

## **BAB III**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Selama menjalani program magang di PT XYZ Finance, penulis ditempatkan pada unit Data Analyst Officer yang berada di bawah Departemen Data Mining, sebagai bagian dari Divisi Customer Relationship Management (CRM) and Digital. Divisi ini berada di bawah tanggung jawab langsung dari Director in Charge, sebagaimana tercantum dalam struktur organisasi perusahaan.

Penulis secara khusus ditempatkan untuk menangani analisis data pelanggan pada lini produk XYZ Multipurpose, dengan fokus utama pada proyek segmentasi pelanggan aktif. Proyek ini bertujuan untuk mengelompokkan pelanggan berdasarkan karakteristik transaksional guna mendukung efektivitas strategi promosi yang lebih terarah.

Koordinasi dilakukan secara langsung dengan Department Head Data Mining sebagai atasan teknis selama magang. Bentuk koordinasi yang dilakukan bersifat rutin dan berlangsung setiap kali penulis menyelesaikan atau mencapai tahapan tertentu dalam proses kerja, seperti setelah menyusun data awal, menyelesaikan proses segmentasi, atau menghasilkan visualisasi hasil. Setiap pencapaian pada tahapan tersebut dilaporkan untuk kemudian diberikan umpan balik atau arahan tindak lanjut.

Selain itu, pihak Human Capital juga secara berkala melakukan monitoring terhadap kinerja dan kedisiplinan penulis selama program magang berlangsung. Hal ini dilakukan sebagai bagian dari proses pembinaan dan evaluasi perkembangan peserta magang di lingkungan kerja perusahaan.

Melalui sistem koordinasi tersebut, penulis memperoleh pemahaman mengenai tata kelola kerja di lingkungan korporat, sekaligus membangun keterampilan komunikasi profesional dalam menyampaikan hasil kerja secara sistematis kepada atasan langsung.

### 3.2 Tugas dan Uraian Kerja Magang

Selama menjalani program magang di PT XYZ Finance, penulis ditempatkan pada tim CRM Data Mining yang berada di bawah Divisi Strategi dan Pengelolaan Data Pelanggan. Tim ini memiliki tanggung jawab utama dalam mengelola data pelanggan secara strategis untuk keperluan analitik, segmentasi, dan otomatisasi distribusi data ke unit-unit bisnis yang relevan. Kontribusi penulis difokuskan pada lima proyek utama berbasis *data analytics pipeline*, mulai dari segmentasi pelanggan hingga pengembangan aplikasi internal. Seluruh proyek dilaksanakan secara kolaboratif dengan supervisi dari analis data dan kepala departemen. Berikut merupakan rincian kegiatan yang telah dilakukan selama masa magang;

Tabel 3. 1 Detail Linimasa Kegiatan Magang

No	Kegiatan	Tanggal Mulai	Tanggal Selesai
<b>Segmentasi Pelanggan XYZ Finance MPF</b>			
1	Pengumpulan, pembersihan, dan validasi data	3 Maret 2025	10 Maret 2025
2	Analisis segmentasi menggunakan RFM dan K-Means	11 Maret 2025	31 Maret 2025
3	Update kepada department head untuk mendapatkan feedback	1 April 2025	4 April 2025
4	Validasi hasil segmentasi dan backtesting	21 April 2025	25 April 2025
5	Transfer knowledge ke tim CRM dan Event	1 Mei 2025	2 Mei 2025
<b>Pengembangan Aplikasi Web Segmentasi</b>			
6	Desain UI/UX dan struktur modul	5 Mei 2025	9 Mei 2025
7	Implementasi modul upload, EDA, segmentasi, dan export	12 Mei 2025	16 Mei 2025
8	Pembuatan dashboard dan distribusi interaktif	19 Mei 2025	23 Mei 2025
9	Uji coba aplikasi dan dokumentasi teknis	2 Juni 2025	6 Juni 2025
<b>Sistem Otomatisasi Distribusi Email</b>			
10	Perancangan arsitektur & format template email	26 Mei 2025	30 Mei 2025
11	Implementasi sistem parsing & distribusi otomatis	2 Juni 2025	6 Juni 2025
12	Validasi cabang & error handling	9 Juni 2025	13 Juni 2025
<b>Pipeline Data Merge Telesales</b>			
13	Perancangan UI dan alur pipeline	2 Juni 2025	6 Juni 2025
14	Pengembangan fitur unggah multi-file, penggabungan, dan penyaringan	9 Juni 2025	20 Juni 2025
15	Finalisasi, dokumentasi, dan pelatihan pengguna	23 Juni 2025	30 Juni 2025

No	Kegiatan	Tanggal Mulai	Tanggal Selesai
Aplikasi Validasi Data KK			
16	Analisis kebutuhan dan struktur data	2 Juni 2025	6 Juni 2025
17	Pengembangan modul validasi KK, NIK, dan fuzzy matching	9 Juni 2025	20 Juni 2025
18	Uji coba aplikasi dan dashboard ringkasan validasi	23 Juni 2025	27 Juni 2025

Seluruh kegiatan magang dilaksanakan secara bertahap, dimulai dari pengenalan lingkungan kerja hingga pelaksanaan lima proyek utama yang mendukung pengambilan keputusan bisnis berbasis data. Kegiatan ini mencakup proses analisis data pelanggan, pengembangan sistem segmentasi, pembuatan dashboard interaktif, otomasi distribusi data leads ke seluruh area operasional, serta validasi dan penggabungan data internal untuk kebutuhan tim telesales dan instansi pengguna lainnya.

Seluruh proyek dirancang untuk menjawab kebutuhan bisnis di divisi CRM, dengan pendekatan *data-driven decision making* yang diterapkan secara sistematis dan kolaboratif. Setiap hasil pengembangan dievaluasi secara berkala melalui diskusi bersama mentor dan tim, guna memastikan akurasi data, efektivitas model, dan kesesuaian implementasi dengan standar perusahaan.

Dukungan berbagai *tools* yang digunakan dalam tiap tahapan kerja. Setiap *tool* memiliki fungsi yang spesifik untuk menunjang kelancaran proses analisis data, pengembangan aplikasi, dan otomasi sistem yang dilakukan. Penggunaan *tools* tersebut menjadi bagian integral dalam strategi penyelesaian tugas, mulai dari eksplorasi data, pemrograman, debugging, visualisasi, hingga dokumentasi.

Adapun rincian *tools* yang digunakan oleh penulis selama pelaksanaan magang akan dijelaskan secara terstruktur pada bagian berikut, lengkap dengan visualisasi logo serta deskripsi fungsional yang relevan dalam konteks implementasi proyek di PT XYZ Finance. Berikut merupakan *tools* yang dipakai untuk mendukung kegiatan magang penulis;

### a) Python

Python merupakan salah satu bahasa pemrograman tingkat tinggi yang dirancang dengan sintaks yang sederhana dan mudah dipahami, sehingga banyak digunakan dalam bidang data science, machine learning, dan pengembangan aplikasi berbasis data [11]. Dalam konteks magang di PT XYZ Finance, Python digunakan sebagai tulang punggung dalam pelaksanaan berbagai proyek berbasis analitik, antara lain membangun sistem segmentasi pelanggan dengan metode Recency, Frequency, Monetary (RFM), menerapkan algoritma K-Means Clustering, serta membangun dashboard analitik interaktif menggunakan Streamlit. Selain itu, Python juga digunakan untuk mengembangkan pipeline otomatis dalam validasi dan penggabungan data internal, termasuk dalam aplikasi validasi data kependudukan yang memanfaatkan teknik fuzzy matching. Kemampuan Python dalam mengelola data dalam skala besar serta integrasinya dengan berbagai pustaka menjadikannya sebagai tools utama dalam pelaksanaan proyek magang ini.

Gambar 3.1 berikut menampilkan logo Python yang menjadi salah satu tools utama dalam pengembangan sistem berbasis data.



Gambar 3. 1 Logo Python [12]

### b) Streamlit

Streamlit merupakan open-source framework berbasis Python yang dirancang khusus untuk membangun antarmuka pengguna (user interface) interaktif dalam bentuk aplikasi web untuk data science dan analisis data [13]. Keunggulan utama Streamlit terletak pada kemudahan implementasi serta integrasinya yang langsung dengan pustaka Python seperti Pandas, NumPy, dan Matplotlib, sehingga memungkinkan para analis dan ilmuwan data untuk menyajikan hasil analisis tanpa perlu keahlian

pengembangan web tingkat lanjut. Framework ini sangat ideal untuk membuat prototipe visualisasi data yang cepat dan responsif.

Dalam proyek magang di PT XYZ Finance, Streamlit digunakan sebagai kerangka kerja utama dalam pengembangan dua aplikasi utama, yaitu Web Segmentasi Pelanggan dan Pipeline Data Merge. Aplikasi pertama menyajikan hasil segmentasi pelanggan dalam bentuk dashboard interaktif, sedangkan aplikasi kedua mengotomatisasi proses penggabungan file telesales dari berbagai wilayah. Selain itu, Streamlit juga diterapkan pada aplikasi validasi data kependudukan yang dilengkapi fitur fuzzy matching, progress tracking, serta output hasil dalam format terstruktur dan visual. Pemanfaatan Streamlit secara strategis mempermudah diseminasi hasil analitik ke unit operasional secara real-time dan intuitif.

Gambar 3.2 berikut menampilkan logo Streamlit yang digunakan sebagai tools utama dalam pengembangan antarmuka aplikasi selama magang berlangsung.



Gambar 3. 2 Logo Streamlit [14]

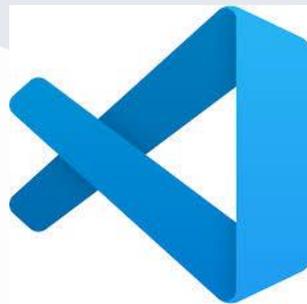
### c) Visual Studio Code (VS Code)

Visual Studio Code (VS Code) merupakan code editor ringan namun kaya fitur yang dikembangkan oleh Microsoft dan banyak digunakan oleh pengembang perangkat lunak serta analis data karena mendukung berbagai bahasa pemrograman dan integrasi ekstensi yang luas [15]. Editor ini mendukung linting, debugging, version control, serta

intelligent code completion (IntelliSense), yang sangat membantu dalam proses pengembangan skrip yang kompleks dan kolaboratif.

Dalam pelaksanaan magang di PT XYZ Finance, VS Code digunakan sebagai integrated development environment (IDE) utama untuk menulis dan menjalankan skrip Python dalam berbagai proyek seperti segmentasi pelanggan, validasi data, dan pipeline otomatisasi. Penggunaan VS Code juga mempermudah integrasi dengan GitHub, sehingga proses versioning, collaboration, dan pengelolaan kode menjadi lebih efisien selama pengembangan berlangsung. Kemudahan navigasi, pengaturan environment, serta plugin pendukung seperti Python Extension dan Jupyter Notebook Support menjadi alasan utama pemilihannya sebagai tools pengembangan utama.

Gambar 3.3 berikut menampilkan logo Visual Studio Code yang digunakan sebagai editor utama dalam pelaksanaan tugas-tugas pengembangan selama magang.



Gambar 3. 3 Logo Visual Studio Code [16]

#### **d) Github**

GitHub adalah platform berbasis web untuk pengelolaan kode sumber dan kolaborasi pengembangan perangkat lunak yang menggunakan sistem version control Git. Platform ini memungkinkan pengembang untuk menyimpan, melacak perubahan, serta berkolaborasi dalam pengembangan proyek secara terstruktur dan terdokumentasi. GitHub menjadi standar industri dalam pengelolaan proyek perangkat lunak modern, terutama pada lingkungan kerja tim yang memerlukan transparansi dan akuntabilitas kode [17].

Selama magang di PT XYZ Finance, GitHub digunakan sebagai media utama untuk pengelolaan versi skrip Python dan dokumentasi proyek. Setiap proyek—mulai dari segmentasi pelanggan, validasi data, hingga pipeline distribusi—disimpan dan diperbarui melalui repository GitHub secara berkala. Fitur seperti commit history, branching, dan pull request sangat membantu dalam proses evaluasi skrip oleh mentor dan dokumentasi perubahan sepanjang proses pengembangan. Dengan memanfaatkan GitHub, tim internal dapat melakukan code review secara efisien sekaligus menjaga integritas skrip sepanjang siklus pengembangan.

Gambar 3.4 berikut menampilkan logo GitHub sebagai platform pengelolaan kode yang digunakan selama kegiatan magang berlangsung.



Gambar 3. 4 Logo Github [18]

#### e) **Microsoft Excel**

Microsoft Excel merupakan perangkat lunak pengolah data tabular yang umum digunakan untuk visualisasi dasar dan pengecekan struktur data [19]. Dengan fitur seperti conditional formatting, filtering, dan highlighting, Excel memudahkan pengguna dalam melakukan verifikasi cepat terhadap hasil keluaran data secara manual. Dalam proyek magang di PT XYZ Finance, Excel tidak digunakan sebagai alat utama dalam pemrosesan atau pembersihan data, tetapi hanya dimanfaatkan untuk memeriksa hasil validasi data yang telah diproses sebelumnya menggunakan Python. Contohnya, Excel digunakan untuk memastikan apakah hasil pencocokan data kependudukan sudah sesuai, atau apakah hasil penggabungan file telesales telah tersusun dengan format tabular yang benar. Dengan antarmuka visual yang praktis,

Excel berfungsi sebagai alat bantu pelengkap untuk validasi tampilan dan struktur hasil akhir sebelum didistribusikan atau digunakan lebih lanjut.

Gambar 3.5 berikut menampilkan logo Microsoft Excel sebagai platform pengelolaan data yang digunakan selama kegiatan magang berlangsung.



Gambar 3. 5 Logo Microsoft Excel [20]

Seluruh *tools* yang telah dijelaskan sebelumnya digunakan secara aktif selama pelaksanaan kegiatan magang guna menunjang kelancaran dan keberhasilan penyelesaian tugas-tugas yang diberikan oleh mentor. Penggunaan *tools* ini berperan penting dalam proses integrasi dan validasi data, pengembangan aplikasi otomatisasi, serta dalam mendukung dokumentasi dan kolaborasi teknis selama proyek berlangsung. Fokus utama kegiatan magang diarahkan pada pengembangan sistem segmentasi pelanggan, otomatisasi distribusi data, serta validasi dan penggabungan data, yang akan dijelaskan secara rinci pada subbab-subbab berikutnya.

### **3.2.1 Segmentasi Pelanggan XYZ Finance MPF**

#### **3.2.1.1 Latar Belakang dan Tujuan Pengembangan**

Segmentasi pelanggan merupakan pendekatan strategis yang bertujuan untuk mengelompokkan pelanggan ke dalam kelompok-kelompok homogen berdasarkan kesamaan karakteristik perilaku atau preferensi tertentu. Dalam konteks PT XYZ Finance, kebutuhan terhadap segmentasi pelanggan menjadi semakin penting seiring dengan rendahnya efektivitas strategi distribusi undangan event promosi yang telah diterapkan sebelumnya. Tingkat kehadiran pelanggan yang rendah dan efektivitas

komunikasi yang kurang optimal menunjukkan bahwa strategi promosi generik tidak lagi relevan di tengah kompleksitas perilaku konsumen saat ini.

Lini bisnis XYZ Finance MPF (Multipurpose Financing) merupakan salah satu unit usaha yang bergerak di bidang pembiayaan multiguna untuk produk seperti elektronik, furnitur, dan perlengkapan rumah tangga. Unit bisnis ini secara rutin menyelenggarakan event promosi untuk meningkatkan akuisisi dan loyalitas pelanggan di berbagai wilayah operasional. Namun, efektivitas event tersebut masih menghadapi tantangan akibat tidak adanya pendekatan segmentasi yang memadai dalam penentuan target undangan.

Berdasarkan latar belakang tersebut, pengembangan sistem segmentasi pelanggan dilakukan dengan memanfaatkan metode RFM (Recency, Frequency, Monetary) yang dikombinasikan dengan algoritma clustering K-Means [21]. Tujuan utama dari inisiatif ini adalah untuk menyusun strategi promosi yang lebih tepat sasaran dengan cara mengidentifikasi kelompok pelanggan potensial yang memiliki kecenderungan untuk merespons undangan promosi secara positif [22]. Dengan adanya sistem segmentasi yang akurat, diharapkan distribusi undangan event dapat dilakukan secara selektif, sehingga meningkatkan efisiensi sumber daya dan efektivitas kampanye.

Selain sebagai upaya optimalisasi promosi, pengembangan segmentasi pelanggan ini juga bertujuan sebagai wadah pembelajaran praktis dalam penerapan analitik data dalam konteks industri. Penulis, yang ditempatkan di Departemen Data Mining dan ditugaskan secara khusus pada lini bisnis MPF, mendapatkan tanggung jawab untuk merancang, mengimplementasikan, dan mengevaluasi model segmentasi berdasarkan data transaksi pelanggan yang tersedia. Keluaran dari proses ini tidak hanya berupa model segmentasi, namun juga insight strategis yang dapat dimanfaatkan oleh tim bisnis untuk pengambilan keputusan yang lebih berbasis data.

Dengan demikian, pengembangan sistem segmentasi pelanggan ini tidak hanya memberikan kontribusi nyata terhadap peningkatan kinerja pemasaran lini bisnis MPF

di bawah PT XYZ Finance, tetapi juga memperkuat kapabilitas organisasi dalam menerapkan prinsip-prinsip data-driven decision making secara berkelanjutan.

### 3.2.1.2 Persiapan dan Pembersihan Data

Langkah awal yang dilakukan dalam proses segmentasi pelanggan adalah melakukan persiapan dan pembersihan data mentah yang diperoleh dari pihak terkait. Data awal diberikan dalam bentuk file teks (.txt) dengan struktur kolom yang dipisahkan oleh delimiter pipa (|). File ini berisi informasi historis pelanggan yang cukup kompleks, seperti tanggal kontrak, jumlah pembiayaan, kategori produk, dan pola transaksi.

Agar data dapat diproses lebih lanjut dalam bentuk tabular yang lebih terstruktur, dilakukan konversi dari file teks ke format Excel (.xlsx). Proses konversi ini dilakukan menggunakan Python, dengan membangun sebuah fungsi khusus bernama `convert_txt_to_excel()`.

```
# Baca file txt dengan delimiter pipe (|)
def convert_txt_to_excel(input_file, output_file):
    """
    Fungsi untuk mengkonversi file txt dengan delimiter pipe ke Excel

    Parameters:
    input_file (str): Path ke file txt input
    output_file (str): Path ke file Excel output
    """

    try:
        # Baca file txt dengan delimiter pipe
        df = pd.read_csv(input_file,
                        delimiter='|',
                        encoding='utf-8',
                        low_memory=False)

        # Tampilkan informasi dasar tentang data
        print("Data berhasil dibaca!")
        print(f"Jumlah baris: {len(df)}")
        print(f"Jumlah kolom: {len(df.columns)}")
        print(f"Nama kolom:")
        for i, col in enumerate(df.columns, 1):
            print(f"{i:2s}. {col}")

        # Tampilkan beberapa baris pertama
        print("\n5 baris pertama data:")
        print(df.head())

        # Tampilkan tipe data setiap kolom
        print("\nTipe data kolom:")
        print(df.dtypes)

        with pd.ExcelWriter(output_file, engine='openpyxl') as writer:
            df.to_excel(writer, sheet_name='Data', index=False)

        # Atur lebar kolom otomatis
        worksheet = writer.sheets['Data']
        for column in worksheet.columns:
            max_length = 0
            column_letter = column[0].column_letter
            for cell in column:
                try:
                    if len(str(cell.value)) > max_length:
                        max_length = len(str(cell.value))
                except:
                    pass
            adjusted_width = min(max_length + 2, 50)
            worksheet.column_dimensions[column_letter].width = adjusted_width

        print(f"File Excel berhasil disimpan ke: {output_file}")

    return df
```

Gambar 3. 6 Tampilan Potongan Kode Fungsi Konversi Data

Kode ini menunjukkan proses utama dari fungsi `convert_txt_to_excel()`, yang menggunakan pustaka `pandas` untuk membaca data dari file teks dengan delimiter pipa

(l) dan mengubahnya ke dalam bentuk Excel menggunakan `to_excel()`. Di dalam fungsi tersebut juga dilakukan proses otomatisasi penyesuaian lebar kolom agar file Excel yang dihasilkan dapat dibaca dengan lebih nyaman. Pendekatan ini dipilih untuk mempermudah eksplorasi awal data sebelum dilakukan segmentasi lebih lanjut.

Setelah proses konversi selesai, file Excel hasil transformasi kemudian dimuat ulang ke dalam environment Python untuk dilakukan eksplorasi awal. Proses ini bertujuan untuk memastikan bahwa seluruh kolom telah terbaca dengan benar, serta mengevaluasi apakah terdapat nilai kosong (missing values) maupun anomali pada tipe data.

Gambar 3.7 menunjukkan potongan kode saat file hasil konversi dibaca kembali menggunakan fungsi `read_excel()` dari pustaka `pandas`. Dalam proses ini, juga dilakukan pengecekan terhadap beberapa nilai unik dari kolom-kolom kategorikal, seperti status rumah (`HOUSE_STAT`), status pernikahan (`MARITAL_STAT`), dan jenis kelamin pelanggan (`CUST_SEX`). Hal ini penting sebagai dasar validasi sebelum tahap pembersihan lanjutan.

```
try:
    data_xyzmultipurpose = pd.read_excel('DATA_XYZFINANCEMULTIPURPOSEFINANCING.xlsx')
    print("Data berhasil dimuat.")
except FileNotFoundError:
    print("Error: File tidak ditemukan. Pastikan file 'DATA_XYZFINANCEMULTIPURPOSEFINANCING.xlsx' ada di direktori yang benar.")
except Exception as e:
    print(f"Terjadi kesalahan: {e}")

data_xyzmultipurpose.head()
```

Gambar 3.7 Potongan Kode Pembacaan File Hasil Konversi Untuk Eksplorasi Awal

	CUST_NO	FIRST_PPC	FIRST_PPC_DATE	FIRST_MPF_DATE	LAST_MPF_DATE	JMH_CON_SBILUM_MPF	MAX_MPF_AMOUNT	MIN_MPF_AMOUNT	AVG_MPF_INST	MPF_CATEGORIES_TAKEN	AVG_MONTHS_BETWEEN_MPF
0	CUST_28781931	REFI	20211111	20220125	20220125	1	4299000	4299000	4.600000e+05	GADGET	NaN
1	CUST_59713242	NMC	20201024	20211208	20211208	1	8712000	8712000	1.342000e+06	GADGET	NaN
2	CUST_80454771	MPF	20191119	20191119	20210312	4	6709000	4347000	6.402500e+05	ELECTRONIC, ELECTRONIC, ELECTRONIC, ELECTRONIC	5.258065
3	CUST_07047257	REFI	20180710	20200923	20220628	3	3999000	2752000	3.876667e+05	GADGET, GADGET, GADGET	10.580643
4	CUST_66430282	MPF	20220607	20220607	20220607	1	3708000	3708000	5.370000e+05	GADGET	NaN

Gambar 3. 8 Hasil pembacaan data untuk eksplorasi awal

Tahap ini berfungsi sebagai fondasi awal sebelum masuk ke eksplorasi data yang lebih mendalam. Melalui pembacaan data hasil konversi, peneliti memastikan bahwa

seluruh kolom telah dimuat dengan benar, sehingga dapat dilanjutkan ke proses pembersihan dan penyesuaian format data sesuai kebutuhan analisis selanjutnya.

Selanjutnya dilakukan pengecekan tipe data untuk memastikan bahwa jenis data yang dimuat sesuai dengan karakteristik masing-masing atribut, khususnya pada kolom tanggal dan numerik yang perlu dikonversi ke tipe data yang tepat. Proses ini dijabarkan melalui potongan kode pada Gambar 3.9.

```

df_dtypes = data_xyzmultipurpose.dtypes.reset_index()
df_dtypes.columns = ['Nama Kolom', 'Tipe Data']

print(df_dtypes)

```

	Nama Kolom	Tipe Data
0	CUST_NO	object
1	FIRST_PPC	object
2	FIRST_PPC_DATE	int64
3	FIRST_MPF_DATE	int64
4	LAST_MPF_DATE	int64
5	JMH_CON_SBLN_MPF	int64
6	MAX_MPF_AMOUNT	int64
7	MIN_MPF_AMOUNT	int64
8	AVG_MPF_INST	float64
9	MPF_CATEGORIES_TAKEN	object
10	AVG_MONTHS_BETWEEN_MPF	float64
11	LAST_MPF_PURPOSE	object
12	LAST_MPF_AMOUNT	int64
13	LAST_MPF_TOP	int64
14	LAST_MPF_INST	int64
15	PPC	object
16	JMH_PPC	int64
17	JMH_CON_NON_MPF	float64
18	PROGRESSION_PATTERNS	object
19	BUSS_UNIT	object
20	MONTH_INST	float64
21	CONTRACT_ACTIVE_DATE	object
22	PRINCIPAL	int64
23	GRS_OP	int64
...		
31	BRANCH_ID	int64
32	AREA	object
33	TOTAL_AMOUNT_MPF	int64
34	TOTAL_PRODUCT_MPF	int64

Gambar 3. 9 Hasil pemeriksaan tipe data atribut untuk validasi struktur dataset

Potongan kode tersebut menampilkan hasil transformasi struktur data menjadi tabel yang merepresentasikan setiap kolom beserta tipe datanya. Langkah ini bertujuan untuk memberikan gambaran menyeluruh mengenai karakteristik awal dataset, sebelum dilakukan konversi lebih lanjut. Terlihat bahwa beberapa atribut seperti *FIRST\_PPC\_DATE*, *LAST\_MPF\_DATE*, dan *BIRTH\_DATE* masih berada dalam format numerik (*int64*) atau teks (*object*), sehingga perlu dikonversi ke tipe *datetime*. Demikian pula, kolom seperti *OCPT\_CODE* dan *NO\_OF\_DEPEND* yang awalnya bertipe *float64* perlu dikonversi ke *int64* agar dapat diproses secara logis dan sesuai konteks.

Setelah dilakukan validasi tipe data, tahap berikutnya adalah melakukan konversi tipe data untuk atribut-atribut yang masih berada dalam format yang tidak sesuai dengan karakteristik aslinya. Sebagai contoh, beberapa atribut tanggal seperti `FIRST_PPC_DATE`, `FIRST_MPF_DATE`, `LAST_MPF_DATE`, `CONTRACT_ACTIVE_DATE`, dan `BIRTH_DATE` masih memiliki tipe data `int64` atau `object`, sehingga perlu dikonversi ke format `datetime` agar dapat digunakan dalam analisis berbasis waktu secara akurat.

Konversi dilakukan menggunakan fungsi `pd.to_datetime()` dengan penyesuaian format dan parameter `errors='coerce'` untuk menghindari kesalahan saat parsing data. Selain itu, dilakukan pula transformasi tambahan dengan membentuk kolom baru `YearMonth` dari atribut `CONTRACT_ACTIVE_DATE` yang telah dikonversi, dengan tujuan mempermudah analisis tren berdasarkan periode bulanan. Di samping itu, diturunkan pula atribut baru `Usia` pelanggan, yang dihitung berdasarkan selisih antara tahun acuan (2024) dan tahun lahir (`BIRTH_DATE`).

Langkah-langkah transformasi tersebut ditampilkan dalam potongan kode pada Gambar 3.10.

```
Data Formatting

data_xyzmultipurpose['FIRST_PPC_DATE'] = pd.to_datetime(data_xyzmultipurpose['FIRST_PPC_DATE'], format='%Y%m%d', errors='coerce')
data_xyzmultipurpose['FIRST_MPF_DATE'] = pd.to_datetime(data_xyzmultipurpose['FIRST_MPF_DATE'], format='%Y%m%d', errors='coerce')
data_xyzmultipurpose['LAST_MPF_DATE'] = pd.to_datetime(data_xyzmultipurpose['LAST_MPF_DATE'], format='%Y%m%d', errors='coerce')
data_xyzmultipurpose['CONTRACT_ACTIVE_DATE'] = pd.to_datetime(data_xyzmultipurpose['CONTRACT_ACTIVE_DATE'], errors='coerce')
data_xyzmultipurpose['BIRTH_DATE'] = pd.to_datetime(data_xyzmultipurpose['BIRTH_DATE'], errors='coerce')

data_xyzmultipurpose['CONTRACT_ACTIVE_DATE'] = pd.to_datetime(data_xyzmultipurpose['CONTRACT_ACTIVE_DATE'], errors='coerce')
data_xyzmultipurpose['YearMonth'] = data_xyzmultipurpose['CONTRACT_ACTIVE_DATE'].dt.to_period('M')

data_xyzmultipurpose['BIRTH_DATE'] = pd.to_datetime(data_xyzmultipurpose['BIRTH_DATE'], errors='coerce')

data_xyzmultipurpose['Usia'] = 2024 - data_xyzmultipurpose['BIRTH_DATE'].dt.year
```

Gambar 3. 10 Potongan kode konversi datetime dan transformasi usia pelanggan

Selain konversi terhadap atribut bertipe tanggal (`datetime`), penyesuaian tipe data juga dilakukan terhadap beberapa atribut numerik yang semula bertipe `float64`, padahal secara konteks seharusnya merupakan bilangan bulat (`integer`). Dua atribut

yang mengalami hal ini adalah OCPT\_CODE dan NO\_OF\_DEPEND, yang kemudian dikonversi ke tipe Int64 menggunakan metode astype().

Langkah ini penting untuk memastikan bahwa seluruh atribut numerik memiliki tipe data yang tepat sebelum dianalisis lebih lanjut, terutama jika akan digunakan dalam perhitungan statistik atau visualisasi kategorikal yang membutuhkan kejelasan representasi bilangan bulat.

Potongan kode untuk proses konversi tersebut ditampilkan pada Gambar 3.6 berikut ini:

```
data_xyzmultipurpose['OCPT_CODE'] = data_xyzmultipurpose['OCPT_CODE'].astype('Int64')
data_xyzmultipurpose['NO_OF_DEPEND'] = data_xyzmultipurpose['NO_OF_DEPEND'].astype('Int64')

print(data_xyzmultipurpose.dtypes)
```

CUST_NO	object
FIRST_PPC	object
FIRST_PPC_DATE	datetime64[ns]
FIRST_MPF_DATE	datetime64[ns]
LAST_MPF_DATE	datetime64[ns]
JMH_CON_SBLM_MPF	int64
MAX_MPF_AMOUNT	int64
MIN_MPF_AMOUNT	int64
AVG_MPF_INST	float64
MPF_CATEGORIES_TAKEN	object
AVG_MONTHS_BETWEEN_MPF	float64
LAST_MPF_PURPOSE	object
LAST_MPF_AMOUNT	int64
LAST_MPF_TOP	int64
LAST_MPF_INST	int64
PPC	object
JMH_PPC	int64
JMH_CON_NON_MPF	float64
PROGRESSION_PATTERNS	object
BUSS_UNIT	object
MONTH_INST	float64
CONTRACT_ACTIVE_DATE	datetime64[ns]
PRINCIPAL	int64
GRS_DP	int64
BIRTH_DATE	datetime64[ns]
...	
TOTAL_PRODUCT_MPF	int64
YearMonth	period[M]
Usia	float64
dtype:	object

Gambar 3. 11 Potongan kode konversi tipe numerik ke int64

Setelah seluruh atribut numerik penting seperti OCPT\_CODE dan NO\_OF\_DEPEND berhasil dikonversi ke dalam tipe data int64, langkah berikutnya adalah melakukan pemetaan struktur data pelanggan secara menyeluruh. Pemetaan ini bertujuan untuk mengelompokkan setiap atribut berdasarkan fungsinya dalam analisis, sehingga mempermudah proses eksplorasi dan penarikan insight yang relevan. Penjelasan lengkap mengenai struktur data pelanggan disajikan pada Tabel 3.2 berikut:

Tabel 3. 2 Struktur Data Pelanggan

No	Nama Kolom	Tipe Data	Kategori	Deskripsi
1	CUST_NO	object	Identitas Demografi &	ID unik pelanggan
2	FIRST_PPC	object	Riwayat Pembiayaan Transaksi &	Jenis produk pertama yang diambil pelanggan
3	FIRST_PPC_DATE	int64	Riwayat Pembiayaan Transaksi &	Tanggal pembelian pertama pelanggan
4	FIRST_MPF_DATE	int64	Riwayat Pembiayaan Transaksi &	Tanggal pertama pelanggan mengambil pembiayaan MPF
5	LAST_MPF_DATE	int64	Riwayat Pembiayaan Transaksi &	Tanggal terakhir pelanggan melakukan transaksi MPF
6	JMH_CON_SBLM_MPF	int64	Riwayat Pembiayaan Transaksi &	Jumlah kontrak sebelum MPF terakhir
7	MAX_MPF_AMOUNT	int64	Pola Pembiayaan & Keuangan	Nilai pembiayaan MPF terbesar yang pernah diambil pelanggan
8	MIN_MPF_AMOUNT	int64	Pola Pembiayaan & Keuangan	Nilai pembiayaan MPF terkecil yang pernah diambil pelanggan
9	AVG_MPF_INST	float64	Pola Pembiayaan & Keuangan	Rata-rata cicilan per bulan pelanggan
10	MPF_CATEGORIES_TAKEN	object	Riwayat Pembiayaan Transaksi &	Kategori produk yang pernah diambil pelanggan
11	AVG_MONTHS_BETWEEN_MPF	float64	Pola Pembiayaan & Keuangan	Rata-rata waktu antar transaksi MPF pelanggan
12	LAST_MPF_PURPOSE	object	Riwayat Pembiayaan Transaksi &	Tujuan pembelian terakhir pelanggan
13	LAST_MPF_AMOUNT	int64	Riwayat Pembiayaan Transaksi &	Nilai transaksi terakhir pelanggan
14	LAST_MPF_TOP	int64	Riwayat Pembiayaan Transaksi &	Produk terakhir yang diambil pelanggan
15	LAST_MPF_INST	int64	Riwayat Pembiayaan Transaksi &	Jumlah cicilan dari pembelian terakhir pelanggan

No	Nama Kolom	Tipe Data	Kategori	Deskripsi
16	PPC	object	Riwayat Pembiayaan Transaksi &	Data PPC (kontrak pembiayaan pelanggan)
17	JMH_PPC	int64	Riwayat Pembiayaan Transaksi &	Jumlah total PPC (kontrak pembiayaan pelanggan)
18	JMH_CON_NON_MPF	float64	Riwayat Pembiayaan Transaksi &	Jumlah kontrak non-MPF yang dimiliki pelanggan
19	PROGRESSION_PATTERNS	object	Pola Pembiayaan & Keuangan	Pola perkembangan pelanggan dalam pembiayaan
20	BUSS_UNIT	object	Data Lokasi & Cabang	Unit bisnis tempat pelanggan terdaftar
21	MONTH_INST	float64	Pola Pembiayaan & Keuangan	Jumlah bulan angsuran pelanggan (tenor cicilan)
22	CONTRACT_ACTIVE_DATE	object	Pola Pembiayaan & Keuangan	Tanggal kontrak aktif terakhir pelanggan
23	PRINCIPAL	int64	Pola Pembiayaan & Keuangan	Nilai pokok pembiayaan pelanggan
24	GRS_DP	int64	Pola Pembiayaan & Keuangan	Uang muka (down payment) yang dibayarkan pelanggan
25	BIRTH_DATE	object	Identitas Demografi &	Tanggal lahir pelanggan
26	CUST_SEX	object	Identitas Demografi &	Jenis kelamin pelanggan
27	EDU_TYPE	object	Identitas Demografi &	Tingkat pendidikan pelanggan
28	OCPT_CODE	float64	Identitas Demografi &	Kode pekerjaan pelanggan
28	OCPT_CODE	float64	Identitas Demografi &	Kode pekerjaan pelanggan
29	HOUSE_STAT	object	Identitas Demografi &	Status kepemilikan rumah
30	MARITAL_STAT	object	Identitas Demografi &	Status pernikahan pelanggan
31	NO_OF_DEPEND	float64	Identitas Demografi &	Jumlah tanggungan pelanggan
32	BRANCH_ID	int64	Data Lokasi & Cabang	ID cabang tempat pelanggan melakukan kontrak
33	AREA	object	Data Lokasi & Cabang	Wilayah geografis pelanggan
34	TOTAL_AMOUNT_MPF	int64	Data Tambahan & Produk	Total nilai pembiayaan MPF yang pernah diambil pelanggan

No	Nama Kolom	Tipe Data	Kategori	Deskripsi
35	TOTAL_PRODUCT_MPF	int64	Data Tambahan & Produk	Total jumlah produk MPF yang pernah dibeli pelanggan

Setelah seluruh atribut data berhasil dipetakan berdasarkan fungsi dan jenisnya dalam Tabel 3.12, langkah selanjutnya adalah memastikan kualitas dan integritas data melalui proses *data cleansing*. Salah satu fokus utama dalam tahap ini adalah penanganan nilai hilang (*missing values*) yang dapat memengaruhi validitas hasil analisis.

Gambar 3.12 menunjukkan hasil pemeriksaan jumlah nilai hilang untuk setiap atribut pada data XYZ Multipurpose. Teridentifikasi bahwa sebagian besar atribut tidak memiliki nilai kosong (*missing*), namun terdapat beberapa atribut dengan proporsi nilai hilang yang signifikan. Beberapa atribut dengan jumlah *missing* terbesar antara lain:

1. JMH\_CON\_NON\_MPF: 735.395 entri kosong
2. AVG\_MONTHS\_BETWEEN\_MPF: 371.382 entri kosong
3. MONTH\_INST: 1.565 entri kosong
4. BIRTH\_DATE dan Usia: masing-masing 95 entri kosong

Atribut demografis seperti CUST\_SEX, EDU\_TYPE, OCPT\_CODE, HOUSE\_STAT, dan MARITAL\_STAT juga memiliki puluhan entri kosong, meskipun dalam jumlah yang relatif kecil.

Kondisi ini memerlukan perlakuan berbeda tergantung pada signifikansi atribut terhadap analisis. Untuk atribut demografis yang memiliki sedikit nilai hilang, dapat dipertimbangkan untuk diimputasi menggunakan metode mode atau most frequent value. Sementara itu, atribut seperti AVG\_MONTHS\_BETWEEN\_MPF dan JMH\_CON\_NON\_MPF yang memiliki proporsi nilai hilang sangat besar, dapat dipertimbangkan untuk dieliminasi dari model analisis jika tidak berdampak besar terhadap hasil utama.

Langkah identifikasi nilai hilang ini menjadi krusial untuk memastikan bahwa proses analitik yang dilakukan selanjutnya dibangun di atas fondasi data yang bersih dan valid.

```
Handling Missing Values

print(data_xyzmultipurpose.isnull().sum())

CUST_NO                0
FIRST_PPC              0
FIRST_PPC_DATE         0
FIRST_MPF_DATE         0
LAST_MPF_DATE          0
JMH_CON_SBLM_MPF       0
MAX_MPF_AMOUNT         0
MIN_MPF_AMOUNT         0
AVG_MPF_INST           0
MPF_CATEGORIES_TAKEN   0
AVG_MONTHS_BETWEEN_MPF 371382
LAST_MPF_PURPOSE       0
LAST_MPF_AMOUNT        0
LAST_MPF_TOP           0
LAST_MPF_INST          0
PPC                    0
JMH_PPC                0
JMH_CON_NON_MPF        735395
PROGRESSION_PATTERNS   0
BUS_S_UNIT             0
MONTH_INST             1565
CONTRACT_ACTIVE_DATE   0
PRINCIPAL              0
GRS_DP                 0
BIRTH_DATE             16
...
Jumlah_Kategori_Produk 0
Year                   0
Month                  0
dtype: int64

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Gambar 3. 12 Pemeriksaan Jumlah Missing Value

Sebagai tindak lanjut dari proses identifikasi nilai hilang, dilakukan penghapusan terhadap atribut `JMH_CON_NON_MPF` sebagaimana ditunjukkan pada Gambar 3.13. Atribut ini dihapus karena memiliki proporsi nilai kosong yang sangat tinggi dan tidak memberikan kontribusi yang signifikan terhadap tujuan analisis. Penghapusan dilakukan untuk menghindari distorsi pada model akibat keberadaan data yang tidak lengkap.

Selanjutnya, pada Gambar 3.14 diperlihatkan langkah imputasi nilai hilang menggunakan metode median. Atribut numerik seperti `MONTH_INST`, `OCPT_CODE`, `NO_OF_DEPEND`, dan `Usia` diisi dengan nilai tengah dari masing-masing distribusinya. Pemilihan metode median didasarkan pada pertimbangan bahwa nilai ini lebih robust terhadap outlier dan cocok digunakan untuk data numerik yang tidak selalu berdistribusi normal.

Langkah-langkah ini penting untuk menjaga integritas dataset sebelum dilakukan tahap eksplorasi dan pemodelan lebih lanjut. Dengan menghapus atribut yang tidak relevan dan mengimputasi nilai hilang secara tepat, dataset menjadi lebih konsisten, valid, dan layak untuk dianalisis.

```
Hapus kolom JMH_CON_NON_MPF karena tidak relevan

data_xyzmultipurpose.drop(columns=['JMH_CON_NON_MPF'], inplace=True)
```

Gambar 3. 13 Penghapusan Atribut JMH\_CON\_NON\_MPF

```
Mengisi Kolom yang null dengan nilai median

data_xyzmultipurpose['MONTH_INST'].fillna(data_xyzmultipurpose['MONTH_INST'].median(), inplace=True)
data_xyzmultipurpose['OCPT_CODE'].fillna(data_xyzmultipurpose['OCPT_CODE'].median(), inplace=True)
data_xyzmultipurpose['NO_OF_DEPEND'].fillna(data_xyzmultipurpose['NO_OF_DEPEND'].median(), inplace=True)
data_xyzmultipurpose['Usia'].fillna(data_xyzmultipurpose['Usia'].median(), inplace=True)
```

Gambar 3. 14 Imputasi Nilai Hilang pada Atribut Numerik

Setelah dilakukan identifikasi terhadap atribut-atribut yang memiliki nilai kosong, langkah selanjutnya adalah penanganan terhadap atribut-atribut tersebut. Pada tahap ini, terdapat dua pendekatan utama yang diterapkan, yaitu:

Gambar 3.15 menunjukkan proses imputasi nilai kosong pada atribut kategorikal menggunakan metode *modus* (nilai yang paling sering muncul). Atribut seperti CUST\_SEX, EDU\_TYPE, HOUSE\_STAT, MARITAL\_STAT, dan AREA yang sebelumnya memiliki nilai hilang dalam jumlah relatif kecil diisi dengan nilai terbanyak dari masing-masing kolom. Pendekatan ini dipilih untuk menjaga representasi kategori dominan tanpa mengubah struktur distribusi data.

```
Mengisi Kolom yang null dengan nilai modus

data_xyzmultipurpose['CUST_SEX'].fillna(data_xyzmultipurpose['CUST_SEX'].mode()[0], inplace=True)
data_xyzmultipurpose['EDU_TYPE'].fillna(data_xyzmultipurpose['EDU_TYPE'].mode()[0], inplace=True)
data_xyzmultipurpose['HOUSE_STAT'].fillna(data_xyzmultipurpose['HOUSE_STAT'].mode()[0], inplace=True)
data_xyzmultipurpose['MARITAL_STAT'].fillna(data_xyzmultipurpose['MARITAL_STAT'].mode()[0], inplace=True)
data_xyzmultipurpose['AREA'].fillna(data_xyzmultipurpose['AREA'].mode()[0], inplace=True)
```

Gambar 3. 15 Imputasi Missing Value dengan Metode Modus

Setelah semua nilai kosong ditangani, tahap pembersihan dilanjutkan dengan proses deteksi dan penghapusan duplikasi data berdasarkan kolom identifikasi unik, yaitu CUST\_NO. Seperti yang ditunjukkan pada Gambar 3.16, ditemukan sebanyak 2.727 entri duplikat dalam dataset. Untuk menjaga validitas analisis, entri-entri duplikat tersebut dihapus, sehingga hanya satu data unik untuk setiap pelanggan yang dipertahankan.

```
# Cek jumlah duplikasi
print("Jumlah duplikasi berdasarkan CUST_NO:", data_xyzmultipurpose.duplicated(subset=['CUST_NO']).sum())

# Jika ada duplikasi, hapus data yang sama berdasarkan CUST_NO
data_xyzmultipurpose = data_xyzmultipurpose.drop_duplicates(subset=['CUST_NO'])

Jumlah duplikasi berdasarkan CUST_NO: 2727
```

Gambar 3. 16 Pemeriksaan dan Penghapusan Data Duplikat

Dengan selesainya tahapan cleansing ini, maka dataset siap digunakan untuk proses analisis lebih lanjut dengan tingkat kebersihan dan konsistensi yang lebih tinggi.

Setelah seluruh data duplikat berhasil dihapus, tahap selanjutnya adalah melakukan normalisasi format data guna menyamakan struktur tipe data pada masing-masing kolom. Langkah ini penting untuk menjamin keseragaman format dan kompatibilitas data pada tahap transformasi dan analisis selanjutnya.

Proses normalisasi dilakukan terhadap tiga jenis kolom utama: atribut waktu (datetime), atribut numerik, dan atribut kategorikal. Pertama, kolom-kolom bertipe waktu seperti FIRST\_PPC\_DATE, FIRST\_MPF\_DATE, LAST\_MPF\_DATE, CONTRACT\_ACTIVE\_DATE, dan BIRTH\_DATE dikonversi ke format datetime

menggunakan `pd.to_datetime()`. Konversi ini memastikan bahwa seluruh kolom tanggal dapat dikenali sebagai objek waktu yang valid, dengan opsi `errors='coerce'` untuk menangani nilai yang tidak sesuai format.

Kedua, kolom numerik seperti `TOTAL_AMOUNT_MPF`, `TOTAL_PRODUCT_MPF`, `MONTH_INST`, dan `Usia` dikonversi ke tipe `float` untuk memastikan dapat diolah secara matematis pada tahap perhitungan statistik atau pembobotan.

Ketiga, untuk kolom kategorikal seperti `MPF_CATEGORIES_TAKEN`, `CUST_SEX`, `EDU_TYPE`, `HOUSE_STAT`, `MARITAL_STAT`, dan `AREA`, dilakukan proses pembersihan dengan menghapus spasi di awal dan akhir string menggunakan fungsi `str.strip()`. Hal ini dilakukan agar nilai kategori tidak salah diidentifikasi akibat inkonsistensi penulisan.

Gambar 3.17 berikut menampilkan potongan kode program yang digunakan dalam proses normalisasi format data:

```
Normalisasi Format Data

date_cols = ['FIRST_PPC_DATE', 'FIRST_MPF_DATE', 'LAST_MPF_DATE', 'CONTRACT_ACTIVE_DATE', 'BIRTH_DATE']
for col in date_cols:
    data_xyzmultipurpose[col] = pd.to_datetime(data_xyzmultipurpose[col], errors='coerce')

numerical_cols = ['TOTAL_AMOUNT_MPF', 'TOTAL_PRODUCT_MPF', 'MONTH_INST', 'Usia']
data_xyzmultipurpose[numerical_cols] = data_xyzmultipurpose[numerical_cols].astype(float)

categorical_cols = ['MPF_CATEGORIES_TAKEN', 'CUST_SEX', 'EDU_TYPE', 'HOUSE_STAT', 'MARITAL_STAT', 'AREA']
data_xyzmultipurpose[categorical_cols] = data_xyzmultipurpose[categorical_cols].apply(lambda x: x.str.strip() if x.dtype == "object" else x)
```

Gambar 3. 17 Normalisasi Format Data terhadap Atribut Tanggal, Numerik, dan Kategorikal

Dengan terselesaikannya tahap ini, dataset telah memiliki struktur yang seragam dan siap untuk dilakukan transformasi fitur RFM sebagai dasar proses segmentasi pelanggan berbasis algoritma clustering.

### 3.2.1.3 Eksplorasi dan Analisis Data Pelanggan

Sebelum dilakukan proses segmentasi menggunakan pendekatan algoritmik, diperlukan pemahaman menyeluruh terhadap karakteristik data pelanggan yang

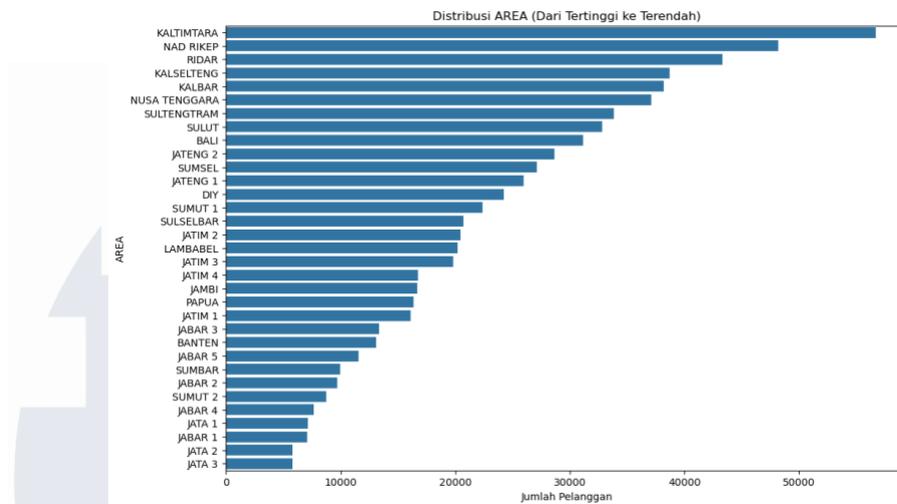
tersedia. Tahapan eksplorasi data (exploratory data analysis/EDA) ini bertujuan untuk mengidentifikasi pola distribusi, hubungan antarvariabel, serta potensi insight yang relevan terhadap strategi pemasaran dan promosi [23]. Analisis dilakukan terhadap berbagai atribut penting seperti wilayah geografis pelanggan, usia, jenis kelamin, riwayat transaksi, serta preferensi produk yang pernah diambil.

Hasil dari eksplorasi ini memberikan landasan awal yang kuat dalam menentukan fitur-fitur yang akan digunakan dalam model segmentasi, sekaligus menjadi sumber insight untuk memahami perilaku pelanggan dalam konteks bisnis pembiayaan multiguna yang dijalankan oleh lini bisnis MPF di bawah PT XYZ Finance.

### **1. Distribusi Pelanggan Berdasarkan Wilayah (AREA)**

Visualisasi pada Gambar 3.18 menyajikan distribusi jumlah pelanggan berdasarkan nilai frekuensi tertinggi hingga terendah untuk masing-masing AREA. Hasil analisis menunjukkan bahwa wilayah KALIMTARA, NAD RIKEP, dan RIDAR merupakan tiga area dengan jumlah pelanggan terbanyak. Sementara itu, wilayah seperti JATA 3, JATA 2, dan JATA 1 tercatat memiliki jumlah pelanggan yang relatif rendah dibandingkan wilayah lainnya.

Temuan ini memberikan insight awal bahwa strategi promosi dapat difokuskan terlebih dahulu pada area dengan basis pelanggan terbesar untuk memaksimalkan potensi konversi undangan. Selain itu, wilayah dengan jumlah pelanggan rendah dapat dijadikan target pengembangan lebih lanjut atau dianalisis lebih dalam untuk memahami hambatan penetrasi pasar. Distribusi ini juga dapat membantu dalam perencanaan logistik, alokasi anggaran promosi, dan prioritas komunikasi di tiap area.

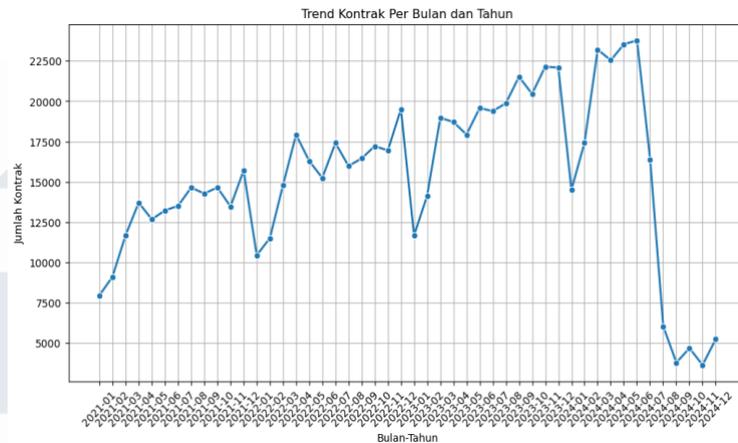


Gambar 3. 18 Visualisasi Distribusi Jumlah Pelanggan Berdasarkan Wilayah AREA

## 2. Tren Jumlah Kontrak per Tahun dan per Bulan

Setelah memahami distribusi pelanggan berdasarkan wilayah, analisis dilanjutkan dengan menelusuri pola temporal dalam data, khususnya terhadap jumlah kontrak pembiayaan yang terjadi setiap bulannya. Analisis tren ini dilakukan guna mengidentifikasi fluktuasi aktivitas pembiayaan dari waktu ke waktu, sekaligus sebagai dasar dalam melihat potensi musiman (*seasonality*) maupun adanya perubahan perilaku pelanggan.

Gambar 3.19 memperlihatkan visualisasi *trendline* jumlah kontrak berdasarkan data agregasi bulanan. Terlihat bahwa secara umum, terdapat tren peningkatan jumlah kontrak dari tahun ke tahun, terutama sejak awal 2017 hingga puncaknya pada pertengahan 2020–2021. Namun demikian, tren ini mengalami penurunan yang sangat signifikan setelah pertengahan tahun 2021, yang kemungkinan besar disebabkan oleh faktor eksternal seperti penurunan daya beli atau kebijakan pembiayaan dari perusahaan.

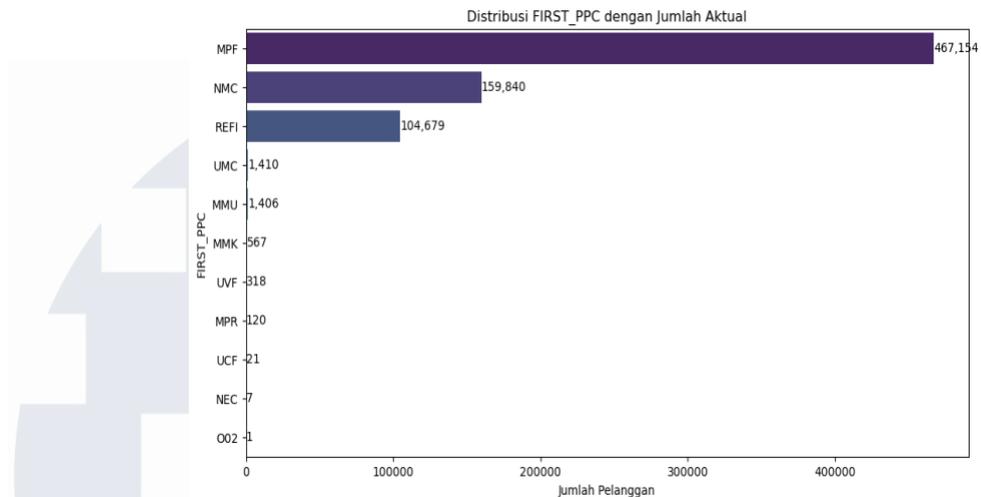


Gambar 3.19 Trend Jumlah Kontrak Per Bulan dan Tahun

### 3. Produk Pertama yang Diambil Pelanggan (FIRST\_PPC)

Setelah memahami tren waktu dari kontrak pembiayaan, analisis selanjutnya difokuskan pada jenis produk pembiayaan pertama yang diambil oleh pelanggan. Variabel FIRST\_PPC memberikan gambaran awal mengenai produk mana yang menjadi pintu masuk pelanggan terhadap layanan pembiayaan MPF. Analisis ini penting untuk mengidentifikasi produk utama yang berperan sebagai entry point dalam hubungan pelanggan dengan perusahaan.

Gambar 3.20 memperlihatkan distribusi variabel FIRST\_PPC, yang menunjukkan bahwa sebagian besar pelanggan memulai kontraknya dengan produk MPF, yakni sebanyak 467.154 pelanggan. Selanjutnya, produk NMC dan REFI menempati urutan kedua dan ketiga dengan jumlah pelanggan masing-masing sebanyak 159.840 dan 104.679. Sementara itu, jenis produk lainnya menunjukkan angka yang jauh lebih rendah, mengindikasikan bahwa kontribusinya terhadap onboarding awal pelanggan relatif kecil.



Gambar 3. 20 Distribusi FIRST\_PPC dengan Jumlah Aktual

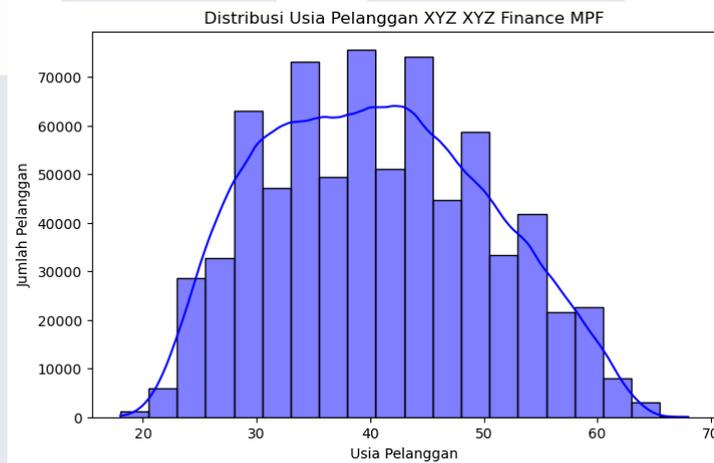
#### 4. Distribusi Usia Pelanggan

Setelah mengevaluasi distribusi produk awal pelanggan, analisis dilanjutkan dengan mengamati karakteristik demografis, khususnya usia pelanggan. Pemahaman terhadap distribusi usia pelanggan sangat penting dalam merancang strategi pemasaran yang lebih tepat sasaran, serta untuk mengetahui segmen usia mana yang paling dominan dalam menggunakan produk pembiayaan XYZ Multipurpose.

Gambar 3.21 memperlihatkan distribusi usia pelanggan dalam bentuk histogram dengan kurva kepadatan (kde). Hasil visualisasi menunjukkan bahwa mayoritas pelanggan berada dalam rentang usia antara 30 hingga 45 tahun, dengan puncak jumlah pelanggan terdapat pada usia sekitar 35–40 tahun. Distribusi ini memiliki kecenderungan berbentuk normal namun sedikit condong ke kiri (left-skewed), yang menandakan bahwa terdapat sebagian kecil pelanggan dari usia lebih tua, namun tidak signifikan secara jumlah.

Distribusi ini memberikan insight bahwa pelanggan XYZ Multipurpose didominasi oleh kelompok usia produktif, yang secara umum memiliki kebutuhan pembiayaan yang tinggi untuk mendukung aktivitas

konsumtif atau produktif mereka. Hal ini menjadi pertimbangan penting dalam menentukan target segmen pada strategi kampanye atau peluncuran produk baru.



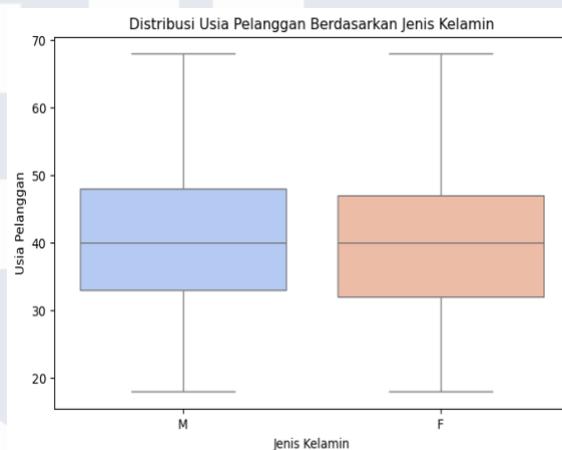
Gambar 3. 21 Distribusi Usia Pelanggan XYZ Finance MPF

## 5. Distribusi Usia Berdasarkan Jenis Kelamin

Selain mengamati distribusi usia secara umum, analisis juga diperluas dengan membedakan pola usia berdasarkan jenis kelamin pelanggan. Pendekatan ini dilakukan untuk mengetahui apakah terdapat perbedaan signifikan dalam sebaran usia antara pelanggan laki-laki dan perempuan, serta untuk memahami potensi segmentasi pasar yang lebih terarah berdasarkan demografi.

Gambar 3.22 menyajikan visualisasi boxplot yang membandingkan distribusi usia pelanggan berdasarkan jenis kelamin. Terlihat bahwa baik pelanggan laki-laki maupun perempuan memiliki median usia yang relatif serupa, yaitu berada di sekitar 38–40 tahun. Rentang usia yang dominan (interkuartil) pada kedua kelompok berkisar antara 30 hingga 50 tahun, yang mengindikasikan bahwa mayoritas pelanggan, tanpa memandang jenis kelamin, berada dalam usia produktif.

Meskipun persebaran nilai minimum dan maksimum menunjukkan keberagaman individu dari berbagai kelompok usia, secara keseluruhan tidak ditemukan perbedaan mencolok antara kedua kelompok. Hasil ini menunjukkan bahwa produk pembiayaan PT XYZ FINANCE menjangkau pria dan wanita secara merata, sehingga pendekatan pemasaran berbasis gender dapat difokuskan pada pesan yang relevan secara usia, tanpa perlu diferensiasi yang tajam berdasarkan jenis kelamin.



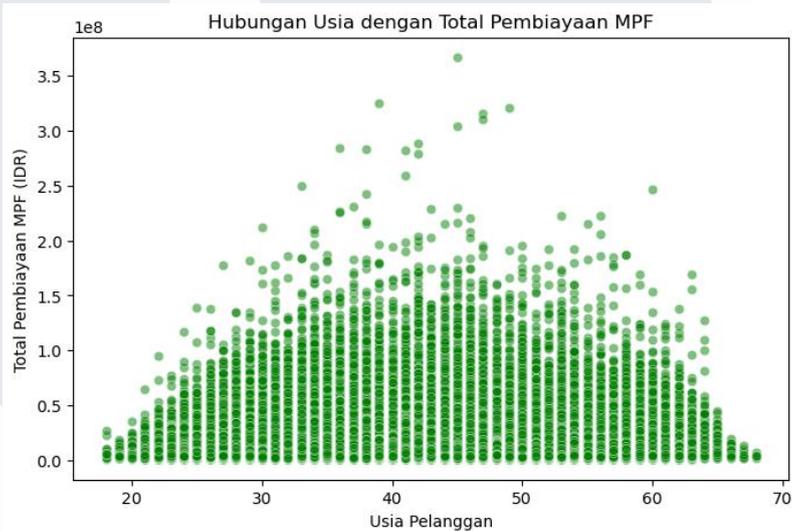
Gambar 3. 22 Distribusi Usia Pelanggan Berdasarkan Jenis Kelamin

## 6. Korelasi Usia dengan Total Pembiayaan

Setelah menganalisis distribusi usia secara umum dan berdasarkan jenis kelamin, eksplorasi dilanjutkan dengan meninjau hubungan antara usia pelanggan dan total pembiayaan Multi Purpose Financing (MPF) yang diambil. Tujuannya adalah untuk mengevaluasi apakah terdapat pola atau kecenderungan tertentu terkait usia dan besarnya nominal pembiayaan.

Gambar 3.23 memperlihatkan scatterplot hubungan antara usia pelanggan (sumbu horizontal) dan total pembiayaan MPF (sumbu vertikal). Terlihat bahwa pelanggan berusia 30–50 tahun cenderung mengambil pembiayaan lebih besar dibanding kelompok usia lainnya.

Distribusi data menunjukkan konsentrasi di rentang usia produktif, dengan puncak pada usia 40–45 tahun, kemudian menurun. Hal ini mencerminkan siklus ekonomi, di mana kelompok usia produktif umumnya memiliki penghasilan stabil, kebutuhan tinggi, serta kapasitas kredit yang lebih besar. Temuan ini menjadi dasar penting dalam perumusan kebijakan pembiayaan berbasis usia yang lebih tepat sasaran.



Gambar 3. 23 Hubungan Usia dengan Total Pembiayaan MPF

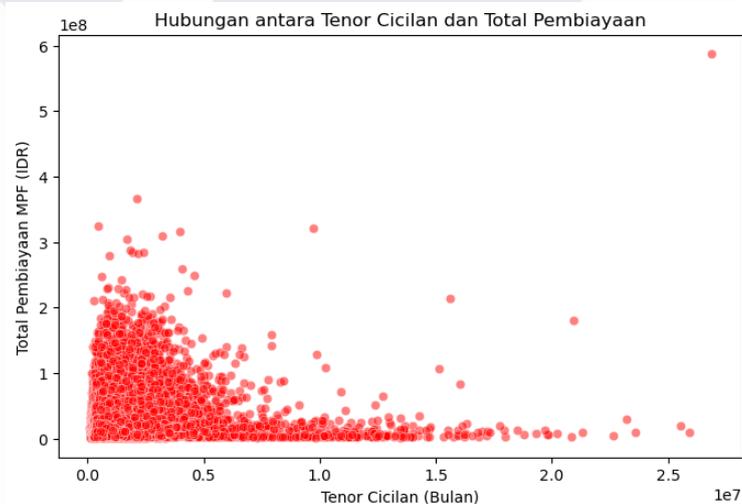
## 7. Korelasi Tenor Cicilan dengan Pembiayaan

Setelah mengkaji keterkaitan antara usia pelanggan dan total pembiayaan MPF, analisis selanjutnya diarahkan pada hubungan antara tenor cicilan (jumlah bulan angsuran) dan besarnya pembiayaan yang diambil oleh pelanggan. Insight ini penting untuk memahami sejauh mana panjang tenor mempengaruhi keputusan pelanggan dalam menentukan nominal pembiayaan.

Gambar 3.24 menyajikan scatterplot yang menunjukkan hubungan antara *tenor cicilan* (sumbu horizontal) dengan *total pembiayaan MPF* (sumbu vertikal). Setiap titik merepresentasikan satu

pelanggan, dengan warna merah dan tingkat transparansi (alpha) yang disesuaikan untuk memperlihatkan kepadatan data. Secara umum, data memperlihatkan pola sebar yang cenderung terkonsentrasi pada tenor pendek dengan nominal pembiayaan rendah hingga menengah.

Meskipun terdapat beberapa outlier pada tenor panjang dan pembiayaan tinggi, mayoritas pelanggan cenderung mengambil pembiayaan dalam nilai tertentu tanpa memperpanjang tenor secara ekstrem. Hal ini mengindikasikan bahwa tenor panjang tidak selalu berkorelasi dengan pembiayaan tinggi, dan keputusan pelanggan kemungkinan lebih dipengaruhi oleh faktor kemampuan bayar bulanan atau preferensi risiko. Temuan ini dapat dijadikan acuan bagi PT XYZ FINANCE dalam menyusun simulasi cicilan yang lebih adaptif terhadap kebutuhan segmen pelanggan.



Gambar 3. 24 Hubungan antara Tenor Cicilan dan Total Pembiayaan

## 8. Rata-rata Pembiayaan per Kategori Produk

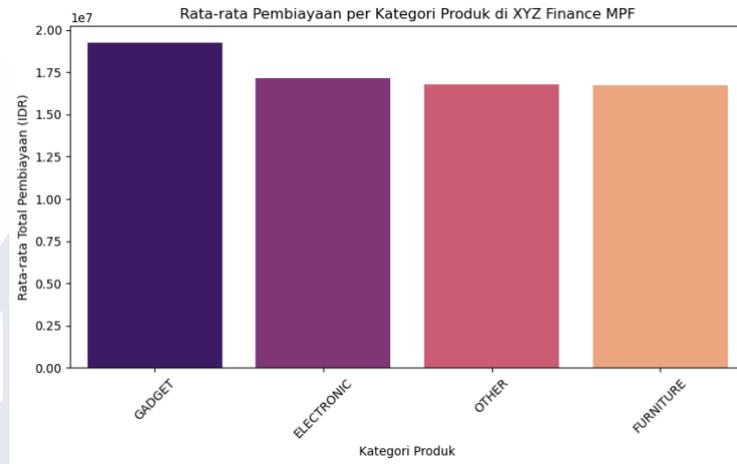
Setelah mengamati hubungan antara variabel-variabel kuantitatif seperti usia dan tenor cicilan dengan total pembiayaan, analisis kemudian dilanjutkan dengan mengevaluasi rata-rata pembiayaan berdasarkan kategori produk yang diambil oleh pelanggan.

Insight ini bertujuan untuk mengidentifikasi jenis produk apa yang cenderung memiliki nilai pembiayaan lebih tinggi, sehingga dapat menjadi acuan dalam menyusun prioritas pemasaran atau pengembangan produk.

Gambar 3.25 menampilkan visualisasi rata-rata pembiayaan untuk setiap kategori produk yang tersedia di PT XYZ FINANCE. Kategori produk diperoleh dari data kolom MPF\_CATEGORIES\_TAKEN, yang diekstraksi menjadi bentuk granular melalui proses eksplorasi data multi-kategori. Hasilnya menunjukkan bahwa **kategori Gadget memiliki rata-rata pembiayaan tertinggi**, diikuti oleh kategori Electronic, Other, dan terakhir Furniture.

Pola ini mencerminkan adanya kecenderungan pelanggan untuk mengambil pembiayaan yang lebih tinggi untuk produk berteknologi seperti gadget dan perangkat elektronik. Hal ini bisa disebabkan oleh harga produk yang relatif mahal, daya tarik terhadap inovasi teknologi terbaru, serta kebutuhan fungsional yang tinggi dari produk-produk tersebut dalam kehidupan sehari-hari. Temuan ini dapat membantu perusahaan dalam mengembangkan strategi promosi dan cross-selling yang disesuaikan berdasarkan kategori produk yang memiliki nilai pembiayaan tinggi.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

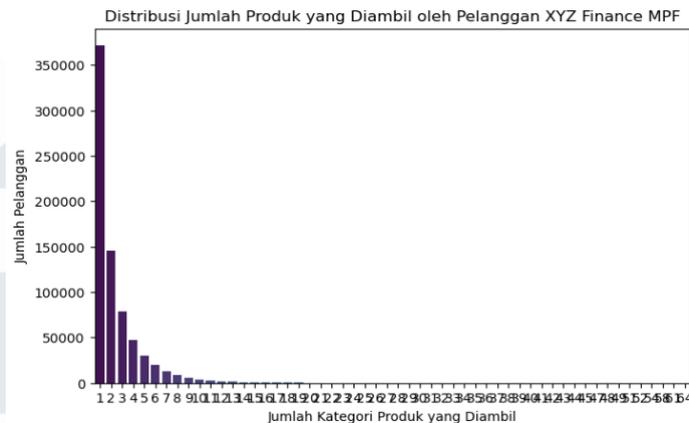


Gambar 3. 25 Rata-rata Pembiayaan per Kategori Produk

Setelah mengevaluasi rata-rata pembiayaan per kategori produk, analisis dilanjutkan dengan mengamati jumlah kategori produk yang diambil per pelanggan. Tujuan analisis ini adalah untuk mengidentifikasi kecenderungan pelanggan dalam mengambil lebih dari satu jenis produk, yang menjadi indikator potensial untuk strategi upselling atau cross-selling.

Gambar 3.26 menunjukkan bahwa meskipun mayoritas pelanggan hanya mengambil satu produk, sebanyak 49,51% atau 364.141 pelanggan tercatat mengambil lebih dari satu kategori. Jumlah pelanggan cenderung menurun seiring bertambahnya kategori yang diambil, namun terdapat segmen kecil yang mengambil hingga lebih dari 20 kategori.

Temuan ini memberikan dasar penting bagi PT XYZ Finance MPF untuk merancang program loyalitas, bundling, serta strategi personalisasi layanan bagi pelanggan yang menunjukkan kecenderungan multi-produk.



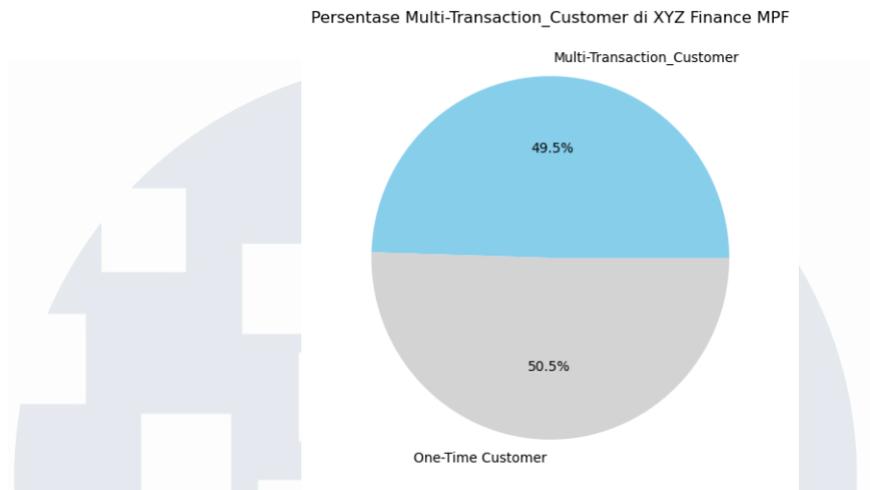
Gambar 3. 26 Distribusi Jumlah Produk yang Diambil oleh Pelanggan

### 9. Proporsi Pelanggan *Multi-Transaction*

Sebagai lanjutan dari analisis jumlah produk yang diambil oleh pelanggan, dilakukan evaluasi terhadap frekuensi transaksi guna mengidentifikasi proporsi pelanggan yang melakukan pembiayaan lebih dari satu kali, atau dikenal sebagai *Multi-Transaction Customer*. Analisis ini penting untuk memahami sejauh mana loyalitas dan kecenderungan retensi pelanggan terhadap layanan yang ditawarkan oleh PT XYZ Finance.

Gambar 3.27 memperlihatkan distribusi pelanggan berdasarkan jumlah transaksi yang dilakukan. Hasil analisis menunjukkan bahwa sebanyak 49,51% pelanggan, atau sekitar 364.141 individu, termasuk dalam kategori *Multi-Transaction Customer*. Artinya, hampir separuh dari total pelanggan pernah kembali untuk mengajukan pembiayaan kedua atau lebih.

Persentase yang relatif seimbang ini menunjukkan adanya potensi besar dalam mempertahankan pelanggan eksisting melalui strategi re-engagement. Temuan ini dapat dimanfaatkan oleh tim pemasaran untuk merancang program loyalitas, komunikasi berbasis frekuensi transaksi, atau penawaran khusus yang ditujukan bagi pelanggan dengan rekam jejak transaksi berulang.



Gambar 3. 27 Persentase Multi-Transaction Customer di PT XYZ FINANCE MPF

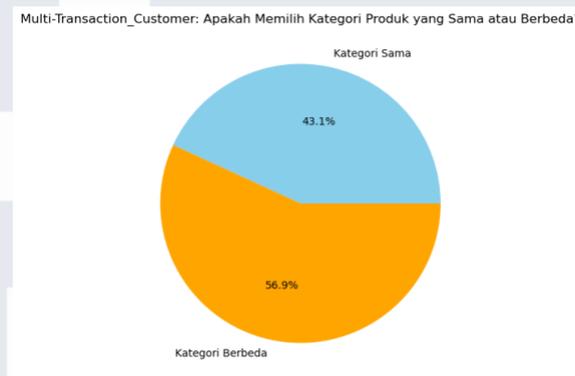
### 10. Preferensi Pelanggan Multi-Transaction Kategori Produk

Setelah mengidentifikasi proporsi pelanggan yang melakukan lebih dari satu transaksi pembiayaan (*Multi-Transaction Customer*), analisis dilanjutkan untuk mengevaluasi apakah pelanggan tersebut cenderung memilih kategori produk yang sama atau berbeda dalam transaksi berikutnya. Wawasan ini penting untuk memahami pola preferensi pelanggan dan potensi pengembangan strategi *cross-selling* yang lebih efektif.

Gambar 3.28 menyajikan hasil visualisasi dalam bentuk diagram lingkaran yang membandingkan proporsi pelanggan yang tetap mengambil produk dari kategori yang sama dengan pelanggan yang berpindah ke kategori berbeda pada transaksi selanjutnya. Berdasarkan hasil analisis, diketahui bahwa sebanyak 56,87% pelanggan memilih kategori produk yang berbeda, sementara 43,13% pelanggan tetap berada pada kategori yang sama.

Temuan ini menunjukkan bahwa sebagian besar pelanggan yang melakukan transaksi ulang cenderung mengeksplorasi produk dari kategori berbeda. Fenomena ini membuka peluang strategis untuk memperluas

penawaran produk dan menyusun paket pembiayaan lintas kategori, yang dapat meningkatkan nilai transaksi per pelanggan. Selain itu, perusahaan juga dapat mengembangkan pendekatan personalisasi berdasarkan preferensi historis pelanggan dalam memilih jenis produk, sehingga strategi pemasaran dapat diarahkan secara lebih presisi.



Gambar 3. 28 Multi-Transaction Customer: Apakah Memilih Kategori Produk yang Sama atau Berbeda?

## 11. Tren Pembiayaan MPF

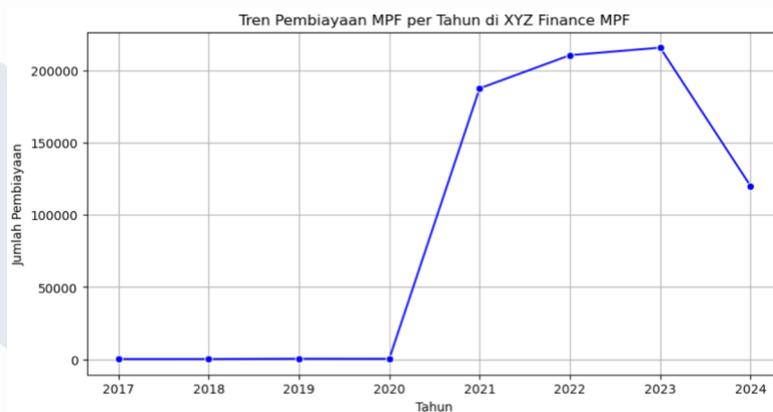
Setelah mengamati karakteristik demografis dan perilaku pengambilan produk, analisis dilanjutkan dengan meninjau tren temporal pembiayaan dari waktu ke waktu. Pemahaman terhadap dinamika pembiayaan berdasarkan dimensi waktu sangat penting untuk mengidentifikasi pola musiman (seasonality), tren pertumbuhan tahunan, serta potensi penurunan yang perlu diantisipasi oleh pihak manajemen.

Gambar 3.29 menunjukkan tren jumlah pembiayaan MPF per tahun. Data memperlihatkan adanya peningkatan yang sangat signifikan sejak tahun 2021, dengan jumlah pembiayaan mencapai puncaknya pada tahun 2023 sebelum mengalami penurunan pada tahun 2024. Hal ini dapat disebabkan oleh berbagai faktor eksternal seperti kondisi ekonomi makro, kebijakan internal perusahaan, atau perubahan perilaku konsumen. Kenaikan tajam pasca-2020 juga dapat dikaitkan dengan pemulihan

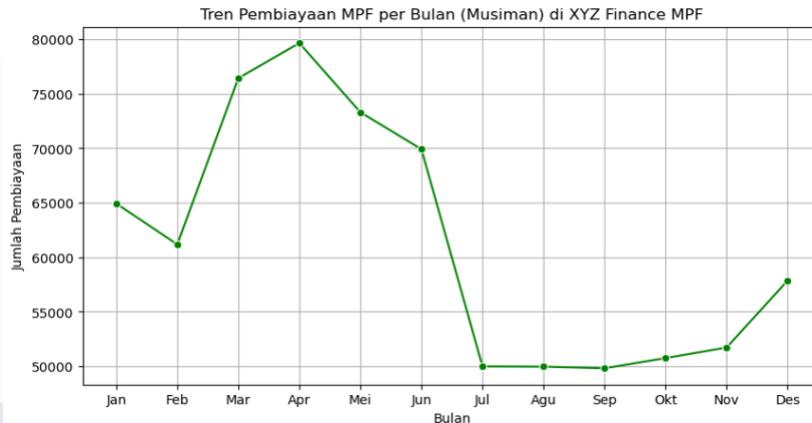
ekonomi pasca-pandemi, yang turut mendorong pertumbuhan permintaan produk pembiayaan.

Sementara itu, Gambar 3.30 memperlihatkan tren pembiayaan MPF berdasarkan bulan (musiman). Terlihat bahwa puncak pembiayaan umumnya terjadi pada bulan April hingga Mei, yang kemungkinan besar berkaitan dengan momen menjelang Hari Raya Idulfitri, di mana kebutuhan konsumtif masyarakat meningkat. Sebaliknya, tren cenderung menurun pada pertengahan tahun (Juli–September) dan kembali meningkat menjelang akhir tahun (Desember), yang dapat dikaitkan dengan momen liburan akhir tahun atau penutupan target tahunan.

Temuan ini memberikan implikasi strategis dalam perencanaan kampanye pemasaran dan alokasi anggaran promosi. Pihak PT XYZ FINANCE dapat memanfaatkan bulan-bulan dengan potensi peningkatan pembiayaan sebagai momentum untuk meningkatkan penetrasi pasar, serta mengantisipasi periode menurun dengan program insentif atau diversifikasi produk.



Gambar 3. 29 Tren Pembiayaan MPF per Tahun di PT XYZ FINANCE MPF



Gambar 3. 30 Tren Pembiayaan MPF per Bulan (Musiman) di PT XYZ FINANCE MPF

### 3.2.1.4 Transformasi Fitur dan Perekayasaan Variabel

Setelah tahap normalisasi format data dilakukan, proses dilanjutkan dengan perekayasaan fitur (feature engineering) untuk menyiapkan data yang lebih representatif dalam proses segmentasi. Tahap ini mencakup proses seleksi dan transformasi beberapa atribut kategorikal menjadi bentuk numerik, serta pembentukan fitur baru berdasarkan informasi yang tersedia.

Langkah awal dalam tahap ini adalah melakukan konversi variabel kategorikal ke dalam bentuk numerik melalui proses *mapping*. Atribut seperti *CUST\_SEX*, *MARITAL\_STAT*, *AREA*, dan *LAST\_MPF\_PURPOSE* masing-masing diberikan nilai angka berdasarkan kategori yang dimiliki. Mapping ini bertujuan untuk mempermudah interpretasi model dan kompatibilitas dalam proses analisis selanjutnya, terutama pada metode clustering yang hanya dapat menerima nilai numerik [24].

Selain itu, dibentuk pula fitur baru dari atribut *MPF\_CATEGORIES\_TAKEN*, yang menunjukkan daftar kategori produk yang pernah diambil pelanggan. Fitur ini dikonversi dengan cara mengelompokkan kategori produk dominan yang dimiliki pelanggan, menjadi empat kelas:

1. Dominan Gadget
2. Dominan Electronic

3. Dominan Furniture
4. Campuran atau Tidak Terdefinisi

Klasifikasi dilakukan dengan mengekstrak kategori dari data string, lalu mengelompokkannya ke dalam kelas dominan berdasarkan urutan kemunculan atau gabungan kategori.

Gambar 3.31 berikut menampilkan potongan kode untuk proses perekayasaan fitur yang dilakukan:

```

Data Formatting
xyzmultipurpose

# Mapping untuk konversi variabel kategorikal menjadi numerik
gender_mapping = {'M': 0, 'F': 1}
marital_mapping = {'S': 0, 'M': 1, 'D': 2}
area_mapping = {area: idx + 1 for idx, area in enumerate(data_xyzmultipurpose['AREA'].unique())}
purpose_mapping = {'GADGET': 1, 'ELECTRONIC': 2, 'FURNITURE': 3, 'OTHER': 4}

# Fungsi untuk mengelompokkan kategori produk yang pernah diambil pelanggan
def categorize_mpf_categories(categories):
    if isinstance(categories, str):
        unique_categories = set(categories.split(', '))
        if unique_categories == {'GADGET'}:
            return 1 # Dominan Gadget
        elif unique_categories == {'ELECTRONIC'}:
            return 2 # Dominan Electronic
        elif unique_categories == {'FURNITURE'}:
            return 3 # Dominan Furniture
        else:
            return 4 # Campuran
    return 4 # Default jika NaN

# Terapkan mapping ke dataset
data_xyzmultipurpose['CUST_SEX'] = data_xyzmultipurpose['CUST_SEX'].map(gender_mapping)
data_xyzmultipurpose['MARITAL_STAT'] = data_xyzmultipurpose['MARITAL_STAT'].map(marital_mapping)
data_xyzmultipurpose['AREA'] = data_xyzmultipurpose['AREA'].map(area_mapping)
data_xyzmultipurpose['LAST_MPF_PURPOSE'] = data_xyzmultipurpose['LAST_MPF_PURPOSE'].map(purpose_mapping)
data_xyzmultipurpose['MPF_CATEGORIES_TAKEN'] = data_xyzmultipurpose['MPF_CATEGORIES_TAKEN'].apply(categorize_mpf_categories)

```

Gambar 3. 31 Penerapan Mapping dan Pengelompokan Kategori Produk ke dalam Fitur Numerik

Selain membentuk fitur berdasarkan atribut eksisting, penulis juga melakukan perekayasaan tambahan untuk menciptakan variabel baru yang dapat memperkaya analisis segmentasi pelanggan. Tiga fitur baru yang dibentuk antara lain adalah: kategori usia pelanggan, status multi-transaction customer, serta kategori pelanggan berdasarkan nilai total pembiayaan. Ketiga fitur ini dirancang untuk memberikan perspektif yang lebih tajam dalam memahami karakteristik dan nilai strategis setiap pelanggan.

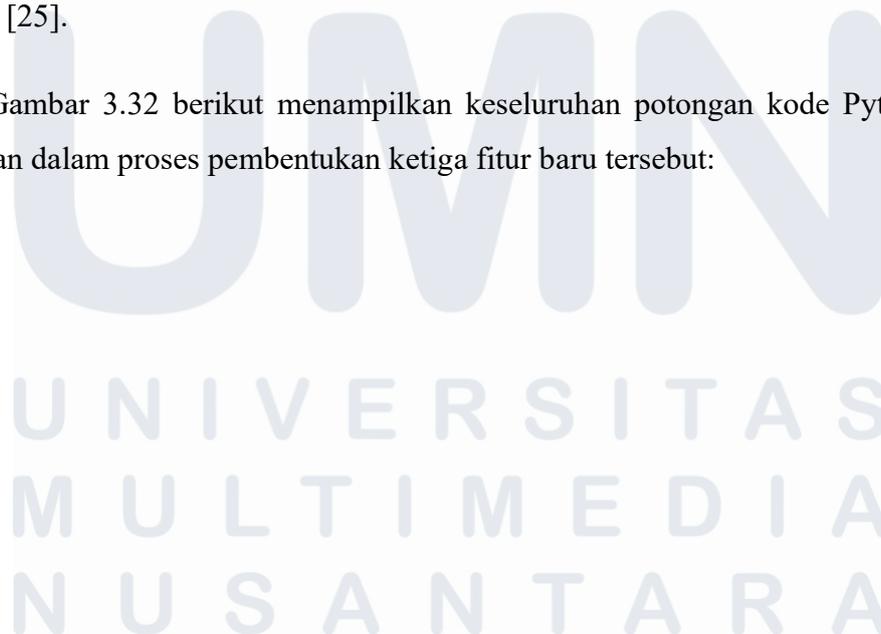
Pertama, dilakukan klasifikasi usia pelanggan ke dalam lima kelompok usia menggunakan fungsi `pd.cut()` dari pustaka *pandas*. Atribut *Usia* dibagi ke dalam

rentang interval tertentu dengan label yang mencerminkan kelompok umur, yaitu: <25, 25–35, 35–45, 45–55, dan 55+. Pengelompokan ini bertujuan untuk memahami dominasi segmen demografis dalam data serta mempermudah analisis komparatif berdasarkan rentang usia.

Kedua, fitur `Multi-Transaction_Customer` dibentuk sebagai variabel biner yang menunjukkan apakah seorang pelanggan tercatat pernah melakukan lebih dari satu transaksi pembiayaan produk Multi-Purpose Financing (MPF). Nilai 1 diberikan untuk pelanggan yang memiliki `TOTAL_PRODUCT_MPF` lebih dari 1, sedangkan nilai 0 diberikan untuk pelanggan dengan hanya satu transaksi atau kurang. Fitur ini berperan penting dalam identifikasi tingkat loyalitas pelanggan terhadap layanan perusahaan.

Ketiga, fitur `Kategori_Customer` digunakan untuk mengelompokkan pelanggan berdasarkan total nominal pembiayaan MPF yang pernah mereka ajukan. Pengelompokan dilakukan secara kuartil menggunakan metode `qcut()` dan menghasilkan empat kategori, yaitu: Low, Medium, High, dan Very High. Klasifikasi ini memungkinkan analisis lanjutan terhadap pelanggan bernilai tinggi (*high-value customers*) dan dapat menjadi acuan dalam strategi personalisasi promosi atau program loyalitas [25].

Gambar 3.32 berikut menampilkan keseluruhan potongan kode Python yang digunakan dalam proses pembentukan ketiga fitur baru tersebut:



```
Kategori Usia Pelanggan

bins = [0, 25, 35, 45, 55, 100]
labels = ['<25', '25-35', '35-45', '45-55', '55+']
data_xyzmultipurpose['Usia_Kategori'] = pd.cut(data_xyzmultipurpose['Usia'], bins=bins, labels=labels, right=False)
]

Multi-Transaction_Customer

data_xyzmultipurpose['Multi-Transaction_Customer'] = data_xyzmultipurpose['TOTAL_PRODUCT_MPF'].apply(lambda x: 1 if x > 1 else 0)
]

Kategori High-Value Customer (Berdasarkan Total Pembiayaan)

data_xyzmultipurpose['Kategori_Customer'] = pd.qcut(data_xyzmultipurpose['TOTAL_AMOUNT_MPF'], q=4, labels=['Low', 'Medium', 'High', 'Very High'])
]
```

Gambar 3. 32 Pembentukan Fitur Tambahan

Ketiga fitur ini menjadi atribut tambahan penting dalam proses analisis selanjutnya, khususnya untuk memahami distribusi pelanggan berdasarkan usia, loyalitas transaksi, serta potensi kontribusi finansial terhadap Perusahaan [26].

### 3.2.1.5 Analisis Feature Importance sebagai Dasar Pemilihan Fitur

Dalam proses segmentasi pelanggan menggunakan algoritma *unsupervised learning*, pemilihan fitur memegang peran yang sangat krusial. Hal ini disebabkan karena algoritma seperti K-Means tidak memiliki variabel target yang dapat dijadikan acuan pembelajaran seperti pada model supervised. K-Means sepenuhnya bergantung pada struktur dan pola kemiripan data berdasarkan nilai-nilai fitur yang digunakan. Oleh karena itu, fitur yang dipilih untuk proses clustering harus mampu merepresentasikan karakteristik utama pelanggan agar klaster yang dihasilkan benar-benar mencerminkan segmentasi yang bermakna dan aplikatif secara bisnis [27].

Untuk mendukung proses pemilihan fitur, dilakukan dua pendekatan analisis: (1) analisis *feature importance* berdasarkan model prediktif, dan (2) analisis visual eksploratif untuk melihat hubungan fitur terhadap keberhasilan komunikasi pelanggan [28].

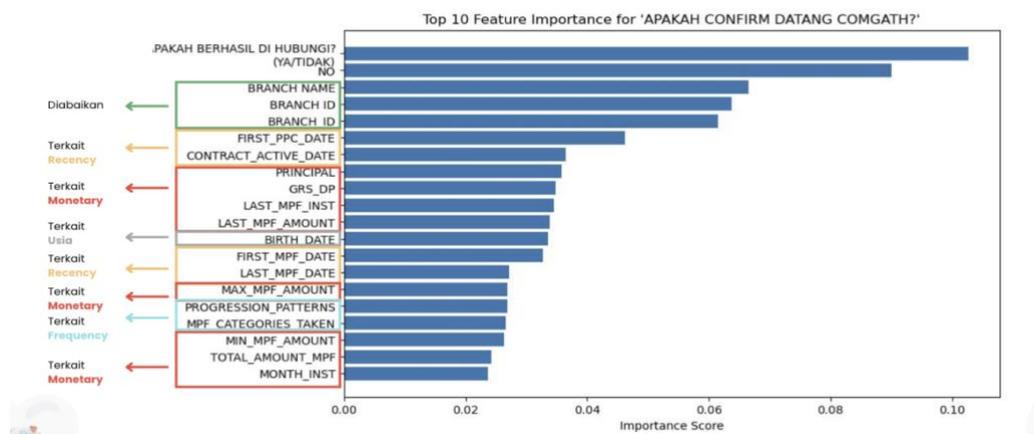
#### 1. Feature Importance Berdasarkan Model

Analisis feature importance dilakukan menggunakan model pembelajaran mesin berbasis pohon keputusan. Model ini digunakan untuk memprediksi keberhasilan pelanggan dalam merespons ajakan untuk hadir dalam sebuah event (variabel target: APAKAH CONFIRM DATANG COMGATH). Model secara otomatis menghitung skor kepentingan (importance score) dari setiap fitur berdasarkan kontribusinya terhadap pemisahan kelas target.

Hasilnya disajikan pada Gambar 3.33, yang menunjukkan sepuluh fitur teratas dengan pengaruh paling signifikan. Beberapa fitur utama yang menonjol meliputi:

- Recency, CONTRACT\_ACTIVE\_DATE, FIRST\_PPC\_DATE → Mewakili dimensi *Recency*
- TOTAL\_PRODUCT\_MPF, MPF\_CATEGORIES\_TAKEN → Mewakili *Frequency*
- TOTAL\_AMOUNT\_MPF, MAX\_MPF\_AMOUNT, PRINCIPAL → Mewakili *Monetary*
- BIRTH\_DATE → Digunakan untuk menghitung *Usia*

Fitur-fitur ini kemudian dijadikan dasar dalam menyusun variabel utama yang digunakan pada proses clustering.



Gambar 3. 33 Top 10 Feature Importance terhadap Kehadiran

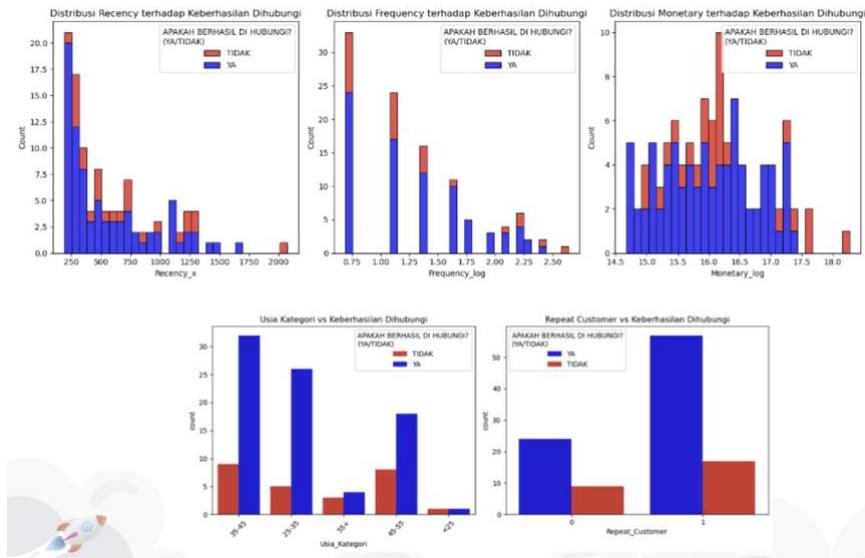
## 2. Validasi Eksploratif melalui Visualisasi Distribusi

Sebagai validasi tambahan, dilakukan analisis eksploratif visual untuk memastikan bahwa fitur-fitur yang terpilih memang menunjukkan hubungan yang konsisten terhadap outcome target. Visualisasi ini membandingkan nilai masing-masing fitur terhadap variabel keberhasilan dihubungi.

Berdasarkan hasil visualisasi pada Gambar 3.34, ditemukan bahwa:

- a) Pelanggan dengan Recency yang lebih kecil (transaksi terbaru) memiliki kemungkinan lebih besar untuk merespons.
- b) Nilai Frequency dan Monetary yang lebih tinggi cenderung berkorelasi dengan keberhasilan dihubungi.
- c) Usia pelanggan yang berada pada rentang 25–45 tahun mendominasi respons positif.
- d) Pelanggan dengan nilai Multi-Transaction\_Customer = 1 (lebih dari satu transaksi) menunjukkan tingkat respons yang lebih tinggi dibandingkan pelanggan satu kali transaksi.

Temuan ini memperkuat keputusan untuk menggunakan lima fitur utama dalam proses segmentasi pelanggan: Recency, TOTAL\_PRODUCT\_MPF, TOTAL\_AMOUNT\_MPF, Multi-Transaction\_Customer, dan Usia.



Gambar 3. 34 Distribusi Fitur Utama terhadap Keberhasilan Komunikasi Pelanggan

### 3.2.1.6 Transformasi dan Standarisasi Fitur untuk Clustering

Setelah menentukan fitur-fitur yang relevan melalui analisis *feature importance*, tahap selanjutnya adalah melakukan transformasi dan standarisasi terhadap fitur-fitur tersebut agar dapat digunakan dalam proses segmentasi pelanggan. Lima fitur utama yang digunakan adalah: Recency, Frequency\_log, Monetary\_log, Multi-Transaction\_Customer, dan Usia\_Segment.

Fitur Recency dihitung sebagai selisih hari antara tanggal referensi (31 Desember 2024) dengan tanggal transaksi terakhir pelanggan (LAST\_MPF\_DATE). Sementara itu, fitur Frequency dan Monetary yang berasal dari jumlah produk MPF dan total pembiayaan, terlebih dahulu dilakukan transformasi logaritmik dengan fungsi  $\log_{1p}$  untuk mengurangi skewness. Fitur Usia\_Segment dibentuk berdasarkan segmentasi rentang usia 25–50 tahun, yang dianggap sebagai kelompok usia produktif yang paling banyak merespons undangan.

untuk menyamakan skala antarfitur sebelum dilakukan proses klusterisasi. Normalisasi ini penting agar fitur dengan nilai besar seperti Monetary\_log tidak mendominasi proses pembentukan kluster.

Setelah proses transformasi dan normalisasi selesai, dilakukan segmentasi pelanggan menggunakan algoritma K-Means dengan jumlah kluster optimal sebanyak 4. Pemilihan nilai  $k = 4$  didasarkan pada evaluasi *elbow method* dan hasil backtest segmentasi terhadap variabel target [29].

```
# Tentukan tanggal referensi
today_date = datetime.datetime(2024, 12, 31)

# Konversi_LAST_MPF_DATE menjadi datetime
data_xyzmultipurpose['LAST_MPF_DATE'] = pd.to_datetime(data_xyzmultipurpose['LAST_MPF_DATE'])

# Hitung Recency
data_xyzmultipurpose['Recency'] = (today_date - data_xyzmultipurpose['LAST_MPF_DATE']).dt.days

rfm = data_xyzmultipurpose.groupby('CUST_NO').agg({
    'Recency': 'min',
    'TOTAL_PRODUCT_MPF': 'sum',
    'TOTAL_AMOUNT_MPF': 'sum',
    'Multi-Transaction_Customer': 'max',
    'Usia': 'max'
}).reset_index()

rfm.rename(columns={'TOTAL_PRODUCT_MPF': 'Frequency', 'TOTAL_AMOUNT_MPF': 'Monetary'}, inplace=True)

rfm['Frequency_log'] = np.log1p(rfm['Frequency'])
rfm['Monetary_log'] = np.log1p(rfm['Monetary'])

rfm['Usia_Segment'] = rfm['Usia'].apply(lambda x: 1 if 25 <= x <= 50 else 0)

features = ['Recency', 'Frequency_log', 'Monetary_log', 'Multi-Transaction_Customer', 'Usia_Segment']
rfm_norm = rfm[features].apply(zscore)

plt.show()

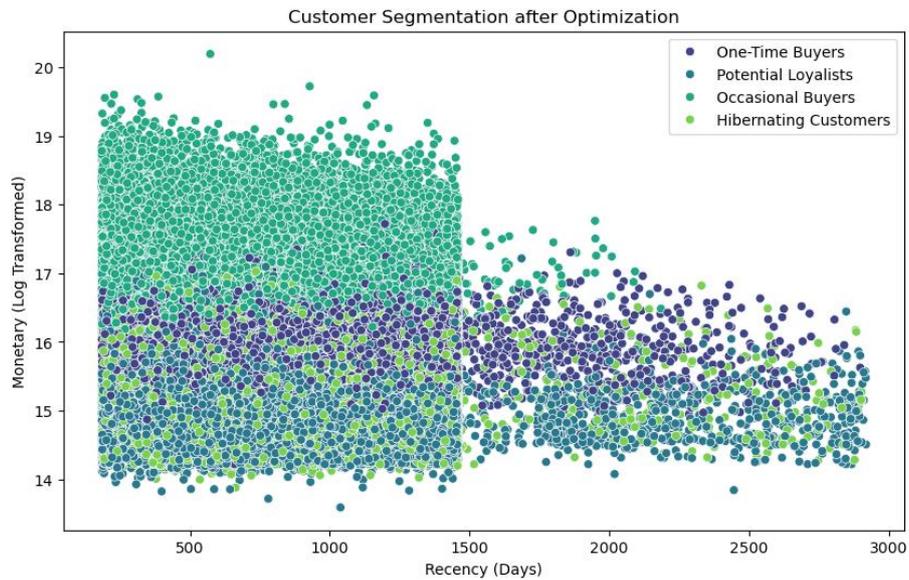
optimal_k = 4
kmeans = KMeans(n_clusters=optimal_k, init='k-means++', random_state=42)
rfm['Cluster'] = kmeans.fit_predict(rfm_norm)
```

Gambar 3. 35 Proses Segmentasi

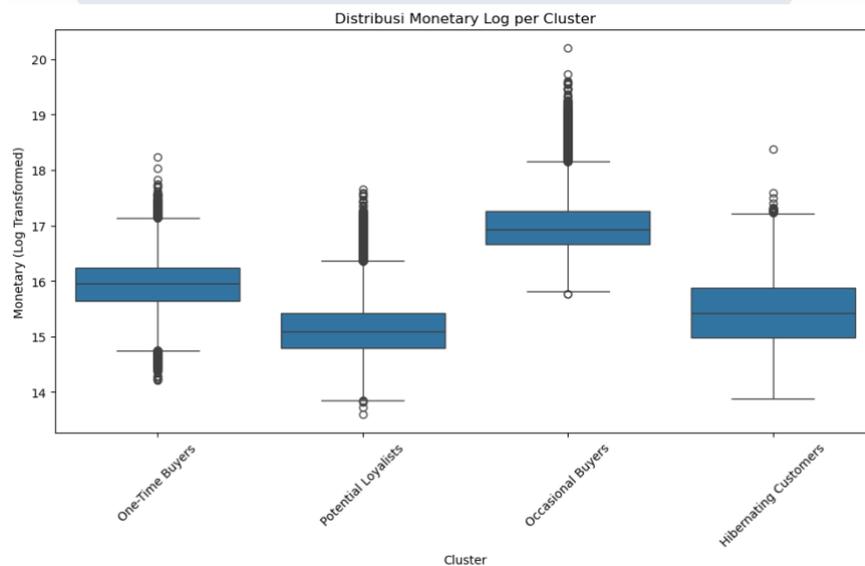
Keempat kluster yang terbentuk dimaknai secara bisnis menjadi:

- a) **Potential Loyalists**
- b) **One-Time Buyers**
- c) **Occasional Buyers**
- d) **Hibernating Customers**

Setiap kluster selanjutnya dipetakan ke dalam keputusan apakah pelanggan tersebut layak untuk diundang kembali atau tidak. Visualisasi hasil segmentasi ditampilkan dalam bentuk scatterplot dan boxplot, sebagaimana dapat dilihat pada Gambar 3.36 dan Gambar 3.37.



Gambar 3. 36 Visualisasi hasil clustering



Gambar 3. 37 Distribusi nilai Monetary (log) pada masing-masing kluster pelanggan

### 3.2.1.7 Analisis Karakteristik Segmentasi Pelanggan

Untuk memahami lebih dalam karakteristik tiap segmen yang dihasilkan, dilakukan perhitungan nilai statistik deskriptif (mean, median, dan standar deviasi) untuk masing-masing fitur pada setiap segmen pelanggan [30]. Analisis ini dilakukan menggunakan fungsi `groupby()` terhadap kolom `Segmentasi_optimal`.

Hasil perhitungan ditampilkan pada Tabel 3.3, yang memperlihatkan bahwa:

- a) **Hibernating Customers** memiliki Recency rata-rata tertinggi, Frequency dan Monetary terendah, serta mayoritas tidak berada dalam kelompok usia produktif (nilai Usia\_Segment = 0).
- b) **Occasional Buyers** menunjukkan frekuensi dan nilai pembiayaan (Monetary) yang tinggi, serta memiliki proporsi besar dari kelompok usia produktif.
- c) **One-Time Buyers** cenderung memiliki frekuensi pembelian satu kali, namun memiliki nilai pembiayaan yang moderat.
- d) **Potential Loyalists** memiliki Recency relatif lebih pendek dari Hibernating, namun masih menunjukkan potensi loyalitas karena termasuk dalam kelompok usia produktif.

Tabel 3. 3 Rangkuman karakteristik statistik tiap segmen pelanggan berdasarkan hasil clustering

Segmentasi Optimal	Recency (Mean, Hari)	Frequency (Mean, Aktual)	Frequency_log (Mean)	Monetary (Mean, Rp)	Monetary_log (Mean)	Multi-Transaction Customer (Mean)	Usia Segment (Mean)
Hibernating Customers	1.158 (Sangat tinggi)	3,68	1,3	Rp9,72 juta	15,99	0,997	76,60%
Occasional Buyers	489 (Menengah)	3,55	1,27	Rp9,97 juta	16,01	0,999	83,80%
One-Time Buyers	791 (Tinggi)	1,00 (Paling rendah)	0,69 (Paling rendah)	Rp4,25 juta (Paling rendah)	15,15	0,000 (tidak lebih dari satu transaksi)	79,10%
Potential Loyalists	692 (Menengah)	7,79 (Tertinggi)	2,05 (Tertinggi)	Rp30 juta (Tertinggi)	17,1	0,9999	80,10%

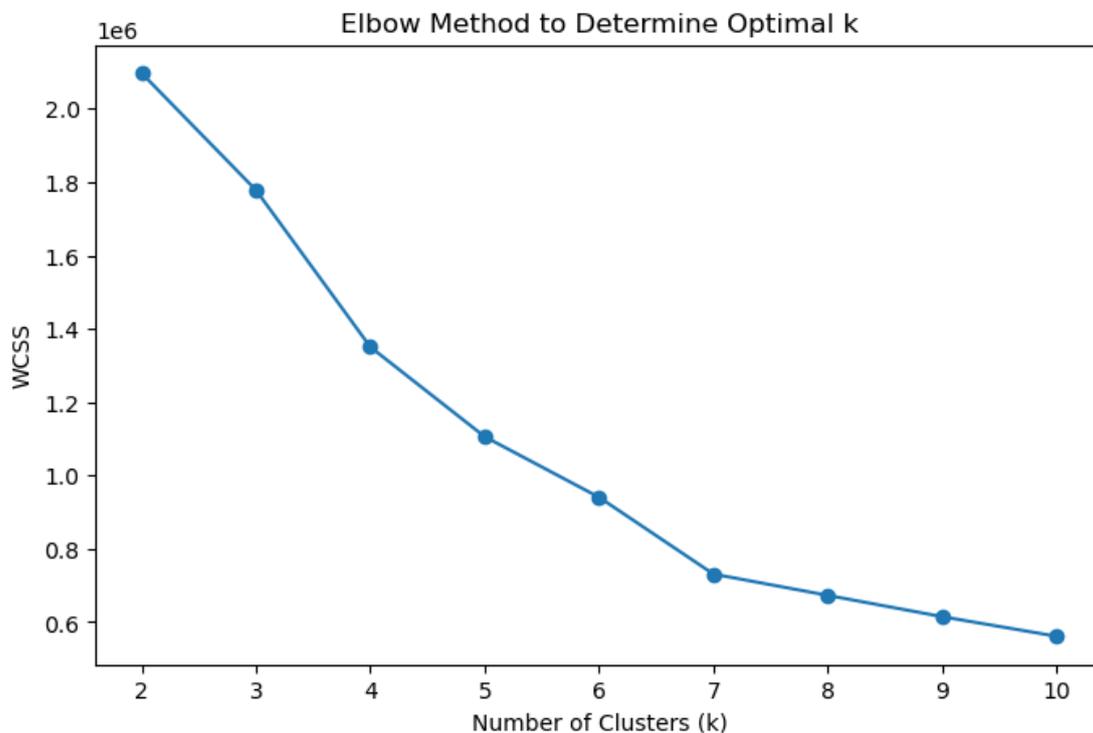
Berdasarkan Tabel 3.3, karakteristik dari masing-masing segmen pelanggan dapat diidentifikasi melalui nilai rata-rata (*mean*) dari variabel-variabel RFM yang telah ditransformasi, serta dua variabel tambahan yaitu *Multi-Transaction\_Customer* dan *Usia\_Segment*[31]. Selanjutnya, keputusan apakah suatu segmen layak diundang atau tidak untuk program pemasaran, dijelaskan melalui analisis terhadap perilaku dan profil masing-masing segmen sebagai berikut:

1. Hibernating Customers tidak termasuk dalam kategori pelanggan yang layak diundang. Hal ini dikarenakan kelompok ini memiliki nilai *Recency* yang sangat tinggi, menunjukkan bahwa pelanggan sudah lama tidak melakukan transaksi. Meskipun terdapat sebagian kecil dari kelompok ini yang memiliki lebih dari satu transaksi (*Multi-Transaction\_Customer* = 0,997), namun probabilitas pelanggan untuk kembali melakukan pembelian tergolong rendah. Selain itu, proporsi kelompok usia produktif juga rendah (76,6%), yang semakin memperkuat potensi churn dari segmen ini.
2. Occasional Buyers merupakan kelompok yang layak diundang. Segmen ini memiliki frekuensi transaksi dan nilai pembiayaan (*Monetary*) yang cukup tinggi, dengan rata-rata *Monetary* mencapai Rp9,97 juta dan *Frequency* sebesar 3,55. Nilai *Multi-Transaction\_Customer* yang mendekati 1 (0,999) menunjukkan bahwa pelanggan dalam kelompok ini aktif bertransaksi lebih dari satu kali. Proporsi usia produktif yang cukup besar (83,8%) turut menunjukkan adanya potensi pembelian lanjutan.
3. One-Time Buyers tidak diundang karena hanya menunjukkan pola pembelian satu kali, yang tercermin dari nilai *Multi-Transaction\_Customer* sebesar 0,000 dan nilai frekuensi terendah di antara seluruh segmen (*Frequency* = 1,00). Meskipun sebagian besar berasal dari kelompok usia produktif (79,1%), rendahnya keterlibatan transaksi menjadikan kelompok ini kurang potensial untuk ditargetkan kembali dalam aktivitas pemasaran.
4. Potential Loyalists merupakan segmen yang layak diundang karena memiliki karakteristik yang menonjol dalam hal nilai pembiayaan (*Monetary*) tertinggi (Rp30 juta) dan frekuensi pembelian yang paling besar (*Frequency* = 7,79). Segmen ini juga didominasi oleh kelompok usia produktif (80,1%), dan memiliki potensi loyalitas yang tinggi untuk dipertahankan sebagai pelanggan tetap.

### 3.2.1.8 Validasi Jumlah Cluster Melalui Fine Tuning

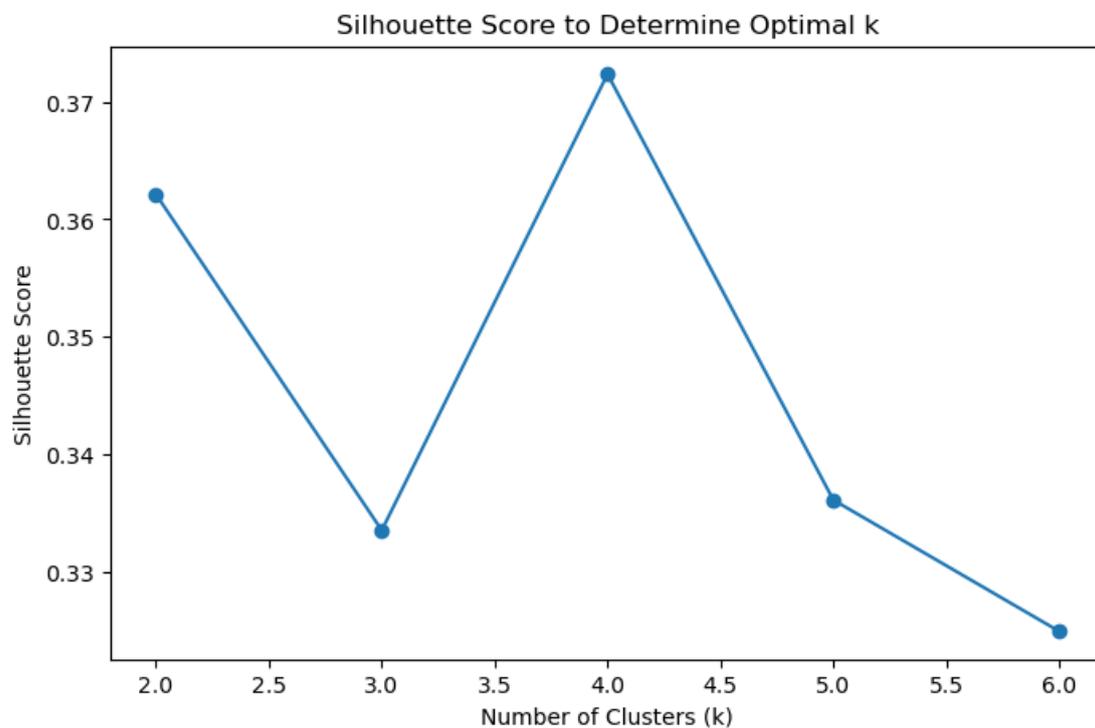
Setelah menentukan jumlah cluster sebanyak empat (4) berdasarkan pertimbangan karakteristik pelanggan serta kebutuhan segmentasi yang relevan secara bisnis, dilakukan proses validasi guna memastikan bahwa jumlah cluster tersebut juga optimal secara statistik. Validasi dilakukan melalui dua pendekatan, yaitu menggunakan metode *Elbow Method* dan evaluasi *Silhouette Score* terhadap hasil segmentasi [32].

*Elbow Method* dilakukan dengan menghitung nilai WCSS (*Within-Cluster Sum of Squares*) pada rentang jumlah cluster dari 2 hingga 10. Titik siku atau “elbow” yang terbentuk pada grafik WCSS menunjukkan jumlah cluster optimal yang memberikan keseimbangan antara kompleksitas model dan tingkat variasi data yang mampu dijelaskan. Pada visualisasi grafik *Elbow Method* di bawah, terlihat bahwa titik siku terjadi pada  $k = 4$ .



Gambar 3. 38 Grafik Elbow Method untuk Menentukan Jumlah Cluster Optimal

Selain itu, dilakukan pula evaluasi dengan menggunakan *Silhouette Score*, yaitu metrik yang mengukur sejauh mana suatu data berada dalam cluster yang benar dibandingkan dengan cluster lainnya. Nilai *Silhouette Score* yang lebih tinggi menunjukkan pemisahan cluster yang lebih baik [33]. Proses ini dilakukan dengan teknik *MiniBatchKMeans* dan pendekatan sampling sebesar 10% dari keseluruhan data, untuk menghemat waktu komputasi tanpa kehilangan representasi karakteristik data. Hasil evaluasi menunjukkan bahwa nilai *Silhouette Score* tertinggi juga diperoleh pada saat jumlah cluster adalah empat ( $k = 4$ ).



Gambar 3. 39 Grafik Silhouette Score terhadap Jumlah Cluster

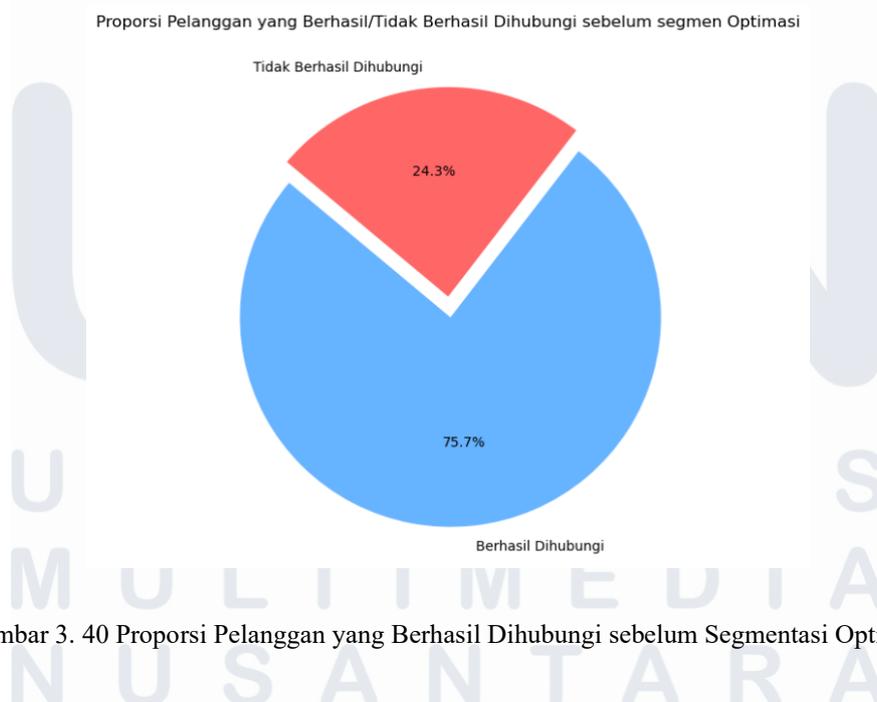
Berdasarkan kedua metode validasi tersebut, dapat disimpulkan bahwa jumlah cluster sebanyak empat (4) adalah keputusan yang optimal, baik secara pertimbangan bisnis maupun analisis statistik. Validasi ini mendukung bahwa pemodelan segmentasi pelanggan yang dilakukan sebelumnya telah sesuai dan dapat dijadikan dasar untuk strategi pemasaran yang lebih terarah [34].

### 3.2.1.9 Validasi Segmentasi melalui Backtesting Kehadiran Pelanggan

Setelah menentukan segmentasi optimal sebanyak empat klaster menggunakan algoritma K-Means, dilakukan proses validasi untuk menilai efektivitas segmentasi tersebut dalam konteks event marketing [35]. Validasi dilakukan dengan mencocokkan hasil segmentasi terhadap data aktual kehadiran pelanggan pada acara yang diselenggarakan, melalui proses *backtesting*.

Pertama, data hasil segmentasi digabungkan (*merge*) dengan data kehadiran pelanggan (Konfirmasi Kehadiran Comgath.xlsx) menggunakan atribut *CUST\_NO*. Selanjutnya, dilakukan normalisasi nilai dalam kolom APAKAH BERHASIL DI HUBUNGI? dan APAKAH CONFIRM DATANG COMGATH? untuk memastikan keseragaman data. Dari hasil penggabungan ini diperoleh sebanyak 1.325 entri pelanggan yang dapat diverifikasi berdasarkan nomor handphone.

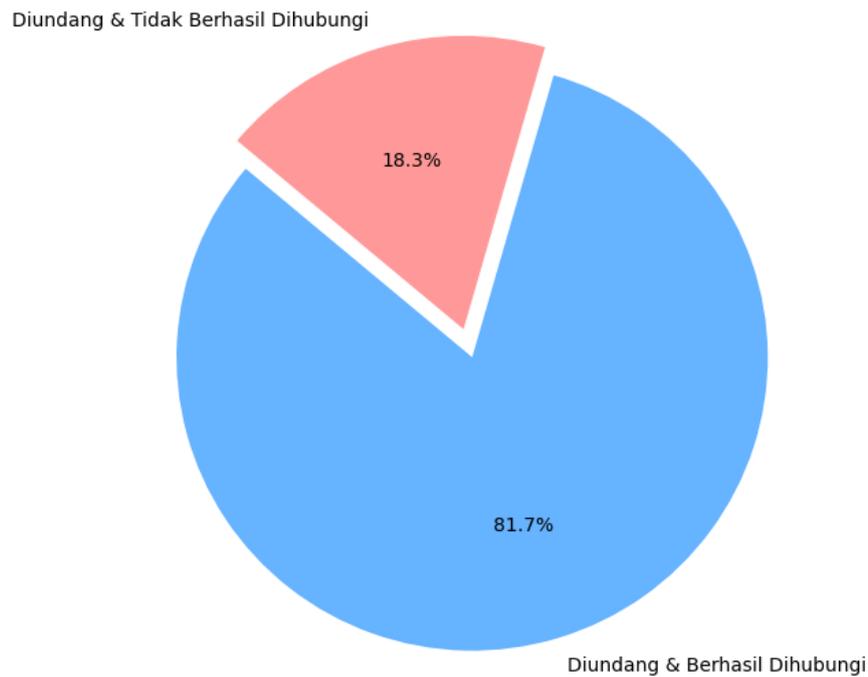
Sebelum proses optimasi segmentasi dilakukan, proporsi keberhasilan dalam menghubungi pelanggan ditunjukkan pada Gambar 3.22. Sebanyak 75,7% pelanggan berhasil dihubungi, sementara sisanya sebesar 24,3% tidak berhasil dihubungi.



Gambar 3. 40 Proporsi Pelanggan yang Berhasil Dihubungi sebelum Segmentasi Optimasi

Setelah dilakukan seleksi pelanggan berdasarkan segmentasi yang optimal (hanya yang layak diundang), proporsi keberhasilan meningkat menjadi 81,7% (Gambar 3.41), yang menunjukkan efektivitas dari strategi seleksi berbasis segmentasi.

Proporsi Pelanggan Diundang yang Berhasil/Tidak Berhasil Dihubungi Setelah segmen Optimasi

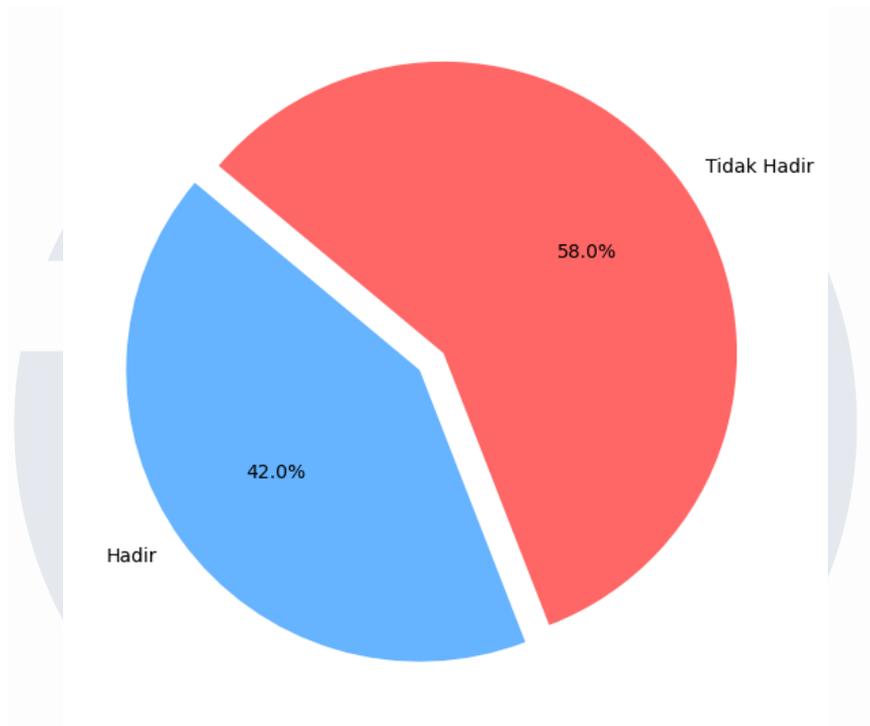


Gambar 3. 41 Proporsi Pelanggan Diundang yang Berhasil Dihubungi setelah Segmentasi Optimasi

Selanjutnya, dilakukan pengukuran tingkat kehadiran pelanggan pada event sebelum penerapan segmentasi. Hasilnya menunjukkan bahwa tingkat kehadiran hanya mencapai 42% (Gambar 3.42), dengan 58% pelanggan tidak hadir meskipun telah dihubungi.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

Proporsi Kehadiran di Event Tanpa Segmen Optimasi

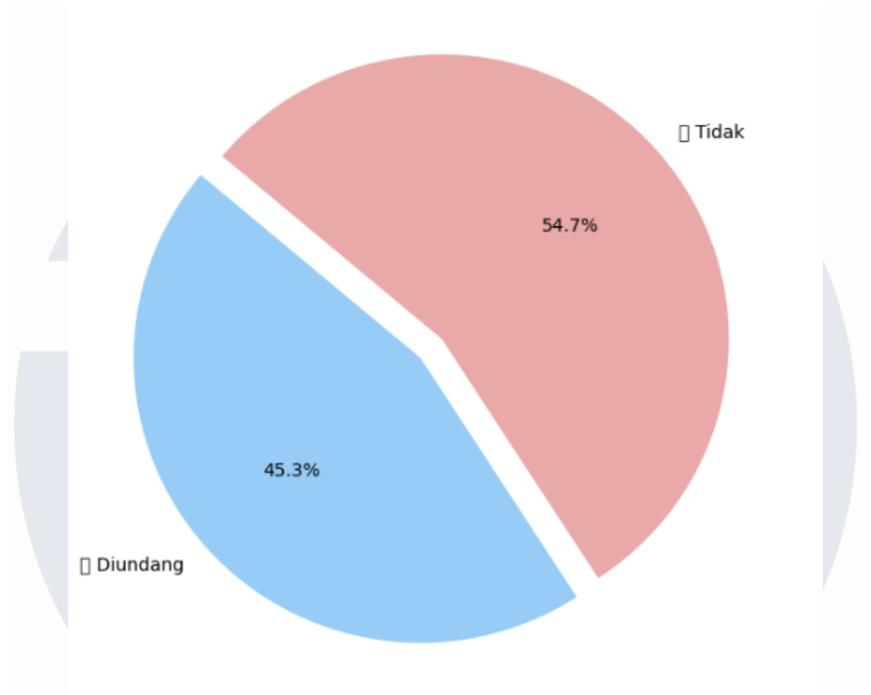


Gambar 3. 42 Proporsi Kehadiran Pelanggan tanpa Segmentasi Optimasi

Sebagai pembandingan, evaluasi dilakukan terhadap efektivitas undangan dan respons berdasarkan hasil segmentasi. Gambar 3.43 memperlihatkan bahwa pelanggan yang diundang berdasarkan hasil segmentasi memiliki tingkat kehadiran sebesar 45,3%. Selain itu, pada Gambar 3.44, segmentasi "Potential Loyalists" merupakan kelompok dengan proporsi kehadiran tertinggi, yaitu sebesar 32,3%, diikuti oleh "One-Time Buyers", "Hibernating Customers", dan "Occasional Buyers".

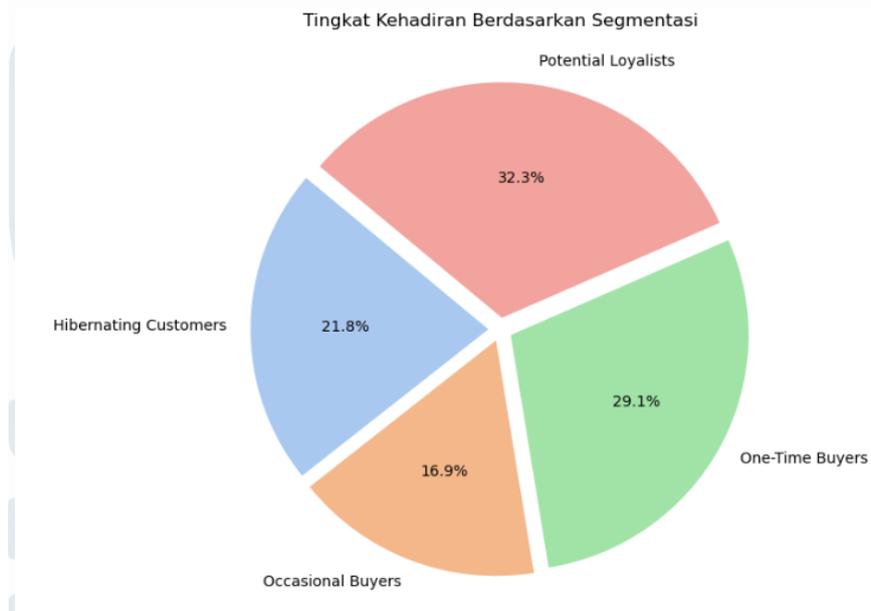
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Tingkat Kehadiran Berdasarkan Status Undangan



Gambar 3. 43 Tingkat Kehadiran Berdasarkan Status Undangan

Tingkat Kehadiran Berdasarkan Segmentasi

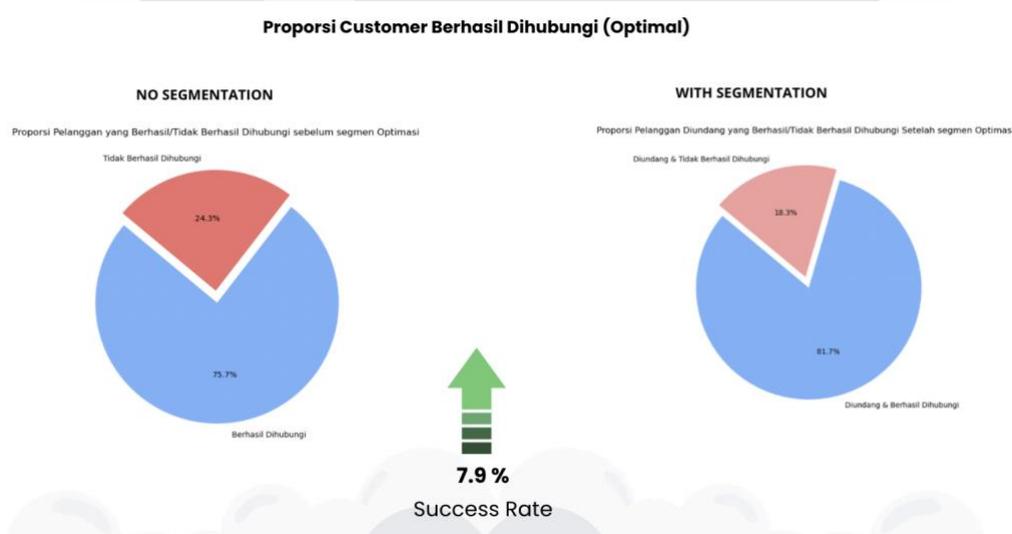


Gambar 3. 44 Tingkat Kehadiran Berdasarkan Segmentasi Pelanggan

Berdasarkan hasil validasi ini, dapat disimpulkan bahwa penggunaan segmentasi pelanggan untuk menentukan sasaran undangan terbukti meningkatkan proporsi pelanggan yang berhasil dihubungi dan berkontribusi pada peningkatan partisipasi pelanggan dalam event Perusahaan [36].

### 3.2.1.10 Evaluasi Efektivitas Segmentasi Melalui Backtesting

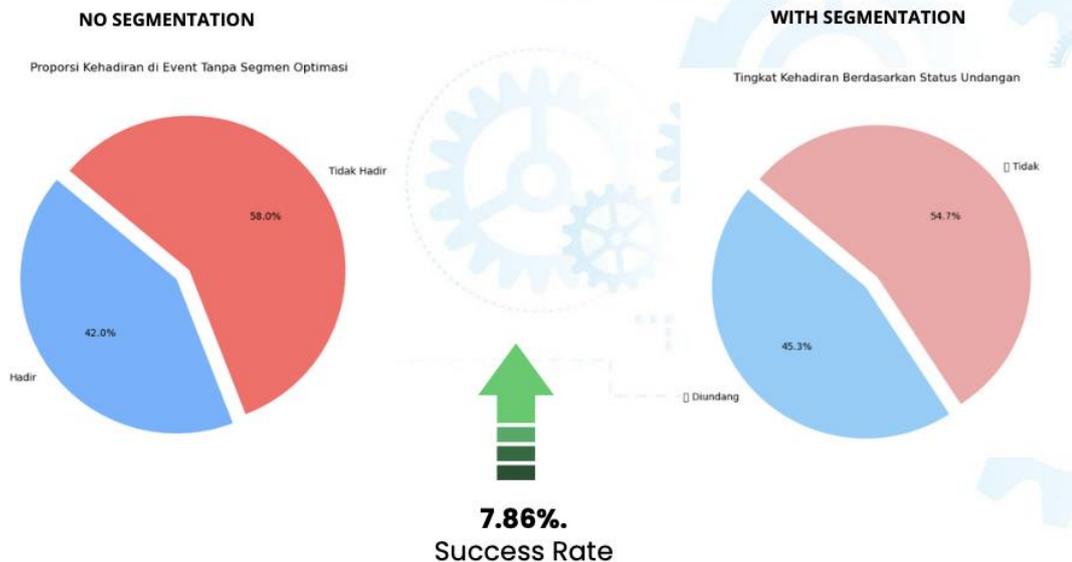
Untuk menguji dampak nyata dari segmentasi yang telah dikembangkan, dilakukan simulasi backtesting dengan membandingkan dua kondisi: tanpa segmentasi (baseline) dan dengan segmentasi (hasil model K-Means optimal). Hasil evaluasi ditunjukkan pada Gambar 3.45 dan Gambar 3.46.



Gambar 3. 45 Perbandingan Proporsi Keberhasilan Menghubungi Pelanggan

Simulasi menunjukkan bahwa keberhasilan menghubungi pelanggan mengalami peningkatan sebesar **+7,9%** setelah penerapan segmentasi. Hal ini mengindikasikan bahwa strategi segmentasi mampu menysasar pelanggan dengan lebih tepat.

## Proporsi Customer Hadir



Gambar 3. 46 Perbandingan Proporsi Kehadiran Pelanggan

Evaluasi terhadap kehadiran pelanggan dalam event juga menunjukkan hasil yang serupa. Berdasarkan hasil backtesting, segmentasi yang diterapkan berpotensi memberikan dampak positif terhadap tingkat kehadiran pelanggan dengan rincian sebagai berikut:

- Tanpa segmentasi: 42,0% pelanggan hadir
- Dengan segmentasi: 45,3% pelanggan hadir
- Peningkatan potensial: +7,86%

Pendekatan segmentasi ini menunjukkan bahwa pelanggan dari segmen yang terpilih memiliki kecenderungan lebih tinggi untuk hadir dalam event. Meskipun hasil ini masih bersifat simulatif dan belum diuji secara operasional, tren tersebut mengindikasikan bahwa segmentasi dapat menjadi alat bantu dalam meningkatkan efektivitas undangan event.

Secara keseluruhan, segmentasi berbasis data ini dapat dijadikan sebagai rekomendasi awal (baseline) dalam menyusun strategi komunikasi dan undangan pada kegiatan event XYZ Multipurpose berikutnya, dengan harapan peningkatan lebih lanjut setelah diimplementasikan di lapangan.

### **3.2.2 Pengembangan Aplikasi Web Segmentasi Pelanggan XYZ Finance MPF**

#### **3.2.2.1 Latar Belakang**

Berdasarkan hasil segmentasi pelanggan yang telah berhasil dilakukan, penulis mengidentifikasi beberapa kendala operasional yang dihadapi tim XYZ Multipurpose. Proses segmentasi memerlukan keahlian programming Python, analisis manual membutuhkan waktu 2-3 hari kerja, dan tidak ada standarisasi interpretasi hasil antar stakeholder. Tim marketing tidak dapat melakukan analisis mandiri ketika ada data baru dan proses sharing melalui Excel statis menyulitkan kolaborasi.

Untuk mengatasi tantangan tersebut, penulis mengembangkan aplikasi web interaktif menggunakan framework Streamlit yang memungkinkan tim operasional melakukan segmentasi pelanggan secara mandiri. Aplikasi dirancang dengan prinsip "self-service analytics" yang dapat diakses melalui browser tanpa instalasi software khusus, meningkatkan efisiensi proses dari 2-3 hari menjadi 15-30 menit [37].

#### **3.2.2.2 Desain dan Arsitektur Aplikasi**

Aplikasi segmentasi pelanggan yang dikembangkan dirancang dengan arsitektur modular guna memastikan kemudahan dalam proses maintenance, kolaborasi tim, dan pengembangan fitur lanjutan. Setiap fungsi utama dalam aplikasi dipisahkan ke dalam berkas atau direktori yang memiliki tanggung jawab spesifik. Pendekatan ini tidak hanya memperjelas struktur kode, tetapi juga meningkatkan skalabilitas sistem ketika diintegrasikan ke dalam proses bisnis perusahaan.

Struktur aplikasi terdiri atas berkas utama app.py yang berfungsi sebagai titik awal eksekusi dan pengatur alur routing. Direktori pages/ berisi berbagai modul yang mencakup proses unggah data (upload.py), eksplorasi data (eda.py), penerapan

algoritma segmentasi (`segmentation.py`), tampilan dashboard interaktif (`dashboard.py`), serta ekspor hasil analisis (`export.py`). Untuk menunjang kebutuhan teknis tambahan, direktori `utils/` menyimpan fungsi pemrosesan (`data_utils.py`) dan visualisasi (`visualization.py`). Selanjutnya, berkas-berkas terkait penataan gaya antarmuka dikonsolidasikan di dalam direktori `styles/`, sedangkan aset statis seperti logo perusahaan disimpan di dalam `assets/`.

Gambar 3.47 berikut menampilkan struktur hierarkis dari aplikasi web dashboard segmentasi pelanggan yang telah dikembangkan selama program magang.

```
├─ app.py           # Main application dan routing
├─ pages/          # Modul halaman aplikasi
│   └─ upload.py   # Upload dan preprocessing data
│   └─ eda.py      # Exploratory data analysis
│   └─ segmentation.py # Algoritma segmentasi
│   └─ dashboard.py # Dashboard interaktif
│   └─ export.py   # Export dan dokumentasi
├─ utils/          # Utility functions
│   └─ data_utils.py # Fungsi pemrosesan data
│   └─ visualization.py # Fungsi visualisasi
├─ styles/         # CSS styling
│   └─ style.css   # Custom styling
└─ assets/         # Static assets (logo, dll)
```

Gambar 3. 47 Struktur Aplikasi Web Dashboard Segmentasi

Sebagaimana terlihat pada struktur di atas, pembagian per modul dilakukan secara eksplisit untuk menghindari duplikasi fungsi dan meningkatkan keterbacaan kode program. Hal ini memungkinkan tim pengembang untuk melakukan pembaruan maupun debugging secara terfokus pada bagian yang relevan tanpa memengaruhi keseluruhan sistem.

### 3.2.2.3 Technology Stack yang Digunakan

Aplikasi dibangun menggunakan beberapa teknologi utama yang dipilih berdasarkan kebutuhan fungsional dan kemudahan penggunaan. Streamlit dipilih sebagai framework utama untuk pengembangan aplikasi web karena kemampuannya dalam rapid prototyping dan native support untuk data science applications. Pandas dan NumPy

digunakan sebagai library utama untuk manipulasi dan analisis data, sementara Scikit-learn diimplementasikan untuk algoritma K-Means clustering sesuai dengan metode yang telah dikembangkan dalam penelitian.

Plotly diintegrasikan untuk menghasilkan visualisasi interaktif yang memungkinkan user melakukan eksplorasi data secara mendalam. OpenPyXL dan XlsxWriter digunakan untuk pemrosesan file Excel, baik untuk input data pelanggan maupun output hasil segmentasi [38]. GitHub dipilih sebagai platform version control dan repository management yang memfasilitasi deployment aplikasi melalui Streamlit Community Cloud.

#### **3.2.2.4 Alur Kerja Aplikasi**

Aplikasi dirancang dengan alur kerja yang mengikuti prinsip user experience yang intuitif dan sequential untuk memastikan kemudahan penggunaan bagi tim non-teknis. Proses dimulai dengan tahap upload data, di mana user mengupload file Excel yang berisi data pelanggan XYZ Multipurpose melalui interface drag-and-drop yang telah disediakan.

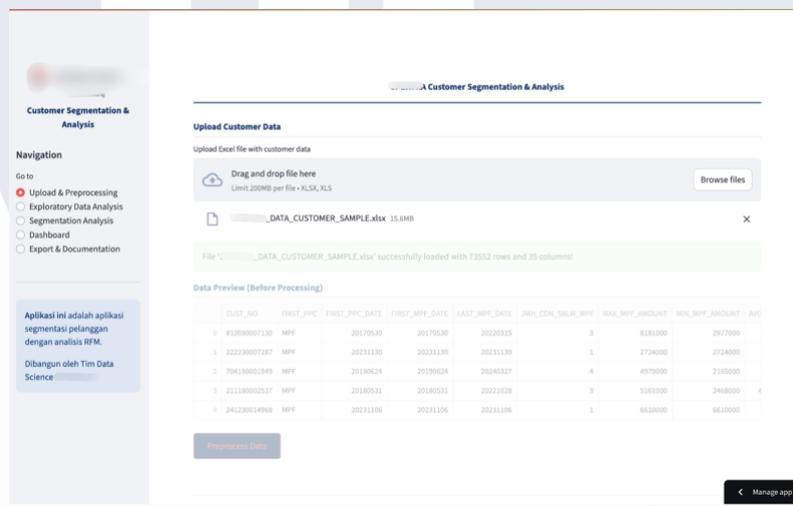
Tahap kedua adalah preprocessing otomatis, di mana sistem melakukan pembersihan dan normalisasi data sesuai dengan standar yang telah ditetapkan dalam penelitian, termasuk konversi tipe data, handling missing values, dan feature engineering. Tahap ketiga memungkinkan user melakukan exploratory data analysis untuk memahami karakteristik data pelanggan melalui visualisasi interaktif.

Tahap keempat merupakan proses segmentasi, di mana sistem menerapkan algoritma RFM dan K-Means clustering dengan metode optimal yang telah dikembangkan untuk menghasilkan empat segmen pelanggan. Tahap kelima menyajikan dashboard interaktif yang menampilkan hasil segmentasi dan rekomendasi undangan berdasarkan karakteristik setiap segmen. Tahap terakhir adalah export, di mana user dapat mengunduh hasil segmentasi dalam berbagai format sesuai kebutuhan operasional.

### 3.2.2.5 Implementasi Fitur-Fitur Utama

#### 1. Upload & Preprocessing Module

Fitur pertama yang diimplementasikan adalah modul upload data yang mengotomatisasi proses preprocessing sesuai dengan standar yang telah ditetapkan dalam penelitian. Modul ini menyediakan interface drag-and-drop untuk upload file Excel dengan validasi format dan ukuran file secara real-time.



Gambar 3. 48 Interface Upload Data

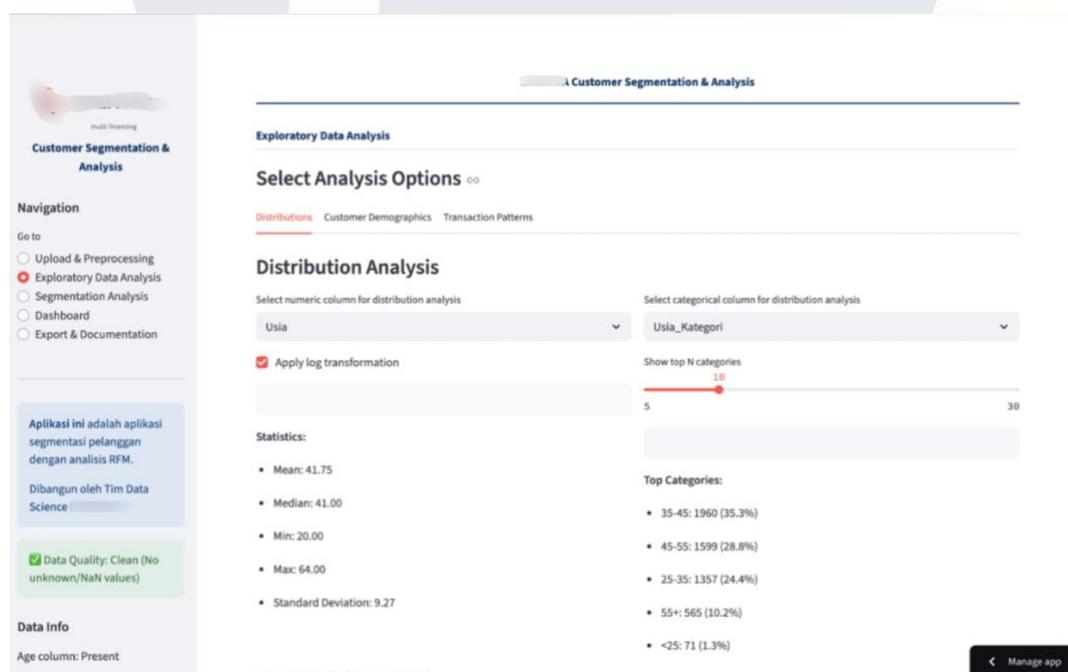
Sistem melakukan konversi otomatis untuk kolom tanggal seperti FIRST\_PPC\_DATE, LAST\_MPF\_DATE, dan BIRTH\_DATE ke format datetime yang konsisten. Proses preprocessing mencakup perhitungan usia pelanggan berdasarkan tahun referensi 2024, normalisasi tipe data untuk kolom numerik, dan penanganan missing values menggunakan metode median untuk data numerik dan modus untuk data kategorikal.

#### 2. Exploratory Data Analysis Module

Modul Exploratory Data Analysis dirancang untuk memberikan pemahaman awal terhadap karakteristik pelanggan sebelum dilakukan proses segmentasi. Modul ini terdiri atas tiga kategori analisis utama yang ditampilkan melalui interface tab interaktif, yaitu: Distribution Analysis, Customer Demographics, dan Transaction Patterns. Setiap

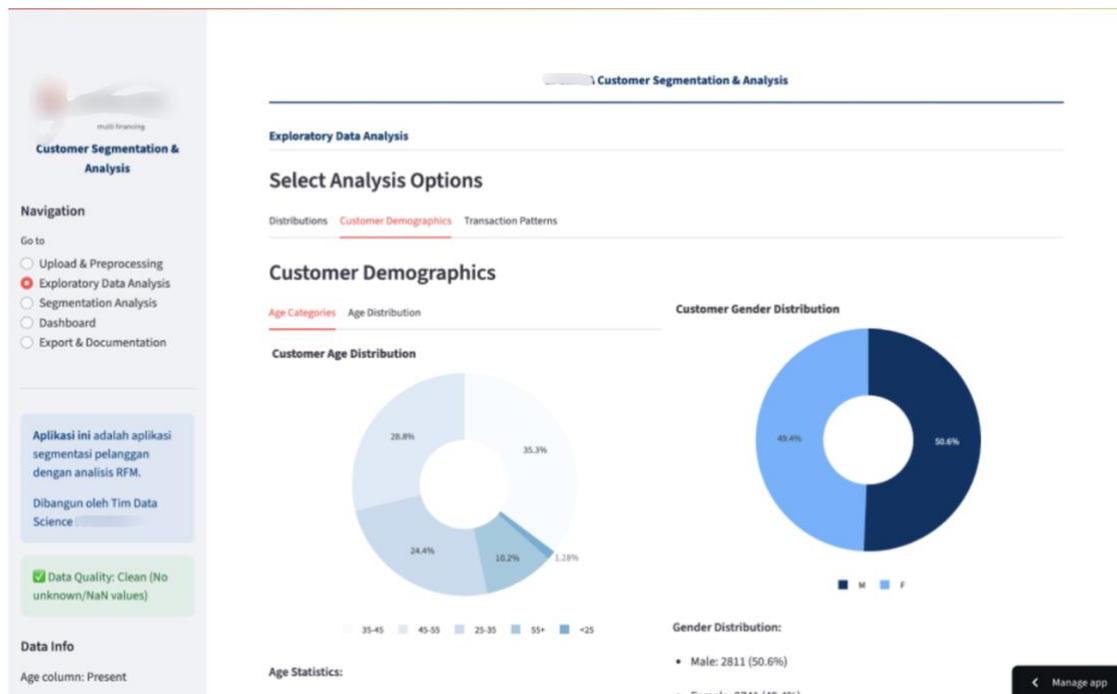
tab menyediakan visualisasi dinamis yang dapat disesuaikan berdasarkan variabel yang dipilih oleh pengguna.

Tab pertama, *Distribution Analysis*, menyajikan analisis sebaran data numerik maupun kategorikal. Pengguna dapat memilih kolom numerik seperti usia pelanggan untuk divisualisasikan dalam bentuk histogram, serta menerapkan transformasi logaritmik guna menyesuaikan distribusi data jika diperlukan. Untuk variabel kategorikal, sistem menyediakan filter top-N categories dengan kontrol slider guna mengatur jumlah kategori yang ditampilkan secara fleksibel. Informasi statistik seperti nilai rata-rata, median, nilai minimum, maksimum, dan simpangan baku juga disediakan secara otomatis untuk mendukung analisis lebih mendalam..



Gambar 3. 49 Interface Exploratory Data Analysis Section 1

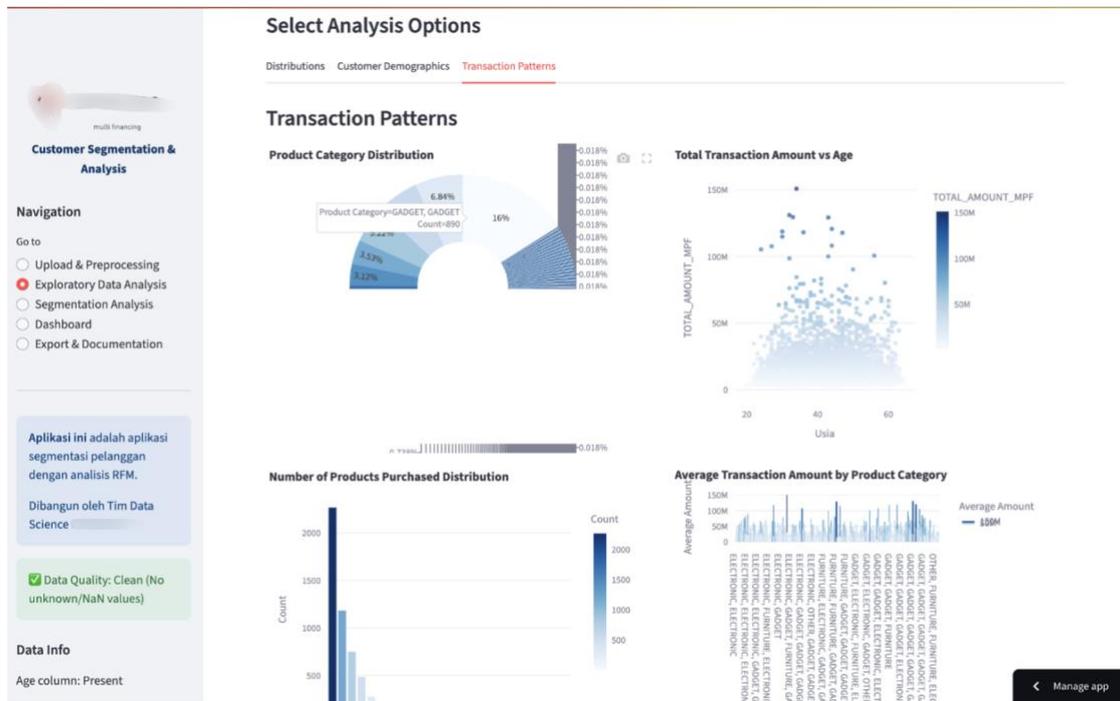
Tab kedua, *Customer Demographics*, menyuguhkan visualisasi yang fokus pada atribut demografis pelanggan, seperti kategori usia, distribusi usia, serta rasio gender. Data ditampilkan dalam bentuk *pie chart* dan *donut chart* untuk mempermudah identifikasi proporsi kelompok pelanggan tertentu. Informasi tambahan seperti jumlah pelanggan laki-laki dan perempuan serta persentasenya turut disajikan sebagai bagian dari ringkasan statistik.



Gambar 3. 50 Interface Exploratory Data Analysis Section 2

Tab ketiga, *Transaction Patterns*, mengkaji pola perilaku pembelian pelanggan berdasarkan kategori produk, jumlah produk yang dibeli, serta hubungan antara usia dengan total nilai transaksi. Visualisasi mencakup *sunburst chart*, *scatter plot*, dan *bar chart* untuk menggambarkan korelasi maupun distribusi frekuensi dengan lebih informatif. Modul ini juga dilengkapi dengan komponen filtering dan interaktivitas untuk eksplorasi yang lebih fleksibel sesuai kebutuhan pengguna.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

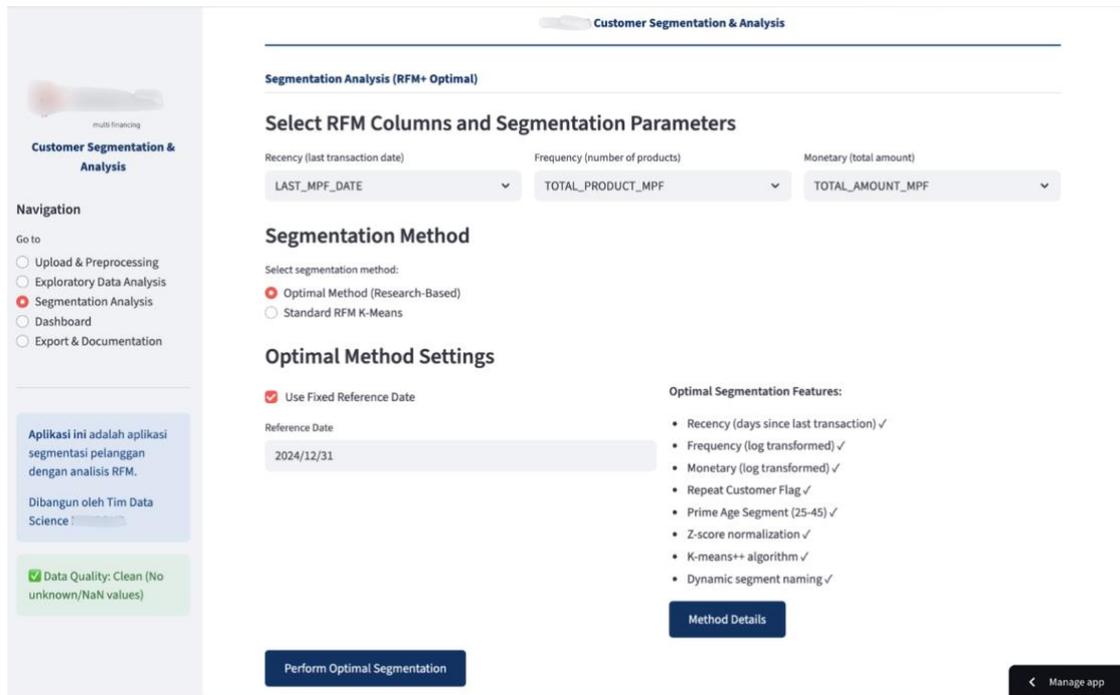


Gambar 3. 51 Dashboard EDA dengan Multiple Tabs dan Visualisasi Interaktif

Secara keseluruhan, modul EDA ini berfungsi sebagai fondasi analisis awal yang membantu tim CRM memahami karakteristik pelanggan secara deskriptif sebelum dilakukan segmentasi yang lebih lanjut menggunakan algoritma clustering.

### 3. Segmentation Analysis Module

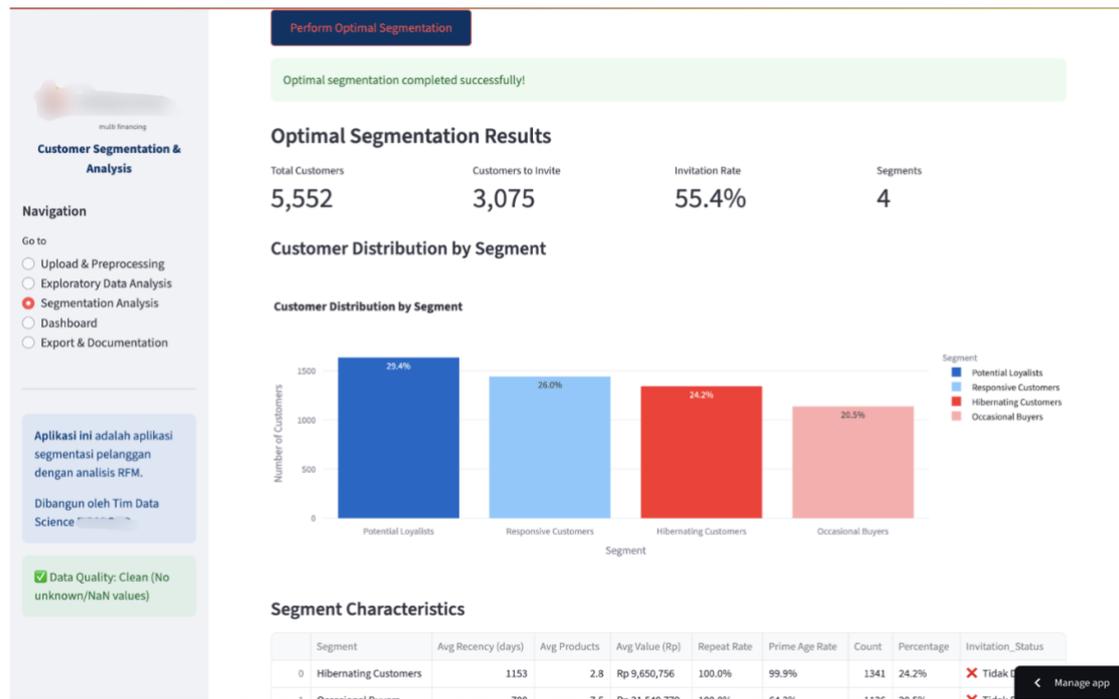
Modul segmentasi merupakan inti dari sistem analitik yang dikembangkan, yang berfungsi untuk mengelompokkan pelanggan berdasarkan perilaku transaksional mereka menggunakan metode RFM (Recency, Frequency, Monetary) yang telah dioptimalkan. Antarmuka modul ini dirancang agar intuitif dan interaktif, memungkinkan pengguna memilih parameter kolom yang relevan melalui dropdown selection yang telah terintegrasi dengan sistem deteksi otomatis terhadap tipe data tanggal dan numerik.



Gambar 3. 52 Interface Segmentation Analysis

Pengguna diberikan opsi untuk memilih dua pendekatan segmentasi, yaitu metode standar *RFM K-Means* dan metode *Optimal (Research-Based)* yang merupakan hasil pengembangan dengan pendekatan *feature engineering* lanjutan. Metode optimal ini mencakup transformasi logaritmik terhadap variabel *Frequency* dan *Monetary*, penambahan indikator pelanggan ulang (*Repeat Customer Flag*), klasifikasi usia produktif (25–45 tahun), serta normalisasi menggunakan metode *Z-Score* sebelum diterapkan algoritma *K-Means++*.

Hasil segmentasi ditampilkan secara ringkas dalam bentuk distribusi pelanggan per segmen, tingkat undangan (invitation rate), serta karakteristik setiap segmen seperti rata-rata *recency*, jumlah produk, nilai transaksi, dan tingkat pengulangan. Pada contoh ini, sistem berhasil mengidentifikasi empat kelompok pelanggan utama: *Potential Loyalists*, *Responsive Customers*, *Occasional Buyers*, dan *Hibernating Customers*.



Gambar 3. 53 Interface Hasil Segmentasi 1

Visualisasi hasil segmentasi juga disajikan dalam bentuk grafik tiga dimensi yang memetakan posisi setiap pelanggan dalam ruang fitur RFM. Visualisasi ini dilengkapi dengan sistem warna untuk membedakan tiap segmen, serta *radar chart* yang menunjukkan kontribusi relatif setiap variabel terhadap segmentasi yang terbentuk.





Gambar 3. 54 Interface Hasil Segmentasi 2

Sebagai pelengkap, sistem menyediakan ringkasan segmentasi dalam format deskriptif serta rekomendasi strategi pemasaran untuk masing-masing kelompok pelanggan. Fitur ini memudahkan tim bisnis dalam menentukan langkah lanjutan, seperti kampanye reaktivasi, pendekatan ulang produk, atau strategi *win-back* pelanggan bernilai tinggi. Pengguna juga dapat mengunduh hasil segmentasi dalam format terstruktur untuk digunakan dalam keperluan analisis lanjutan.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

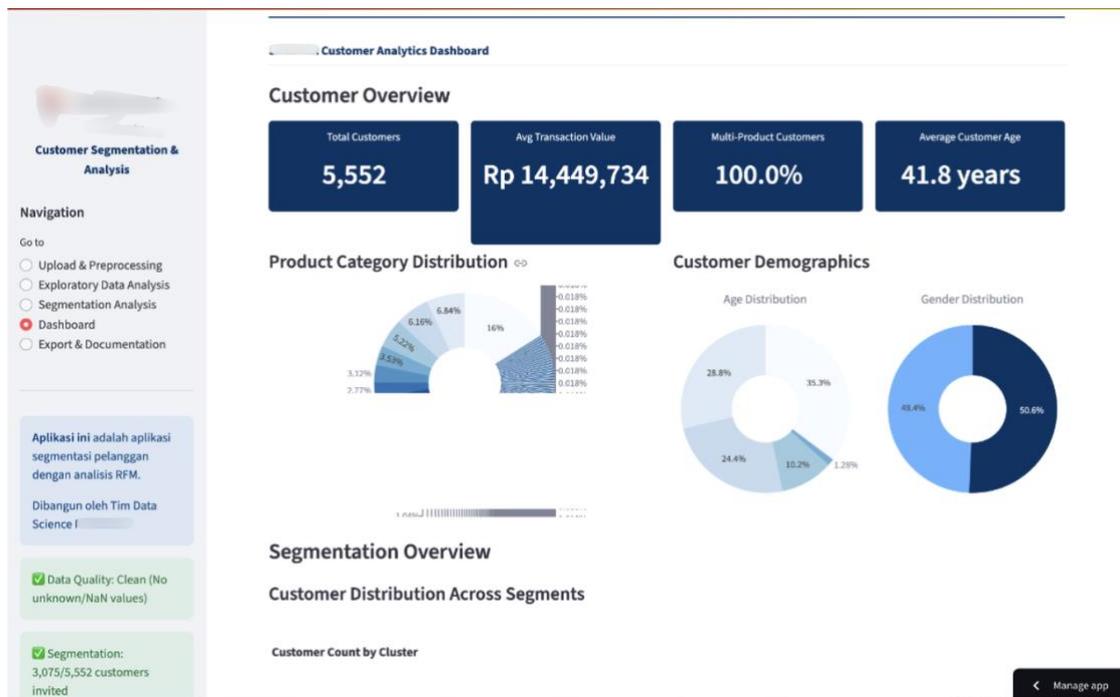


Gambar 3. 55 Interface Hasil Segmentasi 3

Secara keseluruhan, modul ini tidak hanya menyajikan hasil segmentasi yang dapat dipertanggungjawabkan secara metodologis, tetapi juga dilengkapi fitur interpretasi dan strategi yang aplikatif untuk mendukung pengambilan keputusan di level operasional maupun strategis.

#### 4. Interactive Dashboard Module

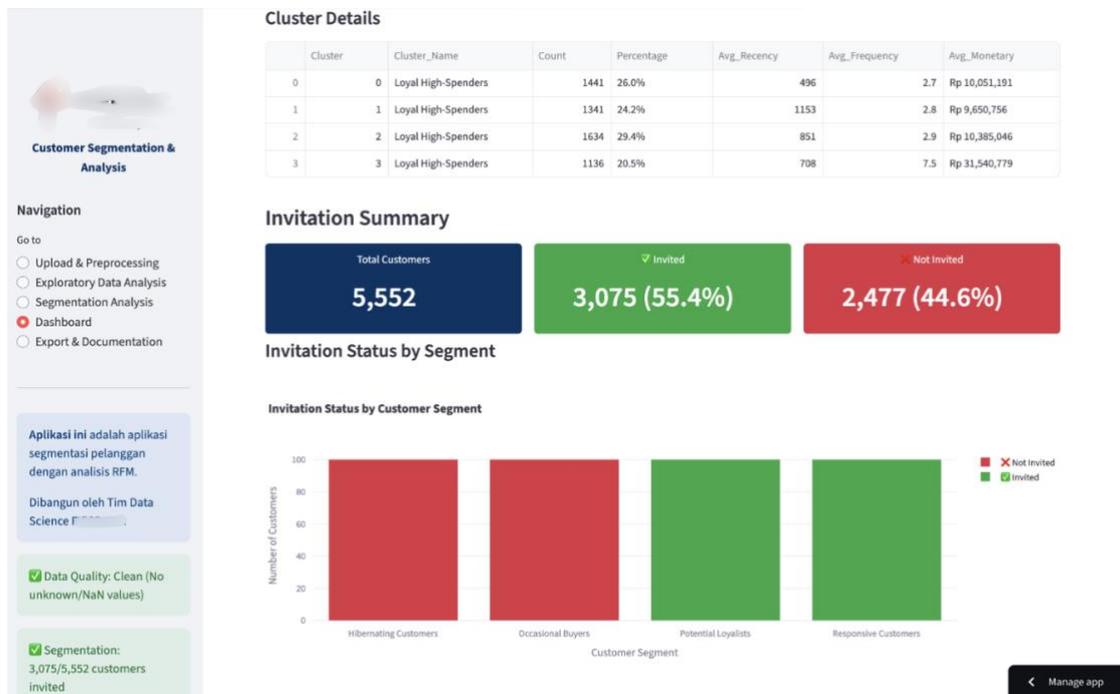
Modul *dashboard* dirancang untuk menyajikan ringkasan hasil segmentasi dalam format visual yang mudah dipahami oleh seluruh pemangku kepentingan, termasuk pihak non-teknis. Antarmuka dashboard menampilkan berbagai indikator kinerja utama (*Key Performance Indicators/KPI*) secara langsung dalam bentuk *cards*, yang meliputi jumlah total pelanggan yang dianalisis, rata-rata nilai transaksi, persentase pelanggan multiproduk, dan rata-rata usia pelanggan.



Gambar 3. 56 Interface Dashboard

Selain itu, dashboard menyediakan visualisasi komprehensif terkait distribusi pelanggan berdasarkan kategori produk dan demografi. Distribusi produk disajikan dalam bentuk sunburst chart yang menunjukkan proporsi relatif tiap kategori produk yang dimiliki oleh pelanggan. Sementara itu, distribusi usia dan jenis kelamin pelanggan divisualisasikan dalam bentuk donut chart yang memberikan gambaran cepat mengenai komposisi demografis pelanggan perusahaan.

Bagian selanjutnya dari dashboard menampilkan informasi segmentasi pelanggan dalam bentuk distribusi antar klaster yang terbentuk. Visualisasi ini ditampilkan dalam bentuk bar chart yang menggambarkan jumlah pelanggan per segmen, serta status undangan (invited vs. not invited) berdasarkan hasil segmentasi. Setiap segmen dapat dianalisis lebih lanjut berdasarkan karakteristik rata-rata seperti recency, frequency, dan monetary yang tercantum pada tabel statistic.

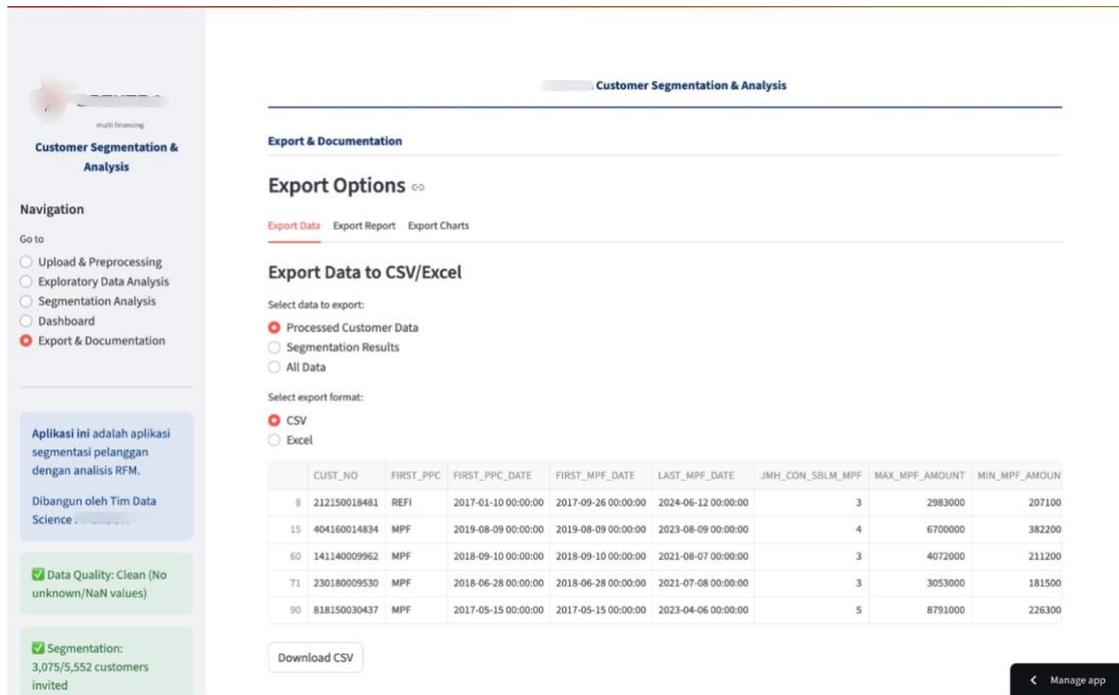


Gambar 3. 57 Dashboard Overview dengan KPI Cards dan Distribusi Segmen

Tampilan yang intuitif ini mendukung pengambilan keputusan secara cepat dan berbasis data (data-driven decision making), khususnya dalam perencanaan kampanye pemasaran, strategi retensi pelanggan, serta evaluasi efektivitas segmentasi. Melalui integrasi visualisasi yang kaya dan data statistik yang ringkas, dashboard berperan sebagai jembatan antara hasil analitik dan eksekusi strategi bisnis yang konkret..

## 5. Export & Documentation Module

Modul Export & Documentation disediakan untuk memfasilitasi kebutuhan operasional pengguna dalam mendistribusikan hasil analisis segmentasi secara fleksibel dan terdokumentasi. Modul ini memungkinkan pengguna untuk mengeksport data hasil analisis dalam berbagai format, baik Comma-Separated Values (CSV) maupun Excel Spreadsheet. Pengguna dapat memilih data yang akan diekspor, mencakup data pelanggan yang telah diproses, hasil segmentasi akhir, ataupun keseluruhan data.



Gambar 3. 58 Interface Export dengan Multiple Format Options dan Preview

Fitur ini dilengkapi dengan preview table yang menampilkan isi data sebelum diunduh, sehingga pengguna dapat memastikan keakuratan dan kelengkapan data yang dipilih. Selain ekspor data tabular, sistem juga menyediakan fitur automated report generation yang menghasilkan dokumen dalam format HTML report berisi executive summary, penjabaran metodologi, visualisasi hasil utama, serta rekomendasi bisnis yang disusun secara otomatis.

Tak hanya itu, visualisasi interaktif yang telah disusun dalam modul dashboard juga dapat diekspor dalam berbagai format gambar seperti PNG atau SVG, sehingga memudahkan tim pemasaran, manajemen, dan analis dalam menyusun presentasi dan laporan internal. Dengan demikian, modul ini berperan penting dalam mengintegrasikan hasil analisis data ke dalam proses kerja nyata perusahaan secara lebih praktis dan terdokumentasi

### **3.2.3 Pengembangan Sistem Otomatisasi Distribusi Email Data Customer di XYZ Finance**

#### **3.2.3.1 Latar Belakang dan Tujuan Pengembangan**

Proses distribusi data leads kepada cabang di XYZ Finance sebelumnya dilakukan secara manual, yang memakan waktu dan rawan kesalahan. Setiap pengiriman email harus disiapkan satu per satu, mulai dari pemisahan data, penyusunan isi, hingga validasi email penerima.

Untuk meningkatkan efisiensi dan akurasi, dikembangkan sistem otomatisasi pengiriman email massal[]. Sistem ini bertujuan untuk:

1. Mengotomatiskan proses distribusi data leads per area;
2. Menyediakan format email HTML yang konsisten dan informatif;
3. Mengidentifikasi cabang tanpa data leads untuk ditindaklanjuti;
4. Mempercepat proses follow-up oleh cabang.

Dengan sistem ini, proses distribusi menjadi lebih cepat, rapi, dan mudah dikendalikan.

#### **3.2.3.2 Desain dan Arsitektur Sistem**

Sistem otomatisasi distribusi email yang dikembangkan oleh penulis untuk PT XYZ Finance dirancang dengan pendekatan modular dan berorientasi pada efisiensi pengiriman informasi. Arsitektur sistem ini terdiri dari tiga komponen utama, yaitu: *pemrosesan data*, *pembangkitan konten email*, dan *distribusi melalui platform Outlook*.

Pertama, modul pemrosesan data bertugas membaca dan membersihkan data dari tiga sumber utama, yakni data leads pelanggan, referensi kategori follow-up, dan daftar email penerima berdasarkan area operasional. Proses ini mencakup validasi alamat email, klasifikasi cabang bermasalah, serta penyusunan tabel HTML yang nantinya akan disisipkan ke dalam isi email.

Selanjutnya, modul pembangkitan email memanfaatkan format HTML yang dirancang secara dinamis. Template ini menyesuaikan konten berdasarkan nama area,

periode distribusi, serta kondisi masing-masing cabang. Selain itu, sistem juga menyisipkan elemen visual seperti logo perusahaan dan petunjuk distribusi agar isi pesan menjadi lebih mudah dipahami oleh penerima.

Terakhir, sistem menghasilkan draft email secara otomatis melalui integrasi dengan Microsoft Outlook. Email akan terisi lengkap dengan daftar penerima, subjek sesuai standar perusahaan, isi tabel data, serta instruksi tindak lanjut yang spesifik. Dengan mekanisme ini, pengguna hanya perlu meninjau dan mengirimkan email tanpa perlu menyusun secara manual satu per satu.

Desain modular ini memastikan bahwa sistem dapat menangani skenario pengiriman massal ke seluruh area operasional PT XYZ Finance secara konsisten, cepat, dan minim kesalahan. Gambar 3.19 berikut menggambarkan arsitektur sistem yang digunakan.



Gambar 3. 59 Diagram Arsitektur Sistem Otomatisasi Email XYZ Finance

### 3.2.3.3 Implementasi Sistem Otomatisasi

#### 1. Input Data dan Validasi

Potongan kode pada Gambar 3.60 menggambarkan implementasi fungsi input dan validasi awal pada sistem distribusi email. Fungsi `export_data_by_area()` bertanggung jawab untuk mengekstraksi alamat email penerima berdasarkan area operasional tertentu, yang dibersihkan terlebih dahulu dari teks tambahan seperti “JUMLAH DATA AREA”. Proses pencocokan dilakukan melalui kolom “AREA” dalam dataframe, kemudian alamat email dikategorikan ke dalam daftar penerima utama (`to_list`) dan daftar tembusan (`cc_list`). Untuk menjamin akurasi, sistem menggunakan fungsi `is_valid_email()` yang memanfaatkan regular expression guna memastikan hanya email dengan format valid yang diproses. Selain itu, sistem mencatat setiap area yang tidak ditemukan dalam data sebagai indikasi adanya “cabang bermasalah”, dan hasil pengecekan tersebut disimpan dalam log untuk keperluan debugging. Dengan pendekatan ini, sistem tidak hanya menjamin keakuratan alamat email, tetapi juga menyediakan mekanisme pelacakan terhadap anomali data pada tahap awal proses distribusi.

```
# == Load Data Penerima Email dari Excel ==
df_email = pd.read_excel("03_PIC Data Leads - 2025.xlsx", sheet_name="PIC 2025")

# == Fungsi Validasi Email ==
def is_valid_email(email):
    return bool(re.match(r"^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,}$", email))

def export_data_by_area(area):
    # Bersihkan area data: JUMLAH DATA AREA
    cleaned_area = area.replace("JUMLAH DATA AREA ", "").strip()
    logging.info(f"DEBUG: Area setelah dibersihkan: {cleaned_area}")

    my_email = "richard.ohendarto@..."

    email_lists = [
        my_email,
        ";".join(df_email.loc[df_email["AREA"] == cleaned_area, "NPK BH"].dropna().astype(str).unique()),
        ";".join(df_email.loc[df_email["AREA"] == cleaned_area, "NPK PH"].dropna().astype(str).unique()),
        ";".join(df_email.loc[df_email["AREA"] == cleaned_area, "NPK MM"].dropna().astype(str).unique()),
        ";".join(df_email.loc[df_email["AREA"] == cleaned_area, "EMAIL"].dropna().astype(str).unique()),
    ]

    to_list = ";".join(filter(None, email_lists))
    to_list = ";".join([email for email in to_list.split(";") if is_valid_email(email)]) # Validasi

    cc_list = ";".join(filter(None, [
        ";".join(df_email.loc[df_email["AREA"] == cleaned_area, "NPK KAWIL"].dropna().astype(str).unique()),
        "heru.pamopase",
        "jonis",
        "ayu.afanuz",
        ";".join(df_email.loc[df_email["AREA"] == cleaned_area, "NPK KORWIL MARKETING"].dropna().astype(str).unique()),
        ";".join(df_email.loc[df_email["AREA"] == cleaned_area, "NPK KORWIL OPERATING"].dropna().astype(str).unique()),
        "VeriPutri.agusetya@...",
        "juniar.suehantose"
    ]))
    cc_list = ";".join([email for email in cc_list.split(";") if is_valid_email(email)]) # Validasi

    # Debugging
    logging.info(f"DEBUG: Area yang dicari di df_email setelah bersih: {cleaned_area}")
    matched_rows = df_email[df_email["AREA"] == cleaned_area]
    if matched_rows.empty:
        logging.warning(f"TIDAK DITEMUKAN area '{cleaned_area}' di df_email!")

    return {"TO": to_list, "CC": cc_list}

# == Load Data Excel ==
file_path = "CEK BAHAN DAN EMAIL BAHAN BH.xlsx"
```

Gambar 3. 60 Potongan Kode Fungsi Input

## 2. Pengolahan dan Pengelompokan Data

Gambar 3.61 menunjukkan proses pembersihan dan pengelompokan data leads berdasarkan area operasional. Tahapan awal diawali dengan menghapus baris yang tidak relevan, termasuk baris dengan nilai kosong pada kolom OFFICE\_CODE dan entri header tambahan yang muncul secara duplikat. Selanjutnya, kolom numerik seperti LEADS\_A, LEADS\_B, dan GRAND\_TOTAL dikonversi menjadi tipe integer untuk mempermudah proses analisis kuantitatif. Sistem kemudian menerapkan fungsi detect\_area() untuk mendeteksi dan menetapkan nilai area berdasarkan kemunculan kata kunci “JUMLAH DATA AREA” pada kolom OFFICE\_CODE maupun NAMA\_CABANG. Nilai area yang hilang diisi menggunakan metode forward fill agar tetap konsisten. Setelah proses tersebut, dilakukan pembersihan lanjutan untuk menghapus baris sisa yang masih mengandung kata kunci ringkasan seperti “JUMLAH DATA AREA” dan “Grand Total”. Terakhir, sistem juga mendeteksi daftar area yang tidak memiliki data leads, yang akan digunakan sebagai referensi dalam identifikasi “cabang bermasalah”. Pendekatan ini memastikan bahwa data yang masuk ke tahap distribusi telah tersaring secara akurat, relevan, dan siap diolah ke dalam format output.

```
df_raw = df_raw.iloc[:, [1, 2, 3, 4, 5]]
df_raw.columns = ["OFFICE_CODE", "NAMA_CABANG", "LEADS_A", "LEADS_B", "GRAND_TOTAL"]
df_raw = df_raw.dropna(subset=["OFFICE_CODE"])

# Hapus baris yang tidak diperlukan (header tambahan)
df_raw = df_raw[df_raw["OFFICE_CODE"] != "OFFICE_CODE"]

# Konversi angka
for col in ["LEADS_A", "LEADS_B", "GRAND_TOTAL"]:
    df_raw[col] = pd.to_numeric(df_raw[col], errors="coerce").fillna(0).astype(int)

def detect_area(row):
    if isinstance(row["OFFICE_CODE"], str) and "JUMLAH DATA AREA" in row["OFFICE_CODE"]:
        return row["OFFICE_CODE"]
    elif isinstance(row["NAMA_CABANG"], str) and "JUMLAH DATA AREA" in row["NAMA_CABANG"]:
        return row["NAMA_CABANG"]
    else:
        return None

df_raw["AREA"] = df_raw.apply(detect_area, axis=1)
df_raw["AREA"] = df_raw["AREA"].fillna(method="ffill") # Isi dengan nilai sebelumnya

# == Hapus Baris yang Tidak Dibutuhkan ==
df_cleaned = df_raw.dropna(subset=["AREA"]) # Hapus baris kosong setelah ffill
df_cleaned = df_cleaned[~df_cleaned["OFFICE_CODE"].astype(str).str.contains("JUMLAH DATA AREA", na=False)]
df_cleaned = df_cleaned[~df_cleaned["NAMA_CABANG"].astype(str).str.contains("Grand Total", na=False)]

# == Daftar Area yang Tidak Memiliki "LEADS_A" ==
area_without_amitra = {"JABAR 1", "JABAR 3", "JATA 1"} # Sesuaikan dengan data Anda
```

Gambar 3. 61 Potongan Kode Fungsi Pengolahan dan Pengelompokan Data

### 3. Pembuatan Template Email Dinamis

Gambar 3.62 menampilkan potongan kode fungsi `generate_html_table()`, yang berfungsi membentuk struktur template email dalam format HyperText Markup Language (HTML). Fungsi ini diawali dengan proses konversi nilai numerik pada kolom leads agar seluruh data tampil dalam format angka yang seragam. Nilai nol digantikan dengan simbol strip (“-”) guna memperjelas interpretasi visual oleh pembaca. Selanjutnya, fungsi menghapus baris yang tidak relevan—khususnya baris dengan nilai `OFFICE_CODE` berupa string literal “OFFICE\_CODE”—guna menjaga kebersihan data dalam tampilan akhir.

Struktur HTML yang dihasilkan mencakup elemen `table` dengan desain visual yang telah ditentukan, seperti background color kuning (#FFF700) pada judul dan ungu (#A8D08D) pada header kolom. Template ini juga menyertakan baris judul yang mencantumkan area distribusi secara dinamis melalui parameter `area.upper()`. Untuk keperluan identifikasi, disiapkan pula list kosong bernama `red_branch_names` yang nantinya akan digunakan untuk menandai cabang-cabang tertentu dengan pewarnaan khusus dalam baris tabel. Secara keseluruhan, pendekatan ini memastikan bahwa email yang dikirim tidak hanya informatif, tetapi juga profesional dan sesuai dengan standar komunikasi visual perusahaan..

```
def generate_html_table(df, area):
    # Pastikan kolom angka dalam format numerik
    df["LEADS_N"] = pd.to_numeric(df["LEADS_N"], errors="coerce").fillna(0).astype(int)
    df["LEADS_A"] = pd.to_numeric(df["LEADS_A"], errors="coerce").fillna(0).astype(int)
    df["GRAND_TOTAL"] = df["LEADS_N"] + df["LEADS_A"]

    # Pastikan nilai 0 diubah menjadi "-"
    df["LEADS_N"] = df["LEADS_N"].apply(lambda x: "-" if x == 0 else f"{x}")
    df["LEADS_A"] = df["LEADS_A"].apply(lambda x: "-" if x == 0 else f"{x}")
    df["GRAND_TOTAL"] = df["GRAND_TOTAL"].apply(lambda x: "-" if x == 0 else f"{x}")

    # Hapus hanya row dengan OFFICE_CODE = "OFFICE_CODE"
    df = df[df["OFFICE_CODE"] != "OFFICE_CODE"]

    # List untuk menyimpan Nama Cabang yang akan menjadi baris merah
    red_branch_names = []

    # Struktur tabel HTML
    html = """
<table border="1" cellpadding="5" cellspacing="0"
    style="border-collapse: collapse; width: 100%; font-family: Arial, sans-serif; text-align: center;">
  <thead>
    <tr style="background-color: #FFF700; font-weight: bold;">
      <th colspan="5">JUMLAH DATA AREA {area.upper()}</th>
    </tr>
    <tr style="background-color: #A8D08D; font-weight: bold;">
      <th>Office Code</th>
      <th>Nama Cabang</th>
      <th>Leads N</th>
      <th>Leads A</th>
      <th>Grand Total</th>
    </tr>
  </thead>
  <tbody>
    """
```

Gambar 3. 62 Potongan Kode Pembuatan Template Email

#### 4. Pengiriman Draft Email

Gambar 3.63 memperlihatkan potongan kode fungsi `send_outlook_email()` yang berfungsi untuk membuat draft email secara otomatis menggunakan pustaka `win32com.client`, yang menghubungkan Python dengan Microsoft Outlook. Proses ini memungkinkan pembentukan draft email dengan atribut lengkap seperti `to`, `cc`, `subject`, dan isi body dalam format HTML secara terprogram dan efisien.

Fungsi ini kemudian diikuti oleh perulangan `for` yang menelusuri setiap area operasional berdasarkan hasil `grouping` kolom "AREA". Untuk setiap area, sistem melakukan pemanggilan fungsi `export_data_by_area()` guna mengambil daftar email penerima yang relevan. Selanjutnya, fungsi `generate_html_table()` digunakan untuk menyusun isi email dalam bentuk tabel HTML serta menghasilkan daftar cabang bermasalah yang akan ditandai secara visual dengan warna khusus.

Jika terdapat cabang yang tidak memiliki data leads atau tidak memberikan umpan balik laporan selama periode waktu tertentu (misalnya >5 bulan), maka informasi tersebut ditampilkan dalam daftar bullet berwarna kuning dengan highlight khusus. Sebaliknya, jika semua cabang memiliki data, sistem menampilkan pesan dengan warna hijau yang menyatakan bahwa tidak terdapat cabang bermasalah. Setiap email hasil dari loop akan disimpan sebagai draft di Outlook, memungkinkan pengguna melakukan pengecekan akhir sebelum pengiriman resmi dilakukan. Pendekatan ini meningkatkan efisiensi komunikasi dan memastikan distribusi data yang akurat ke seluruh area operasional..

U M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

```

# Fungsi `create_draft()` harus dideklarasikan DI LUAR loop `for`

def send_outlook_email(to, cc, subject, body):
    outlook = win32com.client.Dispatch("Outlook.Application")
    mail = outlook.CreateItem(0) # Buat email baru
    mail.To = to
    mail.CC = cc
    mail.Subject = subject
    mail.HTMLBody = body # Format email dalam HTML

    # Jika ingin menambahkan lampiran:
    # mail.Attachments.Add("C:\\path\\to\\file.xlsx")

    mail.Save() # Simpan sebagai draft di Outlook
    logging.info(f"Draft email untuk {subject} berhasil dibuat di Outlook.")

for area, df_area in df_cleaned.groupby("AREA"):
    logging.info(f"DEBUG: Memproses email untuk area: {area}")

    # Ambil daftar email TO dan CC berdasarkan area
    email_data = export_data_by_area(area)

    # Generate HTML tabel dan daftar cabang merah untuk area saat ini
    html_table, red_branch_names = generate_html_table(df_area, area)

    # Buat HTML daftar cabang merah untuk dimasukkan ke email
    if red_branch_names:
        red_branch_html = """
        <p style="font-size: 16px; line-height: 1.6; text-align:center;">
        Berikut Cabang-Cabang yang tidak didistribusikan data Leads NMC karena
        <span style="font-weight: bold; background-color: yellow; padding: 2px 4px;">5 bulan</span>
        berturut-turut atau lebih tidak memberikan report feedback:
        </p>
        <ul style="text-align:center; list-style: none; padding: 0;">
        """
        for branch in red_branch_names:
            red_branch_html += f"<li><span style='color:#000080; font-weight:bold; font-size:15px;'>{branch}</span></li>"
            red_branch_html += "</ul>"
    else:
        red_branch_html = "<p style='color:green; font-weight:bold; font-size:25px; text-align:center;'>Semua cabang memiliki data</p>"

```

Gambar 3. 63 Potongan Kode Pengiriman Draft Email

### 3.2.3.4 Hasil dan Evaluasi

#### 1. Contoh Hasil Draft Email

Sistem otomatisasi yang dikembangkan berhasil menghasilkan draft email dalam format HTML yang telah disesuaikan dengan kebutuhan komunikasi internal perusahaan. Email tersebut mencakup informasi ketersediaan data Good Customer untuk setiap area, dilengkapi dengan tabel distribusi berdasarkan nama cabang, jumlah leads, dan status distribusi. Selain itu, pesan juga memuat informasi penting seperti batas waktu follow-up serta penekanan pada prioritas kerja tim cabang. Gambar 3.64 menunjukkan contoh tampilan draft email hasil keluaran sistem untuk Area

SULUT.





Gambar 3. 64 Contoh Tampilan Draft Email untuk Salah Satu Area

## 2. Efektivitas dan Efisiensi Dibanding Distribusi Manual Sebelumnya

Sebelum penerapan sistem ini, distribusi data dilakukan secara manual dengan menyusun dan mengirimkan email satu per satu oleh tim CRM. Proses tersebut memakan waktu dan rawan kesalahan, terutama dalam format tabel serta alokasi data per area. Setelah diotomatisasi, sistem mampu secara mandiri membangkitkan draft email untuk seluruh area operasional PT XYZ Finance, lengkap dengan personalisasi konten berdasarkan wilayah. Hal ini menghasilkan penghematan waktu yang signifikan dan meningkatkan konsistensi komunikasi lintas cabang, sehingga efisiensi distribusi data meningkat secara substansial.

### 3.2.4 Pengembangan Aplikasi Validasi Data KK

#### 3.2.4.1 Latar Belakang dan Tujuan Pengembangan

Dalam proses bisnis yang melibatkan verifikasi identitas pelanggan, validasi data kependudukan seperti Kartu Keluarga (KK), Nomor Induk Kependudukan (NIK), dan elemen demografis lainnya menjadi aspek krusial untuk memastikan akurasi informasi.

PT XYZ Finance sebagai institusi yang menangani berbagai kebutuhan pembiayaan pelanggan, sering kali menghadapi tantangan terkait keabsahan dan konsistensi data yang dikirimkan oleh calon konsumen maupun mitra kerja.

Berdasarkan kebutuhan tersebut, dikembangkan sebuah aplikasi validasi data KK berbasis web yang dibangun menggunakan framework Streamlit. Aplikasi ini ditujukan untuk mempermudah proses validasi secara otomatis terhadap elemen-elemen penting seperti nomor KK, NIK, nama lengkap, tempat lahir, dan tanggal lahir. Selain itu, aplikasi juga dilengkapi dengan fitur pencocokan fuzzy (fuzzy matching) untuk mengidentifikasi kemungkinan perbedaan minor pada penulisan nama yang umum terjadi akibat kesalahan input[40] [41].

Tujuan dari pengembangan aplikasi ini adalah untuk menyediakan alat bantu validasi data yang efisien, cepat, dan akurat. Sistem ini juga memungkinkan pengguna untuk melacak progres validasi secara real-time melalui tampilan dashboard interaktif, serta melakukan ekspor hasil untuk keperluan pelaporan atau audit internal. Dengan demikian, aplikasi ini tidak hanya meningkatkan keandalan proses verifikasi data, tetapi juga mendukung pengambilan keputusan yang lebih tepat dan berbasis data.

#### **3.2.4.2 Desain dan Arsitektur Sistem**

Aplikasi validasi data KK dikembangkan menggunakan pendekatan modular dan berbasis web, dengan memanfaatkan framework **Streamlit** untuk membangun antarmuka pengguna yang intuitif dan responsif. Arsitektur sistem dirancang agar mampu memproses data dalam jumlah besar secara efisien, sambil tetap mempertahankan kecepatan dan akurasi dalam proses validasi.

Struktur utama sistem dibagi menjadi tiga komponen inti, yaitu:

##### **1. Modul Input dan Preprocessing**

Modul ini menerima file input dari pengguna dalam format Excel (.xlsx) yang berisi data KK, NIK, nama, tempat lahir, dan tanggal lahir. Selanjutnya, sistem melakukan pembersihan data, seperti menghapus entri kosong,

menyamakan format penulisan, serta menyiapkan data untuk proses pencocokan.

## 2. Modul Validasi dan Fuzzy Matching

Proses validasi dilakukan dengan mengecek kesesuaian struktur data seperti panjang digit NIK atau KK, serta keunikan kombinasi data demografis. Selain itu, algoritma fuzzy matching diterapkan untuk membandingkan elemen nama menggunakan tingkat kemiripan tertentu, sehingga mampu mengidentifikasi kesalahan ketik atau variasi penulisan nama.

## 3. Modul Visualisasi dan Monitoring Progres

Hasil validasi ditampilkan dalam bentuk tabel interaktif yang menyoroti baris-baris bermasalah. Selain itu, sistem menyediakan tampilan progres berbasis persentase dan grafik batang untuk menunjukkan distribusi validasi berhasil/gagal. Pengguna juga dapat mengunduh hasil validasi dalam format Excel sebagai dokumentasi.

Secara keseluruhan, arsitektur ini memungkinkan pengguna untuk melakukan validasi data secara end-to-end dalam satu aplikasi, tanpa memerlukan instalasi tambahan atau perangkat khusus. Sistem dapat diakses melalui browser dan mendukung skenario operasional di lingkungan pemerintahan, lembaga keuangan, maupun sektor swasta yang membutuhkan proses verifikasi data kependudukan.

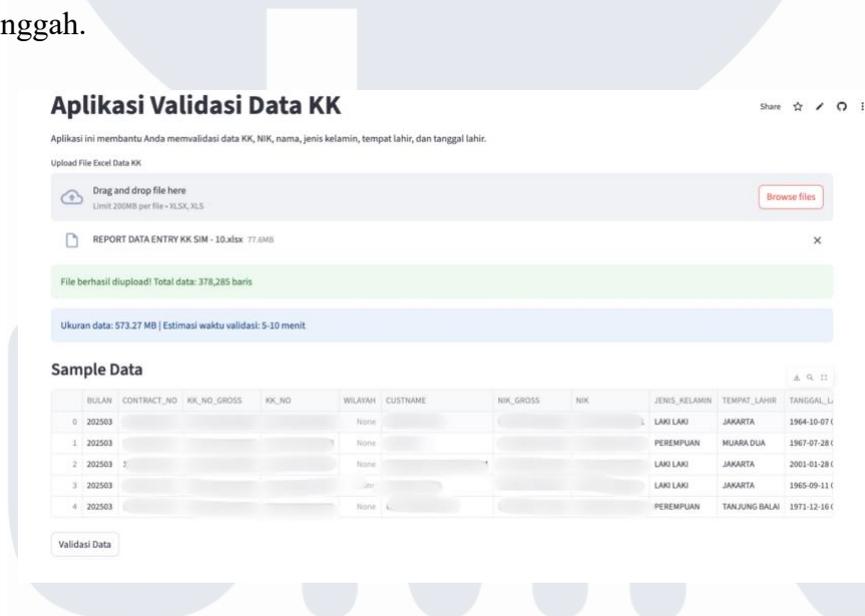
### 3.2.4.3 Implementasi Fitur Utama

Aplikasi validasi data KK pada PT XYZ Finance diimplementasikan dengan sejumlah fitur utama yang dirancang untuk memudahkan proses verifikasi data kependudukan secara sistematis dan interaktif. Seluruh fitur ini dibangun di atas framework *Streamlit*, dengan pemrosesan data menggunakan pustaka Python seperti *pandas* dan *fuzzywuzzy*.

## 1. Upload dan Preprocessing Module

Modul Upload dan Preprocessing berfungsi sebagai tahap awal dalam sistem validasi data kependudukan. Melalui antarmuka yang telah disediakan, pengguna dapat mengunggah file Excel (.xlsx) yang berisi informasi seperti nomor KK, NIK, nama lengkap, tempat lahir, jenis kelamin, dan tanggal lahir. Setelah proses unggah selesai, sistem akan memberikan notifikasi jumlah baris data yang berhasil dimuat, ukuran file, serta estimasi waktu proses validasi.

Secara otomatis, sistem akan menjalankan tahapan preprocessing yang mencakup penghapusan karakter tidak valid, penyamaan format tanggal, serta penghilangan data duplikat. Proses ini bertujuan untuk memastikan data dalam kondisi bersih dan terstruktur sebelum dilanjutkan ke tahap validasi lanjutan. Hasil preprocessing ditampilkan secara langsung dalam bentuk sample data agar pengguna dapat melakukan pengecekan awal terhadap kualitas data yang telah diunggah.



Gambar 3. 65 Interface Upload Data

## 2. Validation & Fuzzy Matching Module

Setelah proses preprocessing selesai, sistem akan melanjutkan ke tahap validasi data untuk memastikan integritas dan konsistensi setiap entri. Validasi dilakukan terhadap beberapa elemen penting, antara lain panjang karakter NIK dan KK,

keberadaan nilai kosong pada kolom-kolom wajib, serta kesesuaian format penulisan nama dan tanggal.

Untuk mendeteksi perbedaan minor dalam penulisan nama atau tempat lahir, sistem mengimplementasikan metode fuzzy matching yang mengukur kemiripan string menggunakan skor kesamaan huruf. Baris data yang memiliki skor kemiripan di bawah ambang batas tertentu akan secara otomatis ditandai sebagai potensi kesalahan, dan ditampilkan dalam bentuk interaktif kepada pengguna. Proses validasi ini juga dilengkapi indikator progres dan estimasi waktu penyelesaian, agar pengguna dapat memantau kinerja sistem secara real-time.

Tampilan antarmuka saat proses validasi berlangsung ditunjukkan pada Gambar 3.66.

The screenshot displays a data validation interface. At the top, a green status bar indicates 'File berhasil diupload! Total data: 378,285 baris'. Below it, a blue bar shows 'Ukuran data: 573.27 MB | Estimasi waktu validasi: 5-10 menit'. The main section is titled 'Sample Data' and contains a table with the following columns: BULAN, CONTRACT\_NO, KK\_NO\_GROSS, KK\_NO, WILAYAH, CUSTNAME, NIK\_GROSS, NIK, JENIS\_KELAMIN, TEMPAT\_LAHIR, and TANGGAL\_L. The table lists five rows of sample data. Below the table is a 'Validasi Data' button. A progress bar is shown with the text 'Memvalidasi tempat lahir...'. Below the progress bar, it says 'Gender validation: 0.2s' and 'Sedang memvalidasi data...'. At the bottom, a blue bar states 'Menggunakan algoritma optimized untuk data besar...'. A 'Manage app' button is visible in the bottom right corner.

	BULAN	CONTRACT_NO	KK_NO_GROSS	KK_NO	WILAYAH	CUSTNAME	NIK_GROSS	NIK	JENIS_KELAMIN	TEMPAT_LAHIR	TANGGAL_L
0	202503				None				LAKI LAKI	JAKARTA	1964-10-07
1	202503								PEREMPUAN	MUARA DUA	1967-07-28
2	202503				None				LAKI LAKI	JAKARTA	2001-01-28
3	202503				None	A			LAKI LAKI	JAKARTA	1965-09-11
4	202503				None				PEREMPUAN	TANJUNG BALAI	1971-12-16

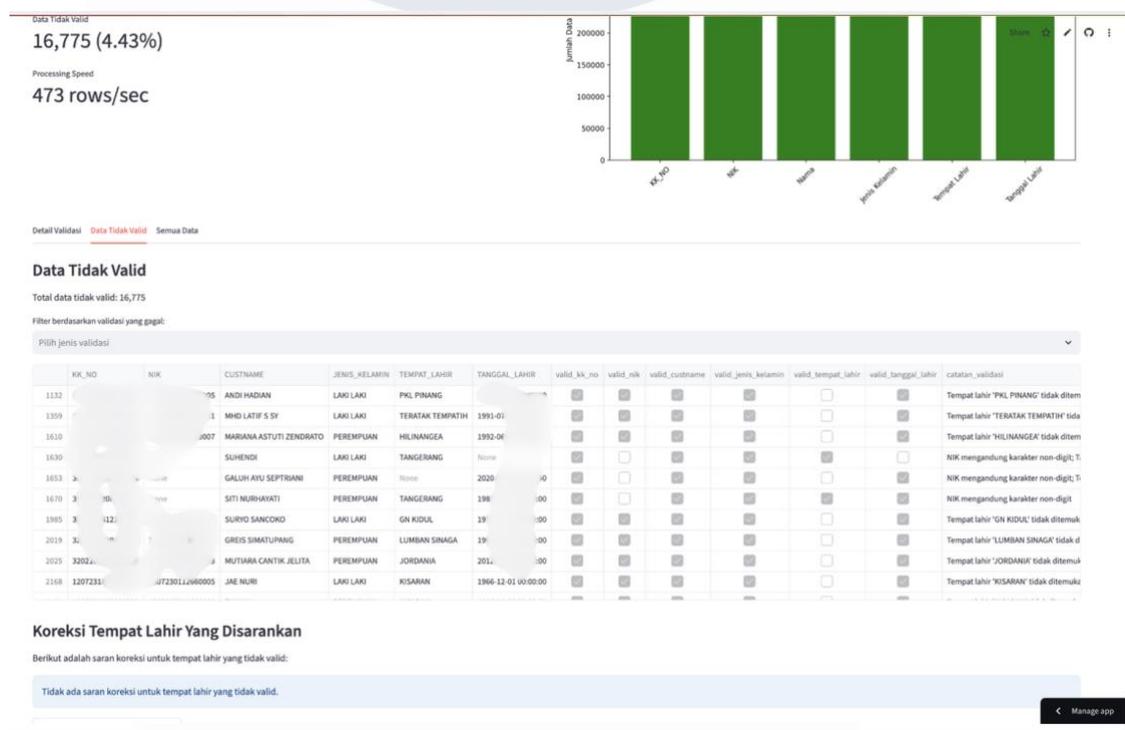
Gambar 3. 66 Interface Saat Data Divalidasi

### 3. Progress Tracker & Visual Feedback Module

Setelah proses validasi selesai dijalankan, sistem menyajikan hasil analisis dalam bentuk visual feedback yang informatif dan interaktif. Informasi utama yang ditampilkan meliputi jumlah total entri tidak valid, kecepatan pemrosesan data, serta persentase data bermasalah dibandingkan keseluruhan dataset. Representasi grafis berupa bar chart ditampilkan untuk memperlihatkan sebaran kesalahan pada masing-masing elemen data seperti KK, NIK, nama, jenis kelamin, tempat lahir, dan tanggal lahir.

Selain grafik, tabel interaktif juga ditampilkan untuk memperinci entri-entri yang tidak lolos validasi. Tabel ini menunjukkan informasi detail untuk setiap baris yang gagal, termasuk penyebab tidak valid, dan indikator validasi per kolom dalam bentuk checkbox. Sistem juga menyediakan fitur penyaringan berdasarkan jenis validasi tertentu, serta saran perbaikan otomatis seperti koreksi ejaan tempat lahir apabila memungkinkan.

Tampilan antarmuka visual dari hasil validasi dan tabel kesalahan ditunjukkan pada Gambar 3.67.



Gambar 3. 67 Interface Visual Feedback



#### **3.2.4.4 Hasil dan Evaluasi**

Hasil implementasi aplikasi validasi data KK pada PT XYZ Finance menunjukkan peningkatan signifikan dalam akurasi dan efisiensi proses verifikasi data kependudukan. Sebelumnya, validasi data dilakukan secara manual menggunakan Excel, yang memerlukan waktu cukup lama dan rentan terhadap human error, khususnya dalam mendeteksi kesalahan penulisan atau duplikasi nama.

Dengan menggunakan aplikasi ini, proses validasi terhadap ratusan hingga ribuan baris data dapat diselesaikan dalam hitungan detik. Fitur fuzzy matching berhasil mengidentifikasi entri dengan ejaan nama yang mirip tetapi tidak identik, yang sebelumnya sering luput dari pemeriksaan manual. Selain itu, indikator progres dan visualisasi hasil validasi memberikan umpan balik langsung kepada pengguna terkait proporsi data yang valid dan perlu diperbaiki.

Secara keseluruhan, aplikasi ini berkontribusi terhadap peningkatan kualitas data customer, mendukung kelancaran proses operasional, dan memperkuat aspek kontrol internal dalam pengelolaan data kependudukan yang digunakan oleh PT XYZ Finance.

#### **3.2.5 Pengembangan Aplikasi Pipeline Data Merge untuk Telesales PT XYZ**

##### **Finance**

##### **3.2.5.1 Latar Belakang dan Tujuan Pengembangan**

Dalam aktivitas harian tim telesales PT XYZ Finance, pengolahan data leads pelanggan menjadi bagian krusial yang memengaruhi efisiensi operasional. Setiap hari, tim menerima sejumlah file Excel dari berbagai sumber yang berisi daftar calon pelanggan untuk dihubungi. Tantangan muncul ketika file-file tersebut memiliki format yang berbeda, dilindungi oleh password, dan berpotensi mengandung data duplikat. Proses penggabungan secara manual tidak hanya memakan waktu, tetapi juga rawan kesalahan yang dapat berdampak pada kualitas distribusi data dan efektivitas aktivitas calling.

Berdasarkan permasalahan tersebut, dikembangkanlah aplikasi Pipeline Data Merge sebagai solusi otomatisasi proses penggabungan data leads pelanggan. Aplikasi ini dirancang untuk mendekripsi file Excel terproteksi, menormalkan struktur data, menghilangkan duplikasi, serta menghasilkan satu file output yang siap digunakan oleh tim telesales tanpa perlu keterlibatan teknis yang kompleks.

Tujuan utama dari pengembangan aplikasi ini adalah untuk meningkatkan efisiensi waktu kerja, mengurangi risiko *human error*, serta memastikan bahwa proses distribusi data kepada telesales berjalan secara lebih terstandarisasi, cepat, dan aman.

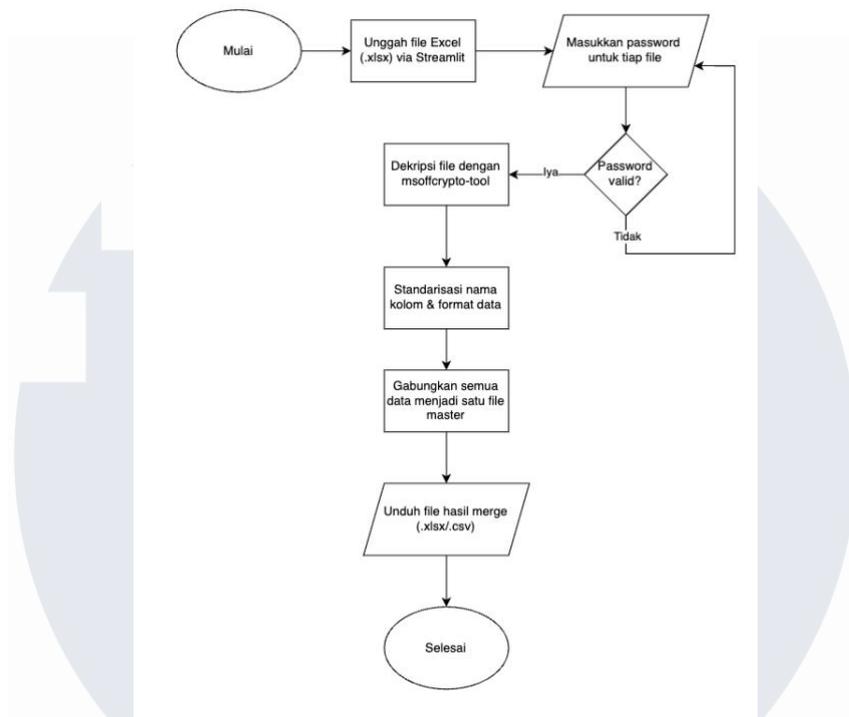
### 3.2.5.2 Desain dan Arsitektur Sistem

Aplikasi *Pipeline Data Merge* dirancang dengan pendekatan modular yang berfokus pada efisiensi pengolahan file Excel yang diproteksi password dan penggabungan data customer dari berbagai sumber. Struktur sistem disusun agar mudah digunakan oleh pengguna non-teknis namun tetap memenuhi standar keamanan dan konsistensi data.

Secara umum, arsitektur sistem ini terdiri dari tiga komponen utama, yaitu: unggah dan dekripsi file, proses standarisasi dan penggabungan data, serta hasil akhir berupa file yang dapat diunduh.

Pengguna dapat mengunggah beberapa file Excel melalui antarmuka Streamlit. Setiap file yang dilindungi kata sandi akan didekripsi langsung di dalam memori (tanpa disimpan secara lokal) menggunakan pustaka `msoffcrypto-tool`. Selanjutnya, sistem akan melakukan standarisasi nama kolom dan tipe data untuk menyelaraskan struktur file sebelum proses deduplikasi dan penggabungan dimulai.

Hasil akhir berupa satu file master yang sudah bersih dari duplikasi dan siap diunduh langsung melalui antarmuka aplikasi.



Gambar 3. 69 Arsitektur Sistem Aplikasi Pipeline Data Merge

### 3.2.5.3 Implementasi Sistem

Implementasi sistem Pipeline Data Merge dilakukan dengan membangun aplikasi berbasis Streamlit yang terhubung langsung dengan pipeline pemrosesan data secara otomatis. Tahapan implementasi terdiri dari pemrosesan file Excel yang diproteksi password, penyatuan data dari berbagai sumber, hingga standarisasi dan penyaringan berdasarkan standar telesales.

Pertama, pengguna mengunggah satu atau beberapa file Excel yang telah diproteksi dengan sandi. Sistem secara otomatis mendekripsi setiap file di memori menggunakan pustaka msosffcrypto-tool, tanpa menyimpan file secara permanen di perangkat atau server. Setelah file berhasil didekripsi, sistem membaca isi file menggunakan openpyxl dan memprosesnya lebih lanjut menggunakan pustaka pandas. Potongan kode yang mengatur mekanisme unggah *file* dan pengelolaan *password* ditunjukkan pada Gambar 3.70.

```

# Container untuk upload file
upload_container = st.container()
with upload_container:
    st.subheader("Upload File Excel")

# Menampilkan password yang digunakan (bisa diaktifkan/nonaktifkan)
show_password = st.checkbox("Tampilkan password yang digunakan")
if show_password:
    st.info(f"Password yang digunakan: {password}")

uploaded_files = st.file_uploader("Pilih satu atau beberapa file Excel", type=["xlsx"], accept_multiple_files=True)

# Jika ada file yang diupload
if uploaded_files:
    st.success(f"{len(uploaded_files)} file telah diupload")

# Option untuk menggunakan sheet yang sama untuk semua file
use_same_sheet = st.checkbox("Gunakan jenis sheet yang sama untuk semua file", value=True)

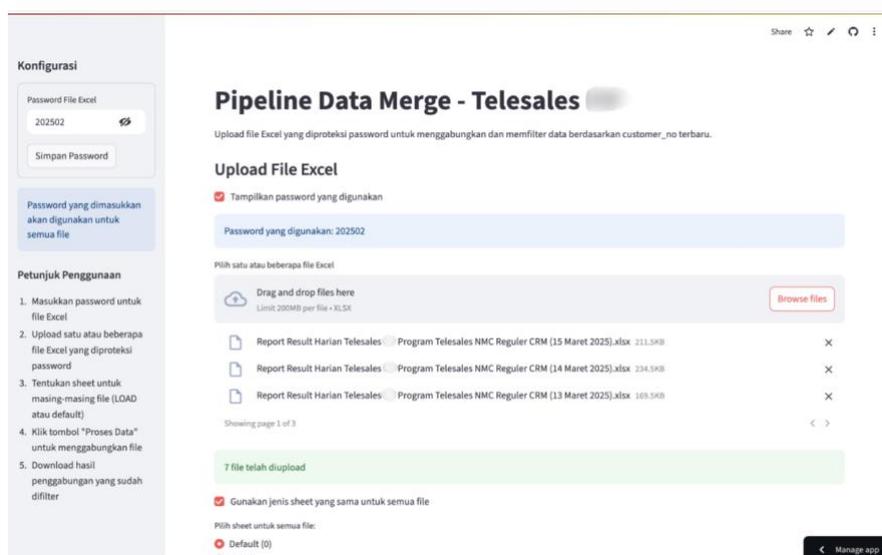
if use_same_sheet:
    global_sheet_choice = st.radio(
        "Pilih sheet untuk semua file:",
        ["Default (0)", "LOAD"],
        key="global_sheet_choice"
    )
    sheet_options = {file.name: "LOAD" if global_sheet_choice == "LOAD" else 0 for file in uploaded_files}

# Tampilkan list file dengan sheet yang dipilih
st.subheader("File yang akan diproses:")
for file in uploaded_files:
    st.text(f"{file.name} - Sheet: {global_sheet_choice}")
else:
    # Opsi untuk sheet yang akan dibaca untuk masing-masing file
    sheet_options = {}
    st.subheader("Pilih sheet untuk masing-masing file:")
    for file in uploaded_files:
        sheet_choice = st.radio(
            f"Sheet untuk {file.name}:",
            ["Default (0)", "LOAD"]
        )

```

Gambar 3. 70 Potongan Kode Upload Data dan Password

Selanjutnya, antarmuka pengguna untuk proses unggah dan pengaturan *file* ditampilkan pada Gambar 3.71. Pengguna diminta untuk memasukkan *password*, memilih opsi *sheet* yang sama atau berbeda untuk setiap file, serta melihat konfirmasi jumlah file yang berhasil diunggah.



Gambar 3. 71 Interface Upload Data dan Password

Tahapan berikutnya adalah proses pembacaan dan penggabungan data. Proses ini mencakup iterasi terhadap setiap file, pembacaan isi file menggunakan metode dekripsi, serta pencatatan hasil log proses. Potongan kode untuk tahapan ini ditunjukkan pada Gambar 3.72, sementara antarmuka pengguna yang sedang menjalankan proses ditampilkan pada Gambar 3.73.

```
# Proses data ketika tombol diklik
if uploaded_files:
    if st.button("Proses Data", key="process_button", type="primary"):
        with st.spinner("Sedang memproses data..."):
            # Inisialisasi untuk menyimpan dataframe dan log
            data_frames = []
            log_messages = []

            # Progress bar
            progress_bar = st.progress(0)

            # Proses setiap file yang diupload
            for i, uploaded_file in enumerate(uploaded_files):
                sheet_to_read = sheet_options[uploaded_file.name]
                st.text(f"Memproses: {uploaded_file.name} (Sheet: {sheet_to_read})")

                # Proses file
                df, error = process_encrypted_excel(uploaded_file, password, sheet_to_read)

                if df is not None:
                    data_frames.append(df)
                    log_messages.append(f"✅ Berhasil membaca: {uploaded_file.name} (Sheet: {sheet_to_read})")
                    log_messages.append(f"    Total baris: {df.shape[0]}")
                else:
                    log_messages.append(f"❌ Gagal membaca {uploaded_file.name}: {error}")

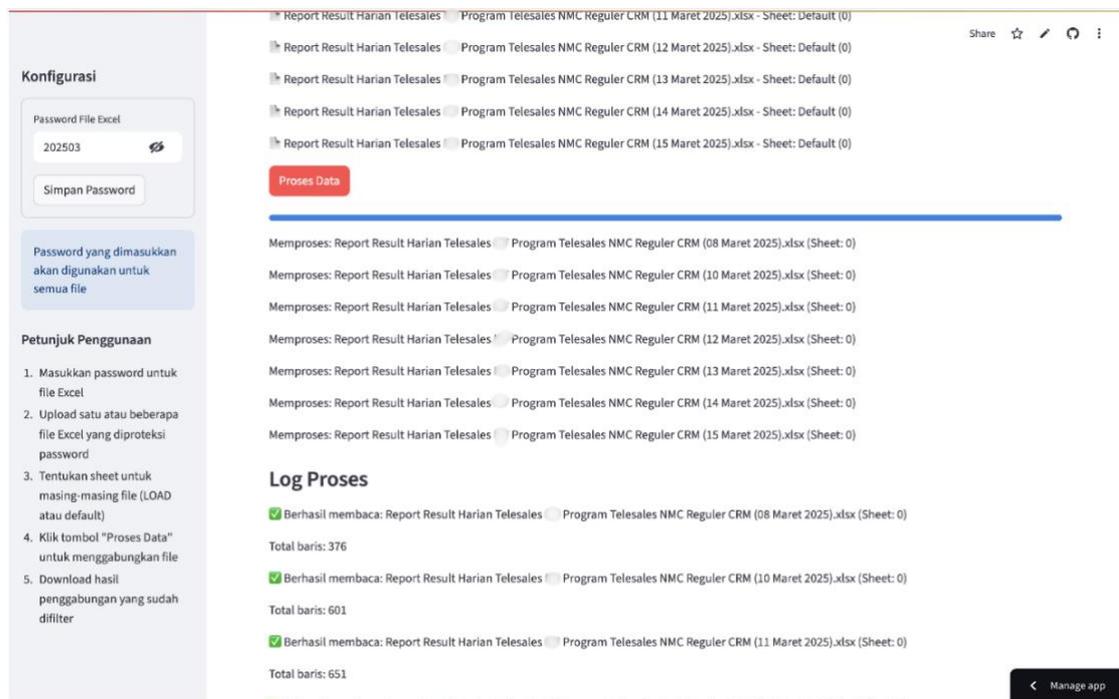
                # Update progress bar
                progress_bar.progress((i + 1) / len(uploaded_files))

            # Tampilkan log
            st.subheader("Log Proses")
            for msg in log_messages:
                st.text(msg)

            # Jika ada data yang berhasil dibaca
            if data_frames:
                # Gabungkan semua dataframe
                st.subheader("Hasil Penggabungan")
                merged_df = pd.concat(data_frames, ignore_index=True)
```

Gambar 3. 72 Potongan Kode Bagian Pemrosesan Data

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3. 73 Interface Saat Data Diproses

Langkah selanjutnya adalah standarisasi data. Proses ini mencakup normalisasi nama kolom, konversi format tanggal ke standar sistem (YYYY-MM-DD), serta pembersihan karakter yang tidak sesuai. Kolom-kolom seperti periode call, nama customer, dan nomor telepon disesuaikan untuk menjaga konsistensi antar file.

Setelah seluruh data digabungkan ke dalam satu dataframe, sistem melakukan proses deduplikasi berdasarkan kombinasi atribut unik, seperti nama customer, nomor telepon, dan tanggal panggilan. Data yang telah tersaring ini kemudian divalidasi untuk memastikan bahwa hanya data yang memenuhi standar yang akan disimpan dalam hasil akhir.

Sebagai keluaran, sistem menampilkan ringkasan jumlah baris data per *file*, hasil deduplikasi, serta menyediakan tautan untuk mengunduh hasil akhir dalam format *Excel*. Potongan kode untuk tahapan ini ditunjukkan pada **Gambar 3.74**.

```

# Tampilkan statistik
col1, col2, col3 = st.columns(3)
with col1:
    st.metric("Total Data (Sebelum Filter)", f"{total_before_filter;}")
with col2:
    st.metric("Total Data (Setelah Filter)", f"{total_after_filter;}")
with col3:
    st.metric("Duplicate Difilter", f"{total_before_filter - total_after_filter;}")

# Tampilkan preview data
st.subheader("Preview Data Hasil Filtering")
st.dataframe(unique_customers.head(10))

# Tampilkan link download
st.subheader("Download Hasil")
col1, col2 = st.columns(2)
with col1:
    st.markdown(get_excel_download_link(unique_customers, "filtered_unique_customers.xlsx"), unsafe_allow_html=True)
    st.caption("Data customer unik (terbaru)")
with col2:
    st.markdown(get_excel_download_link(merged_df, "all_merged_data.xlsx"), unsafe_allow_html=True)
    st.caption("Semua data yang digabungkan")

```

Gambar 3. 74 Potongan Kode Untuk Hasil dan Opsi Download

Seluruh proses dilakukan secara real-time di antarmuka Streamlit yang *user friendly*, sehingga tim telesales dapat menjalankannya tanpa perlu keterampilan teknis lanjutan.

### 3.2.5.4 Hasil dan Evaluasi

Implementasi sistem *Pipeline Data Merge* memberikan dampak langsung terhadap proses kerja tim telesales PT XYZ Finance. Meskipun tidak dilakukan pengujian formal dengan metrik tertentu, efisiensi sistem ini dapat dirasakan secara nyata dalam aktivitas harian pengguna.

Sebelumnya, proses penggabungan data pelanggan dari berbagai file Excel dilakukan secara manual, yang memerlukan waktu dan ketelitian ekstra, terutama karena setiap file dilindungi oleh sandi dan berpotensi mengandung data duplikat. Setelah aplikasi diimplementasikan, pengguna dapat mengunggah banyak file sekaligus, memasukkan sandi sekali, dan mendapatkan hasil gabungan yang sudah terstandarisasi dan bebas duplikasi secara otomatis.

Aplikasi ini juga menyediakan antarmuka pengguna (*user interface*) yang sederhana dan mudah digunakan, sehingga staf non-teknis pun dapat mengoperasikannya tanpa

pelatihan tambahan. Gambar 3.75 berikut memperlihatkan salah satu tampilan antarmuka saat proses penggabungan file telah berhasil diselesaikan dan hasil ditampilkan dalam bentuk tabel.

The screenshot displays a web application interface for data management. On the left, there is a 'Konfigurasi' sidebar with a 'Password File Excel' field set to '202503' and a 'Simpan Password' button. Below it, a 'Petunjuk Penggunaan' section lists five steps: 1. Masukkan password untuk file Excel; 2. Upload satu atau beberapa file Excel yang diproteksi password; 3. Tentukan sheet untuk masing-masing file (LOAD atau default); 4. Klik tombol "Proses Data" untuk menggabungkan file; 5. Download hasil penggabungan yang sudah difilter.

The main area shows a summary of data: 'Total Data (Sebelum Filter): 4,116', 'Total Data (Setelah Filter): 4,109', and 'Duplicate Difilter: 7'. Below this is a 'Preview Data Hasil Filtering' table with the following data:

crm_no	sales_no	customer_main_no	customer_no	load	name	address	bill_address
4115	CRM23001941876	None		Y	NURAINI	None	MW TMUR PAS
3738	CRM21001609908	None		Y	ANWAR GEA	PERUM PURNAMA BLOK IA NOJA	BTN BUNUT JL F
3744	CRM22032544613	None		Y	RONI ANDESKO	JL GAJAH	JL GAJAH RUMA
3743	CRM22032548046	None		Y	EMONIUS ZALUKHU	JL SEGAR I	JL JAWA HANGT
3742	CRM22000162213	None		Y	ELPIN MARBUN	JL PALEMBANG	JL LINTAS TIMUI
3741	CRM22003445109	None		Y	VANENGENI HIA	JL PEMUDA	JL PASIR PUTIH
3740	CRM22000121572	None		Y	DAMIRUS	JL BADAK UJUNG GG M YUNUS	JL BADAK UJUN
3739	CRM22000077746	None		Y	LISKEN HASIBUAN	JL RAMAH KASIH	JL RAMAH KASIH
3737	CRM21001555640	None		Y	DEDY PRATAMA	DUSUN MERANTI JAYA	JL PALAPA UJUN
3746	CRM22032698699	None		Y	ROTUA DERYANTI BR TANJUNG	JL PALAS PASTORAN	JLN PALAS PAST

Below the table, there are 'Download Hasil' options: 'Download filtered unique customers.xlsx' (Data customer unik (terbaru)) and 'Download all merged data.xlsx' (Semua data yang digabungkan). A 'Manage app' button is visible in the bottom right corner.

Gambar 3. 75 Tampilan Interface Hasil dan Opsi Download

Secara keseluruhan, sistem ini membantu mempercepat proses distribusi data untuk keperluan telesales dan mengurangi beban administratif yang sebelumnya dilakukan secara manual. Aplikasi ini menjadi solusi yang praktis dan adaptif terhadap kebutuhan operasional yang dinamis di lingkungan PT XYZ Finance.

### 3.3 Kendala yang Ditemukan

Selama pelaksanaan proyek magang di Departemen CRM Data Mining XYZ Finance, ditemukan sejumlah kendala yang berdampak pada kelancaran proses pengembangan sistem segmentasi pelanggan untuk keperluan event pada lini bisnis XYZ Finance MPF. Kendala-kendala tersebut dapat diklasifikasikan ke dalam empat kategori utama, yakni koordinasi, kompleksitas data, keterbatasan teknis, serta tantangan interpretasi hasil analisis.

Pertama, kendala muncul dari aspek koordinasi dengan pihak-pihak terkait. Beberapa pemangku kepentingan yang berperan penting dalam menyediakan data atau memberikan konfirmasi informasi mengalami beban kerja yang tinggi. Kondisi ini menyebabkan proses komunikasi menjadi tidak optimal dan berdampak pada keterlambatan dalam penyelesaian tahapan proyek tertentu.

Kedua, kompleksitas struktur data menjadi tantangan yang signifikan. Data pelanggan yang digunakan dalam proyek ini berasal dari berbagai sistem internal yang memiliki format dan struktur yang tidak seragam. Ditemukan pula ketidakkonsistenan nilai antarvariabel yang mempersulit proses integrasi dan cleansing data. Hal ini berdampak pada meningkatnya durasi pengerjaan tahap pra-pemrosesan sebelum dilakukan analisis lebih lanjut.

Ketiga, dari sisi teknis, pelaksanaan analisis segmentasi memerlukan daya komputasi yang cukup besar. Dataset yang digunakan memiliki ukuran yang besar, sementara algoritma segmentasi serta proses visualisasi memerlukan penggunaan memori yang tinggi. Akibatnya, waktu eksekusi program menjadi lebih lama dan tidak jarang terjadi kendala berupa crash atau lag pada saat pemrosesan berlangsung.

Keempat, hasil segmentasi yang dihasilkan perlu diterjemahkan ke dalam bentuk strategi operasional yang dapat dipahami dengan mudah oleh tim non-teknis, khususnya bagian pemasaran dan pengelola event. Proses interpretasi hasil analisis memerlukan pendekatan komunikasi yang tepat agar insight yang diperoleh dapat ditindaklanjuti secara efektif dan sesuai dengan kebutuhan operasional perusahaan.

### **3.4 Solusi atas Kendala yang Ditemukan**

Untuk mengatasi berbagai kendala yang muncul selama pelaksanaan proyek segmentasi pelanggan di Departemen CRM Data Mining XYZ Finance, telah diterapkan sejumlah solusi strategis yang bersifat adaptif dan teknis. Solusi ini dirancang agar mampu mengatasi hambatan koordinasi, kompleksitas data, keterbatasan teknis, serta tantangan dalam interpretasi hasil analisis.

Pertama, dalam merespons kendala koordinasi dengan pemangku kepentingan, diterapkan pendekatan komunikasi yang lebih proaktif dan terstruktur. Tim pelaksana secara konsisten melakukan *follow-up* pada waktu yang dinilai optimal, seperti pada jam kerja pagi hari. Selain itu, disusun daftar kebutuhan dan pertanyaan secara sistematis guna meningkatkan efisiensi dan fokus dalam setiap sesi diskusi, sehingga mempercepat proses pengambilan keputusan.

Kedua, untuk mengatasi kompleksitas data yang bersumber dari berbagai sistem internal, dilakukan proses standardisasi format data dan pemetaan (*mapping*) antar atribut. Validasi silang antar dataset turut dilakukan untuk memastikan konsistensi, kelengkapan, dan integritas data sebelum masuk ke tahap analisis lanjutan. Pendekatan ini membantu mengurangi potensi kesalahan akibat inkonsistensi struktur data.

Ketiga, terkait kendala teknis yang berkaitan dengan kapasitas pemrosesan data, dilakukan optimasi terhadap skrip Python yang digunakan dalam analisis. Teknik optimasi meliputi penerapan *vectorization* untuk mengurangi perulangan eksplisit, penghapusan bagian kode yang tidak efisien, serta penggunaan struktur data yang lebih hemat memori. Di samping itu, penggunaan *version control* melalui GitHub memungkinkan dokumentasi dan manajemen perubahan kode secara sistematis dan kolaboratif.

Keempat, untuk menjawab tantangan dalam proses interpretasi hasil segmentasi kepada tim non-teknis, dikembangkan visualisasi data yang informatif dan mudah dipahami dengan bantuan platform interaktif seperti Streamlit. Visualisasi ini dilengkapi dengan rekomendasi tindakan operasional yang sesuai dengan karakteristik masing-masing segmen pelanggan, sehingga hasil analisis dapat langsung diimplementasikan oleh tim pemasaran dan penyelenggara event XYZ Multipurpose.