

Integrasi Sistem Cerdas Berbasis AI untuk Penyaluran Bantuan Langsung Tunai yang Tepat Sasaran

Wresti Andriani^{*1}, Naella Nabila Putri Wahyuning Naja²

^{*1}, Informatika, Universitas Muhammadiyah, Tegal

³Manajemen, Universitas Negeri Semarang, Semarang

e-mail: ¹wresty.andriani@gmail.com, ²naellanabilap@student.unnes.ac.id

Abstrak

Penelitian ini mengembangkan sistem cerdas berbasis AI untuk penyaluran Bantuan Langsung Tunai (BLT) yang tepat sasaran menggunakan algoritma Decision Tree, K-Nearest Neighbors (KNN), dan Naive Bayes. Evaluasi awal menunjukkan akurasi rata-rata model berada di bawah 50%, dengan AUC terbaik sebesar 0.47 pada Naive Bayes. Setelah optimasi menggunakan Particle Swarm Optimization (PSO), algoritma KNN menunjukkan peningkatan terbaik dengan AUC sebesar 0.51, sementara Decision Tree mencapai AUC sebesar 0.49. Sistem ini memanfaatkan data seperti penghasilan, kondisi kesehatan, dan status tempat tinggal untuk menentukan kelayakan penerima BLT. Penelitian ini membuktikan bahwa penggunaan metode AI dengan optimasi mampu meningkatkan efisiensi dan akurasi dalam mendistribusikan BLT secara lebih tepat sasaran, memberikan kontribusi signifikan pada perbaikan sistem bantuan sosial.

Kata Kunci: Sistem Cerdas, Bantuan Langsung Tunai, Particle Swarm Optimization.

Abstract

This study developed an AI-based intelligent system for the accurate distribution of Cash Transfer Program (BLT) using Decision Tree, K-Nearest Neighbors (KNN), and Naive Bayes algorithms. Initial evaluations showed that the average accuracy of the models was below 50%, with Naive Bayes achieving the highest AUC of 0.47. After optimization using Particle Swarm Optimization (PSO), the KNN algorithm demonstrated the best improvement with an AUC of 0.51, while the Decision Tree reached an AUC of 0.49. The system utilizes data such as income, health conditions, and housing status to determine the eligibility of BLT recipients. This study demonstrates that the application of AI methods with optimization can enhance efficiency and accuracy in distributing BLT, providing a significant contribution to improving social assistance systems.

Keywords: Intelligent System, Cash Transfer Program, Particle Swarm Optimization.

I. PENDAHULUAN

Bantuan Langsung Tunai (BLT) adalah salah satu program pemerintah yang bertujuan untuk memberikan bantuan ekonomi langsung kepada masyarakat miskin dan rentan (Adi et al., 2023). Program ini menjadi sangat relevan di masa krisis, seperti pandemi COVID-19, di mana perekonomian masyarakat terguncang (Aldino et al., 2022). BLT memiliki sejumlah keunggulan yang menjadikannya pilihan yang menarik untuk mencapai tujuan sosial-ekonomi ini (Waraningtyas, 2022). Pertama, BLT memungkinkan penerima untuk memiliki kontrol yang lebih besar atas penggunaan bantuan tersebut (Istriawati & Dartanto, 2022). Kedua, BLT memiliki potensi untuk merangsang pertumbuhan ekonomi lokal dengan meningkatkan daya beli penerima (Adi et al., 2023). Ketiga, BLT dapat menjadi respons

yang cepat dan efektif dalam situasi krisis, seperti bencana alam atau pandemi (Malika et al., 2024).

Namun, penyaluran BLT sering menghadapi tantangan, seperti data penerima yang tidak akurat (Wibawani et al., 2021), ketidaktepatan sasaran, serta potensi fraud dalam distribusi serta kurangnya mekanisme seleksi yang berbasis data untuk menentukan prioritas penerima bantuan (Sulaimah et al., 2022) dan adanya data penerima manfaat yang tidak valid atau tumpang tindih (Prasetyo & Kistanti, 2020). Pada situasi ini, adopsi teknologi berbasis kecerdasan buatan (*Artificial Intelligence/AI*) dapat menjadi solusi untuk mengoptimalkan proses penyaluran bantuan agar lebih efisien dan tepat sasaran (Putra et al., 2024).

Beberapa penelitian sebelumnya yang menggunakan sistem cerdas berbasis AI untuk

mengatasi masalah dalam penyaluran dana (Hamonangan et al., 2024) menggunakan mendapatkan akurasi mencapai 71.56%, nilai *recall true* tidak layak sebesar 68.43%, *recall true* layak sebesar 74.80%, *precision* layak sebesar 66.79%, dan *precision* tidak layak sebesar 77.54% dianggap dapat menganalisis dan memvalidasi data penerima bantuan berdasarkan kriteria yang ditetapkan, serta pada penelitian (Kurniadi et al., 2023) menggunakan metode *Naïve Bayes*, menghasilkan nilai akurasi 77,60% model yang dihasilkan dapat diimplementasikan ke dalam sistem aplikasi pendukung keputusan. *KNN* memiliki kelemahan yaitu biaya komputasi yang tinggi pada dataset besar, sensitivitas terhadap dimensionalitas tinggi, dan rentan terhadap outlier serta data tidak seimbang. Selain itu, *KNN* memerlukan normalisasi data dan pemilihan parameter *k* yang optimal untuk kinerja terbaik. *Naïve bayes* juga memiliki kelemahan adalah asumsi independensi antar fitur yang jarang terpenuhi, sehingga dapat menghasilkan prediksi kurang akurat jika fitur saling bergantung (Gunawan et al., 2023). Selain itu, metode ini sensitif terhadap data tidak seimbang dan dapat memberikan hasil buruk pada dataset dengan distribusi yang kompleks atau non-linear.

Penelitian ini akan membandingkan metode *Decision Tree*, *KNN* dan *Naive Bayes* dan dioptimasi menggunakan *Partikel Swam Optimasi* dalam menganalisa bantuan langsung tunai yang tepat sasaran, agar dapat menghasilkan akurasi yang lebih optimal untuk mendukung penyaluran BLT yang lebih tepat sasaran, mengevaluasi kinerja sistem cerdas dalam menganalisis data dan memprioritaskan penerima manfaat dan memberikan rekomendasi praktis bagi pemerintah untuk implementasi sistem berbasis AI dalam program BLT, diharapkan memberikan manfaat, bagi pemerintah membantu dalam membuat kebijakan distribusi bantuan sosial yang lebih efisien dan transparan, meningkatkan keadilan dan ketepatan sasaran dalam penyaluran BLT dan menambah referensi terkait penerapan AI dalam program bantuan sosial. Sehingga pendekatan berbasis AI ini, diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan efektivitas dan efisiensi program BLT, sekaligus memperkuat kepercayaan masyarakat terhadap program pemerintah.

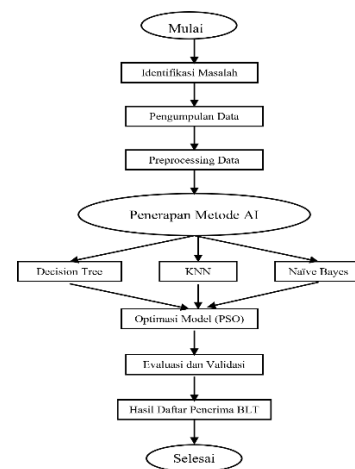
II. METODE PENELITIAN

Penelitian ini merupakan penelitian kuantitatif dengan pendekatan *eksperimental*, dimana algoritma pembelajaran mesin digunakan untuk memprediksi kelayakan penerima Bantuan Langsung Tunai (BLT). Penelitian ini membandingkan kinerja metode *Decision Tree*, *KNN* dan *Naïve Bayes* kemudian dioptimasi menggunakan *Partikel Swam Optimization* untuk mendapatkan akurasi, *precision*, *recall*, *F1 score*, dan *AUC-ROC* yang lebih optimal.

Penelitian dilakukan meliputi tahap pengumpulan data, preprocessing, pengembangan model, hingga analisis hasil kemudian dioptimasi menggunakan algoritma *PSO*, dengan menggunakan data yang diperoleh secara online melalui laman <https://ppid.tegalkota.go.id/cek-bansos-tranparansi-bantuan-keuangan> diperoleh data sebanyak 300 data. Data tersebut memiliki fitur nama penduduk, usia, pekerjaan, jumlah anggota keluarga, penghasilan bulanan (Rp), status penerima bantuan (ya atau tidak), status rumah, pendidikan akhir dan kondisi kesehatan (baik, sakit menahun, atau disabilitas).

Diagram Alur

Individu atau rumah tangga yang berpotensi menjadi penerima BLT berdasarkan kriteria seperti pendapatan rendah, jumlah tanggungan tinggi, dan status pekerjaan tidak tetap, menjadi target ini sedangkan sasarannya yaitu memastikan BLT disalurkan secara tepat kepada individu yang paling membutuhkan. Diagram alurnya seperti pada gambar 1.



Gambar 1. Diagram alur Penelitian.

Gambar 1 menunjukkan alur proses integrasi sistem AI menggunakan metode *Decision Tree*, *KNN* dan *Naïve Bayes* dan dioptimasi menggunakan *Partikel Swam Optimization* hingga menghasilkan output yang diinginkan, yaitu daftar penerima BLT yang valid.

Proses dimulai dengan mengidentifikasi kebutuhan untuk sistem AI yang mampu menyalurkan BLT secara tepat sasaran. Identifikasi masalah, pada tahap ini, masalah utama seperti inefisiensi dan ketidaktepatan penyaluran BLT diidentifikasi.

Pengumpulan Data

Data yang relevan seperti data sosial, ekonomi, dan demografi dari calon penerima BLT, data ini diperoleh secara *online* melalui laman <https://ppid.tegalkota.go.id/cek-bansos-tranparansi-bantuan-keuangan> yaitu daftar calon penerima Bantuan Langsung Tunai Kota Tegal sebanyak 300 data. Data ini mempunyai variabel fitur nama penduduk, usia, pekerjaan, jumlah anggota keluarga, penghasilan bulanan (Rp), status penerima bantuan (ya atau tidak), status rumah, pendidikan akhir dan kondisi kesehatan (baik, sakit menahun, atau disabilitas), dalam bentuk excel.

Preprocessing Data

Preprocessing data yang akan dilakukan pada penelitian ini meliputi *cleansing*, yaitu menghilangkan data yang tidak lengkap atau tidak *relevan* (Gifari et al., 2022), normalisasi yaitu menstandarkan data agar siap digunakan dalam algoritma AI (Ahmadi et al., n.d.) dan transformasi yaitu mengubah format data sesuai kebutuhan analisis (Ichsan et al., 2024).

Pada penelitian ini perubahan yang dilakukan pada proses transformasi data adalah mengubah data nominal menjadi *numeric*, seperti pada Tabel 1.

Tabel 1. Transformasi data

Kategori Pekerjaan		Pendidikan Terakhir	
Pekerjaan	Kategori	Pendidikan	Kategori
Tidak bekerja	1	SD	1
Petani	2	SMP	2
Nelayan	3	SMA	3
Buruh Pabrik	4	D3/S1	4
Pedagang	5	Tidak sekolah	5
Kepemilikan Rumah		Kondisi Kesehatan	

Kepemilikan	Kategori	Kondisi	Kategori
Sewa	1	Baik	1
Menumpang	2	Sakit Menahun	2
Milik Pribadi	3	Disabilitas	3
Penerima Bantuan sebelumnya		Pendapatan	
Status	Kategori	Pendapatan	Kategori
Ya	1	>1.000.000	1
Tidak	2	1.000.000 – 3.000.000	2
		< 3.000.000	3

Kriteria kelayakan ditentukan adalah Pendapatan di bawah batas tertentu <3,000,000, Status pekerjaan seperti tidak bekerja, kondisi kesehatan "Sakit Menahun", serta status tempat tinggal yang memprioritaskan "Menumpang".

Penerapan metode AI

Beberapa algoritma kecerdasan buatan diterapkan untuk analisis, yaitu *Decision Tree* dengan membuat pohon keputusan untuk mengelompokkan penerima yang layak.

KNN (K-Nearest Neighbor) yaitu mengklasifikasikan penerima berdasarkan kemiripan data dengan penerima lain yaitu dengan menghitung Jarak (*Distance*), dalam hal ini menggunakan *Euclidean Distance* (Arinal & Sentosa, 2022), seperti pada persamaan 1.

$$d(x, y) = \sqrt{\sum_{i=1}^n (X_i - y_i)^2} \quad (1)$$

Untuk setiap data baru x , hitung jarak ke semua data. Pilih k untu tetangga terdekat, kemudian urutkan data berdasarkan jarak terdekat dan pilih k data dengan jarak terpendek ke data baru. Untuk *voting* pada klasifikasi, dengan cara menghitung jumlah kemunculan dari setiap kelas dalam k tetangga dan memilih kelas dengan jumlah terbanyak sebagai prediksi.

Naïve Bayes metode yang menggunakan probabilitas untuk menentukan kelayakan penerima. *Naive Bayes* adalah algoritma berbasis probabilitas yang digunakan untuk klasifikasi (Pebdika et al., 2023). Prinsip dasarnya adalah menghitung *probabilitas posterior* dari setiap kelas menggunakan teorema *bayes* dengan asumsi bahwa fitur bersifat independen, seperti pada persamaan 2.

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \quad (2)$$

Di mana: $P(C|X)$ adalah *probabilitas* kelas, C diberikan data X (*probabilitas posterior*), $P(X|C)$ adalah *probabilitas* data X diberikan kelas C (*probabilitas likelihood*), $P(C)$ adalah *probabilitas apriori* kelas C , $P(X)$ adalah *probabilitas* data X (*konstanta* untuk semua kelas, dapat diabaikan dalam perhitungan relatif).

Support Vector Machine (SVM) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi (Rahayu et al., 2023). Tujuannya adalah menemukan *hyperplane optimal* yang memisahkan data ke dalam kelas-kelas dengan margin terbesar. Persamaan tersebut terdiri dari:

1. Persamaan *hyperplane* yaitu sebuah *hyperplane* dalam ruang n dimensi dinyatakan seperti persamaan 3.

$$w \cdot x + b = 0 \quad (3)$$

dimana, w = vektor bobot (weight vektor), x = vektor data input dan b = bias (intersep).

2. *SVM* mencari *hyperplane* yang memaksimalkan *margin*, yaitu jarak terdekat dari titik data ke *hyperplane*. Untuk linear *SVM*, persamaan objektifnya seperti pada persamaan 4 dengan meminimalkan,

$$\frac{1}{2} \|w\|^2 \quad (4)$$

dengan syarat, pada persamaan 5.

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i \quad (5)$$

Dimana y_i adalah label kelas (+1 atau -1), x_i adalah data sampel.

3. *Soft Margin (SVM Nonlinear)* dimana untuk mengatasi data yang tidak sepenuhnya terpisah *linear*, tambahkan variabel *slack* ξ_i , seperti persamaan 6.

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (6)$$

Dengan syarat: $y_i(w \cdot x_i + b) \geq 1 - \xi_i \geq 0$, C adalah parameter regulasi yang mengontrol *trade-off*

antara margin lebar dan penalti terhadap salah klasifikasi.

Random Forest adalah algoritma pembelajaran *ensemble* yang menggunakan banyak pohon keputusan (*Decision Trees*) untuk membuat prediksi. Hasil akhirnya diperoleh dengan menggabungkan prediksi dari setiap pohon melalui *voting* (untuk klasifikasi) (Khaidar et al., 2024). Pohon keputusan dibangun menggunakan subset data yang dipilih secara acak (*bootstrap sampling*) dan subset fitur yang dipilih secara acak. Setiap pohon memberikan prediksi kelas untuk sampel input x . Prediksi akhir diperoleh dari *majoritas voting*, seperti persamaan 7.

$$\hat{y} = \text{mode}(h_1(x), h_2(x), \dots, h_m(x)) \quad (7)$$

Dimana, $h_1(x)$ adalah prediksi dari pohon ke I dan m adalah jumlah pohon dalam hutan.

Setiap *node* dalam pohon keputusan dibagi berdasar kan fungsi *impuritas* seperti *Gini Index*, seperti persamaan 8.

$$Gini = 1 - \sum_{i=1}^n p_i^2 \quad (8)$$

Di mana p_i adalah proporsi sampel dalam kelas i di *node* tersebut.

Particle Swarm Optimization digunakan untuk meningkatkan kinerja model dengan menemukan parameter optimal (Putry et al., 2024). *Particle Swarm Optimization (PSO)* adalah algoritma optimasi berbasis populasi yang terinspirasi oleh perilaku kawanan (*swarm*) seperti burung atau ikan. Algoritma ini bertujuan untuk menemukan solusi optimal dengan memanfaatkan interaksi antara partikel dalam ruang pencarian, seperti persamaan 9.

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \cdot iter \quad (9)$$

Dimana nilai w menentukan pengaruh kecepatan sebelumnya. Biasanya, w dimulai dengan nilai tinggi dan berkurang seiring iterasi.

Hasil dari model dievaluasi menggunakan metrik seperti akurasi, *precision*, *recall*, *F1 score*, dan *AUC-ROC* untuk memastikan keandalan. Akurasi adalah metrik evaluasi yang mengukur proporsi prediksi yang benar terhadap total jumlah prediksi, seperti pada persamaan 10.

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi benar}}{\text{Total prediksi}} \quad (10)$$

Presisi (*Precision*) adalah metrik yang mengukur proporsi prediksi positif yang benar terhadap semua prediksi positif, seperti persamaan 11.

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (11)$$

Dimana, TP adalah jumlah prediksi benar untuk kelas positif dan Fp adalah jumlah prediksi untuk kelas positif.

Recall (Sensitivitas atau True Positive Rate) adalah metrik yang mengukur proporsi prediksi positif yang benar terhadap semua kasus sebenarnya yang positif, seperti persamaan 12.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (12)$$

F1 Score adalah metrik evaluasi yang merupakan rata-rata harmonik dari Presisi dan *Recall*. *F1 Score* digunakan untuk mencapai keseimbangan antara keduanya, terutama pada dataset yang tidak seimbang, seperti pada persamaan 13.

$$F1 \text{ Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (13)$$

F1 Score penting dalam kasus di mana terdapat *trade-off* antara Presisi dan *Recall*. Nilai *F1 Score* berkisar antara 0 dan 1, dimana *F1 Score* = 1, model sempurna (Presisi dan *Recall* = 1) dan *F1 Score* = 0, model tidak memprediksi dengan benar.

III. HASIL DAN PEMBAHASAN

Penelitian yang dilakukan ini menggunakan aplikasi phyton, seperti pada Gambar 2.

```
# Compute metrics again, handling ROC AUC manually
for multi-class
from sklearn.metrics import roc_auc_score.
results = []
for name, model in models.items():
    preds = model.predict(X_test)
    if hasattr(model, "predict_proba"):
        probs = model.predict_proba(X_test)
        roc_auc = roc_auc_score(y_test, probs[:,
1]) if probs.shape[1] > 1 else None
    else:
        roc_auc = None # If predict_proba is not
available
    accuracy = accuracy_score(y_test, preds)
```

```
precision = precision_score (y_test, preds,
average='weighted')
recall = recall_score (y_test, preds,
average='weighted')
f1 = f1_score (y_test, preds,
average='weighted')
results.append ({
    "Model": name,
    "Accuracy": accuracy,
    "Precision": precision,
    "Recall": recall,
    "F1 Score": f1,
    "ROC AUC": roc_auc
})

# Create a DataFrame to display the results
results_df = pd.DataFrame(results)

import ace_tools as tools, tools.
display_dataframe_to_user (name="Comparison of KNN,
Decision Tree, and Naive Bayes Before
Optimization", dataframe=results_df)
```

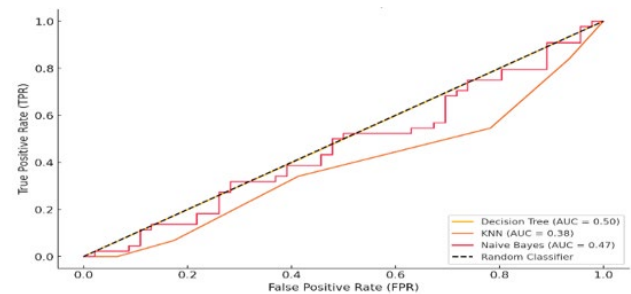
Gambar 2. Aplikasi DT, KNN dan NB

Hasil matrisk evaluasi yang dihasilkan seperti pada Tabel 2.

Tabel 2. Hasil evaluasi DT, KNN dan NB

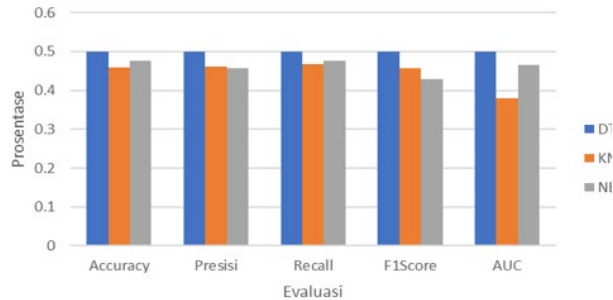
Model	Accuracy	Presisi	Recall	F1Score	AUC
DT	0.5	0.499	0.5	0.499	0.499
KNN	0.46	0.462	0.467	0.458	0.381
NB	0.477	0.457	0.477	0.429	0.465

Selain itu juga menghasilkan gambar grafik ROC seperti Gambar 3.



Gambar 3. Grafik ROC

Hasil evaluasi dari ketiga metode tersebut divisualisasi menggunakan grafik batang seperti pada Gambar 4.



Gambar 4. Grafik perbandingan evaluasi DT, KNN dan NB

Bila ketiga metode ini dioptimasi menggunakan *Optimasi Partikel Swam* dengan menggunakan aplikasi phyton seperti Gambar 5.

```
# Re-train models with default settings to ensure
proper evaluation
optimized_dt = DecisionTreeClassifier (max_depth=5,
min_samples_split=10, random_state=42)
optimized_knn =
KNeighborsClassifier(n_neighbors=5)
optimized_nb = GaussianNB(var_smoothing=1e-9)
optimized_dt.fit (X_train, y_train)
optimized_knn.fit (X_train, y_train)
optimized_nb.fit (X_train, y_train)
# Compute confusion matrices for re-trained models
models_optimized_retrained = {
    "Decision Tree (Re-trained)": optimized_dt,
    "KNN (Re-trained)": optimized_knn,
    "Naive Bayes (Re-trained)": optimized_nb
}
confusion_matrices = {}
for name, model in models_optimized_retrained.
items ():
    preds = model. predict(X_test)
    cm = confusion_matrix (y_test, preds)
    confusion_matrices[name] = cm
# Create a DataFrame to display confusion matrices
confusion_matrix_df = pd. DataFrame (
    {
        "Model": list (confusion_matrices. keys
()),
        "Confusion Matrix": list
(confusion_matrices. values ()),
    }
)
import ace_tools as tools; tools.
display_dataframe_to_user (name="Confusion
Matrices for Re-trained Models",
dataframe=confusion_matrix_df)
```

Gambar 5. Aplikasi phyton *Optimasi Partikel Swam*

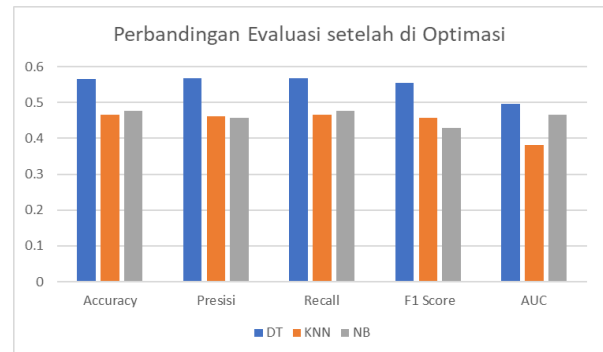
Hasil aplikasi phyton untuk *Optimasi Partikel Swam* untuk metode *Decision Tree*, *KNN* dan *Naïve Bayes* dapat dilihat pada tabel 3.

Tabel 3. Hasil Optimasi PSO dari DT, KNN, NB

Model	Accuracy	Presisi	Recall	F1 Score	AUC
-------	----------	---------	--------	----------	-----

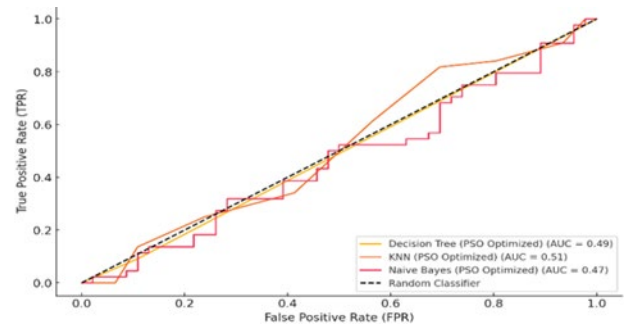
DT	0.566	0.569	0.567	0.556	0.496
KNN	0.467	0.462	0.467	0.458	0.382
NB	0.478	0.457	0.477	0.429	0.465

Grafik batang dari perbandingan ketiga metode setelah dioptimasi seperti pada Gambar 6.



Gambar 6. Hasil PSO dari DT, KNN, NB

Sedangkan gambar grafik ROC seperti pada Gambar 7.



Gambar 7. Grafik ROC perbandingan PSO model

Dari gambar 4 dan 6, terlihat hasil performa model sebelum dan setelah optimasi menggunakan *Particle Swarm Optimization (PSO)*, yaitu untuk metode *Decision Tree (DT)*, *accuracy* meningkat dari 0.5 menjadi 0.566 terjadi peningkatan sebesar (+0.066), *Presisi* meningkat dari 0.499 menjadi 0.569 artinya ada peningkatan sebesar (+0.07), *Recall* meningkat dari 0.5 menjadi 0.567 ada peningkatan sebesar (+0.067), *F1 Score* meningkat dari 0.499 menjadi 0.556 ada peningkatan (+0.057) dan *AUC* sedikit meningkat dari 0.499 menjadi 0.496 ada peningkatan sebesar (+0.003). Jadi, optimasi PSO meningkatkan seluruh metrik untuk model DT, menunjukkan bahwa *PSO* berhasil dalam

menyesuaikan parameter model untuk performa yang lebih baik.

Metode *K-Nearest Neighbors (KNN)*, menghasilkan *accuracy* yang tetap sama di 0.467, *Presisi* hasilnya juga tidak berubah, tetap di 0.462, *Recall* hasilnya pun tidak berubah, tetap di 0.467, begitu juga pada *F1 Score* juga tidak berubah, tetap di 0.458 dan *AUC* yang hasilnya, tetap di 0.382.

Sementara untuk metode *Naive Bayes (NB)* dengan hasil *accuracy* yang meningkat sedikit dari 0.477 menjadi 0.478 (+0.001), *Presisi* pun tidak berubah, tetap di 0.457, begitu pula untuk *Recall*, yang hasilnya, tetap di 0.477 dan *F1 Score* hasilnya juga tidak berubah, tetap di 0.429 serta *AUC*: yang juga tetap di 0.465.

Aplikasi untuk mengetahui daftar penerima BLT seperti pada Gambar 8.

```
# Filter the dataset based on the eligibility
criteria for BLT recipients
# Criteria:
# - Penghasilan Bulanan < 3,000,000
# - Kepemilikan Rumah is "Menumpang" or "Sewa"
# - Kondisi Kesehatan is "Sakit Menahun" or
"Disabilitas"

eligible_recipients = sheet_data [
(sheet_data ['Penghasilan Bulanan (Rp)'] <
3000000) &
(sheet_data ['Kepemilikan Rumah']. isin
(['Menumpang', 'Sewa'])) &
(sheet_data ['Kondisi Kesehatan']. isin
(['Sakit Menahun', 'Disabilitas']))
]

# Display the results to the user
import ace_tools as tools, tools.
display_dataframe_to_user (name="Daftar Penerima
BLT", dataframe=eligible_recipients)
```

Gambar 8. Aplikasi daftar penerima BLT

Hasil perhitungan dalam pencarian penerima bantuan langsung tunai dari 300 orang, diperoleh sebanyak 81 orang dengan perincian diantaranya, seperti pada tabel 4

Tabel 4. Daftar nama Penerima BLT

N0.	Nama Penerima
4	Budi Utama
5	Gilang Santoso
7	Dewi Wijaya
8	Ivan Utama
12	Hana Utama

298	Gilang Rahma

Pada tabel 4, terlihat sampel dari penerima BLT.

IV. KESIMPULAN

Sistem penyaluran Bantuan Langsung Tunai (BLT) sering menghadapi tantangan dalam menargetkan penerima yang benar-benar membutuhkan. Hal ini disebabkan oleh data yang tidak terstruktur dan proses seleksi manual yang kurang efisien. Sebelum dioptimasi, algoritma *Decision Tree*, *KNN*, dan *Naive Bayes* menunjukkan performa yang kurang memuaskan dengan akurasi rata-rata di bawah 0.50. Optimasi *PSO* menghasilkan peningkatan pada algoritma *KNN* dengan *AUC* sebesar 0.51, yang menjadi performa terbaik di antara model lainnya setelah optimasi. *Decision Tree* menunjukkan *AUC* sebesar 0.49 setelah optimasi *PSO*. Kurva *ROC* dari model yang telah dioptimasi menunjukkan bahwa *Decision Tree* dan *KNN* adalah algoritma terbaik dalam memprediksi penerima BLT yang layak, khususnya setelah optimasi *PSO*.

V. REFERENSI

- Adi, M., Putra, P., Putra, A. P., Dirli, A. A., Andriani, W., & Mauldyraharja, R. (2023). Menggali Efektivitas Bantuan Langsung Tunai dalam Membantu Masyarakat. *Concept: Journal of Social Humanities and Education*, 2(4), 189–205.
- Ahmadi, T., Wulandari, A., & Suhatman, H. (n.d.). Sistem Customer Churn Prediction Menggunakan Machine Learning pada Perusahaan ISP.
- Aldino, A. A., Suryono, R. R., & Ambarwati, R. (2022). Analysis of Covid-19 Cash Direct Aid (BLT) Acceptance Using K-Nearest Neighbor Algorithm. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 16(2), 193. <https://doi.org/10.22146/ijccs.70801>
- Arinal, V., & Sentosa, E. (2022). Klasifikasi Tingkat Kesejahteraan RW 006 Kelurahan Kalideres Jakarta Barat dengan Metode K-Nearest Neighbor. *Jurnal Pendidikan Dan Konseling (JPDK)*, 4(4), 5621–5638.
- Gifari, O. I., Adha, M., Freddy, F., & Durrand, F. F. S. (2022). Film Review Sentiment Analysis Using TF-IDF and Support Vector Machine. *Journal of Information Technology*, 2(1), 36–

- 40.
- Gunawan, G., Andriani, W., & Santoso, A. A. (2023). Application of the Naïve Bayes and K-Nearest Neighbor Methods for Classifying Roses. *Jurnal Mantik*, 7(2), 799–809.
- Hamonangan, R., Sari, R. K., Anwar, S., & Hartati, T. (2024). Klasifikasi Algoritma KNN dalam menentukan Penerima Bantuan Langsung Tunai. 6(1), 198–204.
- Ichsan, M., Hasnah, R., Faiz, M., & Musi, S. (2024). Analisis Konvergensi Media: Studi Transformasi Dari Media Analog ke Media Digital. *Triwikrama: Jurnal Ilmu Sosial*, 4(4), 19–30.
<http://ejournal.warunayama.org/index.php/triwikrama/article/view/4422%0Ahttp://ejournal.warunayama.org/index.php/triwikrama/article/download/4422/4097>
- Istriawati, N., & Dartanto, T. (2022). Dampak Bantuan Pangan Non Tunai terhadap Konsumsi Makanan dan Rokok pada Rumah Tangga Miskin di Indonesia. *Journal of Education, Humaniora and Social Sciences (JEHSS)*, 5(2), 1158–1172.
<https://doi.org/10.34007/jehss.v5i2.1407>
- Khaidar, A., Arhami, M., & Abdi, M. (2024). Application of the Random Forest Method for UKT Classification at Politeknik Negeri Lhokseumawe. 4(2), 94–103.
<https://doi.org/10.30811/jaise.v4i2.6131>
- Kurniadi, D., Nuraeni, F., & Firmansyah, M. (2023). Klasifikasi Masyarakat Penerima Bantuan Langsung Tunai Dana Desa Menggunakan Naïve Bayes dan SMOTE. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(2), 309–320.
<https://doi.org/10.25126/jtiik.20231026453>
- Malika, M., Hsb, S. A., Studi, P., Sosial, K., & Utara, U. S. (2024). Peran Bantuan Langsung Tunai (BLT) dalam Meningkatkan Sistem Jaminan Sosial. 19.
- Pebdika, A., Herdiana, R., & Solihudin, D. (2023). Klasifikasi Menggunakan Metode Naive Bayes Untuk Menentukan Calon Penerima Pip. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), 452–458.
<https://doi.org/10.36040/jati.v7i1.6303>
- Prasetyo, P. E., & Kistanti, N. R. (2020). Human capital, institutional economics and entrepreneurship as a driver for quality & sustainable economic growth. *Entrepreneurship and Sustainability Issues*, 7(4), 2575–2589.
[https://doi.org/10.9770/jesi.2020.7.4\(1\)](https://doi.org/10.9770/jesi.2020.7.4(1))
- Putra, V. P., Azinar, A. R., Azhar, S. S., & Imah, E. M. (2024). Identifikasi Penyalahgunaan Dana Bantuan Sosial untuk Penanggulangan Kemiskinan Berdasarkan Cuitan pada Media Sosial Menggunakan Deep Transfer-Learning. *Buletin Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi Dan Komunikasi*, 2(1), 58–66.
- Putry, J. B. E., Sasongko, A. T., & Hadikristanto, W. (2024). Optimasi Decision Tree Menggunakan Particle Swarm Optimization (PSO) pada Risiko Kredit KMG Bank DKI. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(4), 1403–1410.
<https://doi.org/10.57152/malcom.v4i4.1521>
- Rahayu, D. S., Afifah, J., & Intan, S. (2023). Classification of Diabetes Mellitus Using C4 . 5 Algorithm , Support Vector Machine (SVM) and Linear Regression Klasifikasi Penyakit Diabetes Melitus Menggunakan Algoritma C4 . 5 , Support Vector Machine (SVM) dan Regresi Linear. *SENTIMAS: Seminar Nasional Penelitian Dan Pengabdian Masyarakat*, 1(1 SE-), 56–63.
<https://journal.irpi.or.id/index.php/sentimas/article/view/550>
- Sulaiimah, A. R., Meidyustiani, R., & Anwar, S. (2022). Analisis Pengaruh Fraud Diamond Terhadap Fraudulent Financial Statement (Studi Empiris Pada Perusahaan Perbankan Yang Terdaftar Di Bursa Efek Indonesia Periode 2016 – 2020): *Jurnal Sinar Manajemen*, 9(3), 450–462.
<https://www.jurnal.unismuhpalu.ac.id/index.php/JSM/article/view/3004>
- Waraningtyas, H. D. R. (2022). Sistem Pemidanaan Tindak Pidana Korupsi Dana Bantuan Sosial Covid-19 Dalam Perspektif Economic Analysis Of Law.
- Wibawani, S., Hernanda, F., Kusuma, R. G., & Irawan, F. A. (2021). Evaluasi Program BLT Dana Desa Sebagai Jaring Pengaman Sosial Di Desa Kemlagi, Kabupaten Mojokerto. *Syntax Idea*, 3(5), 1205. <https://jurnal.syntax-idea.co.id/index.php/syntax-idea/article/view/1194>