

Analisis Pola Pembelian Obat Menggunakan Algoritma *Direct Hashing & Pruning*

Febrianti Adhania^{1*} dan Iman Paryudi²

^{1,2}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Pancasila, Jakarta

Abstrak. Pelayanan kesehatan masyarakat merupakan aspek penting dalam kehidupan masyarakat saat ini, salah satu contohnya adalah apotek. Banyak apotek belum menerapkan data mining dalam mengelola perusahaannya, sehingga terjadi kendala bagi apoteker dalam mencari obat yang dibutuhkan konsumen. Untuk mengatasi permasalahan tersebut, maka dibutuhkan sebuah sistem yang dapat menganalisis pola pembelian obat berdasarkan data transaksi dengan tujuan mempermudah bagi apoteker dalam penempatan obat yang teratur sehingga mudah untuk mengetahui persediaan stok obat. Penelitian ini menggunakan metode *Direct Hashing & Pruning* dan data yang digunakan merupakan data transaksi obat selama satu bulan. Dalam penelitian ini mendapatkan hasil 1 aturan asosiasi, dengan menggunakan minimum *support* 0.3% dan minimum *confidence* 70%.

Kata kunci— aturan asosiasi; *direct hashing*; obat; *pruning*

1. PENDAHULUAN

Pelayanan kesehatan masyarakat merupakan aspek penting dalam kehidupan masyarakat saat ini. Selain rumah sakit, apotek juga merupakan fasilitas kesehatan yang menjadi tempat dilakukannya kegiatan kefarmasian yang meliputi penyediaan, penyimpanan, penyiapan, dan penjualan obat [1].

Terdapat kendala bagi apoteker dalam mencari obat yang dibutuhkan oleh konsumen dan dalam menentukan obat apa saja yang harus dibeli untuk memenuhi kebutuhan konsumen. Saat ini masih banyak apotek yang belum menerapkan data mining dalam mengelola perusahaannya. Contohnya adalah dalam menentukan pola kecenderungan pembelian obat.

Untuk mengatasi permasalahan diatas, maka dibuat sebuah aplikasi berbasis *web* yang dapat menganalisis pola pembelian obat berdasarkan data transaksi dengan menggunakan algoritma *Direct Hashing & Pruning* (DHP). Aplikasi berbasis *web* ini akan memberikan pola pembelian obat yang paling sering dibeli secara bersamaan berdasarkan data yang diinput oleh apoteker sehingga dapat membantu apoteker dalam penempatan obat yang teratur agar mudah untuk mengetahui persediaan stok obat.

Penelitian ini mengacu pada penelitian sebelumnya yang telah dilakukan oleh Maslihatin et al [2]. Perbedaan penelitian ini dengan penelitian tersebut adalah pada penelitian sebelumnya peneliti menggunakan algoritma Apriori, sedangkan pada penelitian ini menggunakan algoritma DHP. Oleh karena itu, perlu dikaji lebih lanjut mengenai penggunaan algoritma DHP pada aplikasi untuk pola pembelian obat.

2. LANDASAN TEORI

a. Obat

Obat dapat diartikan sebagai komponen atau campuran yang digunakan oleh semua makhluk hidup untuk bagian dalam dan luar tubuh yang dapat mencegah, meringankan, dan menyembuhkan penyakit. Obat-obatan merupakan elemen penting karena diperlukan dalam sebagian besar upaya kesehatan untuk menghilangkan atau menyembuhkan suatu penyakit [3].

Obat sangat penting dalam bidang kesehatan karena dapat mencegah dan menangani berbagai jenis penyakit melalui penggunaan obat. Obat berfungsi untuk diagnosis, mencegah penyakit, menyembuhkan penyakit, memulihkan kesehatan, mengubah fungsi normal tubuh untuk mencapai tujuan tertentu, dan mengurangi rasa sakit [4].

* Corresponding author: Febriadh@gmail.com

b. Association Rule

Association rule adalah sebuah metode yang digunakan untuk menemukan aturan yang menghubungkan *item-item* dalam sebuah transaksi [5]. Salah satu langkah dalam analisis yang sesuai untuk menghasilkan algoritma yang efisien adalah analisis pola frekuensi tinggi. Terdapat dua tolak ukur di dalam mengasosiasi, yaitu *support* dan *confidence* [6], yang dimana *support* adalah nilai presentase kemunculan suatu *itemset* dalam *database*, dan *confidence* adalah nilai kepastian yang menunjukkan kuatnya hubungan antara kombinasi *itemset* yang terbentuk.

Kedua parameter tersebut digunakan dalam metodologi dasar aturan asosiasi. Tahapan metodologi dasar aturan asosiasi dibagi menjadi dua, yaitu [7]:

1. Analisis Pola Frekuensi Tinggi

Pada tahap ini dicari kombinasi *itemset* yang memenuhi syarat nilai minimum *support* yang telah ditentukan. Nilai *support* sebuah *item* dapat dihitung menggunakan rumus (1).

$$Support = \frac{\sum \text{transaksi } X}{\sum \text{seluruh transaksi}} \quad (1)$$

Sedangkan nilai *support* untuk dua *item* atau lebih dapat dihitung menggunakan rumus (2).

$$Support = \frac{\sum \text{transaksi } X \text{ dan } Y}{\sum \text{seluruh transaksi}} \quad (2)$$

2. Pembentukan Aturan Asosiasi

Jika sudah mendapatkan pola frekuensi tinggi, langkah selanjutnya adalah membentuk aturan asosiasi yang memenuhi syarat *confidence*, yaitu dengan menghitung nilai *confidence* untuk aturan asosiasi “Jika X maka Y” = (X → Y). Untuk mendapatkan nilai *confidence* dapat dihitung menggunakan rumus (3).

$$Confidence = \frac{\sum \text{transaksi } X \text{ dan } Y}{\sum \text{transaksi } X} \quad (3)$$

Hasil aturan asosiasi yang dibentuk dapat dievaluasi dengan menggunakan teknik pengujian *lift ratio*. *Lift ratio* merupakan suatu teknik pengujian yang dapat digunakan dalam perhitungan aturan asosiasi untuk mengetahui keakuratan hasil aturan yang terbentuk, terlepas dari *valid* atau tidaknya aturan tersebut. Evaluasi menggunakan teknik *lift ratio* dapat dihitung menggunakan rumus (4) [8].

$$Lift \ ratio = \frac{Confidence (X \rightarrow Y)}{Support \ Y} \quad (4)$$

Aturan asosiasi dikatakan *valid* jika nilai *lift ratio* ≥ 1 , yang dimana setiap *item* memiliki keterkaitan satu sama lain. Sedangkan aturan asosiasi dikatakan tidak *valid* jika nilai *lift ratio* < 1 , yang dimana setiap *item* tidak memiliki keterkaitan sama sekali.

c. Algoritma Direct Hashing & Pruning

Algoritma *Direct Hashing & Pruning* (DHP) merupakan algoritma berbasis *hash* yang efektif untuk menyaring keluar *itemset* yang tidak penting untuk pembangkitan *itemset* selanjutnya. Pada metode ini, nilai *support* dihitung dengan memetakan *item* dari daftar kandidat ke lokasi penyimpanan dalam *tabel hash* [9]. Algoritma DHP mengumpulkan informasi mengenai (k+1)-*itemset* dengan cara seluruh kemungkinan (k+1)-*itemset* di *hash* ke dalam *tabel hash* dengan menggunakan fungsi *hash* [10]. Terdapat tiga tahapan pada algoritma DHP, yaitu [11]:

- 1) Tahap pertama dilakukan untuk menghasilkan satu *set large 1-itemset* dan membangun *tabel hash* untuk *2-itemset*.
- 2) Tahap kedua dilakukan pembangkitan kandidat *itemset* C_k berdasarkan *tabel hash* yang telah dibuat pada iterasi sebelumnya. Kemudian ditentukan *large itemset* dan mengurangi ukuran basis data untuk pembangkitan *itemset* selanjutnya.
- 3) Tahap ketiga pada dasarnya sama dengan tahap kedua tetapi tidak menggunakan *tabel hash*. Tahap kedua ini, dilakukan selama nilai *tabel hash* lebih besar dari minimum *support*.

Rumus *hash* untuk pemrosesan *tabel hash 2-itemset* dapat menggunakan rumus (5) [12].

$$H(XY) = ((Order\ of\ X * 10) + Order\ of\ Y) \bmod\ bil\ prima \quad (5)$$

Dimana :

$H(XY)$ = nilai *hash* dari pasangan *item* X dan Y.

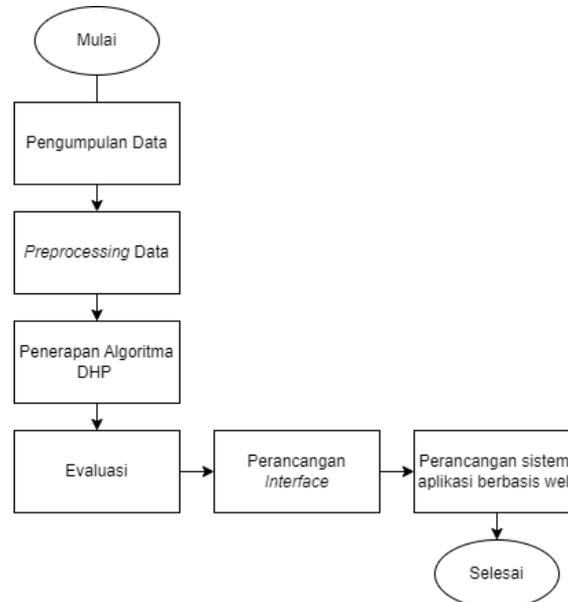
Order X = Nilai X

Order Y = Nilai Y

Mod bil prima = modulus bilangan prima.

3. METODE

Metodologi penelitian merupakan langkah-langkah atau tahapan yang dikerjakan pada penelitian ini, dimulai dari tahap pengumpulan data hingga tahap akhir yang menghasilkan aplikasi berbasis *web*. Tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1 Tahapan Penelitian

a. Pengumpulan Data

Pada tahap pengumpulan data, peneliti mengumpulkan *dataset* yang akan digunakan dalam penelitian ini, *dataset* yang digunakan merupakan data primer yang dikumpulkan dari riwayat transaksi harian penjualan obat selama satu bulan, yaitu bulan Januari 2024. Data yang telah terkumpul, yaitu tanggal transaksi, kode obat, nama obat, nomor *invoice* dll. Data yang digunakan pada penelitian ini sebanyak 5049 data dengan menggunakan dua kolom saja yaitu, tanggal transaksi dan nama obat. Tanggal transaksi digunakan untuk mengelompokkan jumlah obat setiap harinya dapat dilihat pada Gambar 2.

```
# Membaca file CSV
df = pd.read_csv("/content/gdrive/MyDrive/Colab Notebooks/Data_Obat.csv", header=None)
df.head()
```

	0	1	2	3	4	5	6
0	LACTO B	TEMPRA DROPS	GPU SEREH 30 ML	NaN	NaN	NaN	NaN
1	BIOMEGA	RENADINAC 50 MG	ASAM MEF (NOVA)	DEXAMETHASONE GENERIK	NaN	NaN	NaN
2	TERMOREX PLUS 30 ML	MKP CAP LANG 15 ML	LAMBUCID SY 60ML	KELLY LEMON 75 GR	ANDALAN PIL KB	NaN	NaN
3	FAXIDENE 10 MG	GRATAZONE 0,5 MG	OMERIC 300 MG	METFORMIN 500 MG	CAVIPLX TAB	NaN	NaN
4	BEPANTHEN 10 GR	NEW DIATAB	NaN	NaN	NaN	NaN	NaN

Gambar 2 Data Obat

b. Preprocessing Data

Setelah melakukan pengumpulan data, tahap selanjutnya yaitu *preprocessing* data. Data yang telah terkumpul akan diproses menjadi data yang siap digunakan untuk proses analisis. Pada tahap ini, dilakukan *importing module* menggunakan *Pandas* serta *Numpy* yang merupakan *module* untuk *preprocessing* data. *Defaultdict* merupakan *module* untuk menyederhanakan operasi dengan nilai *default*, sedangkan *itertools* merupakan *module* yang menghasilkan set kombinasi untuk analisis. *Pickle* merupakan *module* untuk memuat model yang sudah dilatih atau data yang sudah diproses sebelumnya. Gambar 3 menunjukkan semua *module* yang digunakan.

```
[1] import pandas as pd
import numpy as np
from collections import defaultdict
from itertools import combinations
import pickle
```

Gambar 3 Importing Module

Setelah melakukan tahapan *importing module* yang diperlukan, maka tahap selanjutnya adalah pembersihan data kosong. Pada tahapan ini *dataset* akan diproses dan kemudian dikonversi kedalam format *list of list* agar data dapat disimpan dalam bentuk matriks atau tabel dua dimensi.

```
# Menghapus nilai NaN dan mengonversi data ke dalam format list of lists
df = df.dropna(how='all')
data = df.values.tolist()

# Menghapus nilai NaN dari setiap transaksi
data = [[item for item in transaction if pd.notna(item)] for transaction in data]

# Menampilkan hasil data yang telah dihapus nilai NaN-nya
for i, transaction in enumerate(data[:10]):
    print(f"Transaksi {i+1}: {transaction}")

Transaksi 1: ['LACTO B', 'TEMPRA DROPS', 'GPU SEREH 30 ML']
Transaksi 2: ['BIOMEGA', 'RENADINAC 50 MG', 'ASAM MEF (NOVA)', 'DEXAMETHASONE GENI']
Transaksi 3: ['TERMOREX PLUS 30 ML', 'MKP CAP LANG 15 ML', 'LAMBUCID SY 60ML', 'KE']
Transaksi 4: ['FAXIDENE 10 MG', 'GRATAZONE 0,5 MG', 'OMERIC 300 MG', 'METFORMIN 50']
Transaksi 5: ['BEPANTHEN 10 GR', 'NEW DIATAB']
Transaksi 6: ['MICROLAX SUPP 5 ML', 'RL 500 ML (OTSU)']
Transaksi 7: ['ANTASIDA SYR 60 ML (ERELA)', 'MELOXICAM 15 MG (DEXA)', 'MOLACORT 0.]
Transaksi 8: ['MADU SUMBAWA PRIMA', 'SPASMICAL', 'CAVIPLX TAB']
Transaksi 9: ['VESPERUM', 'TRIANTA SY 60 ML', 'OBH COMBI PLUS 60 ML', 'NEURALGIN']
Transaksi 10: ['DULCOLAX ISI 4', 'DEXTAFEN', 'GRAFIX']
```

Gambar 4 Pembersihan Data Kosong

c. Penerapan Algoritma Direct Hashing & Pruning

Setelah melakukan *preprocessing* data, tahap selanjutnya adalah menerapkan algoritma DHP pada *dataset* yang akan digunakan. Pada tahap ini, yang pertama dilakukan adalah membangun kandidat *itemset* yang kemudian dihitung nilai *support* nya untuk setiap kandidat. Setelah menghitung nilai *support*, selanjutnya adalah melakukan *pruning* atau pemotongan terhadap kandidat *itemset* yang tidak memenuhi nilai minimum *support*. Implementasi algoritma DHP ini menghasilkan aturan asosiasi dari *frequent itemset* yang telah terbentuk.

```
# Implementasi fungsi algoritma DHP dengan optimasi
def generate_candidates(itemset, k):
    """Membangkitkan kandidat itemsets dari itemsets yang diberikan"""
    return list(combinations(itemset, k))

def count_support(transactions, candidates):
    """Menghitung support dari setiap kandidat itemsets"""
    support_count = defaultdict(int)
    for transaction in transactions:
        for candidate in candidates:
            if set(candidate).issubset(transaction):
                support_count[candidate] += 1
    return support_count

def prune_candidates(support_count, min_support):
    """Melakukan pruning terhadap kandidat itemsets yang tidak memenuhi minimum support"""
    return {item: count for item, count in support_count.items() if count >= min_support}

def generate_rules(frequent_itemsets, min_confidence, num_transactions):
    """Menghasilkan aturan asosiasi dari frequent itemsets"""
    rules = []
    for k, itemset_support in enumerate(frequent_itemsets):
        if k == 0: # Handle the case for 1-itemsets
            continue
        for itemset, support in itemset_support.items():
            for i in range(1, len(itemset)):
```

Gambar 5 Penerapan Algoritma DHP

Aturan asosiasi yang terbentuk adalah 1 aturan, dengan menggunakan minimum *support* 0.3% dan minimum *confidence* 70%. Untuk aturan asosiasi dapat dilihat pada Tabel 1.

Tabel 1 Aturan Asosiasi.

<i>Association Rule</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift Ratio</i>
Jika membeli Gasela dan Biomega, maka membeli juga Carbidu 0,75 Mg	0,36%	94,74%	48.81

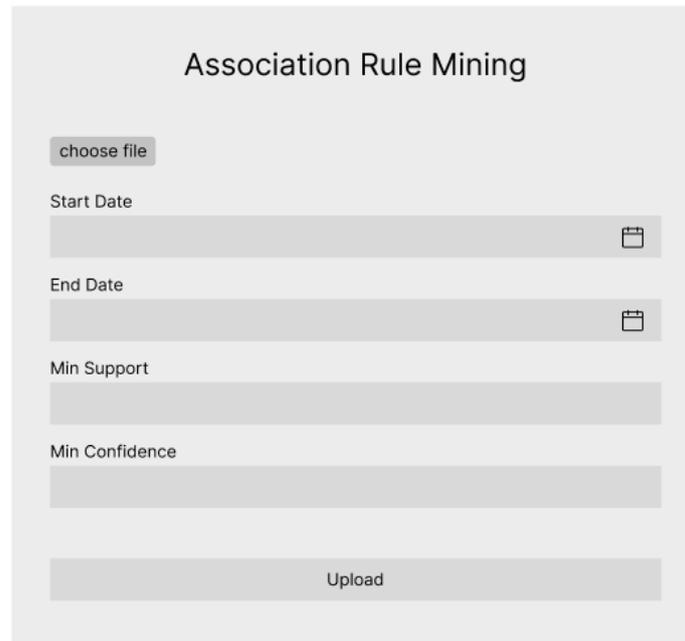
d. Evaluasi

Pada penelitian ini, menggunakan teknik evaluasi *lift ratio* untuk mencari pola pembelian obat. *Lift ratio* digunakan untuk memastikan validitas dari aturan asosiasi yang telah terbentuk. Aturan asosiasi dikatakan valid jika *lift ratio* ≥ 1 , sedangkan aturan asosiasi dikatakan tidak *valid* jika *lift ratio* < 1 .

Nilai *lift ratio* yang didapatkan dari hasil aturan asosiasi pada Tabel 1 adalah 48,81 yang dimana artinya aturan tersebut dikatakan *valid* karena *lift ratio* > 1 . Menggunakan *lift ratio* bersama dengan *confidence* dapat membantu memahami seberapa kuat hubungan antar *item* dalam *dataset*.

e. Perancangan Interface

Setelah tahapan evaluasi, maka tahap selanjutnya adalah merancang *interface* yang bertujuan untuk memberi kenyamanan pengguna ketika menggunakan aplikasi yang telah dirancang. Berikut merupakan *mockup* tampilan sistem yang telah dirancang.



Gambar 6 *Mockup* Tampilan Sistem

f. *Perancangan Sistem / Aplikasi Berbasis Web*

Pada tahap ini, peneliti merancang sebuah *website* menggunakan *framework flask* dan bahasa pemrograman *python*. Langkah pertama dalam pembuatan model *machine learning* kedalam sebuah *website* adalah membuat file *python* yang berisi model *machine learning* yang telah dibuat sebelumnya dan disimpan kedalam format *pickle* ekstensi.pkl dan diberi nama 'dhp_model.pkl'. Untuk menyimpan model kedalam bentuk *pickle* dapat dilihat pada gambar 7.

```
[11] # Menyimpan frequent itemsets dan rules ke dalam file pickle
      model_filename = 'dhp_model.pkl'
      with open(model_filename, 'wb') as f:
          pickle.dump((frequent_itemsets, rules), f)

      print(f"Hasil telah disimpan dalam file: {model_filename}")
```

⇒ Hasil telah disimpan dalam file: dhp_model.pkl

Gambar 7 Menyimpan Model Dalam Format *Pickle*

Setelah menyimpan model dalam format *pickle*, tahap selanjutnya adalah membuat file *app.py*. File ini berisi *script flask* yang memiliki fungsi untuk menghubungkan model asosiasi yang telah disimpan sebelumnya dengan tampilan *web* untuk pola pembelian obat. Proses ini dapat dilihat pada gambar 8.

```
from flask import Flask, request, render_template
import pandas as pd
import os
import pickle
import logging
from collections import defaultdict
from itertools import combinations

# Setup logging
logging.basicConfig(level=logging.DEBUG)

app = Flask(__name__)

# Nama file Pickle
PICKLE_FILE = 'dhp_model.pkl'

# Fungsi untuk memuat data dari file Pickle
def load_data_from_pickle(pickle_file):
    if os.path.exists(pickle_file):
        with open(pickle_file, 'rb') as file:
            return pickle.load(file)
    return None, None
```

Gambar 8 Menghubungkan *Flask* Dengan Model (app.py)

Untuk menjalankan *website* yang telah dibuat sebelumnya, file app.py harus dijalankan terlebih dahulu dengan cara masuk ke dalam direktori yang menyimpan file app.py dan ketik "python app.py" seperti ditunjukkan pada gambar 9.

```
PS C:\Users\Nabila Nuraini\Flask_app> python app.py
* Serving Flask app 'app'
* Debug mode: on
INFO:werkzeug:WARNING: This is a development server.
* Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug: * Restarting with stat
WARNING:werkzeug: * Debugger is active!
```

Gambar 9 Menjalankan *Flask*

4. HASIL

a. Tampilan Input

Gambar 10 merupakan tampilan *input* pada sistem yang telah dibuat, untuk proses pencarian pola pembelian obat diperlukan beberapa data, yaitu file csv yang berisi data transaksi, tanggal transaksi yang ingin dianalisis, nilai minimum *support*, dan nilai minimum *confidence*. Seluruh data akan diproses ke dalam *machine learning*.



Gambar 10 Tampilan *Input* Pada Sistem

b. Tampilan Output

[AMLODIPINE 5 MG (HJ)]	32
[CARBIDU 0,75 MG]	61
[GLIMEPIRIDE 2 (DEXA)]	27
[LERZIN KAPS]	29
[PIROXICAM 20 MG]	95
[SALBUTAMOL 4 MG]	32
[ASAM MEF (HJ)]	44
[SANMOL SY 60 ML]	33
[BIOMEGA, RENADINAC 50 MG]	31
[MOLACORT 0,75 MG, RENADINAC 50 MG]	34
[MOLACORT 0,75 MG, PIROXICAM 20 MG]	51

Association Rules

Antecedent	Consequent	Support	Confidence	Lift Ratio
[BIOMEGA]	[RENADINAC 50 MG]	0.012	0.63	16.95

Gambar 11 Tampilan *Output* Pada Sistem

Gambar 11 merupakan tampilan *output* pada sistem yang telah dibuat, untuk mencari pola pembelian obat perlu meng-*input* beberapa data. Setelah dilakukan proses *input* data, maka pola pembelian obat akan terlihat dibawah *form input*.

5. KESIMPULAN

Berdasarkan dari pembahasan yang telah dilakukan, didapatkan beberapa kesimpulan, yaitu:

1. Penelitian ini menunjukkan bahwa algoritma DHP dapat digunakan secara efektif untuk menganalisis data transaksi. Algoritma ini mampu menemukan pola-pola penting dari data transaksi yang besar.
2. Pola pembelian obat yang dianalisis menggunakan algoritma DHP menghasilkan 1 aturan asosiasi dengan menggunakan nilai minimum *support* sebesar 0.3% dan nilai minimum sebesar *confidence* 70%.

UCAPAN TERIMA KASIH

Penulisan penelitian ini tidak akan selesai tanpa bantuan dari orang-orang di sekeliling penulis yang senantiasa mendukung dan membantu. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu terlaksananya penelitian.

DAFTAR PUSTAKA

- [1] S. Nurajizah, "Analisa Transaksi Penjualan Obat menggunakan Algoritma Apriori," *Jurnal Inovtek Polbeng - Seri Informatika*, vol. 4, pp. 35–44, 2019.
- [2] T. Maslihatin, M. Sulehu, and Darmansyah, "Sistem Asosiasi Penyusunan Obat Pada Apotek Balai Rehabilitasi Badan Narkotika Nasional Baddoka Menggunakan Algoritma Apriori," *Celebes Computer Science Journal*, vol. 2, no. 2, pp. 27–38, 2020.
- [3] S. Fatmawati, H. Fajarini, and R. Ferry Balfas, "Persepsi Masyarakat Desa Parereja tentang Pembelian Obat di Apotek," *JURNAL PENELITIAN MULTIDISIPLIN MAHASISWA*, vol. 1, no. 1, pp. 170–185, Jan. 2024.
- [4] D. I. Risqiyana and N. Oktaviani, "Gambaran Pengetahuan Masyarakat Tentang Obat Paten dan Obat Generik di Desa Simbang Kulon Kecamatan Buaran Kabupaten Pekalongan," *ULIL ALBAB: Jurnal Ilmiah Multidisiplin*, vol. 2, no. 4, pp. 1475–1483, 2023.
- [5] I. Paryudi, A. Ashari, and K. Mustofa, "The Performance of Personality-based Recommender System for Fashion with Demographic Data-based Personality Prediction," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 13, no. 1, pp. 360–368, 2022.
- [6] E. T. Naldy and Andri, "Penerapan Data Mining Untuk Analisis Daftar Pembelian Konsumen Dengan Menggunakan Algoritma Apriori Pada Transaksi Penjualan Toko Bangunan MDN," *Jurnal Nasional Ilmu Komputer*, vol. 2, no. 2, pp. 89–101, May 2021.
- [7] M. Mariko, Kusri, and Sudarmawan, "Perbandingan Algoritma Apriori Dan Algoritma Fp-Growth Untuk Rekomendasi Item Paket Pada Konten Promosi," *EXPLORE*, vol. 11, no. 2, pp. 24–29, 2021.
- [8] Z. Abidin, K. A. Arsyah, and A. Nurdin, "PENERAPAN ALGORITMA APRIORI PADA PENJUALAN SUKU CADANG KENDARAAN RODA DUA (STUDI KASUS: TOKO PRIMA MOTOR SIDOMULYO)," *JURNAL TEKNOINFO*, vol. 16, no. 2, pp. 225–232, Jul. 2022.
- [9] Rupali and G. Gupta, "Apriori Based Algorithms And Their Comparisons," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 7, pp. 79–84, Jul. 2013.
- [10] U. R. Amanda and D. P. Utomo, "Penerapan Data Mining Algoritma Hash Based Pada Data Pemesanan Buah Impor Cv. Green Uni Fruit," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 5, no. 1, pp. 86–93, Oct. 2021, doi: 10.30865/komik.v5i1.3653.
- [11] S. Aggarwal and R. Kaur, "Comparative Study of Various Improved Versions of Apriori Algorithm," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, no. 4, pp. 687–690, Apr. 2013.
- [12] F. Panjaitan, A. Surahman, and T. D. Rosmalasari, "ANALISIS MARKET BASKET DENGAN ALGORITMA HASH-BASED PADA TRANSAKSI PENJUALAN (STUDI KASUS: TB. MENARA)," *Jurnal Teknologi dan Sistem Informasi (JTSI)*, vol. 1, no. 2, pp. 111–119, 2020, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/JTSI>