



INSYST

Journal of Intelligent System and Computation

p-ISSN: 2621-9220

e-ISSN: 2722-1962

Volume 1 Nomor 2, Desember 2019



Published By **Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)**
Institut Sains dan Teknologi Terpadu Surabaya (ISTTS)
formerly **Sekolah Tinggi Teknik Surabaya (STTS)**



Managed By
Departement of Informatics
Institut Sains dan Teknologi Terpadu Surabaya (ISTTS)

INSYST

Journal of Intelligent System and Computation

Volume 01 Nomor 02 Desember 2019

Editor in Chief:

Dr. Yosi Kristian, S.Kom, M.Kom.
Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

Managing Editor:

Dr. Esther Irawati Setiawan, S.Kom., M.Kom.
Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

Reddy Alexandro H., S.Kom., M.Kom.
Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

Editorial Board:

Dr. Ir. Endang Setyati, M.T.
Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

Ir. Edwin Pramana, M.App.Sc, Ph.D
Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

Prof. Dr. Ir. Mauridhi Hery Purnomo, M.T.
Institut Sepuluh November, Indonesia

Hindriyanto Dwi Purnomo, Ph.D.
Universitas Kristen Satya Wacana, Salatiga, Indonesia

Hendrawan Armanto, S.Kom., M.Kom.
Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

Dr. Lukman Zaman PCSW, M.Kom.
Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

Dr. Diana Purwitasari, S.Kom., M.Sc.
Institut Sepuluh November, Indonesia

INSYST

Journal of Intelligent System and Computation

Volume 01 Nomor 02 Desember 2019

Reviewer:

Teguh Wahyono, S.Kom., M.Cs.

Universitas Kristen Satya Wacana, Salatiga, Indonesia

Dr. Anang Kukuh Adisusilo, ST, MT.

Universitas Wijaya Kusuma, Surabaya, Indonesia

Dr. I Ketut Eddy Purnama, ST., MT.

Institut Sepuluh November, Indonesia

Prof. Dr. Benny Tjahjono, M.Sc.

Coventry University, United Kingdom

Dr. Ir. Gunawan, M.Kom.

Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

Dr. Umi Laili Yuhana S.Kom., M.Sc.

Institut Sepuluh November, Indonesia

Dr. Tita Karlita, S.Kom., M.Kom.

Politeknik Elektronika Negeri Surabaya, Indonesia

Dr. Ir. Rika Rokhana, M.T.

Politeknik Elektronika Negeri Surabaya, Indonesia

Dr. I Made Gede Sunarya, S.Kom., M.Cs.

Universitas Pendidikan Ganesha, Indonesia

Dr. Yuni Yamasari, S.Kom., M.Kom.

Universitas Negeri Surabaya, Indonesia

Dr. Adri Gabriel Sooai, S.T., M.T.

Universitas Katolik Widya Mandira, Indonesia

INSYST

Journal of Intelligent System and Computation

Volume 01 Nomor 02 Desember 2019

Daftar Isi

Aspect Based Sentimen Analysis Opini Publik Pada Instagram dengan Convolutional Neural Network Muhammad Arief Rahman, Herman Budiarto, Esther Irawati Setiawan	50
Peringkasan Teks Ekstraktif pada Dokumen Tunggal Menggunakan Metode Restricted Boltzmann Machine Rully Widiastutik, Lukman Zaman P. C. S. W, Joan Santoso	58
Stance Classification Post Kesehatan di Media Sosial Dengan FastText Embedding dan Deep Learning Ernest Lim, Esther Irawati Setiawan, Joan Santoso	65
Pemanfaatan Asynchronous Advantage Actor-Critic Dalam Pembuatan AI Game Bot Pada Game Arcade Evan Kusuma Susanto, Yosi Kristian	74
Identifikasi Foto Wanita Berhijab dari Majalah Untuk Pembuatan Katalog Busana Muslim Otomatis Memanfaatkan Convolutional Neural Network M. Najamudin Ridha, Endang Setyati, Yosi Kristian	85
Identifikasi Biji-Bijian Berdasarkan Ekstraksi Fitur Warna, Bentuk dan Tekstur Menggunakan Random Forest Luthfi Alwi, Arya Tandy Hermawan, Yosi Kristian	92
Perencanaan Perjalanan Wisata Multi Kota dan Negara dengan Algoritma Cuttlefish Hendrawan Armanto, Reynold Kevin, Pickerling Pickerling	99

Aspect Based Sentimen Analysis Opini Publik Pada Instagram dengan Convolutional Neural Network

Muhammad Arief Rahman, *Program Studi Teknologi Informasi, Herman Budianto, Teknologi Informasi Institut Sains dan Teknologi Terpadu Surabaya*, dan Esther Irawati Setiawan, *Teknologi Informasi Institut Sains dan Teknologi Terpadu Surabaya*

Abstrak - Internet sebagai sarana informasi dan komunikasi sudah sangat dikenal di kalangan masyarakat dalam menawarkan kemudahan dan fleksibilitas yang cukup memadai ketika menjadi media. Oleh karena itu opini publik terhadap Operator Telekomunikasi merupakan hal yang sangat penting untuk dijadikan patokan. Kadang-kadang ketika menganalisis ulasan online yang berkembang pesat ini, menjadi sulit untuk mengkategorikan apakah opini pelanggan puas atau tidak puas terhadap produk dan layanan. Selain itu, sebagai bagian dari peningkatan kualitas mereka, organisasi seperti jasa ini perlu mengklasifikasikan aspek produk dan layanan yang paling disukai pelanggan. *Deep Learning* adalah area baru dalam penelitian *Machine Learning*, yang telah diperkenalkan dengan tujuan menggerakkan. *Convolutional Neural Network* adalah salah satu contoh metode *Deep Learning*. Metode *Convolutional Neural Network* diharapkan dapat digunakan dalam pengimplementasian opini publik untuk keperluan data training yang dikumpulkan dari beragam data yang dianalisis kelas sentimennya secara otomatis berdasarkan aspeknya masing-masing. Hasil dari penelitian menunjukkan dari 4 aspek dan 3 sentimen maka didapatkan nilai rata-rata *precision*, *recall*, dan *f1-score* adalah *precision* – 95.80%, *recall*- 88.12%, *f1_score*- 91.62%. Bisa disimpulkan *score representation* ini dapat digunakan untuk klasifikasi sentimen.

Kata kunci : *Convolutional Neural Network*, Analisis Sentimen, *Deep Learning*, *Social Network Analysis*

I. PENDAHULUAN

Internet sebagai sarana informasi dan komunikasi sudah sangat dikenal di kalangan masyarakat dalam menawarkan kemudahan dan fleksibilitas yang cukup memadai ketika menjadi media. Salah satu contohnya adalah Instagram. Instagram adalah sebuah aplikasi berbagi foto dan video yang memungkinkan pengguna mengambil foto, mengambil video, menerapkan filter digital, dan membagikannya ke berbagai layanan jejaring sosial, termasuk milik Instagram sendiri. Toko online biasanya

menyediakan fasilitas untuk memberikan ulasan yang dapat diakses oleh pengunjung. Ulasan berisi deskripsi singkat tentang kesaksian dari pengguna sebelumnya tentang layanan atau barang yang disediakan oleh penjual[1]. *Social Network Services* telah dikembangkan dalam berbagai bentuk[2][3]. Dengan pesatnya perkembangan teknologi ini, semakin banyak orang menuliskan opini mereka tentang produk atau jasa. Hal ini dimanfaatkan sebagai acuan para customer untuk mengetahui pendapat orang lain tentang produk atau jasa yang akan digunakan. Sebaliknya bagi produsen dapat digunakan sebagai media melacak kepuasan pelanggan. Instagram juga dapat digunakan sebagai alat untuk membangun reputasi digital secara global[4].

Operator telekomunikasi fixed-line dan operator multimedia di Indonesia yang memberikan layanan jaringan (*network*), layanan Internet, pusat data, serta layanan hosting dan *cloud computing*. Sebagai operator telekomunikasi, mengetahui kepuasan pelanggan merupakan hal yang sangat penting dikarenakan dapat menjadi patokan untuk perbaikan pada layanan yang diberikan oleh perusahaan. Dengan begitu kualitas layanan yang diberikan kepada masyarakat terutama kepada pelanggan Provider Internet sendiri dapat meningkat. Provider Internet telah memiliki cabang di 101 kota yang tersebar di Indonesia. Oleh karena itu opini publik terhadap Provider Internet merupakan hal yang sangat penting untuk dijadikan patokan.

Untuk menangkap sentimen kata yang digunakan penulis algoritma pembelajaran yang diawasi berdasarkan kesamaan antara kata-kata yang memperhitungkan peringkat ulasan sebelumnya untuk menangkap representasi kata vektor[5]. Analisis sentimen adalah pendekatan yang digunakan Natural Language Processing (NLP) untuk mengekstraksi, mengonversi, dan menafsirkan opini dari sebuah teks dan mengklasifikasikannya menjadi Sentimen positif, negatif atau netral[6]. Pebisnis selalu ingin tahu bagaimana opini public atau pelanggan mereka tentang produk dan layanan yang mereka berikan. Pelanggan berpotensi juga ingin mengetahui pendapat pengguna pelanggan sebelumnya sebelum mereka menggunakan layanan atau membeli produk[7]. Namun, untuk mengevaluasi umpan balik online itu, bukan masalah sederhana. Kadang-kadang ketika menganalisis ulasan online yang berkembang pesat ini, menjadi sulit untuk mengkategorikan apakah opini pelanggan puas atau tidak puas terhadap produk dan layanan. Selain itu, sebagai bagian dari peningkatan kualitas mereka, organisasi seperti jasa ini perlu mengklasifikasikan aspek produk dan layanan yang paling disukai pelanggan. Dan karena nilai informasi ini, terutama untuk menjaga kepercayaan dan retensi pelanggan, semuanya harus

Muhammad Arief Rahman, Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: rahman.malang@gmail.com)

Herman Budianto, Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Jawa Timur, Indonesia (e-mail: hb@stts.edu)

Esther Irawati Setiawan, Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Jawa Timur, Indonesia (e-mail: esther@stts.edu)

dilakukan tepat waktu. Masalahnya adalah sekitar 85% teks yang tersedia di internet memiliki format yang tidak terstruktur, sehingga perlu mengembangkan sistem yang secara otomatis dapat mengklasifikasikan aspek dan sentimen dari data teks online[8].

Deep Learning adalah area baru dalam penelitian *Machine Learning*, yang telah diperkenalkan dengan tujuan menggerakkan *Machine Learning* lebih dekat dengan salah satu tujuan aslinya yaitu *Artificial Intelligence*. *Deep Learning* adalah tentang belajar beberapa tingkat representasi dan abstraksi yang membantu untuk memahami data seperti gambar, suara, dan teks. *Convolutional Neural Network* adalah salah satu contoh metode *Deep Learning*. Berdasarkan penelitian dari Yoon Kim[9] bahwa klasifikasi kalimat menggunakan metode *Convolutional Neural Network* menghasilkan sesuatu yang baik sekali

Metode *Convolutional Neural Network* diharapkan dapat digunakan dalam pengimplementasian opini publik untuk keperluan data training yang dikumpulkan dari beragam data yang dianotasi kelas sentimennya secara otomatis.

II. TINJAUAN PUSTAKA

A. Penggunaan Word Embedding

Pada paper [10] penulis melakukan klasifikasi teks dengan menggunakan word embedding lalu dikombinasikan dengan image embedding agar dapat melakukan tugas baru.. Ada beberapa definisi untuk apa Word Embeddings itu, tetapi dalam pengertian yang paling umum, Word Embeddings adalah representasi numerik dari kata-kata.

Model Word Embeddings dapat dibagi menjadi beberapa kategori utama :

- Metode berbasis hitungan
- Metode prediktif

Secara algoritmik, model-model ini serupa, kecuali bahwa CBOW memprediksi kata-kata target (misalnya 'mat') dari kata-kata konteks sumber ('kucing duduk di'), sedangkan skip-gram melakukan kebalikan dan memprediksi kata konteks-sumber dari target. kata-kata. Dalam model Skip-Gram, jaringan saraf dilatih melalui kumpulan besar di mana tujuan pelatihannya adalah untuk belajar perwakilan kata-kata yang baik dalam memprediksi kata-kata yang dekat. Sifat paling sederhana dari embeddings yang diperoleh dengan semua metode yang dijelaskan di atas adalah bahwa kata-kata yang mirip cenderung memiliki vektor yang sama.

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Gambar 1 Hubungan berpasangan dalam embedding kata[11]

Penulis membahas masalah bahwa model berbasis vektor biasanya diarahkan untuk mewakili kata-kata dalam isolasi dan metode untuk membangun representasi untuk frasa atau

kalimat telah menerima sedikit perhatian dalam literatur. Para penulis menyarankan penggunaan dua operasi komposisi, pengandaan dan penambahan (dan kombinasinya). Dengan cara ini penulis dapat menggabungkan embeddings kata ke frasa atau embeddings kalimat sambil mempertimbangkan properti penting seperti urutan kata dan hubungan semantik antara kata-kata (yaitu jenis komposisi semantik).

B. Penggunaan CNN dengan memanfaatkan emoji

Pada paper [12] penulis menggunakan emoji sebagai bahan acuan lain untuk mendeteksi sentiment tweet dengan metode *Convolutional Neural Network*. ketika menganalisis tweet dengan emoji yang seharusnya menyatakan senang atau marah, kami menemukan bahwa ada sekitar 20 hingga 40% tweets di mana sentimen teks dan emoji tidak konsisten. Mempertimbangkan hasil analisis ini, kami mengasumsikan bahwa sentimen tweet konsisten dengan

emoji kecuali tweet dengan 😊, di mana sentimen emoji 40% tidak konsisten dengan teks. Pada bagian kedua dari makalah ini, tweet dengan emoji selain 😊, yang seharusnya konsisten dengan sentimen teks tweet dengan probabilitas lebih dari 60%, serta tweet yang secara manual dinilai netral, digunakan sebagai pelatihan dan pengujian data model CNN. Pada bagian terakhir, tweet yang mengikuti distribusi sentimen dalam aliran tweet nyata digunakan sebagai data pengujian. Kami membandingkan beberapa variasi penyusunan tanggal pelatihan model CNN, dan mendiskusikan data pelatihan mana dan mana model modalnya yang optimal untuk mengidentifikasi sentimen tweets setelah distribusi sentimen dalam aliran tweets nyata.

Emoji seharusnya menjadi suplemen untuk mengekspresikan sentimen, tetapi survei awal penulis tentang emoji pada tweet Jepang mengungkapkan beberapa kasus yang menarik di mana emoji mengekspresikan sentimen yang tidak konsisten. Untuk menganalisis lebih lanjut hubungan emoji-sentimen, penulis mengumpulkan tweet Jepang dengan emoji oleh Twitter Streaming API1 dari 3 November 2016 hingga 23 Juni 2017. Lalu penulis mencicipi tweet dengan hanya satu emoji (😄, 😊, 😏 atau 😡) dan menemukan tweet dengan 😊 lebih dari yang lain.

Oleh karena itu untuk tweet dengan hanya satu emoji 😊, penulis menilai sentimen tweet secara manual hingga 100 tweet yang mana teks dan emoji mengekspresikan sentimen konsisten ditemukan. Untuk setiap emoji 😊, 😏 atau 😡, penulis menilai sentimen secara manual untuk setiap kelas hingga 50 tweet yang mana teks dan emoji yang mengekspresikan sentimen konsisten ditemukan.

Seperti disebutkan pada tabel 1, tweet dengan 😊 mengandung persentase tweet tertinggi dimana sentimen teks dan emoji tidak konsisten. Jadi, untuk mendapatkan hasil yang lebih baik, penulis memutuskan untuk tidak menggunakan bagian tweet ini. (Meskipun tweet dengan 😏 konten persentase tweet yang relatif tinggi di mana sentimen teks dan emoji tidak konsisten, karena ada jauh lebih sedikit tweet yang marah daripada bahagia tweet, penulis masih memutuskan untuk menggunakan bagian tweet ini). Secara khusus, 3.000 tweet dengan 😊, 3.000 tweet dengan 😏 atau 😡 dipilih secara acak dari koleksi. Selain itu, penulis secara acak memilih 3.000 tweet netral dengan analisis

manual. Berdasarkan hasil bahwa hanya sebagian kecil tweet yang mengekspresikan sentimen yang berlawanan dengan emoji, penulis menganggap bahwa sentimen tweet kira-kira konsisten dengan emoji. Setiap tweet diproses untuk menghilangkan URL, nama pengguna (mis. @Xxx) dan emoji. Selanjutnya, tweet ini diterapkan ke Mecab4 untuk dikonversi menjadi persamaan morfem. Kemudian penulis melakukan validasi silang 10 kali lipat pada kedua CNN dan SVM menggunakan 9.000 tweet yang telah diproses sebelumnya.

TABEL 1.
KLASIFIKASI SENTIMENT DARI TWEET DENGAN EMOJI[12].

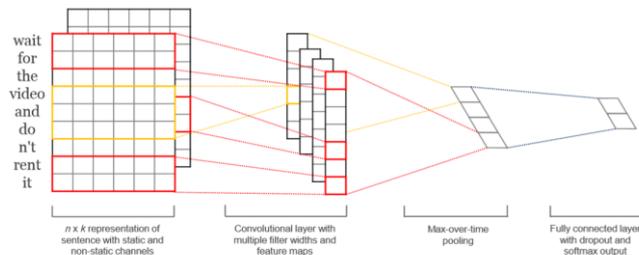
Emoji	Number Of Tweets	Sentiments of emoji and text			
		Collect ed	For manual Analysis	Consist ent	Inconsiste nt
Happy		332573	120	100	20
		37576	91	50	41
Angry		13497	85	50	35
		25161	70	50	20

Untuk keperluan pengujian pada data yang mengikuti distribusi sentimen nyata, penulis secara acak memilih 1.000 Tweet dari 19 Januari 2017 hingga 22 Januari 2017 dan menilai sentimen secara manual. Akibatnya, tweet ini mencakup 180 tweet positif, 45 tweet negatif dan 775 tweet netral. Tweet netral memiliki jumlah yang luar biasa dan tweet negatif kurang dari lima persen.

C. Klasifikasi Kalimat dengan CNN

Pada paper [9] peneliti menggunakan metode Convolutional Neural Network untuk mengklasifikasi sebuah kalimat dan dibandingkan dengan 14 metode lain yaitu RAE : Recursive Autoencoders, MV-RNN: Matrix-Vector Recursive Neural Network, RNTN : Recursive Neural Tensor Network, DCNN : Dynamic Convolutional Neural Network, Paragraph-Vec, CCAE: Combinatorial Category Autoencoders, Sent-Parser: Sentiment analysis-specific parser, NBSVM, MNB: Naive Bayes SVM and Multinomial Naive Bayes, G-Dropout, F-Dropout: Gaussian Dropout and Fast Dropout, Tree-CRF: Dependency tree with Conditional Random Fields, CRF-PR : Conditional Random Fields with Posterior , SVMS : SVM with uni-bi-grams.

Arsitektur model, yang ditunjukkan pada Gambar 1, adalah sedikit variasi dari Arsitektur NNCNCollobert et al. (2011). Misalkan $x_i \in R_k$ menjadi vektor kata k-dimensional yang bersesuaian dengan kata ke-i dalam kalimat.



Gambar 2 Arsitektur model dengan dua saluran untuk contoh kalimat[9]

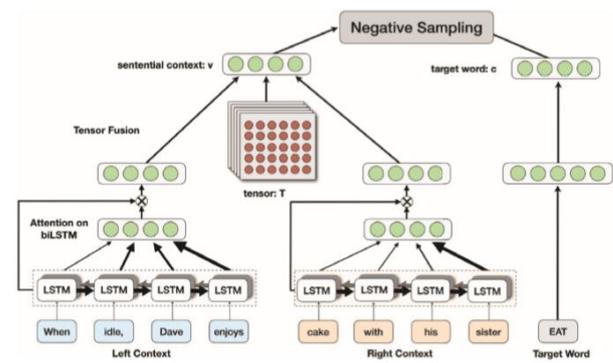
D. Penggunaan Senticnet

Pada paper [13] penulis menggunakan long shot term memory (LSTM) untuk menemukan kata kerja dan kata benda dalam sebuah kalimat menggunakan lexical substitusi

dan membuat 3 level konsep representasi pengetahuan untuk sentiment analisis, produk ini dinamakan dengan SenticNet 5. SenticNet 5 mengkode denotatif dan informasi konotatif yang umumnya dikaitkan dengan objek, tindakan, peristiwa, dan orang di dunia nyata. Ini akan keluar dari penggunaan kata kunci dan kata kunci yang terjadi secara kebetulan, dan bukannya bergantung pada makna tersirat yang terkait dengan konsep akal sehat. Lebih unggul daripada teknik sintaksis murni, SenticNet 5 dapat dengan mudah diekspresikan dengan menggunakan analisis analisis multi-kata yang tidak secara eksplisit menyampaikan emosi, tetapi lebih terkait dengan konsep yang melakukannya.

Sebuah kalimat S dapat direpresentasikan sebagai urutan kata, yaitu, $S = [w_1, w_2, \dots, w_n]$ di mana n adalah jumlah kata dalam kalimat. Kalimat tersebut dapat dibagi menjadi beberapa bagian sehingga prefiks: $[w_1, \dots, w_i - 1]$ membentuk kalimat konteks kiri dengan l kata-kata dan sufiks: $[w_i + 1, \dots, w_n]$ membentuk kalimat konteks yang tepat dengan kata-kata. Di sini, $c = w_i$ adalah kata target. Pada langkah pertama, penulis merepresentasikan kata-kata ini dalam representasi berdimensi rendah yang dikenal sebagai embeddings kata. Khususnya, kami menggunakan embedded word2vec 300 dimensi yang dilatih pada 3 miliar kata Google News corpus. Untuk mengekstrak fitur kontekstual dari subsensi ini, penulis menggunakan model biLSTM pada L dan C secara independent. Ketika biLSTM digunakan, operasi ini diterapkan di kedua arah urutan dan output untuk setiap catatan waktu digabung untuk membentuk representasi keseluruhan di luar kata kunci. Jadi, untuk setiap sensor mata, setelah menerapkan biLSTM, kita mendapatkan matriks fitur representasi berulang.

Model penulis hadir untuk sub-frasa yang penting dalam menyediakan konteks, penulis memasukkan modul perhatian di atas biLSTM untuk kalimat konteks. Modul perhatian terdiri dari jaringan saraf tambahan yang memiliki lapisan tersembunyi diikuti oleh output softmax (Gambar. 1). Ini menghasilkan vektor yang memberikan bobot yang sesuai dengan tingkat relevansi teks konteks di seluruh kalimat. Di bawah ini, kami menggambarkan rumusan perhatian yang diterapkan pada kalimat konteks.



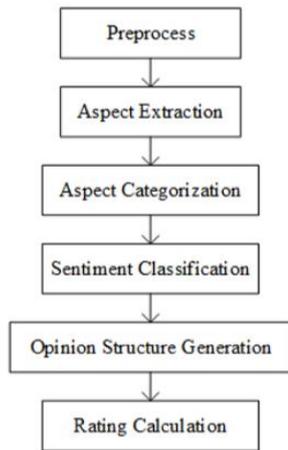
Gambar 3 Framework keseluruhan untuk context dan word embedding generation [13].

Penulis menguji SenticNet 5 (tersedia baik sebagai repositori2 mandiri XML dan sebagai API3) terhadap dua sumber daya sentimen terkenal: Blitzer Dataset (Blitzer, Dredze, dan Pereira 2007) yang kemudian dimodifikasi oleh (Poria et al. 2015) dan Movie Review Dataset dikembangkan (Pang dan Lee 2005). Blitzer Dataset terdiri dari ulasan produk di tujuh domain berbeda. Untuk setiap

domain ada 1.000 ulasan positif dan 1.000 ulasan negatif. Kami memperoleh dataset ini, yang mengandung 3.800 kalimat positif dan 3.410 negatif, dari penulis (Poriaetal.2015). Sebaliknya, Movie Review Dataset telah direstrukturisasi dari dokumen ke level kalimat oleh Socheretal. (Socher et al. 2013c) dan berisi 4.800 kalimat positif dan 4.813 yang negative.

E. Analisa Sentimen berbasis aspek untuk ulasan restoran.

Pada paper [14] penulis melakukan analisis sentiment berdasarkan struktur yang terdiri dari aspek, kategori aspek, dan polaritas sentiment. Ada enam langkah dalam analisis sentimen berbasis aspek: ekstraksi entitas dan kategorisasi, ekstraksi aspek dan kategorisasi, klasifikasi polaritas sentimen, ekstraksi waktu, ekstraksi pemegang opini, dan pembentukan struktur opini.



Gambar 4 Arsitektur Aplikasi[14]

Penulis menerapkan algoritma yang sama seperti dalam kategorisasi aspek untuk klasifikasi sentimen. penulis mendefinisikan dua label sentimen untuk setiap kategori dalam kalimat: positif dan negatif. Misalnya, sebuah kalimat memiliki kategori makanan dan tempat. Setiap kategori memiliki classifier sendiri untuk mengklasifikasikan sentimen kategori dalam kalimat. Untuk mengklasifikasikan polaritas sentimen, penulis menggunakan classifier untuk kategori makanan dan tempat. Untuk pembuatan struktur opini, penulis menggunakan model CBOW untuk menemukan kesamaan antara aspek yang diekstraksi dengan kata-kata awal untuk setiap kategori. Kategori yang memiliki skor kesamaan maksimum akan dipasangkan dengan aspek yang diekstraksi. Misalnya, sebuah kalimat memiliki kategori makanan dan tempat dan penulis ingin memasang "kue" dengan kategori dari kalimat tersebut. Pertama, penulis menemukan skor kesamaan untuk setiap kata benih untuk kategori makanan dan tempat. Skor kesamaan maksimum untuk kategori makanan dan tempat masing-masing adalah 0,2 dan 0,1. Setelah itu, kita bisa memasang "kue" dengan kategori makanan karena skor kesamaan untuk kategori makanan lebih tinggi dari kategori tempat.

Label	Total Sentence
Food	503
Service	97
Price	125
Place	440

Gambar 5 Label untuk kategori aspek dalam data training[14]

Sentence	Food	Service	Price	Place
Saya suka makanannya (I like the food)	Yes	No	No	No
Tempat tetap nyaman, masakan tetap lezat, dan pelayanan tetap ramah (Place is still comfortable, cuisine is still delicious, and service is still friendly)	Yes	Yes	No	Yes

Gambar 6 Contoh training data untuk kategorisasi aspek[14]

Model ekstraksi aspek penulis dapat mengekstraksi beberapa aspek yang tidak muncul dalam data pelatihan, seperti "guramenya" (gurame) dan "jus strawberry" (jus strawberry). Karena model ini menggunakan fitur bag of N-gram, klasifikasi token sangat tergantung dengan kemunculan kata, tag POS, dan cluster CBOW di jendela konteks. Salah satu contoh kesalahan klasifikasi terjadi dalam kalimat "baik rasa dan tempat yang nyaman untuk bersantai" (Baik dalam rasa dan tempat yang nyaman untuk bersantai). Aspek sebenarnya untuk kalimat adalah "rasa" (taste) dan "tempat" (place). Tetapi hasil ekstraksi adalah "tempat". Model gagal untuk mengekstrak "rasa" (rasa) sebagai aspek karena "rasa" (rasa) biasanya muncul bersamaan dengan kata-kata yang berkaitan dengan makanan, seperti "enak" (lezat) dan "makanan" (makanan) dalam data pelatihan. Tetapi kalimat itu tidak memiliki kata-kata yang berhubungan dengan makanan sehingga kesalahan klasifikasi terjadi. Kesalahan klasifikasi lain terjadi pada kalimat "paling enak nasi gorengnya, sate ayam, steak braga permai, dan chicken schnitzel" (makanan yang paling enak adalah nasi goreng, sate ayam, steak braga permai, dan schnitzel ayam). Aspek yang diharapkan dari kalimat itu adalah "nasi gorengnya" (nasi goreng), "sate ayam", "steak braga permai" (steak braga permai), dan "schnitzel ayam" (chicken schnitzel). Tetapi hasil dari ekstraksi aspek adalah "nasi gorengnya". "Sate ayam" (sate ayam) dan "chicken schnitzel" (chicken schnitzel) tidak diekstraksi sebagai aspek karena "sate" (sate) dan "schnitzel" (schnitzel) tidak pernah muncul dalam data pelatihan. Meskipun "steak" dan "braga permai" muncul dalam data pelatihan, "steak" dan "braga permai" tidak pernah muncul secara bersamaan dalam data pelatihan dalam jendela konteks. Selain itu, "braga permai" muncul dalam data pelatihan sebagai nama restoran, bukan bagian dari nama makanan sehingga tidak pernah dilabeli sebagai aspek.

Sentence	Food	Service	Price	Place
Saya suka makanannya (I like the food)	Positive	-	-	-
Tempat tetap nyaman, masakan tetap lezat, dan pelayanan tetap ramah (Place is still comfortable, cuisine is still delicious, and service is still friendly)	Positive	Positive	-	Positive

Gambar 7 Contoh data training untuk klasifikasi sentiment[14]

III. PERANCANGAN

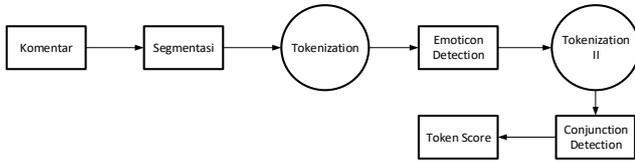
Pada pokok bahasan ini akan dibahas mengenai desain arsitektur dari system, alternative algoritma yang akan digunakan.

a. Pre Processing

Pada proses ini adalah proses yang di lakukan sebelum data siap di import pada sistem, proses ini dilakukan secara manual pada aplikasi pengolahan data office.

- Abbreviation Replacement

Proses ini adalah perubahan singkatan yang dirubah menjadi sebuah teks, singkatan atau kata alay akan dirubah menjadi kata Bahasa Indonesia yang baik oleh tim social media analis.



Gambar 8 Tahap pre-processing.

- Segmentasi

Titik adalah satu-satunya tanda baca dianggap sebagai pemisah pada langkah ini, seperti koma atau titik koma dapat menjadi bagian dari emoticon.

- Tokenization

Pada langkah berikutnya, token yang diambil dari setiap kalimat. Pada kalimat ini hanya spasi putih dipertimbangkan untuk token terpisah, sejak pemisah lainnya, seperti tanda hubung, dapat menjadi bagian dari emoticon.

- Deteksi emoticon

Pencarian classifier dalam teks semua emoticon yang disimpan dalam dua file teks, yang berisi emoticon positif dan negatif.

- Tokenization II

fase tokenization kedua ini, daftar kata hubung (dan, tapi, dll) dianggap sebagai pemisah terkemuka untuk mendapatkan set akhir token untuk setiap kalimat

- Deteksi Conjunction

Langkah selanjutnya terdiri dari mendeteksi dan label kata hubung dengan analisa struktur kalimat dengan daftar kata hubung (dan, tapi, dll). Dalam hal ini, dapat ditemukan komentar tersebut adalah kalimat majemuk atau tidak. Seperti “Biznet kecepatannya bagus sih, **tapi** dari 2 minggu complain belum ditanggapin”

- Token Skor

Tahap berikutnya terdiri dari menetapkan nilai untuk setiap token, 1 jika mengirimkan sentimen positif, 0 jika netral, dan -1 jika negatif.

b. Arsitektur Sistem

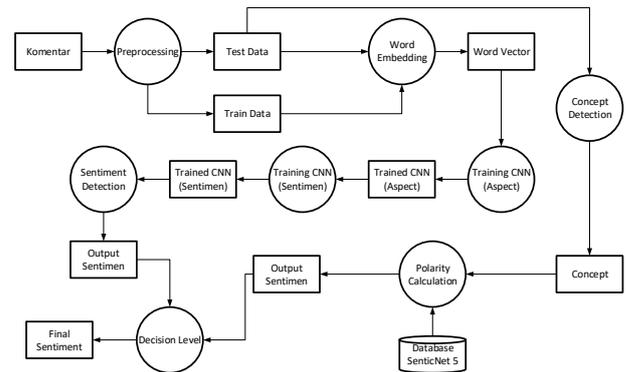
Proses training teks menggunakan metode convolutional neural network untuk ekstraksi fitur yang terdapat pada teks. Proses dilakukan dengan menggunakan dot product dari vector bobot atau kernel vector untuk setiap kata pada kalimat. Setiap kata tersebut direpresentasikan dengan word embedding, untuk dijadikan input pada convolutional neural network. Hasil dari convolutional neural network adalah 3 neuron yaitu positif, negative, dan netral.

Untuk proses pada teks menggunakan SenticNet 5, dilakukan ekstraksi konsep yang akan didapatkan nilai polarity berdasarkan konsep yang terdapat pada kalimat. Setiap konsep akan dilakukan pengecekan pada database SenticNet 5 dan menghasilkan nilai polarity yang digunakan untuk mendeteksi sentiment kalimat tersebut.

c. Proses Word Embedding

Proses ini merupakan teknik untuk merubah kalimat teks menjadi teks vector untuk mewakili makna dari kata

tersebut. Ukuran vector bervariasi dari 50 – 300 dimensional vector. Nilai ini nantinya akan digunakan sebagai input pada proses training convolutional neural network. Setiap kalimat teks akan dihitung sesuai dengan kalimat terpanjang yang dimiliki oleh dataset. Kemudian apabila kalimat tersebut lebih pendek dari panjang kalimat terpanjang, maka sisa untuk melengkapi hingga akhir kalimat akan dinilai 0 atau diberi kata random yang tidak dapat dimengerti agar menjadi nilai 0. Atau yang dikenal dengan istilah padding, dengan menghitung setiap kalimat dengan panjang vector yang sama dengan menambahkan 0 di depan atau belakang kalimat. Untuk merubah kalimat menjadi vector, penulis menggunakan GloVe (Global Vector). Berikut adalah contoh dari word vector.



Gambar 9 Arsitektur Sistem.

	Dimensions					
Word vectors	dog	-0.4	0.37	0.02	-0.34	animal
	cat	-0.15	-0.02	-0.23	-0.23	domesticated
	lion	0.19	-0.4	0.35	-0.48	pet
	tiger	-0.08	0.31	0.56	0.07	fluffy
	elephant	-0.04	-0.09	0.11	-0.06	
	cheetah	0.27	-0.28	-0.2	-0.43	
	monkey	-0.02	-0.67	-0.21	-0.48	
	rabbit	-0.04	-0.3	-0.18	-0.47	
	mouse	0.09	-0.46	-0.35	-0.24	
	rat	0.21	-0.48	-0.56	-0.37	

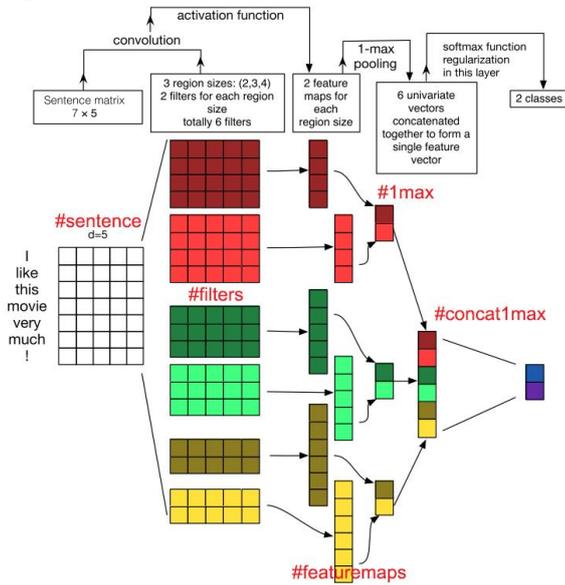
Gambar 10. Word Vector[15]

Setiap word vector memiliki nilai sebanyak dimensi yang telah ditentukan, nilai dari word vector suatu kata bias jadi memiliki kesamaan dengan kata yang sejenisnya. Contohnya adalah *Lion* dan *Cheetah* memiliki nilai yang tidak jauh pada dimensi *animal*.

d. Proses Training CNN

Pada proses ini metode Convolutional neural network akan digunakan untuk melakukan ekstraksi fitur yang terdapat pada teks. Proses yang terjadi pada convolutional layer adalah melakukan dot product dari vector dengan bobot atau kernel vector setiap kata pada kalimat untuk mendapatkan fitur yang lain. Seperti halnya tadi, setiap kata dalam kalimat direpresentasikan dengan word embedding, untuk dapat dijadikan input nilai pada convolutional neural network. Pada tahap awal, CNN akan training data untuk mendeteksi aspek dari kalimat. Sebagai bentuk sebuah analisa yang nantinya diharapkan mampu mempermudah pelanggan atau calon pelanggan untuk mendapatkan informasi ini. Seperti Layanan, Kecepatan, Coverage, dan Promosi Produk Biznet Home. Penelitian ini tahap klasifikasi awal adalah mengelompokan data berdasarkan

kelas-kelas yang disebutkan diatas. Termasuk pelabelan data training yang juga dikelompokkan kedalam kelas tersebut. Pada tahap pertama klasifikasi yang harus dilakukan pada aplikasi adalah klasifikasi text terhadap aspek dimana data yang baru saja diperoleh di proses dengan data yang sudah disimpan sebagai data training. Kemudian akan dilakukan training CNN untuk mendeteksi sentiment dari tiap aspek dengan menghasilkan 3 class yaitu Positif, Netral, dan Negatif.



Gambar 11 Alur Kerja CNN[16]

Arsitektur dari CNN yang akan digunakan yaitu dengan 1 convolutional layer dengan ukuran 2x2 dengan 100 kernel. Setiap convolutional layer diikuti dengan max-pool layer dengan ukuran 2 x 2, lalu terdapat fully connected layer dengan 500 neuron dan diakhiri dengan softmax layer dengan 4 neuron untuk mewakili class aspek (Layanan, Kecepatan, Coverage, dan Promosi) dan 3 neuron untuk mewakili class sentiment (Positif, Netral, dan Negatif).

IV. PENGUJIAN DAN ANALISIS

A. Dataset

Dalam penelitian ini data diambil dari opini publik pada instagram pada akun @biznethome, menggunakan teknik crawling yang dilakukan oleh tim social media analis, jumlah data yang didapat sebanyak 1980, data yang dikumpulkan berupa text yang nantinya akan dilakukan pelabelan manual oleh tim social media analis. Data komentar yang dikumpulkan tidak berasal dari tulisan penulis. Setiap komentar akan diberikan label yang mewakili komentar tersebut. Label terdiri dari 3 jenis, yaitu positif, netral, dan negatif. Contoh data yang sudah terkumpul dapat dilihat pada Tabel 2.

TABEL 2
DISTRIBUSI KELAS PADA DATASET

Aspect	Count	Pos	Netral	Neg
Layanan	692	29	575	88
Kecepatan	122	4	45	73
Coveragae	471	5	454	12
Promosi	695	180	512	3

Contoh 5 sample data yang diperoleh dapat dilihat pada Tabel 3.

Tabel 3
SAMPLE DATA YANG DIPEROLEH

Komentar	Aspek	Sentimen Analisis
Down lagi yaaaaa ??? Hmmm sudah kuduga	Kecepatan	Negatif
BIZNET mau complaining internet mati kemana? Saya posisi di ruko harco mangga dua jalan mangga dua raya Jakarta Pusat 10730	Layanan	Netral
ID 20060219 tolong cek, dari tadi lemot parah	Kecepatan	Netral
Untuk Kecamatan Lubuk Begalung sudah ada coverannya ?	Coverage	Netral
kapan nih biznet bisa ada di lampung nunggu banget nih	Coverage	Positif

B. Pengujian

Pengujian dilakukan pada Google Collaboratory.

i. Preprocessing

Pada tahapan ini dilakukan ekstraksi kata dari data raw yang diperoleh menggunakan librari rake_nltk sehingga didapat kata serapan dari setiap komentar, sehingga data siap untuk dilakukan ke pengujian tahap berikutnya. Hasil penggunaan librari tokenize nltk dapat dilihat pada gambar 12.

Fitted tokenizer on 3164 documents
40000 words in dictionary
Top 3 most common words are: [('kasih', 980), ('terima', 976), ('pakebiznet', 812)]

Gambar 12. Hasil penggunaan tokenizer.

Setelah melakukan tokenizer pada kalimat, maka text tersebut dijadikan sequence untuk di proses pada tahapan selanjutnya. Berikut adalah hasil dari text sequence :

"Cimahi utara admin?" is converted into [30, 4, 59, 137, 1230, 333, 126, 77, 10, 306, 173, 333, 126, 77, 34, 2, 1, 138]

ii. Pengujian Model

- Aspect Model

Kami menggunakan pengaturan yang berbeda dari setiap model. Pada percobaan pertama, kami menggunakan Adagrad optimizer dengan perbandingan learning rate 0.1. Pada percobaan kedua, kami menggunakan Adam optimizer dengan perbandingan learning rate 0.1. Pada percobaan ketiga, kami menggunakan rmsprop optimizer dengan perbandingan learning rate 0.1.

Sebelum melakukan semua eksperimen, kami membandingkan 2 aspect model. Menggunakan Word Embedding (WE) dan model yang lain menggunakan kombinasi Word Embedding dan Convolutional Neural Network (WE+CNN). Hasil perbandingan pada model tersebut dapat dilihat pada table

TABEL 4
DISTRIBUSI KELAS PADA DATASET

Model	Precision	Recall	F1-Score
WE	0.9536	0.6275	0.7568
WE+CNN	0.9585	0.8069	0.8759

- Intell.*, vol. 4, no. 2, p. 169, 2018, doi: 10.20473/jisebi.4.2.169.
- [2] S. Yu, “40 ~ 50대 사용자 유입을 위한 인스타그램 사용성 개선에 관한 연구 A Study on Usability Improvement of Instagram for Users in their 40s and 50s,” vol. 9, no. 9, pp. 177–182, 2018.
- [3] J. Mander, “GWI Social Summary,” *GWI white Pap.*, pp. 1–9, 2015, [Online]. Available: http://www.globalwebindex.net/hubfs/Reports/GWI_Social_Report_-_Q3_2015_Summary.pdf?t=1453306719920&utm_campaign=Insight+Reports&utm_source=hs_automation&utm_medium=email&utm_content=22078263&_hsenc=p2ANqtz-8p_-_sZV7ZRvXkSeZW3f0npXAeby-hKYHmWQbDg5Zs2IYNqh.
- [4] E. Alshawaf and L. Wen, “Understanding digital reputation on Instagram: A case study of social media mavens,” *Proc. 2nd Eur. Conf. Soc. Media ECSSM 2015*, no. July, pp. 19–27, 2015, [Online]. Available: https://www.researchgate.net/publication/280599140_Understanding_Digital_Reputation_on_Instagram_-_A_Case_Study_of_Social_Media_Mavens?enrichId=rgreq-22fe062f9b4f4f2eefe67fe263c89dbb-XXX&enrichSource=Y292ZXJQYWdlOzI4MDU5OTE0MDtBUzoyNjM4Nzg3ODUzNjgwNjRAMTQzO.
- [5] J. Jong, “Predicting Rating with Sentiment Analysis,” pp. 1–5, 2011.
- [6] B. Agarwal, N. Mittal, P. Bansal, and S. Garg, “Sentiment analysis using common-sense and context information,” *Comput. Intell. Neurosci.*, vol. 2015, 2015, doi: 10.1155/2015/715730.
- [7] U. T. Gürsoy, “Social Media Mining and Sentiment Analysis for Brand Management,” pp. 497–511, 2017.
- [8] C. Fiarni, H. Maharani, and R. Pratama, “Sentiment analysis system for Indonesia online retail shop review using hierarchy Naive Bayes technique,” *2016 4th Int. Conf. Inf. Commun. Technol. ICoICT 2016*, vol. 4, no. c, 2016, doi: 10.1109/ICoICT.2016.7571912.
- [9] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” 2014, doi: 10.3115/v1/D14-1181.
- [10] A. Mandelbaum and A. Shalev, “Word Embeddings and Their Use In Sentence Classification Tasks,” 2016, [Online]. Available: <http://arxiv.org/abs/1610.08229>.
- [11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” pp. 1–12, 2013, doi: 10.1162/153244303322533223.
- [12] H. Cui, Y. Lin, and T. Utsuro, “Sentiment Analysis of Tweets by CNN utilizing Tweets with Emoji as Training Data,” 2018, [Online]. Available: http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/.
- [13] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, “SenticNet 5: Discovering Conceptual Primitives for Sentiment Analysis by Means of Context Embeddings,” in *Association for the Advancement of Artificial Intelligence*, 2018, doi: <http://dx.doi.org/10.1016/j.ijporl.2014.01.009>.
- [14] D. Ekawati and M. L. Khodra, “Aspect-based sentiment analysis for Indonesian restaurant reviews,” *Proc. - 2017 Int. Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA 2017*, 2017, doi: 10.1109/ICAICTA.2017.8090963.
- [15] Jayesh Babu Ahire, “Introduction to Word Vectors.” <https://medium.com/@jayeshbahire/introduction-to-word-vectors-ea1d4e4b84bf> (accessed Jan. 09, 2019).
- [16] L. Li, L. Xiao, N. Wang, G. Yang, and J. Zhang, “Text classification method based on convolution neural network,” *2017 3rd IEEE Int. Conf. Comput. Commun. ICC 2017*, vol. 2018-Janua, pp. 1985–1989, 2018, doi: 10.1109/CompComm.2017.8322884.

Peringkasan Teks Ekstraktif pada Dokumen Tunggal Menggunakan Metode *Restricted Boltzmann Machine*

Rully Widiastutik, *Teknik Informatika AKNS*, Lukman Zaman P.C.S.W., *Teknologi Informasi iSTTS*, dan Joan Santoso, *Teknologi Informasi iSTTS*

Abstrak—Penelitian yang dilakukan yaitu menghasilkan peringkasan teks ekstraktif secara otomatis yang dapat membantu menghasilkan dokumen yang lebih pendek dari dokumen aslinya dengan cara mengambil kalimat penting dari dokumen sehingga pembaca dapat memahami isi dokumen dengan cepat tanpa membaca secara keseluruhan. Dataset yang digunakan sebanyak 30 dokumen tunggal teks berita berbahasa Indonesia yang diperoleh dari www.kompas.com pada kategori tekno. Dalam penelitian ini, digunakan sepuluh fitur yaitu posisi kalimat, panjang kalimat, data numerik, bobot kalimat, kesamaan antara kalimat dan centroid, bi-gram, tri-gram, kata benda yang tepat, kemiripan antar kalimat, huruf besar. Nilai fitur setiap kalimat dihitung. Nilai fitur yang dihasilkan ditingkatkan dengan menggunakan metode *Restricted Boltzmann Machine* (RBM) agar ringkasan yang dihasilkan lebih akurat. Untuk proses pengujian dalam penelitian ini menggunakan ROUGE-1. Hasil yang diperoleh dalam penelitian yaitu dengan menggunakan *learning rate* 0.06 menghasilkan *recall*, *precision* dan *f-measure* tertinggi yakni 0.744, 0.611 dan 0.669. Selain itu, semakin besar nilai *compression rate* yang digunakan maka hasil *recall*, *precision* dan *f-measure* yang dihasilkan akan semakin tinggi. Hasil peringkasan teks dengan menggunakan RBM memiliki nilai *recall* lebih tinggi 2.1%, *precision* lebih tinggi 1.6% dan *f-measure* lebih tinggi 1.8% daripada hasil peringkasan teks tanpa RBM. Hal ini menunjukkan bahwa peringkasan teks dengan menggunakan RBM hasilnya lebih baik daripada peringkasan teks tanpa RBM.

Kata Kunci—Natural Language Processing, Peringkasan Teks, *Restricted Boltzmann Machine*, Teks Berita Berbahasa Indonesia.

I. PENDAHULUAN

Dengan perkembangan teknologi internet menyebabkan jumlah data terus meningkat sehingga menyebabkan informasi yang tersedia semakin banyak. Salah satu informasi yang banyak ditemui di media online adalah artikel berita. Artikel berita merupakan salah satu dokumen berbasis teks. Dengan membaca artikel berita, pembaca dapat mengetahui informasi mengenai sesuatu yang sedang

terjadi atau yang telah terjadi. Pembaca dapat mengalami kesulitan serta membutuhkan waktu yang lama dalam memahami isi dokumen apabila dokumen yang akan dibaca banyak dan panjang. Hal ini dapat menyebabkan berkurangnya minat seseorang dalam membaca. Namun apabila terdapat ringkasan dari sebuah teks atau dokumen akan membantu memahami isi dokumen dengan cepat tanpa harus membaca secara keseluruhan. Ringkasan adalah dokumen yang berisi informasi penting atau intisari dari dokumen asli yang dihasilkan dari satu atau lebih dokumen. Ringkasan dokumen yang dibuat secara manual akan membutuhkan waktu lama apabila dokumen yang akan diringkas banyak dan panjang. Oleh karena itu dalam penelitian ini diterapkan suatu sistem peringkasan teks secara otomatis yang dapat membantu dalam penyusunan kalimat mengenai intisari dari dokumen secara cepat[1].

Pada peringkasan teks terdapat dua pendekatan, yang pertama adalah pendekatan ekstraktif. Pendekatan ini menghasilkan ringkasan dengan cara mengambil kalimat-kalimat penting dari dokumen asli dan menyusunnya menjadi dokumen yang lebih pendek. Yang kedua adalah pendekatan abstraktif. Pendekatan ini menghasilkan ringkasan dimana kalimat yang tersusun tidak ada dalam dokumen aslinya[1]. Terdapat dua kategori peringkasan teks yaitu peringkasan dokumen tunggal, dimana peringkasan diekstrak dari satu dokumen dan peringkasan multi dokumen, dimana peringkasan diekstrak dari beberapa dokumen yang saling berhubungan[2].

II. PENELITIAN PENUNJANG

Bab ini terdapat penjelasan dari beberapa jurnal ilmiah yang digunakan untuk perbandingan serta sebagai penunjang pada penelitian ini.

Penelitian yang dilakukan oleh Suputra (2017) yaitu menghasilkan ringkasan berdasarkan skor fitur-fitur penting dari kalimat pada sebuah dokumen. Fitur yang digunakan sebanyak tiga fitur, pertama adalah fitur *keyword* positif, kedua adalah fitur kemiripan antar kalimat, dan ketiga adalah fitur kemiripan kalimat dengan judul dokumen. Dari ketiga fitur tersebut yang memberikan pengaruh yang dominan adalah fitur kemiripan antar kalimat. Data yang digunakan pada saat percobaan yaitu artikel Bahasa Bali yang didapat dari berbagai sumber[3].

Penelitian yang dilakukan oleh Gotami, dkk (2018) yaitu menghasilkan ringkasan yang memiliki makna umum atau luas dengan menggunakan metode *Latent Semantic Analysis*

Rully Widiastutik, Teknik Informatika, Akademi Komunitas Negeri Sumenep, Sumenep, Jawa Timur, Indonesia (e-mail: rullywidiastutik@gmail.com)

Lukman Zaman P.C.S.W., Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: lz@sts.edu)

Joan Santoso, Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: joan@sts.edu)

(LSA). Digunakannya metode tersebut karena mampu mengekstrak struktur semantik atau makna yang tersembunyi pada sebuah kalimat. Untuk penyusunan urutan ringkasan pada tahap ekstraksi ringkasan menggunakan *Cross method LSA*. Data yang digunakan untuk pengujian yaitu 10 artikel berita kesehatan berbahasa Indonesia[4].

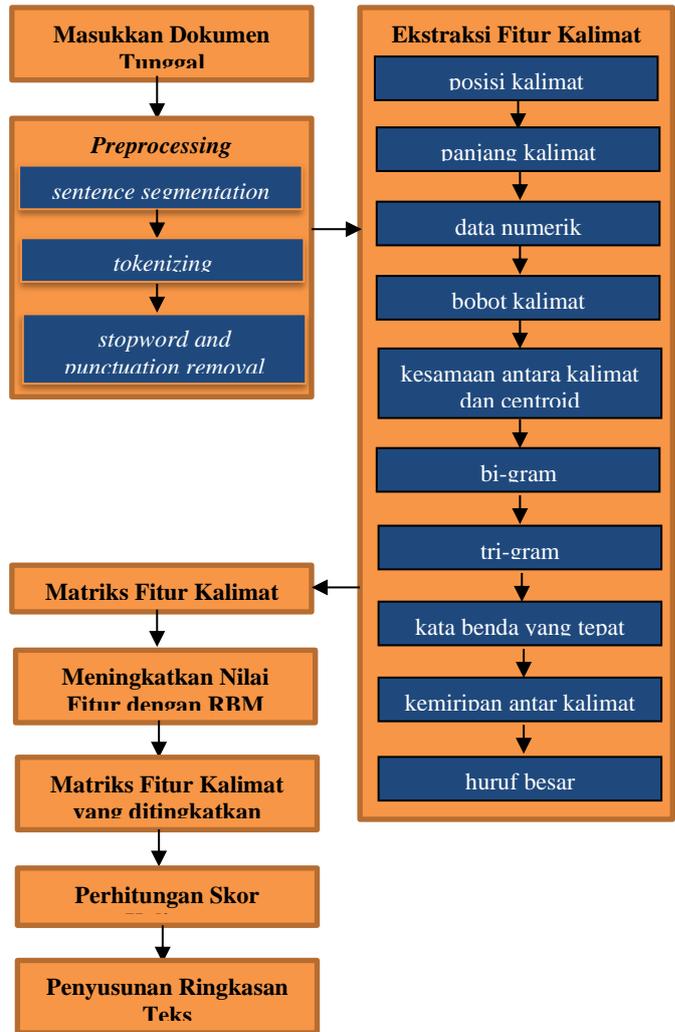
Penelitian yang dilakukan oleh Ambekar, dkk (2018) yaitu menghasilkan peringkasan teks menggunakan *Restricted Boltzmann Machine* (RBM). Terdapat lima fitur yang digunakan yaitu kesamaan judul, posisi kalimat, bobot term, panjang kalimat, skor kata benda yang tepat. Dataset yang digunakan merupakan data input yang tidak berlabel. RBM merupakan algoritma *deep learning* tanpa pengawasan. Dalam penelitiannya menjelaskan bahwa ringkasan dengan menggunakan algoritma *deep learning* tanpa pengawasan hasilnya lebih baik daripada ringkasan dengan menggunakan teknik pembelajaran terawasi karena tidak ada dataset pelatihan yang diperlukan. Sehingga metode tersebut lebih efisien[5].

Penelitian yang dilakukan oleh Elgamel, dkk (2019) yaitu menghasilkan sistem peringkasan teks otomatis pada teks arab. Dalam penelitiannya melakukan perbandingan antara algoritma *deep learning* menggunakan RBM dan algoritma *clustering* menggunakan LSA. Setelah dilakukan pengujian menunjukkan bahwa dengan algoritma *deep learning* menggunakan RBM memberikan hasil yang lebih baik dalam peringkasan teks arab[6].

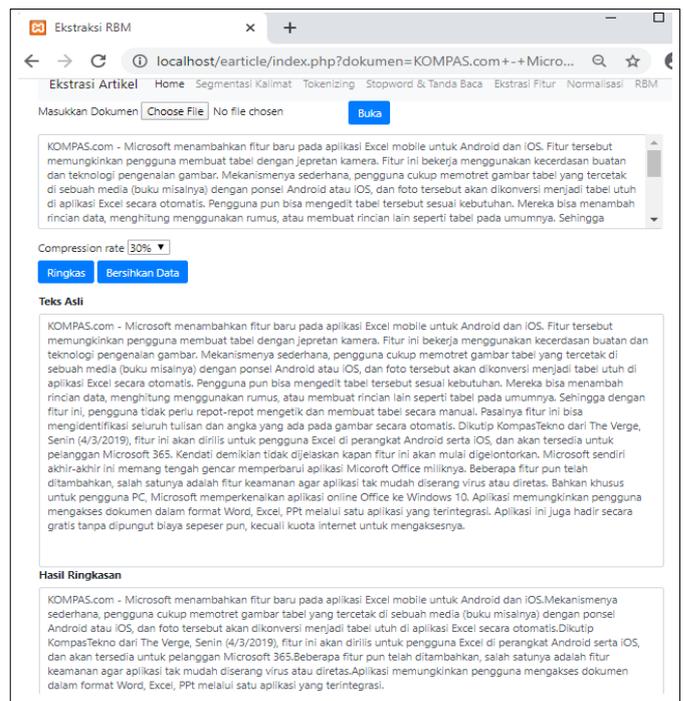
Dari penelitian-penelitian tersebut maka dalam penelitian ini menghasilkan peringkasan teks ekstraktif secara otomatis pada dokumen tunggal menggunakan metode *Restricted Boltzmann Machine* (RBM). Peringkasan teks secara otomatis yang dihasilkan mampu menyusun dokumen dalam bentuk yang lebih pendek dari dokumen asli dengan cara mengambil kalimat penting dokumen asli tersebut. Terdapat sepuluh fitur yang digunakan dalam penelitian ini. Digunakan metode RBM karena dapat memberikan hasil yang lebih baik dalam peringkasan. Dalam penelitian ini, metode RBM digunakan untuk meningkatkan nilai fitur setiap kalimat sehingga ringkasan yang dihasilkan lebih akurat[7].

Gambar 1 merupakan arsitektur sistem dalam penelitian ini. Dokumen teks yang diinputkan dilakukan *preprocessing* untuk menghasilkan *Term Index* yang dapat digunakan pada tahap selanjutnya. Tahap selanjutnya adalah perhitungan nilai fitur setiap kalimat. Terdapat sepuluh fitur yang digunakan. Matriks fitur kalimat dibentuk. Nilai fitur kalimat yang telah dihitung ditingkatkan menggunakan metode RBM. Matriks fitur kalimat yang ditingkatkan dibentuk. Dilakukan perhitungan skor setiap kalimat pada dokumen. Kalimat yang termasuk dalam ringkasan yaitu kalimat yang memiliki urutan teratas berdasarkan skor sesuai *compression rate* yang ditentukan.

Gambar 2 merupakan sistem peringkasan teks ekstraktif pada dokumen artikel berita berbahasa Indonesia menggunakan metode *Restricted Boltzmann Machine*. Pada sistem tersebut user menginput dokumen berupa isi dari artikel berita. Kemudian sistem akan mengolah dokumen tersebut dan akan menampilkan hasil ringkasan sesuai panjang ringkasan yang ditentukan oleh user.



Gambar. 1. Arsitektur Sistem



Gambar. 2. Sistem Peringkasan Dokumen Berita Menggunakan RBM

III. METODOLOGI

A. Preprocessing

Tahap awal pada pemrosesan peringkasan teks yaitu *preprocessing*. Tujuan dilakukan *preprocessing* yaitu untuk menghasilkan *Term Index* dari dokumen teks sehingga dapat digunakan pada tahap selanjutnya[4]. Tahapan *preprocessing* yang digunakan pada penelitian ini meliputi tahapan *sentence segmentation* adalah proses pemisahan kalimat dengan pembatas karakter titik yang diikuti oleh karakter spasi menjadi komponen terpisah. Tahapan selanjutnya adalah *tokenizing*, pada tahapan ini kumpulan kalimat hasil segmentasi dilakukan proses pemisahan setiap kata untuk menjadi data tunggal. Yang terakhir adalah tahapan *stopword and punctuation removal* adalah proses penghapusan kata umum untuk mengurangi jumlah kemunculan kata yang kurang mempunyai arti seperti “dan”, “atau”, “adalah”, “di”, “lalu” dan lain-lain serta proses penghapusan tanda baca kecuali titik karena sebagai pembatas akhir kalimat.

B. Ekstraksi Fitur kalimat

Setelah dilakukan tahap *preprocessing*, tahap selanjutnya adalah perhitungan nilai fitur setiap kalimat. Pada penelitian ini digunakan sebanyak sepuluh fitur yaitu :

1. Fitur Posisi Kalimat

Posisi kalimat adalah letak kalimat dalam sebuah dokumen. Relevansi kalimat dapat diketahui berdasarkan posisinya dalam dokumen. Pada kalimat pertama atau terakhir dari dokumen selalu penting dan memiliki informasi maksimal. Sehingga untuk kalimat pertama atau terakhir, nilai fiturnya yaitu bernilai 1. Sedangkan untuk kalimat lainnya dapat dihitung menggunakan persamaan (1)[8].

$$PK = \frac{N-P}{N} \quad (1)$$

Pada persamaan (1), variabel *PK* adalah hasil perhitungan nilai fitur posisi kalimat untuk kalimat selain kalimat pertama atau terakhir. *N* adalah total jumlah kalimat dalam dokumen dan *P* adalah posisi kalimat dalam dokumen.

2. Fitur Panjang Kalimat

Kalimat yang sangat singkat tidak mengandung banyak informasi. Apabila jumlah kata dalam kalimat kurang dari 3 maka nilai fiturnya bernilai 0. Sedangkan untuk yang lainnya dalam menentukan kalimat penting berdasarkan panjang kalimatnya dapat dihitung menggunakan persamaan (2)[7],[8].

$$Panjang_Kalimat_i = \frac{\text{jumlah kata pada kalimat } i}{\text{jumlah kata pada semua kalimat}} \quad (2)$$

Pada persamaan (2), variabel *Panjang_Kalimat_i* adalah hasil perhitungan nilai fitur panjang kalimat pada kalimat ke-*i*. Untuk menghitung nilai fitur panjang kalimat pada kalimat ke-*i* yaitu jumlah kata pada kalimat ke-*i* dibagi dengan jumlah kata pada semua kalimat.

3. Fitur Data Numerik

Kalimat yang berisi data berupa bilangan juga dipertimbangkan dalam ringkasan karena data berupa bilangan dapat mempresentasikan suatu nilai penting pada dokumen. Fitur data numerik dapat dihitung menggunakan persamaan (3)[8].

$$DataNumerik_i = \frac{\text{jumlah data numerik pada kalimat } i}{\text{jumlah kata pada kalimat } i} \quad (3)$$

Pada persamaan (3), variabel *DataNumerik_i* adalah hasil

perhitungan nilai fitur data numerik pada kalimat ke-*i*. Untuk menghitung nilai fitur data numerik pada kalimat ke-*i* yaitu jumlah data numerik pada kalimat ke-*i* dibagi dengan jumlah kata pada kalimat ke-*i*. Data numerik yaitu data yang berupa bilangan.

4. Fitur Bobot Kalimat

Pada bobot kalimat menggunakan perhitungan *tf-isf* (*term frequency-inverse sentence frequency*). *Tf-isf* didapat dari frekuensi kemunculan kata pada kalimat beserta dengan jumlah dari frekuensi banyak kata yang muncul dalam sebuah dokumen. Frekuensi kemunculan kata pada kalimat dinyatakan dengan *tf*, jumlah kalimat yang terdapat dalam sebuah dokumen dinyatakan dengan *N*, dan jumlah kalimat yang mengandung kata dinyatakan dengan *n*. Fitur bobot kalimat dapat dihitung menggunakan persamaan (4)[9].

$$Bobot_Kalimat_i = tf_i \times isf_i = tf_i \times \left(\log \frac{N}{n_i} \right) \quad (4)$$

Pada persamaan (4), variabel *Bobot_Kalimat_i* adalah hasil perhitungan nilai fitur bobot kalimat pada kalimat ke-*i*. *tf_i* adalah frekuensi kemunculan kata pada kalimat. *N* adalah jumlah kalimat dalam satu dokumen. *n_i* adalah jumlah kalimat dimana kata tersebut muncul.

5. Fitur Kesamaan Antara Kalimat dan Centroid

Kalimat yang fitur *tf-isf* maksimal digunakan sebagai kalimat centroid. Kesamaan antara kalimat dan centroid dapat dihitung menggunakan *cosine similarity* seperti pada persamaan (5)[8].

$$\begin{aligned} Cos_{sim_i} &= \cos(\text{kalimat}_i, \text{centroid}) \\ &= \frac{\text{kalimat}_i \cdot \text{centroid}}{\| \text{kalimat}_i \| \| \text{centroid} \|} \end{aligned} \quad (5)$$

Pada persamaan (5), variabel *Cos_{sim_i}* adalah hasil perhitungan nilai fitur kesamaan antara kalimat ke-*i* dengan kalimat centroid. Variabel *kalimat_i* adalah vektor kalimat ke-*i*, yang akan dibandingkan kemiripannya. Variabel *centroid* adalah vektor kalimat centroid, yang akan dibandingkan kemiripannya. *kalimat_i · centroid* adalah *dot product* antara vektor kalimat ke-*i* dan vektor kalimat centroid. *||kalimat_i||* adalah panjang vektor kalimat ke-*i*. *||centroid||* adalah panjang vektor kalimat centroid. *||kalimat_i|| ||centroid||* adalah *cross product* antara *||kalimat_i||* dan *||centroid||*.

6. Fitur Bi-gram

Bi-gram adalah dua kata yang berdekatan pada setiap kalimat dalam dokumen. Nilai fitur Bi-gram dapat dihasilkan dengan cara menghitung jumlah total Bi-gram pada setiap kalimat[8].

7. Fitur Tri-gram

Tri-gram adalah tiga kata yang berdekatan pada setiap kalimat dalam dokumen. Nilai fitur Tri-gram dapat dihasilkan dengan cara menghitung jumlah total Tri-gram pada setiap kalimat[8].

8. Fitur Kata Benda yang Tepat

Kalimat memberi arti penting jika kalimat tersebut memiliki kata benda yang tepat. Kata benda yang tepat adalah kata yang digunakan untuk mengklasifikasikan orang, tempat, atau benda. Fitur kata benda yang tepat dapat dihitung menggunakan persamaan (6)[8].

$$KB_i = \frac{\text{jumlah kata benda yang tepat pada kalimat } i}{\text{jumlah kata pada kalimat } i} \quad (6)$$

Pada persamaan (6), variabel *KB_i* adalah hasil perhitungan nilai fitur kata benda yang tepat pada kalimat ke-*i*. Untuk

menghitung nilai fitur kata benda yang tepat pada kalimat ke-*i* yaitu jumlah kata benda yang tepat pada kalimat ke-*i* dibagi dengan jumlah kata pada kalimat ke-*i*.

9. Fitur Kemiripan Antar Kalimat

Kemiripan antar kalimat yaitu kata yang muncul dalam kalimat sama dengan kata yang muncul dalam kalimat lain. Kemiripan antar kalimat didapat dari jumlah jarak kedua kalimat dibagi jumlah maksimum jarak kedua kalimat. Fitur kemiripan antar kalimat dapat dihitung menggunakan persamaan (7)[9].

$$KK = \frac{\sum Sim(S_i, S_j)}{Max(\sum Sim(S_i, S_j))} \quad (7)$$

Pada persamaan (7), variabel *KK* adalah hasil perhitungan nilai fitur kemiripan antar kalimat. *Sim(S_i, S_j)* adalah kemiripan antar kalimat ke-*i* dan kalimat ke-*j*. $\sum Sim(S_i, S_j)$ adalah jumlah kemiripan antar kalimat ke-*i* dan kalimat ke-*j*. Dengan *S_i, S_j* diperoleh dari persamaan (8) jarak antar kalimat.

$$S_i, S_j = \frac{\sum_{t=1}^n w_{ti} \times w_{tj}}{\sqrt{\sum_{t=1}^n w_{ti}^2} \times \sqrt{\sum_{t=1}^n w_{tj}^2}} \quad (8)$$

Pada persamaan (8), *w_{ti}* adalah bobot kata *t* pada kalimat *i*. *w_{tj}* adalah bobot kata *t* pada kalimat *j*. *n* adalah jumlah kata dalam kalimat. $\sum_{t=1}^n w_{ti} \times w_{tj}$ adalah jumlah perhitungan skalar dari pembobotan kalimat.

$\sqrt{\sum_{t=1}^n w_{ti}^2} \times \sqrt{\sum_{t=1}^n w_{tj}^2}$ adalah jumlah perhitungan vektor dari pembobotan kalimat.

10. Fitur Huruf Besar

Kalimat yang berisi kata yang mengandung huruf besar mempunyai peluang dalam ringkasan karena kata yang mengandung huruf besar termasuk dalam kata yang dipentingkan pada dokumen. Fitur huruf besar dapat dihitung menggunakan persamaan (9)[10].

$$HB = \frac{\text{banyak kata yang terdapat huruf besar}}{\text{banyak kata dalam kalimat}} \quad (9)$$

Pada persamaan (9), variabel *HB* adalah hasil perhitungan nilai fitur huruf besar pada setiap kalimat. Untuk menghitung nilai fitur huruf besar pada setiap kalimat yaitu banyak kata yang terdapat huruf besar dalam kalimat dibagi dengan banyak kata dalam kalimat.

C. Matriks Fitur Kalimat

Setelah setiap kalimat nilai fiturnya dihitung, selanjutnya matriks fitur kalimat dibentuk yang berupa matriks 2-d yang terdiri dari *S* = (*s*₁, *s*₂, *s*₃, ..., *s*_N) adalah kalimat dalam dokumen dan *S_i* = (*f*₁, *f*₂, *f*₃, *f*₄, *f*₅, *f*₆, *f*₇, *f*₈, *f*₉, *f*₁₀) adalah vektor fitur.

D. Restricted Boltzmann Machine (RBM)

Restricted Boltzmann Machine (RBM) merupakan model generatif probabilistik yang mampu secara otomatis mengekstrak fitur input data dengan menggunakan algoritma pembelajaran tanpa pengawasan. RBM adalah jaringan saraf yang bersifat *stochastic*. Jaringan saraf berarti memiliki unit neuron berupa aktivasi biner yang bergantung pada neuron-neuron yang saling terhubung. Sedangkan *stochastic* berarti aktivasi yang memiliki unsur probabilistik. RBM terdiri dari dua binary unit yaitu *visible layer* dan *hidden layer* serta unit bias. *Visible layer* merupakan state yang akan diobservasi dan *hidden layer* merupakan *feature detectors*. Masing-masing *visible* unit terhubung ke semua

hidden unit yang diwakili oleh array bobot, sehingga masing-masing *hidden* unit juga terhubung ke semua *visible* unit dan unit bias terhubung ke semua *visible* unit dan semua *hidden* unit. Untuk memudahkan proses pembelajaran, jaringan dibatasi sehingga tidak ada *visible* unit terhubung ke *visible* unit lain dan *hidden* unit terhubung ke *hidden* unit lain.

Untuk melatih RBM, sampel dari *training set* yang digunakan sebagai input untuk RBM melalui neuron *visible*, dan kemudian jaringan sampel bolak balik antara neuron *visible* dan *hidden*. Tujuan dari pelatihan yaitu untuk pembelajaran koneksi bobot pada *visible* atau *hidden* dan bias aktivasi neuron sehingga RBM belajar untuk merekonstruksi data input selama fase dimana sampel neuron *visible* dari neuron *hidden*. Setiap proses sampling pada dasarnya berupa perkalian matriks antara sekumpulan sampel pelatihan dan matriks bobot, diikuti dengan fungsi aktivasi neuron yaitu fungsi sigmoid. Sampling antara lapisan *hidden* dan *visible* diikuti oleh modifikasi parameter (dikontrol oleh *learning rate*) dan diulang untuk setiap kelompok data dalam *training set*, dan untuk state sebanyak yang diperlukan untuk mencapai konvergensi[11].

Unit *hidden* diinisialisasi dan diperbaharui menggunakan persamaan (10), dimana *H_j* dari setiap unit *hidden* *j* dengan nilai *V* (binari state unit *visible*) diatur satu dengan probabilitas :

$$p(H_j = 1 | V) = \sigma(b_j + \sum_i V_i W_{ij}) \quad (10)$$

Dimana $\sigma(x)$ adalah fungsi sigmoid

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (11)$$

Pada persamaan (10), *b_j* adalah bias dari unit *hidden*. *V_i* adalah binari state dari *visible* unit *i*. *W_{ij}* adalah bobot antara unit *visible* dan unit *hidden*. *i* (*i* = 1,2,3,...,n) untuk mewakili jumlah dari *visible* neuron. Sedangkan pada persamaan (11), *x* merupakan hasil perhitungan dari *b_j* + $\sum_i V_i W_{ij}$.

Unit *visible* diinisialisasi dan diperbaharui menggunakan persamaan (12), dimana *V_i* dari setiap unit *visible* *i* dengan nilai *H* (binari state unit *hidden*) diatur satu dengan probabilitas :

$$p(V_i = 1 | H) = \sigma(a_i + \sum_j H_j W_{ij}) \quad (12)$$

Pada persamaan (12), *a_i* adalah bias dari unit *visible*. *H_j* adalah binari state dari *hidden* unit *j*. *W_{ij}* adalah bobot antara unit *visible* dan unit *hidden*. *j* (*j* = 1,2,3,...,n) untuk mewakili jumlah dari *hidden* neuron. σ adalah fungsi sigmoid yang dihitung dengan menggunakan persamaan (11).

Metode RBM digunakan dalam penelitian ini karena mampu meningkatkan nilai fitur sehingga ringkasan yang dihasilkan lebih akurat. Matriks fitur kalimat yang diperoleh dari proses sebelumnya dilakukan normalisasi dengan membagi setiap elemennya dengan elemen tertinggi, selanjutnya digunakan sebagai input RBM. Berikut adalah tahapan-tahapan menggunakan RBM[11] :

1. Inisialisasi data
 - a. Lakukan proses pencarian nilai bobot dan bias dengan nilai random yang kecil.
 - b. Tentukan *Learning Rate* dan maksimum Epoch yang akan digunakan.
 - c. Selama Epoch < MaksimumEpoch, lakukan langkah dibawah ini.

d. Selama $datasampel < maksimumdatasampel$, lakukan langkah dibawah ini.

2. Fase Positif

Setelah tahap inialisasi data, selanjutnya adalah tahap fase positif. Fase positif yaitu mengambil data dan sampel dari *hidden* unit.

- a. Menghitung energi aktivasi, probabilitas dan state dari unit *hidden* dengan menggunakan persamaan (10).
- b. Menghitung positif_assosiatif dengan menggunakan persamaan (13). Untuk menghitung positif assosiatif yaitu matriks data sampel dari *visible* neuron ditranspose dikali dengan probabilitas yang dihasilkan dari langkah 2(a).

$$Pos_Asso = (data)^T * P(H_j) \quad (13)$$

Pada persamaan (13), $(data)^T$ merupakan matriks data sampel dari *visible* neuron yang ditranspose. $P(H_j)$ merupakan probabilitas unit *hidden* yang dihasilkan dari langkah 2(a).

3. Fase Negatif

Tahap selanjutnya adalah fase negatif, yaitu merekonstruksi *visible* unit dan data sampel dari *hidden* unit.

- a. Menghitung energi aktivasi dan probabilitas dari unit *visible* dengan menggunakan persamaan (12).
- b. Melakukan langkah 2(a) untuk update *hidden* unit.
- c. Menghitung negatif_assosiatif dengan menggunakan persamaan (14). Untuk menghitung negatif assosiatif yaitu matriks data (probabilitas dari unit *visible* yang diperoleh dari langkah 3(a)) yang ditranspose dikali dengan probabilitas dari unit *hidden* yang dihasilkan dari langkah 3(b).

$$Neg_Asso = (data)^T * P(H_j) \quad (14)$$

Pada persamaan (14), $(data)^T$ merupakan matriks data (probabilitas dari unit *visible* yang diperoleh dari langkah 3(a)) yang ditranspose. $P(H_j)$ merupakan probabilitas unit *hidden* yang dihasilkan dari langkah 3(b).

4. Update Bobot

Tahap selanjutnya adalah update bobot, dengan menggunakan persamaan (15).

$$W_{ij}(\text{baru}) = W_{ij}(\text{lama}) + \Delta W_{ij} \quad (15)$$

$$\Delta W_{ij} = \varepsilon (Pos_Asso - Neg_Asso) \quad (16)$$

Pada persamaan (15), W_{ij} adalah bobot antara unit *visible* dan unit *hidden*. ΔW_{ij} adalah perubahan bobot. Untuk menghitung bobot yang baru yaitu bobot lama ditambah dengan perubahan bobot yang dihasilkan. Sedangkan pada persamaan (16), merupakan perhitungan perubahan bobot. ε adalah *learning rate*. Perhitungan perubahan bobot ΔW_{ij} diperoleh dari *learning rate* dikali dengan selisih antara positif assosiatif dan negatif assosiatif.

5. Hitung Error

Untuk menghitung error menggunakan persamaan (17).

$$Error = \frac{1}{2} \sum_{i=1}^p (O_i - t_i)^2 \quad (17)$$

Pada persamaan (17), Error dihitung dengan pengurangan O_i merupakan data sampel dan t_i merupakan *visible* probabilitas yang dihasilkan dari fase negatif pada langkah 3(a). p adalah jumlah data.

Dalam penelitian ini, pada *visible* layer terdapat sepuluh

neuron karena terdapat sepuluh fitur. *Learning rate* yang digunakan yaitu 0.01, 0.03, 0.06 dan 0.09 serta pembatasan epoch sebanyak 50 epoch.

E. Matriks Fitur Kalimat yang ditingkatkan

Setelah setiap kalimat nilai fiturnya ditingkatkan dengan menggunakan metode RBM, maka diperoleh matriks fitur kalimat yang ditingkatkan.

F. Perhitungan Skor Kalimat

Setelah dihasilkan matriks fitur kalimat yang ditingkatkan, selanjutnya adalah menghitung skor setiap kalimat pada dokumen. Skor kalimat diperoleh dengan cara menghitung nilai total dari nilai fitur yang telah ditingkatkan pada setiap kalimat.

G. Penyusunan Ringkasan Teks

Kalimat diurut secara *descending* berdasarkan skor kalimat. Untuk menentukan kalimat dalam ringkasan yaitu kalimat yang memiliki urutan teratas sesuai *compression rate* yang ditentukan. *Compression rate* yang digunakan dalam penelitian ini yaitu 20%, 30% dan 40%. Kalimat dalam ringkasan diurutkan sesuai dengan posisi asli dalam dokumen.

IV. DATASET ARTIKEL BERITA BERBAHASA INDONESIA

Dataset yang digunakan dalam penelitian ini yaitu dokumen tunggal teks berita berbahasa Indonesia yang diperoleh dari www.kompas.com pada kategori tekno. Artikel berita yang digunakan sebanyak 30 artikel yang diambil dari postingan tanggal 1 Januari 2019 sampai 31 Maret 2019. Dataset yang digunakan dalam penelitian ini berupa isi dari artikel berita. Situs berita online kompas.com digunakan dalam penelitian ini karena kompas.com merupakan salah satu situs berita online di Indonesia yang sering dikunjungi[12]. Sedangkan kategori tekno yaitu mengulas tentang kabar berita terkini dunia IT meliputi gadget terbaru, games, apps, smartphone, review produk, internet, software dan hardware.

Dalam penelitian ini melibatkan seorang pakar yang dapat membantu dalam menentukan hasil ringkasan secara manual. Dimana nantinya hasil ringkasan manual tersebut digunakan sebagai pembanding pada saat melakukan pengujian. Hasil ringkasan manual dari pakar berupa data kuisisioner yang terdiri dari dokumen-dokumen kalimat berita, selanjutnya pakar melakukan ceklist pada dokumen kalimat yang dianggap mengandung informasi penting yang mampu mewakili isi berita secara keseluruhan. Seorang pakar yang terlibat dalam penelitian ini yaitu guru Bahasa Indonesia. Guru Bahasa Indonesia dipilih karena dalam penelitian ini menggunakan dokumen teks berbahasa Indonesia.

V. PENGUJIAN

Metode pengujian ringkasan yang digunakan dalam penelitian ini yaitu intrinsik. Metode intrinsik merupakan metode evaluasi dimana kualitas ringkasan diukur berdasarkan hasil ringkasan yang dihasilkan. Dalam evaluasinya, dilakukan perbandingan antara hasil ringkasan sistem dengan hasil ringkasan manual yang dibuat oleh pakar.

Recall Oriented Understudy for Gisting Evaluation (ROUGE) adalah proses pengujian dengan menghitung jumlah *n-gram* yang sama antara ringkasan sistem dengan ringkasan manual. Dalam penelitian ini menggunakan ROUGE-1 karena memiliki tes signifikan *recall* yang tinggi. ROUGE-1 yaitu menghitung jumlah unigram kata yang sama antara ringkasan sistem dengan ringkasan manual. Parameter untuk evaluasi kinerja adalah *Recall*, *Precision* dan *F-Measure* dengan menggunakan persamaan 18, 19, dan 20[13].

$$\text{ROUGE-N Recall} = \frac{\text{Common } N\text{-grams}(\text{Peer,Reference})}{N\text{-grams}(\text{Reference})} \quad (18)$$

$$\text{ROUGE-N Precision} = \frac{\text{Common } N\text{-grams}(\text{Peer,Reference})}{N\text{-grams}(\text{Peer})} \quad (19)$$

$$\text{ROUGE-N F-Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (20)$$

Pada persamaan (18), persamaan (19), dan persamaan (20) terdapat tiga komponen penting yaitu *Common N-grams(Peer,Reference)* adalah jumlah *n-gram* dalam ringkasan sistem dan ringkasan manual. *N-grams(reference)* adalah jumlah *n-gram* dalam ringkasan manual. *N-grams(peer)* adalah jumlah *n-gram* dalam ringkasan sistem.

Proses pengujian yang dilakukan dalam penelitian ini yaitu pengujian berdasarkan *learning rate*, pengujian berdasarkan *compression rate* dan melakukan perbandingan antara hasil ringkasan menggunakan RBM dan tanpa RBM.

A. Pengujian berdasarkan Learning Rate

TABEL I
HASIL PENGUJIAN BERDASARKAN LEARNING RATE

Learning Rate (ε)	Recall	Precision	F-Measure
0.01	0.728	0.604	0.657
0.03	0.737	0.610	0.664
0.06	0.744	0.611	0.669
0.09	0.736	0.603	0.660

Terdapat empat *learning rate* yang digunakan dalam pengujian ini yaitu 0.01, 0.03, 0.06 dan 0.09. Dalam pengujian digunakan 30 dokumen artikel berita berbahasa Indonesia dengan *compression rate* 40%. Pembatasan epoch yang digunakan sebanyak 50 epoch. Pengujian ini dilakukan untuk mengetahui hasil *recall*, *precision* dan *f-measure* berdasarkan *learning rate* yang diuji. Dari hasil pengujian, tampak bahwa *learning rate* dengan 0.06 menghasilkan *recall*, *precision* dan *f-measure* tertinggi seperti yang terlihat pada tabel I.

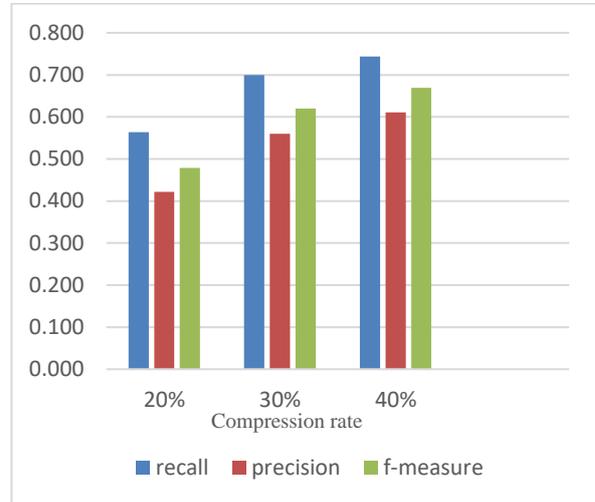
B. Pengujian berdasarkan Compression Rate

Dalam pengujian ini, *compression rate* yang digunakan yaitu 20%, 30% dan 40% dari setiap data dokumen. Sebanyak 30 dokumen artikel berita berbahasa Indonesia yang digunakan. Pengujian ini dilakukan untuk mengetahui hasil *recall*, *precision* dan *f-measure* berdasarkan *compression rate* yang diuji. Pada gambar 3 tampak bahwa hasil *recall*, *precision* dan *f-measure* tertinggi pada saat pengujian dengan menggunakan *compression rate* 40%.

C. Pengujian dengan Membandingkan antara Hasil Peringkasan menggunakan RBM dan tanpa RBM

Pengujian ini dilakukan untuk mengetahui perbandingan antara hasil peringkasan dengan menggunakan RBM dan hasil peringkasan tanpa RBM. Dari hasil pengujian, tampak

bahwa terdapat perbedaan hasil rata-rata *recall*, *precision* dan *f-measure* pada peringkasan menggunakan RBM dengan peringkasan tanpa RBM. Nilai rata-rata *recall*, *precision* dan *f-measure* pada peringkasan menggunakan RBM lebih tinggi daripada nilai rata-rata *recall*, *precision* dan *f-measure* pada peringkasan tanpa RBM seperti yang terlihat pada tabel II.



Gambar 3. Grafik Hasil Pengujian berdasarkan Compression Rate

TABEL II
HASIL PENGUJIAN BERDASARKAN PERBANDINGAN ANTARA MENGGUNAKAN RBM DAN TANPA RBM

Artikel	Dengan RBM			Tanpa RBM		
	R	P	F	R	P	F
1	0.4953	0.3926	0.4380	0.4953	0.3926	0.4380
2	0.6879	0.6136	0.6486	0.6879	0.6136	0.6486
3	0.5773	0.4786	0.5234	0.5773	0.4786	0.5234
4	0.4588	0.4382	0.4483	0.4588	0.4382	0.4483
5	0.8107	0.6009	0.6902	0.6627	0.5068	0.5744
6	0.8065	0.7813	0.7937	0.8065	0.7813	0.7937
7	0.9130	0.6087	0.7304	0.9130	0.6087	0.7304
8	0.7822	0.6870	0.7315	0.7822	0.6870	0.7315
9	1.0000	0.7248	0.8404	1.0000	0.7248	0.8404
10	0.8049	0.5593	0.6600	0.8049	0.5593	0.6600
11	0.7216	0.5983	0.6542	0.7216	0.5983	0.6542
12	0.5833	0.3203	0.4135	0.5476	0.3108	0.3966
13	0.5446	0.4766	0.5083	0.5446	0.4766	0.5083
14	0.6835	0.5243	0.5934	0.6835	0.5243	0.5934
15	0.6458	0.4189	0.5082	0.6563	0.4286	0.5185
16	0.6364	0.5469	0.5882	0.6364	0.5469	0.5882
17	0.9070	0.5821	0.7091	0.9070	0.5821	0.7091
18	0.5960	0.5728	0.5842	0.5960	0.5728	0.5842
19	0.7092	0.5917	0.6452	0.7092	0.5917	0.6452
20	0.6463	0.5521	0.5955	0.5244	0.4624	0.4914
21	0.6981	0.5606	0.6218	0.6981	0.5606	0.6218
22	0.8169	0.7073	0.7582	0.8169	0.7073	0.7582
23	0.8721	0.6466	0.7426	0.8721	0.6466	0.7426
24	0.8542	0.7736	0.8119	0.7292	0.6863	0.7071
25	0.7015	0.4845	0.5732	0.6418	0.4526	0.5309
26	0.5765	0.5385	0.5568	0.5765	0.5385	0.5568
27	0.5301	0.3385	0.4131	0.5301	0.3385	0.4131
28	0.6742	0.6122	0.6417	0.5393	0.4528	0.4923
29	0.5783	0.4948	0.5333	0.5783	0.4948	0.5333
30	0.7000	0.5698	0.6282	0.6857	0.5714	0.6234
Rerata	0.700	0.560	0.620	0.679	0.544	0.602

VI. KESIMPULAN

Dalam melakukan peringkasan teks ekstraktif secara otomatis dengan menggunakan metode *Restricted Boltzmann Machine* (RBM), salah satu yang perlu diperhatikan yaitu penentuan *learning rate* yang akan digunakan agar ringkasan yang dihasilkan baik. Berdasarkan pengujian yang dilakukan dalam penelitian ini, dengan menggunakan *learning rate* 0.06 menghasilkan *recall*, *precision* dan *f-measure* tertinggi yakni 0.744, 0.611 dan 0.669.

Compression rate digunakan dalam peringkasan untuk menentukan panjang ringkasan yang dihasilkan. Berdasarkan pengujian yang dilakukan, semakin besar nilai *compression rate* yang digunakan maka hasil *recall*, *precision* dan *f-measure* yang dihasilkan akan semakin tinggi.

Pada peringkasan teks tanpa RBM diperoleh nilai rata-rata *recall* 0.679, *precision* 0.544 dan *f-measure* 0.602. Sedangkan pada peringkasan teks menggunakan metode RBM diperoleh nilai rata-rata *recall* 0.700, *precision* 0.560 dan *f-measure* 0.620. Peringkasan teks dengan menggunakan metode RBM menghasilkan nilai rata-rata *recall*, *precision* dan *f-measure* lebih tinggi daripada peringkasan teks tanpa menggunakan RBM, walaupun perbedaan hasil yang diperoleh tidak terlalu jauh. Peringkasan teks menggunakan metode RBM memiliki nilai *recall* lebih tinggi 2.1%, *precision* lebih tinggi 1.6% dan *f-measure* lebih tinggi 1.8% daripada peringkasan teks tanpa RBM. Hal ini menunjukkan bahwa peringkasan teks dengan menggunakan RBM hasilnya lebih baik daripada peringkasan teks tanpa RBM.

Pada metode RBM yang digunakan dalam penelitian ini masih perlu dilakukan improvisasi dalam penentuan nilai bobot dan bias yang akan digunakan agar ringkasan yang dihasilkan lebih baik lagi. Untuk penelitian selanjutnya dapat menambahkan fitur selain yang ada pada penelitian ini serta metode ini dapat dikombinasikan dengan metode lain untuk meningkatkan hasil peringkasan.

DAFTAR PUSTAKA

- [1] J. S. Saputra, M. Fachrurrozi, Yunita, "Peringkasan Teks Berita Berbahasa Indonesia Menggunakan Metode Latent Semantic Analysis (LSA) dan Teknik Steinberger & Jezek," in *Prosiding Annual Research Seminar Computer Science and ICT*, 2017.
- [2] S. Irawan, Hermawan, Samsuryadi, "Studi Awal Peringkasan Dokumen Bahasa Indonesia Menggunakan Metode Latent Semantic Analysis dan Maximum Marginal Relevance," in *Prosiding Annual Research Seminar*, 6 Desember 2016.
- [3] I P. G. H. Suputra. (2017, April). Peringkasan Teks Otomatis Untuk Dokumen Bahasa Bali Berbasis Metode Ekstraktif. *Jurnal Ilmu Komputer*. X(1), pp. 33-38. Available: <https://ojs.unud.ac.id/index.php/jik/article/view/39775/24171>
- [4] N. S. W. Gotami, Indriati, R. K. Dewi. (2018, September). Peringkasan Teks Otomatis Secara Ekstraktif Pada Artikel Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Latent Semantic Analysis. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. 2(9), pp. 2821-2828. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2430/905>
- [5] A. Ambekar, K. Shah, M. Agrawal, S. Pawar, A. Shaikh. (2018, June). Text Summarization Using Restricted Boltzmann Machine: Unsupervised Deep Learning Approach. *IJSART*. 4(6), pp. 103-107. Available: <http://ijsart.com/Content/PDFDocuments/IJSARTV4I623858.pdf>
- [6] M. Elgamal, Prof. Dr S. Hamada, Prof. Dr R. Aboelezz and Dr M. Abou-Kreisha. (2019, August). Better Results in Automatic Arabic

- Text Summarization System Using Deep Learning based RBM than by Using Clustering Algorithm based LSA. *International Journal of Scientific & Engineering Research*. 10(8), pp. 781-786. Available: <https://www.ijser.org/researchpaper/Better-Results-in-Automatic-Arabic-Text-Summarization-System-Using-Deep-Learning-based-RBM-than-by-Using-Clustering-Algorithm-based-LSA.pdf>
- [7] S. P. Singh, A. Kumar, A. Mangal, S. Singhal. (2016). Bilingual Automatic Text Summarization Using Unsupervised Deep learning. *IEEE International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. pp. 1195-1200. Available: <https://ieeexplore.ieee.org/document/7754874>
 - [8] N. S. Shirwandkar, Dr. S. Kulkarni. (2018). Extractive Text Summarization using Deep Learning. *IEEE Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. Available: <https://ieeexplore.ieee.org/document/8697465>
 - [9] N. I. Widiastuti, W. K. Afnan. (2017). Fuzzy Logic dan Lexical Chains untuk Peringkasan Teks Otomatis. *Jurnal Sistem Komputer*. 7(1), pp. 5-12. Available: <https://docplayer.info/91149501-Fuzzy-logic-dan-lexical-chains-untuk-peringkasan-teks-otomatis.html>
 - [10] A. Ridok, T. C. Romadhona, "Peringkasan Dokumen Otomatis Menggunakan Metode Fuzzy Model Sistem Inferensi Mamdani," in *Seminar Nasional Teknologi Informasi dan Multimedia*, Yogyakarta, 19 Januari 2013, pp. 19-24.
 - [11] Susilawati, "Algoritma Restricted Boltzmann Machines (RBM) untuk Pengenalan Tulisan Tangan Angka," in *Seminar Nasional Teknologi Informatika, "The Future of Computer Vision"*, 2017, pp. 140-148.
 - [12] D. Branding (2019) "Ini Dia 7 Situs Berita Online di Indonesia yang Sering di Kunjungi," [Online]. <https://www.nataconnexindo.com/blog/ini-dia-7-situs-berita-online-di-indonesia-yang-sering-di-kunjungi>, tanggal akses: 21-Mei-2020.
 - [13] J. Yadav, Dr. Y. K. Meena. (2016, Sept). Use of Fuzzy Logic and WordNet for Improving Performance of Extractive Automatic Text Summarization. *IEEE Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI)*. pp. 2071-2077. Available: <https://ieeexplore.ieee.org/document/7732356>

Rully Widiastutik lahir di Sumenep, Jawa Timur, Indonesia, pada tahun 1988. Menyelesaikan studi S1 di program studi Teknik Informatika Universitas Trunojoyo Madura pada tahun 2010, dan berkarir sebagai dosen di program studi Teknik Informatika Akademi Komunitas Negeri Sumenep. Minat penelitiannya adalah bidang text mining.

Lukman Zaman P. C. S. W berkarir sebagai dosen di program studi Teknologi Informasi Institut Sains dan Teknologi Terpadu Surabaya.

Joan Santoso lahir di Surabaya, Jawa Timur, Indonesia. Telah menyelesaikan pendidikan S1 pada tahun 2011 dan S2 pada tahun 2013 dari Sekolah Tinggi Teknik Surabaya. Minat penelitiannya ialah computational linguistic, information extraction, machine learning, dan big data processing.

Stance Classification Post Kesehatan di Media Sosial Dengan FastText *Embedding* dan *Deep Learning*

Ernest Lim, *Teknologi Informasi ISTTS*, Esther Irawati Setiawan, *Teknologi Informasi ISTTS*, dan Joan Santoso, *Teknologi Informasi ISTTS*

Abstrak— Misinformasi merupakan fenomena yang semakin sering terjadi di media sosial, tidak terkecuali Facebook, salah satu media sosial terbesar di Indonesia. Beberapa penelitian telah dilakukan mengenai teknik identifikasi dan klasifikasi *stance* di media sosial Indonesia. Akan tetapi, penggunaan Word2Vec sebagai *word embedding* dalam penelitian tersebut memiliki keterbatasan pada pengenalan kata baru. Hal ini menjadi dasar penggunaan *fastText embedding* dalam penelitian ini. Dengan menggunakan pendekatan *deep learning*, penelitian berfokus pada performa model dalam klasifikasi *stance* suatu judul *post* kesehatan di Facebook terhadap judul *post* lainnya. *Stance* berupa *for* (setuju), *observing* (netral), dan *against* (berlawanan). Dataset terdiri dari 3500 judul *post* yang terdiri dari 500 kalimat klaim dengan enam kalimat *stance* terhadap setiap klaim. Model dengan *fastText* pada penelitian ini mampu menghasilkan *F1 macro score* sebesar 64%.

Keywords—Bahasa Indonesia, *Deep Learning*, *fastText*, Media Sosial, *Stance Classification*

I. PENDAHULUAN

Penyebaran informasi lewat media sosial memiliki sifat yang berbeda dari penyebaran di dunia nyata. Aspek komunikasi media sosial sangat dipengaruhi oleh fenomena *echo chamber* [1], [2]. Bentuk umum dari fenomena ini adalah kepercayaan *user* terhadap suatu informasi bila informasi tersebut sejalan dengan keyakinan politik *user* [3]. Dengan demikian, misinformasi ataupun *hoax* menyebar dengan cepat di sosial media, seperti Facebook [4]. Selain itu, aspek *bot* juga turut membantu penyebaran *hoax* secara signifikan [5]. Hal ini berdampak pada berbagai masalah penting seperti pemilu [6], penanganan konflik dan krisis [7], dan penanganan wabah [8].

Pada kasus Zika, rumor dibagikan tiga kali lebih banyak dibanding berita yang terverifikasi, meskipun sudah terdapat sosialisasi di media sosial. Kelemahan strategi komunikasi ini disebabkan oleh situasi yang selalu berkembang secara cepat sehingga banyak informasi yang beredar dan mengurangi visibilitas informasi kesehatan yang benar [9].

Oleh sebab itu, penting untuk dilakukan identifikasi dan

klasifikasi *hoax*, terutama di bidang kesehatan. Pendekatan dalam klasifikasi *hoax* pada umumnya hanya klasifikasi konten. Akan tetapi, media sosial memberikan peluang untuk meneliti *stance classification* dengan banyaknya data yang tersedia, seperti profil *user* hingga interaksi pada *post*.

Stance classification adalah teknik untuk mengetahui secara otomatis pandangan penulis teks terhadap suatu *statement*, apakah mendukung, menentang, atau netral [10]. Proses *stance classification* secara umum dapat dibagi menjadi dua tahap, yaitu: pembentukan vektor kata dan klasifikasi *stance*. Dalam pembentukan vektor kata, banyak penelitian *stance classification* yang menggunakan Word2Vec untuk *embedding*. Popularitas Word2Vec ini dikarenakan kemampuannya dalam mengetahui similaritas makna antar kata. Informasi similaritas ini didapatkan dengan memperhatikan kesamaan kata-kata di sekitar kata target [11].

Pada tahap klasifikasi, *Convolutional Neural Network* (CNN) termasuk salah satu *classifier* yang populer untuk *stance classification*. CNN pertama kali digunakan oleh Collobert untuk mengambil fitur penting secara otomatis [12]. Informasi hirarki fitur ini didapatkan melalui operasi *max-over-time pooling* yang dikembangkan lebih lanjut oleh Kim untuk klasifikasi kalimat [13]. Namun, banyak juga metode klasifikasi *stance* yang menggunakan LSTM. Tren ini disebabkan oleh kemampuan LSTM dalam menyimpan informasi sekuens kata dalam kalimat [14].

Dalam konteks bahasa Indonesia, penelitian *stance classification* pada umumnya menggunakan Word2Vec, seperti pada penelitian [15] dan *classifier* LSTM [16]. Hal ini dapat menyebabkan hasil klasifikasi yang kurang akurat karena faktor kesalahan pengetikan. Oleh karena itu, peneliti mengusulkan penggunaan *fastText* saat pembentukan vektor kata dan kalimat dan menggunakan *deep learning*, terutama Bi-LSTM, untuk menghitung probabilitas *stance*.

Usulan tersebut mengacu pada penelitian [16] yang mampu menghasilkan model yang cukup akurat dengan menggunakan Word2Vec dan *classifier* LSTM. Pemilihan *fastText* sendiri berdasarkan kepada kelebihan *fastText* saat membaca kata. Dengan *n-gram character*, sebuah kata akan dibaca per karakter sesuai nilai *n-gram* tersebut [17]. Dengan demikian, permasalahan *Out-of-Vocabulary words* (OOV) dapat teratasi. Bi-LSTM sendiri dipilih karena terbukti lebih akurat dan tidak serentan LSTM untuk *overfitting* [18]. Peneliti juga memilih CNN karena proses pembelajaran yang membutuhkan relatif sedikit *epoch* dan

Ernest Lim, Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: ernest1@mhs.sts.edu)

Esther Irawati Setiawan, Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: esther@stts.edu)

Joan Santoso, Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: joan@stts.edu)

mampu menghasilkan akurasi yang tinggi [19] dan LSTM karena selain menjadi dasar perbandingan dengan penelitian Setiawan [16], LSTM terbukti tepat untuk permasalahan teks yang berhubungan dengan sekuens [20].

Tujuan dari penelitian ini adalah menemukan pendekatan dan parameter *stance classification* yang efektif untuk membantu klarifikasi *hoax* di media sosial. Secara khusus, peneliti mencari kombinasi metode dan pengaturan parameter dalam penggunaan fastText dan *deep learning* untuk klasifikasi *stance* yang paling akurat.

II. PENELITIAN TERKAIT

Penelitian mengenai *stance*, seperti terlihat pada Tabel I, banyak yang menggunakan *word embedding* Word2Vec dan *classifier* CNN dan LSTM.

TABEL I
PENELITIAN STANCE DETECTION DAN CLASSIFICATION

Topik	Riset	Pendekatan	Hasil
<i>Stance Classification</i>	Chen, 2016	Pemodelan CNN + fitur akun media sosial (jumlah <i>like</i> , <i>comment</i>)	Model memberikan hasil akurasi tertinggi dibanding tanpa informasi tambahan dari medsos
<i>Stance Classification</i>	Kochkina, 2017	Pemodelan kalimat rata-rata W2V + jumlah kata negatif + <i>classifier</i> LSTM	Pendekatan memenangkan SemEval 2017 Task 8 Sub-task A
<i>Stance Classification</i>	Lozano, 2017	<i>POS Tagging</i> W2V + fitur akun Twitter + <i>handwritten rules</i> + <i>classifier</i> CNN	Peringkat 4 dalam SemEval 2017 RumourEval 8A, akurasi 74,9%
<i>Stance Detection</i>	Shalini, 2019	GloVe / W2V + <i>classifier</i> BoT / CNN / Bi-LSTM	Bi-LSTM + BoT emb memberi performa tertinggi
<i>Stance Classification</i>	Jannati, 2019	Pemodelan kalimat rata-rata W2V + SVM	<i>F1 macro</i> 63,54%
<i>Stance Classification</i>	Setiawan, 2020	Pemodelan kalimat rata-rata W2V / LSTM W2V + <i>classifier</i> LSTM	Pendekatan <i>full</i> LSTM memberi <i>F1 macro</i> tertinggi

Pada penelitian Chen [21], *tweet* yang dikumpulkan berhasil diklasifikasi dengan akurasi tertinggi dengan menambahkan data media sosial pada input dengan *classifier* CNN. Sementara itu, penelitian Kochkina [22] melakukan *stance classification* pada *tweet* dengan menghitung jumlah kata negatif pada *tweet* dan pemodelan kalimat dengan pendekatan rata-rata. *Word embedding* yang digunakan adalah model Word2Vec *pre-trained* dengan *Google News dataset*. *Classifier* yang digunakan adalah *branch-LSTM*. Pendekatan ini memenangkan kompetisi RumourEval dengan akurasi 78,4%.

Pendekatan yang mirip untuk mengklasifikasi *stance* di media sosial juga dilakukan oleh Lozano [19]. Namun, berbeda dengan Kochkina, Lozano menggunakan *classifier* CNN. Pendekatan ini memperoleh peringkat keempat

dengan akurasi 74,9%. Lozano berargumen CNN memberikan hasil yang akurat pada jumlah data yang kecil tapi juga berisiko *overfitting* secara cepat.

Sementara itu, Shalini [23] mendapat performa model tertinggi dengan Bi-LSTM dan Bag-of-Tricks *embedding* dengan *F1 score* sebesar 79,84%. Model digunakan untuk mendeteksi *stance* suatu *tweet* apakah mendukung atau menolak suatu topik.

Dalam konteks bahasa Indonesia, penelitian yang paling dekat yang juga melakukan *stance classification* antara dua kalimat judul adalah penelitian Setiawan [16]. Penelitian tersebut menunjukkan *stance classification* dapat dilakukan dengan pemodelan kalimat dengan LSTM dan diproses ke *classifier* LSTM dengan *F1 macro* 71%. Dalam penelitian tersebut dilakukan pula uji coba pemodelan kalimat dengan pendekatan rata-rata dan *classifier* LSTM yang menghasilkan *F1 macro* sebesar 60,66%.

Penulis memosisikan penelitian ini sebagai pengembangan penelitian *stance classification* berbahasa Indonesia, terutama dari penelitian [16]. Pengembangan ini dilakukan dengan mengubah *word embedding* menjadi fastText dan menggabungkan model dan fitur-fitur dari penelitian Setiawan [16] dengan pendekatan kalimat pada penelitian Kochkina [22] yang mampu menghasilkan akurasi lebih tinggi. Mengingat penelitian ini meneliti *post* media sosial, maka diambil juga aspek pengambilan data *post* seperti jumlah *like* dan *comment* seperti pada penelitian [21] dan [19].

A. *Stance Classification*

Penelitian *stance* pada awalnya banyak dilakukan pada bidang-bidang diluar topik penyebaran *hoax*, seperti debat kongres [24] dan forum debat online [25]. Namun, pada tahun 2017, diperkenalkan kompetisi *Fake News Challenge* untuk mengidentifikasi *hoax*. Fokus dari kompetisi tersebut adalah mencari *stance* isi berita dengan judul dari artikel berita tersebut. Pada *Fake News Challenge*, pendekatan yang memiliki akurasi tertinggi adalah pendekatan *multi-layer perceptrons* [26] dan CNN dengan *decision tree*. Di Indonesia, salah satu penelitian terbaru mengenai *stance classification* adalah analisis pendapat terhadap berita kesehatan [16]. Dalam paper tersebut, LSTM digunakan untuk mencari vektor kalimat dan sebagai *classifier stance*.

Penelitian ini dirancang sebagai pengembangan dari penelitian analisis pendapat berita kesehatan dalam konteks media sosial. Penelitian ini juga mengeksplorasi model pembentukan vektor kalimat pendekatan sederhana rata-rata dengan *classifier* berbasis *deep learning*. Metode *deep learning* yang dipilih adalah CNN, LSTM, dan Bi-LSTM.

B. NLP dengan CNN

Penggunaan *Convolutional Neural Network* (CNN) untuk penelitian NLP dilakukan pertama kali di tahun 2011 [12]. Konsep CNN dengan lapis *max-pooling* dikembangkan dengan membuat dua kelompok vektor kata dimana setiap kelompok dianggap sebagai *channel*. Dengan *backpropagation* hanya pada satu *channel*, model dapat melatih satu kelompok vektor sesuai penugasan yang ada. Hal ini juga membuat kelompok vektor yang lain statis [13]. Aplikasi CNN untuk *stance classification* sendiri dapat

digabung dengan *automatic rule mining*, *hand written rules*, dan *voting* [19]. Untuk konteks media sosial, penggabungan tersebut adalah model CNN dengan informasi *user* dan *post* [21].

C. NLP dengan LSTM

LSTM sendiri merupakan pengembangan dari RNN sederhana. Bedanya, LSTM memiliki *forget gate* yang Dalam penugasan *stance*, LSTM dapat mengatasi permasalahan kalimat panjang yang dihadapi oleh CNN. Pada penelitian Hanselowski [27], model baru dengan penambahan *stacked LSTM* mampu menghasilkan peningkatan akurasi klasifikasi *Fake News Challenge*.

D. NLP dengan Bi-LSTM

Bi-LSTM telah diaplikasikan dalam *stance detection* seperti pada SemEval 2016 Task 6 dalam konteks Twitter [28]. Pada penelitian Habernalt [29], *bidirectional LSTM* terbukti lebih akurat dalam membandingkan makna kalimat satu dengan yang lain terhadap sebuah topik. Mrowca [30] menggunakan Bi-LSTM untuk mengambil informasi fitur kalimat judul berita dan teks dari isi berita tersebut dalam *Fake News Challenge*. Klasifikasi sendiri dilakukan dengan menggabungkan *output* dari Bi-LSTM dan vektor fitur.

E. Word dan Sentence Embedding

Word embedding berbasis *neural network* dimulai pada tahun 2003 dengan *feedforward neural network model* pada penelitian Bengio [31]. Untuk mengurangi kompleksitas komputasional, Mikolov memperkenalkan Word2Vec yang juga mampu mengekstrak similaritas antar vektor kata dengan *cosine similarity* [11].

FastText [17] adalah hasil pengembangan dari Word2Vec. Berbeda dari Word2Vec, fastText tidak memakai hanya satu kata secara utuh untuk diproses, tapi fastText menggunakan n-gram. Contoh aplikasi n-gram pada kata “program” dengan trigram (n=3) berupa “pro”, “rog”, “ogr”, “gra”, “ram”. Kelebihan fastText adalah waktu proses yang relatif cepat. FastText terbukti efektif dalam klasifikasi teks seperti *sentiment analysis* [32]. Kombinasi CNN dan fastText juga terbukti lebih akurat dalam klasifikasi sentimen dibandingkan menggunakan Word2Vec [33].

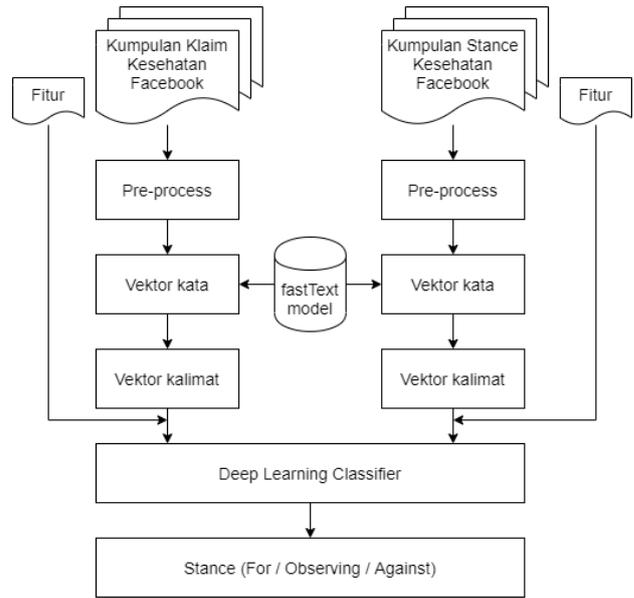
Pada bidang *sentence embedding*, Kiros memperkenalkan *Skip-thought vector*, dimana sebuah kalimat akan ditebak berdasarkan kalimat disekitarnya, seperti konsep *Skip-gram* [34]. Permasalahan durasi pelatihan yang lama dijawab dengan *Quick-thought vectors* [35]. Penurunan waktu pelatihan ini dikarenakan penentuan kalimat berikutnya yang menggunakan *classifier*, bukan *encoder-decoder RNN* seperti *Skip-thought*.

Namun, pembentukan vektor kalimat dengan metode sederhana seperti rata-rata dari *word embedding* juga dapat menghasilkan performa yang bagus [36]. Dalam penelitian Wieting [37], model *paragram-phrase* yang dibentuk dengan rata-rata dari vektor kata mampu mengalahkan model LSTM dalam tugas pengenalan similaritas teks.

III. METODOLOGI PENELITIAN

Proses penelitian ini dimulai dari pengumpulan data, pembentukan vektor kata dan kalimat, pengambilan fitur kalimat dan *post*, dan *deep learning classifier* untuk

klasifikasi *stance*. Gambar 1 menjabarkan alur sistem penelitian ini.



Gambar 1. Alur Sistem Penelitian

Penelitian ini sendiri akan dibatasi dengan hanya mengambil data dari *post* Facebook bertopik kesehatan. Peneliti hanya akan mengambil judul artikel yang *dishare*, judul pada gambar di *post*, atau teks judul pada *post* seperti pada Gambar 2. Panjang dari teks tersebut adalah satu kalimat atau lebih. Klasifikasi akan dilakukan hanya dengan CNN, LSTM, dan Bi-LSTM. Kelas klasifikasi sendiri terdiri dari tiga *stance*, yaitu: *for*, *observing* dan *against*.



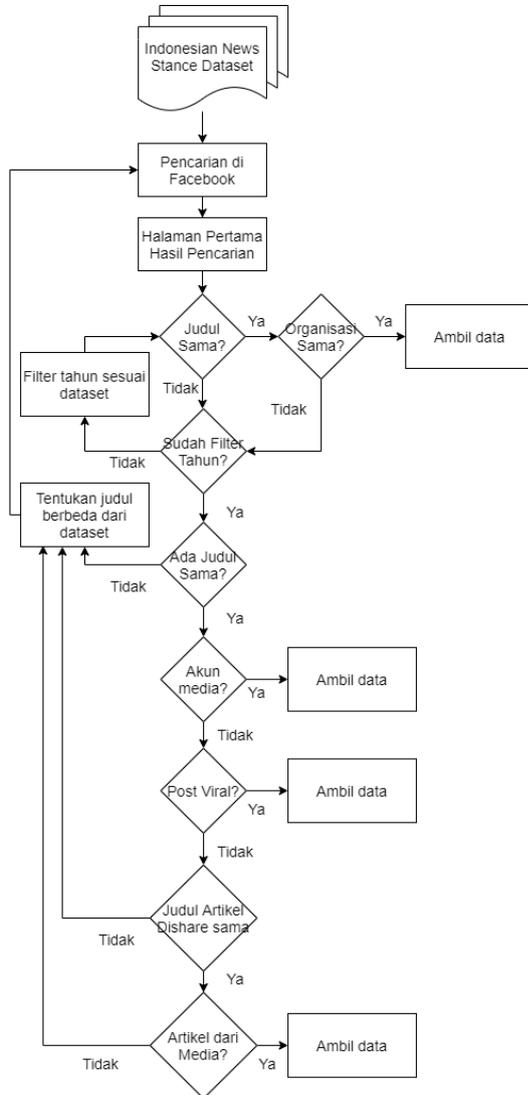
Gambar 2. Contoh Pengambilan Teks dan Post (Kotak Merah)

A. Dataset

Kalimat klaim maupun *stance* didasarkan pada dataset *Indonesian News Stance Dataset* [16]. Dalam penelitian ini, peneliti mencari *post* klaim dan *post* reaksi di Facebook berdasar kalimat klaim dan kalimat-kalimat reaksinya. Saat tidak ditemukan judul *post* Facebook yang berisi teks-teks tersebut, data yang diambil berupa judul-judul *post* Facebook dengan topik lain yang masih termasuk dalam bidang kesehatan. Hal ini mengakibatkan perubahan 350 kalimat klaim (70%) dari total 500 kalimat klaim di *Indonesian News Dataset*. Peneliti juga menambahkan informasi jumlah *like* dan *comment* pada setiap *post*.

Pengambilan data dilakukan secara manual di Facebook. Data yang dicari adalah judul artikel/judul gambar/judul *post* Facebook bersifat klaim, tanggal post klaim, URL *post* klaim, jumlah *like* pada *post*, dan jumlah *comment*. Data

yang dicari pada *post stance* sama dengan *post* klaim dengan penambahan data *stance*. Penentuan klaim dan *stance* dilakukan berdasar opini pengambil data. Data disimpan dalam bentuk file .CSV dan dipisah menjadi kolom-kolom tersendiri. Proses pengambilan data ini dapat diilustrasikan sebagai *flowchart* dalam gambar 3 berikut ini.



Gambar 3. Alur Sistem Pemilihan Post untuk Dataset

Proses pengambilan data *post* klaim dan *post* terkait beserta *stance* diawali dengan pencarian di kotak pencarian Facebook. *Query* pencarian adalah kalimat judul pada *Indonesian News Stance Dataset* dengan nama organisasi pembuat artikel tersebut. Contoh, kalimat pencarian “liputan6 Efek Buruk Minum Susu Kental Manis Tiap Hari” berasal dari artikel liputan6 yang berjudul “Efek Buruk Minum Susu Kental Manis Tiap Hari”. Bila ditemukan *post* berjudul artikel tersebut dari akun organisasi yang sama, maka akan diketik data yang dibutuhkan ke dalam file .CSV.

Bila tidak ditemukan *post* dengan kalimat tersebut pada halaman hasil pertama, pengambil data akan melakukan ulang pencarian dengan pemilihan *filter* tahun *post* sesuai dengan data tahun *post* pada *Indonesian News Stance Dataset*. Bila masih tidak ditemukan *post* dengan judul dan organisasi yang sama, maka dapat dilakukan satu dari dua solusi berikut: dilakukan pengambilan *post* dengan judul

yang sama tapi dengan organisasi atau *user* yang berbeda, atau pengambil data mencari topik kesehatan baru di kotak pencarian di Facebook.

Pengambil data memilih *post* hasil pencarian bertopik baru dengan prioritas utama *post* milik akun organisasi media, mulai dari media berskala nasional, hingga daerah. Hal ini dilakukan untuk memperbesar kemungkinan mendapat *post* dengan reaksi besar. Prioritas kedua adalah *post* perorangan yang *viral*, terlihat dari jumlah *like* yang bisa mencapai satu juta lebih. Prioritas ketiga adalah *post* berjudul sama dengan kalimat topik pencarian milik akun perorangan. Berikut ini, pada Tabel II, adalah salah satu contoh pasangan judul klaim dan judul berkaitan yang didapat.

TABEL II
CONTOH KLAIM DAN STANCE

Judul Klaim	Judul Reaksi	Stance
Dua Studi Ini Buktikan Manfaat Rokok Elektrik untuk Berhenti Merokok	Manfaat Vaping untuk Kesehatan yang Perlu Anda Ketahui	for
	Riset tiga tahun dari Italia pastikan rokok elektrik tidak bahaya	for
	Vape Alias Rokok Elektrik, Apakah Bahaya Bagi Tubuh?	observing
	Kelebihan dan Bahaya Rokok Elektrik untuk Kamu yang Masih Mahasiswa!	observing
	5 Bahaya yang Mengintai di Balik Nikmatnya Vape	against
	Bahaya dan Efek Samping Vape yang Negatif Mengintai Para Penggunaanya	against

Secara total, peneliti mendapat 3.500 judul *post* Facebook bertopik kesehatan yang berbahasa Indonesia. Dari data tersebut, 500 kalimat adalah kalimat klaim. Setiap kalimat klaim memiliki beberapa kalimat reaksi, yaitu: dua kalimat *stance for* (setuju dengan kalimat klaim), dua kalimat *stance observing* (netral terhadap kalimat klaim), dan dua kalimat *against* (bertentangan dengan kalimat klaim) yang juga diambil dan diberi label secara manual. Distribusi data dapat dilihat pada Tabel III.

TABEL III
DISTRIBUSI DATA DALAM DATASET PENELITIAN

Jenis Data	Jumlah
Klaim	500
For	1000
Observing	1000
Against	1000
Total	3500

B. Pembentukan Vektor Kata dan Kalimat

Embedding kata dan kalimat dilakukan untuk memetakan nilai vektor berdasarkan *embedding* fastText. Sebelum dipetakan, setiap kalimat akan melalui *pre-processing* terlebih dahulu.

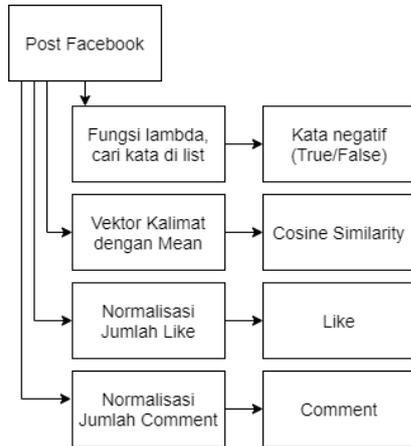
Pre-processing dimulai dengan *case folding*, tokenisasi dengan *word_tokenize* dari nltk, dan *filtering* untuk membuang kata-kata tidak penting seperti “yang”, ”untuk”, ”pada”, ”ke” dan lain-lain berdasar *stopwords* dari Tala [38]. Karakter spesial juga akan dihapus kecuali tanda tanya, tanda hubung, dan 0-9.

Penelitian ini menggunakan *pre-trained* fastText dengan dimensi 300. Dalam setiap kalimat, *token* kata akan diubah

menjadi vektor berdimensi 300 berdasar vektor *pre-trained* fastText cc.id.300.bin.

Semua kalimat akan melalui proses *padding* dengan vektor 0 sehingga semua kalimat akan memiliki besar matriks yang sama dengan kalimat dengan jumlah kata terbanyak. Setelah itu, vektor kalimat klaim dibentuk dengan menghitung rata-rata dari setiap vektor kata dalam kalimat.

C. Pengambilan Fitur Kalimat dan Post



Gambar 4. Pembentukan Fitur

Pembentukan fitur secara garis besar dapat dilihat pada gambar 4. Pengambilan fitur kata negatif dan *cosine similarity* dilakukan pada setiap kalimat klaim dan *stance*. Pada aspek fitur kata negatif, peneliti menambah data Boolean mengenai ada atau tidaknya kata-kata negatif dalam kalimat klaim dan *stance*. Kata-kata negatif tersebut adalah 'tidak', 'bukan', 'jangan', 'tak', 'belum', 'mitos', 'hoaks', dan 'hoax'. Penentuan kata-kata negatif berdasar pada penelitian sebelumnya [16], dimana setiap vektor kalimat akan diperpanjang dengan menambahkan nilai Boolean tadi.

Cosine similarity digunakan untuk mengetahui kemiripan vektor dengan mengukur sudut kosinus dari kedua vektor tersebut. *Cosine similarity* akan menghasilkan nilai 0 hingga 1, dimana hasil yang semakin dekat dengan 1 menunjukkan dua vektor yang memiliki banyak kemiripan. Terlihat pada rumus (1), *cosine similarity* memiliki dua komponen yang dalam penelitian ini adalah kalimat klaim dan kalimat *stance*.

$$cosine\ similarity = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

Pada aspek fitur *post*, peneliti mengambil informasi jumlah *like* dan jumlah komentar setiap *post*. Setelah dinormalisasi, jumlah *like* dan jumlah komentar akan ditambahkan ke vektor kalimat.

D. Fitur Kalimat dan Fitur Media Sosial

Dalam penelitian *stance classification*, Chen [21] memanfaatkan berbagai *metadata* untuk membantu mengetahui *stance* dari *tweet*. Informasi *user* adalah data yang paling penting, diikuti oleh informasi topik dan komentar. Pada penelitian Lozano [19], *metadata* yang

diambil adalah terbagi menjadi empat kriteria, yaitu: 1) jumlah tanda baca dan kata negasi pada *tweet*, 2) jumlah status, *follower*, dan *tweet* favorit *user*, 3) status verifikasi *user*, 4) apakah *tweet* termasuk *tweet reply* dan seberapa panjang thread *tweet* tersebut. Selain itu, Lozano juga menerapkan beberapa aturan dalam proses pelabelan *stance* pada *tweet*. Aturan pertama adalah pemberian label *deny* bila *tweet* mengandung frasa 'not believable'. Aturan kedua adalah pemberian label *support* bila terdapat URL dan tidak mengandung @-mention.

Pada penelitian ini, dalam konteks Facebook, peneliti mengambil *metadata* ada atau tidaknya tanda tanya, jumlah *like*, dan jumlah *comment*. Peneliti juga menentukan beberapa kata negatif yang diasosiasikan dengan *stance against*.

E. Parameter Penting dalam Classifier

Beberapa penelitian tentang parameter penting pada CNN [39] dan LSTM [36]. Pada CNN, ada beberapa parameter yang harus diperhatikan, yaitu: 1) penggunaan representasi vektor kata, 2) *filter region size* yang perlu disesuaikan terhadap penugasannya, 3) peningkatan nilai *feature maps* akan meningkatkan waktu pelatihan model, 4) *1-max pooling* memberikan performa terbaik, dan 5) regularisasi tidak berdampak signifikan terhadap performa.

Pada LSTM, akurasi LSTM sangat dipengaruhi oleh jumlah *token*, pengaturan dimensi disesuaikan dengan tujuan, dan penambahan *hidden unit* yang terkadang justru memperburuk akurasi.

F. Deep Learning Classifier

• Model Classifier CNN

Pemrosesan vektor dengan CNN pada algoritme 7.1 dilakukan dengan beberapa tahap. Pertama, vektor kalimat klaim akan diproses dengan filter dari CNN. Ukuran dari filter ini ditentukan pada saat pembuatan lapis convolutional 1D. Dalam kasus teks, nilai kernel menentukan jumlah baris filter, dan nilai *filters* menjadi panjang filter. Penelitian ini menggunakan *kernel_size* = 1 sehingga filter memproses per 1 kalimat. Setelah menentukan ukuran filter, perhitungan jumlah dari produk per elemen. Dari hasil tersebut, diambil nilai maksimal dengan *1-max pooling*. Program kemudian mengubah *shape* dengan *flatten* sehingga matriks hasil pooling menjadi satu vektor. Vektor tersebut kemudian diproses dengan lapis *dense* untuk mendapat dimensi yang sesuai dengan panjang *one-hot label* dan *softmax* untuk mencari probabilitas *stance* tertinggi pada matriks pasangan vektor tadi.

Parameter *classifier* CNN yang digunakan adalah sebagai berikut: 1) lapis *convolutional 1D* dengan *filters*=304, *kernel_size*=1, aktivasi ReLU, 2) lapis *dropout* sebesar 0,5, 3) lapis *max pooling 1D* dengan *pool size* sebesar satu, 4) lapis *flatten*, 5) lapis *dense* dengan output 50 unit dan aktivasi ReLU, dan 6) lapis *dense* output 3 unit dengan *softmax* untuk mengetahui probabilitas *stance*. Model dibentuk dengan *optimizer* Adam dengan *learning rate* sebesar 0,001.

• Model Classifier LSTM

Pemrosesan vektor dengan LSTM dilakukan dengan menginput vektor gabungan klaim ke dalam lapis LSTM ke

dalam *input gate*. LSTM memiliki sistem *gate* untuk menangani masalah *vanishing gradient*. Sistem proses LSTM terdiri dari lima modul, yaitu: *input gate*, *new memory cell*, *forget gate*, *final memory generation*, dan *output gate*. Dalam *input gate*, program membuat *memory* baru dengan menghitung vektor *input* tadi dengan *activation function* dan melihat *hidden state* dari LSTM sebelumnya, yang saat ini masih tidak ada. Sementara itu, *forget gate* menilai apakah *cell* sebelumnya diperlukan untuk perhitungan *cell* sekarang dengan melihat vektor *input* dan *hidden state* sebelumnya. Nilai dari kedua proses sebelumnya kemudian dijumlahkan dalam *final memory cell* (Ct). Setelah itu, *output gate* menentukan seberapa besar *memory* Ct yang perlu digunakan dalam langkah waktu saat ini. Nilai *output* dari *cell* ini adalah hasil perkalian dari *output gate* dan Ct yang telah melalui lapis *activation function*. Proses ini diulang Kembali untuk *input* vektor gabungan judul yang berkaitan dengan klaim pada langkah waktu berikutnya dengan *hidden state* dari proses dengan vektor gabungan klaim tadi. *Output* dari *cell* ini yang akan diproses ke lapis *dense* dan lapis *output softmax* untuk mendapat *stance*.

- Model Classifier Bi-LSTM

Bi-LSTM memiliki alur yang mirip dengan LSTM, hanya bedanya terdapat satu rangkaian LSTM tambahan yang menghitung dari input paling terakhir hingga input paling awal (*backward LSTM*). Sehingga terdapat dua rangkaian LSTM, *forward LSTM* dan *backward LSTM*. Algoritme classifier Bi-LSTM dapat dilihat pada algoritme 7.3. Langkah pertama dari algoritme, vektor gabungan klaim akan diinputkan terlebih dahulu ke dalam lapis *forward LSTM*. Langkah-langkah ini sama dengan langkah 01-10 pada algoritme 7.2. Program juga akan menjalankan *backward LSTM*. Input langkah waktu pertama adalah vektor terakhir, yaitu vektor gabungan judul berkaitan. Vektor ini diproses Dalam *input gate*, program membuat *memory* baru dengan menghitung vektor *input* tadi dengan *activation function* dan melihat *hidden state* dari LSTM sebelumnya, yang saat ini masih tidak ada. Sementara itu, *forget gate* menilai apakah *cell* sebelumnya diperlukan untuk perhitungan *cell* sekarang dengan melihat vektor *input* dan *hidden state* sebelumnya. Nilai dari kedua proses sebelumnya kemudian dijumlahkan dalam *final memory cell* (Ct). Setelah itu, *output gate* menentukan seberapa besar *memory* Ct yang perlu digunakan dalam langkah waktu saat ini. Nilai *output* dari *cell* ini adalah hasil perkalian dari *output gate* dan Ct yang telah melalui lapis *activation function*. Proses ini diulang untuk *input* vektor gabungan klaim dengan *hidden state* dari proses dengan vektor gabungan judul berkaitan tadi. *Output* dari *cell backward LSTM* ini yang akan diproses ke lapis *dense* dan lapis *output softmax* untuk mendapat *stance*, bersamaan dengan *output* dari *cell forward LSTM*.

Pada *classifier LSTM* dan *Bi-LSTM*, pengaturan parameter yang digunakan adalah: 1) lapis *LSTM/Bi-LSTM* dengan dua langkah waktu, *input* dengan dimensi 304 dan *recurrent dropout* sebesar 0.5, dan *output* 200 unit 2) lapis *dense* dengan aktivasi *ReLU* dan *output* 100 unit, dan 3) lapis *dense output* tiga unit dengan aktivasi *softmax*. Model *LSTM* dan *Bi-LSTM* ini juga dibentuk dengan *optimizer*

Adam dengan *learning rate default* sebesar 0,001.

G. Sistem Evaluasi Performa

Performa model dilihat berdasarkan *F1 score* setiap kelas *stance* dan *F1 macro*. *F1 score* diperoleh dengan menghitung *harmonic mean* dari *precision* dan *recall*, seperti pada rumus 2. Nilai *F1 score* memiliki rentang 0 hingga 1 dimana nilai 1 menunjukkan akurasi maksimal.

$$F1 = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad (2)$$

Precision, dalam konteks klasifikasi, pada rumus 8.2, adalah nilai dari pembagian *true positive* (tp) dengan jumlah dari *true positive* (tp) dan *false positive* (fp). Rumus 3 berikut menunjukkan rumus *precision*.

$$Precision = \frac{tp}{tp + fp} \quad (3)$$

Sementara itu, *recall*, dalam konteks klasifikasi, adalah nilai dari pembagian *true positive* (tp) dengan jumlah dari *true positive* (tp) dan *false negative* (fn). Rumus 4 berikut menunjukkan rumus *recall*.

$$Recall = \frac{tp}{tp + fn} \quad (3)$$

Istilah *true positive* (tp), *true negative* (tn), *false positive* (fp), dan *false negative* (fn) dapat dijabarkan sebagai berikut.

Istilah *true* dan *false* dalam rumus-rumus tersebut mengacu pada kebenaran sebuah prediksi. Sebagai contoh, *classifier* memprediksi sebuah judul memiliki *stance for* terhadap sebuah klaim. Bila label judul tersebut (data Y) pada *dataset* adalah *for*, maka hasil prediksi bersifat *true*. Bila data Y pada *dataset* adalah *observing* atau *against*, maka hasil prediksi bersifat *false*.

Sementara itu, istilah *positive* dan *negative* mengacu pada hasil prediksi yang diharapkan. Contoh, saat menghitung *precision stance observing*, maka semua prediksi yang menghasilkan *stance observing* masuk dalam kategori *positive* dan sisanya, prediksi dengan hasil *stance for* dan *observing*, sebagai anggota kategori *negative*.

Penelitian ini menggunakan fungsi *classification_report* dari *sklearn.metrics* untuk mendapat tabel mengenai performa model. Aspek-aspek yang terdapat dalam tabel tersebut adalah nilai *precision*, nilai *recall*, *F1 score*, dan *macro average F1 score*.

IV. ANALISIS DAN PEMBAHASAN

Untuk mendapatkan arsitektur yang efektif dalam *stance classification*, peneliti menguji coba model dengan beberapa parameter dan metode yang berbeda. Dalam eksperimen, data uji coba adalah 20% dari jumlah data yang ada. Peneliti menggunakan *macro-average F1-score* untuk mengetahui akurasi prediksi semua label dan *F1-score* untuk mengetahui akurasi prediksi pada setiap label.

Dalam setiap uji coba, program akan menghitung *F1 score* dan *F1 macro* dari sepuluh *seed* dari model, sehingga terdapat 10 *classification report* dalam setiap uji coba. Hasil

uji coba adalah rata-rata *F1 score* dan *F1 macro* dari sepuluh *classification report* tersebut.

Aspek-aspek yang diuji cobakan dalam eksperimen ini, yaitu: 1) jenis *word embedding*, 2) pengaruh hasil pelatihan *word embedding*, 3) jenis *classifier*, 4) fitur kalimat dan media sosial, 5) *dropout* dan jenis *activation*, dan 6) *hidden units*.

A. Jenis Word Embedding

Pada uji coba pengaruh jenis *word embedding*, peneliti membandingkan performa antara Word2Vec menggunakan vektor *pre-trained* Kyubyong yang berdimensi 300 (<https://github.com/Kyubyong/wordvectors>) dan fastText cc.id.300.bin yang juga berdimensi 300. Penentuan Word2Vec sebagai perbandingan mengacu pada penelitian sebelumnya yang menggunakan Word2Vec untuk mengklasifikasi pendapat di Twitter dengan metode pemodelan kalimat yang sama dengan penelitian ini [22].

Pada Tabel IV terlihat bahwa ada perbedaan akurasi sebesar 1,1% antara rata-rata *F1 macro* model dengan Word2Vec (52,7%) terhadap model dengan fastText (53,8%). Penggunaan Word2Vec paling akurat dengan LSTM hanya mencapai 55% dan penggunaan fastText paling akurat diperoleh dengan CNN dengan *F1 macro* 55,4%.

TABEL IV
HASIL UJI COBA JENIS WORD EMBEDDING

Pendekatan	<i>F1 for</i>	<i>F1 observing</i>	<i>F1 against</i>	<i>F1 Macro</i>
Word2Vec + CNN	0.454	0.654	0.488	0.532
Word2Vec + LSTM	0.503	0.659	0.488	0.550
Word2Vec + Bi-LSTM	0.453	0.624	0.424	0.500
Rata-rata	0.470	0.646	0.467	0.527
fastText + CNN	0.495	0.824	0.345	0.554
fastText + LSTM	0.492	0.801	0.294	0.529
fastText + Bi-LSTM	0.441	0.817	0.391	0.530
Rata-rata	0.476	0.814	0.343	0.538

B. Pemilihan Hasil Pelatihan Word Embedding

Terlihat pada Tabel V bahwa penggunaan vektor *pre-trained* harus disesuaikan dengan banyaknya fitur yang dipakai dalam penelitian. Bila hanya menggunakan vektor kalimat sebagai *input*, maka wiki.id.bin akan lebih akurat (54,7%) dibandingkan dengan cc.id.300.bin (53,8%). Namun, bila ada penambahan vektor fitur, seperti informasi kata negatif, *cosine similarity*, jumlah *like* dan *comment*, cc.id.300.bin akan menghasilkan model yang lebih akurat dengan *F1 macro* sebesar 60% dibandingkan dengan 50% dengan wiki.id.bin.

TABEL V
HASIL UJI COBA WORD EMBEDDING FASTTEXT PRE-TRAINED

Vektor Pre-Trained	<i>F1 for</i>	<i>F1 observing</i>	<i>F1 against</i>	<i>F1 Macro</i>
cc.id.300.bin				
Tanpa Fitur	0,476	0,814	0,343	0,538
Dengan Fitur	0,549	0,789	0,461	0,600
wiki.id.bin				
Tanpa Fitur	0,484	0,823	0,337	0,547
Dengan Fitur	0,536	0,628	0,406	0,500

C. Jenis Classifier

Dalam Tabel VI, terlihat hasil dari uji coba bila input hanya menggunakan vektor kalimat tanpa penambahan fitur. Ditemukan bahwa metode *classifier* CNN justru memberi *F1 macro* tertinggi dengan cc.id.bin (55,4%). Hasil yang hampir sama juga didapat dengan wiki.id.bin.

TABEL VI
HASIL UJI COBA METODE CLASSIFIER CNN, LSTM, DAN BI-LSTM, DENGAN FASTTEXT CC.ID.300.BIN DAN WIKI.ID.BIN

Classifier	<i>F1 for</i>	<i>F1 observing</i>	<i>F1 against</i>	<i>F1 Macro</i>
cc.id.300.bin				
CNN	0.495	0.824	0.345	0.554
LSTM	0.492	0.801	0.294	0.529
Bi-LSTM	0.441	0.817	0.391	0.530
wiki.id.bin				
CNN	0.512	0.807	0.345	0.553
LSTM	0.445	0.827	0.350	0.540
Bi-LSTM	0.496	0.834	0.315	0.547

D. Fitur Kalimat dan Post

Pada Tabel VII tampak bahwa adanya informasi keberadaan kata negatif dan *cosine similarity* mengakibatkan adanya perbaikan performa yang cukup besar. Akurasi Bi-LSTM meningkat sebesar 9,4% menjadi 62,4% dari 53% tanpa penambahan vektor fitur. Akurasi masing-masing aspek uji coba dengan CNN juga memberikan hasil yang tertinggi, kecuali pada fitur *kata* negatif saja dan fitur *cosine similarity* saja. Nilai *F1 Macro* tertinggi didapat oleh CNN dengan 62,9%. Model Bi-LSTM juga memperoleh hasil yang cukup serupa dengan 62,4%. Nilai *F1 Macro* dari pemodelan kalimat dengan rata-rata dan *classifier* CNN dan Bi-LSTM juga memberikan akurasi yang lebih tinggi daripada hasil uji coba sebesar 60,66 % pada penelitian sebelumnya [16]. Akan tetapi, penambahan fitur *likes* dan *comment* justru memperburuk akurasi dari setiap *classifier*. CNN mengalami penurunan menjadi 60% dan Bi-LSTM menjadi 61,3%.

TABEL VII
HASIL UJI COBA VEKTOR KALIMAT DAN FITUR KALIMAT DAN POST

Classifier	<i>F1 for</i>	<i>F1 observing</i>	<i>F1 against</i>	<i>F1 Macro</i>
Tanpa Vektor Fitur				
CNN	0.495	0.824	0.345	0.554
LSTM	0.492	0.801	0.294	0.529
Bi-LSTM	0.441	0.817	0.391	0.530
Dengan Vektor Fitur Kata Negatif				
CNN	0.579	0.778	0.443	0.600
LSTM	0.552	0.785	0.468	0.602
Bi-LSTM	0.539	0.807	0.512	0.618
Dengan Vektor Fitur Cosine Similarity				
CNN	0.440	0.745	0.432	0.539
LSTM	0.513	0.774	0.312	0.535
Bi-LSTM	0.463	0.793	0.381	0.547
Dengan Vektor Fitur Kata Negatif + Cosine Similarity				
CNN	0.588	0.791	0.511	0.629
LSTM	0.530	0.802	0.481	0.605
Bi-LSTM	0.543	0.822	0.510	0.624
Dengan Vektor Fitur Kata Negatif + Cosine Similarity + Likes dan Comment				
CNN	0.561	0.802	0.433	0.600
LSTM	0.534	0.781	0.451	0.588
Bi-LSTM	0.553	0.785	0.500	0.613

E. Dropout dan Jenis Activation

Dalam Tabel VIII, terbukti bahwa akurasi tertinggi sebesar 62,9% dicapai dengan nilai *dropout* 0,5 dengan *activation* ReLU. Peningkatan nilai *dropout* tampak menurunkan akurasi dari semua model, terutama pada model *classifier* CNN yang hanya menghasilkan *F1 macro* sebesar 37,6%.

TABEL VIII
HASIL UJI COBA DROPOUT DAN ACTIVATION ReLU

Classifier	F1 for	F1 observing	F1 against	F1 Macro
Dropout 0.5 + ReLU				
CNN	0.588	0.791	0.511	0.629
LSTM	0.530	0.802	0.481	0.605
Bi-LSTM	0.543	0.822	0.510	0.624
Dropout 0.75 + ReLU				
CNN	0.591	0.780	0.405	0.592
LSTM	0.538	0.798	0.452	0.597
Bi-LSTM	0.506	0.808	0.511	0.609
Dropout 0.99 + ReLU				
CNN	0.108	0.484	0.172	0.253
LSTM	0.539	0.792	0.384	0.572
Bi-LSTM	0.572	0.791	0.380	0.582

Terbukti pula bahwa *activation function* Leaky ReLU hanya mampu meningkatkan efektifitas sebesar 0,4% pada model LSTM dibandingkan dengan model yang menggunakan ReLU. *F1 Macro* model justru menurun untuk model lain. *F1 Macro* CNN turun 1,4% menjadi 61,5% dan Bi-LSTM turun 2,8% menjadi 59,6%. Leaky ReLU sendiri paling akurat apabila digunakan dengan nilai *dropout* 0,5. Kombinasi ini menghasilkan *F1 macro* tertinggi pada model *classifier* CNN sebesar 61,5% seperti yang tampak pada Tabel IX.

TABEL IX
HASIL UJI COBA DROPOUT DAN ACTIVATION LEAKY ReLU

Classifier	F1 for	F1 observing	F1 against	F1 Macro
Dropout 0.5 + Leaky ReLU				
CNN	0.568	0.780	0.504	0.615
LSTM	0.533	0.809	0.482	0.609
Bi-LSTM	0.519	0.778	0.495	0.596
Dropout 0.75 + Leaky ReLU				
CNN	0.587	0.783	0.406	0.593
LSTM	0.534	0.805	0.485	0.609
Bi-LSTM	0.534	0.772	0.480	0.595
Dropout 0.99 + Leaky ReLU				
CNN	0.138	0.565	0.270	0.324
LSTM	0.555	0.795	0.362	0.571
Bi-LSTM	0.543	0.753	0.395	0.563

F. Hidden Units

Dalam Tabel X, akurasi dapat ditingkatkan dengan penambahan *hidden units*. Pada model *classifier* LSTM, akurasi bertambah 3% dari 60,2% menjadi 63%. Model *classifier* Bi-LSTM pun mengalami peningkatan sebesar 0,6% menjadi 64%.

Dibandingkan dengan penelitian terakhir mengenai *stance classification* dengan *deep learning* [16], penelitian ini memperoleh peningkatan *F1-score* sebesar 3,3% menjadi 64% dari 60,66%. Perbandingan ini hanya berdasarkan hasil

uji coba dengan metode pemodelan kalimat rata-rata dan *classifier* LSTM pada penelitian sebelumnya. Klaim dan judul terkait dalam *dataset* penelitian sebelumnya berjumlah 3.954 judul. Sementara itu, penelitian ini memiliki total 3.500 judul dengan jumlah klaim yang sama dengan penelitian sebelumnya yakni 500. Kedua *dataset* sama-sama memiliki tiga kelas *output* dan berdistribusi sekitar 30% per kelas.

TABEL X
HASIL UJI COBA HIDDEN UNITS

Hidden Units	F1 for	F1 observing	F1 against	F1 Macro
LSTM				
200	0.589	0.803	0.503	0.630
100	0.570	0.794	0.440	0.602
50	0.566	0.792	0.496	0.619
Bi-LSTM				
200	0.588	0.811	0.524	0.640
100	0.562	0.811	0.523	0.634
50	0.578	0.809	0.522	0.637

Akan tetapi, *F1-score* model sebesar 64% ini masih dibawah *F1-score* penelitian Kochkina [22] sebesar 78,4%. Penelitian tersebut menggunakan *dataset* 5568 *tweet*, dengan empat kelas *output*. Distribusi *tweet* per kelas S:D:Q:C adalah 1004:415:464:3685.

V. KESIMPULAN

Stance classification berbahasa Indonesia lebih akurat dilakukan dengan fastText dibandingkan dengan Word2Vec. Word *embedding* fastText sebaiknya digunakan dengan vektor *pre-trained* wiki.id.bin untuk *stance classification* tanpa fitur. Akan tetapi, gunakan cc.id.300.bin bila terdapat informasi fitur kalimat dan *post*. Dalam memilih *classifier*, CNN dan Bi-LSTM adalah dua pilihan terbaik. Terutama bila dilakukan pengaturan *hidden units* pada Bi-LSTM. Penambahan fitur sendiri tidak cukup dengan mencatatkan informasi tanda tanya saja. Perlu ditambahkan informasi ada atau tidaknya kata negatif dalam setiap kalimat dan nilai *cosine similarity* kalimat klaim dan reaksi. Akan tetapi, tidak perlu menambahkan informasi jumlah *like* dan *comment* pada *post* yang justru memperburuk akurasi model. Nilai *dropout* yang paling efektif adalah 0,5 dengan *activation function* ReLU. Diperlukan pula pengaturan *hidden units* agar tidak terlalu sedikit sehingga model dapat menjadi lebih akurat. Penelitian ini dapat dikembangkan lagi pada eksplorasi efek fitur kalimat dan *metadata* media sosial lain terhadap performa. Pemahaman makna kalimat juga dapat dikembangkan dengan pendekatan atensi dan pengaruhnya pada akurasi klasifikasi *stance*.

DAFTAR PUSTAKA

- [1] D. Goldie, M. Linick, H. Jabbar, and C. Lubienski, "Using Bibliometric and Social Media Analyses to Explore the 'Echo Chamber' Hypothesis," *Educ. Policy*, vol. 28, no. 2, pp. 281–305, 2014, doi: 10.1177/0895904813515330.
- [2] S. Jacobson, E. Myung, and S. L. Johnson, "Open media or echo chamber: the use of links in audience discussions on the Facebook Pages of partisan news organizations," *Inf. Commun. Soc.*, vol. 19, no. 7, pp. 875–891, 2016, doi: 10.1080/1369118X.2015.1064461.
- [3] J. J. Van Bavel and A. Pereira, "The Partisan Brain: An Identity-Based Model of Political Belief," *Trends Cogn. Sci.*, vol. 22, no. 3,

- pp. 213–224, 2018, doi: 10.1016/j.tics.2018.01.004.
- [4] F. Zollo and W. Quattrociochi, “Misinformation Spreading on Facebook,” pp. 177–196, 2018, doi: 10.1007/978-3-319-77332-2_10.
 - [5] C. Shao, G. L. Ciampaglia, O. Varol, K. C. Yang, A. Flammini, and F. Menczer, “The spread of low-credibility content by social bots,” *Nat. Commun.*, vol. 9, no. 1, 2018, doi: 10.1038/s41467-018-06930-7.
 - [6] H. Allcott and M. Gentzkow, “Social Media and Fake News in the 2016 Election,” *J. Econ. Perspect.*, vol. 31, no. 2, pp. 211–236, 2017, doi: 10.1257/jep.31.2.211.
 - [7] C. Reuter, S. Stieglitz, and M. Imran, “Social media in conflicts and crises,” *Behav. Inf. Technol.*, vol. 39, no. 3, pp. 241–251, 2020, doi: 10.1080/0144929X.2019.1629025.
 - [8] M. Roy, N. Moreau, C. Rousseau, A. Mercier, A. Wilson, and L. Atlani-Duault, “Ebola and Localized Blame on Social Media: Analysis of Twitter and Facebook Conversations During the 2014–2015 Ebola Epidemic,” *Cult. Med. Psychiatry*, vol. 44, no. 1, pp. 56–79, 2020, doi: 10.1007/s11013-019-09635-8.
 - [9] S. Sommariva, C. Vamos, A. Mantzarlis, L. U. L. Dào, and D. Martinez Tyson, “Spreading the (Fake) News: Exploring Health Messages on Social Media and the Implications for Health Professionals Using a Case Study,” *Am. J. Heal. Educ.*, vol. 49, no. 4, pp. 246–255, 2018, doi: 10.1080/19325037.2018.1473178.
 - [10] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, “Stance and Sentiment in Tweets,” vol. 0, no. 0, 2016, [Online]. Available: <http://arxiv.org/abs/1605.01655>.
 - [11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *ICLR Work.*, pp. 1–12, 2013, [Online]. Available: <http://arxiv.org/abs/1301.3781>.
 - [12] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Suksa, “Natural Language Processing (Almost) from Scratch,” *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011, doi: 10.1109/JMLR.2011.00050.
 - [13] Y. Kim, “Convolutional neural networks for sentence classification,” *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1746–1751, 2014.
 - [14] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference,” *Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process.*, pp. 632–642, 2015, doi: 10.18653/v1/d15-1075.
 - [15] R. Jannati, R. Mahendra, C. W. Wardhana, and M. Adriani, “Stance Classification Towards Political Figures on Blog Writing,” *Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018*, pp. 96–101, 2019, doi: 10.1109/IALP.2018.8629144.
 - [16] E. I. Setiawan, A. Ferdianto, J. Santoso, Y. Kristian, S. Sumpeno, and M. H. Purnomo, “Analisis Pendapat Masyarakat terhadap Berita Kesehatan Indonesia menggunakan Pemodelan Kalimat berbasis LSTM (Indonesian Stance Analysis of Healthcare News using Sentence Embedding,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 1, pp. 8–17, 2020.
 - [17] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.
 - [18] A. Graves and J. Schmidhuber, “Framewise Phoneme Classification with Bidirectional LSTM Networks,” *Proceedings. 2005 IEEE Int. Jt. Conf. Neural Networks, 2005.*, vol. 4, pp. 2047–2052, 2005.
 - [19] M. Garcia Lozano, H. Lilja, E. Tjörnhámmar, and M. Karasalo, “Mama Edha at SemEval-2017 Task 8: Stance Classification with CNN and Rules,” *Proc. 11th Int. Work. Semant. Eval.*, pp. 481–485, 2018, doi: 10.18653/v1/s17-2084.
 - [20] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3104–3112, 2014.
 - [21] W.-F. Chen and L.-W. Ku, “UTCNN: a Deep Learning Model of Stance Classification on Social Media Text,” *Proc. COLING 2016, 26th Int. Conf. Comput. Linguist. Tech. Pap.*, pp. 1635–1645, 2016, [Online]. Available: <http://arxiv.org/abs/1611.03599>.
 - [22] E. Kochkina, M. Liakata, and I. Augenstein, “Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM,” no. 2016, pp. 475–480, 2018, doi: 10.18653/v1/s17-2083.
 - [23] K. Shalini, M. Anand Kumar, and K. Soman, *Deep-Learning-Based Stance Detection for Indian Social Media Text*, vol. 545. Springer Singapore, 2019.
 - [24] M. Thomas, B. Pang, and L. Lee, “Get out the vote: Determining support or opposition from Congressional oor-debate transcripts,” *In Proc. of EMNLP*, no. July, pp. 327–335, 2006.
 - [25] K. Hasan and V. Ng, “Stance Classification of Ideological Debates : Data , Models , Features , and Constraints,” *Proc. SIGDIAL 2013 Conf.*, no. October, pp. 1348–1356, 2013.
 - [26] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, “A simple but tough-to-beat baseline for the Fake News Challenge stance detection task,” pp. 1–6, 2017, [Online]. Available: <http://arxiv.org/abs/1707.03264>.
 - [27] A. Hanselowski *et al.*, “A Retrospective Analysis of the Fake News Challenge Stance Detection Task,” 2018, [Online]. Available: <http://arxiv.org/abs/1806.05180>.
 - [28] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, “Stance Detection with Bidirectional Conditional Encoding,” pp. 876–885, 2016, doi: 10.18653/v1/d16-1084.
 - [29] I. Habernal and I. Gurevych, “Which argument is more convincing? Analyzing and predicting convincings of Web arguments using bidirectional LSTM,” *54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap.*, vol. 3, pp. 1589–1599, 2016, doi: 10.18653/v1/p16-1150.
 - [30] D. Mrowca and E. Wang, “Stance detection for fake news identification,” pp. 1–12, 2017.
 - [31] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A Neural Probabilistic Language Model,” *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003, doi: 10.1080/1536383X.2018.1448388.
 - [32] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” *15th Conf. Eur. Chapter Assoc. Comput. Linguist. EACL 2017 - Proc. Conf.*, vol. 2, pp. 427–431, 2017, doi: 10.18653/v1/e17-2068.
 - [33] I. Santos, N. Nedjah, and L. de Macedo Mourelle, “Sentiment analysis using convolutional neural network with fastText embeddings,” pp. 1–5, 2018, doi: 10.1109/la-cci.2017.8285683.
 - [34] R. Kiros *et al.*, “Skip-thought vectors,” *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, no. 786, pp. 3294–3302, 2015.
 - [35] L. Logeswaran and H. Lee, “An efficient framework for learning sentence representations,” *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–16, 2018.
 - [36] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg, “Fine-grained analysis of sentence embeddings using auxiliary prediction tasks,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–13, 2019.
 - [37] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, “Towards universal paraphrastic sentence embeddings,” *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, pp. 1–19, 2016.
 - [38] F. Z. Tala, “A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia,” *M.Sc. Thesis, Append. D*, vol. pp, pp. 39–46, 2003.
 - [39] Y. Zhang and B. Wallace, “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification,” 2015, [Online]. Available: <http://arxiv.org/abs/1510.03820>.

Ernest Lim adalah lulusan S1 jurusan Arsitektur di Universitas Katolik Parahyangan pada tahun 2016. Ernest kemudian melanjutkan studi S2 Teknologi Informasi di STTS sejak tahun 2018.

Esther Irawati Setiawan menyelesaikan studi S1 di program Teknik Informatika STTS pada tahun 2006. Menyelesaikan studi master Teknologi Informasi STTS pada tahun 2010. Minat penelitian pada bidang *Social Network Analysis* dan *Web Mining*.

Joan Santoso menyelesaikan studi S1 Teknik Informatika di STTS pada tahun 2011. Menyelesaikan studi S2 Teknologi Informasi di STTS pada tahun 2013. Minat penelitian pada bidang *computational linguistics*, *information extraction*, *machine learning*, dan *big data processing*.

Pemanfaatan Asynchronous Advantage Actor-Critic Dalam Pembuatan AI Game Bot Pada Game Arcade

Evan Kusuma Susanto, *Teknik Informatika Sekolah Tinggi Teknik Surabaya*, Yosi Kristian, S.Kom., M.Kom., *Teknik Informatika Sekolah Tinggi Teknik Surabaya*

Abstrak— Asynchronous Advantage Actor-Critic (A3C) adalah sebuah algoritma deep reinforcement learning yang dikembangkan oleh Google DeepMind. Algoritma ini dapat digunakan untuk menciptakan sebuah arsitektur artificial intelligence yang dapat menguasai berbagai jenis game yang berbeda melalui *trial and error* dengan mempelajari tampilan layar game dan skor yang diperoleh dari hasil tindakannya tanpa campur tangan manusia. Sebuah network A3C terdiri dari Convolutional Neural Network (CNN) di bagian depan, Long Short-Term Memory Network (LSTM) di tengah, dan sebuah Actor-Critic network di bagian belakang. CNN berguna sebagai perangkum dari citra output layar dengan mengekstrak fitur-fitur yang penting yang terdapat pada layar. LSTM berguna sebagai pengingat keadaan game sebelumnya. Actor-Critic Network berguna untuk menentukan tindakan terbaik untuk dilakukan ketika dihadapkan dengan suatu kondisi tertentu. Dari hasil percobaan yang dilakukan, metode ini cukup efektif dan dapat mengalahkan pemain pemula dalam memainkan 5 game yang digunakan sebagai bahan uji coba.

Kata Kunci—Artificial Intelligence, Deep Reinforcement Learning, Machine Learning, Reinforcement Learning.

I. PENDAHULUAN

Perkembangan game yang pesat beserta perkembangan teknologi komputer yang juga pesat mendorong peneliti untuk merancang sebuah sistem yang dapat memainkan permainan seperti layaknya manusia [1]–[4]. Pengembangan sistem cerdas ini disebut *artificial intelligence* (AI). Tetapi, biasanya artificial intelligence yang dirancang tersebut biasanya hanya dapat memainkan 1 jenis permainan saja, yaitu permainan yang telah ditentukan oleh pembuatnya. Belum ada AI yang cukup fleksibel untuk memainkan berbagai macam jenis game yang berbeda, tetapi metode pengembangan AI yang berkembang pesat tidak menutup kemungkinan untuk kemunculan AI yang fleksibel.

Metode pengenalan citra dan ekstraksi fitur yang sedang berkembang adalah metode *deep learning*. Deep learning merupakan salah satu arsitektur *neural network* yang memungkinkan input dari suatu neural network berupa gambar, bukan fitur seperti neural network umumnya. Tidak seperti *shallow learning* yang perlu mengekstraksi fitur

secara manual, dan melakukan preprocessing yang banyak. Deep learning mampu mencari dan melatih fitur sendiri, dengan input *raw image*. Karena fitur juga ikut dilatih, maka memungkinkan didapatkannya fitur yang mungkin terlewat jika didesain manual [5].

Metode yang digunakan agar AI dapat belajar sendiri melalui trial and error adalah metode *reinforcement learning* [6]. Dalam penggunaannya, teknik dari reinforcement learning yang akan digunakan adalah *Actor-Critic* [7]. Actor-Critic menyediakan model matematika yang dapat digunakan untuk membantu menentukan langkah apa yang terbaik yang harus diambil oleh AI pada kondisi yang dihadapi.

Dengan adanya kedua metode tersebut, secara teori dengan waktu yang cukup lama dan data yang cukup besar maka dapat didesain sebuah arsitektur neural network yang dapat memainkan berbagai macam jenis game yang berbeda dan dapat belajar dengan sendirinya melalui trial and error.

II. REINFORCEMENT LEARNING

Reinforcement learning adalah mempelajari apa yang harus dilakukan (memetakan situasi ke aksi) untuk memaksimalkan reward yang diperoleh [6]. Pada reinforcement learning, komputer tidak diberi "supervisi" atau diberi panduan oleh user secara langsung. Komputer akan dihadapkan dengan sebuah environment dan belajar dari hasil interaksi antara komputer dengan environment tersebut. User tidak memberi panduan berupa jawaban yang diharapkan dari komputer secara langsung, tetapi user membiarkan komputer berinteraksi dengan environment dan akan memberi reward (imbalan) kepada komputer ketika komputer berhasil melakukan sesuatu hal yang baik dan memberi punishment (hukuman) ketika komputer melakukan suatu hal yang buruk.

Dasar dari algoritma-algoritma reinforcement learning berasal dari Markov Decision Process (MDP) [8]. Markov Decision Process merupakan sebuah formalisasi dari sequential decision making (pengambilan keputusan berurutan), di mana keputusan yang diambil akan mempengaruhi situasi dan reward berikutnya. Dalam MDP dibentuk sebuah model matematika ideal dari sebuah permasalahan reinforcement learning sehingga dapat dibuat pernyataan teoritis terhadap permasalahan tersebut. Dalam MDP terdapat beberapa konsep yang digunakan dalam reinforcement learning yaitu state, transition, action, reward, dan policy. Selain itu digunakan pula formula untuk

3 Oktober 2018 Sekolah Tinggi Teknik Surabaya

Evan Kusuma S, Departemen Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: evanks@rocketmail.com)

Yosi Kristian, Dosen Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: yosi@stts.edu)

mengkalkulasi ekspektasi reward maksimum yang dapat diperoleh. Formula ini adalah Persamaan Bellman (Bellman Equation) [8] yang dirumuskan sebagai berikut:

$$V(s) = \max_a (R(s, a) + \gamma \sum_{s'} T(s, a, s') V(s')) \quad (1)$$

Dalam reinforcement learning terdapat 3 jenis metode dalam menentukan policy terbaik. Metode pertama adalah value based learning. Dalam metode yang bersifat value based lebih diutamakan kalkulasi atau estimasi dari value function setiap state dalam environment. Kalkulasi value function dilakukan dengan menggunakan Persamaan Bellman. Dari value function ini akan dibentuk sebuah policy yang berorientasi untuk memaksimalkan value function yang diperoleh. Salah satu penerapan algoritma value based learning adalah TD-gammon [1], yaitu penggunaan TD learning untuk memainkan permainan backgammon. TD-gammon menjadi dasar bagi banyak penelitian reinforcement learning berikutnya.

Metode berikutnya adalah policy based learning. Pada policy based learning, policy dicari dan ditentukan secara langsung. Policy direpresentasikan sebagai kumpulan parameter (parameterized policy) yang akan dikalkulasi dan dimodifikasi sesuai dengan performa agen (total skor yang diterima) ketika berinteraksi dengan environment dengan policy yang diberikan. Akan dikalkulasi kombinasi parameter terbaik agar skor tertinggi dapat diperoleh. Beberapa cara yang dapat digunakan adalah dengan policy search, algoritma evolutionary [9], atau dengan policy gradient [10].

Metode policy gradient adalah salah satu teknik reinforcement learning di mana dilakukan optimasi terhadap parameterized policy terhadap ekspektasi reward dengan menggunakan gradien. Gradien digunakan untuk mengubah nilai parameter untuk memaksimalkan nilai ekspektasi skor maksimum dan dikalkulasi dengan cara mencari turunan dari cost function (ekspektasi reward maksimum) terhadap masing-masing parameter policy. Contoh dari pemanfaatan metode policy gradient adalah algoritma REINFORCE [11] dan penentuan policy untuk dapat membuat sebuah robot berjalan cepat [12].

III. ACTOR-CRITIC

Actor-critic merupakan jenis algoritma reinforcement learning yang merupakan gabungan dari metode policy based learning dan value based learning. Actor-critic terdiri dari 2 komponen, yaitu actor dan critic [7]. Actor adalah komponen yang membentuk policy, sedangkan critic adalah komponen yang memberikan evaluasi terhadap policy yang dibuat oleh actor.

Dalam metode yang bersifat value based lebih diutamakan kalkulasi atau estimasi dari value function (atau Q function) setiap state dalam environment. Kalkulasi value function dilakukan dengan menggunakan Persamaan Bellman. Dari value function ini akan dibentuk sebuah policy yang berorientasi untuk memaksimalkan value function yang diperoleh.

Pada policy based learning policy dicari dan ditentukan secara langsung. Policy akan dikalkulasi dan dimodifikasi

sesuai dengan performa agen (total skor yang diterima) ketika berinteraksi dengan environment dengan policy yang diberikan.

Metode policy based learning memiliki kelemahan berupa tidak adanya proses belajar dari pengalaman. Hal ini disebabkan karena pada setiap iterasi dilakukan perubahan nilai parameter berdasarkan nilai gradien pada iterasi tersebut, tanpa adanya pengaruh akumulasi dan konsolidasi dari pengalaman masa lalu. Sedangkan metode value based learning yang diutamakan adalah didapatkannya estimasi value function yang akurat dengan harapan policy yang baik dapat ditentukan dari hasil kalkulasi value function. Hal ini menyebabkan tidak adanya jaminan bahwa policy yang dihasilkan adalah policy yang paling optimal [7].

Metode actor-critic dirancang untuk menggabungkan kelebihan dari metode policy based learning dan value based learning [7]. Pada critic digunakan sebuah function approximation untuk estimasi nilai value function sebuah state. Hasil estimasi ini digunakan untuk update parameter policy pada actor menuju arah yang lebih baik. Dengan metode ini dapat dijamin tercapainya local optimum pada hasil policy (karena digunakannya metode gradien). Selain itu, kondisi optimal dapat tercapai lebih cepat karena digunakannya value function.

Terdapat berbagai macam jenis actor critic yang dapat dibentuk, mengingat banyaknya jenis metode estimasi value function yang dapat dilakukan, seperti metode TD learning [13] atau Q learning [14]. Salah satu varian dari metode actor-critic adalah advantage actor critic. Pada metode ini digunakan nilai advantage function dari sebuah aksi yang dilakukan pada sebuah state.

Advantage menyatakan seberapa baik sebuah tindakan apabila dilakukan pada suatu keadaan dibandingkan dengan aksi lain yang dapat dilakukan. Salah satu metode yang dapat digunakan untuk estimasi nilai advantage function adalah metode Generalized Advantage Estimation [15]. Dari metode ini dihasilkan estimasi advantage function dengan pengaturan bias dan variance yang diinginkan.

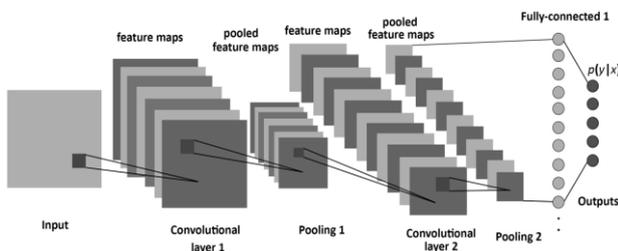
IV. CONVOLUTIONAL NEURAL NETWORK

Neural Network adalah salah satu algoritma dalam Machine Learning yang terinspirasi dari cara kerja otak manusia untuk menyelesaikan suatu masalah. Neural network tersusun dari beberapa layer artificial neuron yang saling terhubung satu dengan yang lain (fully connected) [16]. Neural network menjadi dasar pembentukan arsitektur lain yang lebih kompleks dan didesain secara khusus untuk dapat mengatasi permasalahan tertentu. Neural network biasa digunakan untuk menyelesaikan masalah yang menggunakan input berupa feature vector (matriks 1 dimensi), tetapi kurang baik untuk menerima dan mengolah input berbentuk gambar (matriks 2 atau 3 dimensi) dan input yang bersifat sequential (hasil saat ini dipengaruhi hasil di waktu lampau).

Deep Convolutional Neural Network (DCNN) adalah pengembangan dari neural network untuk permasalahan yang berhubungan dengan citra. Arsitektur dari DCNN ini sendiri didesain sedemikian sehingga menyerupai susunan korteks visual pada kucing dan monyet untuk dapat

mengolah input gambar berupa apa yang dilihat.

Misalnya sebuah neural network perlu memproses gambar dengan ukuran 80 x 80 pixel dengan 3 channel warna (RGB). Pada hidden layer pertama dalam neural network biasa setidaknya terdapat $80 * 80 * 3 = 19200$ bobot untuk sebuah neuron pada layer tersebut. Padahal untuk membuat network yang dapat menyelesaikan permasalahan kompleks dibutuhkan lebih dari sebuah neuron dan sebuah hidden layer. Konsep fully connectivity dalam kasus ini menggunakan terlalu banyak bobot yang dapat menyebabkan terjadinya overfitting, yaitu sebuah keadaan dimana sebuah neural network terlalu menyesuaikan diri dengan data yang dijumpai saat training, sehingga apabila diuji dengan data baru maka neural network tidak akan bekerja dengan baik. Selain itu, informasi struktur spatial dari sebuah gambar akan hilang karena setiap neuron tidak memiliki hubungan antara satu dengan yang lain.



Gambar. 1. Arsitektur Convolutional Neural Network Sederhana

Ide dari CNN sudah muncul pada tahun 1970, namun ide tersebut baru diterbitkan pada sebuah paper pada tahun 1998. Paper tersebut dibuat oleh Yann Lecun, Leon Bottou, Yoshua Bengio, dan Patrick Haffne dengan judul “Gradient-based learning applied to document recognition” yang kemudian menjadi dasar bagi perkembangan CNN modern [17].

Operasi convolution adalah operasi utama dalam DCNN. Pada operasi ini dilakukan ekstraksi fitur laten dari kumpulan pixel input 2 dimensi. Tidak seperti fully connected layer, struktur spatial pada gambar akan tetap dipertahankan oleh convolution layer. Selain itu, jumlah weight yang digunakan juga jauh lebih sedikit dengan adanya weight sharing. Hasil dari proses convolution disebut sebagai feature map atau activation map. Sebuah filter pada operasi convolution dapat mendeteksi sebuah fitur yang terdapat pada sebuah gambar, misal sebuah garis lurus, garis miring, atau bentuk sederhana lain [18].

V. LONG SHORT TERM MEMORY NETWORK

LSTM merupakan salah satu arsitektur Recurrent Neural Network (RNN) [19], yaitu salah satu pengembangan dari neural network biasa yang didesain untuk menangani input yang bersifat bersambung. Yang dimaksud dengan input bersambung adalah output tidak hanya ditentukan oleh input yang baru saja diberikan tetapi juga bergantung pada input yang telah diberikan sebelumnya. Contoh dari input sequential adalah penerjemahan bahasa, di mana arti dari sebuah kata dapat dipengaruhi oleh konteks dari kata

sebelumnya. Oleh sebab itu, RNN didesain agar output dari hasil terakhir dipertimbangkan menjadi input untuk hasil selanjutnya.

Terdapat 2 masalah yang dimiliki oleh RNN, yaitu exploding gradient dan vanishing gradient. Kedua permasalahan ini ditemui tidak hanya pada RNN tetapi pada neural network yang dalam (memiliki banyak layer). RNN dapat dianggap memiliki kedalaman yang sangat besar karena kedalaman RNN ditentukan oleh banyaknya data dalam sequence yang hendak diproses. Dalam kasus seperti penerjemahan sebuah kalimat, input yang diberikan adalah setiap kata dalam kalimat tersebut yang jumlahnya cukup banyak.

Exploding gradient terjadi ketika nilai weight dari network bernilai lebih besar dari 1 dan ukuran network sangat dalam. Seperti yang dapat dilihat pada persamaan backpropagation, nilai dari error akan dipropagasikan ke belakang. Salah satu operasi yang dilakukan adalah nilai error dikalikan dengan weight pada layer sebelumnya. Hal ini menyebabkan dalam proses backpropagation untuk setiap timestep nilai error selalu diperbesar sehingga gradien hasil akhir sangat besar. Nilai gradien yang terlalu besar menyebabkan nilai weight yang diupdate juga besar sehingga network tidak dapat konvergen.

Vanishing gradient adalah kebalikan dari exploding gradient, dimana nilai dari weight network kurang dari 1 dan ukuran network sangat dalam. Hal ini menyebabkan nilai error diperkecil setiap timestep sehingga gradien hilang, nilai update bobot untuk layer di depan akan menjadi kecil sehingga network tidak dapat dilatih dengan baik [20].

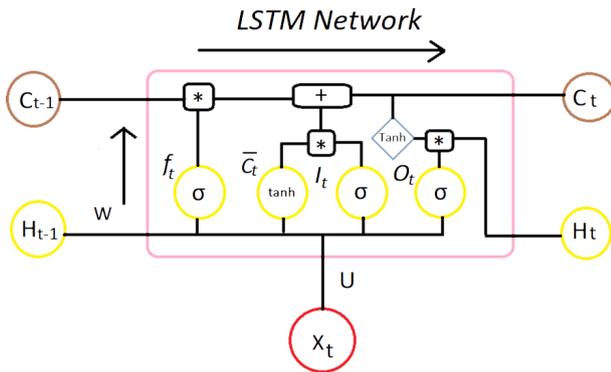
Untuk mengatasi masalah ini, dirancanglah pengembangan dari RNN yaitu Long Short Term Memory network (LSTM) [21]. LSTM pertama kali ditemukan oleh Sepp Hochreiter dan Jürgen Schmidhuber pada tahun 1997. LSTM juga dan disempurnakan oleh banyak orang seperti Felix Gers, Fred Cummins, Santiago Fernandez, Justin Bayer, Daan Wierstra, Julian Togelius, Faustian Gomez, Matteo Gagliolo, dan Alex Graves.

LSTM tersusun dari sebuah cell state yang bersifat sebagai memori untuk LSTM. Dengan adanya memori ini, permasalahan long term dependency (hilangnya efek input lama terhadap training) yang dimiliki oleh RNN dapat diatasi. Nilai dari memory cell pada LSTM diatur oleh 3 gating unit berbeda, yaitu forget gate, input gate, dan output gate.

VI. DEEP Q NETWORK

Deep reinforcement learning merupakan penggabungan dari algoritma reinforcement learning dan algoritma deep learning. Salah satu kegunaan dari deep learning adalah dapat dibentuk menjadi sebuah arsitektur general artificial intelligence [3]. Implementasi dari algoritma reinforcement learning terbatas pada permasalahan-permasalahan yang sederhana. Untuk permasalahan yang kompleks digunakan kumpulan fitur-fitur penting yang ditentukan oleh pembuat program sehingga permasalahan kompleks tersebut dapat disederhanakan. Algoritma deep learning dapat digunakan untuk memproses input yang bersifat kompleks, seperti citra atau data berurutan [18], [21]. Dengan deep learning dapat

dilakukan ekstraksi dan deteksi fitur-fitur penting dari sebuah input dengan dimensionalitas tinggi [3].



Gambar. 2. Struktur LSTM Sederhana

Terdapat beberapa tantangan dalam proses penggabungan antara deep learning dan reinforcement learning. Tantangan pertama adalah perbedaan metode training yang dilakukan, di mana pada deep learning digunakan data dengan jumlah yang sangat banyak dan telah diberi label oleh manusia. Pada reinforcement learning training dilakukan berdasarkan reward yang diterima, di mana reward ini bersifat jarang, terdapat noise, serta terdapat delay antar reward. Permasalahan lainnya adalah pada deep learning data yang digunakan tidak berhubungan antara satu dengan yang lain, sedangkan pada reinforcement learning data yang masuk bersifat berurutan dan memiliki hubungan yang erat antar data. Selain itu data yang diterima pada reinforcement learning akan berubah seiring perubahan policy, di mana pada deep learning distribusi data yang diterima selalu tetap [3].

Deep Q Network adalah sebuah arsitektur deep reinforcement learning yang pertama kali dikembangkan. Deep Q learning dikembangkan oleh DeepMind pada tahun 2013. Dari arsitektur ini dihasilkan sebuah AI yang berhasil mengalahkan nilai AI lain dalam 43 game Atari 2600 [22].

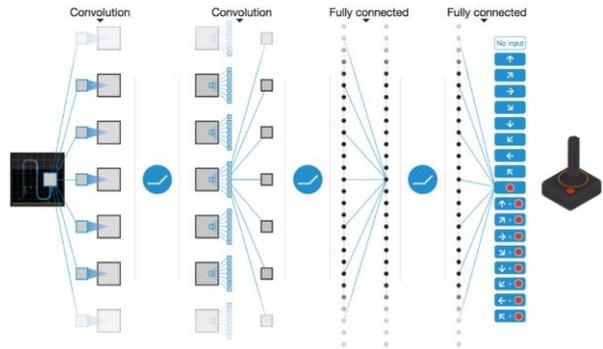
Secara garis besar, Deep Q Network adalah sebuah convolutional neural network yang dilatih menggunakan metode Q learning agar ketika diberi sebuah state (citra yang menggambarkan keadaan game sekarang), network akan menghasilkan estimasi Q function untuk masing-masing aksi yang dapat dilakukan. Aksi yang akan dipilih adalah aksi dengan nilai Q function yang paling tinggi [3]. Metode Q Learning sendiri dapat diformulasikan sebagai berikut:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(R + \gamma \max_{a'} Q(s', a') - Q(s, a)) \quad (2)$$

Untuk menghilangkan korelasi antar data training pada DQN digunakan metode experience replay [23]. Dengan experience replay, pengalaman dari agen tidak langsung dijadikan data training tetapi akan disimpan dalam sebuah replay memory. Pada saat training akan diambil sebuah batch dari experience secara random dari replay memory sebagai data training.

Untuk mengatasi nilai target Q function yang berubah-ubah setiap perubahan policy, digunakan separate target network [22]. Digunakan 2 set parameter berbeda untuk nilai target dan untuk proses training. Parameter dari target network diubah setelah beberapa step tertentu, dimana

jumlah step tersebut adalah angka yang cukup besar, sedangkan network kedua selalu diubah tiap epoch pada training. Dengan target network, perubahan nilai target Q function dapat dikurangi.



Gambar. 3. Arsitektur DQN

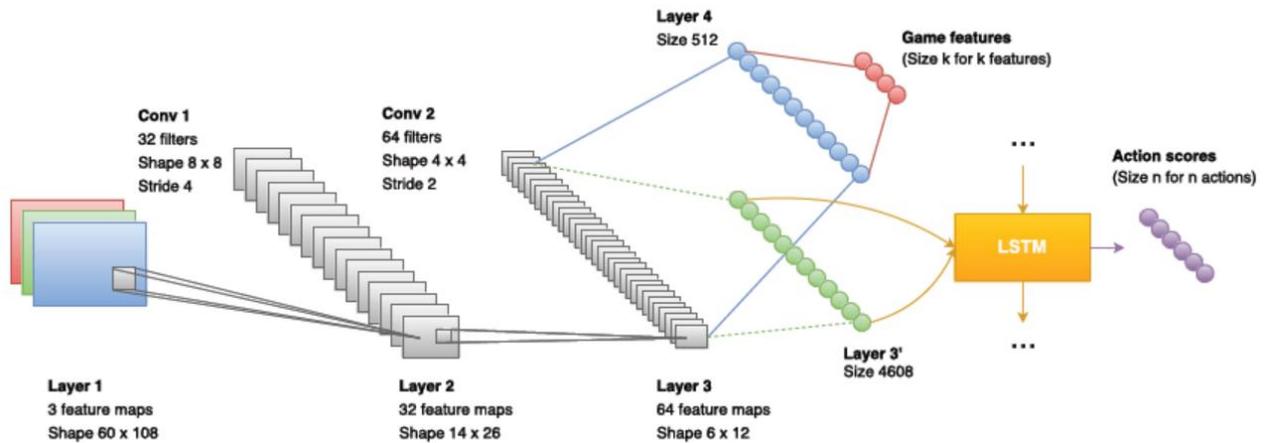
Untuk memperoleh data yang lebih bervariasi, digunakan ϵ -greedy policy pada proses training [3]. Pada fase testing, aksi yang diambil agen adalah aksi dengan nilai Q function tertinggi, sedangkan pada fase training, terdapat peluang sebesar ϵ untuk mengambil aksi random. Hal ini dilakukan untuk mengeksplorasi kemungkinan-kemungkinan yang belum pernah dicoba dan memperbaiki kemungkinan nilai Q function yang salah. Nilai ϵ akan dikurangi seiring berjalannya proses training untuk mengurangi proses eksplorasi pada akhir training.

Pada pengembangan berikutnya dilakukan berbagai modifikasi pada arsitektur dan mekanisme training yang dilakukan dalam DQN sehingga diperoleh hasil yang lebih baik. Pengembangan-pengembangan tersebut antara lain penggunaan prioritized experience replay [24], penggunaan advantage function [25], implementasi DQN secara paralel dengan menggunakan beberapa komputer [26], dan penggunaan actor-critic network untuk environment yang bersifat continuous [27].

VII. ASYNCHRONOUS ADVANTAGE ACTOR CRITIC

Asynchronous Advantage Actor Critic (A3C) adalah sebuah arsitektur deep reinforcement learning baru yang dikembangkan oleh DeepMind pada tahun 2016. Kelebihan dari A3C dibandingkan DQN adalah penggunaan resource yang lebih sedikit, proses training yang lebih cepat, dapat digunakan untuk environment yang bersifat continuous, memiliki kemampuan untuk mengingat state sebelumnya lebih jauh, dan dapat mencapai skor yang lebih tinggi dibanding skor yang dicapai DQN dengan rata-rata skor mencapai 4 kali rata-rata skor DQN [28].

Perbedaan pertama antara A3C dengan DQN adalah proses training dari algoritma A3C berjalan secara asynchronous. A3C memanfaatkan kemampuan multiprocessing sebuah multicore CPU untuk menjalankan beberapa agen sekaligus secara bersamaan. Tiap agen memiliki weight network masing-masing dan berinteraksi dengan environment mereka sendiri pada saat yang bersamaan secara paralel [28]. Kelebihan penggunaan metode ini adalah proses training dapat dipercepat karena tidak diperlukan komunikasi antara beberapa komputer



Gambar. 3. Contoh Arsitektur A3C

berbeda dan dapat dilakukan update parameter dengan metode Hogwild! [29] yang lebih cepat. Selain itu penggunaan memory dapat dikurangi karena experience replay dapat digantikan dengan agen yang bekerja secara paralel.

Perbedaan lainnya adalah digunakannya advantage actor critic untuk menentukan aksi terbaik untuk diambil. Penggunaan advantage function menghasilkan AI yang baik dalam waktu yang lebih cepat dibandingkan Q function [28]. Metode estimasi advantage function yang digunakan adalah generalized advantage estimation [15] untuk menghasilkan estimasi yang dapat diatur keseimbangan antara bias dan variansnya. Selain itu, penggunaan actor-critic membuat AI dapat dimodifikasi dengan mudah untuk menghadapi environment yang bersifat continuous [30].

Perbedaan antara arsitektur DQN dengan A3C adalah pada A3C terdapat sebuah layer LSTM setelah convolutional layer dan Q network digantikan sebuah actor-critic network. LSTM membuat arsitektur dapat digunakan untuk mengatasi environment yang memiliki long term dependency, misalnya untuk menyusuri labirin [31].

VIII. ENVIRONMENT

Bagian terakhir yang diperlukan untuk pembuatan Artificial Intelligence adalah sebuah environment. Istilah environment sendiri merujuk pada program yang berinteraksi dengan AI dengan memberikan observasi atau keadaan saat itu, menerima dan memproses perintah yang diberikan lalu mengembalikan umpan balik berupa skor positif apabila aksi yang dilakukan baik dan skor negatif apabila aksi yang dilakukan buruk.

Environment yang digunakan pada percobaan ini adalah OpenAI gym [32]. Library ini dibuat dengan tujuan untuk membuat sebuah standarisasi yang dapat digunakan untuk penelitian dalam bidang reinforcement learning. Dalam OpenAI gym disediakan kumpulan environment yang dapat digunakan untuk uji coba algoritma reinforcement learning. Environment yang disediakan didesain berdasarkan sebuah interface yang sama sehingga dimungkinkan untuk membuat 1 algoritma untuk menyelesaikan semua environment. Pada penelitian ini digunakan game berbasis Atari yang disediakan oleh OpenAI Gym dan game berbasis HTML5 yang dibuat interfacenya menggunakan standar OpenAI Gym.

Game Atari yang disediakan OpenAI Gym diperoleh dari library Arcade Learning Environment (ALE). Dalam library ini terdapat lebih dari 50 ROM game Atari yang dapat digunakan sebagai environment untuk reinforcement learning. Kegunaan library ALE pertama-tama adalah menyediakan fungsi-fungsi untuk mengendalikan emulator Atari. Selain itu disediakan pula fungsi-fungsi untuk mengakses data-data yang terdapat dalam emulator Atari, seperti tampilan layar, skor, dan nyawa player.

Untuk game HTML5 yang dibuat sendiri, diperlukan bantuan 2 library, yaitu Phaser dan Pypeteer. Phaser adalah sebuah library open source untuk pengembangan game berbasis HTML5 dan javascript. Library ini digunakan untuk pengembangan game pada platform web browser baik pada desktop maupun mobile dengan WebGL dan canvas rendering. Pypeteer merupakan implementasi dari library Puppeter dengan menggunakan Python. Puppeter sendiri adalah sebuah library untuk mengontrol web browser Chrome atau Chromium. Dengan Pypeteer dapat diambil screenshot dari sebuah web page, crawling sebuah website, pengiriman form secara otomatis, otomatis testing webapp, dan sebagainya.

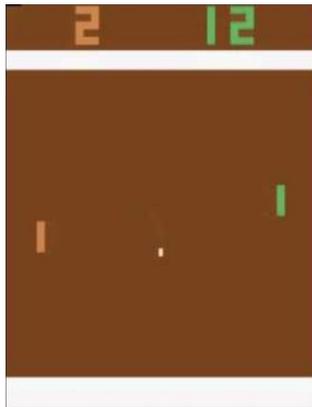
Pada environment dengan standar OpenAI Gym harus terdapat 4 komponen penting, yaitu kemampuan untuk memberikan observation atau keadaan game saat itu, menerima dan memproses perintah yang diberikan, serta memberikan reward untuk tiap aksi yang diberikan. Hasil observasi game dapat diperoleh dari screenshot game pada web browser yang dilakukan dengan bantuan library Pypeteer. Pengiriman perintah dari AI ke game HTML5 dilakukan dengan bantuan Pypeteer. Library Pypeteer dapat mengakses dan mengubah isi variabel pada game sehingga perintah dapat dijalankan. Setelah perintah diberikan, perintah akan dijalankan sesuai logika game yang sudah ditetapkan menggunakan bantuan library Phaser. Reward diperoleh dengan cara mengakses isi variabel skor pada game dengan menggunakan library Pypeteer.

IX. UJI COBA

Terdapat 2 jenis uji coba yang dilakukan, yaitu uji coba hyperparameter dan uji coba perbandingan skor. Pada uji coba perubahan hyperparameter, terdapat 47 skenario uji coba yang akan dilakukan. Masing-masing uji coba dilakukan menggunakan game Pong berbasis konsol Atari.

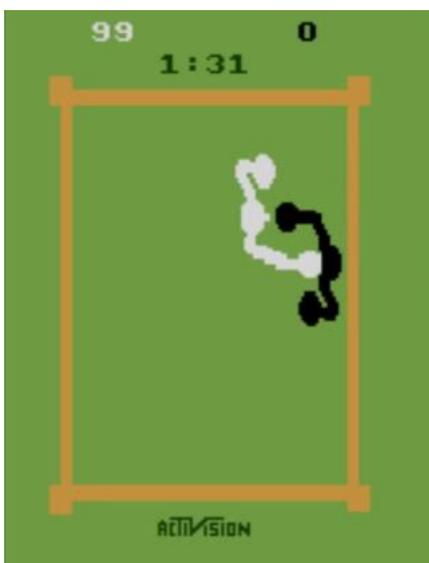
Pada uji coba perbandingan skor, terdapat 5 skenario uji coba, dimana masing-masing skenario digunakan untuk menguji tiap game yang ada. Pada tiap skenario akan diambil nilai yang dapat dicapai AI dan nilai yang dapat dicapai oleh pemain pemula.

Pada uji coba perubahan hyperparameter, terdapat 47 skenario uji coba yang akan dilakukan. Akan dilakukan perubahan hyperparameter yang dapat mempengaruhi kecepatan training dan score yang dapat diperoleh AI. Terdapat 9 hyperparameter yang nilainya hendak diubah, yaitu jumlah worker, jumlah frameskip, ukuran input, update step, gamma, learning rate, jumlah hidden unit LSTM, tau, dan GPU support. Pada uji coba hyperparameter akan digunakan 2 game berbeda, yaitu game Atari Pong dan game Boxing.



Gambar. 4. Game Pong

Tujuan dari game Pong adalah untuk mencegah bola masuk ke gawang sendiri dan berusaha memantulkan bola agar masuk ke gawang lawan di sebelah kiri. Pada uji coba, AI akan diberi skor +1 apabila berhasil memasukkan bola ke gawang lawan, sedangkan jika lawan berhasil memasukkan bola ke gawang AI maka AI akan diberi skor -1. Permainan akan berakhir ketika salah satu pihak mendapat skor 21. Nilai maksimal dari permainan dengan sistem penilaian ini adalah 21 (jika AI menang 21 - 0) dan nilai minimum dari sistem penilaian ini adalah -21 (jika AI kalah 0 - 21).



Gambar. 5. Game Boxing

Tujuan dari game Boxing adalah dapat mengalahkan petinju lawan dengan memukul kepala lawan dan mencegah lawan memukul kepala petinju yang dimainkan. Pukulan yang mengenai bagian tubuh lain tidak dihitung. Sistem pemberian skor pada game ini adalah, AI akan diberi skor +1 apabila berhasil meninju kepala lawan, sedangkan jika lawan berhasil meninju kepala pemain maka AI akan diberi skor -1. Permainan akan berakhir ketika salah satu pihak mendapat skor 100 atau ketika waktu permainan habis. Permainan diberi waktu 2 menit. Nilai maksimal dari permainan dengan sistem penilaian ini adalah 100 (jika AI menang 100 - 0) dan nilai minimum dari sistem penilaian ini adalah -100 (jika AI kalah 0 - 100).

Pada Tabel I ditunjukkan nilai hyperparameter yang dijadikan standar pembandingan setiap percobaan perubahan nilai hyperparameter. Nilai hyperparameter yang tidak diganti akan mengikuti nilai pada Tabel I.

TABEL I
NILAI DEFAULT HYPERPARAMETER

No	Hyperparameter	Nilai
1	Worker	14
2	Frameskip	4
3	Input size	80 × 80 pixel
4	Update step	20
5	Gamma	0.99
6	Learning rate	0.0001
7	LSTM hidden unit	512
8	Tau	1
9	GPU	No

Parameter worker adalah parameter yang menentukan jumlah worker thread yang digunakan oleh AI. Parameter kontrol menggunakan 14 worker thread yang berjalan bersama pada saat training dijalankan.

TABEL II
PENGARUH PARAMETER JUMLAH WORKER PADA GAME PONG

No	Worker	Max Rolling Average	Waktu
1	6	20.3	7:09:40
2	8	20.25	6:47:37
3	10	20.25	7:29:52
4	12	19.95	5:39:34
5	14	20	5:31:58
6	16	19.1	7:33:31

TABEL III
PENGARUH PARAMETER JUMLAH WORKER PADA GAME BOXING

No	Worker	Max Rolling Average	Waktu
1	6	71.2	11:57:06
2	8	77.45	11:47:57
3	10	32	11:46:27
4	12	48.6	11:58:51
5	14	71.4	12:00:59
6	16	6.65	9:38:05

Parameter frameskip adalah parameter yang menentukan jumlah frame yang dilompati oleh AI setiap langkah. Parameter kontrol menggunakan 4 frameskip dengan mengambil max value terhadap 2 frame terakhir.

TABEL IV
PENGARUH PARAMETER FRAMESKIP PADA GAME PONG

No	Frameskip	Max Rolling Average	Waktu
1	3	17.4	1
2	4	20	2
3	5	4.75	3

TABEL V
PENGARUH PARAMETER FRAMESKIP PADA GAME BOXING

No	Frameskip	Max Rolling Average	Waktu
1	3	1.45	10:59:25
2	4	71.4	12:00:59
3	5	87.8	11:49:13

Pada uji coba berikutnya dilakukan perubahan ukuran input. Parameter input size adalah parameter yang menentukan ukuran input yang akan diproses network. Parameter kontrol menggunakan citra berukuran 80 x 80 pixel.

TABEL VI
PENGARUH UKURAN INPUT PADA GAME PONG

No	Input Size	Max Rolling Average	Waktu
1	10x10	15.05	7:30:32
2	20x20	20.15	4:59:13
3	40x40	21	5:46:03
4	60x60	20.05	6:08:27
5	80x80	20	5:31:58

TABEL VII
PENGARUH UKURAN INPUT PADA GAME BOXING

No	Input Size	Max Rolling Average	Waktu
1	10x10	36.1	11:53:20
2	20x20	90.65	10:22:15
3	40x40	99.85	11:20:57
4	60x60	96.7	11:51:41
5	80x80	71.4	12:00:59

Parameter update step adalah parameter yang menentukan jumlah frame dan reward yang disimpan oleh worker sebelum melakukan perhitungan gradien dan melakukan update bobot model. Parameter kontrol menggunakan 20 step sebelum update.

TABEL VIII
PENGARUH PARAMETER UPDATE STEP PADA GAME PONG

No	Update Step	Max Rolling Average	Waktu
1	5	7.85	4:34:40
2	10	19.25	5:09:32
3	15	19.8	5:41:31
4	20	20	5:31:58

TABEL IX
PENGARUH PARAMETER UPDATE STEP PADA GAME BOXING

No	Update Step	Max Rolling Average	Waktu
1	5	95.7	11:32:18
2	10	100	11:24:04

3	15	57.55	11:46:37
4	20	71.4	12:00:59

Parameter gamma adalah parameter yang menentukan jumlah discount factor yang digunakan oleh AI. Parameter kontrol menggunakan nilai gamma sebesar 0.99.

TABEL X
PENGARUH PARAMETER GAMMA PADA GAME PONG

No	Gamma	Max Rolling Average	Waktu
1	0.93	18.8	7:47:09
2	0.95	19.65	7:13:55
3	0.98	20.25	7:04:39
4	0.99	20	5:31:58
5	0.991	20.3	7:00:55
6	0.995	19.8	7:18:25

TABEL XI
PENGARUH PARAMETER GAMMA PADA GAME BOXING

No	Gamma	Max Rolling Average	Waktu
1	0.93	39	11:13:43
2	0.95	49.5	11:51:31
3	0.98	32.75	11:15:42
4	0.99	71.4	12:00:59
5	0.991	75.85	12:00:11
6	0.995	18.85	10:48:38

Parameter learning rate adalah parameter yang menentukan learning rate yang digunakan untuk update bobot model. Parameter kontrol menggunakan learning rate sebesar 0.0001.

TABEL XII
PENGARUH PARAMETER LEARNING RATE PADA GAME PONG

No	Learning Rate	Max Rolling Average	Waktu
1	0.00003	-0.85	7:25:57
2	0.0001	20	5:31:58
3	0.0003	20.03	7:51:39
4	0.001	-1.25	5:44:29

TABEL XIII
PENGARUH PARAMETER LEARNING RATE PADA GAME BOXING

No	Learning Rate	Max Rolling Average	Waktu
1	0.00003	-3.75	9:25:45
2	0.0001	71.4	12:00:59
3	0.0003	-16.25	7:25:19
4	0.001	-25.25	3:28:53

Parameter LSTM hidden unit menyatakan banyaknya hidden unit yang digunakan oleh model. Parameter kontrol menggunakan 512 hidden unit untuk LSTM.

TABEL XIV
PENGARUH UKURAN LSTM PADA GAME PONG

No	LSTM Hidden Unit	Max Rolling Average	Waktu
1	32	19.9	6:14:07
2	64	20	5:05:18
3	128	20.15	7:07:25
4	256	17.8	7:13:15
5	384	19.1	4:39:10
6	512	20	5:31:58

Parameter Tau pada program ini adalah parameter lambda yang digunakan dalam perhitungan Generalized Advantage Estimation. Parameter ini tidak dinamai lambda pada program karena kata “lambda” merupakan sebuah keyword pada bahasa pemrograman Python. Parameter kontrol menggunakan nilai Tau 1.

TABEL XV
PENGARUH UKURAN LSTM PADA GAME BOXING

No	LSTM Hidden Unit	Max Rolling Average	Waktu
1	32	-6.1	10:34:35
2	64	17.6	9:43:10
3	128	6.25	8:56:17
4	256	78.9	11:10:12
5	384	79.35	12:00:40
6	512	71.4	12:00:59

TABEL XVI
PENGARUH PARAMETER TAU PADA GAME PONG

No	Tau	Max Rolling Average	Waktu
1	0.3	18.9	7:58:59
2	0.5	17.7	6:48:25
3	0.7	20.55	7:57:33
4	0.9	19.9	7:11:39
5	1	20	5:31:58
6	1.1	20.4	5:41:19

TABEL XVII
PENGARUH PARAMETER TAU PADA GAME BOXING

No	Tau	Max Rolling Average	Waktu
1	0.3	21.25	11:36:44
2	0.5	38.1	12:00:29
3	0.7	80.2	11:51:47
4	0.9	84.05	11:29:29
5	1	71.4	12:00:59
6	1.1	-19.65	9:38:48

Parameter gpu_id mengatur GPU mana saja yang dapat digunakan oleh program untuk melakukan training. Untuk uji coba ini, akan digunakan sebuah GPU dan akan dilakukan perubahan jumlah worker. Parameter kontrol tidak menggunakan GPU dan menggunakan 14 worker. Khusus pada uji coba ini, proses training dilakukan hanya selama 1 jam untuk masing-masing skenario uji coba.

TABEL XVIII
PENGARUH PENGGUNAAN GPU PADA GAME PONG

No	Worker	Max Rolling Average	Waktu
1	4	10.7	0:56:54
2	6	20.25	0:59:05
3	8	20.45	0:56:08
4	10	20	0:58:06
5	12	20	0:59:05

TABEL XIX
PENGARUH PENGGUNAAN GPU PADA GAME BOXING

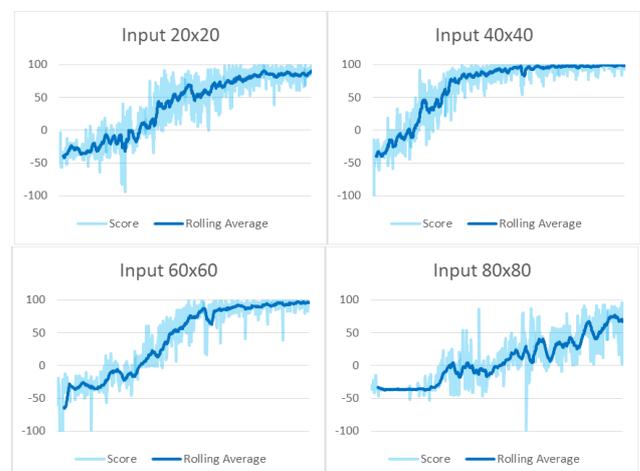
No	Worker	Max Rolling Average	Waktu
1	4	-20.25	0:59:30
2	6	19	0:59:24
3	8	14.3	1:00:01
4	10	19.65	1:00:05
5	12	47.4	0:59:10

Pada uji coba skor akan dilakukan perbandingan antara skor yang berhasil diperoleh oleh AI dengan skor yang diperoleh pemain pemula. Terdapat 5 game berbeda yang akan digunakan dalam uji coba ini, 3 diantaranya adalah game Atari sedangkan 2 lainnya adalah game berbasis HTML5 yang merupakan game buatan sendiri yang telah diintegrasikan dengan library OpenAI gym. Tiga game Atari tersebut antara lain: Pong, Boxing, dan Space Invaders. Sedangkan untuk game HTML5 adalah game balapan sederhana dan game Pinball. Skor yang diperoleh pemain pemula akan digunakan sebagai pembading untuk skor yang diperoleh AI. Akan ditunjukkan perbandingan skor rata-rata, skor tertinggi, dan skor terendah dari pemain pemula beserta AI.

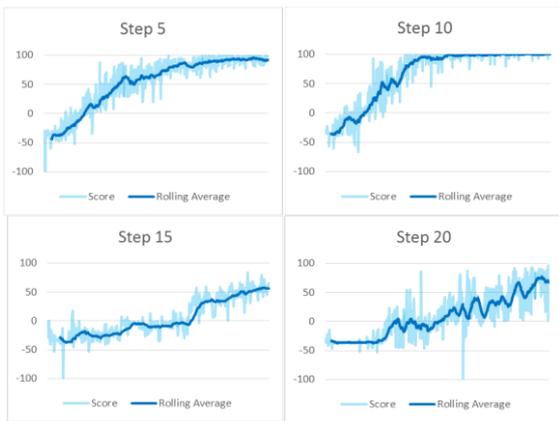
Tujuan dari game Space Invaders adalah untuk menghalau agar alien tidak sampai ke bumi. Sistem pemberian skor pada game ini adalah, alien pada baris 1 dari bawah memberi skor +5 jika ditembak, baris 2 +10, baris 3 +15, dan seterusnya. Kapal induk alien memberi skor +200. Jika player tertembak maka pemain akan kehilangan 1 nyawa (total nyawa pemain 3). Permainan akan berakhir ketika alien berhasil mencapai bumi atau pemain kehabisan nyawa.



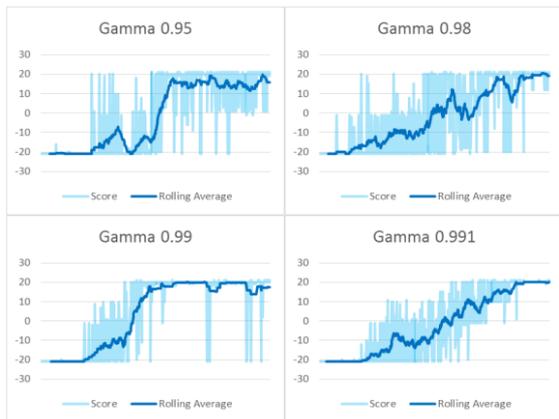
Gambar. 6. Grafik Skor dan Rolling Average dari Perubahan Jumlah Worker pada Game Boxing



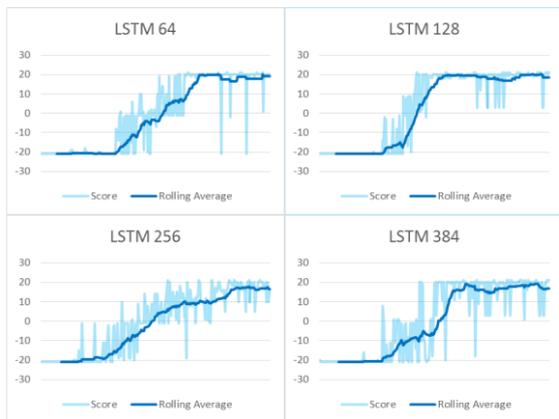
Gambar. 7. Grafik Skor dan Rolling Average dari Perubahan Ukuran Input pada Game Boxing



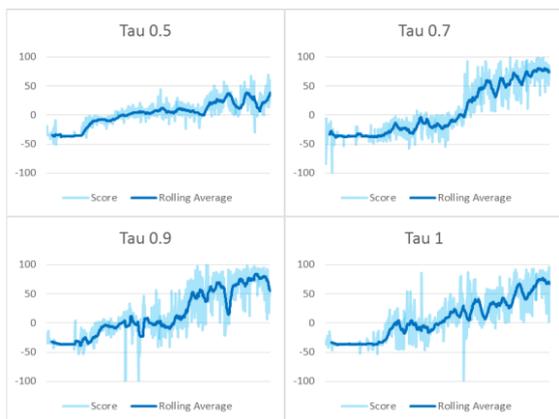
Gambar. 8. Grafik Skor dan Rolling Average dari Perubahan Jumlah Step pada Game Boxing



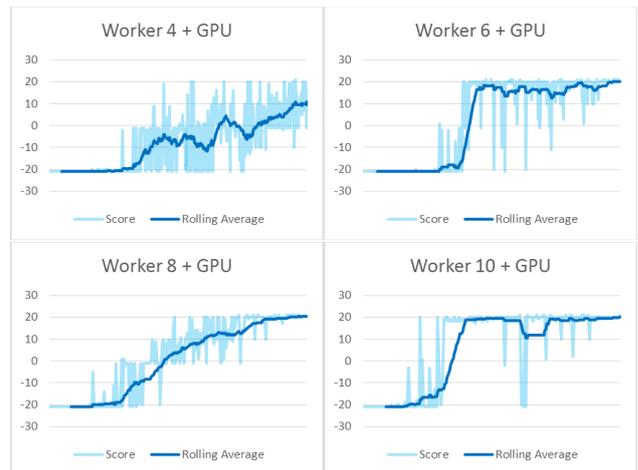
Gambar. 9. Grafik Skor dan Rolling Average dari Perubahan Parameter Gamma pada Game Pong



Gambar. 10. Grafik Skor dan Rolling Average dari Perubahan Ukuran LSTM pada Game Pong



Gambar. 11. Grafik Skor dan Rolling Average dari Perubahan Parameter Tau pada Game Boxing



Gambar. 12. Grafik Skor dan Rolling Average dari Penggunaan GPU pada Game Pong



Gambar. 13. Game Space Invaders

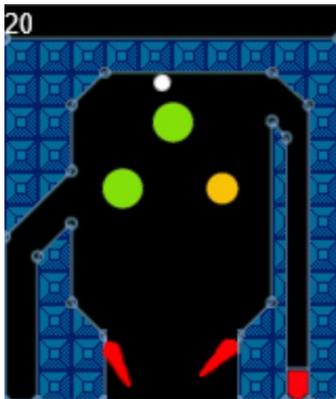
Tujuan dari game Racing adalah mengendarai mobil secepat mungkin untuk mencapai garis finish. Sistem penilaian game ini adalah, AI akan diberi skor apabila berhasil mencapai petak jalan berikutnya. Skor yang diperoleh adalah +20 pada awalnya, namun akan dikurangi terus menerus setiap waktu sehingga untuk mendapatkan skor penuh mobil harus berjalan dengan cepat. Jika berhasil mencapai garis finish, AI akan diberi skor bonus berupa 50% dari skor sebelum mencapai finish.



Gambar. 14. Game Racing

Tujuan dari permainan ini adalah mencegah bola jatuh ke lubang sambil berusaha mendapatkan skor dengan cara mengarahkan agar bola menumbuk bantalan di tengah. Pada game ini, AI akan diberi skor +5 apabila bola berhasil mengenai salah satu dari 3 bantalan di tengah area

permainan. AI akan kalah apabila bola jatuh ke dalam lubang dan tidak dapat dipukul kembali ke atas oleh pemukul.



Gambar.15. Game Pinball

TABEL XX
PERBANDINGAN SKOR PEMAIN PEMULA DENGAN AI PADA GAME ATARI

	Pong		Boxing		Space Invaders	
	Human	AI	Human	AI	Human	AI
Average	-10.1	20.6	7.7	98.7	972.5	2896.5
Max	13	21	20	100	2150	3225
Min	-19	20	3	96	575	2660

TABEL XXI
PERBANDINGAN SKOR PEMAIN PEMULA DENGAN AI PADA GAME HTML5

	Racing		Pinball	
	Human	AI	Human	AI
Average	555	966.5	95	124.5
Max	683	1041	220	350
Min	415	840	30	15

Hasil dari seluruh uji coba AI telah didokumentasikan dalam bentuk video yang dapat dilihat pada website Youtube dengan link <https://tinyurl.com/y8swcx16>.

X. KESIMPULAN

Berdasarkan hasil tersebut dapat disimpulkan beberapa poin yang akan disebutkan dibawah ini.

1. Reinforcement Learning adalah salah satu jenis machine learning yang dapat digunakan untuk membuat sebuah Artificial Intelligence dengan trial and error, tanpa diperlukan handcrafted feature.

2. Algoritma deep learning dapat digabungkan dengan algoritma reinforcement learning untuk mengatasi masalah yang lebih kompleks dan bervariasi, seperti memainkan berbagai macam jenis game berbeda hanya dari mempelajari citra output dari layar.

3. Asynchronous Advantage Actor-Critic (A3C) adalah algoritma deep reinforcement learning yang dapat belajar memainkan beberapa jenis game yang berbeda dengan 1 arsitektur. A3C terdiri dari 3 komponen utama, yaitu Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM), dan Actor-Critic Network.

4. Pada algoritma A3C, Convolutional Neural Network

bertugas merangkul citra input dari layar konsol game dengan cara mengekstraksi fitur penting yang diperoleh dari citra input. Hasil dari Convolutional Neural Network ini menjadi representasi state saat ini dan akan diolah lebih lanjut.

5. Dalam algoritma A3C, Long Short-Term Memory Network berfungsi sebagai pengolah data state yang bersifat sequential untuk merangkul perubahan yang terjadi pada environment. Data representasi state dari Convolutional Neural Network diolah menjadi representasi state dan perubahannya oleh Long Short-Term Memory Network.

6. Actor-Critic Network dalam A3C berfungsi menentukan tindakan terbaik untuk dilakukan pada situasi yang diberikan.

7. Pada paper acuan, algoritma A3C diimplementasikan menggunakan CPU sedangkan pada program acuan dilakukan modifikasi kepada program agar dapat memanfaatkan GPU. Hasilnya adalah proses training dapat dilakukan dengan lebih cepat.

8. Dari hasil percobaan, AI yang dibuat menggunakan A3C memperoleh nilai yang lebih tinggi dibandingkan nilai pemain pemula dalam 5 jenis game yang berbeda.

DAFTAR PUSTAKA

- [1] G. Tesauro, "Temporal difference learning and TD-Gammon," *Commun. ACM*, 1995, doi: 10.1145/203330.203343.
- [2] M. Campbell, a. J. Hoane Jr., and F. Hsu, "Deep Blue," *Artif. Intell.*, vol. 134, no. 1–2, pp. 57–83, 2002, doi: 10.1016/S0004-3702(01)00129-1.
- [3] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv Prepr. arXiv1312.5602*, 2013.
- [4] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, 2017, doi: 10.1038/nature24270.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, 2012, doi: <http://dx.doi.org/10.1016/j.protcy.2014.09.007>.
- [6] R. S. Sutton and A. G. Barto, *Introduction to reinforcement learning*, vol. 135. MIT press Cambridge, 1998.
- [7] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, 2000, pp. 1008–1014.
- [8] R. Bellman, "A Markovian decision process," *J. Math. Mech.*, pp. 679–684, 1957.
- [9] G. Brockman *et al.*, "Evolutionary algorithms for reinforcement learning," *J. Artif. Intell. Res.*, vol. 47, pp. 253–279, Jun. 2013.
- [10] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," *Adv. Neural Inf. Process. Syst. 12*, 1999, doi: 10.1.1.37.9714.
- [11] R. J. Willia, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," *Mach. Learn.*, 1992, doi: 10.1023/A:1022672621406.
- [12] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, 2004, doi: 10.1109/ROBOT.2004.1307456.
- [13] R. S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Mach. Learn.*, 1988, doi: 10.1023/A:1022633531479.
- [14] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, 1992, doi: 10.1007/BF00992698.
- [15] G. Brockman *et al.*, "High-dimensional continuous control using generalized advantage estimation," *Adv. Neural Inf. Process. Syst.*, vol. 47, no. 7540, pp. 693–701, Jun. 2015.
- [16] V. Sharma, S. Rai, and A. Dev, "A Comprehensive Study of Artificial Neural Networks," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, 2012.
- [17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, 1998, doi: 10.1109/5.726791.

- [18] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *CoRR*, vol. abs/1511.0, 2015.
- [19] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent Neural Network based Language Model," *Interspeech*, 2010.
- [20] S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, 1998, doi: 10.1142/S0218488598000094.
- [21] S. Hochreiter and J. Uergen Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Comput.*, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [22] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.
- [23] L. Lin, "Reinforcement Learning for Robots Using Neural Networks," *Report, C.*, 1993.
- [24] V. Mnih *et al.*, "Prioritized Experience Replay," *Int. Conf. Mach. Learn.*, 2015, doi: 10.1038/nature14236.
- [25] Z. Wang, N. de Freitas, and M. Lanctot, "Dueling Network Architectures for Deep Reinforcement Learning," *arXiv Prepr. arXiv1511.06581*, 2015.
- [26] A. Nair *et al.*, "Massively Parallel Methods for Deep Reinforcement Learning," *arXiv:1507.04296*, 2015, doi: 10.1109/IJCNN.2010.5596468.
- [27] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv Prepr. arXiv1509.02971*, 2015.
- [28] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.
- [29] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in neural information processing systems*, 2011, pp. 693–701.
- [30] DeepMind, "Asynchronous Methods for Deep Reinforcement Learning: MuJoCo - YouTube." [Online]. Available: <https://www.youtube.com/watch?v=Ajje08-iPx8&feature=youtu.be>. [Accessed: 01-May-2018].
- [31] DeepMind, "Asynchronous Methods for Deep Reinforcement Learning: Labyrinth - YouTube." [Online]. Available: <https://www.youtube.com/watch?v=nMR5mjCFZCw&feature=youtu.be>. [Accessed: 01-May-2018].
- [32] G. Brockman *et al.*, "OpenAI Gym." 2016.

Evan Kusuma Susanto Lahir di Kota Surabaya, Jawa Timur, Indonesia pada tahun 1996. Saat ini sebagai mahasiswa tingkat akhir dengan jurusan S1 Teknik Informatika di STTS Surabaya. Risetnya berfokus pada machine learning dan deep reinforcement learning.

Yosi Kristian lahir di Tuban, Jawa Timur, Indonesia pada tahun 1981. Beliau menyelesaikan studi S1 di program studi Informatika STTS pada tahun 2004 dan menyelesaikan S2 di program studi Informatika STTS pada tahun 2008. Beliau telah menyelesaikan studi masternya di jurusan Informatika STTS. Saat ini beliau sedang melanjutkan Ph.D di Institut Teknologi Sepuluh November (ITS), Surabaya, Indonesia. Penelitiannya terdiri dari Machine Learning, Intellegence System, dan Computer Vision. Beliau merupakan member dari IAENG.

IDENTIFIKASI FOTO WANITA BERHIJAB DARI MAJALAH UNTUK PEMBUATAN KATALOG BUSANA MUSLIM OTOMATIS MEMANFAATKAN CONVOLUTIONAL NEURAL NETWORK

M. Najamudin Ridha, *STMIK Indonesia Banjarmasin*, Endang Setyati, *Institut Sains dan Teknologi Terpadu Surabaya*, Yosi Kristian, *Institut Sains dan Teknologi Terpadu Surabaya*

Abstrak—Perkembangan Fashion Muslim di Indonesia terus meningkat, disisi lain terobosan baru pada Deep Learning dengan memadukan arsitektur seperti dropout regularizations dan Rectified Linear Unit (ReLU) sebagai fungsi aktivasi dan data augmentation, mampu mencapai terobosan pada large scale image classification. Penelitian ini menggunakan metode deteksi objek wajah dengan Haar Cascades Classification untuk mendapatkan sample dataset wajah dan preprocessing data testing untuk dilanjutkan pada metode machine learning untuk klasifikasi citra dengan Convolutional Neural Network. Dataset yang digunakan adalah kumpulan katalog busana online, dataset yang sudah di preprocessing dibagi menjadi dua kategori, yaitu Hijab untuk semua citra wanita berhijab, dan Non Hijab untuk citra yang bukan wanita berhijab. selanjutnya klasifikasi citra menggunakan data ujicoba majalah digital terbitan Hijabella, Joy Indonesia dan Scarf Indonesia. Semakin besar resolusi citra input untuk preprocessing pada majalah digital, maka akan semakin banyak objek citra yang terdeteksi, dengan meningkatkan jumlah dataset untuk training dan validasi, mampu menambah hasil akurasi yang didapatkan, terjadi peningkatan akurasi pada dataset 2.500 wajah perkategori ke 5.000 wajah perkategori dengan resolusi 720p meningkat dari rata-rata 81.30% menjadi 82.31%, peningkatan rata-rata 1.01% dan tertinggi 2.14%, sedangkan resolusi 1080p meningkat dari rata-rata 83.03% menjadi 83.68%, peningkatan rata-rata 0.65% dan tertinggi 1.73%, akurasi tertinggi adalah sebesar 84.72% menggunakan model dataset 5.000 secara acak perkategori.

Kata Kunci—Fashion Muslim, Majalah Digital, Katalog Online, Hijab, Haar Cascades Classification, Convolutional Neural Network

I. PENDAHULUAN

Perkembangan fashion muslim di Indonesia terus meningkat, sebagian besar didorong oleh banyaknya perancang busana muslim dan suksesnya proses transformasi dari gaya konservatif menjadi lebih kontemporer yang sesuai dengan tipikal generasi muda Indonesia, ada beberapa faktor lain yang mendorong perkembangan tren fashion, mulai dari munculnya banyak komunitas seperti Hijabers Community, Hijabers Mom, sampai diselenggarakannya beragam bazar dan peragaan busana muslim

M. Najamudin Ridha, Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Banjarmasin, Banjarmasin, Kalimantan Selatan, Indonesia (e-mail:admin@komptuterkampus.com).

Endang Setyati, Magister Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail:endang@stts.edu)

Yosi Kristian, Magister Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail:yosi@stts.edu)

Menurut Dirjen Industri Kecil Menengah (IKM) Kementerian Perindustrian, pada tahun 2015 terdapat 20 Juta penduduk Indonesia yang menggunakan hijab, hal ini selaras dengan perkembangan industri fashion muslim sebesar 7 persen setiap tahun. Peran pemerintah dalam mendukung trend fashion muslim sebagai icon fashion Indonesia di mata dunia dengan berbagai usaha yang mendukung para penggiat fashion Indonesia baik berupa bahan baku, teknologi, SDM, pemasaran dan modal.

Dalam dunia Internet, persebaran tren fashion bahkan lebih cepat dari media massa, ada banyak sekali model fashion dari berbagai sumber dan merek. Jenis dan gaya yang tak terbatas. Para pengusaha yang bergerak dibidang fashion saling berlomba-lomba dalam memasarkan produk yang dibuat, banyak katalog dan majalah yang membahas masalah fashion muslim, baik hijab, dress dan lain sebagainya. Untuk membuat katalog secara otomatis diperlukan cara yang dapat memisah fashion tertentu dari kumpulan jenis dan gaya fashion yang ada sesuai dengan kategorinya dengan memanfaatkan Computer Vision dalam proses pengkategorian.

Salah satu kekurangan pada bidang citra digital adalah untuk mengenali object yang ada pada citra secara umum, berbagai kalangan diseluruh dunia mencoba melakukan pendekatan pengenalan citra bagi komputer agar dapat meniru kemampuan manusia dalam mengenali object. Salah satu pendekatan yang berhasil digunakan adalah Neural Network yang terinspirasi dari jaringan syaraf pada manusia. Konsep ini kemudian dikembangkan lebih lanjut dalam Deep Learning.[1].

Banyak metode yang bisa digunakan untuk klasifikasi manusia berdasarkan gaya, raut wajah, pakaian yang digunakan, dan lain sebagainya, saat ini teknik Machine Learning menghasilkan hasil yang sangat bagus untuk mengatasi masalah klasifikasi Machine Learning yang lebih advance, salah satunya adalah Deep Convolutional Neural Network (CNN) yang merupakan metode Deep Learning yang powerful[2]. Namun CNN, seperti metode Deep Learning lainnya, memiliki kelemahan yaitu proses pelatihan model yang lama. Dengan perkembangan perangkat keras, hal tersebut dapat diatasi menggunakan teknologi General Purpose Graphical Processing Unit (GPGPU).

Metode CNN bukanlah ilmu baru, perkembangan awal dari rancangan (Fukushima, 1980) sebuah unsupervised artificial networks dengan nama Neocognitron yang terinspirasi dari system visual biologis rancangan Hubel dan Wiesel. Namun diakhir tahun 1990-an hingga tahun 2000-an, Neural Network

hampir dilupakan karena muncul berbagai algoritma seperti Support Vector Machines, AdaBoost dan lainnya yang dapat dieksekusi lebih cepat dan dengan performa yang lebih baik pada waktu itu, Neural Network kembali mendapatkan perhatian ketika Deep Belief Network (Hinton, 2006) membuat terobosan dengan menjadi model Handwritten Digit Recognition paling akurat yang akhirnya muncul istilah Deep Learning.[3].

Pada tahun 2012 muncul terobosan baru pada deep learning dengan memadukan arsitektur seperti drop out regularizations dan Rectified Linear Unit (ReLU) sebagai fungsi aktivasi dan data augmentation, mampu mencapai terobosan pada large scale image classification (ImageNet) yang memiliki 1000 kategori citra dari 1,2 juta citra melebihi dari performa manusia. model ini dikenal dengan nama AlexNet, proses trainingnya hanya menggunakan 1 fase saja, yaitu supervised backpropagation, sehingga menyebabkan banyak orang tidak percaya lagi pengaruh dari fase unsupervised pre-training[3].

II. HAAR CASCADES CLASSIFIER

Haar like feature atau dikenal sebagai Haar Cascades Classifier merupakan rectangular (persegi) feature, yang memberikan indikasi secara spesifik pada citra, Haar Cascade Classifier berasal dari gagasan Paul Viola dan Michael Jhon, karena itu dinamakan metode Viola & Jhon, ide dari Haar like feature adalah mengenali objek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari image object tersebut. Metode ini memiliki kelebihan komputasi yang sangat cepat, karena hanya tergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah image. Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat kunci utama, yaitu haar like feature, integral image, adaboost learning dan cascade classifier[9].

Haar Feature adalah fitur yang didasarkan pada Wavelet Haar, yaitu gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah). Untuk dua dimensi, satu terang dan satu gelap, selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik. Setiap haar-like feature terdiri dari gabungan kotak – kotak hitam dan putih, seperti pada persamaan .

$$f(x) = \text{SumBlack rectangle} - \text{SumWhite rectangle} \quad (1)$$

Fitur Haar ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang, jika nilai perbedaannya itu diatas nilai threshold, maka dapat dikatakan bahwa fitur tersebut. Nilai dari haar-like feature adalah perbedaan antara jumlah nilai-nilai piksel gray level dalam daerah kotak hitam dan daerah kotak putih. Dimana kotak putih pada haar-like feature dapat dihitung secara cepat menggunakan integral image.

III. CONVOLUTIONAL NEURAL NETWORK

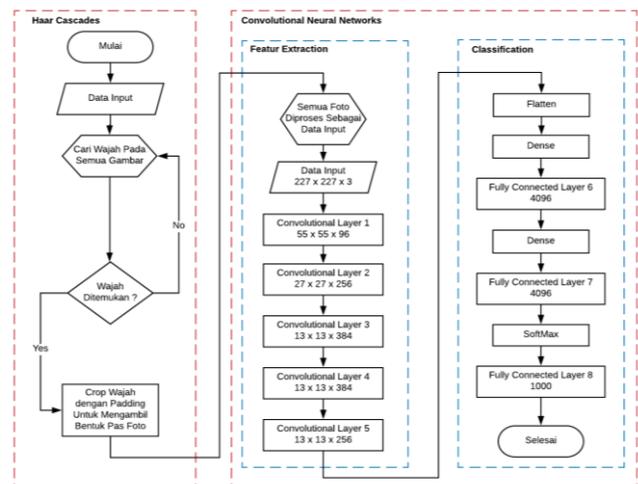
Convolutional Neural Network adalah salah satu metode machine learning dari pengembangan Multi Layer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. Convolutional Neural Network termasuk dalam jenis Deep Neural Network karena dalamnya tingkat jaringan dan

banyak diimplementasikan dalam data citra. Convolutional Neural Network memiliki dua metode; yakni klasifikasi menggunakan feedforward dan tahap pembelajaran menggunakan backpropagation. Cara kerja Convolutional Neural Network memiliki kesamaan pada Multi Layer Perceptron, namun dalam Convolutional Neural Network setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti Multi Layer Perceptron yang setiap neuron hanya berukuran satu dimensi.

IV. IDENTIFIKASI WANITA BERHIJAB

Data input sebagai training adalah kumpulan wajah yang berhijab dan tidak berhijab, untuk data training dikumpulkan dari data katalog busana online yang diambil dari beberapa katalog busana online dan website busana muslim.

Alur sistem pada blok sistem gambar 1 dimana pada data preprocessing untuk mengumpulkan data training dan validasi dari kumpulan katalog online adalah menggunakan metode Haar Cascades Clasification untuk deteksi wajah secara frontal. Setiap wajah yang ditemukan, baik berhijab atau tidak, maka akan diklasifikasi dengan kategori Hijab dan Non hijab, yang kemudian akan diteruskan ke metode Convolutional Neural Network model Alexnet untuk training, hasil model dari training akan disimpan untuk transfer learning dan digunakan kembali untuk klasifikasi foto wajah yang ditemukan Haar Cascades Classification untuk idenfifikasi wajah berhijab atau non hijab.



Gambar. 1. Blok Diagram Arsitektur Sistem

A. Data Input dan Data Output

Data input sebagai testing / ujicoba adalah Majalah digital pada aplikasi Gramedia, dengan mengambil ScreenShoot tiap halaman dan kemudian dikonversi ke bentuk PDF, majalah yang akan digunakan adalah majalah terbitan Hijabella, Joy Indonesia dan Scarf Indonesia sebagai perbandingan hasil tiap majalah. Informasi terbitan dan jumlah halaman dapat dilihat pada tabel I.

Data input sebagai training adalah kumpulan wajah yang berhijab dan tidak berhijab, untuk data training dikumpulkan dari data katalog busana online yang diambil dari beberapa katalog online dan website busana muslim. Data katalog terdapat pada table II.

TABEL I
MAJALAH DIGITAL UNTUK DATA TESTING

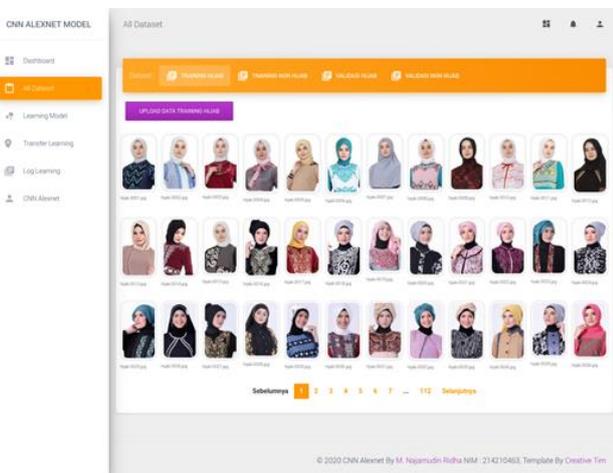
No	Majalah	Edisi	Jumlah
1	Hijabella	16, 30 Nov 2014	132 Hal
2	Hijabella	17, 07 Jan 2015	132 Hal
3	Hijabella	18, 11 Mar 2015	132 Hal
4	Joy Indonesia	Okt 2016	148 Hal
5	Joy Indonesia	Nov 2016	148 Hal
6	Joy Indonesia	Des 2016	148 Hal
7	Scarf Indonesia	31 Jan 2017	120 Hal
8	Scarf Indonesia	21 Des 2017	120 Hal
9	Scarf Indonesia	22 Jan 2018	120 Hal

Informasi terbitan dan jumlah halaman pada majalah digital

TABEL II
KATALOG DIGITAL UNTUK DATA TRAINING

No	Katalog/Website	Tahun
1	Katalog Azzura	Terbitan 2017
2	Katalog Munira	Terbitan 2018
3	Katalog Garsel Fashion	Terbitan 2017 – 2018
4	Katalog Gareu Fashion	Terbitan 2017 – 2018
5	Katalog Garucci	Terbitan 2017 – 2018
6	Katalog Guardno	Terbitan 2017 – 2018
7	Katalog Blackkelly	Terbitan 2017 – 2018
8	Katalog Inficlo	Terbitan 2017 – 2018
9	Katalog Catenzo	Terbitan 2017 – 2018
10	Katalog Golfer	Terbitan 2017 – 2018
11	Katalog Java Seven	Terbitan 2017 – 2018
12	Katalog Basama Soga	Terbitan 2017 – 2018
13	Katalog GShop Geearsy	Terbitan 2017 – 2018
14	Web Najilla Hijab	Diakses 2018
15	Web Miranda Hijab	Diakses 2018
17	Web Jenahara	Diakses 2018
18	Web Rabbani	Diakses 2018
19	Web Jilbab Flow Idea	Diakses 2018
20	Web Elzatta	Diakses 2018
21	Web Zoya	Diakses 2018
22	Web Jenahara	Diakses 2018

Katalog digital yang digunakan untuk data training



Gambar. 2. Contoh output Face Detection dengan Haar Cascade Classification data Training dari katalog digital

B. Preprocessing Dataset Training

Proses ekstraksi wajah pada setiap katalog online, menggunakan metode Haar Cascades Classification untuk menemukan wajah dengan memanfaatkan library OpenCV. Seluruh dataset yang akan digunakan untuk proses training dan validasi model Convolutional Neural Network adalah hasil dari face detection secara frontal dengan metode Haar Cascades Classification, output proses *preprocessing* seperti pada gambar 2.

Setelah wajah ditemukan, untuk data training dan validasi, dilakukan pengelompokan citra dengan kategori “Hijab” dan “Nonhijab”. Rincian pengelompokan dataset pada tabel III.

TABEL III
RINCIAN DATASET YANG DIKUMPULKAN PERKATEGORI

No	Dataset	Ket	Hijab	Non	Total
1	2.500	Training	2.000	2.000	4.000
		Validasi	500	500	1.000
2	5.000	Training	4.000	4.000	8.000
		Validasi	1.000	1.000	2.000
3	2.500 Acak	Training	2.000	2.000	4.000
		Validasi	500	500	1.000
4	5.000 Acak	Training	4.000	4.000	8.000
		Validasi	1.000	1.000	2.000

Katalog digital yang digunakan dibagi menjadi 4 kelompok.

Proses ekstraksi wajah pada majalah digital menggunakan program yang sama dengan ekstraksi wajah pada data training, yaitu menggunakan Haar Cascades Classification, perbandingan hasil ekstraksi wajah dengan resolusi berbeda dapat dilihat pada table IV.

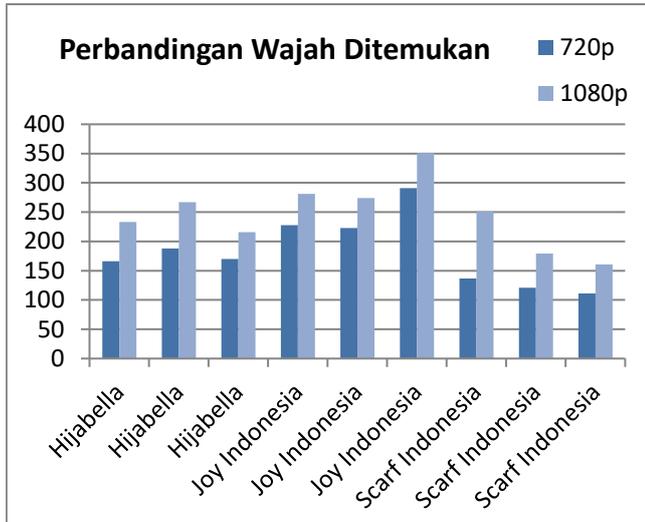
TABEL IV
PERBANDINGAN HASIL EKSTRAKSI WAJAH DENGAN RESOLUSI BERBEDA

No	Majalah	Hal	Wajah Ditemukan	
			720p	1080p
1	Hijabella Ed.16, 30 Nov 2014	132	166	233
2	Hijabella Ed.17, 07 Jan 2015	132	188	267
3	Hijabella Ed.18, 11 Mar2015	132	170	216
4	Joy Indonesia Ed.Okt 2016	148	228	281
5	Joy Indonesia Ed.Nov 2016	148	223	274
6	Joy Indonesia Ed.Des 2016	148	291	351
7	Scarf Indonesia Ed.31 Jan 2017	120	137	251
8	Scarf Indonesia Ed.21 Des 2017	120	121	179
9	Scarf Indonesia Ed.22 Jan 2018	120	111	161

Resolusi 720p menggunakan perangkat Samsung Galaxy A6, sedangkan resolusi 1080p menggunakan perangkat Xiaomi Redmi Note 7.

Dari data table IV, dapat dilihat dengan semakin besar resolusi citra sebuah majalah digital, semakin banyak wajah

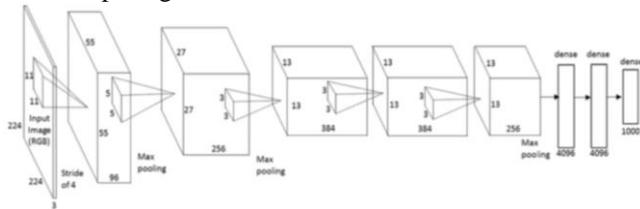
yang ditemukan dengan metode Haar Cascades Classification, Grafik perbandingan hasil ekstraksi wajah dari majalah dapat dilihat pada gambar 3.



Gambar. 3. Grafik Perbandingan Wajah Ditemukan Berdasarkan Resolusi

C. Training Model Convolutional Neural Network

Dataset yang akan digunakan untuk training model Convolutional Neural Network adalah dataset training yang didapatkan dari ekstraksi wajah katalog digital pada tabel 3.2 untuk data training. Arsitektur model yang digunakan adalah AlexNet pada gambar 4.

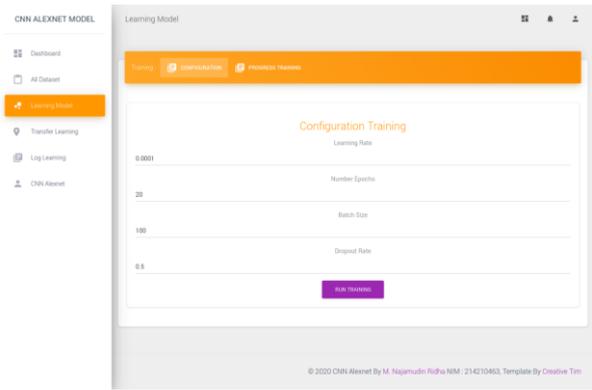


Gambar. 4. Arsitektur Model Alexnet[17]

Proses training dataset menggunakan beberapa model sesuai dengan kategori pada table III. Kemudian akan dilakukan ujicoba model training terhadap majalah digital.

D. Evaluasi Data Testing

Pada bagian preprocessing data testing di tabel IV terdapat perbandingan hasil ekstraksi wajah dengan resolusi berbeda, terbukti bahwa metode Haar Cascades Classification mampu mendeteksi lebih banyak wajah dengan resolusi yang semakin tinggi, untuk menghitung tingkat akurasi model-model yang sudah di training dengan model Convolutional Neural Network, diperlukan uji coba dengan data-data berupa majalah – majalah digital, proses perhitungan akurasi model yang sudah di transfer learning dapat melakukan model Evaluate antara data Testing dengan model yang sudah ada, namun perlu memisahkan hasil ekstraksi wajah dengan Haar Cascades Classification pada majalah digital antara wajah berhijab dan tidak berhijab menyesuaikan model Convolutional Neural Network. Pengaturan parameter untuk training model seperti pada gambar 5.



Gambar. 5. Custom Parameter untuk training model

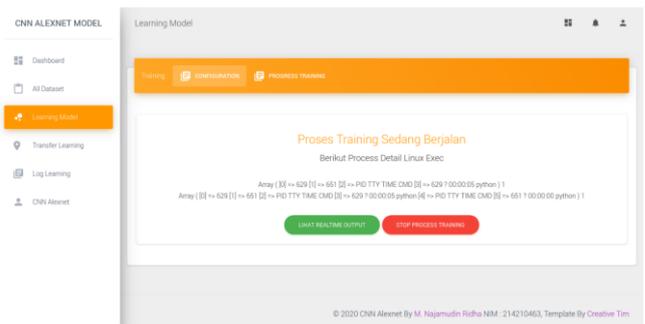
Hasil klasifikasi data wajah akan digunakan untuk testing evaluasi model-model yang sudah di training, pada data training dengan berbagai parameter, data parameter dapat dilihat pada table V.

TABEL V
DATA PARAMETER MODEL TRAINING

No	Nama	Learn	Drop Out	Batch Size	Epoch	Early Stop
1	Test 01	0.0001	0.4	100	100	-
2	Test 02	0.0001	0.4	200	100	-
3	Test 03	0.0001	0.4	300	100	-
4	Test 04	0.0001	0.4	400	100	-
5	Test 05	0.0001	0.4	500	100	-
6	Test 06	0.0001	0.5	100	100	-
7	Test 07	0.0001	0.5	200	100	-
8	Test 08	0.0001	0.5	300	100	-
9	Test 09	0.0001	0.5	400	100	-
10	Test 10	0.0001	0.5	500	100	-
11	Test 11	0.0001	0.4	100	-	25
12	Test 12	0.0001	0.4	200	-	25
13	Test 13	0.0001	0.4	300	-	25
14	Test 14	0.0001	0.4	400	-	25
15	Test 15	0.0001	0.4	500	-	25
16	Test 16	0.0001	0.5	100	-	25
17	Test 17	0.0001	0.5	200	-	25
18	Test 18	0.0001	0.5	300	-	25
19	Test 19	0.0001	0.5	400	-	25
20	Test 20	0.0001	0.5	500	-	25

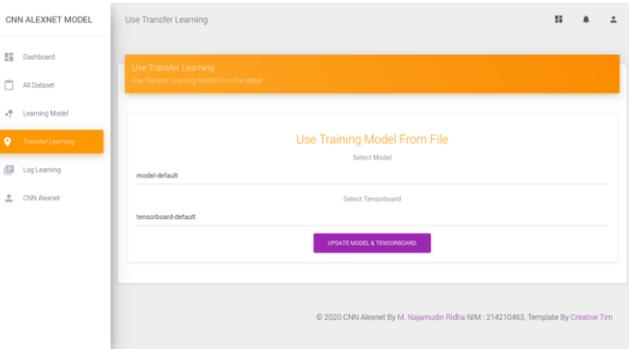
Model-model training dengan berbagai parameter

Proses training dataset pada setiap model akan disimpan dan digunakan untuk transfer learning sebagai ujicoba data testing untuk melihat nilai akurasi model yang telah di training, proses training model dapat dilihat pada gambar 6.



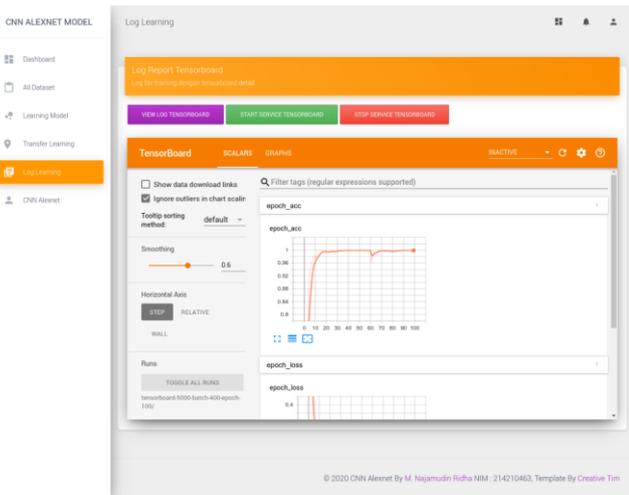
Gambar. 7. Proses Training model dengan Dataset

Setiap majalah yang akan dilakukan untuk ujicoba pada tabel II akan dievaluasi dengan model-model yang dibuat dengan berbagai parameter seperti pada table V, setiap model menggunakan 4 kelompok kategori data training seperti pada tabel III, model hasil training disimpan dengan format hadoop filesystem h5. Untuk memilih model tertentu seperti pada gambar 7.



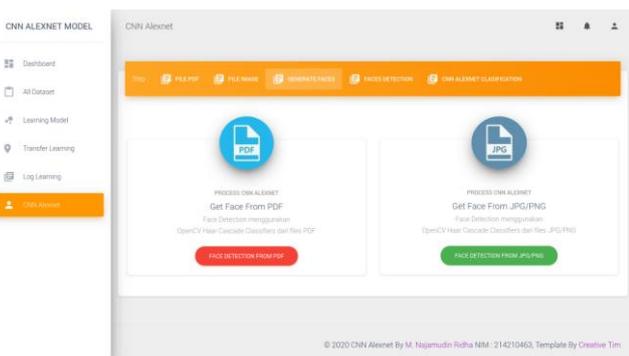
Gambar. 7. Setting Transfer Learning model

Majalah yang diujicoba menggunakan perangkat berbeda dengan resolusi 720p dan 1080p untuk melihat hasil perbandingan perbedaan resolusi terhadap model Convolutional Neural Network yang dibuat. Log proses training dapat dilihat pada gambar 8.



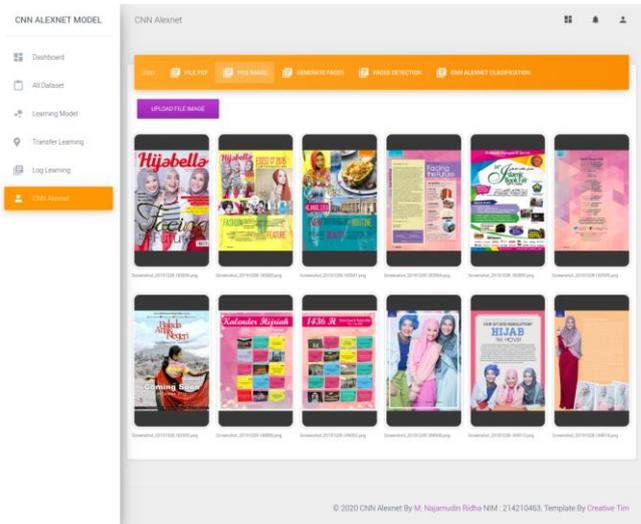
Gambar. 8. Log Proses Training dari Tensorboard

Proses testing data, bisa menggunakan data input berupa file pdf ataupun jpg/jpeg/png, untuk memilih input data testing seperti pada gambar 9.



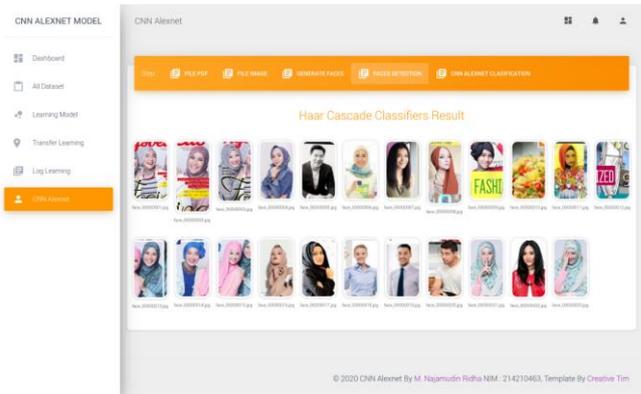
Gambar. 9. Memilih input data Testing

Untuk memudahkan evaluasi data testing, setiap proses klasifikasi dengan Convolutional Neural Network permajalah, contoh 12 halaman pertama majalah Hijabella dari file jpg seperti pada gambar 10.



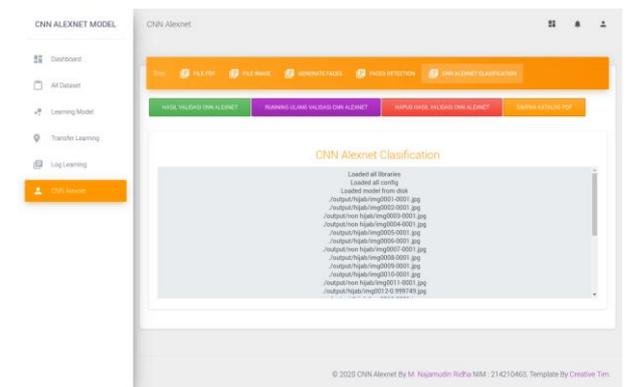
Gambar. 10. Majalah Hijabella untuk Data Testing

Proses Haar Cascade Classification untuk ekstraksi wajah yang ditemukan pada majalah untuk input data testing seperti gambar 11.



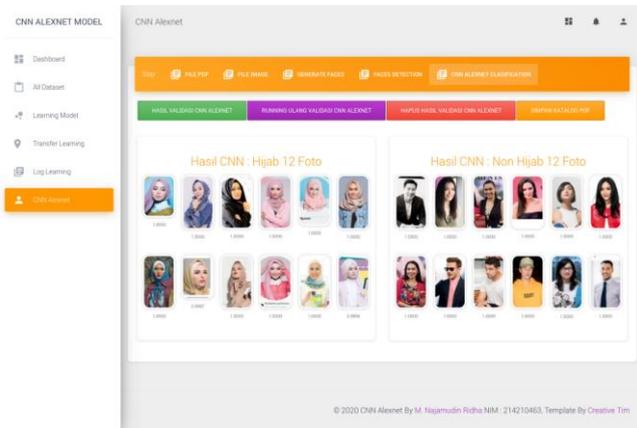
Gambar. 11. Hasil Deteksi Wajah Haar Cascades Classification

Wajah yang berhasil diidentifikasi akan diteruskan ke proses klasifikasi, apakah foto berhijab atau tidak berhijab, proses identifikasi dengan Convolutional Neural Network seperti gambar 12.



Gambar. 12. Proses identifikasi wajah berhijab dengan Convolutional Neural Network

Setelah proses selesai, hasil identifikasi wajah yang berhijab akan dibuat menjadi katalog busana muslim dalam bentuk file pdf, output hasil identifikasi wajah berhijab dengan Convolutional Neural Network seperti pada gambar 13.



Gambar. 13. Output hasil identifikasi wanita berhijab atau tidak dengan Convolutional Neural Network

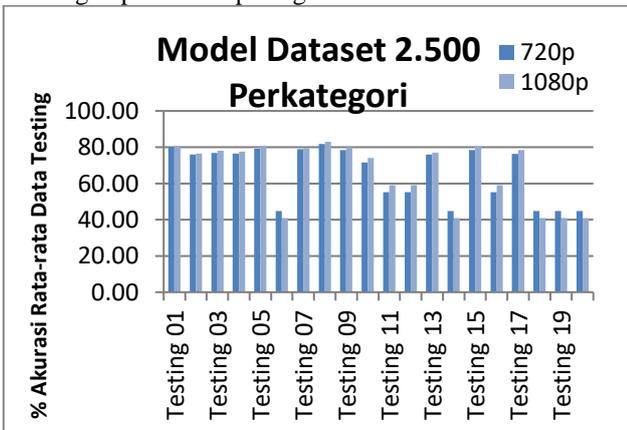
Setiap hasil identifikasi dari metode Convolutional Neural Network permajalah dihitung jumlah akurasi identifikasi wanita berhijab.

Proses perhitungan identifikasi dilakukan pada setiap model-model dengan paramter yang berbeda-beda untuk melihat model terbaik pada table V dan dengan data testing dengan resolusi 720p dan 1080p.

Untuk mempercepat proses akurasi, setiap output wajah yang terdeteksi dari metode Haar Cascades Classification, akan diklasifikasi terlebih dahulu wajah dengan hijab dan non hijab, kemudian akan dibandingkan dengan hasil klasifikasi menggunakan Convolutional Neural Network berdasarkan model-model yang telah di training. Hasilnya digunakan sebagai evaluasi data testing berdasarkan pada model dataset.

1. *Evaluasi Data Testing pada Model 2.500 Dataset*

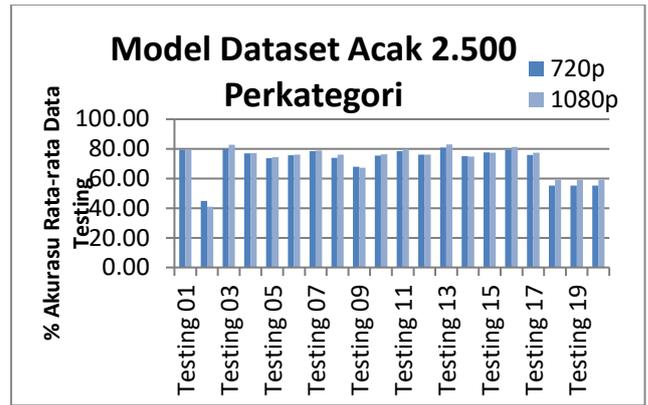
Grafik perbandingan hasil akurasi rata-rata dataset 2.500 perkategori antara 720p dengan 1080p terhadap model data training dapat dilihat pada gambar 14.



Gambar. 14. Grafik Perbandingan Rata-rata akurasi model dataset 2.500 terhadap majalah dengan resolusi 720p dan 1080p

2. *Evaluasi Data Testing pada Model 2.500 Dataset Acak*

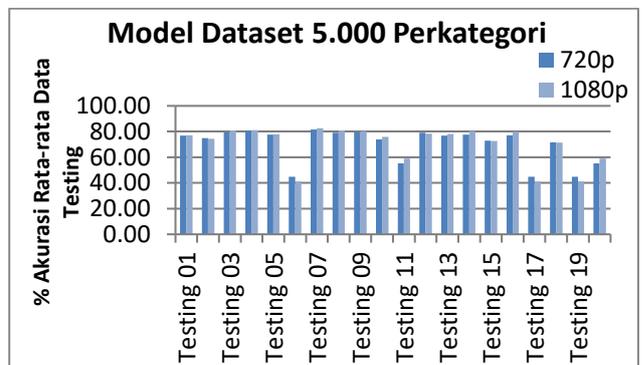
Grafik perbandingan hasil akurasi rata-rata dataset 2.500 Acak perkategori antara 720p dengan 1080p terhadap model data training dapat dilihat pada gambar 15.



Gambar. 15. Grafik Perbandingan Rata-rata akurasi model dataset 2.500 acak terhadap majalah dengan resolusi 720p dan 1080p

3. *Evaluasi Data Testing pada Model 5.000 Dataset*

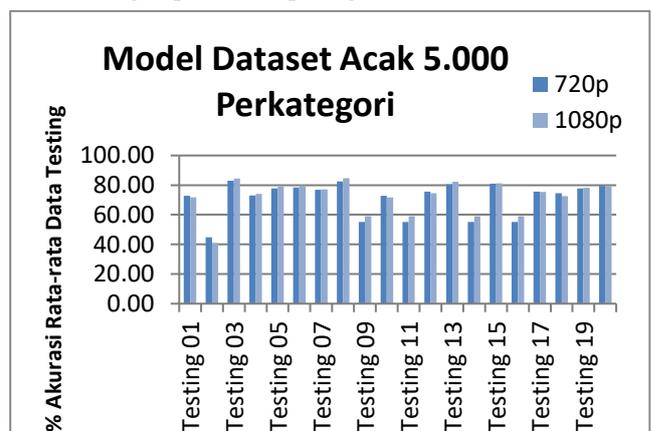
Grafik perbandingan hasil akurasi rata-rata dataset 5.000 perkategori antara 720p dengan 1080p terhadap model data training dapat dilihat pada gambar 16.



Gambar. 16. Grafik Perbandingan Rata-rata akurasi model dataset 5.000 terhadap majalah dengan resolusi 720p dan 1080p

4. *Evaluasi Data Testing pada Model 5.000 Dataset Acak*

Grafik perbandingan hasil akurasi rata-rata dataset 5.000 Acak perkategori antara 720p dengan 1080p terhadap model data training dapat dilihat pada gambar 17.



Gambar. 17. Grafik Perbandingan Rata-rata akurasi model dataset 5.000 Acak terhadap majalah dengan resolusi 720p dan 1080p

E. Hasil Uji Coba Terbaik

Setiap model yang digunakan untuk evaluasi data training menghasilkan nilai akurasi yang berbeda-beda berdasarkan data testing berupa majalah digital. Untuk akumulasi model yang terbaik dari data training sebelumnya adalah dengan

menghitung rata-rata model yang digunakan terhadap majalah-majalah digital yang digunakan. Rata-rata akurasi model dikelompokkan berdasarkan resolusi data input majalah digital resolusi 720p dan majalah digital resolusi 1080p

1. Hasil Uji Coba Data Testing 720p

Hasil uji coba model berdasarkan jenis data training dan kategori dataset yang terbaik untuk perangkat dengan resolusi data input 720p dengan kelompok testing semua majalah terdapat pada tabel VI.

TABEL VI
REKAP UJI COBA TERBAIK BERDASARKAN RESOLUSI 720P

No	Model Dataset	Majalah	Model	Avg Ac. Majalah	Avg Ac. Model
1	2.500	Majalah 03	Testing 08	69.56%	81.81 %
2	2.500 Acak	Majalah 04	Testing 13	77.17%	80.79 %
3	5.000	Majalah 03	Testing 07	74.88%	81.68 %
4	5.000 Acak	Majalah 03	Testing 03	77.02%	82.93 %

Rekap ujicoba terbaik rata-rata berdasarkan resolusi 720p terhadap semua majalah dan model

2. Hasil Uji Coba Data Testing 1080p

Hasil uji coba model berdasarkan jenis data training dan kategori dataset yang terbaik untuk perangkat dengan resolusi data input 1080p dengan kelompok testing semua majalah terdapat pada tabel VII.

TABEL VII
REKAP UJI COBA TERBAIK BERDASARKAN RESOLUSI 1080P

No	Model Dataset	Majalah	Model	Avg Ac. Majalah	Avg Ac. Model
1	2.500	Majalah 09	Testing 08	69.13%	83.06 %
2	2.500 Acak	Majalah 04	Testing 13	77.35%	82.99 %
3	5.000	Majalah 03	Testing 07	73.96%	82.63 %
4	5.000 Acak	Majalah 06	Testing 08	77.11%	84.72 %

Rekap ujicoba terbaik rata-rata berdasarkan resolusi 1080p terhadap semua majalah dan model

Hasil Model terbaik adalah model data training dengan Dataset Acak 5.000 perkategori dan dengan parameter Batch Size 300 dan Epoch 100, menghasilkan rata-rata akurasi 84.72%, sedangkan majalah dengan akurasi tertinggi adalah majalah 04 Joy Indonesia terbitan Oktober 2016 dengan akurati rata-rata adalah 77.35%.

V. KESIMPULAN

Adapun Kesimpulan yang dapat diambil berdasarkan perancangan dan analisa sistem, preprocessing data training, preprocessing data testing, uji coba sistem dan evaluasi hasil berdasarkan model transfer learning, dapat disimpulkan hal-hal berikut ini:

1. Perbedaan Resolusi dataset majalah digital yang digunakan, mempengaruhi hasil wajah yang ditemukan menggunakan Haar Cascades Classification, yaitu terjadi peningkatan wajah ditemukan dari resolusi 720p ke 1080p antara 20.62% sampai dengan 83.21% dengan rata-rata peningkatan pada setiap majalah sebesar 39.15%.

2. Peningkatan wajah yang ditemukan dengan metode Haar Cascades Classification dari perbedaan resolusi mampu menemukan ukuran wajah yang kecil, namun tidak berpengaruh besar pada hasil akurasi identifikasi wajah dengan Convolutional Neural Network, terlihat peningkatan akurasi rata-rata dari perbedaan resolusi hanya sebesar 1.54%.
3. Dengan peningkatan jumlah dataset untuk training dan validasi secara acak, mampu menambah hasil akurasi yang ditemukan, terjadi peningkatan akurasi pada dataset dari 2.500 wajah perkategori ke 5.000 wajah perkategori dengan resolusi 720p tertinggi sebesar 2.14% dan 1080p tertinggi sebesar 1.73%. Akurasi tertinggi adalah pada model dengan 5.000 wajah perkategori dengan data training dan validasi secara acak, yaitu sebesar 84.72%.

DAFTAR PUSTAKA

- [1] I. W. S. E. Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101," J. Tek. ITS, vol. 5, no. 1, pp. 65–69, 2016, [Online]. Available: <http://repository.its.ac.id/48842/>.
- [2] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural network," Comput. Vis. Image Underst., vol. 164, pp. 103–110, 2017, doi: 10.1016/j.cviu.2017.10.004.
- [3] M. Ghifary, "(Deep) Convolutional Neural Networks – Part 1," 2015. .
- [4] D. Putra, Pengolahan Citra Digital. Yogyakarta: Penerbit Andi, 2010.
- [5] M. T. . Hadi Pranoto, S.Kom, "Pengenalan Wajah : Teknik Mendeteksi Lokasi Wajah dalam Gambar dan Video," 2018. .
- [6] OpenCV, "About OpenCV." .
- [7] M. M. Hanugra Aulia Sidharta, S.T, "Introduction To Open CV," 2017.
- [8] R. Adam, "Apa Itu Google Colab," 2019. <https://structilmy.com/2019/05/mengenal-google-colab> (accessed Jan. 31, 2020).
- [9] M. Syarif, P. Studi, T. Informatika, F. I. Komputer, U. Dian, and N. Semarang, "Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login," Techno.com, vol. 14, no. 4, pp. 242–249, 2015.
- [10] S. K. Pavani, D. Delgado, and A. F. Frangi, "Haar-like features with optimally weighted rectangles for rapid object detection," Pattern Recognit., vol. 43, no. 1, pp. 160–172, 2010, doi: 10.1016/j.patcog.2009.05.011.
- [11] N. Sofia, "Convolutional Neural Network." .
- [12] U. S. Utara, U. S. Utara, and U. S. Utara, "Analisis Metode Backpropagation dengan Fungsi Aktivasi Sigmoid Bipolar dan Fungsi Linear dalam Prediksi Pertumbuhan Penduduk," 2019.
- [13] S. Sena, "Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN)," 2017, 2018. <https://medium.com/samuelsena/pengenalan-deep-lear>.
- [14] A. Sharifara, M. S. Mohd Rahim, and Y. Anisi, "A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection," Proc. - 2014 Int. Symp. Biometrics Secur. Technol. ISBAST 2014, no. November 2016, pp. 73–78, 2015, doi: 10.1109/ISBAST.2014.7013097.
- [15] H. Nguyen, S. J. Maclagan, T. D. Nguyen, and T. Nguyen, "Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring," no. Figure 1, 2017, doi: 10.1109/DSAA.2017.31.
- [16] N. Srinivas, H. Atwal, D. C. Rose, G. Mahalingam, K. Ricanek, and D. S. Bolme, "Age, Gender, and Fine-Grained Ethnicity Prediction Using Convolutional Neural Networks for the East Asian Face Dataset," Proc. - 12th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2017 - 1st Int. Work. Adapt. Shot Learn. Gesture Underst. Prod. ASL4GUP 2017, Biometrics Wild, Bwild 2017, Heteroge, pp. 953–960, 2017, doi: 10.1109/FG.2017.118.
- [17] C. N. Networks, "Lecture 9: Understanding and Visualizing Convolutional Neural Networks Fei-Fei," Stanford Convolutional Neural Networks Vis. Recognit., pp. 1–83, 2016.

Identifikasi Biji-Bijian Berdasarkan Ekstraksi Fitur Warna, Bentuk dan Tekstur Menggunakan Random Forest

Luthfi Alwi, *Akademi Pariwisata Majapahit, Arya Tandy Hermawan, Teknologi Informasi Institut Sains dan Teknologi Terpadu Surabaya*, dan Yosi Kristian, *Teknologi Informasi Institut Sains dan Teknologi Terpadu Surabaya*

Abstrak - Proses identifikasi atau pengenalan biji-bijian merupakan aspek penting dalam dunia industri pengolahan pangan. Sebuah industri pangan berskala besar, proses pencampuran beberapa macam biji-bijian dalam pengolahan sebuah produk pangan sangat memperhatikan ketepatan dalam memilih bahan. Agar tidak terjadi kesalahan yang fatal, diperlukan sebuah proses identifikasi dari bahan yang digunakan. Proses pengidentifikasian terhadap beberapa varian biji-bijian dapat dilakukan dengan cara mengekstraksi fitur dari citra (image) dengan menganalisa melalui parameter warna, bentuk dan tekstur serta melakukan proses pengklasifikasian untuk mengukur tingkat keakuratan. Penelitian ini melakukan identifikasi terhadap varian biji-bijian (padi, jagung, kacang tanah dan kedelai) dengan melakukan ekstraksi fitur warna menggunakan RGB dan HSV, ekstraksi fitur bentuk menggunakan Morphological Threshold dan ekstraksi fitur tekstur menggunakan Grey Level Co-occurrence Matrix (GLCM) dan Local Binary Pattern (LBP). Untuk proses pengklasifikasian, peneliti menggunakan metode Random Forest Classifier (RF) untuk mendapatkan tingkat akurasi yang tinggi dengan batasan-batasan yang mempengaruhi keakuratan dalam proses pengklasifikasian untuk dikembangkan dalam proses selanjutnya. Peneliti menggunakan tools MATLAB R2015b untuk proses identifikasi mulai dari proses ekstraksi fitur sampai proses klasifikasinya. Berdasarkan hasil pengujian yang dilakukan didapatkan tingkat akurasi sebesar 99.8 %. Dapat disimpulkan bahwa pengambilan dataset berupa gambar atau image biji-bijian yang diteliti dapat dijadikan patokan untuk pengidentifikasian dan dapat dikembangkan dalam proses selanjutnya.

Kata kunci: *Identifikasi, Ekstraksi Fitur, RGB, HSV, Morphological Threshold., GLCM, LBP, RF, MATLAB*

I. PENDAHULUAN

Proses identifikasi atau pengenalan biji-bijian merupakan aspek penting dalam dunia industri pengolahan pangan. Sebuah industri pangan berskala besar, proses pencampuran beberapa macam biji-bijian dalam pengolahan sebuah produk pangan sangat memperhatikan ketepatan dalam memilih bahan agar tidak terjadi kesalahan dalam proses produksi karena berpengaruh pada hasil akhir dari sebuah produksi. Agar tidak terjadi kesalahan yang fatal, diperlukan sebuah proses identifikasi dari bahan yang digunakan. Dengan sebuah sensor (intelligent camera) yang digunakan dari hasil

sebuah proses identifikasi maka sebuah proses produksi produk pangan dapat berjalan dengan baik dan tidak terjadi kesalahan dalam pencampuran bahan.

Proses pengidentifikasian terhadap beberapa varian biji-bijian dapat dilakukan dengan cara mengekstraksi fitur dari citra (image) dengan menganalisa melalui parameter warna, bentuk dan tekstur serta melakukan proses pengklasifikasian untuk mengukur tingkat keakuratan. Proses identifikasi berdasarkan parameter fitur warna, bentuk dan tekstur dengan metode computer vision diharapkan dapat menjamin tingginya tingkat akurasi dari sebuah varian biji-bijian dan bisa dijadikan acuan untuk penelitian selanjutnya.

Citra digital merupakan gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan atau diskrit nilai digital yang disebut pixel/ picture elements.[1] Citra sebagai salah satu komponen multimedia memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya akan informasi. Maksud dari “citra kaya akan informasi” adalah citra dapat memberikan informasi yang lebih banyak dibandingkan dengan informasi yang disajikan dalam bentuk teks[2]

Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi.[3]

Ekstraksi fitur adalah proses pengindeksan suatu database berupa citra (image) dengan isinya. Salah satu proses ekstraksi fitur adalah menganalisa berdasarkan isi visual seperti warna, bentuk dan tekstur. Setelah proses ekstraksi fitur, dilakukan proses klasifikasi untuk menentukan tingkat keakuratan dari proses identifikasi yang telah dilakukan. [4]

II. TINJAUAN PUSTAKA

A. Image Acquisition

Berdasarkan penelitian yang dilakukan oleh Dongming Li, Yongfu Liu And Liang Gao [5] yang menerangkan bahwa sebuah gambar (image) dapat dipelajari dan dieksplorasi dengan berbagai parameter yang dapat diekstraksikan mulai dari fitur bentuk, warna, ukuran dan tekstur dengan melakukan teknologi pengenalan (recognition) karakteristik dari sebuah gambar (image). Recognition adalah proses untuk memberikan label pada objek sesuai dengan fitur-fitur yang dimiliki oleh objek tersebut.

Recognition dilakukan dimulai dengan proses pengambilan citra (image) yang diperlukan sebagai dataset

Luthfi Alwi, Akademi Pariwisata Majapahit, Surabaya, Jawa Timur, Indonesia (e-mail: luthfi.alwi@yahoo.com)

Arya Tandy Hermawan, Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Jawa Timur, Indonesia (e-mail: arya@stts.edu)

Yosi Kristian, Departemen Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Jawa Timur, Indonesia (e-mail: yosi@stts.edu)

dengan menggunakan beberapa perangkat seperti kamera, scanner, dan lain-lain atau yang sering disebut dengan Image acquisition. Proses pengambilan gambar (image) biji yang dilakukan peneliti dilakukan dengan menggunakan kamera merk Nikon D3400 lensa standart dan perlengkapannya. Selain itu proses pengambilan gambar (image) biji pada penelitian ini dilakukan dengan mendownload melalui Google Image Search. Ini dilakukan untuk mendapatkan varian biji yang bermacam-macam secara acak/random dengan jarak pengambilan gambar yang berbeda-beda karena biji- bijian yang diteliti mempunyai varian lebih dari satu.

B. Otsu Algorithm

Setelah data input didapatkan, selanjutnya dilakukan tahapan selanjutnya yaitu melakukan proses thresholding untuk memisahkan antara image foreground dan background dalam format gray image (Binarization) dimana hasil proses ini digunakan untuk melakukan ekstraksi fitur bentuk. Metode yang digunakan untuk proses thresholding ini adalah Otsu Algorithm. Metode Otsu thresholding diperkenalkan pertama kali oleh Nobuyuki Otsu, dalam jurnal ilmiahnya yang berjudul “A Threshold Selection Method from Grayscale Histogram” pada tahun 1979 Metode Otsu thresholding merupakan metode segmentasi yang cukup akurat dalam mendapatkan daerah yang merupakan obyek tersegmentasi dengan menggunakan histogram grayscale.[6]

Metode Otsu ini didasarkan pada histogramnya. Histogram menunjukkan sebarang nilai intensitas dari tiap piksel pada citra dalam 1 dimensi. Jadi sumbu x biasanya menyatakan level intensitas yang berbeda sedangkan sumbu y menyatakan jumlah piksel yang memiliki nilai intensitas tersebut. Dengan menggunakan histogram kita bisa melakukan pengelompokan terhadap piksel-piksel dalam citra. Pengelompokan ini didasarkan pada nilai ambang batas atau threshold.[7] Sebuah threshold yang mampu memisahkan kelas-kelas sehingga piksel-piksel antar kelas memiliki nilai intensitas yang berbeda maka threshold tersebut dikatakan sudah optimal.

Untuk sebuah gambar, metode segmentasi untuk foreground dan background dilambangkan dengan t, sedangkan untuk foreground dilambangkan dengan w₀, dan nilai mean nya dilambangkan dengan μ₀ sedangkan untuk background dilambangkan dengan w₁, dan nilai mean nya dilambangkan dengan μ₁. Nilai Mean keseluruhan image dihitung dengan rumus:

$$\mu = w_0 \times \mu_0 + w_1 \times \mu_1$$

Sedangkan nilai persamaan yang dibentuk (objective function) dengan menggunakan OTSU Algorithm :

$$g(t) = w_0 \times (\mu_0 - \mu)^2 + w_1 \times (\mu_1 - \mu)^2$$

Dimana g(t) adalah nilai on-class variance dan metode OTSU Algorithm dipakai untuk mendapatkan nilai maksimum dari g(t) dan metode ini dianggap yang terbaik.[8]

Sedangkan untuk ekstraksi fitur warna dan fitur tekstur tidak melalui tahapan ini.

C. Ekstraksi Warna RGB to HSV

Ekstraksi fitur warna cocok untuk menggambarkan dan mewakili gambar berwarna. Ekstraksi fitur warna adalah menganalisa warna dari sebuah citra (image) dimana sebuah citra tersusun dari piksel-piksel yang memiliki ukuran

intensitas warna masing-masing. Sebaran warna di tiap-tiap piksel ditunjukkan oleh histogram. Histogram menunjukkan distribusi piksel berdasarkan intensitas graylevel (derajat keabuan) yang dimiliki tiap-tiap piksel. Penggunaan histogram sebagai metode ekstraksi ciri didasarkan pada perbedaan sebaran atau distribusi piksel di masing-masing gambar. Histogram warna sangat leluasa untuk dibentuk dan dapat disusun dari gambar dengan berbagai rentang warna. Histogram warna dari sebuah gambar diproduksi dengan kuantisasi pada warna di gambar ke dalam bilangan biner, dan menghitung jumlah piksel pada tiap bilangan biner tersebut. [9]

Color Histogram adalah representasi dari distribusi warna dalam suatu gambar. Color Histogram suatu gambar dihasilkan oleh kuantisasi warna dalam gambar menjadi sejumlah piksel dalam satuan bin dan menghitung jumlah piksel gambar dalam setiap satuannya. Color Histogram warna dalam RGB dan HSV ruang warna dengan 8 bin di setiap saluran warna akan digunakan untuk mewakili data gambar.

Pada penelitian ini, ekstraksi fitur warna yang digunakan adalah RGB to HSV dimana hasil ekstraksinya didapatkan 18 fitur yang akan dijadikan parameter untuk pengklasifikasian. Ke-18 fitur itu antara lain R mean, R max, R min, G mean, G max, G min, B mean, B max, B min, H mean, H max, H min, S mean, S max, S min, V mean, V max, dan V min.

D. Ekstraksi Bentuk Morphology Threshold

Ekstraksi fitur bentuk merupakan konfigurasi oleh garis dan kontur. Fitur bentuk dikategorikan bergantung pada teknik yang digunakan. Kategori tersebut adalah berdasarkan batas (boundary-based) dan berdasarkan daerah (region-based). Teknik berdasarkan batas (boundary-based) menggambarkan bentuk daerah dengan menggunakan karakteristik eksternal.

Sebelum dilakukan ekstraksi fitur bentuk, terlebih dahulu melewati tahapan thresholding untuk memisahkan antara image foreground dan background dalam format gray image (Binarization).Metode yang digunakan untuk proses thresholding ini adalah Otsu Algorithm.

Setelah metode Threshold dilakukan, Morphology Extraction Method digunakan untuk mengekstraksi komponen citra yang berguna dalam merepresentasikan dan mendeskripsikan bentuk daerah seperti batas dan rangka.

Proses morfologi suatu image adalah merupakan kumpulan operasi non linear yang berkaitan dengan bentuk atau morphology dalam suatu image. Secara praktek, sistim biner sering digunakan dalam proses morfologi, yaitu proses bit 1 atau yang dikenal dengan foreground dan bit 0 atau background dengan cara merubah bagian tertentu dari foreground menjadi daerah background dan sebaliknya merubah sebagian background untuk menjadi daerah foreground. Manfaat penggunaan proses morfologi yaitu untuk menghilangkan noise yang ada.

Proses dilation dalam morfologi image identik dengan menambahkan pixels dalam lingkup image asal, dengan cara menempatkan satu demi satu pusat penataan elemen untuk masing-masing pixel background. Bila sembarang pixel neighbourhood bernilai pixel foreground (nilai 1) maka pixel background dirubah ke foreground.

Proses erosion merupakan proses menghilangkan pixel dalam lingkup objek image dengan cara meletakkan pusat penataan elemen satu demi satu dalam pixel foreground (nilai

1). Bila terdapat pixel neighbourhood bernilai pixel background (nilai 0), maka nilai foreground tersebut di rubah ke background.

Dengan proses erosi berakibat penyusutan ukuran obyek image sehingga dapat digunakan untuk memisahkan objek yang saling gandeng satu sama lain. Sedangkan dilation akan menaikkan ukurannya sehingga dapat menebalkan objek image dan menyambung object yang terputus ataupun meratakan tepi objek yang rusak.

Opening merupakan proses morfologi dengan menggunakan proses erosi dan dilanjutkan dengan proses dilation menggunakan penataan elemen (structuring element) yang sama. Opening banyak digunakan untuk proses menghilangkan objek kecil dalam suatu image tetapi tetap masih mempertahankan bentuk aslinya.

Sedangkan closing merupakan proses morfologi dengan cara melakukan operasi dilation yang diteruskan dengan operasi erosi dengan menggunakan penataan elemen yang sama. Metode closing digunakan bila ingin menutup lubang objek dengan tetap mempertahankan bentuk aslinya. [10]

Untuk ekstraksi fitur bentuk mencari nilai Rasio “L” (Area/Perimeter), Metric (roundness), Metric, Eccentricity, Equivalent Diameter, Rasio “M” (Major Axis Length/Minor Axis Length) untuk dijadikan parameternya dan didapatkan 5 parameter untuk masing-masing biji.

Untuk parameter Rasio L didapatkan dengan membagi antara Area dan perimeternya karena nilai rata-rata Area dan Perimeter dari image yang digunakan karena jarak pengambilan image yang berbeda-beda.

Begitu juga dengan Rasio M didapatkan dengan membagi antara Major Axis Length dan Minor Axis Length untuk mendapatkan panjang garis rata-rata dari setiap image biji-bijian yang diteliti.

E. Ekstraksi Tekstur Grey Level Co-occurrence Matrix (GLCM) dan Local Binary Pattern (LBP).

Ekstraksi fitur tekstur sebuah image merupakan informasi berupa susunan struktur permukaan suatu gambar. Fitur tekstur cocok untuk memvisualisasikan pola dan sifat permukaan. Ini menyediakan informasi tentang variasi intensitas permukaan gambar.

Ekstraksi fitur tekstur sesuai untuk memvisualisasikan penampakan pola dan permukaan. Hal itu menyediakan informasi terkait variasi intensitas dari permukaan obyek pada gambar. Pada paper ini, metode yang dilakukan adalah dengan Gray Level Co-occurrence Level (GLCM)[11] dan Local Binary Pattern (LBP) [12].

Gray Level Co-occurrence Level (GLCM) adalah metode statistik yang menggunakan hubungan spasial dari piksel. Ini adalah perhitungan dari angka pertemuan dua piksel yang berdekatan dalam sebuah gambar, pada lokasi yang memiliki nilai abu-abu sebesar i dan j. ukuran dari GLCM tergantung pada tingkat keabu-abuan yang dapat mengatur ukuran gambar dengan mengatur nilai intensitas. Untuk membuat

GLCM, dua parameter menunjukkan hubungan piksel dari arah yang berbeda dan jaraknya didefinisikan dalam offset.

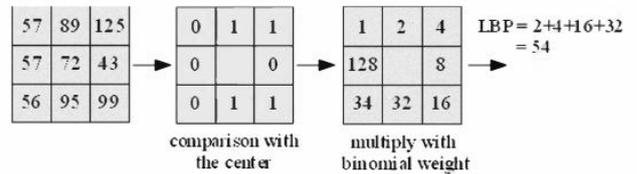
Untuk ekstraksi fitur tekstur dengan metode GLCM (Gray Level Co-occurrence Matrix) mencari nilai rata-rata Contrast, Correlation, Energy, Homogeneity untuk dijadikan parameternya dan didapatkan 16 parameter untuk masing-masing biji. [13] Parameter tekstur yang didapat diturunkan dari metode GLCM ini adalah Contrast, Correlation, Energy, Homogeneity untuk masing-masing sudut. Peneliti menggunakan 4 arah sudut pandang (degree = 0°, 45°, 90°, 135°), 8-jarak ketetanggaan (D = 1, 2,...,8) dan radius = 1.

Pola biner lokal (LBP) adalah jalur non-parameter yang menyederhanakan struktur spasial lokal dari suatu gambar. Ini adalah bentuk invariasi pada perubahan warna menjadi hitam putih. Untuk menghitung LBP, kotak berukuran 3x3 diterapkan pada gambar. Dengan diberi kotak, piksel yang akan lebih terlihat di antar piksel pusat dan sekitarnya dibandingkan dengan menggunakan persamaan. Hasil bilangan biner dikalikan dengan berat binomial yang diberikan untuk tiap piksel. Terakhir, hasil yang didapatkan dijumlahkan untuk memperoleh nilai LBP.

$$LBP(x_c, y_c) = \sum_{n=0}^7 s(i_n - i_c) 2^n$$

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Di mana in dan i_c adalah nilai kerapatan dari piksel pusat dan sekitarnya. [14]



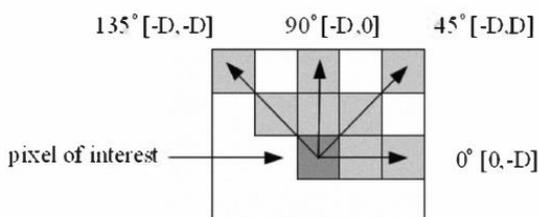
Untuk ekstraksi fitur tekstur dengan metode LBP (Local Binary Pattern) mencari nilai rata-rata dari histogram bin dan didapatkan 59 parameter yang digunakan untuk proses klasifikasi.

F. Random Forest

Setelah dilakukan ekstraksi fitur, dilakukan proses pengklasifikasian dengan berbagai metode untuk mendapatkan tingkat akurasi yang terbaik.

Pada paper ini, setelah proses ekstraksi fitur dilakukan, dan didapatkan beberapa parameter tiap ekstraksi fitur (warna, tekstur, bentuk) selanjutnya melakukan proses klasifikasi dengan menggunakan Random Forest dengan dataset dari nilai parameter yang didapat dari hasil ekstraksi yang tersebut diatas.

Breiman [15] mengusulkan Random Forest (RF), teknik klasifikasi yang disusun dengan menyusun sekumpulan pohon yang diinginkan. RF menggunakan variable respon ciri yang berbeda, yang akan di setiap pohon dan mengubah bagaimana klasifikasi atau pohon regresi terbentuk: setiap titik dipisah menggunakan yang terbaik di antara urutan prediksi yang dipilih secara acak dalam titik itu, dan pohon akan tumbuh meninggi tanpa mengalami pemendekan. Untuk prediksi data baru, RF mengumpulkan semua data hasil dari



pohon-pohon. Hal ini menjadi efektif dan cepat untuk data dengan jumlah yang banyak dan telah menunjukkan apabila dapat digunakan dengan baik dibandingkan program pengelompok yang lain.

Random Forest Classifier [16] merupakan pengembangan dari decision tree dengan menggunakan beberapa tree, dimana setiap tree telah dilakukan training menggunakan sampel individu dan setiap atribut dipecah pada tree yang dipilih antara subset yang bersifat acak. Dan pada proses klasifikasi, individunya didasarkan pada vote dari suara terbanyak pada kumpulan populasi tree.

Pada umumnya model klasifikasi ada dua proses yang akan dilakukan yaitu proses training dan proses testing. Data training didapatkan dari proses ekstraksi fitur (warna, tekstur dan bentuk) dengan berbagai parameternya untuk masing-masing biji yang diteliti seperti yang telah disebutkan diatas.

Sedangkan data testing, mengambil data secara random (acak) untuk masing-masing biji yang sudah dilakukan proses ekstraksi fitur.

Random forest yang dihasilkan memiliki banyak tree, dan setiap tree ditanam dengan cara yang sama. Tree dengan variabel x akan ditanam sejauh mungkin dengan tree dengan variabel y . dan dalam perkembangannya, sejalan dengan bertambahnya data set, maka tree pun ikut berkembang. Untuk prediksi data baru, RF mengumpulkan semua data hasil dari pohon-pohon. Hal ini menjadi efektif dan cepat untuk data dengan jumlah yang banyak. Beberapa fungsi learning yang dihasilkan Random Forest digunakan strategi ensemble "bagging" untuk mengatasi masalah overfitting apabila dihadapkan data set yang kecil.[17]

Hasil proses klasifikasi sebuah dataset secara random/acak dapat diidentifikasi secara tepat dari masing-masing biji yang diteliti. dan menunjukkan tingkat akurasi dari proses identifikasi 4 (empat) biji-bijian yaitu padi, jagung, kacang tanah dan kedelai.

III. PERANCANGAN

Pada pokok bahasan ini akan dibahas mengenai desain arsitektur dari system, alternative algoritma yang akan digunakan. Untuk blok diagram yang mempresentasikan proses penelitian digambarkan pada gambar 5.

A. Pre Processing

Proses pengambilan gambar (image) biji yang dilakukan peneliti dilakukan dengan menggunakan kamera merk Nikon D3400 lensa standart dan perlengkapannya. Selain itu proses pengambilan gambar (image) biji pada penelitian ini dilakukan dengan mendownload melalui Google Image Search. Ini dilakukan untuk mendapatkan varian biji yang bermacam-macam secara acak/random dengan jarak pengambilan gambar yang berbeda-beda karena biji- bijian yang diteliti mempunyai varian lebih dari satu..

B. Thresholding

Proses ini untuk memisahkan antara image foreground dan background dalam format gray image (Binarization) dimana hasil proses ini digunakan untuk melakukan ekstraksi fitur bentuk. Sedangkan untuk ekstraksi fitur warna dan fitur tekstur tidak melalui tahapan ini. Metode yang digunakan untuk proses thresholding ini adalah Otsu Algorithm.

C. Ekstraksi Fitur

Proses pengindeksan suatu database berupa citra (image) dengan isinya. Salah satu proses ekstraksi fitur adalah menganalisa berdasarkan isi visual seperti warna, bentuk dan tekstur.

• Ekstraksi Fitur Warna

Proses menganalisa warna dari sebuah citra (image) dimana sebuah citra tersusun dari piksel-piksel yang memiliki ukuran intensitas warna masing-masing. Sebaran warna di tiap-tiap piksel ditunjukkan oleh histogram. Histogram menunjukkan distribusi piksel berdasarkan intensitas graylevel (derajat keabuan) yang dimiliki tiap-tiap piksel. Penggunaan histogram sebagai metode ekstraksi ciri didasarkan pada perbedaan sebaran atau distribusi piksel di masing-masing gambar.

• Ekstraksi Fitur Tekstur

Merupakan informasi berupa susunan struktur permukaan suatu gambar. Fitur tekstur cocok untuk memvisualisasikan pola dan sifat permukaan. Ini menyediakan informasi tentang variasi intensitas permukaan gambar..

• Ekstraksi Fitur Bentuk

Merupakan konfigurasi oleh garis dan kontur. Fitur bentuk dikategorikan bergantung pada teknik yang digunakan. Kategori tersebut adalah berdasarkan batas (boundary-based) dan berdasarkan daerah (region-based). Teknik berdasarkan batas (boundary-based) menggambarkan bentuk daerah dengan menggunakan karakteristik eksternal.

D. Klasifikasi

Setelah proses ekstraksi fitur dilakukan, dan didapatkan beberapa parameter tiap ekstraksi fitur (warna, tekstur, bentuk) selanjutnya melakukan proses klasifikasi. Metode yang digunakan untuk proses klasifikasi ini adalah menggunakan Random Forest dengan dataset dari nilai parameter yang didapat dari hasil ekstraksi fitur.

IV. PENGUJIAN DAN ANALISIS

A. Dataset

Obyek penelitian adalah image (gambar) dari 4 varian biji-bijian (jagung, padi, kacang tanah, kedelai) dalam format gambar RGB dan JPEG dengan dimensi 300x500 pixel. Jumlah gambar yang diambil untuk 1 varian biji sebanyak 200 gambar. Jadi total gambar yang dijadikan obyek penelitian sejumlah 800 gambar. Untuk uji training sebanyak 75 % dari total gambar dan untuk testing sebanyak 25 % dari total gambar. Beberapa gambar yang menjadi data input:

a. Biji jagung



Gambar 1 Biji jagung

b. Biji kedelai



Gambar 2 Biji kedelai

c. Biji kacang



Gambar 3 Biji kacang

d. Biji padi



Gambar 4 Biji padi

B. Pengujian

Untuk pengerjaan tahap thresholding, tahap ekstraksi fitur (warna, bentuk dan tekstur) sampai tahap klasifikasi menggunakan library MATLAB R2015b.

Untuk uji training sebanyak 75 % dari total gambar dan untuk testing sebanyak 25 % dari total gambar. Untuk uji data training didapat sebanyak 150 gambar untuk masing-masing gambar biji yang sudah didapat.

Hasil Ekstraksi fitur warna dari data training sebanyak 75 % dari total gambar biji Jagung/Kacang Tanah/Kedelai/ Padi berdasarkan fitur warna RGB dan HSV menghasilkan 18 parameter. Ke-18 parameter itu antara lain R mean, R max, R min, G mean, G max, G min, B mean, B max, B min, H mean, H max, H min, S mean, S max, S min, V mean, V max, dan V min.

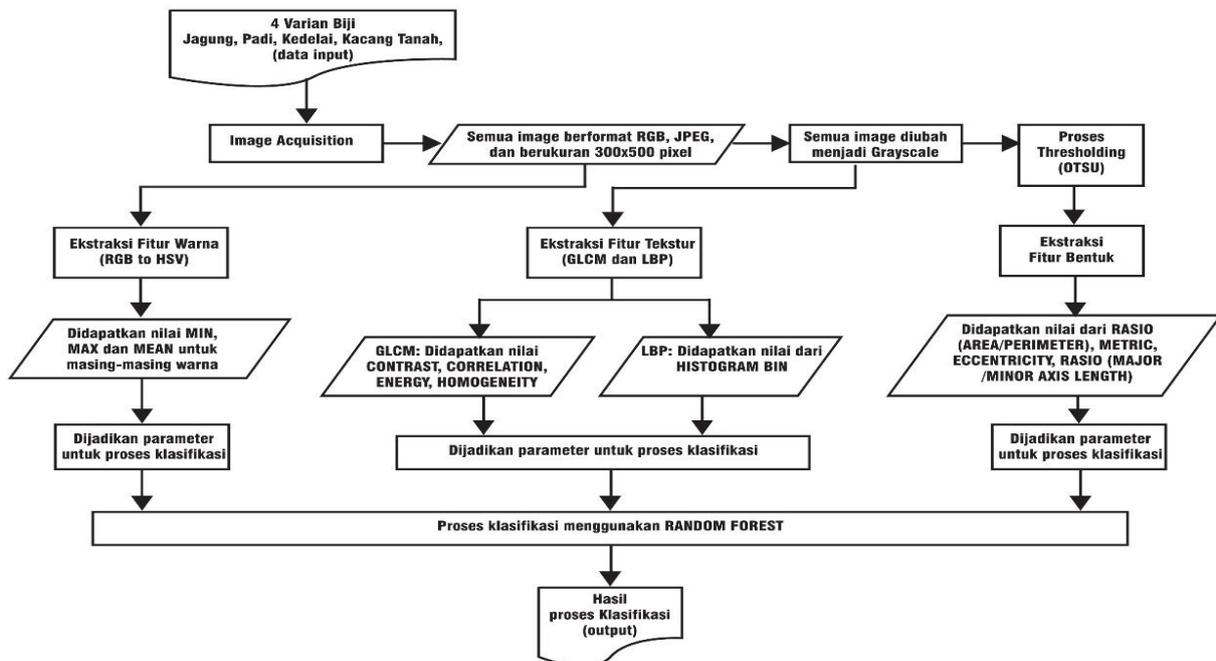
Untuk ekstraksi fitur tekstur dengan metode GLCM (Gray Level Co-occurrence Matrix) mencari nilai rata-rata Contrast, Correlation, Energy, Homogeneity untuk dijadikan parameter dan didapatkan 16 parameter untuk masing-masing biji.

Parameter tekstur yang didapat diturunkan dari metode GLCM ini adalah Contrast, Correlation, Energy, Homogeneity untuk masing-masing sudut. Peneliti menggunakan 4 arah sudut pandang ($degree = 0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}$), 8-jarak ketetanggaan ($D = 1, 2, \dots, 8$) dan radius = 1.

Untuk ekstraksi fitur tekstur dengan metode LBP mencari nilai rata-rata dari histogram bin dan didapatkan 59 parameter yang digunakan untuk proses klasifikasi.

Untuk ekstraksi fitur bentuk mencari nilai Rasio "L" (Area/Perimeter), Metric (roundness), Metric, Eccentricity, Equivalent Diameter, Rasio "M" (Major Axis Length/Minor Axis Length) untuk dijadikan parameter dan didapatkan 5 parameter untuk masing-masing biji.

Untuk pengerjaan uji testing yang diambil dari data training sebanyak 25 %, peneliti melakukan 5 kali uji testing untuk proses pengklasifikasian dengan pengambilan data gambar secara acak (random). Pada setiap uji testing diambil



Gambar 5 Alur Kerja

sebanyak 50 gambar untuk masing-masing gambar biji (jagung, kacang, kedelai, padi) sehingga didapat 200 gambar (image).

Pada uji testing ke-1, didapatkan tingkat akurasi sebesar 99 % dengan tabel confusion matrix dan tabel classification error untuk menampilkan tree dengan tingkat kesalahan yang paling kecil.

Pada uji testing ke-1 ini, tingkat classification error yang didapat terdapat pada number of grown trees ke 56 dengan nilai 0,0233. Dari nilai tersebut dapat dibangun sebuah decision tree untuk menentukan hasil klasifikasi dari sekumpulan data yang sudah dibuat.

TABEL 1
CONFUSION MATRIX TESTING 1

	Jagung	Kacang	Kedelai	Padi
Jagung	50	0	0	0
Kacang	0	50	0	0
Kedelai	0	0	48	2
Padi	0	0	0	50

Pada uji testing ke-2, didapatkan tingkat akurasi sebesar 100 % dengan tabel confusion matrix dan tabel classification error untuk menampilkan tree dengan tingkat kesalahan yang paling kecil.

Pada uji testing ke-2 ini, tingkat classification error yang didapat terdapat pada number of grown trees ke 56 dengan nilai 0,0233. Dari nilai tersebut dapat dibangun sebuah decision tree untuk menentukan hasil klasifikasi dari sekumpulan data yang sudah dibuat.

TABEL 2
CONFUSION MATRIX TESTING 2

	Jagung	Kacang	Kedelai	Padi
Jagung	50	0	0	0
Kacang	0	50	0	0
Kedelai	0	0	50	0
Padi	0	0	0	50

Pada uji testing ke-3, didapatkan tingkat akurasi sebesar 100 % dengan tabel confusion matrix dan tabel classification error untuk menampilkan tree dengan tingkat kesalahan yang paling kecil.

Pada uji testing ke-3 ini, tingkat classification error yang didapat terdapat pada number of grown trees ke 57 dengan nilai 0,0233. Dari nilai tersebut dapat dibangun sebuah decision tree untuk menentukan hasil klasifikasi dari sekumpulan data yang sudah dibuat.

TABEL 3
CONFUSION MATRIX TESTING 3

	Jagung	Kacang	Kedelai	Padi
Jagung	50	0	0	0
Kacang	0	50	0	0
Kedelai	0	0	50	0
Padi	0	0	0	50

Pada uji testing ke-4, didapatkan tingkat akurasi sebesar 100 % dengan tabel confusion matrix dan tabel classification error untuk menampilkan tree dengan tingkat kesalahan yang paling kecil.

Pada uji testing ke-4 ini, tingkat classification error yang didapat terdapat pada number of grown trees ke 57 dengan

nilai 0,0233. Dari nilai tersebut dapat dibangun sebuah decision tree untuk menentukan hasil klasifikasi dari sekumpulan data yang sudah dibuat.

TABEL 4
CONFUSION MATRIX TESTING 4

	Jagung	Kacang	Kedelai	Padi
Jagung	50	0	0	0
Kacang	0	50	0	0
Kedelai	0	0	50	0
Padi	0	0	0	50

Pada uji testing ke-5, didapatkan tingkat akurasi sebesar 100 % dengan tabel confusion matrix dan tabel classification error untuk menampilkan tree dengan tingkat kesalahan yang paling kecil.

Pada uji testing ke-5 ini, tingkat classification error yang didapat terdapat pada number of grown trees ke 56 dengan nilai 0,0233. Dari nilai tersebut dapat dibangun sebuah decision tree untuk menentukan hasil klasifikasi dari sekumpulan data yang sudah dibuat.

TABEL 5
CONFUSION MATRIX TESTING 5

	Jagung	Kacang	Kedelai	Padi
Jagung	50	0	0	0
Kacang	0	50	0	0
Kedelai	0	0	50	0
Padi	0	0	0	50

Peneliti menggunakan metode klasifikasi Random Forest Classifier dengan membangun sebanyak 100 tree untuk mendapatkan keputusan yang terbaik berdasarkan tingkat classification error yang dibuat. Setelah didapatkan nilai dari tingkat classification error, dipilih nilai yang paling kecil pada number of grown trees. Untuk tampilan tree, peneliti memilih menggunakan RULE BASED DECISION TREE FOR CLASSIFICATION berupa text

Pada 5 uji testing yang dilakukan, uji testing ke-1, ke-2 dan ke-5 pada tree ke-56 didapatkan nilai classification error paling kecil yaitu sebesar 0,0233. Sedangkan pada uji testing ke-3 dan ke-4 pada tree ke-57 didapatkan nilai classification error paling kecil yaitu sebesar 0,0233.

Hasil dari penelitian menunjukkan persentase tingkat akurasi dari 5 uji testing adalah testing ke-1 sebesar 99%. Sedangkan tingkat akurasi uji testing ke-2 sampai ke-5 sebesar 100 %.

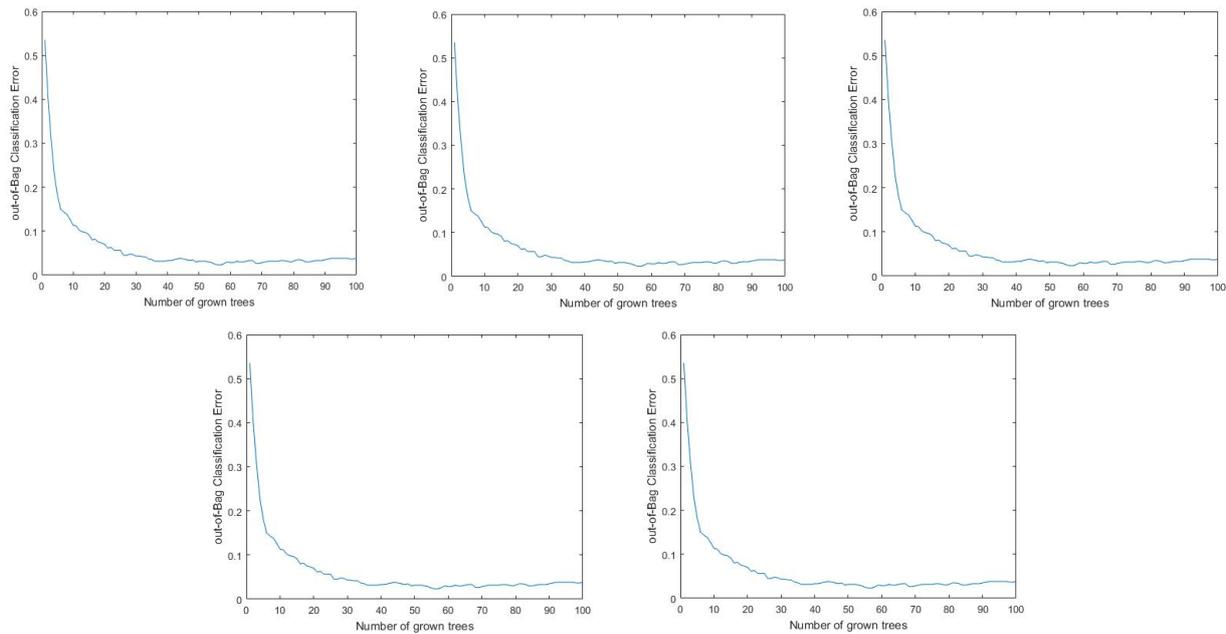
TABEL 6
TINGKAT AKURASI

Test 1	Test 2	Test 3	Test 4	Test 5	Rata-rata
99 %	100 %	100 %	100 %	100 %	99.8 %

Bisa disimpulkan persentase rata-rata dari uji testing untuk pengklasifikasian sebesar 99,8 % dan ini artinya bahwa identifikasi biji-bijian berdasarkan ekstraksi fitur warna, bentuk dan tekstur layak dilakukan.

V. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan didapatkan tingkat akurasi sebesar 99.8 %. Dapat disimpulkan bahwa pengambilan dataset berupa gambar atau image biji-bijian yang diteliti dapat dijadikan patokan untuk pengidentifikasian dan dapat dikembangkan dalam proses selanjutnya.



Gambar 6 Classification Error Testing

VI. DAFTAR PUSTAKA

[1] S. Sinurat *et al.*, “Analisa tekstur citra biji kemiri menggunakan metode filter gabor,” vol. 13, pp. 50–54, 2018.

[2] M. M. Sobel, R. Canny, P. Teguh, K. Putra, N. Kadek, and A. Wirdiani, “Pengolahan Citra Digital Deteksi Tepi Untuk Membandingkan Metode Sobel, Robert dan Canny,” *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 2, no. 2, pp. 253–261, 2016.

[3] E. G. Dianta, “Deteksi Tepi Menggunakan Metode Canny Dengan Matlab Untuk Membedakan Uang Asli dan Uang Palsu,” *J. Jur. Tek. Inform. Fak. Teknol. Ind. Univ. Gunadarma*, vol. 1, no. 1, pp. 1–13, 2012.

[4] I. R. G. A. Sugiarta, M. Sudarma, and I. M. O. Widyantara, “Ekstraksi Fitur Warna, Tekstur dan Bentuk untuk Clustered-Based Retrieval of Images (CLUE),” *Teknol. Elektro*, vol. 16, no. 1, pp. 85–90, 2017.

[5] D. Li, Y. Liu, L. Gao, and T. Management, “Research of Maize Seeds Classification Recognition Based on the Image Processing,” vol. 9, no. 11, pp. 181–190, 2016.

[6] S. I. Syafi’i, R. T. Wahyuningrum, and A. Muntasa, *Segmentasi Obyek Pada Citra Digital Menggunakan Metode Otsu Thresholding*, vol. 13, no. 1, 2016.

[7] M. Otsu, “Optimum Global Thresholding Using Otsu’s Method,” vol. 1, pp. 1–6.

[8] K. Kiliç, I. H. Boyaci, H. Köksel, and I. Küsmenoglu, “A classification system for beans using computer vision system and artificial neural networks,” *J. Food Eng.*, vol. 78, no. 3, pp. 897–904, 2007.

[9] K. Kiratiratanapruk and W. Sinthupinyo, “Color and texture for corn seed classification by machine vision,” *2011 Int. Symp. Intell. Signal Process. Commun. Syst. “The Decad. Intell. Green Signal Process. Commun. ISPACS 2011*, pp. 7–11, 2011.

[10] R. Binary, “Morphological Image Processing,” pp. 1–7, 2017.

[11] B. Pathak, A. Bhuyan, and D. Barooah, “Gray-Level Co-occurrence Matrix Implementation based on Edge Detection Information for Surface Texture Analysis,” pp. 1–5, 2014.

[12] M. Pietikäinen *et al.*, *Computational Imaging and Vision 40*, vol. 40, no. Comput. Vis. Using Local Bin. Patterns. 2011.

[13] D. A. Muhammad, R. R. W. Ken, and E. Siswanto, “Pencarian Citra Digital Berbasis Konten dengan,” *J. Telemat.*, vol. 8, no. 2, pp. 8–13, 1858.

[14] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[15] Y. L. Pavlov, “Random forests,” *Random For.*, pp. 1–122, 2019.

[16] N. Horning, “Random Forests: An algorithm for image classification and generation of continuous fields data sets,” *Int. Conf. Geoinformatics Spat. Infrastruct. Dev. Earth Allied Sci. 2010*, pp. 1–6, 2010.

[17] P. T. T. Hong, T. T. T. Hai, L. T. Lan, V. T. Hoang, V. Hai, and T. T. Nguyen, “Comparative Study on Vision Based Rice Seed Varieties Identification,” *Proc. - 2015 IEEE Int. Conf. Knowl. Syst. Eng. KSE 2015*, pp. 377–382, 2016.

Perencanaan Perjalanan Wisata Multi Kota dan Negara dengan Algoritma Cuttlefish

Hendrawan Armanto, *Teknik Informatika Sekolah Tinggi Teknik Surabaya,*
Reynold Kevin, *Teknik Informatika Sekolah Tinggi Teknik Surabaya,*
dan C. Pickerling, *Teknik Informatika Sekolah Tinggi Teknik Surabaya*

Abstrak — Liburan merupakan saat yang paling tepat untuk melakukan perjalanan wisata bersama keluarga. Sebelum melakukan liburan, tentu saja setiap orang akan menyiapkan rencana perjalanan yang paling efektif dan efisien walaupun merupakan hal yang sulit. Meskipun sudah terdapat berbagai paket tur yang ditawarkan bagi para wisatawan, seringkali yang ditawarkan tidak sesuai dengan keinginan dari masing-masing orang karena faktor selera yang berbeda-beda. Oleh karena itu, dibuatlah sebuah penelitian yang digunakan untuk melakukan perencanaan perjalanan wisata multi kota dan negara berbasis aplikasi mobile. Untuk menyelesaikan, penelitian ini akan menggunakan Algoritma Cuttlefish dalam pembuatan jadwal wisata secara otomatis berdasarkan data input yang telah diisikan oleh pengguna sebelumnya. Pada penelitian ini, pengguna dapat melakukan pembuatan perencanaan perjalanan wisata baik secara manual ataupun otomatis. Selain fitur untuk pembuatan perencanaan perjalanan wisata, juga terdapat fitur untuk melakukan pencarian tiket penerbangan dan juga akomodasi tempat tinggal dengan bantuan TravelPayouts API. Pada pembuatan jadwal wisata pun juga sudah dilengkapi dengan pencarian tiket penerbangan dan akomodasi tempat tinggal sesuai dengan input yang diberikan oleh pengguna sebelumnya. Pengguna juga dapat melihat jadwal wisata yang telah terbentuk dalam bentuk Table View, Map View dan PDF. Dengan dibuatnya penelitian ini diharapkan para pengguna dapat membuat sebuah jadwal wisata berdasarkan selera masing-masing. Jadwal wisata yang dibentuk secara otomatis dengan menggunakan Algoritma Cuttlefish tersebut juga dibuat sedemikian rupa hingga tercipta sebuah jadwal wisata yang efisien. Selain itu, pada tahap akhir dari tahap uji coba disebarkan kuesioner kepada responden. Kuesioner bertujuan untuk mengetahui komentar pengguna mengenai hasil akhir dari penelitian yang telah dibuat demi pengembangan untuk kedepannya. Berdasarkan hasil kuesioner, sebanyak 88% responden menyatakan jadwal wisata yang terbentuk sudah cukup baik.

Kata Kunci — Evolutionary Algorithm, Cuttlefish Algorithm, Perencanaan Perjalanan Wisata.

I. PENDAHULUAN

Seiring dengan berjalannya waktu, saat ini terdapat banyak tempat menarik dan dapat dijadikan sebagai objek wisata. Mulai dari tempat wisata alami seperti pegunungan dan pantai sampai tempat wisata buatan seperti

taman bermain hingga tempat-tempat bersejarah yang ada pada daerah tertentu. Hal ini tentu saja akan mendorong para wisatawan untuk ingin mengunjungi berbagai tempat wisata. Selain itu, dengan adanya banyak tempat wisata yang ada, para wisatawan juga semakin memiliki banyak opsi untuk melakukan perjalanan wisata dan tentu saja akan semakin bingung menentukan sebuah rute perjalanan wisata.

Solusi dari masalah inipun sudah sejak lama dipecahkan dengan berbagai macam pilihan tur yang ditawarkan oleh pihak agensi travel sehingga para wisatawan cukup memilih jenis tur yang ada tanpa perlu repot-repot untuk membuat jadwal wisata. Namun, banyak dari pilihan tur yang telah ditawarkan tidak sesuai dengan keinginan masing-masing orang karena faktor selera tiap orang yang berbeda dan juga budget yang dimiliki. Hal ini tentu saja menimbulkan masalah baru. Untuk mempermudah para wisata dalam mengatur perjalanan wisata khususnya pada 3 negara yaitu Hong Kong, Taiwan dan Korea Selatan, dibentuklah sebuah aplikasi mobile yang digunakan untuk perencanaan perjalanan wisata secara otomatis sesuai dengan keinginan dari pengguna. Selain itu pengguna juga dapat memesan tiket penerbangan dan juga melakukan pencarian terhadap akomodasi tempat tinggal yang diinginkan. Tentunya pembuatan jadwal wisata secara otomatis yang terdapat pada aplikasi tersebut dibentuk dengan bantuan Evolutionary Algorithm [1], yaitu menggunakan Algoritma Cuttlefish [2][3]. Karena sistem ini berbasis pada sebuah aplikasi mobile, sehingga akan dibentuk dengan menggunakan bantuan Vue.js [4], Framework7 [5] yang seluruhnya merupakan bagian dari paket-paket yang terdapat pada Node.js [6].

II. ALGORITMA CUTTLEFISH

Algoritma Cuttlefish merupakan sebuah algoritma yang terinspirasi dari cara binatang sotong untuk bertahan hidup. Algoritma ini akan mensimulasikan mekanisme perubahan warna pada tubuh dari binatang sotong [7]. Algoritma ini ditemukan pada tahun 2013 oleh ilmuwan bernama Adel Sabry Eesa, Adnan Mohsin Abdulazeez Brifciani dan Zeynep Orman [2][3].

Sotong merupakan sebuah binatang laut yang memiliki kemampuan untuk merubah warna kulitnya menjadi sesuai dengan apa yang ada di lingkungan sekitarnya. Sotong akan melakukan mimikri yang merupakan sebuah proses evolusi yang terjadi pada spesies untuk menjadi sama dengan spesies lainnya dengan cara merefleksikan cahaya yang masuk kedalam kulit sotong lalu difreksikan kembali [8].

Hendrawan Armanto, Departemen Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Jawa Timur, Indonesia (e-mail: hendrawan@stts.edu)

Reynold Kevin, Departemen Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Jawa Timur, Indonesia

C. Pickerling, Departemen Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: pickerling@stts.edu).



Gambar. 1. Binatang Sotong

Sotong memiliki 3 jenis komponen kulit yang menumpuk satu sama lain dan nantinya ketiga komponen kulit ini akan saling bekerjasama agar dapat merubah warna kulit sotong menjadi jutaan warna dan motif yang seolah-olah sama persis dengan lingkungan sekitarnya [9]. Berikut adalah 3 jenis komponen kulit yang terdapat pada binatang sotong:

1) *Chromatophores*

Chromatophores merupakan sekumpulan otot dan kantung yang berisikan ribuan pigmen granul. Otot-otot yang terdapat pada chromatophores ini dapat berkontraksi dan relaksi dan menyebabkan kantung dapat membuka atau menutup. Pada saat otot berelaksasi, kantung akan mengecil untuk menyembunyikan pigmen-pigmen yang ada.

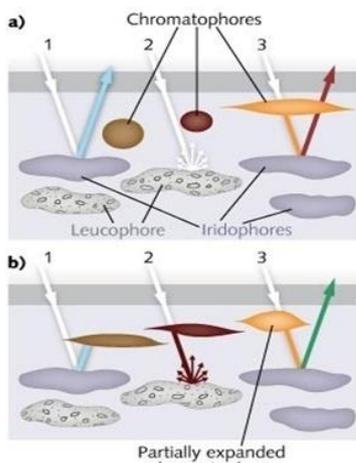
2) *Iridophores*

Iridophores pada sotong akan bekerja untuk menghasilkan warna metalik dan dapat bekerja apabila terkena cahaya. Selain itu, bagian ini juga dapat digunakan binatang sotong untuk membuat bagian kulitnya transparan sehingga seluruh organ dari binatang sotong dapat terlihat.

3) *Leucophores*

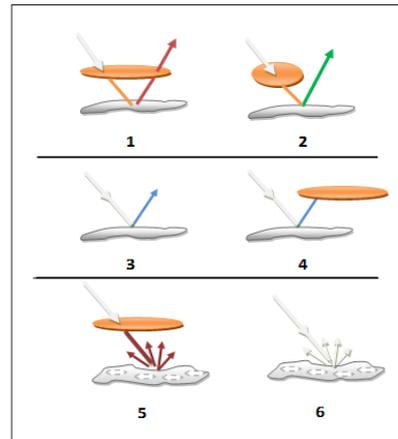
Bagian ini bertanggung jawab untuk menghasilkan sebuah motif bintang-bintang yang terdapat pada binatang sotong. Bagian ini juga merupakan sebuah kumpulan grup-grup sel yang nantinya bisa menyebar dan dapat merefleksikan cahaya yang masuk pada tubuh sotong.

Penampilan atau warna yang terdapat pada binatang sotong ini tergantung pada bagian komponen kulit mana yang terkena cahaya dan nantinya cahaya yang masuk dapat direfleksi baik dari kombinasi antara chromatophores, iridophores, dan leucophores. Proses mekanisme perubah warna pada sotong ini dapat disebut sebagai Global Optimum Solution.



Gambar. 2. Komponen Kulit Sotong

Tujuan dari algoritma ini adalah untuk mendapatkan bentuk dari binatang sotong untuk bersembunyi dengan menyerupai sesuai dengan lingkungannya. Berdasarkan mekanisme perubahan warna kulit yang terjadi pada sotong, Algoritma Cuttlefish akan dibagi menjadi 6 kasus yang mana akan dikumpulkan lagi menjadi 4 grup. Keenam kasus inilah yang nantinya akan mewaliki mekanisme perubahan warna kulit sotong dalam Algoritma Cuttlefish. Untuk setiap kasusnya, akan memiliki 2 proses utama yaitu reflection dan visibility. Reflection merupakan sebuah proses yang menunjukkan mekanisme untuk memantulkan cahaya yang mengenai pada komponen kulit binatang sotong. Visibility adalah suatu proses yang menunjukkan bahwa binatang sotong memiliki bentuk dan motif yang sama seperti daerah sekitarnya. Untuk setiap individu baru nantinya akan berasal dari hasil perhitungan antara reflection dan visibility.



Gambar. 3. 6 Kasus pada Algoritma Cuttlefish

Langkah pertama yang harus dijalankan ada dengan cara melakukan inisialisasi sebuah populasi dengan setiap individu acak didalamnya serta penginputan nilai untuk masing-masing parameter r_1 , r_2 , v_1 dan v_2 . Kemudian dilanjutkan dengan melakukan proses perhitungan untuk mendapatkan sebuah solusi baru dimana populasi akan dibagi sesuai dengan 4 grup sesuai dengan kasus-kasus yang terdapat pada gambar 3. Untuk grup 1 dan 4 bertanggungjawab sebagai local search, sedangkan pada grup 2 dan 3 bertanggungjawab sebagai global search.

$$newp = reflection + visibility \quad (1)$$

Grup 1 (Kasus 1 dan 2) merupakan proses interaksi antara chromatophores dan iridophores. Mula-mula cahaya akan menuju bagian chromatophores dan akan diteruskan menuju komponen iridophores. Semua cahaya tersebut akan dipantulkan kembali. Meningat sel otot pada chromatophores dapat berkontraksi atau berelaksasi sehingga cahaya pantulan pun dapat kembali pada komponen chromatophores atau tidak. Berikut adalah persamaan yang digunakan untuk mendapatkan solusi dari grup 1:

$$reflection_j = R * G_1[i].Points[j] \quad (2)$$

$$visibility_j = V * (Best.Points[j] - G_1[i].Points[j]) \quad (3)$$

Nilai G_n merupakan grup yang sedang dikerjakan lalu $G_n[i].points[j]$ merupakan individu yang ke- i pada grup n

dan menunjuk pada poin ke j . Sedangkan untuk nilai R pada persamaan 3 merupakan sebuah persamaan yang didapatkan dari nilai acak antara parameter r_2 hingga r_1 atau sebaliknya. Hal ini juga berlaku demikian untuk nilai V yaitu merupakan sebuah persamaan yang didapatkan dari nilai acak antara parameter v_2 hingga v_1 atau sebaliknya. Nilai parameter dari r_1 , r_2 , v_1 dan v_2 ini nantinya didapatkan berdasarkan inputan dari pengguna. Untuk $Best.Points[j]$ merupakan sebuah nilai pada individu terbaik yang menunjuk pada poin ke j .

$$R = random() * (r_1 - r_2) + r_2 \quad (4)$$

$$V = random() * (v_1 - v_2) + v_2 \quad (5)$$

Grup 2 (Kasus 3 dan 4) menyimulasikan komponen kulit iridophores menghasilkan solusi baru berdasarkan refleksi cahaya yang terbaik. Karena dalam hal ini komponen iridophores menghasilkan solusi yang terbaik yaitu menghasilkan refleksi cahaya yang warnanya sama dengan organ tubuh pada binatang sotong sehingga dapat diasumsikan bahwa warna yang masuk dan warna yang dihasilkan dari refleksi cahaya adalah sama dan direpresentasikan sebagai best point. Pada grup ini variabel R akan bernilai 1.

$$reflection = R * Best.Point[j] \quad (6)$$

Grup 3 (Kasus 5) merupakan proses komponen kulit leucophores untuk menghasilkan solusi baru berdasarkan refleksi cahaya sekitar. Untuk menemukan kesamaan antara warna yang masuk dengan warna yang dihasilkan dari refleksi cahaya, dapat diasumsikan bahwa warna yang masuk merupakan best solution dan warna yang dihasilkan dari refleksi cahaya adalah nilai yang terdapat di sekitar best solution. Nilai visibility pada grup ini ditunjukkan pada persamaan 7 dimana nilai AV_{Best} yang merupakan rata-rata nilai pada individu terbaik di setiap poinnya.

$$visibility = V * (Best.Point[j] - AV_{Best}) \quad (7)$$

Grup 4 (Kasus 6) menggunakan komponen leucophores untuk menghasilkan solusi yang acak dari cahaya yang masuk. Dengan adanya solusi yang acak ini, diharapkan dapat menghasilkan sebuah solusi baru yang baik sehingga solusi yang digunakan pada kasus ini adalah sebuah solusi yang acak.

Setelah seluruh perhitungan untuk solusi baru selesai dilakukan pada setiap grupnya, akan dilakukan perubahan nilai best apabila solusi baru memiliki nilai fitness yang lebih kecil dari best (dalam hal ini fitness semakin kecil menandakan individu semakin baik). Hal ini akan dilakukan terus menerus hingga individu yang didapatkan sesuai dengan kondisi atau persyaratan yang telah ditetapkan.

III. VUE.JS

Vue.js [4] merupakan suatu framework yang berbasis Javascript dan digunakan oleh para developer untuk membangun sebuah website yang interaktif. Vue.js didesain untuk berfokus pada view layer saja dan juga sangat mudah diimplementasikan dan diintegrasikan dengan library lain

ataupun dengan project yang sudah ada sebelumnya. Karena framework ini berbasis Javascript, sehingga dalam penggunaannya dibutuhkan aplikasi Node.js untuk melakukan proses instalasi cara termudah.

Pada Vue.js, terdapat sebuah fitur yang digunakan pembuatan komponen-komponen yang nantinya dapat digunakan kembali dan juga fitur data binding yang sangat baik. Selain itu, dengan menggunakan Vue.js struktur kodenya pun juga sudah sangat tertata baik dan structural. Fitur menarik lain yang terdapat pada Vue.js yaitu dapat membuat sebuah aplikasi single-page yang memiliki kapabilitas yang tinggi dengan adanya kombinasi dari library dan tool yang sudah disediakan.

Konsep component based yang diusung oleh Vue.js merupakan sebuah konsep dimana setiap blok layout dianggap sebagai komponen yang independen dan memiliki fungsi sendiri yang terisolir sehingga mudah digunakan ulang oleh setiap halaman website apabila diperlukan. Setiap komponen tersebut nantinya akan disusun menjadi blok-blok yang saling terkait dan membangun sebuah tampilan utuh pada sebuah halaman website.

Ketika menggunakan Vue.js, pasti akan selalu terkait dengan data, methods dan juga props karena 3 hal tersebut merupakan hal utama yang terdapat pada Vue.js. Data dalam Vue.js merupakan sekumpulan variabel yang digunakan oleh HTML template dan dapat dimanipulasi serta dimanfaatkan oleh internal komponen masing-masing dan juga tidak dapat diakses oleh komponen lainnya. Jadi sebuah variabel di dalam suatu komponen Vue.js hanya akan berpengaruh terhadap komponen itu sendiri.

Methods merupakan opsi dalam Vue.js yang berubah objek berisi fungsi-fungsi. Fungsi ini sendiri akan memiliki berbagai tujuan yang berbeda-beda dan bervariasi baik memanipulasi data ataupun melakukan sebuah logika pada sebuah aplikasi. Fungsi yang terdapat dalam methods ini juga bisa dijadikan sebuah Event yang bisa dipanggil dengan mudah di bagian HTML template dari komponen yang bersangkutan.

Props merupakan sebuah parameter yang dapat dilempar oleh sebuah komponen kepada komponen lainnya. Props merupakan satu dari beberapa cara untuk mendistribusikan data dari satu komponen ke komponen lainnya.

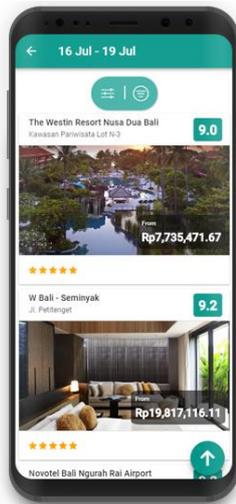
IV. FRAMEWORK7

Framework7 [5] merupakan sebuah framework HTML yang dirilis dibawah lisensi MIT yang bersifat open-source dan terbentuk pada tahun 2014. Framework ini ditujukan untuk pengembangan aplikasi berbasis android maupun iOS yang bersifat hybrid dengan menggunakan HTML, CSS dan Javascript. Framework ini dibentuk dengan tujuan agar para developer dapat membuat sebuah aplikasi mobile tanpa perlu dasar pengetahuan mengenai aplikasi native untuk pembuatan aplikasi mobile itu sendiri.

Meskipun aplikasi berbasis mobile ini dibuat dengan menggunakan HTML, CSS dan Javascript, tampilan dari aplikasi tidak kalah baiknya dengan aplikasi mobile yang dibuat secara native. Jadi dapat dikatakan, Framework7 merupakan sebuah framework website yang sudah menyediakan fitur tampilan responsive untuk aplikasi mobile, sehingga seolah-olah merupakan sebuah aplikasi mobile yang dibuat secara native.

Framework7 ini memang dikhususkan untuk pembuatan aplikasi pada Android maupun iOS saja. Beragam tampilan seperti material design dari google dan komponen-komponen pada iOS juga dapat dipilih agar dapat membuat aplikasi menjadi menarik. Semua itu digunakan dengan menyediakan style-style class pada tag HTML yang sudah dibuat sedemikian rupa menyerupai komponen yang terdapat pada aplikasi mobile serta rangkaian fungsi Javascript yang sudah disediakan oleh Framework7 agar dapat dijalankan layaknya aplikasi mobile.

Framework7 juga sudah didukung dengan bantuan 2 framework Javascript yang dikhususkan untuk pembuatan front-end yaitu React.js dan Vue.js sehingga dapat memperindah tampilan dari desain aplikasi yang telah dibuat. Selain itu sintaks yang terdapat pada Framework7 juga memiliki kemiripan dengan sintaks jQuery. Framework7 juga sudah dilengkapi beragam UI komponen yang mana pengguna hanya perlu melakukan pemanggilan saja untuk menampilkan beragam komponen yang sudah disediakan seperti modal, popup, popover, list view, media list, tab bar, side panel, layout grid, preloader, form, card, button, dll.



Gambar. 4. Contoh Tampilan Framework7

Pada Framework7 juga telah menawarkan fitur yang membuat pengguna dapat merasakan hal yang sama ketika menggunakan aplikasi mobile yang dibuat secara native baik itu dari iOS maupun android seperti swipe back, swipe actions, pull to refresh, infinite scroll, message, dll.

Framework7 sendiri juga sudah menyediakan API yang dapat digunakan untuk pemanggilan fungsi-fungsi seperti menampilkan sebuah popup dan modal dengan menggunakan Javascript dimana dalam penggunaannya memiliki kesamaan dalam menggunakan jQuery. Yang sangat disayangkan dalam penggunaan Framework7 ini adalah dari sisi dokumentasi yang diberikan. Dokumentasi yang diberikan masih belum cukup lengkap karena hanya menjelaskan komponen-komponen serta fitur-fitur yang membuat Framework7 dapat menyerupai aplikasi yang dibuat secara native.

Sedangkan untuk dokumentasi mengenai bagaimana cara mengubah aplikasi dari format HTML dan CSS menjadi format aplikasi mobile baik pada Android maupun iPhone masih belum dimiliki oleh Framework7, sehingga developer harus mempelajari lagi bagaimana penggunaan Apache

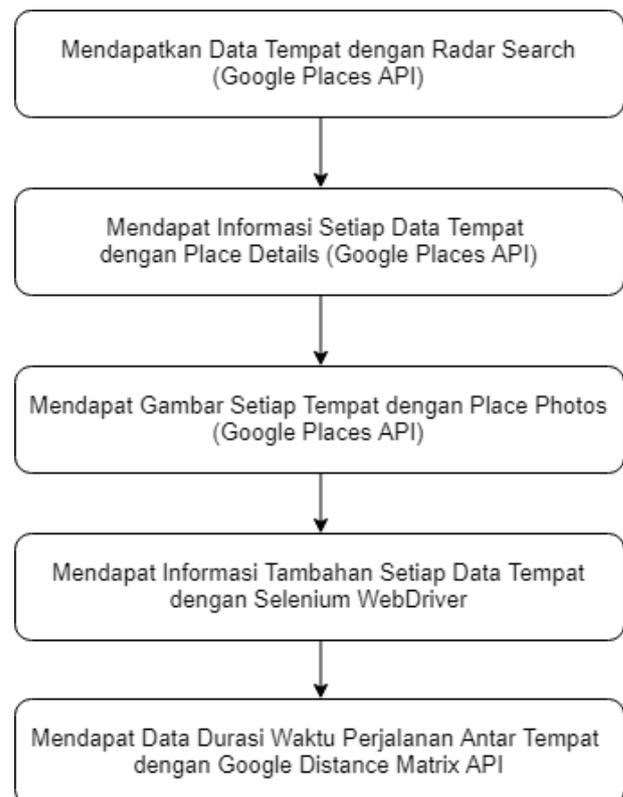
Cordova atau PhoneGap untuk mengubah hasil aplikasi yang berasal dari Framework7 menjadi aplikasi mobile yang ada.

V. DATA

Data tempat yang disediakan pada aplikasi ini merupakan data tempat wisata, tempat makanan serta akomodasi tempat tinggal yang terdapat pada 3 negara yaitu Hong Kong, Taiwan dan Korea Selatan. Pada pengumpulan mengenai informasi data-data yang digunakan pada aplikasi ini akan digunakan dengan bantuan dari Google Places API [10] dan juga Selenium Webdriver [11] sebagai bantuan utama yang nantinya akan dijelaskan pada subbab selanjutnya.

Seluruh data tempat yang didapatkan nantinya juga hanya berdasarkan hasil pencarian dari Google Places API saja tanpa adanya data tambahan yang dicari secara manual. Data-data yang dihasilkan oleh Google Places API nantinya juga akan disortir lagi karena terdapat beberapa data yang informasinya tidak menunjukkan bahwa tempat tersebut bukan merupakan sebuah tempat wisata, tempat makanan ataupun akomodasi tempat tinggal.

Pada Gambar 5 menunjukkan sebuah alur yang digunakan untuk pencarian data-data tempat yang terdapat pada sistem ini. Proses yang pertama kali harus dilakukan dalam pencarian data-data tempat pada aplikasi ini adalah dengan menggunakan fitur yang terdapat pada Google Places API yaitu Radar Search. Pada tahap ini, akan dilakukan proses pencarian data-data sesuai dengan radius yang telah diinputkan serta tipe-tipe tempat kategori yang diinginkan. Untuk setiap tempat akan dilakukan proses pencarian ini sebanyak 4 kali dengan kategori yang berbeda yaitu untuk mendapatkan data tempat wisata sebanyak satu kali, data tempat makanan sebanyak 2 kali dan akomodasi tempat wisata sebanyak satu kali.



Gambar. 5 Alur Pencarian Data Tempat

Setelah mendapatkan data-data tempat berupa place_id yang berasal dari Google Places API, akan dilanjutkan pada proses selanjutnya yaitu pencarian informasi mengenai setiap hasil data tempat yang sudah didapatkan dengan menggunakan fitur Place Details pada Google Places API. Hasil yang didapatkan dengan menggunakan Place Details adalah informasi seperti nama, alamat, jam buka, dll. Kemudian akan dilanjutkan dengan proses ketiga yaitu mendapatkan gambar tempat dengan menggunakan fitur Place Photos yang terdapat pada Google Places API.

Perlu diketahui bahwa proses pencarian data tempat pada proses pertama hingga ketiga hanya berlaku untuk satu pencarian saja dan setelah seluruh data terkumpul baru dilanjutkan pada proses 4 dan 5 yang menggunakan Selenium Webdriver dan Google Distance Matrix API. [12] Setelah seluruh data tempat terkumpul, akan dilanjutkan dengan proses 4 yaitu mencari informasi tambahan seperti deskripsi singkat, rata-rata durasi menetap pada tempat tersebut dengan menggunakan bantuan Selenium WebDriver.

Setelah seluruh data informasi tambahan untuk setiap tempat wisata selesai didapatkan, akan dilakukan proses sortir secara manual terhadap seluruh data tempat yang sudah didapatkan. Setelah pencarian data selesai dilakukan, proses terakhir yang dilakukan adalah mendapatkan data durasi lama perjalanan setiap tempat yang ada dengan menggunakan Google Distance Matrix API. Untuk setiap proses yang terdapat pada gambar 5 akan dilakukan dengan menggunakan bahasa pemrograman C# untuk menghasilkan kumpulan data tempat yang sudah dikonversikan dalam bentuk CSV.

VI. TRAVELPAYOUTS AFFILIATE PROGRAM

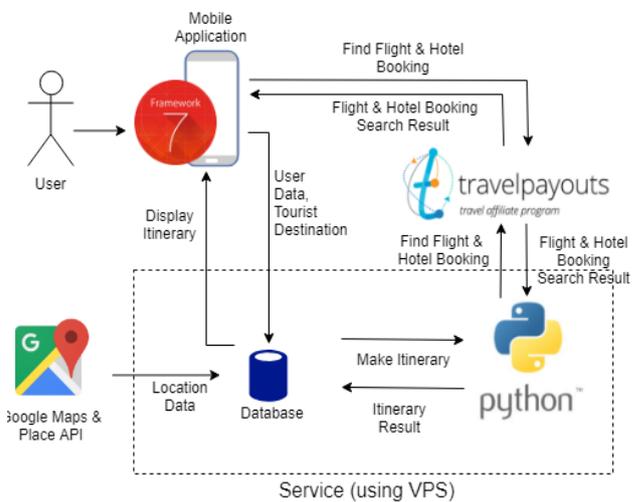
TravelPayouts [13] merupakan sebuah program afiliasi yang bergerak pada bidang travel yang sudah bekerja sama dengan produk-produk seperti JetRadar, Aviasales, Hotellook, dll. Seluruh produk tersebut merupakan sebuah website yang bergerak pada bidang travel namun memiliki fitur utama yang berbeda-beda. JetRadar (Jetradar.com) merupakan salah satu search engine yang terbesar di dunia dan membantu banyak orang dalam membandingkan harga tiket pesawat yang ingin dicari. Aviasales (Aviasales.ru) juga merupakan sebuah search engine yang berguna untuk melakukan pencarian terhadap tiket penerbangan yang berada pada negara Russia dan Ukraina. Sedangkan Hotellook (Hotellook.com) menawarkan fitur untuk melakukan perbandingan harga yang terdapat dalam pencarian akomodasi tempat tinggal yang ada dan tentunya hasil pencarian tersebut berasal dari website travel seperti Booking.com, Agoda.com, Expedia, dll.

Pada program afiliasi TravelPayouts ini telah menyediakan API yang dapat digunakan oleh developer untuk membuat situs pencarian tiket pesawat maupun hotel yang diakses melalui HTTP URL. Jadi dengan menggunakan fitur API ini, developer murni membuat seluruh tampilan dari pencarian tersebut secara manual karena yang hasil yang diberikan akan berupa sebuah JSON yang nantinya akan diubah dalam bentuk tampilan pada sebuah website. Terdapat 5 API yang nantinya dapat digunakan oleh para developer yaitu:

1. API Flight Data Access
2. Affiliate Statistics API

3. Mobile SDK for iOS and Android
4. Flight Search API
5. Hotel Search API

Pada aplikasi ini akan digunakan Flight Search API dan juga Hotel Search API yang digunakan untuk pencarian tiket pesawat serta akomodasi tempat tinggal pada pembuatan jadwal wisata nantinya. Selain itu pada aplikasi ini juga akan menyediakan fitur untuk melakukan pencarian dan pembelian tiket pesawat serta akomodasi tempat tinggal secara independen. Karena aplikasi yang dibuat merupakan sebuah aplikasi mobile yang bersifat hybrid yakni berbasis HTML, sehingga digunakanlah Flight Search API dan Hotel Search API sebagai hasil pencariannya dan untuk desainnya akan dibentuk secara manual seluruhnya. Pada penggunaan akses terhadap Flight Search API dan juga Hotel Search API, akan dibutuhkan request khusus dan dikirimkan pada email aktif yang dimiliki oleh TravelPayouts yang nantinya akan dilakukan verifikasi lebih lanjut apakah request untuk penggunaan akses terhadap kedua API tersebut disetujui oleh pihak TravelPayouts.



Gambar. 6. Alur Sistem

Flight Search API merupakan salah satu API yang disediakan oleh TravelPayouts untuk melakukan hasil pencarian terhadap tiket pesawat dimana dapat diakses melalui HTTP URL dan hasil response yang diberikan dalam bentuk format JSON. Akan terdapat 3 jenis pencarian utama dalam pencarian tiket pesawat nantinya yaitu round trip, one way dan open jaw. Round Trip merupakan sebuah pencarian tiket pesawat dimana akan dilakukan pencarian sekaligus untuk pergi dan pulang. One Way merupakan sebuah pencarian tiket pesawat untuk 1 destinasi saja. Sedangkan Open Jaw merupakan sebuah pencarian tiket pesawat dimana akan dilakukan pencarian lebih dari 1 destinasi yang membentuk suatu rute perjalanan.

Selain adanya fitur untuk melakukan pencarian penerbangan yang terdapat dalam aplikasi ini, juga terdapat sebuah fitur untuk melakukan pencarian terhadap akomodasi tempat tinggal sesuai dengan tempat dan tanggal booking yang dipilih. Pencarian akomodasi tempat tinggal ini juga dapat terwujud dengan bantuan dari TravelPayouts API yaitu Hotel Search API dengan kerjasamanya pada website Hotellook.com. Hotel Search API ini akan memberikan hasil response berupa hasil pencarian terhadap akomodasi tempat tinggal yang berasal dari agensi-agensinya seperti Agoda.com,

Booking.com, Zenhotel.com yang telah bekerja sama dengan Hotellook.

VII. DESAIN SISTEM

Pada sistem ini akan berfokus pada sebuah sistem berbasis aplikasi mobile dimana pengguna dapat membuat sebuah jadwal wisata secara otomatis ataupun manual pada multi kota maupun negara yang terdapat pada 3 negara yaitu Hong Kong, Taiwan dan Korea Selatan. Selain terdapat pembuatan jadwal wisata, sistem ini juga sudah dilengkapi dengan proses pencarian tiket penerbangan serta pencarian akomodasi tempat tinggal dan juga dapat membeli secara langsung. Sistem pencarian baik tiket penerbangan maupun akomodasi tempat tinggal ini juga sudah terintegrasikan dengan sistem pembuatan jadwal wisata serta dapat digunakan secara independen apabila tidak ingin membuat sebuah jadwal wisata yang ada. Selain itu pada sistem ini juga menyediakan sebuah fitur yang digunakan untuk memilih kurs mata uang sehingga para pengguna dapat lebih mudah mengetahui harga yang ada sesuai dengan preferensi yang diinginkan oleh pengguna tersebut.

Pada gambar 6 menunjukkan sebuah alur yang dimiliki oleh sistem dan aplikasi ini akan memiliki nama TripWander. Pada awalnya pengguna dapat membuat sebuah jadwal wisata secara otomatis maupun manual. Pertama-tama pengguna harus mengisi data-data atau informasi yang berkaitan sebelum memulai pembuatan jadwal wisata. Setelah seluruh data telah terisi dan pengguna memutuskan untuk memilih jadwal wisata yang terbuat secara otomatis, maka pengguna harus menunggu hingga jadwal wisata tersebut terbentuk dan akan ditunjukkan melalui notifikasi yang terdapat pada device dari pengguna tersebut.

Jika pengguna memilih untuk membuat jadwal wisatanya sendiri, maka akan dilanjutkan pada halaman selanjutnya untuk mengatur jadwal wisatanya. Hal tersebut juga berlaku pada pengguna yang ingin mengatur ulang jadwal wisata yang telah terbentuk secara otomatis sebelumnya. Selain itu pengguna juga dapat melakukan pencarian terhadap tiket penerbangan serta akomodasi tempat tinggal sesuai dengan tanggal yang diinginkan pada ini yang menggunakan sebuah program afiliasi yang dimiliki oleh TravelPayouts.

Sebelum aplikasi ini terbentuk, seluruh data mengenai tempat-tempat wisata serta akomodasi tempat tinggal yang ada juga sudah didapatkan terlebih dahulu dengan menggunakan bantuan Google Maps API. Pada aplikasi ini juga menggunakan MySQL sebagai database dimana menyimpan seluruh data mengenai informasi pengguna, jadwal wisata, data tempat wisata, data nilai kurs mata uang. Untuk pembuatan aplikasi mobile yang terdapat pada sistem ini menggunakan Framework7 dan Vue.js yang mana merupakan sebuah framework berbasis HTML, CSS dan JavaScript dalam pembuatan aplikasi mobile.

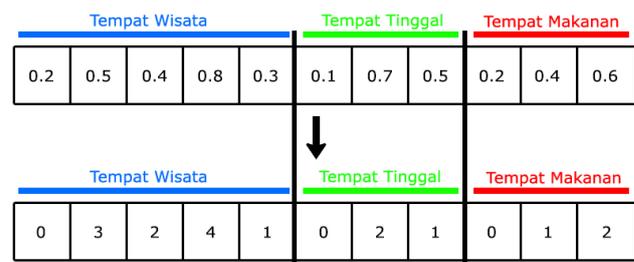
Sedangkan pada backend, akan digunakan Flask [14] yang merupakan sebuah microframework berbasis Python dan digunakan untuk pengaksesan database serta pembuatan sebuah service yang nantinya akan dipanggil oleh aplikasi mobile yang sudah terbuat agar dapat mengakses database yang ada. Selain itu untuk pembuatan jadwal wisata secara otomatis dengan menggunakan Algoritma Cuttlefish juga menggunakan bahasa pemrograman Python. Karena seluruh penggunaan backend

serta pembuatan jadwal wisata yang berbasis Python, maka digunakan sebuah VPS (Virtual Private Server) yang memiliki operating system berbasis Linux dengan versi 14.04. Untuk versi Python yang digunakan adalah Python versi 3.5.4.

VIII. PEMBUATAN JADWAL WISATA OTOMATIS

Sebelum penjelasan tentang bagaimana cara menghitung nilai fitness pada pembuatan jadwal wisata, akan dijelaskan terlebih dahulu mengenai representasi individu yang terdapat pada aplikasi ini. Hal pertama yang dilakukan dalam pembuatan representasi individu adalah dengan membagi masing-masing data tempat wisata, akomodasi tempat tinggal dan tempat makanan yang sudah disortir sesuai dengan budget yang dipilih oleh pengguna ke dalam variabel. Semua data tersebut juga sudah disortir berdasarkan kota yang sedang dipilih.

Kemudian jumlah dimensi yang terdapat pada individu merupakan perpaduan jumlah dari data tempat wisata, akomodasi tempat tinggal dan tempat makanan yang sudah diperoleh dari database. Lalu pada saat proses inialisasi individu baru, akan dilakukan dengan cara mengisi masing-masing dimensi dengan nilai acak antara 0 hingga 1.



Tempat Wisata					Tempat Tinggal			Tempat Makanan		
0.2	0.5	0.4	0.8	0.3	0.1	0.7	0.5	0.2	0.4	0.6
					↓					
Tempat Wisata					Tempat Tinggal			Tempat Makanan		
0	3	2	4	1	0	2	1	0	1	2

Gambar. 7 Ilustrasi Representasi Individu

Setelah seluruh dimensi pada individu baru tersebut telah terisi dengan nilai acak antara 0 hingga 1, nilai yang terdapat pada masing-masing dimensi tersebut akan diubah menjadi indeks yang mana indeks 0 merupakan nilai yang paling kecil dan juga proses pengubahan nilai indeks tersebut akan dibagi menjadi 3 bagian yaitu masing-masing untuk tempat wisata, akomodasi tempat tinggal dan tempat makanan.

Pada gambar 7 merupakan sebuah contoh ilustrasi dari representasi individu yang terdapat dalam pembuatan jadwal wisata. Sebagai contoh terdapat 5 tempat wisata, 3 akomodasi tempat tinggal dan 3 tempat makanan. Lalu jumlah dimensinya akan menjadi 11 pada individu tersebut dan pada masing-masing dimensi akan diisi nilai acak antara 0 hingga 1 dan perlu diketahui pada ilustrasi ini nilai acak memiliki keakuratan 1 angka di belakang koma saja, sedangkan pada implementasinya nilai acak memiliki keakuratan 8 angka di belakang koma.

Setelah masing-masing dimensi telah terisi oleh nilai acak, kemudian akan diubah menjadi indeks sesuai dengan bagiannya dimana indeks terkecil menandakan nilai terendah. Hal ini dilakukan dengan tujuan agar dapat mengetahui tempat mana yang nantinya akan diambil untuk dijadikan jadwal wisata dengan cara pengambilan data sesuai dengan indeks yang terdapat pada dimensi-dimensi tersebut.

Penggunaan Algoritma Cuttlefish pada aplikasi ini hanya sebatas pada 1 kota saja dan nantinya akan dihubungkan pada kota dan negara selanjutnya. Hal ini bertujuan agar

hasil yang didapatkan lebih optimal mengingat banyaknya tempat-tempat yang akan dipilih dan juga penulis masih belum menemukan sebuah metode untuk memicu Algoritma Cuttlefish ini untuk berpindah ke kota atau negara selanjutnya. Meskipun penggunaannya hanya sebatas 1 kota saja, tetapi semuanya sudah terintegrasi agar dapat menghubungkan ke kota dan negara-negara selanjutnya yang ada.

Pada pembuatan jadwal wisata dengan Algoritma Cuttlefish ini juga terdapat beberapa batasan yang digolongkan menjadi soft constraints dan hard constraints. Batasan-batasan soft constraints ini antara lain jam makan dan waktu aktivitas setiap harinya. Sedangkan batasan yang termasuk hard constraints adalah jumlah hari, penerbangan, perpindahan antar kota dan negara, pencarian akomodasi tempat tinggal serta seluruh tempat wisata yang terdapat pada jadwal harus dalam keadaan buka.

Batasan pertama yang dibahas adalah jumlah hari menetap nantinya pada jadwal wisata yang akan dibuat. Pada negara Hong Kong, terdapat 1 kota yang dapat dikunjungi yaitu Hong Kong itu sendiri. Untuk Taiwan, terdapat 5 kota yang dapat dikunjungi yaitu Taipei, Tainan, Taichung, Hsinchu dan Kaohsiung. Sedangkan untuk negara Korea Selatan, terdapat 3 kota yang dapat dikunjungi yaitu Seoul, Busan dan Gyeongju. Untuk setiap negara akan memiliki jumlah minimum hari menetap selama 3 hari dan jumlah hari maksimal yang berbeda-beda yaitu Hong Kong 5 hari, Taiwan 12 hari dan Korea Selatan selama 8 hari.

Penerbangan dalam hal ini berlaku untuk awal keberangkatan dan juga untuk setiap kali berpindah negara. Nantinya tiket pesawat yang akan dipilih merupakan tiket pesawat yang memiliki jam sampai di antara pukul 05.00 hingga 13.00 apabila memungkinkan. Lalu para wisatawan nantinya akan diberikan asumsi waktu selama 60 menit yang digunakan untuk pengambilan barang hingga siap untuk berangkat menuju tempat akomodasi. Apabila wisatawan sampai pada negara tujuan melebihi dari waktu 18.00, maka secara otomatis jadwal wisata akan dilanjutkan keesokan harinya. . Lalu apabila kota yang dikunjungi tidak memiliki bandara, maka akan menuju ke bandara terdekat pada kota yang tersedia. Sebagai contoh, apabila wisatawan hendak berpindah negara dari kota Hsinchu, Taiwan maka wisatawan harus menuju ke bandara yang terdapat pada kota Taipei, Taiwan.

Batasan ketiga merupakan batasan yang terjadi apabila wisatawan akan berpindah kota sehingga batasan ini bersifat opsional. Setiap kali wisatawan akan berpindah kota, maka hal ini selalu terjadi pada pagi hari dimana wisatawan pertama kali menjalankan aktivitas setiap harinya. Untuk transportasi yang ada, akan menggunakan transportasi umum jalur darat sesuai dengan petunjuk yang terdapat pada Google Maps API. Lalu setelah berpindah kota, hal pertama yang dilakukan adalah menuju ke tempat tinggal pada kota yang baru saja dituju untuk melakukan proses check-in yang akan diberi waktu untuk check-in selama 60 menit setiap kali pertama kali mengunjungi tempat akomodasi.

Batasan pindah negara ini hampir sama dengan batasan mengenai pindah kota yaitu bersifat opsional apabila wisatawan mengunjungi lebih dari 2 negara. Wisatawan juga diberikan opsi apabila tidak ingin kembali ke negara asalnya. Pada batasan ini, wisatawan harus sudah sampai pada bandara 90 menit sebelum jam boarding seperti yang tertera pada tiket pesawat. Apabila jam boarding pada tiket

pesawat adalah subuh atau sebelum jam aktivitas dimulai, maka wisatawan juga akan langsung menuju ke bandara 90 menit sebelumnya.

Batasan selanjutnya adalah budget dan berfokus pada pengaruh pembuatan jadwal wisata. Budget yang dimaksudkan ini hanyalah budget yang digunakan dalam pemilihan kualitas akomodasi tempat tinggal yang ada saja karena dalam pengumpulan datanya tidak ditemukan adanya data mengenai biaya yang dikeluarkan untuk melakukan perjalanan dengan transportasi umum dan juga biaya rata-rata yang dikeluarkan untuk masuk pada sebuah tempat wisata. Untuk budget, akan terdapat 2 pilihan yaitu low dan high budget. Low budget merupakan hotel-hotel yang memiliki 3 bintang atau di bawahnya, sedangkan high budget merupakan hotel yang memiliki bintang 4 ke atas.

Pada batasan pencarian akomodasi tempat tinggal sesuai dengan kota yang dituju. Pencarian akomodasi tempat tinggal ini akan didasari pada 2 hal yaitu jarak antara bandara awal atau akhir menuju akomodasi tempat tinggal dan budget. Untuk jarak antara bandara awal atau akhir dengan akomodasi tempat tinggal diusahakan tidak terlalu jauh untuk menghemat waktu perjalanan dan tidak lupa juga dengan batasan-batasan lain yang ada yang nantinya terdapat pada perhitungan fitness.

Pada batasan pemilihan tempat wisata yang nantinya akan dibuat menjadi jadwal perjalanan wisata. Tempat wisata ini dibagi menjadi 4 kategori yaitu must see, culture, nature dan recreation yang digunakan untuk wisatawan dalam pembuatan jadwal sesuai dengan ketertarikannya antara 4 kategori tersebut. Must see merupakan kategori tempat yang dapat dikatakan terkenal pada kotanya dan dalam hal ini tempat wisata tersebut dikategorikan sebagai must see dari jumlah rating yang cukup tinggi yang terdapat pada google. Culture merupakan kategori tempat wisata seperti museum, tempat ibadah, dsb. Nature merupakan kategorit tempat wisata yang berhubungan dengan alam seperti air terjun, pantai, pegunungan, dll. Recreation merupakan kategori tempat wisata seperti taman, taman bermain dan juga mall.

Sebelum pembuatan jadwal wisata, para wisatawan akan memberikan nilai terhadap ketertarikannya terhadap 4 kategori yang sudah disediakan. Nilai yang ada akan memiliki rentang -2 hingga 2. Semakin rendah nilai yang diberikan terhadap kategori yang ada maka wisatawan tidak tertarik terhadap kategori yang sedang diberi penilaian dan hal ini juga berlaku kebalikannya yaitu semakin tinggi nilai yang diberikan maka wisatawan tertarik terhadap kategori tersebut. Pada jadwal wisata ini juga dipastikan setiap tempat wisata yang ada harus dalam keadaan buka. Pada pembuatan jadwal wisata ini juga setiap tempat wisata hanya dapat dikunjungi maksimal satu kali saja.

Untuk batasan jam makan, terdapat sebuah asumsi bahwa pada jadwal wisata yang telah terbuat para wisatawan sudah menerima sarapan dari akomodasi tempat tinggal setiap pagi hari. Lalu para wisatawan harus makan 2 kali sehari baik itu makan pada tempat makan atau pada tempat wisata yang terdapat tempat makan di dalamnya. Apabila wisatawan tidak makan sebanyak 2 kali sehari atau makan disaat jam makan telah selesai, maka akan dikenakan penalty pada perhitungan fitness. Untuk jam makan siang akan memiliki rentang antara pukul 12.00 hingga 14.00 sedangkan untuk jam makan malam akan memiliki rentang antara pukul 18.00 hingga 20.00.

TABEL I
CONTOH PERHITUNGAN TOTAL INTEREST

No	Must See		Culture		Nature		Recreation		Total Interest (Nilai Asli)	Keterangan
	B	A	B	A	B	A	B	A		
1	2	6	2	12	2	12	2	12	0(40)	Nilai <i>interest</i> sama
2	2	6	1	6	0	0	-1	-3	9(9)	-
3	-1	-3	-2	-6	-1	-3	-2	-6	0(-18)	Seluruh nilai <i>interest</i> < 0
4	1	3	2	12	2	12	0	0	24(27)	Melebihi batas maksimal

Batasan yang terakhir merupakan jam aktivitas yang akan dilakukan setiap harinya oleh wisatawan. Sebelumnya, wisatawan harus memilih jam aktivitas setiap harinya yaitu untuk pagi hari memiliki rentang antara pukul 07.00 hingga 09.50 pagi. Sedangkan untuk jam malamnya memiliki rentang antara pukul 20.00 hingga 22.50. Nantinya pada hasil jadwal yang telah terbuat akan diusahakan semaksimal mungkin agar para wisatawan menyelesaikan aktivitas per harinya tepat waktu dan akan dikenakan penalty apabila pulang terlalu cepat atau terlalu larut malam.

Perhitungan rumus fitness yang digunakan dalam pembuatan jadwal wisata dengan Algoritma Cuttlefish adalah menggunakan sistem diskon dan penalti. Perhitungan nilai fitness akan menggunakan total waktu secara keseluruhan dari lama perjalanan antar tempat dan juga durasi menetap pada setiap masing-masing tempat dalam satuan waktu detik. Semakin kecil hasil fitness, maka hasil jadwal juga akan semakin baik. Selain total waktu keseluruhan, akan terdapat penalti apabila melanggar batasan-batasan yang sudah dijelaskan sebelumnya. Perlu juga diingat bahwa seluruh perjalanan wisata ini menggunakan transportasi umum.

Setelah seluruh jadwal wisata terbentuk yang tentunya juga sesuai dengan representasi individu dan batasan yang ada, maka akan dilakukan proses untuk perhitungan nilai fitness total. Nilai fitness total akan didapatkan dari jumlah waktu yang berasal dari durasi lama perjalanan dan durasi waktu menetap pada seluruh tempat yang terdapat pada jadwal yang terbentuk. Kemudian juga akan ditambahkan dengan total nilai penalti yang didapatkan serta akan mendapatkan diskon yang berasal dari jumlah tempat yang dikunjungi serta kecocokan antara tempat wisata yang terpilih dengan nilai interest yang telah diisikan oleh pengguna.

$$fitness = place_time + penalty - (place_time * (place_disc + interest_disc)) \quad (8)$$

Pada rumus 8 merupakan sebuah rumus yang digunakan untuk menghitung nilai fitness secara keseluruhan. *place_time* dan *penalty* merupakan sebuah variabel yang berisikan jumlah waktu yang dilalui dalam detik dan jumlah penalti yang didapat karena melanggar batasan-batasan yang terdapat pada pembuatan jadwal otomatis ini. Kemudian setelah kedua variabel tersebut dijumlahkan akan dilakukan pengurangan dengan nilai dari variabel *place_time* yang dikalikan dengan jumlah presentase antara pengaruh jumlah tempat yang dikunjungi dan nilai interest pada masing-masing tempat wisata.

$$place_disc = \frac{1}{place_target_per_day * real_stay_days * 2} * total_place \quad (9)$$

Pada rumus 9 merupakan rumus yang digunakan untuk mendapatkan presentase jumlah tempat pada jadwal yang telah terbentuk. Variabel *place_target_per_day* merupakan target jumlah tempat wisata dan tempat makanan yang diharapkan dalam 1 hari yaitu bernilai sebesar 10. Kemudian seperti yang sudah dijelaskan, variabel *real_stay_days* merupakan jumlah total hari yang terdapat pada jadwal wisata yang terbentuk (akan dikurangkan 1 hari apabila pada saat datang melebihi pukul 18.00). *total_place* merupakan variabel yang berisikan jumlah total tempat wisata dan tempat makanan yang dikunjungi hingga jadwal selesai terbentuk. Pada rumus tersebut, terdapat sebuah pengali yaitu sebesar 2. Hal ini bertujuan agar nilai yang dihasilkan berkisar antara 0 hingga 0.5.

Sebelum melakukan perhitungan terhadap presentase nilai interest, akan terdapat beberapa aturan mengenai masing-masing nilai interest yang berkisar antara -2 sampai 2:

1. Akan terdapat sebuah nilai *interest_multiplier* bernilai 3 yang nantinya digunakan untuk mengalikan setiap nilai interest pada masing-masing interest. Hal ini dilakukan agar pengaruh nilai interest lebih kuat dengan adanya *interest_multiplier*.
2. Akan terdapat sebuah nilai *like_multiplier* bernilai 2 yang merupakan sebuah multiplier apabila nilai interest bernilai positif. Jadi, setelah dilakukan pengalihan terhadap *interest_multiplier* akan dilakukan pengalihan selanjutnya terhadap *like_multiplier* khusus untuk interest yang bernilai positif kecuali untuk kategori interest *must_see*. Hal ini dilakukan agar nilai interest positif akan lebih dominan daripada nilai yang bernilai negatif, sedangkan untuk kategori *must_see* merupakan sebuah kategori yang bukan berdasarkan selera dari pengguna sehingga tidak diperlukan adanya pengalihan terhadap *like_multiplier*.
3. Apabila nilai dari masing-masing interest adalah sama atau seluruh nilai interest kurang dari 0, maka akan dianggap ketertarikan pengguna terhadap masing-masing interest adalah netral sehingga nilai dari masing-masing interest akan diubah menjadi 0. Nilai total interest pada satu tempat wisatanya nantinya akan dibatasi dengan nilai Sesuai dengan rumus 10.

$$total_interest = 2 * 4 * interest_multiplier \quad (10)$$

Pada rumus 10, *total_interest* akan bernilai 24 jika nilai *interest_multiplier* bernilai 3. Hal ini dilakukan untuk

membatasi nilai maksimal dari total interest pada setiap wisata dan memberikan pengaruh nilai yang lebih pada nilai interest yang bernilai positif.

Berdasarkan aturan-aturan yang baru saja dijelaskan, aturan-aturan tersebut dibuat dengan tujuan agar nilai interest yang diisi oleh pengguna memiliki pengaruh yang lebih dominan terutama nilai yang berisikan positif dan juga agar dapat memperbaiki nilai-nilai interest yang telah diisi oleh pengguna apabila memiliki kesamaan nilai pada seluruh interest ataupun seluruh nilai interest yang berisikan kurang dari 0 yang mana sudah terdapat contoh perhitungan total interest pada tabel I.

Setelah mengetahui beberapa aturan yang terdapat pada perhitungan nilai total interest untuk setiap tempat wisata, terdapat sebuah rumus yang digunakan untuk mendapatkan presentase pengaruh tempat wisata terdapat nilai interest. Nantinya presentase ini dapat menjadi sebuah diskon maupun penalti. Dalam hal ini, penalti yang dimaksudkan jika nilai total fitness yang terdapat pada sebuah tempat wisata bernilai negatif dan sebaliknya jika bernilai positif maka akan menjadi sebuah diskon.

$$interest_disc = \frac{sum_list_interest - total_interest_{min}}{total_interest_{max} - total_interest_{min}} - 0.5 \quad (11)$$

Pada rumus 11 menunjukkan sebuah rumus yang digunakan untuk mendapatkan presentase pengaruh kecocokan antara tempat wisata dengan nilai interest yang telah ditentukan oleh pengguna dengan rentang nilai antara -0.5 hingga 0.5. Hal ini dilakukan apabila nilai presentase maksimal untuk diskon terhadap pengaruh jumlah tempat yang bernilai sebesar 0.5 dan juga nilai presentase maksimal untuk nilai interest yang bernilai 0.5, jika digabungkan nilai presentase maksimalnya akan menjadi 1 sehingga menyebabkan kedua faktor tersebut memiliki bobot pengaruh yang sama.

IX. UJI COBA

Ujicoba yang dilakukan antara lain adalah uji coba parameter yang digunakan untuk mengetahui parameter-parameter yang terbaik untuk digunakan dalam pembuatan jadwal wisata. Kemudian juga dilakukan uji coba skenario yang digunakan untuk mengetahui sebuah hasil jadwal wisata yang dibentuk secara otomatis beserta dengan deskripsi tentang fitness yang ada. Pertama, Uji coba yang dilakukan adalah uji coba parameter dengan menggunakan seluruh data input wisata yang sama yaitu berupa 1 kota saja.

TABEL II
PENGARUH JUMLAH ITERASI TERHADAP PERBUATAN JADWAL WISATA DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
1000	200	0.55	-0.55	1	-1	1128817.20
5000	200	0.55	-0.55	1	-1	723584.87
10000	200	0.55	-0.55	1	-1	612637.52
25000	200	0.55	-0.55	1	-1	286305.10
50000	200	0.55	-0.55	1	-1	252804.40
75000	200	0.55	-0.55	1	-1	252804.40

Kota yang digunakan adalah Kota Taipei dan perjalanan wisata dimulai selama 3 hari yaitu dari tanggal 18 Mei 2018 hingga 20 Mei 2018. Pada uji coba ini juga diasumsikan

akan melakukan perjalanan pada 1 negara saja lalu kembali ke kota asal, sehingga untuk jam kedatangan yang ada adalah pukul 10.15 dan jam keberangkatan untuk pulang ke negara asal adalah pukul 21.15. Selain itu pada uji coba ini juga tidak digunakan interest atau dapat dikatakan seluruh nilai interest yang ada bernilai 0.

Pada waktu untuk memulai serta mengakhiri aktivitas adalah mulai dari pukul 08.00 hingga 21.30 setiap harinya. Pada uji coba ini, parameter yang akan diuji coba adalah jumlah iterasi, jumlah populasi, nilai parameter *r1*, *r2*, *v1* dan *v2*. Untuk uji coba ini pun juga sudah menggunakan random seed yang berasal dari library pada Python yang bertujuan untuk menghasilkan nilai acak yang sama meskipun dilakukan pada waktu yang berbeda.

Berdasarkan hasil uji coba untuk masing-masing parameter yang terdapat pada Algoritma Cuttlefish untuk pembuatan jadwal otomatis dan telah tercantum mulai dari tabel II hingga tabel VII, berhasil didapatkan nilai parameter terbaik yang digunakan pada pembuatan jadwal wisata dengan Algoritma Cuttlefish adalah jumlah iterasi sebesar 50000, jumlah populasi sebesar 300, nilai parameter *v1* dan *v2* sebesar 1 dan -1 serta nilai parameter *r1* dan *r2* yang bernilai sebesar 0.5 dan -0.5.

TABEL III
PENGARUH JUMLAH POPULASI TERHADAP PERBUATAN JADWAL WISATA DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	50	0.55	-0.55	1	-1	296317.30
50000	100	0.55	-0.55	1	-1	206155.92
50000	150	0.55	-0.55	1	-1	529473.50
50000	200	0.55	-0.55	1	-1	252804.40
50000	250	0.55	-0.55	1	-1	114071.50
50000	300	0.55	-0.55	1	-1	111680.63
50000	350	0.55	-0.55	1	-1	115408.88
50000	400	0.55	-0.55	1	-1	126658.43

TABEL IV
PENGARUH PARAMETER V TERHADAP PERBUATAN JADWAL WISATA DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	300	0.55	-0.55	0.25	-0.25	150273.00
50000	300	0.55	-0.55	0.5	-0.5	122142.37
50000	300	0.55	-0.55	0.75	-0.75	135548.20
50000	300	0.55	-0.55	1	-1	111680.63
50000	300	0.55	-0.55	1.25	-1.25	123374.60
50000	300	0.55	-0.55	1.5	-1.5	127177.60
50000	300	0.55	-0.55	2	-2	133491.00

TABEL V
PENGARUH PARAMETER R TERHADAP PERBUATAN JADWAL WISATA DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	300	0.25	-0.25	1	-1	145588.77
50000	300	0.5	-0.5	1	-1	109656.00
50000	300	0.75	-0.75	1	-1	114336.42
50000	300	1	-1	1	-1	133681.22
50000	300	1.25	-1.25	1	-1	119116.15
50000	300	1.5	-1.5	1	-1	114336.42
50000	300	2	-2	1	-1	114336.42

TABEL VI
PENGARUH PARAMETER R1 TERHADAP PERBUATAN JADWAL WISATA
DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	300	0.25	-0.5	1	-1	115747.12
50000	300	0.5	-0.5	1	-1	109656.00
50000	300	0.75	-0.5	1	-1	117234.52
50000	300	1	-0.5	1	-1	114336.42
50000	300	1.25	-0.5	1	-1	144618.8
50000	300	1.5	-0.5	1	-1	117804
50000	300	1.75	-0.5	1	-1	114336.42
50000	300	2	-0.5	1	-1	119116.15

TABEL VII
PENGARUH PARAMETER V1 TERHADAP PERBUATAN JADWAL WISATA
DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	300	0.5	-0.5	0.25	-1	126823.35
50000	300	0.5	-0.5	0.5	-1	127916.23
50000	300	0.5	-0.5	0.75	-1	136507.20
50000	300	0.5	-0.5	1	-1	109656.00
50000	300	0.5	-0.5	1.25	-1	112376.60
50000	300	0.5	-0.5	1.5	-1	158921.87
50000	300	0.5	-0.5	1.75	-1	132779.33
50000	300	0.5	-0.5	2	-1	128078.90

Uji coba berikutnya adalah uji coba skenario. Pada uji coba ini akan dijabarkan secara lengkap mulai dari akomodasi tempat tinggal yang dipilih dan juga perhitungan nilai fitness beserta dengan data input awal sesuai dengan yang terdapat pada tabel VIII. Setelah hasil jadwal terbentuk yang berasal dari data input yang terdapat pada tabel VIII, telah didapatkan nilai fitness sebesar 73271.84 dan terdapat sebuah hasil jadwal wisata yang dapat dilihat pada tabel IX

TABEL IX
JADWAL WISATA UJI COBA SKENARIO

No	Nama	Jenis	Waktu	Interest
Hari Pertama – 13 September 2018				
1	Hong Kong International Airport	Bandara	13.50 – 14.50	-
2	Mandarin Oriental, Hong Kong	Hotel	15.40 – 16.40	-
3	Keung Kee Meat Shop	Tempat Makan	17.00 – 17.45	-
4	Central-Mid-Levels Escalators	Tempat Wisata	18.07 – 19.07	must_see
5	Bubba Gump	Tempat Makan	19.49 - 21.19	-
6	Mandarin Oriental, Hong Kong	Hotel	22.01 – 22.01	-
Hari Kedua – 14 September 2018				
7	Mandarin Oriental, Hong Kong	Hotel	08.00 – 08.00	-
8	Lung Fu Shan Country Park	Tempat Wisata	08.18 – 09.03	recreation
9	Dr Sun Yat-sen Museum	Tempat Wisata	09.18 – 10.48	culture
10	Airport Core Programme Exhibition Centre	Tempat Wisata	11.56 – 12.26	recreation
11	Main Street Corner Café	Tempat Makan	13.42 – 15.42	-
12	Peak Tower	Tempat Wisata	17.12 – 18.42	must_see, recreation
13	Yardbird	Tempat Makan	19.29 – 21.29	-
14	Mandarin Oriental, Hong Kong	Hotel	21.43 – 21.43	-
Hari Ketiga – 15 September 2018				
15	Mandarin Oriental, Hong Kong	Hotel	08.00 – 08.00	-
16	Hong Kong Avenue of Comic Stars	Tempat Wisata	08.16 – 09.46	recreation
17	Noon Day Gun	Tempat Wisata	10.11 – 12.11	culture
18	ABC Kitchen	Tempat Makan	12.41 – 13.26	-
19	Soho, Hong Kong	Tempat Wisata	13.45 – 15.15	must_see, recreation
20	Mandarin Oriental, Hong Kong	Hotel	15.28 – 16.28	-
21	Hong Kong International Airport	Bandara	17.19 – 17.19	-

dengan mayoritas tempat wisatanya memiliki interest recreation dan must_see.

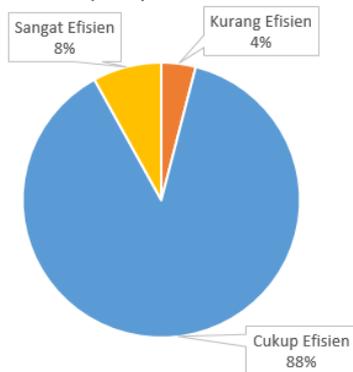
Dari 8 tempat wisata yang akan dikunjungi pada jadwal wisata untuk uji coba 2 ini, dapat dilihat terdapat 2 tempat wisata yang memiliki interest culture, 1 tempat wisata yang memiliki interest must_see, 3 tempat wisata yang memiliki interest recreation dan 2 tempat wisata yang memiliki interest must_see, recreation. Sehingga dapat disimpulkan terdapat 3 dari 8 data atau 37.5% data yang memiliki tipe interest must_see, 2 dari 8 data atau 25% data yang memiliki tipe interest culture, 5 dari 8 data atau 62.5% data yang memiliki tipe interest recreation dan tidak terdapat 1 pun tempat wisata yang memiliki tipe interest nature.

TABEL VIII
DATA INPUT UJI COBA SKENARIO

Tujuan	Hong Kong (3 Hari) – Hong Kong
Jumlah Wisatawan	Dewasa : 2
Tanggal Keberangkatan	12 September 2018
Kota Asal	Surabaya, Indonesia (Kembali)
Interest	Must See : 1 Culture : 0 Nature : -2 Recreation : 2
Budget	High
Waktu Aktivitas	08.00 – 21.30

Selain uji coba skenario, juga telah dilakukan uji coba berupa kuesioner yang dilakukan pada 25 responden. Sebanyak 88% responden mengatakan bahwa hasil jadwal wisata secara otomatis yang terbentuk sudah cukup efisien dan sesuai dengan keinginan pengguna dan terdapat 8% yang merasa bahwa hasil jadwal wisata yang terbentuk

sudah sangat efisien. Sedangkan terdapat 4% responden yang berpendapat bahwa hasil jadwal wisata yang terbentuk masih belum cukup efisien dan untuk visualisasi data tersebut sudah terdapat pada gambar 8. Dari tanggapan responden tersebut, dapat disimpulkan bahwa hasil jadwal wisata yang terbentuk secara otomatis sudah cukup efisien, namun masih harus ditingkatkan kembali dari sisi perhitungan fitness dan dari data tempat yang disediakan.



Gambar. 8 Hasil Uji Coba Kepuasan Pengguna Terhadap Hasil Jadwal

X. KESIMPULAN

Dari hasil uji coba yang dilakukan, Algoritma Cuttlefish mampu menghasilkan sebuah jadwal perjalanan wisata yang sudah diintegrasikan dengan pencarian tiket penerbangan dan akomodasi tempat tinggal dengan bantuan TravelPayouts API.

Berdasarkan hasil uji coba terhadap pengaruh jumlah iterasi, jumlah populasi, nilai parameter $v1$, $v2$, $r1$ dan $r2$, maka pada penerapan Algoritma Cuttlefish dalam pembuatan jadwal wisata menggunakan nilai parameter 50000 pada jumlah iterasi, 300 pada jumlah populasi, 1 dan -1 pada $v1$ dan $v2$, 0.5 dan -0.5 pada $r1$ dan $r2$.

Database mengenai tempat baik tempat wisata, tempat makanan maupun akomodasi tempat tinggal serta data mengenai durasi waktu perjalanan sangat berpengaruh dalam proses perencanaan perjalanan wisata ini, semakin akurat dan baik data yang berhasil dikumpulkan maka semakin baik pula hasil jadwal yang berhasil terbentuk mengingat seluruh data yang berhasil dikumpulkan berasal dari data yang terdapat pada Google Maps saja.

Berdasarkan hasil uji coba kuesioner yang diberikan pada responden, sebanyak 88% responden menyatakan jadwal wisata yang terbentuk sudah cukup baik serta 8% responden menyatakan bahwa jadwal wisata yang dibentuk sudah sangat baik dan 4% responden sisanya menyatakan bahwa jadwal wisata yang ada masih kurang baik.

DAFTAR PUSTAKA

[1] D. Simon, *Evolutionary Optimization Algorithms, Biologically Inspired and Population-Based Approaches to Computer Intelligence*, J. Wiley, and Sons, Inc, 2013.

[2] A. S. Eesa, A. M. Abdulazeez, Z. Orman, Cuttlefish Algorithm – A Novel Bio-Inspired Optimization Algorithm, *International Journal of Scientific and Engineering Research*, Vol. 4, Issue 9, September, 2013.

[3] A. S. Eesa, A. M. Abdulazeez, Z. Orman, A New Tool for Global Optimization Problems- Cuttlefish Algorithm, *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering* Vol. 8, No. 9, 2014.

[4] O. Filipova, *Learning Vue.js 2*, Packt Publishing Ltd, 2016, pp. 53-65.

[5] <https://framework7.io/>.

[6] Eduonix Learning Solutions, *Learn Node.js by Building 6 Projects: Build six practical and instructive Node.js projects*, Packt Publishing Ltd, 2018.

[7] <http://www.thecephalopodpage.org/>.

[8] R. T. Hanlon, and J. B. Messenger, *Cephalopod Behavior*, Cambridge: Cambridge University Press, 1996.

[9] R. A. Cloney, and S. L. Brocco, Chromatophore organs, reflector cells, iridocytes and leucophores in cephalopods, *Am. Zool.* 1983.

[10] https://developers.google.com/places/web-service/intro?hl=en_US.

[11] Z. Zhan, *Selenium WebDriver Recipes in C#: Second Edition*, Apress, 2015.

[12] <https://developers.google.com/maps/documentation/distance-matrix/intro#DistanceMatrixRequests>

[13] <https://www.travelpayouts.com/>

[14] G. C. Hillar, *Building RESTful Python Web Services*, Packt Publishing Ltd, 2016, pp.189-301.

Hendrawan Armanto lahir di Surabaya, Indonesia, pada tahun 1986. Menyelesaikan studi S1 di program studi Teknik Informatika STTS pada tahun 2008. Menyelesaikan studi masternya pada program studi Teknologi Informasi STTS. Minat penelitian adalah bidang Artificial Intelligent, Evolutionary Algorithm, dan Game Development.

Reynold Kevin lahir di Kota Surabaya, Jawa Timur, Indonesia pada tahun 1996. Saat ini sebagai mahasiswa tingkat akhir dengan jurusan S1 Teknik Informatika di STTS Surabaya. Risetnya berfokus pada Evolutionary Algorithm dan pembuatan aplikasi mobile hybrid.

C. Pickerling lahir di Surabaya, Indonesia, pada tahun 1986. Menyelesaikan studi S1 di program studi Teknik Informatika STTS pada tahun 2008. Menyelesaikan studi masternya pada program studi Teknologi Informasi STTS pada tahun 2016. Minat penelitian adalah pada bidang ilmu software engineering dan evolutionary algorithm.

INSYST

Journal of Intelligent System and Computation

Volume 01 Nomor 02 Desember 2019

Author Guidelines

- Manuscript should be written in Indonesia and be submitted online via journal website. Online Submission will be charged at no Cost
- Manuscript should not exceed 15 pages including embedded figures and tables, without any appendix, and the file should be in Microsoft Office (.doc/.docx). [download template](#)
- Title, Abstract and Keywords must be written in bilingual
- Title should be less than 15 words
- Abstracts consists of no more than 200 words, contains the essence of the article and includes a brief background, objectives, methods and results or findings of the study. Abstract is written in one paragraph.
- Keywords are written in Indonesia and English three to five words/phrases, separated with coma and consist of important words/phrases from the article.
- Author's name, affiliation, affiliation address and email. State clearly and include country's name on your affiliation address.
- The main text of the writing should be consists of: Introduction, Method, Result and Discussion, and Conclusion; followed by Acknowledgment and Reference
- Introduction State adequate background, issues and objectives, avoiding a detailed literature survey or a summary of the results. Explain how you addressed the problem and clearly state the aims of your study.
- Used method is the scientific in the form of study of literature, observation, surveys, interviews, Focus Group Discussion, system testing or simulation and other techniques commonly used in the world of research. It is also recommended to describe analysis techniques used briefly and clearly, so that the reader can easily understand.
- Results should be clear, concise and not in the form of raw data. Discussion should explore the significance of the results of the work, not repeat them. Avoid extensive citations and discussion of published literature. INSYST will do the final formatting of your paper.
- Conclusion should lead the reader to important matter of the paper. Authors are allowed to include suggestion or recommendation in this section. Write conclusion, suggestion and/or recommendation in narrative form (avoid of using bulleting and numbering)
- Acknowledgments. It is highly recommended to acknowledge a person and/or organizations helping author(s) in many ways. Sponsor and financial support acknowledgments should be included in this section. Should you have lots of parties

to be acknowledged, state your acknowledgments only in one paragraph. Avoid of using bulleting and numbering in this section

- The number of references are not less than 10 with at least 8 primary references. Primary references are include journal, thesis, disertasion and all kinds of research reports. All refferences must come from source published in last 7 years.
- Figure and table should be in black and white, and if it is made in color, it should be readable when it is later printed in black and white.
- Figure and table should be clearly readable and in a proportional measure to the overall page.

Tim Redaksi

Journal of Intelligent System and Computation

Departement of Informatics

Institut Sains dan Teknologi Terpadu Surabaya

Jl. Ngagel Jaya Tengah 73-77 Surabaya

Email: insyst@istts.ac.id

Website: <https://jurnal.stts.edu/index.php/INSYST/index>