BAB 4
HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Penelitian ini dilakukan untuk mengetahui apakah ada pola hubungan antar kategori buku yang sering dipinjam bersamaan di perpustakaan. Untuk itu, digunakan pendetakatan *data mining* melalui metode *association rules mining* dengan pemanfaatan algoritma *Frequent Pattern Growth* (FP-Growth). Dari total 11.220 transaksi peminjaman, hanya 38% atau 4.306 transaksi yang mencakup peminjaman dua kategori buku atau lebih. Transaksi ini menjadi dasar pembentukan pola asosiasi, sementara 62% sisanya merupakan transaksi tunggal yang tidak dapat digunakan dalam pembentukan pola asosiasi.

Nilai *minimum support* yang ditetapkan sebesar 0,00675 dihitung menggunakan pendekatan *geometric mean* dari frekuensi masing-masing kategori buku. Hasil penerapan algoritma FP-Growth dengan nilai ambang tersebut menghasilkan 89 *frequent itemset* yang digunakan untuk membentuk 90 aturan asosiasi. Evaluasi dilakukan menggunakan tiga metrik, yaitu *lift*, *leverage*, dan *conviction*. Hasil evaluasi menunjukkan bahwa hanya sebagian kecil aturan yang menunjukkan kekuatan asosiasi yang tinggi, yaitu:

- 22 aturan (24,4%) memiliki nilai *lift* > 1
- 25 aturan (27,8%) menunjukkan *leverage* positif
- 22 aturan (24,4%) memiliki *conviction* > 1

Untuk memastikan validitas secara statistik, dilakukan uji *Fisher Exact Test* dengan koreksi *False Discovery Rate* menggunakan metode *Benjamini-Hochberg*. Berdasarkan pengujian tersebut, diperoleh 50 aturan yang signifikan secara statistik. Dari hasil evaluasi dan validasi, hanya delapan aturan yang memenuhi seluruh kriteria evaluasi dan signifikan secara statistik *(p-value < 0,05).* Beberapa aturan yang lolos seleksi antara lain:

- Jika seseorang meminjam buku *Sastra Lain*, terdapat kemungkinan sebesar 54% bahwa ia juga akan meminjam *Sastra Amerika*.
- Jika seseorang meminjam *Ilmu Politik*, kemungkinan 40% ia juga meminjam *Agama-agama Lain*.
- Jika seseorang meminjam *Matematika*, kemungkinan 16% ia juga meminjam *Pendidikan*.

Aturan-aturan tersebut mencerminkan kebiasaan pemustaka dalam meminjam buku, sehingga dapat dimanfaatkan dalam pengembangan sistem rekomendasi buku, membantu dalam penempatan koleksi di rak, serta perencanaan pengadaan koleksi yang lebih sesuai dengan kebutuhan pemustaka.

4.2 TAHAP PEMBERSIHAN DATA

Langkah awal dalam proses analisis adalah pembersihan data. Data peminjaman buku yang diperoleh mengandung informasi tambahan seperti baris kosong dan kolom yang tidak relevan. Oleh karena itu, dilakukan proses pembersihan untuk memastikan hanya data yang relevan yang akan digunakan dalam analisis lebih lanjut. Kode program pembersihan data disusun dalam bentuk fungsi bernama `read_data()`. Fungsi ini digunakan untuk membaca *file spreadsheet,* mendeteksi baris *header* yang valid, serta menyaring hanya kolom-kolom yang relevan untuk proses analisis.

```
import pandas as pd

def read_data(file_path):
    raw_data = pd.read_excel(file_path, header=None)
    header_row_index = None

    for index, row in raw_data.iterrows():
        if 'No. Anggota' in row.values:
            header_row_index = index
            break

 if header_row_index is None:
        raise ValueError("Baris Header tidak ditemukan dalam data.")

    cleaned_data = pd.read_excel(file_path, skiprows=header_row_index)
    cleaned_data = cleaned_data
        .loc[:, ~cleaned_data.columns.str.contains('^Unnamed')]
```

```
    required_columns = [
        'No. Anggota',
        'Judul Buku',
        'Nomor Klasifikasi',
        'Tgl. Pinjam',
    ]
    cleaned_data = cleaned_data
        .loc[:,cleaned_data.columns.isin(required_columns)]
    return cleaned_data
```

Setelah fungsi `read_data()` dijalankan pada *dataset*, diperoleh data yang bersih dengan struktur seperti pada Tabel 4.1 berikut:

Tabel 4.1 Contoh Data Setelah Pembersihan

No. Anggota	Judul Buku	Nomor Klasifikasi	Tgl. Pinjam
3328112803800002	Amerika dan Dunia Memperdebatkan Bentuk Baru P...	327 AME	2025-06-16
3328186007070004	Menulis Sebagai Suatu Keterampilan Berbahasa	411 TAR m c.1	2025-05-26
3328101105740002	Binatang Pertamaku	590 AJI b c.3	2025-05-17
3175032203770008	Kecupan	741.5 MAN k	2025-05-16
2023015	Pelajar Teladan	808.06 Sug p c.1	2025-05-06

Tabel diatas menyajikan contoh data setelah dilakukan pembersihan. Setiap baris berisi informasi penting yang diperlukan, yaitu no. anggota, judul buku, nomor klasifikasi, dan tanggal peminjaman. Informasi yang tidak relevan telah dieliminasi agar siap digunakan dalam proses selanjutnya.

4.3 INTEGRASI DATA

Tahap integrasi data dilakukan untuk menggabungkan informasi peminjaman buku dengan klasifikasi yang sesuai berdasarkan sistem *Dewey Decimal Classification* (DDC). Tujuan dari tahap ini adalah menyelaraskan setiap entri peminjaman dengan klasifikasi bukunya.

4.3.1 Pembacaan Data Kategori Buku

Langkah awal dalam proses integrasi adalah membaca data klasifikasi buku. Data ini berisi informasi mengenai kode klasifikasi dan deskripsi kategori yang mengacu pada sistem klasifikasi perpustakaan, yaitu *Dewey Decimal Classification* (DDC). Berikut ini merupakan fungsi yang digunakan untuk membaca dan menyesuaikan format data kategori:

```
def read_classification(file_path):
    df = pd.read_excel(file_path)
    df = df[["Kode", "Kategori"]]

    df["Kode"] = df["Kode"].astype(str).str.zfill(3)

    df = df.rename(columns={
        "Kode": "Kode Klasifikasi",
        "Kategori": "Kategori Buku"
    })
    return df
```

Fungsi `read_classification()` akan membaca data kategori yang telah diunggah oleh pengguna, kemudian mengatur kolom kode agar terdiri dari tiga digit dengan menambahkan nol di depan jika diperlukan (misalnya 20 menjadi 020). Kolom juga dinamai ulang agar sesuai dengan format yang akan digunakan. Struktur data hasil dari fungsi ini dapat dilihat pada tabel dibawah ini:

Tabel 4.2 Stuktur Data Kategori Buku

Kode Klasifikasi	Kategori Buku
320	Ilmu Politik
410	Linguistik
590	Zoologi
740	Desain dan Gambar
800	Sastra

4.3.2 Ekstraksi Kode Klasifikasi dari Data Peminjaman

Nomor Klasifikasi yang terdapat pada data transaksi peminjaman hasil pembersihan sebenarnya sudah menunjukkan kode klasifikasi buku sesuai dengan sistem *Dewey Decimal Classification* (DDC). Namun, data tersebut belum

menyertakan deskripsi kategori secara jelas. Agar informasi ini dapat digunakan dalam analisis, diperlukan proses ekstraksi untuk mengambil bagian kode numerik dari setiap data pada kolom *Nomor Klasifikasi*.

Kode tersebut kemudian dicocokkan dengan data klasifikasi yang berisi deskripsi dari masing-masing kategori. Proses ekstraksi ini dilakukan menggunakan fungsi `extract_classification()` sebagai berikut:

```
import re

def extract_classification(row):
    teks = str(row["Nomor Klasifikasi"]).strip().lower()

    if re.search(r'fik', teks, re.IGNORECASE):
        return "fik"

    match = re.findall(r'\d{2,3}', teks)

    if match:
        return match[0].zfill(3)
    else:
        return None
```

Fungsi ini bekerja dengan mendeteksi angka dua hingga tiga digit pada nilai *Nomor Klasifikasi*, yang biasanya terletak di awal dan menunjukkan kode klasifikasi DDC. Angka tersebut kemudian dibulatkan ke bawah ke kelipatan sepuluh untuk menyederhanakan kategorisasi, lalu dikembalikan dalam format tiga digit (misalnya 813 menjadi 810). Selain itu, apabila ditemukan kata *"fik"* dalam *Nomor Klasifikasi*, maka entri tersebut dikategorikan sebagai *"fik",* yang merupakan penanda untuk koleksi fiksi atau *non*-DDC.

4.3.3 Penggabungan Data

Setelah data kode klasifikasi berhasil diekstraksi dari kolom *Nomor Klasifikasi* dan data kategori buku telah disiapkan, tahap selanjutnya adalah melakukan proses penggabungan kedua data tersebut. Tujuannya adalah agar setiap entri peminjaman buku tidak hanya memiliki kode klasifikasi numerik, tetapi juga dilengkapi dengan deskripsi kategori berdasarkan sistem *Dewey Decimal Classification* (DDC). Proses integrasi ini dilakukan menggunakan fungsi `decode_classification()`, yang menggabungkan data transaksi peminjaman

dengan data klasifikasi kategori menggunakan metode *merge* berdasarkan kolom *"Kode Klasifikasi"*. Berikut adalah implementasi fungsinya:

```
def decode_classification(df_book, df_classification):
    df_result = df_book.copy()

    # Ekstraksi kode klasifikasi
    df_result["Kode Klasifikasi"] = df_result.apply(
        extract_classification, axis=1)

    df_result = df_result[df_result["Kode Klasifikasi"].notna()]
        .reset_index(drop=True)

    df_result = df_result.merge(
        df_classification, on="Kode Klasifikasi", how="left")

    # Tangani klasifikasi khusus 'fik'
    df_result["Kategori Buku"] = df_result.apply(
        lambda row: "Fiksi" if
            row["Kode Klasifikasi"].lower() == "fik" else
            row["Kategori Buku"], axis=1
        )
    return df_result
```

Fungsi di atas terdiri atas beberapa langkah utama:

1. Menyalin data transaksi ke dalam variabel baru sebagai dasar pemrosesan.
2. Menjalankan proses ekstraksi kode klasifikasi dari kolom *Nomor Klasifikasi* menggunakan fungsi `extract_classification()`.
3. Membuang data yang tidak memiliki hasil ekstraksi valid.
4. Melakukan penggabungan (*merge*) dengan data klasifikasi, agar setiap kode dapat dihubungkan dengan deskripsi kategori DDC yang sesuai.
5. Melakukan penangangan jika ditemukan kode *"fik",* maka akan dikategorikan sebagai *"Fiksi",* karena klasifikasi ini tidak termasuk dalam sistem DDC.

Setelah fungsi `decode_classification()` dijalankan, data yang telah dibersihakan akan memiliki dua atribut tambahan, yaitu *"Kode Klasifikasi"* dan *"Kategori Buku"*. Untuk menjaga akurasi dan konsistensi data, entri yang tidak memiliki kecocokan dengan klasifikasi yang tersedia akan dikeluarkan dari hasil akhir. Contoh hasil akhir dari proses integrasi ditampilkan pada tabel berikut:

Tabel 4.3 Tabel Contoh Data Setelah Integrasi Kategori Buku

No. Anggota	Judul Buku	Nomor Klasifikasi	Tgl. Pinjam	Kode Klasifikasi	Kategori Buku
3328112803800002	Amerika dan Dunia Memperdebatkan Bentuk Baru P...	327 AME	2025-06-16	320	Ilmu Politik
3328186007070004	Menulis Sebagai Suatu Keterampilan Berbahasa	411 TAR m c.1	2025-05-26	410	Linguistik
3328101105740002	Binatang Pertamaku	590 AJI b c.3	2025-05-17	590	Zoologi
3175032203770008	Kecupan	741.5 MAN k	2025-05-16	740	Desain dan Gambar
2023015	Pelajar Teladan	808.06 Sug p c.1	2025-05-06	800	Sastra

4.4 TAHAP TRANSFORMASI DATA

Setelah data peminjaman buku berhasil melalui tahap pembersihan dan integrasi dengan data klasifikasi, proses berikutnya adalah transformasi data. Transformasi dilakukan dengan mengelompokkan data peminjaman berdasarkan kombinasi unik antara nomor anggota dan tanggal peminjaman. Setiap kombinasi ini dianggap sebagai satu transaksi, sehingga dalam satu baris data dapat mencakup semua kategori buku yang dipinjam dalam kunjungan tersebut.

Untuk menjaga keunikan data, jika dalam satu transaksi terdapat beberapa buku dari kategori yang sama, maka kategori tersebut hanya dicatat satu kali. Transformasi ini diimplementasikan melalui fungsi `transform_data()` berikut:

```
def transform_data(df_book):
  df = df_book.copy()
  df['Tgl. Pinjam']=pd.to_datetime(df['Tgl. Pinjam'], errors='coerce')

  grouped_data = df.groupby(['No. Anggota', 'Tgl. Pinjam']).agg({
    'Judul Buku': lambda x: list(x),
    'Kategori Buku': lambda x: list(dict.fromkeys(x.dropna())),
  }).reset_index().rename(
     columns={
       'Judul Buku': 'Daftar Buku',
       'Kategori Buku': 'Daftar Kategori Buku',
  }).sort_values(by='Tgl. Pinjam', ascending=False).reset_index(
     drop=True)
  return grouped_data
```

Fungsi ini melakukan beberapa tahapan antara lain:

1. Menyalin data transaksi ke dalam variabel baru sebagai dasar pemrosesan.
2. Mengonversi kolom *"Tgl. Pinjam"* menjadi format *datetime*.
3. Mengelompokkan data berdasarkan *"No. Anggota"* dan *"Tgl. Pinjam"*.
4. Menggabungkan semua judul buku dan kategori buku dalam transaksi tersebut ke dalam format daftar (*list*).
5. Menghapus duplikasi kategori dalam satu transaksi dengan menggunakan `dict.fromkeys()` agar setiap kategori hanya tercatat satu kali.

Sebelum proses transformasi dilakukan, data peminjaman terdiri atas 23.197 baris, di mana setiap baris merepresentasikan satu buku yang dipinjam oleh seorang anggota. Hal ini menyebabkan terjadinya redundansi, karena satu transaksi dapat tercatat dalam beberapa baris terpisah jika lebih dari satu buku dipinjam. Contoh hasil data setelah proses transformasi ditampilkan pada Tabel 4.4 berikut:

Tabel 4.4 Contoh Hasil Transformasi Data Peminjaman Buku

No. Anggota	Tgl. Pinjam	Daftar Buku	Daftar Kategori Buku
3328112803800002	2025-06-16	[Amerika dan Dunia Memperdebatkan Bentuk Baru P...]	[Ilmu Politik]
3328186007070004	2025-05-26	[Menulis Sebagai Suatu Keterampilan Berbahasa]	[Linguistik]
3328101105740002	2025-05-17	[Binatang Pertamaku]	[Zoologi]
3175032203770008	2025-05-16	[Kecupan]	[Desain dan Gambar]
2023015	2025-05-06	[Pelajar Teladan]	[Sastra]

Setelah transformasi, struktur data diubah sedemikian rupa sehingga setiap baris merepresentasikan satu transaksi unik, yaitu kombinasi dari satu anggota dan satu tanggal peminjaman. Informasi buku dan kategori buku yang dipinjam dikelompokkan dalam dua kolom baru berbentuk daftar: *"Daftar Buku"* dan *"Daftar Kategori Buku"*. Hasil akhir menunjukkan bahwa terdapat 11.219 transaksi unik yang ditemukan.

4.5 SELEKSI DATA

Setelah proses transformasi selesai dilakukan, tahap selanjutnya adalah seleksi data. Tahap ini bertujuan untuk menyaring data hasil transformasi agar hanya menyisakan transaksi yang relevan untuk dianalisis lebih lanjut. Dalam tahap ini, hanya atribut *"Daftar Kategori Buku"* yang digunakan, karena atribut lain seperti *"No. Anggota", "Tgl. Pinjam",* maupun *"Daftar Buku"* tidak memberikan kontribusi langsung terhadap analisis asosiasi antar item. Atribut-atribut tersebut lebih berfungsi sebagai identitas transaksi, bukan representasi hubungan antar kategori buku. Seleksi data dilakukan dengan cara mengambil hanya kolom *"Daftar Kategori Buku" dari* hasil transformasi kemudian menyaring transaksi yang memiliki lebih dari satu kategori buku karena transaksi dengan hanya satu kategori tidak dapat membentuk asosiasi. kode program yang digunakan untuk melakukan proses seleksi data adalah sebagai berikut:

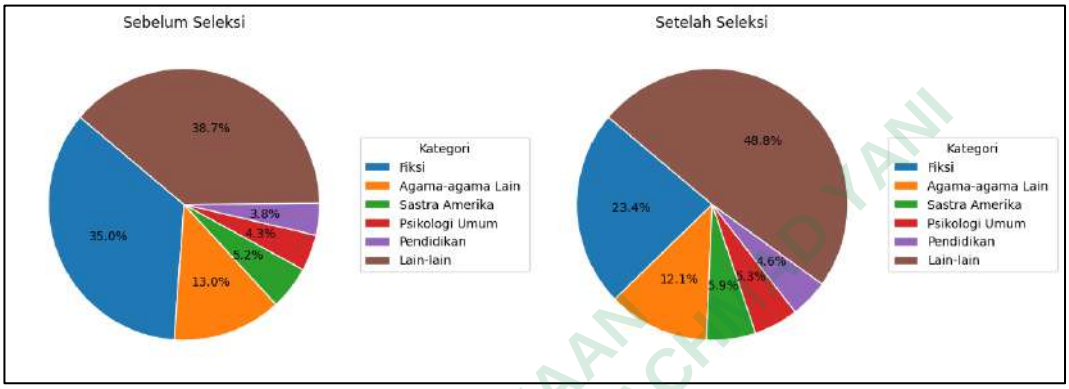
```
df_selected = df_transformed[['Daftar Kategori Buku']].copy()
df_selected = df_selected[df_selected['Daftar Kategori Buku']
   .map(len) > 1].reset_index(drop=True)
```

Setelah proses seleksi dilakukan, jumlah transaksi yang memenuhi kriteria untuk dianalisis adalah sebanyak 4.305 transaksi, dari total 11.219 transaksi hasil transformasi. Hal ini menunjukkan bahwa hanya sekitar 38% dari keseluruhan data yang telah ditransformasikan mengandung lebih dari satu kategori buku yang memenuhi syarat untuk dianalisis. Sisanya, yaitu sekitar 62%, merupakan transaksi tunggal yang tidak dapat membentuk relasi antar item, sehingga dikeluarkan dari proses analisis. Contoh hasil proses seleksi data ditampilkan pada tabel 4.5 berikut:

Tabel 4.5 Hasil Seleksi Data Peminjaman Buku

	Daftar Kategori Buku
1	[Pendidikan, Agama-agama Lain]
2	[Bahasa Inggris, Linguistik]
3	[Ilmu Sosial, Bahasa Inggris]
4	[Agama-agama Lain , Desain & Gambar]
5	[Administrasi Publik, Hukum]

Untuk memastikan bahwa proses seleksi data tidak menimbulkan bias terhadap representasi kategori buku, dilakukan perbandingan distribusi persentase kategori sebelum dan sesudah seleksi. Tujuan dari proses ini adalah untuk melihat apakah ada kategori tertentu yang mendominasi yang dapat memengaruhi hasil pola asosiasi. Perbandingan ini divisualisasikan pada Gambar 4.1 berikut:



Gambar 4.1 Persentase Kategori Buku Sebelum dan Sesudah Seleksi

Berdasarkan hasil visualisasi, terlihat adanya penurunan proporsi kategori Fiksi setelah proses seleksi data. Hal ini menunjukkan bahwa kategori Fiksi lebih sering muncul dalam transaksi tunggal. Sebaliknya, terjadi peningkatan proporsi kategori yang lainnya, serta distribusi yang lebih merata pada kategori Pendidikan, Psikologi Umum, Sastra Amerika.

4.6 PENERAPAN ALGORITMA FP – GROWTH

Setelah data melalui tahap seleksi dan hanya menyisakan transaksi yang mengandung lebih dari satu item, tahap berikutnya adalah penerapan algoritma *Frequent Pattern Growth* (FP-Growth). Penerapan algoritma dimulai dengan penentuan nilai *minimum support* sebagai ambang batas frekuensi kemunculan pola yang dianggap signifikan.

4.6.1 Menentukan *Minimum Support*

Dalam proses penambangan pola asosiasi, pemilihan nilai *minimum support* merupakan tahap yang sangat krusial karena secara langsung memengaruhi jumlah serta kualitas aturan yang dihasilkan. Nilai *support* yang terlalu tinggi berisiko mengabaikan pola-pola penting yang kurang dominan, sedangkan nilai yang terlalu

rendah justru dapat menghasilkan terlalu banyak aturan yang tidak relevan (Han et al., 2012). Untuk menghindari pendekatan yang *arbitrer* atau berbasis eksperimen manual, penentuan *nilai minimum support* mengadopsi pendekatan berbasis statistik, yaitu dengan menggunakan rata-rata geometrik dari frekuensi kemunculan item. Pendekatan ini dikembangkan berdasarkan prinsip yang dijelaskan oleh Tan et al., (2019), yang menyebutkan bahwa *geometric mean* lebih tepat digunakan sebagai ukuran tendensi sentral untuk data dengan distribusi tidak simetris (*skewed*). Nilai rata-rata geometrik dihitung dengan rumus berikut:

$$GeoMean = \sqrt[n]{f1 \cdot f2 \cdot \cdots \cdot fn} \tag{10}$$

di mana:

- $f1 \cdot f2 \cdot \cdots \cdot fn$ adalah frekuensi kemunculan masing-masing item unik dalam dataset.
- n adalah jumlah item unik.

Nilai *GeoMean* tersebut kemudian dinormalisasi terhadap jumlah total transaksi untuk memperoleh nilai *minimum support* yang proporsional terhadap ukuran dataset dengan N adalah jumlah total transaksi, sesuai rumus berikut:

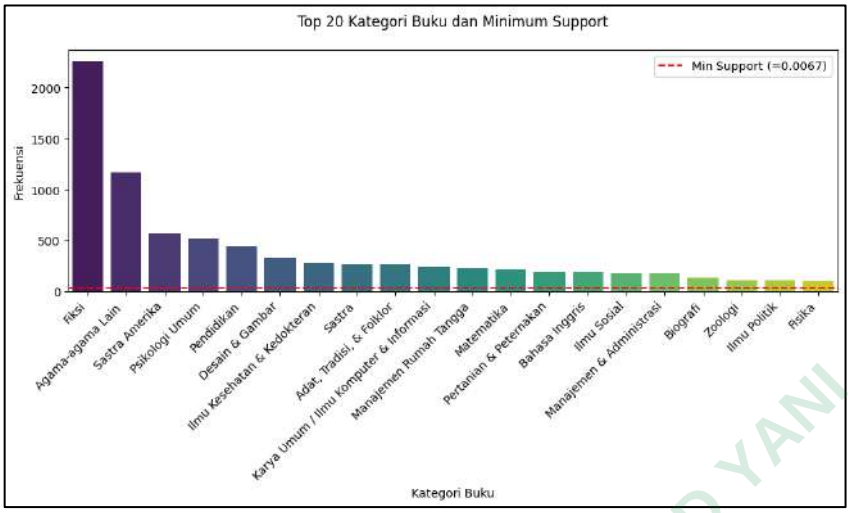
$$Minimum\ Support = \frac{GeoMean}{N} \tag{11}$$

Pendekatan ini dianggap lebih stabil dibandingkan penggunaan rata-rata aritmetika karena tidak terpengaruh secara ekstrem oleh item-item yang sangat sering atau sangat jarang muncul. Selain itu, metode ini bersifat *data-driven*, sehingga mampu menyesuaikan diri dengan karakteristik distribusi data aktual. Adapun implementasi perhitungan nilai *minimum support* dilakukan melalui kode program berikut:

```
from collections import Counter
import pandas as pd
import numpy as np

def determine_min_support_geometric(item_lists: pd.Series):
  item_counts = Counter()
  for items in item_lists.dropna():
      item_counts.update(items)

  item_freq = pd.Series(item_counts).sort_values(ascending=False)
  total_transactions = len(item_lists)

  # Filter hanya item dengan frekuensi > 0
  frequencies = item_freq[item_freq > 0]
  # Hitung rata-rata geometrik
  log_sum = np.sum(np.log(frequencies))
  geo_mean = np.exp(log_sum / len(frequencies))

  min_support = geo_mean / total_transactions

  info = {
    "Total Transaksi": total_transactions,
    "Jumlah Item Unik": len(frequencies),
    "Rata-rata Geometrik Frekuensi": round(geo_mean, 5),
    "Minimum Support (GeoMean)": round(min_support, 5)
  }

  return round(min_support, 5), item_freq, info
```

Setelah fungsi `determine_min_support_geometric()` dijalankan pada data hasil seleksi, diperoleh nilai *minimum support* sebesar 0,00675 yang berarti suatu kategori buku harus muncul setidaknya dalam 0,675% dari total 4.306 transaksi untuk dapat dikategorikan sebagai *frequent itemset*.

Nilai ambang ini bersifat adaptif terhadap struktur data, karena dihitung langsung dari rata-rata geometrik frekuensi kemunculan item. Terdapat 83 kategori buku unik yang ditemukan, dengan nilai rata-rata geometrik frekuensi kemunculannya tercatat sebesar 29,06248 kali. Distribusi frekuensi kemunculan kategori buku dalam dataset menunjukkan adanya ketimpangan (*skewed distribution*) di mana beberapa kategori sangat dominan, sedangkan sebagian besar kategori lainnya memiliki frekuensi rendah.



Gambar 4.2 Distribusi Frekuensi Kemunculan 15 Kategori Buku Teratas

Visualisasi pada Gambar 4.2 menampilkan 15 kategori buku dengan frekuensi kemunculan tertinggi dalam dataset hasil seleksi. Grafik ini memberikan gambaran terhadap dominasi item tertentu yang mungkin memengaruhi hasil akhir pola asosiasi yang ditemukan.

4.6.2 Menyiapkan Data Transaksi Untuk Algoritma FP – Growth

Setelah nilai *minimum support* berhasil ditentukan, langkah selanjutnya adalah menyiapkan data dalam format yang sesuai untuk proses penambangan pola asosiasi. Format yang dibutuhkan oleh algoritma FP-Growth adalah data dalam bentuk transaksi biner, di mana setiap transaksi direpresentasikan sebagai baris, dan setiap kategori buku direpresentasikan sebagai kolom dengan nilai biner *True* atau *False*. Untuk melakukan transformasi, digunakan modul `TransactionEncoder` yang disediakan oleh *library* `mlxtend`. Modul ini secara otomatis mengubah struktur *data list of lists menjadi one-hot encoded DataFrame*. Adapun kode program untuk melakukan transformasi data transaksi adalah sebagai berikut:

```
from mlxtend.preprocessing import TransactionEncoder

def prepare_transactions(data_series):
    te = TransactionEncoder()
    te_array = te.fit_transform(data_series)
    df_transformed = pd.DataFrame(te_array, columns=te.columns_)
    return df_transformed
```

Fungsi di atas menerima *data_series* berupa `pandas.Series` yang berisi daftar kategori buku untuk setiap transaksi. Data tersebut kemudian dikodekan ke dalam bentuk biner dengan menggunakan metode `fit_transform()` dari objek `TransactionEncoder`. Hasilnya adalah sebuah *DataFrame* di mana setiap kolom mewakili satu kategori buku, dan setiap nilai menunjukkan apakah kategori tersebut terdapat dalam transaksi yang bersangkutan.

Setelah fungsi `prepare_transactions()` dijalankan, diperoleh struktur data dalam bentuk *one-hot encoding* yang siap digunakan sebagai input dalam algoritma FP-Growth. Sebagai ilustrasi, Tabel 4.6 berikut menampilkan 10 transaksi pertama dengan 5 kategori buku pertama hasil proses encoding.

Tabel 4.6 Contoh Hasil *One-Hot Encoding* Data Kategori Buku

Adat, Tradisi, & Folklor	Administrasi Publik	Agama	Agama-agama Lain	Ajaran Kristen & Moralitas
False	False	False	True	False
False	False	False	False	False
False	False	False	False	False
False	False	False	True	False
False	True	False	False	False
True	False	False	True	False
False	False	False	True	False
False	False	False	True	False
False	False	False	False	False
True	False	False	False	False

4.6.3 Penerapan Algoritma FP-Growth

Tahap selanjutnya adalah penerapan algoritma FP-Growth untuk mengekstraksi *frequent itemsets*. *Itemset* yang dihasilkan pada tahap ini akan menjadi dasar untuk pembentukan aturan asosiasi (*association rules*) pada langkah selanjutnya. Nilai *minimum support* yang digunakan dalam proses ini adalah sebesar 0,00675. Nilai tersebut merepresentasikan ambang minimum proporsi kemunculan *itemset* dalam keseluruhan transaksi dimana suatu kombinasi kategori

buku harus muncul dalam setidaknya 0,675% dari total 4.306 transaksi agar dapat dianggap sebagai *frequent item*. Penerapan algoritma dilakukan menggunakan fungsi `fpgrowth()` dari pustaka `mlxtend.frequent_patterns`. Kode program yang digunakan adalah sebagai berikut:

```
from mlxtend.frequent_patterns import fpgrowth

frequent_itemsets = fpgrowth(
   df_prepared,
   min_support=min_support,
   use_colnames=True
)
```

Parameter `min_support` diatur berdasarkan nilai *minimum support* yang telah ditentuan sebelumnya, sementara `use_colnames=True` digunakan untuk memastikan bahwa nama-nama kategori buku tetap tampil dalam format asli, bukan dalam bentuk indeks. Setelah fungsi dijalankan, diperoleh sebanyak 89 *frequent itemsets* yang memenuhi kriteria *minimum support*. Setiap *itemset* disertai dengan nilai *support*, yaitu proporsi transaksi di mana kombinasi kategori buku tersebut ditemukan. Tabel 4.7 berikut menyajikan beberapa hasil *frequent itemsets* yang berhasil diekstraksi:

Tabel 4.7 Contoh *Frequent Itemsets* Hasil Penerapan Algoritma FP-Growth

No	***Support***	***Itemset***
1	0,27078	Agama-agama Lain
2	0,10241	Pendidikan
3	0,05016	Matematika
4	0,02368	Ilmu Politik
5	0,00859	Agama-agama Lain, Manajemen & Administrasi

Tabel tersebut menunjukkan adanya beberapa kategori buku yang cenderung dipinjam bersamaan oleh anggota perpustakaan. Misalnya, kombinasi kategori Fiksi dan Manajemen & Administrasi muncul bersama dalam sekitar 0,975% dari seluruh transaksi. Sementara itu, kategori Agama-agama Lain muncul secara mandiri dalam lebih dari 27% transaksi, menjadikannya salah satu kategori paling dominan dalam data yang dianalisis.

4.6.4 Pembentukan Aturan Asosiasi

Setelah memperoleh *frequent itemsets* dari hasil penerapan algoritma FP-Growth, tahap selanjutnya adalah pembentukan aturan asosiasi. Pembentukan aturan dilakukan menggunakan fungsi `association_rules()` dari pustaka `mlxtend.frequent_patterns` dengan parameter `metric="confidence"` dan `min_threshold=0.0`. Konfigurasi ini digunakan untuk mengekstraksi seluruh aturan yang mungkin terbentuk tanpa melakukan penyaringan awal terhadap nilai *confidence* agar seluruh potensi hubungan antar kategori buku dapat dianalisis secara menyeluruh. Kode program yang digunakan adalah berikut:

```
from mlxtend.frequent_patterns import association_rules

all_rules = association_rules(
  frequent_itemsets,
  metric="confidence",
  min_threshold=0.00
)
```

Fungsi tersebut menghasilkan sebanyak 89 aturan asosiasi dalam bentuk pasangan *antecedents* dan *consequents*. Contoh aturan tersebut dapat dilihat pada Tabel 4.8 berikut.

Tabel 4.8 Contoh *Frequent Itemsets* Hasil Algoritma FP-Growth

Antecedent	Consequent	Antecedent Supp	Consequent Support	Support	Confidence
Adat, Tradisi, & Folklor	Fiksi	0.0611	0.5248	0.0360	0.5894
Sastra	Fiksi	0.0615	0.5248	0.0337	0.5472
Sastra Lain	Sastra Amerika	0.0218	0.1321	0.0118	0.5426
Desain & Gambar	Fiksi	0.0755	0.5248	0.0395	0.5231
Hiburan, Permainan, & Olahraga	Fiksi	0.0139	0.5248	0.0072	0.5167

Salah satu metrik yang dihasilkan dalam pembentukan aturan adalah *support* yang menunjukkan proporsi transaksi yang mengandung kombinasi *antecedents* dan *consequents* secara bersamaan. Semakin tinggi nilai *support*,

semakin sering kombinasi tersebut muncul dalam data transaksi. Selanjutnya, metrik *confidence* digunakan untuk mengukur sejauh mana kemungkinan *consequents* muncul dalam sebuah transaksi apabila *antecedents* telah diketahui ada di dalam transaksi tersebut. Selain itu, terdapat pula metrik *antecedent support* dan *consequent support*. *Antecedent support* menunjukkan proporsi kemunculan *antecedents* secara mandiri dalam seluruh transaksi, sedangkan *consequent support* mengukur seberapa sering *consequents* muncul secara terpisah dalam data.

Sebagai ilustrasi, aturan pertama yang terdiri dari *Adat, Tradisi, & Folklor* sebagai *antecedent* dan *Fiksi* sebagai *consequent* memiliki nilai *support* sebesar 0,0360, yang menunjukkan bahwa kombinasi kedua kategori tersebut muncul dalam 3,60% dari seluruh transaksi. Nilai *confidence* sebesar 0,5894 mengindikasikan bahwa dari seluruh transaksi yang memuat kategori *Adat, Tradisi, & Folklor*, sekitar 58,94% di antaranya juga memuat kategori *Fiksi*. Adapun *antecedent support* sebesar 0,0611 menunjukkan bahwa kategori *Adat, Tradisi, & Folklor* secara mandiri muncul dalam 6,11% transaksi, sedangkan *consequent support* sebesar 0,5248 menunjukkan bahwa kategori *Fiksi* muncul dalam lebih dari separuh jumlah transaksi.

4.7 Evaluasi Aturan Asosiasi

Setelah seluruh aturan asosiasi berhasil dibentuk, tahap selanjutnya adalah melakukan evaluasi terhadap kualitas aturan-aturan tersebut. Evaluasi ini dilakukan dengan menggunakan empat metrik, yaitu *lift, leverage, conviction* dan *fisher's exact test*.

4.7.1 Evaluasi Menggunakan Metrik *Lift*

Metrik *lift* digunakan untuk mengukur kekuatan asosiasi antara item pada bagian *antecedent* dan *consequent* dalam suatu aturan. Nilai *lift* menunjukkan sejauh mana kemunculan *antecedent* berpengaruh terhadap kemunculan *consequent*, dibandingkan jika keduanya muncul secara acak. Semakin tinggi nilai *lift*, semakin kuat hubungan antar item dalam aturan tersebut. Interpretasi nilai *lift* dapat dijabarkan sebagai berikut:

- *lift* > 1 mengindikasikan adanya asosiasi positif antara *antecedent* dan *consequent*. Artinya, kemunculan *antecedent* meningkatkan kemungkinan munculnya *consequent* dalam transaksi yang sama.
- *lift* = 1 menunjukkan bahwa *antecedent* dan *consequent* bersifat independen, sehingga tidak terdapat hubungan asosiasi di antara keduanya.
- *lift* < 1 menunjukkan adanya asosiasi negatif, kemunculan *antecedent* justru menurunkan kemungkinan munculnya *consequent* dalam suatu transaksi.

Evaluasi metrik *lift* dilakukan terhadap seluruh aturan asosiasi yang telah terbentuk. Untuk memperjelas distribusi kekuatan asosiasi, aturan-aturan tersebut dikelompokkan berdasarkan rentang nilai *lift*, sebagaimana ditunjukkan pada potongan kode berikut.

```
# Filter aturan dengan nilai lift > 1
lift_positive = all_rules[all_rules['lift'] > 1]
# Filter aturan dengan nilai lift = 1
lift_neutral = all_rules[all_rules['lift'] == 1]
# Filter aturan dengan nilai lift < 1
lift_negative = all_rules[all_rules['lift'] < 1]
```

Dari total 90 aturan asosiasi yang dihasilkan, sebanyak 22 aturan (24,44%) memiliki nilai *lift* lebih dari 1. Tidak ditemukan aturan dengan nilai *lift* sama dengan 1. Sementara itu, 68 aturan (75,56%) memiliki nilai *lift* kurang dari 1. Tabel berikut menyajikan beberapa contoh aturan dengan nilai *lift* tertinggi:

Tabel 4.9 Contoh Aturan Asosiasi dengan Nilai *Lift* lebih dari 1

Antecedent	Consequent	Support	Confidence	Lift
Sastra Lain	Sastra Amerika	0,0118	0,5426	4,1059
Sastra Amerika	Sastra Lain	0,0118	0,0896	4,1059
Ilmu Kesehatan & Kedokteran	Manajemen Rumah Tangga	0,0093	0,1476	2,8374
Manajemen Rumah Tangga	Ilmu Kesehatan & Kedokteran	0,0093	0,1786	2,8374
Sosial Ekonomi	Administrasi & Organisasi	0,0118	0,1067	2,1569

Nilai *lift* sebesar 4,1059 pada aturan antara kategori *Sastra Lain* dan *Sastra Amerika* menunjukkan bahwa kedua kategori tersebut cenderung muncul secara bersamaan dalam satu transaksi peminjaman. Temuan ini mengindikasikan adanya keterkaitan minat antara kedua kategori, di mana pengguna yang meminjam buku dari *Sastra Lain* juga kemungkinan tertarik pada *Sastra Amerika*, atau sebaliknya. Sebaliknya, aturan dengan nilai *lift* kurang dari 1 menunjukkan lemahnya asosiasi antar item yang terlibat. Tabel berikut menyajikan beberapa contoh aturan dengan nilai *lift* di bawah 1:

Tabel 4.10 Contoh Aturan Asosiasi dengan Nilai *Lift* kurang dari 1

Antecedent	Consequent	Support	Confidence	Lift
Desain & Gambar	Fiksi	0,0394	0,5230	0,9966
Fiksi	Desain & Gambar	0,0394	0,0752	0,9966
Hiburan, Permainan, & Olahraga	Fiksi	0,0071	0,5166	0,9844
Fiksi	Hiburan, Permainan, & Olahraga	0,0071	0,0137	0,9844
Zoologi	Fiksi	0,0125	0,5046	0,9615

4.7.2 Evaluasi Menggunakan Metrik *Leverage*

Setelah dilakukan evaluasi menggunakan metrik *lift*, langkah selanjutnya adalah menilai kekuatan asosiasi antar item menggunakan metrik *leverage*. Metrik *leverage* mengukur selisih antara probabilitas kemunculan bersama antara *antecedent* dan *consequent* dengan probabilitas yang diharapkan apabila keduanya bersifat independen. Nilai ini menunjukkan seberapa besar kemungkinan dua item muncul bersama dalam transaksi melebihi ekspektasi jika keduanya tidak saling terkait (independen). Interpretasi nilai *leverage* adalah sebagai berikut

- *leverage* > 0 menunjukkan adanya asosiasi positif, yaitu aturan tersebut terjadi lebih sering dibandingkan yang diharapkan secara acak.
- *leverage* = 0 menunjukkan tidak adanya asosiasi yakni *antecedent* dan *consequent* muncul secara independen.
- *leverage* < 0 menunjukkan adanya asosiasi negatif, di mana *antecedent* dan *consequent* cenderung tidak muncul bersama dalam satu transaksi.

Evaluasi *leverage* dilakukan terhadap seluruh 90 aturan asosiasi yang telah terbentuk pada proses sebelumnya. Potongan kode berikut digunakan untuk mengelompokkan aturan berdasarkan nilai *leverage*:

```
# Filter aturan dengan nilai leverage > 0
leverage_positive = all_rules[all_rules['leverage'] > 0]

# Filter aturan dengan nilai leverage = 0
leverage_neutral  = all_rules[all_rules['leverage'] == 0]

# Filter aturan dengan nilai leverage < 0
leverage_negative = all_rules[all_rules['leverage'] < 0]
```

Berdasarkan hasil evaluasi menggunakan metrik *leverage*, terdapat 25 aturan (27,78%) yang memiliki nilai lebih dari 0. Tidak ditemukan aturan dengan nilai *leverage* sama dengan 0. Sementara itu, sebanyak 65 aturan (72,22%) memiliki nilai *leverage* kurang dari 0. Beberapa contoh aturan dengan *leverage* positif tertinggi ditampilkan pada Tabel 4.11 berikut:

Tabel 4.11 Contoh Aturan dengan *Leverage* Positif

Antecedent	Consequent	Support	Confidence	Lift	Leverage
Sastra Lain	Sastra Amerika	0,0118	0,5425	0,0089	4,1058
Sastra Amerika	Sastra Lain	0,0118	0,0896	0,0089	4,1058
Ilmu Kesehatan & Kedokteran	Manajemen Rumah Tangga	0,0092	0,1476	0,0060	2,8373
Manajemen Rumah Tangga	Ilmu Kesehatan & Kedokteran	0,0092	0,1785	0,0060	2,8373
Fiksi	Adat, Tradisi, & Folklor	0,0359	0,0685	0,0039	1,1229

Mayoritas aturan menunjukkan nilai *leverage* negatif, yang mengindikasikan bahwa kombinasi item dalam aturan-aturan tersebut muncul lebih jarang daripada yang diharapkan berdasarkan probabilitas acak. Beberapa contoh aturan dengan nilai *leverage* negatif ditampilkan pada Tabel 4.12 berikut:

Tabel 4.12 Contoh Aturan dengan *Leverage* Negatif

Antecedent	Consequent	Support	Confidence	Lift	Leverage
Hiburan, Permainan, & Olahraga	Fiksi	0,007	0,516	0,984	-0,0001
Fiksi	Hiburan, Permainan, & Olahraga	0,007	0,013	0,984	-0,0001
Desain & Gambar	Fiksi	0,039	0,523	0,996	-0,0001
Sastra Amerika	Sastra	0,006	0,052	0,856	-0,0011
Sastra	Sastra Amerika	0,006	0,113	0,856	-0,0011

4.7.3 Evaluasi Menggunaan Metrik *Conviction*

Conviction adalah salah satu metrik yang digunakan untuk menilai seberapa kuat suatu aturan asosiasi dalam bentuk implikasi $A \rightarrow B$. Nilai *conviction* menunjukkan seberapa besar kemungkinan *consequent* akan terjadi ketika *antecedent* muncul, dibandingkan jika *consequent* tidak terjadi secara keseluruhan.

- *conviction* > 1 menunjukkan bahwa kemunculan *antecedent* cenderung diikuti oleh *consequent*, yang mengindikasikan adanya hubungan positif.
- *conviction* = 1 menunjukkan bahwa *antecedent* dan *consequent* bersifat independen (tidak saling memengaruhi).
- *conviction* < 1 menunjukkan bahwa kehadiran *antecedent* justru menurunkan kemungkinan munculnya *consequent*, yang mencerminkan hubungan negatif.

Evaluasi terhadap aturan asosiasi berdasarkan nilai *conviction* dilakukan dengan menggunakan potongan kode berikut:

```
# Filter aturan dengan nilai conviction > 0
conviction_positive = all_rules[all_rules['conviction'] > 1]
# Filter aturan dengan nilai conviction == 0
conviction_neutral = all_rules[all_rules['conviction'] == 1]
# Filter aturan dengan nilai conviction < 0
conviction_negative = all_rules[all_rules['conviction'] < 1]
```

Hasil evaluasi *conviction* terhadap aturan asosiasi yang telah dihasilkan menunjukkan bahwa sebanyak 22 aturan (24,44%) memiliki nilai *conviction* lebih dari 1. Tidak ditemukan aturan dengan nilai *conviction* sama dengan 1 (0%). Beberapa aturan dengan nilai *conviction* tertinggi disajikan pada tabel berikut:

Tabel 4.13 Contoh Aturan dengan *Conviction* Tertinggi

Antecedent	Consequent	Support	Confidence	Lift	Conviction
Sastra Lain	Sastra Amerika	0,011	0,542	4,105	1,897
Ilmu Politik	Agama-agama Lain	0,009	0,401	1,484	1,219
Adat, Tradisi, & Folklor	Fiksi	0,035	0,589	1,122	1,157
Manajemen Rumah Tangga	Ilmu Kesehatan & Kedokteran	0,009	0,178	2,837	1,140
Sastra Amerika	Sastra Lain	0,011	0,089	4,105	1,074

Sementara itu, sebanyak 68 aturan (75,56%) memiliki nilai *conviction* kurang dari 1. Beberapa aturan dengan nilai *conviction* kurang dari 1 ditunjukkan pada tabel dibawah ini:

Tabel 4.14 Contoh Aturan dengan *Conviction* Kurang dari 1

Antecedent	Consequent	Support	Confidence	Lift	Conviction
Fiksi	Hiburan, Permainan, & Olahraga	0,007	0,013	0,984	0,999
Fiksi	Desain & Gambar	0,039	0,075	0,996	0,999
Fiksi	Zoologi	0,012	0,023	0,961	0,999
Fiksi	Biologi Umum	0,008	0,016	0,882	0,997
Fiksi	Etika	0,008	0,015	0,833	0,996

Aturan-aturan dengan *conviction* kurang dari 1 tidak direkomendasikan sebagai dasar pengambilan keputusan karena menunjukkan hubungan asosiasi yang tidak konsisten dan cenderung acak.

4.7.4 Validasi Statistik Menggunakan *Fisher Exact Test*

Algoritma FP-Growth mampu mengidentifikasi aturan asosiasi berdasarkan frekuensi kemunculan item. Namun, tidak semua aturan yang dihasilkan dapat dianggap signifikan secara statistik. Mengingat data peminjaman buku dalam penelitian ini memiliki karakteristik *long-tail distribution* dan tingkat *sparsity* yang tinggi, diperlukan validasi yang dapat mengukur kekuatan keterkaitan antar item secara lebih objektif. Salah satu metode yang dapat digunakan adalah *Fisher Exact Test,* sebuah pendekatan statistik yang umum diterapkan untuk menguji asosiasi antara dua variabel kategorik dalam bentuk tabel kontingensi 2x2 (Hahsler, 2006).

Pendekatan yang digunakan adalah *two-sided test,* yang menguji apakah terdapat asosiasi positif maupun negatif secara signifikan antara dua item. Uji ini lebih konservatif dibandingkan *greater test* karena mempertimbangkan dua arah kemungkinan asosiasi (Agresti, 2002). Tingkat signifikansi yang digunakan adalah $\alpha = 0,05$, sesuai dengan standar dalam analisis statistik *inferensial* (Wasserman, 2004). Sebuah aturan asosiasi dikatakan signifikan secara statistik apabila nilai *p-value* hasil uji lebih kecil dari 0,05, yang menunjukkan bahwa asosiasi tidak terjadi secara kebetulan pada tingkat kepercayaan 95%.

Karena pengujian dilakukan terhadap banyak aturan sekaligus, maka diterapkan koreksi untuk *multiple testing* menggunakan metode *Benjamini-Hochberg (FDR).* Koreksi ini bertujuan untuk mengurangi risiko *false discovery rate* akibat pengujian berulang, tanpa mengorbankan kekuatan uji (*statistical power*), dan tetap menjaga kendali terhadap kesalahan Tipe I (Benjamini & Hochberg, 1995). Potongan kode berikut digunakan untuk mengevaluasi signifikansi aturan asosiasi dengan menghitung nilai *p-value* menggunakan *Fisher Exact Test*, serta menerapkan koreksi *multiple testing* menggunakan metode *Benjamini-Hochberg* (FDR):

```
import pandas as pd
import numpy as np
from scipy.stats import fisher_exact
from statsmodels.stats.multitest import multipletests

def fisher_test_with_correction(rules_df, onehot_df, alpha=0.05,
   correction_method='fdr_bh'):
```

```
    p_values, is_valid = []
    for _, row in rules_df.iterrows():
        antecedents = list(row['antecedents']) if isinstance(
            row['antecedents'], (frozenset, set)) else row['antecedents']
        consequents = list(row['consequents']) if isinstance(
            row['consequents'], (frozenset, set)) else row['consequents']
        if not all(item in onehot_df.columns for item in antecedents +
            consequents):
            p_values.append(np.nan)
            is_valid.append(False)
            continue
        A = onehot_df[antecedents].all(axis=1)
        B = onehot_df[consequents].all(axis=1)
        a = (A & B).sum()
        b = (A & ~B).sum()
        c = (~A & B).sum()
        d = (~A & ~B).sum()
        if (a + b == 0) or (c + d == 0) or (a + c == 0) or (b + d == 0):
            p_values.append(np.nan)
            is_valid.append(False)
            continue
        try:
            _, p = fisher_exact([[a, b], [c, d]], alternative='two-sided')
        except Exception:
            p = np.nan
        p_values.append(p)
        is_valid.append(True)
    result_df = rules_df.copy()
    result_df['fisher_p_value'] = p_values
    result_df['fisher_p_value_corrected'] = np.nan
    result_df['is_significant'] = False
    valid_mask = pd.Series(is_valid, index=result_df.index)
    if valid_mask.any():
       corrected_p = multipletests(
          result_df.loc[valid_mask, 'fisher_p_value'],
          alpha=alpha,
          method=correction_method
       )[1]
       result_df.loc[valid_mask,'fisher_p_value_corrected']=corrected_p
       result_df.loc[valid_mask, 'is_significant'] = corrected_p < alpha
    return result_df
```


Berdasarkan hasil pengujian diperoleh sebanyak 50 aturan yang signifikan secara statistik berdasarkan nilai *p-value* yang telah dikoreksi menggunakan metode *Benjamini-Hochberg* (FDR). Temuan ini menunjukkan bahwa meskipun banyak aturan memiliki nilai *support* yang rendah, asosiasi yang terbentuk tetap memiliki makna signifikan secara statistik. Tabel 4.15 menyajikan sebagian aturan yang lolos uji signifikansi tersebut:

Tabel 4.15 Contoh aturan yang lolos uji signifikansi *fisher exact test*

Antecedent	Consequent	Supp	Conf	Lift	p-FET	p-FDR	Sig
Agama - agama Lain	Fiksi	0,112	0,415	0,792	2,99e-18	4,49e-17	True
Fiksi	Agama - agama Lain	0,112	0,214	0,792	2,99e-18	4,49e-17	True
Pendidikan	Agama - agama Lain	0,270	0,019	0,185	1,60e-05	4,51e-05	True
Agama - agama Lain	Pendidikan	0,019	0,070	0,686	1,60e-05	4,51e-05	True
Pendidikan	Fiksi	0,034	0,337	0,643	7,94e-17	7,15e-16	True

Keterangan:

- *p-FET* adalah *p-value* dari *Fisher Exact Test*
- *p-FDR* adalah *p-value* setelah koreksi *Benjamini-Hochberg*
- *Sig* menunjukkan aturan tersebut signifikan atau tidak.

4.7.5 Seleksi Aturan Asosiasi

Setelah dilakukan evaluasi terhadap seluruh aturan asosiasi, langkah selanjutnya adalah menyaring aturan-aturan yang dianggap paling relevan. Aturan terpilih adalah yang memenuhi seluruh kriteria berikut:

- *lift* > 1 menunjukkan asosiasi positif antara *antecedent* dan *consequent*.
- *leverage* > 0 menunjukkan bahwa kemunculan keduanya lebih sering daripada yang diharapkan secara acak.
- *conviction* > 1 menandakan peningkatan kepercayaan terhadap *consequent* jika *antecedent* hadir dan bukan secara kebetulan.
- *p-value* < *0,05* (setelah koreksi *Benjamini-Hochberg*) menunjukkan signifikansi statistik berdasarkan *fisher exact test*.

Potongan kode berikut digunakan untuk menyaring aturan yang memenuhi seluruh kriteria tersebut:

```
selected_rules = all_rules[
    (all_rules['lift'] > 1) &
    (all_rules['leverage'] > 0) &
    (all_rules['conviction'] > 1) &
    (all_rulse['is_significant'] == True)
]
```

Hasil dari proses seleksi menghasilkan 8 aturan yang memenuhi seluruh kriteria evaluasi yang telah ditetapkan. Aturan-aturan ini dapat dianggap sebagai yang paling relevan dan bermakna. Tabel berikut menyajikan daftar aturan tersebut:

Tabel 4.16 Aturan Asosiasi yang Lolos Seleksi Evaluasi dan Validasi

Antecedent	Consequent	Lift	Lev	Conv	p-FET	p-FDR	Sig
Sastra Amerika	Sastra Lain	4,103	0,009	1,074	2,99e-18	4,49e-17	True
Sastra Lain	Sastra Amerika	4,105	0,009	1,897	2,99e-18	4,49e-17	True
Ilmu Kesehatan & Kedokteran	Manajemen Rumah Tangga	2,837	0,006	1,112	1,60e-05	4,51e-05	True
Manajemen Rumah Tangga	Ilmu Kesehatan & Kedokteran	2,837	0,006	1,140	1,60e-05	4,51e-05	True
Matematika	Pendidikan	1,582	0,002	1,071	7,94e-17	7,15e-16	True
Pendidikan	Matematika	1,582	0,002	1,031	5,35e-03	9,64e-03	True
Agama-agama Lain	Ilmu Politik	1,484	0,003	1,011	4,50e-03	8,80e-03	True
Ilmu Politik	Agama - agama Lain	1,484	0,003	1,219	4,50e-03	8,80e-03	True

Keterangan:

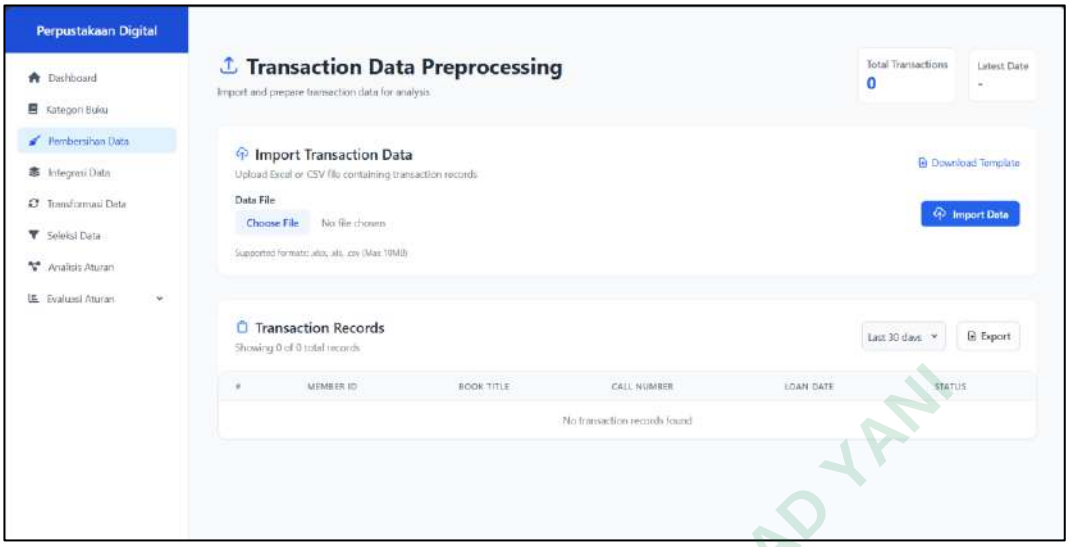
- *p-FET* adalah *p-value* dari *fisher exact test*
- *p-FDR* adalah *p-value* setelah koreksi *Benjamini-Hochberg*
- *Sig* menuntjukan aturan tersebut signifikan atau tidak.

4.8 PRESENTASI PENGETAHUAN

Pada tahap akhir dari proses analisis, hasil yang telah diperoleh perlu disajikan dalam bentuk yang dapat memudahkan pengguna dalam memahami dan memanfaatkan pengetahuan yang telah ditemukan melalui antarmuka sederhana.

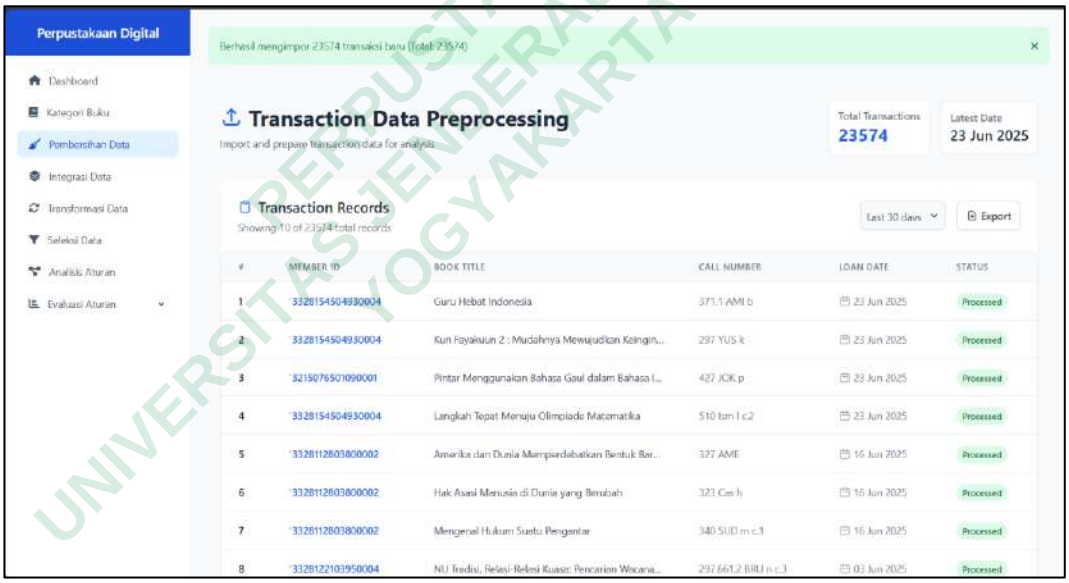
4.8.1 Halaman Pembersihan Data

Pada saat pertama kali mengakses sistem, pengguna akan diarahkan ke halaman Pembersihan Data dan diminta untuk mengimpor data peminjaman buku yang tersedia dalam format Excel atau CSV.



Gambar 4.3 Tampilan Halaman Pembersihan Data

Setelah data berhasil diimpor, sistem akan menyimpan data ke dalam *database* kemudian menampilkannya pada layar dalam bentuk tabel seperti berikut:

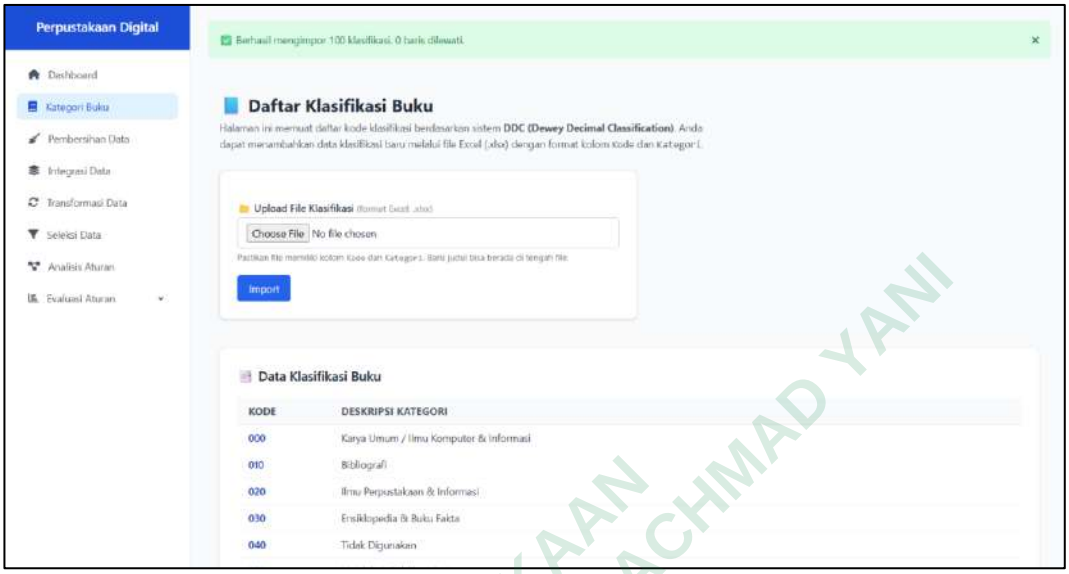


Gambar 4.4 Tampilan Halaman Pembersihan Data Setelah Berhasil Menyimpan Data

4.8.2 Halaman Kategori Buku

Halaman ini menampilkan daftar klasifikasi buku dalam bentuk tabel. Pengguna juga dapat mengunggah *file* Excel (.xlsx) yang berisi data klasifikasi, dengan kolom Kode dan Kategori untuk melengkapi data klasifikasi buku yang

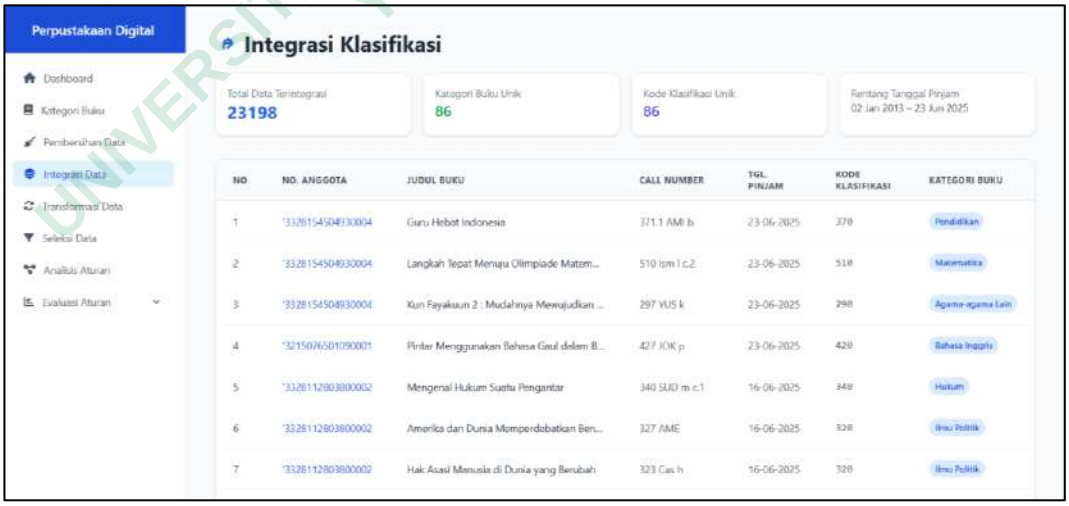
sudah ada pada *database* sebelumnya. Apabila sistem mendeteksi data yang sama dengan data yang ada disistem, data tersebut akan dilewati dan tidak ditambahkan.



Gambar 4.5 Tampilan Halaman Kategori Buku

4.8.3 Halaman Integrasi Data

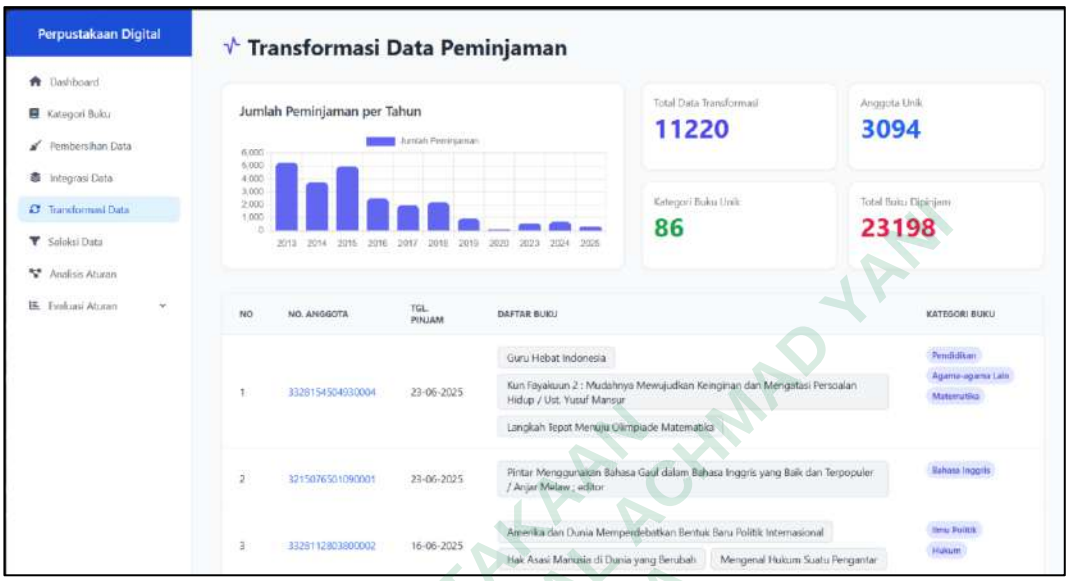
Pada halaman Integrasi Data, sistem akan mengambil data peminjaman buku dan data klasifikasi buku dari *database* kemudian sistem akan melakukan proses integrasi data pada kedua data tersebut. Sistem menampilkan data transaksi peminjaman buku yang telah disatukan dengan data klasifikasi buku.



Gambar 4.6 Tampilan Halaman Integrasi Data

4.8.4 Halaman Transformasi Data

Halaman ini menampilkan data peminjaman buku yang berhasil ditransformasikan dalam bentuk tabel seperti gambar berikut:



Gambar 4.7 Tampilan Halaman Transformasi Data

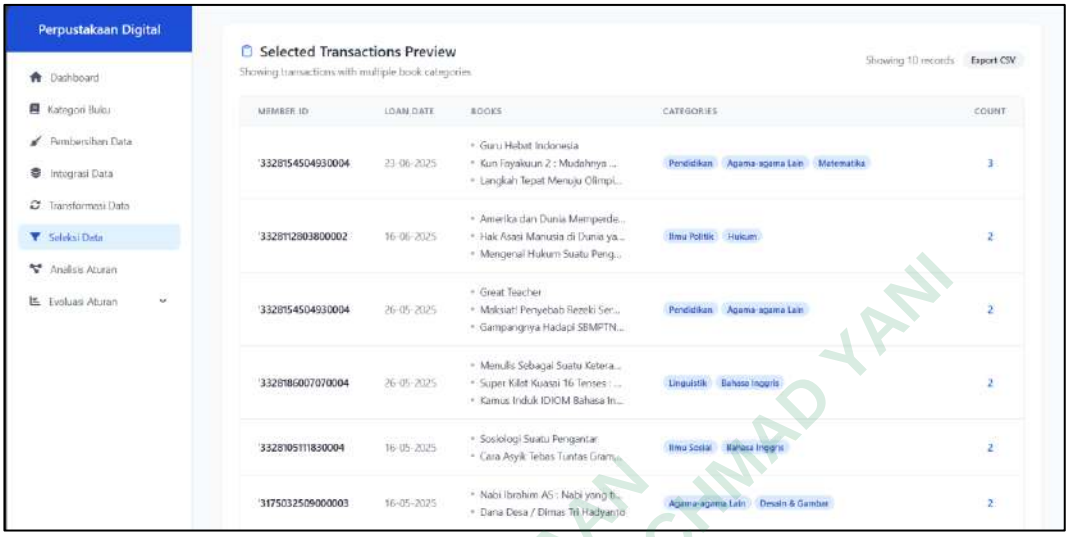
4.8.5 Halaman Seleksi Data

Halaman ini menampilkan grafik yang perbandingan jumlah transaksi peminjaman berdasarkan kategori buku sebelum dan setelah proses seleksi.

Gambar 4.8 Tampilan Halaman Transformasi Data

Selain itu, sistem menampilkan informasi kategori yang paling sering dipinjam oleh pemustaka. Sistem juga menampilkan data peminjaman buku yang

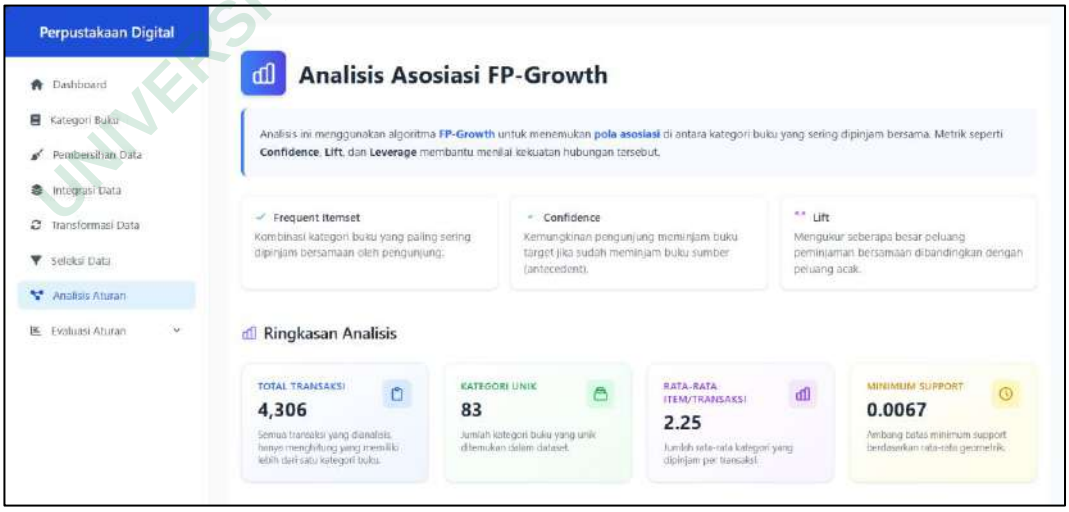
telah diseleksi kedalam bentuk tabel. Pengguna juga dapat mengekspor data tersebut kedalam format excel apabila diperlukan.



Gambar 4.9 Tampilan Data Peminjaman Buku yang Telah Diseleksi

4.8.6 Halaman Analisis Aturan

Halaman Analisi Aturan menampilkan hasil dari proses implementasi algoritma FP-Growth beserta penjelasan untuk untuk memudahkan pengguna memahami proses tersebut. Halaman ini juga menampilkan ringkasan hasil proess implementasi yang mencakup jumlah transaksi, kategori yang ditemukan, nilai *minimum support* yang digunakan dalam analisis dan yang lainnya.

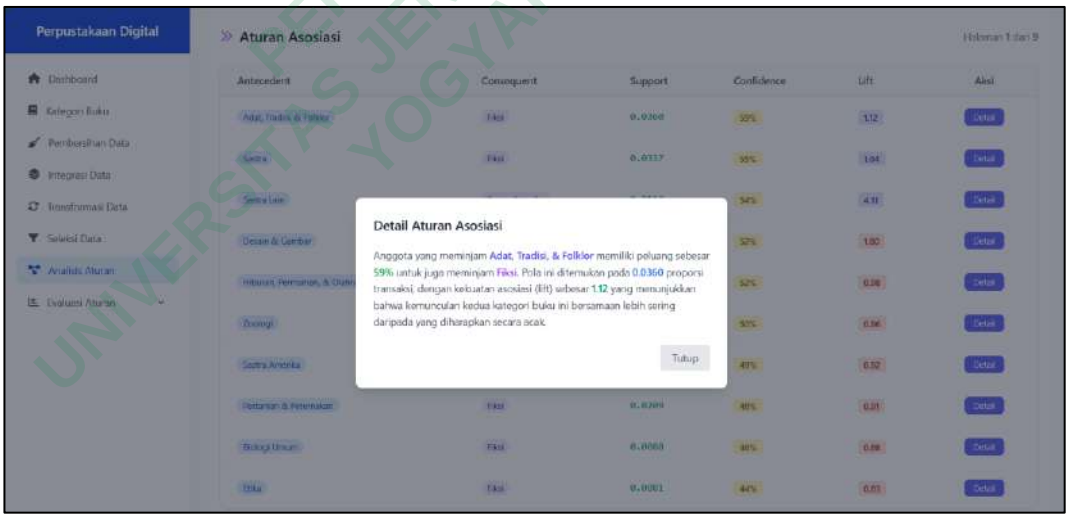


Gambar 4.10 Tampilan Halaman Analisis Aturan

Sistem juga menampilkan hasil *Frequent itemset* yang telah diekstrak dalam bentuk tabel. Pengguna juga dapat mencari kategori tertentu di dalam tabel untuk menemukan pola asosiasi yang relevan dengan kategori yang dicari.

Gambar 4.11 Tampilan *Frequent Itemset* yang Telah Diekstraksi

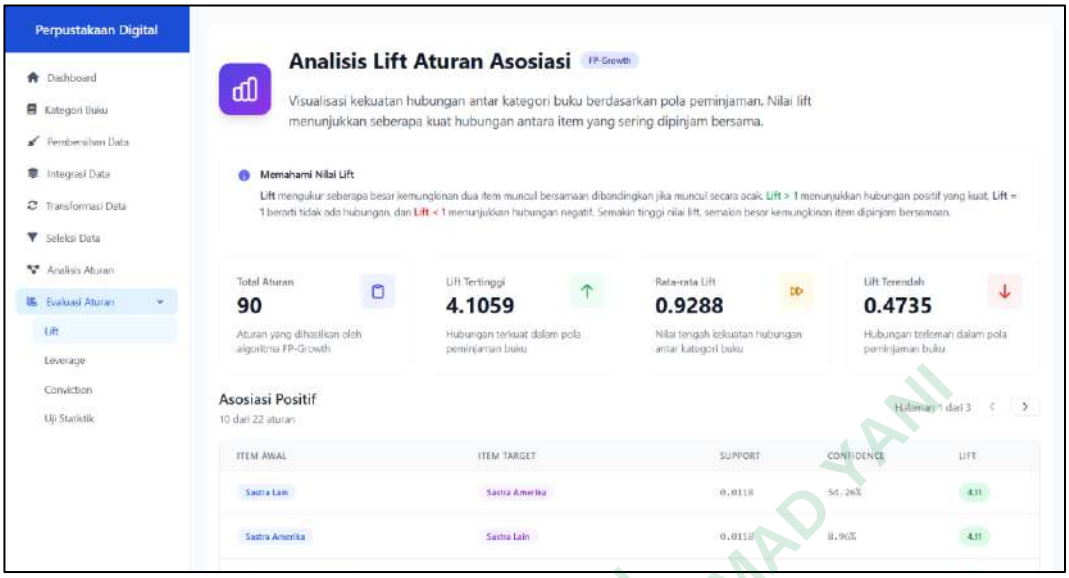
Sistem juga menampilkan aturan-aturan asosiasi yang terbentuk dalam bentuk tabel dengan penjelasan untuk setiap aturan.



Gambar 4.12 Tampilan Aturan Asosiasi yang Terbentuk

4.8.7 Halaman Evaluasi Aturan - *Lift*

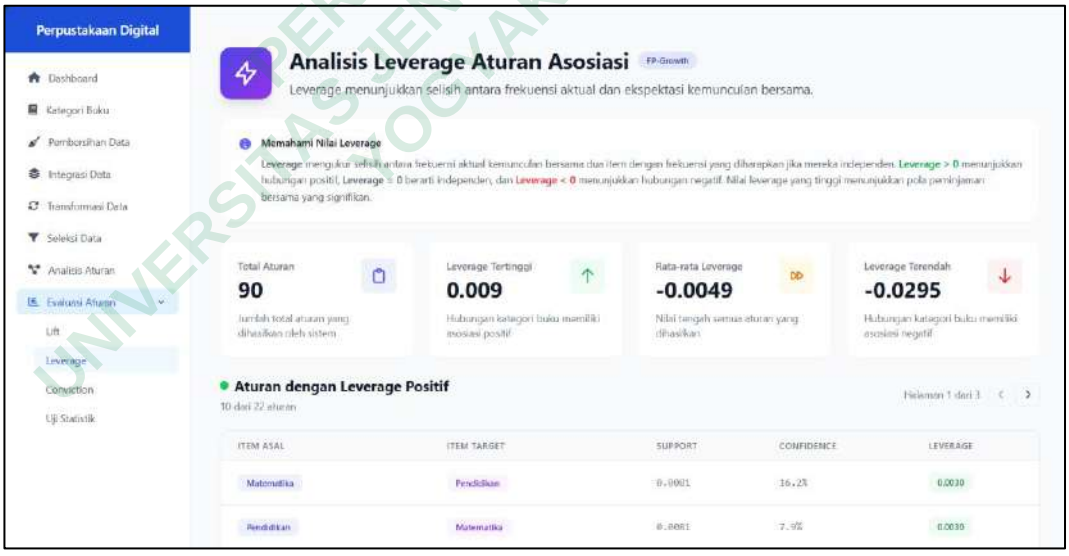
Halaman Evaluasi Aturan – *Lift* menampilkan hasil evaluasi terhadap aturan-aturan yang telah terbentuk menggunakan metrik *lift*. Halaman ini juga menyajikan penjelasan mengenai nilai *lift* beserta interpretasinya.



Gambar 4.13 Tampilan Halaman Evaluasi Aturan - *Lift*

4.8.8 Halaman Evaluasi Aturan - *Leverage*

Halaman Evaluasi Aturan – *Leverage* menampilkan hasil dari proses evaluasi aturan-aturan yang terbentuk menggunakan metrik *leverage*. Halaman ini juga menampilkan penjelasan mengenai nilai *leverage* beserta interpretasinya.

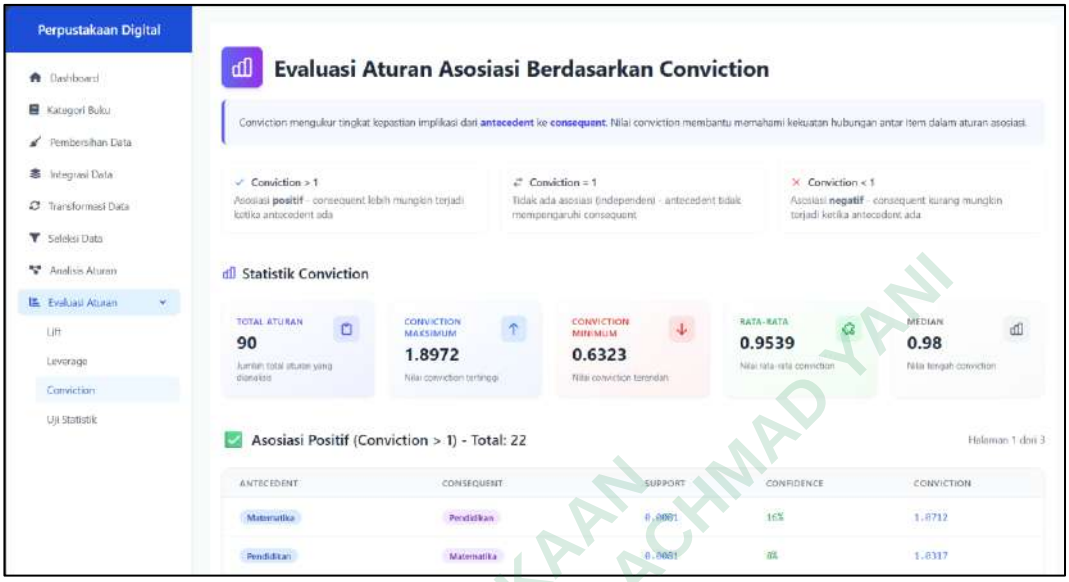


Gambar 4.14 Tampilan Halaman Evaluasi Aturan - *Leverage*

4.8.9 Halaman Evaluasi Aturan - *Conviction*

Halaman Evaluasi Aturan – *Conviction* menampilkan hasil evaluasi aturan-aturan asosiasi menggunakan metrik *conviction*. Aturan-aturan dikelompokkan

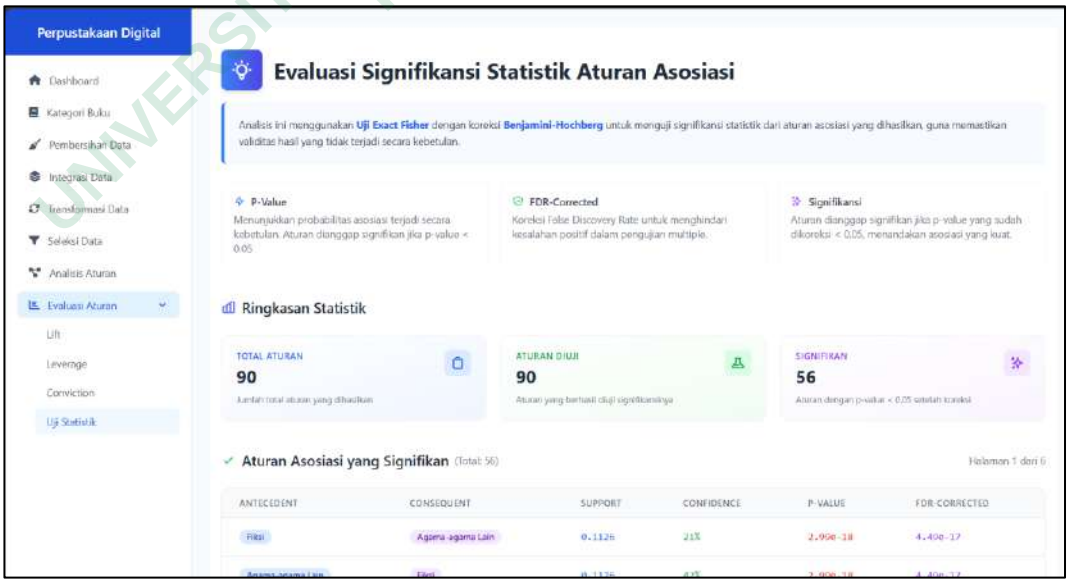
berdasarkan nilai *convictionn*ya. Halaman ini juga menampilkan statistik nilai maksimum, minimum, rata-rata, dan median *conviction*.



Gambar 4.15 Tampilan Halaman Evaluasi Aturan - *Conviction*

4.8.10 Halaman Evaluasi Aturan – *Fisher's Exact Test*

Pada halaman ini, sistem mengevaluasi signifikansi statistik aturan asosiasi menggunakan uji *fisher exact test* dengan koreksi *Benjamini-Hochberg*. Halaman ini menampilkan ringkasan statistik, termasuk jumlah aturan yang diuji, aturan yang signifikan, serta nilai *p-value* yang dikoreksi.



Gambar 4.16 Tampilan Halaman Evaluasi Aturan – *Fisher's Exact Test*

4.9 PEMBAHASAN

Bagian ini membahas seluruh tahapan dan temuan yang diperoleh dari proses analisis data peminjaman buku perpustakaan menggunakan algoritma FP-Growth. Pembahasan meliputi penyiapan data, transformasi dan seleksi, hingga pembentukan dan evaluasi aturan asosiasi.

4.9.1 Penyiapan Data dan Seleksi Data

Data awal yang digunakan merupakan data transaksi peminjaman buku dengan struktur baris per item. Setelah dilakukan proses pembersihan dan pemetaan call number menjadi kode klasifikasi DDC, data berhasil dikonversi menjadi format yang dapat digunakan untuk analisis asosiasi kategori buku.

Transformasi data menghasilkan 11.219 transaksi unik, yang didefinisikan sebagai kombinasi unik antara nomor anggota dan tanggal peminjaman. Namun, dari jumlah tersebut hanya 4.305 transaksi (38%) yang terdiri atas dua kategori atau lebih, dan memenuhi syarat untuk dianalisis lebih lanjut menggunakan algoritma asosiasi. Hal ini mencerminkan bahwa mayoritas transaksi hanya memuat satu kategori buku, sehingga tidak dapat memberikan kontribusi dalam pembentukan relasi antar kategori buku.

4.9.2 Distribusi Kategori Buku dan Karakteristik Data

Distribusi kategori buku menunjukkan bahwa beberapa kategori memiliki frekuensi kemunculan yang tinggi, seperti Fiksi, Agama-agama Lain, dan Pendidikan. Namun, proporsi kategori Fiksi menurun signifikan setelah proses seleksi, karena cenderung muncul sebagai kategori tunggal dalam transaksi. Hal ini mengindikasikan bahwa meskipun Fiksi merupakan kategori populer, ia jarang dipinjam bersama dengan kategori lain. Sebaliknya, kategori seperti Pendidikan, Sastra, dan Ilmu Sosial lebih sering muncul dalam kombinasi sehingga memiliki potensi lebih besar dalam pembentukan asosiasi.

4.9.3 Penentunan *Minimum Support* dan Penerapan FP-Growth

Nilai *minimum support* ditentukan menggunakan pendekatan *geometric mean* dari frekuensi kategori buku, menghasilkan ambang 0,00675. Pendekatan ini

digunakan untuk menghindari bias akibat distribusi frekuensi yang tidak merata. Setelah penerapan algoritma FP-Growth, ditemukan 89 *itemset* yang memenuhi nilai *minimum support*. Beberapa *itemset* terdiri dari kombinasi dua atau lebih kategori buku seperti:

- Agama-agama Lain
- Pendidikan
- Fiksi, Hiburan, Permainan, & Olahraga

Namun, tidak semua kombinasi yang sering muncul menunjukkan asosiasi yang kuat atau relevan, sehingga perlu evaluasi lebih lanjut dengan metrik tambahan.

4.9.4 Pembentukan dan Evaluasi Aturan Asosiasi

Setelah *frequent itemset* berhasil diperoleh, tahap selanjutnya adalah pembentukan aturan asosiasi menggunakan parameter *confidence* dengan nilai awal 0,0. Pendekatan ini bertujuan untuk mengekstraksi seluruh aturan potensial tanpa melakukan penyaringan awal, sehingga seluruh kemungkinan hubungan implikatif antar kategori buku dapat dianalisis secara menyeluruh.

Setiap aturan yang terbentuk merepresentasikan hubungan potensial antara dua atau lebih kategori buku berdasarkan kebiasaan peminjaman pengguna perpustakaan. Namun, hasil analisis menunjukkan bahwa tidak semua aturan memiliki kekuatan asosiasi yang tinggi. Evaluasi menggunakan tiga metrik utama, yaitu *lift*, *leverage*, dan *conviction*, menunjukkan distribusi sebagai berikut:

- Sebanyak 22 aturan (24,4%) memiliki nilai *lift* > 1, yang mengindikasikan adanya asosiasi positif antara kategori buku yang dianalisis.
- Sebanyak 25 aturan (27,8%) menunjukkan nilai *leverage* positif, menandakan deviasi yang bermakna dari ekspektasi acak.
- Sebanyak 22 aturan (24,4%) memiliki *conviction* > 1, yang mencerminkan kestabilan arah implikatif dari *antecedent* ke *consequent*.

Mayoritas aturan yang terbentuk tidak menunjukkan asosiasi yang kuat secara metrik evaluasi. Oleh karena itu, dilakukan validasi statistik tambahan menggunakan *fisher exact test* dengan koreksi *Benjamini-Hochberg*.

Hasil evaluasi dan validasi menunjukkan bahwa hanya delapan aturan yang memenuhi seluruh kriteria evaluasi. Aturan-aturan ini dianggap paling relevan, konsisten, dan bermakna. Aturan-aturan asosiasi yang lolos seleksi disajikan dalam bentuk implikatif berikut:

- Jika seseorang meminjam buku dari kategori Sastra Lain, maka terdapat kemungkinan sebesar 54% bahwa ia juga akan meminjam buku dari kategori Sastra Amerika.
- Jika seseorang meminjam buku dari kategori Ilmu Politik, maka terdapat kemungkinan sebesar 40% bahwa ia juga akan meminjam buku dari kategori Agama-agama Lain.
- Jika seseorang meminjam buku dari kategori Manajemen Rumah Tangga, maka terdapat kemungkinan sebesar 18% bahwa ia juga akan meminjam buku dari kategori Ilmu Kesehatan & Kedokteran.
- Jika seseorang meminjam buku dari kategori Ilmu Kesehatan & Kedokteran, maka terdapat kemungkinan sebesar 15% bahwa ia juga akan meminjam buku dari kategori Manajemen Rumah Tangga.
- Jika seseorang meminjam buku dari kategori Matematika, maka terdapat kemungkinan sebesar 16% bahwa ia juga akan meminjam buku dari kategori Pendidikan.
- Jika seseorang meminjam buku dari kategori Pendidikan, maka terdapat kemungkinan sebesar 8% bahwa ia juga akan meminjam buku dari kategori Matematika.
- Jika seseorang meminjam buku dari kategori Sastra Amerika, maka terdapat kemungkinan sebesar 9% bahwa ia juga akan meminjam buku dari kategori Sastra Lain.
- Jika seseorang meminjam buku dari kategori Agama-agama Lain, maka terdapat kemungkinan sebesar 3,5% bahwa ia juga akan meminjam buku dari kategori Ilmu Politik.

4.9.5 Keterbatasan dan Implikasi Temuan

Hasil analisis pada penelitian yang dilakukan menunjukkan keberhasilan dalam mengidentifikasi sejumlah pola asosiasi antar kategori buku berdasarkan data transaksi peminjaman. Namun demikian, terdapat beberapa keterbatasan yang memengaruhi keluasan dan kedalaman temuan, baik dari sisi struktur data maupun kekuatan asosiasi yang terbentuk.

1. Tingginya proporsi transaksi tunggal

   Sebanyak 62% dari total transaksi hanya memuat satu kategori buku, sehingga tidak dapat digunakan dalam pembentukan aturan asosiasi. Hal ini membatasi jumlah data yang dapat dianalisis, dan secara langsung mempengaruhi jumlah serta keragaman pola asosiasi yang dapat dihasilkan.

2. Dominasi kategori dengan frekuensi tinggi namun asosiasi lemah

   Kategori *Fiksi* merupakan salah satu yang paling sering muncul dalam dataset dan juga banyak terlibat dalam aturan asosiasi. Namun, berdasarkan evaluasi metrik, sebagian besar aturan yang melibatkan *Fiksi* memiliki nilai *lift* < 1 dan *conviction* < 1, yang menunjukkan asosiasi lemah. Kondisi ini menunjukkan bahwa tingginya frekuensi kemunculan suatu kategori tidak selalu sejalan dengan kekuatan asosiasinya, dan bahkan dapat menimbulkan noise dalam proses penambangan pola.

3. Jumlah aturan signifikan terbatas

   Dari total 90 aturan yang terbentuk, hanya 8 aturan yang memenuhi seluruh kriteria evaluasi. Aturan-aturan yang lolos cenderung terkonsentrasi pada kategori tertentu dan belum mencerminkan kebiasaan peminjaman pengguna secara menyeluruh.

Walaupun demikian, aturan - aturan yang lolos evaluasi tetap memberikan gambaran tentang kebiasaan pengguna dalam meminjam buku. Pola-pola ini dapat dijadikan dasar untuk pengembangan layanan perpustakaan yang lebih sesuai dengan kebutuhan pengguna.