

Pengembangan Aplikasi ETL Untuk Pre Processing Business Intelligence

1st Imam Algiza
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

imamalgiza@student.telkomuniversity.
ac.id

2nd Roswan Latuconsina.
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

roswan@telkomuniversity.ac.id

3rd Purba Daru Kusuma
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

purbodaru@telkomuniversity.ac.id

Abstrak — Dalam era digital yang berkembang pesat, pengelolaan data menjadi kunci perkembangan perusahaan, sehingga kebutuhan akan aplikasi Business Intelligence (BI) yang mampu mengumpulkan, menganalisis, dan mengonversi data menjadi wawasan berharga semakin meningkat. Proses analisis data sering kali terkendala data yang tidak terstruktur atau kurang terkelola, serta keterbatasan keterampilan analitik di banyak organisasi. Dengan aplikasi ETL untuk Pre-Processing BI, fokus utama adalah integrasi sumber data, ekstraksi, pemodelan, analisis, dan visualisasi data, termasuk penggunaan algoritma pembelajaran mesin untuk pemodelan prediktif yang memberikan wawasan jelas. Aplikasi ini dirancang untuk menyederhanakan analisis dan visualisasi data, membersihkan data dari kekacauan, serta memastikan pengujian dan optimisasi berjalan lancar demi menghasilkan output sesuai ekspektasi.

Kata kunci — Business Intelligence, Analisis Data, Visualisasi Data, Machine Learning.

I. PENDAHULUAN

Dalam era digital yang berkembang pesat, kemampuan mengelola dan memanfaatkan data menjadi faktor kunci kesuksesan organisasi, terutama bagi perusahaan baru yang ingin bersaing di pasar global. Data memiliki potensi besar untuk memberikan wawasan berharga yang mendukung pengambilan keputusan strategis, peningkatan efisiensi operasional, dan keunggulan kompetitif. Namun, banyak perusahaan pemula belum optimal memanfaatkan data, sehingga peluang besar tersebut sering terabaikan.

Sebaliknya, perusahaan mapan yang telah mengadopsi teknologi seperti Business Intelligence (BI) menikmati keunggulan signifikan. BI mampu mengolah data mentah menjadi wawasan yang mudah dipahami untuk mendukung keputusan strategis, mengidentifikasi peluang baru, dan meningkatkan efisiensi operasional. Dalam persaingan global, kemampuan merespons perubahan pasar secara cepat dan akurat menjadi sangat penting, dan BI memberikan fondasi kuat bagi perusahaan untuk mencapai hal tersebut[1].

Sayangnya, banyak perusahaan baru masih menghadapi tantangan dalam mengadopsi BI karena keterbatasan sumber daya atau pengetahuan teknis. Hal ini membuat mereka berisiko tertinggal dari pesaing yang lebih dahulu memanfaatkan teknologi tersebut. Oleh karena itu, penting bagi perusahaan baru untuk segera memahami nilai strategis pengelolaan data dan mulai menerapkan teknologi BI agar mampu bertahan dan berkembang dalam persaingan yang semakin ketat.

II. KAJIAN TEORI

Business Intelligence (BI) adalah serangkaian proses, teknologi, dan alat yang digunakan untuk mengolah data mentah menjadi informasi yang relevan dan bermanfaat dalam mendukung pengambilan keputusan. BI memungkinkan organisasi memahami data mereka, mengidentifikasi pola, tren, dan wawasan strategis untuk meningkatkan efisiensi operasional serta daya saing. Proses utama dalam BI meliputi pengumpulan, analisis, dan penyajian data, yang disampaikan melalui laporan atau dashboard visual agar mudah dipahami, bahkan oleh pengguna tanpa latar belakang teknis. Dalam era digital, BI menjadi elemen penting untuk mendorong pengambilan keputusan berbasis data (data-driven decision-making).

Power BI adalah salah satu alat Business Intelligence yang dikembangkan oleh Microsoft, dirancang untuk membantu organisasi dalam mengelola, menganalisis, dan memvisualisasikan data. Power BI memungkinkan integrasi dengan berbagai sumber data, seperti Excel, SQL Server, dan aplikasi SaaS, yang memberikan fleksibilitas tinggi dalam pengolahan data. Alat ini memiliki antarmuka pengguna yang intuitif, sehingga pengguna non-teknis dapat dengan mudah membuat visualisasi, dashboard, dan laporan interaktif. Fitur-fitur seperti Query Editor dan Data Analysis Expressions (DAX) memberikan kemampuan untuk memodelkan data dengan fleksibilitas yang mendukung analisis mendalam[2].

Keunggulan utama Power BI terletak pada kemampuannya untuk menyajikan data dalam berbagai bentuk visualisasi, seperti grafik batang, pie chart, atau scatter plot, yang interaktif dan informatif. Selain itu, Power BI mendukung kolaborasi real-time melalui layanan Power BI Service, memungkinkan laporan dan dashboard dipublikasikan secara online untuk diakses oleh tim atau pemangku kepentingan kapan saja. Dengan skalabilitas yang tinggi, Power BI dapat digunakan oleh individu maupun organisasi besar, menjadikannya salah satu solusi BI yang paling populer[3].

Meski memiliki keunggulan signifikan, Power BI juga memiliki keterbatasan, seperti ketergantungan pada koneksi internet untuk pembaruan real-time dan keterbatasan pengolahan data yang sangat besar pada perangkat keras yang kurang memadai. Namun, dengan biaya yang relatif terjangkau, fitur yang komprehensif, dan kemudahan penggunaannya, Power BI tetap menjadi pilihan ideal bagi perusahaan yang ingin memulai transformasi digital mereka.

dengan mengadopsi Business Intelligence sebagai bagian dari strategi pengambilan keputusan berbasis data.

III. METODE

A. ETL (Extract, Transform and Load)

Proses ETL adalah solusi yang dirancang untuk mengatasi tantangan utama yang dihadapi oleh banyak UKM dalam mengadopsi teknologi Business Intelligence (BI). Organisasi atau perusahaan kecil sering kali kesulitan dalam mengelola dan menganalisis data dari berbagai sumber karena keterbatasan sumber daya dan kemampuan teknis. Dengan menggunakan metode ETL, data dari berbagai sumber dapat diintegrasikan, dibersihkan, dan diolah menjadi informasi yang siap digunakan untuk pengambilan keputusan yang lebih baik [4].

1. *Extract* (Ekstraksi):

Tahap ini dimulai dengan mengunggah file CSV yang berisi data mentah dari berbagai sumber. Data tersebut kemudian di-convert dan dimasukkan ke dalam database MySQL, sebuah langkah penting yang memungkinkan integrasi data dari berbagai sumber ke dalam satu platform sentral.

2. *Transformation* (Transformasi):

Setelah data diekstraksi, langkah berikutnya adalah transformasi data. Pada tahap ini, data disaring, dibersihkan, dan disesuaikan dengan kebutuhan analisis. Misalnya, data yang redundan atau tidak relevan dihapus, sementara data yang hilang diisi atau diimputasi sesuai kebutuhan analisis. Proses ini juga mencakup penerapan aturan bisnis tertentu, sehingga data yang dihasilkan sesuai dengan kebutuhan spesifik dari organisasi.

3. *Load* (Memuat):

Tahap terakhir adalah memuat data yang sudah diolah ke dalam sistem penyimpanan tujuan. Dengan data yang sudah terstruktur dan bersih, organisasi dapat dengan mudah mengakses dan menggunakan data ini untuk analisis lebih lanjut atau untuk pelaporan yang dibutuhkan.

B. Data Visualization Algorithm

Visualisasi data merupakan aspek penting dalam BI yang memungkinkan organisasi untuk memahami data yang kompleks melalui representasi grafis yang sederhana. Berikut adalah beberapa jenis visualisasi yang digunakan.

1. Grafik Batang (Bar):

Digunakan untuk membandingkan kuantitas antara berbagai kategori. Ini memudahkan pengguna untuk melihat perbedaan antar kategori secara visual, baik dalam format vertikal maupun horizontal.

2. Diagram Lingkaran (Pie):

Efektif untuk menunjukkan proporsi atau persentase dari suatu keseluruhan. Setiap bagian dari lingkaran mewakili kontribusi relatif dari sebuah kategori terhadap total.

3. Scatter Chart:

Digunakan untuk mengidentifikasi hubungan atau pola antara dua variabel. Ini sangat berguna untuk mendeteksi outliers atau data yang tidak sesuai dengan tren umum.

4. Line Chart:

Digunakan untuk menunjukkan tren atau perubahan nilai seiring dengan waktu. Grafik ini membantu pengguna melacak perubahan data dari waktu ke waktu.

5. Donut Chart:

Variasi dari grafik pie yang memungkinkan penambahan informasi tambahan di tengah lingkaran, seperti total keseluruhan atau kategori utama.

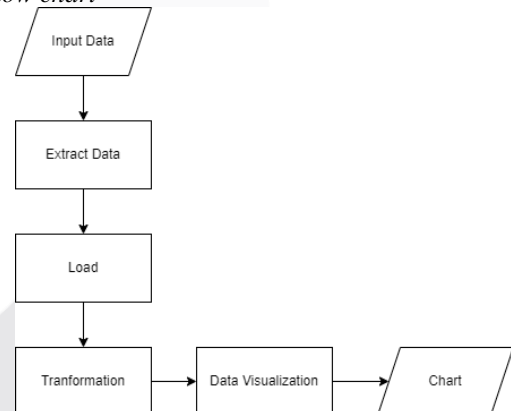
6. Table Chart:

Menyajikan data dalam format tabel yang memungkinkan pembaca untuk membandingkan nilai antar kategori secara mudah dan terstruktur.

Untuk mengatasi tantangan yang dihadapi organisasi atau perusahaan kecil dalam mengadopsi BI, metode ETL dan visualisasi data yang dijelaskan di atas merupakan solusi efektif. Proses ETL memungkinkan organisasi atau perusahaan untuk mengintegrasikan berbagai sumber data menjadi satu database terpusat, sementara visualisasi data memudahkan interpretasi dan penggunaan data untuk pengambilan keputusan. Melalui penggunaan algoritma yang efisien, organisasi atau perusahaan kecil dapat mengatasi keterbatasan teknis dan sumber daya, serta memanfaatkan BI untuk meningkatkan efisiensi operasional dan bersaing di pasar yang kompetitif.

Dalam konteks ini, kajian teori dan latar belakang yang telah disusun menunjukkan pentingnya adopsi BI bagi organisasi atau perusahaan kecil. Metode yang dipaparkan memberikan solusi konkret yang dapat diterapkan oleh organisasi atau perusahaan kecil untuk mengatasi keterbatasan tersebut dan memanfaatkan data dengan lebih efektif, sehingga dapat beroperasi lebih efisien dan mengambil keputusan yang lebih tepat berdasarkan analisis data yang mendalam.

C. Flow chart



Gambar 1
Flow chart system

Pada gambar 1 merupakan *flow chart*, alur ini menggambarkan bagaimana data mentah dikumpulkan, diproses, dan akhirnya disajikan dalam bentuk visualisasi yang mudah dipahami untuk mendukung pengambilan keputusan yang lebih baik dalam organisasi.

IV. HASIL DAN PEMBAHASAN

A. Proses ETL (Extract, Transform and Load)

Aplikasi yang dikembangkan memiliki kemampuan untuk mengintegrasikan data dari sumber berbasis CSV. Data dalam format ini diproses melalui tahapan Extract, Transform, Load (ETL) untuk memastikan bahwa data mentah dari berbagai sumber dapat diolah menjadi informasi yang siap digunakan. Proses ini mencakup ekstraksi data dari file CSV, transformasi data untuk membersihkan dan

mengonversi data mentah ke dalam format yang sesuai, serta memuat data ke dalam sistem aplikasi untuk analisis lebih lanjut.

Data yang telah diinputkan dan tersimpan di dalam database MySQL maka sistem otomatis melakukan *load* data dan ditampilkan pada halaman web. Pada tabel 1 berisi tentang *library* yang digunakan pada *data processing*.

TABEL 1
Library Python

Library yang digunakan	
Pandas	Untuk manipulasi dan analisis data dalam python.
Matplotlib	Membuat visualisasi data dalam bentuk grafik 2D.
Seaborn	Untuk visualisasi data yang dibangun di atas Matplotlib.
Json	Standar python untuk membaca dan menulis data dalam bentuk format JSON (JavaScript Object Notation).
Os	Standar python yang menyediakan fungsi untuk berinteraksi dengan sistem operasi, seperti mengelola file dan direktori.

Pada tahap pemrosesan data melibatkan proses *data transformation* yang meliputi *data converting*, *data cleaning*, *fill missing data* dan *data aggregation*. Proses transformasi melibatkan berbagai jenis operasi yang bertujuan untuk meningkatkan kualitas data, mengonversi data menjadi format yang sesuai, dan memastikan bahwa data siap untuk analisis. Dalam implementasi ini, data diambil dari database MySQL berdasarkan tabel dan kolom yang dipilih oleh *user*. Data yang diambil kemudian dikonversi menjadi format JSON. Data yang telah diambil dari database disimpan dalam file *data.json*, yang memungkinkan *script* Python untuk membaca data dalam format yang dapat diproses lebih lanjut. *Script* Python kemudian dieksekusi menggunakan perintah *shell* dari PHP. *Script* ini memuat data dari file *data.json* dan menghasilkan visualisasi menggunakan *library matplotlib*. Di dalam python, data dari *data.json* dibaca dan diproses, kemudian *library matplotlib* digunakan untuk membuat grafik visualisasi berdasarkan data tersebut.

```
def convert_data_types(df):
    for column in df.columns:
        try:
            df[column] = pd.to_numeric(df[column], errors='raise')
        except ValueError:
            df[column] = df[column].astype(str)
    return df
```

GAMBAR 2
Convert data

Gambar 1 merupakan proses konversi data yang memiliki fungsi melakukan iterasi melalui setiap kolom dalam *DataFrame* untuk memastikan setiap kolom diperiksa dan dikonversi sesuai kebutuhan. Mencoba mengonversi kolom ke tipe data numerik menggunakan *pd.to_numeric()*. Jika kolom sudah numerik atau dapat dikonversi ke numerik, tipe data kolom tersebut akan diubah ke numerik. Jika konversi gagal (misalnya, karena ada nilai yang tidak bisa dikonversi ke numerik), *ValueError* akan ditangani dan kolom tersebut akan diubah tipe datanya menjadi string.

Tahapan pembersihan data memastikan kualitas data yang baik sebelum melakukan analisis lebih lanjut. Mengisi nilai yang hilang dalam *DataFrame* untuk meningkatkan

integritas dan kualitas data sehingga dapat digunakan dalam analisis lebih lanjut. Proses dimulai dengan melakukan iterasi atau perulangan melalui setiap kolom dalam *DataFrame*. Untuk setiap kolom, pertama-tama diperiksa apakah tipe data kolom tersebut adalah *numeric* atau *non-numeric* (kategorikal atau string).

Jika Kolom adalah *Numeric*

- Nilai yang hilang diisi dengan median kolom tersebut.
- Median adalah nilai tengah dari data yang diurutkan. Ini mengurangi pengaruh outlier (nilai yang sangat tinggi atau rendah) yang dapat mendistorsi hasil analisis.

Jika Kolom adalah *Non-Numeric*

- Nilai yang hilang diisi dengan modus kolom tersebut.
- Modus adalah nilai yang paling sering muncul dalam data. Ini memastikan bahwa nilai yang paling representatif untuk kategori tersebut digunakan sebagai pengganti nilai yang hilang.

```
def fill_missing_values(df):
    for column in df.columns:
        if pd.api.types.is_numeric_dtype(df[column]):
            # Fill missing values in numeric columns with median
            df[column] = df[column].fillna(df[column].median())
        else:
            # Fill missing values in non-numeric columns with mode
            df[column] = df[column].fillna(df[column].mode()[0])
    return df
```

GAMBAR 3
Fill missing value

Pada gambar 2 proses pembersihan data ini menggunakan metode median atau nilai tengah dari sekumpulan data yang telah diurutkan. Jika jumlah data adalah ganjil, median adalah nilai yang berada di posisi tengah. Jika jumlah data adalah genap, median adalah rata-rata dari dua nilai tengah.

Jika jumlah data ganjil:

$$\text{Median} = \frac{x_{(n+1)}}{2} \quad (1)$$

Jika dataset berisi [3,5,7], maka $n = 3$ (jumlah data adalah ganjil). Posisi median adalah $\frac{(3+1)}{2} = 2$ maka pada posisi kedua dalam data yang diurutkan, yaitu $x_2 = 5$

Jika jumlah data genap:

$$\text{Median} = \frac{x_{\frac{n}{2}} + x_{(\frac{n}{2})+1}}{2} \quad (2)$$

Jika dataset berisi [3,5,7,9], maka $n = 4$ (jumlah data adalah genap). Posisi median adalah rata-rata dari nilai pada posisi kedua dan ketiga

$$\frac{x_{\frac{4}{2}} + x_{(\frac{4}{2})+1}}{2} = \frac{x_2 + x_3}{2} \quad (3)$$

Nilai pada posisi kedua adalah $x_2 = 5$ dan nilai pada posisi ketiga adalah $x_3 = 7$ maka median adalah $\frac{5+7}{2} = 6$

```
def remove_duplicates(df):
    df.drop_duplicates(inplace=True)
    return df
```

GAMBAR 4
Remove duplicate data

Gambar 3 merupakan proses menghapus data yang sama. Menggunakan *drop_duplicates()* secara otomatis mengidentifikasi baris-baris yang duplikat berdasarkan nilai di semua kolom. Jika ada baris yang memiliki nilai yang sama persis di semua kolom, baris-baris tersebut dianggap duplikat. Setelah baris-baris duplikat diidentifikasi,

drop_duplicates() akan menghapus semua baris duplikat, hanya menyisakan satu baris untuk setiap set data yang unik.

```
def aggregate_data(df, x_column, y_column):
    if df.empty:
        print("DataFrame is empty after cleaning.")
        return pd.DataFrame()

    if pd.api.types.is_numeric_dtype(df[y_column]):
        aggregated_df = df.groupby(x_column)[y_column].agg('sum').reset_index()
    else:
        aggregated_df = df[[x_column, y_column]].drop_duplicates()

    return aggregated_df
```

Gambar 5
Aggregate data

Pada gambar 4 merupakan proses *aggregate data*. Dalam fungsi ini, data dikelompokkan berdasarkan nilai dalam kolom *x_column*, dan nilai dalam kolom *y_column* dijumlahkan untuk setiap grup. Jika *DataFrame* kosong setelah pembersihan, fungsi akan mencetak pesan dan mengembalikan *DataFrame* kosong. Data dikelompokkan berdasarkan nilai-nilai unik di kolom *x_column* menggunakan *df.groupby(x_column)*. Untuk setiap grup, nilai-nilai dalam kolom *y_column* dijumlahkan menggunakan *.agg('sum')*. Hasil agregasi dikembalikan sebagai *DataFrame* baru dengan hasil perhitungan yang telah diringkas. Tabel 2 dan 3 berisi contoh data yang sebelum diagregasi dan setelah diagregasi.

Jika isi dataframe seperti berikut:

TABEL 2
Data sebelum diagregasi

x_column	y_column
A	10
A	20
B	30
B	40

Maka data setelah diagregasi akan menjadi:

TABEL 3
Data setelah diagregasi

x_column	y_column
A	30
B	70

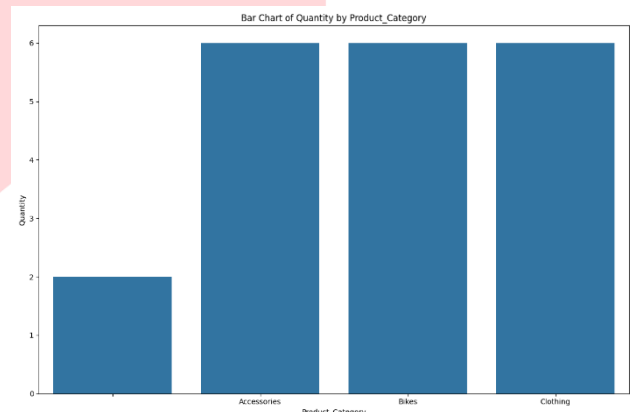
Pengujian aplikasi dilakukan melalui beberapa metode, termasuk alpha testing, beta testing, dan stress test, untuk memastikan bahwa aplikasi berfungsi sesuai ekspektasi:

1. Alpha Testing: Dilakukan secara internal untuk menguji semua fitur aplikasi. Hasilnya menunjukkan bahwa aplikasi berhasil memproses data dalam format CSV, menghasilkan visualisasi sesuai kebutuhan, dan mampu menangani data mentah dengan tingkat akurasi tinggi.
2. Beta Testing: Dilakukan oleh pengguna akhir untuk mengevaluasi pengalaman pengguna. Mayoritas responden memberikan umpan balik positif terhadap antarmuka aplikasi yang sederhana dan kemudahan dalam menggunakan fitur visualisasi data.
3. Stress Test: Menguji performa aplikasi dalam menangani data dalam jumlah besar. Aplikasi mampu memproses data hingga batas tertentu tanpa penurunan performa yang signifikan, meskipun terdapat batasan pada perangkat keras tertentu.
4. Penetration Test: Dilakukan untuk memastikan keamanan data yang diproses dalam aplikasi. Hasilnya menunjukkan bahwa aplikasi memiliki tingkat

keamanan yang memadai dengan perlindungan terhadap akses tidak sah.

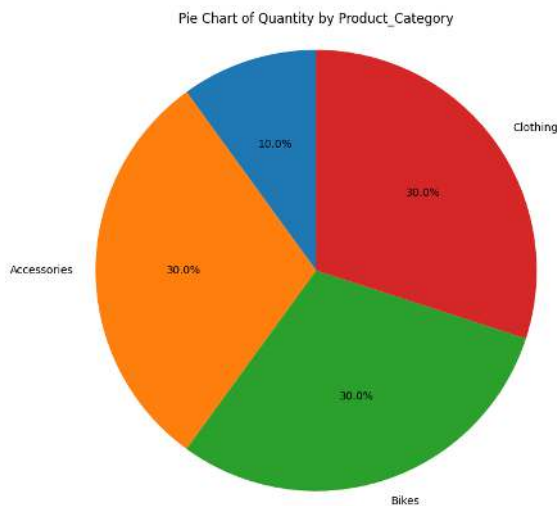
B. Data Visualization Algorithm

Data visualization melakukan proses mengubah data menjadi representasi grafis seperti grafik, diagram, atau peta, sehingga informasi dalam data dapat disajikan secara visual dan lebih mudah dipahami. Beberapa jenis visualisasi yang umum digunakan meliputi *bar chart*, *pie chart*, *scatter chart*, *donut chart*, *line chart* dan *table visualization*. Menggunakan data dari *DataFrame* proses ini melibatkan persiapan gambar, pembuatan grafik, penambahan judul dan label, pengaturan layout, dan akhirnya menyimpan gambar ke lokasi yang ditentukan. Masing-masing fungsi menggunakan metode dari *matplotlib* dan *seaborn* untuk menghasilkan visualisasi yang informatif dan mudah dibaca.



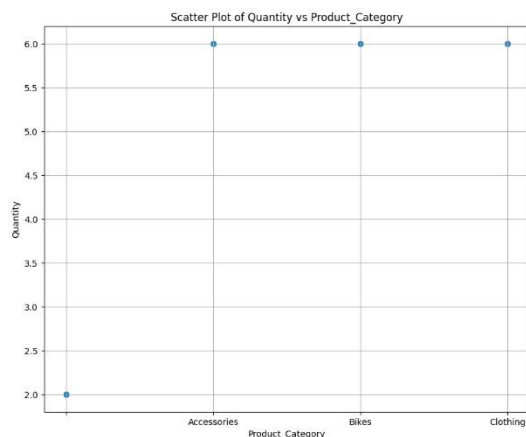
GAMBAR 6
Bar Chart

Pada gambar 5 merupakan hasil pengujian visualisasi data menggunakan *bar chart*. Keuntungan utama dari *bar chart* adalah kemudahannya dalam menginterpretasikan perbandingan antar kategori, yang sangat berguna dalam komunikasi data yang efektif. Grafik ini juga sangat fleksibel, dapat menangani data dalam jumlah kecil hingga besar, dan dapat dimodifikasi menjadi bentuk bertumpuk atau berkelompok untuk analisis lebih lanjut. *Bar chart* sangat sesuai untuk berbagai penggunaan, seperti membandingkan metrik bisnis, menampilkan hasil survei, statistik kesehatan, atau prestasi pendidikan. Untuk hasil terbaik, penting untuk menjaga kejelasan dengan label yang jelas dan skala yang tepat, serta menghindari desain yang terlalu rumit yang tidak menambah nilai informasi [5].



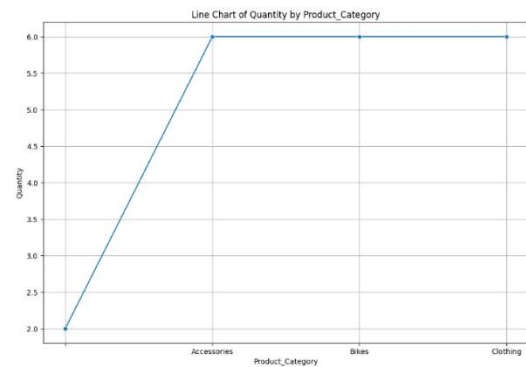
GAMBAR 7
Pie Chart

Gambar 6 merupakan hasil visualisasi menggunakan *pie chart*. Keunggulan utama dari *pie chart* terletak pada kemampuannya untuk mengkomunikasikan informasi dengan cara yang cepat dan mudah dipahami, membuatnya ideal untuk menggambarkan komposisi data dalam laporan bisnis, presentasi pendidikan, atau analisis demografis. Penggunaan warna yang berbeda untuk setiap kategori juga memperjelas distingsi antar kategori, meningkatkan keefektifan grafik dalam menyampaikan informasi [6].



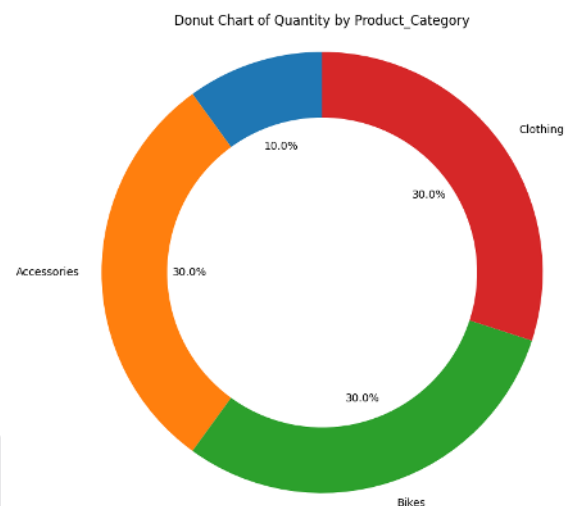
GAMBAR 8
Scatter Chart

Gambar 7 merupakan hasil visualiasi data menggunakan *scatter chart*. *Scatter chart* digunakan untuk melihat hubungan atau pola antara dua variabel. Dalam kode ini, *scatter chart* menampilkan nilai y terhadap x, dengan setiap titik mewakili pasangan nilai dari data. Kegunaan dari *scatter chart* ini untuk menemukan hubungan atau korelasi antara dua variabel dan mengidentifikasi tren atau pola data. *Scatter chart* ini sangat berguna untuk mmendeteksi *outliers* atau data yang tidak sesuai dengan tren umum [5].



GAMBAR 9
Line Chart

Gambar 8 merupakan hasil visualisasi data menggunakan *Line chart*, yang digunakan untuk menunjukkan tren atau perubahan nilai y seiring perubahan x. Dalam visualisasi ini, setiap titik data dihubungkan dengan garis, membantu mengidentifikasi arah perubahan data. Dengan line chart ini user dapat melacak perubahan data dari waktu ke waktu atau sepanjang suatu urutan, mengidentifikasi tren naik atau turun dan berguna untuk data time-series atau data yang bersifat sekuensial [7].



GAMBAR 10
Donut chart

Gambar 9 merupakan hasil visualisasi data menggunakan *donut chart*. Keuntungan dari *donut chart* terletak pada estetikanya yang lebih modern dan kemampuannya untuk mengurangi kepadatan visual yang sering terjadi pada *pie chart* penuh. Ruang kosong di tengah memberikan kelegaan visual dan bisa digunakan untuk menyampaikan pesan kunci atau data yang menonjol secara efektif. Grafik ini sangat berguna untuk menampilkan data yang membutuhkan konteks tambahan atau ketika ingin menekankan aspek tertentu dari data yang disajikan [7].

Table with Total (Total rows: 5)

Product_Category	Quantity
	2.0
Accessories	6.0
Bikes	6.0
Clothing	6.0
Total	20.0

GAMBAR 11
Table visualization

Table visualization sangat berguna untuk menampilkan data yang terstruktur dan memungkinkan pembaca untuk dengan mudah membandingkan nilai-nilai antar kategori. Keuntungan utama dari *table* adalah menampilkan data yang kompleks dalam bentuk yang mudah dipahami dan dianalisis, serta memberikan detail yang presisi di mana ini sangat berguna dalam konteks laporan mengenai analisis data untuk setiap detail data yang dianalisis [7].

D. Whitebox Testing Data Transformation

TABEL 4
Whitebox testing data transformation

Test Case Description	Steps	Actual Result
Mengkonversi Tipe Data	Panggil <i>convert_data_types(df)</i> dengan tipe data campuran	Kolom berhasil dikonversi ke tipe data yang sesuai
Mengisi Nilai yang Hilang	Panggil <i>fill_missing_values(df)</i> dengan nilai yang hilang	Nilai yang hilang berhasil diisi dengan median atau mode
Menghapus Duplikasi	Panggil <i>remove_duplicates(df)</i> dengan baris duplikat	Baris duplikat berhasil dihapus
Mengagregasi Data	Panggil <i>aggregate_data(df, x_column, y_column)</i>	<i>DataFrame</i> berhasil teragregasi dengan benar

Hasil pengujian pada tabel 4 menunjukkan bahwa semua pengujian data transformasi berhasil. Proses *whitebox testing* ini memastikan bahwa setiap bagian dari *script* berfungsi dengan baik dan dapat diandalkan untuk memproses data dengan benar.

E. Whitebox Testing Data Visualization

TABEL 5
Whitebox testing data visualization

Test Case Description	Steps	Actual Result
Membuat Bar Chart	Panggil <i>create_bar_chart(df, x_column, y_column, output_folder)</i>	<i>Bar Chart</i> berhasil disimpan
Membuat Pie Chart	Panggil <i>create_pie_chart(df, x_column, y_column, output_folder)</i>	<i>Pie Chart</i> berhasil disimpan

Membuat Donut Chart	Panggil <i>create_donut_chart(df, x_column, y_column, output_folder)</i>	<i>Donut Chart</i> berhasil disimpan
Membuat Line Chart	Panggil <i>create_line_chart(df, x_column, y_column, output_folder)</i>	<i>Line Chart</i> berhasil disimpan
Membuat Scatter Chart	Panggil <i>create_scatter_plot(df, x_column, y_column, output_folder)</i>	<i>Scatter Chart</i> berhasil disimpan
Membuat Table Visualization	Panggil <i>create_table_chart(df, x_column, y_column, output_folder)</i>	<i>Table Visualization</i> berhasil disimpan

Hasil pengujian pada tabel 5 menunjukkan bahwa semua pengujian data visualisasi berhasil. Tidak ada error atau masalah yang ditemukan selama pengujian ini, sehingga *script* algoritma ini siap digunakan untuk kebutuhan visualisasi data.

V. KESIMPULAN

Aplikasi ETL (Extract, Transform, Load) untuk Pre-Processing Business Intelligence (BI) yang dikembangkan dalam penelitian ini telah terbukti efektif dalam mengintegrasikan berbagai sumber data menjadi satu database terpusat. Visualisasi data yang dihasilkan dari aplikasi ini memudahkan interpretasi dan penggunaan data untuk pengambilan keputusan yang lebih baik. Dengan menggunakan algoritma yang efisien, aplikasi ini dapat membantu organisasi atau perusahaan kecil mengatasi keterbatasan teknis dan sumber daya, sehingga mereka dapat bersaing di pasar yang kompetitif. Implementasi BI melalui metode ETL ini memungkinkan perusahaan untuk memanfaatkan data secara lebih efektif, meningkatkan efisiensi operasional, dan mendukung pengambilan keputusan yang lebih tepat.

REFERENSI

- [1] datagalaxy.com, "Gartner's top trends in data & analytics for 2023." Diakses: 14 Agustus 2024. [Daring]. Tersedia pada: <https://www.datagalaxy.com/en/blog/gartner-recap-top-trends-in-data-and-analytics-for-2023/>
- [2] I. A. A. Alsibhawi, J. B. Yahaya, dan H. B. Mohamed, "Business Intelligence Adoption for Small and Medium Enterprises: Conceptual Framework," 1 April 2023, MDPI. doi: 10.3390/app13074121.
- [3] Brett Powell, Microsoft Power BI cookbook: Creating business intelligence solutions of analytical data models, reports, and dashboards. Packt Publishing Ltd., 2017.
- [4] N. A. El-Adaileh dan S. Foster, "Successful business intelligence implementation: a systematic literature

- review,” 14 November 2019, Emerald Group Holdings Ltd. doi: 10.1108/JWAM-09-2019-0027.
- [5] T. Barker dan J. Westfall, *Pro Data Visualization Using R and JavaScript*. Berkeley, CA: Apress, 2022. doi: 10.1007/978-1-4842-7202-2.
- [6] Bernadette Bautista, “Creating Compelling Visuals: Pie Charts for Effective Data Storytelling,” *storytellingwithcharts.com*. Diakses: 15 Agustus 2024. [Daring]. Tersedia pada: <https://www.storytellingwithcharts.com/creating-compelling-visuals-pie-charts-effective-data-storytelling/>
- [7] M. Khan dan S. Shah Khan, “Data and Information Visualization Methods, and Interactive Mechanisms: A Survey,” 2011.

