

JURNAL TELEMATIKA MIKOM



ISSN: 2527-9033 (*online*)
2085-725X (*cetak*)

VOL. 15 No. 2
SEPTEMBER 2023



MAGISTER ILMU KOMPUTER
PROGRAM PASCA SARJANA TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

JURNAL TELEMATIKA MKOM

Pelindung

Rektor Universitas Budi Luhur
Direktur Riset dan PPM

Penanggung Jawab

Dr. Deni Mahdiana, S.Kom., M.M., M.Kom
(Dekan Fakultas Teknologi Informasi)

Editor in Chief

Dr. Utomo Budiyanto, S.Kom., M.Sc

Assistant Journal In Chief

Safitri Juanita, S.Kom, M.T.I

Associate (Handing) Editor :

Devit Setiono, S.Kom., M.Kom
Nidya Kusumawardhany S.Kom., M.Kom

Alamat Redaksi

Kantor Fakultas Teknologi Informasi
Jl. Ciledug Raya No.99, RT.10/RW.3, Petukangan Utara
Pesanggrahan, Kota Jakarta Selatan, DKI Jakarta 12260
e-mail : telematika.mkom@budiluhur.ac.id

Jurnal Telematika MKOM adalah jurnal yang diterbitkan oleh Program Pascasarjana Magister Ilmu Komputer Universitas Budi Luhur yang memuat hasil-hasil penelitian dengan topik-topik penelitian yang berasal dalam cakupan rumpun ilmu Komputer khususnya studi penelitian dasar dan terapan dalam Rekayasa Komputasi Terapan dan Teknologi Sistem Informasi

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadiran Tuhan Yang Maha Esa atas berkat, rahmat dan karunia-Nya sehingga Jurnal Ilmiah Telematika MKOM Volume 15 Nomor 2 September 2023 dapat terbit sesuai yang direncanakan.

Jurnal penelitian ini terbit sebagai bentuk kepedulian Universitas Budi Luhur (UBL) dalam meningkatkan mutu penelitian dan publikasi yang dilakukan oleh Dosen, mahasiswa ataupun praktisi di perguruan tinggi. Semoga Jurnal Telematika MKOM dapat menjadi referensi bagi para peneliti di Indonesia dan meningkatkan kualitas dari publikasi penelitian di Indonesia.

Seluruh personalia Jurnal Telematika MKOM mengucapkan terima kasih kepada penulis sebagai penyumbang artikel ilmiah, karena tanpa sumbangan artikel ilmiah dan penelitian dari penulis maka mustahil jurnal ilmiah Telematika MKOM dapat diterbitkan, terima kasih juga kepada semua pihak yang selalu memberikan dukungan kepada jurnal Telematika MKOM sehingga dapat hingga saat ini.

Terima kasih dan selamat membaca.

Jakarta, September 2023

Editor in Chief

Jurnal Telematika MKOM

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
Diagnosis Ispa Berbasis Web Dengan Metode <i>Forward Chaining</i> dan <i>Certainty Factor</i> Pada Sistem Pakar	
Rilo Anggoro Saputra, Hari Soetanto.....	48-56
Pengenalan Citra Wajah Pada Sistem Presensi Karyawan Menggunakan Metode K-Nearest Neighbor Dan Haversine Formula	
Muh. Mustafa Daniel Noya, Mardi Hardjianto	57-64
Prediksi Performa Pesepakbola Pada Kompetisi Liga Champions Eropa 2021 Menggunakan Algoritme Decision Tree Classifier	
Maret Ismoyo, Mohammad Syafrullah, Purwanto Purwanto, Painem Painem.....	65-74
Aplikasi Presensi Berbasis Internet of Things untuk Mengaktifkan Kamera Pada Raspberry PI	
Asep Cahyana, Imelda Imelda.....	75-85
Pendekatan Hybrid Pada Sistem Peringkat Teks Artikel Berita Bahasa Inggris Menggunakan <i>Natural Language Processing</i>	
Farah Raihanunnisa, Muhammad Arhami, Rahmad Hidayat.....	86-92

DIAGNOSIS ISPA BERBASIS WEB DENGAN METODE FORWARD CHAINING DAN CERTAINTY FACTOR PADA SISTEM PAKAR

Rilo Anggoro Saputra^{1*}, Hari Soetanto²

^{1,2}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, RT.10/RW.2, Petukangan Utara, Pesanggrahan, Jakarta, Indonesia
e-mail koresponden: 1911500146@student.budiluhur.ac.id

(received: 24/07/2023, revised: 14/08/2023, accepted: 18/08/2023)

Abstrak

Sistem pakar merupakan cabang dari kecerdasan buatan yang mensimulasikan sebuah penilaian atau pengambilan sebuah keputusan dari seorang yang memiliki keahlian dan pengalaman di bidang tertentu dengan tujuan untuk memecahkan sebuah permasalahan. Salah satu penerapan sistem pakar dapat dilakukan pada bidang kedokteran untuk mendiagnosa sebuah penyakit. Seperti pada kasus sistem pakar akan diterapkan untuk mendiagnosa Infeksi Saluran Pernafasan Akut (ISPA) merupakan salah satu penyakit yang menyerang saluran pernapasan yang meliputi hidung, tenggorokan, bronkus, dan juga paru-paru. ISPA terjadi karena disebabkan oleh virus serta bakteri, yang dimana gejala umum yang terjadi meliputi pilek, nyeri dalam tenggorokan, demam, sampai kelelahan. Di Indonesia ISPA bisa terjadi pada bayi, anak-anak, orang dewasa, sampai orang tua. Namun penyakit ini sering dianggap remeh bagi sebagian masyarakat Indonesia, pada tahun 2018 Kementerian Kesehatan Republik Indonesia (Kemenkes RI) mendata sebanyak 186.809 ribu masyarakat Jawa Barat mengalami permasalahan Infeksi Saluran Pernafasan (ISPA). Berdasarkan pertimbangan tersebut maka diperlukanlah sebuah aplikasi pendukung dengan menggunakan metode *forward chaining* dan *certainty factor* yang bertujuan agar dapat membantu para tenaga kesehatan untuk mendiagnosa secara cepat, sebagai tindakan preventif pada saat pengecekan tahap awal. Metode *Forward Chaining* melibatkan antara pembentukan aturan untuk suatu penyakit berdasarkan gejala yang ada, sementara penggunaan metode *Certainty Factor* untuk menghitung tingkat probabilitas dari setiap penyakit yang keluar berdasarkan gejala yang dipilih. Dengan menggabungkan kedua metode tersebut dapat menghasilkan nilai akurasi yang cukup akurat, studi kasus yang dilakukan pada penelitian ini menggunakan 10 data pasien dengan menghasilkan tingkat akurasi sebesar 90%.

Kata Kunci: *Certainty Factor*, *Forward Chaining*, Ispa, Sistem Pakar, Virus dan Bakteri.

WEB-BASED ISPA DIAGNOSIS WITH FORWARD CHAINING AND CERTAINTY FACTOR METHODS IN EXPERT SYSTEMS

Abstract

Expert system is a branch of artificial intelligence that simulates the judgment or decision-making of an individual with expertise and experience in a specific field, aiming to solve a particular problem. One application of expert systems can be found in the medical field, where they are used to diagnose diseases. For instance, in the case of an expert system applied to diagnose Acute Respiratory Infection (ARI), which is a disease affecting the respiratory tract, including the nose, throat, bronchi, and lungs. ARI can be caused by viruses and bacteria, and common symptoms include a runny nose, sore throat, fever, and fatigue. In Indonesia, ARI can affect people of all ages, from infants and children to adults and the elderly. However, this disease is sometimes underestimated by some Indonesian communities. In 2018, the Ministry of Health of the Republic of Indonesia reported that 186,809 individuals in West Java experienced respiratory infections. Considering these factors, there is a need for a supportive application using the forward chaining and certainty factor methods. The goal is to aid healthcare professionals in quickly diagnosing ARI as a preventive measure during the initial screening phase. The Forward Chaining method involves creating rules for a specific disease based on observed symptoms, while the Certainty Factor method calculates the probability level of each potential disease based on the selected symptoms. By combining these two methods, a reasonably accurate accuracy level can be achieved. In the case study conducted for this research, ten patient data were utilized, resulting in an accuracy level of 90%.

Keywords: *Ispe*, Expert System, *Forward Chaining*, *Certainty Factor*, Viruses And Bacteria

1. Pendahuluan

Di Indonesia penyakit Infeksi Saluran Pernafasan Akut (ISPA) merupakan salah satu penyakit yang biasa terjadi pada masyarakat di Indonesia yang mengganggu suatu proses pernafasan pada seseorang, infeksi ini umumnya terjadi karena adanya suatu organisme virus yang menyerang organ manusia seperti hidung, trakea, dan bahkan paru-paru. Berdasarkan data Riset Kesehatan Dasar (Riskesmas) pada tahun 2018 sebanyak 1 Juta lebih penduduk Indonesia mengalami infeksi pada saluran pernafasan. Dalam menangani penyakit ISPA, diagnosis yang tepat dan cepat sangatlah diperlukan untuk dapat menentukan jenis bahkan tingkat keseriusan dari gejala yang dialami dari seorang pengidap ISPA. Namun, untuk menentukan diagnosis yang akurat dan cepat dapat menjadi sulit, sebab gejala dari setiap penyakit yang ada seringkali mempunyai kemiripan. Oleh karena itu perlu adanya sistem yang dapat membantu seorang dokter atau tenaga kesehatan dalam mengambil suatu keputusan dari seorang pasien yang mengidap ISPA dengan tepat.

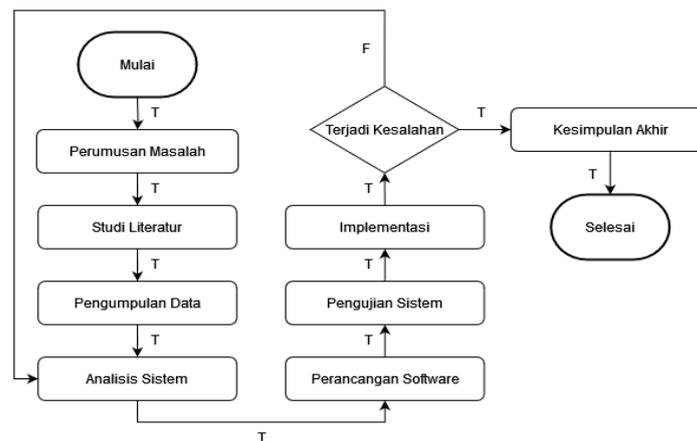
ISPA merupakan suatu permasalahan yang hampir terjadi pada setiap tahunnya. Pada dasarnya ISPA dapat berlangsung hingga 2 minggu lamanya [1]. ISPA merupakan peradangan akut yang terjadi pada saluran pernafasan atas maupun bagian bawah yang disebabkan oleh sebuah virus, bakteri, dan juga riketsia [2] Penyakit ISPA menjadi salah satu penyakit yang biasanya menular secara tidak disadari oleh kalangan masyarakat, yang dapat menyebabkan berbagai spektrum penyakit dari penyakit tanpa gejala atau infeksi ringan sampai infeksi parah sampai menyebabkan kematian [3]

Sistem pakar ini dibangun dengan menggabungkan 2 metode yaitu *forward chaining* dan *certainty factor*. *Forward chaining* merupakan sebuah metode pencarian dengan fakta yang diketahui sistem[4] Metode ini biasa dikenal dengan pelacakan kedepan yang digunakan dalam mengambil kesimpulan atau bahkan solusi berdasarkan fakta dan aturan-aturan (*rules*) yang ada untuk menghasilkan sebuah kesimpulan baru[5] Metode ini dimulai dari mengumpulkan *rules* awal atau sebuah informasi yang diberikan, lalu system akan menganalisa *rules* yang sesuai dengan fakta yang ada sesuai dengan basis pengetahuan. *Certainty Factor* merupakan sebuah metode yang dapat mendefinisikan kepercayaan seorang pakar untuk mengelola ketidakpastian dalam suatu aturan yang ada [6]. *Certainty Factor* telah dikembangkan oleh Shortliffe dan Buchanan pada tahun 1970an untuk MYCIN sebagai sebuah sistem pakar untuk mendiagnosa penyakit infeksi darah dan meningitis.

Berdasarkan studi literatur terhadap penelitian yang telah dilakukan oleh Teuku Feraldy Ramadhani dan kedua rekannya dalam penelitian menggunakan metode *forward chaining* dapat menghasilkan tingkat akurasi lebih tinggi sebesar 94%, dibandingkan dengan Wahyuni I. dan rekannya yang sebesar 90,91% [7] *Forward chaining* digunakan digunakan untuk membuat pengelompokan berbagai jenis gejala dalam setiap penyakit. Penelitian yang dilakukan oleh Subrianto Chandra dan kedua rekannya dalam menggunakan metode *certainty factor* dapat menghasilkan tingkat akurasi lebih besar yaitu 86,67% dibandingkan dengan dengan Hasan dan ketiga rekannya yang memiliki tingkat akurasi sebesar 80% [8]. *Certainty factor* digunakan untuk membuat suatu perhitungan probabilitas dari pembobotan nilai keragu-raguan seorang *user* dalam proses *anamnesa*, serta dihitungkan dengan bobot nilai seorang pakar. Dengan memanfaatkan data yang diberikan melalui pengetahuan seorang pakar dengan 22 gejala dan 10 penyakit, sistem pakar dapat memberikan sebuah keputusan yang cepat dalam melakukan diagnosis penyakit, hal ini dapat membantu dokter maupun paramedis dalam mendiagnosa seorang pasien yang mengidap ISPA.

2. Metode Penelitian

Dalam melakukan penelitian diperlukan beberapa rangkaian tahapan yang dilakukan, hal ini bertujuan agar dapat membuat suatu penelitian menjadi terarah. Berikut ini adalah penjabaran dari setiap tahapan metode yang dilakukan. Alur dari penerapan penelitian dijelaskan Pada Gambar 1.



Gambar 1. Alur Penelitian

2.1 Sistem Pakar

Sistem pakar adalah suatu cabang atau anak-anak dari *Artificial Intelligence* (AI) yang mulai dikembangkan pada pertengahan tahun 1960[9]. Sistem pakar yang baik ialah sebuah sistem yang dapat dirancang agar dapat menyelesaikan suatu permasalahan tertentu baik dari skala mudah sampai rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan seorang pakar [5].

2.2 Pengumpulan Data

Tahap pengumpulan data terbagi menjadi 2 bagian, pertama dengan mendapatkan data pasien yang memiliki diagnosa ISPA kepada klinik As-Shofa Purwakarta. Setelah itu, melakukan sebuah wawancara dengan seorang pakar untuk membahas penjelasan lebih akurat mengenai penyakit yang telah didapat serta gejala-gejala yang terdapat dalam sebuah penyakit.

2.3 Forward Chaining

Forward chaining adalah teknik mencari sebuah fakta yang diketahui dan kemudian membandingkannya dengan klausa IF dari aturan *IF-THEN*. Aturan diterapkan jika fakta ada dan memenuhi bagian IF, maka sebuah fakta baru akan dimasukkan ke dalam database setiap kali aturan diterapkan. Informasi masukan dapat berupa data, bukti, temuan atau gejala. Sedangkan kesimpulan dapat berupa tujuan, hipotesa, penjelasan atau diagnosa. Sehingga arah pencarian penalaran ke depan atau runut maju dimulai dari data menuju tujuan, dari bukti menuju hipotesis, atau dari gejala menuju diagnosa [7] dengan menggunakan logika "OR" dan "AND" untuk menghasilkan sebuah kesimpulan berdasarkan gejala yang dialami oleh seorang pasien.

2.4 Certainty Factor

Certainty Factor merupakan sebuah metode yang digunakan untuk menghadapi sebuah permasalahan yang tidak memiliki kepastian terhadap jawaban yang diterima oleh seorang pakar [10] *Certainty factor* digunakan untuk mendeteksi seberapa banyak ukuran kepastian berdasarkan fakta atau aturan yang ditemukan[11]. Serta untuk menghitung tingkat akurasi dari setiap penyakit. Berikut merupakan rumus dari metode *certainty factor* menurut David McAllister, yaitu:

1. Rumus ini dipakai apabila belum adanya nilai *certainty factor* pada setiap gejala.

$$CF(H,E) = MB(H,E) - MD(H,E) \quad (1)$$

2. CF dengan *premis*/gejala (ciri) tunggal (*single premise rules*).

$$CF_{\text{fold}}(n) = CF(P) * CF(U) \quad (2)$$

3. Apabila terdapat kaidah yang menyimpulkan serupa (*similarity concluded rules*) atau lebih dari satu *symptom* (gejala).

$$CF_{\text{combine}} = CF_{\text{fold}}(n) + CF_{\text{fold}}(n+1) * (1 - CF_{\text{fold}}(n)) \quad (3)$$

4. Lalu untuk menghitung persentase penyakit.

$$CF_{\text{persentase}} = CF_{\text{combine}} * 100 \quad (4)$$

Dengan ketentuan sebagai berikut:

- CF(H,E) = CF dari hipotesis (H) dipengaruhi oleh *simptom/gejala* atau disebutkan sebagai *evidence* (E). Nilai besaran CF berada dalam rentang 0 hingga 1. Nilai 0 mengarah pada ketidakpercayaan secara total, dan nilai 1 menunjukkan kepercayaan total.
- MB(H,E) = Besaran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesis (H) yang dipengaruhi oleh *simptom/gejala/evidence* (E).
- MD(H,E) = Besaran kenaikan ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis (H) yang dipengaruhi *simptom/gejala/evidence* (E).
- CF(P) = Nilai besaran CF dari pakar.
- CF(U) = Nilai besaran CF dari *user*/pengguna.
- CFold = Nilai dari hasil perkalian CF pakar dan CF *user*.
- CFcombine = Nilai hasil perkalian dari semua *simptom/gejala/evidence*(E).

3. Hasil Dan Pembahasan

Hasil penelitian menjelaskan terkait data yang digunakan pada saat pelaksanaan penelitian, studi kasus untuk menjelaskan bagaimana proses dari penerapan metode, hasil penerapan metode yang dipakai pada aplikasi yang dibangun, dan hasil pengujian serta keluaran tingkat akurasi dari aplikasi. Dengan menggunakan metode *Certainty Factor* dan *Forward Chaining* untuk mendiagnosa penyakit terhadap salah satu pasien, terdapat keluaran penyakit tuberculosis sebesar 100%.

3.1. Gejala Dan Penyakit

Berdasarkan hasil dari wawancara yang dilakukan dengan pakar dan menganalisis data pasien yang terjangkit ISPA ditentukan 10 penyakit pada Tabel 1 dan 22 gejala pada Tabel 2 pada penelitian ini.

Tabel 1. Data Penyakit

Kode	Nama Penyakit
D01	<i>Influenza</i>
D02	Sinusitis
D03	Laringitis
D04	Faringitis
D05	Pneumonia
D06	Asma
D07	Tuberculosis
D08	Bronkitis
D09	Tonsillitis
D10	<i>Covid-19</i>

Tabel 2. Data Gejala

Kode	Nama Gejala
G01	Hidung Tersumbat
G02	Pilek
G03	Bersin
G04	Pembengkakan di Bagian Leher
G05	Batuk Tak Berdahak
G06	Demam
G07	Batuk Bedahak
G08	Penurunan Berat Badan
G09	Sakit Kepala
G10	Nyeri Menelan
G11	Sesak Nafas
G12	Sakit Tenggorokan
G13	Nafsu Makan Menurun
G14	Berkurangnya Kemampuan Indra Penciuman

Kode	Nama Gejala
G15	Berkurangnya Kemampuan Indra Perasa
G16	Alergi
G17	Kondisi Lingkungan
G18	Hidung Berair
G19	Riwayat Penyakit Keluarga
G20	Batuk Lebih dari 2 Minggu
G21	Batuk Berdarah
G22	Pembengkakan pada Amandel

3.2. Aturan

Aturan keputusan berfungsi untuk mempermudah dalam merancang *knowledge base* dengan bentuk kondisi yang tersedia antara *IF – THEN*. Pada Tabel 3 menampilkan Aturan *Rule*.

Tabel 3. Aturan

No	Rule	THEN
1	If G1, G2, G3, G5, G6, G7, G9, G11, G14, G15, G16, G17, G18	G01
2	If G1, G2, G5, G6, G9, G11, G14, G15, G16, G17, G18, G20	G02
3	If G5, G6, G9, G10, G11, G12, G14, G15, G17, G20	G03
4	If G5, G6, G7, G9, G10, G11, G12, G15, G17, G20	G04
5	If G5, G7, G8, G11, G12, G17	G05
6	If G1, G2, G5, G7, G11, G12, G16, G17, G19	G06
7	If G5, G6, G7, G8, G11, G13, G17, G19, G20, G21	G07
8	If G5, G6, G7, G8, G9, G10, G11, G12, G13, G20	G08
9	If G4, G6, G8, G9, G10, G11, G12, G13, G20, G22	G09
10	If G1, G2, G3, G5, G6, G7, G9, G11, G12, G13, G14, G15, G17, G18	G10

3.3. Nilai Keyakinan Pakar

Berdasarkan hasil form yang telah diberikan terhadap pakar terbentuk nilai keyakinan pakar dilampirkan pada Tabel 4 pada penelitian ini.

Tabel 4. Nilai Keyakinan Pakar

Kode	D01	D02	D03	D04	D05	D06	D07	D08	D09	D10
G01	1	0.5	0	0	0	0.20	0	0	0	0.30
G02	1	0.5	0	0	0	0.50	0	0	0	0.20
G03	3	0	0	0	0	0	0	0	0	0.20
G04	0	0	0	0	0	0	0	0	0.70	0
G05	0.5	0.75	0.75	1	1	0.50	0.75	0.75	0	0.50
G06	0.75	0.25	0.50	0.50	0	0	0.75	0.50	0.75	1
G07	0.5	0	0	0.50	0.50	0.30	1	0.50	0	1
G08	0	0	0	0	0.50	0	1	0.50	0.50	0
G09	1	1	0.50	0.50	0	0	0	0.50	1	0.50
G10	0	0	1	1	0	0	0	0.50	1	0
...

3.4. Perhitungan *Certainty Factor*

Contoh untuk kasus untuk pasien pengidap gangguan ISPA dengan gejala yang dialami oleh seorang pasien adalah batuk tak berdahak (G005), demam (G006), batuk berdahak (G007), nyeri menelan (G010), sesak nafas (G011), nafsu makan menurun (G013), kondisi lingkungan (G017), Riwayat penyakit keluarga (G019), batuk lebih dari 2 minggu (G020), pembengkakan pada amandel(G021).

Tabel 5. Perhitungan Tuberkulosis

Gejala	MB-MD (CF1)	CF User (CF2)	Cfold (CF1 * CF2)	CFCcombined (CFold + CFciri*(1-CF))
G005	1.0	0.6	0.6	0.76
G006	0.5	0.8	0.4	0.88
G007	0.5	1	0.5	0.9088
G010	0.3	0.8	0.24	0.93616
G017	0.5	0.6	0.3	0.948928
G011	0.5	0.4	0.2	0.96322816
G019	0.7	0.4	0.28	0.988968448
G020	0.7	1	0.7	1
G021	1	1	1	1
G013	0.6	0.6	0.36	1

Pada Tabel 5 merupakan perhitungan yang sudah diinputkan berdasarkan gejala dan jawaban CF yang pasien rasakan. Jika dalam perhitungan manual maka akan menghasilkan hasil sebagai berikut:

Perhitungan 1

$$= 0.6+0.4 * (1-0.6)$$

$$= 0.6+0.4 * (0.4)$$

$$= 0.76$$

Perhitungan 2

$$= 0.76+0.5 * (1-0.76)$$

$$= 0.76 +0.5 * (0.24)$$

$$= 0.88$$

Perhitungan 3

$$= 0.88+0.24 * (1-0.88)$$

$$= 0.88+0.24 * (0.12)$$

$$= 0.9088$$

Perhitungan 4

$$= 0.9088+0.3 * (1-0.9088)$$

$$= 0.9088+0.3 * (0.0912)$$

$$= 0.93616$$

Perhitungan 5

$$= 0.93616+0.2 * (1-0.93616)$$

$$= 0.93616+0.2 * (0.06384)$$

$$= 0.948928$$

Perhitungan 6

$$= 0.948928+0.28 * (1-0.948928)$$

$$= 0.948928+0.28 * (0.051072)$$

$$= 0.96322816$$

Perhitungan 7

$$= 0.96322816+0.7 * (1-0.96322816)$$

$$= 0.96322816+0.7 * (0.3677184)$$

$$= 0.988968448$$

Perhitungan 8

$$= 0.988968448+1 * (1-0.988968448)$$

$$= 0.988968448+1 * (0.011031552)$$

$$= 1$$

Perhitungan 9

$$= 1+0.36 * (1-1)$$

$$= 1+0.36 * (0)$$

$$= 1$$

Perhitungan 10

$$= 1+0 * (1-1)$$

$$= 1+0 * (0)$$

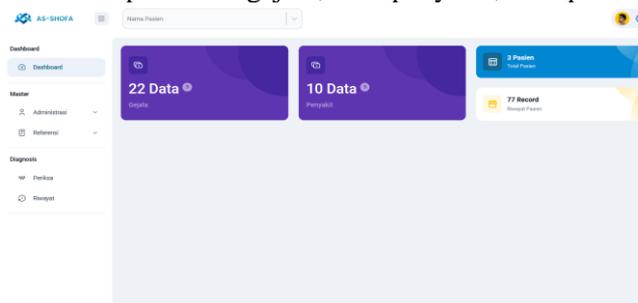
$$= 1$$

Dari hasil perhitungan CF gabungan maka nilai kepastian penyakit Tuberkulosis memilih nilai tertinggi dengan nilai kepastian sebesar 1 atau 100%.

3.5. Tampilan Aplikasi

3.5.1. Tampilan Menu Dashboard

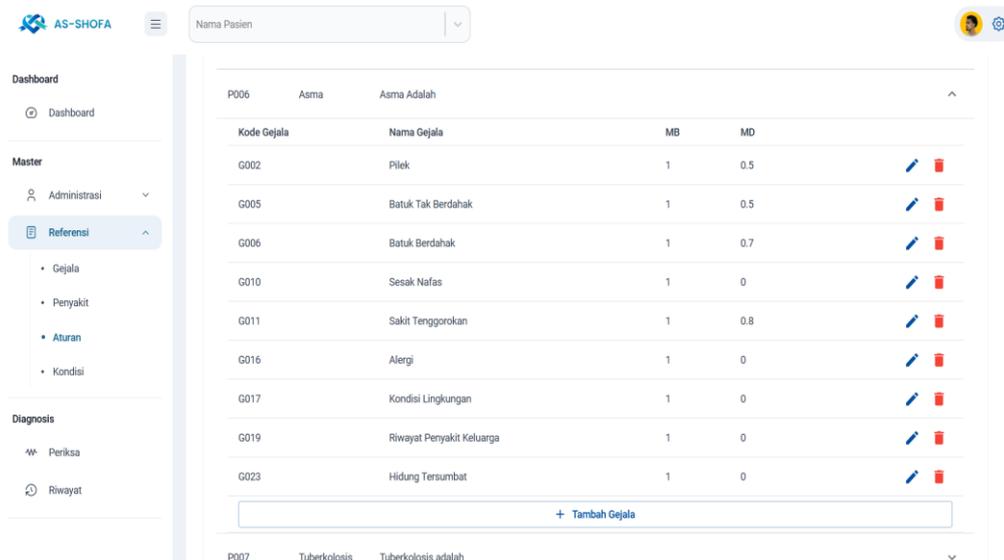
Menu *dashboard* pada Gambar 2 merupakan tampilan utama pada aplikasi sistem pakar, yang berisikan beberapa informasi seperti total gejala, total penyakit, total pasien, dan total Riwayat.



Gambar 2. Tampilan Menu Dashboard

3.5.2. Tampilan Menu Aturan

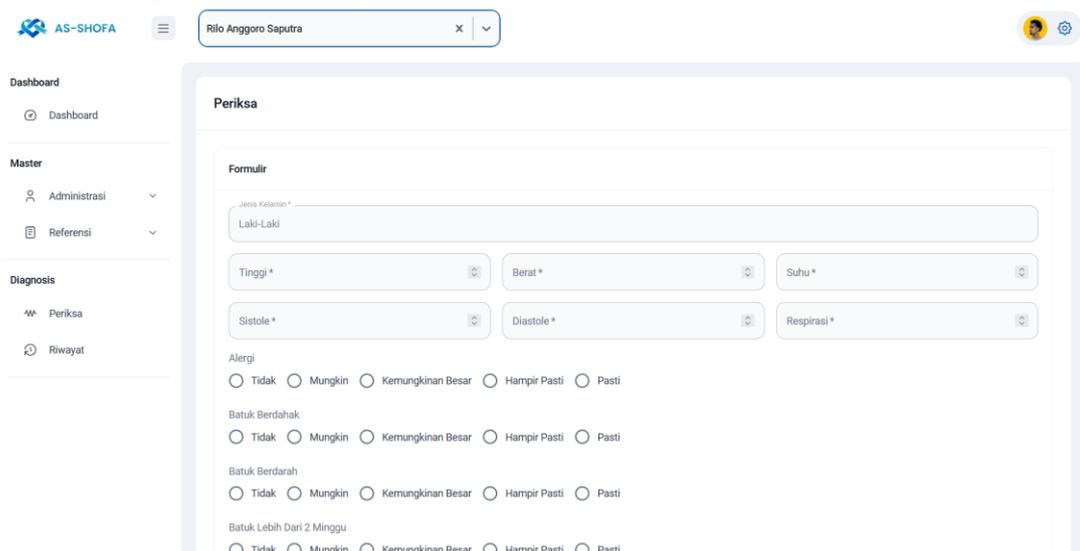
Menu Aturan pada Gambar 3 merupakan menu untuk mengoperasikan kedua metode, yaitu menginput gejala untuk penerapan metode *forward chaining* dan memasukan nilai MB dan MD untuk penerapan awal metode *certainty factor*.



Gambar 3. Tampilan Menu Aturan

3.5.3. Tampilan Menu Pemeriksaan

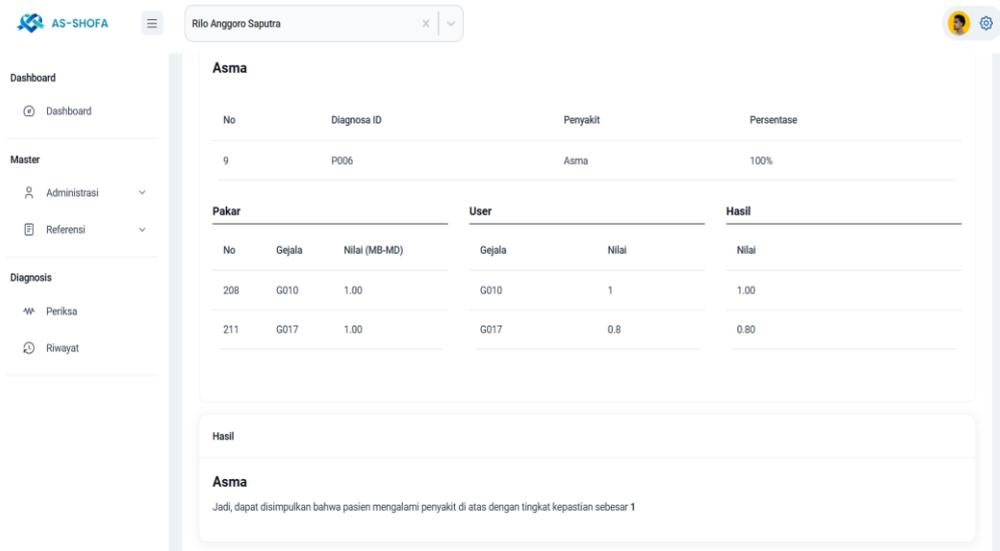
Menu pemeriksaan pada Gambar 4 merupakan menu untuk melakukan sebuah kajian terhadap pasien yang ingin diperiksa untuk melihat diagnosa awal sebagai langkah preventif sebelum melakukan pengecekan lebih lanjut oleh pakar.



Gambar 4. Tampilan Menu Pemeriksaan

3.5.4. Tampilan Hasil Pemeriksaan

Menu hasil pemeriksaan pada Gambar 5 merupakan tampilan pada saat pasien selesai melakukan pengecekan terhadap kajian-kajian yang ada.



Gambar 5. Tampilan Hasil Pemeriksaan

3.6. Hasil Pengujian

Pada Tabel 6 adalah hasil pengujian model aplikasi dengan metode *black box* oleh pakar. Terdapat 7 komponen dan 23 poin pengujian. Pengujian dilakukan dan mendapatkan hasil bahwa aplikasi dapat berjalan dengan cukup baik tanpa adanya kesalahan pada aplikasi.

Tabel 6. *Black Box Testing*

No	Komponen	Pengujian	Hasil yang diharapkan	Hasil
1	Dashboard	Mengklik menu yang terdapat pada <i>sidebar</i>	Dapat mengakses semua halaman yang ada melalui <i>sidebar</i>	Berhasil
2	Aturan	Mengklik menu aturan pada <i>sidebar</i>	Dapat mengakses halaman aturan	Berhasil
		Mengklik tombol “tambah gejala” dan menginput nilai MB dan MD	Dapat menambahkan gejala pada suatu penyakit sebagai keterhubungan antara gejala dan penyakit	Berhasil
3	Periksa	Mengklik menu periksa pada <i>sidebar</i>	Dapat mengakses halaman periksa	Berhasil
		Mengisi kajian terhadap pasien sesuai kondisi pasien	Dapat mengeluarkan hasil dari penyakit berdasarkan kajian yang diisi	Berhasil
...

Pengujian dilakukan dengan melakukan uji coba terhadap 10 pasien. Terdapat 1 data pasien yang hasilnya tidak sesuai dengan hasil yang dikeluarkan dari sistem. Berdasarkan data yang sesuai dengan hasil pendiagnosaan akhir yang dilakukan oleh seorang pakar maka dapat disimpulkan dengan menggunakan metode *forward chaining* dan *certainty factor* dapat memiliki tingkat akurasi sebesar 90%

4. Kesimpulan

Hasil penelitian menunjukkan bahwa metode *forward chaining* dan *certainty factor* sangat efektif untuk mendiagnosis ISPA berdasarkan 10 data penyakit dan 22 gejala, dengan tingkat akurasi mencapai 90% dari 10 data pasien. Pelayanan yang efisien tercapai karena dokter dapat melihat pengkajian awal yang telah dilakukan oleh seorang paramedis saat pasien bertemu dengan dokter.

Untuk masa depan, sistem pakar mendiagnosa ISPA diharapkan dapat dikembangkan lebih lanjut. Salah satu langkahnya adalah tidak membatasi pada satu jenis penyakit sajam tetapi dapat diimplementasikan untuk berbagai fasilitas kesehatan. Hal ini akan memberikan kemudahan dan efisiensi waktu dalam pelayanan. Selain itu, penelitian selanjutnya diharapkan dapat menggabungkan

metode lainnya atau mencari metode baru yang lebih unggul dalam menganalisis data untuk menyelesaikan suatu permasalahan. Dengan demikian, dapat menciptakan sistem pakar yang lebih efektif dan efisien dari yang telah di buat saat ini.

Daftar Pustaka

- [1] W. Budhyanti, Lisnaini, and M. Chandra, *Penanganan Infeksi Saluran Pernafasan Akut (ISPA) Pada Anak*, 1st ed. UKI Press, Amhhpta IKAPI, 2021.
- [2] L. Situmeang, *Pencegahan Dan Pengendalian Ispa*, 1st ed. Eureka Media Aksara, 2023.
- [3] N. Aprilla, E. Yahya, and P. Studi Sarjana Keperawatan Universitas Pahlawan Tuanku Tambusai, "Hubungan Antara Perilaku Merokok Pada Orang Tua Dengan Kejadian Ispa Pada Balita Di Desa Pulau Jambu Wilayah Kerja Puskesmas Kuok Tahun 2019", [Online]. Available: <http://journal.universitaspahlawan.ac.id/index.php/ners>
- [4] R. Hasanah, Muhasshanah, and F. Helmi, "Sistem Pakar Diagnosa Penyakit Saluran Pernapasan Menggunakan Metode Forward Chaining," *Jar's*, vol. 1, no. 1, 2022, [Online]. Available: <https://www.ejournalwiraraja.com/index.php/FT>
- [5] L. Abdul Hafiz and D. Andreswari, "Sistem Pakar Untuk Diagnosa Penyakit Tulang Berbasis Web Menggunakan Metode Forward Chaining," 2018. [Online]. Available: <http://ejournal.unib.ac.id/index.php/rekursif/>
- [6] K. Martiwi Sukiakhy, O. Aulia, J. Informatika, and F. Matematika dan Ilmu Pengetahuan Alam, "Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Gangguan Mental Pada Anak Berbasis Web," *Cyberspace: Jurnal Pendidikan Teknologi informasi*, vol. 6, 2022.
- [7] T. Feraldy Ramadhani, I. Fitri, and E. Tri Esti Handayani, "Sistem Pakar Diagnosa Penyakit ISPA Berbasis Web Dengan Metode Forward Chaining," *JOINTECS*, vol. 3, no. 1, 2020.
- [8] S. Chandra, Y. Yunus, and S. Sumijan, "Sistem Pakar Menggunakan Metode Certainty Factor untuk Estetika Kulit Wanita dalam Menjaga Kesehatan," *Jurnal Informasi dan Teknologi*, Dec. 2020, doi: 10.37034/jidt.v2i4.70.
- [9] Y. Yuliana, P. Paradise, and K. Kusriani, "Sistem Pakar Diagnosa Penyakit Ispa Menggunakan Metode Naive Bayes Classifier Berbasis Web," *CSRID (Computer Science Research and Its Development Journal)*, vol. 10, no. 3, p. 127, Mar. 2021, doi: 10.22303/csrid.10.3.2018.127-138.
- [10] F. Haikal Hasan and M. Syaifuddin, "Sistem E-Healthcare Untuk Mendiagnosa Penyakit Laringitis Dengan Menggunakan Metode Certainty Factor," *Jurnal CyberTech*, vol. 1, no. 1, pp. 195–206, 2018.
- [11] Susilo Hermawan, "Sistem Pakar Metode Forward Chaining dan Certainty Factor untuk Mengidentifikasi Penyakit Pertusis pada Anak," *Rangteknikjournal*, 2018.

Pengenalan Citra Wajah pada Sistem Presensi Karyawan Menggunakan Metode K-Nearest Neighbor dan Haversine Formula

Muh. Mustafa Daniel Noya^{1*}, Mardi Hardjianto²

^{1,2}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, RT.10/RW.2, Petukangan Utara, Pesanggrahan, Jakarta, Indonesia
e-mail koresponden: tafanoya@gmail.com

(received: 28/07/2023, revised: 14/08/2023, accepted: 15/08/2023)

Abstrak

Penerapan pengolahan citra wajah pada sistem presensi karyawan menjadi kebutuhan penting di era teknologi. PT. SISMEDIKA telah melakukan implementasi sistem presensi karyawan dengan menggunakan teknologi sidik jari (*finger print*). Namun, hal tersebut dapat menghambat karyawan jika terjadi cedera pada jari ataupun jari dalam keadaan basah. Oleh karena itu, penelitian ini membuat aplikasi presensi dengan pendeteksian wajah menggunakan metode *K-Nearest Neighbor*. Sistem ini juga dilengkapi teknologi *geotagging* dengan GPS dan pemanfaatan *Google Maps API* untuk memilih dan membatasi area presensi dan melakukan perhitungan jarak antara kedua titiknya dengan metode *Haversine Formula*. Penelitian ini bertujuan untuk menghindari faktor-faktor yang dapat menjadi penghambat bagi karyawan dalam melakukan presensi. Hasil penelitian menunjukkan metode *Haversine Formula* berjalan baik, karyawan tidak dapat presensi di luar area yang ditentukan. Penggunaan dataset foto individu dengan 10 label. Identifikasi wajah menggunakan K-NN mencapai akurasi 0,99784, presisi 0,991, dan recall 0,979.

Kata Kunci: Haversine Formula, K-Nearest Neighbor, Pengenalan Citra Wajah, Sistem Presensi Karyawan.

FACE IMAGE RECOGNITION IN EMPLOYEE ATTENDANCE SYSTEM USING THE K-NEAREST NEIGHBOR METHOD AND HAVERSINE FORMULA

Abstract

The application of facial image processing in employee attendance systems is an essential need in the technological era. PT SISMEDIKA has implemented an employee attendance system using fingerprint technology. However, this can hinder employees if there is an injury to the finger or the finger is wet. Therefore, this research uses the *K-Nearest Neighbor* method to make a presence application with face detection. This research makes a system that features *geotagging* technology with GPS utilizing the *Google Maps API* to select and limit the attendance area and calculate the distance between the two points using the *Haversine Formula* method. This research aims to avoid factors that can hinder employees' attendance. The results show that the *Haversine Formula* method works well; employees cannot take attendance outside the designated area by using individual photo datasets with ten labels. Face identification using K-NN achieved an accuracy of 0.99784, precision of 0.991, and recall of 0.979.

Keywords: Employee Attendance System, Face Image Recognition, Haversine Formula, K-Nearest Neighbor.

1. Pendahuluan

Sistem presensi merupakan aspek yang signifikan dalam kehidupan sehari-hari dimana sistem ini berperan sebagai salah satu metode pencatatan kehadiran seseorang, baik itu di tempat kerja, sekolah, maupun kampus [1]. Penulis lain juga mendefinisikan "Sistem Presensi" sebagai suatu sistem yang bertujuan untuk mengelola kehadiran karyawan dalam sebuah lembaga atau institusi. Sistem ini secara otomatis mencatat kehadiran karyawan dan menggunakan data kehadiran tersebut sebagai sumber laporan untuk memenuhi kebutuhan pengelolaan karyawan [2]. Pada umumnya sebuah sistem presensi dapat mencatat waktu kehadiran, waktu kepulangan, jumlah hadir, jumlah izin disertai keterangan izin

karena sakit atau kepentingan lainnya. Pada saat ini banyak perusahaan yang telah melengkapi kantornya dengan sistem presensi yang saling terintegrasi antara karyawan dan atasan dengan menerapkan teknologi *fingerprint* untuk sistemnya. Adapun kelemahan dari sistem presensi berbasis *fingerprint* ini yaitu sensor *finger* yang tidak dapat mendeteksi dengan baik pada kondisi-kondisi tertentu seperti ketika kondisi jari sedang basah, lecet karena luka dan hal lainnya. Hal ini tentu dapat mempersulit karyawan dalam melakukan presensi.

PT. Sistem Integrasi Medika (SISMEDIKA) adalah perusahaan teknologi informasi dengan fokus pengembangan terhadap sistem informasi rumah sakit, klinik dan laboratorium di berbagai kota yang tersebar di Indonesia agar dapat beroperasi secara lengkap dan terintegrasi. Pada saat ini, perusahaan tersebut sedang menjalani proyek pengerjaan sistem rumah sakit di RSPAD Gatot Subroto, Jakarta Pusat. Permasalahan terdapat pada perusahaan yang saat ini masih melakukan presensi dengan sistem dan teknologi *fingerprint* yang memiliki kelemahan-kelemahan yang telah disebutkan sebelumnya. Oleh karena itu, peneliti bermaksud untuk membuat sistem presensi karyawan berbasis web dengan menerapkan teknik pengenalan wajah pada setiap karyawan sebagai solusi dari permasalahan yang ada.

Program mengidentifikasi karyawan menggunakan pengenalan wajah, yaitu membandingkan pola gambar wajah karyawan dengan pola gambar foto wajah karyawan yang tersimpan di komputer. Pada penelitian ini pengenalan wajah diimplementasikan menggunakan metode *K-Nearest Neighbor* (KNN) dan perhitungan jarak antara dua titik dengan metode *Haversine Formula*. *K-Nearest Neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran (*neighbor*) yang jaraknya paling dekat dengan objek tersebut [3].

KNN akan mengelompokkan hasil perhitungan dengan data latih yang mempunyai kerabat terbanyak dalam nilai jangkauan yang ditentukan. Jarak antara data latih dan data uji dihitung menggunakan persamaan Euclidean [4]. Hasil dari penelitian terdahulu dengan judul “Implementasi *Face Recognition* pada Absensi Kehadiran Mahasiswa Menggunakan Metode *Haar Cascade Classifier*” [5] melakukan tahapan dan pengujian penelitian dengan metode *haar cascade* dalam proses pengenalan wajah yaitu sebanyak 25 data wajah dengan ketentuan jarak terdekat 30 cm dan jarak terjauh 150cm dan dihasilkan bahwa pengujian dengan status berhasil dikenali dengan benar sebanyak 19 orang dan status tidak berhasil dikenali dengan benar sebanyak 6 orang. Selanjutnya diperoleh hasil nilai akurasi sebesar 76%. Nilai akurasi tersebut dibandingkan dengan metode K-NN pada proses yang sama yaitu pengenalan wajah sebanyak 10 data wajah karyawan yang berbeda dengan jarak rata-rata 60cm mendapatkan hasil akurasi keseluruhan sebesar 99%.

2. Metode Penelitian

2.1 Data Penelitian

Data yang digunakan dalam penelitian ini berasal dari lembaga tempat riset ini dilakukan untuk program ini. Data tersebut merupakan informasi mengenai karyawan yang bekerja sebagai tenaga ahli.

Tabel 1. Data Karyawan

Nama Pegawai	Alamat	No. Telp
Niken	Jl Acondion B1 Q/12, Dki Jakarta	0854-555-507
Rico	Jl Pertanian Raya Komplek SLB A Pembina, Dki Jakarta	0859-555-688
Viko	Jl Tebet Tmr Raya 4, Dki Jakarta	0853-555-793
Laurenza	Jl P Jayakarta 35, Dki Jakarta	0878-555-96
Muthiara	Jl Raya Serdang F/31 RT 007/01, Dki Jakarta	0855-555-513
Riki	Jl Dr Susilo 1, Dki Jakarta	0852-555-120
Novinda	Jl Pinangsia Raya Glodok Harco B1 D/223, Dki Jakarta	0878-555-143
Andreas	Ged Sudirman Tower Kav-60, Jakarta	0878-555-907
Daniel	Jl Falatehan I 34, Jakarta	0838-555-690
Fahri	Jl Jagakarsa I 17, Dki Jakarta	0838-555-190

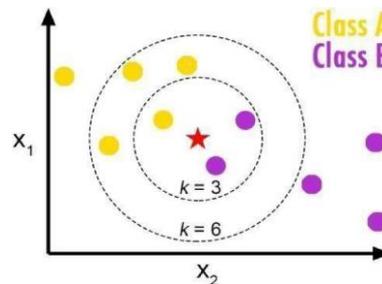
2.2 Penerapan Metode

Metode KNN terdiri dari dua tahap yaitu tahap pembelajaran (*training*) dan tahap klasifikasi atau pengujian (*testing*). Pada tahap *training*, algoritma ini hanya menyimpan vektor-vektor fitur dan

klasifikasi dari data *training*. Sedangkan pada tahap klasifikasi, fitur-fitur yang sama dihitung untuk data yang akan diuji (yang belum diklasifikasikan). Jarak antara vektor baru ini dengan seluruh vektor data pembelajaran dihitung dan beberapa k tetangga terdekat diambil. Perhitungan jarak ketetangga menggunakan algoritma *euclidean* sebagaimana ditunjukkan dalam persamaan 1.

$$euc = \sqrt{(a_1 - b_1)^2 + \dots + (a_n - b_n)^2} \quad (1)$$

Dimana $a = a_1, a_2, \dots, a_n$, dan $b = b_1, b_2, \dots, b_n$ mewakili n nilai atribut dari dua *record* untuk atribut dengan nilai kategori. Sebuah titik akan diprediksi jenisnya berdasarkan pada klasifikasi terbanyak dari *neighbor* di sekitarnya, ilustrasinya dapat dilihat pada Gambar 1 dibawah ini:



Gambar 1. Ilustrasi penggunaan nilai k pada metode KNN

Haversine formula adalah metode perhitungan jarak antara dua titik di bumi berdasarkan panjang garis lurus antara keduanya, dengan memperhitungkan kelengkungan bumi [5]. Formula *Haversine* terdapat pada persamaan 2.

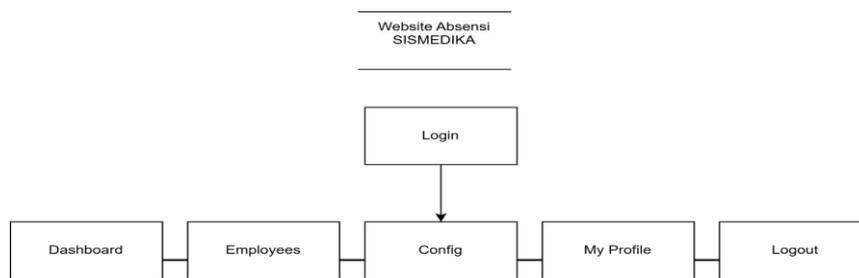
$$\begin{aligned} x &= (\text{lon}2 - \text{lon}1) \cdot \cos\left(\frac{\text{lat}1 + \text{lat}2}{2}\right) \\ y &= (\text{lat}2 - \text{lat}1) \\ d &= \sqrt{(x \cdot x) + (y \cdot y)} \cdot R \end{aligned} \quad (2)$$

Keterangan:

- $lat1$ = latitude awal
- lon = longitude awal
- $lat2$ = latitude tujuan
- $lon2$ = longitude tujuan
- x = longitude (lintang)
- y = latitude (bujur)
- d = jarak
- R = jari-jari bumi sebesar 6371 (km) 1 derajat = 0,0174532925

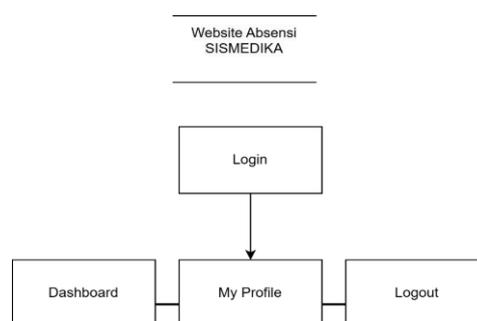
2.3 Rancangan Menu

Struktur menu di bawah ini merupakan rancangan menu website yang terbagi menjadi dua yaitu tampilan menu dari sisi *manajer* dan *employee*. Pada user *manager* berikut terdapat lima menu yang dapat digunakan. Setelah user login dengan status sebagai *manager* maka user akan mengakses tampilan menu *dashboard* sebagai menu *default*. Selanjutnya user dapat mengakses menu *employees* untuk melihat dan menambahkan user akun karyawan, menu *config*, *my profile* dan menu *logout* untuk keluar dari aplikasi.



Gambar 2. Rancangan Menu *Manager*

Selanjutnya, pada rancangan menu user karyawan terdapat tiga menu yang dapat digunakan. Setelah user *login* dengan status sebagai *employee* maka user akan mengakses tampilan menu *dashboard* sebagai menu *default*. User karyawan dapat mengakses menu *my profile* dan menu *logout* untuk keluar dari aplikasi.



Gambar 3. Rancangan Menu *Employee*

3 Hasil Dan Pembahasan

Beberapa tahapan metode yang digunakan dalam penelitian ini yaitu melakukan penentuan dan pengambilan data titik lokasi karyawan untuk melakukan presensi. Jika titik lokasi tervalidasi masuk dalam *range* jarak yang telah ditentukan maka langkah selanjutnya adalah karyawan harus menginputkan data wajah kedalam sistem sebagai validasi kedua untuk melakukan presensi. Setelah melewati kedua tahap tersebut maka karyawan tercatat hadir secara sistem dan dapat menggunakan sistem presensi tersebut sebagai pelaporan kegiatan di hari itu.

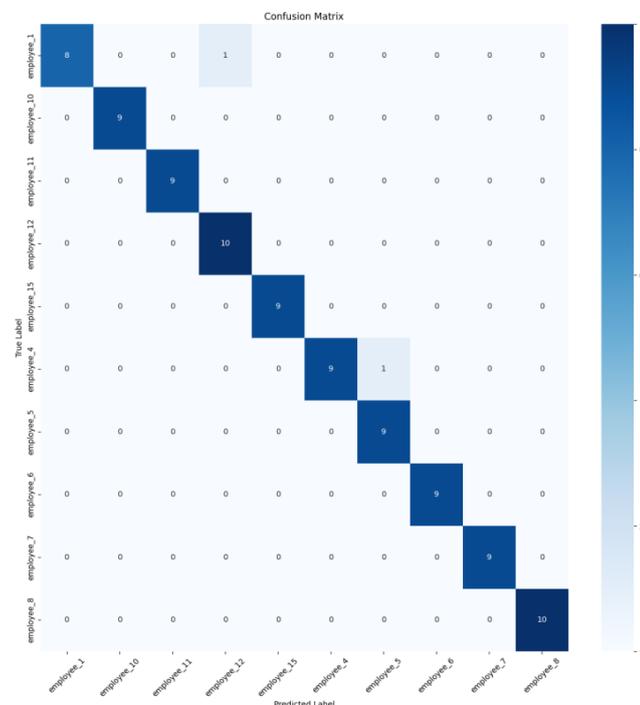
3.2 Pengujian K-Nearest Neighbor (K-NN)

Tabel 2. Pengujian KNN

No.	Nama Karyawan	Data Wajah Setelah Proses Training
1.	Fahri	
2.	Andreas	

No.	Nama Karyawan	Data Wajah Setelah Proses Training
3.	Laurenza	

Pada Tabel 2 terdapat sampel data berupa gambar hasil proses *training* dari metode KNN. Sebanyak 90 data uji dievaluasi dan hasil prediksinya direpresentasikan dalam bentuk *confusion matrix*. *Confusion matrix* adalah sebuah tabel yang mengindikasikan jumlah data uji yang terklasifikasi dengan benar dan jumlah data uji yang terklasifikasi secara salah [7]. *Confusion matrix* menggambarkan seberapa baik model KNN dalam mengklasifikasikan data ke dalam kategori yang benar. *Confusion matrix* pada Gambar 4 merupakan hasil perhitungan dari 10 label yang masing-masing label memiliki nilai tersendiri. Terdapat 10 nilai *True Positive* dengan range nilai 8-10 dan 2 nilai *True Negative* dengan nilai 1.



Gambar 4. *Confusion Matrix*

Selanjutnya, Tabel 3 merupakan hasil pengukuran akurasi keseluruhan, presisi maupun *recall* masing-masing label atau kelas dengan model kostum KNN. Tabel ini menghasilkan akurasi *f1-score* dengan nilai 98% dan akurasi *support* sebesar 93%, *macro average* untuk presisi, *recall*, *f1-score* dengan nilai 98%, *support* sebesar 93% serta *weighted average* untuk presisi, *recall*, *f1-score* dengan nilai 98%, *support* sebesar 93%.

Tabel 3. Hasil Pengukuran

Label	Precision	Recall	F1-Score	Support
Rico (employee_5)	1.00	1.00	1.00	9
Daniel (employee_1)	1.00	0.89	0.94	9
Andreas (employee_11)	1.00	1.00	1.00	9
Fahri (employee_12)	0.91	1.00	0.95	10
Viko (employee_6)	1.00	1.00	1.00	9
Muthiara (employee_8)	1.00	1.00	1.00	10
Niken (employee_4)	1.00	0.90	0.95	10

Label	Precision	Recall	F1-Score	Support
Riki (employee_15)	1.00	1.00	1.00	9
Laurenza (employee_7)	1.00	1.00	1.00	9
Novinda (employee_10)	1.00	1.00	1.00	9
accuracy			0.98	93
macro avg	0.98	0.98	0.98	93
weighted avg	0.98	0.98	0.98	93

Berdasarkan Tabel 3 dan Gambar 4 hasil *confusion matrix* pengujian *custom* model KNN, maka perolehan nilai akurasi, presisi dan *recall* menggunakan rumus perhitungan sebagai berikut:

3.2.1 Akurasi Model

Nilai Akurasi Keseluruhan didapat dengan total akurasi dibagi banyaknya k [8]. TP dan TN berdasarkan Gambar 5 merupakan diagonal utama dari *confusion matrix*. Formula akurasi terdapat pada persamaan 1.

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Akurasi = \frac{926}{928} = 0.99784$$

3.2.2 Presisi Model

Presisi merupakan perbandingan antara jumlah prediksi positif yang tepat dengan total hasil dari seluruh prediksi positif [9]. Formula presisi terdapat pada persamaan 2.

$$Presisi\ Keseluruhan = \frac{Total\ Presisi\ Seluruh\ Label}{Total\ Label} \quad (2)$$

$$Presisi = \frac{9.91}{10} = 0.991$$

3.2.3 Recall Model

Menurut Pendit, dalam pengertian *Recall* yang dikemukakan oleh Lancaster, merupakan rasio dari jumlah dokumen yang dapat ditemukan kembali melalui suatu proses pencarian informasi.[10]. Formula recall terdapat pada persamaan 3.

$$Recall\ Keseluruhan = \frac{Total\ Recall\ Seluruh\ Label}{Total\ Label} \quad (3)$$

$$Recall = \frac{9.79}{10} = 0.979$$

3.3 Pengujian Haversine Formula

Pengujian ini dilakukan dengan lokasi Wisma Bermis RSPAD Gatot Subroto dengan koordinat -6,177430936922497, 106,83566061526238 dan radius maksimal 20 meter. Hasil dari pengujian *blackbox* tahap ini dapat dilihat pada Tabel 4.

Tabel 4. *Black Box Haversine Formula*

No	Skenario	Titik Lokasi	Hasil Diharapkan	Kesimpulan Akhir
1	presensi di lokasi yang ditentukan	lat - 6,1777268, lon 106,8355965	sistem mengizinkan untuk melakukan pengujian tahap kedua	sesuai <i>valid</i>
2	presensi di luar lokasi yang ditentukan	lat - 6,1780842, lon 106,8343466	sistem tidak mengizinkan untuk melakukan pengujian tahap kedua dan memberikan pesan "You are too far from the attendance area"	sesuai <i>valid</i>

No	Skenario	Titik Lokasi	Hasil Diharapkan	Kesimpulan Akhir
3	melakukan presensi masuk	lat - 6,1777268, lon 106,8355965	sistem menerima dan menampilkan pesan "Berhasil mengisi presensi kehadiran!"	sesuai <i>valid</i>
4	melakukan presensi pulang	lat - 6,1777268, lon 106,8355965	sistem menerima dan menampilkan pesan "Berhasil mengisi presensi kepulangan!"	sesuai <i>valid</i>
5	melakukan presensi saat tidak ada jadwal	lat - 6,1777268, lon 106,8355965	sistem memberikan pesan "tidak ada presensi untuk hari ini"	sesuai <i>valid</i>

Berdasarkan hasil *blackbox testing* maka diperoleh nilai akurasi, presisi, *recall* sebagai berikut:

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN}$$

$$= \frac{5}{5}$$

$$akurasi = 1$$

Selanjutnya untuk nilai presisi sebagai berikut:

$$Presisi\ Keseluruhan = \frac{Total\ Presisi\ Seluruh\ Label}{Total\ Label}$$

$$= \frac{5}{5}$$

$$Presisi = 1$$

Selanjutnya untuk nilai *recall* sebagai berikut:

$$Recall\ Keseluruhan = \frac{Total\ Recall\ Seluruh\ Label}{Total\ Label}$$

$$= \frac{5}{5}$$

$$Presisi = 1$$

4 Kesimpulan

Berdasarkan penulisan penelitian yang telah dilakukan dapat disimpulkan bahwa penggunaan metode *Haversine Formula* dengan bantuan *google maps API* terbukti mampu mencegah karyawan untuk melakukan presensi di tempat yang tidak seharusnya. Begitu juga dengan algoritma metode *K-Nearest Neighbor* (KNN) dapat diimplementasikan dan berjalan dengan baik serta efektif dalam melakukan identifikasi citra wajah. Pada penelitian ini menghasilkan akurasi *f1-score* sebesar 0.98 dan akurasi *support* sebesar 93. Selain itu, dihasilkan juga nilai *macro avg* pada *precision* sebesar 0.98, nilai *macro avg* pada *recall* sebesar 0.98, nilai *macro avg* pada *f1-score* sebesar 0.98 dan nilai *macro avg* pada *support* sebesar 93. Nilai *macro avg* berguna ketika ada ketidak seimbangan jumlah sampel antara kelas-kelas yang ada.

Ucapan Terima Kasih

Peneliti mengucapkan banyak terimakasih kepada PT. Sistem Integrasi Medika, Ibu Herlina Suganda selaku Direktur Operasional, Bapak Andreas Brahmantiyo selaku Koordinator Lapangan, Muhammad Akbar Bagaskoro dan Ramdhan Dadan selaku programmer, Mutiara Panghurina dan Aini Cahyaning Putri selaku implementor serta senior dan teman-teman rekan kerja lainnya yang telah mengizinkan, memberikan dukungan dan kontribusi dalam menyediakan data dan informasi yang diperlukan untuk keperluan penelitian ini.

Daftar Pustaka

- [1] I. Kurniawan, T. Irfan, P. Studi Elektronika Industri, J. Teknik Elektro, P. Negeri Jakarta, and J. G. Siwabessy, "Sistem Presensi Berbasis Face Recognition," 2019, Diakses pada 24 Juli 2023.
- [2] F. Pratama and P. Putra, "Pengembangan Sistem Presensi Untuk Work From Home (WFH) dan Work From Office (WFO) Selama Pandemi Covid-19," 2022, Diakses pada 24 Juli 2023.
- [3] M. M. Baharuddin, H. Azis, and T. Hasanuddin, "Analisis Performa Metode K-Nearest Neighbor Untuk Identifikasi Jenis Kaca," *ILKOM Jurnal Ilmiah*, vol. 11, no. 3, pp. 269–274, Dec. 2019, doi: 10.33096/ilkom.v11i3.489.269-274, Diakses pada 24 Juli 2023.
- [4] M. M. Baharuddin, H. Azis, and T. Hasanuddin, "Analisis Performa Metode K-Nearest Neighbor Untuk Identifikasi Jenis Kaca," *ILKOM Jurnal Ilmiah*, vol. 11, no. 3, pp. 269–274, Dec. 2019, Diakses pada 24 Juli 2023, doi: 10.33096/ilkom.v11i3.489.269-274.
- [5] M. Kamal Syarifudin, R. Titi, K. Sari, and M. K. Syarifudin, "Perbandingan Metode Euclidean Distance dan Haversine Distance pada Aplikasi Sistem PPDB dan algoritma K-Means Untuk Menentukan Kebijakan Peraturan Zonasi," vol. 15, no. 4, pp. 1979–276, 2022, Diakses pada 24 Juli 2023, doi: 10.30998/faktorexacta.v15i4.14795.
- [6] S. H. Bariah, M. Irsad, and S. Putra, "Penerapan Metode Waterfall Pada Perancangan Sistem Informasi Pengolahan Data Nilai Siswa", Diakses pada 24 Juli 2023.
- [7] A. Indriani, "Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier." [Online], Diakses pada 24 Juli 2023, Available: www.bluefame.com
- [8] M. Rangga, A. Nasution, and M. Hayaty, "Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter," *Jurnal Informatika*, vol. 6, no. 2, pp. 212–218, 2019, [Online], Diakses pada 24 Juli 2023, Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- [9] K. L. Kohsasih, Z. Situmorang, and I. Artikel, "Analisis Perbandingan Algoritma C4.5 Dan Naïve Bayes Dalam Memprediksi Penyakit Cerebrovascular," *Jurnal Informatika*, vol. 9, no. 1, pp. 13–17, 2022, [Online], Diakses pada 24 Juli 2023, Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- [10] : Martin and L. Nilawati, "Recall dan Precision Pada Sistem Temu Kembali Informasi Online Public Access Catalogue (OPAC) di Perpustakaan," vol. 21, no. 1, pp. 77–84, 2019, Diakses pada 24 Juli 2023, doi: 10.31294/p.v20i2.

PREDIKSI PERFORMA PESEPAKBOLA PADA KOMPETISI LIGA CHAMPIONS EROPA 2021 MENGGUNAKAN ALGORITME *DECISION TREE CLASSIFIER*

Maret Ismoyo^{1*}, Mohammad Syafrullah², Purwanto³, Painem⁴

^{1,2,3,4}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, RT.10/RW.2, Petukangan Utara, Pesanggrahan, Jakarta, Indonesia
e-mail koresponden: maretismoyo@gmail.com

(received: 28/07/2023, revised: 14/08/2023, accepted: 15/08/2023)

Abstrak

Sepakbola merupakan salah satu olahraga paling terkenal dengan jumlah penggemar terbanyak di seluruh dunia. Selain itu, dalam dunia Sepakbola terdapat berbagai kompetisi yang diikuti oleh banyak pesepakbola profesional. Oleh karena itu, terdapat minat yang cukup besar dari berbagai pihak (internal maupun eksternal) untuk mengevaluasi atau mengukur performa para pesepakbola yang bertanding. Dalam mengevaluasi pemain, beberapa faktor kompleks dapat mempengaruhi staf pelatih dalam menilai pemain. Selain itu, pihak lain juga masih jarang melakukan penelitian terkait prediksi performa pesepakbola. Masalah prediksi tersebut karena membutuhkan pengetahuan dan penggunaan indikator data sesuai dengan topik penelitian tentang prediksi performa atlet sepakbola yang akan dilakukan. Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah sistem prediksi yang dapat memprediksi performa pemain dengan menggunakan metode *Decision Tree Classifier* dimana metode ini mengklasifikasikan data menjadi tiga kelas. Data yang digunakan sebanyak 876 data yang terbagi menjadi *data training* dan *testing* dengan tiga kelas performa pemain yaitu *Bad*, *Normal* dan *Good*. Hasil penelitian ini menunjukkan akurasi terbaik diperoleh dengan perbandingan data 80% : 20%, dengan menggunakan nilai parameter *max_depth* = 6 dan *max_leaf_nodes* = 4. Hasil penelitian ini menghasilkan nilai akurasi sebesar 97.73%. nilai presisi sebesar 97.76%. nilai recall sebesar 97.73%. nilai skor F1 sebesar 97.74% dan tingkat kesalahan 2.27%.

Kata kunci: *Data Mining*, *Decision Tree Classifier*, Prediksi, Performa Pesepakbola, Sepakbola.

PREDICTION OF FOOTBALLER PERFORMANCE IN THE 2021 EUROPEAN CHAMPIONS LEAGUE COMPETITION USING THE *DECISION TREE CLASSIFIER ALGORITHM*

Abstract

Football is one of the most famous sports with the most fans worldwide. In the world of football, there are various competitions in which many professional footballers participate. Therefore, there is considerable interest from various parties (internal and external) to evaluate or measure the performance of competing footballers. Several complex factors can influence the coaching staff in assessing players. In addition, other parties rarely conduct research related to predicting footballers' performance. The prediction problem requires knowledge and the use of data indicators by the research topic on predicting football athletes' future performance. Thus, this study proposed to predict player performance using the decision tree classifier method. This method classifies data into three classes. The data used is 876, divided into training and testing data with three classes of player performance, namely *Bad*, *Normal*, and *Good*. This study's results show the best accuracy using the split method with 80%: 20%, using the parameter values *max_depth* = 6 and *max_leaf_nodes* = 4. This study resulted in an accuracy value of 97.73%, a precision value of 97.76%, a recall value of 97.73%, an F1 score value of 97.74%, and an error rate of 2.27%.

Keywords: *Data Mining*, *Decision Tree Classifier*, *Footballer Performance*, *Prediction*, *Soccer*.

1. Pendahuluan

Olahraga merupakan salah satu unsur dalam kehidupan manusia yang jika diberdayakan melalui berbagai upaya yang sesuai, maka akan diperoleh hasil dan/atau prestasi yang mengagumkan [1]. Sepakbola adalah salah satu olahraga terkenal dengan jumlah penggemar terbanyak di dunia. Terdapat banyak kompetisi seperti Liga Champions Eropa, Piala Dunia, dan lain-lain yang disaksikan oleh penggemar dari seluruh dunia. Banyak hasil pengamatan dan juga analisis datang dari berbagai kalangan terkait performa baik para pemain maupun tim secara keseluruhan. Pada dasarnya masing-masing tim punya mekanisme analisis sendiri yang diterapkan guna mengetahui faktor penyebab kesuksesan, maupun kemunduran, kekalahan dan hal lainnya atas tim mereka. Namun berbagai penelitian mencoba menerapkan beberapa pendekatan ilmiah guna melakukan hal tersebut, khususnya untuk mengukur performa dari seorang pesepak bola.

Adapun pihak internal seperti pemilik klub serta staf kepelatihan berkeinginan untuk mengetahui pemain mana yang menjadi aktor keberhasilan timnya dalam memperoleh kemenangan, sehingga proses perekrutan dan rotasi pemain dapat berjalan sesuai dengan kebutuhan klub. Sedangkan pada pihak eksternal, ada fans klub dan media (penyiaran, *online* maupun cetak) yang ingin mengetahui pemain mana saja yang harus diapresiasi atau dikritik performanya selama suatu kompetisi sepakbola berlangsung [2].

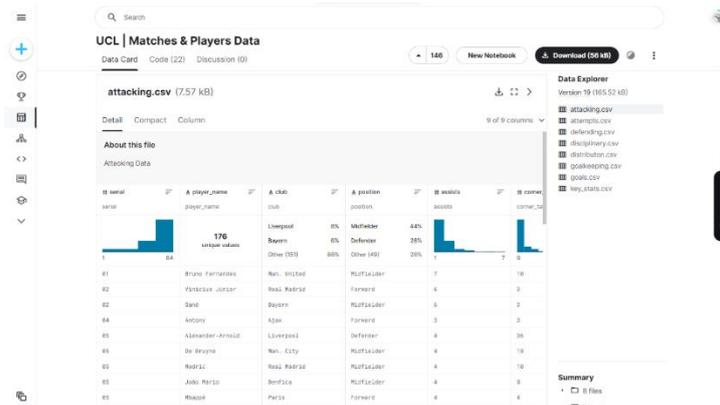
Umumnya, sepakbola adalah olahraga yang membutuhkan kerjasama tim, jadi terdapat faktor-faktor rumit didalamnya yang dapat mempengaruhi pengambilan keputusan seorang manajer/pelatih dalam menilai pemainnya, antara lain: faktor fisik, taktik, mental, teknik serta strategi permainan [3]. Maka dari itu, penting bagi staf kepelatihan untuk menggunakan prediksi yang disertai indikator performa yang mampu menggambarkan aspek-aspek spesifik dari kinerja pemain selama diatas lapangan. Mengingat salah satu tugas utama seorang manajer/pelatih yaitu mengevaluasi performa pemain, sehingga hasil feedback/umpan balik dari staf kepelatihan dapat diberikan kepada para pemainnya [4].

Terdapat beberapa metode yang telah diterapkan pada aktivitas penelitian terdahulu terkait dengan prediksi performa pesepakbola, antara lain dengan algoritme *Naïve Bayes*, *Random Forest* [5], *Support Vector Machine* (SVM), dan *K-Nearest Neighbor* (KNN) [6]. Pada penelitian tersebut, *dataset* yang digunakan adalah daftar statistik 150.000 pemain sepakbola pada gim *Football Manager 2017* yang diperoleh dari situs *Kaggle.com*. Sementara pada penelitian ini, *dataset* yang digunakan adalah 8 file berformat **.csv* yang berisi daftar statistik pemain sepakbola pada kompetisi Liga Champions Eropa 2021/2022 yang diperoleh dari situs *Kaggle.com*. Adapun pada penelitian ini menerapkan algoritme *Decision Tree Classifier* untuk dapat mengklasifikasikan performa pesepakbola ke dalam 3 kelas/*class* yaitu *Bad/Buruk*, *Normal/Biasa*, dan *Good/Bagus*. Adapun pemilihan nama 3 kelas/*class* ini mirip seperti yang digunakan dalam penelitian [7], yaitu *performer*, *moderate*, dan *failure*. Tujuan penggunaan 3 kelas pada penelitian ini yaitu agar model prediksi mampu menentukan kategori kelas/*class* untuk setiap data masukan yang belum diketahui *class*-nya [8].

2. Metode Penelitian

2.1. Data Penelitian

Dataset mentah yang digunakan pada penelitian ini adalah data statistik pesepakbola selama keikutsertaannya pada ajang Liga Champions Eropa musim 2021/2022 yang diperoleh dari situs *Kaggle.com* dengan kata kunci pencarian “UCL 2021-22 | Players Data” yang memuat 8 file berformat **.csv*. Adapun halaman situs *Kaggle.com* sebagai sumber *dataset* mentah dapat dilihat pada gambar 1.

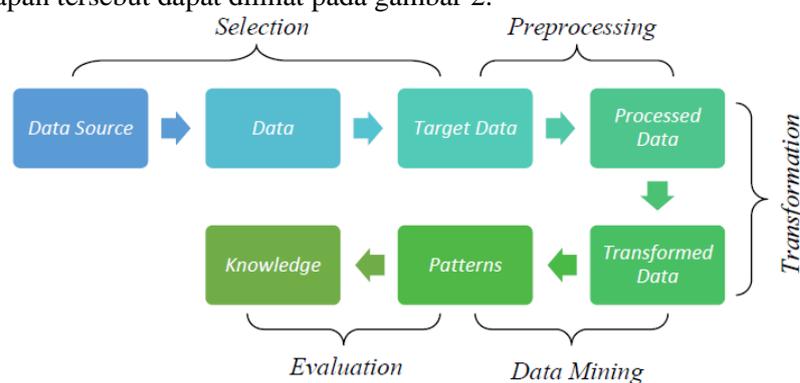


Gambar 1. Tampilan situs Kaggle.com

Berdasarkan pada gambar 1, dapat terlihat bahwa terdapat 8 file dengan format *.csv di sidebar bagian kanan, antara lain *attacking.csv* yang berisi 176 baris data, *attempts.csv* yang berisi 546 baris data, *defending.csv* yang berisi 627 baris data, *disciplinary.csv* yang berisi 584 baris data, *distribution.csv* yang berisi 608 baris data, *goalkeeping.csv* yang berisi 53 baris data, *goals.csv* yang berisi 183 baris data dan *key_stats.csv* yang berisi 747 baris data.

2.2. Penerapan Metode yang Digunakan

Penambangan data atau istilah lainnya *data mining* merupakan sub-bidang ilmu komputer yang memilah/menyortir sekumpulan data dalam jumlah besar untuk mengetahui pola atau tren, dalam rangka memecahkan masalah melalui analisis data [6]. *Data mining* menggunakan fungsi-fungsi statistik seperti varians, standar deviasi, dll; dimana fungsi-fungsi tersebut sangat berharga dalam mempelajari kehandalan serta relasi antar data [9]. Penambangan data sering dianggap identik dengan proses KDD (*Knowledge Discovery in Database*), dimana proses tersebut terdiri dari deretan langkah-langkah yang diterapkan untuk mengekstrak pola-pola unik yang menggambarkan pengetahuan dari data [10]. Di dalam penambangan data, terdapat 5 tahapan yang sering dilakukan dalam analisis data [11]. Adapun 5 tahapan tersebut dapat dilihat pada gambar 2.



Gambar 2. Tahapan Data Mining

2.2.1. Seleksi Data/Data Selection

Proses menyeleksi data dilakukan dengan tujuan untuk memilih atribut yang relevan pada data, sehingga proses analisis data untuk menentukan kategori performa pemain dapat dilakukan. Pada penelitian ini, didapatkan 10 atribut dan 1 target atribut yang dibutuhkan, antara lain: *position*, *pass_accuracy*, *pass_completed*, *crossing_accuracy*, *crossing_completed*, *freekicks_taken*, *match_played*, *goals*, *assists*, *distance_covered*, dan *performance* sebagai target atribut, serta diperoleh data sebanyak 603 baris data.

2.2.2. Pra-pemrosesan/*Preprocessing*

Pra-pemrosesan/*Preprocessing* merupakan suatu tahapan membersihkan data yang telah diseleksi dari duplikasi serta dari data tidak relevan/tidak konsisten. Adapun tahapan pra-pemrosesan yang dilakukan antara lain:

- a. Menghapus *outlier data* / data bias
 Tahap ini menjelaskan bahwa jumlah baris data yang sebelumnya berjumlah 603 data berubah menjadi 522 data. Hal itu disebabkan karena *dataset* ini terdapat *outlier* sebanyak 81 data bias. Oleh sebab itu, data bias tersebut harus dihilangkan supaya tidak mempengaruhi kinerja pada tahapan selanjutnya.
- b. Mengecek *missing value* / nilai kosong pada masing-masing atribut
 Sesuai *dataset* yang telah diperiksa, tidak ditemukan *missing value* karena pada keterangan angka *missing value* di semua atribut menunjukkan angka 0, yang berarti semua atribut pada *dataset* ini bersih dari data kosong/*missing value*.

2.2.3. Transformasi Data/*Data Transformation*

Proses transformasi data dilakukan untuk mengubah bentuk data kedalam format yang sesuai dengan algoritme klasifikasi yang diterapkan, dimana algoritme *Decision Tree Classifier* pada penelitian ini diperlukan transformasi data untuk meningkatkan performa model prediksinya. Adapun tahapan transformasi data yang dilakukan antara lain:

- a. *Label Encoding*
 Tahapan mengubah bentuk data yang sebelumnya berbentuk kategorikal, lalu diubah menjadi bentuk numerik dengan menggunakan fungsi *map()*. Hal ini bertujuan agar komputer mampu membaca data-data kategorikal kedalam bentuk integer. Adapun pada penelitian ini, tahapan encoding dilakukan pada atribut *performance* dan atribut *position* yang dapat dilihat pada tabel 1 dan tabel 2.

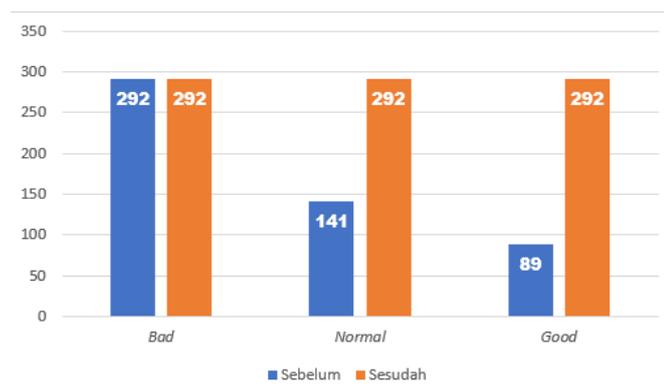
Tabel 1. Encoding atribut *performance*

Keterangan	Encoding
<i>Bad</i>	0
<i>Normal</i>	1
<i>Good</i>	2

Tabel 2. Encoding atribut *position*

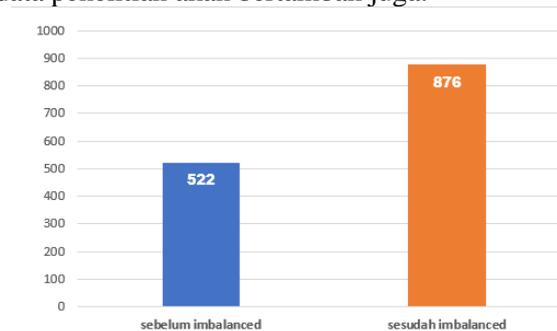
Keterangan	Encoding
<i>Goalkeeper</i>	0
<i>Defender</i>	1
<i>Midfielder</i>	2
<i>Forward</i>	3

- b. *Handling Imbalanced Data*
 Tahapan menyeimbangkan jumlah bobot pada atribut-atribut yang memiliki jumlah bobot yang sedikit, dengan cara *balanced* / diseimbangi dengan atribut yang memiliki jumlah bobot terbanyak.



Gambar 3. Perbandingan jumlah data setiap *class*

Berdasarkan gambar 3, terdapat 2 warna *barchart*/grafik batang yang dimana warna biru merupakan tanda warna untuk data yang belum melalui tahap *handling imbalanced data*, sedangkan yang warna *orange* merupakan tanda warna untuk data yang telah melalui tahap *handling imbalanced data*. Pada grafik berwarna biru, jumlah data pada kelas *bad*, *normal*, dan *good* berbeda-beda, sementara pada grafik berwarna orange, jumlah data pada kelas *bad*, *normal*, dan *good* sudah sama jumlahnya. Hal itu bisa terjadi karena adanya proses *over-sampling*, dimana kelas/*class* yang jumlah datanya lebih sedikit akan disamakan jumlah datanya dengan kelas/*class* yang paling banyak jumlah datanya dengan menggunakan data *dummy* pada modul SMOTENC (*Synthetic Minority Over-sampling Technique for Nominal and Continuous*) yang ada pada *library* sklearn. Dengan ditambahkan dengan data *dummy*, akan menyebabkan jumlah data penelitian akan bertambah juga.

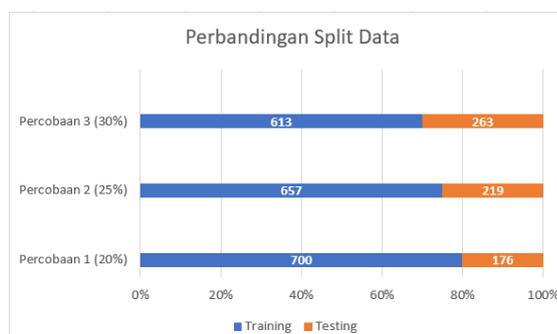


Gambar 4. Perbandingan jumlah data penelitian

Dapat dilihat pada gambar 4, *dataset* yang sebelumnya sebanyak 522 data saat sebelum ditambahkan data *dummy*, akan bertambah menjadi 876 data setelah ditambahkan data *dummy*.

c. *Splitting Data*

Tahapan pembagian data latih dan data uji yang akan digunakan pada proses pemodelan/modelling. Penggunaan data training dan data testing pada penelitian ini memiliki porsi yang berbeda-beda pada setiap pengujiannya. Hal ini bertujuan agar model prediksi yang telah dilatih/training dengan menggunakan data training yang lebih banyak porsinya dibandingkan dengan data testing, dapat melakukan proses klasifikasi secara tepat ketika menggunakan masukan/input data baru. Adapun porsi pembagian datanya dapat dilihat pada gambar 5.



Gambar 5. *Splitting* data penelitian

d. *Data Normalization*

Tahapan yang bertujuan mengubah *values* dari sebuah feature yang berbeda ukuran skala menjadi feature yang ukuran skalanya sama. Dengan dilakukannya tahapan ini, memungkinkan model memiliki performa dan kestabilan pemrosesan yang lebih baik. Pada penelitian ini,

penulis menerapkan tahapan ini dengan menggunakan *library StandardScaler()*. Adapun contoh penerapannya pada atribut di penelitian ini dapat dilihat pada Tabel 3.

Tabel 3. Data Normalization

Atribut	Sebelum	Sesudah
<i>position</i>	2	=> 0.19130654273419248
<i>pass_accuracy</i>	85	=> 1.4677626144204212
<i>pass_completed</i>	165	=> -0.6148431049757687
<i>cross_accuracy</i>	0	=> -0.8821824236284991
<i>cross_completed</i>	0	=> -0.7297656515595429
<i>freekicks_taken</i>	14	=> -0.2843575099627252
<i>match_played</i>	6	=> -0.512483455587928
<i>goals</i>	0	=> 1.4350583528242042
<i>assists</i>	0	=> -0.820562626039796
<i>distance_covered</i>	66.4	=> -0.7326085221454094

2.2.4.

Data/Data Mining

Pemodelan merupakan tahapan utama pada *data mining* setelah data penelitian selesai dibersihkan serta di transformasikan. Tujuan dari tahapan ini yaitu untuk menentukan algoritme yang cocok digunakan pada model prediksi. Dalam tahap pemodelan, dilakukan beberapa pengaturan *tuning* parameter pada setiap algoritme klasifikasi untuk meningkatkan nilai performanya. Pada penelitian ini, penulis menggunakan fungsi *GridSearchCV()* untuk memilih nilai *tuning* parameter terbaik secara otomatis. Adapun contoh pengaturan *tuning* parameter menggunakan *GridSearchCV()* untuk algoritme DTC dapat dilihat pada Tabel 4.

Tabel 4. Hasil *GridSearch* algoritme DTC

Hyperparameter Tuning	Hasil nilai Tuning terbaik
<i>max_depth</i>	6
<i>max_leaf_nodes</i>	6

Keterangan:

- Nilai *max_depth* yang digunakan penulis yaitu: 2-11
- Nilai *max_leaf_nodes* yang digunakan penulis yaitu: 2-11
- Semakin banyak parameter yang di *tuning*, maka semakin lama waktu komputasi yang dibutuhkan fungsi *Grid Search()* untuk menentukan nilai *tuning* terbaik.

2.2.5. Evaluasi/Evaluation

Setelah melakukan tahapan pemodelan, langkah selanjutnya terdapat tahapan untuk mengevaluasi kinerja dari algoritme klasifikasi yang akan digunakan sebagai model prediksi performa pesepakbola. Pada kasus klasifikasi, untuk mengukur kinerja algoritmenya, perlu melihat nilai-nilai *performance metrics* pada *confusion matrix*. Pada penelitian ini, penulis melakukan 3 kali percobaan dengan menggunakan 5 *performance metrics* pada *confusion matrix*, antara lain: *Accuracy*, *Precision*, *F1-score*, *Recall*, dan *Error rate*.

2.3. Rancangan Pengujian

Adapun pada penelitian ini, untuk proses pengujian yang diterapkan yaitu dengan menggunakan metode tabel *confusion matrix* disajikan dalam bentuk tabel matriks yang memvisualisasikan performa model klasifikasi pada serangkaian *data testing* yang nilai aktualnya diketahui [6]. Tabel *confusion matrix* dapat digambarkan seperti pada gambar 6.

		Aktual	
		1 (Positive)	0 (Negative)
Prediksi	1 (Positive)	TP (True Positive)	FP (False Positive)
	0 (Negative)	FN (False Negative)	TN (True Negative)

Gambar 6. Tabel *confusion matrix*

Penambahan

Keterangan:

- *True Positive* (TP) = jumlah nilai *true* yang diprediksi *true*
- *False Positive* (FP) = jumlah nilai *false* tetapi diprediksi sebagai *true*
- *True Negative* (TN) = jumlah nilai *false* yang diprediksi *true*
- *False Negative* (FN) = jumlah nilai *true* tetapi diprediksi sebagai *false*

Adapun beberapa *performance metrics* beserta persamaannya dibawah ini yang umum digunakan untuk menghitung kinerja model dalam kasus klasifikasi [12] yaitu:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

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

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

$$F1-score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

Berikut ini adalah penjelasan mengenai 4 *performance metrics* yang umum digunakan [13] :

- *Accuracy* = menunjukkan nilai performa model klasifikasi.
- *Precision* = menunjukkan efisiensi model klasifikasi pada setiap kategori.
- *Recall* = menunjukkan efisiensi model klasifikasi yang mampu diklasifikasikan secara penuh.
- *F1-score* = menunjukkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan.

3. Hasil dan Pembahasan

Langkah selanjutnya, dilakukan perhitungan performa dari model yang digunakan sebanyak 3 kali pengujian dengan porsi data latih dan data uji yang berbeda pada setiap pengujian, dengan menerapkan tabel *Confusion Matrix* dan 5 *performance metrics* yaitu: *accuracy*, *precision*, *recall*, *f1-score*, dan *error rate*. Adapun parameter yang digunakan pada ketiga pengujian ini ada 2, yaitu *max_depth* = 6 dan *max_leaf_nodes* = 4.

3.1. Pengujian 1 (80% Data Training:20% Data Testing)

Adapun pada pengujian pertama ini, *dataset* yang berjumlah 876 data hasil *reshape* akan dibagi menjadi 80% untuk data latih dan 20% untuk data uji, dimana rasio perbandingan untuk data latih sebanyak 700 data dan data uji sebanyak 176 data. Adapun tabel *Confusion Matrix* pada pengujian 1 dapat dilihat pada Tabel 5.

Tabel 5. *Confusion Matrix* pengujian 1

		Aktual			
		<i>Bad</i>	<i>Normal</i>	<i>Good</i>	
Prediksi	<i>Bad</i>	57 (TP) (cell 1)	1 (FN) (cell 2)	0 (FP) (cell 3)	Jml. Performa <i>Bad</i> = 58
	<i>Normal</i>	0 (FP) (cell 4)	57 (TP) (cell 5)	2 (FN) (cell 6)	Jml. Performa <i>Normal</i> = 59
	<i>Good</i>	0 (FP) (cell 7)	1 (FN) (cell 8)	58 (TP) (cell 9)	Jml. Performa <i>Good</i> = 59

Berikut ini adalah perhitungan 5 *performance metrics* pada pengujian 1.

$$Accuracy = \frac{TP}{TP + FP + FN} = \frac{172}{172 + 0 + 4} = \frac{172}{176} \times 100\% = 97.73\%$$

$$\begin{aligned}
 Precision &= \frac{Prec. Bad + Prec. Normal + Prec. Good}{3} \\
 &= \frac{100\% + 96.61\% + 96.67\%}{3} = 97.76\% \\
 Recall &= \frac{TP}{TP + FN} = \frac{172}{172 + 4} \times 100\% = 97.73\% \\
 F1-score &= \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 97.76\% \times 97.73\%}{97.76\% + 97.73\%} = 97.74\% \\
 Error rate &= 100\% - Accuracy = 100\% - 97.73\% = 2.27\%
 \end{aligned}$$

Setelah menghitung 5 *performance metrics*, selanjutnya dilakukan perbandingan performa model jika tidak menggunakan parameter seperti pada Tabel 6.

Tabel 6. Pengaruh *hyperparameter tuning* pada pengujian 1

<i>Hyperparameter Tuning</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Error rate</i>
Ya	97.73 %	97.76 %	97.73 %	97.74 %	2.27 %
Tidak	97.16 %	97.21 %	97.17 %	97.17 %	2.84 %

3.2. Pengujian 2 (75% Data Training: 25% Data Testing)

Adapun pada pengujian kedua ini, *dataset* yang berjumlah 876 data hasil *reshape* akan dibagi menjadi 75% untuk data latih dan 25% untuk data uji, dimana rasio perbandingan untuk data latih sebanyak 657 data dan data uji sebanyak 219 data. Adapun tabel *Confusion Matrix* pada pengujian 2 dapat dilihat pada Tabel 7.

Tabel 7. *Confusion Matrix* pengujian 2

		Aktual			
		<i>Bad</i>	<i>Normal</i>	<i>Good</i>	
Prediksi	<i>Bad</i>	72 (TP) (cell 1)	1 (FN) (cell 2)	0 (FP) (cell 3)	Jml. Performa <i>Bad</i> = 73
	<i>Normal</i>	0 (FP) (cell 4)	70 (TP) (cell 5)	3 (FN) (cell 6)	Jml. Performa <i>Normal</i> = 73
	<i>Good</i>	0 (FP) (cell 7)	2 (FN) (cell 8)	71 (TP) (cell 9)	Jml. Performa <i>Good</i> = 73

Berikut ini adalah perhitungan 5 *performance metrics* pada pengujian 2.

$$\begin{aligned}
 Accuracy &= \frac{TP}{TP + FP + FN} = \frac{213}{213 + 0 + 6} = \frac{213}{219} \times 100\% = 97,26\% \\
 Precision &= \frac{Prec. Bad + Prec. Normal + Prec. Good}{3} \\
 &= \frac{100\% + 95.89\% + 95.94\%}{3} = 97.28\% \\
 Recall &= \frac{TP}{TP + FN} = \frac{213}{213 + 6} \times 100\% = 97.26\% \\
 F1-score &= \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 97.28\% \times 97.26\%}{97.28\% + 97.26\%} = 97.27\% \\
 Error rate &= 100\% - Accuracy = 100\% - 97.26\% = 2.74\%
 \end{aligned}$$

Setelah menghitung 5 *performance metrics*, selanjutnya dilakukan perbandingan performa model jika tidak menggunakan parameter seperti pada Tabel 8.

Tabel 8. Pengaruh *hyperparameter tuning* pada pengujian 2

<i>Hyperparameter Tuning</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Error rate</i>
Ya	97.26%	97.28 %	97.26 %	97.27 %	2.74 %
Tidak	96.80%	96.83 %	96.80 %	96.81 %	3.20 %

3.3. Pengujian 3 (70% Data Training: 30% Data Testing)

Adapun pada pengujian ketiga ini, *dataset* yang berjumlah 876 data hasil *reshape* akan dibagi menjadi 70% untuk data latih dan 30% untuk data uji, dimana rasio perbandingan untuk data latih sebanyak 613 data dan data uji sebanyak 263 data. Adapun tabel *Confusion Matrix* pada pengujian 3 dapat dilihat pada Tabel 9.

Tabel 9. *Confusion Matrix* pengujian 3

		Aktual			
		<i>Bad</i>	<i>Normal</i>	<i>Good</i>	
Prediksi	<i>Bad</i>	86 (TP) (cell 1)	1 (FN) (cell 2)	0 (FP) (cell 3)	Jml. Performa <i>Bad</i> = 87
	<i>Normal</i>	0 (FP) (cell 4)	83 (TP) (cell 5)	5 (FN) (cell 6)	Jml. Performa <i>Normal</i> = 88
	<i>Good</i>	0 (FP) (cell 7)	2 (FN) (cell 8)	86 (TP) (cell 9)	Jml. Performa <i>Good</i> = 88

Berikut ini adalah perhitungan 5 *performance metrics* pada pengujian 3.

$$Accuracy = \frac{TP}{TP + FP + FN} = \frac{255}{255 + 0 + 8} = \frac{255}{263} \times 100\% = 96.96\%$$

$$Precision = \frac{Prec. Bad + Prec. Normal + Prec. Good}{3} = \frac{100\% + 96.51\% + 94.51\%}{3} = 97.01\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{255}{255 + 8} \times 100\% = 96.96\%$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 97.01\% \times 96.96\%}{97.01\% + 96.96\%} = 96.98\%$$

$$Error rate = 100\% - Accuracy = 100\% - 96.95\% = 3.04\%$$

Setelah menghitung 5 *performance metrics*, selanjutnya dilakukan perbandingan performa model jika tidak menggunakan parameter seperti pada Tabel 10.

Tabel 10. Pengaruh *hyperparameter tuning* pada pengujian 3

<i>Hyperparameter Tuning</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Error rate</i>
Ya	96.96 %	97.01 %	96.96 %	96.98 %	3.04 %
Tidak	96.58 %	96.65 %	96.59 %	96.59 %	3.42 %

4. Kesimpulan

Berdasarkan penelitian terkait performa pesepakbola yang telah dijalankan dengan menggunakan *algoritme Decision Tree Classifier*, dapat dibuatkan kesimpulan jika penggunaan *hyperparameter tuning* pada algoritme ini memiliki pengaruh yang besar pada nilai performa model prediksi yang digunakan. Adapun algoritme *Decision Tree Classifier* dapat diterapkan untuk studi kasus klasifikasi performa pesepakbola dengan nilai akurasi tertinggi pada pengujian 1 sebesar 97.73% bila menerapkan *hyperparameter tuning*. Jika dibandingkan kalau tidak menerapkan *hyperparameter tuning*, maka nilai akurasinya hanya sebesar 97.16%.

Daftar Pustaka

- [1] Ridwan M, "Kondisi Fisik Pemain Sekolah Sepakbola (SSB) Kota Padang," *J. Performa Olahraga*, vol. 5, pp. 65–72, 2020.
- [2] T. Kharrat, I. G. McHale, and J. L. Peña, "Plus–minus player ratings for soccer," *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 726–736, 2019, doi: 10.1016/j.ejor.2019.11.026.
- [3] Aditya Akbar Islami, "Keterampilan Psikologis Pemain Sepakbola Ditinjau dari Posisi Bermain dan Kelompok Usia," pp. 1–23, 2021.
- [4] R. Das, B. Jhahharia, and P. K. Das, "Prediction Model of Success and Failure in Football Competitions," *Int. J. Res. Pedagog. Technol. Educ. Mov. Sci.*, vol. 11, pp. 12–19, 2022, doi: 10.55968/uniaca.2022.11.2.3.
- [5] V. C. Pantzalis and C. Tjortjis, "Sports Analytics for Football League Table and Player Performance Prediction," 11th Int. Conf. Information, Intell. Syst. Appl. IISA 2020, 2020, doi: 10.1109/IISA50023.2020.9284352.
- [6] V. C. Pantzalis, "Sports Analytics Algorithms for Performance Prediction," *Int. Hell. Univ.*, no. December, 2019.
- [7] K. Bijukumar, "Artificial intelligence to predict sports performance," *Int. J. Phys. Educ. Sport. Heal.*, vol. 6, no. 3, pp. 238–241, 2019, [Online]. Available: www.kheljournal.com
- [8] E. H. Harahap, L. Muflikhah, and B. Rahayudi, "Implementasi Algoritma Support Vector Machine (SVM) Untuk Penentuan Seleksi Atlet Pencak Silat," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 10, pp. 3843–3848, 2018.
- [9] Y. L. Deep Madhvani, Ronik Moradiya, "Data Mining System for Selection of Best Basket Ball Team," *Int. Res. J. Eng. Technol.*, vol. 8, no. 6, pp. 3405–3407, 2021.
- [10] G. Theron, "The use of Data Mining for Predicting Injuries in Professional Football Players," *duo.uio.no*, 2020.
- [11] P. Koudoumas, "Sports Analytics Algorithms for Performance Prediction," *Int. Hell. Univ.*, no. January, 2021.
- [12] Y. Min, Y. Yin, and K. Khin, "Comparing the Performance of Machine Learning Algorithms for Human Activities Recognition using WISDM Dataset," *Int. J. Comput.*, vol. 38, pp. 61–72, 2020.
- [13] P. Pugsee and P. Pattawong, "Football match result prediction using the random forest classifier," *ACM Int. Conf. Proceeding Ser.*, pp. 154–158, 2019, doi: 10.1145/3358528.3358593.

APLIKASI PRESENSI BERBASIS *INTERNET OF THINGS* UNTUK MENGAKTIFKAN KAMERA PADA RASPBERRY PI

Asep Cahyana^{1*}, Imelda Imelda²

^{1,2}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, RT.10/RW.2, Petukangan Utara, Pesanggrahan, Jakarta, Indonesia
e-mail koresponden: asepcahyana48@gmail.com

(*received*: 04/08/2023, *revised*: 15/08/2023, *accepted*: 21/08/2023)

Abstrak

Salah satu penilaian kinerja karyawan bagi Perusahaan adalah melalui presensi. Saat ini, berbagai alat presensi menjamur, salah satunya adalah fitur presensi dengan menggunakan pengenalan wajah. Namun, terdapat kelemahan pada alat presensi berbasis wajah, yaitu proses validasi pengenalan wajah yang kurang akurat sehingga menyebabkan hasil presensi yang tidak valid. Oleh karena itu, penelitian ini mengusulkan sebuah model presensi berbasis Internet of Things (IoT) yang diharapkan dapat mengurangi tingkat kesalahan pendeteksian. Alat yang digunakan dalam penelitian ini adalah sensor berbasis RFID. Setiap model wajah karyawan dikorelasikan dengan RFID dan dapat digunakan sebagai presensi setiap karyawan. Selain itu, untuk menangani penyebaran virus COVID-19, ditambahkan juga sensor suhu sehingga karyawan yang memiliki suhu tubuh di atas 37,5 C tidak dapat melakukan presensi. Selain itu, notifikasi kehadiran akan dikirimkan ke perangkat Android karyawan. Parameter yang digunakan dalam pengujian adalah keberhasilan presensi menggunakan RFID dan face recognition dengan waktu pemrosesan kurang dari 2,5 detik. Hasil dari penelitian ini menunjukkan bahwa sistem dapat berjalan dengan baik, dengan persentase keberhasilan transaksi mencapai 90%.

Kata kunci: Aplikasi Presensi, Android, *Internet Of Things* (IoT), RFID, Sensor Infrared.

ATTENDANCE APPLICATION BASED ON INTERNET OF THINGS (IOT) FOR ACTIVATE CAMERAS ON RASPBERRY PI

Abstract

One of the employee performance assessments for the Company is through attendance. Currently, various attendance tools are proliferating, one of which is the attendance feature using face recognition. However, there are weaknesses in face-based attendance tools, namely the inaccurate face recognition validation process, which causes invalid attendance results. Therefore, this research proposes an Internet Of Things (IoT)-based attendance model expected to reduce the detection error rate. The tool used in this research is an RFID-based sensor. Each employee's face model is correlated with an RFID and can be used as each employee's presence. In addition, to deal with the spread of the COVID-19 virus, a temperature sensor is also added so that employees who have a body temperature above 37.5 C cannot take attendance. In addition, the attendance notifications will sent to employees' Android devices. The parameters used in testing are the success of attendance using RFID and face recognition with a processing time of less than 2.5 seconds. The results of this study show that the system can run well, with the percentage of successful transactions reaching 90%.

Keywords: Android, Internet Of Things (IoT), Infrared Sensor, Presence Application, RFID.

1. Pendahuluan

Perkembangan teknologi dan jaringan internet telah memungkinkan semakin banyak perangkat untuk terhubung secara online. Hal ini membuka peluang baru dalam berbagai bidang, termasuk transportasi, kesehatan, manufaktur, pertanian, dan banyak lagi. Salah satu teknologi yang memanfaatkan berkembangnya jaringan komputer ialah *Internet of Things* (IoT). Menurut [1] *Internet of Things* (IoT) merupakan suatu konsep dimana suatu benda atau objek ditanamkan dengan teknologi seperti sensor dan perangkat lunak untuk keperluan komunikasi, kontrol, koneksi dan pertukaran informasi dengan perangkat lain selama masih terhubung dengan internet. IoT memungkinkan

pengumpulan data yang luas dan real time secara otomatis yang dapat memberikan manfaat yang signifikan dalam hal efisiensi, produktivitas, dan kenyamanan.

Perkembangan IoT dapat dimanfaatkan, salah satunya dalam sistem presensi pegawai. Fungsi perangkat presensi berbasis IoT mempermudah pegawai dalam mengetahui status presensinya. Pegawai tidak perlu khawatir jika status presensinya belum tersimpan. Perangkat IoT memberikan notifikasi ke handphone pegawai sesaat setelah melakukan transaksi presensi [2].

PT. Rekabio memiliki kebutuhan untuk membuat perangkat presensi yang dapat mengirimkan notifikasi ke pegawai secara real time dan mengirimkan log presensi ke server secara aman. Belajar dari pengalaman saat pandemi Covid-19 dimana diperlukan alat pengecek suhu tubuh pegawai sehingga pegawai yang memiliki suhu tubuh tinggi tidak diperbolehkan untuk bekerja dikantor [3]. Maka dibuatlah alat presensi berbasis RFID yang dilengkapi dengan sensor pembaca suhu.

Pada penelitian sebelumnya [4], [5] dan [6] menggunakan RFID sebagai presensi. Penelitian [2] dan [7] menggunakan face recognition sebagai presensi. Penelitian ini mengusulkan penggabungan teknologi RFID (*Radio Frequency Identification*) [4] dan sensor suhu Infrared pada perangkat Raspberry Pi berbasis *Internet of Things (IoT)* [8]. Teknologi ini dipilih karena RFID memiliki id yang bersifat unik sehingga dapat digunakan sebagai pengenalan. Selain itu karena berbasis sensor gelombang elektromagnetis sehingga dapat di gunakan tanpa perlu bersentuhan langsung dengan alat dapat mengurangi potensi penyebaran virus Covid-19 [5]. Selain itu ditambahkan pula pengenalan wajah sebagai verifikasi tambahan untuk menghindari penggunaan presensi oleh pegawai lain.

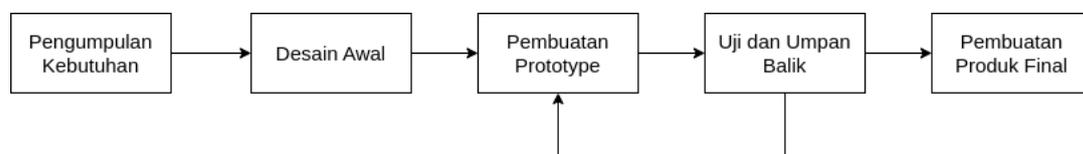
Pengiriman data dari perangkat IoT ke server ini menggunakan tunnel VPN. Tunnel VPN digunakan untuk mengamankan pengiriman data [9]. Tunnel VPN ini juga sebagai private network yang memungkinkan server untuk melakukan koneksi ke perangkat IoT melalui internet tanpa perlu ip publik statis pada setiap perangkat [10] [11].

Data presensi menentukan status kehadiran pegawai yang berkaitan dengan upah dan penilaian. Kesalahan presensi dapat menyebabkan kesenjangan diantara pegawai. Maka dibutuhkan alat presensi yang memiliki tingkat akurasi yang tinggi dengan tingkat kesalahan seminim mungkin. Dengan adanya penelitian ini diharapkan dapat menjadi solusi bagi permasalahan tersebut.

2. Metode Penelitian

2.1. Tahapan penelitian

Pada penelitian ini menggunakan metode prototipe. Metode prototipe adalah cara pendekatan dalam pengembangan perangkat lunak atau sistem. Dalam metode ini, kita membuat versi awal atau model sederhana dari produk yang ingin dibuat dan mengujinya sebelum mengembangkan versi final yang lebih lengkap. Tahapan penelitian terdapat pada Gambar 1.



Gambar 1. Tahapan Penelitian

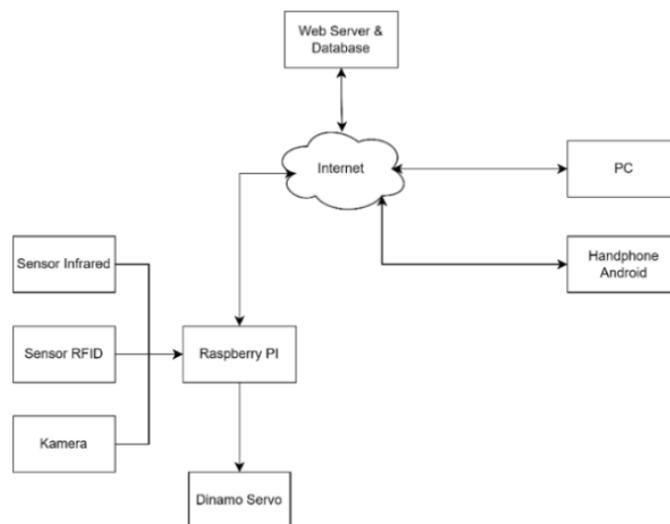
2.2. Pengumpulan Kebutuhan

Pada tahap pengumpulan kebutuhan ditentukan apa saja kebutuhan fungsionalitas dari perangkat yang akan dibuat. Terdapat beberapa kebutuhan yaitu:

- Perangkat dapat digunakan absen menggunakan RFID tag dan pengenalan wajah serta terdapat sensor suhu yang mengecek suhu tubuh pegawai.
- Data transaksi absen dapat dikirim ke server melalui jaringan internet.
- Pengguna mendapatkan notifikasi sesaat setelah melakukan presensi

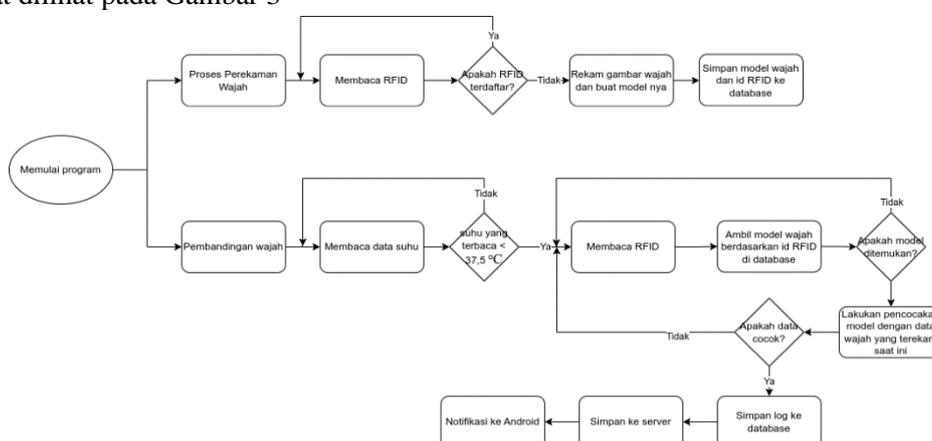
2.3. Desain Awal

Setelah kebutuhan terkumpul, langkah selanjutnya adalah merancang konsep awal sistem. Sistem dibuat dalam mock-up yang sederhana. Sistem terdiri dari beberapa bagian yaitu bagian input, bagian pemrosesan dan bagian output. Pada bagian input terdapat dua sensor yaitu sensor Infrared dan sensor RFID. Suhu tubuh yang dideteksi oleh sensor Infrared akan memicu diaktifkannya sensor RFID dan kamera sehingga pegawai dapat melakukan presensi. Data yang didapat dari sensor RFID dan kamera kemudian diproses oleh Raspberry PI, hasilnya disimpan di database lokal dan kemudian disimpan ke database server. Selain itu Raspberry Pi akan menggerakkan motor servo untuk membuka pintu berdasarkan status presensi. Pada bagian output adalah berupa notifikasi yang diterima user melalui aplikasi Android, serta data transaksi yang dapat dilihat pada aplikasi web server. Untuk lebih jelasnya dapat dilihat pada Gambar 2.



Gambar 2. Perancangan Sistem

Pada Gambar 2 terdapat tiga sistem terpisah yang dihubungkan melalui jaringan internet yaitu Raspberry PI sebagai perangkat IOT, web server sebagai tempat penyimpanan dan pemrosesan data serta perangkat pc atau handphone sebagai alat pengakses data. Sedangkan alur data dari rancangan diatas dapat dilihat pada Gambar 3



Gambar 3. Alur Kerja Program

Gambar 3 memperlihatkan dua proses utama. Pertama proses Perekaman, dimulai dengan pembacaan id dari tag RFID menggunakan RFID reader. Dari pembacaan RFID didapatkan dua jenis value yaitu *id* dan *text*. Id kemudian dicocokkan dengan data yang terdapat di database lokal Raspberry PI. Jika data tidak ditemukan maka dapat dilanjutkan ke langkah selanjutnya yaitu perekaman wajah

dan pembuatan model menggunakan *OpenCV* untuk kemudian disimpan ke *database* lokal. Pada proses pembuatan model terdapat dua proses utama yaitu pendeteksian wajah pada gambar dengan metode HOG dan ekstraksi ciri gambar wajah untuk disimpan ke *database* [12]. Proses kedua adalah perbandingan wajah. Dimulai dengan pembacaan data suhu. Jika suhu pegawai kurang dari 37,5 °C maka dilanjutkan dengan pembacaan data RFID. Data tersebut kemudian digunakan untuk mendapatkan model wajah yang telah disimpan di *database* dari tahap sebelumnya. Data model kemudian dicocokkan dengan wajah gambar karyawan dari kamera. Jika terdapat kecocokan maka akan dibuat data absen dan disimpan ke *database* lokal. Selain penyimpanan di lokal data riwayat absen juga dikirim ke server melalui tunnel VPN. Tunnel VPN digunakan untuk membuat jaringan private di internet sehingga server dapat melakukan koneksi ke perangkat Raspberry PI secara aman. Data yang diterima di server kemudian akan disimpan di *database*. Request ke FCM juga akan dikirim untuk memberikan notifikasi pada perangkat pengguna.

2.4. Pembuatan *Prototype*

Pada analisis kebutuhan ini lebih berfokus pemenuhan kebutuhan dari sisi *hardware*. Untuk membangun sistem yang dapat digunakan untuk melakukan perekaman presensi maka dibutuhkan komponen untuk mendeteksi dan menyimpan data log presensi. Selain itu karena alat ini dilengkapi dengan pendeteksi wajah maka diperlukan juga kamera. Komponen-komponen yang diperlukan secara lengkap dapat dilihat pada tabel 1.

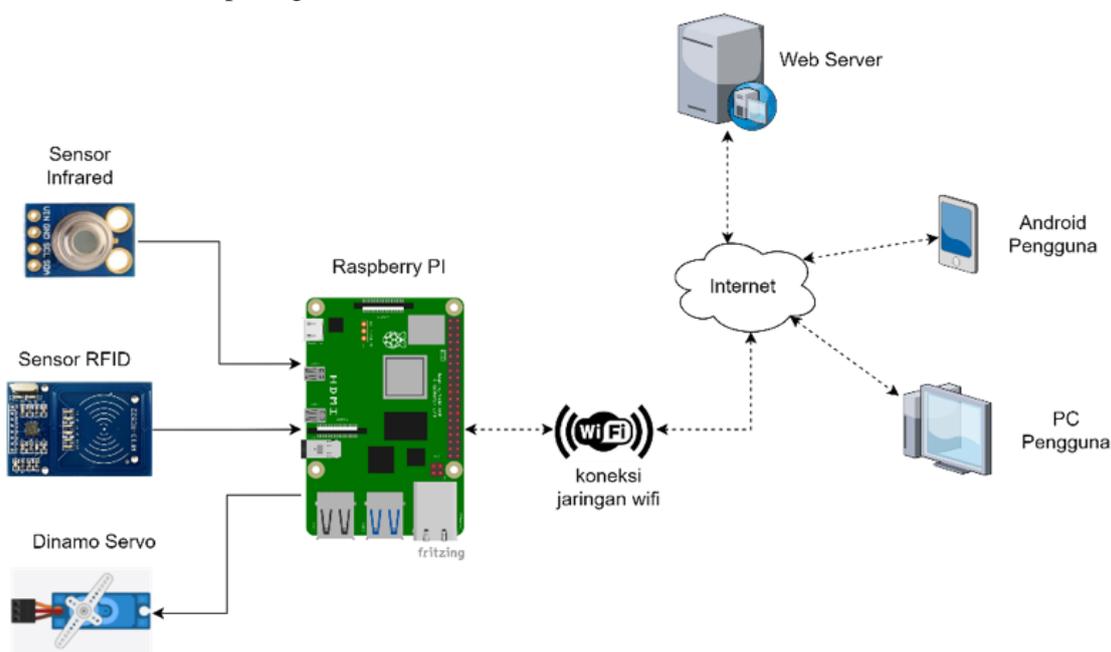
Tabel 1. Daftar komponen-komponen yang digunakan

No	Nama Komponen	Penjelasan
1	Raspberry PI 4	Menurut [13] Raspberry Pi adalah komputer papan tunggal <i>SBC (single-board computer)</i> yang berukuran seperti kartu kredit, yang dapat digunakan untuk program perkantoran, permainan komputer, dan sebagai pemutar multimedia untuk video beresolusi tinggi. Pada penelitian ini fungsi raspberry PI 4 Sebagai pusat kendali yang menerima input dari <i>RFID reader</i> dan kamera, kemudian memproses dan mengirimkannya ke server
2	<i>RFID reader</i>	Menurut [6] <i>RFID</i> atau <i>Radio Frequency Identification</i> menggunakan beberapa jalur gelombang sebagai pemancar sinyal. Namun, jalur yang paling banyak digunakan adalah jalur UHF yang memiliki frekuensi antara 865-866 MHz dan 902-928 MHz. Pada <i>RFID</i> Tag, kode yang ditulis berupa data 96 bit yang terdiri dari 8 bit <i>header</i> , 28 bit nama organisasi pengelola data, 24 bit objek untuk mengidentifikasi jenis produk, dan 36 bit terakhir berupa nomor seri unik untuk setiap tag. Pada penelitian ini fungsi <i>RFID reader</i> untuk membaca data cari tag <i>RFID</i>
3	Sensor suhu <i>Infrared</i>	Menurut [3] Sensor MLX9061 merupakan sebuah sensor yang digunakan untuk mengukur suhu menggunakan cahaya <i>inframerah</i> . Sensor ini secara khusus dirancang untuk mendeteksi energi inframerah dan secara otomatis mengkalibrasi energi inframerah dalam skala suhu. Sensor MLX90614 berfungsi sebagai pendeteksi intensitas radiasi inframerah yang dipancarkan oleh objek. Keluaran dari sensor ini sudah berbentuk digital karena telah dilengkapi dengan <i>ADC (Analog-to-Digital Converter)</i> . Prinsip kerjanya adalah dengan menangkap energi panas yang dipancarkan dari inframerah yang dimiliki oleh setiap benda, kemudian dikonversikan menjadi besaran suhu. Pada penelitian ini fungsi Sensor suhu <i>Infrared</i> untuk membaca data suhu pegawai
4	Modul kamera	Menurut [2] Kamera adalah seperangkat peralatan dengan kelengkapan yang tugasnya mengabadikan suatu objek sebagai gambar yang merupakan hasil proyeksi ke sistem lensa. Kamera adalah alat yang digunakan dalam fotografi. Dalam perkembangannya, kamera digunakan untuk membuat gambar atau menyimpannya ke dalam <i>film/memory card</i> . Pada penelitian ini fungsi kamera untuk merekam wajah guna sebagai inputan pada sistem deteksi wajah

No	Nama Komponen	Penjelasan
5	Monitor Raspberry PI	Berfungsi sebagai media keluaran yang memudahkan user untuk mengatur arah wajah terhadap kamera dan sebagai media informasi
6	Adapter	Berfungsi untuk memberikan supply daya pada perangkat Raspberry PI

2.5 Perancangan *Hardware*

Pada perancangan perangkat keras, semua modul dirangkai sedemikian rupa agar saling terhubung dan dapat dioperasikan. Gambar 4 menunjukkan rancangan keseluruhan sistem IoT. Terdapat dua sensor yang berperan sebagai inputan untuk *Raspberry PI* yaitu sensor Infrared dan sensor RFID. *Raspberry PI* mengolah data inputan untuk kemudian mengirimkannya ke aplikasi web di *server*. Aplikasi web dapat diakses melalui perangkat *android* dan PC.

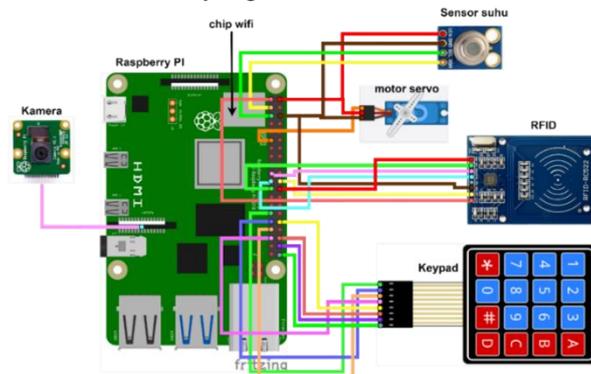


Gambar 4. Rancangan Seluruh sistem IOT

Berikut langkah-langkah perancangan perangkat keras:

- Rangkaian *Raspberry PI* dengan Kamera
 Pada rangkaian ini kamera difungsikan untuk merekam tampilan wajah karyawan dan kemudian data tersebut diolah pada *Raspberry PI* menggunakan OpenCV untuk kemudian dilakukan validasi. Pada *Raspberry PI* telah disediakan port untuk menghubungkan *Raspberry PI* dengan modul kamera sehingga dapat dengan mudah dipasang.
- Rangkaian *Raspberry PI* dengan sensor *RFID*
 Pada rangkaian ini digunakan sensor *RFID RC522*. Sensor *RFID RC522* memiliki tujuh pin yang harus dihubungkan ke *Raspberry PI*.
- Rangkaian *Raspberry PI* dengan sensor *Infrared MLX9061*
 Sensor *Infrared MLX90614* memiliki empat pin yang perlu dihubungkan ke board *Raspberry PI*.
- Rangkaian *Raspberry PI* dengan *Keypad*
 Keypad memiliki delapan pin yang perlu dihubungkan ke board *Raspberry PI*.
- Rangkaian *Raspberry PI* dengan dinamo servo
- Dinamo servo memiliki tiga pin yang perlu dihubungkan ke board *Raspberry PI*.
- Rangkaian Keseluruhan

Berikut adalah rangkaian keseluruhan alat yang dibuat:



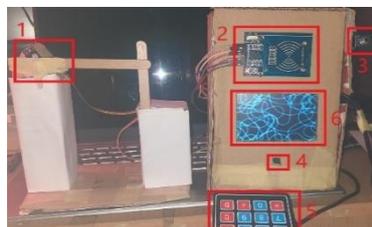
Gambar 5. Rangkaian Keseluruhan

3. Hasil dan Pembahasan

3.1. Implementasi Metode

a. Implementasi Pembacaan RFID

Pembacaan *RFID* dilakukan menggunakan modul *RFID RC522*. Hasil dari pembacaan ini didapatkan dua nilai yaitu id dan text. Pada penelitian ini id digunakan sebagai *identifier*. Setiap pegawai memiliki satu id *RFID* yang unik. Pada Gambar 6 memperlihatkan rangkaian keseluruhan alat.



Gambar 6. Gambar Alat yang telah dibuat

Keterangan Gambar 6:

1.Motor Servo, 2. RFID Reader, 3. Sensor suhu *infrared*, 4. Kamera, 5. Keypad, 6.LCD.

b. Implementasi Pengenalan Wajah

Proses pengenalan data wajah terdiri dari dua bagian yaitu proses perekaman (*enroll*) dan proses perbandingan (*presensi*).

1. Proses Perekaman

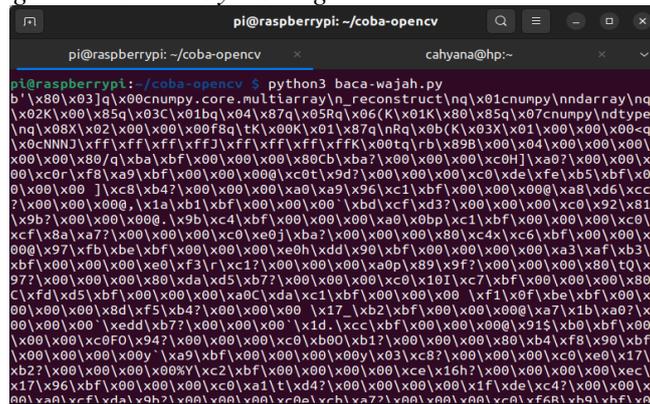
Perekaman wajah dilakukan setelah pegawai memindai tag *RFID*. Pada proses ini diambil tiga gambar melalui kamera. Dari gambar tersebut dilakukan pendeteksian lokasi wajah dengan menggunakan *HOG*. Gambar 7 menunjukkan wajah sebelum dan sesudah *HOG*.



Gambar 7. Model Wajah

Setelah lokasi wajah berhasil di deteksi dilakukan ekstraksi *feature* wajah. Pada penelitian ini ekstraksi *feature* wajah menggunakan model yang telah disediakan oleh ResNet. Alasan penggunaan model Resnet ialah karena model ini telah dilatih sebelumnya menggunakan dataset yang besar dan beragam untuk dapat mengenali dan mengekstraksi fitur-fitur wajah dengan baik. Hasil dari ekstraksi ini menghasilkan kumpulan vektor-vektor *face encoding*.

Vektor *face encoding* ini memiliki panjang 128 dimensi. Agar vektor *face encoding* dapat disimpan di database maka perlu di ubah menjadi *byte string*. Gambar 8 menunjukkan vektor *face encoding* dalam bentuk *byte string*.



Gambar 8. Vektor *face encoding* di dalam bentuk byte string

2. Proses Perbandingan

Pada proses perbandingan data model diambil dari database berdasarkan id *RFID*. Model tersebut kemudian dibandingkan dengan gambar wajah dari kamera. Sebelum di bandingkan kedua data di buat menjadi vektor *face encoding*. Pada perbandingan ini dihitung jarak *Euclidean* untuk menentukan apakah kedua data tersebut dekat atau jauh dalam ruang fitur. Gambar 9 memperlihatkan hasil perhitungan jarak *Euclidean* pada dua foto orang yang sama dengan ekspresi berbeda dengan foto dua orang yang berbeda.



Gambar 9. Perbandingan jarak Euclidean

c. Implementasi Relasi RFID dengan Model Wajah

Data id RFID kemudian direlasikan dengan vektor *face encoding* yang telah didapatkan dari proses sebelumnya. Relasi yang terbentuk ialah one to many dimana pada satu id RFID dapat berelasi dengan banyak vektor *face encoding*. Gambar 10 memperlihatkan vektor *face encoding* yang telah tersimpan di database.

Table: templates

id	id_user	template	created_at
1 27	9	BLOB	2023-06-21 ...
2 28	9	BLOB	2023-06-21 ...
3 29	9	BLOB	2023-06-21 ...
4 30	10	BLOB	2023-06-21 ...
5 31	10	BLOB	2023-06-21 ...
6 32	10	BLOB	2023-06-21 ...
7 33	11	BLOB	2023-06-21 ...

Gambar 10. Vektor *face encoding* di basis data

d. Implementasi Pengiriman Data ke Server

Setelah data pegawai tervalidasi kemudian data presensi dikirim ke *server*. Pengiriman data presensi ini menggunakan *metode Rest API*. Data dikirim dalam format *json* seperti yang terlihat pada Gambar 11.

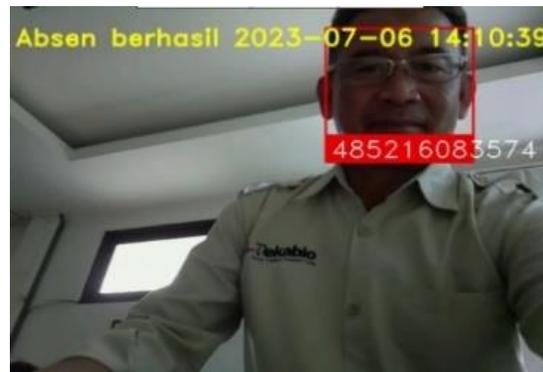
```
{
  "id_kartu": "1006428061936",
  "ip_mesin": "192.168.1.10",
  "waktu": "2023-07-02 07:05:01"
}
```

Gambar 11. Vektor *face encoding* di basis data

3.2. Pengujian

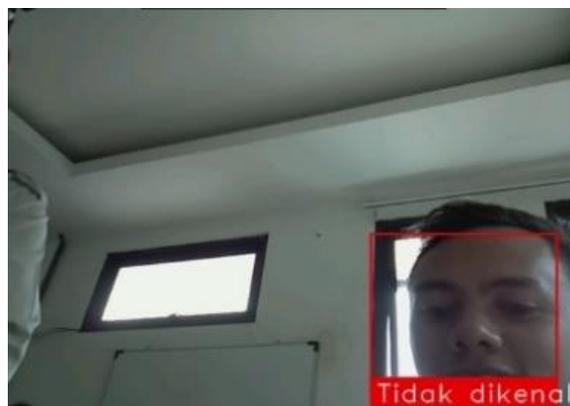
a. Pengujian Pengenalan Wajah

Pengujian pengenalan wajah dilakukan dengan dua skenario yaitu pegawai presensi dengan menggunakan kartu pengenal sendiri dan pegawai absen menggunakan kartu pengenal pegawai lain. Pada Gambar 12 memperlihatkan bahwa pegawai sukses melakukan presensi dengan kartu pengenalnya sendiri.



Gambar 12. Pegawai berhasil presensi

Pada Gambar 13 terlihat bahwa pegawai tidak dapat melakukan presensi dengan menggunakan kartu pengenal orang lain.



Gambar 13. Pegawai tidak dapat melakukan presensi

b. Pengujian Presensi

Pengujian kecepatan pengenalan wajah dilakukan dengan menggunakan dua data pegawai. Pegawai melakukan presensi sebanyak 10 kali dengan posisi wajah yang berbeda-beda kemudian dicatat waktu yang dibutuhkan untuk melakukan deteksi. Waktu ini dihitung dari pertama pegawai melakukan pembacaan RFID sampai waktu presensi diterima. Presensi dinyatakan berhasil jika user berhasil melakukan presensi pada RFID dan pengenal wajah dengan lama pemrosesan kurang dari 2.5 detik.

Tabel 2. Hasil Pengujian Absen

No	Ke	Waktu	Lama (detik)	Suhu	Status
1	1	2023-07-27 11:23:49	1.83	31.61	berhasil
2	2	2023-07-27 11:23:55	0.61	31.39	berhasil
3	3	2023-07-27 11:24:03	1.87	31.11	berhasil
4	4	2023-07-27 11:24:07	2.92	31.33	gagal
5	5	2023-07-27 11:24:10	1.62	30.09	berhasil
6	6	2023-07-27 11:24:14	1.92	32.09	berhasil
7	7	2023-07-27 11:24:17	1.87	30.33	berhasil
8	8	2023-07-27 11:24:21	1.73	31.15	berhasil
9	9	2023-07-27 11:24:26	4.22	32.05	gagal
10	10	2023-07-27 11:24:35	1.82	31.05	berhasil
11	11	2023-07-27 11:24:38	1.81	30.05	berhasil
12	12	2023-07-27 11:24:43	1.05	31.63	berhasil
13	13	2023-07-27 11:24:49	0.9	31.35	berhasil
14	14	2023-07-27 11:24:51	0.08	31.35	berhasil
15	15	2023-07-27 11:24:57	2.00	31.29	berhasil
16	16	2023-07-27 11:25:04	2.16	31.91	berhasil
17	17	2023-07-27 11:25:40	1.96	31.41	berhasil
18	18	2023-07-27 11:25:43	1.84	31.95	berhasil
19	19	2023-07-27 11:25:46	1.99	31.97	berhasil
20	20	2023-07-27 11:27:35	1.94	32.09	berhasil
Rata-rata			1.81		

Tingkat keberhasilan dapat dihitung dengan rumus persentase pada persamaan 1.

$$\text{Persen (\%)} = (\text{jumlah berhasil}) / (\text{jumlah total}) \times 100\% \quad (1)$$

$$\text{Persen} = \frac{18}{20} \times 100\% = 90\%$$

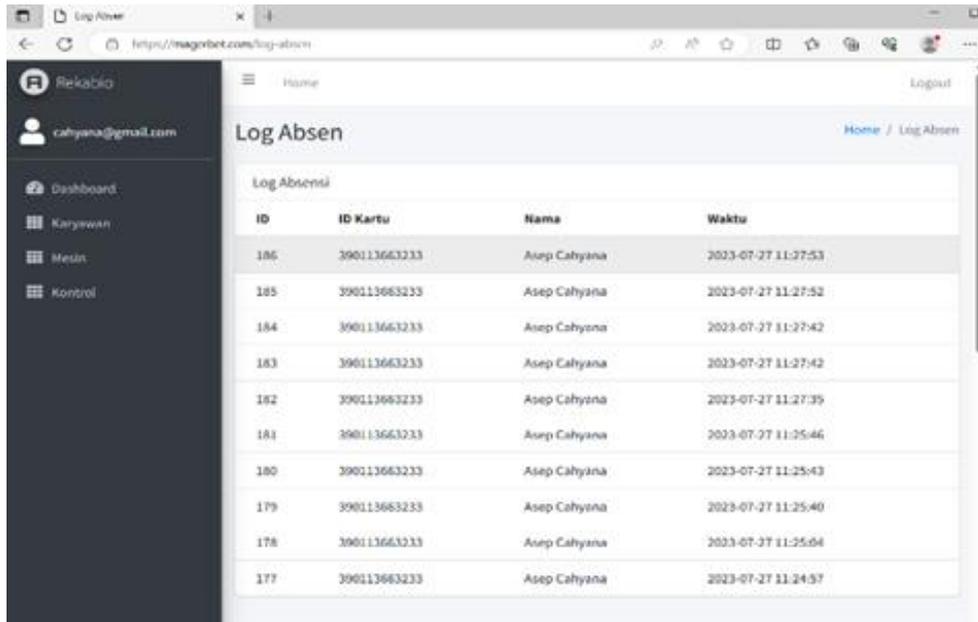
Dari dua pengujian pada Tabel 2, diketahui bahwa rata-rata waktu presensi adalah 1.81 detik dengan tingkat keberhasilan presensi 90%. Dari Tabel 3 dapat diketahui bahwa dari 10 percobaan presensi dengan suhu diatas 37.5 °C adalah gagal.

Tabel 3. Hasil pengujian presensi pada suhu lebih dari 37.5 °C

No	Ke	Waktu	Suhu	Status
1	1	2023-08-17 20:30:16	38.07	gagal
2	2	2023-08-17 20:30:21	38.39	gagal
3	3	2023-08-17 20:30:24	38.43	gagal
4	4	2023-08-17 20:30:26	37.71	gagal
5	5	2023-08-17 20:30:28	37.99	gagal
6	6	2023-08-17 20:30:31	37.75	gagal
7	7	2023-08-17 20:30:33	37.63	gagal
8	8	2023-08-17 20:30:37	37.67	gagal
9	9	2023-08-17 20:30:39	37.75	gagal
10	10	2023-08-17 20:30:42	37.55	gagal

c. Pengujian Pengiriman Data ke *Server*

Pengujian pengiriman data ke *server* dilakukan dengan cara melakukan presensi kemudian dilakukan pengecekan pada web apakah data telah berhasil disimpan. Pada Gambar 14 memperlihatkan data yang telah berhasil disimpan di *server*.

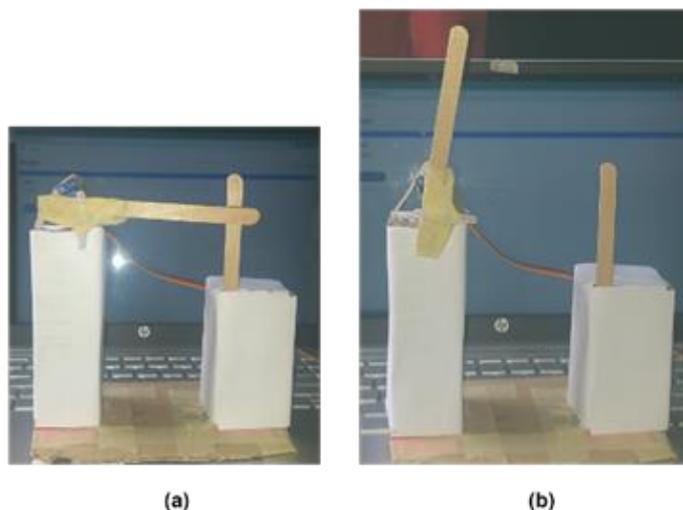


ID	ID Kartu	Nama	Waktu
186	390113663233	Asep Cahyana	2023-07-27 11:27:53
185	390113663233	Asep Cahyana	2023-07-27 11:27:52
184	390113663233	Asep Cahyana	2023-07-27 11:27:42
183	390113663233	Asep Cahyana	2023-07-27 11:27:42
182	390113663233	Asep Cahyana	2023-07-27 11:27:39
181	390113663233	Asep Cahyana	2023-07-27 11:25:46
180	390113663233	Asep Cahyana	2023-07-27 11:25:43
179	390113663233	Asep Cahyana	2023-07-27 11:25:40
178	390113663233	Asep Cahyana	2023-07-27 11:25:04
177	390113663233	Asep Cahyana	2023-07-27 11:24:57

Gambar 14. Data presensi

d. Pengujian Kontrol

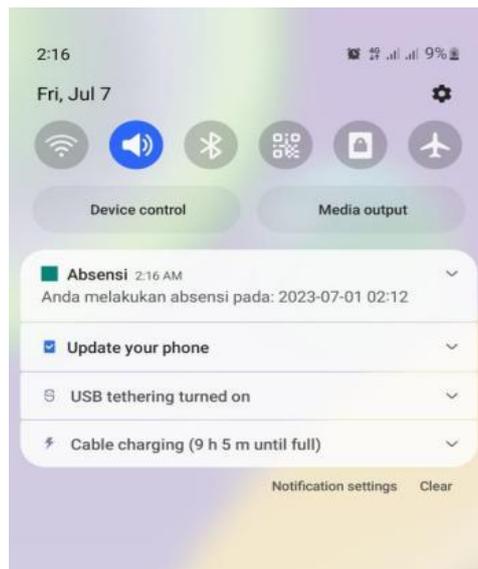
Pengujian kontrol dilakukan dengan cara membukan halaman kontrol kemudian memilih ip mesin yang tersedia. Kemudian tekan tombol Buka Gerbang untuk mengirim perintah membuka gerbang. Pintu akan terbuka selama 3 detik kemudian tertutup kembali. Gambar 15 menunjukkan pengujian kontrol berhasil.



Gambar 15. Pada foto (a) Pintu tertutup, dan foto (b) Pintu terbuka

e. Pengujian Pengiriman Notifikasi Presensi

Pengujian pengiriman notifikasi presensi dilakukan dengan cara melakukan presensi kemudian di cek apakah perangkat *Android* pegawai mendapatkan notifikasi presensi. Pada gambar 16 memperlihatkan notifikasi telah berhasil dikirim.



Gambar 16. Notifikasi Presensi pada perangkat Android

4. Kesimpulan

Berdasarkan hasil penelitian maka kami menyimpulkan bahwa penggunaan alat presensi menggunakan RFID dan pengenalan wajah meningkatkan yang disertai sensor suhu memiliki persentase keberhasilan transaksi 90% dan perangkat presensi berhasil diintegrasikan dengan aplikasi *web* dan *android*. Adapun saran untuk penelitian selanjutnya yaitu; ditambahkan implementasi untuk mendeteksi pemalsuan wajah menggunakan gambar atau video, kode program ditulis dalam Bahasa *python* kadang berjalan lambat, ini dapat ditingkatkan dengan menulis program dalam bahasa *C++* dan ditambahkan sensor cahaya yang dapat menyalakan lampu untuk menjamin ketersediaan cahaya pada saat presensi.

Daftar Pustaka

- [1] R. Setiawan, "Memahami Apa Itu Internet of Things," Dicoding.Com, 08 09 2021. [Online]. [Accessed 06 06 2023].
- [2] M. Noviansyah And S. Sopyan, "Sistem Pengamanan Otomatis Dengan Pengenalan Wajah Berbasis Internet Of Things," *Jurnal Akrab Juara*, Pp. 172-182, 2022.
- [3] Z. Aulia And Busran, "Pendeteksi Suhu Tubuh Menggunakan Sensor Infrared Mlx90614 Sebagai Upaya Pencegahan Covid-19 Di Institut Teknologi Padang," *Jurnal TEKNOIF*, Vol. 9, 2021.
- [4] Tukadi, Abdullah And S. Muharom, "Rancang Bangun Aplikasi Presensi Menggunakan Smartcard RFID Berbasis Web," *Cyclotron*, Vol. 4, 2021.
- [5] T. Johan, "Sistem Presensi Dosen Menggunakan Rfid Untuk Mendukung Administrasi Pendidikan Tinggi Berbasis Less Contact Pada Masa Pandemi Covid-19," *Jurnal Tika*, Vol. 6, Pp. 270-276, 2021.
- [6] R. Hidayat, F. Y. Limpraptono And M. Ardita, "Rancang Bangun Alat Absensi Karyawan Menggunakan RFID Dan ESP32Cam Berbasis Internet Of Things," *Prosiding SENIATI*, Pp. 137-145, 2022.
- [7] A. Roihan, N. Rahayu And D. S. Aji, "Perancangan Sistem Kehadiran Face Recognition Menggunakan Mikrokomputer Berbasis Internet Of Things," *Technomedia Journal*, Vol. 5, 2021.
- [8] Y. Hendrian And R. A. A. Rais, "Perancangan Alat Ukur Suhu Tubuh Dan Hand Sanitizer Otomatis Berbasis IOT," *Jurnal Infortech*, Vol. 3, 2021.
- [9] T. Rahman, G. M. V. T. Mariatmojo, H. Nurdin and H. Kuswanto, "Implementasi VPN Pada VPS Server Menggunakan Openvpn Dan Raspberry Pi," *Teknika*, Vol. 11, 2022.
- [10] I. M. Alviendra, E. Setijadi And G. Kusrahardjo, "Pengembangan Dan Penerapan Sistem Virtual Private Network (VPN) Pada Internet Of Things (IOT) Menggunakan Simulasi," *Jurnal Teknik ITS*, Vol. 11, 2022.
- [11] A. P. Sari, Sulistiyono And N. Kemala, "Perancangan Jaringan Virtual Private Network Berbasis Ip Security Menggunakan Router Mikrotik".
- [12] Z. Nandy, I. Sengupta, S. Mondal, S. Sahil and M. S. Pal, "Electronic Attendance Recorder And Confirmation System Using Facial Identification Modules in Python," *IJRASET*, Vol. 10, 2022.
- [13] M. Riadi, "Raspberry Pi (Definisi, Fungsi, Jenis, Spesifikasi Dan Pemrograman)," *Kajian Pustaka*, 17 12 2020. [Online]. [Accessed 05 06 2023].

PENDEKATAN *HYBRID* PADA SISTEM PERINGKAS TEKS ARTIKEL BERITA BAHASA INGGRIS MENGGUNAKAN *NATURAL LANGUAGE PROCESSING*

Farah Raihanunnisa¹, Muhammad Arhami², Rahmad Hidayat^{2*}

¹Teknik Informatika, Teknologi Informasi dan Komputer, Politeknik Negeri Lhokseumawe

²Teknologi Informasi dan Komputer, Politeknik Negeri Lhokseumawe, Lhokseumawe
Jl. Banda Aceh-Medan Km. 280,3, Buketrata, Mesjid Punteut, Blang Mangat, Lhokseumawe, Aceh, Indonesia
e-mail koresponden: rahmad_hidayat@pnl.ac.id

(received: 11/08/2023, revised: 22/08/2023, accepted: 24/08/2023)

Abstrak

Kegiatan mengumpulkan informasi melalui sejumlah artikel yang dilakukan dalam kehidupan sehari-hari baik oleh kalangan pelajar, peneliti, jurnalis, dan sebagainya, memakan waktu yang relatif lama. Hal ini menimbulkan masalah ketika seseorang harus mengumpulkan informasi yang cukup dalam waktu yang terbatas. Penelitian ini bertujuan untuk membuat sebuah sistem peringkasan teks otomatis yang dapat menghasilkan ringkasan yang relevan dan informatif sehingga membantu penggal informasi untuk dapat menemukan informasi penting dalam sebuah artikel dengan waktu yang lebih sedikit dibandingkan dengan membaca keseluruhan artikel. Sistem peringkasan teks otomatis yang diajukan menerapkan NLP (*Natural Language Processing*) dengan pendekatan *hybrid*. Pendekatan *hybrid* merupakan gabungan dari dua teknik, yaitu teknik peringkasan ekstraktif dan teknik peringkasan abstraktif. Peringkasan ekstraktif merupakan peringkasan yang dilakukan dengan mengekstrak kalimat dari dokumen asli. Sedangkan peringkasan abstraktif dilakukan dengan menghasilkan kalimat baru mendekati peringkasan yang dihasilkan oleh manusia. Peringkasan ekstraktif yang dilakukan menggunakan algoritma *Textrank*, sedangkan teknik peringkasan abstraktif dilakukan dengan menerapkan arsitektur *Transformer*. *Textrank* merupakan pendekatan berbasis *graph*, sedangkan *transformer* merupakan rangkaian algoritma berbasis *encoder decoder*. Pengujian model dilakukan dengan menerapkan teknik pengujian *ROUGE* (*Recall Oriented Understudy for Gisting Evaluation*), dimana *rouge* melakukan pengujian berdasarkan n-gram kata. Hasil yang diperoleh pada penelitian ini menunjukkan nilai *F1-Score* 0.34 pada *ROUGE-1*, 0.15 pada *ROUGE-2*, dan 0.25 pada *ROUGE-L*.

Kata kunci: Arsitektur *Transformer*, Pendekatan *Hybrid*, Peringkasan Teks Otomatis, *Recall Oriented Understudy for Gisting Evaluation*, *Textrank*.

A *HYBRID* APPROACH TO ENGLISH NEWS ARTICLE TEXT SUMMARIZATION SYSTEM USING *NATURAL LANGUAGE PROCESSING*

Abstract

The activity of gathering information through a number of articles carried out in everyday life by students, researchers, journalists, and so on, takes a relatively long time. This creates a problem when one has to gather sufficient information in a limited time. This study aims to create an automatic text summary system that can produce relevant and informative summaries so as to help information seekers to be able to find important information in an article in less time than reading the entire article. The proposed automatic text summary system applies NLP (*Natural Language Processing*) with a hybrid approach. The hybrid approach is a combination of two techniques, namely extractive summary techniques and abstractive summary techniques. Extractive summarization is carried out using the *Textrank* algorithm, while abstractive summarization techniques are carried out by applying the *Transformer* architecture. Model testing is carried out by applying the *ROUGE* (*Recall Oriented Understudy for Gisting Evaluation*) testing technique. The result in this study shows *F1-Score* value 0.34 on *ROUGE-1*, 0.15 on *ROUGE-2*, and 0.25 on *ROUGE-L*.

Keywords: *Automatic Text Summarizer, Hybrid Approach, Recall Oriented Understudy for Gisting Evaluation, Textrank, Transformer Architecture*

1. Pendahuluan

Natural Language Processing atau Pemrosesan Bahasa Alami merupakan bidang dalam *Artificial Intelligence* yang mempelajari cara komunikasi antara manusia dan komputer. Bidang ini mendalami bagaimana suatu mesin seolah dapat memahami dan mengerti bahasa alami yang digunakan manusia [1]. Salah satu bagian dari NLP adalah *Automatic Text Summarization* atau Peringkat Teks Otomatis. Fokus terhadap bidang ini pertama kali dipelajari oleh Lun sekitar enam puluh tahun yang lalu dan telah menyita perhatian dalam beberapa tahun terakhir [2]. Peringkat teks otomatis merupakan sebuah teknik yang membangun ringkasan singkat dan akurat dari sebuah dokumen yang panjang [3]. Peringkat teks otomatis memberikan gambaran informasi keseluruhan dari dokumen hanya melalui beberapa kalimat yang cukup dibaca dalam waktu yang singkat. Hal ini akan membantu pekerjaan dalam mengumpulkan informasi penting melalui sejumlah dokumen tekstual yang menghabiskan waktu yang lama untuk dibaca secara keseluruhan. Alasan tersebut menjadikan peringkat teks otomatis sebagai salah satu teknik yang diminati dan mulai dipelajari oleh manusia. Tujuan utama dari peringkat otomatis yaitu untuk mendapatkan informasi penting yang menjadi inti dari sebuah dokumen.

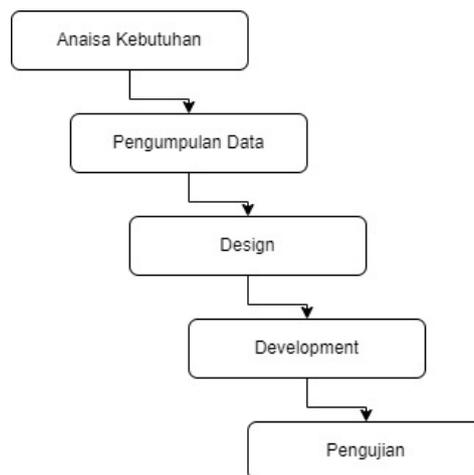
Informasi merupakan kumpulan data dan fakta yang begitu penting dan banyak digunakan dalam seluruh aspek kegiatan manusia. Informasi dapat tersedia dalam berbagai jenis media baik cetak maupun digital. Artikel merupakan salah satu media sumber informasi yang cukup sering digunakan untuk berbagai keperluan. Seorang jurnalis harus membaca banyak artikel sebagai informasi tambahan dan referensi dalam membuat sebuah berita. Seorang pelajar, peneliti, penulis harus berhadapan dengan banyak artikel untuk mengumpulkan data dan informasi yang nantinya akan dijadikan landasan terhadap penelitian atau karya tulis yang akan diangkat. Ada beberapa permasalahan yang diperoleh para peneliti, pelajar, atau masyarakat umum ketika akan melakukan pengumpulan informasi diantaranya adalah : Membutuhkan waktu yang relative lama karena harus membaca keseluruhan dokumen yang panjang dan Perlunya perbaikan dan peningkatan relevansi dalam pengembangan peringkat teks otomatis

Penelitian sebelumnya telah menunjukkan berbagai metode peringkat teks otomatis dengan akurasi yang berbeda beda. Penelitian yang dilakukan oleh Ade Naufal, peringkat teks otomatis dengan menggunakan algoritma Binary Firefly dan menunjukkan nilai F1 score 0.46 pada ROUGE-1, 0.34 pada ROUGE-2, dan 0.42 pada ROUGE-L [4]. Penelitian yang dilakukan oleh Leni, peringkat teks otomatis dengan menggunakan algoritma *Textrank* pada dokumen berbahasa Indonesia, namun sayangnya Leni tidak menjabarkan hasil pengujian secara spesisik [5]. Penelitian menerapkan *deep learning* dengan algoritma RNN pada peringkat teks otomatis menunjukkan hasil nilai F1 score 0.43 pada ROUGE-1, 0.20 pada ROUGE-2 dan 0.39 pada ROUGE-L [6]. Penelitian menerapkan pendekatan Hybrid dengan menggabungkan algoritma LSA dan RBM pada peringkat teks otomatis dan menunjukkan penurunan nilai F1 score yang cukup drastis dibandingkan menggunakan kedua metode secara terpisah [7].

Dari seluruh penelitian yang pernah dilakukan belum pernah ada penelitian yang melakukan peringkat dengan pendekatan *Hybrid* yang menggabungkan algoritma *Textrank* dan *Transformer Architecture*. Oleh karena itu, penelitian ini mengajukan sistem peringkat teks otomatis dengan menerapkan pendekatan *Hybrid* yang menggabungkan peringkat ekstraktif dan abstraktif dengan menggunakan algoritma *Textrank* dan *Transformer Architecture* yang diharapkan memiliki nilai keakuratan yang lebih baik.

2. Metode Penelitian

Secara sederhana alur tahapan pembuatan sistem untuk penelitian terdapat pada Gambar 1.



Gambar 1. Alur Tahapan Penelitian

Berikut ini adalah penjelasan tahapan alur penelitian yang terdapat pada Gambar 1:

- Analisa Kebutuhan, pada tahapan ini akan dilakukan peninjauan lebih lanjut terkait dengan kebutuhan system yang dibangun, apakah system ini dibutuhkan, dan seperti apa detailnya system yang dibutuhkan
- Pengumpulan Data, penelitian ini hanya melibatkan data sekunder, yaitu data yang akan digunakan untuk membangun system peringkas teks otomatis, data yang digunakan merupakan kumpulan dataset CNN/Daily Mail yang berisi artikel berita dan ringkasan yang dibuat secara manual.
- Rancang dan Bangun Sistem, pada tahapan ini dilakukan perancangan sistem termasuk design UML, design *User Interface*, dan dilanjutkan dengan pembangunan sistem peringkas teks otomatis
- Pengujian, pada tahapan ini dilakukan untuk menunjukkan performa dari pendekatan yang digunakan dalam peringkas teks otomatis, hasil dari tahapan ini yang menunjukkan bahwa metodologi yang diajukan mampu menghasilkan ringkasan yang relevan atau tidak.

2.2. Metodologi Penelitian

Penelitian ini mengajukan Teknik *Hybrid* dalam membangun ringkasan. Teknik *hybrid* yang dimaksud merupakan kombinasi dari teknik peringkasan ekstraktif dan abstraktif. Secara garis besar proses peringkasan terbagi kedalam tiga tahapan sebagaimana yang terlihat pada Gambar 2.



Gambar 2. Alur Peringkasan

Gambar 2. Merupakan tampilan dari alur peringkasan secara keseluruhan, berikut merupakan tahapan peringkasan:

- Preprocessing*
Tahap *preprocessing* merupakan tahapan awal yang bertujuan untuk menyiapkan teks sedemikian rupa sehingga siap untuk diringkas. Tahapan ini meliputi proses segmentasi, tokenisasi, menghapus *stopwords* dan tanda baca, dan lemmatisasi.
- Peringkasan Ekstraktif
Peringkasan ekstraktif bertujuan untuk ekstraksi kalimat penting dalam keseluruhan dokumen. Adapun metode yang digunakan adalah *TextRank*. *TextRank* merupakan model peringkasan berbasis grafik dalam pemrosesan teks yang diajukan oleh Mihalchea dan Paul. Ada dua pembelajaran

unsupervised yang diterapkan yakni, model untuk ekstraksi *keyword* dan model untuk ekstraksi kalimat [8]. *TextRank* sendiri merupakan perkembangan dari *PageRank*, dan graph yang dihasilkan pada model ini merupakan *graph* tidak berarah [9].

TextRank merupakan *method* berbasis *graph* yang dapat digunakan untuk meringkas dokumen tunggal. Pada metode ini kalimat-kalimat akan direpresentasikan sebagai *node* yang dihubungkan oleh *edge*, dimana *edge* merupakan representasi *similarity* atau kemiripan antar kalimat. Adapun perhitungan kemiripan menggunakan rumus *cosine similarity*. *Cosine similarity* digunakan untuk menghitung jarak antara dua vektor. Sebagai contoh *x* dan *y* adalah dua vektor yang akan digunakan. Formula dari *cosine similarity* terdapat pada persamaan 1.

$$\cos(x,y) = \frac{x \cdot y}{|x| \cdot |y|} \quad (1)$$

dengan:

x = vektor *x*

y = vektor *y*

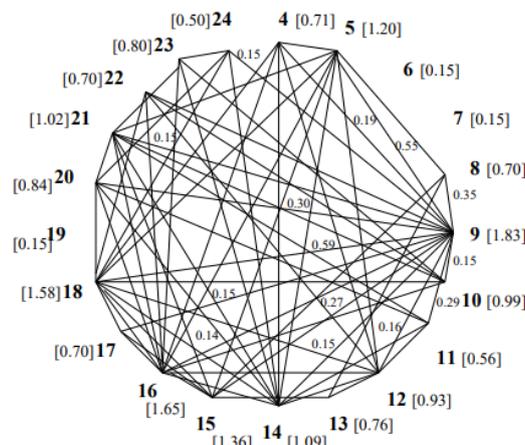
$$|x| = \sqrt{(x_1)^2 + (x_2)^2 + \dots + (x_n)^2}$$

$$|y| = \sqrt{(y_1)^2 + (y_2)^2 + \dots + (y_n)^2}$$

Perbedaan dua vektor diberikan oleh :

$$\text{dis}(x, y) = 1 - \cos(x, y)$$

Gambar 3. merupakan gambaran pembobotan pada algoritma *TextRank* dimana 24 kalimat saling terhubung dengan *edge* yang merupakan perhitungan *similarity* terhadap kalimat lainnya.



Gambar 3. Pembobotan *TextRank* [5]

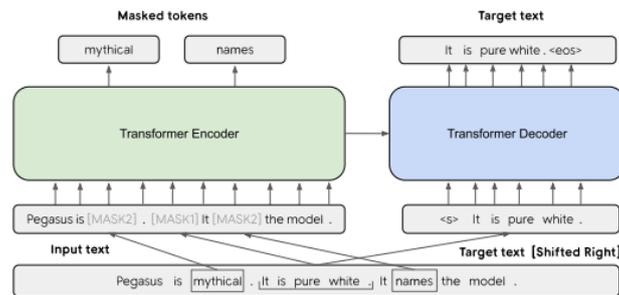
Adapun tahap peringkasan ekstarktif terdiri dari :

1. Mengubah kalimat menjadi vektor dengan menggunakan *sentence embedding*.
2. Menghitung kemiripan antar kalimat melalui vektor yang telah dibangun menggunakan *cosine similarity*.
3. Merepresentasikan *graph* dimana kalimat akan dijadikan *node* dan dihubungkan oleh nilai kemiripan kalimat tersebut dengan kalimat lainnya.
4. Setelah *graph* serta bobot direpresentasikan selanjutnya adalah proses perangkangan, dimana kalimat dengan nilai *top rank* akan dimasukkan kedalam ringkasan. Pemilihan dimulai dari kalimat dengan skor tertinggi dan seterusnya.

c. Peringkasan Abstraktif

Transformer architecture diimplementasikan dalam tahapan peringkasan abstraktif. Secara garis besar struktur dalam arsitektur *transformer* terbagi menjadi dua yaitu *Encoder* dan *decoder*. *Encoder* berfungsi untuk memproses kalimat *input* dan *decoder* berfungsi untuk membangun kalimat *output*.

Masing – masing *encoder* dan *decoder* mengandung *multiple layers* dari *self attention* dan *feed forward neural networks* [10]. Implementasi *transformer architecture* dalam penelitian ini dilakukan dengan memanfaatkan Pegasus model. Pegasus merupakan sebuah *pretrained* model yang dikembangkan diatas *transformer architecture* [11]. Pegasus dilatih pada sekitar 1,5 juta dataset dari berbagai kumpulan dataset *summarization* seperti Xsum, CNN, Wikihow, Pubmed, Billsum, dan lain sebagainya. Arsitektur Pegasus dapat dilihat pada gambar 4.



Gambar 4. Arsitektur Pegasus Model [8]

Pegasus mengimplementasikan standard transformer encoder – decoder dengan mengaplikasikan GSG dan MLM.

1. *Gap Sentences Generation*

Selama masa *pre-training* dokumen ditransformasikan kedalam *gap sentences*. *Gap sentences* dilakukan dengan menghapus sebagian kalimat dari dokumen dan menggunakan dokumen sisa untuk memprediksi sebagian kalimat yang telah dihapus.

2. *Masked Language Model*

Dalam MLM dipilih token tertentu secara acak dari suatu dokumen untuk ditutup atau disebut dengan istilah *mask*. Dan selanjutnya dengan dokumen yang tersisa model akan dilatih untuk memprediksi token yang telah ditutup tersebut.

Model pegasus menggunakan kedua teknik GSG dan MLM selama masa pretraining yang bertujuan untuk memahami makna, pola bahasa dan relasi antar setiap kata. GSG berfokus untuk memahami struktur dan konteks dokumen, sedangkan MLM berfokus untuk mempelajari representasi token[8].

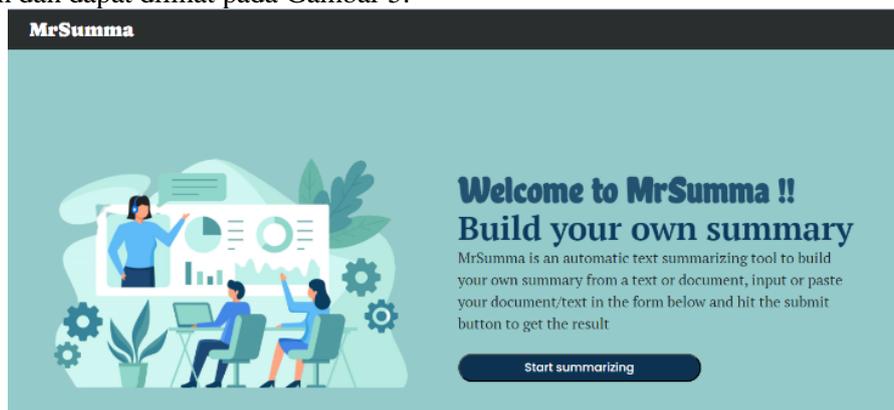
3. Hasil dan Pembahasan

Pada bagian ini berisi analisis, hasil implementasi ataupun pengujian serta pembahasan dari topik penelitian, yang bisa dibuat terlebih dahulu metodologi penelitian. Bagian ini juga merepresentasikan penjelasan yang berupa penjelasan, gambar, tabel dan lainnya.

3.1. User Interface

a. Halaman Depan

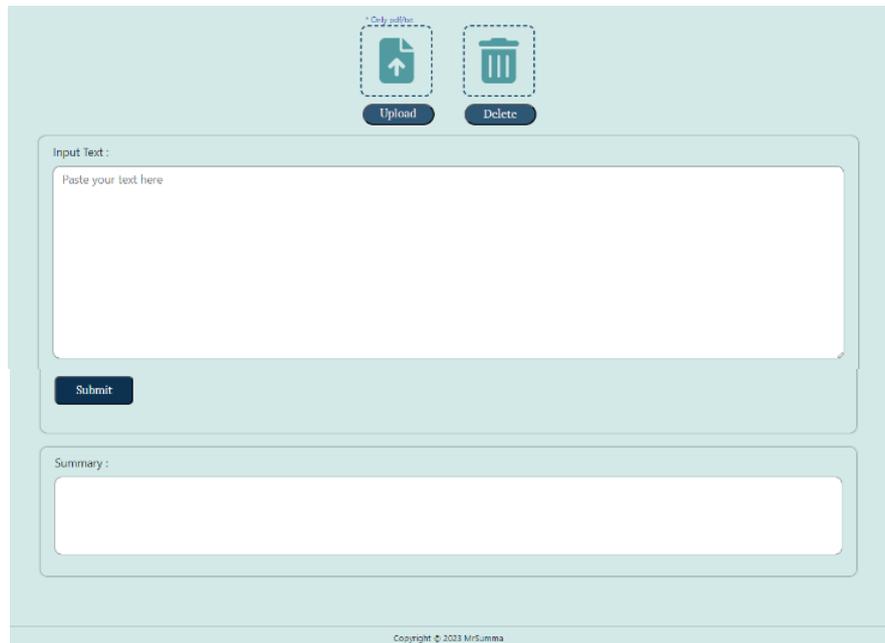
Tampilan *user interface* halaman depan pada sistem peringkask teks otomatis hanya terdiri dari satu halaman dan dapat dilihat pada Gambar 5.



Gambar 5. Halaman Depan Sistem Peringkat Teks Otomatis

b. Form Peringkat Teks Otomatis

Pada *user interface* terdapat sebuah form untuk peringkat teks otomatis yang dapat digunakan untuk memasukkan dokumen atau teks yang ingin diringkas. Proses *input* dapat dilakukan melalui paste teks pada *textarea* ataupun melalui tombol *upload* dokumen. Selanjutnya untuk mendapatkan hasil ringkasan maka dapat digunakan tombol submit. Dan hasil ringkasana akan tampil pada *form output*. Jika ingin memasukkan teks baru maka *user* dapat menggunakan tombol delete untuk membersihkan teks area. Tampilan form peringkat terdapat pada Gambar 6.



Gambar 6. Form Peringkat Teks Otomatis

3.2. Pengujian

Pada bagian ini merupakan hasil ringkasan dari proses pengujian menggunakan matriks *ROUGE* (*Recall Oriented Understudy for Gisting Evaluation*). *ROUGE* merupakan standar pengujian yang selama ini digunakan dalam peringkat. Peringkat dilakukan pada 100 artikel CNN/Daily Mail yang dilengkapi dengan ringkasan manual sebagai hasil aktual atau sebagai referensi yang digunakan sebagai sampel pengujian.

Tabel 1. Sampel Hasil Pengujian 100 Artikel

Nama Dokumen	F1-Score ROUGE-1	F1-Score ROUGE-2	F1-Score ROUGE-L
doc0	0.518519	0.379747	0.469136
doc1	0.46	0.244898	0.32
doc2	0.142857	0	0.122449
doc3	0.408163	0.170213	0.367347
doc4	0.307692	0.105263	0.205128
doc5	0.273684	0.064516	0.189474
doc6	0.336634	0.141414	0.217822
doc7	0.690476	0.536585	0.547619
doc8	0.47619	0.196721	0.31746
...
doc95	0.242424	0	0.151515
doc96	0.36	0.183673	0.3
doc97	0.347826	0.059701	0.144928
doc98	0.314607	0.068966	0.202247

Adapun hasil rata-rata pengujian dari 100 artikel yang diperoleh terdapat pada Tabel 2.

Tabel 2. Hasil Keseluruhan Pengujian

	Average	Low	High
F1-Score ROUGE-1	0.34	0.2	0.69
F1-Score ROUGE-2	0.15	0.	0.49
F1-Score ROUGE-L	0.25	0.02	0.66

Hasil menunjukkan nilai *F1-Score* tertinggi 0.69 pada *ROUGE-1*, 0.49 pada *ROUGE-2*, dan 0.66 pada *ROUGE-L*, rata-rata untuk *F1-Score* 0.34 pada *ROUGE-1*, 0.15 pada *ROUGE-2*, dan 0.25 pada *ROUGE-L*.

4. Kesimpulan

Penelitian ini menyimpulkan bahwa implementasi hybrid pada peringkasan teks otomatis menunjukkan hasil *F1-Score* 0.34 pada *ROUGE-1*, 0.15 pada *ROUGE-2*, dan 0.25 pada *ROUGE-L*. Nilai ini menunjukkan penurunan jika dibandingkan dengan implementasi teknik abstraktif dengan Pegasus model sebelum digabungkan dengan TextRank. Disisi lain proses peringkasan memakan waktu yang lebih sedikit dibandingkan dengan penerapan teknik abstraktif dasar. Sehingga, dapat disimpulkan bahwa mengaplikasikan pendekatan hybrid dengan kombinasi TextRank dan *Transformer Architecture* memiliki keakuratan lebih kecil dari penerapan *Transformer Architecture* secara murni, namun mengungguli dari segi kecepatan

Daftar Pustaka

- [1] H. Bryan *et al.*, "Pemanfaatan *Text Summarization Dengan Support Vector Machine Dan K-Nearest Neighbor* Pada Analisis Sentimen Untuk Mempermudah Pengguna Membaca Review Game Steam," *J. Infra*, vol. 10, no. 1, pp. 31–36, 2022.
- [2] Y. J. Kumar, O. S. Goh, H. Basiron, N. H. Choon, and P. C. Suppiah, "A review on automatic text summarization approaches," *J. Comput. Sci.*, vol. 12, no. 4, pp. 178–190, 2016, doi: 10.3844/jcssp.2016.178.190.
- [3] A. Raj, S. E. M, and D. P. S, "A Systematic Survey on Multi-document Text Summarization," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 10, no. 6, pp. 3148–3153, 2021, doi: 10.30534/ijatcse/2021/111062021.
- [4] A. N. Ammar and S. Suyanto, "Peringkasan Teks Ekstraktif Menggunakan Binary Firefly Algorithm," *Indo-JC*, vol. 5, no. September, pp. 31–42, 2020, doi: 10.21108/indojc.2020.5.2.440.
- [5] L. Pertiwi, "Penerapan Algoritma Text Mining, Steaming Dan Texrank Dalam Peringkasan Bahasa Inggris," *Bimasati*, vol. 1, no. 3, pp. 100–104, 2022.
- [6] D. Suleiman and A. Awajan, "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges," *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/9365340.
- [7] R. Varma, "SJSU ScholarWorks A Hybrid Approach for Multi-document Text Summarization," San Jose State University, 2019.
- [8] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," *Proc. 2004 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2004 - A Meet. SIGDAT, a Spec. Interes. Gr. ACL held conjunction with ACL 2004*, vol. 85, pp. 404–411, 2004.
- [9] J. Pragantha, Eris, and V. C. M, "Penerapan Algoritma TextRank untuk Automatic Summarization pada Dokumen Berbahasa Indonesia," *Ilmu Tek. dan Komuter*, vol. 1, no. 1, pp. 71–78, 2017.
- [10] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.
- [11] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-Training with extracted gap-sentences for abstractive summarization," *37th Int. Conf. Mach. Learn. ICML 2020*, vol. PartF16814, pp. 11265–11276, 2020.