



TEKNOLOGI dan rekayasa

- APLIKASI SISTEM PAKAR BERBASIS ANDROID UNTUK DIAGNOSIS PENYAKIT GASTROESOPHAGEAL REFLUX DISEASE (GERD) DENGAN METODE CERTAINTY FACTOR 78
Marla Nur Assyifa
- IMPLEMENTASI CHATBOT PADA PENDAFTARAN MAHASISWA BARU MENGGUNAKAN RECURRENT NEURAL NETWORK 91
Tjut Awaliyah Zuraiyah, Dian Kartika Utami, Degi Herlambang
- KLASIFIKASI CITRA KEBAKARAN DAN NON KEBAKARAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK 102
Zaky Farhan Abror
- IMPLEMENTASI RENCANA PERJALANAN WISATA DI KOTA BOGOR MENGGUNAKAN ALGORITMA GREEDY BERBASIS WEBSITE 114
Nur Najmi Sania, Illmiyati Sari Tiya Noviyanti
- PERBANDINGAN DETEKSI TEPI SOBEL DAN ROBERT UNTUK PENDETEKSIAN KESAMAAN CITRA BERDASARKAN WARNA 131
M. Ridwan Dwi Septian, Febriana, A. Ramadona Nilawati
- PENGARUH JUMLAH SUDU PADA TURBIN ANGIN SUMBU VERTIKAL TERHADAP DISTRIBUSI KECEPATAN DAN TEKANAN 141
Ridwan, Abdul Latief
- IMPLEMENTASI ALGORITMA ASSOCIATION RULE UNTUK PROMOSI PRODUK BERBASIS WEBSITE PADA BENGKEL DELTA JAYA MOTOR 152
Anggara Fajri Afif, Ericks Rahmat Swedia, Margi Cahyanti

DEWAN REDAKSI JURNAL TEKNIK REKAYASA

Penanggung Jawab

Prof. Dr. E.S. Margianti, S.E., M.M.
Prof. Suryadi Harmanto, SSI., M.M.S.I.
Drs. Agus Sumin, M.M.S.I.

Dewan Editor

Dr. Desti Riminarsih, S.Si, M.Si, Universitas Gunadarma
Dr. Dina Indarti, S.Si, M.Si, Universitas Gunadarma
Dr. Ir. Asep Mohamad Noor, M.T., Universitas Gunadarma
Ajib Setyo Arifin, S.T., M.T., Ph.D., Universitas Indonesia
Dr. Sunny Arief Sudiro, STMIK Jakarta STI&K

Mitra Bebestari

Prof. Dr. Sarifuddin Madenda, Universitas Gunadarma
Prof. Dr.-Ing. Adang Suhendra, S.Si, Skom, Msc, Universitas Gunadarma
Prof. Ir. Busono Soerowirdjo, Msc, Phd, Universitas Gunadarma
Prof. Dr. Rer.Nat. A. Benny Mutiara, Universitas Gunadarma
Prof. Dr. Ir. Bambang Suryawan, MT, Universitas Gunadarma
Prof. Dr. B.E.F. da Silva, Universitas Indonesia
Prihandoko, S.Kom, MIT, Phd, Universitas Gunadarma
Dr. Tubagus Maulana Kusuma, Skom., Mengsc., Universitas Gunadarma
Dr. Ir. Rakhma Oktavina, MT., Universitas Gunadarma
Dr. RR Sri Poernomo Sari, M.T., Universitas Gunadarma
Dr. Lussiana ETP, Ssi., M.T., STMIK Jakarta STI&K

Sekretariat Redaksi

Universitas Gunadarma
Jalan Margonda Raya No. 100 Depok 16424
Phone : (021) 78881112 ext 516.

JURNAL ILMIAH TEKNOLOGI DAN REKAYASA

NOMOR 2, VOLUME 24, AGUSTUS 2019

DAFTAR ISI

APLIKASI SISTEM PAKAR BERBASIS ANDROID UNTUK DIAGNOSIS PENYAKIT <i>GASTROESOPHAGEAL REFLUX DISEASE (GERD)</i> DENGAN METODE <i>CERTAINTY FACTOR</i> Marla Nur Assyifa	78
IMPLEMENTASI <i>CHATBOT</i> PADA PENDAFTARAN MAHASISWA BARU MENGGUNAKAN <i>RECURRENT NEURAL NETWORK</i> Tjut Awaliyah Zuraiyah, Dian Kartika Utami, Degi Herlamban	91
KLASIFIKASI CITRA KEBAKARAN DAN NON KEBAKARAN MENGGUNAKAN <i>CONVOLUTIONAL NEURAL NETWORK</i> Zaky Farhan Abror	102
IMPLEMENTASI RENCANA PERJALANAN WISATA DI KOTA BOGOR MENGGUNAKAN ALGORITMA GREEDY BERBASIS <i>WEBSITE</i> Nur Najmi Sania, Ilmiyati Sari Tiya Noviyanti	114
PERBANDINGAN DETEKSI TEPI SOBEL DAN ROBERT UNTUK PENDETEKSIAN KESAMAAN CITRA BERDASARKAN WARNA M. Ridwan Dwi Septian, Febriani, A. Ramadona Nilawati	131
PENGARUH JUMLAH SUDU PADA TURBIN ANGIN SUMBU VERTIKAL TERHADAP DISTRIBUSI KECEPATAN DAN TEKANAN Ridwan, Abdul Latief	141
IMPLEMENTASI ALGORITMA <i>ASSOCIATION RULE</i> UNTUK PROMOSI PRODUK BERBASIS <i>WEBSITE</i> PADA BENGKEL DELTA JAYA MOTOR Anggara Fajri Afif, Ericks Rahmat Swedia, Margi Cahyanti	152

APLIKASI SISTEM PAKAR BERBASIS ANDROID UNTUK DIAGNOSIS PENYAKIT GASTROESOPHAGEAL REFLUX DISEASE (GERD) DENGAN METODE CERTAINTY FACTOR

Marla Nur Assyifa

*Fakultas Teknologi Industri Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
marlasyifa26@gmail.com*

Abstrak

Masyarakat pada zaman sekarang masih memiliki kesadaran yang rendah terhadap kebiasaan kecil yang merusak kesehatan. Contoh dari kebiasaan kecil tersebut adalah waktu makan yang tidak teratur atau menunda waktu makan. Kebiasaan menunda waktu makan membuat lambung memproduksi asam berlebih. Jika lambung memproduksi asam berlebih maka dapat memicu penyakit maag dan pada tingkat yang lebih tinggi dapat memicu munculnya salah satu penyakit pencernaan yaitu Gastroesophageal Reflux Disease (GERD) yang akan membahayakan organ tubuh lainnya. Banyak masyarakat yang menganggap penyakit tersebut adalah hal kecil yang tidak perlu penanganan dokter. Aplikasi sistem pakar berbasis Android dirancang dan dibuat untuk mendiagnosis penyakit GERD dengan menampilkan besarnya kepercayaan gejala terhadap penyakit tersebut. Besarnya nilai kepercayaan merupakan hasil perhitungan dengan menggunakan metode Certainty Factor. Pengujian sistem menunjukkan bahwa sistem mampu melakukan diagnosis penyakit GERD berdasarkan gejala-gejala yang diderita pasien. Hasil diagnosis berupa persentase yang terbagi menjadi dua kategori yaitu akut dan kronis. Hasil persentase diperoleh dari perhitungan metode Certainty Factor yang menunjukkan tingkat kebenaran dan keakuratan dari kemungkinan penyakit GERD. Berdasarkan pengujian kepada 30 penderita GERD diperoleh persentase keakuratan sistem pakar sebesar 80%. Keakuratan sistem pakar diukur berdasarkan perbandingan antara hasil diagnosis pakar kesehatan dengan hasil diagnosis aplikasi sistem pakar.

Kata Kunci: *Android, aplikasi, GERD, metode Certainty Factor, sistem pakar.*

Abstract

Today's society still has a low awareness of small habits that damage health. Examples of these small habits are irregular meal times or delayed meal times. The habit of delaying mealtime makes the stomach produce excess acid. If the stomach produces excess acid, it can trigger heartburn and at a higher level can trigger the emergence of one digestive disease, namely Gastroesophageal Reflux Disease (GERD) which will harm other organs of the body. Many people think that the disease is a small thing that does not need to be treated by a doctor. The Android-based expert system application is designed and made to diagnose GERD by displaying the amount of symptom confidence in the disease. The amount of the trust value is the result of calculations using the Certainty Factor method. System testing shows that the system is able to make a diagnosis of GERD based on the symptoms suffered by the patient. The diagnosis results in the form of a percentage which is divided into two categories, namely acute and chronic. The percentage results obtained from the calculation of the Certainty Factor method which shows the level of truth and accuracy of the possibility of GERD disease. Based on testing of 30 GERD sufferers, the percentage of expert system accuracy is 80%. The accuracy of the expert system is measured based on the comparison between the results of the diagnosis of the health expert and the diagnosis results of the expert system application.

Keywords: *Android, Certainty Factor method, expert system, GERD*

PENDAHULUAN

Pada zaman sekarang gaya hidup masyarakat yang selalu ingin praktis dan mengarah tidak sehat mengakibatkan tubuh rentan terhadap penyakit. Penyakit tercipta dari kebiasaan kecil seperti menunda waktu makan hingga tidak teraturnya pola makan. Seringnya menunda waktu makan membuat lambung memproduksi asam yang berlebih sehingga memicu timbulnya penyakit *Gastroesophageal Reflux Disease* (GERD). Penyakit GERD dianggap ringan oleh masyarakat, namun jika dibiarkan maka penyakit ini menjadi berbahaya karena dapat mengiritasi lapisan esofagus dan membuatnya menjadi meradang. Pendeteksian secara dini mengenai penyakit GERD pada lambung perlu dilakukan untuk mencegah penyakit yang lebih lanjut dan agar dapat diketahui sudah sejauh mana penyakit yang diderita oleh masyarakat. Seiring waktu, peradangan dapat merusak lapisan esofagus yang menyebabkan komplikasi seperti pendarahan, penyempitan esofagus, dan dapat menyerang organ-organ tubuh lainnya [1].

Perkembangan teknologi memungkinkan untuk membantu masyarakat terutama dalam pendeteksian secara dini mengenai penyakit GERD melalui sistem pakar. Sistem pakar merupakan suatu program kecerdasan buatan yang bertujuan untuk meniru keahlian yang dimiliki oleh pakar dalam menyelesaikan suatu masalah pada berbagai bidang. Perkembangan sistem pakar pada era ini menjadi lebih

mudah karena didukung oleh pesatnya perkembangan teknologi. Sistem pakar dapat diimplementasikan ke dalam beberapa platform aplikasi yaitu Android, Desktop, dan *Website*. Android merupakan salah satu sistem operasi *mobile* yang saat ini lebih diminati penggunaannya oleh masyarakat karena beragam fitur dapat ditampilkan untuk memenuhi kebutuhan. Fitur yang ditawarkan berupa aplikasi untuk menunjang pemakai melakukan suatu kegiatan dengan mudah dan cepat salah satunya yaitu aplikasi sistem pakar. Sistem pakar berbasis Android ini memanfaatkan media *mobile* yang kini lebih marak digunakan dibandingkan dengan media lainnya.

Implementasi sistem pakar pada bidang kesehatan banyak digunakan karena sistem pakar dipandang sebagai sarana untuk menyimpan pengetahuan pakar pada program komputer. Sistem pakar menghasilkan keputusan yang lebih cepat dan konsisten. Implementasi yang diterapkan dalam bidang kesehatan yaitu sistem pakar untuk mendiagnosis suatu penyakit. Sistem pakar dapat membantu aktivitas para pakar sebagai asisten yang berpengalaman dan mempunyai pengetahuan yang dibutuhkan. Pada penyusunan sistem pakar dilakukan kombinasi kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu [2].

Penelitian terdahulu terkait sistem pakar sudah cukup berkembang terutama yang berkaitan dengan bidang kesehatan.

Metode yang digunakan untuk sistem pakar juga semakin beragam. Penelitian terdahulu mengenai sistem pakar yaitu sistem pakar berbasis Android untuk mendiagnosis penyakit TB Paru pada anak menggunakan metode logika fuzzy [3]. Penelitian lain misalnya aplikasi sistem pakar berbasis *mobile* untuk diagnosis meningitis secara detil menggunakan metode *forward chaining* [4]. Sistem pakar berbasis web juga dikembangkan untuk mendiagnosa penyakit lambung dengan implementasi metode CBR (*Case Based Reasoning*) [5]. Metode yang digunakan dalam sistem pakar selain logika *fuzzy*, *forward chaining*, CBR adalah metode *Certainty Factor* [6]. Penelitian mengenai sistem pakar menggunakan metode *Certainty Factor* untuk menentukan titik akupunktur dalam menangani penyakit insomnia dilakukan oleh Gunawan dan Wardoyo. Sistem pakar yang dibuat pada penelitian tersebut memiliki akurasi sebesar 93.33% [7]. Pada penelitian

yang dilakukan Hasan, Sholeha, Tetik, dan Kusri menggunakan metode *Certainty Factor* dalam pembuatan sistem pakar untuk mendiagnosa penyakit kolesterol dan asam urat berbasis web. Hasil penelitian tersebut menunjukkan bahwa sistem pakar yang dibuat memiliki akurasi 80% dalam mendiagnosa penyakit kolesterol dan asam urat [8].

Berdasarkan uraian tersebut maka tujuan penelitian ini adalah mengembangkan suatu aplikasi sistem pakar yang dapat digunakan untuk mendiagnosis penyakit GERD akut maupun kronis. Metode yang digunakan pada penelitian ini adalah metode *Certainty Factor*.

METODE PENELITIAN

Tahapan penelitian dimulai dengan perencanaan, analisis kebutuhan, perancangan struktur navigasi, perancangan tampilan aplikasi, dan pengujian aplikasi yang dijelaskan dalam Gambar 1.



Gambar 1. Tahapan Pembuatan Aplikasi

Pada tahap perencanaan dilakukan pengumpulan data-data serta informasi-

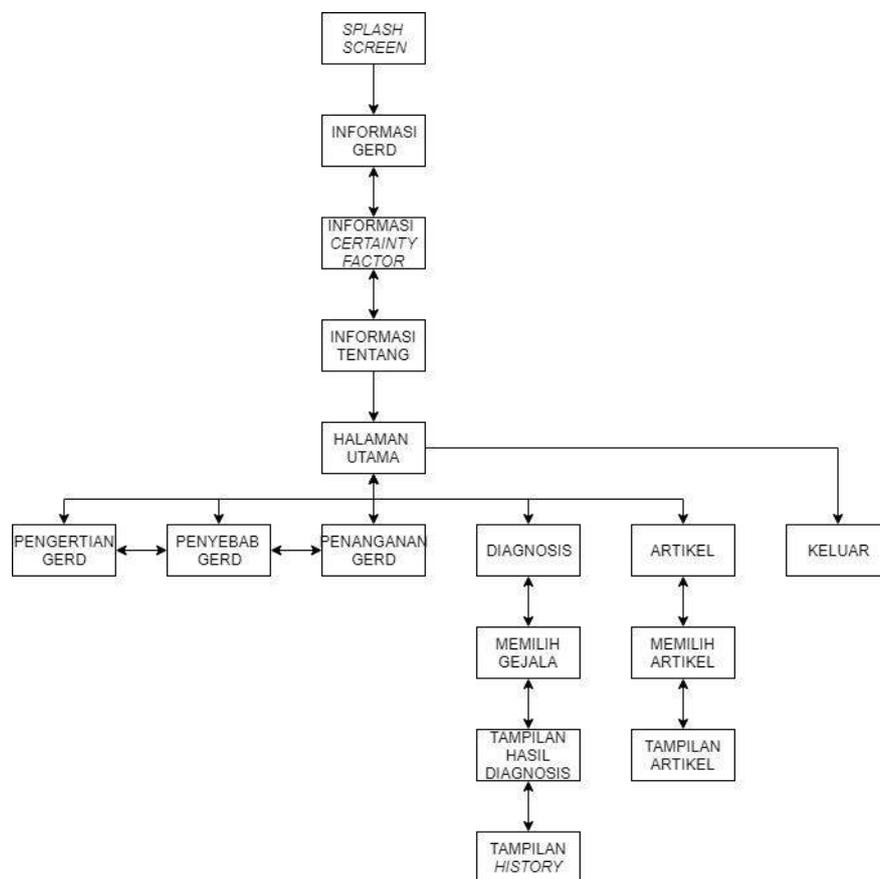
informasi yang berhubungan dengan penyakit *Gastroesophageal Reflux Disease* (GERD).

Pengumpulan data dilakukan dengan dua cara yaitu wawancara dan studi pustaka. Wawancara dilakukan dengan dr. Elma Cita Maghfira untuk mendapatkan nilai *Certainty Factor* atau nilai kepercayaan ahli dari masing-masing gejala untuk perhitungan nilai diagnosis sehingga menghasilkan kategori penyakit seperti akut atau kronis yang diderita oleh pengguna. Analisis kebutuhan. Pada tahap ini dilakukan analisis kebutuhan pengguna, spesifikasi *software* dan *hardware* yang diperlukan untuk pembuatan aplikasi. Kebutuhan pengguna yaitu untuk membangun aplikasi sistem pakar berbasis Android.

Perancangan struktur navigasi digunakan sebagai penuntun alur sebuah

aplikasi multimedia, sedangkan diagram UML digunakan untuk memvisualisasikan interaksi antara pengguna dengan sistem. Tahap perancangan tampilan aplikasi dilakukan untuk memudahkan dalam pembuatan aplikasi agar aplikasi yang dihasilkan menarik dan mudah digunakan. Pengujian aplikasi dilakukan menggunakan metode pengujian *black box*, pengujian aplikasi pada beberapa perangkat *mobile* Android, dan pengujian *alpha* terhadap 30 orang pasien penderita penyakit GERD.

Struktur navigasi yang digunakan adalah struktur navigasi campuran atau gabungan dari struktur navigasi linier dan hierarki.



Gambar 2. Struktur Navigasi

Pada Gambar 2 diberikan struktur navigasi aplikasi. Halaman pertama yang ditampilkan pada aplikasi adalah *splash screen* kemudian halaman selanjutnya berisi informasi GERD, informasi *Certainty Factor*, dan informasi tentang aplikasi. Pada halaman utama terdapat 6 cabang menu seperti pengertian GERD, penyebab GERD, penanganan GERD, diagnosis, artikel, dan keluar. Halaman diagnosis mempunyai cabang seperti memilih gejala yang berisi daftar gejala, tampilan hasil diagnosis yang berisi hasil perhitungan menggunakan metode *Certainty Factor*, dan tampilan *history* yang berisi basis data pengguna yang telah melakukan diagnosis pada aplikasi sistem pakar ini. Halaman artikel mempunyai cabang interaksi yang dapat dilakukan yaitu memilih artikel dan menampilkan isi artikel. Simbol anak panah 2 arah seperti pada halaman utama dan penanganan GERD menunjukkan bahwa dari halaman penanganan bisa kembali ke halaman utama dan begitu juga sebaliknya.

Metode *Certainty Factor* (CF) merupakan bagian dari *Certainty Theory* yang diperkenalkan pertama kali oleh Shorliffe Buchanan dalam pembuatan MYCIN mencatat bahwa dokter sering kali menganalisa informasi yang ada dengan ungkapan seperti mungkin, kemungkinan besar, dan hampir pasti. MYCIN merupakan aplikasi sistem pakar awal yang dirancang untuk mengidentifikasi bakteri yang menyebabkan infeksi berat. Tim MYCIN menggunakan metode CF guna menggambarkan tingkat kepercayaan pakar terhadap masalah yang sedang dihadapi. *Certainty Factor* juga berguna untuk mengatasi ketidakpastian dalam menentukan penyakit yang mempunyai gejala yang sama. Metode *Certainty Factor* cocok digunakan dalam sistem pakar untuk mengukur sesuatu apakah pasti atau tidak pasti dalam mendiagnosis penyakit. Perhitungan dengan menggunakan metode *Certainty Factor* dalam sekali hitung hanya dapat mengolah 2 data saja sehingga keakuratan data dapat terjaga [5].

Tabel 1. Aturan-aturan Nilai Kepercayaan

Kepercayaan	Nilai CF
Tidak pasti	-1.0 sampai 0.79
Hampir Tidak Pasti	-0.8 sampai -0.59
Kemungkinan Tidak	-0.6 sampai -0.69
Mungkin Tidak	-0.4 sampai 0.19
Tidak Tahu	-0.2 sampai 0.2
Mungkin	0.4 sampai 0.59
Kemungkinan Besar	0.6 sampai 0.79
Hampir Pasti	0.8 sampai 0.89
Pasti	0.9 sampai 1.0

Perhitungan yang diperlukan sebagai gejala dari CF sesuai dengan yang telah

dipilih oleh pengguna. Gambar 3 merupakan alur perhitungan menggunakan algoritma.



Gambar 3. Algoritma Perhitungan

Pada Gambar 3 diberikan algoritma perhitungan nilai CF untuk mendapatkan nilai diagnosis penyakit yang diderita pengguna. Perhitungan selanjutnya dilakukan sebagai berikut. Pada sesi konsultasi sistem, pengguna diberi dua pilihan jawaban yang masing-masing memiliki bobot 0 untuk jawaban Tidak dan 1 untuk jawaban Ya. Penyakit dengan kode P01 untuk GERD akut dan P02 kode untuk GERD kronis.

Hasil diagnosis yang ditampilkan pada aplikasi sistem pakar ini berupa nilai

persentase beserta hasil kategori penyakitnya yaitu GERD akut atau GERD kronis. Wawancara yang dilakukan dengan seorang pakar memperoleh hasil berupa gejala-gejala penyakit GERD dengan nilai kepercayaan terhadap masing-masing gejala. Pakar tersebut memberikan sebuah nilai kepercayaan untuk setiap gejala berdasarkan aturan-aturan nilai kepercayaan pada metode *Certainty Factor*. Rentang nilai kepercayaan yang dapat diberikan dimulai dari -1 atau tidak pasti hingga 1 atau pasti.

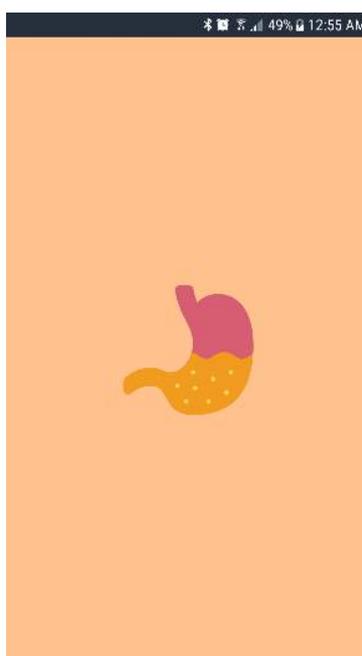
Tabel 2. Gejala Penyakit GERD dan Nilai CF Pakar

Kode Gejala	Nama Gejala	Nilai CF	GERD kronis	GERD akut
G01	Mual	0.7	X	
G02	Muntah	0.8	X	
G03	Suara serak	0.3	X	
G04	Terasa sakit ketika menelan	0.8	X	
G05	Otot perut terasa kejang	0.6	X	
G06	Rasa panas di bagian dada	0.9		X
G07	Rasa asam di mulut	0.8		X
G08	Rasa terbakar di ulu hati	1		X
G09	Rasa terbakar di tenggorokan	1		X
G10	Penurunan berat badan	0.5		X

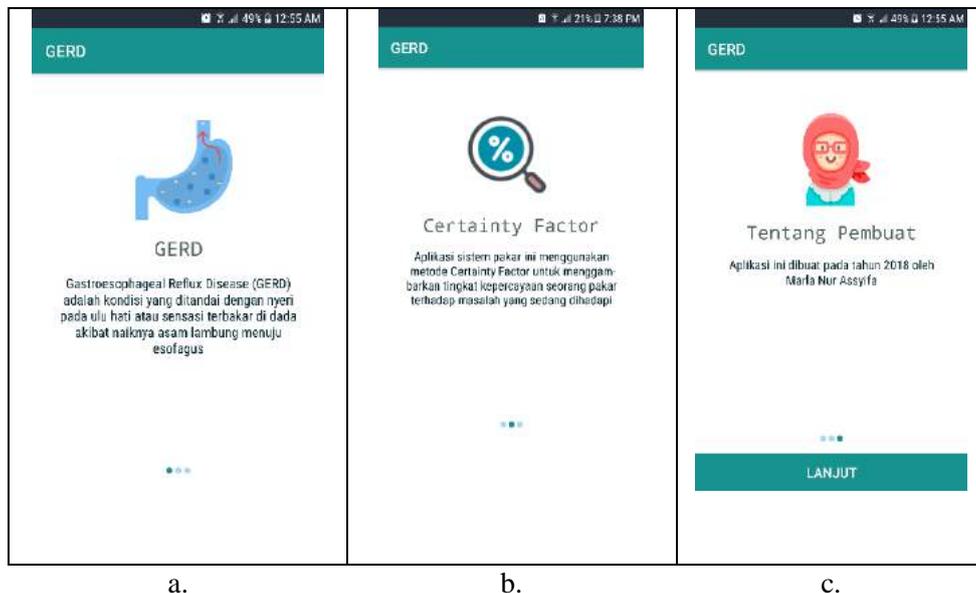
Pada Tabel 2 berisi daftar gejala penyakit GERD beserta kodenya dan nilai *Certainty Factor* (CF) yang diberikan oleh pakar. Kode dan gejala tersebut dimasukkan ke dalam rumus perhitungan metode *Certainty Factor*. Jika pengguna memilih gejala nomor 1, 2, 3, 5, dan 10 maka dikategorikan sebagai penyakit GERD akut atau ringan. Gejala nomor 4, 6, 7, 8, dan 9 merupakan kategori penyakit GERD kronis.

HASIL DAN PEMBAHASAN

Hasil penelitian ini berupa aplikasi sistem pakar berbasis Android. Halaman *splash screen* ditampilkan setelah pengguna membuka aplikasi sistem pakar untuk mendiagnosis penyakit GERD. Tampilan halaman *splash screen* dapat dilihat pada Gambar 4.



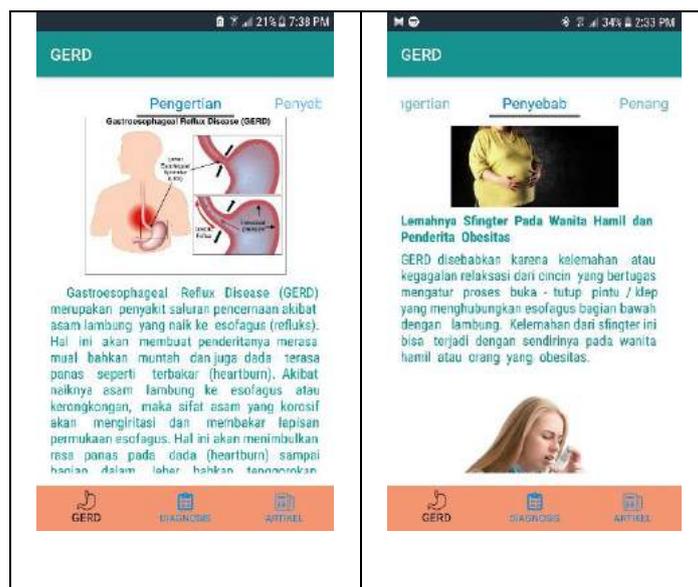
Gambar 4. Tampilan Halaman *Splash Screen*



Gambar 5. Tampilan Halaman Awal Slide 1 (a), Slide 2 (b), Slide 3 (c)

Pada Gambar 5 diberikan tampilan halaman awal untuk slide 1, 2 dan 3. Slide 1 berisi ilustrasi gambar lambung dan informasi singkat mengenai penyakit GERD. Slide kedua yang berisi ilustrasi gambar persentase

yang menunjukkan hasil perhitungan metode CF berupa persentase dan informasi singkat mengenai metode CF tersebut. Slide ketiga yang berisi ilustrasi gambar orang yang menunjukkan pembuat aplikasi.



a. b.

Gambar 6. Tampilan Halaman Utama

Pada Gambar 6 diberikan tampilan halaman utama yang berisi 3 submenu yaitu menu GERD, diagnosis, dan artikel. Pada waktu masuk ke halaman utama, pengguna

langsung melihat menu GERD yang berisi informasi pengertian GERD (Gambar 6a), penyebab (Gambar 6b), dan penanganan penyakit GERD.



Gambar 7. Tampilan Halaman Menu Diagnosis

Pada Gambar 7a. diberikan tampilan halaman data diri yang berada di dalam menu diagnosis. Pengguna harus mengisi data diri seperti nama, jenis kelamin, dan umur. Halaman data diri memiliki 2 tombol yaitu *next* dan *history*. Tombol *next* berfungsi untuk berpindah ke halaman gejala penyakit dan tombol *history* untuk berpindah ke halaman *history* yang berisi basis data hasil diagnosis pengguna. Pada Gambar 7b. diberikan tampilan halaman menu diagnosis yang berisi daftar gejala penyakit GERD. Terdapat 10 gejala yang harus dipilih oleh pengguna berdasarkan dengan kondisi pengguna tersebut.

Halaman berikutnya yang ditampilkan setelah pengguna selesai memilih gejala adalah halaman hasil diagnosis yang berisi hasil perhitungan menggunakan metode *Certainty Factor* (Gambar 7c). Halaman *history* dibuat untuk menyimpan data hasil diagnosis pengguna yang telah melakukan diagnosis pada aplikasi sistem pakar ini.

Pada Gambar 8 diberikan tampilan halaman *history* untuk menyimpan data hasil diagnosis yang berisi nama pengguna yang telah dimasukkan pada halaman data diri, tanggal diagnosis, dan hasil persentase. Tampilan halaman menu artikel berisi daftar artikel dapat

dilihat pada Gambar 9. Terdapat 10 artikel yang memuat informasi mengenai lambung dan penyakit GERD seperti fakta-fakta unik lambung, cara kerja asam lambung, dan lain-lain.



Gambar 8. Tampilan Halaman *History*



a.

b.

Gambar 9. Tampilan Halaman Menu Artikel , a. daftar artikel, b. salah satu artikel

Pada Gambar 9a diberikan tampilan halaman menu artikel yang berisi 10 daftar judul artikel disertai dengan gambar dan sumbernya. Pada Gambar 9b diberikan tampilan

halaman isi artikel yang berisi gambar ilustrasi dan isi dari artikel tersebut. Artikel di atas ditampilkan jika judul dari artikel tersebut dipilih oleh pengguna pada halaman sebelumnya.

Semua artikel yang terdapat di halaman daftar artikel memuat tema yang sama yaitu mengenai kesehatan lambung dan penyakit GERD.

Langkah selanjutnya adalah pengujian

aplikasi terhadap beberapa perangkat *mobile* berbasis Android dengan spesifikasi yang berbeda. Hasil pengujian ditampilkan dalam Tabel 3.

Tabel 3. Hasil Pengujian Perangkat *Mobile*

No	Perangkat	Spesifikasi	Hasil Pengujian
1.	Samsung Galaxy S6	Layar 5.1" Android OS v7.0 (Nougat) CPU Quad-core 1.3 GHz	Dapat diinstalasi dan berjalan dengan baik
2.	Asus Zenfone 3	Layar 5.5" Android OS v8.0 (Oreo) CPU Octa-core 2.0 GHz	Dapat diinstalasi dan berjalan dengan baik
3.	Samsung Galaxy Grand 2	Layar 5.25" Android OS v4.4.2 (Kitkat) CPU Quad-core 1.2 GHz	Dapat diinstalasi dan berjalan dengan baik
4.	Sony Xperia Z1 Compact D5503	Layar 4.3" Android OS v5.0.2 (Lollipop) CPU Qualcomm MSM8974	Dapat diinstalasi dan berjalan dengan baik
5.	Samsung Galaxy Y S5360	Layar 3.0" Android OS v2.3 (Gingerbread) CPU 830 MHz ARMv6	Tidak dapat diinstalasi karena masalah pada sistem operasi.

Pada Tabel 3 ditunjukkan bahwa kelima perangkat *mobile* dengan sistem operasi Android yang telah diuji dapat disimpulkan bahwa aplikasi layanan pengaduan ini dapat berjalan dengan baik pada 5 perangkat Android, sementara pada 1 perangkat Android tidak dapat diinstalasi. Aplikasi ini lebih cocok pada layar berukuran 5.0" karena tampilan yang terlihat lebih rapi dan gambar yang sesuai dengan ukuran layar, serta minimal sistem operasi Android yang dipakai adalah Android Kitkat 4.4.

Uji coba penggunaan aplikasi melibatkan 30 responden untuk mengetahui tingkat akurasi aplikasi dibandingkan dengan hasil diagnosis pakar. Berdasarkan uji coba tersebut diperoleh informasi bahwa terdapat perbedaan hasil diagnosis. Contoh perbedaan

hasil diagnosis tersebut seperti pada hasil diagnosis menggunakan aplikasi menyatakan pasien nomor 4 menderita penyakit GERD Kronis, namun hasil diagnosis dokter menyatakan bahwa pasien tersebut menderita penyakit GERD Akut. Perbedaan hasil diagnosis berikutnya terdapat pada pasien nomor 11, 12, 13, 22, dan 24. Perbandingan hasil diagnosis aplikasi sistem pakar dengan hasil diagnosis dokter sebesar 80% karena terdapat 6 data yang berbeda dari 30 data yang tersedia.

KESIMPULAN DAN SARAN

Pada penelitian ini telah berhasil dibuat sistem pakar untuk mendiagnosis penyakit *Gastroesophageal Reflux Disease* (GERD) menggunakan metode *Certainty Factor*.

Sistem pakar ini digunakan untuk mendiagnosis GERD yang dialami pasien termasuk pada kategori akut atau kronis berdasarkan gejala-gejalanya. Berdasarkan hasil pengujian sistem dinyatakan bahwa sistem pakar untuk mendiagnosis penyakit *Gastroesophageal Reflux Disease* (GERD) yang dibangun menggunakan metode *Certainty Factor* dapat memberikan akurasi hasil diagnosis penyakit *Gastroesophageal Reflux Disease* (GERD) yang baik. Hasil diagnosis akan memberikan solusi bagi pasien untuk pencegahan dan pengobatan dari penyakit yang dialami pasien.

Pada penelitian selanjutnya dapat dilakukan penyempurnaan dan pengembangan seperti menambah beberapa pakar agar nilai CF pakar beragam dan tidak hanya bergantung kepada seorang pakar sehingga menghasilkan informasi yang lebih baik dan lengkap serta akurat.

DAFTAR PUSTAKA

- [1] A. D. Susanto, A. R. L. Syafruddin, N. Sawitri, W. H. Wiyono, F. Yunus, dan S. Prasetyo, “Gambaran klinis dan endoskopi penyakit refluks gastroesofagus (PRGE) pada pasien asma persisten sedang di RS Persahabatan, Jakarta,” *Jurnal Respirologi Indonesia*, vol. 29, no. 1, hal. 31 – 38, 2009.
- [2] M. Arifin, Slamini, dan W. E. Y. Retnani, “Penerapan metode certainty factor untuk sistem pakar diagnosis hama dan penyakit pada tanaman tembakau,” *Jurnal Berkala Sainstek*, vol. 5, no. 1, hal. 21 – 28, 2017.
- [3] M. Jumiyati, B. Pramono, dan L. O. Hasanudin, “Aplikasi sistem pakar diagnosis penyakit tb paru pada anak dengan metode logika fuzzy berbasis Android,” *semanTIK*, vol.1, no. 1, hal. 25 – 32, 2015.
- [4] M. P. Hardiyanti, R. R. Isnanto, dan I. P. Windasari, “Aplikasi sistem pakar berbasis mobile untuk diagnosis dini meningitis,” *Jurnal Teknologi dan Sistem Komputer*, vol. 5, no. 2, hal. 83 – 88, 2017.
- [5] F. Akmal dan S. Winiarti, “Sistem pakar untuk mendiagnosa penyakit lambung dengan implementasi metode CBR (Case-Based Reasoning) berbasis web,” *Jurnal Sarjana Teknik Informatika*, vol. 2, no.1, hal. 790 – 800, 2014.
- [6] A. Syatibi, “Sistem pakar diagnosa awal penyakit kulit sapi berbasis web dengan menggunakan metode certainty factor,” *Tesis*, Program Studi Magister Sistem Informasi, Universitas Diponegoro, Semarang, 2012.
- [7] E. P. Gunawan dan R. Wardoyo, “An expert system using certainty factor for determining insomnia acupoint,” *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, vol. 12, no. 2, hal. 119 – 128, 2018.
- [8] P. Hasan, E. W. Sholeha, Y. N. Tetik, dan Kusriani, “Sistem pakar diagnosa penyakit kolesterol dan asam urat

menggunakan metode certainty factor,”
*Jurnal Ilmiah Sistem Informasi dan
Teknik Informatika (SISFOTENIKA)*,
vol. 9, no. 1, hal. 47 – 58, 2019.

IMPLEMENTASI *CHATBOT* PADA PENDAFTARAN MAHASISWA BARU MENGGUNAKAN *RECURRENT NEURAL NETWORK*

¹Tjut Awaliyah Zuraiyah, ²Dian Kartika Utami, ³Degi Herlambang

^{1,2,3}Program Studi Ilmu Komputer, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Pakuan

Jl. Pakuan, Tegallega, Kota Bogor, Jawa Barat, 16143

¹tjut.awaliyah@unpak.ac.id, ²diankartikautami@unpak.ac.id, ³ddegih@gmail.com

Abstrak

Chatbot adalah perangkat lunak yang dapat berkomunikasi dengan manusia menggunakan bahasa alami. Model percakapan menggunakan kecerdasan buatan agar mampu memahami ucapan pengguna dan memberi tanggapan yang relevan dengan masalah yang dibahas oleh pengguna. Pendaftaran mahasiswa baru memerlukan banyak informasi mengenai prosedur pendaftaran di perguruan tinggi. Website pendaftaran online di Universitas Pakuan masih sebatas berisi informasi umum. Penelitian ini bertujuan untuk membuat suatu aplikasi *Chatbot* otomatis yang dapat berkomunikasi dengan manusia mengenai informasi pendaftaran mahasiswa baru di Universitas Pakuan menggunakan *Recurrent Neural Network (RNN)* untuk klasifikasi teks. Aplikasi *Chatbot* diimplementasikan menggunakan bahasa pemrograman Python dan Telegram API. Tahapan pada implementasi *Chatbot* terdiri dari preprocessing, transformasi data ke format .JSON, pelatihan data, bag of word dan full connection. Pengujian aplikasi *Chatbot* menggunakan data sebanyak 251 kalimat pertanyaan tentang pendaftaran mahasiswa baru di Universitas Pakuan. Hasil pengujian menunjukkan bahwa *Chatbot* dapat menjawab pertanyaan mengenai pendaftaran mahasiswa baru dengan akurasi sebesar 88%, presisi sebesar 95% dan recall sebesar 92%.

Kata Kunci: *Chatbot*, pendaftaran mahasiswa baru, *RNN*, Python, Telegram API.

Abstract

Chatbot is software that can communicate with humans using natural language. The conversation model uses artificial intelligence to be able to understand the user's words and provide responses that are relevant to the problem discussed by the user. Enrollment for new students requires a lot of information about registration procedures in tertiary institutions. The online registration website at Pakuan University is still limited to general information. This study aims to create an automated *Chatbot* application that can communicate with humans about new student registration information at Pakuan University using the *Recurrent Neural Network (RNN)* for text classification. The *Chatbot* application is implemented using the Python programming language and the Telegram API. The stages in the *Chatbot* implementation consist of preprocessing, data transformation to .JSON format, data training, bag of words and full connection. Testing the *Chatbot* application uses data as much as 251 question sentences about enrolling new students at Pakuan University. The test results show that *Chatbot* can answer questions regarding registration of new students with an accuracy of 88%, a precision of 95% and a recall of 92%.

Keywords: *Chatbot*, registration of new students, Python, Telegram API.

PENDAHULUAN

Sebagian besar masyarakat ingin mendapat informasi yang mudah, cepat dan akurat. Pendaftaran calon mahasiswa baru memerlukan banyak informasi prosedur pendaftaran di setiap perguruan tinggi. Sarana informasi pendaftaran *online* di Universitas Pakuan melalui halaman *website* masih sebatas informasi umum. Akan tetapi, calon pendaftar membutuhkan komunikasi dua arah untuk menanyakan informasi lebih detail lagi. Terdapat satu tempat ruang informasi untuk melayani calon pendaftar yang ingin bertanya lebih lanjut namun tidak efisien karena pertanyaan berulang dan serupa berkaitan dengan pendaftaran. Untuk mendukung kebutuhan informasi yang akurat dan *up-to-date* terkait dengan pendaftaran mahasiswa baru, maka diperlukan sebuah sistem berbasis teknologi informasi yang dapat merangkum dengan baik berbagai data pendaftaran serta menampilkan informasinya pada pengguna.

Chatbot juga dikenal sebagai *talkbot*, *chatterbox*, *Bot*, *IM bot* atau *artificial conversational entity*. *Chatbot* adalah program komputer yang meniru percakapan manusia dalam format alami termasuk teks atau bahasa lisan menggunakan teknik kecerdasan buatan seperti *Natural Language Processing* (NLP), gambar dan pemrosesan video, serta analisis audio [1].

Chatbot dapat digunakan dalam bidang pendidikan sebagai media pembelajaran. Penelitian yang dilakukan oleh Wijaya,

Sarosa, dan Tolle mengenai pembuatan *Chatbot* sebagai media pembelajaran bahasa Java pada *Google Classroom* dan *Facebook Messenger*. Hasil pengujian *Chatbot* menunjukkan bahwa nilai ujian siswa lebih tinggi pada kelas yang menggunakan *Chatbot* dibandingkan kelas yang menggunakan pembelajaran secara konvensional [2]. Penelitian lain mengenai *Chatbot* sebagai *virtual assistant* berbasis *website* menggunakan AIML yang memberikan informasi kepada mahasiswa mengenai Program Studi Teknik Informatika, Universitas Muhamadiyah Malang. Selain itu, pada *Chatbot* dapat ditambahkan pengetahuan baru jika terdapat pertanyaan yang tidak terdapat pada basis data *Artificial Linguistic Internet Computer Entity* (ALICE). Akurasi *Chatbot* yang dibuat pada penelitian tersebut sebesar 80% [3].

Beberapa *Chatbot* tersedia tetapi pengguna harus melakukan pengaturan frase kunci secara manual, mengusulkan dan mengembangkan pertanyaan yang sering diajukan [4]. *Recurrent Neural Networks* (RNN) merupakan arsitektur *deep learning* yang populer serta sangat menjanjikan untuk menyelesaikan berbagai persoalan yang terkait dengan NLP. *Chatbot* secara otomatis menanggapi pengguna menggunakan RNN dalam bentuk *Long Short-Term Memory* (LSTM) untuk klasifikasi teks [5]. Model RNN digunakan agar mesin dapat memahami bahasa manusia, mulai dari cara berkomunikasi, mendengarkan, mengenali percakapan, hingga memahami tata bahasa dan aksen [6].

Pada penelitian ini dibuat suatu aplikasi *Chatbot* otomatis yang dapat berkomunikasi dengan pengguna mengenai informasi pendaftaran mahasiswa baru di Universitas Pakuan. Implementasi aplikasi *Chatbot* pada penelitian ini menggunakan *Recurrent Neural Network* (RNN).

METODE PENELITIAN

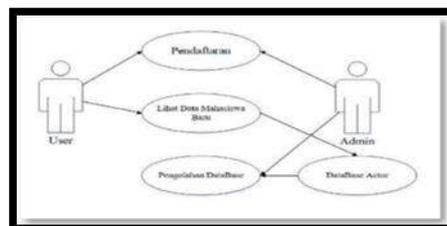
Tahap Perencanaan

Tahap proses perencanaan ini merupakan tahapan awal yang berupa pengumpulan data yang dibutuhkan dalam pembuatan *Chatbot* untuk calon mahasiswa baru Universitas Pakuan. Pada tahap ini mencari permasalahan yang terjadi menyangkut pendaftaran calon mahasiswa baru serta menentukan bagaimana solusi pemecahan masalahnya. Tahapan ini dilakukan untuk mendapatkan data pendaftaran

dan proses pendaftaran secara langsung melalui wawancara dan studi literatur. Penelitian ini mengumpulkan data secara langsung ke lapangan menggunakan teknik pengumpulan data wawancara pegawai BAAK maupun calon mahasiswa baru sehingga dapat diambil data yang diperlukan guna kepentingan penelitian ini. Selanjutnya dilakukan pembuatan sebuah dokumen yaitu kuisisioner yang akan diberikan ke setiap responden guna mencatat setiap jawaban yang nantinya akan menjadi sebuah sumber data penelitian ini.

Tahap perancangan

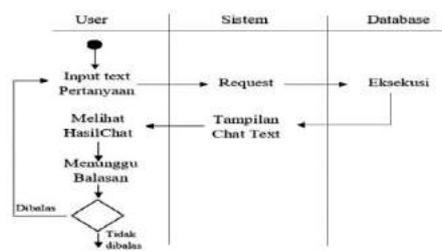
Tahapan perancangan secara detail digunakan untuk menentukan sistem yang akan dibangun meliputi pembuatan *use case diagram*, *activity diagram* dan *sequence diagram*. Gambar 2 menunjukkan *use case diagram* untuk pendaftaran mahasiswa baru.



Gambar 2. *Use Case Diagram*

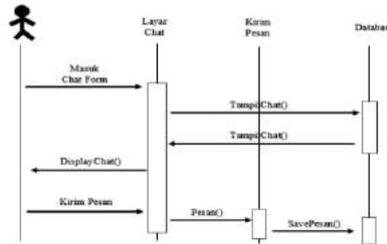
Activity diagram merupakan model analisis yang digunakan atau menggambarkan sebuah proses aktivitas. Gambar 3 menunjukkan

activity diagram aplikasi *Chatbot* pendaftaran mahasiswa baru.



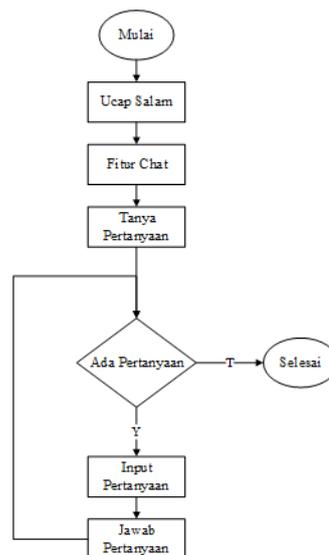
Gambar 3. *Activity Diagram Chatbot*

Sequence diagram menggambarkan menerima sebuah *message*. Gambar 4 objek dan relasinya termasuk kronologi memperlihatkan *sequence diagram Chatbot* (urutan) perubahan secara logis setelah pendaftaran mahasiswa baru.



Gambar 4. *Sequence Diagram Chatbot*

Flowchart sistem dibangun guna dapat memberikan gambaran seutuhnya dari sistem terlihat pada Gambar 5. kebutuhan data yang ada. Adapun *flowchart* sistem terlihat pada Gambar 5.



Gambar 5. *Flowchart Sistem*

Tahap Implementasi

Pembangunan aplikasi ini menggunakan pemrograman bahasa Python. Pengujian data menggunakan *tool tensorflow* atau *deep learning* guna mendapat tingkat akurasi tinggi untuk jawaban dari pertanyaan yang dimasukkan pengguna. Koneksi terhadap *Telegram API* digunakan sebagai media

hosting untuk *platform Telegram* secara *realtime*.

Implementasi *Chatbot* secara detail terbagi menjadi beberapa tahapan, yaitu tahap *preprocessing*, tahap transformasi data menjadi *file .JSON*, lalu tahap pelatihan data, tahap *bag of words*, dan tahap terakhir *full connection*.

a. *Text Preprocessing*

Tahapan *text preprocessing* adalah tahapan dimana aplikasi melakukan seleksi data yang diproses pada setiap dokumen. Proses *preprocessing* meliputi 4 tahapan yaitu *case folding*, *tokenizing*, *filtering* dan *stemming*.

b. Transformasi Data Menjadi File .JSON

Trasformasi data menjadi sebuah *file .JSON* diperlukan untuk sistem agar dapat membaca setiap pola pertanyaan yang muncul beserta target jawabannya. Cara mentransformasikan data ke dalam *file .JSON* ialah dengan memisahkan setiap kata yang ada didalam data menjadi beberapa bagian yaitu:

1. *Tags* (kategori) merupakan sebuah pengetahuan atau kategori yang menjadi acuan sistem dalam menentukan respon.
2. *Pattern* (pola) adalah sebuah rangkaian huruf yang diharapkan sesuai atau cocok dengan satu atau bahkan lebih dengan masukan pengguna.
3. *Responses* (output) adalah hasil jawaban yang akan dikeluarkan berdasarkan *index tags* dan *pattern* yang ditentukan oleh sistem.

c. Pelatihan Data

Setelah data terkumpul, sistem menguji data agar mendapat tingkat akurasi *output* yang baik dan memuaskan. Langkah pelatihan data sebagai berikut:

1. Mengubah bentuk pola pada data .JSON ke dalam bentuk matriks.

2. *Flattering* atau memasukan setiap baris matriks ke dalam *node input layer* yang akan memberi sinyal ke *hidden layer*.

3. Menilai *output* setelah selesai perhitungan dari *input layer* ke *hidden layer* yang dimana akan diberikan setiap bobot oleh sistem.

d. *Bag of Words*

Pada tahap ini dilakukan transformasi nilai *bag of words* dengan mentransformasi teks *input* dari pengguna menjadi bentuk bilangan biner.

e. *Full Connected*

Setelah mendapat nilai biner dari teks *input*, maka dimasukan nilai biner tersebut ke dalam *input layer* yang nantinya akan memberi sinyal kepada *hidden layer* dari arsitektur *neural network*.

Tahap Pengujian

Pengukuran performa dari aplikasi *Chatbot* yang dibuat pada penelitian ini berdasarkan hasil *confusion matrix* dengan menghitung tingkat *accuracy*, *precision* dan *recall*.

a. *Accuracy*

Akurasi merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Perhitungan akurasi menggunakan Persamaan (1)

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (1)$$

b. *Precision*

Presisi merupakan rasio prediksi benar

positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Perhitungan presisi menggunakan Persamaan (2).

$$Presisi = \frac{TP}{FP+TP} * 100\% \quad (2)$$

c. *Recall*

Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. *Recall* dihitung menggunakan Persamaan (3).

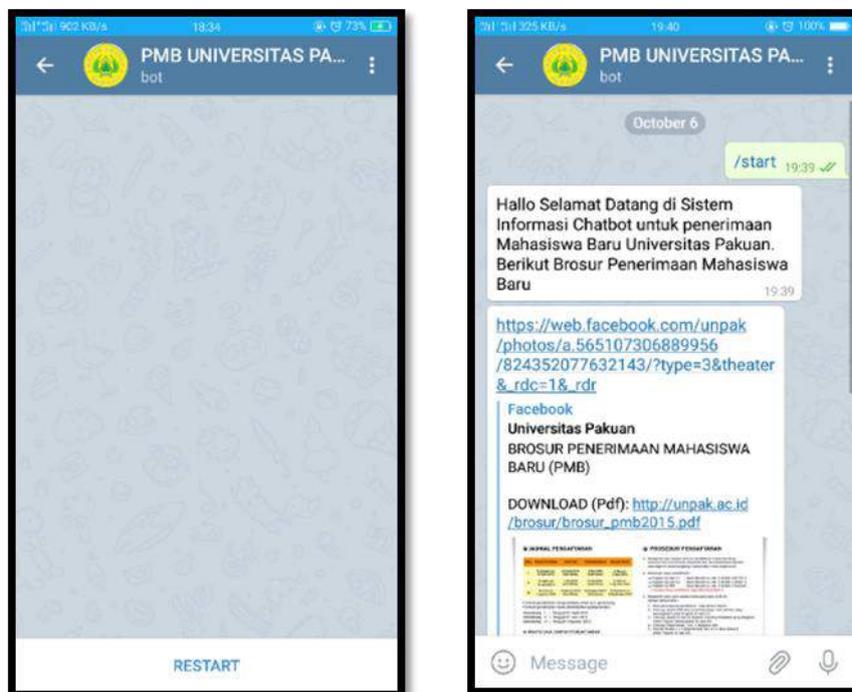
$$Recall = \frac{TP}{FN+TP} * 100\% \quad (3)$$

HASIL DAN PEMBAHASAN

Pada aplikasi *Chatbot* pendaftaran

mahasiswa baru Universitas Pakuan terdapat halaman *chat* atau pesan menggunakan *platform* Telegram. Dengan menggunakan *RNN* sistem dengan mudah menjawab pertanyaan dari pengguna. Apabila *Chatbot* tidak mengerti apa yang dimasukkan oleh pengguna maka *Chatbot* akan menanyakan kembali dan mengoreksi pertanyaan yang diajukan oleh pengguna sesuai dengan format sistem yang ada.

Halaman *chat* Telegram merupakan tampilan awal ketika pengguna membuka aplikasi Telegram terlihat pada Gambar 6 (a) dan (b). Pada saat pengguna membuka kontak *Chatbot* maka sistem akan otomatis mengirimkan gambar mengenai pendaftaran mahasiswa baru Universitas Pakuan.



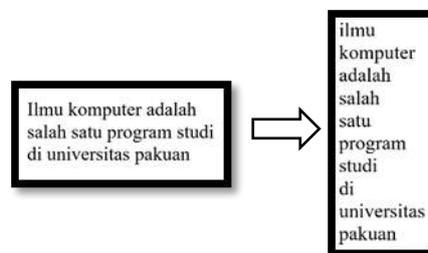
Gambar 6. (a) Halaman Telegram, (b) Halaman *Chatbot Start*

Pada tahap *text preprocessing* dimulai dengan tahap *case folding*. Dalam penelitian ini *case folding* digunakan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Selanjutnya dilakukan tahap *tokenizing* yang merupakan tahapan pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Contoh hasil tahap *tokenizing* dapat dilihat pada Gambar 7.

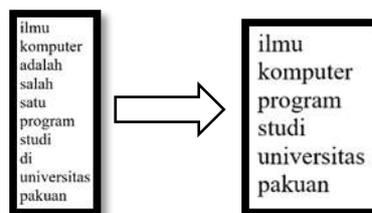
Tahap *text preprocessing* selanjutnya yaitu tahap *filtering*. Pada tahap ini digunakan

stoplist (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting). Contoh *filtering* dapat dilihat pada Gambar 8.

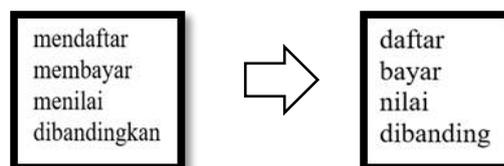
Tahap *stemming* diperlukan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen dan juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk yang berbeda. Gambar 9 merupakan contoh hasil tahap *stemming* *Chatbot*.



Gambar 7. Hasil Tahap *Tokenizing*



Gambar 8. Hasil Tahap *Filtering*



Gambar 9. Hasil Tahap *Stemming*

Setelah data terkumpul lalu sistem menguji data agar mendapat tingkat akurasi *output* yang baik dan memuaskan. Tabel 1

merupakan contoh data pelatihan yang digunakan pada penelitian ini.

Tabel 1. Sample Pelatihan Data

Tags		Pattern		Target
Salam	Hei	apa kabar	hello	Ada yang bisa dibantu?
Daftar	bagaimana cara daftar	ilmu komputer	fasilitas	Lihat Brosur ini...
Terimakasih	terimakasih	thanks	makasih	Kapan pun!

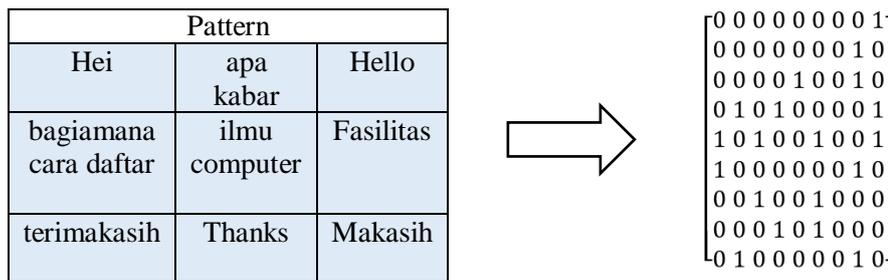
Tabel 1 merupakan contoh sebuah file .JSON akan digunakan dalam pelatihan data. Pelatihan data dilakukan agar mesin dapat mengenali *pattern* atau pola teks *input* pengguna yang nanti akan digunakan untuk membantu menjawab pertanyaan dari pengguna pada sistem *Chatbot*. Langkah-langkah pada pelatihan data sebagai berikut:

1. Mengubah bentuk pola pada data

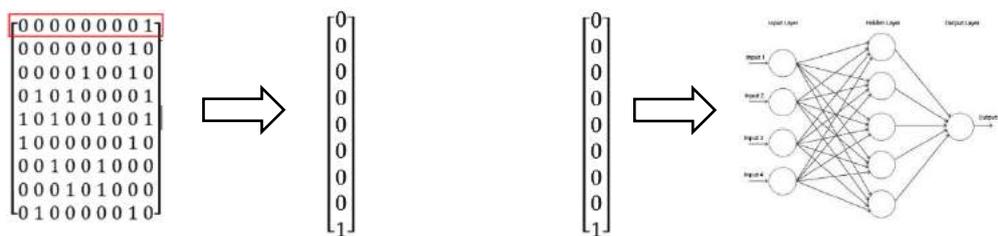
.JSON ke dalam bentuk matriks seperti pada Gambar 10.

2. *Flattering* atau memasukan setiap baris matriks ke dalam *node input layer* yang akan memberi sinyal ke *hidden layer*. Contoh hasil *flattering* dapat dilihat pada Gambar 11.

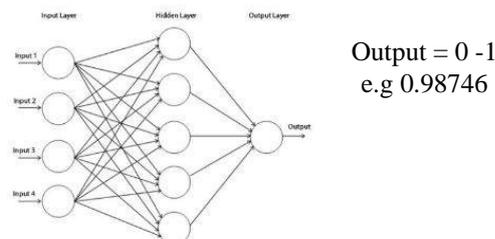
3. Nilai *output* pelatihan data terlihat pada Gambar 12.



Gambar 10. Hasil Mengubah Pola ke Matriks



Gambar 11. Hasil *Flattering*



Gambar 12. Nilai *Output* Pelatihan Data

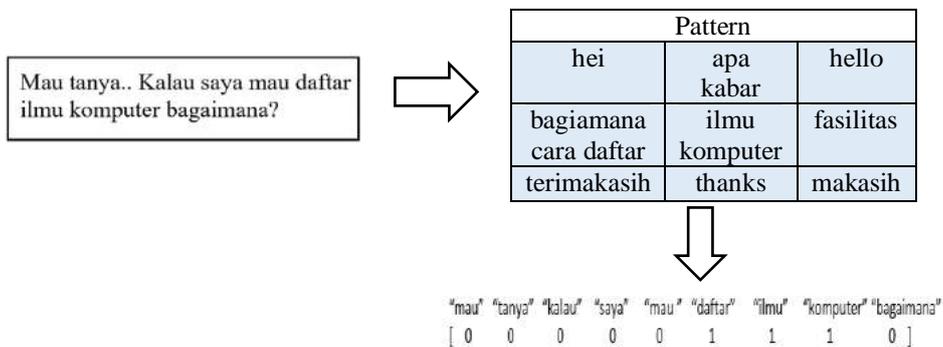
Bobot pada hidden layer tidak dapat dimasukkan langsung oleh pengguna karena bobot merupakan nilai yang otomatis diberikan oleh sistem, dan disimpan untuk dijadikan tingkat akurasi oleh pola sebelumnya.

Tahap selanjutnya dilakukan transformasi nilai *bag of words*. Contoh hasil transformasi nilai *bag of words* terlihat pada Gambar 13.

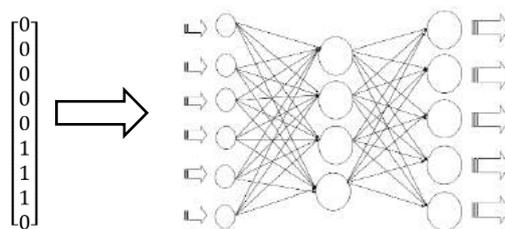
Setelah mendapat nilai biner dari teks *input*, maka pada tahap *full connected*, nilai biner tersebut dimasukan ke dalam *input layer*

yang nantinya akan memberi sinyal kepada *hidden layer* dari arsitektur *neural network* seperti pada Gambar 14. Nilai *output* ditentukan oleh sistem setelah melakukan perhitungan dari *input layer* ke *hidden layer*, sehingga dapat menentukan jawaban apa yang akan diberikan oleh sistem *Chatbot*.

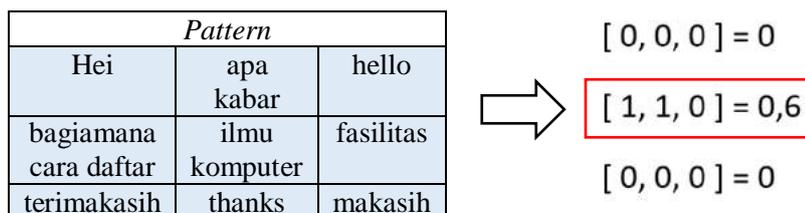
Setelah didapatkan nilai-nilai output dari setiap pola, kemudian diambil nilai yang paling tinggi untuk menjawab pertanyaan dilihat dari hasil penentuan respon seperti pada Gambar 15.



Gambar 13. Hasil Transformasi Nilai *Bag of Words*



Gambar 14. Tahap *Full Connected*



Gambar 15. Contoh Penentuan Respon

Jumlah data secara keseluruhan data uji sebanyak 50 data. Kemudian sebanyak 300 data, untuk pengujian data didapatkan hasil uji seperti terlihat pada latih sebanyak 250 dan akan diuji dengan Tabel 2.

Tabel 2. Nilai *Confusion Matrix*

	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	40	3
<i>Negative</i>	2	4

Pengukuran *performance* berdasarkan menghitung tingkat *accuracy*, *precision* dan hasil *confusion matrix* pada Tabel 2 dengan *recall* sebagai berikut:

$$\text{Akurasi} = \frac{40 + 4}{40 + 5 + 3 + 2} = \frac{44}{50} = 88\%$$

$$\text{Presisi} = \frac{40}{40 + 2} = \frac{40}{42} = 95\%$$

$$\text{Recall} = \frac{40}{40 + 3} = \frac{40}{43} = 92\%$$

Berdasarkan hasil perhitungan performa aplikasi, aplikasi memiliki akurasi sebesar 99%, presisi sebesar 95%, dan *recall* sebesar 92%.

KESIMPULAN DAN SARAN

Pada penelitian ini telah berhasil mengimplementasikan aplikasi *Chatbot* menggunakan *Recurrent Neural Networks* (RNN) untuk pelatihan data yang dapat membantu menemukan jawaban yang sesuai dari pertanyaan pengguna. JSON digunakan untuk menyimpan data serta menampilkan data-data yang tersimpan menggunakan perangkat lunak Python untuk membantu dalam pembuatan dan pengembangan aplikasi ini. Pada nilai-nilai pada *confusion matrix* didapatkan tingkat akurasi aplikasi sebesar

88%, presisi sebesar 95% dan *recall* sebesar 92%. Pembuatan aplikasi ini bertujuan untuk mempermudah pengguna dalam mendapatkan informasi yang cepat dan akurat secara *realtime*. Pengguna dapat langsung mendapatkan informasi saat itu juga dengan menggunakan aplikasi Telegram tanpa harus datang jauh ke kampus yang cukup jauh bagi sebagian orang.

Penelitian selanjutnya dapat mengembangkan aplikasi *Chatbot* dari sisi keamanan aplikasi. Selain itu, pada penelitian selanjutnya dapat menggunakan metode *deep learning* lainnya sehingga memberikan hasil akurasi yang lebih baik.

DAFTAR PUSTAKA

- [1] K. Bala, M. Kumar, S. Hulawale, dan S. Pandita, "Chat-bot for college management

- system using A.I,” *International Research Journal Of Engineering and Technology (IRJET)*, vol. 4, no. 11, hal. 2030 – 2033, 2017.
- [2] M. H. Wijaya, M. Sarosa, dan H. Tolle, “Rancang bangun Chatbot pembelajaran Java pada Google Classroom dan Facebook Messenger,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 5, no. 3, hal. 287 – 296, 2018.
- [3] Maskur, “Perancangan Chatbot pusat informasi mahasiswa menggunakan AIML sebagai virtual assistant berbasis web”, *KINETIK*, vol. 1, no. 3, hal. 123 – 128, 2016.
- [4] S. P. Lende dan M. M. Raghuwanshi, “Question answering system on education acts using NLP techniques,” Dipresentasikan pada IEEE World Conference on Futuristic Trends in Research and Innovation for Social Welfare, Coimbatore, India, 2016.
- [5] P. Muangkammuen, N. lutiruk, dan K. R. Saikaew, “Automated Thai-FAQ Chatbot using RNN-LSTM,” In Proc. 22nd IEEE International Computer Science and Engineering Conference (ICSEC), 2018, hal. 1 – 4.
- [6] P. Tanwar, T. V. Prasad, dan K. Datta, “An effective reasoning algorithm for question answering system,” *International Journal of Advanced Computer Science and Applications (IJACSA) Special Issue on Natural Language Processing*, hal. 52 – 57, 2014.

KLASIFIKASI CITRA KEBAKARAN DAN NON KEBAKARAN MENGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

Zaky Farhan Abror

Fakultas Teknologi Industri Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
zakyfarhan114@gmail.com

Abstrak

Kebakaran menimbulkan kerugian yang dapat membahayakan jiwa maupun harta benda. Lokasi dan waktu terjadinya kebakaran sulit untuk diprediksi. Citra dari kamera CCTV dapat digunakan untuk memantau terjadinya kebakaran pada suatu lokasi. Dengan pemanfaatan perkembangan teknologi saat ini, data citra tersebut dapat diolah sehingga memberikan informasi terjadinya kebakaran pada suatu lokasi dengan lebih cepat dibandingkan secara manual. Salah satu metode yang dapat digunakan dalam pengolahan citra kebakaran adalah deep learning. Salah satu model deep learning yang memiliki kinerja terbaik dalam pengenalan dan klasifikasi citra yaitu Convolutional Neural Network (CNN). Penelitian terdiri dari beberapa tahap yaitu pengumpulan data, pra-pemrosesan data, pembagian data, perancangan arsitektur model CNN, pelatihan, pengujian, dan hasil. Jumlah data citra yang digunakan pada penelitian ini yaitu 950 citra yang terdiri dari 700 data pelatihan, 170 data validasi, dan 80 data pengujian. Arsitektur model CNN yang digunakan terdiri dari dua lapisan konvolusi dan dua lapisan pooling. Pada pelatihan, model dikenalkan dengan pola citra kebakaran dan non kebakaran kemudian divalidasi. Pada tahap pengujian, model mengklasifikasikan citra kebakaran dan non kebakaran. Tingkat akurasi model pada tahap pelatihan sebesar 98.8% dan pada tahap pengujian sebesar 90%.

Kata Kunci: CNN, deep learning, kebakaran, klasifikasi.

Abstract

Fires cause losses that can endanger lives and property. The location and time of the fire are difficult to predict. The image from a CCTV camera can be used to monitor the occurrence of a fire at a location. By utilizing current technological developments, the image data can be processed so as to provide information on the occurrence of fires at a location faster than manually. One method that can be used in fire image processing is deep learning. One of the deep learning models that has the best performance in image recognition and classification is Convolutional Neural Network (CNN). The study consisted of several stages, namely data collection, data pre-processing, data sharing, architectural design of the CNN model, training, testing, and results. The amount of image data used in this study is 950 images consisting of 700 training data, 170 validation data, and 80 testing data. The CNN architecture model used consists of two convolution layers and two pooling layers. In training, the model is introduced to fire and non-fire image patterns and then validated. In the testing phase, the model classifies fire and non-fire images. The accuracy of the model at the training stage was 98.8% and at the testing stage it was 90%.

Keywords: classification, CNN, deep learning, fire.

PENDAHULUAN

Kebakaran merupakan salah satu bencana nonalam yang dapat terjadi di hutan, lahan, perumahan atau pemukiman padat penduduk. Definisi kebakaran hutan dan lahan menurut Badan Nasional Penanggulangan Bencana (BNPB) adalah suatu keadaan di mana hutan dan lahan dilanda api, sehingga mengakibatkan kerusakan hutan dan lahan yang menimbulkan kerugian ekonomis dan atau nilai lingkungan. Definisi kebakaran secara umum menurut BNPB adalah situasi dimana bangunan pada suatu tempat seperti rumah atau pemukiman, pabrik, pasar, gedung, dan lain-lain dilanda api yang menimbulkan korban dan/atau kerugian [1]. Berdasarkan definisi kebakaran tersebut, kebakaran menimbulkan kerugian yang dapat membahayakan jiwa maupun harta benda.

Lokasi dan waktu terjadinya kebakaran sulit untuk diprediksi. Pemberian info terjadinya kebakaran di suatu tempat pada pihak berwenang yang dapat menanggulangi kebakaran memerlukan waktu jika dilakukan secara manual. Saat ini, kamera CCTV pada beberapa lokasi telah banyak dipasang untuk memantau setiap saat kejadian di lokasi tersebut. Citra dari kamera CCTV dapat digunakan untuk memantau terjadinya kebakaran pada suatu lokasi. Dengan pemanfaatan perkembangan teknologi saat ini, data citra tersebut dapat diolah sehingga memberikan informasi terjadinya kebakaran pada suatu lokasi dengan lebih cepat dibandingkan secara manual.

Penelitian mengenai pengolahan citra untuk identifikasi titik api telah dilakukan oleh Hardiyanto dan Sartika. Pada penelitian tersebut, pengolahan citra yang dilakukan adalah melakukan konversi citra foto udara dari UAV (*Unmanned Aerial Vehicle*) pada ruang warna RGB ke HSV. Hasil penelitian menunjukkan bahwa sistem deteksi api dari citra digital foto udara memiliki akurasi yang rendah jika objek api terlalu tinggi, titik api yang tertutup oleh asap tebal, dan titik api yang terlalu kecil [2].

Metode lain yang dapat digunakan dalam pengolahan citra kebakaran adalah *Machine Learning* (ML). ML adalah metode yang dapat belajar dari data (*data-driven method*) sehingga menjadi cerdas. Cerdas dalam artian memiliki kemampuan generalisasi terhadap data baru yang belum dipelajari sebelumnya. *Deep learning* adalah salah satu kelas dari algoritma *machine learning* yang memiliki *multiple layer* yang terdiri dari unit-unit *processing* yang nonlinier [3]. Salah satu model *deep learning* yang memiliki kinerja terbaik dalam pengenalan dan klasifikasi citra yaitu *Convolutional Neural Network* (CNN) [4,5]. CNN mampu melakukan proses pembelajaran mandiri untuk pengenalan objek, ekstraksi objek dan klasifikasi citra [6]. CNN merupakan arsitektur yang terinspirasi oleh sistem safaf biologi yang terdiri dari konvolusi, *pooling*, dan *fully-connected layers* [7]. Pada CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, sehingga metode ini cocok untuk pemrosesan dengan *input* berupa citra [8].

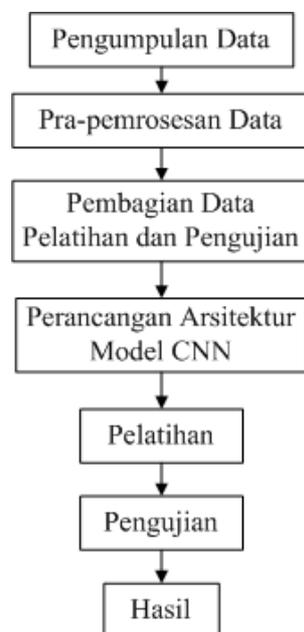
Penelitian menggunakan CNN dalam mengklasifikasi citra telah dilakukan oleh beberapa peneliti. Pada penelitian yang dilakukan oleh Suartika, Wijaya, dan Soelaiman membahas mengenai klasifikasi citra menggunakan CNN pada basis data Caltech 101. Hasil penelitian menunjukkan bahwa CNN cukup handal dalam mengklasifikasi citra pada basis data Caltech 101 [9]. Setiawan melakukan penelitian mengenai klasifikasi penyakit pada citra daun menggunakan CNN. Hasil penelitian yang dilakukan oleh Setiawan menunjukkan bahwa CNN dapat mengklasifikasi 9 jenis penyakit pada citra daun secara otomatis dengan akurasi yang baik [10]. Implementasi CNN terhadap citra instrumen alat musik gamelan menggunakan *keras* dilakukan oleh Pradana.

Hasil uji coba menunjukkan bahwa klasifikasi terhadap citra instrumen alat musik gamelan yaitu bonang, gambang dan kendang memiliki akurasi yang baik [11].

Oleh karena itu, pada penelitian ini dilakukan klasifikasi pada citra kebakaran dan non kebakaran menggunakan CNN. Hasil klasifikasi citra diharapkan dapat mendeteksi dini terjadinya kebakaran pada lokasi tertentu sehingga dapat ditanggulangi dengan lebih cepat.

METODE PENELITIAN

Penelitian ini bertujuan untuk mengklasifikasikan citra kebakaran dan non kebakaran menggunakan *Convolutional Neural Network*. Adapun tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Skema Penelitian

Berdasarkan Gambar 1, tahap awal penelitian dilakukan pengumpulan data citra

kebakaran dan non kebakaran. Setelah data dikumpulkan lalu dilakukan pra-pemrosesan

data citra. Tahap selanjutnya adalah pembagian data citra yaitu data untuk pelatihan dan pengujian. Setelah pembagian data selesai maka dilanjutkan dengan perancangan model *Convolutional Neural Network* (CNN). Model CNN yang telah dirancang lalu digunakan untuk proses pelatihan dan dilanjutkan untuk proses pengujian.

Pengumpulan Data

Dataset penelitian ini adalah citra kebakaran dan non kebakaran. Data tersebut diperoleh dengan cara *crawling* pada situs pencarian Google [12]. Jumlah citra yang terdapat pada *dataset* adalah 950 citra dengan *format* jpg.

Pra-pemrosesan Data

Sebelum dilakukan proses klasifikasi menggunakan CNN, citra terlebih dahulu dilakukan pra-pemrosesan. Citra kebakaran dan non kebakaran yang telah dikumpulkan memiliki ukuran piksel yang berbeda-beda. Oleh karena itu, tahapan pra-pemrosesan data yang dilakukan adalah mengubah ukuran piksel citra asli sehingga setiap citra memiliki ukuran yang sama yaitu 150×150 piksel.

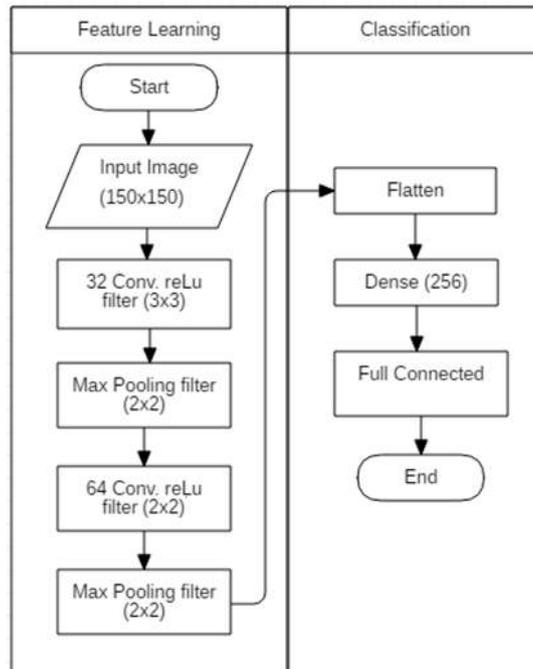
Pembagian Data

Data citra kebakaran dan non kebakakan selanjutnya dibagi menjadi tiga jenis, yaitu data pelatihan, data validasi, dan data pengujian. Data pelatihan digunakan untuk melakukan pembelajaran, data validasi

digunakan untuk melakukan pengujian terhadap hasil pembelajaran pada tahap pelatihan dan data pengujian digunakan untuk melakukan pengujian setelah pelatihan. Data pelatihan sebanyak 700 citra, data validasi sebanyak 170 citra, dan data pengujian sebanyak 80 citra.

Perancangan Arsitektur Model CNN

Pada rancangan model CNN yang digunakan pada penelitian ini, citra masukan yang digunakan berukuran $150 \times 150 \times 3$. Dengan 150×150 merupakan panjang dan lebar citra serta 3 merupakan komponen warna yang dimiliki yaitu RGB (*Red, Green, Blue*). CNN yang digunakan terdiri dari 2 lapisan konvolusi dan 2 lapisan *pooling*. Citra masukan akan dikonvolusi pada langkah pertama dengan ukuran *filter layer* 3×3 , dimana proses konvolusi ini akan mereduksi baris dan kolom pada citra. Tahap selanjutnya adalah melakukan operasi *maxpooling*. Operasi yang dilakukan yaitu perkalian matriks antara hasil dari konvolusi sebelumnya yang berukuran 3×3 dengan *filter pooling* 2×2 . Pada tahap kedua dilakukan operasi konvolusi dengan ukuran *filter layer* 2×2 dan *filter pooling* 2×2 . Setelah proses konvolusi dan *pooling* telah selesai dilakukan selanjutnya proses *flatten* dan *fully connected*. Proses ini bertujuan untuk mengubah *feature map* hasil *pooling layer* ke dalam bentuk vektor. Gambar 2 merupakan rancangan dari model *Convolutional Neural Network* yang digunakan pada penelitian ini.



Gambar 2. Rancangan Arsitektur Model CNN

Pelatihan

Pada proses pelatihan, ada beberapa nilai parameter yang diinisialisasi antara lain jumlah *epoch* dan *learning rate*. Jumlah *epoch* dan *learning rate* yang memberikan akurasi optimum tidak diketahui. Oleh karena itu, pada penelitian ini dilakukan pelatihan menggunakan beberapa jumlah *epoch* dan beberapa nilai *learning rate* sehingga diperoleh jumlah *epoch* dan *learning rate* yang memberikan akurasi yang optimum.

Pengujian

Setelah melakukan proses pelatihan pada model, dilakukan proses pengujian untuk menguji model tersebut mengklasifikasi citra sesuai dengan kelasnya. Proses pengujian akan dilakukan menggunakan data pengujian sebanyak 80 citra.

Hasil Klasifikasi

Penentuan baik atau tidaknya performa suatu model klasifikasi dapat dilihat dari parameter pengukuran performanya, yaitu tingkat akurasi, sensitivitas, dan presisi. Untuk menghitung faktor-faktor tersebut diperlukan sebuah matriks yang biasa disebut *confusion matrix*. Beberapa nilai di dalam *confusion matrix* yaitu: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Seluruh kemungkinan kejadian sebenarnya positif (P) dan seluruh kemungkinan kejadian sebenarnya negatif (N). Nilai tersebut dapat digunakan untuk menghitung akurasi dengan Persamaan (1).

$$\text{Akurasi} = \frac{TP + TN}{P + N} \quad (1)$$

Sementara untuk menghitung tingkat presisi prediksi kejadian dapat digunakan Persamaan (2).

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2)$$

Sensitivitas dapat dihitung dengan menggunakan Persamaan (3).

$$\text{Sensitifitas} = \frac{TP}{TP + FN} \quad (3)$$

HASIL DAN PEMBAHASAN

Hasil Pengumpulan Data

Data yang telah dikumpulkan untuk klasifikasi citra kebakaran dan non kebakaran berjumlah 950 citra. Contoh data citra yang telah dikumpulkan dapat dilihat pada Gambar 3.



Gambar 3. Contoh Data Citra

Hasil Pra-pemrosesan Data

Data citra yang terkumpul memiliki ukuran yang berbeda sehingga dilakukan pra-pemrosesan data agar seluruh citra masukan memiliki ukuran yang sama. Gambar 4(a) merupakan contoh citra asli sebelum

dilakukan perubahan ukuran citra dan Gambar 4(b) adalah citra yang sudah dilakukan perubahan ukuran. Ukuran citra asli yaitu 2800×1866 piksel dan ukuran citra setelah dilakukan perubahan ukuran yaitu 150×150 piksel.



(a)



(b)

Gambar 4. Hasil Perubahan Ukuran Citra

Hasil Pembagian Data

Setelah pra-pemrosesan data citra dilakukan lalu data tersebut dibagi menjadi tiga jenis data dengan jumlah yang berbeda.

Data pelatihan sebanyak 700 citra, data validasi sebanyak 170 citra, dan data pengujian sebanyak 80 citra yang dapat dilihat pada Tabel 1.

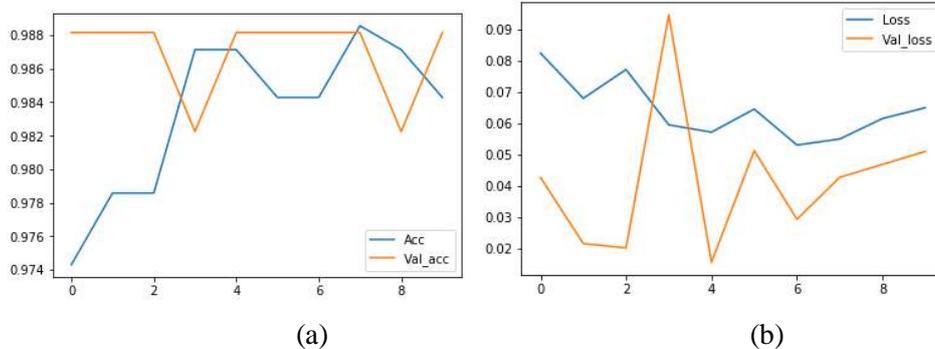
Tabel 1. Jumlah Data Pelatihan, Validasi, dan Pengujian

<i>Dataset</i>	Data Pelatihan	Data Validasi	Data Pengujian	Jumlah
Citra Kebakaran	350	85	40	475
Citra Non Kebakaran	350	85	40	475
Jumlah	700	170	80	950

Hasil Pelatihan dengan Beberapa Epoch

Pada tahap pelatihan dilakukan dengan beberapa jumlah *epoch* yaitu 10, 25, dan 50. Hasil pelatihan dengan *epoch* 10 dan nilai *learning rate* 0.0001 diperoleh nilai *loss* yaitu

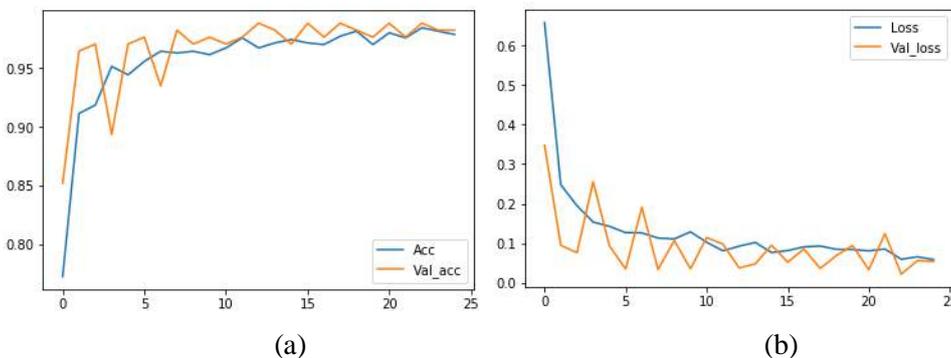
6.4%, nilai akurasi yaitu 98.4%, nilai validasi *loss* yaitu 51%, dan nilai validasi akurasi yaitu 98.8% dengan waktu 5 menit. Grafik hasil pelatihan dengan *epoch* 10 dapat dilihat pada Gambar 5.



Gambar 5. Grafik Hasil Pelatihan dengan Epoch 10

Gambar 5(a) merupakan grafik model akurasi dan Gambar 5(b) merupakan grafik model *loss*. Berdasarkan Gambar 5, akurasi

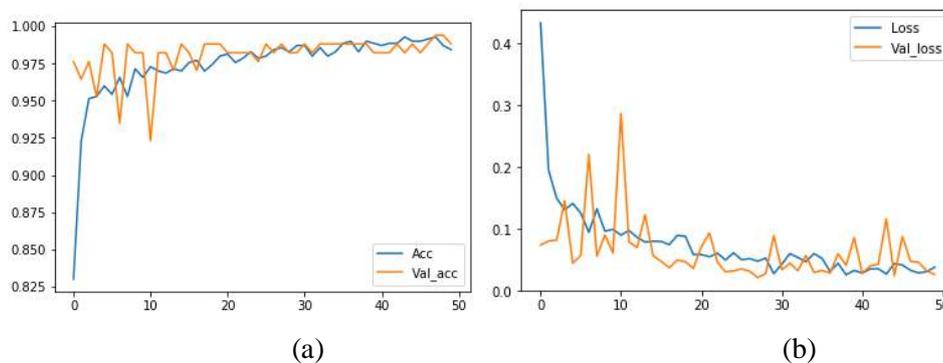
model hasil pelatihan dengan *epoch* 10 belum optimum karena grafik yang dihasilkan tidak konvergen.



Gambar 6. Grafik Hasil Pelatihan dengan Epoch 25

Selanjutnya dilakukan pelatihan dengan *epoch* 25 dan nilai *learning rate* 0.0001. Berdasarkan hasil pelatihan diperoleh nilai *loss* yaitu 7.5%, nilai akurasi yaitu 98.2%, nilai validasi *loss* yaitu 8.4% dan nilai validasi akurasi yaitu 98.8% dengan waktu 18 menit. Gambar 6 menunjukkan grafik hasil pelatihan dengan *epoch* 25. Berdasarkan Gambar 6, akurasi model hasil pelatihan dengan *epoch* 25 sudah optimum karena grafik yang

dihasilkan konvergen. Grafik hasil pelatihan dengan *epoch* 50 dan nilai *learning rate* 0.0001 dapat dilihat pada Gambar 7. Nilai *loss* hasil pelatihan yaitu 3.81%, nilai akurasi yaitu 98.4%, nilai validasi *loss* yaitu 2.6%, dan nilai validasi akurasi yaitu 98.8 dengan waktu 24 menit. Berdasarkan Gambar 7, akurasi model hasil pelatihan dengan *epoch* 50 juga optimum karena grafik yang dihasilkan konvergen.



Gambar 7. Grafik Hasil Pelatihan dengan *Epoch* 50

Tabel 2 merupakan perbandingan nilai *loss*, akurasi, validasi *loss*, dan validasi akurasi dengan beberapa jumlah *epoch* yaitu 10, 25, dan 50. Berdasarkan Tabel 2, *loss* terkecil terjadi pada *epoch* 50 yaitu 3.81%, akurasi tertinggi

terjadi pada saat *epoch* 10 dan 50, validasi *loss* terendah terjadi pada saat *epoch* 50, dan validasi akurasi tertinggi terjadi pada semua *epoch*. Oleh karena itu dapat disimpulkan bahwa pada *epoch* 50 memiliki akurasi yang optimum.

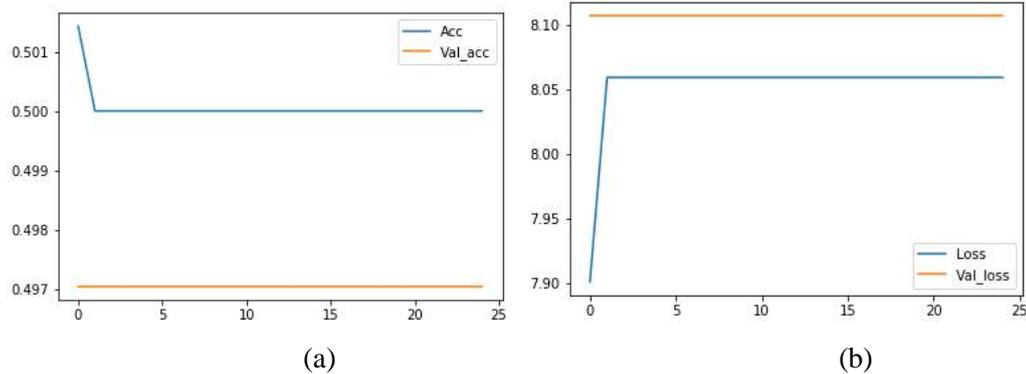
Tabel 2. Perbandingan *Epoch*

	10	25	50
<i>Loss</i>	6.4%	7.5%	3.81%
<i>Accuracy</i>	98.4%	98.2%	98.4%
Validasi <i>Loss</i>	51%	8.4%	2.6%
Validasi <i>Accuracy</i>	98.8%	98.8%	98.8%

Hasil Pelatihan dengan Beberapa *Learning Rate*

Hasil pelatihan dengan nilai *learning rate* 0.01 diperoleh nilai *loss* yaitu 805.9%, nilai akurasi yaitu 50%, nilai validasi *loss* yaitu 810.67%, dan nilai validasi akurasi yaitu

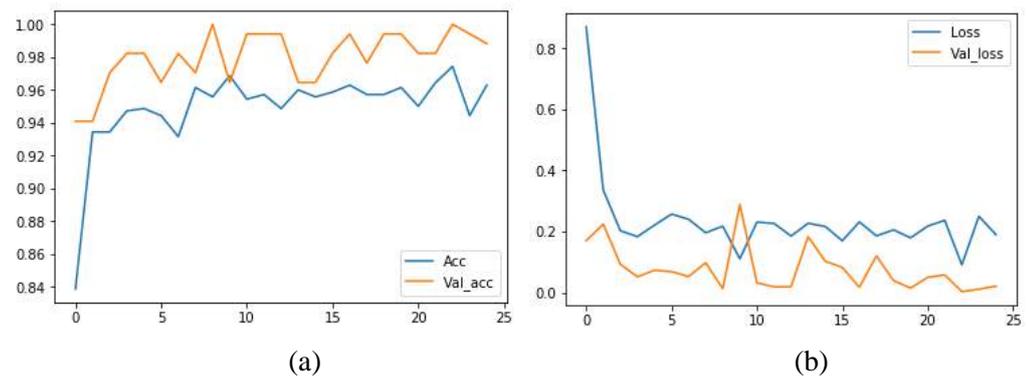
49.7% dengan waktu 10.5 menit. Gambar 8 menunjukkan grafik hasil pelatihan dengan nilai *learning rate* 0.01. Akurasi model hasil pelatihan dengan nilai *learning rate* 0.01 belum optimum karena grafik yang dihasilkan tidak konvergen.



Gambar 8. Grafik Hasil Pelatihan dengan *Learning Rate* 0.01

Hasil pelatihan dengan nilai *learning rate* 0.001 diperoleh nilai *loss* yaitu 1.89%, nilai akurasi yaitu 96.2%, nilai validasi *loss* yaitu 2%, dan nilai validasi akurasi yaitu 98.8%

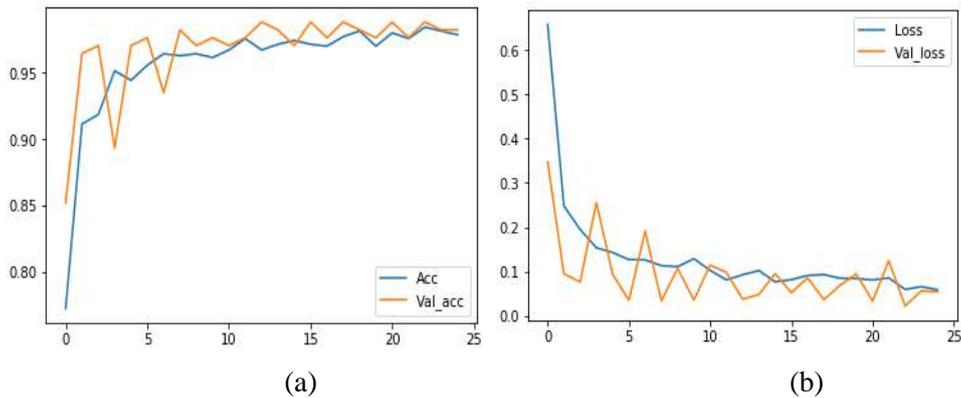
dengan waktu 13.2 menit. Akurasi model hasil pelatihan dengan nilai *learning rate* 0.001 juga belum optimum karena grafik yang dihasilkan tidak konvergen seperti terlihat pada Gambar 9.



Gambar 9. Grafik Hasil Pelatihan dengan *Learning Rate* 0.001

Hasil pelatihan dengan nilai *learning rate* 0.001 diperoleh nilai *loss* yaitu 7.5%, nilai akurasi yaitu 98.2%, nilai validasi *loss* yaitu 8.4%, dan nilai validasi akurasi yaitu 98.8%

dengan waktu 17.6 menit. Akurasi model hasil pelatihan dengan nilai *learning rate* 0.0001 sudah optimum karena grafik yang dihasilkan konvergen seperti terlihat pada Gambar 10.



Gambar 10. Grafik Hasil Pelatihan dengan *Learning Rate* 0.0001

Tabel 3 menunjukkan perbandingan nilai *loss*, akurasi, validasi *loss*, dan validasi akurasi beberapa *learning rate*, yaitu 0.01, 0.001 dan 0.0001. Berdasarkan Tabel 3, *loss* terkecil terjadi pada *learning rate* 0.001 yaitu 1.89%, akurasi tertinggi terjadi pada saat

learning rate 0.0001, validasi *loss* terendah terjadi pada saat *learning rate* 0.02, dan validasi akurasi tertinggi terjadi pada *learning rate* 0.001 dan 0.0001. Oleh karena itu, nilai *learning rate* 0.0001 sudah memberikan akurasi yang optimum dan grafik model yang konvergen.

Tabel 3. Perbandingan *Learning Rate*

	0.01	0.001	0.0001
<i>Loss</i>	805.9%	1.89%	7.5%
<i>Accuracy</i>	50%	96.2%	98.2%
Validasi <i>Loss</i>	810.67%	2%	8.4%
Validasi <i>Accuracy</i>	49.7%	98.8%	98.8%

Hasil Pengujian

Hasil dari proses pengujian akan ditampilkan dalam sebuah *confusion matrix* pada Tabel 4. Berdasarkan Tabel 4, hasil yang didapatkan dari pengujian terhadap model yang telah dilatih cukup baik. Hal ini dapat dilihat dengan prediksi benar yang didapatkan.

Prediksi citra kebakaran diklasifikasikan dengan benar ke dalam kebakaran sebanyak 36 citra dengan kesalahan sebanyak 4 citra yang diklasifikasikan ke dalam kelas non kebakaran. Prediksi citra non kebakaran dengan benar sebanyak 40 citra dan tidak terdapat kesalahan klasifikasi ke dalam kelas kebakaran.

Tabel 4. *Confusion Matrix*

Matriks		Kelas Prediksi	
		Kebakaran	Non Kebakaran
Kelas Sebenarnya	Kebakaran	36	4
	Non Kebakaran	0	40

Berdasarkan Tabel 4 dapat dihitung akurasi, presisi, dan sensitivitas yang didapat setelah melakukan proses pelatihan.

$$\text{Akurasi} = \frac{76}{80} \times 100\% = 90\%$$

$$\text{Presisi} = \frac{36}{36} \times 100\% = 100\%$$

$$\text{Sensitivitas} = \frac{36}{40} \times 100\% = 90\%$$

Berdasarkan Tabel 4, terlihat bahwa akurasi dari data citra non kebakaran lebih baik dibandingkan data citra kebakaran. Ini terjadi dikarenakan citra non kebakaran yang diuji memiliki kondisi citra yang berbeda. Pada citra non kebakaran, citra cenderung memiliki warna yang bervariasi dibandingkan warna citra kebakaran yang didominasi warna merah. Ketika citra memiliki warna merah yang sedikit maka hasilnya akan kurang baik, sehingga tingkat kecocokan model dengan citra yang diuji berkurang. Hal ini yang mempengaruhi hasil dari pengujian terhadap model sehingga tingkat akurasi yang didapatkan lebih kecil.

KESIMPULAN DAN SARAN

Implementasi *Convolutional Neural Network* untuk klasifikasi citra kebakaran dan non kebakaran didapatkan hasil yang baik. CNN yang digunakan terdiri dari 2 lapisan konvolusi, 2 lapisan *pooling* 2×2, 1 lapisan *flatten* layer dan 1 lapisan *dense*. Tingkat akurasi yang diperoleh dari model *Convolutional Neural Network* dengan nilai *learning rate* 0,0001 dan jumlah *epoch* sebanyak 50 mendapatkan akurasi validasi

sebesar 98.8% dan validasi *loss* sebesar 2.6%. Oleh karena itu dapat disimpulkan bahwa implementasi *deep learning* menggunakan CNN mampu melakukan klasifikasi citra kebakaran dan non kebakaran dengan baik. Hasil klasifikasi pada data baru sebanyak 80 citra memiliki akurasi 90%.

Klasifikasi citra kebakaran dan non kebakaran pada penelitian ini belum bersifat *real time*. Pada pengembangan penelitian selanjutnya, proses klasifikasi citra ini dapat bersifat *real time* sehingga dapat diimplementasikan sehingga dapat digunakan oleh umum dalam bentuk aplikasi.

DAFTAR PUSTAKA

- [1] Badan Nasional Penanggulangan Bencana, *Definisi dan jenis bencana*, Jakarta, 2019.
- [2] D. Hardiyanto dan D. A. Sartika, "Identifikasi titik api menggunakan pengolahan citra foto udara," Simposium Nasional RAPI XVII, FT UMS, 2018, hal 51 – 58.
- [3] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, hal. 85 – 117, 2015.
- [4] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," Prosiding Twenty-Second International Joint

- Conference on Artificial Intelligence, 2011, hal. 1237 – 1242.
- [5] A. Krizhevsky, I. Sutskever, dan G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Prosiding Advances in Neural Information Processing System (NIPS)*, 2012, hal. 1097 – 1105.
- [6] C. Zhang, I. Sargent, X. Pan, A. Gardiner, J. Hare, dan P. M. Atkinson, “VPRS-based regional decision fusion of CNN and MRF classifications for very fine resolution remotely sensed images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 8, hal. 1 – 15, 2018.
- [7] F. Hu, G. S. Xia, J. Hu, dan L. Zhang, “Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery,” *Remote Sensing*, vol. 7, no. 11, hal. 14680 – 14707, 2015.
- [8] E. Maggiori, Y. Tarabalka, G. Charpiat, dan P. Alliez, “Convolutional neural networks for large scale remote-sensing image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 5, no. 2, hal. 645 – 657, 2016.
- [9] I. W. E. P. Suartika, A. Y. Wijaya, dan R. Soelaiman, “Klasifikasi citra menggunakan *convolutional neural network* (CNN) pada caltech 101,” *Jurnal Teknik ITS*, vol. 5, no. 1, hal. 65 – 69, 2016.
- [10] M. Setiawan, “Klasifikasi penyakit pada citra daun menggunakan convolutional neural networks,” *Skripsi*, Institut Pertanian Bogor, Bogor, 2018.
- [11] S. C. Pradana, “Implementasi convolutional neural network terhadap instrumen alat musik gamelan menggunakan keras,” *Skripsi*, Universitas Islam Indonesia, Yogyakarta, 2018.
- [12] C. Olston dan M. Najork, “Web crawling,” *Foundations and Trends in Information Retrieval*, vol. 4, no. 3, hal. 176 – 246, 2010.

IMPLEMENTASI RENCANA PERJALANAN WISATA DI KOTA BOGOR MENGGUNAKAN ALGORITMA GREEDY BERBASIS WEBSITE

¹Nur Najmi Sania, ²Ilmiyati Sari

^{1,2}Fakultas Teknologi Industri Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat

¹najmisania@student.gunadarma.ac.id, ²ilmiyati@staff.gunadarma.ac.id

Abstrak

Perencanaan perjalanan wisata merupakan hal penting, terutama pada daerah yang sering menjadi destinasi wisata, salah satunya adalah kota Bogor. Dengan adanya rencana rute perjalanan, maka wisatawan dapat dengan mudah melihat gambaran perjalanan yang diinginkan serta dapat mempersingkat waktu ketika sampai di tempat tujuan. Penelitian ini bertujuan untuk menentukan jarak tempuh terpendek dan jumlah optimal tempat wisata yang dapat dikunjungi di kota Bogor menggunakan algoritma Greedy dengan permasalahan Knapsack. Algoritma Greedy digunakan untuk menentukan rute perjalanan wisata di kota Bogor dengan jarak tempuh terpendek. Permasalahan Knapsack digunakan untuk menentukan jumlah optimal tempat wisata di kota Bogor yang dapat dikunjungi dengan waktu yang dimiliki oleh wisatawan. Algoritma Greedy dengan permasalahan Knapsack tersebut diimplementasikan ke dalam sebuah website. Terdapat 20 tempat wisata di kota Bogor pada website. Objek wisata pada website diklasifikasikan menjadi 3 kategori yaitu wisata alam, tempat rekreasi, dan museum. Pengguna website memasukkan titik awal keberangkatan, lamanya perjalanan, alokasi waktu, pemilihan tempat wisata yang akan dikunjungi dan lamanya waktu kunjungan. Hasil pengujian blackbox menunjukkan bahwa semua fungsi pada website dapat berjalan dengan baik.

Kata kunci: algoritma Greedy, kota Bogor, permasalahan Knapsack, rencana perjalanan wisata, website

Abstract

Planning a tour is important, especially in areas that are often tourist destinations, one of which is Bogor. With the planned itinerary, tourists can easily see the desired trip description and can shorten the time when they arrive at their destination. This study aims to determine the shortest distance and the optimal number of tourist attractions that can be visited in Bogor using the Greedy algorithm with the Knapsack problem. Greedy algorithm is used to determine the route of tourist travel in Bogor with the shortest distance traveled. The Knapsack problem is used to determine the optimal number of tourist attractions in Bogor that can be visited with the time owned by tourists. Greedy's algorithm with the Knapsack problem is implemented on a website. There are 20 tourist attractions in Bogor on the website. Attractions on the website are classified into 3 categories namely natural attractions, recreation areas, and museums. Website users enter the starting point of departure, the length of the trip, the allocation of time desired, the selection of tourist attractions to be visited and the length of time of the visit. Blackbox testing results show that all functions on the website can work well.

Keywords: Bogor city, greedy algorithms, knapsack problem, travel plans, website

PENDAHULUAN

Pariwisata adalah suatu perjalanan yang dilakukan orang untuk sementara waktu, yang diselenggarakan dari suatu tempat ke tempat lain meninggalkan tempatnya semula, dengan suatu perencanaan dan dengan maksud bukan untuk berusaha atau mencari nafkah di tempat yang dikunjungi, tetapi semata-mata untuk menikmati kegiatan pertamasyaan dan rekreasi atau untuk memenuhi keinginan yang beraneka ragam [1]. Perencanaan perjalanan sebelum pergi berpariwisata merupakan sebuah hal yang penting. Dengan adanya rencana perjalanan, maka wisatawan dapat dengan mudah melihat gambaran perjalanan yang diinginkan serta dapat mempersingkat waktu ketika sampai di tempat tujuan dan tidak perlu memikirkan rencana tempat-tempat yang akan dikunjungi.

Algoritma Greedy menjadi salah satu alternatif untuk membantu proses pencarian waktu tempuh terpendek dan jumlah optimal tempat wisata yang dapat dikunjungi dari waktu yang dimiliki oleh wisatawan. Algoritma Greedy adalah algoritma yang dapat memecahkan masalah langkah demi langkah dan merupakan salah satu metode yang digunakan dalam masalah optimasi [2]. Algoritma Greedy memiliki prinsip utama yaitu mengambil sebanyak mungkin apa yang dapat diperoleh saat ini [3]. Banyaknya tempat wisata yang dapat dikunjungi dari waktu yang dimiliki oleh wisatawan merupakan salah satu permasalahan *Knapsack*.

Permasalahan *Knapsack* merupakan salah satu permasalahan yang dapat diselesaikan dengan algoritma Greedy. Waktu yang dibutuhkan untuk menyelesaikan permasalahan *Knapsack* menggunakan algoritma Greedy yang diimplementasikan ke dalam perangkat lunak menjadi lebih singkat dibandingkan melakukan perhitungan secara manual dan algoritma *Brute Force* [4].

Bogor merupakan kota yang sering dijadikan tujuan oleh beberapa kota di sekitarnya seperti Jakarta, Depok, Tangerang dan Bekasi. Menurut data Dinas Kebudayaan dan Pariwisata kota Bogor, rata-rata orang yang berwisata di kota Bogor sekitar 3 hingga 5 juta wisatawan pada tahun 2013 hingga 2016 [5]. Bahkan kota Bogor selalu dikunjungi warga Belanda yang tinggal di Batavia sebelum kemerdekaan karena cuaca yang sejuk dan nyaman. Selain memiliki cuaca yang sejuk, Bogor juga memiliki beberapa tempat wisata yang dapat dikunjungi antara lain wisata alam, tempat rekreasi, dan museum.

Beberapa penelitian mengenai perencanaan wisata telah dilakukan. Penelitian yang dilakukan oleh Qomaruddin, Alawy dan Sugiono mengenai aplikasi penentuan rute perjalanan wisata di kabupaten Jember menggunakan algoritma Dijkstra. Hasil uji coba menunjukkan bahwa aplikasi menghasilkan waktu yang lebih cepat 4 menit dibandingkan dengan perhitungan secara manual [6]. Penelitian mengenai pencarian jalur tercepat rute perjalanan wisata di kota Malang menggunakan algoritma *Tabu Search*

telah dilakukan oleh Varita, Setyawati, dan Rahadi. Pada penelitian tersebut, algoritma *Tabu Search* dapat menentukan rute tercepat yang optimal dengan biaya terendah menggunakan parameter panjang, volume, dan kepadatan jalan. Kekurangan algoritma *Tabu Search* yaitu parameter yang digunakan dalam algoritma *Tabu Search* tidak benar-benar saling bebas sehingga dapat mempengaruhi parameter lain [7]. Pada penelitian yang dilakukan oleh Debby, Beng, dan Wasino membahas mengenai *website* rencana perjalanan wisata di Boloyali, Solo, dan Klaten. Pada penelitian tersebut menggunakan algoritma Greedy untuk menentukan obyek wisata yang akan menjadi rekomendasi dikunjungi setiap hari oleh wisatawan. Wisatawan dapat membuat rencana perjalanan wisata sendiri untuk daerah Boyolali, Solo, dan Klaten menggunakan *website* [8]. Penelitian lain mengenai penggunaan algoritma Greedy untuk perencanaan wisata telah dilakukan oleh Ghozali, Setiawan, dan Furqon. Pada penelitian tersebut menghasilkan aplikasi perencanaan wisata di Malang Raya dengan algoritma Greedy. Dengan menggunakan algoritma Greedy dapat dihasilkan solusi optimum dalam pencarian rute terpendek yang dapat membantu merumuskan hasil rekomendasi rute wisata di Malang Raya [9].

Oleh karena itu, pada penelitian ini dibuat suatu sistem berbasis *website* yang dapat memberikan informasi tempat wisata di kota Bogor serta membuat rencana

perjalanan menggunakan algoritma Greedy dengan permasalahan *Knapsack*. *Website* dapat membantu wisatawan yang berkunjung ke daerah kota Bogor mengetahui tempat wisata yang akan dikunjungi selama berada di kota Bogor. Dalam penelitian ini permasalahan *Knapsack* digunakan untuk memperoleh jumlah tempat wisata yang dapat dikunjungi di kota Bogor sesuai dengan waktu yang dimiliki oleh wisatawan yang paling optimal. Algoritma Greedy digunakan untuk menentukan rute perjalanan dengan waktu perjalanan yang paling optimal.

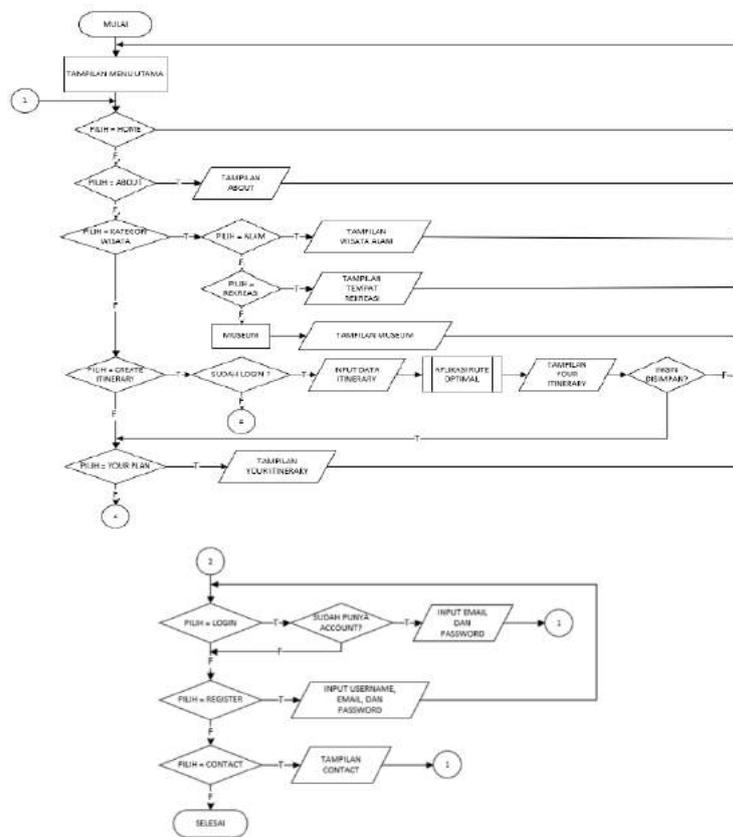
METODE PENELITIAN

Website yang dibuat memberikan informasi mengenai 20 tempat wisata di kota Bogor dan rute perjalanan wisata untuk tempat wisata yang ingin dikunjungi oleh pengguna. Objek wisata pada *website* diklasifikasikan menjadi 3 kategori yaitu wisata alam, tempat rekreasi, dan museum. Kategori wisata alam memuat 7 tempat wisata yaitu Curug Bidadari Sentul Paradise Bogor, Taman Wisata Alam Gunung Pancar, Curug Leuwi Lieuk Sentul, Telaga Warna, Curug Leuwi Hejo, Gunung Batu Jonggol, dan Taman Nasional Gunung Gede Pangrango. Selanjutnya, kategori tempat rekreasi memuat 7 tempat wisata yaitu Kebun Raya Bogor, Kuntum *Farmfield*, *The Jungle Waterpark* Bogor, *Devoyage* Bogor, Taman Safari Indonesia, *Little Venice* Puncak, dan Taman Sakura. Lalu, kategori museum memuat 6

tempat yaitu Museum Zoologi Bogor, Museum Nasional Sejarah Alam Indonesia, Museum Tanah dan Pertanian, Museum Perjuangan Bogor Museum dan Monumen PETA, dan Lulu *Trick Eye Museum*.

Flowchart *website* rencana perjalanan wisata di kota Bogor dapat dilihat pada Gambar 1. Berdasarkan Gambar 1, pada tampilan awal *website* terdapat menu utama.

Selanjutnya pada menu utama terdapat menu *Home, About, Kategori Wisata, Create Itinerary, Your Plan, Login, Register, dan Contact*. Pada menu Kategori Wisata terdapat tiga pilihan menu yaitu alam, rekreasi, dan museum. Pada menu *Create Itinerary*, pengguna harus memiliki akun sehingga membuat rencana perjalanan sesuai yang diinginkan.



Gambar 1. *Flowchart Website*

Pengguna memasukkan titik awal keberangkatan, lamanya perjalanan (berupa tanggal), alokasi waktu yang diinginkan, pemilihan tempat wisata yang akan dikunjungi dan lamanya waktu kunjungan (dalam menit) pada *website*. Langkah-langkah

untuk menghasilkan rute perjalanan wisata dalam penelitian ini sebagai berikut:

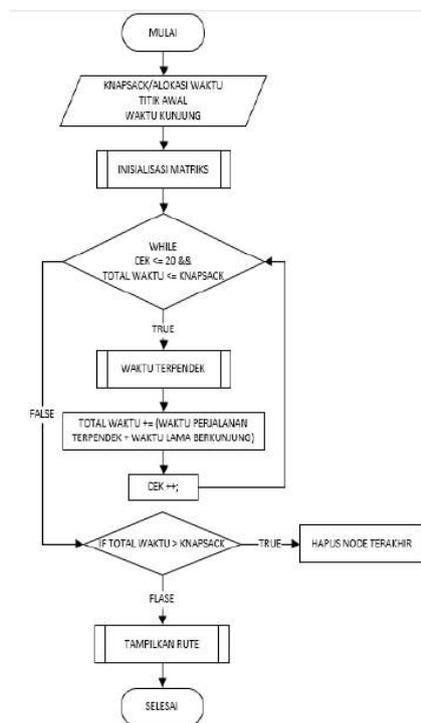
1. Merepresentasikan masalah perjalanan wisata kedalam bentuk graf. Tempat wisata yang dipilih direpresentasikan sebagai simpul dengan bobotnya adalah waktu

kunjung, serta jalan yang menghubungkan setiap tempat wisata adalah sisi yang bobotnya adalah waktu tempuh.

2. Menggunakan algoritma Greedy pada graf yang dihasilkan untuk mencari waktu tempuh terpendek atau rute optimal.
3. Menggunakan permasalahan *Knapsack* untuk mengoptimalkan jumlah tempat wisata yang dapat dikunjungi.

Gambar 2 menjelaskan bagaimana alur *website* untuk menampilkan rute yang paling

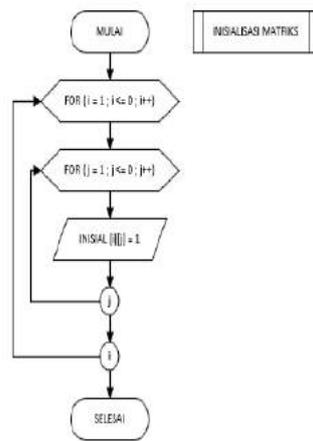
optimal. *Knapsack* pada penelitian ini yaitu total alokasi waktu wisata yang digunakan oleh wisatawan. Perhitungan dimulai dari titik awal keberangkatan yang dimasukkan oleh pengguna. Selanjutnya dilakukan inisialisasi matriks untuk mengidentifikasi apakah nilai dari titik tersebut dihitung atau tidak. Jika total waktu wisata lebih dari *knapsack* maka rute terakhir dihapus. Jika total waktu wisata tidak lebih dari *knapsack* maka ditampilkan rute yang optimal.



Gambar 2. *Flowchart* Rute Optimal

Flowchart inisialisasi matriks dapat dilihat pada Gambar 3. Semua nilai pada matriks diinisialisasi dengan nilai 1 dengan ukuran matriks yaitu 20×20 karena terdapat 20 tempat wisata yang digunakan dalam

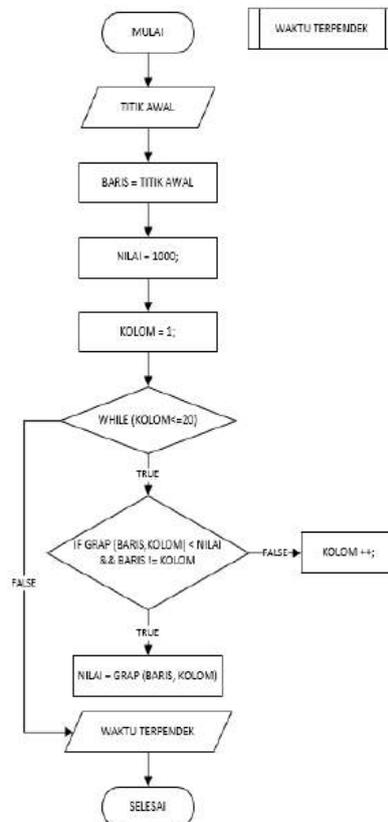
perhitungan. Jika nilai pada matriks telah dihitung maka nilai matriks diubah menjadi 0. Pada perhitungan selanjutnya nilai matriks yang telah diubah menjadi 0 tidak akan dihitung lagi.



Gambar 3. *Flowchart* Inisialisasi Matriks

Flowchart perhitungan waktu terpendek dapat dilihat pada Gambar 3. Titik awal keberangkatan yang telah dimasukkan oleh pengguna digunakan untuk memulai baris pada matriks. Inisialisasi nilai pada

penelitian ini yaitu 1000. Nilai terkecil dari kolom pada baris matriks akan menjadi rute selanjutnya. Selanjutnya dihitung pada sistem sampai alokasi waktu (*knapsack*) tidak mencukupi.



Gambar 4. *Flowchart* Perhitungan Waktu Terpendek

HASIL DAN PEMBAHASAN

Nama 20 tempat wisata yang berada di kota Bogor dinotasikan dengan huruf A sampai T yang dapat dilihat pada Tabel 1. Data waktu tempuh antar 20 tempat wisata di kota Bogor dapat dilihat pada Tabel 2.

Tabel 1. Notasi 20 Tempat Wisata di Kota Bogor

Node	Nama Tempat Wisata
A	Kebun Raya Bogor
B	Kuntum <i>Farmfield</i>
C	<i>The Jungle Waterpark</i> Bogor
D	<i>Devoyage</i> Bogor
E	Taman Safari Indonesia
F	Museum Zoologi Bogor
G	Museum Nasional Sejarah Alam Indonesia
H	Museum Tanah dan Pertanian
I	Museum Perjuangan Bogor
J	Museum dan Monumen PETA
K	Curug Bidadari Sentul Paradise Bogor
L	Taman Wisata Alam Gunung Pancar
M	Leuwi Lieuk Sentul
N	Telaga Warna
O	Curug Leuwi Hejo
P	Gunung Batu Jonggol
Q	Taman Nasional Gunung Gede Pangrango
R	<i>Little Venice</i> Puncak
S	Taman Sakura, Kebun Raya Cibodas
T	Lulu <i>Trick Eye Museum</i>

Tabel 2. Data Waktu Tempuh Antar 20 Tempat Wisata di Kota Bogor

X	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	0	24	15	18	51	1	2	16	12	6	50	37	85	67	61	98	82	83	87	33
B	20	0	27	30	42	21	22	26	31	26	53	40	88	57	64	102	79	80	84	30
C	23	27	0	2	58	23	17	29	26	21	58	47	94	75	70	105	89	90	94	40
D	25	30	3	0	59	27	19	31	29	22	61	50	97	77	73	107	92	93	97	43
E	66	63	82	84	0	67	68	72	78	72	84	82	128	23	109	117	45	47	50	17
F	23	26	14	17	57	0	11	22	8	6	49	35	83	63	59	96	81	82	86	33
G	12	27	24	27	55	12	0	11	7	5	48	34	82	62	58	95	79	80	84	31
H	4	27	16	18	61	22	10	0	10	8	53	39	87	67	62	97	81	82	86	33
I	16	32	28	30	61	17	6	25	0	8	54	40	87	68	63	97	81	82	86	32
J	13	28	29	31	56	13	15	21	23	0	47	33	81	62	56	97	79	80	84	31
K	45	53	58	60	67	45	47	52	52	44	0	33	81	73	57	98	94	95	99	45
L	33	44	49	51	68	34	35	41	42	33	34	0	55	75	31	68	93	94	98	44
M	81	93	96	99	119	82	83	88	88	82	83	55	0	127	106	58	130	123	135	93
N	53	53	67	70	20	54	55	60	62	60	76	68	117	0	93	98	27	29	32	25
O	57	69	71	74	95	57	59	64	63	57	58	33	113	101	0	47	116	112	121	69
P	92	102	105	106	114	93	94	97	97	93	99	68	58	96	48	0	92	84	97	106
Q	74	73	86	88	47	74	75	79	81	78	98	91	135	27	118	96	0	26	15	51
R	75	74	87	89	48	76	77	80	83	80	99	91	125	28	116	87	25	0	30	53
S	80	79	91	94	53	80	81	84	87	84	103	96	140	32	124	102	18	31	0	56
T	25	25	37	40	17	25	26	30	35	30	49	41	90	24	67	104	51	53	55	0

Hasil Perhitungan Dengan Algoritma Greedy

Berikut ini merupakan contoh perhitungan dengan algoritma Greedy. Misalkan total alokasi yang dimiliki adalah 8 jam (480 menit), perjalanan dilakukan selama dua hari dan perjalanan dimulai dari titik Kebun Raya Bogor (A). Lokasi dan alokasi waktu kunjungan masing-masing tempat wisata yang akan dikunjungi sebagai berikut:

1. *Kuntum Farmfield* (B): 60 menit

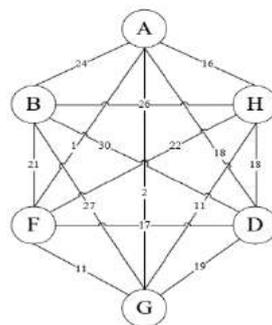
2. *Devoyage* Bogor (D): 200 menit

3. Museum Zoologi Bogor (F): 120 menit

4. Museum Nasional Sejarah Alam Indonesia (G): 120 menit

5. Museum Tanah dan Pertanian (H): 180 menit

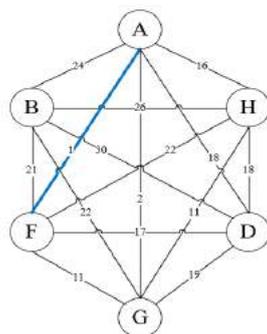
Lintasan graf dari node A, B, D, F, G, dan H dan bobot setiap sisi yang menyatakan waktu perjalanan dari setiap node satu ke node lain seperti ditunjukkan pada Gambar 5.



Gambar 5. Tampilan Graf dari Kasus Perhitungan

Perhitungan pertama yaitu mencari lintasan yang memiliki nilai bobot minimum dari masing-masing sisi. Lintasan pertama memilih sisi yang berdekatan dengan node A, yaitu sisi AB dengan bobot 24, sisi AH

dengan bobot 16, sisi AF dengan bobot 1, sisi AG dengan bobot 2 dan sisi AD dengan bobot 18. Sisi yang dipilih yaitu sisi AF karena mempunyai bobot lebih kecil dari yang lain yaitu 1 seperti ditunjukkan pada Gambar 6.

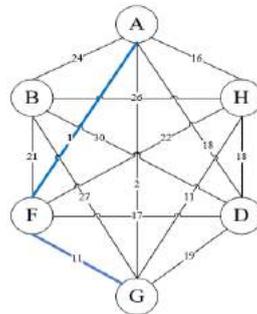


Gambar 6. Graf Rute dari Node A ke Node F

Selanjutnya memilih sisi yang berdekatan dengan sisi AF, yaitu sisi FB dengan bobot 21 maka bobot totalnya 22.

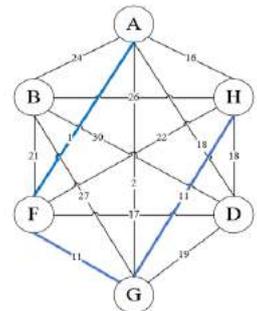
Lalu sisi FH dengan bobot 22 maka bobot totalnya 23. Selanjutnya sisi FD dengan bobot 17 maka bobot totalnya 18. Sisi FG dengan

bobot 11 maka bobot totalnya 12. Sisi yang bobot terkecil seperti ditunjukkan pada dipilih yaitu sisi FG karena mempunyai Gambar 7.



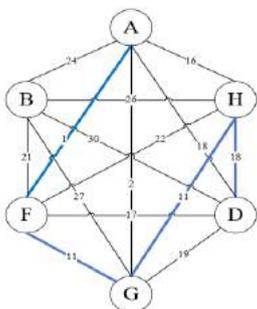
Gambar 7. Graf Rute dari Node F ke Node G

Selanjutnya memilih sisi yang totalnya 38. Sisi GD dengan bobot 19 maka berdekatan dengan sisi FG, yaitu sisi GB bobot totalnya 46. Sisi yang dipilih yaitu sisi dengan bobot 27 maka bobot totalnya 39. GH karena mempunyai bobot terkecil seperti Lalu sisi GH dengan bobot 11 maka bobot ditunjukkan pada Gambar 8.



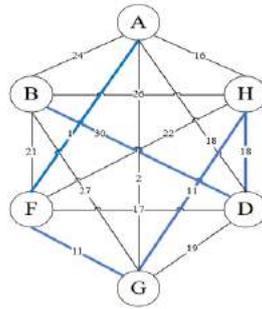
Gambar 8. Graf Rute dari Node G ke Node H

Selanjutnya memilih sisi yang berdekatan dengan bobot 18 maka bobot totalnya 54. Sisi dengan sisi GH, yaitu sisi HB dengan bobot yang dipilih yaitu sisi HD karena mempunyai 26 maka bobot totalnya 64 dan sisi HD bobot terkecil seperti ditunjukkan pada Gambar 9.



Gambar 9. Graf rute dari node H ke node D

Selanjutnya memilih sisi yang dengan bobot 30 dengan bobot totalnya 84 berdekatan dengan sisi HD, yaitu sisi DB menit seperti ditunjukkan pada Gambar 10.



Gambar 10. Graf Rute dari Node D ke Node B

Rute yang optimal dengan waktu tempuh terpendek adalah A – F – G – H – D – B dengan waktu 84 menit, yaitu Kebun Raya Bogor – Museum Zoologi Bogor – Museum Nasional Sejarah Alam Indonesia – Museum Tanah dan Pertanian – *Devoyage* Bogor – Kuntum *Farmfield*.

Hasil dari rute tersebut belum termasuk perhitungan waktu perjalanan dengan waktu kunjung. Perhitungan kedua yaitu menjumlahkan waktu tempuh dengan waktu kunjung wisata untuk mendapatkan total alokasi waktu dari masing-masing tempat wisata yang dipilih. Perhitungan dari kasus yang di atas sebagai berikut:

- A - F = 1 + 120 = 121 menit
- F - G = 11 + 120 = 131 menit
- G - H = 11 + 180 = 191 menit
- H - D = 18 + 200 = 218 menit
- D - B = 30 + 60 = 90 menit

Total waktu kunjung dan perjalanan adalah 750 menit atau 12 jam 30 menit sedangkan total alokasi waktu yang dimiliki dalam satu hari hanya 480 menit atau 8 jam. Oleh karena itu, untuk mendapatkan rute perjalanan wisata optimal dibutuhkan

perhitungan yang ketiga yaitu mencari titik terdekat dari titik awal. Titik yang memiliki nilai terkecil maka akan dipilih sebagai rute selanjutnya dan menjadi titik awal untuk perhitungan selanjutnya. Apabila sisa waktu yang dimiliki tidak mencukupi untuk perhitungan selanjutnya, maka perhitungan selesai. Sisa tempat wisata yang tidak terhitung akan ditempatkan di hari berikutnya dengan perhitungan yang sama. Perhitungan dari kasus yang di atas sebagai berikut:

- A - B = 24 menit
- A - D = 18 menit
- A - F = 1 menit
- A - G = 2 menit
- A - H = 16 menit

Berdasarkan perhitungan di atas maka A – F adalah nilai terkecil dan F titik yang dipilih. Total waktu yang dimiliki yaitu waktu tempuh + waktu kunjung = 1 + 120 = 121 menit. Sisa waktu yang dimiliki menjadi 480 – 121 = 359 menit. Kemudian perhitungan dilanjutkan dari titik F.

- F - B = 21 menit
- F - D = 17 menit
- F - G = 11 menit

F - H = 22 menit

Berdasarkan perhitungan di atas maka F - G adalah nilai terkecil dan G titik yang dipilih. Total waktu yang dimiliki menjadi waktu tempuh + waktu kunjung = 11 + 120 = 131 menit. Sisa waktu yang dimiliki menjadi $359 - 131 = 228$ menit. Kemudian perhitungan dilanjutkan dari titik G.

G - B = 22 menit

G - D = 19 menit

G - H = 11 menit

Berdasarkan perhitungan di atas maka G - H adalah nilai terkecil dan H titik yang dipilih. Total waktu yang dimiliki menjadi waktu tempuh + waktu kunjung = 11 + 180 = 191 menit. Sisa waktu yang dimiliki menjadi $228 - 191 = 37$ menit. Kemudian perhitungan dilanjutkan dari titik H.

H - B = 22 menit

H - D = 19 menit

Berdasarkan perhitungan di atas maka H - D adalah nilai terkecil dan D titik yang dipilih. Total waktu yang dimiliki menjadi

waktu tempuh + waktu kunjung = 18 + 200 = 218 menit. Waktu yang tersisa hanya 37 menit jadi tidak mencukupi, maka perhitungan berhenti di titik H. Jadi rute perjalanan optimal yang memenuhi dengan perhitungan ketiga adalah:

1. F: Museum Zoologi Bogor = 120 menit
2. G: Museum Nasional Sejarah Alam Indonesia = 120 menit
3. H: Museum Tanah dan Pertanian = 180 menit

Dengan total waktu kunjung dan perjalanan adalah 443 menit atau 7 jam 23 menit. Setelah mendapatkan rute perjalanan optimal, langkah berikutnya yaitu mencari hasil perjalanan sebenarnya dengan perhitungan keempat yaitu menjumlahkan waktu perjalanan (waktu tempuh) dengan waktu kedatangan (waktu mulai perjalanan), serta menjumlahkan waktu kedatangan (waktu mulai perjalanan) dengan waktu kunjung. Hasil rencana perjalanan wisata pada hari pertama dapat dilihat pada Tabel 3.

Tabel 3. Hasil Rencana Perjalanan Wisata Hari Pertama

Lokasi	Waktu Tempuh (dalam menit)	Waktu Kedatangan	Waktu Kunjung (dalam menit)	Waktu Selesai
A - F	1	09.01	120	11:01
F - G	11	11.12	120	13:12
G - H	11	13.23	180	16:23

Berdasarkan Tabel 3 waktu kedatangan dimulai dari jam 09:00 - 17:00. Terdapat 3 tempat wisata saja yang dapat dikunjungi di hari pertama, yaitu Museum Zoologi Bogor, Museum Nasional Sejarah Alam Indonesia,

dan Museum Tanah dan Pertanian. Perjalanan selanjutnya dialihkan ke hari kedua dengan waktu kedatangan (waktu mulai perjalanan) yang sama yaitu jam 09:00 - 17:00 seperti ditunjukkan pada Tabel 4.

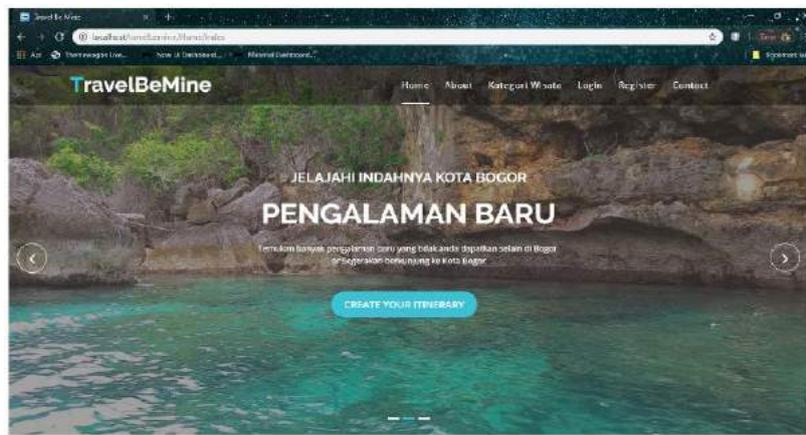
Tabel 4. Hasil Rencana Perjalanan Wisata Hari Kedua

Lokasi	Waktu Tempuh (dalam menit)	Waktu Kedatangan	Waktu Kunjung (dalam menit)	Waktu Selesai
H – D	18	09.18	200	12:38
D – G	30	13.08	60	14:08

Tampilan Website

Pada halaman *Home* terdapat menu *Home*, *About*, *Kategori Wisata*, *Login*, *Register* dan *Contact*. Selanjutnya dibawah

menu-menu tersebut terdapat *slide image* serta tombol untuk *Create Itinerary*. Tampilan halaman *Home* dapat dilihat pada Gambar 11.



Gambar 11. Tampilan Halaman *Home*

Pada halaman *About* terdapat *header* yang sama seperti halaman *Home*. Halaman *About* menjelaskan kegunaan *website*

perencanaan wisata kota Bogor kepada pengguna. Tampilan halaman *about* dapat dilihat pada Gambar 12.



Gambar 12. Tampilan Halaman *About*

Pada halaman *Kategori Wisata* terdapat 4 tombol yaitu *All*, *Alam*, *Rekreasi*, dan

Museum. Selanjutnya di bawah tombol tersebut terdapat beberapa gambar

objek wisata sesuai kategori yang dipilih. dilihat pada Gambar 13. Tampilan halaman Kategori Wisata dapat



Gambar 13. Tampilan Halaman Kategori Wisata

Pada halaman *Login* terdapat logo yang akan mengarahkan kembali ke halaman *Home*. Pada halaman *Login*, pengguna dapat memasukkan akun yang telah terdaftar pada *website*. Jika pengguna belum memiliki akun maka akan diarahkan ke halaman Registrasi. Tampilan halaman *Login* dapat dilihat pada Gambar 14.



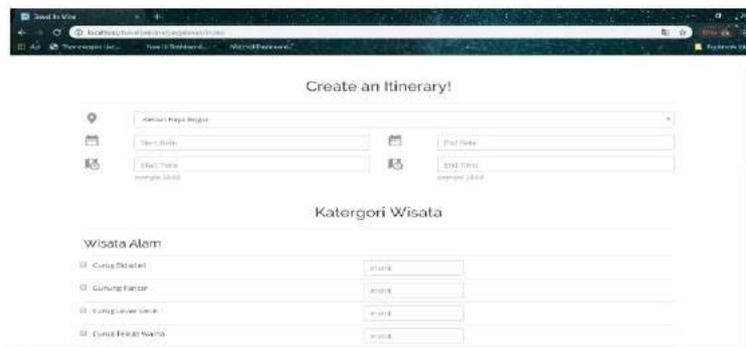
Gambar 14. Tampilan Halaman *Login*

Pada halaman Registrasi, pengguna yang belum memiliki akun dapat melakukan registrasi akun dengan memasukkan *username*, *e-mail*, dan *password*. Tampilan halaman registrasi dapat dilihat pada Gambar 15.



Gambar 15. Tampilan Halaman Registrasi

Tampilan halaman *Create Itinerary* serta berakhirnya perjalanan, pemilihan dapat dilihat pada Gambar 16. Pada halaman *Create Itinerary*, pengguna memasukkan titik awal keberangkatan, waktu dan tanggal mulai tempat wisata yang akan dikunjungi dan lamanya waktu kunjungan di setiap tempat wisata (dalam menit).



Gambar 16. Tampilan Halaman *Create Itinerary*

Pada halaman *Your Plan* menampilkan *history* perjalanan yang telah dilakukan oleh pengguna. Tampilan *history* perjalanan yang akan dilakukan dan halaman *Your Plan* seperti pada Gambar 17.



Gambar 17. Tampilan Halaman *Your Plan*

Pada halaman *Your Itinerary* terdapat informasi rute perjalanan wisata sesuai dengan objek wisata, durasi wisata, dan tanggal yang telah dimasukkan oleh pengguna.

Rencana perjalanan yang ditampilkan merupakan rencana perjalanan yang optimal dengan jarak tempuh terpendek. Tampilan halaman *Your Itinerary* dapat dilihat pada Gambar 18.

Your Itinerary!

Hari 1

Lokasi	Jam Operasional	Durasi Perjalanan (Dalam Menit)	Waktu Kedatangan	Durasi Wisata (Dalam menit)	Waktu Selesai
Museum Zoologi	08:00 - 17:00	1	09:01	120	11:01
Museum Nasional Sejarah Alam Indonesia	08:00 - 16:00	11	11:12	135	13:12
Museum Tanah dan Pertanian	08:00 - 10:00	11	13:23	180	15:23

Hari 2

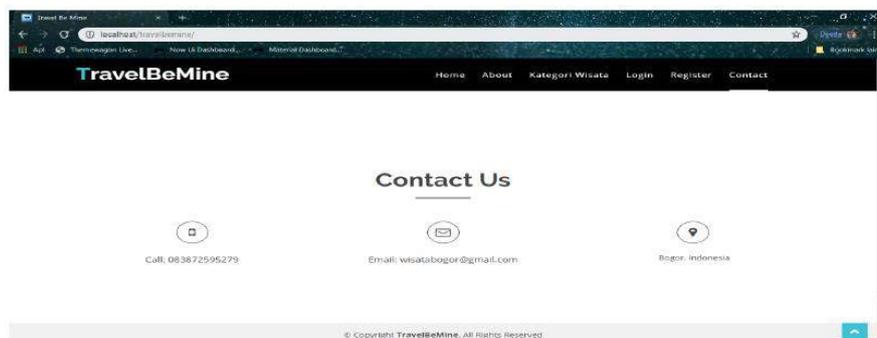
Lokasi	Jam Operasional	Durasi Perjalanan (Dalam Menit)	Waktu Kedatangan	Durasi Wisata (Dalam menit)	Waktu Selesai
Kantun Fairfield	08:00 - 18:00	27	09:27	60	10:27
Devoyage Diger	10:00 - 19:00	30	10:57	200	14:17

[SAMBUN PERJALANAN](#)

Gambar 18. Tampilan Halaman *Your Itinerary*

Pada halaman *Contact* terdapat informasi kontak pemilik *website*. Tampilan

halaman *Contact* dapat dilihat pada Gambar 19.



Gambar 19. Tampilan Halaman *Contact*

Hasil uji coba *website* menggunakan *blackbox testing* pada beberapa *browser* dapat dilihat pada Tabel 5. Berdasarkan Tabel 5, semua fungsi pada *website* dapat berjalan dengan baik pada Google Chrome, Opera, dan

Internet Explorer. Pada *browser* Internet Browser Smartphone, fungsi tidak berjalan dengan baik pada tombol *Create Itinerary*. Fungsi juga tidak berjalan baik pada menu *Create Itinerary* menggunakan browser Mozila.

Tabel 5. Hasil *Blackbox Testing*

No.	Fungsi	Internet Browser Smartphone	Mozila	Google Chrome	Opera	Internet Explorer
1.	Memilih Menu <i>Home</i>	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
2.	Memilih Menu <i>About</i>	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
3.	Memilih Menu Kategori Wisata	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
4.	Memilih Menu <i>Login</i>	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
5.	Memilih Menu <i>Register</i>	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
6.	Menekan Tombol <i>Create Itinerary</i>	Fungsi tidak berjalan dengan baik karena tombol <i>Create Itinerary</i> tidak muncul	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
7.	Memilih Menu <i>Create Itinerary</i>	Fungsi berjalan dengan baik	Fungsi tidak berjalan dengan baik karena <i>checkbox</i> tidak muncul pada kategori wisata	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
8.	Memilih Menu <i>Your Plan</i>	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
9.	Menekan Tombol Lihat Rute	Fungsi berjalan dengan baik tetapi tampilan sedikit tidak rapi	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
10.	Menekan Tombol <i>Delete</i>	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik
11.	Memilih Menu <i>Contact</i>	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik	Fungsi berjalan dengan baik

12.	Memilih Menu Logout	Fungsi berjalan dengan baik				
-----	---------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

KESIMPULAN DAN SARAN

Pada penelitian ini telah dibuat suatu *website* rencana perjalanan wisata di kota Bogor menggunakan algoritma Greedy dengan permasalahan *Knapsack*. *Website* memberikan informasi mengenai 20 tempat wisata di Kota Bogor. Pengguna *website* dapat menentukan rute perjalanan wisata di kota Bogor dengan waktu tempuh yang paling optimal. Hasil uji coba *website* menggunakan pengujian *blackbox* menunjukkan bahwa semua fungsi pada *website* dapat berjalan dengan baik.

Rute yang dihasilkan pada *website* hanya mempertimbangkan waktu pada saat tertentu tidak mempertimbangkan waktu yang riil. Pada penelitian selanjutnya dapat dibuat rute dengan menggunakan waktu yang riil. Selain itu, jumlah tempat wisata dapat ditambah dalam pilihan yang dimasukkan pengguna sehingga lebih banyak alternatif tempat wisata yang dapat dijangkau oleh wisatawan di kota Bogor.

DAFTAR PUSTAKA

- [1] R. Sihite, *Tourism industry (kepariwisataan)*. Surabaya: SIC, 2000.
- [2] Y. Kurniasari, *Penerapan algoritma Greedy*. Jakarta: PT. Gramedia Pustaka Utama, 2006.
- [3] R. Munir, "Strategi algoritmik," Sekolah Teknik Informatika dan Elektro, Institut Teknologi Bandung, Tech. Report, 2005.
- [4] D. Rachmawati dan A. Chandra, "Implementasi algoritma Greedy untuk menyelesaikan masalah knapsack problem," *Jurnal Ilmiah Sains dan Komputer (SAINTIKOM)*, vol. 12, no. 3, hal. 185–192, 2013.
- [5] B. A. Cahyono, *Kota Bogor dalam angka Bogor city in figures*. Bogor: BPS Kota Bogor, 2018.
- [6] M. Qomaruddin, M. T. Alawy, dan Sugiono, "Perancangan aplikasi penentu rute terpendek perjalanan wisata di kabupaten Jember menggunakan algoritma Dijkstra," *Science Electro*, vol. 6, no. 2, hal. 31 – 39, 2017.
- [7] I. Varita, O. Setyawati, dan D. Rahadi, "Pencarian jalur tercepat rute perjalanan wisata dengan algoritma Tabu Search," *Jurnal EECCIS*, vol. 7, no. 2, hal. 185 – 190, 2013.
- [8] B. Debby, J. T. Beng, dan Wasino, "Itinerary wisata Boyolali-Solo-Klaten," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 6, no. 1, hal. 22 – 27, 2018.
- [9] A. E. Ghozali, B. D. Setiawan, dan M. T. Furqon, "Aplikasi perencanaan wisata di Malang Raya dengan algoritma Greedy," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vo. 1, no. 12, hal. 1459–1467, 2017.

PERBANDINGAN DETEKSI TEPI SOBEL DAN ROBERT UNTUK PENDETEKSIAN KESAMAAN CITRA BERDASARKAN WARNA

¹M. Ridwan Dwi Septian, ²Febriani, ³A. Ramadona Nilawati

¹Fakultas Teknologi Industri Universitas Gunadarma, ^{2,3}Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat

¹ridwandwiseptian@staff.gunadarma.ac.id, ²febriani@staff.gunadarma.ac.id,

³rama@staff.gunadarma.ac.id

Abstrak

Penelitian ini bertujuan untuk mendeteksi kesamaan dua citra berdasarkan warna dan deteksi tepi citra menggunakan metode deteksi tepi antara Sobel dan Robert. Hasil masing-masing deteksi selanjutnya dibandingkan untuk mendapatkan metode deteksi tepi terbaik. Proses pencocokan citra dilakukan dengan fitur histogram, teknik cropping, dan operator deteksi tepi (Robert dan Sobel). Terdapat dua tahapan yaitu pra-proses untuk membangun basis data citra dan proses CBIR untuk menentukan kemiripan citra berdasarkan citra input. Tahapan pra-proses diawali dengan citra yang dimasukkan dan ditransformasi ke histogram warna untuk mendapatkan nilai RGB (Red, Green, Blue). Selanjutnya nilai histogram warna disimpan ke dalam basis data. Tahapan proses CBIR diawali dengan deteksi tepi citra masukan menggunakan operator deteksi tepi Robert dan Sobel untuk mendapatkan nilai garis tepi dari citra input. Teknik cropping dari nilai garis tepi dilakukan untuk memotong sudut luar dari deteksi tepi sehingga citra yang didapatkan menjadi optimal. Selanjutnya citra tersebut ditransformasi ke histogram warna untuk mendapatkan nilai RGB. Tahapan akhir adalah proses pencocokan antara citra input dengan citra pada database dengan mencocokkan nilai RGB menggunakan jarak Euclidean. Hasil penelitian menunjukkan bahwa operator Robert mendapatkan hasil yang lebih optimal dibandingkan operator Sobel.

Kata Kunci: CBIR, deteksi tepi, kesamaan citra, Robert, Sobel.

Abstract

This study aims to determine the similarity based on the colors and patterns of the shape of the image, by comparing edge detection between Sobel and Robert. The image matching process is performed using the histogram feature, cropping techniques, and edge detection operators (Robert and Sobel). There are two stages, namely preprocessing and CBIR process to determine the similarity of the image. The preprocessing stage begins with the input image and is transformed into a histogram model to get the RGB values (Red, Green, Blue). Then the histogram values is stored in the database. The CBIR process stage begins with the input image and is transformed to the edge detection operator to get the edge value from the input image. Cropping technique is used to cut out the edge of the edge detection so that the image obtained is optimal. Then the image is transformed into a histogram model to get the RGB value. The final stage is the process of matching the input image (preprocessing) and input image (CBIR process) by matching the RGB value using the Euclidean Distance. The results of this study indicate that the Robert operator gets more optimal results than the Sobel operator.

Keywords: CBIR, image similarity, Robert, Sobel.

PENDAHULUAN

Citra terdiri dari dua jenis yaitu citra kontinu dan citra diskrit. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal analog misalnya mata manusia dan kamera analog. Citra diskrit dihasilkan melalui proses digitalisasi terhadap citra kontinu. Beberapa sistem optik dilengkapi dengan fungsi digitalisasi sehingga mampu menghasilkan citra diskrit misalnya kamera digital dan *scanner*. Citra diskrit disebut juga citra digital. Komputer yang umum dipakai saat ini dapat digunakan untuk mengolah citra digital [1].

Salah satu unsur citra adalah warna. Ruang warna citra ada bermacam-macam misalnya Salah satu ruang warna adalah ruang warna RGB. Pada ruang warna RGB setiap warna memperlihatkan komponen *spectral primary red, green, dan blue*. Model ini didasarkan pada sistem koordinat kartesian. *Sub-space* warna yang dicari adalah kubus dimana nilai RGB pada tiga sudut *cyan, magenta, dan yellow* ada pada tiga sudut lain, hitam adalah origin dan putih adalah titik paling jauh dari origin. Dalam model ini, *grayscale* (titik-titik nilai *equal* RGB) diperluas dari hitam ke putih, sepanjang garis gabungan dua titik. Perbedaan warna dalam model ini adalah titik-titik yang berada di dalam kubus dan didefinisikan oleh penyebaran vektor dari origin [2].

Sistem pencarian dan proses temu kenali citra berbasis *Content-Based Image*

Retrieval (CBIR) merupakan bagian penting dalam teknologi temu kenali informasi citra dan diaplikasikan dalam beberapa bidang seperti industri [3], forensik [4], *remote sensing* [5,6] dan kedokteran.

Berkembangnya pemakaian citra digital dan besarnya ukuran jumlah data citra digital, ditindaklanjuti dengan banyaknya peneliti melakukan penelitian untuk meningkatkan efisiensi pencarian dan pengindeksan data citra.

Beberapa penelitian yang berhubungan dengan CBIR telah banyak dilakukan dengan metode, operator, dan fitur yang berbeda-beda. Penelitian yang dilakukan oleh Mehtre, Mohan, dan Desai mengenai CBIR menggunakan dua metode *color matching* yaitu *color table method* dan *distance method* [7]. Penelitian lain terkait CBIR juga dilakukan pada bidang kesehatan menggunakan metode optimisasi Wavelet [8]. Pengenalan citra berdasarkan warna juga digunakan untuk pengenalan citra bendera negara [9].

Berdasarkan masalah di atas, penelitian ini bertujuan membuat aplikasi untuk melihat kemiripan suatu citra berdasarkan nilai dari histogram warna menggunakan deteksi tepi Sobel dan Robert. Penggunaan pengolahan citra sebagai dasar untuk mengetahui apakah antara citra yang dimasukkan memiliki kesamaan atau tidak dengan citra yang terdapat pada basis data. Kesamaan sebuah citra yang diujikan menggunakan nilai histogram dan dicocokkan dengan nilai histogram yang terdapat di dalam basis data dengan mengukur jarak kemiripan menggunakan jarak *Euclidean*.

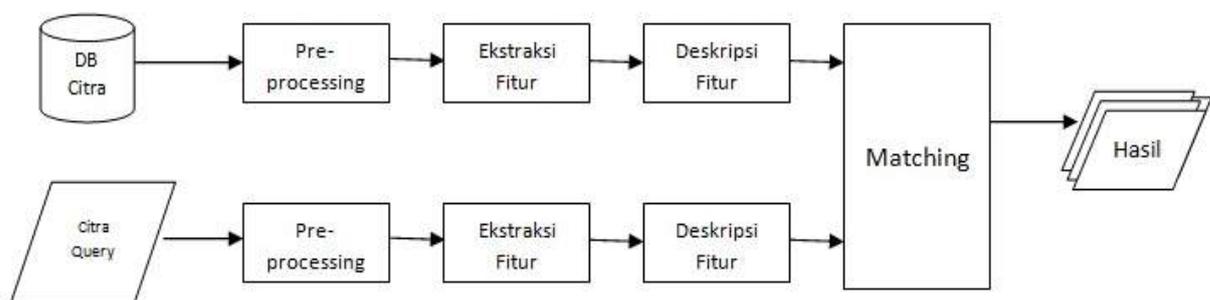
METODE PENELITIAN

Pada penelitian ini digunakan 50 citra berukuran 500×500 piksel yang diambil dari

buku “*Crystal Power, Crystal Healing*” [10] sebagai basis data. Pada Tabel 1 diberikan 50 daftar citra yang digunakan dalam penelitian ini.

Tabel 1. Basis Data Citra

No	Nama Batu	Jenis Batu	No	Nama Batu	Jenis Batu
1	Alexandrite	Mulia	26	Morion	Mulia
2	Aquamarine	Mulia	27	Pacific Opal	Mulia
3	Berlian	Mulia	28	Peridot	Mulia
4	Black Diamonds	Mulia	29	Red Diamond	Mulia
5	Blue Sapphire	Mulia	30	Red Topaz	Mulia
6	Burgundy	Mulia	31	Rose Alabaster	Mulia
7	Chrysolite	Mulia	32	Rose	Mulia
8	Citrine	Mulia	33	Rubi	Mulia
9	Cobalt	Mulia	34	Safir	Mulia
10	Crystal	Mulia	35	Serendibite	Mulia
11	Emerald	Mulia	36	Shadow Crystal	Mulia
12	Erinite	Mulia	37	Smoked Topaz	Mulia
13	Garnet	Mulia	38	Smoky Quartz	Mulia
14	Hyacinth	Mulia	39	Swarovsk	Mulia
15	Indian Red	Mulia	40	Tanzanite	Mulia
16	Indian Sapphire	Mulia	41	Topaz	Mulia
17	Indicolite	Mulia	42	Tourmaline	Mulia
18	Jadeite	Mulia	43	Turquoise	Mulia
19	Light Azore	Mulia	44	Yogo Sapphire	Mulia
20	Light Colorado Topaz	Mulia	45	Zamrud	Mulia
21	Light Olivine	Mulia	46	Zircon	Mulia
22	Light Peach	Mulia	47	Zoisite	Mulia
23	Light Topaz	Mulia	48	White Coral	Mulia
24	Lighthyst	Mulia	49	Red Coral	Mulia
25	Lime	Mulia	50	Star Ruby	Mulia

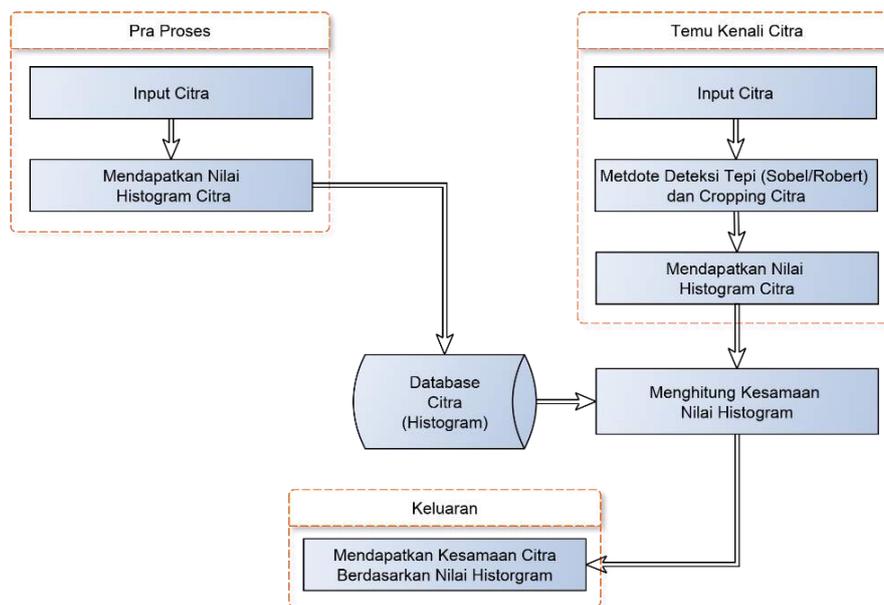


Gambar 1. Bagan Sistem Temu Kenali Citra

Pada Gambar 2 diberikan bagan pengembangan sistem temu kenali citra yang dibuat pada penelitian ini. Secara garis besar terbagi menjadi 3 proses yaitu pra-proses, temu kenali citra dan keluaran berupa nilai kesamaan dua citra.

Pada Gambar 1, cara kerja CBIR diawali dengan pra-proses pada citra di basis

data dan citra *query*. Selanjutnya dilakukan ekstraksi fitur citra dapat berupa warna, tekstur maupun bentuk. Berdasarkan fitur citra yang diperoleh selanjutnya dilakukan pencocokan antara citra *query* dengan citra pada basis data. Pada penelitian ini digunakan jarak *Euclidean* untuk mengukur jarak kemiripan citra.

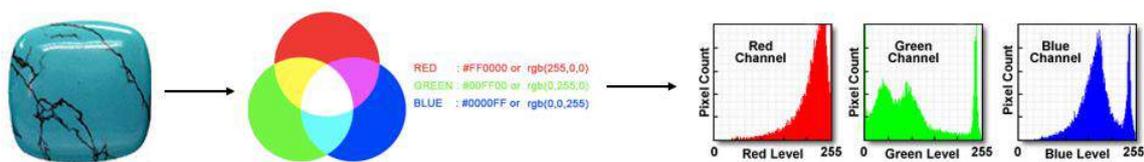


Gambar 2. Bagan Sistem Temu Kenali Citra

Pada tahap pra-proses dilakukan transformasi citra ke ruang warna RGB sehingga didapatkan nilai histogram citra berdasarkan ruang warna RGB. Tahap selanjutnya pada pra-proses adalah tahap penyimpanan nilai citra ke basis data yang berisikan ID, Nama, R, G, B, dan FILENAME. ID dijadikan sebagai *primary key* untuk citra *input*, NAMA adalah sebagai penamaan citra, dan R adalah nilai dari warna merah, G adalah nilai dari warna hijau, B

adalah nilai dari warna biru, dan FILENAME adalah posisi file basis data *input*.

Pada bagian pra-proses citra di-*input* untuk mendapatkan nilai histogram dengan mengambil data citra RGB. Gambar 3 ditunjukkan proses transformasi citra ke RGB dan mendapatkan nilai histogram warna. Setelah diperoleh nilai histogram citra dalam ruang warna RGB maka fitur citra disimpan ke dalam basis data dengan menyimpan nilai R, G, dan B.



Gambar 3. Transformasi Citra ke RGB dan Nilai Histogram

Tahap berikutnya adalah temu kenali citra yang diawali dengan proses deteksi tepi citra dan *cropping* citra. Citra hasil proses ini selanjutnya ditransformasi ke ruang warna RGB untuk mendapatkan nilai histogram citra. Langkah selanjutnya adalah pencocokan citra *input* dengan citra yang ada dalam basis data menggunakan jarak *Euclidean* untuk melihat jarak nilai kemiripan berdasarkan nilai RGB apakah antara citra *input* dan citra

yang terdapat di dalam basis data. Semakin kecil jarak *Euclidean* menunjukkan dua buah citra tersebut semakin mirip.

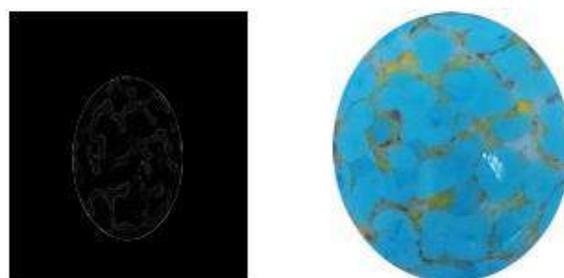
Pada proses temu kenali citra (CBIR) langkah yang dilakukan sama seperti pada tahap pra-proses kecuali pada tahap awal yaitu citra *input* melewati proses deteksi tepi. Pada penelitian ini deteksi tepi menggunakan dua operator yaitu Sobel dan Robert. Pada Gambar 4 ditampilkan salah satu citra hasil deteksi tepi.



Gambar 4. Citra Input Transformasi ke Operator Deteksi Tepi

Tahap berikutnya adalah optimalisasi bentuk citra hasil deteksi tepi menggunakan

teknik *cropping*. Pada Gambar 5 ditampilkan hasil teknik *cropping* dari hasil deteksi tepi citra.



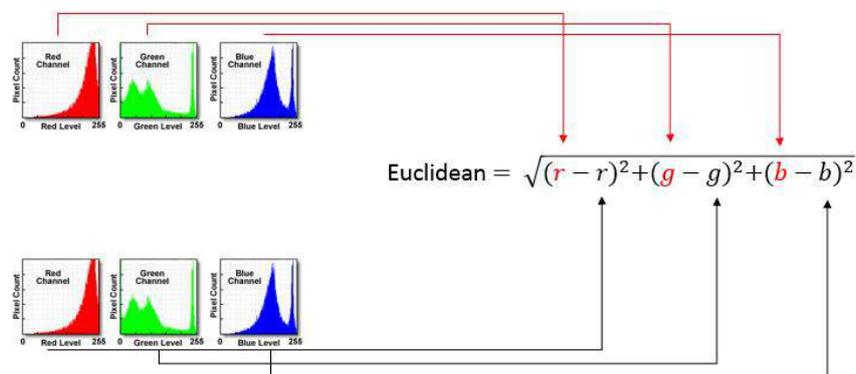
Gambar 5. Transformasi Deteksi Tepi ke Teknik *Cropping*

Setelah melalui teknik *cropping*, citra *input* tersebut dilakukan proses seperti pra-proses di mana citra *input* tersebut ditransformasi ke bentuk RGB dan mengambil nilai RGB berdasarkan nilai histogram citra. Tahap selanjutnya adalah mencocokkan nilai histogram citra yang terdapat di dalam basis data dengan nilai histogram citra *input*. Metode yang digunakan untuk mencocokkan nilai

histogram citra adalah metode pengukuran jarak *Euclidean*. Perhitungan jarak *Euclidean* menggunakan formula seperti pada Persamaan 1.

$$\text{jarak Euclidean} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

Cara menghitung nilai jarak *Euclidean* pada citra *input* dan citra pada basis data yang diilustrasikan pada Gambar 6.



Gambar 6. Perhitungan Jarak *Euclidean*

Pada Gambar 6 diilustrasikan perhitungan jarak *Euclidean* dari citra *input* dengan citra yang ada pada basis data menggunakan formula pada Persamaan (1). Semakin kecil jarak *Euclidean* antara citra *input* dengan citra yang ada pada basis data maka kedua citra tersebut semakin mirip.

HASIL DAN PEMBAHASAN

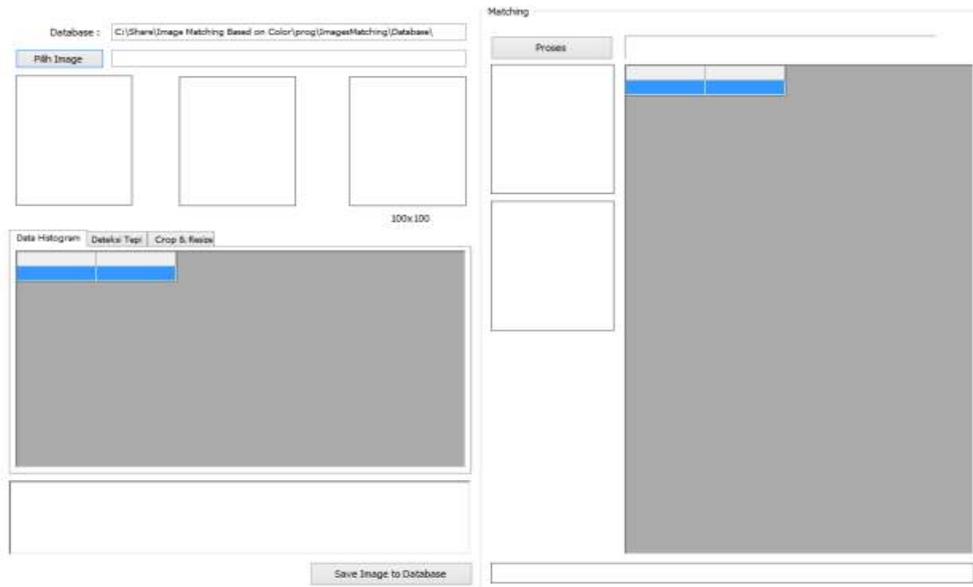
Perbandingan pencocokan citra dengan adanya pra-proses deteksi tepi Robert dan Sobel ditampilkan dalam aplikasi pendeteksi

kesamaan citra. Tampilan utama dari aplikasi tersebut dapat dilihat pada Gambar 7.

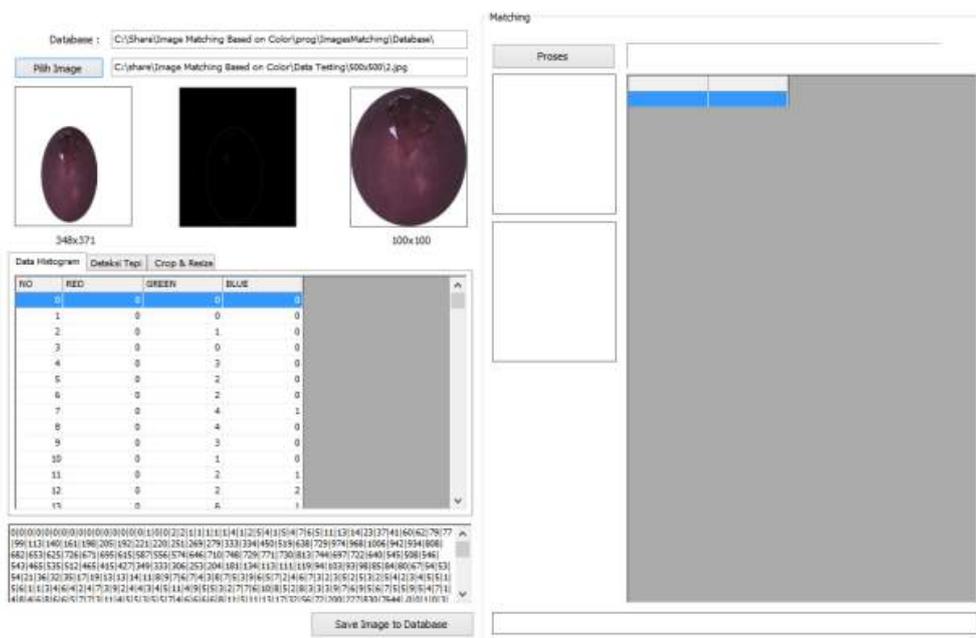
Setelah pengguna memasukkan citra ke dalam aplikasi maka tampilan aplikasi menjadi seperti pada Gambar 8.

Jika citra telah dimasukkan pada aplikasi maka tampilan aplikasi dapat dilihat pada Gambar 8. Aplikasi akan menampilkan nilai histogram warna RGB, hasil deteksi tepi dan hasil proses *cropping*.

Pada Gambar 9 ditunjukkan hasil dari kemiripan untuk citra *Star Ruby* yang terdapat pada basis data.



Gambar 7. Tampilan Utama Aplikasi



Gambar 8. Tampilan Aplikasi Setelah Input Citra



Gambar 9. Jarak Euclidean

Pada tahap pengujian digunakan 10 citra batu mulia sebagai *citra input* untuk dicocokkan

ke dalam basis data. Hasil perhitungan jarak *Euclidean* dapat dilihat pada Tabel 2.

Tabel 2. Jarak *Euclidean* dari Hasil Penelitian

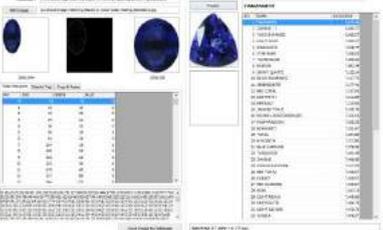
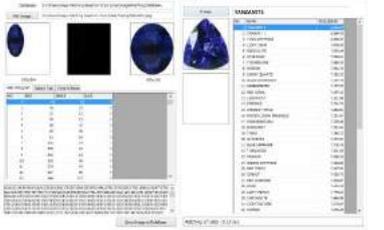
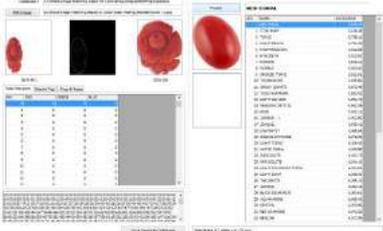
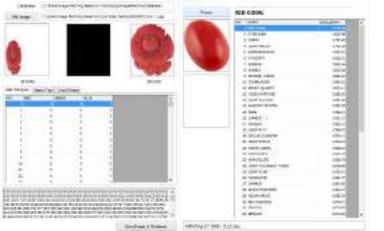
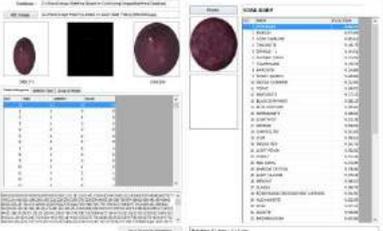
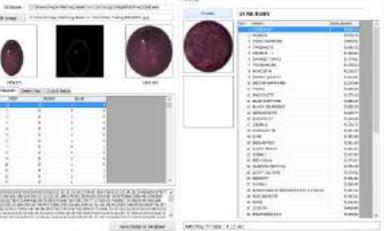
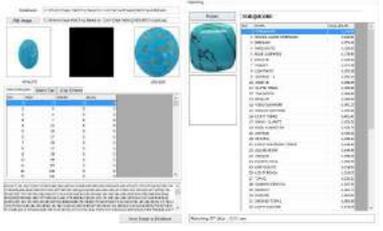
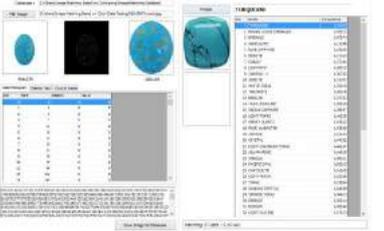
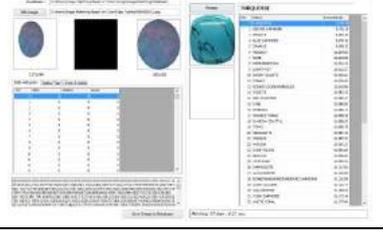
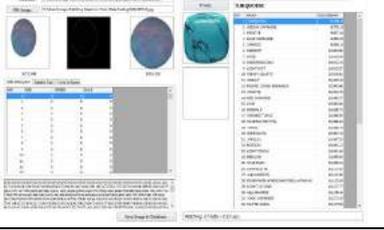
No	Citra <i>Input</i>	Citra <i>Output</i>	Nama Batu	Jarak <i>Euclidean</i>
1			White Coral	16632.56
2			Star Ruby	8082.52
3			Tanzanite	6642.46
4			Indian Sapphire	7818.95
5			Zamrud – 1	5630.81
6			Red Coral	5576.57
7			Turquoise	5724.07
8			Topaz	1766.69
9			Morion	1393.76
10			Turquoise	9748.39

Pada Tabel 2 ditunjukkan hasil citra *output* dan nilai jarak *Euclidean* pada setiap citra *input*. Masing-masing citra pada basis data dihitung jarak *Euclidean* dengan citra *input*. Citra *output* merupakan citra pada basis data yang memiliki jarak *Euclidean* terpendek.

Pada Tabel 3 ditunjukkan hasil perbandingan jarak *Euclidean* dengan

menggunakan operator Robert dan operator Sobel. Berdasarkan Tabel 3, pada 6 citra *input* yang berbeda, diperoleh jarak *Euclidean* yang berbeda pada kedua operator deteksi tepi. Berdasarkan 6 citra tersebut, terdapat 5 citra *input* dengan deteksi tepi Robert yang memiliki jarak *Euclidean* lebih kecil dibandingkan deteksi tepi Sobel.

Tabel 3. Hasil Perbandingan Jarak Euclid Berdasarkan Operator Deteksi Tepi

No	Robert [R]	Sobel [S]	E[R]	E[S]
1			66246	66247
2			557657	557606
3			808252	808261
4			1546969	1663256
5			572155	572407
6			97836	97839

KESIMPULAN DAN SARAN

Pada penelitian ini dilakukan pendeteksian kesamaan citra berdasarkan

warna pada citra batu mulia menggunakan dua metode deteksi tepi yaitu Roberts dan Sobel. Berdasarkan hasil uji coba, deteksi tepi Robert memiliki 9 nilai jarak *Euclidean* yang

lebih baik dibandingkan dengan deteksi tepi Sobel.

Pada penelitian lebih lanjut dapat dilakukan temu kenali citra menggunakan fitur lain seperti fitur bentuk atau tekstur. Selain itu, metode temu kenali citra dapat dikembangkan menggunakan pendekatan *deep learning*.

DAFTAR PUSTAKA

- [1] E. Prasetyo, *Pengolahan citra digital dan aplikasinya menggunakan Matlab*, Yogyakarta: Andi, 2011.
- [2] Shapiro, G. Linda, Stockman and C. George, *Computer vision*, Prentice Hall: New Jersey, 2002.
- [3] D. Tao dan X. Tang, "Nonparametric discriminant analysis in relevance feedback for content based image retrieval," In Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR), 2004, hal. 1013 – 1016.
- [4] G. AlGarni dan M. Hamiane, "A novel technique for automatic shoeprint image retrieval," *Forensic Science International*, vol. 181, no. 1, hal 10 – 14, 2008.
- [5] A. Hafiane, S. Chaudhuri, G. Seetharaman, dan B. Zavidovique, "Region-based CBIR in GIS with local space filling curves to spatial representation," *Pattern Recognition Letters*, vol. 27, no. 4, hal. 259 – 267, 2006.
- [6] Z. Xie, "A rotation- and flip-invariant algorithm for representing spatial continuity information of geographic images in content based image retrieval," *Computers and Geosciences*, vol. 30, no. 9 – 10, hal. 1093 – 1104, 2004.
- [7] B. M. Mehtre, S. K. Mohan, dan A. Desai, "Color matching for image retrieval", 2018. [Daring]. Tersedia pada: https://www.comp.nus.edu.sg/~mohan/papers/col_match.pdf. [Diakses: 12 Desember 2018].
- [8] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, dan C. Roux, "Wavelet optimization for content-based image retrieval in medical database," *Medical Image Analysis, Elsevier*, vol. 14, no. 2, hal. 227 – 241, 2010.
- [9] M. Cahyanti, R. A. Salim, dan M. Wisuda, "Implementasi pengolahan citra untuk pengenalan citra bendera negara berdasarkan warna," Dalam Prosiding Seminar Nasional Riset Teknologi Informasi, 2016, hal. 128 – 137.
- [10] M. Gienger, *Crystal power, crystal healing*, London: Cassell Illustrated, 2015

PENGARUH JUMLAH SUDU PADA TURBIN ANGIN SUMBU VERTIKAL TERHADAP DISTRIBUSI KECEPATAN DAN TEKANAN

¹Ridwan, ²Abdul Latief

^{1,2} Program Studi Teknik Mesin, Fakultas Teknologi Industri Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
ridwan@staff.gunadarma.ac.id, latiefconstantine@gmail.com

Abstrak

Angin merupakan salah satu sumber energi baru terbarukan. Sistem Konversi Energi Angin (SKEA) dapat digunakan untuk memanfaatkan angin menjadi energi mekanik atau energi listrik. Komponen utama SKEA terdiri dari rotor dengan sudu sebagai penggerak utama. Salah satu kelemahan turbin angin adalah efisiensinya yang masih relatif rendah, sehingga masih terus diteliti khususnya terkait dengan sudu (*blade*) rotor turbin. Pada penelitian ini dilakukan simulasi terhadap sudu turbin angin sumbu vertikal Savonius tipe U yang dipasang pada rotor turbin. Ada 3 kategori rotor yang disimulasikan yaitu: rotor dengan jumlah sudu/*blade* 2, 3 dan 4. Kecepatan angin yang diterapkan/dialirkan pada simulasi sudu ini adalah 5 m/dtk. Simulasi ini menggunakan alat bantu perangkat lunak SolidWorks/flow simulatian berbasis Finite Element Analysis (FEA). Tujuan penelitian ini akan dilihat/diketahui besarnya distribusi kecepatan dan distribusi tekanan pada masing-masing sudu tersebut. Berdasarkan hasil simulasi diketahui bahwa rotor dengan jumlah *blade* tiga memberikan distribusi kecepatan yang lebih tinggi yakni (4,8 m/dtk) dibanding dengan rotor dengan empat *blade* empat (4,5 m/dtk) dan dua *blade* (4,3 m/dtk). Turbin dengan jumlah *blade* tiga memberikan perbedaan distribusi tekanan maksimum dan minimum paling rendah dibanding turbin dengan empat *blade* dan dua *blade*, yakni 21,32 Pascal.

Kata Kunci: Angin, distribusi kecepatan, distribusi tekanan, savonius, solidWorks.

Abstract

Wind is one of the renewable energy sources that can be utilized as mechanical or electrical energy through the Wind Energy Conversion System (SKEA) The main components of the SKEA consist of rotors with blades as the main movers. One of the disadvantages of wind turbines is that their efficiency is still relatively low, so In this research, simulation on the Savonius vertical axis type U wind turbine blade is mounted on the turbine rotor, there are 3 simulated rotor categories: rotor with the number of blades / blade 2, 3 and 4. The wind speed applied / flowed in this blade simulation is 5 m / s, this simulation is used by SolidWorks / flow simulatian software based on Finite Element Analysis (FEA) The purpose of this research will be to see / know the magnitude of the velocity distribution and pressure distribution on each of the blades. From the simulation results it is known that the rotor of three blades provides a better / higher velocity distribution (4.8 m / sec) compared to four-blade rotors (4.5 m / sec) and two blades (4.3 m / sec). Turbines with three blades provide the maximum and minimum pressure distribution differences compared to turbines with four blades and two blades 21.32 Pascal.

Keywords: pressure distribution, Savonius, solidWorks, velocity distribution, wind.

PENDAHULUAN

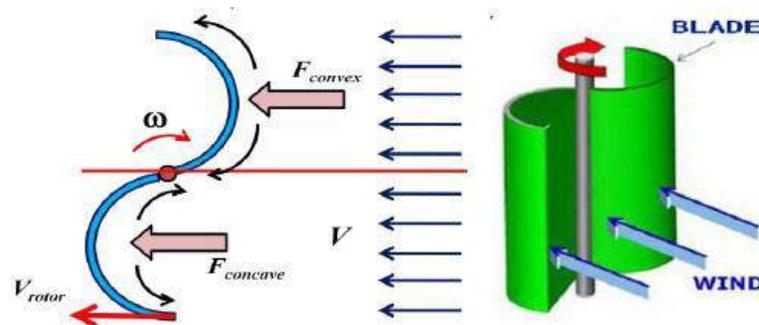
Angin merupakan salah satu sumber energi baru terbarukan yang dapat dimanfaatkan menjadi energi mekanik atau listrik melalui suatu konversi yang dinamakan Sistem Konversi energi angin (SKEA)[1]. Komponen utama SKEA terdiri dari rotor dengan sudu sebagai penggerak utama, generator sebagai pengubah energi mekanik menjadi energi listrik dan lain-lain. Energi angin merupakan salah satu energi yang ramah terhadap lingkungan. Energi ini dapat menjadi solusi terhadap pemanasan global dan berpotensi menggantikan energi dari fosil.[2]

Pemenuhan energi dalam suatu negara menjadi suatu hal yang sangat penting karena sangat mempengaruhi perkembangan industri di negara tersebut. Salah satu komoditi unggulan dalam peradaban modern adalah energi [2,3]. Jumlah konsumsi energi telah menjadi indikator untuk standar hidup dan tingkat industrilisasi. Saat ini, hampir

90% dari energi dunia berasal dari pembakaran bahan bakar fosil, yaitu batubara, minyak petroleum, gas alam, dan lain-lain.[4]

Trend penggunaan energi baru terbarukan termasuk energi angin mengalami peningkatan dari tahun ke tahun yang diprediksi akan mengalami kenaikan penggunaan mencapai 10 % dari konsumsi energi dunia pada tahun 2030. [2,5]. Kondisi alam dan geografis di Indonesia sangat cocok diterapkan turbin angin sumbu vertikal. Savonius turbin termasuk dalam salah satu jenis turbin sumbu vertikal.

Turbin angin sumbu vertikal tipe savonius sebagaimana diperlihatkan pada Gambar 1 memiliki kelebihan yakni dapat bekerja pada kecepatan angin yang relatif rendah dibanding dengan turbin angin sumbu horisontal [6]. Selain itu turbin jenis ini memiliki konstruksi yang sederhana dibanding jenis sumbu horisontal [7]. Prinsip kerja turbin Savonius dapat dilihat pada Gambar 1 berikut ini.



Gambar. 1 Skematik turbin angin savonius dua sudu dengan gaya hambat [5,8].

Jumlah sudu pada rotor turbin sangat mempengaruhi performa turbin angin sumbu

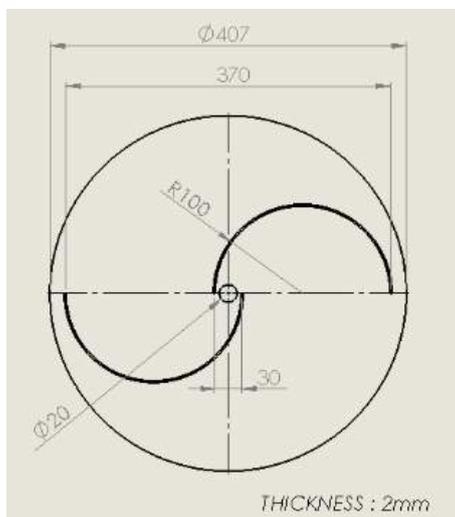
vertikal khususnya tipe savonius. Pada kecepatan angin (*wind speed*) yang relatif

sekitar 3 -5 m/s performa turbin meningkat seiring dengan penambahan jumlah sudu turbin [9]. Tujuan penelitian ini adalah menganalisis pengaruh jumlah sudu terhadap performa turbin yang mencakup distribusi kecepatan dan distribusi tekanan yang terjadi. Distribusi tekanan dan distribusi kecepatan yang diterima sudu turbin sangat mempengaruhi performa turbin tersebut dan berbanding lurus dengan putaran poros yang dapat dihasilkan turbin.

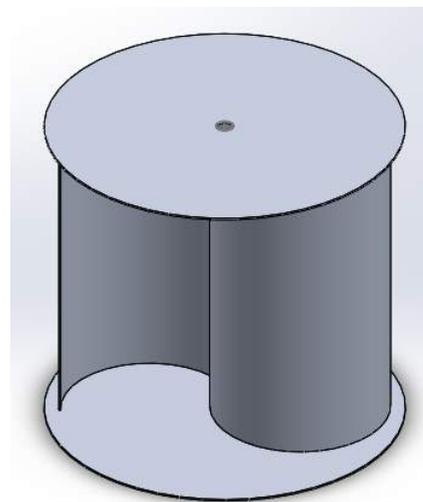
METODE PENELITIAN

Penelitian ini dilakukan dengan metode simulasi numerik dengan bantuan perangkat lunak solidworks (*flow simulations*) berbasis *Finite Element Analysis*

(*FEA*). Pada penelitian ini disimulasikan sudu turbin angin sumbu vertikal Savonius berbentuk U dengan 3 macam jumlah sudu (*blade*) yaitu 2, 3 dan 4 sudu. Kecepatan angin yang diterapkan atau dialirkan melalui sudu yaitu seragam sebesar 5 m/dtk. Model masing-masing sudu dibuat dalam dua dimensi (2D). Selanjutnya model sudu tersebut dibuat dalam model tiga dimensi (3D) dengan dimensi ukuran diameter sudu (d) = 200 mm, tinggi rotor = 370 mm, dan ketebalan blade dan endplate = 2 mm. Desain sudu dapat dilihat pada Gambar 2, Gambar 3 dan Gambar 4. Pada Gambar 2 sampai Gambar 4 terlihat pada pandangan/tampak atas dan tampak isometri/3D masing – masing untuk desain 2 sudu, 3 sudu dan 4 sudu.



(a)

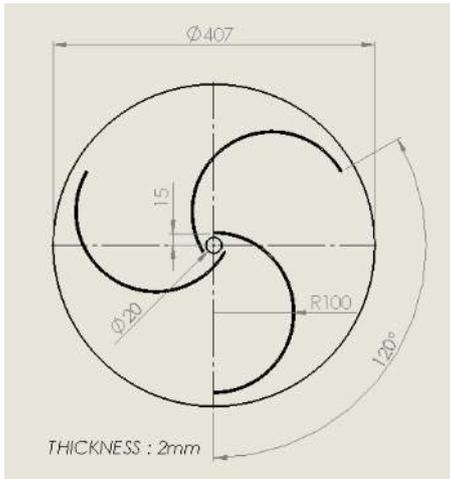


(b)

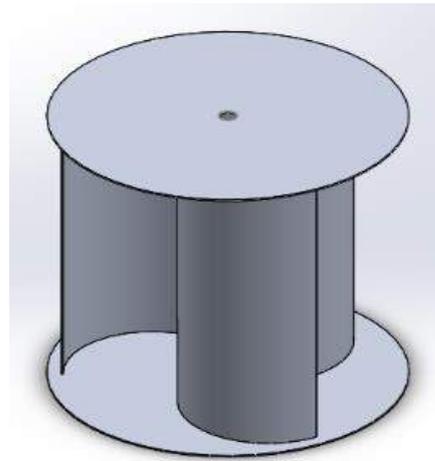
Gambar 2. Turbin Savonius 2 Sudu : (a) tampak atas, (b) tampak isometri/3D

Pada Gambar 2 ditunjukkan desain sudu untuk turbin savonius dengan banyaknya sudu adalah 2. Pada Gambar 2 (a) desain sudu

disajikan dalam 2 dimensi yang merupakan tampak atas, sedangkan pada Gambar 2 (b) merupakan desain sudu dalam 3 dimensi.



(a)

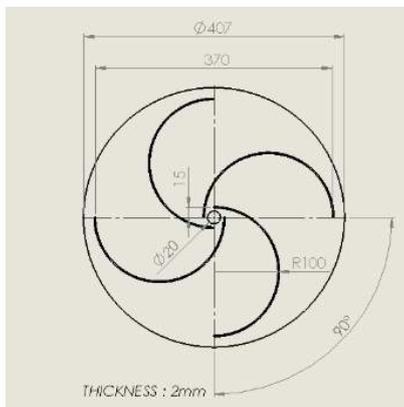


(b)

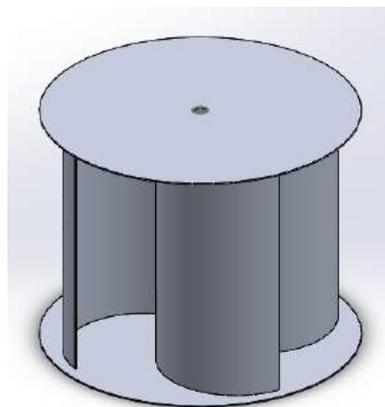
Gambar 3. Turbin Savonius 3 Sudu : (a) tampak atas, (b) tampak isometri/3D

Pada Gambar 3 ditunjukkan desain sudu untuk turbin savonius dengan banyaknya sudu adalah 3. Pada Gambar 3 (a) desain sudu disajikan dalam tampak atas

sehingga berupa desain 2 dimensi. Gambar 2 (b) merupakan desain sudu dalam 3 dimensi sesuai dengan desain 2 dimensi pada Gambar 3 (a).



(a)

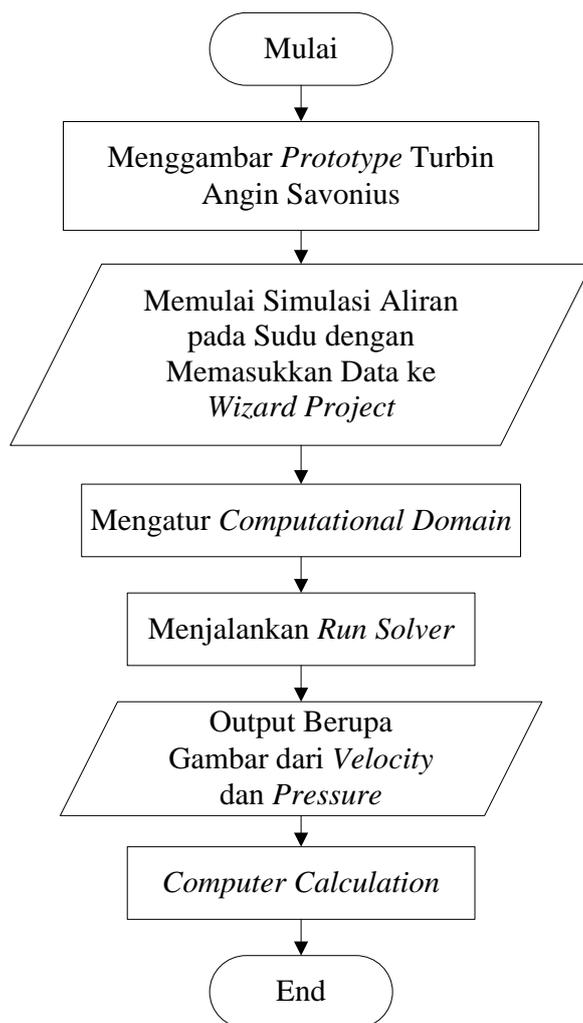


(b)

Gambar 4. Turbin Savonius 4 Sudu : (a) tampak atas, (b) tampak isometri/3D

Pada Gambar 4 ditunjukkan desain sudu untuk turbin savonius dengan banyaknya sudu adalah 4. Pada Gambar 4 (a) desain sudu

disajikan dalam 2 dimensi yang merupakan tampak atas, sedangkan pada Gambar 4 (b) merupakan desain sudu dalam 3 dimensi.



Gambar 5. Diagram Alir Simulasi

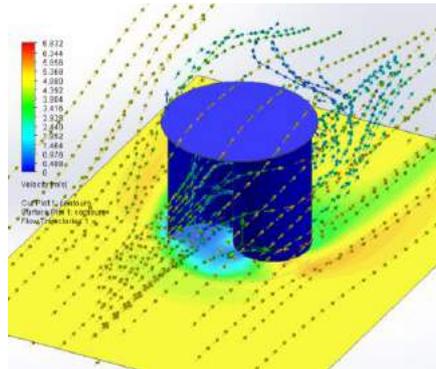
Gambar 5 merupakan *flow chart* proses simulasi. Penelitian dimulai dengan desain sudu sampai 3D yang dilanjutkan dengan mengatur kondisi batas dan domain Tahap berikutnya adalah menjalankan solver, running/simulasi. *Output* yang ditampilkan adalah data berupa distribusi kecepatan dan distribusi tekanan masing-masing sudu, serta melakukan analisis data hasil simulasi. Kecepatan angin dalam simulasi ini diterapkan/dialirkan seragam ke masing-masing

sudu/blade yakni 5 m/dtk.

HASIL DAN PEMBAHASAN

Hasil/*output* dari simulasi yang dilakukan ditampilkan dalam bentuk distribusi kecepatan dan distribusi tekanan untuk masing-masing rotor turbin dengan jumlah sudu atau blade 2,3, dan 4 blade. Sebagaimana dapat di-lihat pada Gambar 6 sampai dengan Gambar 11.

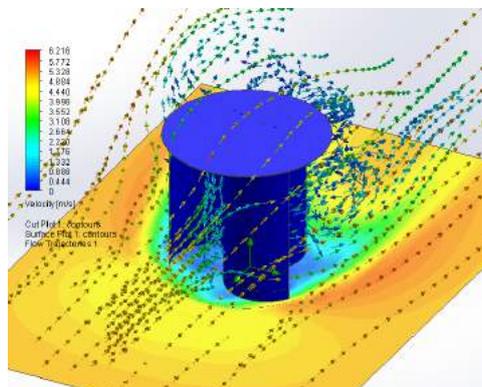
Distribusi Kecepatan



Gambar 6. Distribusi kecepatan pada turbin 2 blades

Pada Gambar 6 ditunjukkan distribusi kecepatan pada turbin dengan jumlah blade sebanyak dua buah. Pada gambar tersebut terlihat distribusi kecepatan paling rendah yakni 0 m/det ditandai dengan warna biru tua pada blade. Garis-garis arus yang menggambarkan kecepatan angin mengenai dan melalui sudu melintas melewati sudu dari sisi atas, samping kanan dan bawah dari blade. Kontur warna yang terjadi mulai dari biru tua, biru muda hijau, dan didominasi warna kuning dan hijau pada dasar sudu. Pada belakang sudu terlihat semacam olakan

(wake) dan selanjutnya arus berbelok. Distribusi kecepatan tertinggi yang terjadi pada bagian luar sudu berkisar 1,4 m/det sampai 4,3 m/det. Distribusi kecepatan aliran angin pada turbin dengan 2 blade tidak terlihat kecepatan angin yang mendekati kecepatan angin 5 m/dtk sebagaimana kecepatan angin yang dialirkan/disimulasikan pada blade. Hal ini diakibatkan karena turbin dengan jumlah sudu dua angin langsung terdistribusi secara merata kesamping sudu karena terjadi keseimbangan pada dua sudu tersebut pada kedua sisinya.

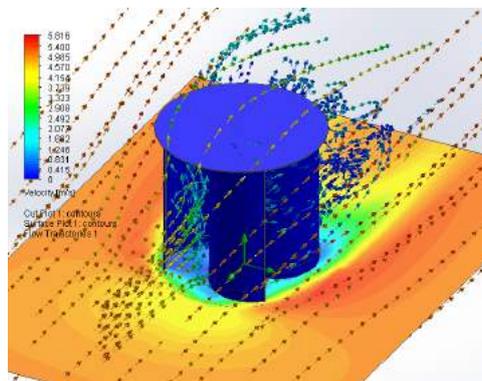


Gambar 7. Distribusi kecepatan pada turbin 3 blades

Pada Gambar 7 terlihat distribusi kecepatan pada turbin dengan jumlah blade sebanyak tiga buah. Pada gambar tersebut terlihat distribusi kecepatan paling rendah yakni 0 m/det dan tertinggi berada pada 4,8 m/dtk. Garis-garis arus yang menggambarkan kecepatan angin mengenai dan melalui sudu melintas melewati sudu dari sisi atas, samping kanan dan tengah sudu. Kontur warna lebih didominasi warna kuning tua kemerah-merahan dan terlihat distribusi kecepatannya lebih merata dan lebih rapat garis arusnya dibanding pada sudu dengan jumlah blade dua buah. Distribusi kecepatan terlihat lebih merata pada setiap sudu, dan terjadi kecepatan yang tetap tinggi yaitu di atas 4 m/dtk pada setiap sudu sehingga hal ini sangat bagus dan berpotensi untuk memberikan dorongan pada sudu sehingga

pada akhirnya akan memberikan kecepatan putar yang lebih tinggi pada rotor turbin. Kecepatan rotor turbin berbanding lurus dengan konversi energi putar (mekanik) rotor menjadi energi listrik pada generator yang dapat menghasilkan listrik, sebagai konversi dari putaran rotor bilamana dihubungkan dengan generaor listrik.

Turbin dengan jumlah blade tiga dari hasil simulasi terlihat menyebabkan distribusi kecepatan yang lebih tinggi dan merata dengan baik pada setiap sudu turbin. Hal ini diakibatkan karena posisi turbin sumbu vertikal ini memberikan keseimbangan kecepatan angin bila mana pada sudu satu telah dilewati angin selanjutnya diteruskan ke sudu dua dan sudu tiga, sehingga efektifitas angin yang dialirkan mampu terdistribusi dengan baik ke masing-masing sisi sudu.



Gambar 8. Distribusi kecepatan pada turbin 4 blades

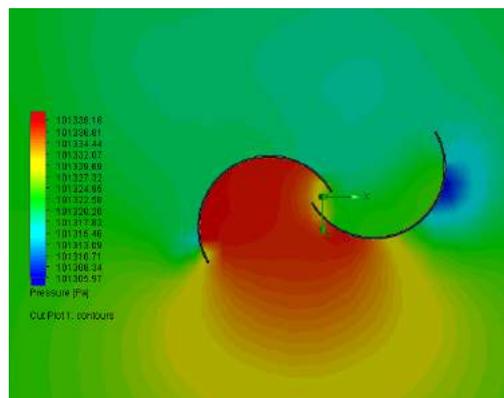
Pada Gambar 8 terlihat distribusi kecepatan pada turbin dengan jumlah blade sebanyak empat buah/bilah. Pada gambar tersebut terlihat distribusi kecepatan paling rendah yakni 0 m/det dan tertinggi berada pada 4,5 m/dtk. Garis-garis arus yang

menggambar-kan kecepatan angin mengenai dan melalui sudu melintas melewati sudu dari sisi atas, dan samping sudu. Kontur warna lebih didominasi warna kuning tua, dan terlihat distribusi kecepatannya lebih terkonsentrasi dibelakang blade, dan terjadi

olakan (*wake*) yang tinggi. Olakan yang tinggi akan kurang baik bagi putaran rotor yang juga dapat mengakibatkan hambatan pada sudu sehingga sebagian gaya dorong aliran angin tidak sepenuhnya mendorong sudu dan tidak maksimalnya konversi putaran nantinya ke rotor blade.

Semakin tinggi distribusi kecepatan yang terjadi pada rotor akan semakin baik karena memungkinkan memberikan gaya dorong pada sudu dan diteruskan pada poros rotor sehingga dapat menghasilkan energi mekanik, energi listrik berguna sesuai dengan kebutuhan/aplikasinya.

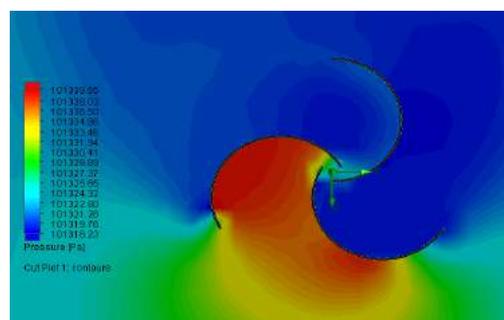
Distribusi Tekanan



Gambar 9. Distribusi tekanan pada turbin angin 2 blades

Pada Gambar 9 terlihat distribusi tekanan pada turbin dengan jumlah blade dua, distribusi tekanan minimum terjadi pada simulasi ini yaitu 101.305,97 Pascal dan tertinggi sebesar 101.339,18. Pascal dengan selisih sebesar 33,21 Pascal. Hasil simulasi dengan dua blade ini terlihat tekanan akibat

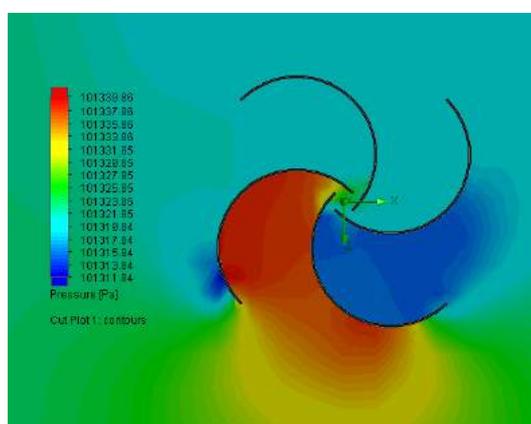
kecepatan angin tidak merata secara luas. Pada kedua blade lebih terkonsentrasi pada satu sudu sehingga tidak optimal dalam memberikan gaya dorong atau torsi pada sudu yang berakibat langsung pada kemampuan rotor melalui blade melakukan putaran poros turbin.



Gambar 10. Distribusi tekanan pada turbin angin 3 blade

Gambar 10 menampilkan distribusi tekanan pada turbin dengan jumlah blade tiga, distribusi tekanan minimum terjadi adalah 101.318,23 Pascal dan tertinggi sebesar 101.339, 23 Pascal, dengan selisih 21, 32 Pascal. Hasil simulasi dengan tiga blade ini terlihat tekanan akibat kecepatan angin lebih merata secara luas pada pada sekitar blade, sehingga lebih optimal dalam memberikan gaya dorong atau torsi pada sudu yang berakibat langsung pada

kemampuan rotor melalui blade melakukan putaran poros turbin. Distribusi tekanan dekat rotor lebih tinggi dan tersebar secara luas hal ini dapat terjadi karena turbin dengan tiga blade mampu menahan angin dan selanjutnya mendistribusikan lebih cepat pada rotor berikutnya sambil berputar. Salah satu kelebihan dari turbin angin sumbu vertikal adalah dengan kecepatan angin yang relatif rendah sekitar 0,5 m/s sampai 3 meter perdetik sudah mampu memutar rotor turbin.



Gambar 11. Distribusi tekanan pada turbin angin 4 blade

Pada Gambar 11 terlihat distribusi tekanan pada turbin dengan jumlah blade empat. Distribusi tekanan minimum terjadi adalah 101.311,84 Pascal dan tertinggi sebesar 101.339,86. Pascal dengan selisih/perbedaan sebesar 28,02 Pascal. Hasil simulasi dengan empat blade ini terlihat tekanan akibat kecepatan angin kurang merata dibanding pada turbin dengan tiga blade, tapi masih lebih baik dibanding turbin dengan dua blade. Tekanan kurang terdistribusi secara merata. Tekanan yang terjadi pada sudu satu, jauh berbeda dengan sudu dua, tiga

dan empat. Turbin dengan jumlah sudu empat memiliki masa yang lebih tinggi dari turbin dengan jumlah blade tiga sehingga tekanan dari dorongan /kecepatan angin sebagian terserap/terdisipasi dari massa rotor ini, sehingga efektifitas turbin dengan tiga blade lebih baik dalam distribusi tekanan dari turbin empat blade. Semakin kecil perbedaan antara distribusi/sebaran tekanan yang tertinggi dan terendah maka akan berdampak baik. Hal ini berbanding lurus dengan performa rotor turbin karena berbanding langsung dengan energi putar/energi mekanik,

torsi serta daya yang bisa dibangkitkan melalui putaran poros rotor turbin.

KESIMPULAN DAN SARAN

Jumlah sudu pada turbin angin sumbu vertikal jenis savonius mempengaruhi distribusi kecepatan dan distribusi tekanan angin yang melalui dan bekerja pada sudu/blade turbin. Distribusi kecepatan maksimum terjadi pada turbin dengan jumlah sudu sebanyak 3 yaitu 4,8 m/dtk, dan distribusi kecepatan turbin dengan empat sudu lebih tinggi dibanding dengan turbin dengan jumlah sudu dua. Distribusi kecepatan tertinggi pada turbin dengan empat blade yakni 4,5 m/dtk, sedangkan pada turbin dengan jumlah sudu dua distribusi tekanan tertinggi hanya 4,3 m/dtk. Turbin dengan jumlah blade tiga memberikan perbedaan distribusi tekanan maksimum dan minimum paling rendah dibanding turbin dengan empat blade dan dua blade, yakni 21,32 Pascal. Turbin dengan empat blade memiliki perbedaan distribusi tekanan maksimum dan minimum sebesar 28,02 Pascal dan turbin dengan dua blade sebesar 33,21 Pascal.

Pada penelitian selanjutnya dapat dikembangkan dengan menambah variasi bentuk jumlah sudu dan variasi besar kecepatan angin yang dialirkan pada sudu turbin. Diperlukan pengujian eksperimen pada terowongan angin (*wind tunnel*) untuk melihat perbandingan antara hasil simulasi dan hasil eksperimen.

DAFTAR PUSTAKA

- [1] J. V. Akwa,, G. A. d. S. Júnior, and A. P. Petry. "Discussion on the verification of the overlap ratio influence on performance coefficients of a Savonius wind rotor using computational fluid dynamics." *Renewable energy*, vol 38, no. 1, hal. 141-149, 2012.
- [2] BP, *BP Energy Outlook*, Jakarta : ESDM, 2017.
- [3] S. M. Irvan, "Savonius wind turbine vertical design for source of energy for lighting the highways," Tesis, Universitas Gunadarma, Depok, 2017.
- [4] M. E. Prasetya, "Studi kinerja turbin angin sumbu horizontal naca 4412 dengan modifikasi sudu tipe flat pada variasi sudut kemiringan 0°, 10°, 15°," Tesis, UMS, Surakarta, 2015.
- [5] Ridwan, I. Setyawan, dan A. Latief, "Pengaruh Jumlah Sudu dan Jumlah Fin pada Sudu terhadap Unjuk Kerja Turbin Angin Savonius Tipe U," *SEMRESTEK 2018 Proceedings*, 2018, hal. 225-234.
- [6] F. Wenehenubun, A. Saputra, and H. Sutanto, "An experimental study on the performance of Savonius wind turbines related with the number of blades" *Energy Procedia*, vol. 68, hal. 297 - 304, 2015.

- [7] N.H.Mahmoud, A.A.El-Haroun, E.Wahba, and M.H.Nasef, "An experimental study on improvement of Savonius rotor performance," *Alexandria Engineering Journal*, vol. 51, hal. 19-25, 2012.
- [8] Al-Kayiem, H. H. Bhayo, B. A. Assadi, and Mohsen, "Comparative critique on the design parameters and their effect on the performance of S-rotors," *Renewable Energy*, vol. 99, hal. 1306-1317, 2016.
- [9] F. H. Napitupulu, dan S. Siregar. "Perancangan turbin vertikal axis savonius dengan menggunakan 8 buah sudu lengkung," *Jurnal Dinamis*, vol. 1, no. 13, hal. 24 – 36, 2013.

IMPLEMENTASI ALGORITMA ASSOCIATION RULE UNTUK PROMOSI PRODUK BERBASIS WEBSITE PADA BENGKEL DELTA JAYA MOTOR

¹Anggara Fajri Afif, ²Ericks Rahmat Swedia, ³Margi Cahyanti

^{1,3}Fakultas Ilmu Komputer Universitas Gunadarma, ²Fakultas Teknologi Industri Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat

¹anggaraafif@gmail.com, ²ericks_rs@staff.gunadarma.ac.id, ³margi@staff.gunadarma.ac.id

Abstrak

Bengkel Delta Jaya Motor merupakan toko yang menjual sparepart motor yang hampir setiap hari dikunjungi konsumen. Adanya kegiatan transaksi penjualan setiap hari membuat data semakin lama akan semakin bertambah banyak. Pada saat ini toko Delta Jaya Motor belum memanfaatkan ketersediaan jumlah data yang besar untuk menentukan strategi bisnis, seperti menentukan produk yang sering dibeli oleh konsumen, menentukan strategi pemasaran, dan mengatur ketersediaan stok produk. Kemajuan teknologi informasi juga belum dimanfaatkan, aplikasi website adalah suatu wadah informasi yang bersifat realtime sehingga mempermudah konsumen memperoleh informasi. Dengan adanya aplikasi website diharapkan memudahkan proses bisnis pada toko Delta Jaya Motor seperti konsumen dapat langsung melihat produk di halaman website beserta stok dan produk lain yang sering dibeli bersamaan. Penelitian ini membahas analisis dan penerapan data mining untuk melihat pola pembelian konsumen Delta Jaya motor menggunakan algoritma association rule berbasis website. Dengan menggunakan algoritma association rule berhasil diperoleh sebelas aturan asosiasi dengan memberikan batas minimum support 13% dan minimum confidence 30%. Hasil dari aturan asosiasi memberikan informasi untuk membantu pihak manajemen atau pemilik bengkel dalam menentukan strategi bisnisnya, seperti mengatur produk yang sering dibeli, menentukan strategi pemasaran dan mengatur ketersediaan stok produk.

Kata Kunci: aplikasi web, association rule, data mining.

Abstract

Delta Jaya Motor Workshop is a shop that sells motorcycle spare parts that are visited by consumers almost every day. The existence of sales transaction activities every day makes the data more and more will increase. At present Delta Jaya Motor stores have not utilized the availability of large amounts of data to determine business strategies, such as determining products that are often bought by consumers, determining marketing strategies, and managing the availability of product stock. Advances in information technology have also not yet been exploited, website applications are real time information containers that make it easier for consumers to obtain information. The website application is expected to facilitate business processes in Delta Jaya Motor stores such as consumers can immediately see the product on the website page along with stock and other products that are often purchased together. This study discusses the analysis and application of data mining to see the purchase patterns of Delta Jaya motor consumers using the website-based association rule algorithm. By using the association rule algorithm, eleven association rules were obtained by providing a minimum support limit of 13% and a minimum confidence of 30%. The results of the association rules provide information to assist the management or workshop owner in determining their business strategies, such as managing products that are frequently bought, determining marketing strategies and managing product stock availability.

Keywords: association rule, data mining, web application.

PENDAHULUAN

Saat ini teknologi informasi berkembang begitu cepat seiring dengan berkembangnya ilmu pengetahuan. Teknologi informasi ini memiliki banyak sekali peranan dan dampaknya dalam berbagai bidang, salah satunya pada bidang bisnis. Banyaknya persaingan dalam dunia bisnis khususnya dalam industri penjualan, membuat para pelaku bisnis harus selalu memikirkan strategi-strategi dan terobosan yang dapat meningkatkan penjualan dan pemasaran produk yang dijual. Salah satunya dengan pemanfaatan data transaksi penjualan produk [1].

Algoritma *association rule* termasuk salah satu teknik dalam *data mining*. Teknik ini bekerja dengan mencari keterhubungan antara satu *item* dengan *item* lainnya. Algoritma telah diterapkan dalam berbagai bidang antara lain analisis data cuaca untuk menentukan perkiraan cuaca [2] dan analisis menggunakan *data mining* untuk mengetahui obat yang sering dibeli [3]. Pada penelitian yang dilakukan oleh Cahyanti, Mujahidin, dan Swedia menggunakan algoritma *association rule* untuk menganalisis nilai mahasiswa di Universitas Gunadarma sehingga dapat diketahui kecenderungan pola mahasiswa jika mengulang mata kuliah tertentu [4]. Penelitian lain mengenai algoritma *association rule* juga telah dilakukan oleh Nurdin. Pada penelitian tersebut menggunakan algoritma *association rule* untuk menganalisis penjualan barang

pada supermarket sejahtera dalam mengatur tata letak produk yang sering dibeli [5].

Industri besar seperti PT. Unilever juga melakukan analisis asosiasi untuk implementasi produknya dengan hasil informasi baru yang dapat memudahkan distributor menemukan strategi yang tepat untuk melakukan pemasaran pada produk PT. Unilever Indonesia [6]. Aturan asosiasi ini juga paling sering dimanfaatkan untuk promosi produk guna mengetahui keterhubungan antara produk yang sering dibeli dengan cara melakukan analisis terhadap pola kebiasaan konsumen dalam membeli suatu produk serta hubungan dengan produk lain yang dibeli [7]. Dengan mengetahui produk-produk apa saja yang telah dibeli secara bersamaan oleh konsumen, maka pihak manajemen dapat menentukan kebijakan-kebijakan untuk meningkatkan penjualan di tokonya, seperti mengatur dan menampilkan produk-produk unggulan di suatu halaman produk.

Toko Delta Jaya Motor merupakan usaha ritel yang menyediakan berbagai aksesoris *sparepart* motor. Saat ingin memilih aksesoris, konsumen terkadang belum mengetahui apa saja yang ingin diganti. Dengan menggunakan metode *association rule*, konsumen dapat mengetahui dengan cepat produk mana yang sering dibeli dan mungkin dibutuhkan. Saat ini proses bisnis yang ada di toko Delta Jaya Motor masih konvensional, seperti penulisan nota transaksi penjualan menggunakan kwitansi yang ditulis tangan dan belum tersedianya

layanan informasi teknologi untuk menampilkan *sparepart* dan stok produk.

Oleh karena itu, toko Delta Jaya Motor memerlukan suatu sarana informasi dan layanan penjualan berbasis *website* yang bisa memenuhi kebutuhan dan memudahkan perusahaan dalam menjalankan kegiatan bisnis. Pemanfaatan teknologi informasi dapat memudahkan transaksi serta memberikan informasi tentang produk-produk unggulan agar penjualan toko semakin meningkat. Pada penelitian ini

membahas mengenai penggunaan algoritma *association rule* untuk promosi produk berbasis *website* pada toko Delta Jaya Motor.

METODE PENELITIAN

Penerapan *data mining* menggunakan teknik *association rule* yang dilakukan pada penelitian ini memiliki beberapa tahapan. Tahap-tahap penelitian dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1, tahapan awal penelitian adalah melakukan pengumpulan data transaksi penjualan yang didapatkan dari hasil wawancara dan survei ke toko Delta Jaya Motor. Ada 11 produk pada toko yang digunakan pada penelitian ini. Produk-produk tersebut adalah knalpot, *velg*, *gear*, suspensi depan, selang rem, *stang*, suspensi belakang, *handle*, kaliper, *arm* dan *footstep*. Data transaksi penjualan yang telah terjadi sejak awal tahun

2017, yang berjumlah 1000 data transaksi kemudian dikumpulkan untuk dianalisis dengan menggunakan *association rule*. Setelah hasil asosiasi diperoleh, kemudian dilakukan perancangan halaman situs web dan basis data untuk mengimplementasikan algoritma *association rule* pada sistem penjualan yang berbasis web.

Algoritma Association Rule

Algoritma *association rule* terdiri dari dua langkah sebagai berikut [8]:

1. Mencari semua hubungan antar *item* berdasarkan frekuensi pembelian,
2. Menentukan aturan asosiasi yang memenuhi kriteria minimum *support* dan *confidence*. Berdasarkan frekuensi pembelian antar *item*.

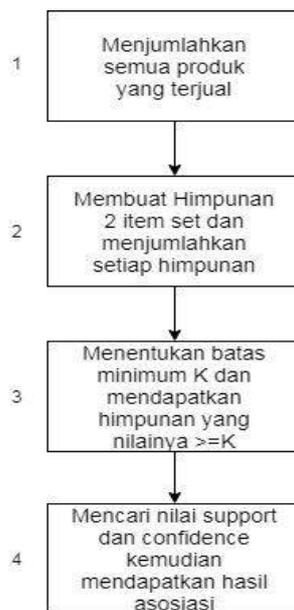
Rumus perhitungan untuk mencari nilai *support* dan *confidence* didefinisikan secara berurutan seperti pada Persamaan (1) dan (2) [5].

$$Support = \frac{\sum \text{Item yang dibeli sekaligus}}{\sum \text{Jumlah seluruh transaksi}} \times 100\% \quad (1)$$

Confidence =

$$\frac{\sum \text{Item yang dibeli sekaligus}}{\sum \text{Jumlah transaksi pada bagian antecedent}} \times 100\% \quad (2)$$

Item yang dibeli sekaligus adalah jumlah dari seluruh barang yang dibeli secara bersamaan pada satu waktu transaksi, yang merupakan gabungan 2 *item* set. Seluruh transaksi adalah jumlah transaksi. Transaksi *antecedent* adalah jumlah suatu produk yang terjual dari seluruh transaksi. Gambar 2 merupakan diagram alur algoritma *association rule*.



Gambar 2. Diagram Alur Teknik *Association Rule*

HASIL DAN PEMBAHASAN

Jumlah data yang diolah sebanyak 1000 transaksi dan terdapat 11 *item* produk. Data penjualan setiap produk yang sudah dijumlah dapat dilihat pada Tabel 1.

Pada tahap selanjutnya dilakukan penggabungan himpunan sebanyak dua *item* set setiap produk untuk mengetahui jumlah pembelian produk secara bersamaan. Proses ini menghasilkan 55 himpunan seperti terlihat pada Tabel 2.

Setelah mendapatkan jumlah keseluruhan produk yang dibeli secara bersamaan dari setiap himpunan, selanjutnya adalah menentukan nilai minimum yaitu $>= 130$. Hal tersebut berarti jika jumlah seluruh nilai produk yang dibeli secara bersamaan pada setiap himpunan lebih atau sama dengan 130 maka akan masuk ke dalam aturan asosiasi. Himpunan yang terbentuk dapat dilihat pada Tabel 3.

Salah satu contoh perhitungan untuk produk knalpot dan hasil asosiasi dapat dilihat pada Tabel 4.

Langkah selanjutnya adalah mencari nilai $support \times confidence$ seperti terlihat pada Tabel 5.

Hasil asosiasi dari produk knalpot diurutkan berdasarkan hasil perkalian $support$ dan $confidence$ yang paling besar seperti terlihat pada Tabel 6.

Tabel 1. Data Transaksi Penjualan

ID Transaksi	Knalpot	Velg	Gear	Stang	Suspensi Depan (SD)	Suspensi Belakan (SB)	Handle	Kaliper	Selang	Arm	Footstep
1	1	0	0	1	0	1	0	1	0	0	0
2	0	1	1	0	1	0	1	0	0	1	0
3	0	0	0	0	0	1	0	1	0	0	0
4	0	1	0	1	0	0	0	0	0	1	0
5	1	0	0	0	1	1	1	1	0	0	1
.....
1000	1	1	1	0	0	1	1	1	1	1	0
Σ	424	368	368	424	438	449	435	440	393	417	434

Tabel 2. Himpunan Dua Item Set

Kombinasi	Σ	Kombinasi	Σ	Kombinasi	Σ	Kombinasi	Σ
Knalpot, Velg	90	Velg, Handle	156	Stang, SB	135	SB, Selang	159
Knalpot, Gear	102	Velg, Kaliper	147	Stang, Handle	188	SB, Arm	247
Knalpot, Stang	166	Velg, Selang	126	Stang, Kaliper	248	SB, Footstep	181
Knalpot, SD	234	Velg, Arm	162	Stang, Selang	147	Handle, Kaliper	141
Knalpot, SB	169	Velg, Footstep	172	Stang, Arm	148	Handle, Selang	125
Knalpot, Handle	149	Gear, Stang	127	Stang, Footstep	169	Handle, Arm	155
Knalpot, Kaliper	190	Gear, SD	110	SD, SB	163	Handle, Footstep	221
Knalpot, Selang	208	Gear, SB	158	SD, Handle	117	Kaliper, Selang	125
Knalpot, Arm	170	Gear, Handle	221	SD, Kaliper	187	Kaliper, Arm	125
Knalpot, Footstep	155	Gear, Kaliper	136	SD, Selang	234	Kaliper, Footstep	181
Velg, Gear	135	Gear, Selang	122	SD, Arm	166	Selang, Arm	152
Velg, Stang	123	Gear, Arm	162	SD, Footstep	170	Selang, Footstep	109
Velg, SD	140	Gear, Footstep	189	SB, Handle	140	Arm, Footstep	154
Velg, SB	209	Stang, SD	137	SB, Kaliper	156		

Tabel 3. Himpunan yang Terbentuk

Kombinasi	Σ	Kombinasi	Σ	Kombinasi	Σ	Kombinasi	Σ
Knalpot, SD	234	Velg, Kaliper	147	Stang, Selang	147	SB, Footstep	181
Knalpot, Stang	166	Velg, Arm	162	Stang, Arm	148	Handle, Kaliper	141
Knalpot, SB	169	Velg,	172	Stang,	169	Handle, Arm	155
		Footstep		Footstep			
Knalpot,	149	Gear, SB	158	SD, SB	163	Handle, Footstep	221
Handle							
Knalpot,	190	Gear,	221	SD, Kaliper	187	Kaliper, Footstep	181
Kaliper		Handle					
Knalpot, Selang	208	Gear,	136	SD, Selang	234	Selang, Arm	152
		Kaliper					
Knalpot, Arm	170	Gear, Arm	162	SD, Arm	166	Arm, Footstep	154
Knalpot,	155	Gear,	189	SD, Footstep	170		
Footstep		Footstep					
Velg, Gear	135	Stang, SD	137	SB, Handle	140		
Velg, SD	140	Stang, SB	135	SB, Kaliper	156		
Velg, SB	209	Stang,	188	SB, Selang	159		
		Handle					
Velg, Handle	156	Stang,	248	SB, Arm	247		
		Kaliper					

Tabel 4. Perhitungan Untuk Produk Knalpot.

No	Aturan yang diperoleh	Support	Confidence
1	Knalpot-Suspensi Depan (SD)	$(234/1000)*100\% = 23,4\%$	$(234/424)*100\%=55\%$
2	Knalpot-Stang	$(166/1000)*100\% = 16,6\%$	$(166/424)*100\%=39\%$
3	Knalpot-Suspensi Belakang(SB)	$(169/1000)*100\% = 16,9\%$	$(169/424)*100\%=40\%$
4	Knalpot-Handle	$(149/1000)*100\% = 14,9\%$	$(149/424)*100\% = 35\%$
5	Knalpot-Kaliper	$(190/1000)*100\% = 19\%$	$(190/424)*100\% = 44\%$
6	Knalpot-Selang	$(208/1000)*100\% = 20,8\%$	$(208/424)*100\% = 49\%$
7	Knalpot-Arm	$(170/1000)*100\% = 17\%$	$(170/424)*100\% = 41\%$
8	Knalpot-Footstep	$(155/1000)*100\% = 15,5\%$	$(155/424)*100\% = 36\%$

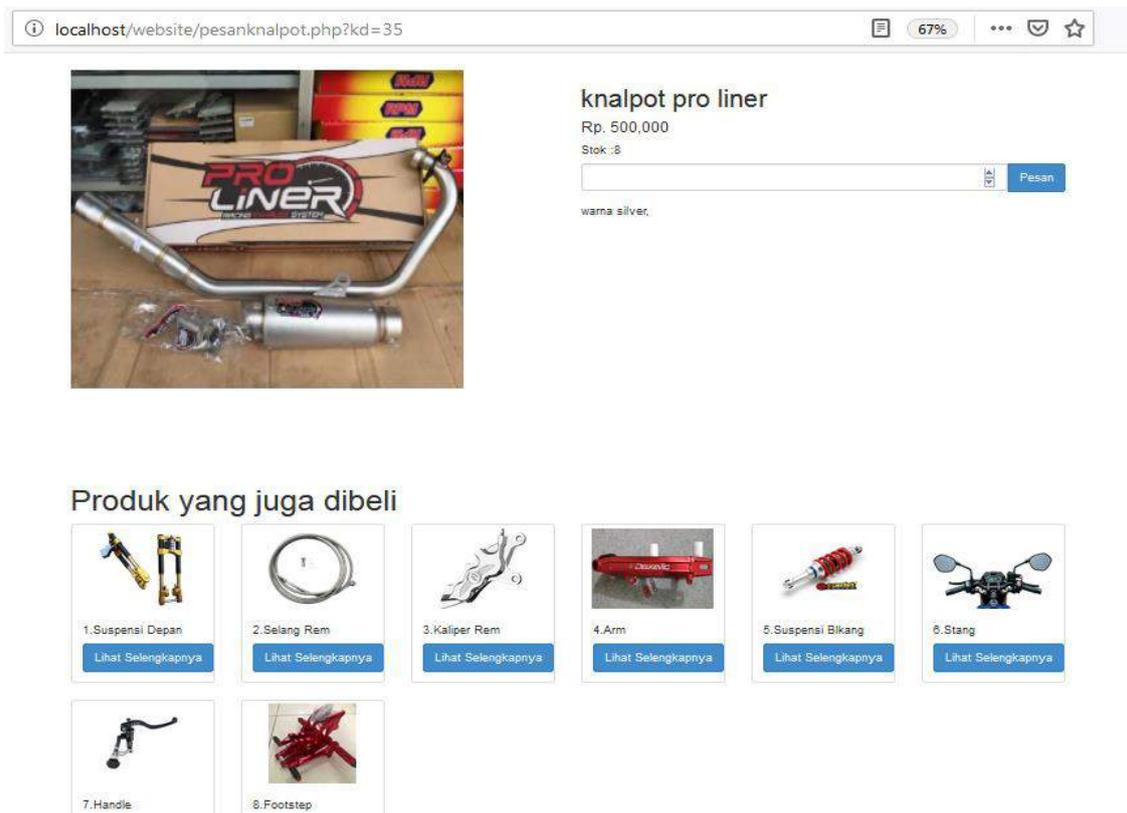
Tabel 5. Hasil Dari Perhitungan Support dan Confidence

No	If Antecedent then Consequent	Support	Confidence	Support vs Confidence
1	Knalpot-Suspensi Depan (SD)	23,4%	55%	0,1265
2	Knalpot-Stang	16,6%	39%	0,0647
3	Knalpot-Suspensi Belakang (SB)	16,9%	40%	0,0676
4	Knalpot-Handle	14,9%	35%	0,0525
5	Knalpot-Kaliper	19%	44%	0,0836
6	Knalpot-Selang	20,8%	49%	0,1029
7	Knalpot-Arm	17%	41%	0,0697
8	Knalpot-Footstep	15,5%	36%	0,0558

Tabel 6. Hasil Aturan Asosiasi

Peringkat	Produk	Support vs Confidence
1	Knalpot- Suspensi Depan (SD)	0,1265
2	Knalpot- Selang	0,1029
3	Knalpot- Kaliper	0,0836
4	Knalpot- Arm	0,0697
5	Knalpot- Suspensi Belakang (SB)	0,0676
6	Knalpot- Stang	0,0647
7	Knalpot- Handle	0,0525
8	Knalpot- Footstep	0,0558

Tampilan situs web untuk produk knalpot setelah dilakukan aturan asosiasi dengan mengurutkan dari nilai perkalian yang paling besar dapat dilihat pada Gambar 3.



Gambar 3. Tampilan Hasil Asosiasi untuk Produk Knalpot

Penelitian ini menemukan 11 (sebelas) asosiasi dari perhitungan yang telah dilakukan aturan asosiasi untuk semua produk. Hasil sebagai berikut:

1. Hasil yang didapat pada aturan pertama yaitu saat memilih produk knalpot maka akan muncul produk selang, kaliper, *arm*, *stang*, suspensi belakang, *handle* dan *footstep*.
2. Hasil yang didapat pada aturan kedua yaitu saat memilih produk *velg* maka akan muncul produk suspensi belakang, *footstep*, *arm*, *handle*, kaliper, suspensi depan, dan *gear*.
3. Hasil yang didapat pada aturan ketiga yaitu saat memilih produk *gear* maka akan muncul produk *handle*, *footstep*, *arm*, suspensi belakang, kaliper dan *velg*.
4. Hasil yang didapat pada aturan keempat yaitu saat memilih produk *stang* maka akan muncul produk kaliper, *handle*, knalpot, *footstep*, *arm*, selang, suspensi depan dan suspensi belakang.
5. Hasil yang didapat pada aturan kelima yaitu saat memilih produk suspensi depan maka akan muncul produk knalpot, selang, kaliper, *footstep*, *arm*, suspensi belakang, *velg* dan *stang*.
6. Hasil yang didapat pada aturan keenam yaitu saat memilih produk suspensi belakang maka akan muncul produk *arm*, *velg*, *footstep*, knalpot, suspensi depan, selang, kaliper, *handle*, dan *stang*.
7. Hasil yang didapat pada aturan ketujuh yaitu saat memilih produk *handle* maka akan muncul produk *gear*, *footstep*, *stang*, *velg*, *arm*, knalpot, kaliper, dan suspensi belakang.
8. Hasil yang didapat pada aturan kedelapan yaitu saat memilih produk kaliper maka akan muncul produk *stang*, knalpot, suspensi sepan, *footstep*, suspensi belakang dan *handle*.
9. Hasil yang didapat pada aturan kesembilan yaitu saat memilih produk selang maka akan muncul produk suspensi depan, knalpot, suspensi belakang, *arm* dan *stang*.
10. Hasil yang didapat pada aturan kesepuluh yaitu saat memilih produk *arm* maka akan muncul produk suspensi belakang, knalpot, suspensi depan, *gear*, *velg*, *handle*, *footstep*, selang, dan *stang*.
11. Hasil yang didapat pada aturan kesebelas yaitu saat memilih produk *footstep* maka akan muncul produk *handle*, *gear*, suspensi belakang, kaliper, *velg*, suspensi depan, *stang*, knalpot dan *arm*.

KESIMPULAN DAN SARAN

Analisis dan penerapan data *mining* berbasis *website* sudah dilakukan dengan perhitungan menggunakan teknik *asosiation rule* untuk menghasilkan keterhubungan antara produk *sparepart*. Penelitian ini berhasil menemukan sebelas aturan asosiasi dengan memberikan batas minimum *support* 13% dan minimum *confidence* 30%.

Pada penelitian ini, jumlah produk terdiri dari 11 *item sparepart* dan 1000 transaksi yang dianalisis. Pada penelitian lebih lanjut dapat menambahkan lebih banyak

produk dan transaksi yang dianalisis sehingga mendapatkan hasil asosiasi yang lebih bervariasi.

DAFTAR PUSTAKA

- [1] Gresshinta, “Analisis dan penerapan data mining untuk melihat pola permintaan konsumen minimarket Yogamart dengan menggunakan algoritma association rule,” *Skripsi*, Universitas Gunadarma, Depok, 2018.
- [2] M. Fauzy, K. Rahmat, dan I. Asror, “Penerapan metode association rule menggunakan algoritma Apriori pada simulasi prediksi hujan wilayah kota Bandung,” *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 2, no.2, 2016.
- [3] S. Despitara dan H. Tursina, “Analisis asosiasi pada transaksi obat menggunakan data mining dengan algoritma Apriori,” *Jurnal Sistem dan Teknologi Informasi*, vol. 1, no.1, 2016.
- [4] M. Cahyanti, M. Mujahidin, dan E. R. Swedia, “Penerapan algoritma Apriori association rule untuk analisa nilai mahasiswa di Universitas Gunadarma,” *SEMNAS TEKNO MEDIA*, vol. 5, no.1, 2017.
- [5] D. A. Nurdin, “Penerapan data mining untuk menganalisis penjualan barang dengan menggunakan metode Apriori pada supermarket sejahtera Lhoksumawe,” *Jurnal Techsi*, vol. 6, no. 1, hal. 134-155, 2015
- [6] Faisal, “Penerapan metode association rule mining untuk analisis dan implementasi teknik data mining dalam memprediksi strategi pemasaran produk Unilever,” *Jurnal INSTEK*, Vol. 3, No.1, 2018.
- [7] R. R. Rereug, “Penerapan data mining dengan memanfaatkan metode association rule untuk promosi produk,” *Jurnal Teknologi Rekayasa*, vol. 3, no.1, hal. 89 – 98, 2018.
- [8] T. D. Larose, *Discovering knowledge in data: introduction to data mining*. Hoboken, New Jersey: John Wiley & Sons, 2005.