

BAB IV

HASIL DAN PEMBAHASAN

4.1. Indikator Evaluasi Model

Penelitian ini menggunakan pendekatan klasifikasi biner untuk mengelompokkan produk ke dalam dua kategori, yaitu “Diminati” dan “Tidak Diminati”. Dalam konteks penelitian ini, model pembelajaran mesin tidak hanya diharapkan mampu menghasilkan prediksi yang benar secara umum, tetapi juga memiliki kemampuan yang baik dalam membedakan kedua kelas secara proporsional. Mengingat distribusi data uji menunjukkan adanya ketidakseimbangan antara jumlah produk pada kedua kelas tersebut, evaluasi kinerja model tidak dapat semata-mata didasarkan pada nilai akurasi.

Dasar utama pengukuran performa model dalam penelitian ini adalah confusion matrix. Matriks ini merepresentasikan hasil klasifikasi ke dalam empat komponen, yaitu True Negative (TN), False Positive (FP), False Negative (FN), dan True Positive (TP). TN menunjukkan jumlah produk yang benar diprediksi sebagai “Tidak Diminati”, sedangkan TP menunjukkan jumlah produk yang benar diprediksi sebagai “Diminati”. FP merepresentasikan produk yang sebenarnya tidak diminati tetapi diprediksi sebagai diminati, sementara FN menunjukkan produk yang sebenarnya diminati namun tidak berhasil dikenali oleh model. Keempat komponen tersebut menjadi dasar perhitungan seluruh metrik evaluasi yang digunakan dalam penelitian ini.

Accuracy digunakan sebagai indikator awal untuk menggambarkan proporsi total prediksi yang benar terhadap keseluruhan data uji. Namun, pada data dengan distribusi kelas yang tidak seimbang, akurasi saja belum cukup untuk menggambarkan performa model secara menyeluruh. Oleh sebab itu, evaluasi dilengkapi dengan precision, recall, dan F1-score, khususnya untuk menilai kinerja model pada kelas “Diminati”. Precision menunjukkan tingkat ketepatan model ketika memprediksi suatu produk sebagai diminati, recall mengukur kemampuan model dalam mendeteksi seluruh produk yang benar-benar diminati, sedangkan F1-score merupakan rata-rata harmonik dari precision dan recall yang mencerminkan keseimbangan antara ketepatan dan kelengkapan deteksi.

Selain itu, penelitian ini juga menggunakan Area Under Curve (AUC) dari kurva ROC sebagai indikator tambahan untuk menilai kemampuan model dalam membedakan kedua kelas secara keseluruhan. Nilai AUC yang semakin mendekati 1 menunjukkan bahwa model memiliki kemampuan diskriminatif yang semakin baik, sedangkan nilai yang mendekati 0,5 menunjukkan performa yang cenderung setara dengan tebakan acak. Dengan demikian, penggunaan accuracy, precision, recall, F1-score, dan AUC secara bersama-sama memungkinkan evaluasi model dilakukan secara lebih menyeluruh, sehingga interpretasi hasil penelitian tidak hanya berfokus pada satu metrik, melainkan mencerminkan kinerja model secara utuh.

4.2. Implementasi Python

Pada penelitian ini, proses pemodelan klasifikasi dilakukan menggunakan bahasa pemrograman Python dengan bantuan beberapa pustaka pengolahan data

dan *machine learning*, yaitu *pandas*, *numpy*, *matplotlib*, dan *scikit-learn*. Python dipilih karena mampu mendukung seluruh tahapan analisis secara terstruktur dan efisien, mulai dari membaca data, melakukan prapemrosesan, membentuk fitur, melatih model, hingga mengevaluasi hasil klasifikasi. Dengan dukungan pustaka tersebut, proses analisis dapat dilakukan dalam satu alur kerja yang runtut sehingga memudahkan pengolahan data dan interpretasi hasil penelitian.

Data dalam penelitian ini dibagi menggunakan pendekatan *time-based split* ke dalam dua bagian, yaitu data latih dan data uji dengan proporsi 80:20 berdasarkan urutan tanggal transaksi. Pembagian ini dilakukan dengan mengurutkan terlebih dahulu seluruh tanggal unik pada dataset, kemudian membaginya ke dalam masing-masing kelompok sesuai proporsi yang telah ditetapkan. Pendekatan ini dipilih agar urutan waktu pada data tetap terjaga, sehingga pelatihan dan pengujian model lebih mencerminkan kondisi nyata, yaitu memprediksi data pada periode berikutnya berdasarkan data dari periode sebelumnya. Variabel yang digunakan sebagai prediktor mencakup *Nama_Produk*, *Harga_Satuan*, fitur historis jumlah pembelian dan frekuensi transaksi, statistik agregat harian, serta fitur kalender seperti hari, tanggal, dan bulan. Sementara itu, variabel target yang diprediksi adalah *Label*, yang terdiri atas dua kelas, yaitu *Diminati (1)* dan *Tidak Diminati (0)*.

Dalam penerapannya, model *Support Vector Machine (SVM)* dibangun dengan melakukan standarisasi fitur terlebih dahulu agar perbedaan skala antarvariabel numerik tidak memengaruhi proses pembentukan *hyperplane*. Sebaliknya, model *Random Forest* dilatih menggunakan fitur yang sama tanpa

memerlukan standarisasi khusus, karena algoritma ini bekerja berdasarkan pembentukan pohon keputusan yang relatif tidak sensitif terhadap skala data. Dengan demikian, kedua model dibangun dari informasi yang sama, tetapi tetap disesuaikan dengan karakteristik masing-masing metode.

Setelah proses pelatihan selesai, model dievaluasi menggunakan data uji untuk melihat kemampuan generalisasinya dalam mengklasifikasikan produk ke dalam kategori Diminati dan Tidak Diminati. Evaluasi dilakukan menggunakan confusion matrix serta beberapa metrik kinerja, yaitu accuracy, precision, recall, F1-score, dan AUC. *Confusion matrix* digunakan untuk melihat distribusi prediksi benar dan salah pada masing-masing kelas, sedangkan metrik evaluasi digunakan untuk menilai performa model secara kuantitatif. Melalui pendekatan ini, hasil klasifikasi dari model SVM dan Random Forest dapat dibandingkan secara objektif sehingga dapat ditentukan model yang memberikan performa terbaik dalam penelitian ini.

```
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt

from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.impute import SimpleImputer

from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    roc_auc_score, confusion_matrix, classification_report, ConfusionMatrixDisplay
)
```

Gambar 4. 1 Kode Import Library Python

Gambar 4.1 menunjukkan tahap awal implementasi Python untuk membangun model klasifikasi pada data penjualan atau minat produk. Baris pertama mengimpor pustaka dasar, yaitu pandas untuk membaca dan mengelola data tabular, numpy untuk operasi numerik, re untuk pemrosesan teks sederhana, serta matplotlib.pyplot untuk visualisasi hasil. Selanjutnya, kode mengambil komponen dari scikit-learn yang dipakai dalam alur pemodelan. ColumnTransformer dan Pipeline digunakan agar proses prapengolahan dan pelatihan model dapat disusun rapi dalam satu rangkaian kerja. OneHotEncoder dipakai untuk mengubah variabel kategorik menjadi bentuk numerik, StandardScaler untuk menstandarkan skala data, dan SimpleImputer untuk mengisi nilai yang hilang. Pada tahap pemodelan, digunakan dua algoritma klasifikasi, yaitu SVC sebagai representasi Support Vector Machine dan RandomForestClassifier sebagai metode berbasis ensemble. Terakhir, berbagai metrik evaluasi diimpor, seperti accuracy, precision, recall, F1-score, AUC, confusion matrix, classification report, dan ConfusionMatrixDisplay, agar performa model dapat dianalisis secara menyeluruh, terukur, dan konsisten.

```

df = pd.read_excel(FILE_PATH)
df["Tanggal"] = pd.to_datetime(df["Tanggal"], errors="coerce", dayfirst=True)

# Konversi "Rp63.000" -> 63000
def rp_to_int(x):
    if pd.isna(x):
        return np.nan
    s = str(x).replace("Rp", "").replace(".", "").replace(",", "").strip()
    s = re.sub(r"\s+", "", s)
    return int(s) if s else np.nan

df["Harga_Satuan"] = df["Harga_Satuan"].apply(rp_to_int)

# Ambil kolom yang dipakai
df = df[["Tanggal", "Nama_Produk", "Harga_Satuan", "Jumlah_Pembelian", "Frekuensi_Transaksi"]].copy()

# Agregasi jika ada duplikat Tanggal-Produk
df = df.groupby(["Tanggal", "Nama_Produk"], as_index=False).agg(
    Harga_Satuan=("Harga_Satuan", "median"),
    Jumlah_Pembelian=("Jumlah_Pembelian", "sum"),
    Frekuensi_Transaksi=("Frekuensi_Transaksi", "sum")
)

```

Gambar 4. 2 Load Data dan Preprocessing Awal

Gambar 4.2 ini menampilkan tahap preprocessing dataset menggunakan pandas dan konversi kolom tanggal menjadi format datetime. Selanjutnya, harga produk yang awalnya berupa string diubah menjadi tipe numerik menggunakan fungsi `rp_to_int`. Proses ini memastikan integritas data kuantitatif sebelum dilakukan analisis. Selain itu, agregasi data duplikat berdasarkan kombinasi Tanggal dan Nama_Produk dilakukan untuk memperoleh nilai median harga, total kuantitas pembelian, dan jumlah transaksi, sehingga dataset siap untuk proses pembuatan label dan ekstraksi fitur berikutnya. Tahap ini menjadi fondasi penting dalam workflow karena kualitas preprocessing langsung memengaruhi akurasi prediksi model klasifikasi.

```

# 2) LABEL: DIMINATI = TOP-K HARIAN BERDASARKAN KUANTITAS
# =====
df["rank_qty"] = df.groupby("Tanggal")["Jumlah_Pembelian"].rank(method="first", ascending=False)
df["Label"] = ((df["Jumlah_Pembelian"] > 0) & (df["rank_qty"] <= TOP_K)).astype(int)

# =====
# 3) FITUR KONTEKS HARIAN (SHIFT 1 HARI -> anti leakage)
# =====
daily = df.groupby("Tanggal").agg(
    TotalQty_All=("Jumlah_Pembelian", "sum"),
    MeanQty_All=("Jumlah_Pembelian", "mean"),
    StdQty_All=("Jumlah_Pembelian", lambda s: s.std(ddof=0)),
    MaxQty_All=("Jumlah_Pembelian", "max"),
    NonzeroProducts=("Jumlah_Pembelian", lambda s: (s > 0).sum())
).reset_index().sort_values("Tanggal")

```

Gambar 4. 3 Pembuatan Label Top-K dan Fitur Konteks

Gambar 4.3 ini menampilkan proses penentuan label produk diminati menggunakan metode Top-K harian, di mana produk dengan kuantitas pembelian tertinggi diberi label 1, sedangkan lainnya 0. Selain itu, ditunjukkan pembuatan fitur konteks harian seperti total, rata-rata, standar deviasi, maksimum, jumlah produk non-zero, dan cutoff Top-K yang digeser satu hari untuk mencegah data leakage. Fitur per produk juga dibuat dengan lag 1 dan 7 hari serta rolling mean 3 dan 7 hari, memberikan informasi tren historis pembelian. Blok kode ini penting karena menghubungkan data mentah dengan variabel prediktor yang akan digunakan dalam pemodelan machine learning.

```

# 5) SPLIT TIME-BASED (TRAIN/TEST)
# =====
unique_dates = sorted(model_df["Tanggal"].unique())
n = len(unique_dates)

train_end = int(n * TRAIN_RATIO)

train_dates = set(unique_dates[:train_end])
test_dates = set(unique_dates[train_end:])

train = model_df[model_df["Tanggal"].isin(train_dates)].copy()
test = model_df[model_df["Tanggal"].isin(test_dates)].copy()

FEATURES = [
    "Nama_Produk", "Harga_Satuan",
    "Qty_Lag1", "Qty_Lag7", "Qty_MA3", "Qty_MA7",
    "Freq_Lag1", "Freq_Lag7",
    "TotalQty_All_Lag1", "MeanQty_All_Lag1", "StdQty_All_Lag1",
    "MaxQty_All_Lag1", "NonzeroProducts_Lag1", "CutoffTopK_Lag1",
    "DOW", "DOM", "Month"
]

X_train, y_train = train[FEATURES], train["Label"]
X_test, y_test = test[FEATURES], test["Label"]

```

Gambar 4. 4 Pembagian Dataset dan Pipeline Model

Gambar 4.4 menampilkan pembagian dataset secara berbasis waktu (time-based split) menjadi dua bagian, yaitu train dan test, berdasarkan urutan tanggal transaksi untuk menjaga urutan temporal data. Pada tahap ini juga dilakukan pemisahan antara variabel prediktor dan variabel target. Variabel prediktor merupakan atribut yang digunakan dalam proses klasifikasi, sedangkan variabel target yang diprediksi adalah Label. Hasil pembagian ini kemudian menghasilkan data `X_train`, `y_train`, `X_test`, dan `y_test` yang digunakan pada proses pelatihan dan pengujian model.

```

# 9) EVALUASI TEST
# =====
prob_test_svm = svm.predict_proba(X_test)[:, 1]
prob_test_rf = rf.predict_proba(X_test)[:, 1]

# Threshold default = 0.5
pred_test_svm = (prob_test_svm >= 0.5).astype(int)
pred_test_rf = (prob_test_rf >= 0.5).astype(int)

def eval_metrics(y_true, y_pred, y_prob):
    return {
        "Accuracy": accuracy_score(y_true, y_pred),
        "Precision": precision_score(y_true, y_pred, zero_division=0),
        "Recall": recall_score(y_true, y_pred, zero_division=0),
        "F1": f1_score(y_true, y_pred, zero_division=0),
        "AUC": roc_auc_score(y_true, y_prob) if y_true.nunique() == 2 else np.nan
    }

res_svm = eval_metrics(y_test, pred_test_svm, prob_test_svm)
res_rf = eval_metrics(y_test, pred_test_rf, prob_test_rf)

print("\n=== HASIL TEST ===")
print("SVM :", res_svm)
print("RF  :", res_rf)

```

Gambar 4. 5 Proses Evaluasi Model metrik

Kode pada gambar menunjukkan tahap evaluasi model setelah proses pelatihan selesai dilakukan. Pertama, model SVM dan Random Forest dilatih menggunakan data latih melalui perintah `fit(X_train, y_train)`. Setelah itu, kedua model menghasilkan probabilitas prediksi pada data uji dengan `predict_proba`, khususnya probabilitas untuk kelas positif atau kelas “Diminati”. Nilai probabilitas ini kemudian dikonversi menjadi label biner menggunakan batas 0,5, sehingga model dapat menghasilkan keputusan akhir dalam bentuk 0 atau 1. Selanjutnya, dibuat fungsi `eval_metrics()` untuk menghitung beberapa ukuran kinerja utama, yaitu accuracy, precision, recall, F1-score, dan AUC. Penggunaan fungsi ini membuat proses evaluasi menjadi lebih ringkas, konsisten, dan mudah diterapkan pada lebih dari satu model. Bagian `zero_division=0` digunakan agar perhitungan tetap aman ketika ada kondisi tertentu pada data. Terakhir, hasil evaluasi masing-

masing model disimpan dalam `res_svm` dan `res_rf` untuk dibandingkan secara objektif pada tahap analisis hasil penelitian.

4.3. Data Training

Tabel 4. 1 Data Training Penjualan Harian

Tanggal	Nama Produk	Harga Satuan	Jumlah Pembelian	Frekuensi Transaksi
1/6/2025	Beras	Rp63.000	10	6
1/6/2025	Minyak Goreng	Rp15.000	2	1
1/6/2025	Gula Pasir	Rp14.000	1	1
1/6/2025	Garam	Rp4.000	2	2
1/6/2025	Telur	Rp2.000	108	10
1/6/2025	Gas Elpiji	Rp22.000	2	2
1/6/2025	Rokok	Rp25.000	23	14
1/6/2025	Aqua	Rp5.000	18	10
1/6/2025	Mie Instan	Rp3.000	45	13
1/6/2025	Sabun Batang	Rp4.000	8	4
1/6/2025	Tepung	Rp10.000	2	2
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
12/8/2025	Detergen	Rp15.000	1	1
12/8/2025	Kecap	Rp3.000	0	0
13/8/2025	Beras	Rp63.000	5	1
13/8/2025	Minyak Goreng	Rp15.000	7	4
13/8/2025	Gula Pasir	Rp14.000	3	2
13/8/2025	Garam	Rp4.000	1	1
13/8/2025	Telur	Rp2.000	42	8
13/8/2025	Gas Elpiji	Rp22.000	4	3
13/8/2025	Rokok	Rp25.000	14	9
13/8/2025	Aqua	Rp5.000	8	4
13/8/2025	Mie Instan	Rp3.000	31	7

Tabel data training ini menampilkan sebagian data penjualan yang telah masuk ke kelompok data latih setelah dataset dibagi menggunakan metode time-based split dengan perbandingan 80:20. Pembagian dilakukan dengan mengurutkan

data terlebih dahulu berdasarkan tanggal transaksi, kemudian 80% data pada urutan awal digunakan sebagai data training dan 20% sisanya digunakan sebagai data testing. Cara ini dipakai agar urutan waktu tetap terjaga, sehingga model belajar dari data penjualan pada periode sebelumnya sebelum diuji pada periode sesudahnya. Pada cuplikan data training yang ditampilkan, terlihat bahwa pada 1 Juni 2025 produk Telur memiliki jumlah pembelian tertinggi, yaitu 108 unit dengan 10 kali transaksi, disusul Mie Instan sebanyak 45 unit dan Rokok sebanyak 23 unit. Sementara itu, pada bagian akhir cuplikan data, yaitu 13 Agustus 2025, Telur kembali menunjukkan jumlah pembelian yang tinggi, yaitu 42 unit, diikuti Mie Instan sebanyak 31 unit dan Rokok sebanyak 14 unit. Secara umum, data training ini menunjukkan bahwa setiap produk memiliki pola penjualan yang berbeda-beda, sehingga data tersebut menjadi dasar bagi model untuk mempelajari karakteristik produk yang diminati dan tidak diminati.

4.4. Data Testing

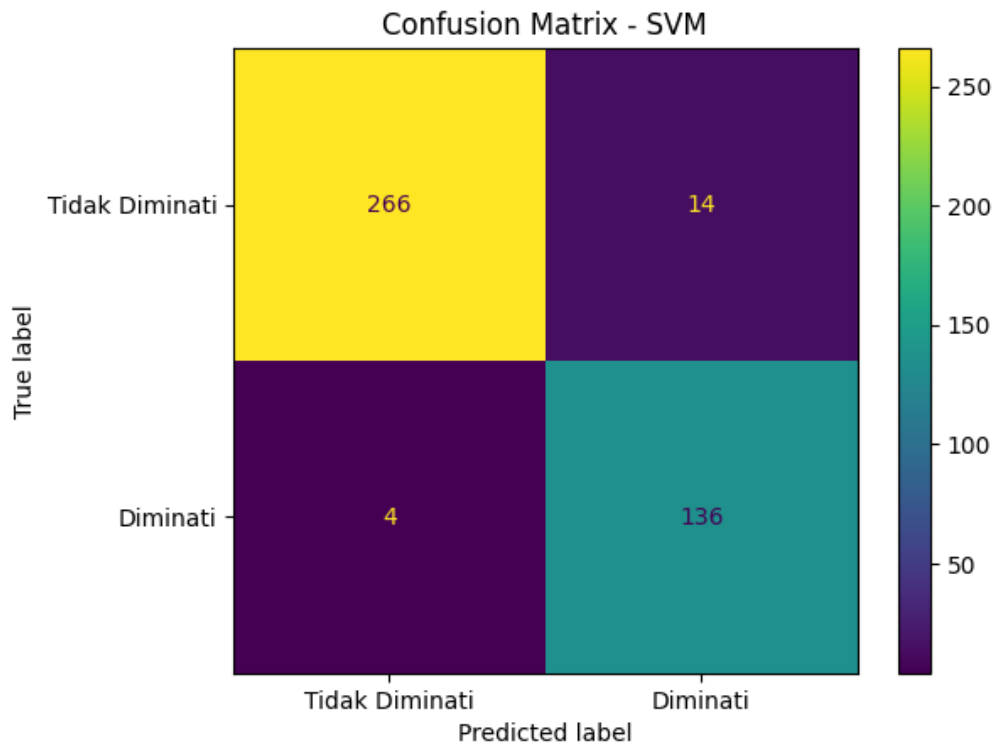
Tabel 4. 2 Data Testing Penjualan Harian

Tanggal	Nama Produk	Harga Satuan	Jumlah Pembelian	Frekuensi Transaksi
13/8/2025	Sabun Batang	Rp4.000	5	1
13/8/2025	Tepung	Rp10.000	1	1
13/8/2025	Kopi	Rp2.000	2	1
13/8/2025	Susu	Rp12.000	3	3
13/8/2025	Detergen	Rp15.000	3	3
13/8/2025	Kecap	Rp3.000	1	1
14/8/2025	Beras	Rp63.000	6	4
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
31/8/2025	Gas Elpiji	Rp22.000	4	1
31/8/2025	Rokok	Rp25.000	14	6
31/8/2025	Aqua	Rp5.000	13	9

31/8/2025	Mie Instan	Rp3.000	41	9
31/8/2025	Sabun Batang	Rp4.000	2	2
31/8/2025	Tepung	Rp10.000	1	1
31/8/2025	Kopi	Rp2.000	2	1
31/8/2025	Susu	Rp12.000	5	2
31/8/2025	Detergen	Rp15.000	3	2
31/8/2025	Kecap	Rp3.000	0	0

Tabel data testing ini menampilkan sebagian data penjualan yang masuk ke kelompok data uji setelah dataset dibagi menggunakan metode time-based split dengan perbandingan 80:20. Pembagian dilakukan dengan mengurutkan seluruh data berdasarkan tanggal transaksi, kemudian 80% data pada periode awal digunakan sebagai data training dan 20% data pada periode akhir digunakan sebagai data testing, sehingga urutan waktu tetap terjaga dan pengujian model dapat menggambarkan kondisi yang lebih nyata. Berdasarkan cuplikan data yang ditampilkan, pada 13 Agustus 2025 terlihat beberapa produk dengan jumlah pembelian yang relatif rendah, seperti Sabun Batang sebanyak 5 unit, Susu dan Detergen masing-masing 3 unit, serta Kecap sebanyak 1 unit. Sementara itu, pada 31 Agustus 2025, Mie Instan menjadi produk dengan jumlah pembelian tertinggi, yaitu 41 unit dengan 9 kali transaksi, disusul Rokok sebanyak 14 unit dan Aqua sebanyak 13 unit. Pada tanggal yang sama, Kecap tercatat bernilai 0, yang menunjukkan bahwa tidak terjadi pembelian untuk produk tersebut. Secara keseluruhan, data testing ini memperlihatkan adanya variasi permintaan antarproduk pada periode akhir pengamatan dan digunakan untuk menilai kemampuan model dalam mengenali pola produk yang diminati dan tidak diminati pada data baru.

4.5. Hasil Model SVM



Gambar 4. 6 Confusion Matrix SVM

Berdasarkan Gambar 4.6, confusion matrix pada model Support Vector Machine (SVM) menunjukkan bahwa model memiliki kinerja klasifikasi yang sangat baik dalam membedakan dua kategori, yaitu “Tidak Diminati” dan “Diminati”. Nilai True Negative (TN) sebesar 266 menunjukkan bahwa model mampu mengklasifikasikan dengan benar sebagian besar produk yang termasuk kategori “Tidak Diminati”. Di sisi lain, nilai True Positive (TP) sebesar 136 menunjukkan bahwa model juga sangat baik dalam mengenali produk yang benar-benar termasuk kategori “Diminati”. Meskipun demikian, masih terdapat kesalahan klasifikasi yang ditunjukkan oleh nilai False Positive (FP) sebesar 14, yaitu produk yang sebenarnya tidak diminati tetapi diprediksi sebagai diminati, serta False

Negative (FN) sebesar 4, yaitu produk yang sebenarnya diminati tetapi diprediksi sebagai tidak diminati. Jumlah kesalahan yang relatif kecil ini menandakan bahwa model SVM memiliki kemampuan prediksi yang tinggi dan cukup andal untuk digunakan dalam klasifikasi tingkat minat produk.

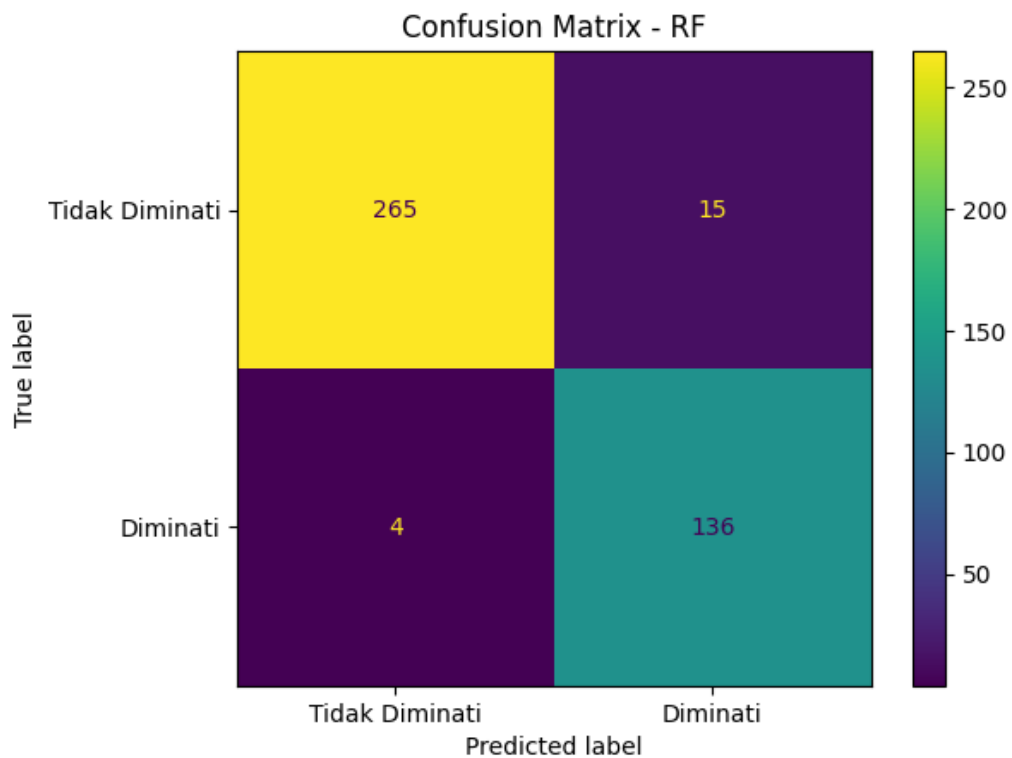
Tabel 4. 3 Hasil Evaluasi Svm

Kategori	Accuracy	AUC	Precision	Recall	F1-score	Support
Keseluruhan Model	0.9571	0.9896	-	-	-	420
Tidak Diminati	-	-	0.9852	0.9500	0.9673	280
Diminati	-	-	0.9067	0.9714	0.9379	140

Berdasarkan tabel evaluasi metrik di atas, model SVM menunjukkan kinerja yang sangat baik dalam mengklasifikasikan produk ke dalam kategori Diminati dan Tidak Diminati. Secara keseluruhan, model memperoleh accuracy sebesar 0,9571, yang berarti model mampu memprediksi dengan benar sekitar 95,71% dari seluruh data uji sebanyak 420 data. Selain itu, nilai AUC sebesar 0,9896 menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam membedakan dua kelas secara keseluruhan. Jika dilihat per kelas, pada kategori Tidak Diminati model menghasilkan precision 0,9852, recall 0,9500, dan F1-score 0,9673 dari 280 data, yang menunjukkan bahwa model sangat baik dalam mengenali produk yang tidak diminati. Sementara itu, pada kategori Diminati, model memperoleh precision 0,9067, recall 0,9714, dan F1-score 0,9379 dari 140 data. Nilai recall yang tinggi pada kelas Diminati menunjukkan bahwa sebagian besar produk yang benar-benar diminati berhasil dikenali oleh model. Secara umum, hasil ini menunjukkan bahwa model SVM tidak hanya memiliki tingkat ketepatan yang tinggi secara keseluruhan,

tetapi juga mampu menjaga keseimbangan performa pada kedua kelas, sehingga layak digunakan untuk mendukung klasifikasi minat produk pada penelitian ini.

4.6. Hasil Model Random Forest



Gambar 4. 7 Confusion Matrix Random Forest

Berdasarkan Gambar 4.7, confusion matrix model Random Forest menunjukkan bahwa model memiliki kemampuan klasifikasi yang sangat baik dalam membedakan produk ke dalam kategori “Tidak Diminati” dan “Diminati”. Nilai True Negative (TN) sebesar 265 mengindikasikan bahwa model mampu mengidentifikasi dengan benar sebagian besar produk yang termasuk kategori “Tidak Diminati”. Sementara itu, nilai True Positive (TP) sebesar 136 menunjukkan bahwa model juga sangat baik dalam memprediksi produk yang benar-benar termasuk kategori “Diminati”. Meskipun demikian, masih terdapat sejumlah kecil

kesalahan klasifikasi, yaitu False Positive (FP) sebesar 15 dan False Negative (FN) sebesar 4. Nilai FP menunjukkan adanya produk yang sebenarnya tidak diminati tetapi diprediksi sebagai diminati, sedangkan nilai FN menunjukkan adanya produk yang sebenarnya diminati tetapi diprediksi sebagai tidak diminati. Secara keseluruhan, hasil ini menegaskan bahwa model Random Forest memiliki tingkat akurasi klasifikasi yang tinggi dan dapat digunakan sebagai model yang andal dalam mendukung prediksi minat produk.

Tabel 4. 4 Hasil Evaluasi Random Forest

Kategori	Accuracy	AUC	Precision	Recall	F1-score	Support
Keseluruhan Model	0.9547	0.9908	-	-	-	420
Tidak Diminati	-	-	0.9851	0.9464	0.9654	280
Diminati	-	-	0.9007	0.9714	0.9347	140

Berdasarkan tabel di atas, model Random Forest menunjukkan kinerja yang sangat baik dalam mengklasifikasikan produk ke dalam kategori Diminati dan Tidak Diminati. Secara keseluruhan, model memperoleh accuracy sebesar 0,9547, yang berarti mampu memberikan prediksi yang benar pada sekitar 95,47% dari total 420 data uji. Selain itu, nilai AUC sebesar 0,9908 menunjukkan bahwa model memiliki kemampuan yang sangat kuat dalam membedakan kedua kelas secara keseluruhan. Jika dilihat pada masing-masing kelas, untuk kategori Tidak Diminati model menghasilkan precision sebesar 0,9851, recall sebesar 0,9464, dan F1-score sebesar 0,9654 dari 280 data, yang menunjukkan bahwa model sangat baik dalam mengenali produk yang tidak diminati. Sementara itu, pada kategori Diminati, model memperoleh precision sebesar 0,9007, recall sebesar 0,9714, dan F1-score sebesar 0,9347 dari 140 data. Nilai recall yang tinggi pada kelas Diminati

menunjukkan bahwa sebagian besar produk yang benar-benar diminati berhasil dikenali oleh model. Secara umum, hasil ini memperlihatkan bahwa model Random Forest memiliki performa yang tinggi, baik dari sisi ketepatan prediksi secara keseluruhan maupun dalam menjaga keseimbangan klasifikasi pada kedua kelas.

4.7. Perbandingan Kinerja Model

Pada hasil model ini, perbandingan difokuskan pada metrik uji karena pengujian dilakukan pada data yang tidak digunakan saat pelatihan dan validasi. Model pertama yang digunakan adalah SVM, dengan Akurasi 95,71%, AUC 0,9896, dan Precision untuk kategori *Tidak Diminati* sebesar 98,52%, serta Recall 95,00%. Untuk kategori *Diminati*, Precision SVM adalah 90,67%, dan Recall 97,14%. Model kedua, yaitu Random Forest (RF), menunjukkan Akurasi yang sedikit lebih rendah, yakni 95,47%, namun AUC RF sedikit lebih tinggi, yaitu 0,9908. Selain itu, precision untuk kategori *Tidak Diminati* sedikit lebih rendah dibandingkan SVM, yaitu 98,51%, dan Recall 94,64%. Untuk kategori *Diminati*, precision RF adalah 90,07%, sementara recall-nya sama dengan SVM, yaitu 97,14%.

Dalam hal Akurasi, kedua model menunjukkan performa yang sangat baik dan serupa, dengan selisih yang sangat kecil (hanya 0,24%), yang menunjukkan bahwa keduanya mampu melakukan klasifikasi dengan tingkat kesalahan yang rendah. AUC kedua model menunjukkan angka yang sangat tinggi, mengindikasikan kemampuan model dalam memisahkan kedua kelas dengan baik. Perbedaan utama terlihat pada Precision dan F1-Score untuk kategori *Tidak Diminati*, di mana SVM sedikit lebih unggul, tetapi Recall menunjukkan nilai yang

lebih stabil pada kedua model. Ini mengindikasikan bahwa kedua model cenderung berhasil mendeteksi kelas positif dan negatif dengan tingkat sensitivitas yang tinggi.

Perbandingan per kelas juga menunjukkan performa yang hampir setara, meskipun SVM menunjukkan sedikit keunggulan pada kategori *Tidak Diminati*. Dengan Recall untuk kategori *Tidak Diminati* sebesar 95,00% pada SVM dibandingkan 94,64% pada RF, serta F1-Score yang lebih tinggi pada SVM, yaitu 96,73% berbanding 95,54% pada RF. Kelas *Diminati* menunjukkan bahwa kedua model memiliki performa yang sangat mirip, dengan Recall yang sangat tinggi, tetapi ada sedikit perbedaan pada Precision, di mana SVM sedikit lebih unggul. Keseluruhan, meskipun perbedaan antara kedua model tidak signifikan, SVM sedikit lebih unggul dalam hal keakuratan dan deteksi kategori tertentu.

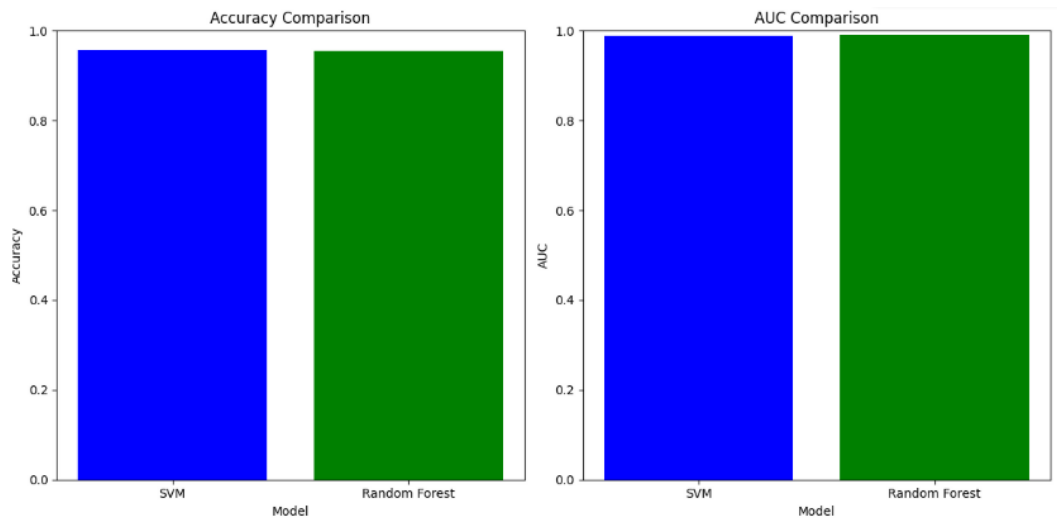
4.7.1. Perbandingan Metrik Utama

Perbandingan metrik utama pada tahap pengujian dilakukan untuk menilai kinerja generalisasi kedua model pada data yang tidak dilibatkan dalam pelatihan dan validasi. Berdasarkan hasil evaluasi, Model SVM yang menggunakan kernel RBF memperoleh akurasi uji sebesar 95,71%, sementara Model Random Forest mencapai akurasi 95,47%. Selisih yang sangat kecil (0,24%) ini menunjukkan bahwa kedua model memiliki kemampuan yang hampir setara dalam memprediksi label pada data uji. Namun, perbedaan kecil tersebut tidak terlalu signifikan dan menunjukkan bahwa kedua model dapat menghasilkan performa yang baik pada data yang belum pernah dilihat sebelumnya.

Selain akurasi, metrik AUC digunakan untuk mengukur seberapa baik model dalam memisahkan antara kelas positif dan negatif. Model SVM

memperoleh nilai AUC sebesar 0,9896, sedangkan Model Random Forest sedikit lebih tinggi dengan nilai AUC 0,9908. Meskipun perbedaan AUC antara kedua model sangat kecil, AUC yang mendekati angka 1 pada kedua model menandakan bahwa keduanya sangat baik dalam membedakan antara kelas *Diminati* dan *Tidak Diminati*. Secara keseluruhan, AUC memperkuat hasil yang diperoleh dari akurasi dengan menunjukkan bahwa kedua model mampu menangani masalah klasifikasi dengan sangat baik.

Dalam kesimpulan, meskipun Model SVM sedikit lebih unggul dalam hal akurasi, kedua model menunjukkan performa yang sangat baik berdasarkan metrik utama yang diuji. Secara analitis, perbandingan antara akurasi dan AUC menunjukkan bahwa kedua model relatif seimbang dalam kinerjanya. Selanjutnya, perlu dicatat bahwa meskipun kedua model menunjukkan hasil yang serupa, perbedaan kecil di antara mereka mungkin relevan dalam konteks aplikasi praktis, tergantung pada kebutuhan spesifik pengguna, misalnya dalam hal kecepatan pemrosesan atau kebutuhan untuk mengurangi overfitting.



Gambar 4. 8 Perbandingan Accuracy dan AUC

Gambar di atas menunjukkan perbandingan dua metrik utama pada data uji, yaitu Akurasi dan AUC, antara Model SVM dan Random Forest, keduanya menunjukkan kinerja yang hampir setara. Model SVM mencatatkan akurasi 95,71% dan AUC 0,9896, sementara Model Random Forest sedikit lebih rendah dengan akurasi 95,47% tetapi memiliki AUC sedikit lebih tinggi yaitu 0,9908. Meskipun ada sedikit perbedaan antara kedua model ini, perbedaan tersebut tidak signifikan, dengan keduanya menunjukkan kemampuan klasifikasi yang sangat baik. Grafik perbandingan menegaskan bahwa kedua model sangat efektif dalam memisahkan kelas, namun Random Forest sedikit lebih unggul pada AUC.

4.7.2. Perbandingan Perkelas

Tabel 4. 5 Perbandingan Metrik Perkelas SVM vs Rf

Kelas	Precision SVM	Recall SVM	F1 SVM	Precision RF	Recall RF	F1 Rf	Support
Tidak Diminati	0.9852	0.9500	0.9673	0.9851	0.9464	0.9654	280
Diminati	0.9067	0.9714	0.9379	0.9007	0.9714	0.9347	140

Berdasarkan Tabel 4.5, kedua model menunjukkan performa yang sangat baik dalam mengklasifikasikan produk ke dalam kategori Diminati dan Tidak Diminati. Pada kelas Diminati, model SVM memperoleh precision 0,9067, recall 0,9714, dan F1-score 0,9379, sedangkan Random Forest memperoleh precision 0,9007, recall 0,9714, dan F1-score 0,9347. Pada kelas Tidak Diminati, performa kedua model juga relatif seimbang, dengan SVM sedikit unggul pada recall dan F1-score. Secara umum, hasil ini menunjukkan bahwa SVM sedikit lebih unggul dibandingkan Random Forest pada accuracy, precision, dan F1-score, sedangkan Random Forest sedikit unggul pada nilai AUC. Dengan demikian, SVM dapat dinyatakan sebagai model yang sedikit lebih baik secara keseluruhan dalam penelitian ini.

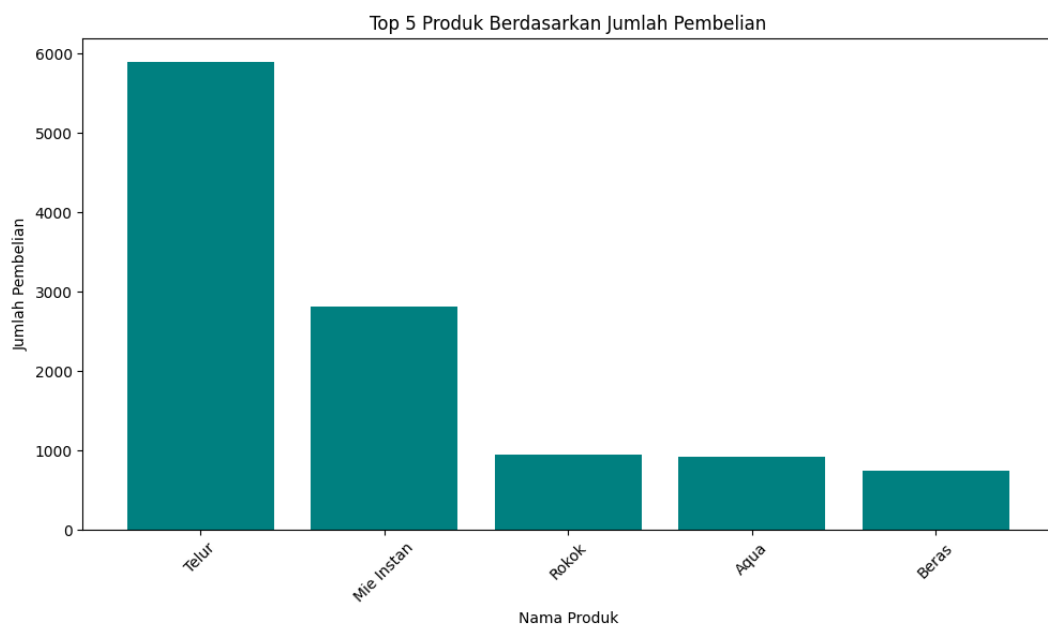
4.8. Identifikasi Produk Diminati

Tabel 4. 6 Top Kuantitas Produk

Produk	Kuantitas
Telur	5893
Mie Instan	2819
Rokok	943
Aqua	916
Beras	748

Tabel 4.6 menunjukkan lima produk dengan total kuantitas tertinggi berdasarkan data transaksi yang dianalisis. Telur menempati posisi pertama dengan jumlah 5893 unit, diikuti oleh Mie Instan sebanyak 2819 unit, Rokok 943 unit, Aqua

916 unit, dan Beras 748 unit. Jika dijumlahkan, total kuantitas lima produk teratas tersebut mencapai 11420 unit. Dari jumlah tersebut, Telur memberikan kontribusi paling besar, yaitu sekitar 51,64% dari total kuantitas Top-5. Selain itu, rentang kuantitas pada kelompok lima produk teratas berada antara 748 unit hingga 5893 unit, yang menunjukkan adanya perbedaan volume penjualan yang cukup besar antarproduk. Berdasarkan data tersebut, Telur dapat diidentifikasi sebagai produk dengan tingkat permintaan paling tinggi dibandingkan produk lainnya, sedangkan Mie Instan menempati posisi kedua dengan selisih kuantitas yang cukup jauh. Temuan ini menunjukkan bahwa kedua produk tersebut memiliki peran yang menonjol dalam struktur penjualan selama periode pengamatan.



Gambar 4. 9 Top Kuantitas Produk

Gambar 4.9 menunjukkan diagram batang kuantitas lima produk dengan penjualan tertinggi di Toko Sembako Mentari berdasarkan data transaksi yang dianalisis. Telur menempati posisi pertama dengan total kuantitas 5893 unit, diikuti

oleh Mie Instan sebanyak 2819 unit, Rokok 943 unit, Aqua 916 unit, dan Beras 748 unit. Perbedaan kuantitas yang cukup besar antara Telur dan produk lainnya menunjukkan bahwa Telur merupakan produk dengan tingkat permintaan paling dominan selama periode pengamatan. Visualisasi ini memperkuat hasil pada Tabel 4.6 bahwa Telur dan Mie Instan menjadi dua produk yang paling menonjol dalam struktur penjualan, sehingga keduanya perlu menjadi perhatian utama dalam pengelolaan stok dan perencanaan persediaan toko.