

Sejarah Singkat Streamlit

Streamlit adalah sebuah platform open-source yang telah merevolusi cara pengembang dan data scientist membangun aplikasi web interaktif. Diluncurkan pada tahun 2019, Streamlit dirancang untuk menyederhanakan proses pengembangan dengan memungkinkan pengguna untuk membuat aplikasi web menggunakan bahasa pemrograman Python, tanpa perlu keahlian khusus dalam pemrograman web yang kompleks seperti JavaScript atau HTML. Dengan filosofi "Pythonic" yang mendasarinya, Streamlit memungkinkan pengembang untuk fokus pada logika bisnis dan analisis data, sementara platform ini mengurus tampilan dan interaksi web.

Kesederhanaan penggunaannya dan kemampuannya untuk memungkinkan pembuatan prototipe dan aplikasi data yang interaktif dengan cepat telah membuat Streamlit meraih popularitas yang pesat di kalangan komunitas pengembang dan data scientist. Dengan dukungan dari komunitas yang terus berkembang, Streamlit terus mengalami pertumbuhan, termasuk peluncuran Streamlit for Teams yang memungkinkan berbagi aplikasi dengan tim secara mudah. Sebagai proyek open-source, Streamlit menjadi alat yang tak ternilai bagi mereka yang ingin dengan cepat membangun aplikasi web data interaktif, dashboard, dan prototipe dengan Python.

Pemrosesan Data

Pemrosesan data sebagai berikut:

1. Mengimpor beberapa pustaka yang diperlukan, seperti ``pandas`` untuk manipulasi data, ``RandomForestClassifier`` dari ``sklearn`` untuk pembuatan model Random Forest, ``train_test_split`` untuk membagi data menjadi data pelatihan dan uji, ``streamlit`` untuk membuat aplikasi web, ``LabelEncoder`` untuk mengkodekan label kategori, ``accuracy_score`` untuk mengukur akurasi model, ``matplotlib.pyplot`` untuk membuat visualisasi, dan ``plotly.express`` untuk visualisasi data dengan Plotly.
2. Membaca data dari file CSV menggunakan ``pd.read_csv()`` dan mengonversi kolom 'TGL_INVC' menjadi format tanggal yang dapat diolah.
3. Mengekstraksi komponen tanggal (hari, bulan, tahun) dari kolom 'TGL_INVC' dan menghapus kolom tersebut karena tidak diperlukan lagi.
4. Membuat sebuah sidebar di aplikasi Streamlit yang memungkinkan pengguna untuk memilih apakah ingin melihat data SALES atau RETUR.
5. Data dibagi menjadi dua: ``sales_data`` untuk data SALES dan ``retur_data`` untuk data RETUR. Selain itu, data juga di-filter sesuai pilihan pengguna dengan variabel ``filtered_data``.
6. Menghitung jumlah QTYSALES atau RETUR untuk setiap barang sesuai dengan filter yang dipilih oleh pengguna. Hasil perhitungan ini disimpan dalam ``sales_return_by_item`` dengan mengganti nama kolom menjadi 'JUMLAH BARANG' atau 'QTYRETUR' sesuai dengan filter.
7. Menampilkan nama barang yang banyak terjual (SALES) atau banyak dikembalikan (RETUR) sesuai dengan filter yang dipilih oleh pengguna.
8. Menambahkan kolom "Jumlah QTYSALES" pada Data Tabel SALES dan mengganti nama kolom jika filter adalah RETUR, dan menambahkan tanda minus pada jumlah RETUR.
9. Data dibagi menjadi fitur (X) dan target (y) untuk melatih model Random Forest. Objek ``LabelEncoder`` digunakan untuk mengkodekan kolom-kolom kategori.
10. Mengukur sejauh mana model Random Forest akurat dengan akurasi dan mengonversinya ke persentase.
11. Menampilkan hasil analisis dalam bentuk tabel, grafik batang, dan grafik clustered column yang membandingkan akurasi antara SALES dan RETUR.

12. Seluruh pemrosesan data dan visualisasi tersebut terintegrasi ke dalam aplikasi web menggunakan Streamlit, sehingga pengguna dapat dengan mudah mengakses dan menginterpretasi hasil analisis data tersebut.

Random Forest (Decision Tree)

Pada kode di atas, terdapat implementasi dari algoritma Random Forest. Random Forest adalah salah satu algoritma yang populer dalam machine learning yang digunakan untuk tugas klasifikasi dan regresi. Algoritma ini memanfaatkan teknik ensemble learning dengan menggabungkan banyak pohon keputusan (decision tree) menjadi sebuah "hutan" yang dapat menghasilkan prediksi yang lebih akurat.

Pertama, data dibagi menjadi dua bagian, yaitu data pelatihan (X_{train} , y_{train}) dan data uji (X_{test} , y_{test}) menggunakan metode `train_test_split`. Data pelatihan digunakan untuk melatih model, sedangkan data uji digunakan untuk menguji sejauh mana model dapat membuat prediksi yang akurat.

Selanjutnya, model Random Forest dibangun dengan menginisialisasi objek `clf` (classifier) dengan 100 pohon keputusan (`n_estimators=100`) dan penggunaan nilai `random_state=42` untuk memastikan hasil yang dapat direproduksi. Setelah itu, model tersebut dilatih (`fit`) menggunakan data pelatihan (X_{train} , y_{train}).

Setelah model dilatih, langkah selanjutnya adalah melakukan prediksi pada data uji (X_{test}) menggunakan model yang telah dibuat. Hasil prediksi disimpan dalam variabel `y_pred`.

Untuk mengukur sejauh mana model Random Forest ini akurat, digunakan metrik akurasi (`accuracy_score`). Akurasi mengukur seberapa banyak prediksi yang benar dari keseluruhan prediksi yang dibuat oleh model. Hasil akurasi tersebut kemudian dikonversi menjadi persentase (`accuracy_percentage`) agar lebih mudah dipahami.

Interface / Hasil Program SALES

Penjelasan Tabel diatas:

Tabel "Hasil Barang Penjualan (Sales) Paling Tinggi" memberikan informasi tentang produk atau barang yang memiliki penjualan tertinggi dalam data SALES. Tabel ini berisi beberapa kolom sebagai berikut:

1. **Nama Barang** (Nama Barang): Kolom ini berisi nama produk atau barang yang memiliki penjualan tertinggi dalam periode waktu yang dianalisis.
2. **Jumlah Penjualan** (Jumlah Penjualan): Kolom ini menunjukkan jumlah total produk yang terjual (QTYSALES) untuk produk yang tercantum dalam kolom "Nama Barang." Jumlah penjualan diukur dalam unit barang yang terjual.
3. **Tanggal Paling Banyak Penjualan** (Tanggal Paling Banyak Penjualan): Kolom ini mencantumkan tanggal (hari) ketika penjualan produk tersebut mencapai jumlah tertinggi. Ini adalah tanggal ketika produk tersebut paling laris terjual dalam periode waktu yang dianalisis.
4. **Bulan Paling Banyak Penjualan** (Bulan Paling Banyak Penjualan): Kolom ini mencantumkan bulan ketika penjualan produk tersebut mencapai jumlah tertinggi. Ini adalah bulan ketika produk tersebut paling laris terjual dalam periode waktu yang dianalisis.
5. **Hasil Barang Penjualan (SALES) paling Tinggi** : MAXITON SOFT CAP (BTL/30S), dengan Jumlah Penjualan 3,523 Barang. Kemudian Tanggal Paling Banyak Penjualannya pada Tanggal **30**, dan Bulan Paling Banyak Penjualannya pada Bulan **4**.

```
# Konversi 'TGL_INVC' menjadi format tanggal yang dapat diolah
data['TGL_INVC'] = pd.to_datetime(data['TGL_INVC'], format='%d-%m-%Y')

# Ekstraksi komponen tanggal dari kolom 'TGL_INVC'
data['TGL_DAY'] = data['TGL_INVC'].dt.day
data['TGL_MONTH'] = data['TGL_INVC'].dt.month
data['TGL_YEAR'] = data['TGL_INVC'].dt.year

# Hapus kolom 'TGL_INVC' yang tidak diperlukan lagi
data.drop(columns=['TGL_INVC'], inplace=True)

# Sidebar
st.sidebar.title("Filter")
filter_sales_return = st.sidebar.selectbox("Tampilkan SALES atau RETUR?",
["SALES", "RETUR"])

# Filter data SALES
```

```

sales_data = data[data['TRS_TYPE'] == "SALES"]

# Filter data RETUR
retur_data = data[data['TRS_TYPE'] == "RETUR"]

# Filter data berdasarkan pilihan SALES atau RETUR
filtered_data = data[data['TRS_TYPE'] == filter_sales_return]

# Menghitung jumlah QTYSALES atau RETUR untuk setiap barang
if filter_sales_return == "SALES":
    sales_return_by_item =
filtered_data.groupby('NAMABARANG')['QTYSALES'].sum().reset_index()
    sales_return_by_item.rename(columns={'QTYSALES': 'JUMLAH BARANG'},
inplace=True)
elif filter_sales_return == "RETUR":
    sales_return_by_item =
filtered_data.groupby('NAMABARANG')['QTYSALES'].sum().reset_index()
    sales_return_by_item.rename(columns={'QTYSALES': 'QTYRETUR'}, inplace=True)

# Menampilkan nama barang yang banyak di SALES jika filter adalah SALES
most_item_name = ""
most_item_qty = 0
if filter_sales_return == "SALES":
    most_item = sales_return_by_item.nlargest(1, 'JUMLAH BARANG')
    most_item_name = most_item['NAMABARANG'].values[0]
    most_item_qty = most_item['JUMLAH BARANG'].values[0]

# Menampilkan nama barang yang banyak di RETUR jika filter adalah RETUR
if filter_sales_return == "RETUR":
    most_item = sales_return_by_item.nlargest(1, 'QTYRETUR')
    most_item_name = most_item['NAMABARANG'].values[0]
    most_item_qty = most_item['QTYRETUR'].values[0]

# Menambahkan kolom "Jumlah QTYSALES" pada Data Tabel SALES
if filter_sales_return == "SALES":
    sales_return_by_item =
filtered_data.groupby('NAMABARANG')['QTYSALES'].sum().reset_index()
    sales_return_by_item.rename(columns={'QTYSALES': 'Jumlah QTYSALES'},
inplace=True)
elif filter_sales_return == "RETUR":
    # Menambahkan kolom "Jumlah QTYSALES (dengan minus)" pada Data Tabel RETUR
    sales_return_by_item =
filtered_data.groupby('NAMABARANG')['QTYSALES'].sum().reset_index()
    sales_return_by_item.rename(columns={'QTYSALES': 'QTYRETUR'}, inplace=True)

```

```

    sales_return_by_item['QTYRETUR'] = -sales_return_by_item['QTYRETUR'] #
    Tambahkan tanda minus

# Split data untuk melatih model Random Forest
X = filtered_data[['TGL_DAY', 'TGL_MONTH', 'TGL_YEAR', 'NAMABARANG',
'NAMAOUTLET']]
y = filtered_data['NAMABARANG']

# Buat objek LabelEncoder untuk NAMABARANG dan NAMAOUTLET
label_encoder = LabelEncoder()
X['NAMABARANG'] = label_encoder.fit_transform(X['NAMABARANG'])
X['NAMAOUTLET'] = label_encoder.fit_transform(X['NAMAOUTLET'])

# Tampilkan tanggal dan bulan paling banyak di SALES dan RETUR jika ada data
most_common_sales_date = filtered_data[filtered_data['TRS_TYPE'] ==
'SALES']['TGL_DAY'].mode()
most_common_sales_month = filtered_data[filtered_data['TRS_TYPE'] ==
'SALES']['TGL_MONTH'].mode()
most_common_return_date = filtered_data[filtered_data['TRS_TYPE'] ==
'RETUR']['TGL_DAY'].mode()
most_common_return_month = filtered_data[filtered_data['TRS_TYPE'] ==
'RETUR']['TGL_MONTH'].mode()

# Menghitung jumlah QTYSALES pada SALES untuk setiap barang
sales_qty_by_item =
sales_data.groupby('NAMABARANG')['QTYSALES'].sum().reset_index()
sales_qty_by_item.rename(columns={'QTYSALES': 'Jumlah Penjualan'}, inplace=True)

# Menghitung jumlah QTYSALES pada RETUR untuk setiap barang dengan tanda minus
retur_qty_by_item =
retur_data.groupby('NAMABARANG')['QTYSALES'].sum().reset_index()
retur_qty_by_item.rename(columns={'QTYSALES': 'Jumlah Penjualan (dengan minus)'},
inplace=True)
retur_qty_by_item['Jumlah Penjualan (dengan minus)'] = -retur_qty_by_item['Jumlah
Penjualan (dengan minus)']

# Gabungkan data SALES dan RETUR
combined_data = pd.concat([sales_qty_by_item, retur_qty_by_item],
ignore_index=True)

# Jumlahkan QTYSALES untuk SALES dengan NAMABARANG yang sama
sales_result = sales_data.groupby("NAMABARANG")["QTYSALES"].sum().reset_index()
sales_result.rename(columns={'NAMABARANG': 'Nama Barang', 'QTYSALES': 'Jumlah
Penjualan'}, inplace=True)

```

```

# Jumlahkan QTYSALES untuk RETUR dengan NAMABARANG yang sama
retur_result = retur_data.groupby("NAMABARANG")["QTYSALES"].sum().reset_index()
retur_result.rename(columns={'NAMABARANG': 'Nama Barang', 'QTYSALES': 'Jumlah Pengembalian'}, inplace=True)

# Temukan hasil maksimum pada SALES
max_sales = sales_result[sales_result["Jumlah Penjualan"] == sales_result["Jumlah Penjualan"].max()]

# Temukan hasil minimum pada RETUR (dengan menggunakan tanda minus)
min_retur = retur_result[retur_result["Jumlah Pengembalian"] == retur_result["Jumlah Pengembalian"].min()]

# Gabungkan kolom 'TGL_DAY' dan 'TGL_MONTH' pada tabel max_sales
max_sales['Tanggal Paling Banyak Penjualan'] = most_common_sales_date.iloc[0] if not most_common_sales_date.empty else 'Tidak ada data'
max_sales['Bulan Paling Banyak Penjualan'] = most_common_sales_month.iloc[0] if not most_common_sales_month.empty else 'Tidak ada data'

# Gabungkan kolom 'TGL_DAY' dan 'TGL_MONTH' pada tabel min_retur
min_retur['Tanggal Paling Banyak Pengembalian'] = most_common_return_date.iloc[0] if not most_common_return_date.empty else 'Tidak ada data'
min_retur['Bulan Paling Banyak Pengembalian'] = most_common_return_month.iloc[0] if not most_common_return_month.empty else 'Tidak ada data'

# Tampilkan hasil dalam aplikasi Streamlit
if filter_sales_return == "SALES":
    st.title("Analisis Perhitungan Barang Terjual (Sales) dan Barang Pengembalian (Retur) Menggunakan Algoritma Random Forest")
    st.subheader("Hasil Barang Penjualan (Sales) Paling Tinggi :")
    st.write(max_sales)
    st.subheader("Data dan Jumlah Penjualan (Sales) Berdasarkan Kategori Barang:")
    st.write(sales_result)
elif filter_sales_return == "RETUR":
    st.title("Analisis Perhitungan Barang Terjual (Sales) dan Barang Pengembalian (Retur) Menggunakan Algoritma Random Forest")
    st.subheader("Hasil Barang Pengembalian (Retur) Paling Tinggi:")
    st.write(min_retur)
    st.subheader("Data dan Jumlah Pengembalian (Retur) Berdasarkan Kategori Barang:")
    st.write(retur_result)

# Rename nama kolom data tabel
filtered_data.rename(columns={'TRS_TYPE': 'Type', 'NAMAOUTLET': 'Nama Outlet', 'NAMABARANG': 'Nama Barang',

```

```

                                'QTYSALES': 'Jumlah', 'TGL_DAY': 'Tanggal/Hari',
                                'TGL_MONTH': 'Bulan', 'TGL_YEAR': 'Tahun'}},

inplace=True)

# Tampilkan tabel data
st.subheader("Data Tabel:")
st.write(filtered_data)

```

Penjelasan Program Diatas:

1. Konversi Tanggal: Tanggal dalam kolom 'TGL_INVC' dikonversi menjadi format yang dapat diolah oleh Python, yaitu format tanggal 'dd-mm-yyyy'. Selanjutnya, komponen tanggal (hari, bulan, tahun) diekstraksi dari kolom 'TGL_INVC'.
2. Hapus Kolom Tanggal: Kolom 'TGL_INVC' yang sudah diolah lebih lanjut dihapus karena tidak diperlukan lagi.
3. Sidebar Filter: Dibuat sidebar di aplikasi Streamlit untuk memungkinkan pengguna memilih apakah ingin melihat data SALES atau RETUR.
4. Filter Data SALES dan RETUR: Data dibagi menjadi dua, yaitu `sales_data` untuk data SALES dan `retur_data` untuk data RETUR. Selain itu, data juga difilter sesuai pilihan pengguna dengan variabel `filtered_data`.
5. Perhitungan Jumlah Barang: Jumlah QTYSALES atau RETUR dihitung untuk setiap barang sesuai dengan filter yang dipilih oleh pengguna. Hasil perhitungan ini disimpan dalam `sales_return_by_item` dengan mengganti nama kolom menjadi 'JUMLAH BARANG' atau 'QTYRETUR' sesuai dengan filter.
6. Nama Barang Terbanyak: Menampilkan nama barang yang paling banyak terjual (SALES) atau paling banyak dikembalikan (RETUR) sesuai dengan filter yang dipilih oleh pengguna.
7. Kolom Jumlah QTYSALES: Menambahkan kolom "Jumlah QTYSALES" pada Data Tabel SALES dan mengganti nama kolom jika filter adalah RETUR. Selain itu, tanda minus juga ditambahkan pada jumlah RETUR.
8. Persiapan Data untuk Model: Data dibagi menjadi fitur (X) dan target (y) untuk melatih model Random Forest. Objek `LabelEncoder` digunakan untuk mengkodekan kolom-kolom kategori.
9. Tanggal dan Bulan Terbanyak: Menghitung tanggal dan bulan paling banyak dalam SALES dan RETUR jika ada data yang sesuai.
10. Perhitungan Jumlah Penjualan: Menghitung jumlah QTYSALES pada SALES untuk setiap barang. Hasilnya disimpan dalam `sales_qty_by_item`.
11. Perhitungan Jumlah Pengembalian: Menghitung jumlah QTYSALES pada RETUR untuk setiap barang dengan tanda minus. Hasilnya disimpan dalam `retur_qty_by_item`.
12. Gabungkan Data SALES dan RETUR: Menggabungkan data SALES dan RETUR ke dalam satu dataframe yang disebut `combined_data`.
13. Perhitungan Jumlah Penjualan dan Pengembalian: Menghitung jumlah QTYSALES untuk SALES dan RETUR dengan mengelompokkannya berdasarkan NAMABARANG. Hasilnya disimpan dalam dataframe `sales_result` dan `retur_result`.
14. Temuan Maksimum dan Minimum: Menemukan hasil maksimum pada SALES (barang terbanyak terjual) dan hasil minimum pada RETUR (barang paling sedikit dikembalikan). Selain itu, tanggal dan bulan paling banyak juga dicatat.
15. Tampilan Hasil: Hasil analisis ditampilkan dalam aplikasi Streamlit. Jika filter adalah SALES, informasi tentang barang terbanyak terjual dan tabel data penjualan SALES ditampilkan. Jika filter

adalah RETUR, informasi tentang barang paling sedikit dikembalikan dan tabel data pengembalian RETUR ditampilkan.

16. Renaming Kolom: Nama kolom dalam data tabel diubah sesuai dengan yang lebih deskriptif.
17. Tampilan Data Tabel: Data tabel yang sudah diolah ditampilkan dalam aplikasi.

Tabel Data dan Jumlah Penjualan (SALES) berdasarkan Kategori Barang

Tabel "Data dan Jumlah Penjualan (Sales) Berdasarkan Kategori Barang" merupakan tabel yang menampilkan informasi tentang penjualan berdasarkan kategori barang (NAMABARANG) pada data SALES. Tabel ini berisi beberapa kolom sebagai berikut:

1. **Nama Barang** (Nama Barang): Kolom ini berisi nama barang atau produk yang terjual dalam transaksi SALES.
2. **Jumlah Penjualan** (Jumlah Penjualan): Kolom ini menunjukkan jumlah total produk yang terjual (QTYSALES) untuk setiap barang atau produk. Jumlah penjualan diukur dalam unit barang yang terjual.

Tabel ini membantu dalam melihat performa penjualan dari berbagai barang atau produk yang terdapat dalam data SALES. Dengan melihat tabel ini, dapat mengidentifikasi produk mana yang paling laku terjual berdasarkan jumlah penjualannya dalam periode tertentu.

Data Tabel Sales:

Penjelasan Tabel diatas:

Tabel "Data Sales" adalah tabel yang berisi data penjualan dari jenis transaksi "Sales." Berikut adalah penjelasan untuk setiap kolom dalam tabel ini:

1. **Type** (Jenis Transaksi): Kolom ini menyatakan jenis transaksi, yang dalam konteks ini adalah "Sales." Ini adalah kategori yang membedakan transaksi penjualan dari barang-barang dengan jenis transaksi lainnya, seperti "Retur."
2. **Nama Outlet**: Kolom ini berisi informasi tentang nama outlet atau lokasi di mana transaksi penjualan dilakukan. Outlet ini adalah tempat di mana pelanggan membeli produk.
3. **Nama Barang**: Kolom ini mencantumkan nama produk atau barang yang dijual dalam transaksi. Ini adalah item atau produk yang dibeli oleh pelanggan.
4. **Jumlah Terjual**: Kolom ini menunjukkan jumlah produk yang terjual dalam transaksi tersebut. Jumlah ini diukur dalam unit barang yang terjual, dan jumlahnya mencerminkan kuantitas produk yang dibeli oleh pelanggan.
5. **Tanggal**: Kolom ini berisi tanggal transaksi penjualan. Ini adalah tanggal ketika transaksi terjadi.
6. **Bulan**: Kolom ini mencantumkan bulan ketika transaksi penjualan dilakukan. Ini membantu dalam memahami pola penjualan selama bulan-bulan tertentu.

7. **Tahun:** Kolom ini menunjukkan tahun ketika transaksi penjualan berlangsung. Ini membantu dalam menganalisis tren penjualan dari tahun ke tahun.

Jumlah Total Data Sales : 23199 Data

Tabel ini memberikan gambaran rinci tentang setiap transaksi penjualan, termasuk informasi tentang produk yang terjual, tanggal dan lokasi transaksi, dan jumlah yang dibeli. Data ini berguna untuk menganalisis performa penjualan secara lebih mendalam dan mengidentifikasi tren penjualan seiring waktu.

Diagram Jumlah Penjualan Barang (SALES) berdasarkan Kategori Barang

Grafik Akurasi Model Random Forest :

Akurasi Model Random Forest adalah ukuran sejauh mana model Random Forest yang telah dilatih dapat memprediksi dengan benar kelas atau label dari data uji yang digunakan untuk menguji model. Akurasi ini diukur dalam persentase dan menggambarkan seberapa akurat model dalam memprediksi kelas yang benar.

Dalam kode yang telah diberikan, telah dilatih model Random Forest menggunakan data pelatihan (X_{train}) dan labelnya (y_{train}). Kemudian, model tersebut diuji menggunakan data uji (X_{test}) dan hasil prediksi dari model dibandingkan dengan label yang sebenarnya (y_{test}). Akurasi dihitung dengan membandingkan berapa banyak prediksi yang benar dengan total jumlah prediksi.

Grafik Akurasi Model Random Forest yang ditampilkan dalam aplikasi Streamlit adalah grafik batang yang menunjukkan akurasi model untuk kategori "SALES" dan "RETUR". Grafik ini membantu untuk membandingkan seberapa baik model Random Forest dalam memprediksi kedua kategori tersebut.

Grafik tersebut menggunakan Plotly Express untuk membuat visualisasi yang interaktif. Dalam grafik ini, sumbu x menunjukkan kategori (SALES atau RETUR), sedangkan sumbu y menunjukkan akurasi dalam bentuk persentase. Grafik ini dapat memberikan pemahaman yang lebih baik tentang seberapa baik model berkinerja dalam memprediksi kategori SALES dan RETUR, sehingga dapat menilai kinerja model dengan lebih baik. Dalam contoh kode yang telah diberikan, model Random Forest memiliki akurasi sekitar 93.91% untuk SALES dan 90.48% untuk RETUR.

HASIL PROGRAM RETUR

Tabel "Hasil Barang Pengembalian (Retur) Paling Tinggi" adalah tabel yang memberikan informasi tentang barang atau produk yang memiliki jumlah pengembalian (RETUR) paling tinggi dalam data RETUR. Tabel ini berisi beberapa kolom sebagai berikut:

1. **Nama Barang (Nama Barang):** Kolom ini berisi nama barang atau produk yang mengalami pengembalian dalam transaksi RETUR.
2. **Jumlah Pengembalian (Jumlah Pengembalian):** Kolom ini menunjukkan jumlah total produk yang dikembalikan (QTYSALES dengan tanda minus) untuk setiap barang atau produk dalam data RETUR. Jumlah pengembalian diukur dalam unit produk yang dikembalikan.
3. **Tanggal Paling Banyak Pengembalian (Tanggal Paling Banyak Pengembalian):** Kolom ini mencantumkan tanggal (hari) yang paling sering terjadi pengembalian barang untuk produk tersebut. Data ini memberikan informasi tentang tanggal di mana pengembalian paling sering terjadi.
4. **Bulan Paling Banyak Pengembalian (Bulan Paling Banyak Pengembalian):** Kolom ini mencantumkan bulan yang paling sering terjadi pengembalian barang untuk produk tersebut. Data ini memberikan informasi tentang bulan di mana pengembalian paling sering terjadi.
5. **Hasil Barang Pengembalian (RETUR) paling Tinggi :** CHAMPS MULTIVITAMIN PINNEAPPLE (BTL/30), dengan Jumlah Pengembalian -232 Barang. Kemudian Tanggal Paling Banyak Pengembalian pada Tanggal 29, dan Bulan Paling Banyak Pengembalian pada Bulan 3.

Tabel ini membantu dalam melihat produk mana yang mengalami tingkat pengembalian paling tinggi. Informasi ini dapat digunakan untuk mengidentifikasi produk yang memiliki masalah atau memerlukan perbaikan, serta mengelola stok produk yang cenderung mengalami pengembalian.

Data dan Jumlah Pengembalian (RETUR) Berdasarkan Kategori Barang:

Data Tabel RETUR :

Tabel "Data Retur" memberikan informasi terkait dengan transaksi RETUR dalam data yang sedang dianalisis. Tabel ini terdiri dari beberapa kolom sebagai berikut:

1. Type (Jenis): Kolom ini menunjukkan jenis transaksi, yang dalam hal ini adalah "Retur." Ini mengindikasikan bahwa transaksi tersebut adalah pengembalian produk.
2. Nama Outlet: Kolom ini mencantumkan nama toko atau outlet tempat transaksi RETUR dilakukan. Informasi ini membantu dalam mengidentifikasi toko atau outlet mana yang paling sering menerima pengembalian produk.
3. Nama Barang: Kolom ini berisi nama produk atau barang yang dikembalikan oleh pelanggan. Ini adalah barang yang tidak memenuhi harapan pelanggan dan harus dikembalikan.
4. Jumlah Terjual (Jumlah Terjual): Kolom ini mencantumkan jumlah produk yang dikembalikan dalam transaksi RETUR. Jumlah ini diukur dalam unit produk yang dikembalikan.
5. Tanggal: Kolom ini mencantumkan tanggal ketika transaksi RETUR dilakukan. Ini adalah tanggal ketika produk dikembalikan oleh pelanggan.
6. Bulan: Kolom ini mencantumkan bulan ketika transaksi RETUR dilakukan. Ini membantu dalam melihat pola pengembalian produk selama beberapa bulan.
7. Tahun: Kolom ini mencantumkan tahun ketika transaksi RETUR dilakukan. Ini membantu dalam melacak tren pengembalian produk dari tahun ke tahun.

Jumlah Total Data Retur: -1052 Data

Tabel "Data Retur" memberikan gambaran tentang transaksi RETUR yang terjadi, termasuk produk apa yang dikembalikan, dari mana produk tersebut dikembalikan, berapa jumlah produk yang dikembalikan, dan kapan transaksi RETUR tersebut terjadi. Informasi ini penting untuk mengelola pengembalian produk dan memastikan kepuasan pelanggan.

Diagram Pengembalian (RETUR) Barang berdasarkan Kategori :

Akurasi Mode Random Forest :

Screenshoot Aplikasi :

Menu Sales

Screenshot RETUR :

