

PREDIKSI STATUS INTEGRASI SPBU DI WILAYAH TELKOM MAKASSAR MENGUNAKAN METODE DECISION TREE

1st Muh. Nur
School of Computing
Telkom University
Bandung, Indonesia

awinkrun@student.telkomuniversity.ac.id

2nd Indwiarti
School of Computing
Telkom University
Bandung, Indonesia

indwiarti@telkomuniversity.ac.id

3rd Fitriyani
School of Computing
Telkom University
Bandung, Indonesia

fitriyani@telkomuniversity.ac.id

Abstrak — Digitalisasi SPBU bertujuan meningkatkan efisiensi operasional melalui integrasi perangkat seperti dispenser, display, nozzle, dan EDC. Masalah utama adalah integrasi perangkat digitalisasi ke sistem yang belum optimal, sehingga menyebabkan data tidak akurat dan laporan tidak konsisten. Penelitian ini menggunakan metode Decision Tree untuk memprediksi status integrasi perangkat berdasarkan 534 data operasional SPBU di wilayah Telkom Makassar dengan 45 atribut.

Ketidakeimbangan data membuat model lebih akurat mendeteksi Partial Integrasi dibandingkan Full Integrasi. Data diseimbangkan menggunakan SMOTE sebelum diterapkan pada model Decision Tree, yang dievaluasi melalui akurasi, precision, recall, dan F1-score. Hyperparameter tuning dilakukan untuk meningkatkan performa.

Hasil penelitian menunjukkan bahwa setelah data diseimbangkan menggunakan SMOTE, model Decision Tree berhasil meningkatkan kemampuan dalam mengenali kelas minoritas. Evaluasi akhir setelah SMOTE dan hyperparameter tuning menghasilkan akurasi sebesar 69%, precision 62%, recall 89%, dan F1-score 73%. Hasil ini menunjukkan bahwa model mampu memprediksi status integrasi perangkat dengan lebih seimbang, mendukung pengambilan keputusan untuk meningkatkan sistem digitalisasi SPBU.

Kata kunci—Digitalisasi SPBU, Decision Tree, Status Integrasi, SMOTE, Hyperparameter Tuning

I. PENDAHULUAN

Digitalisasi SPBU bertujuan meningkatkan efisiensi operasional melalui integrasi perangkat seperti PC POS, ATG Console, Router, Switch, Dispenser, dan EDC [1]. Integrasi ini mendukung pencatatan transaksi yang akurat, pemantauan stok secara real-time, serta pengelolaan pembayaran yang lebih efisien [2].

Sistem berbasis Internet of Things (IoT) dan cloud computing diterapkan di SPBU untuk mengatasi ketidaksesuaian stok dan meningkatkan transparansi distribusi bahan bakar. Di Wilayah Telkom Makassar, teknologi ini dikembangkan untuk meningkatkan efisiensi operasional. IoT memungkinkan pemantauan tangki secara

real-time [3]. Cloud computing memastikan data selalu tersedia dan dapat diakses kapan saja [4].

Salah satu tantangan dalam digitalisasi SPBU adalah status integrasi perangkat, yaitu Full Integrasi, di mana semua perangkat terhubung ke sistem utama untuk pemantauan real-time, dan Partial Integrasi, di mana tidak semua perangkat terhubung, sehingga menyebabkan ketidaksesuaian data dan keterlambatan laporan.

Penelitian ini bertujuan mengklasifikasi Status Integrasi SPBU menggunakan algoritma Decision Tree dan mengembangkan model untuk mengidentifikasi pola dalam data operasional SPBU [5]. Dataset yang digunakan adalah Analisis Data Operasional (ADO) di Wilayah Telkom Makassar.

Penelitian [6] berjudul “Classification of Adult Income Using Decision Tree” menggunakan metode Decision Tree untuk mengklasifikasi pendapatan dewasa dengan dataset Adult Income dari UCI Machine Learning Repository. Hasilnya, model Decision Tree mencapai akurasi 85% dalam memprediksi kelas pendapatan individu.

Penelitian [7] berjudul “The Evaluation of Online Education Course Performance Using Decision Tree Mining Algorithm” menganalisis kinerja kursus online dengan algoritma Decision Tree. Dataset berasal dari data kinerja kursus keuangan pada semester pertama tahun akademik 2019-2020 di sebuah universitas. Hasilnya menunjukkan bahwa faktor seperti minat pada matematika, sikap belajar, kehadiran, dan penyelesaian tugas mandiri berpengaruh signifikan terhadap prestasi mahasiswa.

Penelitian [8] berjudul “Predicting Heart Disease using Tree-based Model” menggunakan Decision Tree untuk memprediksi penyakit jantung dengan dataset heart disease dari UCI Machine Learning Repository. Hasilnya menunjukkan bahwa Decision Tree Classifier memiliki akurasi 81% untuk pasien dengan penyakit jantung dan 82% untuk pasien tanpa penyakit jantung.

Berdasarkan latar belakang yang telah dijelaskan, penelitian ini berfokus pada bagaimana mengimplementasikan metode Decision Tree untuk

memprediksi status integrasi pada unit SPBU di Wilayah Telkom Makassar dan mengevaluasi tingkat akurasi model dalam melakukan prediksi tersebut.

Penelitian ini bertujuan untuk memprediksi status integrasi SPBU menggunakan metode Decision Tree serta mengukur kinerja model klasifikasi yang dihasilkan.

II. KAJIAN TEORI

A. Stasiun Pengisian Bahan Bakar Umum (SPBU)

SPBU (Stasiun Pengisian Bahan Bakar Umum) adalah fasilitas yang digunakan untuk menyalurkan dan memasarkan bahan bakar minyak (BBM) serta produk lain yang berhubungan dengan kendaraan [9]. SPBU memiliki berbagai standar operasional dalam pembangunan dan pengelolaannya, termasuk aspek ekonomi, teknis, keselamatan kerja, serta lokasi [10].

Terdapat tiga jenis SPBU berdasarkan kepemilikan dan operasionalnya, yaitu, SPBU COCO dimiliki dan dioperasikan langsung oleh perusahaan pengelola BBM. SPBU CODO dimiliki perusahaan BBM, dikelola oleh mitra atau dealer. SPBU DODO dimiliki dan dioperasikan oleh pihak swasta dengan lisensi perusahaan BBM. [10].

Dispenser adalah perangkat utama SPBU untuk menyalurkan bahan bakar, dilengkapi sistem pengukuran volume dan harga [11]. Display adalah bagian dari dispenser yang menampilkan informasi seperti volume bahan bakar yang dikeluarkan, harga per liter, dan total biaya transaksi. Nozzle adalah ujung selang dispenser yang berfungsi untuk mengalirkan bahan bakar ke dalam tangki kendaraan [12].



Gambar 1. Dispenser, Display dan Nozzle

Electronic Data Capture (EDC) di SPBU memproses pembayaran digital dan terhubung ke sistem transaksi serta dispenser. Jika dispenser terintegrasi, EDC mendukung Self-Service Payment dengan input nomor polisi atau Payment Console untuk transaksi otomatis. Jika tidak terintegrasi, EDC tidak dapat digunakan pada dispenser tersebut [13].



Gambar 2. EDC

ATG Console (Automatic Tank Gauging) memonitor stok bahan bakar di tangki penyimpanan. Tangki pendam dilengkapi ATG Probe untuk mendeteksi volume bahan bakar secara otomatis, sedangkan tangki atas tidak dimonitor ATG karena ukurannya kecil dan berada di atas tanah [14].

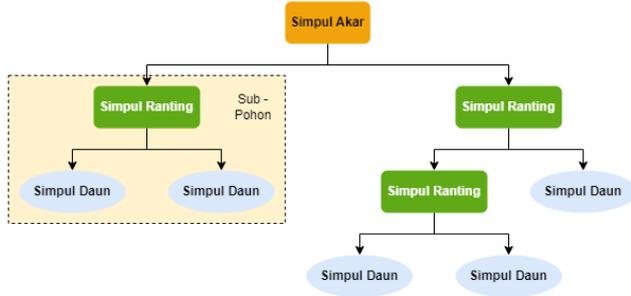


Gambar 3. ATG Console

B. Decision Tree

Decision Tree adalah salah satu teknik klasifikasi yang banyak digunakan dan dapat diterapkan pada data dalam jumlah besar. Ini adalah pembelajaran mesin terawasi berbasis pohon yang digunakan untuk mengklasifikasikan atau membuat prediksi berdasarkan jalur bagaimana pertanyaan sebelumnya dijawab [8]. Decision Tree merupakan model berturut-turut yang menyatukan rangkaian pengujian dasar secara efisien dan kohesif dimana fitur numerik dibandingkan dengan nilai ambang batas pada setiap pengujian [15]. Aturan konseptual jauh lebih mudah untuk dibangun daripada bobot numerik dalam jaringan saraf koneksi antar simpul. Terutama untuk tujuan pengelompokan, Decision Tree digunakan. Selain itu, Decision Tree adalah model klasifikasi yang biasa digunakan dalam Data Mining. Simpul dan cabang terdiri dari setiap pohon. Setiap simpul mewakili fitur dalam kategori yang akan diklasifikasikan dan setiap subset menentukan nilai yang dapat diambil oleh simpul [5]. Karena analisisnya yang sederhana dan ketepatannya pada berbagai bentuk data, Decision Tree telah menemukan banyak bidang implementasi [16].

Struktur Decision Tree



Gambar 4. Struktur Decision Tree

Seperti yang ditunjukkan pada Gambar 4, pohon terdiri dari simpul dan cabang, dengan langkah pembentukan berupa pembelahan, penghentian, dan pemangkasan [6]. Terdapat tiga jenis simpul: simpul akar sebagai titik awal pembagian, simpul ranting sebagai pemisah data berdasarkan atribut, dan simpul daun sebagai titik akhir keputusan. Setiap simpul merepresentasikan fitur yang diklasifikasikan menjadi dua subset eksklusif, sementara cabang menunjukkan aliran hasil dari simpul akar.

Untuk menghitung nilai root node pada decision tree, digunakan Gini index atau Gini impurity. Gini index adalah ukuran yang digunakan untuk menilai ketidakmurnian atau ketidakpastian dalam suatu set data. Nilainya berkisar antara 0 hingga 1, di mana nilai 0 menunjukkan bahwa semua elemen dalam set berasal dari kelas yang sama (murni), sedangkan nilai mendekati 1 menunjukkan distribusi elemen yang merata di antara semua kelas (tidak murni) [17].

Gini index untuk suatu set data T yang terdiri dari n kelas didefinisikan sebagai:

$$G(T) = 1 - \sum_{j=1}^n p_j^2 \quad (1)$$

Keterangan:

$G(T)$ = Gini index untuk set data T

n = Jumlah kelas dalam set data

p_j = Frekuensi relatif dari kelas j dalam set T , yang dihitung sebagai:

$P_j = \frac{N_j}{N}$ di mana N_j adalah jumlah elemen dari kelas j dan N adalah total jumlah elemen dalam set T .

Ketika membangun pohon keputusan, kita ingin memilih variabel dan nilai yang akan memisahkan data menjadi dua subset T_1 dan T_2 . Gini index untuk pemisahan ini dihitung sebagai:

$$Gini_{split}(T) = \frac{N_1}{N} G(T_1) + \frac{N_2}{N} G(T_2) \quad (2)$$

Keterangan:

$Gini_{split}(T)$ Gini index untuk pemisahan data T menjadi dua subset T_1 dan T_2

N_1 = Jumlah elemen dalam subset T_1

N_2 = Jumlah elemen dalam subset T_2

N = Jumlah elemen dalam subset T

$G(T_1)$ = Gini index untuk subset T_1

$G(T_2)$ = Gini index untuk subset T_2

C. Synthetic Minority Oversampling Technique (SMOTE)

Metode untuk menangani masalah ketidakseimbangan kelas dalam data klasifikasi. Ketidakseimbangan kelas terjadi ketika jumlah sampel dari satu kelas jauh lebih besar

dibandingkan dengan kelas lainnya, yang dapat menyebabkan model klasifikasi menjadi bias terhadap kelas mayoritas dan mengabaikan kelas minoritas [18].

Synthetic Minority Oversampling Technique (SMOTE) bekerja dengan membuat sampel sintetis baru untuk kelas minoritas guna mengatasi masalah ketidakseimbangan kelas dalam dataset [19]. Berikut adalah langkah-langkah utama dalam algoritma SMOTE:

1. Identifikasi Sampel Kelas Minoritas
SMOTE hanya beroperasi pada kelas yang jumlah sampelnya lebih sedikit dibandingkan kelas lainnya.
2. Menentukan K-Tetangga Terdekat
Untuk setiap sampel dalam kelas minoritas, SMOTE mencari k tetangga terdekat menggunakan metode seperti Euclidean distance.
3. Membuat Sampel Sintetis
Dari k tetangga terdekat, dipilih secara acak salah satu tetangga (x). Sampel sintetis (x_{baru}) dihitung menggunakan interpolasi linear, yang ditunjukkan pada persamaan (3).
$$(x_{baru} = x_i + (x^{\wedge} - x_i) \times \delta) \quad (3)$$

Dimana:
 x_i = adalah sampel asli kelas minoritas.
 x^{\wedge} = adalah tetangga terdekat yang dipilih.
 δ = adalah bilangan acak antara 0 dan 1.
4. Menambahkan Sampel Sintetis ke Dataset
Proses ini diulang hingga jumlah sampel di kelas minoritas meningkat sesuai kebutuhan.

Dengan demikian, SMOTE meningkatkan kinerja model pembelajaran mesin dengan memastikan kelas minoritas memiliki representasi yang lebih baik tanpa sekadar menduplikasi sampel yang ada, sehingga mengurangi risiko overfitting [19].

D. Korelasi Pearson

Korelasi Pearson adalah metode analisis korelasi yang populer untuk mengidentifikasi hubungan linear antara dua variabel. Diperkenalkan oleh Karl Pearson sekitar tahun 1900, metode ini memerlukan hipotesis dan hanya dapat diterapkan pada data berskala interval atau rasio. Koefisien korelasi (r) berkisar antara -1 hingga 1, di mana nilai mendekati 1 menunjukkan hubungan positif yang kuat, nilai mendekati -1 menunjukkan hubungan negatif yang kuat, dan nilai mendekati 0 menunjukkan hubungan yang lemah [20]. Tanda koefisien juga penting, dengan nilai positif menunjukkan hubungan searah dan nilai negatif menunjukkan hubungan berlawanan. Korelasi Pearson bertujuan menentukan koefisien korelasi (r) untuk mengukur tingkat hubungan serta kontribusi antara variabel bebas (x) dan variabel terikat (y) [21]. Rumus Korelasi Pearson ditunjukkan pada persamaan (4).

$$r_{xy} = \frac{N \cdot \sum XY - (\sum X) \cdot (\sum Y)}{\sqrt{\{N \cdot \sum X^2 - (\sum X)^2\} \cdot \{N \cdot \sum Y^2 - (\sum Y)^2\}}} \quad (4)$$

Dimana:

x = Variabel pertama

y = Variabel kedua

n = Banyaknya pengamatan

E. Confusion Matrix

Confusion matrix adalah tabel yang digunakan dalam machine learning untuk mengevaluasi kinerja model klasifikasi. Confusion matrix menggambarkan performa model dengan membandingkan prediksi yang dilakukan oleh model dengan nilai sebenarnya dari data yang diuji [22]. Berikut representasi Confusion Matrix.

Tabel 1. Confusion Matrix

	POSITIVE (1)	NEGATIVE (0)
POSITIVE (1)	TP	FP
NEGATIVE (0)	FN	TN

Berdasarkan representasi Confusion Matrix pada Tabel 1, maka dijelaskan bahwa:

1. True Positive (TP): Prediksi positif yang benar
2. True Negative (TN): Prediksi negatif yang benar
3. False Positive (FP): Prediksi positif yang salah (Type I error)
4. False Negative (FN): Prediksi negatif yang salah (Type II error)

Untuk menghitung Precision, recall, F1-score, dan akurasi dapat digunakan rumus berikut:

- A. Metrik Precision: Rasio jumlah sampel yang diklasifikasikan positif dengan benar terhadap jumlah seluruh sampel yang diklasifikasikan positif. [22]. Rumus Precision ditunjukkan pada persamaan (5).

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

- B. Metrik recall: Rasio jumlah sampel yang diklasifikasikan secara benar sebagai positif dengan jumlah seluruh sampel positif sebenarnya [22]. Rumus recall ditunjukkan pada persamaan (6).

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

- C. Metrik Akurasi: Rasio jumlah sampel yang diklasifikasikan dengan benar (hasil sebenarnya) terhadap jumlah total sampel yang diuji [22]. Rumus akurasi ditunjukkan pada persamaan (7).

$$Akurasi = \frac{TP+TN}{N} = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

- D. Metrik F1-Score: rata-rata harmonik dari Precision dan Recall. Artinya, ketika nilai Precision ditingkatkan, Recall cenderung turun, dan sebaliknya [23]. Rumus F1-score ditunjukkan pada persamaan (8).

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \frac{Precision*Recall}{Precision+Recall} \quad (8)$$

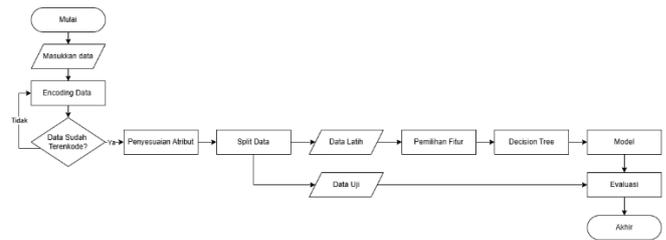
F. Hyperparameter Tuning

Hyperparameter tuning adalah proses mencari nilai optimal untuk hyperparameter dalam model machine learning, yang harus ditentukan sebelum training dimulai karena tidak dipelajari secara otomatis. Tujuan utamanya adalah meningkatkan performa dan akurasi model dengan menemukan kombinasi hyperparameter yang optimal [24]. Salah satu metodenya adalah Grid Search, yaitu teknik yang secara sistematis mencari kombinasi hyperparameter dalam grid yang telah ditentukan dengan mengevaluasi model untuk setiap kombinasi yang ada [25].

III. PERANCANGAN SISTEM

A. Desain Perancangan Sistem

Berikut adalah Flowchart Decision Tree, yang dapat dilihat dalam Gambar 5.



Gambar 5. Flowchart Decision Tree

Proses dimulai dengan memasukkan data ke dalam sistem, kemudian dilakukan encoding untuk mengonversi data kategori menjadi numerik. Setelah itu, data disesuaikan melalui penyesuaian atribut, lalu dibagi menjadi data latih dan data uji. Dari data latih, fitur relevan dipilih untuk membangun model Decision Tree. Model yang terbentuk kemudian dievaluasi menggunakan data uji untuk mengukur kinerjanya, dan proses berakhir setelah evaluasi selesai.

B. Dataset

Pada penelitian ini, data yang digunakan adalah Analisis Data Operasional Unit SPBU di Wilayah Telkom Makassar. Data yang diuji terakhir diperbarui pada Maret 2024 dan terdiri dari 534 baris dan 45 kolom, dengan 13 fitur bertipe float64, 7 fitur bertipe int64, dan 25 fitur bertipe object. Variabel target dalam dataset ini adalah Status Integrasi, yang merupakan variabel biner untuk memprediksi apakah SPBU memiliki dispenser dengan Partial Integrasi atau Full Integrasi.

Jumlah SPBU yang telah didigitalisasi sebanyak 110 SPBU. Setiap SPBU memiliki berbagai merek dispenser, dengan masing-masing SPBU memiliki lebih dari satu dispenser. Setiap dispenser dilengkapi dengan display, nozzle, dan perangkat EDC. Daftar atribut dataset dapat dilihat pada Tabel 2.

Tabel 2. Daftar Atribut Dataset

No.	Atribut
1	SPBU
2	REGIONAL
3	WITEL
4	STATUS INTEGRASI
5	STATUS DIGITALISASI
6	ALAMAT
7	KOTA/KAB
8	JUMLAH DISPENSER
9	TATSUNO C
10	TATSUNO NC
11	GILBARCO C
12	GILBARCO NC
13	TOKHEIM C
14	TOKHEIM NC
15	TATSUNO ADAT C
16	TATSUNO ADAT NC
17	WAYNE C
18	WAYNE NC

No	Atribut
19	TOMINAGA C
20	TOMINAGA NC
21	SOMO PETRO C
22	SOMO PETRO NC
23	ENE C
24	ENE NC
25	LG C
26	LG NC
27	PRIME C
28	PRIME NC
29	SANKI C
30	SANKI NC
31	TOKICO C
32	TOKICO NC
33	CENSTAR C
34	CENSTAR NC
35	PRIMER C
36	PRIMER NC
37	DISPLAY CONNECT
38	DISPLAY NOT CONNECT
39	NOZZLE CONNECT
40	NOZZLE NOT CONNETC
41	PRODUCT
42	EDC A920
43	EDC Z90
44	EDC N5
45	EDC A930

Berdasarkan Tabel 2, Dataset mencakup informasi SPBU, jumlah dispenser, merek dispenser, perangkat EDC, status integrasi, status digitalisasi, alamat, dan wilayah operasional. Setiap dispenser dikategorikan sebagai C (Connect) untuk terintegrasi dan NC (Not Connect) untuk tidak terintegrasi. Dataset juga mencatat kondisi perangkat EDC, baik yang berfungsi maupun rusak. Proses encoding dilakukan untuk mengubah data kategorikal menjadi numerik. Variabel target Status Integrasi diencoding menjadi biner: Partial Integrasi = 0 dan Full Integrasi = 1. Pada variabel Merek Dispenser, kategori C diencoding menjadi 1 dan NC menjadi 0. Untuk variabel EDC, EDC Rusak diencoding menjadi 0, sedangkan kategori lainnya menjadi 1. Dua atribut baru dibuat: DISPENSER untuk hasil encoding kategori C dan NC, serta EDC untuk hasil encoding perangkat EDC. Fitur utama dalam klasifikasi meliputi Dispenser, EDC, Display Connect, Display Not Connect, Nozzle Connect, dan Nozzle Not Connect, sementara atribut tidak relevan dibuang. Data kemudian dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian. Data latih digunakan untuk membangun model, sedangkan data uji mengevaluasi kinerja model dalam melakukan prediksi.

IV. HASIL IMPLEMENTASI

Hasil klasifikasi menggunakan Decision Tree dievaluasi dengan akurasi, precision, recall, F1-score, dan confusion matrix. Distribusi kelas Status Integrasi dianalisis sebelum dan sesudah SMOTE, diikuti pembagian data (80% latih, 20% uji). Analisis korelasi antar variabel dilakukan untuk memahami hubungan fitur, dan model dievaluasi untuk mengukur efektivitas metode.

A. Distribusi Kelas Pada Variabel Status Integrasi dan Proses Penyeimbangan Menggunakan SMOTE

Analisis distribusi kelas dilakukan untuk memeriksa keseimbangan data pada variabel target (Status Integrasi). Ketidakseimbangan data dapat berdampak pada performa model klasifikasi, terutama dalam mendeteksi kelas minoritas. Hasil distribusi data ditampilkan pada Tabel 3.

Tabel 3. Distribusi Data Kelas Variabel Status Integrasi

Kelas	Sebelum SMOTE	Sesudah SMOTE
Partial Integrasi	357	357
Full Integrasi	177	357

Distribusi awal Status Integrasi menunjukkan ketidakseimbangan, dengan 357 data untuk Partial Integrasi dan 177 data untuk Full Integrasi, sebagaimana ditampilkan pada Tabel 3. SMOTE diterapkan untuk menyeimbangkan data tanpa mengurangi data asli, meningkatkan jumlah Full Integrasi menjadi 357, sehingga total data menjadi 714. Distribusi seimbang ini memungkinkan model belajar lebih representatif, mengurangi bias, dan meningkatkan akurasi prediksi.

B. Split Data Setelah Penyeimbangan dengan SMOTE.

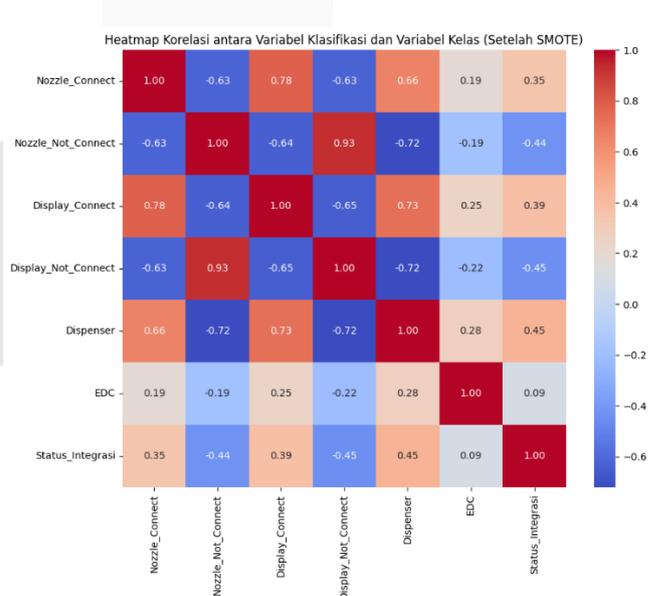
Setelah SMOTE, dataset dibagi menjadi 80% data latih untuk melatih model dan 20% data uji untuk mengevaluasi performa prediksi. Berikut adalah jumlah data setelah dilakukan split pada Tabel 4.

Tabel 4. Split Data Latih dan Data Uji

Data Latih	Data Uji
571	143

C. Korelasi Antara Variabel Kelas dan Variabel Klasifikasi.

Matriks korelasi digunakan untuk memahami hubungan antara fitur dalam dataset dan variabel target. Matriks ini memberikan wawasan tentang kontribusi setiap fitur terhadap klasifikasi. Heatmap korelasi dapat dilihat pada Gambar 5.



Gambar 6. Korelasi antara Variabel Kelas dan Variabel Klasifikasi

Dari Gambar 6, heatmap korelasi menampilkan hubungan antara variabel kelas dengan variabel-variabel klasifikasinya.

Heatmap ini memberikan visualisasi yang jelas mengenai kekuatan dan arah hubungan antarvariabel.

Selanjutnya, korelasi antara variabel-variabel tersebut dapat dilihat secara rinci pada Tabel 4.3, yang menyajikan nilai korelasi dalam bentuk numerik untuk memberikan pemahaman yang lebih terperinci mengenai hubungan antarvariabel.

Tabel 5. Nilai Korelasi Variabel

Variabel	Korelasi
Dispenser	0.452760
Display_Connect	0.387332
Nozzle_Connect	0.353081
EDC	0.089893
Nozzle_Not_Connect	-0.437716
Display_Not_Connect	-0.449217

Tabel 5 menunjukkan nilai korelasi antara beberapa variabel yang terkait dengan pengukuran atau kondisi tertentu. Berikut adalah penjelasan masing-masing variabel beserta nilai korelasinya:

1. Dispenser (0.452760): Korelasi positif moderat, menunjukkan bahwa semakin tinggi nilai Dispenser, semakin tinggi pula nilai variabel lain yang berkorelasi dengannya. Hubungan ini cukup kuat, tetapi tidak sangat signifikan.
2. Display_Connect (0.387332): Korelasi positif yang cukup rendah, menunjukkan adanya hubungan antara Display_Connect dan variabel lain, meskipun pengaruhnya tidak terlalu kuat.
3. Nozzle_Connect (0.353081): Korelasi positif lemah, menunjukkan bahwa hubungan antara Nozzle_Connect dan variabel lain ada, tetapi kekuatannya rendah.
4. EDC (0.089893): Korelasi sangat rendah, menunjukkan bahwa hubungan antara EDC dan variabel lain hampir tidak ada atau sangat lemah. Pengaruhnya terhadap variabel lain tidak signifikan.
5. Nozzle_Not_Connect (-0.437716): Korelasi negatif moderat, yang berarti semakin tinggi nilai Nozzle_Not_Connect, semakin rendah nilai variabel lain yang berkorelasi dengannya. Hubungan ini cukup signifikan dalam arah yang berlawanan.
6. Display_Not_Connect (-0.449217): Korelasi negatif moderat, menunjukkan bahwa ketika Display_Not_Connect meningkat, variabel lain yang berkorelasi dengannya cenderung menurun dengan kekuatan yang cukup kuat.

Secara keseluruhan, nilai-nilai korelasi ini menggambarkan hubungan antara variabel-variabel yang diuji. Korelasi positif menunjukkan hubungan langsung, sedangkan korelasi negatif menunjukkan hubungan terbalik. Semakin tinggi atau rendah nilai korelasi, semakin kuat hubungan antara variabel, sementara nilai yang mendekati nol menunjukkan hubungan yang sangat lemah atau tidak ada hubungan.

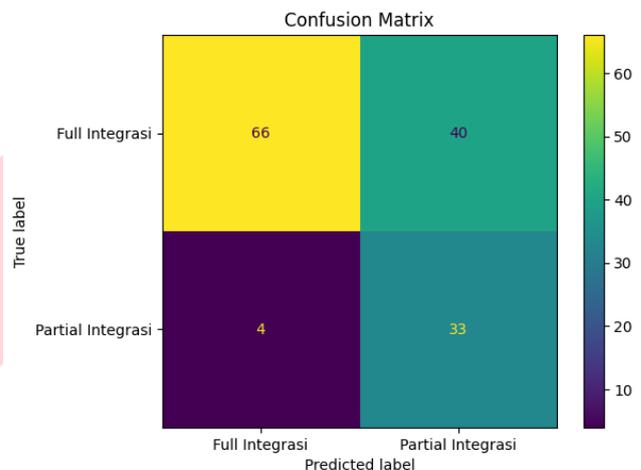
D. Evaluasi Model Klasifikasi.

Evaluasi model dilakukan dengan menggunakan Confusion Matrix untuk menghitung metrik-metrik klasifikasi seperti Precision, Recall, F1-Score, dan Akurasi.

Hasil evaluasi ini membantu mengukur performa model dalam memprediksi data uji.

a. Hasil Confusion Matrix

Setelah membangun model *Decision Tree*, diperoleh parameter model sebagai berikut: Kedalaman maksimum = 9, sampel daun minimum = 30, dan sampel split minimum = 58. Parameter ini menunjukkan kompleksitas pohon keputusan yang dihasilkan dalam proses klasifikasi. Hasil evaluasi model ditunjukkan melalui *confusion matrix* pada Gambar 7.



Gambar 7. Hasil Confusion Matrix

Dari hasil confusion matrix pada Gambar 6, dijelaskan performa model dalam memprediksi setiap kelas:

1. True Positive (TP): 66 data yang sebenarnya Full Integrasi (1) berhasil diprediksi dengan benar sebagai Full Integrasi (1).
2. False Positive (FP): 40 data yang sebenarnya Partial Integrasi (0) salah diprediksi sebagai Full Integrasi (1).
3. False Negative (FN): 4 data yang sebenarnya Full Integrasi (1) salah diprediksi sebagai Partial Integrasi (0).
4. True Negative (TN): 33 data yang sebenarnya Partial Integrasi (0) berhasil diprediksi dengan benar sebagai Partial Integrasi (0).

Menganalisis metrik evaluasi model, yaitu akurasi, precision, recall, dan F1-score untuk menilai kinerja model klasifikasi. Hasil evaluasi dapat dilihat pada Tabel 6, yang menunjukkan performa model berdasarkan prediksi terhadap kelas Full Integrasi (1) dan Partial Integrasi (0).

Tabel 6. Evaluasi Hasil Klasifikasi

Akurasi	Precision	Recall	F1-Score
69%	62%	94%	75%

Berdasarkan Tabel 6, Evaluasi Hasil Klasifikasi, model *Decision Tree* yang digunakan memiliki akurasi sebesar 69%, dengan nilai precision 62%, recall 94%, dan F1-score 75%.

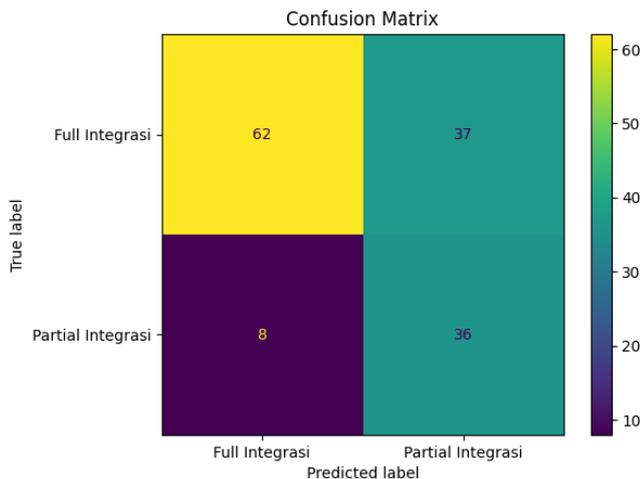
1. Akurasi menunjukkan seberapa sering model memprediksi dengan benar dibandingkan total data uji. Dengan akurasi 69%, model dapat memprediksi secara benar 69% dari total data.

- Precision mengindikasikan bahwa hanya 62% dari prediksi Full Integrasi (1) yang benar, sementara sisanya merupakan kesalahan klasifikasi dari Partial Integrasi (0). Nilai precision yang rendah menunjukkan bahwa model sering salah mengklasifikasikan Partial Integrasi (0) sebagai Full Integrasi (1).
- Recall mengukur seberapa banyak data Full Integrasi (1) yang berhasil dideteksi dengan benar oleh model. Nilai Recall 94% menunjukkan bahwa model sangat baik dalam mengenali kasus Full Integrasi (1).
- F1-Score sebagai harmonik rata-rata antara Precision dan Recall menunjukkan model memiliki keseimbangan yang cukup baik dalam mendeteksi Full Integrasi (1), tetapi masih ada ketidakseimbangan dalam mengenali Partial Integrasi (0).

b. Hasil Hyperparameter Tuning

Setelah melakukan evaluasi terhadap metrik klasifikasi, langkah selanjutnya adalah Hyperparameter Tuning untuk meningkatkan performa model Decision Tree. Hyperparameter tuning bertujuan untuk menyesuaikan parameter model secara optimal agar hasil klasifikasi lebih akurat dan seimbang.

Hyperparameter tuning menggunakan GridSearch dengan 5 folds menghasilkan kombinasi optimal: kriteria = 'gini', kedalaman maksimum = 5, sampel daun minimum = 2, dan sampel split minimum = 2. Dari parameter tersebut, diperoleh hasil confusion matrix yang ditampilkan pada Gambar 8.



Gambar 8. Hasil Confusion Matrix Hyperparameter Tuning

Setelah dilakukan Hyperparameter Tuning, hasil Confusion Matrix pada Gambar 7, menunjukkan perubahan kinerja model. Berikut evaluasi berdasarkan Confusion Matrix tersebut.

- True Positive (TP): 62 data Full Integrasi (1) diprediksi dengan benar sebagai Full Integrasi (1).
- False Positive (FP): 37 data Partial Integrasi (0) salah diklasifikasikan sebagai Full Integrasi (1). Model masih memiliki kecenderungan untuk mengklasifikasikan Partial Integrasi (0) sebagai Full

Integrasi (1) kesalahan klasifikasi masih cukup tinggi.

- False Negative (FN): 8 data Full Integrasi (1) salah diklasifikasikan sebagai Partial Integrasi (0).
- True Negative (TN): 36 data Partial Integrasi (0) diprediksi dengan benar sebagai Partial Integrasi (0). Model mengalami sedikit peningkatan dalam mengenali Partial Integrasi (0) dibandingkan sebelum tuning.

Hasil evaluasi klasifikasi setelah tuning dapat dilihat pada Tabel 7, yang menunjukkan performa model berdasarkan akurasi, precision, recall, dan F1-score.

Tabel 7. Evaluasi Klasifikasi Hyperparameter Tuning

Akurasi	Precision	Recall	F1-Score
69%	62%	89%	73%

Berdasarkan hasil evaluasi setelah tuning pada Tabel 4.5, model mencapai akurasi 69%, dengan precision 62%, recall 89%, dan F1-score 73%. Akurasi tetap stabil, precision tidak berubah signifikan, recall sedikit menurun, namun F1-score menunjukkan keseimbangan antara precision dan recall.

c. Hasil Evaluasi

Hasil evaluasi model Decision Tree sebelum dan setelah dilakukan hyperparameter tuning. Analisis mencakup perbandingan performa model berdasarkan metrik evaluasi, seperti akurasi, precision, recall, dan F1-score, untuk menilai dampak tuning terhadap kinerja model. Perbandingan Confusion Matrix sebelum dan sesudah hyperparameter tuning pada model Decision Tree disajikan pada Tabel 4.6. Evaluasi ini bertujuan untuk memahami perubahan jumlah prediksi benar dan salah pada kelas positif maupun negatif setelah tuning dilakukan.

Tabel 8. Perbandingan Confusion Matrix Sebelum dan Sesudah Hyperparameter Tuning

Metrik	Sebelum Tuning	Setelah Tuning	Perubahan
True Positive (TP)	66	62	Menurun
False Positive (FP)	40	37	Menurun
False Negative (FN)	4	8	Meningkat
True Negative (TN)	33	36	Meningkat

Berikut adalah evaluasi berdasarkan Tabel 8, Confusion Matrix sebelum dan sesudah Hyperparameter Tuning.

- True Positive berkurang dari 66 ke 62, yang berarti 4 data yang sebelumnya dikenali dengan benar kini salah diklasifikasikan sebagai Partial Integrasi (0).
- False Positive berkurang dari 40 ke 37, menunjukkan bahwa model lebih sedikit salah dalam mengklasifikasikan Partial Integrasi sebagai Full Integrasi.
- False Negative meningkat dari 4 ke 8, yang berarti model sedikit lebih sering salah dalam mengenali Full Integrasi dibandingkan sebelumnya.
- True Negative meningkat dari 33 ke 36, menunjukkan bahwa model lebih baik dalam mengenali Partial Integrasi setelah tuning.

Secara keseluruhan, meskipun ada penurunan pada True Positive dan peningkatan pada False Negative, pengurangan False Positive dan peningkatan True Negative menunjukkan bahwa model cenderung lebih akurat dalam mengenali kelas

Partial Integrasi setelah tuning, meskipun ada trade-off dalam pengenalan kelas Full Integrasi.

Perbandingan hasil evaluasi model sebelum dan sesudah hyperparameter tuning dilakukan menggunakan metrik akurasi, precision, recall, dan F1-score. Evaluasi ini bertujuan untuk menilai dampak tuning terhadap performa model. Perubahan yang terjadi, baik peningkatan maupun penurunan pada metrik tertentu, dapat dilihat pada Tabel 9.

Tabel 9. Perbandingan Evaluasi Hasil Klasifikasi Sebelum dan Sesudah Hyperparameter Tuning

Metrik	Sebelum Tuning	Sesudah Tuning	Perubahan
Akurasi	69%	69%	Tetap Sama
Precision	62%	62%	Tetap Sama
Recall	94%	89%	Menurun
F1-Score	75%	73%	Menurun

Berdasarkan Tabel 9, evaluasi hasil klasifikasi sebelum dan sesudah hyperparameter tuning menunjukkan perubahan pada performa model. Berikut penjelasannya mengenai perbedaan akurasi, precision, recall, dan F1-score yang dihasilkan, serta dampaknya terhadap kualitas model.

1. Akurasi model mencapai 69%, menunjukkan kemampuan model dalam melakukan prediksi yang benar. Setelah dilakukan hyperparameter tuning, akurasi tetap stabil tanpa perubahan. Hal ini menunjukkan bahwa perubahan pada parameter model tidak memberikan dampak signifikan terhadap jumlah prediksi yang benar secara keseluruhan.
2. Precision tetap berada di angka 62%, menunjukkan bahwa tuning tidak memberikan perubahan dalam mengurangi false positive. Hal ini mengindikasikan model tetap memiliki kemampuan yang sama dalam memastikan prediksi positif benar adanya.
3. Recall mengalami penurunan, yang berarti model lebih sering gagal mendeteksi kelas positif yang sebenarnya (false negative meningkat). Yang berarti lebih banyak data Full Integrasi salah diklasifikasikan sebagai Partial Integrasi.
4. F1-Score, sebagai gabungan dari precision dan recall, juga mengalami penurunan. Penurunan ini menunjukkan bahwa keseimbangan antara precision dan recall sedikit memburuk, yang kemungkinan besar disebabkan oleh turunya recall.

Hyperparameter tuning yang dilakukan tidak meningkatkan performa model Decision Tree secara keseluruhan. Tidak ada peningkatan pada akurasi dan precision, sementara recall dan F1-score justru mengalami penurunan. Perubahan parameter yang diterapkan kemungkinan mengurangi sensitivitas model terhadap kelas positif, menyebabkan penurunan recall.

V. KESIMPULAN

Berdasarkan hasil implementasi dan evaluasi model Decision Tree untuk memprediksi status integrasi SPBU di Wilayah Telkom Makassar, dapat disimpulkan bahwa metode ini berhasil diimplementasikan, melibatkan proses penyeimbangan data SMOTE dan hyperparameter tuning. Namun, model awal dengan akurasi 69% tidak mengalami peningkatan signifikan setelah tuning.

Meskipun akurasi tetap stabil di 69%, terjadi penurunan pada metrik recall dan F1-score. Hal ini mengindikasikan bahwa tuning yang dilakukan belum mampu mengoptimalkan performa model secara signifikan, dan eksplorasi lebih lanjut terhadap metode atau parameter lain mungkin diperlukan untuk mencapai hasil prediksi yang baik.

REFERENSI

- [1] Timoty Agustian Berutu, Dina Lorena Rea Sigalingging, Gaby Kasih Valentine Simanjuntak, and Friska Siburian, "Pengaruh Teknologi Digital terhadap Perkembangan Bisnis Modern," *Neptunus: Jurnal Ilmu Komputer Dan Teknologi Informasi*, vol. 2, no. 3, pp. 358–370, Jul. 2024, doi: 10.61132/neptunus.v2i3.258.
- [2] E. Indrayani, *Transformasi Digital untuk Masa Depan Pelayanan Publik di Indonesia*. Bandung: CV Cendekia Press, 2024.
- [3] A. I. Hidayat, A. Agunawan, Y. Mahendra, and W. Cahyani, "Penerapan IoT pada Sistem Deteksi Kadar Air dan Level Tangki Stasiun SPBU," *Jurnal Teknik*, vol. 21, no. 2, pp. 142–154, Dec. 2023, doi: 10.37031/jt.v21i2.421.
- [4] M. Syarifuddin, J. Simarmata, A. Maulidinnawati Abdul Kadir Parewe, I. Firman Ashari, and S. Ramadhani Arifin, *Cloud Computing: Konsep dan Implementasi*. Yayasan Kita Menulis, 2023. [Online]. Available: <https://www.researchgate.net/publication/370815388>
- [5] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, Mar. 2021, doi: 10.38094/jastt20165.
- [6] Roland Fiagbe, "Classification of Adult Income Using Decision Tree," *Data Science and Data Mining*, pp. 2–5, 2023.
- [7] Y. Yang, "The Evaluation of Online Education Course Performance Using Decision Tree Mining Algorithm," *Complexity*, pp. 1–13, Apr. 2021, doi: 10.1155/2021/5519647.
- [8] Emil Agbemade, "Predicting Heart Disease using Tree-based Model," *Data Science and Data Mining*, 2023.
- [9] Achmad Nuruddin Syaifulloh, Nur Iriawan, and Pratnya Paramitha Oktaviana, "ANALYSIS OF THE PATTERN DISTRIBUTION OF SURABAYA REGIONAL PUBLIC FUEL STATIONS (SPBU) USING SPATIAL POISSON POINT PROCESS," *JURNAL SAINS DAN SENI ITS*, vol. 8, no. 2, 2019.
- [10] S. Oleh and V. Berlinda, "PERJANJIAN KERJASAMA PENGELOLAAN USAHA SPBU DEALER OWNED DEALER OPERATED (DODO) DI INDONESIA (Studi pada Stasiun Pengisian Bahan Bakar Umum (SPBU) 24.345.28 di Kabupaten Tulang Bawang)," 2018.
- [11] SUHARNO, "PENGOPERASIAN PERALATAN SPBU," 2020.
- [12] administrator, "Pengertian Nozzle : Fungsi, Jenis, Komponen, Cara Kerja dan Contoh," Empat Pilar.

- Accessed: Feb. 11, 2025. [Online]. Available: <https://www.empatpilar.com/pengertian-nozzle/>
- [13] A.- Fauzi, A. Fathurohman, and T. N. Pratyaksa, "Manage Sistem Operasi Digitalisasi SPBU Pertamina Daerah Operasional PT. Telkom Indonesia Regional IV Witel Semarang," *JURNAL KOMPUTER DAN TEKNOLOGI INFORMASI*, vol. 1, no. 2, Jul. 2023, doi: 10.26714/jkti.v1i2.12919.
- [14] A. N. Aliansyah *et al.*, "Simulasi Sistem Monitoring Tangki Pendam yang Terintegrasi dengan Mesin Dispenser SPBU Berbasis PLC," vol. 07, pp. 01–06, 2022, [Online]. Available: <http://ojs.uho.ac.id/index.php/jfe/>
- [15] I. S. Damanik, A. P. Windarto, A. Wanto, Poningsih, S. R. Andani, and W. Saputra, "Decision Tree Optimization in C4.5 Algorithm Using Genetic Algorithm," *J Phys Conf Ser*, vol. 1255, no. 1, pp. 1–7, Aug. 2019, doi: 10.1088/1742-6596/1255/1/012012.
- [16] J. Mrva, S. Neupauer, L. Hudec, J. Sevcech, and P. Kapec, "Decision Support in Medical Data Using 3D Decision Tree Visualisation," in *2019 E-Health and Bioengineering Conference (EHB)*, IEEE, Nov. 2019, pp. 1–4. doi: 10.1109/EHB47216.2019.8969926.
- [17] Y. Setiawan, O. A. Permata, and M. P. Yuda, "Decision Tree based Data Modelling for First Detection of Thalassemia Major," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 13, no. 1, pp. 49–56, Feb. 2024, doi: 10.32736/sisfokom.v13i1.1949.
- [18] E. Muchai and L. Odongo, "Comparison of Crisp and Fuzzy Classification Trees Using Gini Index Impurity Measure on Simulated Data," *Eur Sci J*, vol. 10, no. 18, pp. 130–134, Jun. 2014.
- [19] K. Ujaran, K. Ridwan, E. Heni Hermaliani, M. Ernawati, and C. Author, "Penerapan Metode SMOTE Untuk Mengatasi Imbalanced Data Pada," *Computer Science (CO-SCIENCE)*, vol. 4, no. 1, Jan. 2024.
- [20] A. Ghina Nurfanseptra, L. Muflikhah, and B. D. Setiawan, "Deteksi Mutasi Epidermal Growth Factor Receptor pada Kanker Paru Menggunakan Extreme Gradient Boosting," *Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 4, pp. 2548–964, Apr. 2025, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [21] A. Berbasis *et al.*, "Web-Based Application for Data Analysis Using Pearson Bivariate Correlation," vol. 18, no. 1, pp. 39–50, Apr. 2021.
- [22] P. Putu Narisya Ardhaneswari and I. Wayan Candra Suwitra, "Pearson Correlation Analysis In Determining The Relationship Between Price And Sales Volume Of Wardah Matte Lip Cream On The Shopee E-Commerce Platform Analisis Korelasi Pearson Dalam Menentukan Hubungan Harga dengan Volume Penjualan Wardah Matte Lip Cream Pada Platform E-Commerce Shopee," vol. 1, no. 2, pp. 85–92, Mar. 2025.
- [23] Mahmoud M., "Confusion Matrix in Binary Classification Problems: A Step-by-Step Tutorial," *Journal of Engineering Research (ERJ)*, vol. 6, no. 5, pp. 1–12, 2022.
- [24] A. Tasnim, Md. Saiduzzaman, M. A. Rahman, J. Akhter, and A. S. Md. M. Rahaman, "Performance Evaluation of Multiple Classifiers for Predicting Fake News," *Journal of Computer and Communications*, vol. 10, no. 09, pp. 1–21, 2022, doi: 10.4236/jcc.2022.109001.
- [25] Balaram Panda, "A survey on application of Population Based Algorithm on Hyperparameter Selection," 2019.
- [26] Md Riyad Hossain, Douglas Timmer, and Hiram Moya, "Machine learning model optimization with hyper-parameter tuning approach," *Conference Paper*, p. 7, Aug. 2021.