

BAB V

SIMPULAN DAN SARAN

5.1 SIMPULAN

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan bahwa sistem rekomendasi kategori produk berbasis *algoritma Naive Bayes* dengan pendekatan *predictive analytics* dapat membantu PT. Kilaumas Lestari Mulia dalam mengenali preferensi pelanggan secara lebih akurat dan efisien. Maka dapat disimpulkan beberapa hal yaitu sebagai berikut :

1. Penelitian ini berhasil menghasilkan sistem rekomendasi kategori produk berbasis *algoritma Naive Bayes* yang mampu memanfaatkan data pencarian pengguna untuk memprediksi kategori produk yang relevan. Sistem ini terbukti efektif dalam membantu mengenali minat calon pembeli secara otomatis dan mendukung efisiensi dalam penyusunan strategi promosi maupun pengelolaan stok barang.
2. Penerapan metode *predictive analytics* dalam penelitian ini memberikan hasil prediksi yang akurat dengan memanfaatkan data historis penjualan. *Algoritma Naive Bayes* dapat mengklasifikasikan kata kunci pencarian ke dalam kategori produk dengan performa yang baik, serta mampu memberikan dasar pengambilan keputusan yang lebih cepat dan berbasis data.
3. Penggunaan framework *Streamlit* dalam pengembangan antarmuka aplikasi memberikan kemudahan dalam membangun dashboard interaktif yang ringan, responsif, dan *user-friendly*. Hal ini mempermudah *admin* dan *user* dalam mengakses, mencari produk, dan melihat hasil prediksi tanpa memerlukan instalasi atau konfigurasi sistem yang rumit.

5.2 SARAN

Agar sistem yang dikembangkan dapat terus ditingkatkan dan digunakan secara optimal, beberapa saran yang dapat diberikan antara lain:

1. Pengayaan Dataset : Sistem akan menjadi lebih akurat jika dataset yang digunakan mencakup periode waktu yang lebih panjang, data riil dari aktivitas pencarian pengguna di situs e-commerce perusahaan, dan variasi produk yang lebih luas.
2. Penambahan Fitur Personalisasi : Sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur personalisasi berbasis riwayat pencarian individu, sehingga rekomendasi menjadi lebih spesifik dan relevan untuk tiap pengguna.
3. Integrasi dengan Sistem Operasional : Sebaiknya aplikasi rekomendasi ini diintegrasikan dengan sistem manajemen inventaris dan pemasaran perusahaan agar informasi hasil prediksi dapat langsung digunakan untuk mengatur stok dan promosi produk secara otomatis.
4. Pengujian dengan Data Nyata (*Real-World Testing*) : Untuk meningkatkan *Validitas* sistem, disarankan dilakukan pengujian lanjutan dengan data pencarian dari pengguna aktual pada platform digital perusahaan.
5. Keamanan dan Privasi Data: Seiring dengan penggunaan data pengguna, perlu diperhatikan aspek keamanan data dan kepatuhan terhadap regulasi privasi, seperti dengan melakukan enkripsi data pencarian dan menambahkan otorisasi akses data yang ketat.

DAFTAR PUSTAKA

- Aljohani, A. (2023). *Predictive Analytics and Machine Learning for Real-Time Supply Chain Risk Mitigation and Agility*.
- Bokonda, P. L., Touhami, K. O., & Souissi, N. (2021). *Predictive Analytics Using Machine Learning: Review of Trends and Methods*.
- Buga, R., Buzea, C. G., Agop, M., Ochiuz, L., Vasincu, D., Popa, O., . . . Eva, L. (2025). *Streamlit Application and Deep Learning Model for Brain Metastasis Monitoring After Gamma Knife Treatment*.
- Chen, Y., Li, C., & Wang, H. (2021). *Big Data and Predictive Analytics for Business intelligence: A Bibliographic Study (2000 2021)*.
- F, Fan, H., Li, K., Li, X., Song, T., Zhang, W., . . . Du, B. (2019). *CoVSCode: A Novel Real-Time Collaborative Programming Environment for Lightweight IDE*.
- Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). *Recommendation Sitem: Algorithms, Challenges, Metrics, and Business Opportunities*.
- Haryanto. (2024). *Analisis Big Data dan Artificial Intelligence (AI): dalam Industri Khususnya Prediksi Penyakit Jantung dengan Phytton*.
- Karabila, I., Nossayba, N., El-Ansari, A., & Alami, N. (2023). *Enhancing Collaborative Filtering-Based Recommender Sistem Using Sentiment Analysis*.
- Kurniawan, A., Rifa'i, A., Nafis, M. A., Sefrida, N., & Patria, H. (2022). *Pemilihan Metode Predictive Analytics dengan Machine Learning untuk Analisis dan Strategi Peningkatan Kualitas Kredit Perbankan*.
- Mahajan, P., Udin, S., Hajati, F., & Moni, M. A. (2023). *Ensemble Learning for Disease Prediction: A Review*.
- Priharsari, D. (2022). *Sistematic Literature Review di Bidang Sistem Informasi dan Ilmu Komputer*.
- Rrmoku, K., Selimi, B., & Ahmedi, L. (2022). *Application of Trust in Recommender Systems—Utilizing*.
- Shahbazi, Z., Hazra, D., Park, S., & Byun, Y. C. (2020). *Toward Improving the Prediction Accuracy of Product Recommendation Sistem Using Extreme Gradient. Department of Computer Engineering*.
- Troussas, C., Krouska, A., Koliarakis, A., & Sgouropoulou, C. (2023). *Harnessing the Power*

of *User-Centric Artificial Intelligence: Customized Recommendations and Personalization in Hybrid Recommender Systems.*

Yang, Z., Ren, J., Zhang, Z., Sun, Y., Zang, C., Wang, M., & Wang, L. (2023). *A New Three-Way Incremental Naive Bayes Classifier.*



DAFTAR RIWAYAT HIDUP



Data Pribadi

Nama Lengkap : Nico Pratama
Tempat/Tanggal Lahir : Jakarta, 16 Mei 2002
Jenis Kelamin : Laki-laki
Alamat Lengkap : Villa Grand Tomang Blok C.1 No.1 RT: 02 / RW: 17,
Tangerang (15131), Banten
Agama : Kristen
Telepon : 0895328901650
E-mail : nicopratama1643@gmail.com

Pendidikan Formal

2009 – 2015 SDN Gebang Raya 1
2015 – 2019 SMPN 12 Tangerang
2019 – 2021 SMKN 3 Tangerang
2021 – selesai Universitas Buddhi Dharma Program Studi Teknik
Informatika, Peminatan Data Science

Tangerang, 05 Agustus 2025

Nico Pratama

LAMPIRAN

RE(Recruitment Elicitation)

No	Daftar Analisa Kebutuhan Sistem	Keterangan
1	Bagaimana sistem mengelola otentikasi pengguna dan hak akses berdasarkan peran (admin/user) untuk fitur-fitur seperti prediksi kategori?	√
2	Bagaimana sistem menampilkan, mencari, dan memfilter produk berdasarkan kategori, termasuk kemampuan untuk menampilkan produk terlaris dan memberikan saran pencarian?	√
3	Bagaimana system dapat memanfaatkan riwayat pencarian pengguna untuk mendukung proses prediksi kategori produk?	√
4	Seberapa akurat model Naïve Bayes dalam memprediksi kategori produk berdasarkan pola penjualan historis?	√
5	Apakah hasil prediksi dari metode Naïve Bayes dapat meningkatkan relevansi rekomendasi bagi pengguna?	√

Tangerang, 25-06-2025

Pembimbing



(Rino M.kom)

NIDN : 0420058502

Responden



(Nathania)

(Direktur)

Mahasiswa



(Nico Pratama)

20211000016

No	Daftar Analisa Kebutuhan Sistem	Keterangan
1	Bagaimana sistem mengelola otentikasi pengguna dan hak akses berdasarkan peran (admin/user) untuk fitur-fitur seperti prediksi kategori?	√
2	Bagaimana sistem menampilkan, mencari, dan memfilter produk berdasarkan kategori, termasuk kemampuan untuk menampilkan produk terlaris dan memberikan saran pencarian?	√
3	Bagaimana system dapat memanfaatkan riwayat pencarian pengguna untuk mendukung proses prediksi kategori produk?	√
4	Seberapa akurat model Naïve Bayes dalam memprediksi kategori produk berdasarkan pola penjualan historis?	√
5	Apakah hasil preidksi dari metode Naïve Bayes dapat meningkatkan relevansi rekomendasi bagi pengguna?	√

Tangerang, 25-06-2025

Pembimbing



(Rino M.kom)

NIDN : 0420058502

Responden



(Pama Trisnadi)

Mahasiswa



(Nico Pratama)

20211000016

No	Daftar Analisa Kebutuhan Sistem	Keterangan
1	Bagaimana sistem mengelola otentikasi pengguna dan hak akses berdasarkan peran (admin/user) untuk fitur-fitur seperti prediksi kategori?	√
2	Bagaimana sistem menampilkan, mencari, dan memfilter produk berdasarkan kategori, termasuk kemampuan untuk menampilkan produk terlaris dan memberikan saran pencarian?	√
3	Bagaimana system dapat memanfaatkan riwayat pencarian pengguna untuk mendukung proses prediksi kategori produk?	√
4	Seberapa akurat model Naïve Bayes dalam memprediksi kategori produk berdasarkan pola penjualan historis?	√
5	Apakah hasil preiksi dari metoe Naïve Bayes dapat meningkatkan relevansi rekomendasi bagi pengguna?	√

Tangerang, 25-06-2025

Pembimbing

Responden

Mahasiswa



(Rino M.kom)

NIDN : 0420058502



(Ajeng)



(Nico Pratama)

20211000016

No	Daftar Analisa Kebutuhan Sistem	Keterangan
1	Bagaimana sistem mengelola otentikasi pengguna dan hak akses berdasarkan peran (admin/user) untuk fitur-fitur seperti prediksi kategori?	√
2	Bagaimana sistem menampilkan, mencari, dan memfilter produk berdasarkan kategori, termasuk kemampuan untuk menampilkan produk terlaris dan memberikan saran pencarian?	√
3	Bagaimana system dapat memanfaatkan riwayat pencarian pengguna untuk mendukung proses prediksi kategori produk?	√
4	Seberapa akurat model Naïve Bayes dalam memprediksi kategori produk berdasarkan pola penjualan historis?	√
5	Apakah hasil preiksi dari metoe Naïve Bayes dapat meningkatkan relevansi rekomendasi bagi pengguna?	√

Tangerang, 25-06-2025

Pembimbing



(Rino M.kom)

NIDN : 0420058502

Responden



(Sintya. A.)

Mahasiswa



(Nico Pratama)

20211000016

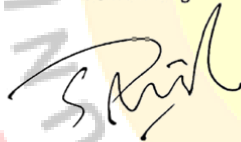
No	Daftar Analisa Kebutuhan Sistem	Keterangan
1	Bagaimana sistem mengelola otentikasi pengguna dan hak akses berdasarkan peran (admin/user) untuk fitur-fitur seperti prediksi kategori?	√
2	Bagaimana sistem menampilkan, mencari, dan memfilter produk berdasarkan kategori, termasuk kemampuan untuk menampilkan produk terlaris dan memberikan saran pencarian?	√
3	Bagaimana system dapat memanfaatkan riwayat pencarian pengguna untuk mendukung proses prediksi kategori produk?	√
4	Seberapa akurat model Naïve Bayes dalam memprediksi kategori produk berdasarkan pola penjualan historis?	√
5	Apakah hasil preiksi dari metoe Naïve Bayes dapat meningkatkan relevansi rekomendasi bagi pengguna?	√

Tangerang, 25-06-2025

Pembimbing

Responden

Mahasiswa



(Rino M.kom)

NIDN : 0420058502





(Nico Pratama)

20211000016

Lampiran 1- 5 Requirement Elicitation

No	Daftar Analisa Kebutuhan Sistem	Keterangan
1	Bagaimana sistem mengelola otentikasi pengguna dan hak akses berdasarkan peran (admin/user) untuk fitur-fitur seperti prediksi kategori?	√
2	Bagaimana sistem menampilkan, mencari, dan memfilter produk berdasarkan kategori, termasuk kemampuan untuk menampilkan produk terlaris dan memberikan saran pencarian?	√
3	Bagaimana system dapat memanfaatkan riwayat pencarian pengguna untuk mendukung proses prediksi kategori produk?	√
4	Seberapa akurat model Naïve Bayes dalam memprediksi kategori produk berdasarkan pola penjualan historis?	√
5	Apakah hasil preiksi dari metoe Naïve Bayes dapat meningkatkan relevansi rekomendasi bagi pengguna?	√

Tangerang, 25-06-2025

Pembimbing

Responden

Mahasiswa



(Rino M.kom)

NIDN : 0420058502



(Frans Vito)



(Nico Pratama)

20211000016

No	Daftar Analisa Kebutuhan Sistem	Keterangan
1	Bagaimana sistem mengelola otentikasi pengguna dan hak akses berdasarkan peran (admin/user) untuk fitur-fitur seperti prediksi kategori?	√
2	Bagaimana sistem menampilkan, mencari, dan memfilter produk berdasarkan kategori, termasuk kemampuan untuk menampilkan produk terlaris dan memberikan saran pencarian?	√
3	Bagaimana system dapat memanfaatkan riwayat pencarian pengguna untuk mendukung proses prediksi kategori produk?	√
4	Seberapa akurat model Naïve Bayes dalam memprediksi kategori produk berdasarkan pola penjualan historis?	√
5	Apakah hasil preiksi dari metoe Naïve Bayes dapat meningkatkan relevansi rekomendasi bagi pengguna?	√

Tangerang, 25-06-2025

Pembimbing

Responden

Mahasiswa



(Rino M.kom)

NIDN : 0420058502



(SUTRISNO)



(Nico Pratama)

20211000016

No	Daftar Analisa Kebutuhan Sistem	Keterangan
1	Bagaimana sistem mengelola otentikasi pengguna dan hak akses berdasarkan peran (admin/user) untuk fitur-fitur seperti prediksi kategori?	√
2	Bagaimana sistem menampilkan, mencari, dan memfilter produk berdasarkan kategori, termasuk kemampuan untuk menampilkan produk terlaris dan memberikan saran pencarian?	√
3	Bagaimana system dapat memanfaatkan riwayat pencarian pengguna untuk mendukung proses prediksi kategori produk?	√
4	Seberapa akurat model Naïve Bayes dalam memprediksi kategori produk berasarkan pola penjualan historis?	√
5	Apakah hasil preiksi dari metoe Naïve Bayes dapat meningkatkan relevansi rekomendasi bagi pengguna?	√

Tangerang, 25-06-2025

Pembimbing



(Rino M.kom)

NIDN : 0420058502

Responden



(Yogi Sarawan)

Mahasiswa



(Nico Pratama)

20211000016

LAMPIRAN 2

Pengesahan Skripsi Oleh Perusahaan PT. Kilaumas Lestari Mulia

LEMBAR PENGESAHAN SELESAI PROYEK MINOR/SKRIPSI

Dinyatakan bahwa mahasiswa:

Nama Mahasiswa : Nico Pratama
Universitas : Universitas Budhi Dharma
Nomor Induk Mahasiswa : 20211000016

Telah melaksanakan kegiatan PROYEK MINOR/SKRIPSI*) dari 03 Maret 2025 sampai dengan 06 Juni 2025 dengan baik.

Nama Instansi : PT Kilumas Lestari Mulia
Alamat : Komplek KKGJ Jl. Pipa No.15, Belimbing, Kec.
Kosambi, Kabupten Tangerang, Banten 15212

Tangerang, 25 Juni 2025


PT. KILAUMAS LESTARI MULIA
Nathania
Direktur

LAMPIRAN 3

Kartu Bimbingan

**UNIVERSITAS BUDDHI DHARMA**
Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang
021 5517853 / 021 5586822 admin@buddhidharma.ac.id

KARTU BIMBINGAN TA/SKRIPSI

NIM : 20211000016
Nama Mahasiswa : NICO PRATAMA
Fakultas : Sains dan Teknologi
Program Studi : Teknik Informatika
Jenjang : Strata Satu
Tahun Akademik/Semester : 2024/2025 Genap
Dosen Pembimbing : Rino, M.Kom
Judul Skripsi : PREDIKSI REKOMENDASI KATEGORI PRODUK MENGGUNAKAN
METODE PREDICTIVE ANALYTICS NAIVE BAYES PADA PT
KILAUMAS LESTARI MULIA

Tanggal	Catatan	Paraf
04 Maret 2025	pengajuan topik, jurnal dan panduan skripsi	
18 Maret 2025	persetujuan topik dan pengajuan bab 1 dan 2	
08 April 2025	revisi bab 1 dan 2	
22 April 2025	acc bab 1,2 dan pengajuan bab 3	
29 April 2025	revisi bab 3	
13 Mei 2025	acc bab 3 dan pengajuan bab 4	
27 Mei 2025	revisi bab 4	
03 Juni 2025	acc bab 4 dan pengajuan bab 5 serta demo program	
10 Juni 2025	acc bab 5 dan acc sidang	

Mengetahui
Ketua Program Studi



Hartana Wijaya, M.Kom

Tangerang, 21 July 2025
Pembimbing



Rino, M.Kom

LAMPIRAN 4

Codingan Aplikasi

```
import Streamlit as st
import pandas as pd
import plotly.express as px
import numpy as np
from model import load_and_train, prediksi_sampai_desember_2025
from thefuzz import process

# Definisi hubungan antar produk berdasarkan analisis sebelumnya.
# Kunci bisa berupa KATEGORI_BARANG atau NAMA_BARANG.
# Nilai adalah daftar KATEGORI_BARANG atau NAMA_BARANG yang
berhubungan.
PRODUCT_RELATIONSHIPS = {
    # Hubungan antar Kategori PVC (Pipa, Fitting, Lem)
    "AQUALON": ["LANGGENG", "MASTER", "PIPA RUCIKA", "VINILON",
"YUTA"],
    "LANGGENG": ["AQUALON", "MASTER", "PIPA RUCIKA", "VINILON",
"YUTA"],
    "MASTER": ["AQUALON", "LANGGENG", "PIPA RUCIKA", "VINILON"],
    "PIPA RUCIKA": ["AQUALON", "LANGGENG", "MASTER", "VINILON"],
    "VINILON": ["AQUALON", "LANGGENG", "MASTER", "PIPA RUCIKA"],

    # Hubungan antar Kategori PEX/PPR
    "RIFENG": ["WESTPEX", "YUTA"], # Rifeng (fitting PEX, selang fleksibel)
berhubungan dengan Westpex (pipa PEX) dan Yuta (sanitary).
    "WESTPEX": ["RIFENG"], # Westpex (pipa PEX) berhubungan dengan Rifeng
(fitting PEX).

    # Hubungan Kategori Umum Talang
    # (Aksesori spesifik Talang akan ditangani di level nama barang)
    "AQUALON_TALANG_KATEGORI_UMUM": ["LANGGENG", "MASTER"], #
Menggunakan suffix untuk membedakan kategori umum Aqualon dari kategori spesifik
talang Aqualon jika diperlukan pemisahan lebih lanjut.
    "LANGGENG_TALANG_KATEGORI_UMUM": ["AQUALON", "MASTER"],
    "MASTER_TALANG_KATEGORI_UMUM": ["AQUALON", "LANGGENG"],

    # Hubungan Spesifik Nama Barang (NAMA_BARANG -> list of NAMA_BARANG
atau KATEGORI_BARANG)

    # Lem Vinilon dan Pipa PVC
    "LEM VINILON BOTOL 60 GR": ["PIPA VINILON", "PIPA RUCIKA",
"MASTER", "LANGGENG"], # LEM VINILON ke kategori pipa PVC

    # Water Meter Cover dan Water Meter
    "WATER METER COVER (BIRU MUDA) AQUALON": ["WM DRAT+SEAL TS
1 X 3/4 LG", "WM DRAT+SEAL TS 1 X 1/2 LG", "WM DRAT TS 1 X 1/2 LG", "WM
```

POLOS TS 1/2 LG", "WM POLOS TS 3/4 LG"],
 "WATER METER COVER (BIRU TUA) AQUALON": ["WM DRAT+SEAL TS 1 X 3/4 LG", "WM DRAT+SEAL TS 1 X 1/2 LG", "WM DRAT TS 1 X 1/2 LG", "WM POLOS TS 1/2 LG", "WM POLOS TS 3/4 LG"],
 "WATER METER COVER (KUNING) AQUALON": ["WM DRAT+SEAL TS 1 X 3/4 LG", "WM DRAT+SEAL TS 1 X 1/2 LG", "WM DRAT TS 1 X 1/2 LG", "WM POLOS TS 1/2 LG", "WM POLOS TS 3/4 LG"],

Talang dan Aksesorisnya (spesifik nama barang)
 "TALANG U 6 PUTIH AQUALON": ["GANTUNGAN TALANG U 6 PUTIH AQUALON", "TUTUP TALANG U 6 PUTIH AQUALON", "CORONG TALANG U 6 PUTIH AQUALON", "SAMBUNGAN TALANG U 6 PUTIH AQUALON"],
 "TALANG RG 8 LG": ["GANTUNGAN PVC RG 8 LG", "TUTUP RG 8 LG", "CORONG RG 8 LG", "SAMBUNGAN RG 8 LG", "BELOKAN RG 8 LG"],
 "TALANG RG 6 LG": ["GANTUNGAN PVC RG 6 LG", "TUTUP RG 6 LG", "CORONG RG 6 LG", "SAMBUNGAN RG 6 LG", "BELOKAN RG 6 LG"],
 "TALANG LANGGENG SG 220": ["SAMBUNGAN SG 220 LG", "TUTUP SG 220 LG", "GANTUNGAN SG 220 LG", "CORONG SG 220 LG", "BELOKAN SG 220 LG"],
 "TALANG LANGGENG SG 140": ["GANTUNGAN SG 140 LG", "TUTUP SG 140 LG", "CORONG SG 140 LG", "SAMBUNGAN SG 140 LG", "BELOKAN SG 140 LG"],
 "TALANG MASTER PUTIH": [],
 "TALANG MASTER ABU": [],

Pipa Conduit dan Junction
 "PIPA CONDUIT RIFENG / RIIFO 20 MM": ["PIPA MASTER CONDUIT 20 MM", "PIPA CONDUIT VINILON 20 MM", "CONDUIT JUNC 1/2 x 3 LG"],
 "PIPA MASTER CONDUIT 20 MM": ["PIPA CONDUIT RIFENG / RIIFO 20 MM", "PIPA CONDUIT VINILON 20 MM", "CONDUIT JUNC 1/2 x 3 LG"],
 "PIPA CONDUIT VINILON 20 MM": ["PIPA CONDUIT RIFENG / RIIFO 20 MM", "PIPA MASTER CONDUIT 20 MM", "CONDUIT JUNC 1/2 x 3 LG"],
 "CONDUIT JUNC 1/2 x 3 LG": ["PIPA CONDUIT RIFENG / RIIFO 20 MM", "PIPA MASTER CONDUIT 20 MM", "PIPA CONDUIT VINILON 20 MM"],

Hidroponik dan Pipa PVC
 "STANDING UNIT HINDROPONIK AQUALON": ["AQUALON", "LANGGENG", "MASTER", "PIPA RUCIKA", "VINILON"], # Berhubungan dengan kategori pipa PVC secara umum

Sika (semua produk Sika berhubungan satu sama lain)
 "SIKA": ["SIKA"] # Ini adalah kasus khusus: jika produk adalah Sika, tampilkan semua produk Sika lainnya.
 }

Definisi struktur kategori dalam kategori
 # Ini adalah data baru yang akan digunakan untuk tampilan di halaman Home.
 NESTED_CATEGORIES = {
 "Pipa": {
 "KLOFF/MASTER": [

"PIPA KLOFF/MASTER D 1 1/2", "PIPA KLOFF/MASTER D 1 1/4", "PIPA KLOFF/MASTER D 2",
 "PIPA KLOFF/MASTER D 2 1/2", "PIPA KLOFF/MASTER D 3", "PIPA KLOFF/MASTER D 4",
 "PIPA KLOFF/MASTER D 5", "PIPA KLOFF/MASTER D 6", "PIPA KLOFF/MASTER D 8",
 "PIPA KLOFF/MASTER AW 1", "PIPA KLOFF/MASTER AW 1 1/2", "PIPA KLOFF/MASTER AW 1 1/4",
 "PIPA KLOFF/MASTER AW 2", "PIPA KLOFF/MASTER AW 2 1/2", "PIPA KLOFF/MASTER AW 3",
 "PIPA KLOFF/MASTER AW 3/4", "PIPA KLOFF/MASTER AW 4", "PIPA KLOFF/MASTER AW 5",
 "PIPA KLOFF/MASTER AW 6", "PIPA KLOFF/MASTER AW 8", "PIPA MASTER CONDUIT 20 MM"
],
 "LG (Langgeng)": [
 "PIPA LANGGENG C 1 1/2", "PIPA LANGGENG C 2", "PIPA LANGGENG C 3", "PIPA LANGGENG C 4",
 "PIPA LANGGENG C 5/8", "PIPA LANGGENG PUTIH D 16", "PIPA C 5/8"
],
 "MASPION": [
 "PIPA MASPION D 2 1/2"
],
 "RIFENG / RIIFO": [
 "PIPA CONDUIT RIFENG / RIIFO 20 MM", "PEX AL PEX 1/2 @ 100 MT RIFENG"
],
 "RUCIKA": [
 "PIPA RUCIKA JIS ABU D 1 1/2", "PIPA RUCIKA JIS ABU D 2", "PIPA RUCIKA JIS ABU D 3",
 "PIPA RUCIKA JIS ABU D 4", "PIPA RUCIKA JIS ABU D 5", "PIPA RUCIKA JIS ABU D 6",
 "PIPA RUCIKA JIS ABU AW 1", "PIPA RUCIKA JIS ABU AW 1 1/2", "PIPA RUCIKA JIS ABU AW 1 1/4",
 "PIPA RUCIKA JIS ABU AW 2", "PIPA RUCIKA JIS ABU AW 2 1/2", "PIPA RUCIKA JIS ABU AW 3",
 "PIPA RUCIKA JIS ABU AW 3/4", "PIPA RUCIKA JIS ABU AW 4", "PIPA RUCIKA JIS ABU AW 5",
 "PIPA RUCIKA JIS ABU AW 6", "PIPA RUCIKA JIS ABU AW 8", "PIPA RUCIKA JIS ABU AW 10",
 "PIPA RUCIKA PUTIH D 1 1/4", "PIPA RUCIKA PUTIH D 1 1/2", "PIPA RUCIKA PUTIH D 2",
 "PIPA RUCIKA PUTIH D 2 1/2", "PIPA RUCIKA PUTIH D 3", "PIPA RUCIKA PUTIH D 4",
 "PIPA RUCIKA PUTIH D 5", "PIPA RUCIKA PUTIH D 6", "PIPA RUCIKA PUTIH D 8",
 "PIPA RUCIKA PUTIH D 10", "PIPA RUCIKA PUTIH D 12", "PIPA RUCIKA PUTIH AW 1",
 "PIPA RUCIKA PUTIH AW 1 1/2", "PIPA RUCIKA PUTIH AW 1 1/4", "PIPA RUCIKA PUTIH AW 1/2",


```

    "TEE ELBOW 90 AW ABU 1 AQUALON", "TEE ELBOW 90 AW PUTIH 1
    AQUALON",
    "TEE PUTIH 1 AQUALON", "TEE PUTIH 1/2 AQUALON", "TEE PUTIH 3/4
    AQUALON"
  ],
  "LG (Langgeng)": [
    "BALL VALVE TS 1/2 LG", "BELOKAN DV 3/4 LG", "DOP DV 3 LG",
    "DOP DV 4 LG",
    "DOP DV 6 LG", "DOP DV 8 LG", "DOP SARINGAN 3 (DRAIN CAP - DV)
    LG", "KDD TS 3/4 LG",
    "KDD TS 3/4 X 1 LG", "KNEE DV 2 1/2 LG", "KNEE DV 3 LG", "KNEE DV
    4 LG",
    "SDL TS 1 X 3/4 LG", "SDL TS 1 1/2 LG", "SDL TS 1 1/4 LG", "SDL TS 1/2 X
    1 LG",
    "SDL TS 3/4 LG", "SDL TS 3/4 X 1 LG", "SDL TS JIS 1 1/4 X 1 LG", "SDD
    TS 1 1/2 LG",
    "TDD TS 3/4 X 1/2 LG", "TEE DV 3 LG", "TEE TS 3/4 LG", "VS TS 2 X 1/2
    LG",
    "VT 3/4 x 1/2 TS LG", "WM DRAT TS 1 X 1/2 LG", "WM DRAT+SEAL TS 1
    X 1/2 LG",
    "WM DRAT+SEAL TS 1 X 3/4 LG", "WM POLOS TS 1/2 LG", "WM POLOS
    TS 3/4 LG"
  ],
  "RIFENG": [
    "EQUAL STRAIGHT UNION 1216 X 1216 RIFENG", "FEMALE ELBOW
    1216 X 1/2 F RIFENG",
    "MALE ELBOW 1216 X 1/2 M RIFENG", "MALE STRAIGHT UNION (SDL)
    GREEN 20 X 1/2 RIFENG",
    "MALE UNION 1216 X 1/2 M RIFENG", "PLASTICK SOCKET (KLEM)
    GREEN 1 RIFENG",
    "PLASTICK SOCKET (KLEM) GREEN 1/2 RIFENG", "PLASTICK SOCKET
    (KLEM) GREEN 3/4 RIFENG",
    "STRAIGHT ELBOW 1216 X 1216 RIFENG", "UNEQUAL TEE 1620 X 1216
    X 1620 RIFENG"
  ]
},
"Talang Air & Aksesoris": {
  "AQUALON": [
    "CORONG TALANG U 6 PUTIH AQUALON", "GANTUNGAN TALANG U
    6 PUTIH AQUALON",
    "SAMBUNGAN TALANG U 6 PUTIH AQUALON", "TALANG U 6 PUTIH
    AQUALON", "TUTUP TALANG U 6 PUTIH AQUALON"
  ],
  "KLOFF/MASTER": [
    "TALANG MASTER ABU", "TALANG MASTER PUTIH"
  ]
},
"LG (Langgeng)": [
  "BELOKAN RG 6 LG", "BELOKAN RG 8 LG", "BELOKAN SG 140 LG",
  "BELOKAN SG 220 LG",
  "CORONG RG 6 LG", "CORONG RG 8 LG", "CORONG SG 140 LG",

```

"CORONG SG 220 LG",
 "GANTUNGAN PVC RG 6 LG", "GANTUNGAN PVC RG 8 LG",
 "GANTUNGAN SG 140 LG", "GANTUNGAN SG 220 LG",
 "SAMBUNGAN RG 6 LG", "SAMBUNGAN RG 8 LG", "SAMBUNGAN SG
 140 LG", "SAMBUNGAN SG 220 LG",
 "TALANG LANGGENG SG 140", "TALANG LANGGENG SG 220",
 "TALANG RG 6 LG", "TALANG RG 8 LG",
 "TUTUP RG 6 LG", "TUTUP RG 8 LG", "TUTUP SG 140 LG", "TUTUP SG
 220 LG"
]
 },
 "Bahan Kimia Konstruksi": {
 "SIKA": [
 "SIKA ACCELERATOR 900 ML", "SIKA ANCHORFIX 1 (300ML)", "SIKA
 ANCHORFIX 2 (300 ML)",
 "SIKA ANTISOL S (20 KG)", "SIKA BOOM AP 750 ML", "SIKA CERAM
 150 CA 25 KG",
 "SIKA CERAM 180 GA 25 KG", "SIKA CERAM 200 TA 25 KG", "SIKA
 CERAM 200 TA 5 KG",
 "SIKA CERAM 255 BIG SLAB 25 KG", "SIKA CHAPDUR EXTRA GREEN.
 25KG",
 "SIKA CHAPDUR NATURAL. 25KG", "SIKA COAT PLUS GREY. 20KG",
 "SIKA COAT PLUS GREY. 4 KG",
 "SIKA COAT PLUS STONE GREY 4 KG", "SIKA COAT PLUS STONE
 GREY. 20KG",
 "SIKA COAT PLUS WHITE 20 KG", "SIKA COAT PLUS WHITE 4 KG",
 "SIKA CRETE FLOOR LEVEL 25 KG",
 "SIKA DECKSEAL GREY 1 KG", "SIKA DECKSEAL GREY. 20KG", "SIKA
 DECKSEAL GREY. 4KG",
 "SIKA DECKSEAL WHITE. 20KG", "SIKA DECKSEAL WHITE. 4KG",
 "SIKA FLEX 140 CONST 600 ML",
 "SIKA GROUT 215 (25 KG)", "SIKA LATEX. 20 KG", "SIKA MONOTOP
 613", "SIKA MONOTOP 615 HB",
 "SIKA SHIELD P24 S 3MM", "SIKA TILE GROUT BLACK", "SIKA TILE
 GROUT BLUE",
 "SIKA TILE GROUT CREAM", "SIKA TILE GROUT GREEN", "SIKA TILE
 GROUT GREY",
 "SIKA TILE GROUT IVORY", "SIKA TILE GROUT MED BROWN. 1KG",
 "SIKA TILE GROUT RED",
 "SIKA TILE GROUT WHITE", "SIKA TILE GROUT YELLOW", "SIKA
 VISCOCRETE 3115N. 20LT",
 "SIKA VISCOCRETE 3115N. 5LT", "SIKA WATERPROOFING MORTAR. 5
 KG", "SIKABOND NV. 10KG",
 "SIKACIM BONDING (900 ML)", "SIKACIM CONCRETE 5 LT", "SIKACIM
 CONCRETE 900 ML",
 "SIKACIM LATEX 900 ML", "SIKACIM THINBED (40 KG) SIKAMUR
 130",
 "SIKACIM TILEGROUT ADDITIVE 900 ML", "SIKACRETE 08 SCC (35
 KG)",
 "SIKADUR 20 CRACK SEAL AB 160ML", "SIKADUR 20 CRACK SEAL AB

```

850ML",
    "SIKADUR 31 CF (1.2KG)", "SIKADUR 31 CF NORMAL (6 KG)",
    "SIKADUR 52 ID (1.2KG)",
    "SIKADUR 732", "SIKAFLEX 11FC GREY (310ML)", "SIKAFLEX 11FC
    WHITE (310ML)",
    "SIKAFLEX PRO 3 CONCRETE GREY 600 ML", "SIKAMENT NN 20 KG",
    "SIKASET ACCELERATOR 20 KG",
    "SIKATOP 107 PLUS (36KG)", "SIKATOP 107 PLUS (4 KG)", "SIKATOP
    107 SEAL SET (25 KG)",
    "SIKATOP 121", "SIKATOP LIQUID", "SIKAWALL 161 PLASTERMIX 40
    KG",
    "SIKAWALL 181 SKIMCOAT (40 KG)"
    ],
    },
    "Perlengkapan Sanitasi & Kran": {
        "YUTA": [
            "ANGLE VALVE AGSD 1/2 YUTA", "BIBCOCK ABS BAMHW 1/2 YUTA",
            "BIDET SET STF-W (WHITE) 1/2 YUTA", "FLOAT VALVE PLASTIC BALL
            YBF 1 YUTA",
            "FLOAT VALVE PLASTIC BALL YBF 1/2 YUTA", "FLOAT VALVE
            PLASTIC BALL YBF 3/4 YUTA",
            "FLOAT VALVE STAINLESS 1 1/2 YUTA", "FLOAT VALVE STAINLESS 2
            YUTA",
            "FOOT VALVE 1 ( PFV ) YUTA", "FOOT VALVE 3/4 ( PFV ) YUTA",
            "JET SHOWER HEAD STH - CHROME 1/2 YUTA", "JET SHOWER HEAD
            STH - W 1/2 YUTA",
            "MEDIUM BIBCOCK BMLF 1/2 YUTA", "SINK TAP TSBF 1/2 YUTA",
            "SINK TAP TSF 1/2 YUTA"
        ],
        },
        "Lain-lain": {
            "AQUALON": [
                "STANDING UNIT HINDROPONIK AQUALON", "WATER METER
                COVER (BIRU MUDA) AQUALON",
                "WATER METER COVER (BIRU TUA) AQUALON", "WATER METER
                COVER (KUNING) AQUALON"
            ],
            "LG (Langgeng)": [
                "CONDUIT JUNC 1/2 x 3 LG"
            ],
            "RIFENG": [
                "BALL VALVE PPR 1 RIFENG", "BALL VALVE PPR 1/2 RIFENG", "BALL
                VALVE PPR 3/4 RIFENG",
                "FLEXIBLE HOSE RF-F100 RIFENG", "FLEXIBLE HOSE RF-F20 RIFENG",
                "FLEXIBLE HOSE RF-F30 RIFENG", "FLEXIBLE HOSE RF-F60 RIFENG"
            ],
            "VINILON": [
                "LEM VINILON BOTOL 60 GR"
            ],
        ],
    },
}

```

```

}

# ==== LOGIN SETUP ====
USERS = {
    "admin": {"password": "admin123", "role": "admin"},
    "user": {"password": "user123", "role": "user"}
}

st.set_page_config(page_title="Rekomendasi Produk", layout="wide")

# Inisialisasi session state
if "logged_in" not in st.session_state:
    st.session_state.logged_in = False
    st.session_state.role = None
    st.session_state.username = None
if "produk_id_terakhir_diklik" not in st.session_state:
    st.session_state.produk_id_terakhir_diklik = None

def try_login(username, password):
    if username in USERS and USERS[username]["password"] == password:
        st.session_state.logged_in = True
        st.session_state.role = USERS[username]["role"]
        st.session_state.username = username
        st.success(f"Berhasil login sebagai {username} ({st.session_state.role})")
    else:
        st.session_state.login_failed = True

def login_form():
    st.title("\U0001F510 Login Sistem Rekomendasi Produk")
    with st.form("login_form"):
        username = st.text_input("Username")
        password = st.text_input("Password", type="password")
        submitted = st.form_submit_button("Login")
        if submitted:
            try_login(username, password)

    if st.session_state.get("login_failed", False):
        st.error("Username atau password salah.")
    if st.session_state.get("logout_success", False):
        st.success("Anda berhasil logout.")
        st.session_state.logout_success = False

# Jika belum login
if not st.session_state.logged_in:
    login_form()
    st.stop()

# ==== APLIKASI UTAMA ====
st.title("\U0001F6D2 Sistem Rekomendasi Produk PT. KILAUMAS LESTARI")

```

```

MULIA")

if "kategori_history" not in st.session_state:
    st.session_state.kategori_history = []
if "kategori_rekomendasi" not in st.session_state:
    st.session_state.kategori_rekomendasi = []
if "terakhir_diklik" not in st.session_state:
    st.session_state.terakhir_diklik = None

model, bulan_cols, df = load_and_train()
df['TOTAL_PENJUALAN'] = df[bulan_cols].sum(axis=1)

params = st.query_params
kategori_param = params.get("kategori")
produk_id_param = params.get("produk_id")

menu = st.sidebar.radio("\U0001F4CC Navigasi", ["\U0001F3E0 Home", "\U0001F50D
Cari Produk", "\U0001F3AF Prediksi Kategori"])

with st.sidebar:
    st.markdown("----")
    st.markdown(f"\U0001F464 Login sebagai: `{st.session_state.username}`
({st.session_state.role})")
    if st.button("Logout"):
        st.session_state.logged_in = False
        st.session_state.username = None
        st.session_state.role = None
        st.session_state.logout_success = True
        st.rerun()

# Fungsi tampilkan produk
def tampilkan_produk(dataframe, kolom_per_baris=3):
    rows = dataframe.iterrows()
    while True:
        cols = st.columns(kolom_per_baris)
        for i in range(kolom_per_baris):
            try:
                idx, row = next(rows)
                with cols[i]:
                    button_key = f"produk_{row['NAMA_BARANG']}_{idx}"
                    if st.button(f"{row['NAMA_BARANG']}", key=button_key):
                        st.query_params.update(produk_id=idx)
                        st.session_state.terakhir_diklik = row['KATEGORI_BARANG']
                        st.session_state.produk_id_terakhir_diklik = idx
                        st.rerun()

            st.markdown(f"""
                <div style="border: 1px solid #e0e0e0; border-radius: 10px; padding:
10px;
                background-color: #fff8f0; text-align: center; height: 150px;

```

```

        box-shadow: 2px 2px 8px rgba(0,0,0,0.1); transition: 0.3s;">
        <p style="color:#555; margin: 4px 0;">
            <strong>Kategori:</strong> <span
style="color:#2980b9;">{row['KATEGORI_BARANG']}</span>
        </p>
        <p style="color:#555; margin: 4px 0;"><strong>Penjualan:</strong>
{int(row['TOTAL_PENJUALAN'])}</p>
        <p style="color:#555; margin: 4px 0;"><strong>Satuan:</strong>
{row['SATUAN']}</p>
    </div>
    """, unsafe_allow_html=True)
except StopIteration:
    return

def get_related_products(clicked_product_name, clicked_product_category,
df_all_products):
    related_targets = []

    # 1. Cek hubungan berdasarkan NAMA_BARANG yang spesifik
    if clicked_product_name in PRODUCT_RELATIONSHIPS:
        related_targets.extend(PRODUCT_RELATIONSHIPS[clicked_product_name])

    # 2. Cek hubungan berdasarkan KATEGORI_BARANG
    if clicked_product_category in PRODUCT_RELATIONSHIPS:
        # Penanganan khusus untuk kategori SIKA: semua produk Sika saling berhubungan
        if clicked_product_category == "SIKA":
            return df_all_products[
                (df_all_products['KATEGORI_BARANG'] == "SIKA") &
                (df_all_products['NAMA_BARANG'] != clicked_product_name)
            ].copy()
        else:
            related_targets.extend(PRODUCT_RELATIONSHIPS[clicked_product_category])

    # Penanganan untuk kategori talang umum
    if "TALANG" in clicked_product_category: # Jika kategori produk mengandung kata
"TALANG"
        if "AQUALON" in clicked_product_category and
"AQUALON_TALANG_KATEGORI_UMUM" in PRODUCT_RELATIONSHIPS:
            related_targets.extend(PRODUCT_RELATIONSHIPS["AQUALON_TALANG_KATE
GORI_UMUM"])
        elif "LANGGENG" in clicked_product_category and
"LANGGENG_TALANG_KATEGORI_UMUM" in PRODUCT_RELATIONSHIPS:
            related_targets.extend(PRODUCT_RELATIONSHIPS["LANGGENG_TALANG_KAT
EGORI_UMUM"])
        elif "MASTER" in clicked_product_category and
"MASTER_TALANG_KATEGORI_UMUM" in PRODUCT_RELATIONSHIPS:

```

```

related_targets.extend(PRODUCT_RELATIONSHIPS["MASTER_TALANG_KATEGORI_UMUM"])

# Hapus duplikasi dari daftar target yang berhubungan
related_targets = list(set(related_targets))

if not related_targets:
    return pd.DataFrame() # Tidak ada produk berhubungan yang ditemukan

# Filter df_all_products berdasarkan target yang ditemukan
# Target bisa berupa NAMA_BARANG atau KATEGORI_BARANG

mask_by_name = df_all_products['NAMA_BARANG'].isin(related_targets)
mask_by_category = df_all_products['KATEGORI_BARANG'].isin(related_targets)

# Gabungkan mask
filtered_df = df_all_products[mask_by_name | mask_by_category].copy()

# Pastikan produk yang diklik itu sendiri tidak termasuk dalam daftar yang berhubungan
filtered_df = filtered_df[filtered_df['NAMA_BARANG'] != clicked_product_name].copy()

return filtered_df

# === HOME ===
if menu == "\U0001F3E0 Home":
    st.subheader("\U0001F4C1 Kategori Produk Berjenjang")

    # Tampilkan kategori utama, lalu merek, lalu produk
    for main_category, brands_dict in NESTED_CATEGORIES.items():
        st.markdown(f"### \U0001F539 {main_category}")

        # Iterasi merek di bawah setiap kategori utama
        for brand, products_list in brands_dict.items():
            with st.expander(f"\U0001F4A1 **{brand}**"):
                cols = st.columns(3) # Tampilkan 3 produk per baris
                for i, product_name in enumerate(products_list):
                    with cols[i % 3]:
                        # Cari produk di DataFrame berdasarkan nama barang
                        # Menggunakan .copy() untuk menghindari SettingWithCopyWarning
                        product_data = df[df['NAMA_BARANG'] == product_name].copy()

                        if not product_data.empty:
                            # Ambil data produk pertama yang cocok (jika ada duplikat)
                            row = product_data.iloc[0]
                            idx = product_data.index[0] # Ambil index produk untuk produk_id

                            button_key = f"produk_{product_name}_{idx}"

```

```

        if st.button(f"{product_name}", key=button_key):
            st.query_params.update(produk_id=idx)
            st.session_state.terakhir_diklik = row['KATEGORI_BARANG'] #
Simpan kategori merek
            st.session_state.produk_id_terakhir_diklik = idx
            st.rerun()

        st.markdown(f"""
padding: 10px;
            <div style="border: 1px solid #e0e0e0; border-radius: 10px;
                background-color: #fff8f0; text-align: center; height: 150px;
                box-shadow: 2px 2px 8px rgba(0,0,0,0.1); transition: 0.3s;">
                <p style="color:#555; margin: 4px 0;">
                    <strong>Kategori:</strong> <span
style="color:#2980b9;">{row['KATEGORI_BARANG']}</span>
                </p>
                <p style="color:#555; margin: 4px
0;"><strong>Penjualan:</strong> {int(row['TOTAL_PENJUALAN'])}</p>
                <p style="color:#555; margin: 4px 0;"><strong>Satuan:</strong>
{row['SATUAN']}</p>
                </div>
            """, unsafe_allow_html=True)
        else:
            st.info(f"Produk '{product_name}' tidak ditemukan di dataset.")
            st.markdown("---") # Garis pemisah antar kategori utama

# Logika tampilan detail produk atau produk terlaris setelah kategori utama
ditampilkan
if produk_id_param:
    try:
        produk_id = int(produk_id_param)
        clicked_product = df.loc[produk_id]
        clicked_product_name = clicked_product["NAMA_BARANG"]
        clicked_product_category = clicked_product["KATEGORI_BARANG"]

        st.subheader(f"\U0001F4E6 Detail Produk: **{clicked_product_name}**")
        st.markdown(f"""
            <div style="border: 1px solid #e0e0e0; border-radius: 10px; padding: 15px;
                background-color: #e6f7ff; text-align: center; margin-bottom: 20px;
                box-shadow: 2px 2px 8px rgba(0,0,0,0.1);">
                <h3 style="color: #FF5733;">{clicked_product_name}</h3>
                <p style="color:#888888; margin: 5px 0;">
                    <strong>Kategori:</strong> <span
style="color:#007BFF;">{clicked_product_category}</span>
                </p>
                <p style="color:#888888; margin: 5px 0;"><strong>Total
Penjualan:</strong> {int(clicked_product['TOTAL_PENJUALAN'])}</p>
                <p style="color:#888888; margin: 5px 0;"><strong>Satuan:</strong>
{clicked_product['SATUAN']}</p>
            </div>
        """)

```

```

"""', unsafe_allow_html=True)

# Mendapatkan dan menampilkan produk-produk yang berhubungan
related_products_df = get_related_products(clicked_product_name,
clicked_product_category, df)

if not related_products_df.empty:
    st.subheader("\U0001F4B5 Produk-Produk Rekomendasi")
    tampilkan_produk(related_products_df)
else:
    st.info(f'Tidak ada produk lain yang berhubungan dengan
**{clicked_product_name}** dalam data yang tersedia.')

# Tombol untuk menyembunyikan detail produk
if st.button("Sembunyikan Detail Produk"):
    st.query_params.clear()
    st.session_state.produk_id_terakhir_diklik = None
    st.rerun()

except KeyError:
    st.error("ID produk tidak valid atau produk tidak ditemukan.")
    st.query_params.clear()
    st.session_state.produk_id_terakhir_diklik = None
    st.rerun()
except Exception as e:
    st.error(f'Terjadi kesalahan saat menampilkan detail produk atau produk
berhubungan: {e}')
    st.query_params.clear()
    st.session_state.produk_id_terakhir_diklik = None
    st.rerun()

elif not produk_id_param and not kategori_param: # Tampilkan terlaris hanya jika
tidak ada param aktif
    st.subheader("\U0001F525 Daftar Produk Terlaris")
    produk_terlaris = df.sort_values(by='TOTAL_PENJUALAN', ascending=False)
    tampilkan_produk(produk_terlaris.head(12))

# === CARI ===
elif menu == "\U0001F50D Cari Produk":
    st.subheader("\U0001F50E Cari Produk atau Kategori")

    # Kotak pencarian selalu ditampilkan di bagian atas
    search_query = st.text_input("Ketik nama produk atau kategori:")

    # Prioritas 1: Tampilkan detail produk dan produk berhubungan jika produk_id_param
aktif
    # Ini akan mengambil alih tampilan pencarian jika ada produk yang dipilih
    if produk_id_param:
        try:

```

```

produk_id = int(produk_id_param)
clicked_product = df.loc[produk_id]
clicked_product_name = clicked_product["NAMA_BARANG"]
clicked_product_category = clicked_product["KATEGORI_BARANG"]

st.subheader(f"\U0001F4E6 Detail Produk: **{clicked_product_name}**")
# Markup untuk detail produk
st.markdown(f"""
    <div style="border: 1px solid #e0e0e0; border-radius: 10px; padding: 15px;
        background-color: #e6f7ff; text-align: center; margin-bottom: 20px;
        box-shadow: 2px 2px 8px rgba(0,0,0,0.1);">
        <h3 style="color: #FF5733;">{clicked_product_name}</h3>
        <p style="color: #888888; margin: 5px 0;">
            <strong>Kategori:</strong> <span
style="color: #007BFF;">{clicked_product_category}</span>
        </p>
        <p style="color: #888888; margin: 5px 0;"><strong>Total
Penjualan:</strong> {int(clicked_product['TOTAL_PENJUALAN'])}</p>
        <p style="color: #888888; margin: 5px 0;"><strong>Satuan:</strong>
{clicked_product['SATUAN']}</p>
    </div>
    """, unsafe_allow_html=True)

# Mendapatkan dan menampilkan produk-produk yang berhubungan
related_products_df = get_related_products(clicked_product_name,
clicked_product_category, df)

if not related_products_df.empty:
    st.subheader("\U0001F4B5 Produk-Produk Rekomendasi")
    tampilkan_produk(related_products_df)
else:
    st.info(f"Tidak ada produk lain yang berhubungan dengan
    **{clicked_product_name}** dalam data yang tersedia.")

# Tombol untuk menyembunyikan detail produk dan kembali ke tampilan
pencarian umum
if st.button("Sembunyikan Detail Produk"):
    st.query_params.clear() # Hapus semua query params (termasuk produk_id)
    st.session_state.produk_id_terakhir_diklik = None # Hapus juga dari session
state

    st.rerun() # Muat ulang aplikasi

# Hentikan eksekusi lebih lanjut di halaman ini untuk run saat ini
# agar tidak menampilkan hasil pencarian di bawah detail produk
st.stop()

except KeyError:
    st.error("ID produk tidak valid atau produk tidak ditemukan.")
    st.query_params.clear() # Hapus parameter yang tidak valid
    st.session_state.produk_id_terakhir_diklik = None

```

```

        st.rerun() # Muat ulang
    except Exception as e:
        st.error(f"Terjadi kesalahan saat menampilkan detail produk atau produk
berhubungan: {e}")
        st.query_params.clear()
        st.session_state.produk_id_terakhir_diklik = None
        st.rerun() # Muat ulang

# Prioritas 2: Tampilkan hasil pencarian jika kotak pencarian berisi teks (dan
produk_id_param tidak aktif)
if search_query:
    hasil = df[df['NAMA_BARANG'].str.contains(search_query, case=False, na=False)
|
df['KATEGORI_BARANG'].str.contains(search_query, case=False,
na=False)]

    if hasil.empty:
        semua_produk = df['NAMA_BARANG'].tolist() +
df['KATEGORI_BARANG'].tolist()
        suggestion, score = process.extractOne(search_query, semua_produk)
        st.warning(f"Tidak ditemukan hasil untuk: **{search_query}**")
        if score > 60:
            st.info(f"Mungkin maksud Anda: **{suggestion}**")
            hasil = df[df['NAMA_BARANG'].str.contains(suggestion, case=False,
na=False) |
df['KATEGORI_BARANG'].str.contains(suggestion, case=False,
na=False)]

        if not hasil.empty:
            kategori_teratas = hasil['KATEGORI_BARANG'].mode()[0]
            if kategori_teratas in st.session_state.kategori_rekomendasi:
                st.session_state.kategori_rekomendasi.remove(kategori_teratas)
                st.session_state.kategori_rekomendasi.append(kategori_teratas)
                st.session_state.terakhir_diklik = kategori_teratas
            if len(st.session_state.kategori_rekomendasi) > 5:
                st.session_state.kategori_rekomendasi.pop(0)

            st.success(f"Menampilkan hasil untuk: **{search_query}**")
            tampilkan_produk(hasil)

# Prioritas 3: Tampilkan produk terlaris jika tidak ada pencarian aktif dan tidak ada
detail produk yang ditampilkan
else: # Ini adalah 'else' dari `if search_query:`
    st.info("Tampilkan produk terlaris karena belum ada input pencarian atau detail
produk disembunyikan.")
    tampilkan_produk(df.sort_values(by='TOTAL_PENJUALAN',
ascending=False).head(10))

# === PREDIKSI ===
elif menu == "\U0001F3AF Prediksi Kategori":

```

```

st.subheader("\U0001F52E Prediksi Kategori Berdasarkan Pencarian")

if st.session_state.role != "admin":
    st.warning("Fitur ini hanya tersedia untuk admin.")
    st.stop()

if not st.session_state.kategori_rekomendasi:
    st.info("Belum ada pencarian dilakukan. Silakan lakukan pencarian di menu 'Cari Produk' terlebih dahulu.")
else:
    # Gunakan kategori terakhir yang dicari/diklik sebagai "nama produk" dalam pesan akhir
    # Ini akan menjadi kategori barang yang terakhir diinteraksi pengguna.
    nama_produk_terpilih = st.session_state.terakhir_diklik if
st.session_state.terakhir_diklik else "Produk Terpilih"

    st.info(f"Melakukan prediksi untuk kategori yang terakhir dicari: **{'',
'.join(st.session_state.kategori_rekomendasi)}**")

    # Filter products based on categories in session state
    df_predict_target =
df[df["KATEGORI_BARANG"].isin(st.session_state.kategori_rekomendasi)].copy()

    if df_predict_target.empty:
        st.warning("Tidak ada data produk yang cocok untuk kategori yang dicari.")
    else:
        # Hitung total penjualan dan label untuk setiap produk dalam grup yang difilter
        df_predict_target["TOTAL_PENJUALAN"] =
df_predict_target[bulan_cols].sum(axis=1)

        # Gunakan median dari grup yang difilter untuk mendefinisikan "Tinggi" vs
        "Rendah"
        median_penjualan_group = df_predict_target["TOTAL_PENJUALAN"].median()
        df_predict_target["LABEL_PENJUALAN"] =
(df_predict_target["TOTAL_PENJUALAN"] > median_penjualan_group).astype(int)

        # --- Hitung Parameter Naive Bayes dari grup yang difilter ---
        # Parameter ini merepresentasikan karakteristik penjualan "Tinggi" dan
        "Rendah"
        # di antara produk-produk dalam kategori yang sedang dipilih.

        # Pastikan ada sampel untuk penjualan tinggi dan rendah sebelum menghitung
        rata-rata/varians
        if 1 in df_predict_target["LABEL_PENJUALAN"].unique():
            mean_tinggi = df_predict_target[df_predict_target["LABEL_PENJUALAN"]
== 1][bulan_cols].mean().values
            # Menambahkan 1e-6 untuk mencegah masalah varians nol
            var_tinggi =
np.clip(df_predict_target[df_predict_target["LABEL_PENJUALAN"] ==
1][bulan_cols].var().values, 1e-6, None)

```

```

    prior_tinggi = df_predict_target['LABEL_PENJUALAN'].mean()
else: # Tidak ada penjualan tinggi di grup yang difilter
    mean_tinggi = np.zeros(len(bulan_cols))
    var_tinggi = np.ones(len(bulan_cols)) * 1e-6
    prior_tinggi = 1e-6 # Prior yang sangat rendah untuk "tinggi" jika tidak
ditemukan penjualan tinggi

    if 0 in df_predict_target['LABEL_PENJUALAN'].unique():
        mean_rendah = df_predict_target[df_predict_target['LABEL_PENJUALAN']
== 0][bulan_cols].mean().values
        # Menambahkan 1e-6 untuk mencegah masalah varians nol
        var_rendah =
np.clip(df_predict_target[df_predict_target['LABEL_PENJUALAN'] ==
0][bulan_cols].var().values, 1e-6, None)
        prior_rendah = 1 - prior_tinggi
    else: # Tidak ada penjualan rendah di grup yang difilter
        mean_rendah = np.zeros(len(bulan_cols))
        var_rendah = np.ones(len(bulan_cols)) * 1e-6
        prior_rendah = 1e-6 # Prior yang sangat rendah untuk "rendah" jika tidak
ditemukan penjualan rendah

# Normalisasi prior jika diperlukan (seharusnya berjumlah 1)
total_priors = prior_tinggi + prior_rendah
if total_priors > 0:
    prior_tinggi /= total_priors
    prior_rendah /= total_priors
else: # Seharusnya tidak terjadi jika df_predict_target tidak kosong
    prior_tinggi = 0.5
    prior_rendah = 0.5

# Fungsi log Gaussian untuk stabilitas
def gaussian_log_prob(x, mean, var):
    coeff = -0.5 * np.log(2.0 * np.pi * var)
    exponent = -((x - mean) ** 2) / (2 * var)
    return coeff + exponent

# Menggunakan rata-rata penjualan dari seluruh grup yang difilter untuk prediksi
kategori grup
x_input_group = df_predict_target[bulan_cols].mean().values

log_likelihood_tinggi = np.sum(gaussian_log_prob(x_input_group, mean_tinggi,
var_tinggi))
log_likelihood_rendah = np.sum(gaussian_log_prob(x_input_group,
mean_rendah, var_rendah))

# Menambahkan 1e-10 pada log(prior) untuk stabilitas numerik
log_posterior_tinggi = np.log(prior_tinggi + 1e-10) + log_likelihood_tinggi
log_posterior_rendah = np.log(prior_rendah + 1e-10) + log_likelihood_rendah

# Untuk stabilitas numerik: kurangi log-posterior maksimum sebelum

```

eksponensiasi

```
max_log_posterior = max(log_posterior_tinggi, log_posterior_rendah)
prob_tinggi = np.exp(log_posterior_tinggi - max_log_posterior)
prob_rendah = np.exp(log_posterior_rendah - max_log_posterior)

total_prob = prob_tinggi + prob_rendah

# Normalisasi probabilitas
if total_prob > 0:
    prob_tinggi /= total_prob
    prob_rendah /= total_prob
else: # Fallback jika probabilitas berjumlah nol (sangat jarang terjadi)
    prob_tinggi = 0.5
    prob_rendah = 0.5

kategori_prediksi = "TINGGI" if prob_tinggi > prob_rendah else "RENDAH"

# --- Bagian output yang dimodifikasi ---
st.markdown("### \U0001F4CA Hasil Perhitungan Naive Bayes Manual")
st.write(f"Prior kelas tinggi: **{prior_tinggi:.4f}**")
st.write(f"Prior kelas rendah: **{prior_rendah:.4f}**")
st.write(f"Posterior kelas tinggi: **{prob_tinggi:.4f}**")
st.write(f"Posterior kelas rendah: **{prob_rendah:.4f}**")

# Pesan prediksi akhir yang mencantumkan nama_produk_terpilih
if kategori_prediksi == "TINGGI":
    st.success(f"Kategori penjualan produk **{nama_produk_terpilih}**  
diprediksi: **{kategori_prediksi}**")
else:
    st.info(f"Kategori penjualan produk **{nama_produk_terpilih}** diprediksi:  
**{kategori_prediksi}**")

st.markdown("---")
st.markdown(f"***Catatan:** Pertimbangkan penambahan stok jika prediksi  
penjualan tinggi untuk memenuhi permintaan, atau optimalisasi stok jika prediksi rendah  
guna mencegah penumpukan dan mengurangi biaya.")
```

