

Volume 5, No.1 2024

JISE

Journal of Internet and Software Engineering



ISSN 2797-9016



9

<https://ugm.id/jise>

The journal published by
Department of Electrical Engineering and Informatics
Vocational College, Universitas Gadjah Mada

EDITORIAL TEAM

Journal of Internet and Software Engineering (JISE)

Editor-in-Chief

Ganjar Alfian, Universitas Gadjah Mada, Indonesia

Computer Networks Section Editor

Sahirul Alam, Universitas Gadjah Mada, Indonesia

Software Engineering Section Editor

Firma Syahrian, Universitas Gadjah Mada, Indonesia

Applied Artificial Intelligence Section Editor

Yuris Mulya Saputra, Universitas Gadjah Mada, Indonesia

Editorial Board

Ronald Adrian, Universitas Gadjah Mada, Indonesia

Wijayanti Dwi Astuti, Universitas Gadjah Mada, Indonesia

Dinar Nugroho Pratomo, Universitas Gadjah Mada, Indonesia

Anni Karimatul Fauziah, Universitas Gadjah Mada, Indonesia

Filip Benes, VSB-Technical University of Ostrava, Czech Republic

Muhammad Syafrudin, Sejong University, South Korea

Umar Farooq, Coventry University, United Kingdom

Layout Editor

Andi Fariel, Universitas Gadjah Mada, Indonesia

Muhammad Rizal Pahleviannur, Universitas Gadjah Mada, Indonesia

<https://ugm.id/jise>

The journal published by

Department of Electrical Engineering and Informatics

Vocational College, Universitas Gadjah Mada

Sekip unit III, Caturtunggal, Terban,

Kec. Gondokusuman, Kab. Sleman, D.I. Yogyakarta 55281

- 1. PEMODELAN PREDIKSI KADAR GULA DARAH PADA PASIEN DIABETES MENGGUNAKAN METODE REGRESI LINEAR** 1-8
Hanan Hanafi, Ganjar Alfian, Tri Widodo, Muhammad Syafrudin
- 2. PURWARUPA SISTEM KLASIFIKASI LEGALITAS INVESTASI BERBASIS ALGORITMA BIDIRECTIONAL LONG SHORT TERM MEMORY** 9-14
Muhammad Navi Nugraha, Muhammad Arrofiq
- 3. ANALISIS TINGKAT AKURASI METODE PENDETEKSIAN PLAGIARISME IDE DENGAN MENGGUNAKAN YAKE DAN SENTENCE TRANSFORMER** 15-22
Salsabila Laily Rahma, Umar Taufiq
- 4. ANALISIS KOMPARASI PADA TEKNIK BLACK BOX TESTING (STUDI KASUS: WEBSITE LARS)** 23-28
Salmania Jesamine Putri, Divi Galih Prasetyo Putri, Widhy Hayuhardhika Nugraha Putra
- 5. KOMPARASI METODE AUTOMASI DAN HYBRID PADA PENGUJIAN APLIKASI MOBILE WEBRTC MENGGUNAKAN APPIUM** 29-36
Gabriela Anggerita Jasmin, Divi Galih Prasetyo Putri
- 6. PENGEMBANGAN SISTEM INFORMASI TUGAS AKHIR BERBASIS WEB PROGRAM STUDI TRPL SV UGM** 37-42
Siti Qotijah, Kholifah Murniati
- 7. ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN PENGGUNA APLIKASI STARBUCKS MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE** 43-49
Muhammad Adin Palimbani, Rochana Prih Hasuti, Rian Adam Rajagede

Pemodelan Prediksi Kadar Gula Darah pada Pasien Diabetes Menggunakan Metode Regresi Linear

Hanan Hanafi¹, Ganjar Alfian^{1,*}, Tri Widodo², Muhammad Syafrudin³

¹Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada, Indonesia;
hananhanafi@mail.ugm.ac.id

²Program Studi Pendidikan Teknologi Informasi, Universitas Teknologi Yogyakarta, Indonesia;
triwido@uty.ac.id

³Department of Artificial Intelligence, Sejong University, Republic of Korea;
udin@sejong.ac.kr

*Korespondensi: ganjar.alfian@ugm.ac.id;

Abstract – Predicting blood glucose levels in the future can help diabetic patients to take preventive action earlier so that they can control their blood glucose levels. This study proposed a blood glucose levels prediction model using linear regression method. Time series data of blood glucose levels from 30 type 1 diabetic patients were used as a single input parameter with the additional utilization of statistical features. The dataset was tested using several algorithms, namely Linear Regression, Lasso Regression, Ridge Regression, eXtreme Gradient Boosting (XGB), and K-Nearest Neighbor (KNN). The prediction models have been tested and showed that the Linear Regression method outperformed other models with average Root Mean Square Error (RMSE) values of 15 patients tested were 5.024, 12.488, and 20.635, Mean Absolute Error (MAE) values were 2.891, 8.272, and 14.926 and Coefficient of Determination (R^2) values were 0.962, 0.741, and 0.39 for the Prediction Horizon (PH) of 5 minutes, 15 minutes, and 30 minutes, respectively. The results of the prediction model in this study were implemented in a website-based information system. In this system, the user can predict blood glucose levels in the future by utilizing a history of blood glucose levels in the previous 30 minutes. Users can also see the visualization of data based on the movement of blood glucose levels with different time interval. This system is expected to help diabetic patients to predict their future blood glucose levels so that they can control their blood glucose levels and avoid bad health conditions in the future.

Keywords – Prediction Model, Blood Glucose, Diabetes, Linear Regression

Intisari – Mengetahui kadar gula darah di masa depan akan dapat membantu penderita diabetes dalam melakukan tindakan preventif lebih awal, sehingga dapat mengontrol kadar gula darah dan penyakit diabetesnya. Penelitian ini mengusulkan pemodelan prediksi kadar gula darah pada pasien diabetes menggunakan metode regresi linear. Dataset yang digunakan adalah data *time series* dari kadar gula darah pada 30 anak penderita diabetes tipe 1. Dataset tersebut digunakan sebagai parameter *input* tunggal dengan tambahan pemanfaatan fitur statistik yang diuji menggunakan beberapa algoritma, yaitu *Linear Regression*, *Lasso Regression*, *Ridge Regression*, *eXtreme Gradient Boosting (XGB)*, dan *K-Nearest Neighbor (KNN)*. Pada tahap evaluasi performa model menunjukkan bahwa metode regresi linear lebih baik dari model prediksi lainnya. Hasilnya menunjukkan untuk *Prediction Horizon (PH)* pada 5 menit, 15 menit, dan 30 menit didapat nilai rata-rata *Root Mean Squared Error (RMSE)* dari 15 pasien yang diuji sebesar 5,024, 12,488, dan 20,635, nilai *Mean Absolute Error (MAE)* sebesar 2,891, 8,272, dan 14,926 serta nilai *Coefficient of Determination (R^2)* sebesar 0,962, 0,741, dan 0,39. Hasil model prediksi pada penelitian ini diimplementasi dan divisualisasikan ke sistem informasi berbasis *website*. Dalam sistem tersebut pengguna dapat memprediksi kadar gula darah di masa depan dengan berdasarkan riwayat kadar gula darah pada waktu 30 menit sebelumnya. Pengguna juga dapat melihat visualisasi data pergerakan kadar gula darah berdasarkan rentang waktu tertentu. Sistem ini diharapkan dapat membantu pasien diabetes untuk memprediksi kadar gula darah pada masa depan, sehingga dapat mengontrol kadar gula darahnya dan menghindari kondisi kesehatan yang buruk pada masa depan.

Kata kunci – Model Prediksi, Gula Darah, Diabetes, Regresi Linear

I. PENDAHULUAN

Diabetes adalah penyakit kronis yang disebabkan oleh ketidakmampuan tubuh untuk memproduksi hormon insulin atau karena penggunaan yang tidak efektif dari produksi insulin [1]. Berdasarkan klasifikasi saat ini, diabetes terbagi menjadi 2 jenis yaitu diabetes tipe 1 dan diabetes tipe 2 [2]. Diabetes ditandai dengan tingginya kadar gula dalam darah dan dapat menyebabkan kerusakan serius pada jantung, pembuluh darah, ginjal, dan saraf. Gejala diabetes ditandai dengan mudah haus, sering buang air kecil, cepat lelah, penglihatan kabur, dan berat badan yang turun. Diabetes

adalah ancaman yang serius bagi kesehatan dunia yang merupakan salah satu dari empat prioritas penyakit tidak menular yang ditindaklanjuti oleh para pemimpin dunia [3]. Diabetes merupakan peringkat ke sembilan penyebab kematian terbanyak secara global pada Tahun 2019, meningkat sebesar 70% sejak Tahun 2000 [4]. Secara global, diperkirakan 422 juta orang dewasa hidup dengan diabetes pada Tahun 2014. Selama dekade terakhir, prevalensi diabetes meningkat lebih besar di negara-negara berpenghasilan rendah dan menengah dibandingkan negara berpenghasilan tinggi. Diabetes menyebabkan 1,5 juta kematian pada tahun 2012 serta tambahan 2,2 juta kematian

dari penyakit kardiovaskular, ginjal kronis, dan tuberkulosis yang berhubungan atau komplikasi dengan diabetes [3].

Diabetes tipe 1 tidak dapat dicegah dengan pengetahuan saat ini, sehingga hal efektif yang dapat dilakukan oleh pasien atau penderita diabetes tipe 1 adalah mencegah diabetes tipe 2 dan komplikasinya seperti berolahraga secara teratur, makan makanan sehat, menghindari rokok, dan mengontrol tekanan darah [3]. Mengetahui prediksi kadar gula darah di masa depan juga dapat membantu seorang pasien atau penderita diabetes untuk mengontrol kadar gula darahnya dan melakukan tindakan preventif lebih awal, sehingga dapat menghindari kondisi yang fatal di masa yang akan datang.

Memprediksi kadar gula darah di masa depan dapat menggunakan metode berbasis *machine learning*. Metode *machine learning* sudah banyak digunakan di berbagai area dan dapat meningkatkan performa sistem, seperti pada area *Internet of Things* [5] dan *Natural Language Processing* [6]. Penelitian sebelumnya menunjukkan bahwa sistem prediksi kadar gula darah pada pasien diabetes dapat dibangun menggunakan metode *machine learning*. Pappada dkk [7] mengusulkan model prediksi kadar gula darah pada pasien diabetes menggunakan *Neural Network* (NN) dengan memasukkan dari 10 pasien menggunakan data nilai *Continuous Glucose Monitoring* (CGM), dosis insulin, nilai meteran glukosa, asupan gizi, gaya hidup, dan faktor emosional. Pada penelitiannya menggunakan *Prediction Horizon* (PH) 75 menit yang menghasilkan nilai RMSE 43,9 mg/dL.

Penelitian ini menjelaskan pentingnya penggunaan metode *machine learning*, khususnya model prediksi berbasis regresi linear, dalam meningkatkan efektivitas pemantauan kadar gula darah pada penderita diabetes. Keunggulan penelitian ini terletak pada kemampuan model prediksi untuk memberikan estimasi akurat mengenai tingkat gula darah masa depan. Dengan mengintegrasikan model terlatih (*trained model*) ini ke dalam sistem monitoring gula darah berbasis web, penelitian ini tidak hanya menciptakan solusi yang lebih canggih, tetapi juga praktis. Integrasi tersebut akan memungkinkan pemantauan gula darah secara *real-time*, memberikan informasi yang sangat berharga bagi pasien dan tenaga medis. Kemampuan untuk memprediksi masa depan membuka pintu bagi intervensi yang lebih dini dan personalisasi perawatan, yang pada gilirannya dapat mengoptimalkan manajemen penyakit diabetes.

II. DASAR TEORI

Penelitian sebelumnya telah menghasilkan temuan yang signifikan, menunjukkan bahwa penerapan model prediksi menggunakan algoritma *machine learning* dapat memberikan hasil yang akurat dalam memprediksi kadar gula darah. Dengan memanfaatkan data pasien yang luas dan beragam, seperti riwayat medis, pola makan, dan faktor gaya hidup, metode ini mampu memberikan informasi yang berharga untuk mendukung upaya diagnosis dan pengelolaan diabetes secara lebih efektif. Georga dkk [8] mengusulkan menggunakan model prediksi dengan metode *Support Vector*

Regression (SVR) untuk prediksi gula darah pada pasien diabetes tipe 1. Model yang diusulkan tersebut menggunakan data gula darah, insulin, makanan, serta aktivitas fisik dari 27 pasien diabetes tipe 1. Evaluasi model tersebut menggunakan *Prediction Horizon* (PH) 15 menit, 30 menit, dan 60 menit dengan hasil RMSE masing-masing didapat 5,21 mg/dL, 6,03 mg/dL, dan 7,14 mg/dL.

Penelitian Ben Ali dkk [9] mengusulkan model prediksi kadar gula darah pada pasien diabetes tipe 1 menggunakan *Artificial Neural Network* (ANN) hanya dengan menggunakan masukkan *Continuous Glucose Monitoring* (CGM) sebagai masukkan tunggal. Model tersebut diuji dengan menggunakan data dari 13 pasien diabetes tipe 1. Hasil evaluasi model prediksi tersebut didapat nilai RMSE 6,43 mg/dL, 7,46 mg/dL, dan 8,13 mg/dL, dan 9,03 mg/dL dengan *Prediction Horizon* (PH) 15 menit, 30 menit, 45 menit, dan 60 menit.

Penelitian Hamdi dkk [10] menggunakan model prediksi dengan metode *Support Vector Regression* (SVR) untuk prediksi kadar gula darah pada pasien diabetes tipe 1. Pada penelitiannya model tersebut menggunakan data 12 pasien dengan memasukkan hanya menggunakan data kadar gula darah atau *Continuous Glucose Monitoring* (CGM) sebagai masukkan tunggal model. Hasil evaluasi performa model tersebut didapat nilai RMSE 9,44 mg/dL untuk *Prediction Horizon* (PH) 15 menit, 10,78 untuk *Prediction Horizon* (PH) 30 menit, 11,82 mg/dL untuk *Prediction Horizon* (PH) 45 menit, dan 12,95 mg/dL untuk *Prediction Horizon* (PH) 60 menit.

Penelitian lain dilakukan oleh Martinsson dkk [11] mengusulkan model prediksi kadar gula darah menggunakan *Recurrent Neural Network* dengan metode *Long Short-Term Memory* (LSTM). Model tersebut menggunakan data kadar gula darah atau *Continuous Glucose Monitoring* (CGM) dari 6 pasien. Performa model dievaluasi dengan menggunakan *Prediction Horizon* (PH) 30 menit dan 60 menit. Evaluasi tersebut didapat nilai rerata RMSE 18,87 mg/dL pada prediksi 30 menit dan 31,40 mg/dL pada prediksi 60 menit.

Alfian dkk [12] mengusulkan model pembelajaran mesin untuk memprediksi kadar gula darah pada pasien diabetes menggunakan metode *Artificial Neural Network* (ANN). Dalam penelitiannya tersebut menggunakan data kadar gula darah pada pasien diabetes tipe 1 untuk memprediksi kadar gula darah pada *Prediction Horizon* (PH) 30 menit dan 60 menit selanjutnya. Model prediksi tersebut diuji kepada 12 anak yang mengalami Diabetes tipe 1. Berdasarkan hasil pengujiannya tersebut didapat model ANN mengungguli model lain dengan rata-rata RMSE 2,82 mg/dL, 6,31 mg/dL, 10,65 mg/dL, dan 15,33 mg/dL untuk *Prediction Horizon* (PH) 15 menit, 30 menit, 45 menit, dan 60 menit.

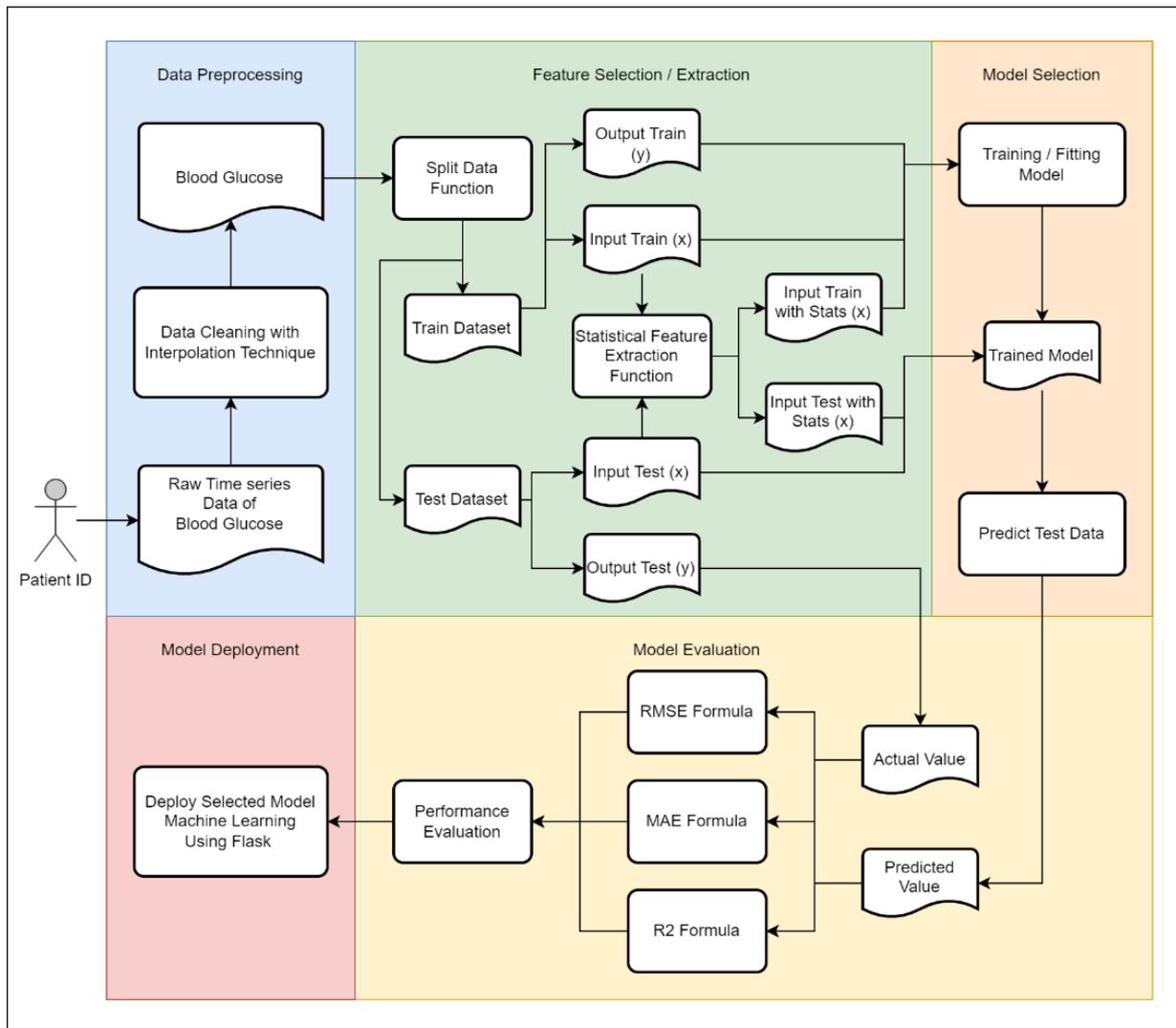
Penelitian ini mengusulkan model prediksi kadar gula darah pada pasien diabetes menggunakan metode *Linear Regression*. Kemudian hasil analisis dan pengembangan

model prediksi pada penelitian ini diimplementasi ke sistem informasi berbasis *website*. Pengguna juga dapat melihat visualisasi data pergerakan kadar gula darah berdasarkan waktu dalam bentuk diagram. Sistem ini diharapkan dapat membantu pasien diabetes untuk memprediksi kadar gula darah pada masa depan, sehingga dapat menghindari hal buruk atau kondisi yang fatal bagi penderita diabetes pada masa depan.

III. METODOLOGI

Pada penelitian ini menggunakan dataset berbentuk *time series* atau historis dari kadar gula darah pada 30 pasien anak diabetes tipe 1. Dataset tersebut berasal dari jurnal penelitian yang berjudul *Evaluation of Counter-regulatory Hormone Responses during Hypoglycemia and the Accuracy of Continuous Glucose Monitors in Children with T1DM* yang diterbitkan oleh Jaeb Center for Health Research (JCHR). Dataset tersebut digunakan sebagai parameter *input* tunggal dengan tambahan pemanfaatan data statistik atau *statistical feature* dalam jarak waktu tertentu (*window*).

Membangun sistem prediksi kadar gula darah pada pasien diabetes menggunakan metode regresi linear pada tahap awalnya adalah membuat model prediksi. Pada tahapan pembuatan model prediksi dilakukan analisis dan perancangan model prediksi. Analisis awal yang dilakukan adalah menganalisis data yang akan digunakan untuk membuat model prediksi dengan melakukan pra-pemrosesan data dan ekstraksi fitur untuk mendapatkan data masukan dan keluaran yang diinginkan. Hasil ekstraksi fitur tersebut digunakan untuk melatih dan menguji algoritma yang dipilih untuk digunakan dalam sistem. Hasil pengujian algoritma tersebut dievaluasi untuk mendapatkan model prediksi dengan tingkat akurasi terbaik. Selanjutnya, hasil model prediksi tersebut akan diimplementasikan ke dalam sistem berbasis *website* yang dapat digunakan untuk memprediksi kadar gula darah sesuai masukan yang dibutuhkan sistem. Tahapan dalam analisis dan perancangan model prediksi yang dilakukan dapat dilihat pada Gambar 1.



Gambar 1. Skema analisis dan perancangan model prediksi gula darah

Data pre-processing atau pra-pemrosesan data adalah tahap pertama yang diperlukan sebelum *machine learning* diterapkan, karena algoritma belajar dari data dan hasil *machine learning* sangat bergantung pada data yang tepat yang diperlukan untuk penyelesaian masalah tertentu yang disebut *feature* [13]. Pada penelitian ini digunakan teknik interpolasi atau *data interpolation*.

Feature extraction atau *feature selection* adalah proses untuk mendapatkan nilai masukan dan keluaran yang

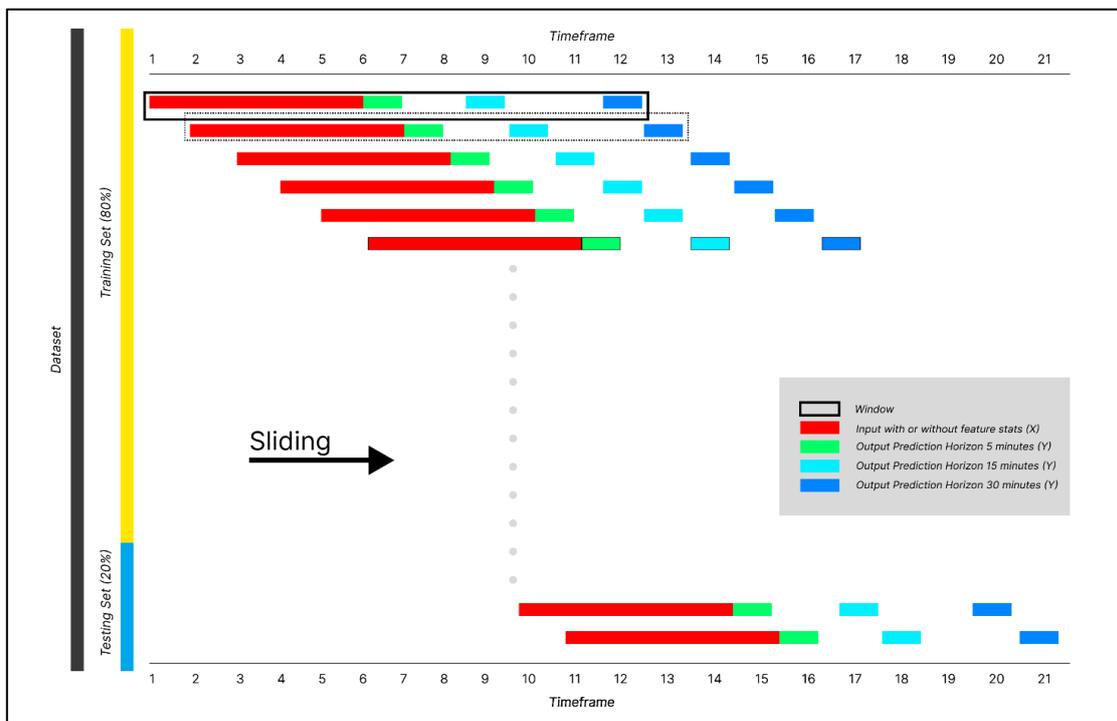
diinginkan untuk digunakan pada algoritma model *machine learning*. Pada tahap ini dilakukan pembagian data menjadi data latih dengan porsi 80% dan data uji dengan porsi 20% dari keseluruhan dataset. Masing-masing dalam data latih dan data uji di dalamnya terdapat data *input* dan *output*. Kemudian dilakukan ekstraksi fitur untuk mendapatkan fitur statistika dari *input* data latih dan data uji. Fitur statistika yang diekstraksi dan digunakan ditunjukkan pada Tabel 1.

Tabel 1. Fitur statistika yang diekstraksi

Nama	Deskripsi
Min	Nilai minimum dari kadar gula darah dalam satu <i>window</i>
Max	Nilai maksimum dari kadar gula darah dalam satu <i>window</i>
Mean	Nilai rata-rata dari kadar gula darah dalam satu <i>window</i>
Std	Nilai standar deviasi dari kadar gula darah dalam satu <i>window</i>
Diff	Nilai perbandingan dari kadar gula darah tertinggi dengan terendah dalam satu <i>window</i>
Median	Nilai tengah dari kadar gula darah dalam satu <i>window</i>
Kurtosis	Nilai keruncingan dari distribusi kadar gula darah dalam satu <i>window</i>
Skew	Nilai asimetris dari distribusi kadar gula darah dalam satu <i>window</i>

Gambar 2 menunjukkan ilustrasi mengubah data *time series* menjadi format *input* (x) dan *output* (y) dengan menggunakan pendekatan teknik *sliding windows*. Pertama, *dataset* dibagi dua menjadi data latih (*training set*) dan data uji (*testing set*) dengan porsi pembagian 80% dan 20%. Kemudian, data latih dan data uji tersebut diubah menjadi format masukan (X) dan keluaran (Y) dengan *Prediction*

Horizon (PH) yang diinginkan, pada penelitian ini digunakan PH 5 menit, 15 menit, dan 30 menit. Selanjutnya, data masukan dilakukan ekstraksi fitur statistik sebagai penambahan parameter masukan yang kemudian akan diuji dengan membandingkan performa model tanpa fitur statistik dengan model yang menggunakan fitur statistik.



Gambar 2. Ilustrasi data *time series* menggunakan pendekatan *sliding windows*

Model selection atau *modeling* adalah proses pemilihan model algoritma *machine learning* yang digunakan dalam melatih dan menguji dataset yang telah diolah sebelumnya. Pada penelitian ini menggunakan metode algoritma *Linear Regression*, *Lasso Regression*, *Ridge Regression*, *Support Vector Regression* (SVR), *eXtreme Gradient Boosting* (XGB), dan *K-Nearest Neighbor* (KNN). Model dari metode algoritma tersebut masing-masing akan dilatih menggunakan data latih yang sudah diproses pada tahap sebelumnya. Kemudian, akan dievaluasi menggunakan data uji pada tahap selanjutnya.

Penggunaan model algoritma pada implementasinya hanya menjalankan fungsi dalam pustaka python dengan memasukkan parameter yang dibutuhkan. Tetapi, dalam fungsi tersebut terdapat persamaan matematika yang dapat dihitung sendiri secara manual. Model algoritma *Linear Regression* dapat dihitung menggunakan persamaan (1-3).

$$Y_i = A + BX \quad (1)$$

$$A = \frac{(\sum Y)(\sum X^2) - (\sum X)(\sum XY)}{n(\sum X^2) - (\sum X)^2} \quad (2)$$

$$B = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2} \quad (3)$$

Dimana Y_i adalah nilai yang ingin diprediksi (*dependent variable*), X adalah nilai yang digunakan untuk memprediksi (*independent variable*), A adalah nilai *intercept* (nilai rata-rata pada variabel Y_i apabila nilai pada variabel X bernilai 0), dan B adalah nilai slope (nilai seberapa besar kontribusi yang diberikan oleh variabel X terhadap variabel Y).

Penelitian ini menggunakan nilai kadar gula darah pada 30 menit sebelumnya dengan jangka waktu 5 menit, sehingga variabel independen (X) memiliki lebih dari 1 nilai. Kasus tersebut disebut dengan *Multivariate Linear Regression* yang dapat dihitung menggunakan persamaan (4).

$$Y_i = A + B_1X_1 + B_2X_2 + \dots + B_nX_n \quad (4)$$

Dimana B_nX_n adalah nilai koefisien regresi dari independen variabel X , dan n merupakan jumlah banyaknya variabel independen atau titik datanya.

Model evaluation atau evaluasi performa model adalah langkah penting untuk menilai kualitas model prediksi. Beberapa metrik evaluasi yang dapat digunakan adalah *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), dan *coefficient of determination* (R^2). Semakin kecil nilai MSE, RMSE, MAE, dan MAPE serta semakin besar nilai R -squared, maka semakin baik performa model tersebut. Pada penelitian ini metrik yang digunakan adalah RMSE, MAE, dan R^2 untuk mengevaluasi performa model prediksi.

IV. HASIL DAN PEMBAHASAN

A. Evaluasi Performa Model

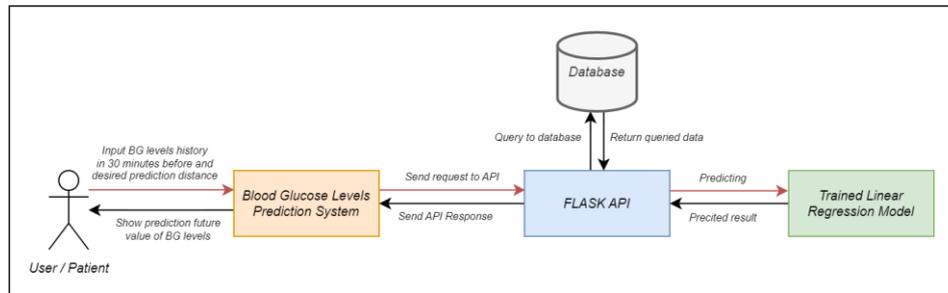
Setelah model prediksi dilatih dengan menggunakan data latih yang telah diolah sebelumnya, selanjutnya adalah melakukan pengujian pada model prediksi tersebut. Pengujian model prediksi ini berfungsi untuk mengevaluasi performa model algoritma *machine learning* yang digunakan untuk mendapatkan model dengan akurasi yang terbaik. Pengujian model prediksi dilakukan dengan melakukan prediksi pada model dengan menggunakan data uji. Data uji tersebut menggunakan 15 data pasien diabetes dari total 30 pasien dalam dataset. Hasil prediksi data uji tersebut dibandingkan dengan data aktual atau sebenarnya menggunakan metrik evaluasi. Metrik evaluasi yang digunakan dalam pengujian model prediksi ini adalah RMSE, MAE, dan R^2 . Performa model prediksi yang diuji dan dievaluasi yaitu model tanpa fitur statistik dengan model yang menggunakan fitur statistik.

Tabel 2 menunjukkan perbandingan metrik evaluasi pada model prediksi tanpa menggunakan fitur statistika, sedangkan Tabel 3 menunjukkan perbandingan metrik evaluasi pada model prediksi dengan menggunakan fitur statistika. Berdasarkan tabel perbandingan metrik evaluasi tersebut, model yang memiliki performa terbaik secara umum adalah regresi linear dengan rata-rata RMSE dan MAE yang paling kecil dan nilai R^2 yang tinggi. Selanjutnya metrik evaluasi performa model dengan fitur statistik dievaluasi untuk melihat apakah dengan menggunakan fitur statistika pada masukan model memiliki pengaruh pada performa model.

B. Sistem Prediksi Kadar Gula Darah Berbasis Website

Secara umum gambaran skema rancangan sistem yang akan dibangun ditunjukkan pada Gambar 3. Pada rancangan sistem tersebut pengguna memasukkan seluruh data yang dibutuhkan dalam sistem berbasis *website*, kemudian masukkan user tersebut akan dikirim sebagai *payload data* ke API yang dibangun menggunakan pustaka python flask, selanjutnya dalam API tersebut data yang diterima akan diproses menggunakan model prediksi yang telah dilatih. Hasil prediksi tersebut dikembalikan dalam bentuk *response data* API ke sistem berbasis *website* dan secara otomatis hasil prediksi tersebut ditampilkan pada halaman *website* sistem.

Halaman *get prediction* merupakan halaman fitur utama dalam sistem ini yang akan menampilkan hasil prediksi kadar gula darah. Pada prosesnya halaman di awal akan tampil masukkan yang harus diisi oleh pengguna, kemudian selanjutnya akan diproses data masukkan tersebut dan akan ditampilkan hasil prediksinya pada halaman selanjutnya. Hasil implementasi rancangan antarmuka halaman *get prediction* dapat dilihat pada Gambar 4. Setelah sistem berhasil memprediksi kadar gula darah selanjutnya, sistem akan menampilkan hasil prediksi berdasarkan waktu yang ditentukan berupa tampilan grafik atau diagram garis yang dapat dilihat pada Gambar 4.



Gambar 3. Ilustrasi skema rancangan sistem yang dibangun

Tabel 2. Perbandingan metrik evaluasi performa model tanpa menggunakan fitur statistik

Model	Metrik	Prediction Horizon (Menit)		
		5	15	30
Linear Regression	RMSE (mg/dL)	5.046±1.617	12.574±3.107	20.312±5.936
	MAE	2.875±1.386	8.212±2.204	14.62±4.469
	R ²	0.962±0.07	0.72±0.622	0.463±0.896
Lasso Regression	RMSE (mg/dL)	5.101±1.272	12.398±3.141	20.021±6.272
	MAE	3.032±1.114	8.242±2.131	14.557±4.722
	R ²	0.954±0.101	0.715±0.658	0.48±0.871
Ridge Regression	RMSE (mg/dL)	5.047±1.617	12.574±3.107	20.312±5.936
	MAE	2.876±1.387	8.212±2.204	14.62±4.469
	R ²	0.962±0.07	0.72±0.622	0.464±0.895
Support Vector Regression (SVR)	RMSE (mg/dL)	6.308±1.856	14.358±2.345	23.053±6.502
	MAE	4.247±1.97	10.513±2.357	17.618±5.319
	R ²	0.935±0.124	0.663±0.716	0.262±1.383
Extreme Gradient Boosting (XGB)	RMSE (mg/dL)	10.335±4.083	16.037±5.438	22.297±8.135
	MAE	6.99±2.689	11.757±4.152	17.093±6.383
	R ²	0.871±0.224	0.746±0.318	0.584±0.337
K-Nearest Neighbor (KNN)	RMSE (mg/dL)	6.473±1.967	15.047±2.425	24.584±7.033
	MAE	4.357±1.873	10.938±2.198	18.761±5.696
	R ²	0.946±0.076	0.666±0.638	0.236±1.271

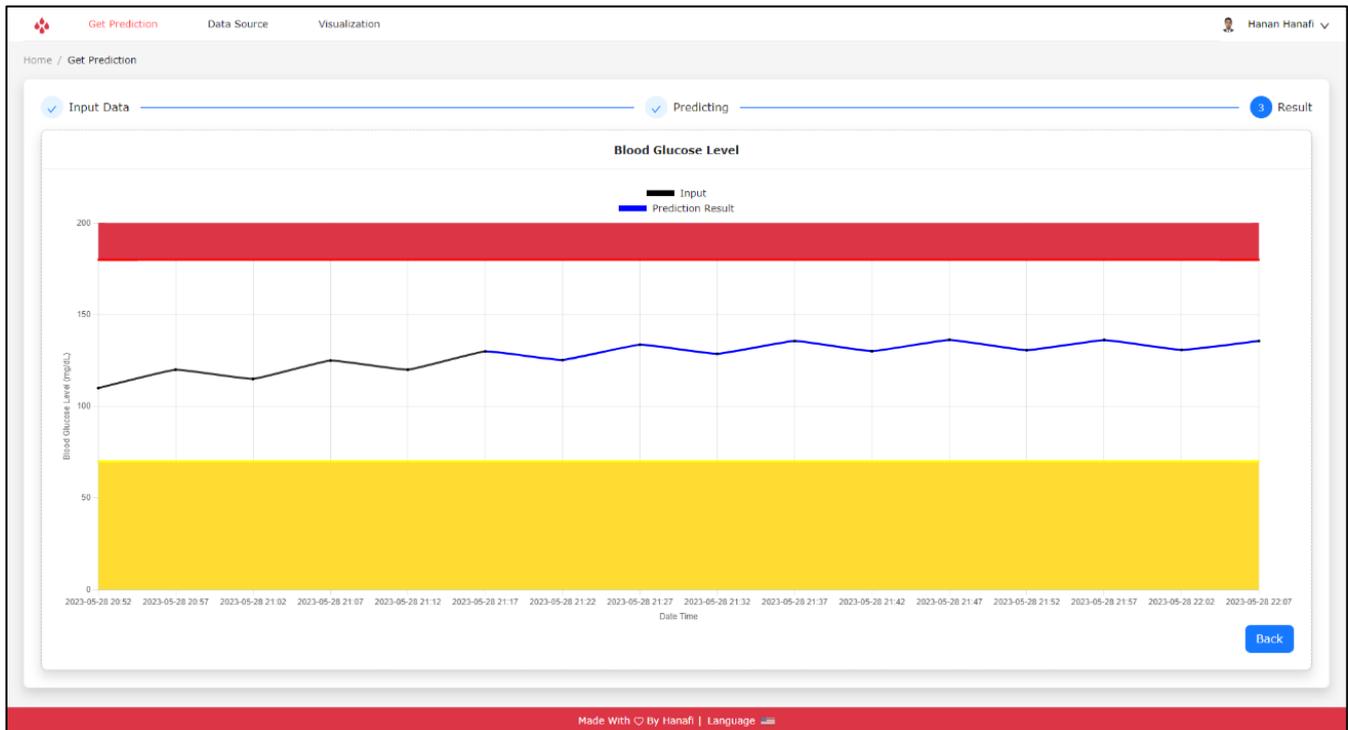
Tabel 3. Perbandingan metrik evaluasi performa model dengan menggunakan fitur statistik

Model	Metrik	Prediction Horizon (Menit)		
		5	15	30
Linear Regression	RMSE (mg/dL)	5.024±1.758	12.488±3.249	20.635±6.162
	MAE	2.891±1.449	8.272±2.368	14.926±4.71
	R ²	0.962±0.069	0.741±0.554	0.39±1.162
Lasso Regression	RMSE (mg/dL)	5.223±1.379	12.476±3.165	20.284±6.505
	MAE	3.089±1.187	8.333±2.195	14.752±4.94
	R ²	0.952±0.104	0.729±0.599	0.441±0.991
Ridge Regression	RMSE (mg/dL)	5.024±1.757	12.483±3.245	20.603±6.15
	MAE	2.89±1.448	8.256±2.346	14.905±4.691
	R ²	0.962±0.069	0.741±0.555	0.402±1.119
Support Vector Regression (SVR)	RMSE (mg/dL)	6.201±1.729	13.976±2.524	22.294±6.377

Model	Metrik	Prediction Horizon (Menit)		
		5	15	30
Extreme Gradient Boosting (XGB)	MAE	4.181±1.824	10.107±2.247	17.078±5.143
	R ²	0.938±0.119	0.687±0.662	0.318±1.307
	RMSE (mg/dL)	12.321±4.012	17.779±5.9	23.794±8.527
	MAE	9.134±3.026	13.531±4.646	18.552±6.617
K-Nearest Neighbor (KNN)	R ²	0.828±0.271	0.707±0.311	0.547±0.307
	RMSE (mg/dL)	6.465±1.528	14.105±2.23	23.519±6.523
	MAE	4.389±1.566	10.176±1.822	18.002±5.197
	R ²	0.943±0.091	0.715±0.554	0.351±0.993

Pada Gambar 4 juga terdapat informasi nilai minimum dan maksimum dari kadar gula darah. Kadar gula darah yang terlalu tinggi atau terlalu rendah dapat menimbulkan masalah kesehatan seperti komplikasi penyakit hingga

risiko kematian. Kadar gula darah yang terlalu tinggi atau *hyperglycemia* adalah kondisi dimana kadar gula darah melebihi 180 mg/dL, sedangkan kadar gula darah yang terlalu rendah atau *hypoglycemia* adalah kondisi dimana kadar gula darah kurang dari 70 mg/dL [14].



Gambar 4. Implementasi halaman *get prediction* bagian hasil prediksi

V. SIMPULAN

Berdasarkan hasil evaluasi performa model prediksi dan pengujian sistem prediksi yang telah dilakukan, didapat kesimpulan bahwa secara keseluruhan metode *Linear Regression* memiliki performa lebih baik daripada *Ridge Regression*, *Lasso Regression*, *Support Vector Regression* (SVR), *eXtreme Gradient Boosting* (XGB), dan *K-Nearest Neighbor* (KNN). Hasil evaluasi juga menunjukkan penambahan fitur statistik dapat meningkatkan performa model regresi linear dengan hasil pada *Prediction Horizon* (PH) 5 menit, 15 menit, dan 30 menit didapat nilai rata-rata

RMSE dari pengujian data 15 pasien adalah 5,024, 12,488, dan 20,635, nilai MAE adalah 2,891, 8,272, dan 14,926 serta nilai R² adalah 0,962, 8,272, dan 14,926. Kemudian, model prediksi regresi linear tersebut berhasil diimplementasi dan divisualisasikan ke dalam sistem berbasis *website*, sehingga sistem dapat digunakan oleh pengguna atau pasien diabetes untuk memprediksi kadar gula darah pada masa depan.

REFERENSI

- [1] Kementerian Kesehatan Republik Indonesia, "Pola Hidup Sehat dan Deteksi Dini Bantu Kontrol Gula Darah Pada Penderita Diabetes," 2022. <https://www.kemkes.go.id/article/view/21111600001/blood-sugar-control-for-dm-patient-through-healthy-lifestyle-and-early-detection.html>
- [2] World Health Organization, *Classification of diabetes mellitus*. Geneva: World Health Organization, 2019. [Online]. Available: <https://apps.who.int/iris/handle/10665/325182>
- [3] World Health Organization, *Global report on diabetes*. Geneva: World Health Organization, 2016. [Online]. Available: <https://apps.who.int/iris/handle/10665/204871>
- [4] World Health Organization, *Improving diabetes outcomes for all, a hundred years on from the discovery of insulin: report of the Global diabetes summit*. Geneva: World Health Organization, 2021. [Online]. Available: <https://apps.who.int/iris/handle/10665/350930>
- [5] Yuris Mulya Saputra, Ganjar Alfian, and Muhammad Qois Huzyan Octava, "Perancangan Federated Learning Berbasis Homomorphic Encryption untuk Perangkat Internet of Things," *J. Internet Softw. Eng.*, vol. 4, no. 1, May 2023, doi: 10.22146/jise.v4i1.6378.
- [6] Rochana Prih Hastuti, Vellya Riona, and Margareta Hardiyanti, "Content Retrieval dengan Fasttext Word Embedding pada Learning Management System Olimpiade," *J. Internet Softw. Eng.*, vol. 4, no. 1, May 2023, doi: 10.22146/jise.v4i1.6766.
- [7] S. M. Pappada *et al.*, "Neural Network-Based Real-Time Prediction of Glucose in Patients with Insulin-Dependent Diabetes," *Diabetes Technol. Ther.*, vol. 13, no. 2, pp. 135–141, Feb. 2011, doi: 10.1089/dia.2010.0104.
- [8] E. I. Georga *et al.*, "Multivariate Prediction of Subcutaneous Glucose Concentration in Type 1 Diabetes Patients Based on Support Vector Regression," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 1, pp. 71–81, 2013, doi: 10.1109/TITB.2012.2219876.
- [9] J. Ben Ali, T. Hamdi, N. Fnaiech, V. Di Costanzo, F. Fnaiech, and J.-M. Ginoux, "Continuous blood glucose level prediction of Type 1 Diabetes based on Artificial Neural Network," *Biocybern. Biomed. Eng.*, vol. 38, no. 4, pp. 828–840, Jan. 2018, doi: 10.1016/j.bbe.2018.06.005.
- [10] T. Hamdi, J. Ben Ali, V. Di Costanzo, F. Fnaiech, E. Moreau, and J.-M. Ginoux, "Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm," *Biocybern. Biomed. Eng.*, vol. 38, no. 2, pp. 362–372, Jan. 2018, doi: 10.1016/j.bbe.2018.02.005.
- [11] J. Martinsson, A. Schliep, B. Eliasson, and O. Mogren, "Blood Glucose Prediction with Variance Estimation Using Recurrent Neural Networks," *J. Healthc. Inform. Res.*, vol. 4, no. 1, pp. 1–18, Mar. 2020, doi: 10.1007/s41666-019-00059-y.
- [12] G. Alfian *et al.*, "Blood glucose prediction model for type 1 diabetes based on artificial neural network with time-domain features," *Biocybern. Biomed. Eng.*, vol. 40, no. 4, pp. 1586–1599, Oct. 2020, doi: 10.1016/j.bbe.2020.10.004.
- [13] A. Holzinger, "Big Data Calls for Machine Learning," in *Encyclopedia of Biomedical Engineering*, R. Narayan, Ed., Oxford: Elsevier, 2019, pp. 258–264. doi: 10.1016/B978-0-12-801238-3.10877-3.
- [14] American Diabetes Association, "Hypoglycemia (Low Blood Glucose)." <https://diabetes.org/healthy-living/medication-treatments/blood-glucose-testing-and-control/hypoglycemia>

Purwarupa Sistem Klasifikasi Legalitas Investasi Berbasis Algoritma *Bidirectional Long Short Term Memory*

Muhammad Navi Nugraha¹, Muhammad Arrofiq^{1,*}

¹Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
navinugraha15@gmail.com

*Korespondensi: rofiq@ugm.ac.id;

Abstract – *Illegal investment has become a serious problem in Indonesia, causing significant financial losses for investors and the economy. Manual verification of investment company legality has proven ineffective and time-consuming. Therefore, this research aims to enhance the effectiveness of investment legality verification by utilizing machine learning technology for automated prediction of investment messages. The study employs text classification methodology to categorize Indonesian investment messages from social media platforms, particularly Telegram, into legal or illegal categories. The research objective is to design a prototype investment legality detection system. The dataset used in this study is sourced from Telegram, comprising two variants with 2996 and 210 data points, respectively, covering legal and illegal investment messages. The research process involves data collection, preprocessing, classification model training, evaluation, and the development of a web application for investment message detection. Two algorithms, BiLSTM and LSTM, are implemented and compared in this research. The evaluation considers dataset size, dataset split ratio, utilization of BiLSTM and LSTM algorithms, confusion matrix analysis, and the web application. The findings reveal that the text classification model with BiLSTM algorithm and a dataset size of 2996 and 70:30 split ratio achieves the highest performance compared to other variants. The BiLSTM model achieves an accuracy of 96%, precision of 98%, and recall of 93%. The web application successfully detects investment messages using the top-performing model and operates according to the designated design.*

Keywords – *Illegal Investments, Text classification, LSTM, Telegram, Natural Language Processing, Ruby on Rails.*

Intisari – Investasi ilegal telah menjadi permasalahan serius di Indonesia, yang mengakibatkan kerugian finansial yang signifikan bagi investor dan perekonomian. Pengecekan legalitas perusahaan investasi secara manual belum efektif dan memakan waktu. Oleh karena itu, penelitian ini memanfaatkan teknologi *machine learning* dengan metode *text classification* untuk melakukan prediksi otomatis terhadap pesan investasi. Metode *text classification* digunakan dalam penelitian ini untuk mengategorikan pesan investasi berbahasa Indonesia dari media sosial, terutama Telegram, menjadi legal atau ilegal. Tujuan penelitian ini adalah merancang purwarupa sistem deteksi legalitas investasi. *Dataset* yang digunakan diperoleh dari media sosial Telegram, terdiri dari dua varian dengan jumlah 2996 data dan 210 data yang mencakup pesan investasi legal dan ilegal. Proses penelitian meliputi pengumpulan data, *preprocessing*, pelatihan model klasifikasi, evaluasi, dan pembuatan aplikasi web untuk deteksi pesan investasi. Dalam penelitian ini, dua algoritma, yaitu BiLSTM dan LSTM, diimplementasikan dan dibandingkan. Pengujian dilakukan dengan mempertimbangkan jumlah data dalam *dataset*, rasio pembagian *dataset*, penggunaan algoritma BiLSTM dan LSTM, *confusion matrix*, serta aplikasi web. Hasil pengujian menunjukkan bahwa model klasifikasi teks dengan algoritma BiLSTM dan jumlah data 2996 dengan rasio pembagian 70:30 memberikan performa tertinggi dibandingkan varian lain. Model dengan algoritma BiLSTM tersebut mampu mencapai akurasi sebesar 96%, presisi 98%, dan *recall* 93%. Aplikasi web berhasil mendeteksi pesan investasi menggunakan model dengan performa tertinggi dan berjalan sesuai dengan rancangan yang telah ditetapkan.

Kata kunci – *Investasi Ilegal, Text classification, LSTM, Telegram, Natural Language Processing, Ruby on Rails.*

I. PENDAHULUAN

Peningkatan jumlah investor di pasar modal Indonesia pada Desember tahun 2022 telah meningkat sebanyak 37,5 persen menjadi 10,3 juta investor dari sebelumnya hanya 7,48 juta investor per Desember 2021 [1]. Hal tersebut menandakan bahwa minat masyarakat Indonesia untuk berinvestasi meningkat dengan didukungnya simplikasi teknologi dalam pembukaan rekening investasi. Peningkatan jumlah investor tersebut menjadi pasar yang menarik untuk para pengusaha mendapatkan profit dengan cara menawarkan berbagai keuntungan investasi di platform masing-masing. Akan tetapi, ada juga peningkatan yang memprihatinkan dari penawaran investasi di Indonesia, yaitu kenaikan skema investasi ilegal di Indonesia yang menjadi isu kritical baik bagi investor maupun pihak berwenang.

Akhir-akhir ini kasus investasi ilegal marak terjadi di Indonesia sehingga menimbulkan kerugian bagi korban.

Investasi disebut ilegal apabila melakukan kegiatan dengan tidak memiliki izin dari regulator jasa keuangan karena tidak memenuhi syarat untuk melakukan kegiatan investasi. Begitu juga melakukan penipuan investasi pada instrumen saham, obligasi, komoditas, kemitraan atau jenis investasi lainnya yang dilakukan dengan janji dan kesepakatan yang menipu [2]. Kasus ini menyebabkan kerugian besar secara finansial bagi para investor maupun ekonomi. Sebagaimana pihak Satgas Waspada Investasi menyebutkan bahwa kerugian keseluruhan masyarakat Indonesia akibat penipuan investasi ilegal selama 5 tahun terakhir telah mencapai 123,5 triliun rupiah [3]. Pendekatan yang digunakan oleh pelaku investasi ilegal untuk menarik calon korban adalah melalui informasi iklan berbasis teks di media sosial dan platform *online*. Menurut survei Populix, dari seribu orang responden, mayoritas sebanyak 68 persen responden tertarik dengan investasi karena memperoleh informasi produk investasi melalui media sosial [4]. Dari kondisi tersebut menunjukkan

bahwa media sosial memiliki pengaruh yang signifikan terhadap minat calon investor untuk menaruh dana mereka dalam investasi ilegal.

Direktorat Jenderal Kekayaan Negara pada tahun 2022 menyarankan upaya pencegahan penipuan investasi ilegal di platform *online* dapat dilakukan dengan pengecekan legalitas perusahaan yang menawarkan investasi. Hendaknya perusahaan investasi sudah mendapatkan izin dan diawasi oleh regulator di sektor jasa keuangan. Salah satu cara untuk melakukan pengecekan legalitas perusahaan investasi secara manual adalah dengan mengakses situs resmi Otoritas Jasa Keuangan (OJK) dan otoritas keuangan lainnya. Namun, cara pengecekan secara manual dinilai kurang efektif karena setiap individu perlu melakukan pengecekan satu per satu legalitas di situs OJK sehingga memerlukan waktu lama bagi calon investor. Salah satu solusi yang lebih efektif adalah dengan menggunakan teknologi *machine learning* yang mampu melakukan prediksi secara otomatis. Metode *machine learning* yang dapat digunakan untuk melakukan klasifikasi berdasarkan legalitas investasi adalah *text classification*. Suatu teknik dari *Natural Language Processing* (NLP) yang dapat menetapkan kategori pada *unstructured text* dari data berformat teks [5]. Penelitian terdahulu tentang *text classification* menunjukkan bahwa dengan menggunakan algoritma Naive Bayes pada *text classification* memiliki hasil akurasi sebesar 82,6% untuk mendeteksi berita *hoax* berbahasa Indonesia secara daring [6]. Penelitian lain oleh Norman dkk. (2018) menambahkan bahwa penggunaan algoritma *Support Vector Machine* (SVM) dan *Natural Language Processing* (NLP) lebih efektif dan akurat dalam memprediksi kategori dari ulasan suatu produk pada *e-commerce*. Pratama [7] melakukan penelitian *text classification* data SMS berbahasa Indonesia untuk deteksi spam dengan mengimplementasikan algoritma *Long-Short Term Memory* (LSTM). Hasilnya menunjukkan bahwa LSTM (akurasi 94%) memiliki akurasi 27% lebih tinggi dibandingkan dengan metode Naive Bayes (akurasi 67%) dan 24% lebih tinggi dari metode K-Nearest Neighbor (KNN) (akurasi 70%). Penelitian serupa pernah dilakukan oleh Nugroho dkk. [8] dimana *text classification* untuk deteksi depresi dan kecemasan pengguna Twitter menggunakan pengembangan dari LSTM, yaitu *Bidirectional LSTM* memiliki tingkat akurasi yang lebih tinggi dibandingkan LSTM. Berdasarkan uraian di atas, maka akan dilakukan perbandingan performa BiLSTM dan LSTM dalam *text classification* pesan investasi. Algoritma yang menunjukkan performa terbaik diharapkan dapat diaplikasikan untuk mendeteksi legalitas investasi dari teks pesan investasi berbahasa Indonesia dan mengategorikannya menjadi ilegal atau legal. Kategorisasi ini dapat digunakan untuk mengidentifikasi investasi ilegal yang berisi informasi palsu atau penawaran keuntungan yang tidak realistis dari media sosial. Hal tersebut seringkali diindikasikan sebagai skema investasi ilegal.

Penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan mengembangkan sebuah sistem purwarupa yang dapat mengklasifikasikan legalitas investasi dengan

lebih akurat dan mampu mengekstrak informasi lebih lengkap. Sistem purwarupa ini akan mengimplementasikan *text classification* menggunakan algoritma *Bidirectional Long Short-Term Memory* (BiLSTM) dengan menerapkan pustaka algoritma BiLSTM dari Keras. Sistem akan mengambil teks berbahasa Indonesia yang telah diekstraksi oleh model yang dilatih di TensorFlow menggunakan *dataset* yang dikumpulkan dari grup investasi Telegram. Hasil deteksi akan ditampilkan aplikasi web interaktif yang dikembangkan menggunakan Ruby on Rails dan di-*deploy* di fly.io. Penelitian ini juga akan mencakup pengujian performa berupa akurasi, presisi, dan *recall* dari *text classification* menggunakan metode pengujian *confusion matrix*. Dengan memanfaatkan teknologi *machine learning* dengan metode *text classification* dan algoritma BiLSTM, diharapkan dapat membantu mengurangi risiko calon investor terkena investasi ilegal dengan mendeteksi legalitas investasi dan potensi penipuan investasi. Tidak hanya itu, sistem ini diharapkan dapat diimplementasikan dalam bidang pengawasan investasi untuk membantu mendeteksi investasi ilegal di Indonesia.

II. DASAR TEORI

Penipuan investasi di platform media sosial seperti Telegram telah menjadi masalah yang semakin meningkat dalam beberapa tahun terakhir. Putri dan Musyafak [9], meneliti jenis-jenis modus yang sering digunakan oleh pelaku penipuan di platform media sosial, termasuk Telegram. Mereka menemukan bahwa undian berhadiah dan penawaran investasi adalah modus yang paling sering digunakan. Studi ini menunjukkan bahwa pelaku penipuan sering menggunakan daya tarik keuntungan besar untuk menarik korban. Selain itu, pelaku penipuan juga sering menggunakan testimoni palsu atau memanipulasi data untuk menunjukkan bahwa investasi tersebut telah memberikan keuntungan besar kepada investor lain. Studi tersebut dapat dijadikan rujukan untuk mengidentifikasi pola penipuan investasi ilegal di Telegram.

Salah satu pendekatan yang dapat digunakan untuk mendeteksi penipuan investasi di Telegram adalah dengan menggunakan *text classification* yang merupakan salah satu metode dalam *machine learning*. Metode *text classification* memungkinkan komputer untuk mengategorikan teks ke dalam dua kategori, seperti ilegal atau legal.

Beberapa penelitian telah mengimplementasikan *machine learning* dalam *text classification*. Sebagai contoh, penelitian oleh Mutawalli dkk. [10] menggunakan metode (SVM) untuk melakukan *text classification* sosial media Twitter. Hasil penelitian menunjukkan bahwa SVM dapat menghasilkan akurasi yang cukup tinggi sebesar 80% dalam *text classification*.

Selain algoritma SVM, ada juga penelitian yang menggunakan algoritma lain untuk *text classification*. Contohnya, penelitian oleh Ardiani dkk [11] mengimplementasikan algoritma Naive Bayes untuk klasifikasi cuitan Twitter masyarakat terhadap pembangunan Kota Pontianak. Hasilnya menunjukkan bahwa Naive Bayes

mencapai tingkat akurasi sebesar 72% dari jumlah data latih sebesar 450 dan data uji sebesar 50.

Meskipun terdapat berbagai metode yang dapat digunakan dalam *text classification*, penggunaan algoritma LSTM memiliki performa yang lebih tinggi. Dalam sebuah penelitian yang dilakukan oleh Pratama [7], algoritma LSTM diimplementasikan untuk mengklasifikasikan teks data SMS spam berbahasa Indonesia. Hasil penelitian menunjukkan bahwa LSTM memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan dua algoritma lainnya, yaitu Naïve Bayes dan KNN. Algoritma LSTM mencapai tingkat akurasi sebesar 94%, sedangkan Naïve Bayes hanya sebesar 67% dan KNN sebesar 70%.

Studi serupa yang dilakukan oleh Nugroho dkk. [8] membandingkan kemampuan algoritma untuk mengidentifikasi kecemasan dan depresi pengguna Twitter. LSTM dan BiLSTM dibandingkan dengan semua model pembelajaran mesin tradisional. Pengujian dengan *confusion matrix* menunjukkan bahwa model algoritma tradisional seperti SVM, Naive Bayes, MLP, KNN, dan DT memiliki akurasi yang lebih rendah daripada LSTM dan BiLSTM. Namun, hasil pengujian yang dibandingkan menunjukkan bahwa BiLSTM memiliki akurasi 94.12%, sedangkan LSTM memiliki akurasi 84.91%.

Penelitian lain yang dilakukan oleh Fadli & Hidayatullah [12] mengimplementasikan BiLSTM dalam *text classification* untuk mengidentifikasi *cyberbullying* pada cuitan berbahasa Indonesia. Hasilnya menunjukkan bahwa BiLSTM mencapai tingkat akurasi 93.84% sedangkan dengan metode lain seperti LSTM tingkat akurasinya sebesar 92.02%. dan urutan kata yang kompleks dalam bahasa Indonesia.

Berdasarkan uraian di atas, sebagian besar penelitian terdahulu pada *text classification* menggunakan algoritma *machine learning*, seperti SVM, Naïve Bayes, dan LSTM. Sebagian besar penelitian tersebut masih berfokus pada *dataset* dari platform seperti Twitter dan SMS. Penelitian yang menggunakan *dataset* dari Telegram masih terbatas. Selain itu, algoritma BiLSTM dan LSTM memiliki performa yang lebih tinggi dibandingkan algoritma lainnya. Oleh karena itu, penggunaan algoritma LSTM dan Bi-LSTM dalam *text classification* perlu diuji performanya untuk mendeteksi penipuan investasi dari data pesan Telegram.

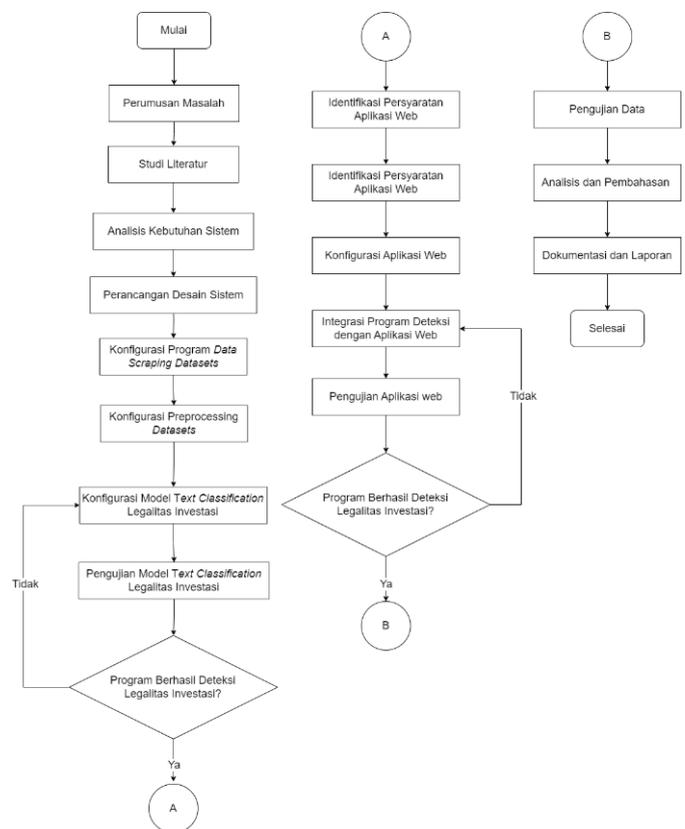
III. METODOLOGI

Penelitian ini terdiri dari beberapa tahapan yang dapat dilihat pada Gambar 1. Tahap awal adalah perumusan masalah dan studi literatur untuk memperdalam wawasan mengenai topik penelitian. Selanjutnya, kebutuhan sistem baik perangkat lunak maupun perangkat keras ditentukan. Tahap berikutnya adalah perancangan sistem dan instalasi program *scraper* untuk mengumpulkan *dataset* pesan investasi di Telegram. Setelah mengumpulkan *dataset*, langkah selanjutnya adalah melatih model *text classification* menggunakan algoritma BiLSTM dan LSTM, serta melakukan pengujian performa menggunakan metode

confusion matrix. Model dengan performa terbaik akan digunakan untuk purwarupa sistem deteksi investasi ilegal.

Selanjutnya, dilakukan pengujian *text classification* secara lokal menggunakan Flask API. Jika berhasil, dilanjutkan dengan pembuatan web interaktif menggunakan *framework* Ruby on Rails. Jika sistem deteksi tidak berhasil secara lokal, dilakukan *troubleshooting* pada konfigurasi program. Setelah berhasil membuat aplikasi web interaktif, dilakukan integrasi program deteksi dengan aplikasi web. Setelah integrasi selesai, dilakukan pengujian sistem deteksi menggunakan data yang telah disiapkan.

Sistem deteksi legalitas investasi diuji menggunakan data pesan iklan investasi dari *dataset* yang telah disiapkan. Hasil deteksi akan divisualisasikan pada halaman *website*, dan data hasil deteksi juga dapat diunduh dalam format CSV. Langkah terakhir adalah pembuatan laporan dan dokumentasi berdasarkan analisis dan pembahasan hasil pengujian sistem.



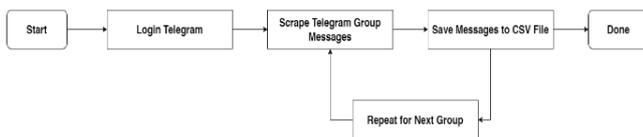
Gambar 1. Diagram alir

A. Alat dan Bahan

Dalam melaksanakan penelitian ini, dibutuhkan perangkat keras dan perangkat lunak guna menunjang penelitian yang akan dilakukan. Perangkat keras terdiri dari Laptop/PC dan SSD eksternal. Sedangkan perangkat lunak meliputi *Visual Studio Code*, Python, Google Collaboratory, PostgreSQL, Github, Tensorflow, Keras, Flask, Ruby on Rails.

B. Pengumpulan Data

Tahap pengumpulan *dataset* dimulai dengan melakukan *data scraping* pesan iklan investasi dari grup Telegram. Diagram alir pada Gambar 2, menunjukkan langkah-langkah dalam program tersebut, di mana program *login* ke Telegram menggunakan API *credentials* yang valid. Selanjutnya, program mengambil pesan-pesan dari grup yang telah ditentukan sebelumnya. Pesan-pesan yang berhasil diambil kemudian disimpan dalam *file* CSV. Proses pengambilan pesan dari grup Telegram diulang hingga semua grup telah diproses. Setelah semua grup diproses, program akan berakhir.



Gambar 2. Diagram alir telegram *scraping*

Dua *dataset* legal dan ilegal hasil *scraping* pesan investasi dari grup Telegram yang masih terpisah diberikan label secara manual untuk menandakan pesan tersebut termasuk kategori legal atau ilegal. Kemudian, *dataset* yang telah diberikan label digabungkan menjadi sebuah *dataset* yang di dalamnya berisi pesan legal dan ilegal. *Dataset* tersebut terdapat dua jenis yang digunakan dalam pengujian model *text classification*, yaitu *dataset* yang berjumlah 2996 data (Gambar 3) dan 210 data (Gambar 4).



Gambar 3. Komposisi jumlah label legalitas *dataset* 2996 data



Gambar 4. Komposisi jumlah label legalitas *dataset* 210 data

C. Preprocessing

Dataset yang berhasil dikumpulkan akan melalui tahap *preprocessing* sebagai proses membersihkan *noise* dan beberapa hal yang tidak diperlukan dalam *dataset*. Langkah-langkah *preprocessing* yang digunakan meliputi [7] (i) *Folding case* (ii) *Filtering* (iii) *Tokenisasi* (iv) *Stemming* berbahasa Indonesia (v) *Stopwords Removal*.

Berdasarkan Tabel 1, dapat diidentifikasi bahwa langkah-langkah *preprocessing* yang telah dilakukan berhasil menghilangkan *noise* dan meningkatkan kualitas data untuk digunakan dalam model *text classification*.

Tabel 1. *Dataset* setelah melalui *preprocessing*

Teks	Legalitas
belum sempat buka aplikasi akseleran suka tinggal campaign jangan khawatir akseleran punya fitur utk kamu yang sangat sibuk tp tetap bisa laku dana nama auto lending apa sih auto lending	1
real testimoni terima kasih telah joint di channel kami moga dana manfaat ya kak rizki arisha yang mau ikut investasi dana silah japri admin juwai musah	0

D. Text classification

Data berupa pesan investasi diklasifikasi ke dalam kelas-kelas legalitas berupa kelas legal dan kelas ilegal menggunakan algoritma BiLSTM dan LSTM. Secara detail, dilakukan perbandingan performa dengan dua varian *dataset* yaitu, dengan jumlah datanya 2996 dan 210. Selain itu, jumlah *epoch* yang digunakan untuk setiap model divariasikan dengan kelipatan 8 dengan nilai *epoch* paling rendah 16 dan paling tinggi 72. Hal tersebut dikarenakan nilai *epoch* yang paling sesuai untuk model tidak dapat dipastikan sehingga perlu menguji beberapa *epoch* untuk mencapai akurasi yang optimal. Proses pelatihan model dilakukan berulang-ulang untuk mencapai konvergensi nilai bobot. Rasio pembagian *data training* dan *data testing* yang digunakan adalah 30:70, 50:50, dan 70:30.

E. Evaluasi

Confusion matrix adalah sebuah matriks yang digunakan untuk mengevaluasi performa model klasifikasi dengan membandingkan hasil prediksi dan data aktualnya. Untuk menghitung kinerja model, dapat digunakan metrik seperti akurasi, presisi, dan *recall* berdasarkan rumus yang tercantum pada persamaan (1), (2), dan (3). Rumus-rumus tersebut adalah sebagai berikut:

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

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

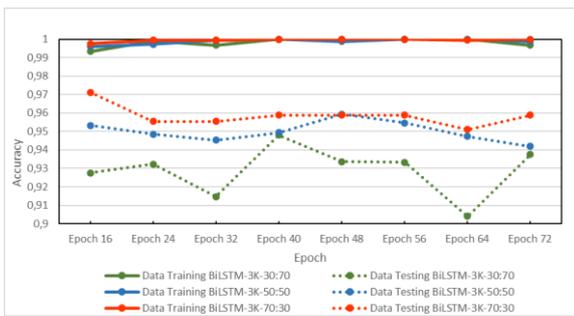
$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Model

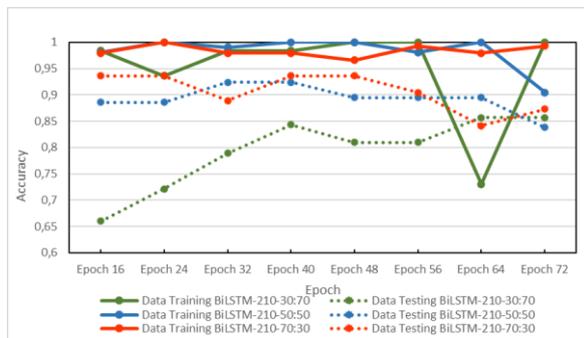
Pengujian pelatihan model dilakukan untuk mengetahui pengaruh jumlah *epoch* dan rasio pembagian data terhadap performa model dalam mengklasifikasikan pesan investasi selama pelatihan. Variasi jumlah *epoch* adalah 8, 16, 24, 32, 40, 48, 56, 64, dan 72. Sedangkan rasio pembagian *data training* dan *data testing* yang digunakan adalah 30:70, 50:50, dan 70:30.

Hasil pengujian pada varian model BiLSTM dengan 2996 data, seperti yang terlihat dalam Gambar 5, menunjukkan tingkat akurasi yang cenderung tinggi pada *epoch* 16, 48, dan 72. Namun, perlu diperhatikan bahwa pada *epoch* 72, varian dengan rasio pembagian 50:50 lebih akurat daripada varian lainnya. Pada *epoch* tersebut, model telah mencapai atau mendekati konvergensi dimana penyesuaian bobot model telah mencapai puncaknya. Selain itu, tingkat akurasi yang dicapai meningkat seiring dengan jumlah data latihan yang digunakan. Dengan rasio pembagian 70:30 antara data latihan dan pengujian, tingkat akurasi yang dicapai lebih tinggi.



Gambar 5. Perbandingan performa latih model BiLSTM varian 2996 data

Jika dibandingkan dengan model varian BiLSTM yang memiliki jumlah data 2996. Hasil pengujian model varian BiLSTM dengan jumlah data 210 berbeda. Hasil tersebut ditampilkan dalam Gambar 6.



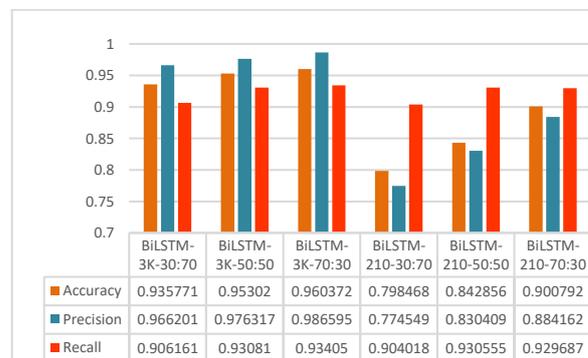
Gambar 6. Perbandingan performa latih model BiLSTM varian 210 data

Hasil pengujian model varian BiLSTM dengan jumlah data 210 digambarkan pada Gambar 6. Dalam pengujian varian model tersebut, model menunjukkan tingkat akurasi

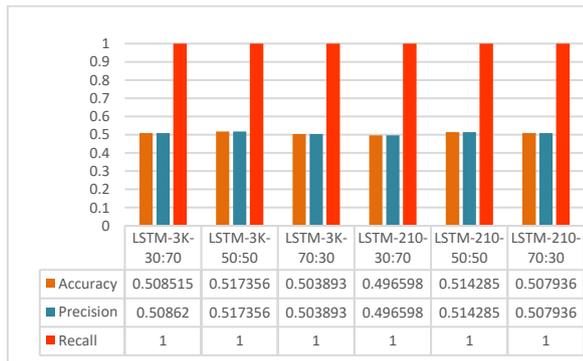
yang tinggi pada *epoch* 40, tetapi kemudian menurun setelahnya. Akurasi meningkat sedikit pada *epoch* 72 yang menunjukkan bahwa model telah mencapai atau hampir mencapai titik konvergensi saat pelatihan dilakukan pada *epoch* 40. Temuan juga menunjukkan bahwa semakin banyak data pelatihan yang digunakan dalam model, maka hasilnya semakin akurat. Hal ini ditunjukkan dengan perbandingan data 70:30 antara data pelatihan dan pengujian. Namun, model dengan jumlah 210 data ini menunjukkan kinerja yang lebih buruk dengan akurasi hanya 0,843. Hasil akurasi tersebut didapatkan dari menghitung rata-rata semua akurasi model dengan jumlah 210 data.

Pengujian selanjutnya adalah untuk mengetahui performa akurasi, presisi, dan *recall data testing* dengan pengukuran *confusion matrix*. Pengujian *confusion matrix* pada model *text classification* dilakukan pada dua algoritma, LSTM dan BiLSTM. Masing-masing algoritma diuji dengan dua varian *dataset*, masing-masing dengan 2996 data dan 210 data. Selain itu, *dataset* ini juga diuji dengan berbagai rasio pembagian data pelatihan dan pengujian, yaitu 30:70, 50:50, dan 70:30.

Berdasarkan Gambar 8, model LSTM tanpa *layer bidirectional* tidak mampu mengekstraksi informasi konteks secara optimal sehingga performanya rendah dibandingkan dengan model BiLSTM. Hal tersebut terlihat pada Gambar 8, bahwa semua performa varian model LSTM konstan sama. Hasil tersebut berbanding terbalik dengan performa varian model BiLSTM seperti tertampil pada Gambar 7. Varian model dengan split *dataset* rasio 70:30 dan *dataset* berjumlah 2996 data menghasilkan tingkat akurasi (96%), presisi (98%), dan *recall* (93,4%) yang lebih tinggi dibandingkan dengan varian lain dari model BiLSTM dan model LSTM. Hal ini mengindikasikan bahwa penggunaan *layer bidirectional*, jumlah *dataset* yang lebih banyak, dan rasio split *dataset* 70:30 memberikan performa signifikan yang berdampak pada peningkatan akurasi, presisi, dan *recall* dalam klasifikasi pesan investasi.



Gambar 7. Hasil perbandingan performa model BiLSTM

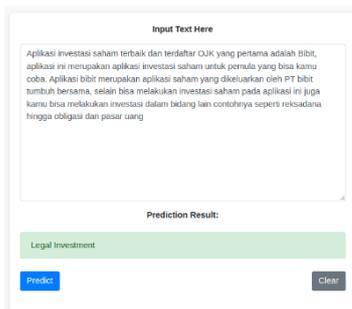


Gambar 8. Hasil perbandingan performa model LSTM

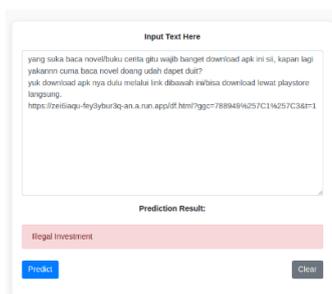
B. Hasil Pengujian Fungsionalitas Situs

Setelah menguji performa dari semua varian model, maka selanjutnya adalah menyimpan varian model dengan performa tertinggi ke dalam *file* berformat H5. Model tersebut digunakan sebagai API prediksi oleh sistem berbasis web.

Berdasarkan Gambar 9 dan 10, situs purwarupa ini berfungsi dengan baik. Situs dapat memprediksi pesan investasi ilegal dan legal. Selain itu, API model *text classification* dapat memberikan respons yang cepat dan efektif terhadap masukan teks pesan investasi.



Gambar 9. Hasil prediksi investasi legal



Gambar 10. Hasil prediksi investasi ilegal

V. SIMPULAN

Berdasarkan hasil pengujian yang telah dicapai, dapat disimpulkan bahwa Purwarupa Sistem Pencegahan Penipuan Investasi dengan Metode *Text classification* Berbasis Algoritma *Bidirectional LSTM* berhasil dirancang dan mampu memprediksi pesan investasi dengan performa akurasi, presisi, dan *recall* yang tinggi. Model yang menghasilkan performa tertinggi adalah varian model yang

menggunakan algoritma BiLSTM dengan jumlah 2996 data dan rasio pembagian *dataset* 70:30. Penelitian yang telah dilakukan masih memiliki beberapa kekurangan yang perlu diperbaiki dan dikembangkan di masa mendatang. Hal yang perlu dikembangkan adalah menambah *dataset*, menambah pengujian *epoch* dan *split dataset*, dan mengoptimalkan performa model yang menggunakan algoritma LSTM dengan menambahkan beberapa parameter.

REFERENSI

- [1] KSEI, "Pencapaian Tahun 2022 yang Positif, Tumbuhkan Semangat Baru pada Tahun 2023," 2022. [Online]. Available: https://www.ksei.co.id/files/uploads/press_releases/press_file/id-id/215_berita_pers_pencapaian_tahun_2022_yang_positif_tumbuhkan_semangat_baru_pada_tahun_2023_20230102182734.pdf.
- [2] W. Albrecht, C. A. C. Albrecht and M. Zimelman, *Fraud examination: Cengage Learning*, 2018.
- [3] cnnindonesia.com, "Rugi Masyarakat Akibat Investasi Bodong Melesat Jadi Rp109 T di 2022," 2022. [Online]. Available: <https://www.cnnindonesia.com/ekonomi/20221117193808-78-875287/rugi-masyarakat-akibat-investasi-bodong-melesat-jadi-rp109-t-di-2022>.
- [4] Databoks, "Media Sosial yang Digunakan Responden untuk Mencari Informasi Produk Investasi," Oktober 2022. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2022/12/01/ini-media-sosial-paling-efektif-untuk-promosi-produk-investasi>.
- [5] K. Kowsari, D. E. Brown, M. H., K. J. M., M. S. Gerber and L. E. Barnes, "HDLTex: Hierarchical Deep Learning for Text Classification," *16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 364-371, 2017.
- [6] F. Rahutomo, I. Y. R. Pratiwi and D. M. Ramadhani, "Eksperimen Naïvebayes pada Deteksi Berita Hoax Berbahasa Indonesia," 2019. [Online]. Available: <https://jurnal.kominfo.go.id/index.php/jpkop/article/view/1805/1122>
- [7] E. D. Pratama, "Implementasi Model Long-Short Term Memory (LSTM) pada Klasifikasi Teks Data SMS Spam Berbahasa Indonesia. The Journal on Machine Learning and Computational Intelligence (JMLCI), 1(2)," 2022. Available: <https://jmlci.unesa.ac.id/index.php/home/article/view/12/>.
- [8] K. S. Nugroho, I. Akbar and I. Affi Nizar S., "Deteksi Depresi dan Kecemasan Pengguna Twitter Menggunakan Bidirectional LSTM," 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2301.04521>.
- [9] F. A. Putri and N. Musyafak, "Pesan Komunikasi dalam Penipuan Digital Pada Masa Pandemi COVID-19," 2022. [Online]. Available: https://www.researchgate.net/profile/Najahan-Musyafak/publication/368660294_Pesan_Komunikasi_Dalam_Penipuan_Digital_Pada_Masa_Pandemi_Covid-19_Communication_Messages_In_Digital_Deception_During_The_Covid-19_Pandemic/links/63f36c6731cb6a6d1d19b551/PESAN-KO.
- [10] L. Mutawalli, M. T. A. Zaen and W. Bagye, "Klasifikasi Teks Sosial Media Twitter Menggunakan Support Vector Machine (Studi Kasus Penusukan Wiranto)," 2019. [Online]. Available: <https://ejournal.stmiklombok.ac.id/index.php/jire/article/view/117/77>.
- [11] L. Ardiani, H. Sujaini and Tursina, "Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak," *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, vol. 8(2), pp. pp.183-190, 2020.
- [12] H. F. Fadli and A. F. Hidayatullah, "Identifikasi Cyberbullying pada Media Sosial Twitter Menggunakan Metode LSTM dan BiLSTM," 2023. [Online]. Available: <https://journal.uui.ac.id/AUTOMATA/article/view/17364/10897>.

Analisis Tingkat Akurasi Metode Pendeteksian Plagiarisme Ide dengan Menggunakan *Yake* dan *Sentence Transformer*

Salsabila Laily Rahma¹, Umar Taufiq^{1,*}

¹Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
salsabilalaily@mail.ugm.ac.id

*Korespondensi: umartaufiq8284@ugm.ac.id;

Abstract – The utilization of Artificial Intelligence (AI) with unsupervised learning techniques can be beneficial in detecting idea plagiarism as it can automatically identify similarities and differences between textual documents without requiring labeled data or specialized training. Idea plagiarism involves inserting a summary from one text document into another, making it challenging to detect using conventional plagiarism detection methods. This research develops a method to address issues in detecting idea plagiarism and evaluates the accuracy level of the developed method. The approach utilizes a novel approach by leveraging Python libraries implementing AI techniques, specifically *Yake* as a keyword extraction algorithm and *Sentence Transformer* as a text similarity computation algorithm on the PAN dataset (Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection). The PAN dataset is publicly available and specifically designed for research in the field of plagiarism detection. The dataset used in this study is the PAN13-14 summary obfuscation dataset, and it contains ground truth data used as a reference for measuring the accuracy of the developed method. The research findings indicate that the *Sentence Transformer* method with *Yake* at a threshold of 0.1 achieves the highest accuracy for the Plagiarized category, with F-score values of 0.3175 and 0.3217 on the testing and training datasets, respectively. On the other hand, the *Sentence Transformer* method with a threshold of 0.6 achieves the highest accuracy for the Non Plagiarized category, with F-score values of 0.8905 and 0.8907 on the testing and training datasets, respectively.

Keywords – Artificial Intelligence, Idea Plagiarism Detection, *Yake*, *Sentence Transformer*, Python, Accuracy, PAN13-14 Dataset

Intisari – Penggunaan Artificial Intelligence dengan teknik *unsupervised learning* dapat berguna dalam pendeteksian plagiarisme ide karena dapat mengidentifikasi kemiripan dan perbedaan antara dokumen teks tanpa memerlukan data berlabel atau pelatihan khusus. Plagiarisme ide melibatkan penyisipan ringkasan dari satu dokumen teks ke dalam dokumen teks lainnya, sehingga membuatnya sulit terdeteksi menggunakan metode pendeteksian plagiarisme standar. Penelitian ini mengembangkan metode untuk mengatasi permasalahan dalam deteksi plagiarisme ide dan menguji tingkat akurasi level dokumen dari metode yang dikembangkan. Metode yang dikembangkan ini menggunakan pendekatan baru dengan memanfaatkan *library Python* yang mengimplementasikan AI pada teknik *unsupervised learning* yaitu metode *Yake* sebagai algoritma pengekstrak kata kunci dan *Sentence Transformer* sebagai algoritma untuk menghitung kemiripan teks pada dataset PAN. *Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (PAN)* adalah kumpulan data yang bersifat *public* dan secara khusus dikembangkan untuk penelitian dalam bidang pendeteksian plagiarisme. Dataset PAN yang digunakan dalam penelitian ini adalah dataset PAN13-14 summary obfuscation dengan sebuah *ground truth* yang menjadi acuan dalam pengukuran akurasi dari metode yang dikembangkan. Hasil penelitian menunjukkan bahwa metode *Sentence Transformer* dengan *Yake* pada *threshold* 0.1 memiliki akurasi tertinggi untuk kategori *Plagiarized* dengan nilai *F-score* pada dataset *testing* dan dataset *training* secara berturut-turut adalah 0.3175 dan 0.3217, sementara metode *Sentence Transformer* dengan *threshold* 0.6 memiliki akurasi tertinggi untuk kategori *Non Plagiarized* dengan nilai *F-score* pada dataset *testing* dan dataset *training* secara berturut-turut adalah 0.8905 dan 0.8907.

Kata kunci – Kecerdasan Buatan, Deteksi Plagiarisme Ide, *Yake*, *Sentence Transformer*, Python, Akurasi, Dataset PAN13-14

I. PENDAHULUAN

Seiring dengan bertambahnya jumlah penggunaan teknologi internet dan ketersediaan data yang semakin meningkat, hal ini membuat banyak akademisi, peneliti hingga pelajar yang menggunakan data dari internet untuk kebutuhan pribadi. Dalam rangka pencegahan tindakan plagiarisme yang marak terjadi, seringkali para pengguna mengubah teks, mengganti istilah hingga menggunakan sinonim sebagai pengganti kata aslinya, melakukan parafrase, mengganti kalimat aktif menjadi kalimat pasif atau sebaliknya hingga menggunakan teknik lainnya. Hal ini menjadikan tindakan plagiarisme ide atau gagasan menjadi sebuah masalah yang sulit dideteksi menggunakan metode pendeteksian plagiarisme standar [1].

Plagiarisme ide merupakan tindakan mengutip ide orang lain sebagai miliknya sendiri tanpa menyebutkan sumbernya. Plagiarisme dalam dokumen teks yang bersifat tersembunyi

(*obfuscated*) tidak dapat terdeteksi oleh metode deteksi plagiarisme yang hanya menyandarkan kinerjanya pada deteksi kesamaan teks saja. Inti permasalahan yang akan diselesaikan dalam penelitian ini adalah kesulitan dalam mendeteksi plagiarisme ide, terutama dalam dokumen teks yang memiliki *obfuscated plagiarism* yang sulit dideteksi oleh metode deteksi plagiarisme konvensional. Oleh karena itu, pengembangan metode deteksi plagiarisme ide dengan memanfaatkan teknologi informasi yang canggih perlu untuk dilakukan untuk mencegah plagiarisme dan menjaga keunikan sumber informasi. [2]

Pendeteksian plagiarisme mengacu pada serangkaian kegiatan di bidang teknologi informasi yang terdiri dari proses *preprocessing* dan normalisasi teks, representasi dokumen, hingga perhitungan kemiripan. Proses *preprocessing* dan normalisasi teks akan menghilangkan tanda baca, kata penghubung, *comments*, *blanks* dan *extra lines*. Proses

representasi dokumen adalah proses untuk merepresentasikan dokumen ke dalam bentuk yang dapat dibaca oleh sistem, dalam hal ini dapat dilakukan dengan perbandingan karakter *string* antara dua buah dokumen. Proses perhitungan kemiripan dapat dilakukan dengan mengimplementasikan algoritma untuk mendeteksi kasus plagiarisme dengan didasarkan pada *matrix* perangkat lunak [3].

Di dalam sistem pendeteksian plagiarisme, terdapat metodologi yang diterapkan yaitu *Artificial Intelligence* (AI) yang memanfaatkan teknik *unsupervised learning*. Hal ini dapat berguna dalam pendeteksian plagiarisme karena dapat menyimulasikan pemikiran manusia menggunakan data sehingga dapat menghasilkan perilaku tanpa campur tangan manusia dalam menanggapi prediksi tersebut. Keuntungan teknik *unsupervised learning* berupa dapat menggunakan data yang tidak berlabel untuk mencari struktur, pola, *cluster*, pengelompokan atau observasi terkait dalam data *input* [4].

Penelitian ini menghadirkan kontribusi baru dalam upaya pencegahan plagiarisme dengan mengusulkan penggunaan kombinasi algoritma yang menggunakan pendekatan teknik pembelajaran tanpa pengawasan (*unsupervised learning*) sebagai metode pendeteksian plagiarisme ide yang akan diuji pada *dataset summary obfuscation*. Pendekatan ini dianggap baru karena belum secara luas dikembangkan dan diterapkan dalam deteksi plagiarisme ide. Hal inilah yang mendorong penulis untuk melakukan penelitian dengan harapan akan memberikan kontribusi penting dan solusi yang lebih efektif dalam mengembangkan metode pendeteksian plagiarisme ide.

Yake merupakan sebuah *library Python* yang menggunakan pendekatan teknik pembelajaran tanpa pengawasan (*unsupervised learning*) sehingga tidak perlu dilatih pada kumpulan dokumen tertentu. *Yake* dapat digunakan untuk mengotomatiskan proses ekstraksi kata kunci secara cepat, dimana dapat mengidentifikasi kata kunci penting dan paling relevan dengan akurasi tinggi dalam teks. *Yake* akan menghasilkan daftar kata kunci *final* yang telah diurutkan berdasarkan skor relevansinya [5].

Sentence Transformer merupakan sebuah *framework Python* yang menggunakan pendekatan teknik pembelajaran tanpa pengawasan (*unsupervised learning*) sehingga tidak membutuhkan data *training* berlabel. *Sentence Transformer* dapat digunakan untuk penyematan kalimat dan teks dengan lebih dari 100 bahasa. Penyematan ini dapat dibandingkan dengan *cosine similarity* untuk menemukan kalimat dengan arti yang mirip. Hal ini dapat berguna untuk mencari kemiripan kalimat dalam teks [6].

Berdasarkan metode pendeteksian plagiarisme yang telah tersedia secara *public*, maka terdapat sebuah dasar pemikiran pada penelitian ini untuk mengembangkan metode pendeteksian plagiarisme ide dengan memanfaatkan *Yake* dan *Sentence Transformer*, serta menguji tingkat akurasi level dokumen pada metode pendeteksian plagiarisme ide yang dikembangkan pada dataset *PAN13-14 summary obfuscation*. Dataset *Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (PAN)* adalah kumpulan data yang bersifat *public* dan secara khusus dikembangkan untuk penelitian dalam bidang pendeteksian plagiarisme. Dataset *PAN13-14* yang digunakan terdiri dari dokumen teks

berbahasa Inggris yang terbagi menjadi *dataset testing set* dan *dataset training set*. Setiap *dataset* ini memiliki pasangan dokumen teks yang terdiri dari dokumen asli atau sumber (*source*) dan dokumen yang dicurigai melakukan plagiarisme terhadap dokumen asli (*suspicious*), serta dokumen *ground truth* yang berisi daftar pasangan dokumen *source* dan *suspicious* beserta status *plagiat* atau *bukan plagiat* [7].

Dalam penelitian ini, dibutuhkan pengujian metode pendeteksian plagiarisme dengan penetapan nilai *threshold* dari 0.1 - 0.9 untuk mengetahui tingkat akurasi yang dihasilkan. Penelitian ini juga menggunakan *ground truth* yang berguna untuk memvalidasi *threshold* yang mendekati kebenaran. *Yake* akan dikembangkan untuk mendeteksi plagiarisme dengan menghitung persamaan kata kunci dalam suatu pasangan dokumen teks. Metode *Sentence Transformer* juga dapat digunakan untuk mendeteksi kemiripan teks secara mandiri maupun dalam kombinasi dengan *Yake*. Perhitungan plagiarisme pada metode *Sentence Transformer* secara mandiri dilakukan dengan cara memanfaatkan model *transformer* yang dapat menghasilkan representasi vektor dari penyematan teks yang diambil dari pasangan dokumen sumber (*source*) dan dokumen yang mencurigakan (*suspicious*).

Representasi vektor yang dihasilkan dari *Sentence Transformer* akan dibandingkan menggunakan *cosine similarity* untuk menentukan tingkat kemiripan antara kalimat-kalimat. Sementara perhitungan plagiarisme pada metode *Sentence Transformer* dalam kombinasi dengan *Yake* dilakukan dengan cara mengotomatiskan proses ekstraksi kata kunci antar pasangan dataset, lalu kata kunci tanpa skor akan dimasukkan ke dalam model *transformer*. Hasilnya adalah representasi vektor yang dapat dibandingkan menggunakan *cosine similarity* untuk mengidentifikasi kalimat-kalimat dengan arti yang mirip sebagai indikasi adanya plagiarisme. Hasil penelitian menunjukkan bahwa metode *Sentence Transformer* dengan *Yake* pada *threshold* 0.1 memiliki akurasi tertinggi untuk kategori *Plagiarized* dengan nilai *F-score* pada *dataset testing* dan *dataset training* secara berturut-turut adalah 0.3175 dan 0.3217, sementara metode *Sentence Transformer* dengan *threshold* 0.6 memiliki akurasi tertinggi untuk kategori *Non Plagiarized* dengan nilai *F-score* pada *dataset testing* dan *dataset training* secara berturut-turut adalah 0.8905 dan 0.8907.

II. DASAR TEORI

Adanya metode pendeteksian plagiarisme *Yake* dan *Sentence Transformer* serta tersedianya *dataset PAN 13-14 summary obfuscation* yang telah dilengkapi dengan *ground truth*, memberikan dasar pemikiran untuk mengembangkan metode pendeteksian plagiarisme ide serta menguji tingkat akurasi level dokumen pada metode yang dikembangkan melalui pemanfaatan teknik *artificial intelligence*. Metode pendeteksian plagiarisme ini merupakan bentuk implementasi dari *artificial intelligence*, khususnya pada teknik *unsupervised learning*. *Yake* tidak perlu dilatih pada kumpulan dokumen tertentu untuk keperluan ekstraksi kata kunci pada teks, sementara *Sentence Transformer* tidak

membutuhkan data *training* berlabel untuk keperluan mencari kemiripan kalimat pada teks [2, 4, 5, 6, 8]. *Dataset PAN 13-14 summary obfuscation* dapat menjadi data yang digunakan pada penelitian untuk mengetahui tingkat akurasi level dokumen karena terdiri dari dataset *testing* dan *training* yang mana setiap *dataset* memiliki dokumen *source* dan *suspicious*. [7].

Penelitian terkait penerapan metode ekstraksi kata kunci dan perhitungan kemiripan semantik dengan penyematan dan perbandingan kalimat telah banyak dilakukan untuk optimisasi guna menguji akurasi dalam kinerjanya. Penelitian [9], sebuah sistem dibuat untuk melakukan ekstraksi frasa kunci dari data *social media* menggunakan *Sentence Transformer* dengan model *deep learning BERT* sebagai model pembelajaran mesin yang menggunakan proses penyematan *tweet* dan kemudian menggunakan matrik kemiripan untuk menghitung skor kemiripan. Selain itu, dilakukan penggabungan semua frasa kunci dalam satu dokumen dengan penerapan nilai ambang batas untuk proses pemeringkatan sehingga didapatkan frasa kunci terbaik. Penelitian [10], sebuah sistem dibuat untuk mengembangkan metode untuk *semantic textual similarity* (STS) dengan cara *SBERT* untuk mendapatkan penyisipan kalimat yang bermakna secara semantik yang dapat dibandingkan menggunakan *cosine similarity*.

Penelitian [11], sebuah sistem dibuat untuk implementasi relevansi teks guna menemukan dokumen topik tertentu melalui serangkaian proses yang terdiri dari *text preprocessing*, proses ekstraksi kata kunci dengan algoritma *Porter Stemming*, pembobotan kata kunci dengan cara perhitungan relevansi teks dengan *cosine similarity* yang akan menghasilkan nilai antara 0 dan 1, dimana nilai yang semakin mendekati 1 maka akan semakin tinggi kemiripan kedua dokumen. Penelitian [12], sebuah sistem bernama *FacTeR-Check* dibuat untuk melakukan pengecekan fakta atau mengekstraksi informasi mendalam tentang penyebaran hoaks dengan cara pengambilan *tweet* yang terkait hoaks untuk memfilter *tweet* yang paling relevan dengan *XLM-RoBERTa Transformer* untuk mengevaluasi kemiripan semantik dan ekstraksi kata kunci dengan *KeyBERT* serta menemukan kemiripan dengan menggunakan *Named-Entity Recognition*. Penelitian [13], sebuah sistem dibuat untuk mendeteksi plagiarisme pada dokumen teks dengan metode algoritma *Jaccard Similarity* pada dokumen *source* dan dokumen *suspicious* pada *dataset PAN* tahun 2013 dan 2014.

Penelitian-penelitian di atas hanya berfokus pada metode ekstraksi kata kunci dan perhitungan kemiripan semantik dengan penyematan dan perbandingan kalimat. Sejauh ini belum ada penelitian yang berfokus pada gabungan dari metode ekstraksi kata kunci dengan perhitungan kemiripan dengan *transformer* kalimat untuk mengukur akurasi level dokumen dalam hal deteksi plagiarisme ide pada *dataset* yang diakui dan digunakan oleh peneliti secara umum. Penelitian ini menggunakan pendekatan baru pada pengembangan metode pendeteksian plagiarisme ide yaitu dengan menggabungkan konsep ekstraksi kata kunci dan perhitungan kemiripan teks. Metode ini unik dan belum pernah diteliti sebelumnya sehingga patut dikembangkan dalam penelitian

sehingga menjadikan penelitian ini berpotensi memberikan hasil yang inovatif untuk keperluan analisis dan pengembangan metode di masa yang akan datang.

Dengan adanya teknologi serta ketersediaan *dataset* yang dapat mendukung penelitian, maka hal ini menciptakan sebuah ide atau dasar pemikiran pada penelitian ini yang bertujuan untuk untuk mengembangkan metode pendeteksian plagiarisme ide dengan memanfaatkan *Yake* dan *Sentence Transformer*, serta menguji tingkat akurasi level dokumen pada metode pendeteksian plagiarisme ide yang dikembangkan pada *dataset PAN13-14 summary obfuscation*. Pengukuran akurasi level dokumen menggunakan formula *Recall (Rd)*, *Precision (Pd)*, dan *F-score (Fd)* [14].

Penelitian ini menggunakan metode yang menjadi acuan karena sudah dikembangkan oleh peneliti sebelumnya yaitu metode *Sentence Transformer*, sementara metode yang dimodifikasi dalam penelitian adalah metode *Yake Similarity*, *Yake Similarity (Modif.#1)*, dan gabungan metode *Sentence Transformer* dengan tambahan *Yake*.

Metode *Yake Similarity* adalah sebuah metode yang dikembangkan dari algoritma *yake* dalam rangka menghitung kemiripan pada teks. Secara *default*, *yake* merupakan sebuah algoritma pengekstrak kata kunci. Penelitian ini mengusung pendekatan yang unik yaitu dengan menghitung persamaan kata kunci yang dihasilkan oleh *yake* pada dua dokumen teks *source* dan *suspicious*. Nilai kemiripan didapatkan dari hasil perhitungan jumlah kata kunci yang sama antara 2 dokumen (*irisan*) / jumlah kata kunci keseluruhan pada 2 dokumen. Berdasarkan data hasil pengujian setelah metode ini diimplementasikan, masih ditemukan adanya kelemahan pada metode ini yaitu nilai akurasi maksimal hanya berada pada 0.5 sehingga kelemahan metode ini perlu ditingkatkan.

Metode *Yake Similarity (Modif.#1)* adalah sebuah metode penyempurnaan dari metode *Yake Similarity* guna meningkatkan nilai akurasinya. Penelitian ini menghitung persamaan kata kunci yang dihasilkan oleh *yake* pada dua dokumen teks *source* dan *suspicious*. Nilai kemiripan didapatkan dari hasil perhitungan jumlah kata kunci yang sama antara 2 dokumen (*irisan*) / gabungan kata kunci antara 2 dokumen (*union*). Perhitungan kemiripan pada metode ini berhasil meningkatkan akurasi dari metode *Yake* sebelumnya, dengan rata-rata kenaikan akurasi pada *dataset testing* sebesar 0.001721797107 dan pada *dataset training* sebesar 0.001528264308.

Metode *Sentence Transformer* adalah sebuah metode yang menjadi sebuah acuan atau referensi karena metode ini sudah dikembangkan oleh peneliti sebelumnya. Metode ini mampu menemukan kemiripan kalimat dalam teks dengan teknik *sentence embeddings* dan perbandingan dengan *cosine similarity*. Dalam implementasinya, metode ini akan memasukkan kalimat atau paragraf dari dokumen *source* dan *suspicious* ke dalam model *transformer* sehingga menghasilkan sebuah *source embeddings* dan *suspicious embeddings* berupa vektor yang berukuran tetap. Kemudian kedua vektor ini dibandingkan dengan *cosine similarity*. *Cosine similarity* merupakan metode bawaan (*built-in*) dari *Sentence Transformer Library*.

Persamaan (1) dan (2) menunjukkan formula yang diterapkan dalam metode *cosine similarity* untuk membandingkan tingkat kemiripan dua buah vektor yang merupakan representasi dari teks. Persamaan tersebut menjadi landasan utama dalam metodologi penelitian ini.

$$\text{Cosine Similarity}(A, B) = (A \cdot B) / (||A|| * ||B||) \quad (1)$$

$$\frac{(a_1b_1+a_2b_2+a_3b_3+\dots+a_{384}b_{384})}{\sqrt{a_1^2+a_2^2+a_3^2+\dots+a_{384}^2}\sqrt{b_1^2+b_2^2+b_3^2+\dots+b_{384}^2}} \quad (2)$$

Metode *Sentence Transformer* dan *Yake* adalah sebuah metode yang dikembangkan dalam penelitian ini. Metode ini merupakan contoh implementasi dari gabungan konsep ekstraksi kata kunci dan perhitungan kemiripan teks sebagai sebuah pendekatan baru yang belum ada sebelumnya. Dalam implementasinya, metode ini akan melakukan proses ekstraksi kata kunci dengan *yake* lalu sistem akan mengambil kata kunci tanpa skor dari dokumen *source* dan *suspicious* untuk dimasukkan ke dalam model *transformer* sehingga menghasilkan *source embeddings* dan *suspicious embeddings* berupa vektor yang berukuran tetap untuk dibandingkan dengan *cosine similarity*. Dengan demikian, pengembangan metode ini diharapkan mampu menghasilkan nilai akurasi lebih baik karena *sentence transformer* dapat langsung memproses kata kunci paling relevan yang telah dihasilkan oleh *Yake*.

Penelitian ini dimaksudkan untuk menguji tingkat akurasi level dokumen pada metode pendeteksian plagiarisme ide yang dikembangkan. Hal ini cukup sulit dilakukan karena terdapat proses pengelompokan (*clustering*) *dataset* sesuai dengan *ground truth* yang harus diselesaikan sebelum proses perhitungan nilai kemiripan pada metode pendeteksian plagiarisme ide dilakukan. Penggunaan *dataset PAN 13-14 summary obfuscation* yang sudah dilengkapi dengan *ground truth* memberikan kelebihan yaitu dapat digunakan untuk memvalidasi *threshold* yang mendekati kebenaran.

Tabel 1 menunjukkan statistik *dataset PAN13-14* yang digunakan dalam penelitian. Hal ini bertujuan agar sistem dapat menguji setiap pasangan *dataset (source document dan suspicious document)* seperti yang telah ditentukan pada *ground truth* agar tidak menimbulkan banyak kombinasi jika membandingkan dokumen *source* dan dokumen *suspicious* secara keseluruhan.

Tabel 1. Statistik dataset pada *PAN13-14 summary obfuscation* menurut Potthast dkk [15]

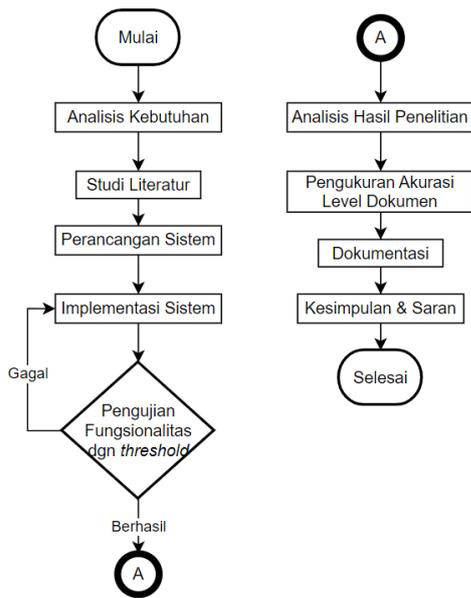
Documents	Training set		Testing set	
	suspicious	source	suspicious	source
Non-Plagiarized	947	-	949	-
Plagiarized	238	-	236	-
Total	1185	237	1185	237

III. METODOLOGI

Metode penelitian yang dilakukan berfokus pada pengembangan metode dan pengukuran akurasi level dokumen pada metode yang dikembangkan dengan memanfaatkan *Yake* dan *Sentence Transformer*.

- A. Analisis Kebutuhan yaitu dengan melakukan riset mengenai kebutuhan dan spesifikasi sistem, serta kebutuhan perangkat lunak dan perangkat keras yang digunakan untuk mendukung penelitian.
- B. Tahap perancangan yaitu dengan membuat rencana dan rancangan mengenai program sistem yang terdiri atas dua kegiatan yaitu studi literatur dan perancangan sistem. Studi literatur dilakukan dengan mengumpulkan literatur-literatur yang berkaitan dengan perangkat lunak. Perancangan sistem dilakukan dengan mempersiapkan dataset, instalasi dan konfigurasi sistem, perancangan struktur direktori dan desain sistem yang akan digunakan pada penelitian.
- C. Implementasi teknik ekstraksi kata kunci dan perhitungan kemiripan teks dengan *Yake* dan *Sentence Transformer*. Langkah-langkah yang perlu dijalankan pada tahapan ini adalah seperti pembuatan file berisi kata kunci, pengelompokan file berisi kata kunci dan dataset PAN, pembuatan file berisi nilai kemiripan dengan metode pendeteksian plagiarisme, serta pengujian *threshold* pada metode pendeteksian plagiarisme.
- D. Pengujian sistem dilakukan dengan cara pengujian fungsionalitas sistem untuk memastikan bahwa proses menerima *input* dan menghasilkan *output* sudah berfungsi sebagaimana mestinya dan pengujian terkait dengan pengukuran tingkat akurasi level dokumen pada metode pendeteksian plagiarisme dengan menghitung *recall*, *precision* dan *F-score* berdasarkan data jumlah pasangan dokumen yang diidentifikasi oleh sistem sebagai *plagiat* maupun *non plagiat* (baik yang sukses maupun gagal terdeteksi).

Gambar 1 menunjukkan langkah-langkah proses penelitian dengan digambarkan menggunakan diagram alir yang memvisualisasikan secara sistematis urutan proses yang dilakukan dalam penelitian. Proses yang tertera pada penelitian ini dilakukan sesuai dengan metodologi yang telah ditetapkan. Metodologi yang mendasari penelitian ini adalah metode *waterfall*. Pendekatan ini dipilih sebagai kerangka kerja yang paling sesuai untuk mencapai tujuan penelitian dengan efektif dan efisien, memastikan kelancaran pelaksanaan serta konsistensi hasil yang diperoleh dalam penelitian.



Gambar 1. Skema rangkaian yang digunakan pada penelitian

IV. HASIL DAN PEMBAHASAN

Untuk melakukan evaluasi terhadap keakuratan metode pendeteksian plagiarisme ide dengan menggunakan *Yake* dan *Sentence Transformer*, dilakukan pengujian dengan menetapkan nilai ambang batas *threshold* mulai dari 0.1 - 0.9 untuk menentukan apakah suatu dokumen dikatakan plagiat atau tidak. Pengujian ini dilakukan untuk dataset *PAN13-14 summary obfuscation* dengan jumlah dokumen *source* dan dokumen *suspicious* serta *ground truth* yang dapat digunakan. Pada dataset *testing*, terdiri dari jumlah dokumen *source* sebanyak 3169 dan jumlah dokumen *suspicious* sebanyak 1826. Pada dataset *training*, terdiri dari jumlah dokumen *source* sebanyak 3230 dan jumlah dokumen *suspicious* sebanyak 1827. Semua eksperimen dilakukan dengan menggunakan bahasa pemrograman *Python* dengan menguji 1185 pasangan dokumen *source* dan dokumen *suspicious* pada dataset *testing* dan dataset *training* berdasarkan data *ground truth*.

Selanjutnya perbandingan akurasi tingkat dokumen antara metode *Yake* dengan *Sentence Transformer* dapat dijelaskan sebagai berikut. Sesuai dengan ekspektasi seperti yang tampak pada Tabel 2 - 5, hasil penelitian yang diperoleh menunjukkan bahwa pada kategori *plagiarized*, tingkat akurasi tertinggi diperoleh pada kombinasi metode *Yake* dan *Sentence Transformer* (*threshold* 0.1), sebagaimana dibuktikan dengan adanya nilai *F-score* tertinggi pada dataset *testing* dan dataset *training* secara berturut-turut adalah 0.3175 dan 0.3217. Sementara pada kategori *non plagiarized*, tingkat akurasi tertinggi diperoleh pada metode *Sentence Transformer* (*threshold* 0.6) dengan nilai *F-score* pada dataset *testing* dan dataset *training* yaitu sebesar 0.8905 dan 0.8907.

Akurasi yang diperoleh dari metode *Sentence Transformer* dengan tambahan *Yake* dapat meningkatkan akurasi dalam deteksi plagiarisme ide. Hal ini dikarenakan

Yake mampu memberikan kata kunci paling relevan dari teks dengan akurasi tinggi sehingga kinerja *Sentence Transformer* dalam mencari kemiripan dalam kalimat akan semakin baik. Hasilnya adalah berupa sistem dapat mengidentifikasi jumlah pasangan dokumen lebih banyak daripada metode lain yang dikembangkan dalam penelitian.

Hasil penelitian ini juga menunjukkan bahwa akurasi pada metode *Yake* tanpa adanya metode *Sentence Transformer* perlu ditingkatkan. Metode *Yake* menunjukkan ketidakseimbangan performa metode. Hal ini dapat dilihat pada akurasi yang terlalu rendah pada kategori *Plagiarized* dan akurasi yang terlalu tinggi pada kategori *Non Plagiarized*. Oleh karena itu, metode ini tidak dijadikan sebagai hasil kesimpulan meskipun memiliki akurasi tertinggi. Berbeda halnya dengan metode *Sentence Transformer* yang mampu membedakan atau mengklasifikasikan kasus *Plagiarized* dan *Non Plagiarized* dengan lebih baik berdasarkan variasi *threshold*. Sementara pada metode *Sentence Transformer* dengan tambahan metode *Yake* mampu meningkatkan akurasi level dokumen menjadi lebih baik.

Tabel 2 menunjukkan perbandingan *F-score* pada metode pendeteksian plagiarisme yang diujikan pada dataset *testing* pada kategori *Plagiarized*. Berdasarkan hasil pengujian, akurasi level dokumen tertinggi pada kategori *Plagiarized* yaitu terdapat pada *threshold* 0.1 dengan menggunakan metode kombinasi antara *Yake* dan *Sentence Transformer*. Hal ini dapat dilihat bahwa pada kasus *plagiarized*, nilai *Fd* tertinggi pada dataset *testing* adalah 0.3175.

Tabel 2. Akurasi pada dataset testing (Plagiarized)

<i>Threshold</i>	<i>Yake</i>	<i>Sentence</i>	<i>Sentence Yake</i>
0.1	0.2321	0.2967	0.3175
0.2	0.0574	0.2496	0.2712
0.3	-	0.1838	0.1269
0.4	-	0.1188	0.0752
0.5	-	0.0785	0.04
0.6	-	0.0331	0.0402
0.7	-	0.0084	0.0402
0.8	-	-	0.0324
0.9	-	-	0.0247

Tabel 3 menunjukkan perbandingan *F-score* pada metode pendeteksian plagiarisme yang diujikan pada dataset *training* pada kategori *Plagiarized*. Berdasarkan hasil pengujian, akurasi level dokumen tertinggi pada kategori *Plagiarized* yaitu terdapat pada *threshold* 0.1 dengan menggunakan metode kombinasi antara *Yake* dan *Sentence Transformer*. Hal ini dapat dilihat bahwa pada kasus *plagiarized*, nilai *Fd* tertinggi pada dataset *training* adalah 0.3217.

Tabel 3. Akurasi pada dataset training (Plagiarized)

Threshold	Yake	Sentence	Sentence Yake
0.1	0.2369	0.3194	0.3217
0.2	0.0084	0.2454	0.2655
0.3	-	0.1715	0.1798
0.4	-	0.1088	0.1194
0.5	-	0.0775	0.0923
0.6	-	0.0569	0.0781
0.7	-	0.0084	0.056
0.8	-	-	0.056
0.9	-	-	0.056

Tabel 4 menunjukkan perbandingan *F-score* pada metode pendeteksian plagiarisme ide yang diujikan pada *dataset testing* pada kategori *Non Plagiarized*. Berdasarkan hasil pengujian, metode *Sentence Transformer* dengan *threshold* 0.6 terbukti memiliki akurasi tertinggi dalam mendeteksi pasangan dokumen yang diidentifikasi sebagai *bukan plagiat* dengan nilai *F-score* yaitu 0.8905.

Tabel 4. Akurasi pada dataset testing (Non Plagiarized)

Threshold	Yake	Sentence	Sentence Yake
0.1	0.8917	0.5458	0.4552
0.2	0.8918	0.7673	0.7707
0.3	0.8894	0.849	0.8583
0.4	0.8894	0.879	0.8831
0.5	0.8894	0.8889	0.8868
0.6	0.8894	0.8905	0.8873
0.7	0.8894	0.8898	0.8873
0.8	0.8894	0.8894	0.888
0.9	0.8894	0.8894	0.8886

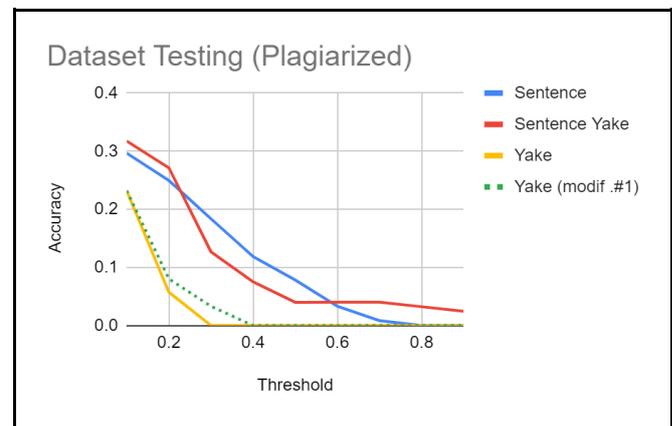
Tabel 5 menunjukkan perbandingan *F-score* pada metode pendeteksian plagiarisme ide yang diujikan pada *dataset training* pada kategori *Non Plagiarized*. Berdasarkan hasil pengujian, metode *Sentence Transformer* dengan *threshold* 0.6 terbukti memiliki akurasi tertinggi dalam mendeteksi pasangan dokumen yang diidentifikasi sebagai *bukan plagiat* dengan nilai *F-score* yaitu 0.8907.

Gambar 2 menunjukkan grafik perbandingan *F-score* pada metode pendeteksian plagiarisme ide yang diujikan pada *dataset testing* pada kategori *Plagiarized*. Dalam hal ini, metode yang dimodifikasi dalam penelitian adalah metode *Sentence Transformer* dan *Yake* (merah), *Yake* (kuning), dan *Yake modif.#1* (hijau), sementara metode yang menjadi acuan karena sudah dikembangkan oleh peneliti sebelumnya adalah metode *Sentence Transformer* (biru).

karena sudah dikembangkan oleh peneliti sebelumnya adalah metode *Sentence Transformer* (biru).

Tabel 5. Akurasi pada dataset training (*Non Plagiarized*)

Threshold	Yake	Sentence	Sentence Yake
0.1	0.8877	0.543	0.4539
0.2	0.8888	0.747	0.762
0.3	0.8884	0.8393	0.8602
0.4	0.8884	0.8738	0.8877
0.5	0.8884	0.8873	0.8882
0.6	0.8884	0.8907	0.8884
0.7	0.8884	0.8888	0.8887
0.8	0.8884	0.8884	0.8887
0.9	0.8884	0.8884	0.8887

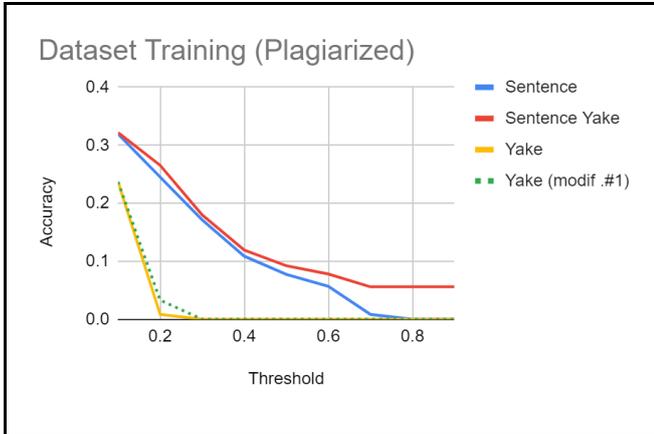
Gambar 2. Perbandingan Akurasi pada *Dataset Testing* untuk kategori *Plagiarized*

Gambar 3 menunjukkan grafik perbandingan *F-score* pada metode pendeteksian plagiarisme ide yang diujikan pada *dataset training* pada kategori *Plagiarized*. Dalam hal ini, metode yang dimodifikasi dalam penelitian adalah metode *Sentence Transformer* dan *Yake* (merah), *Yake* (kuning), dan *Yake modif.#1* (hijau), sementara metode yang menjadi acuan karena sudah dikembangkan oleh peneliti sebelumnya adalah metode *Sentence Transformer* (biru).

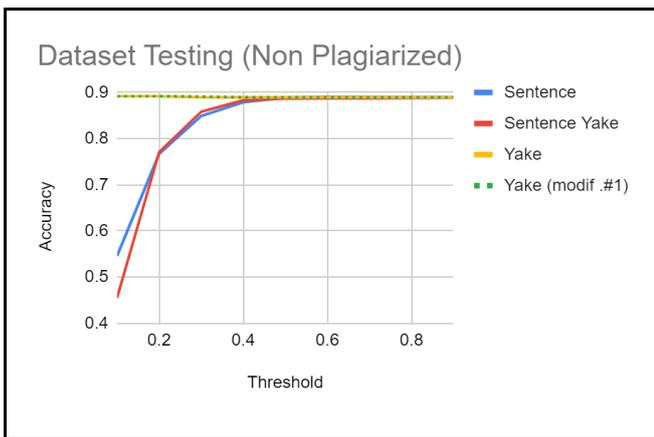
Gambar 4 menunjukkan grafik perbandingan *F-score* pada metode pendeteksian plagiarisme ide yang diujikan pada *dataset testing* pada kategori *Non Plagiarized*. Dalam hal ini, metode yang dimodifikasi dalam penelitian adalah metode *Sentence Transformer* dan *Yake* (merah), *Yake* (kuning), dan *Yake modif.#1* (hijau), sementara metode yang menjadi acuan karena sudah dikembangkan oleh peneliti sebelumnya adalah metode *Sentence Transformer* (biru).

Gambar 5 menunjukkan grafik perbandingan *F-score* pada metode pendeteksian plagiarisme ide yang diujikan pada *dataset training* pada kategori *Non Plagiarized*. Dalam hal ini, metode yang dimodifikasi dalam penelitian adalah metode

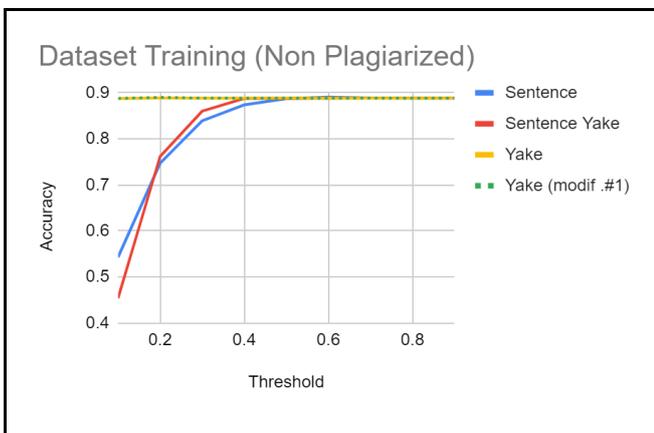
Sentence Transformer dan *Yake* (merah), *Yake* (kuning), dan *Yake modif.#1* (hijau), sementara metode yang menjadi acuan karena sudah dikembangkan oleh peneliti sebelumnya adalah metode *Sentence Transformer* (biru).



Gambar 3. Perbandingan Akurasi pada *Dataset Training* untuk kategori *Plagiarized*



Gambar 4. Perbandingan Akurasi pada *Dataset Testing* untuk kategori *Non Plagiarized*



Gambar 5. Perbandingan Akurasi pada *Dataset Training* untuk kategori *Non Plagiarized*

V. SIMPULAN

Pada penelitian ini telah dibuat sebuah rancang bangun sistem pendeteksian plagiarisme ide dengan metode *Yake* dan *Sentence Transformer* untuk mengukur akurasi level dokumen pada *dataset PAN13-14 summary obfuscation*. Berdasarkan hasil penelitian, dapat disimpulkan bahwa semakin rendah nilai batas *threshold* yang ditetapkan dalam suatu pengujian, maka jumlah pasangan dokumen yang terdeteksi sebagai *plagiat* dalam sistem akan semakin meningkat, sementara jumlah pasangan dokumen yang terdeteksi sebagai *bukan plagiat* akan semakin menurun.

Berdasarkan penelitian yang telah dilakukan, maka diperoleh beberapa saran dalam upaya pengembangan dan perbaikan penelitian di masa depan. Adapun saran yang dapat dilakukan untuk peneliti selanjutnya meliputi a.) melakukan penelitian dengan mengembangkan metode untuk meningkatkan akurasi level dokumen, b.) melakukan penelitian dengan menggunakan *dataset* selain berbahasa Inggris, c.) melakukan penelitian dengan menggunakan sampel *dataset* yang lebih besar.

REFERENSI

- [1] V. Bhuyar and S.N. Deshmukh. 2023. "Analysis of Support Tools for Plagiarism Detection." International Conference on Applications of Machine Intelligence and Data Analytics: ICAMIDA 2022, ACSR 105, pp. 38-46, 2023.
- [2] Taufiq, Umar. 2023. Named Entity Recognition dan Dependency Parsing untuk Ekstraksi Konsep yang Lebih Baik dalam Deteksi Plagiarisme. Disertasi. Program Studi S3 Ilmu Komputer. Universitas Gadjah Mada. Yogyakarta.
- [3] A. S. Bin-Habtoor and M.A. Zaher. 2012. "A Survey on Plagiarism Detection Systems," International Journal of Computer Theory and Engineering, vol. 4, no. 2, pp. 1-5.
- [4] H. Garg, J. M. Chatterjee, R. N. Thakur. 2023. "A Roadmap For Enabling Industry 4.0 By Artificial Intelligence," USA: John Wiley & Sons, Inc. and Scrivener Publishing LLC.
- [5] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt. 2019. "Yake! Keyword Extraction from Single Documents using Multiple Local Features," Information Sciences, vol. 509, 2020, pp. 257-289, Elsevier Inc.
- [6] Reimers, Nils. 2022. "Sentence Transformer Documentation," dilihat pada 6 Juni 2023 halaman situs <https://www.sbert.net/>
- [7] M. N. Mansoor and M.S.H. Al-Tamimib, 2022. "Computer-based plagiarism detection techniques: A comparative study," 13th International Journal of Nonlinear Analysis and Applications (IJNAA), 2022, pp. 10-12.
- [8] Patel, Ankur A., 2019. "Hands-On Unsupervised Learning Using Python: How to Build Applied Machine Learning Solutions from Unlabeled Data," USA: O'Reilly Media, Inc. (Penerbit O'Reilly), pp. 3-26. dilihat pada 12 Juni 2023 halaman situs https://www.google.co.id/books/edition/Hands_On_Unsupervised_Learning_Using_Pyt/-SKJDwAAQBAJ?hl=en&gbpv=1&dq=unsupervised+learning+algorith+AI&printsec=frontcover
- [9] R. Devika, S. Vairavasundaram, C. S. J. Mahenthara, V. Varadarajan and K. Kotecha, "A Deep Learning Model Based on BERT and Sentence Transformer for Semantic Keyphrase Extraction on Big Social Data," in *IEEE Access*, vol. 9, pp. 165252-165261, 2021, doi: 10.1109/ACCESS.2021.3133651.
- [10] Reimers, Nils and Gurevych, Iryna. 2019. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," dilihat pada 6 Juni 2023 halaman situs <https://arxiv.org/abs/1908.10084>.

- [11] D. Gunawan, C.A. Sembiring, M.A Budiman, "The Implementation of Cosine Similarity to Calculate Text Relevance Between Two Documents," 2018. IOP Conference Series: Journal of Physics (J.Phys.), pp. 1-6.
- [12] A. Martin, J. Huertas-Tato, A. Huertas-García, G. Villar-Rodríguez, D. Camacho. 2022. "FacTeR-Check: Semi-automated fact-checking through semantic similarity and natural language inference," Knowledge Based Systems, vol. 251, pp. 1-17. Spain: Elsevier Inc.
- [13] Agustian, Surya. 2021. "Pendekatan Semantik Dalam Deteksi Berbagai Tipe Plagiarisme Pada Dokumen Teks," Jurnal Teknik Informatika, vol.14, no.2, pp. 1-14, (file PDF). diunduh pada 13 Juni 2023 <https://journal.uinjkt.ac.id/index.php/ti/article/downloadSuppFile/2841/5618> halaman situs
- [14] Potthast, M, Hagen, M, Gollub, T, Tippmann, M, Kiesel, J, Rosso, P, Stamatatos, E, dan Stein, B, 2013b, Overview of the 5th International Competition on Plagiarism Detection. In Forner, P, Navigli, R, Tufis, D, dan Ferro, N, editors, Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013, volume 1179 of CEUR Workshop Proceedings. CEUR-WS.org.
- [15] Potthast, M, Gollub, T, Hagen, M, Tippmann, M, Kiesel, J, Rosso, P, Stamatatos, E, dan Stein, B, September 2013a. PAN13 Originality: Text Alignment. URL <https://doi.org/10.5281/zenodo.3715980>.

Analisis Komparasi pada Teknik *Black Box Testing* (Studi Kasus: *Website Lars*)

Salmania Jesamine Putri¹, Divi Galih Prasetyo Putri^{1,*}, Widhy Hayuhardhika Nugraha Putra²

¹Departemen Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;

salmania2001@gmail.com

²Universitas Brawijaya;

widhy@ub.ac.id

*Korespondensi: divi.galih@ugm.ac.id;

Abstract – Currently the use of software has dominated almost all fields of knowledge. It is important for developers to ensure the quality of a software so that it is suitable for use by the general public. The quality of the software, one of which can be determined from the output produced, whether it is in accordance with user needs. Software testing is an important activity in the Software Development Life Cycle (SDLC) phase to ensure quality software. There are various methods that may be used in software testing, one of which is black box testing which tests the functionality of a system and does not require the tester to understand program code. In order to achieve optimal testing results, it is necessary to determine the most suitable design of test cases for a software. Equivalence Class Partitioning (ECP), Boundary Value Analysis (BVA), and Decision Table (DT) are commonly used black box testing techniques. This study aims to compare these three technique to determine which technique is most effectively applied to a software. The sample used for testing is the Lars website, an application developed to assist in the hospital accreditation process. The results of testing each technique are measured using standard testing metrics to see which technique is the most optimal. The results obtained by this study are that the ECP outperforms the other technique in catching failures, measured from the calculation of the failed test case metric with a percentage of 51.8% compared to the BVA technique with a result of 33.3% and 46% for DT.

Keywords – Black Box Testing, Equivalence Class Partitioning, Boundary Value Analysis, Decision Table, Website Lars.

Intisari – Saat ini penggunaan perangkat lunak sudah mendominasi hampir seluruh bidang ilmu pengetahuan. Merupakan hal yang penting bagi pengembang untuk memastikan kualitas suatu perangkat lunak sehingga layak digunakan oleh khalayak umum. Kualitas perangkat lunak salah satunya dapat ditentukan dari output yang dihasilkan apakah sudah sesuai dengan kebutuhan pengguna. Pengujian perangkat lunak merupakan salah satu aktivitas yang penting dalam fase Software Development Life Cycle (SDLC) untuk memastikan perangkat lunak yang berkualitas. Terdapat berbagai metode yang mungkin digunakan dalam pengujian perangkat lunak, satu diantaranya adalah black box testing yang menguji fungsionalitas suatu sistem dan tidak mengharuskan penguji untuk memahami kode program. Dalam rangka mencapai hasil pengujian yang optimal, perlu menentukan perancangan kasus uji yang paling tepat digunakan pada suatu perangkat lunak. Equivalence Class Partitioning (ECP), Boundary Value Analysis (BVA), dan Decision Table (DT) merupakan teknik pengujian pada black box yang umum digunakan. Penelitian ini bertujuan untuk membandingkan tiga teknik tersebut, sehingga dapat menentukan teknik mana yang paling efektif diterapkan pada suatu perangkat lunak. Sampel yang digunakan untuk pengujian adalah website Lars yang merupakan aplikasi untuk membantu proses akreditasi rumah sakit. Hasil dari pengujian masing-masing teknik diukur menggunakan standard testing metrics untuk melihat teknik mana yang paling optimal. Hasil yang didapatkan penelitian ini adalah teknik ECP lebih unggul dalam menangkap kegagalan, diukur dari perhitungan metrik test case failed dengan persentase 51.8% dibandingkan teknik BVA dengan hasil 33.3% dan DT 46%.

Kata kunci – Black Box Testing, Equivalence Class Partitioning, Boundary Value Analysis, Decision Table, Website Lars.

I. PENDAHULUAN

Pengujian perangkat lunak (*software testing*) adalah proses menganalisis perangkat lunak untuk menemukan perbedaan antara kebutuhan perangkat lunak dengan kondisi yang sebenarnya atau proses evaluasi dari suatu perangkat lunak [1]. Pada [2] mengemukakan peran *testing* dalam proses pengembangan perangkat lunak atau *Software Development Life Cycle* (SDLC) sangat penting dan merupakan proses yang cukup menantang. *Software testing* memiliki berbagai macam pendekatan yang bisa digunakan, diantaranya *white box* dan *black box testing*. *White box testing* berfokus pada struktur internal dan implementasi kode program, sedangkan *black box testing* tidak memperhatikan struktur internal namun fokus pada *input* dan *output* program [3]. [4] *Black box testing* adalah

pengujian pada fungsionalitas perangkat lunak, biasanya saat melakukan pengujian dengan metode *black box* tester akan berinteraksi dengan antarmuka sistem dengan memberikan *input* dan memeriksa *output* tanpa mengetahui bagaimana cara kerja dan di mana input dikerjakan.

Terdapat berbagai teknik pada *black box testing* yang dapat diterapkan, namun menurut [4] sekalipun memilih salah satu teknik pengujian dari *black box testing*, hasil pengujian dapat bervariasi karena adanya perbedaan teknik dalam mengidentifikasi kebutuhan sistem dan menentukan perancangan kasus uji. Untuk mencegah kemungkinan kesalahan dalam perangkat lunak yang tidak terdeteksi dengan baik, penting untuk

menerapkan teknik *black box testing* yang tepat. Dengan melakukan hal ini, hasil dari pengujian dapat dipercaya dan mengurangi risiko terjadinya kerugian yang signifikan, seperti pemborosan waktu, tenaga, dan biaya. Karena kompleksitas dari setiap aplikasi yang akan diuji sudah pasti berbeda, maka diperlukan analisis komparasi dari berbagai teknik pengujian pada *black box*.

PT Andhara Prima Kreatif (APIK) merupakan perusahaan industri kreatif berbasis teknologi yang menyediakan berbagai layanan kebutuhan produk digital dan penyediaan talenta. PT APIK telah mengembangkan berbagai macam aplikasi yang sudah dapat digunakan secara umum, satu diantaranya yaitu aplikasi Lars. Aplikasi Lars dikembangkan untuk memberikan survei akreditasi rumah sakit berdasarkan peraturan perundang-undangan yang berlaku. Dalam pengembangan aplikasi Lars, proses pengujian diperlukan untuk mengukur kualitas dan kelayakan aplikasi. Oleh karena itu, PT APIK memerlukan analisis komparasi untuk menentukan teknik yang paling efektif digunakan pada proses pengujian aplikasi yang dikembangkan. Analisis komparasi teknik *black box testing* dilakukan dengan menghitung beberapa *software testing metrics*.

II. DASAR TEORI

A. Black Box Testing

Black box testing atau biasa disebut *behavioural testing* adalah metode pengujian perangkat lunak berdasarkan perilakunya, bukan menguji struktur atau kode internal dari suatu program dan berfokus pada *input* dan *output* yang telah dicapai [5]. Menurut [6] *black box testing* merupakan pengujian yang dilakukan berdasarkan spesifikasi kebutuhan dan tidak memerlukan pemeriksaan kode pada aplikasi. Proses ini murni dilakukan berdasarkan sudut pandang pengguna. *Black box testing* berperan penting dalam proses pengujian perangkat lunak untuk membantu validasi fungsionalitas dari sistem. Kelebihan dari metode ini adalah pengujian tidak perlu memiliki pengetahuan khusus mengenai bahasa pemrograman dan juga pengetahuan tentang implementasi.

B. Equivalence Class Partitioning

Equivalence Class Partitioning (ECP) adalah teknik pengujian perangkat lunak yang membagi input nilai menjadi nilai valid dan tidak valid dan memilih representasi dari setiap data uji [7]. Menurut [8] tujuan dari *equivalence partitioning* adalah untuk membagi domain input sistem pengujian menjadi kelas-kelas atau kelompok. Semua input di kelas yang sama memiliki pengaruh serupa pada sistem yang diuji dan mewakili kondisi tertentu. Tabel 1 menggambarkan aturan nilai pengujian pada teknik ECP.

Tabel 1. Aturan Input ECP

Rule	Keterangan
<i>Min</i>	Kondisi ketika nilai input berada di bawah/kurang dari <i>rule</i> yang ada

Rule	Keterangan
<i>Middle</i>	Kondisi ketika nilai input sesuai dengan <i>rule</i> yang ada
<i>Max</i>	Kondisi ketika nilai input berada di atas/lebih dari <i>rule</i> yang ada

C. Boundary Value Analysis

Boundary Value Analysis (BVA) adalah teknik pada *black box testing* yang menguji batas atas dan batas bawah nilai yang dimasukkan ke dalam sistem [9]. Pada intinya, teknik ini memilih data uji yang dekat dengan batas domain data untuk menghasilkan *input* uji di dekat batas dan menemukan *failure* yang disebabkan kesalahan implementasi pada nilai batas. Dalam proses pengembangan program, *developer* cenderung mengabaikan kondisi batas, sehingga *defect* cenderung terkonsentrasi di dekat batas. Oleh karena itu, tes data yang dipilih berada dalam batas atau dekat dengan batas. Dalam pengertian itu, teknik BVA adalah perluasan dan penyempurnaan dari teknik *Equivalence Class*. Tabel 2 menggambarkan aturan nilai pengujian pada teknik BVA.

Tabel 2. Aturan Input BVA

Rule	Keterangan
<i>Min(-)</i>	Tepat di bawah batas bawah
<i>Min(+)</i>	Tepat di atas batas bawah
<i>Middle</i>	Berada pada <i>range</i> pada aturan pengujian
<i>Max(-)</i>	Tepat di bawah batas atas
<i>Max(+)</i>	Tepat di atas batas atas

D. Decision Table

Decision Table (DT) merupakan teknik pada *black box testing* yang digunakan untuk menguji perangkat lunak pada input yang berbeda kombinasi dengan menggabungkan input dan output dan merangkumnya ke dalam tabel. *Decision Table* juga sering disebut sebagai tabel *cause-and-effect* karena tabel pada *Decision Table* berisikan beberapa sebab dan akibat [10]. Pada teknik DT, digunakan sejumlah kombinasi input untuk dapat menentukan output yang harus muncul sesuai dengan kebutuhan sistem [11]. Tabel 3 menggambarkan aturan nilai pengujian pada teknik ECP.

Tabel 3. Aturan Input DT

Kondisi	R1	R2	R3	R4
<i>Username (T/F)</i>	F	T	F	T
<i>Password (T/F)</i>	F	F	T	T
<i>Output (E/S)</i>	E	E	E	S

Keterangan:

T = True

F = False

E = Error

S = Success

R = Rule Pengujian
 $R1$ = Rule Pengujian 1
 $R2$ = Rule Pengujian 2, dst. Total number of combinations = 2^n

E. Selenium IDE

Selenium adalah *software testing framework* yang digunakan untuk mengotomatisasi aplikasi *web* yang banyak digunakan untuk pengujian fungsional, pengujian regresi, dan pengujian performa. Selenium IDE merupakan salah satu *tools* Selenium berbasis *web* yang pada awalnya diimplementasikan sebagai *add-on* pada *Firefox*, namun saat ini sudah dapat digunakan di setiap *browser web*. Beberapa kelebihan Selenium IDE, yaitu memberikan dukungan *multi-browser*, tidak memerlukan pemahaman mendalam terhadap bahasa pemrograman, dapat mengatur *breakpoint* dan *debug*, dapat melakukan *record-and-playback*, serta penggunaannya yang mudah [12].

F. Software Testing Metrics

Software Testing Metrics berperan penting dalam proses pengujian dan evaluasi produk perangkat lunak dengan menyediakan standar dan pengukuran yang objektif. Dengan menggunakan *software testing metrics*, pengembang juga dapat mengukur dan memantau sejauh mana perkembangan perangkat lunak [13] [14] [15]. Dalam penelitian ini *metrics* yang digunakan dapat dilihat di persamaan (1), (2), (3) dan (4).

$$Test\ Case\ Executed\ \% = \left(\frac{Test\ Case\ Executed}{Test\ Case\ Written} \right) \times 100\% \quad (1)$$

Keterangan:

Test Case Executed = total *test case* yang dapat dieksekusi

Test Case Written = total *test case* yang dibuat

$$Test\ Case\ Passed\ \% = \left(\frac{Test\ Case\ Passed}{Total\ Test\ Case} \right) \times 100\% \quad (2)$$

Keterangan:

Test Case Passed = total *test case* yang berhasil

Total Test Case = total *test case* yang diuji

$$Test\ Case\ Failed\ \% = \left(\frac{Test\ Case\ Failed}{Total\ Test\ Case} \right) \times 100\% \quad (3)$$

Keterangan:

Test Case Failed = total *test case* yang gagal

Total Test Case = total *test case* yang diuji

$$Total\ Test\ Case = \text{jumlah test case yang dibuat} \quad (4)$$

Keterangan:

Total *test case* dapat memperkirakan lamanya waktu pengujian.

Semakin banyak *test case*, maka waktu yang dibutuhkan untuk pengujian semakin lama.

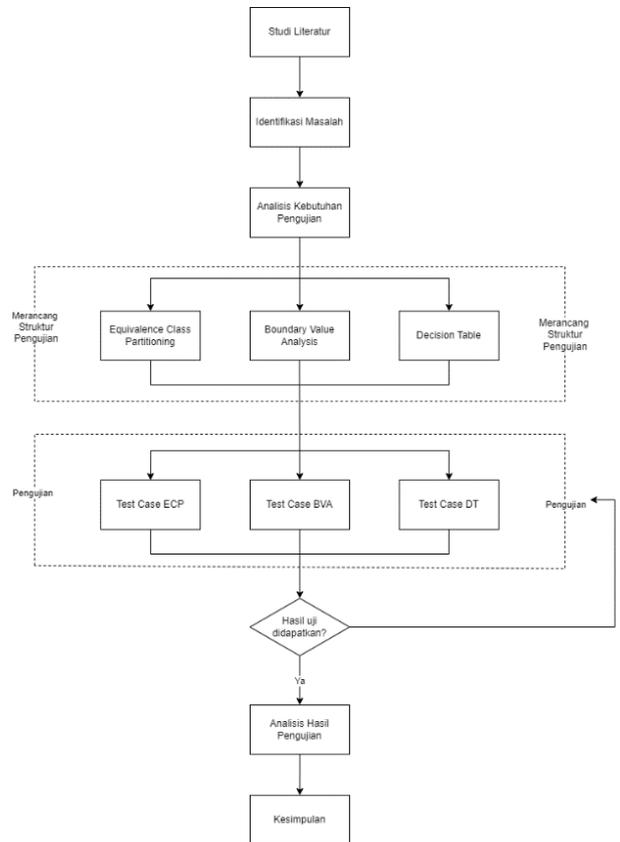
III. METODOLOGI

A. Objek Penelitian

Objek pada penelitian ini adalah *website* Lars, yaitu *software* yang dikembangkan oleh PT Andhara Prima Kreatif (PT APIK) yang digunakan untuk membantu proses akreditasi rumah sakit.

B. Tahapan Penelitian

Tahapan penelitian diilustrasikan pada Gambar 1.



Gambar 1. Alur Penelitian

C. Rancangan Pengujian

Pengujian *website* Lars dilakukan secara otomatis dengan menggunakan bantuan *tools* Selenium IDE. Untuk melakukan pengujian, diperlukan *test case* dan data yang akan diinputkan pada pengujian.

Contoh pada Tabel 4 dan 5 adalah data input jika suatu elemen fitur memiliki rentang nilai valid dari angka 1 sampai 9999. Tabel 4 menunjukkan data input yang akan digunakan dalam metode ECP.

Tabel 4. Contoh Data Input ECP

Kondisi	Contoh Data
Invalid	-1
Valid	1000
Invalid	100000

Keterangan:

Nilai invalid = di bawah rentang nilai terendah

Nilai valid = di antara rentang nilai yang sudah ditentukan

Nilai invalid = di atas rentang nilai tertinggi

Tabel 5 menunjukkan data input yang akan digunakan dalam metode BVA.

Tabel 5. Contoh Data Input BVA

Kondisi	Contoh Data
Batas bawah	0
Batas atas	2
Nilai tengah	2000
Batas bawah 2	9998
Batas atas 2	10000

Keterangan:

Nilai batas bawah= nilai tepat di bawah rentang terendah

Nilai batas atas = nilai tepat di atas rentang terendah

Nilai tengah = nilai di antara rentang yang sudah ditentukan

Nilai batas bawah 2= nilai tepat di bawah rentang tertinggi

Nilai batas atas 2 = nilai tepat di atas rentang tertinggi

Tabel 6 menunjukkan data input yang akan digunakan dalam metode DT.

Tabel 6. Contoh Data Input DT

Kondisi	Contoh Data
True	55167
	77777
False	11234567890
	1234!@
	F1235
	FFFFF
	1

Tabel 6 adalah contoh data untuk elemen fitur kode pos yang terdiri dari 5 angka.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian ECP

Tabel 7 menunjukkan hasil pengujian metode ECP.

Tabel 7. Hasil Pengujian ECP

Elemen Fitur	Pass	Fail	Total Test Case
Nama Rumah Sakit	2	1	3
Nomor Telepon	2	1	3
Email	3	2	5
Password	2	1	3
Konfirmasi Password	2	1	3
Email (login)	2	1	3
Password (login)	2	1	3
Email (forget password)	2	1	3
Induk RS	1	2	3

Elemen Fitur	Pass	Fail	Total Test Case
Kapasitas	1	2	3
NIB	1	2	3
Izin Operasional	1	2	3
Alamat Lengkap	1	2	3
Kode Pos	1	2	3
Nama Direktur	1	2	3
Email Direktur	1	2	3
Nomor Direktur	1	2	3
Divisi	1	2	3
Total	27	29	56

B. Hasil Pengujian BVA

Tabel 8 menunjukkan hasil pengujian metode BVA.

Tabel 8. Hasil Pengujian BVA

Elemen Fitur	Pass	Fail	Total Test Case
Nama Rumah Sakit	4	1	5
Nomor Telepon	4	1	5
Email	5	2	7
Password	4	1	5
Konfirmasi Password	4	1	5
Email (login)	5	1	6
Password (login)	4	1	5
Email (forget password)	4	1	5
Induk RS	3	2	5
Kapasitas	3	2	5
NIB	3	2	5
Izin Operasional	3	2	5
Alamat Lengkap	3	2	5
Kode Pos	1	4	5
Nama Direktur	3	2	5
Email Direktur	3	2	5
Nomor Direktur	3	2	5
Divisi	3	2	5
Total	62	31	93

C. Hasil Pengujian DT

Tabel 9 menunjukkan hasil pengujian metode DT.

Tabel 9. Hasil Pengujian DT

Teknik	Pass	Fail	Total Test Case
ECP	27	29	56
BVA	62	31	93
DT	39	45	75

D. Perbandingan Hasil Pengujian

Tabel 10 menunjukkan perbandingan hasil pengujian.

Tabel 10. Perbandingan Hasil

Fitur	Pass	Fail	Total Test Case
Registrasi	31	1	32
Login	4	0	4
Forget Password	1	1	2
Profil	3	33	36

E. Perhitungan *Standard Testing Metrics*

1. *Test Case Executed*

$$\text{Test Case Executed ECP \%} = \left(\frac{56}{56}\right) \times 100\% = 100\%$$

$$\text{Test Case Executed BVA \%} = \left(\frac{93}{93}\right) \times 100\% = 100\%$$

$$\text{Test Case Executed DT \%} = \left(\frac{74}{74}\right) \times 100\% = 100\%$$

2. *Test Case Passed*

$$\text{Test Case Passed ECP \%} = \left(\frac{27}{56}\right) \times 100\% = 48.2\%$$

$$\text{Test Case Passed BVA \%} = \left(\frac{62}{93}\right) \times 100\% = 66.7\%$$

$$\text{Test Case Passed DT \%} = \left(\frac{40}{74}\right) \times 100\% = 54\%$$

3. *Test Case Failed*

$$\text{Test Case Failed ECP \%} = \left(\frac{29}{56}\right) \times 100\% = 51.8\%$$

$$\text{Test Case Failed BVA \%} = \left(\frac{62}{93}\right) \times 100\% = 33.3\%$$

$$\text{Test Case Failed DT \%} = \left(\frac{34}{74}\right) \times 100\% = 46\%$$

4. *Total Test Case*

Tabel 11 menunjukkan hasil perhitungan *total test case*.

Tabel 11. *Total Test Case*

Teknik	Total Test Case
ECP	56
BVA	93
DT	74

F. Perbandingan Metriks

Tabel 12 menunjukkan hasil perbandingan metriks.

Tabel 12. Perbandingan Metriks

Metrics	ECP	BVA	DT
<i>Executed</i>	100%	100%	100%
<i>Pass</i>	48.2%	66.7%	54%
<i>Fail</i>	51.8%	33.3%	46%
<i>Total Test Case</i>	Low	High	Medium

Dari hasil yang ditunjukkan pada Tabel 12, teknik ECP mampu menemukan banyak kegagalan, dibuktikan dengan persentase *test case fail* yang tinggi jika dibandingkan dua teknik lain. Hasil ini sesuai dengan hasil penelitian terdahulu yang dilakukan oleh [16] yang menunjukkan teknik ECP mampu menangkap paling banyak kegagalan dengan waktu yang paling singkat dibandingkan teknik BVA, PWT, dan DT.

Berdasarkan pengujian yang telah dilakukan pada *website* Lars, teknik ECP dapat menangkap lebih banyak kegagalan karena memiliki perbandingan tes negatif yang lebih tinggi dibandingkan BVA, sedangkan DT lebih memfokuskan pada logika kombinasi setiap fitur. Pada ECP, jumlah minimal tes untuk setiap fitur adalah tiga dengan perbandingan negatif tes dibanding positif tes adalah 2:1, sedangkan minimal tes pada BVA adalah lima dengan perbandingan tes negatif dibanding tes positif adalah 2:3, sedangkan DT mencoba segala kemungkinan yang dapat terjadi.

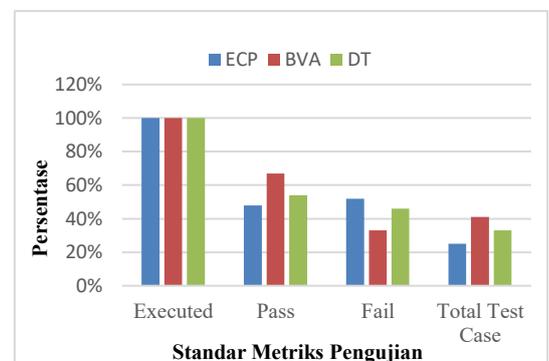
Teknik ECP menjadi teknik yang paling efektif untuk diterapkan pada pengujian *website* Lars karena ECP mampu menguji seluruh *test case* yang telah dibuat dan berhasil menangkap lebih banyak kegagalan dengan jumlah kasus pengujian yang paling sedikit. Selain itu, jumlah kasus uji pada teknik ECP paling sedikit dan jenis input data tidak harus spesifik seperti BVA.

Teknik BVA mampu menemukan kesalahan yang lebih spesifik dibandingkan ECP dan DT. Namun berdasarkan perhitungan metriks pada penelitian ini, teknik BVA menjadi teknik yang paling tidak disarankan untuk pengujian *website* Lars karena membutuhkan banyak kasus uji dan tidak mendeteksi banyak kegagalan, serta membutuhkan waktu pengujian yang lebih lama. Data yang digunakan pada teknik BVA juga harus spesifik pada batas-batas nilai yang relevan dengan perubahan, sehingga memerlukan pemahaman dan ketelitian dalam menentukan nilai input.

Teknik DT memiliki jenis perancangan kasus uji yang berbeda dengan ECP dan BVA karena tidak berfokus pada satu elemen dan pengujian karakter, tetapi menguji berbagai kombinasi input dan kondisi yang berbeda. Pengujian ini membantu memahami bagaimana perubahan dalam satu kondisi dapat mempengaruhi hasil pengujian.

G. Grafik

Gambar 2 menunjukkan grafik hasil perhitungan metriks *software testing*.



Gambar 2. Grafik Hasil Perhitungan Metriks

V. SIMPULAN

Hasil pengujian dengan teknik ECP menunjukkan *test case failed* sebesar 51.8%, pada teknik BVA sebesar 33.3%, dan teknik DT 46%. Berdasarkan hasil tersebut, *website* Lars perlu melakukan perbaikan fungsionalitas karena masih banyak ditemukan *bug*.

Teknik yang paling efektif untuk digunakan pada *website* Lars adalah ECP yang memiliki tingkat keberhasilan dalam menemukan *bug* sebesar 51.8% dan jumlah kasus pengujian yang paling sedikit dibandingkan dengan BVA dan DT.

Bagi peneliti selanjutnya, dapat melakukan penelitian dengan mengimplementasikan lebih banyak teknik. Selain itu, dapat melakukan penelitian lebih lanjut mengenai penggabungan teknik ECP dan DT yang dapat diimplementasikan pada aplikasi lain.

VI. REFERENSI

- [1] M. E. Khan, "Different Approaches to Black Box Testi Technique for Finding Errors," vol. 2, p. 10, 2011.
- [2] R. V. Binder, *Testing Object-Oriented Systems: Mode Patterns, and Tools*, Addison-Wesley Professional, 1999.
- [3] Z. A. Hamza dan M. Hammad, "Web and Mobile Applicatio Testing using Black and White Box approaches," *IEEE*, p. 2019.
- [4] K. I. Seo dan E. M. Choi, "Comparison of Five Black-box Testi Methods for Object-Oriented Software," *IEEE*, p. 8, 2006.
- [5] U. K. Tiwari dan S. Kumar, "Components Integration-Effk Graph: A Black Box Testing and Test Case Generation Techniq for Component-Based Software," *Springer Link*, p. 15, 2017.
- [6] S. Nidhra dan J. Dondeti, "Black Box and White Box Testi Techniques - A Literature Review," *International journal embedded systems and applications*, vol. 2, no. 2, pp. 29-5 2012.
- [7] M. Sholeh, I. Gifas, Cahiman dan M. A. Fauzi, "Black B Testing on ukmbantul.com Page with Boundary Value Analy: and Equivalence Partitioning Methods," *IOP Publishing*, p. 2021.
- [8] K. Naik dan P. Tripathy, *Software Testing and Quality Assuran Theory and Practice*, A JOHN WILEY & SONS, INC PUBLICATION, 2008.
- [9] I. P. A. Prayudha, R. S. Hartati dan Y. Divayana, "Bounda Value Analysis Testing Techniques on Learning Managemen System Applications," *International Journal of Engineering a Emerging Technology*, vol. 4, p. 4, 2019.
- [10] J. Joosten, A. E. Permanasari dan T. B. Adji, "The Use Decision Table for Reducing Complex Rules in Softwa Testing," *IOP Science*, p. 8, 2020.
- [11] Meenu dan Navita, "Study and Analysis of Software Testing *IJRITCC*, vol. 3, no. 12, p. 6674-6678, 2015.
- [12] 2023. [Online]. Available: <https://www.selenium.dev>. [Diaks 15 April 2023].
- [13] P. M. Jacob dan D. M. Prasanna, "A Comparative Analysis Black Box Testing Strategies," *IEEE*, no. 16, p. 6, 2016.
- [14] I. G. S. Aryandana, *Pengukuran Performa Metode Softwa Testing Equivalence Class Partitioning dan Boundary Val Analysis*, Yogyakarta, 2019.
- [15] Hamilton. [Online]. Available: <ps://www.guru99.com/software-testing-metrics-complete-orial.html>. [Diakses June 2023].
- [16] Xu, L. Chen, C. Wang dan O. Rud, "A Comparative Study Black-Box Testing with Open Source Applications," *IEEE*, 6, 2016.

Komparasi Metode Automasi dan *Hybrid* pada Pengujian Aplikasi *Mobile Webrtc* Menggunakan Appium

Gabriela Anggerita Jasmin¹, Divi Galih Prasetyo Putri^{1,*}

¹Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada
anggeritaj@mail.ugm.ac.id

*Korespondensi: divi.galih@ugm.ac.id;

Abstract – Information technology has transformed communication and interaction in society. It allows people to connect virtually, regardless of distance. Virtual consultations are common, facilitated by applications with video call features using WebRTC technology. WebRTC enables real-time audio, visual, and data communication. Testing is crucial to minimize bugs, but WebRTC's complex workflow makes it challenging. Software testing can be manual or automated. Manual testing involves using two devices to test WebRTC directly. Appium, an open-source tool, is used for automated testing. Appium supports multiple platforms and programming languages, without needing access to source code for device functionalities like the camera and microphone. It's not tied to any specific testing framework. The testing process covers the entire application system, ensuring optimal functionality and addressing all issues. The testing method choice depends on requirements. Testing results favor the automated method in terms of time efficiency, requiring less time. However, the hybrid method surpasses it in test coverage and effectiveness, covering more test cases.

Keywords – End-to-End, Appium, Testing, WebRTC, Automasi, Hybrid

Intisari – Teknologi informasi telah mengubah komunikasi dan interaksi dalam masyarakat. Ini memungkinkan orang untuk terhubung secara virtual, tanpa memandang jarak. Konsultasi virtual menjadi umum, difasilitasi oleh aplikasi dengan fitur panggilan video menggunakan teknologi WebRTC. WebRTC memungkinkan komunikasi audio, visual, dan data secara real-time. Pengujian sangat penting untuk meminimalkan bug, tetapi alur kerja yang kompleks dari WebRTC membuatnya menantang. Pengujian perangkat lunak dapat dilakukan secara manual atau otomatis. Pengujian manual melibatkan penggunaan dua perangkat untuk menguji WebRTC secara langsung. Appium, sebuah alat sumber terbuka, digunakan untuk pengujian otomatis. Appium mendukung berbagai platform dan bahasa pemrograman, tanpa perlu akses ke kode sumber untuk fungsi perangkat seperti kamera dan mikrofon. Appium tidak terikat pada kerangka pengujian tertentu. Proses pengujian meliputi seluruh sistem aplikasi, memastikan fungsi optimal dan menangani semua masalah. Pemilihan metode pengujian tergantung pada kebutuhan. Hasil pengujian lebih mendukung metode otomatis dalam hal efisiensi waktu, membutuhkan waktu yang lebih sedikit. Namun, metode hibrida melampaui metode tersebut dalam cakupan pengujian dan efektivitas, mencakup lebih banyak kasus pengujian.

Kata kunci – End-to-End, Appium, Pengujian, WebRTC, Automasi, Hybrid

I. PENDAHULUAN

Pengaruh teknologi informasi pada kehidupan manusia sangat besar. Dalam berkomunikasi, teknologi informasi memungkinkan orang untuk terhubung secara virtual tanpa memperdulikan jarak. Dengan adanya konektivitas yang luas, pertukaran informasi menjadi merata dan dapat diakses tanpa terbatas oleh waktu dan lokasi [1]. Namun, teknologi ini juga memberikan dampak negatif seperti penyebaran berita palsu dan tindak kejahatan siber. Di sisi lain, teknologi informasi membawa banyak manfaat, seperti kemudahan dalam mendapatkan informasi dan berkomunikasi. Konsultasi virtual menjadi lebih mudah dengan aplikasi berfitur panggilan video menggunakan teknologi WebRTC. WebRTC juga dapat beroperasi di berbagai peramban web, lintas platform atau sistem operasi, baik pada komputer maupun perangkat *mobile*. Kehadiran teknologi ini juga memberikan banyak keuntungan bagi pengguna [2]. Namun, menerapkan WebRTC dalam aplikasi membutuhkan pengujian yang rumit.

Pengujian perangkat lunak bisa dilakukan secara manual atau otomatis. Dalam pengujian manual, dua perangkat digunakan untuk menguji WebRTC secara langsung. Pengujian otomatis menggunakan alat sumber terbuka

bernama Appium, yang mendukung berbagai platform dan bahasa pemrograman. Proses pengujian meliputi seluruh sistem aplikasi, dengan tujuan memastikan fungsionalitas optimal dan menyelesaikan semua masalah. Pilihan metode pengujian tergantung pada kebutuhan.

Oleh karena itu, pada penelitian ini dilakukan eksplorasi metode pengujian pada aplikasi yang mengimplementasikan *WebRTC*. Setiap metode yang diimplementasikan akan dievaluasi menggunakan metrik test coverage, time, dan test case effectiveness.

Pengujian hanya mengacu pada satu alur bisnis tertentu yang menuju penggunaan WebRTC. Pengujian tidak mencakup pengaruh penggunaan koneksi internet. Aplikasi yang diuji merupakan aplikasi berbasis *mobile* dengan sistem operasi Android.

Hasil pengujian diharapkan dapat memberikan informasi kepada pengembang tentang bug yang ditemukan dan menjadi panduan dalam melakukan perbaikan bug pada pengembangan selanjutnya.

II. DASAR TEORI

A. Pengujian Perangkat Lunak

1) Pengujian Antarmuka Pengguna Perangkat Lunak

UI merupakan komponen krusial dalam aplikasi Android karena berfungsi sebagai antarmuka yang langsung dilihat oleh pengguna. Kualitas UI yang baik memudahkan pengguna untuk memahami aktivitas yang terjadi dalam aplikasi. Pengujian UI melibatkan berbagai tahap dan komponen seperti tombol, input teks, dan menu dropdown. Hal ini penting karena fungsionalitas UI berdampak langsung pada kualitas perangkat lunak [3].

Pengujian merupakan tahapan penting dalam pengembangan aplikasi Android yang bertujuan untuk memastikan kualitas perangkat lunak. Hasil pengujian mempengaruhi pengembangan selanjutnya. Strategi pengujian yang tepat dapat membantu dalam mendeteksi lebih banyak bug dan menghemat biaya, sementara penerapan strategi yang kurang tepat dapat mengakibatkan bug terlewat dan biaya yang meningkat [4].

2) Pengujian End-to-end

Pengujian end-to-end adalah pengujian sistem perangkat lunak secara menyeluruh, yang melibatkan semua komponen dan modul untuk memastikan integritasnya. Tujuannya adalah untuk menemukan masalah dan memverifikasi bahwa sistem berfungsi dengan baik dalam skenario penggunaan yang sebenarnya. Proses pengujian ini meliputi perencanaan, pembuatan skenario pengujian, pelaksanaan tes, dan analisis hasilnya. Keuntungannya termasuk pengujian yang komprehensif terhadap alur kerja sistem dan mendeteksi masalah sebelum sistem diimplementasikan. Namun, pengujian ini bisa kompleks dan memerlukan sumber daya yang signifikan.

3) Pengujian Black Box

Pengujian perangkat lunak bertujuan untuk memvalidasi dan memverifikasi bahwa perangkat lunak telah dibangun sesuai dengan kebutuhan yang telah ditetapkan. Dalam proses pengujian, dilakukan pemeriksaan untuk menemukan bug atau kesalahan sedini mungkin, sehingga dapat segera diperbaiki oleh pengembang. Pengujian perangkat lunak penting untuk menjamin kualitas perangkat lunak yang dikembangkan.

Ada dua jenis pengujian yang umum digunakan, yaitu white box testing dan black box testing. Black box testing berfokus pada fungsionalitas perangkat lunak, dengan mendeteksi kesalahan yang berkaitan dengan fungsi, antarmuka, struktur data, performa, inisialisasi, dan terminasi [5].

4) Pengujian Automasi Perangkat Lunak

Pengujian perangkat lunak dapat dilakukan secara manual atau automasi. Pengujian automasi menggunakan tools atau framework untuk mengotomatisasi proses pengujian. Dalam

pengujian automasi, skrip atau kode program digunakan untuk menjalankan serangkaian tes secara otomatis, termasuk simulasi interaksi pengguna dengan perangkat lunak. Keuntungan pengujian automasi adalah cakupan pengujian yang luas, kemampuan untuk melakukan pengujian berulang-ulang secara otomatis, dan penghematan waktu. Selain itu, pengujian automasi juga dapat mendeteksi banyak bug atau error, dan dokumentasi pengujian dilakukan dengan baik [3].

B. Alat Pengujian Automasi

1) Appium

Appium adalah tools pengujian automasi open source yang dapat digunakan untuk menguji aplikasi *mobile* native, aplikasi web seluler, dan aplikasi hybrid. Appium mendukung pengujian lintas platform, sehingga skrip yang sama dapat digunakan untuk menguji aplikasi di berbagai sistem operasi *mobile*. Appium menggunakan protokol WebDriver untuk mengendalikan aplikasi *mobile* dan melakukan berbagai aksi seperti mengklik dan memasukkan teks. Appium kompatibel dengan berbagai bahasa pemrograman dan menggunakan alat pengujian UIAutomator untuk Android dan XCUITest untuk iOS. Dengan Appium, tester dapat memeriksa dan mengotomatisasi setiap elemen dalam aplikasi *mobile* sesuai dengan skenario pengujian yang telah disusun [6].

2) UiAutomator2

UiAutomator2 Appium adalah framework pengujian otomatis untuk perangkat Android. Framework ini memungkinkan pengujian aplikasi native, hybrid, dan web *mobile* secara otomatis pada emulator dan perangkat fisik. UiAutomator2 Appium berfungsi berdasarkan protokol W3C WebDriver dengan ekstensi khusus untuk menangani skenario-skenario khusus sistem operasi. Framework ini menggunakan server UiAutomator2 yang mengoperasikan perintah-perintahnya dan memanfaatkan framework UiAutomator yang dikembangkan oleh Google. UiAutomator2 Appium terintegrasi dengan baik dalam versi-versi terbaru Android dan menyediakan alat pengujian otomatis untuk berinteraksi dengan perangkat, termasuk alat seperti UiSelector untuk mengambil elemen-elemen dari aktivitas utama emulator [7].

3) JavaScript

JavaScript adalah bahasa pemrograman ringan dan lintas platform yang digunakan untuk pengembangan halaman web dan aplikasi. Selain di browser, JavaScript juga bisa digunakan di sisi server menggunakan Node.js.

Dalam pengujian perangkat lunak, JavaScript sangat penting. Untuk pengujian aplikasi *mobile* secara otomatis, JavaScript bisa digunakan dengan framework seperti Appium atau WebDriverIO. Dengan JavaScript, pengujian aplikasi *mobile* bisa mencakup tindakan seperti mengklik, memasukkan teks, dan memverifikasi elemen aplikasi. Kombinasi JavaScript dan Appium memberikan kekuatan dalam otomatisasi pengujian aplikasi *mobile*.

Dalam pengujian aplikasi *mobile* dengan JavaScript, penting untuk memilih kerangka kerja yang tepat, memahami struktur aplikasi dan platform yang dituju, serta menerapkan prinsip pengujian yang baik. Terdapat juga perpustakaan JavaScript khusus seperti *selendroid*, *Detox*, atau *WebdriverIO* yang dapat meningkatkan efisiensi dan kemudahan dalam pengujian. Dengan JavaScript, pengujian aplikasi *mobile* dapat dilakukan di berbagai platform seperti *iOS* dan *Android*, dengan melakukan berbagai aksi dan verifikasi pada elemen aplikasi.

4) Node.js

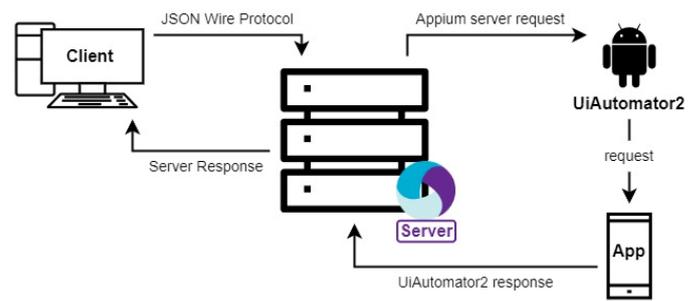
Node.js adalah lingkungan runtime JavaScript yang menggunakan mesin JavaScript V8 dari Google Chrome. Dengan Node.js, pengembang dapat menjalankan kode JavaScript di sisi server untuk membangun aplikasi web yang efisien dan skalabel. Node.js menggunakan model non-blokir I/O yang memungkinkan pengolahan paralel yang responsif terhadap permintaan. Hal ini membuatnya cocok untuk mengembangkan aplikasi jaringan yang membutuhkan penanganan banyak permintaan secara bersamaan. Node.js juga dilengkapi dengan NPM (Node Package Manager) yang menyediakan ribuan paket modul siap pakai untuk memperluas fungsionalitas Node.js.

C. WebSocket

WebSocket adalah protokol komunikasi real-time yang memfasilitasi komunikasi dua arah antara klien dan server. Protokol ini memiliki keunggulan dalam komunikasi langsung, mengurangi beban protokol, dan mendukung komunikasi lintas domain. WebSocket sering digunakan dalam aplikasi web seperti chat, pembaruan data real-time, dan kolaborasi online. Dalam pengujian aplikasi WebSocket, penting untuk menguji fungsionalitas komunikasi real-time, pembaruan data langsung, penanganan kesalahan, dan integrasi dengan komponen lainnya. Pengujian ini melibatkan simulasi pesan, pemantauan respons server, dan pengujian beban untuk memastikan performa yang optimal ketika terdapat banyak koneksi WebSocket.

D. Arsitektur Pengujian

Pada pengujian dengan Appium, skrip pengujian ditulis menggunakan bahasa pemrograman yang didukung oleh klien Appium. Skrip ini berisi perintah dan asersi untuk mengontrol aplikasi, mengambil data, dan memverifikasi hasilnya. Selama pengujian, server Appium menerima perintah dari skrip dan mengirimkannya ke driver UiAutomator2. Driver ini menjalankan perintah pada perangkat Android yang sedang diuji. Respons dari perangkat Android dikirim kembali ke server Appium, yang kemudian meneruskannya ke klien Appium. Dengan arsitektur ini, Appium memungkinkan pengujian otomatis aplikasi Android dengan pengendalian yang lengkap, serta kemampuan untuk menguji fungsionalitas dan memverifikasi perilaku aplikasi di perangkat Android. Arsitektur pengujian diilustrasikan pada Gambar 1.



Gambar 1. Arsitektur Pengujian

E. Test Case

Test case (kasus pengujian) adalah spesifikasi detail yang memberikan petunjuk tentang langkah-langkah, data, dan hasil yang diharapkan dalam pengujian. Test case digunakan untuk menguji fungsionalitas perangkat lunak dengan mengikuti prosedur yang telah ditentukan sebelumnya. Test case yang baik dapat menemukan kesalahan dan cacat yang belum terdeteksi sebelumnya, bukan hanya untuk memverifikasi bahwa program berfungsi dengan benar. Hasil pengujian dengan test case akan dibandingkan dengan hasil yang diharapkan, dan jika ada perbedaan, dilakukan perbaikan pada kode program [9].

F. WebRTC

WebRTC (Web Real-Time Communication) adalah framework open source yang memungkinkan komunikasi real-time seperti panggilan suara, panggilan video, dan pertukaran data langsung dalam aplikasi web dan *mobile*. WebRTC mendukung penggunaan perangkat keras pada perangkat *mobile* seperti kamera dan mikrofon, sehingga memudahkan integrasi fitur panggilan video dan suara langsung. Pengguna dapat melakukan komunikasi peer-to-peer tanpa perlu menginstal plugin atau perangkat tambahan [10]. WebRTC menghubungkan berbagai platform dan perangkat melalui protokol yang disediakan oleh website resmi WebRTC menggunakan JavaScript API.

III. METODOLOGI

A. Spesifikasi Perangkat

Dalam penelitian ini spesifikasi perangkat lunak dan perangkat keras yang digunakan adalah sebagai berikut.

Perangkat Lunak

- 1) Sistem operasi Android 10
- 2) Sistem operasi Windows 11 pro 64-bit
- 3) Visual Studio Code
- 4) Appium
- 5) UiAutomator2

Perangkat Keras

- 1) Laptop Asus Zenbook Flip 14 UX462DA
 - RAM 8 GB
 - Processor AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
 - SSD 256 GB
- 2) Samsung Galaxy M20
 - RAM 3 GB
 - ROM 32 GB
 - Processor Exynos 7904
 - Android 10
 - Kamera belakang 13MP
 - Kamera depan 8MP
- 3) Redmi Note 11 Pro
 - RAM 8 GB
 - ROM 128 GB
 - Processor MediaTek Helio G96
 - Android 13
 - Kamera belakang 108MP
 - Kamera depan 16MP

B. Tahapan Penelitian

Pengerjaan penelitian dimulai dengan menganalisis sistem kerja aplikasi dan memahami alur aplikasi (Gambar 2). Selanjutnya, test case dirancang untuk menguji bagian-bagian penting dalam aplikasi. Test case akan direview untuk memastikan kematangannya. Setelah test case siap, dilakukan persiapan kode untuk pengujian otomatis dengan menggunakan Appium. Pengujian otomatis dilakukan dan diikuti oleh pengujian manual. Setelah semua pengujian selesai, dilakukan analisis untuk membandingkan hasil pengujian dengan metrik yang telah ditetapkan, seperti test coverage, time, dan test case effectiveness. Perhitungan metrik pengujian dapat memberikan penilaian terhadap kecacatan yang muncul [8].

1) Test Coverage

Test coverage adalah metrik yang mengevaluasi sejauh mana pengujian telah dilakukan terhadap kode atau sistem yang diuji. Metrik ini dapat mengindikasikan kelengkapan pengujian dan tujuannya adalah untuk menentukan seberapa banyak kode atau fungsionalitas aplikasi yang telah diuji dan sejauh mana pengujian tersebut dilakukan secara efektif [8]. Ada juga konsep scenario coverage yang mengukur sejauh mana skenario atau kasus pengujian telah dijalankan. Hal ini membantu dalam mengevaluasi cakupan pengujian dan memastikan bahwa area yang memadai telah diuji. Dalam menghitung persentase *test coverage*, dapat dilihat pada persamaan (1).

$$TC = \frac{ETC}{TTC} \times 100\% \quad (1)$$

TC = Test Coverage

ETC = Executed Test Case

TTC = Total Test Case

2) Time

Test time adalah durasi yang diperlukan untuk menjalankan tes dan mendapatkan hasilnya. Metrik ini digunakan untuk mengevaluasi efisiensi dan efektivitas proses pengujian. Durasi test time dapat berbeda-beda tergantung pada kompleksitas test case, ukuran sistem, konfigurasi perangkat keras dan lunak, serta performa lingkungan pengujian. Penting untuk memantau dan mengelola test time secara efektif untuk memastikan penyelesaian tepat waktu aktivitas pengujian dan mengidentifikasi masalah potensial yang dapat mempengaruhi proses pengujian secara keseluruhan. Manajemen test time yang efisien melibatkan optimisasi proses eksekusi tes, memberikan prioritas pada test case kritis, menggunakan alat otomatisasi, dan menerapkan strategi pengujian paralel atau terdistribusi jika memungkinkan. Dengan mengelola test time dengan baik, proses pengujian dapat dilakukan dengan cepat, meningkatkan produktivitas, dan menghasilkan perangkat lunak berkualitas sesuai dengan jadwal yang ditargetkan. Metrik ini dihitung dari total waktu eksekusi pengujian.

3) Test Effectiveness

Mengukur kekurangan dalam pengujian perangkat lunak adalah suatu disiplin yang penting, tidak peduli apakah pengujian telah diotomatisasi atau tidak [8]. Tujuan utama pengujian perangkat lunak adalah mengidentifikasi bug atau cacat dalam perangkat lunak, baik itu bug yang valid maupun yang tidak valid atau tidak relevan. Jika ditemukan bug yang valid, langkah-langkah perbaikan dan pengujian ulang harus dilakukan, sementara bug yang tidak valid dapat ditutup atau diabaikan.

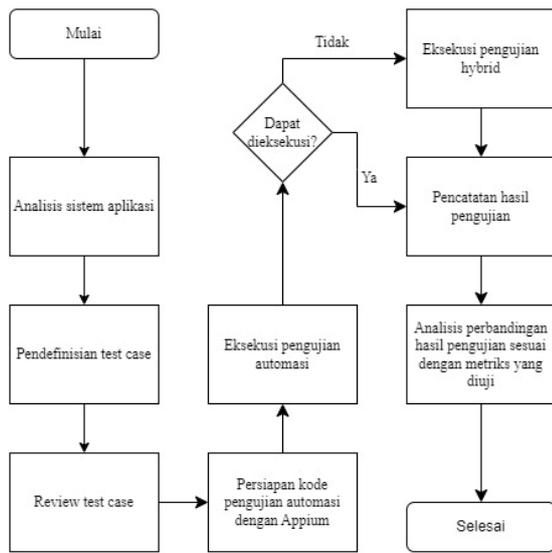
Efektivitas pengujian perangkat lunak dapat diukur dengan persentase jumlah bug yang ditemukan dibandingkan dengan jumlah total test case yang dijalankan. Rumus untuk menghitung metrik efektivitas test case adalah: jumlah bug yang ditemukan dibagi oleh jumlah total *test case* dikalikan dengan 100% yang dapat dilihat pada persamaan (2).

$$TCE = \frac{DD}{TTC} \times 100\% \quad (2)$$

TCE = Test Case Effectiveness

DD = Defects Detected

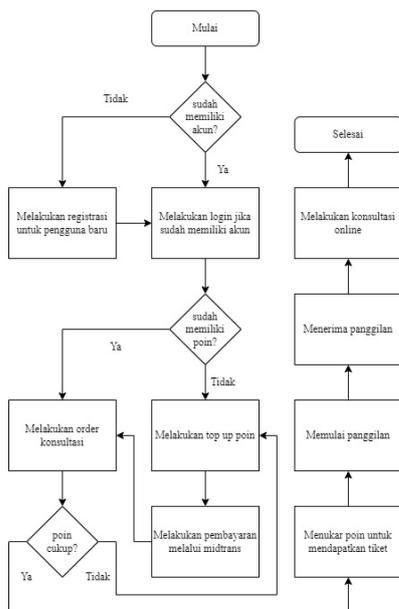
TTC = Total Test Case



Gambar 2. Tahapan Pengujian

C. Analisis Sistem Aplikasi

Alur bisnis aplikasi Ngampu dimulai dengan registrasi pengguna baru. Pengguna kemudian dapat melakukan login untuk mengakses aplikasi. Selanjutnya, pengguna dapat melakukan top up poin dan melakukan pembayaran melalui midtrans. Setelah itu, pengguna dapat memesan konsultasi melalui menu konsultan. Jika poin cukup, pesanan dapat berhasil dibuat. Jika tidak, pengguna dapat melakukan top up terlebih dahulu. Setelah pesanan konsultasi terbuat, panggilan video sesuai jadwal yang dipilih dapat dimulai. Sesi konsultasi dimulai setelah konsultan menerima panggilan dari pengguna. Analisis sistem aplikasi dapat dilihat pada Gambar 3.



Gambar 3. Alur Bisnis Aplikasi Ngampu

D. Test Case

Pengujian dilaksanakan dengan acuan test case yang telah dibuat seperti yang dapat dilihat pada Tabel 1.

Tabel 1. Test Case

ID	Test Case
TC01	Register valid
TC02	Register dengan Mengosongkan Nama
TC03	Register dengan Mengosongkan Email
TC04	Register dengan Mengosongkan Password
TC05	Register dengan Nama Menggunakan Angka
TC06	Register dengan Email Tanpa @
TC07	Register dengan Email Tanpa . (dot)
TC08	Register dengan Email yang Sudah Terdaftar
TC09	Register dengan Password Tanpa Huruf Kapital
TC10	Register dengan Password Tanpa Huruf Kecil
TC11	Register dengan Password Tanpa Angka
TC12	Register dengan Password Tanpa Simbol
TC13	Register dengan Password Kurang dari Delapan Karakter
TC14	Register dengan Menekan Tombol Daftar Berulang Kali
TC15	Login Valid
TC16	Login dengan Mengosongkan Email
TC17	Login dengan Mengosongkan Password
TC18	Login dengan Email Tanpa @
TC19	Login dengan Email Tanpa . (dot)
TC20	Login dengan Email yang Belum Terdaftar
TC21	Login dengan Password Tanpa Huruf Kapital
TC22	Login dengan Password Tanpa Huruf Kecil
TC23	Login dengan Password Tanpa Angka
TC24	Login dengan Password Tanpa Simbol
TC25	Login dengan Password Kurang dari Delapan Karakter
TC26	Login dengan Password yang Tidak Valid Lebih dari Tiga Kali
TC27	Login dengan Menekan Tombol Login Berulang Kali
TC28	Forgot Password Valid
TC29	Forgot Password dengan Mengosongkan Email
TC30	Forgot Password dengan Email Tanpa @
TC31	Forgot Password dengan Email Tanpa . (dot)
TC32	Forgot Password dengan Email yang Belum Terdaftar
TC33	Top Up dengan Memilih Pilihan Nominal yang Tersedia dan Sudah Login
TC34	Top Up dengan Mengisi Jumlah Nominal dan Sudah Login
TC35	Top Up dengan Mengisi Nominal Nol dan Sudah Login
TC36	Top Up Tanpa Mengisi Nominal dan Sudah Login
TC37	Top Up dengan Memilih Pilihan Nominal yang Tersedia dan Sudah Login namun Unverified
TC38	Top Up dengan Mengisi Jumlah Nominal dan Sudah Login namun Unverified

ID	Test Case
TC39	Top Up dengan Mengisi Nominal Nol dan Sudah Login namun Unverified
TC40	Top Up Tanpa Mengisi Nominal dan Sudah Login namun Unverified
TC41	Top Up dengan Memilih Pilihan Nominal yang Tersedia dan Belum Login
TC42	Top Up dengan Mengisi Jumlah Nominal dan Belum Login
TC43	Top Up dengan Mengisi Nominal Nol dan Belum Login
TC44	Top Up Tanpa Mengisi Nominal dan Belum Login
TC45	Memesan Konsultasi Valid Sudah Login
TC46	Memesan Konsultasi Tanpa Mengisi Judul Konsultasi Sudah Login
TC47	Memesan Konsultasi Tanpa Memilih Waktu Konsultasi Sudah Login
TC48	Memesan Konsultasi dengan Poin Kosong Sudah Login
TC49	Memesan Konsultasi dengan Menekan Tombol Lanjutkan Berulang Kali
TC50	Memesan Konsultasi Valid Belum Login
TC51	Memesan Konsultasi Tanpa Mengisi Judul Konsultasi Belum Login
TC52	Memesan Konsultasi Tanpa Memilih Waktu Konsultasi Belum Login
TC53	Melihat Tiket Konsultasi yang Aktif Sudah Login
TC54	Melihat Tiket Konsultasi yang Aktif Belum Login
TC55	Melakukan Panggilan ke User Lain yang Belum Login
TC56	Melakukan Panggilan ke User Lain yang Sudah Login
TC57	Menerima Panggilan dari User Lain
TC58	Mematikan dan Menyalakan Kamera
TC59	Mematikan dan Menyalakan Mikrofon
TC60	Switch Kamera
TC61	Menekan Tombol Kembali Saat Melakukan Panggilan

ID	Automated	Hybrid
TC05	Passed	Passed
TC06	Passed	Passed
TC07	Passed	Passed
TC08	Warning	Warning
TC09	Passed	Passed
TC10	Passed	Passed
TC11	Passed	Passed
TC12	Passed	Passed
TC13	Passed	Passed
TC14	Warning	Warning
TC15	Passed	Passed
TC16	Passed	Passed
TC17	Passed	Passed
TC18	Passed	Passed
TC19	Passed	Passed
TC20	Passed	Passed
TC21	Passed	Passed
TC22	Passed	Passed
TC23	Passed	Passed
TC24	Passed	Passed
TC25	Passed	Passed
TC26	Failed	Failed
TC27	Cannot be Tested	Passed
TC28	Passed	Passed
TC29	Warning	Warning
TC30	Passed	Passed
TC31	Passed	Passed
TC32	Passed	Passed
TC33	Passed	Passed
TC34	Passed	Passed
TC35	Passed	Passed
TC36	Passed	Passed
TC37	Passed	Passed
TC38	Passed	Passed
TC39	Passed	Passed
TC40	Passed	Passed
TC41	Passed	Passed
TC42	Passed	Passed
TC43	Passed	Passed
TC44	Passed	Passed
TC45	Passed	Passed
TC46	Warning	Warning
TC47	Warning	Warning
TC48	Passed	Passed
TC49	Cannot be Tested	Passed
TC50	Passed	Passed
TC51	Warning	Warning
TC52	Passed	Passed
TC53	Passed	Passed
TC54	Passed	Passed
TC55	Passed	Passed
TC56	Passed	Passed
TC57	Warning	Passed
TC58	Passed	Passed
TC59	Passed	Passed
TC60	Passed	Passed
TC61	Failed	Failed

IV. HASIL DAN PEMBAHASAN

A. Pengujian Fungsionalitas

Tabel 2 menyajikan data bahwa pada pengujian automasi, terdapat lima puluh test case yang lulus uji, tujuh test case yang mendapatkan peringatan, dua test case yang tidak lulus uji, dan dua test case yang tidak dapat diuji. Sementara itu, pada pengujian hybrid, terdapat lima puluh tiga test case yang lulus uji, enam test case yang mendapatkan peringatan, dan dua test case yang tidak lulus uji.

Tabel 2. Hasil Pengujian

ID	Automated	Hybrid
TC01	Passed	Passed
TC02	Passed	Passed
TC03	Passed	Passed
TC04	Passed	Passed

B. Perbandingan Metode

1) Test Coverage

Dalam Tabel 3 disajikan data pengujian yang dilakukan, metode automasi berhasil menjalankan lima puluh sembilan test case, dengan lima puluh test case yang berhasil, tujuh test case mendapatkan peringatan, dan dua test case yang gagal. Namun, terdapat dua test case yang tidak dapat dilakukan, yaitu test case menekan tombol login dan lanjutkan pada pemesanan konsultasi secara berulang kali, karena tombol tersebut tidak dapat ditekan lebih dari sekali.

Tabel 3. Perbandingan Metrik Test Coverage

Status	Automated	Hybrid
<i>Passed</i>	50	53
<i>Warning</i>	7	6
<i>Failed</i>	2	2
<i>Cannot be Tested</i>	2	0
TOTAL	61	61

Berdasarkan metrik cakupan pengujian, perhitungan persentase cakupan pengujian pada masing-masing metode adalah sebagai berikut:

- Automated testing : $\frac{59}{61} \times 100\% = 96.72\%$
- Hybrid testing : $\frac{61}{61} \times 100\% = 100\%$

Sementara itu, metode hybrid berhasil menjalankan semua enam puluh satu test case, dengan lima puluh tiga test case yang berhasil, enam test case mendapatkan peringatan, dan dua test case yang gagal.

2) Time

Berdasarkan hasil pengujian yang disajikan pada Tabel 4, pengujian dilakukan dalam tiga percobaan. Metode automasi menunjukkan waktu yang lebih singkat dibandingkan metode hybrid. Rata-rata pengujian automasi membutuhkan waktu 1728,737 detik, sedangkan pengujian hybrid membutuhkan waktu 1791,002 detik dengan catatan *test case* yang dapat dieksekusi pada pengujian automasi lebih sedikit dibandingkan dengan pengujian *hybrid*. Selain itu juga seperti yang dipaparkan pada alur pengujian, metode *hybrid* melalui proses yang lebih panjang.

Tabel 4. Perbandingan Metrik Time

	Automated	Hybrid
Time I	1705,508 s	1777,987 s
Time II	1775,812 s	1833,733 s
Time III	1704,891 s	1761,287 s

3) Test Case Effectiveness

Dari hasil pengujian yang disajikan pada Tabel 5, didapatkan hasil bahwa dari empat puluh tujuh *test case* negatif, *automated testing* dapat mendeteksi *defect* sebanyak empat puluh lima. Sedangkan *hybrid testing* dapat mendeteksi sejumlah empat puluh enam *defect*.

Tabel 5. Perbandingan Metrik Test Case Effectiveness

	Automated	Hybrid
Defects Detected	40	41

- Automated : $\frac{40}{61} \times 100\% = 65.574\%$
- Hybrid : $\frac{41}{61} \times 100\% = 67.21\%$

Defect yang ditemukan terdiri dari *test case* tipe positif yang berstatus *failed* dan juga *test case* negatif yang berstatus *passed*.

V. SIMPULAN

Berdasarkan jumlah temuan bug pada pengujian pada aplikasi Ngampu, antara pengujian automasi dengan metode *hybrid* tidak jauh berbeda. Metode automasi mendeteksi lima bug *suggestion*, tiga bug *medium*, dan tiga bug *critical* dari total enam puluh satu *test case*. Pada pengujian *hybrid*, dari enam puluh satu *test case* ditemukan sebanyak lima bug *suggestion*, dua bug *medium*, dan satu bug *critical*. Dari hasil tersebut, dapat disimpulkan bahwa aplikasi Ngampu sudah layak untuk rilis dengan beberapa perbaikan

Berdasarkan metrik *test coverage*, metode hybrid jauh lebih baik dibandingkan dengan metode automation karena memiliki cakupan pengujian yang lebih luas yaitu sebesar 100%. Berdasarkan metrik *time*, metode automation jauh lebih cepat dibandingkan dengan metode hybrid dengan rata-rata lama pengujian selama 1728,737 detik. Berdasarkan metrik *test case effectiveness*, metode *hybrid* lebih unggul dari metode automasi persentase keefektifan sebesar 67.21%

REFERENSI

- [1] R. Ainanda dan E. V. Haryanto, "Rancang Bangun Aplikasi E-meeting Menggunakan WebRTC (Web Real time Communication) Design and Build E-meeting Applications Using WebRTC (Web Real time Communication)," 2020.
- [2] T. Abdulghani, M. Maulana, dan H. Gozali, "Sistem Konsultasi dan Bimbingan Online Berbasis Web Menggunakan Webrtc (Studi Kasus : Fakultas Teknik Universitas Suryakencana)," *Media Jurnal Informatika*, vol. 11, no. 2, 2019, [Daring]. Tersedia pada: <http://jurnal.unsur.ac.id/mjinformatika>
- [3] M. M. Muhtadi, M. Dhandy Friyadi, dan A. Rahmani, "Analisis GUI Testing pada Aplikasi E-Commerce menggunakan Katalom," 2019.
- [4] H. Singh, S. K. Jha, D. Gupta, dan A. V. Singh, "GUI Testing Android Application," dalam *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICRITO56286.2022.9965072.
- [5] L. Setiyani, "Techno Xplore Jurnal Ilmu Komputer dan Teknologi Informasi PENGUJIAN SISTEM INFORMASI INVENTORY PADA PERUSAHAAN DISTRIBUTOR FARMASI MENGGUNAKAN METODE BLACK BOX TESTING," 2019.
- [6] A. R. Rambe, "Pengujian Otomatis dengan Teknik Black Box Menggunakan Appium," 2022.
- [7] L. Baird, Z. Shan, dan V. Namboodiri, "Automated Dynamic Detection of Self-Hiding Behavior," dalam *Proceedings - 2019 IEEE 16th International Conference on Mobile Ad Hoc and Smart Systems Workshops, MASSW 2019*, Institute of Electrical and Electronics Engineers Inc., Nov 2019, hlm. 87–91. doi: 10.1109/MASSW.2019.00024.

- [8] T. Garrett dan B. Gauf, "Useful Automated Software Testing Metrics," 2011. [Daring]. Tersedia pada: <http://www.thefreedictionary.com/metric>
- [9] A. N. Hasibuan dan T. Dirgahayu, "Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna," 2021.
- [10] R. Y. Rahmanda, E. Sakti Pramukantoro, dan W. Yahya, "Perancangan dan Implementasi Kelas Virtual FILKOM Universitas Brawijaya dengan Memanfaatkan Teknologi WebRTC (Web Real-Time Communication)," 2018. [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>

Pengembangan Sistem Informasi Tugas Akhir Berbasis Web Program Studi TRPL SV UGM

Siti Qotijah¹, Kholifah Murniati^{2*};

¹Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
sitiqotijah2018@mail.ugm.ac.id

²Departemen Ekonomi dan Bisnis, Sekolah Vokasi, Universitas Gadjah Mada;
kholifahmurniati99@mail.ugm.ac.id

*Korespondensi: kholifahmurniati99@mail.ugm.ac.id;

Abstract - The Final Project (TA) or Final Project (PA) is one of the requirements for graduation to get a Bachelor of Applied Science (S.Tr.) degree in the D4 Software Engineering Technology (TRPL) program at SV UGM. Academic responsibilities include processing final project data and student graduation, from registration to the submission of the final project report. Although the Simaster system is currently in place, there are some incomplete features, such as the submission of final project proposals from faculty projects, which must be done outside the system using Google Form. Students who have passed the defense must create a self-approval signature sheet for examiners, and the submission of final project reports is not yet integrated into the system. Therefore, the author plans to develop a final project information system using PHP, the Codeigniter framework, and the MySQL database. Testing is done manually and with automation testing using a tool called Selenium. Manual testing results show that the system can manage registration to the submission of final project reports well and without bugs.

Keywords: Registration, Information System, Final Project, Exam

Intisari - Tugas Akhir (TA) atau Proyek Akhir (PA) merupakan salah satu syarat untuk kelulusan dan mendapatkan gelar Sarjana Terapan (S.Tr.) pada prodi D4 Teknologi Rekayasa Perangkat Lunak (TRPL) SV UGM. Salah satu tanggung jawab akademik yaitu memproses data proyek akhir dan kelulusan mahasiswa. Mulai dari pendaftaran tugas akhir sampai pengumpulan laporan proyek akhir. Proses tersebut sudah ada pada sistem Simaster sekarang ini, akan tetapi ada beberapa *fitur* yang belum ada seperti fitur pengajuan tugas akhir dari proyek dosen. Proses pengajuan tersebut harus dilakukan di luar sistem dengan menggunakan *google form* sehingga pihak akademik butuh waktu untuk membuat *form* di setiap periode pendaftaran tugas akhir. Mahasiswa yang sudah dinyatakan lulus sidang harus membuat lembar tanda tangan persetujuan penguji sendiri serta pengumpulan laporan proyek akhir belum ada di sistem tersebut. Berdasarkan permasalahan-permasalahan tersebut, maka pengembangan sistem perlu untuk dilakukan untuk membantu dan mempermudah mengelola pendaftaran proyek akhir hingga pengumpulan laporannya. Pengembangan bisa membantu pihak Akademik, Dosen, dan Mahasiswa untuk proses proyek akhir sampai pengumpulan laporan. Pengembangan Sistem Informasi Tugas Akhir ini dibangun dengan menggunakan bahasa pemrograman PHP, *framework* Codeigniter, dan *database* MySQL. Pengujian sistem dilakukan secara manual dengan mencoba langsung dan juga melalui *automation testing* dengan alat bernama selenium. Hasil pengujian secara manual sudah sesuai dengan *input output* pengguna dan bisa digunakan dalam mengelola pendaftaran proyek akhir hingga pengumpulan laporan akhir yang sudah disahkan. Pengujian dengan selenium pada halaman akademik/kaprodi, dosen dan mahasiswa menunjukkan berhasil dan tidak ada *bug* sistem.

Kata kunci: Pendaftaran, Sistem Informasi, Proyek Akhir, Sidang

I. PENDAHULUAN

Sarjana Terapan Teknologi Rekayasa Perangkat Lunak (TRPL) adalah salah satu program studi yang berada di bawah naungan Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada. Salah satu tanggung jawab akademik yaitu memproses data proyek akhir dan kelulusan mahasiswa. Mulai dari pendaftaran tugas akhir hingga pengumpulan laporan proyek akhir.

Proses tersebut sudah ada pada sistem Simaster saat ini, akan tetapi ada beberapa *fitur* yang belum ada seperti *fitur* pengajuan tugas akhir dari proyek dosen. Proses pengajuan

tersebut harus dilakukan di luar sistem dengan menggunakan *google form* sehingga pihak akademik membutuhkan waktu harus membuat *form* di setiap periode pendaftaran tugas akhir. Fitur kuota bimbingan dosen belum ada di sistem simaster untuk memudahkan rekap jumlah bimbingan dosen. Fitur pembuatan jadwal sidang melalui *export excel*, sehingga memerlukan waktu saat akan dilakukan penjadwalan. Mahasiswa yang sudah dinyatakan lulus sidang harus membuat lembar tanda tangan persetujuan penguji sendiri, sehingga membutuhkan waktu untuk membuatnya, serta

pengumpulan laporan proyek akhir belum ada di sistem tersebut.

Berdasarkan permasalahan-permasalahan tersebut, maka pengembangan sistem perlu untuk dilakukan untuk membantu dan mempermudah mengelola pendaftaran proyek akhir hingga pengumpulan laporan yang sudah disahkan. Pengembangan bisa membantu pihak Akademik, Dosen, dan Mahasiswa untuk proses proyek akhir sampai pengumpulan laporan.

II. DASAR TEORI

A. Proyek Akhir

Tugas Akhir (TA) atau Proyek Akhir (PA) merupakan salah satu syarat untuk kelulusan dan mendapatkan gelar Sarjana Terapan (S.Tr.) dalam prodi D4 Teknologi Rekayasa Perangkat Lunak (TRPL) SV UGM. Dimana mahasiswa membuat laporan akhir dari suatu permasalahan yang telah diselesaikan.

B. Sistem Informasi

Sistem informasi adalah sistem yang ada dalam suatu organisasi yang mengintegrasikan kebutuhan harian untuk memproses transaksi, mendukung operasional, mencerminkan aktivitas manajerial dan strategis organisasi, serta memberikan laporan kepada pihak eksternal yang memerlukan informasi tersebut [1].

C. Website

Website atau lebih dikenal dengan web dapat diartikan sebagai suatu kumpulan halaman yang menampilkan berbagai macam informasi berupa teks, data, gambar diam maupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun dinamis, yang dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau *hyperlink* [2].

D. Metode Waterfall

Metode *waterfall* diuraikan memiliki beberapa tahapan pengembangan, yaitu perencanaan, analisis, desain, implementasi dan pemeliharaan sistem yang sudah berhasil dibangun [3].

E. PHP

PHP dalam pengembangan web adalah untuk mengatur tampilan konten sesuai dengan situasi yang diinginkan. PHP digunakan untuk membuat web yang dinamis, yang memungkinkan pengolahan data dari *form* dan interaksi dengan pengguna. Selain itu, PHP juga dapat digunakan untuk

menyimpan data ke dalam *database* dan memodifikasi tampilan web sesuai dengan *input* dari pengguna [4].

F. Codeigniter

CodeIgniter adalah salah satu contoh *framework* tersebut, yang menyediakan berbagai jenis *library* yang mempermudah pengembangan program tanpa perlu menulis semua kode dari awal. CodeIgniter dibangun menggunakan konsep pengembangan *MVC (Model-View-Controller)*. Pengontrol akan menggunakan model untuk mengakses data dan *view* untuk menampilkan data pada *browser* yang digunakan oleh pengguna [5].

G. MySQL

MySQL adalah salah satu program *Database Management System (DBMS)*. Keunggulan MySQL meliputi kemudahan akses dengan biaya gratis, keandalan yang terjamin, kemampuan untuk mendapatkan pembaruan terbaru, dan ketersediaan forum-forum yang membantu pengguna mengatasi kendala saat menggunakan MySQL [6].

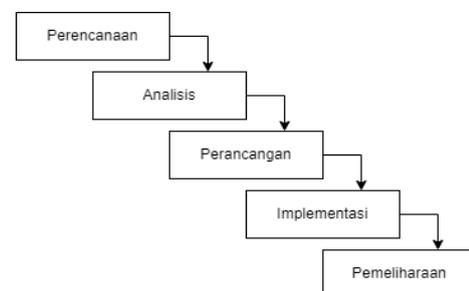
H. XAMPP

XAMPP adalah sebuah perangkat lunak gratis yang mendukung berbagai sistem operasi dan merupakan gabungan dari beberapa program [7].

III. METODOLOGI

A. Tahapan Penelitian

Pada pengembangan Sistem Informasi Tugas Akhir Berbasis Web Program Studi TRPL menggunakan metodologi *waterfall*. Dimana metodologi *waterfall* didasarkan pada pengembangan yang berulang. Pengembangan akan dimulai dengan tahapan pengembangan, yaitu perencanaan, analisis, desain, implementasi dan pemeliharaan sistem yang sudah berhasil dibangun atau seperti pada Gambar 1.



Gambar 1. Tahapan model *waterfall* [3]

B. Kebutuhan Sistem

1. Pengguna Sistem

Dalam pengembangan sistem informasi ini terdapat beberapa pengguna sistem, meliputi:

- Staf Akademik
- Kaprodi
- Dosen
- Mahasiswa

2. Kebutuhan fungsional

Kebutuhan fungsional menjelaskan mengenai gambaran secara umum proses yang ada pada sistem yang akan dibangun. Berikut ini adalah kebutuhan fungsional dari sistem yang sudah dibangun dan akan dikembangkan:

- a) Akademik dan kaprodi dapat melihat status mahasiswa.
- b) Akademik dan kaprodi dapat mengubah status pengajuan proyek akhir mahasiswa.
- c) Akademik dan kaprodi dapat melihat progres bimbingan mahasiswa yang sudah terdaftar sebagai pengguna.
- d) Akademik dan kaprodi dapat melakukan *import* dokumen (data *user* mahasiswa, dosen).
- e) Akademik dan kaprodi dapat melihat *submit* laporan akhir mahasiswa.
- f) Dosen dapat mengajukan dan mengubah usulan proyek.
- g) Dosen dapat melihat kuota dan mahasiswa bimbingannya.
- h) Dosen dapat melihat jadwal menguji mahasiswa.
- i) Mahasiswa dapat mengajukan usulan ide atau memilih proyek untuk dijadikan proyek akhir.
- j) Mahasiswa dapat mengajukan perubahan judul proyek akhir.
- k) Mahasiswa dapat menambahkan catatan bimbingan ke dosen pembimbing.
- l) Mahasiswa dapat *men-generate* lembar pengesahan
- m) Mahasiswa dapat *men-upload* laporan proyek akhir.

3. Kebutuhan Non-fungsional

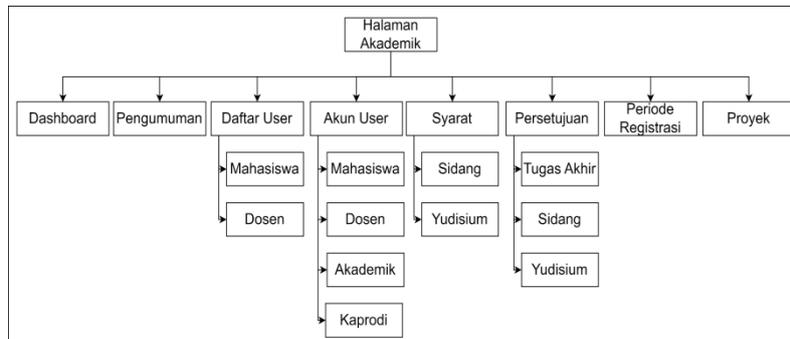
Persyaratan non-fungsional mencakup persyaratan sistem yang melampaui persyaratan fungsional, atau dikenal juga sebagai persyaratan dukungan sistem. Berikut ini adalah kebutuhan non-fungsional dari sistem ini, yaitu:

- a) Sistem berbasis web.
- b) Sistem hanya dapat diakses oleh akademik, kaprodi, dosen, dan mahasiswa Program Studi Teknologi Rekayasa Perangkat Lunak SV UGM.
- c) Akun dosen dan mahasiswa dibuat melalui akademik.
- d) Seluruh pengguna sistem hanya bisa menggunakan sistem ketika sudah login.
- e) Sistem mampu berjalan pada sistem operasi komputer Windows.

Langkah analisis dilakukan sebelum memulai pengembangan untuk mengidentifikasi masalah yang ada dalam sistem dan menetapkan kebutuhan yang harus dipenuhi selama proses pengembangan.

A. Struktur Menu

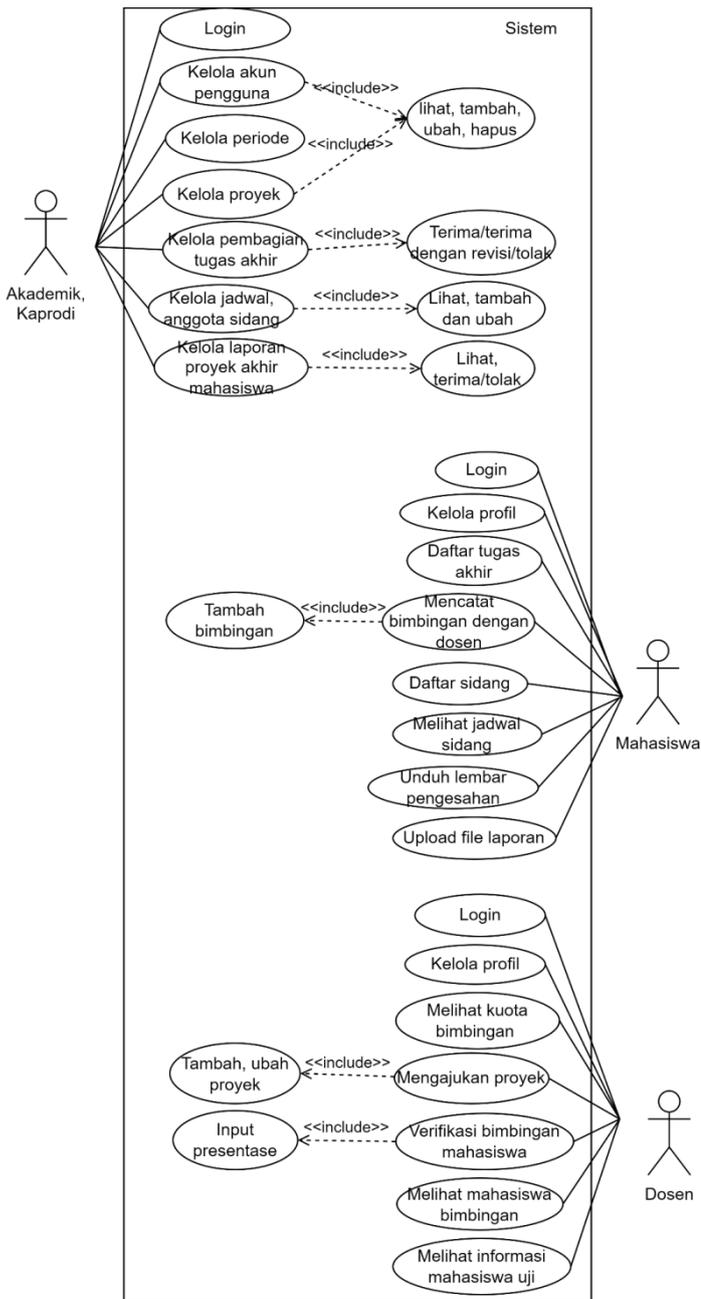
Sistem ini akan diaplikasikan pada lingkup sistem informasi pengajuan tugas akhir mahasiswa untuk prodi TRPL SV UGM, akademik, dosen dan kaprodi. Penggambaran *fitur* pada sistem yang terdapat pada sistem ditunjukkan struktur menu yang ada pada Gambar 2.



Gambar 2. Struktur menu

4. Rancangan Proses

Rancangan proses menjelaskan gambaran proses-proses atau alur yang berjalan pada sistem yang dibangun. rancangan proses akan dibuat *menggunakan use case diagram* seperti pada Gambar 3.



Gambar 3. Use case diagram

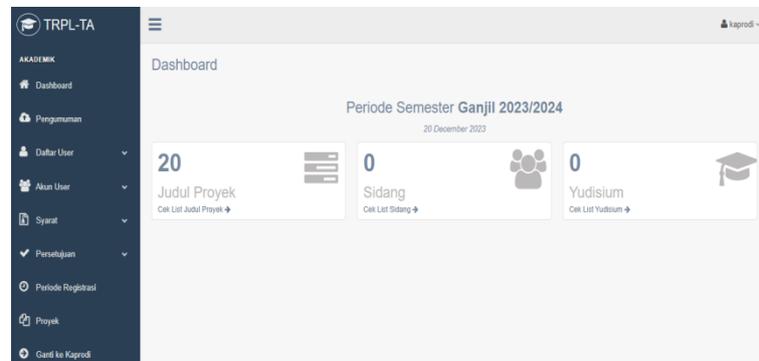
4. PHP versi 5.6, HTML, CSS, dan javascript sebagai bahasa pemrograman pada sisi server.
5. MySQL versi 10.4.11 sebagai basis data.
6. Codeigniter versi 3.1.6 sebagai framework.
7. Google Chrome sebagai web browser.
8. Visual Studio Code versi 1.46.1 sebagai text editor.

Pada proses implementasi sistem terdapat peralatan perangkat keras yang berguna untuk membangun sistem berbasis web ini antara lain:

1. Processor Intel® Celeron® CPU N4000 @1.10GHz
2. RAM 4.00 GB
3. Hard Disk 500 GB
4. Monitor LCD 14”
5. Keyboard
6. Mouse

B. Implementasi Sistem

Implementasi sistem pengajuan tugas akhir ini menggunakan codeigniter, dengan konsep MVC (model, view, controller). Konsep MVC akan memisahkan bagian untuk mengatur tampilan dan fungsi sistem. Untuk melakukan implementasi tampilan akan menggunakan framework Codeigniter. Tampilan dari halaman utama sistem dapat dilihat pada Gambar 4. Halaman utama sistem akan berisikan informasi umum dan daftar alur belajar yang dimiliki oleh sistem.



Gambar 4. Tampilan halaman utama

IV. HASIL DAN PEMBAHASAN

A. Spesifikasi Lingkungan Pengembangan

Perangkat lunak yang berguna untuk mengembangkan Sistem Informasi Tugas Akhir Berbasis Web Program Studi TRPL ini antara lain:

1. Windows 10 sebagai sistem operasi.
2. XAMPP versi 3.2.4 sebagai paket aplikasi Apache, MySQL, dan PHP.
3. Apache versi 2.4.41 sebagai web server.

C. Pengujian

Pada tahap ini dimana sistem akan diuji kelayakannya. Pengujian sistem pada penelitian ini menggunakan teknik *automation testing* dan *manual test* dengan menggunakan perangkat lunak bernama selenium guna mengevaluasi performa sistem. Pengujian manual dengan melakukan perancangan skenario pengujian, tahap pengujian dilakukan untuk mengecek apakah *input* dan *output* dari sistem berjalan dengan baik sesuai dengan fungsionalitasnya. Skenario pengujian ditunjukkan pada Tabel 1., Tabel 2. dan Tabel 3.

Tabel 1. Skenario Pengujian Halaman Akademik/Kaprodi

No	Pengujian	Aksi
1	Import data user mahasiswa	Upload file excel dan pilih kolom, klik tombol import
2	Persetujuan tugas akhir	Terima atau terima dengan revisi atau tolak pengajuan tugas akhir
3	Ubah data dosen dan kuota bimbingan	Isi data masukan
4	Cek bimbingan mahasiswa dan dosen	Lihat <i>persentase</i> bimbingan mahasiswa dan dosen
5	Buat jadwal sidang	Isi data masukan
6	Cek pengumpulan laporan	Klik <i>button</i> setuju atau tidak

Tabel 2. Skenario Pengujian Halaman Dosen

No	Pengujian	Aksi
1	Tambah proyek	Isi data masukan
2	Ubah proyek	Klik <i>button</i> edit dan isi data masukan
3	Verifikasi bimbingan, dan <i>persentase</i> bimbingan	Klik <i>button</i> aksi terima bimbingan, dan isi data masukan
4	Kuota bimbingan	Cek kuota bimbingan
5	Jadwal menguji (sidang)	Cek jadwal menguji

Tabel 3. Skenario Pengujian Halaman Mahasiswa

No	Pengujian	Aksi
1	Daftar tugas akhir	Isi data masukan
2	Ubah tugas akhir	Klik edit dan isi data masukan
3	Tambah catatan bimbingan	Isi data masukan
4	Daftar sidang	Klik tombol daftar dan isi data masukan
5	Generate lembar pengesahan	Klik <i>button download</i> lembar pengesahan
6	Submit laporan final	Upload laporan akhir <i>final</i>

D. Hasil Automation Testing

1. Bagian akademik/kaprodi

Testing yang dilakukan pada bagian akademik/kaprodi meliputi: cek bimbingan dosen dan mahasiswa, edit kuota dosen, persetujuan proyek akhir, cek pengumpulan laporan. Hasil testing pada bagian akademik/kaprodi dapat dilihat pada Gambar 5.

Log	Reference
Running 'akademik'	
1. open on https://ta-siti.cosmolab.space/akademik/dashboard OK	
2. setWindowSize on 1382x744 OK	
3. click on linkText=Daftar User OK	
4. click on linkText=Dosen OK	
5. click on css= odd:nth-child(1) .btn OK	
6. click on linkText=Akun User OK	
7. click on linkText=Dosen OK	
8. click on css= odd:nth-child(1) .btn-primary OK	
9. click on id=kuota_mahasiswa OK	
10. type on id=kuota_mahasiswa with value 9 OK	
11. click on css= .btn-success OK	
12. click on linkText=Persetujuan OK	
13. click on linkText=Tugas Akhir OK	
14. click on css= even:nth-child(2) .fa OK	
15. click on linkText=Persetujuan OK	
16. click on linkText=Yudisium OK	
17. click on css= even:nth-child(2) .btn OK	
18. click on css= .fa-eye OK	
19. selectWindow on handle=\${win804} OK	
'akademik' completed successfully	

Gambar 5. Hasil testing pada bagian akademik/kaprodi

2. Bagian dosen

Testing yang dilakukan pada bagian dosen meliputi: cek kuota bimbingan dan verifikasi bimbingan. Hasil testing pada bagian dosen dapat dilihat pada Gambar 6.

Log	Reference
Running 'dosen'	
1. open on https://ta-siti.cosmolab.space/dosen OK	
2. setWindowSize on 1382x744 OK	
3. click on linkText=Mahasiswa OK	
4. click on linkText=Bimbingan OK	
5. click on css= even:nth-child(6) .fa-tasks OK	
6. Trying to find linkText=Respon... OK	
'dosen' completed successfully	

Gambar 6. Hasil testing pada bagian dosen

3. Bagian mahasiswa

Testing yang dilakukan pada bagian mahasiswa meliputi: pengajuan proyek akhir, tambah catatan bimbingan dan klik *download* lembar pengesahan. Hasil testing pada bagian mahasiswa dapat dilihat pada Gambar 7.

Log	Reference
Running 'mahasiswa'	
1. open on https://ta-siti.cosmolab.space/mahasiswa/pengajuan/tugasakhir OK	
2. setWindowSize on 1382x744 OK	
3. click on linkText=Edit OK	
4. click on name=proyektiga OK	
5. select on name=proyektiga with value label=Sistem Akademik (dosen) OK	
6. click on css=.form-group:nth-child(4) > .btn OK	
'mahasiswa' completed successfully	
2. setWindowSize on 1382x744 OK	
3. click on linkText=Data OK	
4. click on linkText=Tugas Akhir OK	
5. click on linkText=Tambah Bimbingan OK	
6. click on name=subject OK	
7. type on name=subject with value Bab 2 OK	
8. click on name=description OK	
9. type on name=description with value Laporan bimbingan bab 2 OK	
'mahasiswa' completed successfully	
1. open on https://ta-siti.cosmolab.space/mahasiswa/sidang OK	
2. setWindowSize on 1382x744 OK	
3. click on css=.fa-file-pdf-o OK	
4. selectWindow on handle=\${win8284} OK	
'mahasiswa' completed successfully	

Gambar 7. Hasil testing pada bagian mahasiswa

V. SIMPULAN

Berdasarkan hasil penelitian Pengembangan Sistem Informasi Tugas Akhir Berbasis Web Prodi Teknologi Rekayasa Perangkat Lunak (TRPL) pada penelitian ini, dapat diperoleh kesimpulan bahwa, telah berhasil dilakukan Pengembangan Sistem Informasi Tugas Akhir Prodi TRPL Berbasis Web yang dapat memudahkan akademik, kaprodi, dosen, dan mahasiswa dalam proses pengajuan proyek akhir, sidang, dan pengumpulan laporannya. Pengembangan Sistem Informasi Tugas Akhir Prodi TRPL Berbasis Web ini telah sesuai dengan SOP terbaru, dan juga memperbaiki beberapa *fitur* yang sebelumnya sudah ada, sesuai dengan kebutuhan *fungsionalitasnya*. Akademik dapat memantau mahasiswa, dosen pembimbing terkait proyek akhir. Akademik atau kaprodi dapat mengubah, dan melihat kuota bimbingan dosen. Mahasiswa bisa mendapatkan lembar pengesahan dengan mudah. Adanya sistem ini dapat menghemat dalam penggunaan kertas.

REFERENSI

- [1] Jogyanto., *Analysis Dan Design System Informasi: Pendekatan Terstruktur Teori Dan Praktik Aplikasi Bisnis*. Yogyakarta: Penerbit Andi, 2005
- [2] A. M. Rudianto, *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta: Andi Offset, 2011.
- [3] Pressman, R.S., *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I*, Yogyakarta: Andi, 2015.
- [4] Nixon, R., *Learning PHP, Mysql and Javascript*, O'Reilly Media, Inc., 2009.
- [5] Basuki, A.P., *Membangun Web Berbasis PHP dengan Framework CodeIgniter*, Yogyakarta: Penerbit Lokomedia, 2010.
- [6] Hidayatullah, P. & K., J. K., *Pemrograman Web*. Bandung: Informatika, 2014.
- [7] Buana, I. and Setia, K., *Jago pemrograman PHP*. Jakarta: Dunia Komputer, 2014.

Analisis Sentimen Berbasis Aspek pada Ulasan Pengguna Aplikasi Starbucks Menggunakan Algoritma *Support Vector Machine*

Muhammad Adin Palimbani¹, Rochana Prih Hasuti^{1,*}, Rian Adam Rajagede²

¹Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
madinpalimbani09@mail.ugm.ac.id

²Departemen Informatika, Universitas Islam Indonesia;
rian.adam@uii.ac.id

*Korespondensi: rochana.prih.h@ugm.ac.id;

Abstract – The development of digital transformation in all aspects of business today has changed the way companies add values to their business. One example is the coffee shop business company called Starbucks, which utilizes the trend of mobile application technology by developing a "Loyalty Rewards App". However, there are many negative reviews on the Google Play Store that developers should pay attention to. There has been no research on these reviews so that sentiment and information related to important aspects such as usability aspects of the application are still unknown. This study aims to determine the sentiment of Starbucks application user reviews as well as aspects of application usability namely learnability, efficiency, errors and satisfaction, using the SVM method, with three kernel functions namely Linear, Polynomial and RBF in classification cases. The data preprocessing stage is followed by feature extraction using TF-IDF. To get better classification results, hyperparameter tuning is also performed on the SVM model using GridSearchCV. The dataset is obtained from scraping user reviews of the Starbucks application on the Google Play Store. The results showed that the classification modeling using SVM has a fairly good performance with an average score of 88.96% accuracy score, 66.85% f1-score, 75.77% precision and 64.68% recall. Meanwhile, the analysis results show that the majority of sentiments are negative in all aspects of the application, especially in the errors aspect, which indicates a high error rate of the system.

Keywords – Aspect Based Sentiment Analysis, Support Vector Machine, Starbucks, Usability Aspects

Intisari – Perkembangan transformasi digital di semua aspek bisnis saat ini telah mengubah cara perusahaan memberikan nilai tambah bagi bisnis mereka. Salah satu contohnya yaitu perusahaan bisnis kedai kopi Starbucks yang memanfaatkan tren teknologi aplikasi *mobile* dengan mengembangkan "Loyalty Rewards App". Namun, banyak ditemukan ulasan negatif di Google Play Store yang harus diperhatikan pihak pengembang. Belum ada penelitian pada ulasan tersebut sehingga sentimen dan informasi terkait aspek penting seperti aspek-aspek *usability* aplikasi masih belum diketahui. Penelitian ini bertujuan untuk mengetahui sentimen ulasan pengguna aplikasi Starbucks serta aspek-aspek *usability* aplikasi yaitu *learnability*, *efficiency*, *errors* dan *satisfaction*, menggunakan metode SVM, dengan menguji tiga jenis kernel yaitu Linear, Polinomial dan RBF dalam kasus klasifikasi. Tahap *preprocessing* data diikuti dengan ekstraksi fitur menggunakan TF-IDF. Untuk mendapatkan hasil klasifikasi yang lebih baik, juga dilakukan Hyperparameter tuning pada model SVM menggunakan GridSearchCV. Dataset diperoleh dari hasil *scraping* ulasan pengguna aplikasi Starbucks di Google Play Store. Hasil penelitian menunjukkan bahwa pembuatan model klasifikasi menggunakan SVM memiliki performa yang cukup baik yaitu dengan rata-rata skor dari skor *accuracy* sebesar 88,96%, *f1-score* 66,85%, *precision* 75,77% dan *recall* 64,68%. Sementara itu, hasil analisis menunjukkan bahwa mayoritas sentimen bernilai negatif di seluruh aspek aplikasi terutama pada aspek *errors*, yang menandakan tingginya tingkat kesalahan pada sistem.

Kata kunci – Analisis Sentimen Berbasis Aspek, Support Vector Machine, Starbucks, Usability Aspects

I. PENDAHULUAN

Integrasi teknologi ke dalam semua aspek bisnis menjadi hal yang tak terelakkan di era transformasi digital saat ini. Bagi perusahaan, integrasi teknologi memiliki peranan dalam meningkatkan efisiensi operasional bisnis mereka. Salah satu bentuk integrasi teknologi tersebut yakni pemanfaatan aplikasi berbasis *mobile*. Berdasarkan data statistika terkait jumlah unduhan aplikasi seluler di seluruh dunia dari tahun 2016-2023, terjadinya peningkatan pengguna perangkat *mobile* yang cukup signifikan setiap tahunnya [1]. Hal ini memberikan peluang pemasaran yang luar biasa bagi suatu perusahaan guna meningkatkan nilai tambah bisnis mereka.

Perusahaan Starbucks merupakan salah satu perusahaan kedai kopi terbesar di dunia yang telah mengintegrasikan

perangkat *mobile* ke dalam bisnis mereka. Teknologi tersebut dikenal dengan sebutan "Starbucks Loyalty Rewards Mobile App". Salah satu contoh program loyalitas yang sering digunakan oleh perusahaan saat ini [2]. Sebuah perangkat *mobile* yang diintegrasikan dengan program Starbucks Rewards dimana hampir semua pertumbuhan bisnis Starbucks berasal dari program ini dan menjadi aplikasi *restaurant* paling populer sejak 2018 (*The Manifest 2018 Consumer App Survey*) [3]. Oleh karena itu, performa aplikasi sangat mempengaruhi penjualan bisnis mereka dalam hal ini keterlibatan aktivitas *user* pelanggan terhadap program Starbucks Rewards.

Hasil observasi ulasan pengguna di Google Play Store per 30 April 2023 menunjukkan nilai 1/5 lebih banyak daripada

nilai di atasnya. *Rating* aplikasi pun hanya memperoleh nilai 2.9/5 dari total keseluruhan ulasan pengguna. Hasil pelabelan manual yang dilakukan oleh 3 orang annotator juga menunjukkan banyak sekali ulasan negatif yang ditemukan dengan jumlah persentase mencapai 67,29% dari seluruh data sampel. Hal ini menunjukkan bahwa performa aplikasi dinilai sangat kurang sehingga *user* pelanggan memberi penilaian rendah. Akan tetapi, ulasan ini masih belum diketahui terkait informasi aspek mana yang dianggap kurang.

II. DASAR TEORI

Melalui pendekatan analisis sentimen [4] berbasis aspek-aspek *usability* aplikasi [5], penelitian ini dapat mengekstraksi aspek dan sentimen di tiap aspek *learnability*, *efficiency*, *memorability*, *errors* dan *satisfaction*. Penentuan aspek didasarkan pada teori Nielsen Norman Group yang berkaitan dengan “*The Design’s Functionality*” [6] dimana dapat diperoleh kesimpulan seberapa baik performa aplikasi bagi *user* pelanggan dilihat dari polaritas sentimen di setiap aspeknya.

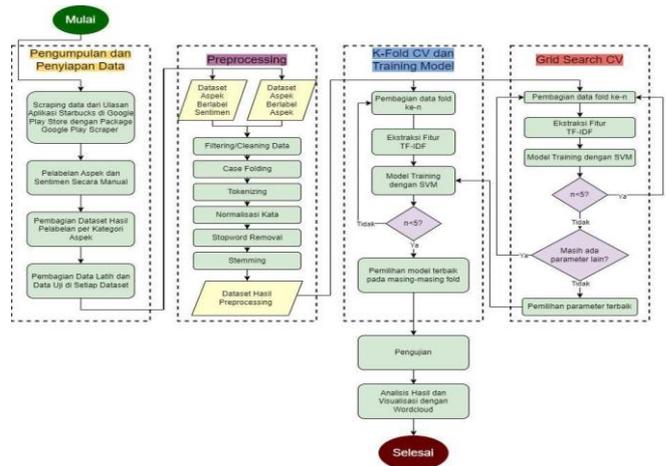
Penelitian [7], yang melakukan analisis sentimen berbasis aspek-aspek *usability* aplikasi KAI Access dengan menggunakan algoritma *Support Vector Machine* mampu menghasilkan model klasifikasi dengan performa yang baik. Diikuti dengan penanganan *imbalanced data* menggunakan *Random Oversampling*, penelitian ini mampu mengetahui sentimen dan informasi pada aspek *usability* aplikasi. Performa model klasifikasi yang dihasilkan pun cukup baik yaitu dengan rata-rata skor nilai *accuracy* mencapai 88.96%, *F1-score* 66.85%, presisi 75.77% dan *recall* 64.68%. Hal inilah yang mendasari penggunaan algoritma *Support Vector Machine* pada penelitian ini.

Berdasarkan latar belakang di atas, eksperimen dilakukan untuk menganalisis setiap aspek dan sentimen pada ulasan aplikasi Starbucks dengan menerapkan algoritma *Support Vector Machine*. Eksperimen diharapkan mampu menentukan model klasifikasi terbaik untuk mengetahui sentimen serta informasi di setiap kelas sentimen pada masing-masing aspek yang telah ditentukan. Analisis pada permasalahan ini belum pernah dilakukan sebelumnya, sehingga diharapkan dapat memberikan wawasan berharga kepada tim pengembangan aplikasi serupa. Selain itu, kontribusi dari penelitian ini adalah pada pengumpulan dan anotasi pada dataset baru, yang selanjutnya akan dipublikasikan terbuka sehingga dapat digunakan dalam penelitian selanjutnya di masa depan.

III. METODOLOGI

A. Tahapan Penelitian

Tahapan penelitian dimulai dari pengumpulan data, penyiapan data, *preprocessing*, *training* model klasifikasi, evaluasi dengan *5-Fold Cross Validation*, penentuan Hyperparameter terbaik dan analisis hasil. Tahapan penelitian secara jelas diilustrasikan pada Gambar 1.



Gambar 1. Diagram alir penelitian

B. Pengambilan Data

Dataset diperoleh dengan teknik *scraping* menggunakan *package Google Play Scraper* pada python. Dataset yang digunakan adalah dataset bahasa Indonesia dengan jumlah secara keseluruhan sebanyak 1714 dari 11600 ulasan yang ada di Google Play Store. Akan tetapi, data yang akan digunakan pada penelitian ini hanya 1500 ulasan saja.

C. Pelabelan Data

Proses pelabelan data dilakukan secara manual dimana anotasi kategori aspek pada setiap data ulasan berdasarkan observasi data secara langsung serta pelabelan dipecah menjadi dua yakni pelabelan aspek dan sentimen beraspek. Untuk pelabelan aspek, penentuan kategori aspek pada setiap data ulasan ditentukan berdasarkan teori *usability* atau kebergunaan suatu aplikasi pada Nielsen Norman Group terkait “*The Design’s Functionality*” [6] dimana pengaruh desain antarmuka suatu aplikasi sangat mempengaruhi kemudahan pengguna dalam menggunakan suatu aplikasi. Aspek-aspek *usability* yang dimaksud adalah sebagai berikut:

1. *Learnability* yaitu tingkat kemudahan aplikasi untuk dipelajari pengguna.
2. *Efficiency*, yaitu tingkat produktivitas pengguna akibat menggunakan aplikasi.
3. *Memorability*, yaitu tingkat kemudahan aplikasi untuk diingat pengguna pada tata cara penggunaannya.
4. *Errors*, yaitu tingkat kesalahan sistem pada aplikasi.
5. *Satisfaction*, yaitu kepuasan pengguna terhadap aplikasi yang digunakan.

D. Pembagian Data

Dataset yang sudah dilabeli, akan dibagi menjadi 8 dataset. Pertama, dataset dibagi berdasarkan jumlah kategori aspek yang ada, yaitu 4 dataset berlabel aspek untuk kategori aspek *learnability*, *efficiency*, *errors* dan *satisfaction*. Kedua, dibuat 4 dataset berlabel sentimen berdasarkan data pada

dataset berpelabelan aspek yang memiliki label 1 (data yang mengandung aspek). Yang terakhir, dilakukan pembagian data latih dan data uji untuk setiap dataset yang ada.

E. Preprocessing

Dataset yang telah dilabeli berdasarkan kategori aspek dan sentimen akan dilakukan *preprocessing* data yang terdiri dari *Case Folding*, *Symbol Removal*, Normalisasi kata, Tokenisasi dan *Stemming*. Hasil scraping, *preprocessing* dan *labelling* dataset yang dianotasi secara manual telah diterbitkan pada github [8]

F. Ekstraksi Fitur (TF-IDF)

Proses ekstraksi fitur dilakukan dengan menggunakan *library Scikit Learn* pada bahasa pemrograman python yang dapat memberi bobot setiap kata atau fitur berdasarkan bobot TF-IDF pada data latih [9]. Ekstraksi fitur dengan metode ini dilakukan pada data latih dan data validasi di setiap pembagian *fold* data pada tahap *K-Fold Cross Validation*.

G. Pembuatan Model Klasifikasi

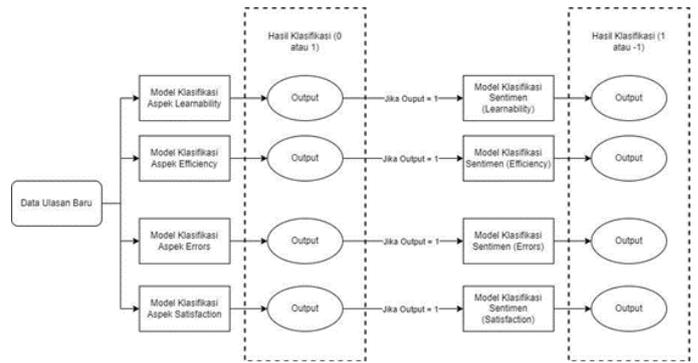
Pembuatan model klasifikasi menggunakan algoritma *Support Vector Machine (SVM)* dengan tiga jenis kernel yaitu Linear, Polinomial dan RBF [10]. Penerapan Hyperparameter Tuning menggunakan *Grid Search Cross Validation* juga dilakukan pada penelitian ini guna mendapatkan kombinasi terbaik dari Hyperparameter SVM yang menghasilkan klasifikasi lebih optimal. Tidak hanya itu, pembuatan model klasifikasi dengan metode SVM tanpa Hyperparameter Tuning juga dilakukan dengan tujuan pembandingan hasil akurasi model yang dibuat. Metode *K-Fold Cross Validation* juga diterapkan sebagai proses validasi. Proses ini dilakukan untuk mengetahui performa model pada distribusi data latih dan data uji yang berbeda beda.

H. Pengujian

Tahap pengujian dilakukan untuk menguji performa model pada data uji. Performa model diukur dari nilai *confusion matrix* yakni akurasi, presisi, *recall* dan *f1-score* yang diperoleh setelah proses pengujian.

I. Analisis Hasil dan Visualisasi Data

Analisis hasil dilakukan berdasarkan hasil implementasi model terbaik hasil pengujian pada data uji guna mendapatkan hasil klasifikasi aspek dan sentimen di setiap kategori aspek. Berikut ilustrasi penggunaan model untuk mengekstrak aspek dan mengklasifikasi polaritas sentimen pada data uji diilustrasikan pada Gambar 2.



Gambar 2. Alur Penerapan Model Klasifikasi

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengambilan Data

Dari proses pengambilan data menggunakan teknik *scraping* dengan menggunakan *package google play scraper*, diperoleh 1500 data ulasan berbahasa indonesia aplikasi Starbucks di Google Play Store. Perolehan data ulasan diambil dari bulan Maret 2019 sampai bulan April 2023 dilihat pada Tabel 1.

Tabel 1. Hasil Sampel Scraping Data

Username	Ulasan	Waktu	Rating
Najla Desniwa	Gabisa di dowload	2023-03-21 14:25:02	1
Tati Papatungan	tidak bisa di sign in parah banget aplikasi ini	2023-03-21 13:51:35	1
selabintana haha	Dih tollol sign up susah kali	2023-03-21 10:44:34	1
Lana fauziiia	Mau log in INVALID trs. Tolong diperbaikiiiiiiiiiiii	2023-03-19 19:03:46	1
Data Unlimited	Order aja loading nya g karu2 an, app luar kalau udh masuk indo ya gini	2023-03-17 8:50:38	1

B. Hasil Pelabelan Data

Pada tabel 2, label 0 dan 1 digunakan untuk menandai ada atau tidaknya aspek pada suatu ulasan sedangkan, untuk label 1, -1 dan 0 pada tabel 3, menunjukkan adanya sentimen positif dan negatif pada ulasan yang mengandung aspek.

Tabel 2. Sampel Hasil Pelabelan Manual Aspek

Ulasan	Learnability	Efficiency	Errors	Satisfaction
gabisa ambil rewards padahal starnya udah cukup	0	0	1	0
Semakin gak jelas... update ni aplikasi aja harus login accout google segala. Terlalu ribetttt. Tolong diperbaiki.	1	1	0	0
aplikasi dibuat itu untuk memprmudah.				
Bukan untuk mempersulit pengguna aplikasi. Oke, bisa dipahami untik menjaga data custemer, tapi bukan dengan cara seperti ini.. terlalu bertele-tele				
Starbucks cabang di Palangka Raya, Kalimantan Tengah, gk ada di appnya, gimana coba?	0	0	1	0
Biasa login pake no hp sekarang minta email.kaga jelas amat sih....	0	1	0	0
Mungkin sebaiknya ada bahasa Indonesianya	0	0	1	0

Tabel 3. Sampel Hasil Pelabelan Manual Sentimen Beraspek

Ulasan	Learnability	Efficiency	Errors	Satisfaction
gabisa ambil rewards padahal starnya udah cukup	0	0	-1	0

Ulasan	Learnability	Efficiency	Errors	Satisfaction
Semakin gak jelas... update ni aplikasi aja harus login accout google segala. Terlalu ribetttt. Tolong diperbaiki.	-1	-1	0	0
Starbucks cabang di Palangka Raya, Kalimantan Tengah, gk ada di appnya, gimana coba?	0	0	-1	0
Biasa login pake no hp sekarang minta email.kaga jelas amat sih....	0	-1	0	0
Mungkin sebaiknya ada bahasa Indonesianya	0	0	1	0

C. Hasil Pembagian Data Latih dan Data Uji

Dari Tabel 4 dan Tabel 5, dapat diketahui jumlah masing-masing data untuk data latih dan data uji hasil yang merupakan dari pembagian data pada setiap dataset.

Tabel 4. Jumlah Data Dari Hasil Pembagian Data Berlabel Aspek

Aspek	Data Latih	Data Uji	Total
Learnability	1200	300	1500
efficiency	1200	300	1500
errors	1200	300	1500
satisfaction	1200	300	1500
Total			6000

Tabel 5. Jumlah Data Dari Hasil Pembagian Data Berlabel Sentimen

Aspek	Data Latih	Data Uji	Total
Learnability	103	26	129
Efficiency	102	25	127
Errors	670	168	838
Satisfaction	409	102	511
Total			1605

D. Analisis Word Cloud pada Data Latih

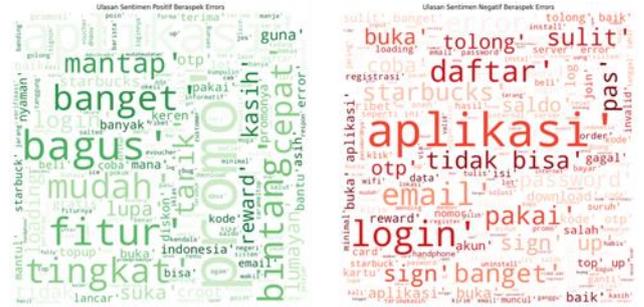
Tahapan ini bertujuan untuk mengetahui fitur-fitur kata paling berpengaruh dan mempunyai bobot paling tinggi pada sentimen positif dan sentimen negatif di setiap aspek data latih sehingga mengetahui pengaruh hasil prediksi model klasifikasi di akhir nanti. Gambar 3 menunjukkan aspek *efficiency*. Gambar 4 menunjukkan aspek *learnability*. Gambar 5 menunjukkan aspek *errors*. Gambar 6 menunjukkan aspek *satisfaction*.



Gambar 3. Awan Kata Sentimen Positif (Hijau) dan Negatif (Merah) pada Aspek *Efficiency*



Gambar 4. Awan Kata Sentimen Positif (Hijau) dan Negatif (Merah) pada Aspek *Learnability*



Gambar 5. Awan Kata Sentimen Positif (Hijau) dan Negatif (Merah) pada Aspek *Errors*



Gambar 6. Awan Kata Sentimen Positif (Hijau) dan Negatif (Merah) pada Aspek *Satisfaction*

E. Hasil Pembuatan Model Klasifikasi

Hasil pembuatan model klasifikasi menggunakan tiga jenis kernel pada algoritma *Support Vector Machine* ditampilkan pada Tabel 6.

Tabel 6. *Hyperparameter* Terbaik Hasil *Grid Search Cross Validation*

Model Klasifikasi	Jenis Klasifikasi	C	Gamma	Kernel
<i>Learnability</i>	Aspek	0.1	0.01	Linear
	Sentimen	10	0.1	RBF
<i>Efficiency</i>	Aspek	10	0.1	RBF
	Sentimen	10	0.1	RBF
<i>Errors</i>	Aspek	10	0.1	RBF
	Sentimen	10	1	RBF
<i>Satisfaction</i>	Aspek	10	1	RBF
	Sentimen	10	0.1	RBF

F. Hasil Validasi Model Klasifikasi

Hasil validasi model klasifikasi menggunakan tiga jenis kernel pada algoritma *Support Vector Machine* ditampilkan pada Tabel 7.

Tabel 7. Hasil Performa Validasi Model Klasifikasi *Support Vector Machine* dengan Hyperparameter Terbaik

Model Klasifikasi	Jenis Klasifikasi	C	Gamma	Kernel
Learnability	Aspek	0.1	0.01	Linear
	Sentimen	10	0.1	RBF
Efficiency	Aspek	10	0.1	RBF
	Sentimen	10	0.1	RBF
Errors	Aspek	10	0.1	RBF
	Sentimen	10	1	RBF
Satisfaction	Aspek	10	1	RBF
	Sentimen	10	0.1	RBF

G. Analisis Umum

Dari hasil penelitian ini menunjukkan bahwa pembuatan model klasifikasi menggunakan algoritma *Support Vector Machine* diikuti dengan Hyperparameter terbaik, menghasilkan performa yang baik dalam pembuatan decision boundary pada klasifikasi data ulasan aplikasi Starbucks dengan perolehan rata-rata skor akurasi sebesar 88.96%, f1-score 66.85%, presisi 75.77%, dan recall 64.68%. Akan tetapi, nilai tersebut belum bisa dikatakan sempurna karena tidak adanya penanganan *imbalanced data* terutama pada dataset sentimen berespek *errors* dan aspek *learnability*. Hal ini dikarenakan karena kondisi *imbalanced data* menjadi masalah dalam klasifikasi karena *classifier learning* akan condong memprediksi ke kelas data mayoritas daripada kelas minoritas. Akibatnya, dihasilkan akurasi prediksi yang baik terhadap kelas data latih yang banyak (mayoritas) sedangkan untuk kelas data latih yang sedikit (minoritas) akan dihasilkan akurasi prediksi yang buruk. Pada dataset sentimen berespek *errors*, hasil *labelling* menunjukkan jumlah sentimen positif sebanyak 45 ulasan dan sentimen negatif sebanyak 793 ulasan dengan hasil akurasi model sebesar 90.75%, *f1-score* 61.40%, presisi 60.40% dan *recall* 62.43% seperti yang ditunjukkan pada tabel 7. Pada dataset aspek *learnability*, hasil *labelling* menunjukkan jumlah ulasan yang mengandung aspek

learnability sebanyak 129 ulasan dan tidak mengandung aspek *learnability* sebanyak 1371 ulasan dengan hasil akurasi model sebesar 92,25%, f1-score 47,98%, presisi 46.125% dan recall 50%. Kedua dataset latih ini menunjukkan hasil *f1-score* yang rendah yang artinya model klasifikasi memiliki nilai presisi dan *recall* yang buruk dan ini berpengaruh pada kedekatan antara data aktual dan data prediksinya. Walaupun *imbalanced data* tidak berpengaruh terhadap nilai akurasi model, klasifikasi SVM dengan Hyperparameter Tuning tetap mampu menghasilkan nilai akurasi yang baik di setiap kelas aspeknya. Hasil pengujian model klasifikasi ditampilkan pada Tabel 8.

Tabel 8. Hasil Pengujian Model Klasifikasi

Jenis Klasifikasi	Kategori Aspek	Avg. Accuracy	Avg. F1-Score	Avg. Precision	Avg. Recall
Aspek	<i>Learnability</i>	92.25	47.98	46.125	50.0
	<i>Efficiency</i>	98.67	95.78	97.89	93.90
	<i>Errors</i>	99.42	99.41	99.37	99.44
Sentimen	<i>Satisfaction</i>	98.5	98.36	98.03	98.73
	<i>Learnability</i>	97.92	91.30	95.12	88.40
	<i>Efficiency</i>	98.25	91.82	95.83	88.48
Average	<i>Errors</i>	90.75	61.40	60.40	62.43
	<i>Satisfaction</i>	95.92	93.27	96.45	90.81
Average		96.46	84.92	86.15	84.02

V. SIMPULAN

Berdasarkan hasil labelling terhadap data ulasan pengguna aplikasi Starbucks, diketahui bahwa mayoritas sentimen bernilai negatif untuk seluruh aspek aplikasi, terutama pada aspek *errors* yang merupakan aspek yang paling banyak dibicarakan pada data ulasan dengan persentase sentimen negatif mencapai 73,46% dari keseluruhan sentimen negatif berespek *errors*, dengan keluhan-keluhan yang berkaitan dengan *login*, *sign up*, kode otp, *top up* dan penukaran *reward* pada aplikasi Starbucks.

Dengan mengimplementasikan algoritma *Support Vector Machine* menggunakan Hyperparameter Tuning Grid Search CV terbaik pada data ulasan yang mengandung mayoritas kelas negatif di setiap aspek, hasil rata-rata skor dari model klasifikasi tersebut diperoleh nilai *accuracy* sebesar 88.96%, *f1-score* 66.85%, *precision* 75.77%, dan *recall* 64.68%.

Pada penelitian selanjutnya, diharapkan menggunakan data ulasan berbahasa Inggris yang memiliki jumlah data lebih besar daripada berbahasa Indonesia sehingga hasil pembuatan model algoritma menjadi lebih akurat dalam melakukan prediksi di setiap aspeknya. Tidak hanya itu, dengan mencoba menerapkan metode penanganan *imbalanced data* pada hasil *labelling* aspek di aspek *learnability* dan sentimen beraspek *errors* guna mendapatkan nilai *f1-score* dan presisi yang lebih baik dari penelitian ini.

REFERENSI

- [1] L. Ceci, 'Number of Mobile App Downloads Worldwide From 2016 to 2023'. [Online]. Available: <https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>
- [2] L. A. Maulana, S. J. Putra, and N. Hasanati, 'Pengaruh Program Loyalitas Terhadap Loyalitas Konsumen Aplikasi Go-Jek di Indonesia', *Fak. Sains Dan Teknol. Univ. Islam Negeri Syarif Hidayatullah Jkt.*, no. 17 Oct 2019, [Online]. Available: <https://repository.uinjkt.ac.id/dspace/handle/123456789/48364>
- [3] C. E. Orozco Mora, *Analysis of User's Sentiment Towards Starbucks on Twitter*. 2020. doi: 10.13140/RG.2.2.22406.60484.
- [4] C. Cahyaningtyas, Y. Nataliani, and I. R. Widiyanti, 'Analisis sentimen pada rating aplikasi Shopee menggunakan metode Decision Tree berbasis SMOTE', *Fak. Teknol. Inf. Univ. Kristen Satya Wacana*, no. Vol. 18 No. 2 (2021), Nov. 2021, doi: <https://doi.org/10.24246/aiti.v18i2.173-184>.
- [5] M. Bangsa, S. Priyanta, and Y. Suyanto, 'Aspect-Based Sentiment Analysis of Online Marketplace Reviews Using Convolutional Neural Network', *IJCCS Indones. J. Comput. Cybern. Syst.*, vol. 14, p. 123, Apr. 2020, doi: 10.22146/ijccs.51646.
- [6] J. Nielsen, 'Usability 101: Introduction to Usability'. [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- [7] H. Mustakim, 'Analisis Sentimen Berbasis Aspek Pada Ulasan Pengguna Aplikasi KAI Access Menggunakan Support Vector Machine dan Random Oversampling', *ETD Univ. Gadjah Mada*, Sep. 2021, [Online]. Available: <https://etd.repository.ugm.ac.id/penelitian/detail/204690>
- [8] 'Starbucks App Reviews Dataset'. [Online]. Available: <https://github.com/adinplb/asba-sbuxapp-skripsi/tree/main>
- [9] E. S. Oktaviani, 'Analisis Sentimen Berbasis Aspek Menggunakan Metode Support Vector Machine pada Data Ulasan Restaurant', *Univ. Gadjah Mada*, Agustus 2020, [Online]. Available: <https://etd.repository.ugm.ac.id/penelitian/detail/191001>
- [10] A. Z. Praghakusma and N. Charibaldi, 'Komparasi Fungsi Kernel Metode Support Vector Machine untuk Analisis Sentimen Instagram dan Twitter (Studi Kasus: Komisi Pemberantasan Korupsi)', *J. Sarj. Tek. Inform.*, vol. 9, No.2, pp. 33–42, Jun. 2021, doi: <http://journal.uad.ac.id/index.php/JSTIF/index>.