

TESIS

**ANALISIS PERBANDINGAN KINERJA ALGORITMA APRIORI,
FP-GROWTH DAN ECLAT DALAM MENEMUKAN POLA FREKUENSI
PADA DATASET INA-CBG'S**



Disusun oleh:

Nama : Eka Wahyu Pujlharto
NIM : 22.55.2332
Konsentrasi : Digital Transformation Intelligence

PROGRAM STUDI S2 INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2024

TESIS

**ANALISIS PERBANDINGAN KINERJA ALGORITMA APRIORI,
FP-GROWTH DAN ECLAT DALAM MENEMUKAN POLA FREKUENSI
PADA DATASET INA-CBG'S**

**COMPARATIVE ANALYSIS OF THE PERFORMANCE OF THE
APRIORI, FP-GROWTH AND ECLAT ALGORITHMS IN FINDING
FREQUENCY PATTERNS IN THE INA-CBG'S DATASET**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Eka Wahyu Pujlharto
NIM : 22.55.2332
Konsentrasi : Digital Transformation Intelligence

PROGRAM STUDI S2 INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2024

HALAMAN PENGESAHAN

**ANALISIS PERBANDINGAN KINERJA ALGORITMA APRIORI,
FP-GROWTH DAN ECLAT DALAM MENEMUKAN POLA FREKUENSI
PADA DATASET INA-CBG'S**

**COMPARATIVE ANALYSIS OF THE PERFORMANCE OF THE APRIORI,
FP-GROWTH AND ECLAT ALGORITHMS IN FINDING FREQUENCY
PATTERNS IN THE INA-CBG'S DATASET**

Dipersiapkan dan Disusun oleh

Eka Wahyu Pujiharto

22.55.2332

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 10 Juni 2024

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 10 Juni 2024

Rektor

Prof. Dr. M. Suvanto, M.M.

NIK. 190302001

HALAMAN PERSETUJUAN

**ANALISIS PERBANDINGAN KINERJA ALGORITMA APRIORI, FP-GROWTH
DAN ECLAT DALAM MENEMUKAN POLA FREKUENSI PADA DATASET INA-
CBG'S**

**COMPARATIVE ANALYSIS OF THE PERFORMANCE OF THE APRIORI, FP-
GROWTH AND ECLAT ALGORITHMS IN FINDING FREQUENCY PATTERNS
IN THE INA-CBG'S DATASET**

Dipersiapkan dan Disusun oleh

Eka Wahyu Pujiharto

22.55.2332

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 10 Juni 2024

Pembimbing Utama

Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

Anggota Tim Penguji

Dhani Ariatmanto, M.Kom., Ph.D.
NIK. 190302197

Pembimbing Pendamping

Drs. Asro Nasiri, M.Kom.
NIK. 190302152

Hanafi, S.Kom., M.Eng., PhD
NIK. 190302024

Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 10 Juni 2024
Direktur Program Pascasarjana

Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Eka Wahyu Pujiharto
NIM : 22.55.2332
Konsentrasi : Digital Transformation Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
**Analisis Perbandingan Kinerja Algoritma Apriori, Fp-Growth Dan Eclat
Dalam Menemukan Pola Frekuensi Pada Dataset Ina-Cbg's**

Dosen Pembimbing Utama : Prof. Dr. Kusriani, M.Kom.
Dosen Pembimbing Pendamping : Drs. Asro Nasiri, M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 10 Juni 2024
Yang Menyatakan,



Eka Wahyu Pujiharto

HALAMAN PERSEMBAHAN

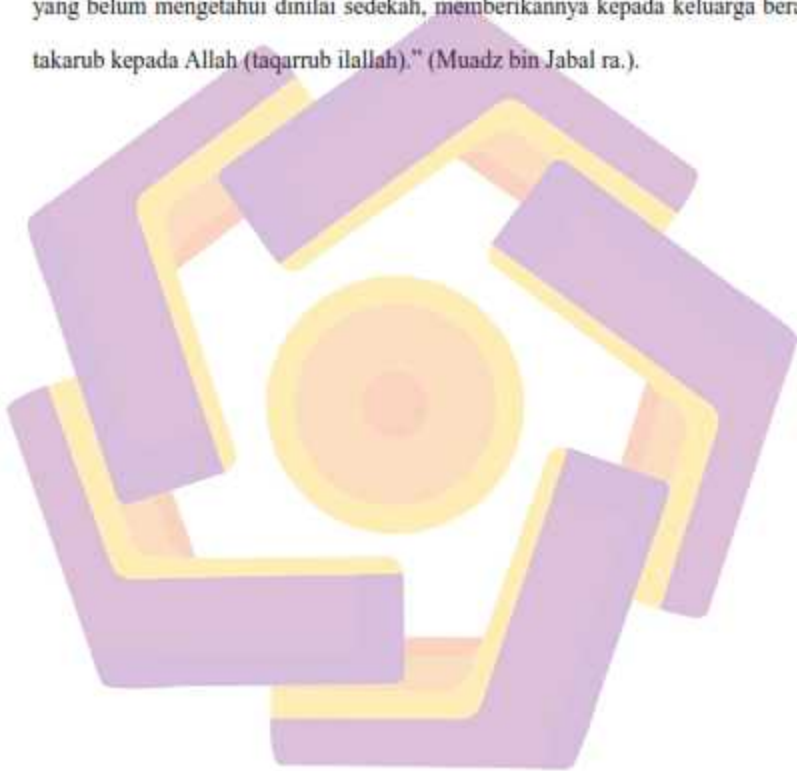
Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, dan shalawat serta salam semoga terlimpah bagi Nabi Muhammad SAW, kepada keluarga, sahabat, dan para pengikutnya sampai akhir zaman.

Alhamdulillah atas dukungan serta doa dari orang-orang tercinta, akhirnya tesis ini dapat terselesaikan dengan baik. Oleh karena itu dengan rasa bangga dan suka cita, tesis ini saya persembahkan untuk :

1. Mamah (lola), Istri (Nur Ika Fidhadia), Anak (Alfarzan Hanif Rizqullaah), dan adik saya (Kurnia Dwi Wahyuni) yang telah memberikan dukungan moral maupun materi serta doa, motivasi dan kesabaran kepada saya.
2. Ibu Prof. Dr. Kusriani, M.Kom., dan Bapak Drs. Asro Nasiri, M.Kom., selaku dosen pembimbing saya yang telah memberikan arahan, motivasi, serta bimbingan positif dalam menyelesaikan tesis ini sehingga tesis ini dapat terselesaikan dengan baik.
3. Sahabat seperjuangan saya yang berharga, Pak Widdi, Mba Elisa yang telah memberikan semangat, arahan dan doa untuk tidak menyerah dalam berjuang.
4. Keluarga besar MTI Amikom, khususnya kelas PJJ MTI Konsentrasi Digital Transformation Intelligence yang telah menjadi teman-teman saya semasa kuliah pasca sarjana. Serta semua pihak yang telah membantu dan mendukung saya yang tidak saya sebutkan satu persatu.

HALAMAN MOTTO

“Belajar ilmu karena Allah adalah takwa, menuntutnya adalah ibadah, mempelajarinya adalah tasbih, membahasnya adalah jihad, mengajarkannya kepada yang belum mengetahui dinilai sedekah, memberikannya kepada keluarga berarti takarub kepada Allah (taqarrub ilallah).” (Muadz bin Jabal ra.).



KATA PENGANTAR

Segala puji bagi Allah SWT, yang telah memberikan segala nikmat yang tak terhingga nilainya kepada hamba-Nya. Alhamdulillahirobbil'alamiin dengan izin Allah SWT, penulis dapat menyelesaikan tesis ini yang berjudul "ANALISIS PERBANDINGAN KINERJA ALGORITMA APRIORI, FP-GROWTH DAN ECLAT DALAM MENEMUKAN POLA FREKUENSI PADA DATASET INACBG'S". Penyusunan tugas akhir ini merupakan salah satu persyaratan kelulusan untuk memperoleh gelar Magister Informatika di Universitas AMIKOM Yogyakarta.

Dengan tersusunnya tesis ini, penulis mengucapkan terima kasih banyak kepada :

1. Orang tua, saudara, anak dan istri tercinta.
2. Bapak Prof. Dr. M. Suyanto, M.M selaku Rektor Universitas AMIKOM Yogyakarta.
3. Ibu Dr. Kusriani, M.Kom selaku Ketua Pascasarjana dan Ketua Program S2 Teknik Informatika Universitas AMIKOM Yogyakarta juga sebagai Dosen Pembimbing Pertama.
4. Bapak Drs. Asro Nasiri, M.Kom selaku Dosen Pembimbing Kedua.
5. Seluruh Dosen Pascasarjana PJJ MTI Universitas AMIKOM Yogyakarta.
6. Teman-teman seperjuangan PJJ MTI Universitas AMIKOM Yogyakarta.

Penulis menyadari jika tesis ini masih jauh dari kata sempurna, semoga tesis ini bermanfaat tidak hanya bagi penulis melainkan juga bermanfaat bagi pembaca dan dapat memberikan inspirasi untuk pengembangan penelitian selanjutnya.

Yogyakarta, 10 Juni 2024

Penulis

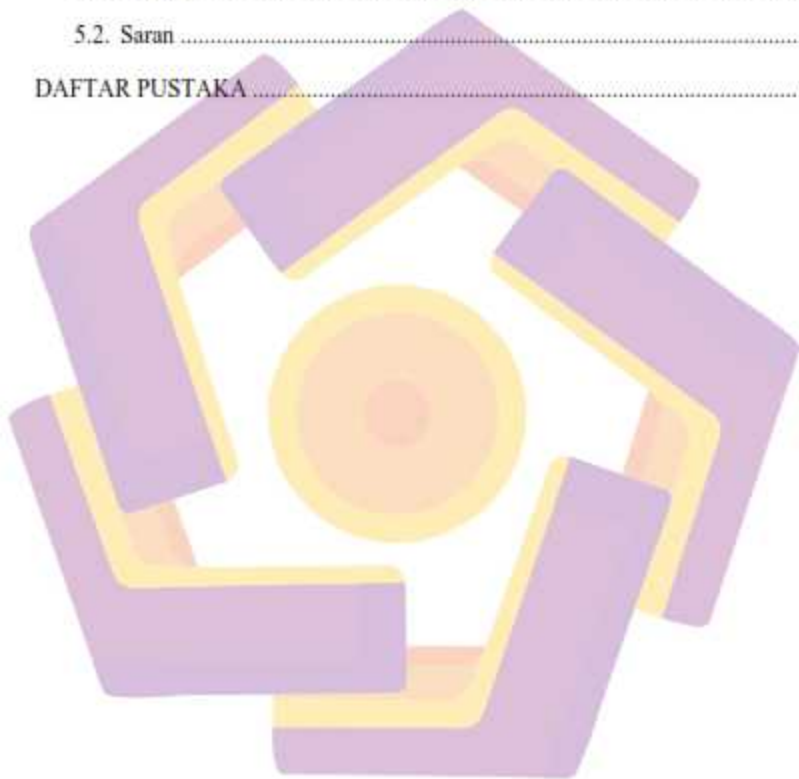


DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
INTISARI.....	xv
<i>ABSTRACT</i>	xvi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	7
1.3. Batasan Masalah.....	7
1.4. Tujuan Penelitian.....	7
1.5. Manfaat Penelitian.....	8
BAB II TINJAUAN PUSTAKA.....	9
2.1. Tinjauan Pustaka.....	9
2.2. Keaslian Penelitian.....	16

2.3. Landasan Teori.....	22
2.3.1. Analisis Asosiasi (Association Rule Mining).....	22
2.3.2. Metodologi Dasar Aturan Asosiasi.....	22
2.3.3. Algoritma Apriori.....	24
2.3.4. Algoritma FP-Growth.....	27
2.3.5. Algoritma Equivalence Class Transformation (ECLAT).....	28
2.3.6. Menghitung performance algoritma.....	30
BAB III METODE PENELITIAN.....	32
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	32
3.2. Metode Pengumpulan Data.....	32
3.3. Metode Analisis Data.....	34
3.4. Alur Penelitian.....	35
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	40
4.1. Persiapan Data.....	40
4.1.1. Preprocessing Data.....	40
4.1.2. Pemanggilan Dataset.....	42
4.2. Perancangan dan Pemodelan Sistem.....	43
4.2.1. Pola Frekuensi Tinggi.....	43
4.2.2. Pembentukan Aturan Asosiatif.....	50
4.2.3. Analisa Aturan Asosiatif.....	60
4.3. Analisis Perbandingan Kinerja Algoritma.....	62

4.3.1. Perbandingan Waktu Proses	62
4.3.2. Perbandingan Kapasitas Memori	69
BAB V PENUTUP	77
5.1. Kesimpulan	77
5.2. Saran	78
DAFTAR PUSTAKA	79



DAFTAR TABEL

Tabel 2.1. Matriks Literatur Review dan Posisi Penelitian.....	16
Tabel 2.2. Proses Perhitungan Algoritma Apriori.....	26
Tabel 3.1. Variabel Data INACBG's.....	37
Tabel 4.1. Hasil Perhitungan Nilai Support pada Itemset.....	49
Tabel 4.2 Hasil Aturan Asosiatif Algoritma Apriori.....	51
Tabel 4.3. Hasil Aturan Asosiasi Algoritma Apriori Nilai Confidence > 0.6.....	52
Tabel 4.4. Hasil Aturan Asosiatif Algoritma FP-Growth.....	54
Tabel 4.5. Hasil Aturan Asosiatif Algoritma FP-Growth Nilai Confidence > 0.6	55
Tabel 4.6. Hasil Aturan Asosiatif Algoritma ECLAT.....	57
Tabel 4.7. Hasil Aturan Asosiatif Algoritma ECLAT Nilai Confidence > 0.6.....	58
Tabel 4.8. Hasil Aturan Asosiatif Ketiga Algoritma.....	60
Tabel 4.9. Hasil Aturan Asosiatif Algoritma Nilai Support > 0.02 dan Nilai Confidence > 0.6.....	60
Tabel 4.10. Hasil Perhitungan Waktu Proses Algoritma Apriori.....	63
Tabel 4.11. Hasil Perhitungan Waktu Proses Algoritma FP-Growth.....	64
Tabel 4.12. Hasil Perhitungan Waktu Proses Algoritma ECLAT.....	66
Tabel 4.13. Tabel Pengujian Algoritma.....	67
Tabel 4.14. Hasil Perhitungan Kapasitas Memori Algoritma Apriori.....	70
Tabel 4.15. Hasil Perhitungan Kapasitas Memori Algoritma FP-Growth.....	71
Tabel 4.16. Hasil Perhitungan Kapasitas Memori Algoritma ECLAT.....	72

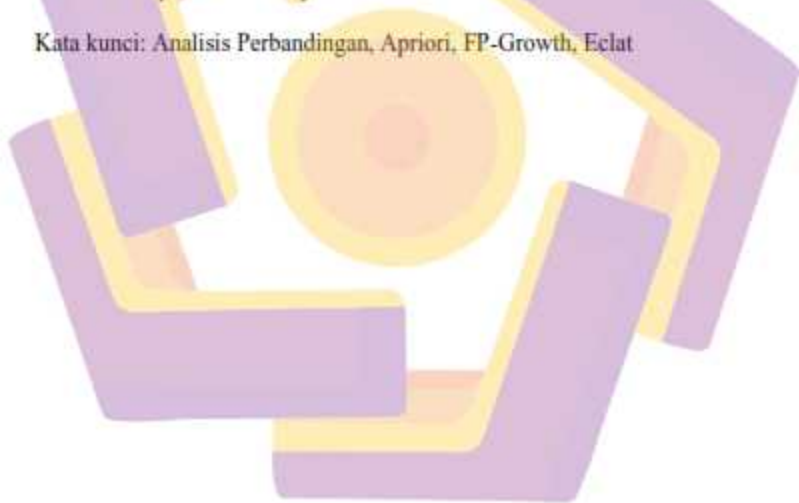
DAFTAR GAMBAR

Gambar 3.1. Alur Penelitian.....	36
Gambar 4.1. File inacbg-ranap-2022.txt.....	40
Gambar 4.2. Proses Tahap 1 Konversi TXT ke Sheet Excel.....	41
Gambar 4.3. Proses Tahap 2 Konversi TXT ke Sheet Excel.....	41
Gambar 4.4. File Sheet Excel Hasil Konversi.....	41
Gambar 4.5. Isi Dataset DIAGLIST.....	42
Gambar 4.6. Info Dataset DIAGLIST.....	43
Gambar 4.7. Matrix Itemset.....	44
Gambar 4.8. Frekuensi Itemset Algoritma FP-Growth.....	45
Gambar 4.9. Frekuensi Itemset Algoritma Apriori.....	46
Gambar 4.10. Frekuensi Itemset Algoritma ECLAT.....	48
Gambar 4.11. Grafik Rata-rata Waktu Proses Algoritma.....	66
Gambar 4.12. Grafik Rata-rata Penggunaan Memori pada Proses Algoritma.....	73

INTISARI

Setiap fasilitas kesehatan seperti rumah sakit, klinik dan puskesmas yang bekerjasama dengan BPJS wajib melakukan klaim pembiayaan atas perawatan kesehatan terhadap pasien menggunakan tarif INACBG's (Indonesian - Case Based Groups). Tarif INACBG's merupakan paket layanan yang didasarkan kepada pengelompokan diagnosa penyakit yang menggunakan kode ICD-10. Penelitian ini bertujuan menemukan pola frekuensi pada dataset INA-CBG's terutama kombinasi diagnosa agar diketahui kombinasi diagnosa apa saja yang sering muncul untuk bahan evaluasi lebih lanjut oleh pihak manajemen fasilitas kesehatan. Penelitian ini membandingkan kinerja Algoritma Apriori, FP-Growth dan Eclat. Nilai Akurasi Lift Ratio dan Rule Asosiasi ketiga algoritma didapatkan nilai yang sama, tetapi untuk waktu komputasi dan pemakaian memori pada Algoritma Eclat lebih banyak daripada Algoritma Apriori dan Fp-Growth, maka dapat disimpulkan bahwa Algoritma FP-Growth dan Apriori lebih cocok untuk dijadikan solusi dalam menemukan pola frekuensi pada dataset INACBG's.

Kata kunci: Analisis Perbandingan, Apriori, FP-Growth, Eclat



ABSTRACT

Every health facility such as hospitals, clinics and community health centers that collaborate with BPJS is required to make funding claims for health care for patients using INACBG's (Indonesian - Case Based Groups) rates. INACBG's tariff is a service package that is based on disease diagnosis groupings using ICD-10 codes. This research aims to find frequency patterns in the INA-CBG's dataset, especially combinations of diagnoses, in order to find out what combinations of diagnoses frequently appear for further evaluation by health facility management. This research compares the performance of the Apriori, FP-Growth and Eclat algorithms. The accuracy values of the Lift Ratio and Rule Association of the three algorithms obtained the same value, but the computing time and memory usage of the Eclat Algorithm is more than the Apriori and Fp-Growth Algorithms, so it can be concluded that the FP-Growth and Apriori Algorithms are more suitable to be used as solutions in found frequency patterns in INACBG's dataset.

Keyword: Comparative Analysis, Apriori, FP-Growth, Eclat



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pada era digital dan informasi yang semakin berkembang, jumlah data yang dihasilkan oleh berbagai organisasi dan entitas semakin melonjak dengan cepat. Fenomena ini dikenal sebagai "ledakan data" atau "data explosion". Di tengah kelimpahan data ini, organisasi di berbagai sektor, termasuk bisnis, kesehatan, pendidikan, dan lainnya, mencoba untuk mengambil manfaat maksimal dari data mereka. Salah satu cara untuk mendapatkan wawasan berharga dari data adalah dengan menemukan pola frekuensi atau pola-pola yang muncul secara konsisten dalam data.

Pada bidang kesehatan, Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan menggunakan tarif INA-CBG's (Indonesia Case Based Groups) pada pola pembayaran Jaminan Kesehatan Nasional (JKN). Ketentuan ini sesuai dengan Peraturan Presiden No 111 Tahun 2013 sebagai revisi dari Perpres No 12 Tahun 2013 mengenai Jaminan Kesehatan.

Setiap fasilitas kesehatan seperti rumah sakit, klinik dan puskesmas yang bekerjasama dengan BPJS wajib melakukan klaim pembiayaan atas perawatan kesehatan terhadap pasien menggunakan tarif INACBG's (Indonesian - Case Based Groups). Tarif INACBG's ini merupakan besaran pembayaran klaim oleh BPJS Kesehatan kepada Fasilitas Kesehatan Tingkat Lanjutan atas paket layanan

yang didasarkan kepada pengelompokan diagnosis penyakit (Permenkes RI Nomor 69 Tahun 2013).

Pola pembayaran INACBG's ini menggunakan sistem paket, sehingga fasilitas kesehatan dituntut untuk lebih mengoptimalkan biaya pelayanan. Semua biaya pelayanan baik itu tindakan medis, tindakan non medis, obat, penunjang, dan biaya perawatan lainnya yang dikeluarkan oleh fasilitas kesehatan dikemas dalam satu paket pembayaran berdasarkan group diagnosa dengan sistem grouping tarif pada sistem klaim INACBG's.

Kendala yang terjadi pada fasilitas kesehatan dengan adanya sistem pembayaran klaim biaya pasien menggunakan INACBG's adalah fasilitas kesehatan mengalami kesulitan untuk menganalisa data klaim, terkait penetapan diagnosa, prosedur tindakan dan variabel lainnya. Tarif INACBGs menggunakan sistem koding dengan kode ICD-10 untuk diagnosis serta kode ICD-9-CM untuk prosedur/tindakan. Pengelompokan kode diagnosis dan prosedur dilakukan dengan menggunakan grouper UNU (UNU Grouper). UNU-Grouper adalah Grouper casemix yang dikembangkan oleh United Nations University/UNU (Permenkes Nomor 69 Tahun 2013). Kombinasi dari kode diagnosa, prosedur, dan variabel lain sangat menentukan Tarif INACBG's, ketidak tepatan penentuan diagnosa dan prosedur akan mengakibatkan kerugian dikarenakan terlalu besar biaya yang dikeluarkan dibandingkan dengan pembayaran klaim dengan tarif grouping hasil dari sistem INACBG's.

Data mining adalah proses untuk mendapatkan informasi yang berguna dari basis data yang besar dan perlu diekstraksi agar menjadi informasi baru dan dapat

membantu dalam pengambilan keputusan (Suntoro, 2019). Algoritma data mining menjadi sangat penting dalam menganalisis data untuk mengidentifikasi pola-pola kombinasi kode diagnosa dan kode prosedur ini. Pola-pola ini dapat mengungkap hubungan yang tidak terlihat secara langsung antara berbagai variabel dalam data. Algoritma untuk mengenali pola dari data yaitu algoritma asosiasi, ada tiga algoritma asosiasi yang umum digunakan diantaranya adalah Apriori, FP-Growth, dan ECLAT.

Telah dilakukan beberapa penelitian yang meneliti tentang algoritma asosiasi yaitu diantaranya, penelitian pertama yang berjudul "*Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores*", penelitian ini bertujuan membandingkan metode Apriori, Apriori TID, dan FP-Growth untuk menentukan perilaku transaksi konsumen. Penelitian ini menggunakan data yang diperoleh dari database Market Basket Analysis untuk menentukan jumlah transaksi yang dilakukan oleh konsumen selama beberapa musim. Hasil penelitian menunjukkan bahwa FP-Growth menggunakan memori yang lebih banyak daripada kedua algoritma lainnya, tetapi memiliki performa terbaik. Algoritma Apriori-TID lebih cepat dan menggunakan memori yang lebih ringan..

Penelitian kedua adalah penelitian yang berjudul "*Evaluation of Apriori, FP growth and Eclat association rule mining algorithms*". Penelitian ini bertujuan menemukan hubungan menarik antara entitas dalam catatan penjualan yang besar dan dapat membantu dalam berbagai proses pengambilan keputusan bisnis, seperti analisis perilaku belanja pelanggan, pemasaran, dan perancangan persediaan.

Penelitian ini mengevaluasi tiga algoritma (Apriori, FP Growth, dan Eclat) untuk melakukan "mining rule association" dengan himpunan entitas yang umum. Penulis memeriksa ketiga algoritma ini dengan melihat jumlah transaksi, dukungan minimum, penggunaan memori, dan waktu eksekusi.

Penelitian ketiga adalah penelitian yang berjudul "*Discovering Future Earnings Patterns through FP-Growth and ECLAT Algorithms with Optimized Discretization*". Penelitian ini bertujuan untuk menemukan pola pendapatan masa depan terkait dari laporan keuangan perusahaan yang terdaftar di Bursa Efek Thailand. Algoritma FP-Growth dan Eclat digunakan untuk mengidentifikasi pola pendapatan masa depan yang sangat terkait.

Penelitian keempat, "*Determination of the best rule-based analysis results from the comparison of the Fp-Growth, Apriori, and TPQ-Apriori Algorithms for recommendation systems*", membandingkan tiga algoritma ini dengan dataset yang sama untuk menemukan dan memastikan pembentukan aturan yang optimal dan konsisten. Dataset koperasi pegawai Telkom NTB, yang dapat diunduh di situs Kaggle, digunakan untuk melakukan pengujian. Pada penggunaan algoritma fp-growth RapidMiner, hasilnya akan lebih optimal, tetapi akan lebih buruk jika dataset yang diolah kecil. Tidak seperti algoritma FP-growth Apriori pada Weka, di mana rule yang dihasilkan lebih maksimal jika dataset yang digunakan besar dan optimal jika dataset yang digunakan kecil. Untuk membandingkan, algoritma fp-growth Apriori tradisional dan TPQ-Apriori, Apriori tradisional, FP-growth, dan TPQ-Apriori digunakan. Untuk pengujian rule, hasil rule TPQ-Apriori dibandingkan dengan algoritma Fp-Growth pada alat Rapidminer dan Weka, serta

algoritma Apriori pada alat Weka. Hasil pengujian menunjukkan bahwa TPQ-Apriori memiliki kemampuan konsistensi dan optimalisasi yang lebih besar.

Penelitian kelima, "Penerapan Association Rule Dalam Bidang Kesehatan Pada Rumah Sakit X", Penelitian ini bertujuan untuk menemukan association rule mengenai penyakit Diabetes Melitus berdasarkan karakteristik pasien, gejala dan diagnosis penyakit berdasarkan data yang didapat pada Rumah Sakit X. Data yang diambil adalah 100 data riwayat penyakit pasien. Pengolahan data menggunakan software Rapid Miner dengan algoritma FP-Growth. Diperoleh 12 rule yang valid dari 13 rule yang ditemukan. Association rule yang ditemukan dapat digunakan sebagai informasi pendukung terkait diagnosis dan analisis penyakit untuk segera dicari tindakan yang tepat dalam penanganan pasien berdasarkan histori penyakit pasien yang pernah berobat di Rumah Sakit X.

Penelitian keenam di bidang kesehatan yang menggunakan algoritma asosiasi berjudul "Penerapan Association Rule Menggunakan Algoritma Apriori Pada Poliklinik Penyakit Dalam (Studi Kasus: Rumah Sakit Umum Daerah Bintang)". Untuk menemukan aturan asosiasi berdasarkan jenis kelamin, umur, keluhan, dan kode ICD10, penulis menggunakan Algoritma Apriori. Hasil aturan asosiasi adalah 22 dengan data 1668 dan minimal dukungan 6% dan kepercayaan 50%. Penelitian ini menghasilkan aturan asosiasi yang dapat digunakan oleh rumah sakit dan dinas kesehatan untuk melakukan sosialisasi dan penyuluhan kesehatan kepada masyarakat.

Salah satu tahap analisis asosiasi yang menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien adalah analisis pola frekuensi tinggi

(frequent pattern mining). Penting tidaknya suatu asosiasi dapat diketahui dengan dua tolak ukur, yaitu: support dan confidence. Support (nilai penunjang) adalah persentase kombinasi item tersebut dalam database, sedangkan confidence (nilai kepastian) adalah kuatnya hubungan antar-item dalam aturan asosiasi.

Dengan menganalisis data INACBG's diharapkan dapat menghasilkan pola keterhubungan antara variabel kode diagnosa dengan kode prosedur. Pola asosiasi ini dapat diperoleh dengan menggunakan metode association rule. Penambangan aturan asosiasi adalah cara untuk menemukan hubungan antara item dalam kumpulan data. Aturan asosiasi terdiri dari dua fase, yaitu: menemukan kombinasi item yang sering muncul dari kumpulan item yang ada, dan mendefinisikan condition dan result untuk aturan conditional association rule.

Proses akhir dari penambangan aturan asosiasi yaitu menunjukkan kekuatan hubungan asosiasi dari item set dan pembentukan hasil akhir asosiasi menggunakan lift ratio. Lift Ratio merupakan suatu ukuran (parameter) untuk mengetahui kekuatan aturan asosiasi (association rule) yang telah terbentuk dari nilai support dan confidence. Nilai lift ratio digunakan sebagai penentu apakah aturan asosiasi valid atau tidak valid (Santosa, 2007).

Dari latar belakang masalah ini peneliti akan melakukan analisis perbandingan kinerja tiga algoritma asosiasi yaitu Apriori, FP-Growth dan Eclat untuk menemukan pola frekuensi data pada dataset INA-CBG's. Dengan membandingkan performa Apriori, FP-Growth, dan ECLAT, untuk mengetahui tingkat dari kepercayaan dan keakuratan suatu data yang dihasilkan dan untuk mengukur nilai kepercayaan dari suatu kode diagnosa dan kode prosedur.

1.2. Rumusan Masalah

Berdasarkan pemaparan latar belakang masalah di atas, dapat diperoleh perumusan masalah dari penelitian ini adalah sebagai berikut:

- a. Berapa nilai confidence dan conviction yang dihasilkan dari masing-masing metode algoritma yang diterapkan.
- b. Berdasarkan analisis perhitungan waktu proses dan kapasitas memori yang digunakan algoritma Apriori, FP-Growth dan Eclat, algoritma mana yang lebih baik untuk digunakan dalam menemukan pola frekuensi dan aturan asosiasi pada dataset INACBG's?.

1.3. Batasan Masalah

Berikut adalah batasan-batasan yang diterapkan dalam penelitian ini:

- a. Penelitian ini menggunakan dataset INA-CBG's pasien rawat inap di RSD Gunung Jati Kota Cirebon.
- b. Membandingkan tiga algoritma asosiasi yaitu *apriori*, *fp-growth*, dan *eclat* untuk mengukur tingkat kepercayaan rules yang dihasilkan tiap algoritma.
- c. Hasil penelitian mengetahui nilai kepercayaan (*confidence*) dan nilai *Lift Ratio* yang dihasilkan tiap algoritma.

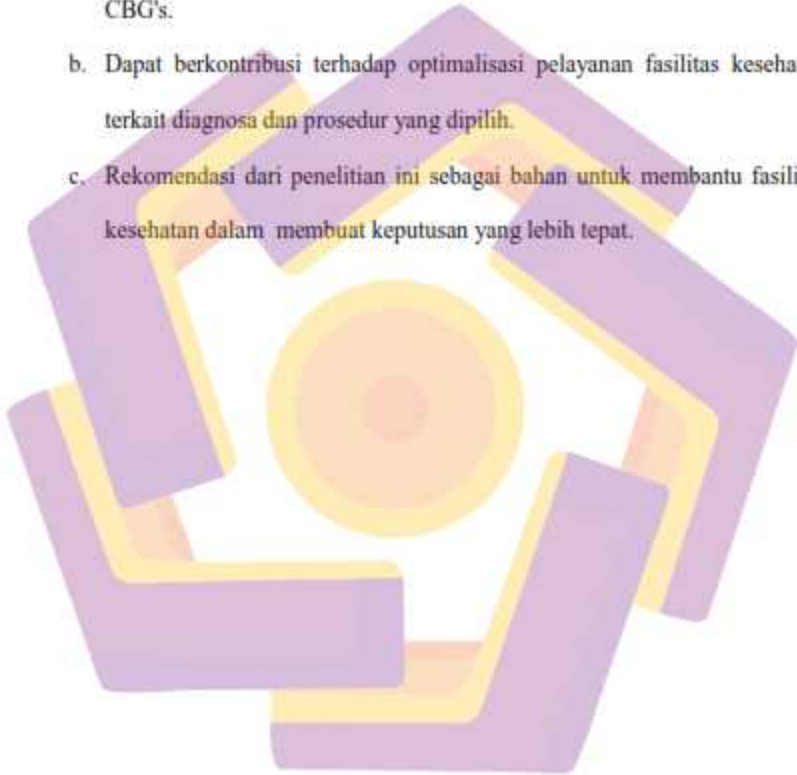
1.4. Tujuan Penelitian

Maksud dan tujuan dari penelitian ini adalah untuk melakukan perbandingan kinerja algoritma Apriori, FP-Growth, dan ECLAT dalam menemukan pola frekuensi pada dataset INA-CBG's serta waktu yang dibutuhkan dalam pemrosesan data.

1.5. Manfaat Penelitian

Berikut adalah beberapa manfaat dari penelitian ini:

- a. Memberikan panduan untuk peneliti selanjutnya dalam memilih algoritma yang paling sesuai untuk analisis asosiasi pada dataset serupa dengan INA-CBG's.
- b. Dapat berkontribusi terhadap optimalisasi pelayanan fasilitas kesehatan terkait diagnosa dan prosedur yang dipilih.
- c. Rekomendasi dari penelitian ini sebagai bahan untuk membantu fasilitas kesehatan dalam membuat keputusan yang lebih tepat.



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Beberapa penelitian yang berkaitan dengan topik penelitian yang diambil adalah sebagai berikut :

Penelitian pertama, "*Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores*" mencoba menemukan pola frekuensi data melalui data mining. Pada penelitian ini, pola pembelian sembako di toko sembako digunakan, dan kemudian dianalisis berdasarkan musim. Penelitian ini membandingkan metode Apriori, Apriori TID, dan FP-Growth untuk menentukan perilaku transaksi konsumen. Penelitian ini menggunakan data yang diperoleh dari database Market Basket Analysis untuk menentukan jumlah transaksi yang dilakukan oleh konsumen selama beberapa musim. Hasil penelitian menunjukkan bahwa FP-Growth menggunakan memori yang lebih banyak daripada kedua algoritma lainnya, tetapi memiliki performa terbaik. Algoritma Apriori-TID lebih cepat dan menggunakan memori yang lebih ringan. Pada penelitian ini masih ada kekurangan yaitu terbatasnya jumlah sampel data yang akan mempengaruhi validitas dan generalisasi hasil penelitian. Penelitian ini juga hanya terbatas meneliti satu variabel atau aspek tertentu, sehingga tidak mencakup semua faktor yang relevan dalam analisis.

Penelitian kedua adalah penelitian yang berjudul "*Evaluation of Apriori, FP growth and Eclat association rule mining algorithms*". Penelitian ini menemukan

hubungan menarik antara entitas dalam catatan penjualan yang besar dan dapat membantu dalam berbagai proses pengambilan keputusan bisnis, seperti analisis perilaku belanja pelanggan, pemasaran, dan perancangan persediaan. Penelitian ini mengevaluasi tiga algoritma (Apriori, FP Growth, dan Eclat) untuk melakukan "mining rule association" dengan himpunan entitas yang umum. Penulis memeriksa ketiga algoritma ini dengan melihat jumlah transaksi, dukungan minimum, penggunaan memori, dan waktu eksekusi. Data eksperimental digunakan untuk melakukan evaluasi algoritma ini. Hasilnya adalah kesimpulan dan saran tentang keefektifan dan efisiensi masing-masing algoritma dalam konteks yang dijelaskan. Pada penelitian ini kurang menggambarkan tujuan dari penelitian, apa yang ingin dicapai dengan hasil perolehan aturan-aturan afiliasi dalam kumpulan data besar ini, juga hasil dari perhitungan kinerja ketiga algoritma asosiasi tidak memberikan rincian tentang bagaimana perhitungan ini digunakan.

Penelitian ketiga adalah penelitian yang berjudul "*Discovering Future Earnings Patterns through FP-Growth and ECLAT Algorithms with Optimized Discretization*". Penelitian ini bertujuan untuk menemukan pola pendapatan masa depan terkait dari laporan keuangan perusahaan yang terdaftar di Bursa Efek Thailand. Enam puluh delapan deskriptor akuntansi memberikan dasar untuk memperkirakan probabilitas suatu kenaikan atau penurunan laba pada tahun anggaran berikutnya. Sebelum menghasilkan pola terkait dari akuntansi deskriptor, data numerik diubah menjadi rentang diskrit optimal menggunakan teknik diskritisasi yang dioptimalkan. Algoritma FP-Growth dan Eclat digunakan untuk mengidentifikasi pola pendapatan masa depan yang sangat terkait. Pada penelitian

ini bertujuan untuk mengidentifikasi pola pendapatan masa depan yang berbeda dari penelitian lain. Namun, tidak ada penjelasan yang cukup tentang tujuan spesifik penelitian ini atau masalah yang ingin dipecahkan. Pada penelitian ini telah ditemukan empat peningkatan pola pendapatan dan sembilan penurunan yang terkait erat. Namun, tidak ada penjelasan lebih lanjut atau contoh konkret dari temuan ini.

Penelitian keempat, *"Determination of the best rule-based analysis results from the comparison of the Fp-Growth, Apriori, and TPQ-Apriori Algorithms for recommendation systems"*, membandingkan tiga algoritma ini dengan dataset yang sama untuk menemukan dan memastikan pembentukan aturan yang optimal dan konsisten. Dataset koperasi pegawai Telkom NTB, yang dapat diunduh di situs Kaggle, digunakan untuk melakukan pengujian. Dengan mempertimbangkan hasil pengujian aturan dasar dengan dataset Kopegtel sebesar seratus persen, lima puluh persen, dan dua puluh persen dari total volume dataset, dapat disimpulkan bahwa dataset yang lebih besar akan diolah. Jika menggunakan algoritma fp-growth RapidMiner, hasilnya akan lebih optimal, tetapi akan lebih buruk jika dataset yang diolah kecil. Tidak seperti algoritma FP-growth Apriori dan Weka, di mana rule yang dihasilkan lebih maksimal jika dataset yang digunakan besar dan optimal jika dataset yang digunakan kecil. Untuk membandingkan, algoritma fp-growth Apriori tradisional dan TPQ-Apriori, Apriori tradisional, FP-growth, dan TPQ-Apriori digunakan. Untuk pengujian rule, hasil rule TPQ-Apriori dibandingkan dengan algoritma Fp-Growth pada alat Rapidminer dan Weka, serta algoritma Apriori pada alat Weka. Hasil pengujian menunjukkan bahwa TPQ-Apriori memiliki

kemampuan konsistensi dan optimalisasi yang lebih besar. Kekurangan dari penelitian ini adalah kurangnya detail tentang dataset: Meskipun penelitian ini menyebutkan bahwa dataset telah melalui tahap preprocessing, namun tidak dijelaskan secara detail tentang karakteristik dataset, seperti jumlah atribut, tipe data, dan distribusi nilai-nilai dalam dataset. Informasi ini penting untuk memahami konteks dan relevansi hasil penelitian. Pada penelitian ini tidak adanya evaluasi kinerja yang komprehensif: Penelitian ini hanya menggunakan metrik-metrik sederhana seperti support, confidence, dan lift untuk mengevaluasi kinerja algoritma. Tidak ada evaluasi kinerja yang lebih komprehensif, seperti waktu eksekusi, penggunaan memori, atau analisis kompleksitas algoritma.

Penelitian kelima, "Penerapan Association Rule Dalam Bidang Kesehatan Pada Rumah Sakit X", Penelitian ini bertujuan untuk menemukan association rule mengenai penyakit Diabetes Melitus berdasarkan karakteristik pasien, gejala dan diagnosis penyakit berdasarkan data yang didapat pada Rumah Sakit X. Data yang diambil adalah 100 data riwayat penyakit pasien. Pengolahan data menggunakan software Rapid Miner dengan algoritma FP-Growth. Diperoleh 12 rule yang valid dari 13 rule yang ditemukan. Association rule yang ditemukan dapat digunakan sebagai informasi pendukung terkait diagnosis dan analisis penyakit untuk segera dicari tindakan yang tepat dalam penanganan pasien berdasarkan histori penyakit pasien yang pernah berobat di Rumah Sakit X. Dengan prosedur yang telah dilakukan menggunakan software Rapid Miner didapatkan 12 aturan valid dimana memiliki nilai lift ratio > 1 yang digunakan sebagai acuan untuk mendapatkan sebuah knowledge. Aturan ini dapat digunakan sebagai informasi pada rumah sakit

terkait diagnosis yang sering terjadi pada pasien sehingga pihak rumah sakit dapat langsung mengambil tindakan yang tepat kepada pasien. Penelitian ini mengalami kekurangan yaitu variabel diagnosa yang digunakan hanya 1 diagnosa, tidak menyertakan diagnosa sekunder atau diagnosa penyerta atau komplikasi lainnya. Dataset yang digunakan terbatas hanya 100 data riwayat pasien, yang akan mempengaruhi dari hasil perhitungan asosiasi rule.

Penelitian keenam di bidang kesehatan yang menggunakan algoritma asosiasi berjudul "Penerapan Association Rule Menggunakan Algoritma Apriori Pada Poliklinik Penyakit Dalam (Studi Kasus: Rumah Sakit Umum Daerah Bintang)". Untuk menemukan aturan asosiasi berdasarkan jenis kelamin, umur, keluhan, dan kode ICD10, penulis menggunakan Algoritma Apriori. Hasil aturan asosiasi adalah 22 dengan data 1668 dan minimal dukungan 6% dan kepercayaan 50%. Penelitian ini menghasilkan aturan asosiasi yang dapat digunakan oleh rumah sakit dan dinas kesehatan untuk melakukan sosialisasi dan penyuluhan kesehatan kepada masyarakat. Penelitian ini tidak mempertimbangkan komplikasi diagnosa penyerta, hanya menggunakan satu variabel diagnosa utama.

Pada bidang kesehatan, Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan menggunakan tarif INA-CBG's (*Indonesia Case Based Groups*) versi pertama yakni versi 4.0 pada pola pembayaran Jaminan Kesehatan Nasional (JKN). Versi ini mulai diberlakukan pada 2014. Ketentuan ini sesuai dengan Peraturan Presiden No 111 Tahun 2013 sebagai revisi dari Perpres No 12 Tahun 2013 mengenai Jaminan Kesehatan.

Setiap Fasilitas Kesehatan baik itu Rumah Sakit, Puskesmas, Klinik yang bekerjasama dengan BPJS menggunakan pola pembayaran INACBG's. Pola Pembayaran INACBG merupakan pola pembayaran dengan 1.077 kelompok tarif yang terdiri dari 789 kode group/kelompok rawat inap dan 288 kode group/kelompok rawat jalan, dengan menggunakan sistem kode ICD-10 untuk diagnosa dan sistem kode ICD-9-CM untuk prosedur/ tindakan. UNU Grouper (UNU Grouper) digunakan untuk mengelompokkan diagnosis dan kode prosedur. Dikembangkan oleh United Nations University (UNU), UNU-Grouper adalah versi casemix dari Grouper.

Dataset INACBG's merupakan dataset yang dimiliki oleh setiap Fasilitas Kesehatan baik itu Rumah Sakit maupun Puskesmas dan Klinik, dengan dataset ini diharapkan Fasilitas Kesehatan dapat menganalisa data pelayanan kesehatan untuk menemukan pola frekuensi atau pola-pola data yang muncul secara konsisten dalam data kesehatan tersebut. Sementara itu belum ada penelitian yang memfokuskan tentang pengenalan pola-pola frekuensi pada dataset INACBG's.

Tantangan yang ada pada pengolahan dataset ini adalah, kode diagnose ICD10 memiliki jumlah total kode diagnosa di ICD-10 lebih dari 70.000 dan tidak dapat ditentukan secara pasti. Hal ini dikarenakan beberapa alasan :

1. Struktur berjenjang : ICD-10 memiliki struktur kode yang berjenjang, di mana setiap kode dapat memiliki sub-kode yang lebih spesifik. Contohnya, kode A00 untuk "Demam berdarah Dengue" memiliki sub-kode A00.0 untuk "Demam berdarah Dengue tanpa komplikasi", A00.1 untuk "Demam berdarah Dengue dengan komplikasi syok dan/atau kegagalan organ", dan seterusnya.

2. Penambahan kode baru: WHO secara berkala menambahkan kode baru ke ICD-10 untuk mencerminkan perkembangan penyakit dan kondisi medis baru.
3. Variasi antar Negara : Beberapa negara mengadopsi versi ICD-10 dengan modifikasi dan penambahan kode lokal mereka sendiri. (who.int, 2023)

Perkembangan dataset ICD-10 akan mempengaruhi kinerja dari algoritma yang digunakan nanti, karena semakin besar data yang diproses seiring dengan berkembangnya variasi diagnose yang akan ditemukan.



2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian

Analisis Perbandingan Kinerja Algoritma Apriori, Fp-Growth Dan Eclat Dalam Menemukan Pola Frekuensi Pada Dataset

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores</i>	Andi Ilhamsyah Idris, Eliyah A M Sampetoding, Valian Yoga Pudy Ardhana, Irene Maritsa Adrisumatri Sakri, Hidayatullah Ruslan, Esther Sanda Manapa, IJCS (International Journal of Informatics and Computer Science) Tahun 2022	Membandingkan metode Apriori, Apriori-TID, dan FP-Growth dalam menentukan perilaku transaksi konsumen dan menghitung kuantitas transaksi konsumen dalam beberapa musim berdasarkan data yang diperoleh dari database Market Basket Analysis	Ketiga algoritma menghasilkan aturan serupa dalam analisis keranjang pasar pada berbagai musim. Dalam penelitian ini, ditemukan bahwa transaksi tertentu memiliki nilai confidence tertinggi. Evaluasi kinerja ketiga algoritma juga dilakukan berdasarkan jumlah aturan yang dihasilkan, penggunaan memori, dan waktu respons.	Disarankan untuk memperluas cakupan data dengan mengambil sampel dari berbagai toko kelontong atau supermarket untuk mendapatkan hasil yang lebih representatif. Penelitian ini hanya fokus pada analisis keranjang pasar berdasarkan musim. Disarankan untuk melakukan analisis lebih mendalam, seperti analisis segmentasi konsumen atau analisis asosiasi antar produk, untuk mendapatkan wawasan yang lebih detail tentang perilaku konsumen.	Perbedaan terletak pada metode yang digunakan dan teknik pengujian algoritma. Kesamaan nya terletak pada tujuan penelitian yaitu untuk melakukan komparasi 3 algoritma.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	<i>Evaluation of Apriori, FP growth and Eclat association rule mining algorithms</i>	Dr. V Srinadh, International Journal of Health Sciences, Tahun 2022	Mengevaluasi tiga jenis algoritma, yaitu Apriori, FP Growth, dan Eclat, untuk melakukan "association rule mining"	Algoritma Apriori dan Fpgrowth menggunakan format data horizontal, yang mengharuskan pemindaian basis data beberapa kali dan menyebabkan waktu komputasi yang cukup lama. Eclat memiliki keunggulan dalam hal waktu komputasi. Eclat menghilangkan pemindaian basis data berulang dan mengkonsumsi waktu yang lebih sedikit daripada Apriori dan Fpgrowth tetapi sangat tergantung pada data masukan dan sumber daya yang tersedia. Jika diperhitungkan faktor lain selain waktu, hasilnya mungkin bervariasi tergantung pada faktor-faktor tersebut.	Dataset yang digunakan masih sederhana, disarankan menggunakan dataset yang lebih kompleks	Terdapat kesamaan 3 algoritma yang diteliti, dataset yang digunakan pada penelitian ini berbeda dengan dataset yang akan diteliti

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	<i>Discovering Future Earnings Patterns through FP-Growth and ECLAT Algorithms with Optimized Discretization</i>	Putthiporn Thanathamabe, Siriporn Sawangarereak . Emerging Science Journal, ISSN 2610-9182, Tahun 2022	Penggunaan Algoritma FP-Growth dan Eclat untuk mengidentifikasi pola laba masa depan dari laporan keuangan pada Bursa Efek Thailand.	Algoritma FP-Growth dan Eclat berhasil menemukan beberapa pola yaitu : - empat pola peningkatan pendapatan yang terkait kuat dengan kategori aset dan kewajiban dalam keuangan - Sembilan pola penurunan pendapatan yang terkait kuat lainnya terkait dengan kewajiban, pendapatan, dan kategori beban dalam laporan keuangan. dan mengidentifikasi sepuluh akuntansi penting descriptor yang mencerminkan laba masa depan.	Data yang diproses hanya berasal dari Bursa Efek Thailand (SET), yang didirikan sebagai sumber pendanaan jangka panjang untuk perusahaan besar. sedangkan untuk pasar usaha kecil dan menengah yakni Pasar Alternatif Investasi (MAI) tidak diproses.	Persamaan penelitian ini pada tujuan penelitian yaitu menemukan pola frekuensi pada data. tetapi hanya menggunakan 2 algoritma asosiasi

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	<i>Determination of the best rule-based analysis results from the comparison of the Fp-Growth, Apriori, and TPQ-Apriori Algorithms for recommendation systems</i>	Moch Syahrir . Lalu Zazuli Azhar Mardedi, Jurnal Manajemen Teknologi dan Informatika, Tahun 2023	Melakukan perbandingan algoritma Fp-growth, Apriori, dan TPQ-Apriori untuk menganalisis hasil rule dari ketiga algoritma tersebut pada data penjualan department store Kopegtel NTB yang telah diunggah di situs Kaggle.	Semakin besar dataset yang akan diolah, hasilnya akan lebih optimal jika menggunakan algoritma fp-growth RapidMiner, namun kurang maksimal jika dataset yang akan diolah merupakan dataset kecil. Berbeda dengan menggunakan algoritma FP-growth Apriori dan Weka dimana rule yang dihasilkan kurang maksimal jika dataset yang digunakan besar dan optimal jika datasetnya kecil. Metode yang digunakan sebagai perbandingan adalah algoritma Apriori tradisional fp-growth dan TPQ-Apriori, Apriori Tradisional	Tidak memberikan informasi lengkap tentang karakteristik dataset yang digunakan. Sangat penting untuk menyertakan detail tentang jumlah data, jenis data, dan karakteristik dataset lainnya, karena ini dapat memengaruhi hasil penelitian.	Ada kesamaan pada tujuan penelitian yaitu membandingkan antara 3 algoritma asosiasi untuk menemukan aturan asosiasi pada sebuah dataset, tetapi algoritma yang digunakan sedikit berbeda,

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				<p>FP-growth, dan TPQ-Apriori. Untuk pengujian rule, hasil rule TPQ-Apriori dibandingkan dengan algoritma Fp-Growth pada tools Rapidminer dan Weka serta algoritma Apriori pada tools Weka. Dari hasil pengujian, TPQ-Apriori mampu lebih optimal dan konsisten.</p>		
5	<p>Penerapan Association Rule Dalam Bidang Kesehatan Pada Rumah Sakit X</p>	<p>Karina Intan Indradewi, Tia Atika Putri, Seminar Nasional IENACO 2019 ISSN : 2337 - 4349</p>	<p>Menemukan association rule mengenai penyakit Diabetes Melitus berdasarkan karakteristik pasien, gejala dan diagnosis penyakit berdasarkan data yang didapat pada Rumah Sakit X.</p>	<p>Penerapan Association Rule menggunakan algoritma FP-Growth dengan prosedur yang telah dilakukan menggunakan software Rapid Miner didapatkan 12 aturan valid dimana memiliki nilai lift ratio > 1 yang digunakan sebagai acuan untuk mendapatkan sebuah knowledge.</p>	<p>Disarankan untuk memperluas cakupan data tidak hanya untuk 1 diagnosa penyakit.</p>	<p>Dataset yang digunakan sama-sama pada bidang kesehatan tetapi variabel yang digunakan berbeda dengan yang akan diteliti. Algoritma yang digunakan pada penelitian ini hanya menggunakan 1 algoritma asosiasi.</p>

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Penerapan Association Rule Menggunakan Algoritma Apriori Pada Poliklinik Penyakit Dalam (Studi Kasus: Rumah Sakit Umum Daerah Bintan)	Nola Ritha, Eka Suswanti Wisnu Pebriadi, Jurnal Sains dan Informatika, ISSN e-ISSN: 2598-5841, Tahun 2021	Penerapan Algoritma Apriori dalam menentukan aturan asosiasi berdasarkan jenis kelamin, umur, kelurahan, dan kode ICD10.	Algoritma Apriori dapat digunakan untuk menentukan aturan asosiasi pada poliklinik Penyakit Dalam Rumah Sakit Umum Daerah Bintan. Penelitian ini berhasil menemukan sebanyak 22 aturan asosiasi dengan memberikan batas minimal support sebesar 6% dan minimal confidence sebesar 50%. Dan hanya 9 aturan asosiasi yang bisa digunakan dan 13 aturan asosiasi yang didapatkan dari total 22 aturan asosiasi tidak bisa digunakan, karena informasi dari 13 hasil aturan asosiasi tersebut memuat informasi yang kurang spesifik.	Disarankan dapat memperluas cakupan data, tidak hanya pada salah satu poliklinik penyakit dalam saja, tetapi menggunakan data semua poliklinik lainnya. Variabel diagnosa hanya menggunakan 1 diagnosa utama saja, belum memperhitungkan diagnosa penyerta atau komplikasi	Dataset yang digunakan sama-sama pada bidang kesehatan tetapi variabel yang digunakan berbeda dengan yang akan diteliti. Algoritma yang digunakan pada penelitian ini hanya menggunakan 1 algoritma asosiasi.

2.3. Landasan Teori

Menurut Gartner Group, "Data mining atau penambangan data adalah proses menemukan korelasi, pola, dan tren baru yang ada pada sejumlah data yang besar dan disimpan dalam repositori, dengan menggunakan teknik data mining serta teknik statistik dan matematika." (Larose, 2005).

2.3.1. Analisis Asosiasi (Association Rule Mining)

Analisis Asosiasi adalah metode penambangan data untuk mengidentifikasi hubungan asosiatif antara sekelompok item. Istilah "analisis keranjang belanja" juga berlaku untuk analisis asosiasi. Analisis keranjang belanja mengidentifikasi atribut atau karakteristik yang serupa dan menyelidiki hubungan antara atribut tersebut dengan mengidentifikasi pedoman untuk menghitung korelasi antara dua atau lebih atribut. (Larose, 2005).

Definisi lain dari analisis asosiasi adalah proses mengidentifikasi semua aturan asosiatif yang memenuhi kondisi minimal untuk dukungan (*minimal support*) dan keyakinan (*minimal confidence*). (Ulumuddin dan Juanita, 2018).

2.3.2. Metodologi Dasar Aturan Asosiasi

Ada 2 metodologi dasar dalam aturan asosiasi (Kusrini, 2009), yaitu sebagai berikut :

a. Analisis Pola Frekuensi Tinggi

Pada tahap ini dicari kombinasi item yang memenuhi syarat minimum dari nilai *support* pada database. Persentase item atau kombinasi item yang ada pada database disebut nilai *support* (penunjang) . Berikut ini cara mencari nilai *support* sebuah item dengan rumus persamaan sebagai berikut :

$$\text{Support (A)} = \frac{\text{Jumlah Transaksi Untuk A}}{\text{Total Transaksi}} \quad (2.1)$$

Rumus nilai *support* pada rumus Persamaan 2.1 menjelaskan bahwa nilai *support* diperoleh dengan cara membagi jumlah transaksi yang mengandung item A (satu item) dengan total jumlah keseluruhan transaksi.

Sedangkan untuk nilai *support* dari 2 item diperoleh dari rumus persamaan sebagai berikut:

$$\text{Support (A,B)} = \frac{\sum \text{Transaksi Untuk A dan B}}{\sum \text{Transaksi}} \quad (2.2)$$

Rumus Persamaan 2.2 menjelaskan bahwa nilai *support* 2-itemsets diperoleh dengan cara membagi jumlah transaksi yang mengandung item A dan item B dengan total jumlah keseluruhan transaksi.

b. Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tinggi telah diidentifikasi, tahap selanjutnya adalah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan cara menghitung nilai *confidence* aturan asosiatif A B. Rumus persamaan di bawah ini digunakan untuk menghitung nilai *confidence* aturan A B :

$$\text{Confidence} = P(B|A) = \frac{\sum \text{Transaksi Untuk A dan B}}{\sum \text{Transaksi A}} \quad (2.3)$$

Nilai *confidence* dihitung menggunakan rumus Persamaan 2.3 dengan membagi jumlah transaksi yang mengandung item A dan item B dengan jumlah total transaksi yang hanya mengandung item A, seperti terlihat pada rumus di atas. Aturan asosiasi harus diurutkan menurut $\text{Support} \times \text{Confidence}$ untuk memilihnya, kemudian aturan diambil sebanyak n aturan yang memiliki hasil terbesar.

2.3.3 Algoritma Apriori

Algoritma Apriori digunakan untuk menghitung jumlah aturan asosiasi antar item. Aturan asosiasi mendefinisikan bagaimana dua atau lebih item terkait satu sama lain. (trivusi.web.id , 2022)

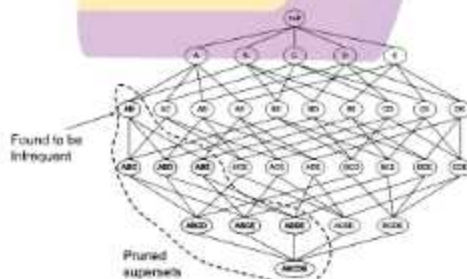
Ilmuwan R. Agrawal dan Srikan mengemukakan algoritma ini pada tahun 1994. Algoritma ini membantu dalam mengidentifikasi produk yang dapat dibeli bersamaan dan biasanya digunakan untuk analisis keranjang belanja. Apriori juga dapat digunakan di bidang medis untuk menemukan reaksi obat pada pasien.

Algoritma Apriori adalah pilihan populer dalam data mining, karena dapat digunakan untuk mencari informasi yang paling sering muncul dalam database. Itemset terdiri dari items data transaksi yang ditemukan dalam database.

Itemset yang paling sering muncul akan dipilih oleh algoritma Apriori agar dapat digunakan untuk menentukan aturan asosiasi dan menggambarkan tren umum dalam database.

a. Cara Kerja Algoritma Apriori

Algoritma Apriori menggunakan strategi “bottom-up” untuk mengidentifikasi kumpulan item yang paling sering muncul.



Gambar 2.1. Cara kerja algoritma apriori (sumber: medium.com)

Pada Gambar 2.1. Cara kerja algoritma apriori (sumber: medium.com), menggambarkan bahwa untuk menentukan kandidat itemset dengan efisien, metode ini melakukan pendekatan menggunakan *Breadth-first search* dan struktur data Hash tree.

Pendekatan dari bottom-up memuli dengan setiap item dalam daftar itemset. Setelah itu, kandidat dibentuk melalui *self-join*. Kita memperluas panjang itemset satu item yang membentuk subhimpunan setiap iterasi.

Subhimpunan kemudian diuji. Lakukan *pruning* (pemangkasan) jika ditemukan itemset berisi subhimpunan yang jarang.

Proses ini diulangi sampai tidak ada lagi itemset yang berhasil diturunkan dari data.

b. Algoritma Apriori memiliki langkah-langkah sebagai berikut :

1. Tentukan *support* dari *itemset* (dengan ukuran $k = 1$) dalam *database*.
Pada tahap ini akan menghasilkan himpunan kandidat.
2. Jika ditemukan *support* lebih kecil dari ambang batas (*threshold*) yang diberikan maka lakukan pemangkasan (*pruning*) kumpulan kandidat.
3. Buat himpunan ukuran $k + 1$ dengan menggabungkan himpunan item yang paling sering muncul., dan ulangi himpunan di atas hingga tidak ada lagi itemset yang dapat dibentuk.

c. Komponen pada Algoritma Apriori

Pada algoritma Apriori, terdapat 3 komponen penting, yaitu : *Support*, *Confidence* dan *Lift*.

Sebagai contoh studi kasus :

Pada sebuah minimarket A memiliki 2000 data transaksi pembelian. 2000 data transaksi tersebut terdiri dari pembelian 200 kopi sachet, dan 500 minyak goreng.

Dari 500 transaksi pembelian minyak goreng tersebut, terdapat 100 transaksi gabungan antara pembelian kopi sachet dan minyak goreng.

Proses perhitungannya dapat dilihat Tabel 2.2. Proses perhitungan algoritma apriori sebagai berikut :

Tabel 2.2. Proses perhitungan algoritma apriori

Keterangan	Rumus	Hasil
<i>Support</i> (kopi) dihitung dengan rumus Persamaan 2.1	jumlah transaksi kopi / total transaksi semua produk	$200/2000 = 0.1 = 10\%$
<i>Support</i> (minyak) dihitung dengan rumus Persamaan 2.1	jumlah transaksi minyak / total transaksi semua produk	$500 / 2000 = 0.4 = 40\%$
<i>Support</i> (kopi dan minyak) dihitung dengan rumus Persamaan 2.2	jumlah transaksi kombinasi kopi dan minyak / total transaksi semua produk	$100 / 2000 = 0.05 = 5\%$
<i>Confidence</i> dihitung dengan rumus Persamaan 2.3	jumlah transaksi kombinasi kopi dan minyak / total transaksi kopi	$100/200 = 0.5 = 50\%$
<i>Lift Ratio</i> dihitung dengan rumus Persamaan 2.6	$\text{support}(\text{kopi dan minyak}) / (\text{support}(\text{kopi}) * \text{support}(\text{minyak}))$	$0.05 / (0.1 * 0.4) = 1.25$

Pelanggan mungkin membeli kopi sachet dan minyak goreng pada saat yang sama. Ini disebut *Confidence*.

Seiring dengan penjualan minyak goreng, ada peningkatan rasio penjualan kopi. Ketika minyak goreng dan kopi dijual dalam satu sachet, kemungkinan seseorang membeli keduanya adalah 1 poin 25 kali lebih besar daripada membeli satu-satunya. Nilai *lift* di bawah satu menunjukkan bahwa pelanggan jarang membeli dua barang secara bersamaan. Sebaliknya, kombinasi yang lebih baik selalu memiliki nilai yang lebih tinggi, dan konsumen lebih cenderung membelinya sekaligus.

2.3.4. Algoritma FP-Growth

FP-Growth merupakan salah satu algoritma asosiasi yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. (Samuel, 2008)

FP-Growth merupakan algoritma pencarian *frequent item set* yang diperoleh dari FP-tree dengan cara menjelajahi pohon dari bawah ke atas (Tan, Steinbach, dan Kumar, 2006).

Dapat disimpulkan, bahwa FP-Growth merupakan algoritma pencarian kumpulan data yang sering muncul dari kumpulan data dengan menggunakan teknik pembuatan struktur data Pohon.

Tujuan dari algoritma FP-Growth adalah sama dengan Apriori yaitu mencari hubungan antar item di dalam sebuah dataset yang telah ditentukan (*Association Rules*).

Tahapan Algoritma FP-Growth :

1. Sediakan set data

2. Pengurutan didasarkan pada nilai frekuensi tertinggi dari item yang muncul
3. Membentuk *FP-tree*
4. Membentuk *Conditional Pattern Base*
5. Membentuk *Conditional FP-tree*
6. Membentuk *Frequent Pattern Generated*
7. Mencari *frequency 2 Itemset*
8. Mencari *Support 2 Itemset* 9 Mencari *Confidence 2 Itemset*

2.3.5. Algoritma Equivalence Class Transformation (ECLAT)

Algoritma asosiasi yang lainnya yang sering dipakai adalah *Equivalence Class Transformation* (ECLAT). Algoritma ECLAT ditemukan oleh M. J. Zaki pada tahun 1997. Dalam algoritma ECLAT mencari item dari yang paling sering muncul hingga item yang paling jarang muncul dengan dilakukan tanpa memperhatikan urutan apapun, proses pemindaian dilakukan sekali tanpa melakukan perulangan. (Mustofa and Irhamah, 2019). Proses pencarian itemset akan terus dilakukan pencarian selagi masih terdapat itemset yang tersisa. Sebuah *Transaction Id List* (TID List) akan menyimpan data dari setiap *itemset*, kemudian TID list yang mengandung transaksi itemset yang sama (*frequent itemset*) akan diurutkan. Lalu *k-itemset* disusun ke dalam kelas-kelas menurut kriteria tertentu dan dibentuk dengan partisi suatu himpunan (*equivalence class*), untuk mendapatkan $(k+1)$ -*itemset* maka dilakukan penggabungan pasangan *frequent k-itemset* berdasarkan kelas yang sama.

Kemampuannya untuk mengungkapkan asosiasi yang kuat di antara item-item yang berbeda dan efisiensi dalam menangani data yang besar adalah dua keuntungan menggunakan algoritma ECLAT.

Membedakan dataset menjadi serangkaian itemset vertikal yang masing-masing terdiri dari satu item adalah cara algoritma ini berfungsi. Selanjutnya, algoritma ini menghitung dukungan (*support*) dari setiap itemset untuk mengidentifikasi itemset yang sering muncul bersama.

Langkah-langkah algoritma ini sebagai berikut :

1. Langkah pertama adalah mengidentifikasi semua item dalam dataset dan membuat itemset awal yang hanya mengandung satu item.
2. Selanjutnya, algoritma akan menghitung dukungan (*support*) untuk setiap set item. dukungan (*support*) adalah jumlah transaksi yang mencakup itemset tersebut dibagi dengan jumlah total transaksi.
3. Setelah itu, itemset-itemset yang memiliki dukungan (*support*) di atas batas minimum akan dipertahankan, sedangkan itemset-itemset yang tidak memiliki dukungan akan dihapus.
4. Algoritma kemudian akan membentuk itemset baru yang lebih panjang dengan menggabungkan itemset yang dipertahankan.
5. Proses penggabungan *itemsets* dilakukan secara berulang hingga tidak ada lagi *itemsets* baru yang terbentuk.
6. Terakhir, algoritma ini akan menghasilkan semua *itemset* yang memenuhi batas minimum dukungan (*support*).

Sebagai contoh, algoritma ini dapat membantu kita menemukan pola yang ada pada dataset penjualan dengan memperoleh data item yang sering dibeli bersama seperti "roti", "susu", dan "coklat".

2.3.6. Menghitung performance algoritma

Mengukur akurasi dari algoritma asosiasi menggunakan nilai *confidence* dan *conviction*. Nilai *confidence* adalah nilai akurasi atau biasa disebut nilai kepastian, besar atau kecilnya nilai *confidence* menentukan akurat atau tidaknya rule tersebut. Rule kurang akurat jika *confidence* kecil. Tetapi jika nilai *confidence*-nya besar maka dapat dipastikan *rules* tersebut akurat. Nilai *confidence* menentukan kuatnya hubungan antar item dalam algoritma asosiasi.

Setelah nilai *confidence* sudah ditentukan, selanjutnya adalah menentukan nilai *conviction*. Untuk mengetahui tingkat akurasi dari dan kekuatan *rule* yang terbentuk digunakan nilai *conviction*.

Nilai range pada *conviction* ini berada pada nilai 0,5 1...., dengan ketentuan *conviction* dianggap memiliki nilai tak terhingga (*infinite*) apabila nilai dari *confidence* ($A \rightarrow B$) sama dengan 1. *Conviction* menghasilkan nilai rule yang semakin jauh dari 1 bahkan sampai tak hingga (∞), maka akan dianggap semakin akurat. (Han, Kamber, dan Pei, 2006).

Nilai *Conviction* menggunakan rumus persamaan sebagai berikut :

$$\text{Conviction}(A \rightarrow B) = \frac{1 - \text{Support}(B)}{1 - \text{Confidence}(A \rightarrow B)} \quad (2.4)$$

Lift Ratio adalah suatu ukuran (*parameter*) untuk mengetahui kekuatan aturan asosiasi (*association rule*) yang telah terbentuk dari nilai *support* dan

confidence. Nilai *lift ratio* biasanya digunakan sebagai penentu apakah aturan asosiasi valid atau tidak valid.

Antecedent merupakan sebab yang menjadikan item *consequent*. Sedangkan *consequent* adalah sebuah akibat atau juga item yang akan dibeli setelah membeli *antecedent*. Nilai dari *expected confidence* dihitung dengan rumus persamaan sebagai berikut :

$$\text{Expected confidence} = \frac{\sum \text{Transaksi mengandung B}}{\text{Jumlah Transaksi}} \quad (2.5)$$

Lift ratio dapat dihitung dengan 2 cara sebagai berikut :

1. Membandingkan antara *confidence* untuk suatu aturan dibagi dengan *expected confidence* (Santosa, 2007), dengan rumus persamaannya sebagai berikut :

$$\text{Lift Ratio} = \frac{\text{Confidence}}{\text{Expected Confidence}} \quad (2.6)$$

2. Membandingkan nilai *support* A B dibagi dengan perkalian antara nilai *support* A dan nilai *support* B (trivusi.web.id, 2022), dengan rumus persamaan sebagai berikut :

$$\text{Lift Ratio} = \frac{\text{Support (A,B)}}{(\text{Support(A)} * \text{Support(B)})} \quad (2.7)$$

Penulis akan menggunakan rumus Persamaan 2.7 untuk menghitung *lift ratio* dalam penelitian ini. Jika nilai *lift Ratio* lebih besar dari 1 menunjukkan adanya manfaat dari aturan tersebut. Lebih tinggi nilai *lift ratio*, lebih besar kekuatan asosiasinya. (Santosa, 2007)

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Adapun jenis, sifat dan pendekatan penelitian ini adalah sebagai berikut :

1. Jenis Penelitian Eksperimen

Pelatihan ini dilakukan untuk mengetahui algoritma yang memiliki tingkat kepercayaan tertinggi dalam menentukan hasil asosiasi yang dihasilkan.

2. Sifat Penelitian Deskriptif

Penelitian ini menjelaskan tentang tahapan-tahapan dalam melakukan pengukuran kinerja dari ketiga algoritma.

3. Pendekatan penelitian kuantitatif

Penelitian ini menggunakan pendekatan kuantitatif yaitu hasil dari pengukuran kinerja dari ketiga algoritma disajikan dengan data berupa angka dan diagram.

3.2. Metode Pengumpulan Data

Langkah-langkah yang dilakukan dalam pengumpulan data adalah sebagai berikut :

1. Identifikasi Sumber Data :

Identifikasi sumber data yang diperlukan untuk penelitian ini, yaitu data klaim INACBG's dari Rumah Sakit Daerah Gunung Jati Kota Cirebon.

2. Perizinan dan Persetujuan :

Mengajukan surat izin kepada Rumah Sakit Daerah Gunung Jati untuk mengakses dan menggunakan data klaim INACBG's yang diperlukan untuk penelitian.

3. Pengumpulan Data Klaim INACBG's :
 - a. Identifikasi periode waktu yang akan menjadi fokus penelitian.
 - b. Dapatkan data klaim INACBG's dari Rumah Sakit Daerah Gunung Jati Kota Cirebon.
4. Validasi dan Verifikasi Data :
 - a. Lakukan validasi dan verifikasi data yang telah dikumpulkan untuk memastikan integritas dan keakuratan data.
 - b. Periksa keberadaan nilai yang hilang atau tidak valid, duplikat data, dan pastikan konsistensi dalam format data.
5. Pengolahan dan Pengorganisasian Data :
 - a. Lakukan proses pengolahan data untuk membersihkan dan merapikan data yang dikumpulkan.
 - b. Organisasikan data dalam format yang sesuai untuk analisis berikutnya, seperti spreadsheet atau database.
6. Dokumentasi dan Privasi Data :
 - a. Pastikan untuk menjaga kerahasiaan dan privasi data pasien sesuai dengan aturan dan peraturan yang berlaku.
 - b. Dokumentasikan metode dan langkah-langkah pengumpulan data yang telah dilakukan untuk memudahkan replikasi dan validasi penelitian.

3.3. Metode Analisis Data

Dalam melakukan penelitian ini penulis menggunakan sistem analisa data kuantitatif dengan pendekatan metode analisis deskriptif, analisa data dilakukan dengan tahapan sebagai berikut :

1. Identifikasi Dataset INA-CBG's;
2. Mengeksplorasi dataset INA-CBG's untuk memahami struktur dan karakteristiknya.
3. Identifikasi variabel atau atribut yang akan digunakan dalam analisis.
4. Melakukan reprocessing data dengan mengecek nilai-nilai yang hilang atau data yang tidak valid jika diperlukan.
5. Normalisasi data dengan mengecek jika dataset memiliki atribut dengan skala yang berbeda, normalisasikan data untuk memastikan perbandingan yang adil antara algoritma.
6. Encoding data dengan merubah kategorikal menjadi bentuk numerik jika algoritma yang akan digunakan memerlukan data numerik.
7. Menentukan parameter-parameter yang sesuai untuk setiap algoritma, seperti nilai support threshold dan confidence threshold
8. Menerapkan algoritma Apriori, fp-growth, dan Eclat pada dataset INA-CBG's dengan parameter yang telah ditentukan.
9. Mengekstrak aturan asosiasi yang dihasilkan oleh setiap algoritma.
10. Mencatat waktu eksekusi dan penggunaan sumber daya (memori, CPU) untuk setiap algoritma.

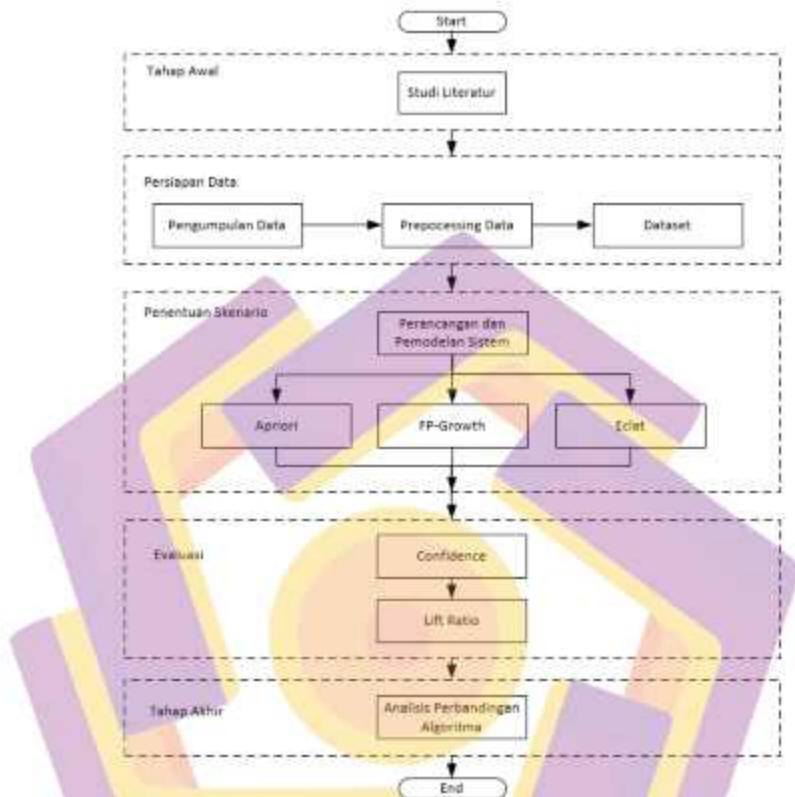
11. Evaluasi Kinerja Algoritma dengan menghitung metrik kinerja seperti *support*, *confidence*, *lift*, *conviction*, dan *lift ratio* untuk masing-masing aturan yang dihasilkan.
12. Bandingkan hasil aturan yang dihasilkan oleh ketiga algoritma.
13. Analisis hasil berdasarkan metrik kinerja dan tujuan penelitian.
14. Membuat visualisasi hasil berupa grafik atau tabel untuk mempresentasikan hasil perbandingan algoritma.
15. Membuat kesimpulan dari hasil analisis data.
16. Membuat laporan penelitian yang mencakup semua langkah di atas.

Tools yang digunakan untuk membantu analisa data, penulis menggunakan bahasa pemrograman python dengan memanfaatkan *library-library* yang menunjang analisa asosiasi ketiga algoritma tersebut.

3.4. Alur Penelitian

Alur penelitian disini menjelaskan mengenai tahapan-tahapan atau prosedur penelitian yang akan dilaksanakan secara terurut dan sistematis agar pelaksanaan penelitian dapat berjalan dengan baik.

Adapun alur penelitian ini dapat dilihat pada gambar 2 sebagai berikut :



Gambar 3.1. Alur penelitian

1. Tahap Awal

Alur dari penelitian ini berawal dari identifikasi masalah pada objek penelitian yang dilakukan serta studi pustaka yaitu dengan mengumpulkan dan menganalisa publikasi-publikasi yang telah dilakukan sebelumnya serta mengambil beberapa referensi teori dari beberapa buku yang telah terbit dan berkaitan dengan

penelitian yang dilakukan sebagai rujukan untuk menentukan metode algoritma yang sesuai untuk penyelesaian permasalahan.

2. Persiapan Data

Pada tahap persiapan data, peneliti melakukan pengumpulan data klaim INACBG's pada RSD Gunung Jati Kota Cirebon. Data yang diambil adalah data pasien Rawat Inap selama satu tahun. Variabel yang ada pada data INACBG's adalah sebagai berikut :

Tabel 3.1. Variabel data INACBG's

No	Nama Variabel	Keterangan Variabel
1	KODE_RS	Kode Rumah Sakit
2	KELAS_RS	Kelas Rumah Sakit
3	KELAS_RAWAT	Kelas Perawatan
4	KODE_TARIF	Kode Tarif INACBG
5	ADMISSION_DATE	Tgl Masuk
6	DISCHARGE_DATE	Tgl Keluar
7	BIRTH_DATE	Tanggal Lahir
8	BIRTH_WEIGHT	Berat Lahir
9	SEX	Jenis Kelamin
10	DISCHARGE_STATUS	Status Pulang
11	DIAGLIST	List Diagnosa (ICD 10)
12	PROCLIST	List Prosedur (ICD 9)
13	INACBG	Kode Group Diagnosa
14	DESKRIPSI_INACBG	Keterangan Group Diagnosa
15	TARIF_INACBG	Tarif INACBG
16	TARIF_RS	Tarif Rumah Sakit
17	LOS	Length Of Stay
18	NAMA_PASIEN	Nama Pasien
19	MRN	Medical Record Number
20	UMUR_TAHUN	Umur dalam Tahun
21	UMUR_HARI	Umur dalam Hari
22	DPJP	Dokter Penanggung Jawab
23	SEP	No SEP
24	NOKARTU	No. Kartu BPJS

Data yang sudah diambil kemudian dilakukan pre processing data yaitu melakukan pembersihan data dan transformasi data, setelah itu mengambil variabel Diaglist (list diagnose kode ICD 10) untuk dijadikan dataset yang dibutuhkan pada proses selanjutnya.

3. Penentuan Skenario

Pada tahap ini dilakukan proses pengolahan data terhadap dataset yang telah disiapkan sebelumnya, membuat model untuk masing-masing algoritma yang digunakan, dengan tahapan sebagai berikut :

- a. Mencari frekuensi item untuk masing-masing item data.
- b. Menghitung nilai mean atau rerata frekuensi dari ketiga algoritma tersebut.
- c. Mencari nilai support masing-masing item data.
- d. Menentukan nilai minimum support untuk memfilter data yang dibutuhkan untuk proses selanjutnya.
- e. Mencari asosiasi rule yang terbentuk dari hasil filter data.

4. Evaluasi

Pada tahap ini mengevaluasi asosiasi rule yang sudah ditemukan dengan mencari nilai confidence dan nilai lift ratio untuk setiap algoritma yang digunakan.

5. Tahap Akhir

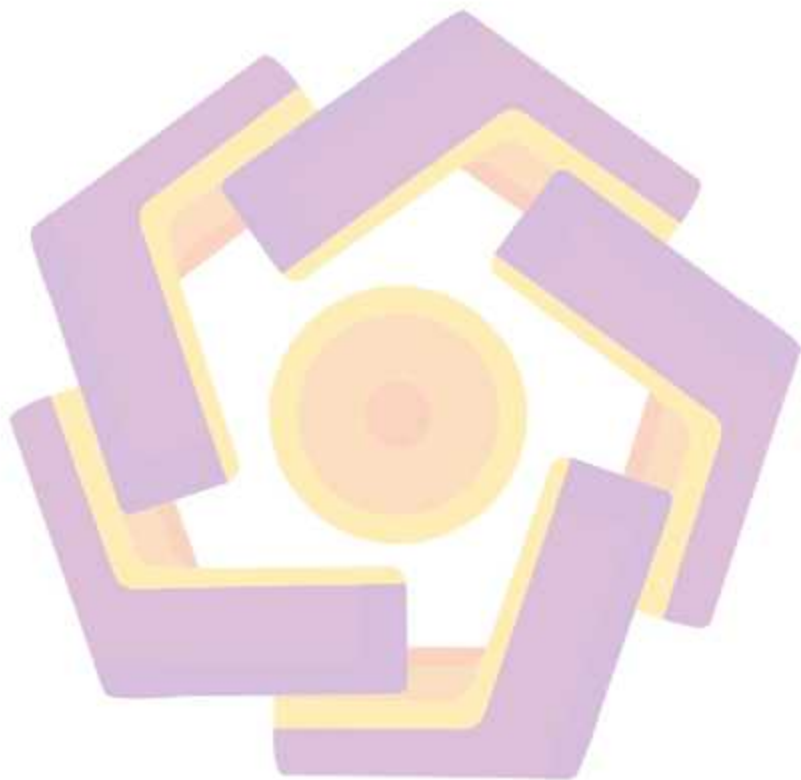
Pada tahap akhir menganalisa asosiasi rule dengan tahapan :

- a. Mendeskripsikan antecedent dan consequent pada asosiasi rule yang terbentuk dalam bentuk kode ICD menjadi deskripsi diagnosa sesuai dengan Buku Kode Klasifikasi Penyakit dan Tindakan Medis ICD 10 dan ICD 9.
- b. Memilih asosiasi rule terbaik dengan nilai confidence $>60\%$ dan nilai lift ratio >1 .

Membandingkan asosiasi rule yang terbaik dari ketiga model algoritma.

Hasil dari perbandingan ini maka ditemukan algoritma yang paling cocok untuk

digunakan dalam menemukan pola frekuensi dan aturan asosiasi pada dataset INACBG's.



BAB IV

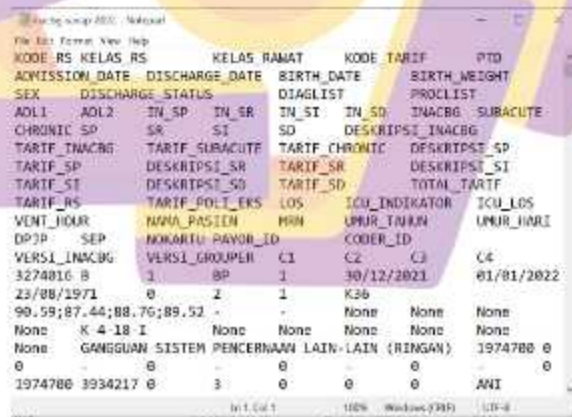
HASIL PENELITIAN DAN PEMBAHASAN

4.1. Persiapan Data

Data diambil dari aplikasi E-Klaim INA-CBG, periode data pelayanan rawat inap tanggal 1 Januari 2022 sampai dengan 31 Desember 2022, dihasilkan sebuah file berekstensi .txt, kemudian diberi nama *inacbg-ranap-2022.txt*.

4.1.1 Preprocessing Data

Agar File *inacbg-ranap-2022.txt* dapat dibaca dengan jelas tiap variabelnya maka file tersebut dibuka menggunakan excel, dan dilakukan proses *delimited*, dapat dilihat pada gambar *Gambar 4.2. Proses tahap 1 konversi TXT ke sheet excel* dan *Gambar 4.3. Proses tahap 2 konversi TXT ke sheet excel*, adapapun file tersebut berisi seperti pada *Gambar 4.1. File inacbg-ranap-2022.txt*.



KODE_RS	KELAS_RS	KELAS_RAMAT	KODE_TARIF	PTD			
ADMISSION_DATE	DISCHARGE_DATE	BIRTH_DATE	BIRTH_WEIGHT				
SEX	DISCHARGE_STATUS	DIAGLIST	PROCLIST				
ADL1	ADL2	IN_SP	IN_ER	IN_SI	IN_SO	INACBG	SUBACUTE
CHRONIC_SP	SR	SI	SD	DESKRIPSI_INACBG			
TARIF_INACBG	TARIF_SUBACUTE	TARIF_CHRONIC	DESKRIPSI_SP				
TARIF_SP	DESKRIPSI_SR	TARIF_SR	DESKRIPSI_SI				
TARIF_SI	DESKRIPSI_SO	TARIF_SO	TOTAL_TARIF				
TARIF_RS	TARIF_POLT_EKS	LOS	ICU_INDIKATOR	ICU_LOS			
VENT_HOUR	NAMA_PASIEEN	HRH	UMUR_TAHUN	UMUR_HARI			
DP3P	SEP	NOMORU_PAVOB_ID	CODER_ID				
VERSI_INACBG	VERSI_GROUPLEM	C1	C2	C3	C4		
3274816	B	1	BP	1	30/12/2021	01/01/2022	
23/08/1971	0	2	1	K36			
90.59;87.44;88.76;89.52	-	-	-	None	None	None	
None	K 4 18 1	None	None	None	None	None	
None	GANGGUAN SISTEM PENCERNAAN LAIN-LAIN (RINGAN)					1974780	0
0	0	0	0	0	0	0	0
1974780	3934217	0	3	0	0	0	ANI

Gambar 4.1. File *inacbg-ranap-2022.txt*

Dataset tersebut memiliki 24 variabel data utama dan 36 variabel pelengkap lainnya. Setelah itu dipilih variabel DIAGLIST dengan cara menghapus variabel lain, kemudian disave menjadi file .csv diberi nama inacbg-ranap-2022.csv.

4.1.2. Pemanggilan Dataset

Proses selanjutnya adalah menggunakan *google drive* untuk menyimpan *dataset* dan menggunakan *google colaboratory* digunakan untuk menjalankan program dengan menggunakan bahasa pemrograman python. Untuk mengolah dataset tersebut dimulai dengan proses pemanggilan dataset dengan menggunakan perintah sebagai berikut :

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
dataset =
pd.read_csv("/content/drive/MyDrive/ADataset/inacbg-ranap-
2022.csv")
dataset
```

Setelah dieksekusi maka akan menampilkan isi dari dataset tersebut, sebagai berikut :

	DIAGLIST
0	K30
1	D105.R090
2	K899
3	H184
4	C439
...	...
14667	B24,D381,E678,J189,J40,J179
14668	D214,D630,K921
14669	E119,E40,E975,E860,094,N184,N380,R54,D638
14670	C56,D630
14671	J189,B909

14672 rows * 1 columns

Gambar 4.5. Isi dataset DIAGLIST

Untuk menampilkan info dari dataset DIAGLIST dengan mengetikan sintak sebagai berikut :

```
dataset.info()
```

Setelah dieksekusi maka akan menampilkan info dari dataset tersebut, sebagai berikut :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14672 entries, 0 to 14671
Data columns (total 1 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   DIAGLIST    14672 non-null  object
dtypes: object(1)
memory usage: 114.8+ KB
```

Gambar 4.6. Info dataset DIAGLIST

Hasil dari info dataset, dataset memiliki 14672 baris dan 1 kolom, dan tidak ada data yang Null atau kosong.

4.2. Perancangan dan Pemodelan Sistem

Untuk mendapatkan aturan asosiasi pada dataset diperlukan 2 metodologi dasar dalam aturan asosiasi, yaitu analisis pola frekuensi tinggi dan pembentukan aturan asosiatif.

4.2.1. Pola Frekuensi Tinggi

Pada tahap ini dicari kombinasi item yang memenuhi syarat minimum dari nilai *support* pada database. Persentase item atau kombinasi item yang ada pada database disebut nilai *support* (penunjang).

Untuk menghitung frekuensi dataset maka perlu membuat matrix itemset untuk menampilkan kemunculan kode ICD10 dengan cara mengeksekusi sintak sebagai berikut :

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
data = list(dataset.apply(lambda x:x.split(',') or ''))

te = TransactionEncoder()
te_ary = te.fit(data).transform(data)
df = pd.DataFrame(te_ary, columns=te.columns_)
df
```

Menghasilkan matrix itemset seperti dibawah ini :

	1410	1411	1441	1451	1455	1470	1484	1489	1490	1500	1530	1540	1554	1555	1561	1561	1570	1582	1588	1592	
0	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
1	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
2	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
3	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
4	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
...
14697	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
14698	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
14699	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
14670	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu
14671	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu	Falsu

Gambar 4.7. Matrix Itemset

Pada Gambar 4.7. Matrix Itemset diatas diketahui ukuran dari dataset DIAGLIST memiliki 14672 baris dan 2151 kolom.

Langkah selanjutnya menghitung frekuensi dataset dengan menghitung nilai support tiap-tiap itemset menggunakan algoritma apriori, fp-growth dan ECLAT.

4.2.1.1. Pola Frekuensi Tinggi Algoritma FP-Growth

Pada fp-growth sintak untuk menampilkan frekuensi itemset pada dataset dengan nilai minimal support = 0.01 adalah sebagai berikut :

```
fpgrowth(df, min_support=0.01, use_colnames=True)
```

Menghasilkan data itemset seperti gambar dibawah ini :

	support	itemssets
0	0.031080	{K011}
1	0.079403	{D649}
2	0.019493	{I500}
3	0.013972	{D696}
4	0.019786	{P599}
...		
63	0.018130	{E880, D649}
64	0.010564	{Z370, O429}
65	0.013154	{A099, E96}
66	0.027058	{Z370, O621}
67	0.012745	{Z511, C20}

68 rows x 2 columns

Gambar 4.8. Frekuensi itemset algoritma fp-growth

Untuk mendapatkan nilai *support*, sebagai contoh pada Gambar 4.8.

Frekuensi itemset algoritma pada *index* 67 dimana $(Z511 \rightarrow C20)$ mendapat nilai *support* 0,012745 diperoleh dari perhitungan sebagai berikut :

Diketahui :

Jumlah transaksi *dataset* DIAGLIST = 14672, Antecedents (A) = Z511,

Consequents (C) = C20, dan Frekuensi kemunculan $(A \rightarrow C) = 187$

Dengan menggunakan rumus persamaan 2.2, maka :

$$Support(A \rightarrow C) = \frac{\sum \text{Transaksi}(A \rightarrow C)}{\sum \text{Transaksi}}$$

$$Support = \frac{187}{14672} = 0.012745$$

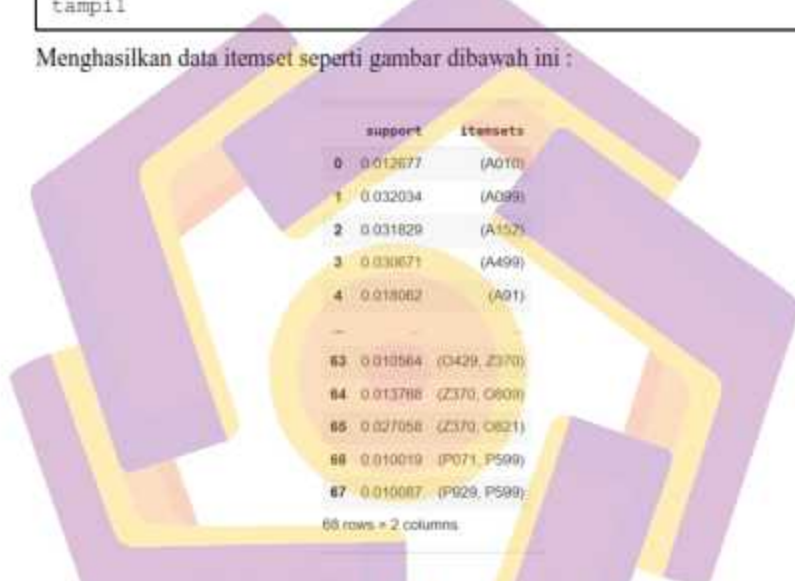
Berdasarkan Gambar 4.8. Frekuensi itemset algoritma , diketahui terdapat 68 *itemssets* dengan nilai *support* lebih dari atau sama dengan 1% dan didapatkan *itemssets* dengan lebih dari 1 kategori yang disebut sebagai *antecedents* dan *consequents*. *Itemssets* inilah yang akan diproses lebih lanjut.

4.2.1.2. Pola Frekuensi Tinggi Algoritma Apriori

Pada apriori sintak untuk menampilkan frekuensi *itemset* pada dataset dengan nilai minimal *support* = 0.01 adalah sebagai berikut :

```
from mlxtend.frequent_patterns import apriori
apriori(df, min_support=0.01, use_colnames=True)
tampil=apriori(df, min_support=0.01, use_colnames=True)
tampil
```

Menghasilkan data *itemset* seperti gambar dibawah ini :



	support	itemsets
0	0.012677	(A010)
1	0.032034	(A099)
2	0.031829	(A157)
3	0.030671	(A499)
4	0.018062	(A01)
...
63	0.010564	(O429, Z370)
64	0.013708	(Z370, C600)
65	0.027056	(Z370, C821)
66	0.010019	(P071, P599)
67	0.010087	(P929, P599)

68 rows x 2 columns

Gambar 4.9. Frekuensi itemset algoritma apriori

Untuk mendapatkan nilai *support*, sebagai contoh pada Gambar 4.9. Frekuensi *itemset* algoritma apriori pada *index* 67 dimana (P929 → P599) mendapat nilai *support* 0.010087 diperoleh dari perhitungan sebagai berikut :

Diketahui :

Jumlah transaksi dataset DIAGLIST = 14672, *Antecedents* (A) = P929, *Consequents* (C) = P599, dan Frekuensi kemunculan (A → C) = 148

Dengan menggunakan rumus persamaan 2.2, maka :

$$\text{Support}(A \rightarrow C) = \frac{\sum \text{Transaksi}(A \rightarrow C)}{\sum \text{Transaksi}}$$

$$\text{Support} = \frac{148}{14672} = 0.010087$$

Berdasarkan Gambar 4.9. Frekuensi itemset algoritma apriori, diketahui terdapat 68 *itemsets* dengan nilai *support* lebih dari atau sama dengan 1% dan didapatkan *itemsets* dengan lebih dari 1 kategori yang disebut sebagai *antecedents* dan *consequents*. *Itemsets* inilah yang akan diproses lebih lanjut.

4.2.1.3. Pola Frekuensi Tinggi Algoritma ECLAT

Pada ECLAT sintak untuk menampilkan frekuensi *itemset* pada dataset dengan nilai minimal *support* = 0.01 adalah sebagai berikut :

```
from pyECLAT import Example1, Example2
from pyECLAT import ECLAT
import pandas as pd

ex1 = Example1().get()
eclat_instance = ECLAT(data=ex1, verbose=True)

get_ECLAT_indexes,
get_ECLAT_supports =
eclat_instance.fit(min_support=0.01, min_combination=1, max_combination=
2, separator=' & ', verbose=True)

df = pd.DataFrame(get_ECLAT_supports.items(), columns=['itemsets',
'support'])
df
```

Menghasilkan data itemset seperti gambar dibawah ini :

	itemsets	support
0	{A91}	0.018062
1	{B99}	0.012200
2	{E43}	0.014449
3	{E870}	0.014449
4	{O809}	0.014449
...
63	{J969, E870}	0.014517
64	{J969, N179}	0.011791
65	{F88, A099}	0.013154
66	{D64, E880}	0.018130
67	{Z370, O429}	0.010564

68 rows x 2 columns

Gambar 4.10. Frekuensi *itemset* algoritma ECLAT

Untuk mendapatkan nilai *support*, sebagai contoh pada Gambar 4.10.

Frekuensi *itemset* algoritma pada *index* 67 dimana ($Z370 \rightarrow O429$) mendapat nilai *support* 0.010564 diperoleh dari perhitungan sebagai berikut :

Diketahui :

Jumlah transaksi dataset DIAGLIST = 14672, *Antecedents* (A) = Z370, *Consequents* (C) = O429, dan Frekuensi kemunculan ($A \rightarrow C$) = 155

Dengan menggunakan rumus persamaan 2.2, maka :

$$Support(A \rightarrow C) = \frac{\sum Transaksi(A \rightarrow C)}{\sum Transaksi}$$

$$Support = \frac{155}{14672} = 0.010564$$

Berdasarkan Gambar 4.9. Frekuensi *itemset* algoritma apriori, diketahui terdapat 68 *itemsets* dengan nilai *support* lebih dari atau sama dengan 1% dan didapatkan *itemsets* dengan lebih dari 1 kategori yang disebut sebagai *antecedents* dan *consequents*. *Itemsets* inilah yang akan diproses lebih lanjut.

4.2.1.4. Perhitungan Pola Frekuensi Tinggi Algoritma Apriori, FP-Growth dan ECLAT

Langkah selanjutnya setiap algoritma dilakukan perhitungan frekuensi dataset dari nilai *support* = 0.001 sampai dengan nilai *support* = 0.02, maka hasilnya dapat dilihat pada Tabel 4.1. Hasil perhitungan nilai *support* sebagai berikut :

Tabel 4.1. Hasil perhitungan nilai *support*

Support (>)	Jumlah Itemset N=1	Jumlah Itemset N=2	Jumlah Itemset N=3
0.001	395	546	276
0.002	237	236	86
0.003	166	126	35
0.004	136	93	20
0.005	107	61	10
0.006	94	45	5
0.007	84	30	2
0.008	67	20	1
0.009	58	17	-
0.010	56	12	-
0.011	53	8	-
0.012	50	7	-
0.013	44	5	-
0.014	41	3	-
0.015	34	2	-
0.016	33	2	-
0.017	31	2	-
0.018	29	2	-
0.019	28	1	-
0.020	25	1	-
0.030	18	-	-

Tabel 4.1. Hasil perhitungan nilai *support* (Lanjutan)

Support (>)	Jumlah Itemset N=1	Jumlah Itemset N=2	Jumlah Itemset N=3
0.040	5	-	-
0.050	3	-	-
0.060	2	-	-
0.070	1	-	-
0.080	-	-	-

Dari hasil perhitungan pada Tabel 4.1. Hasil perhitungan nilai *support* didapatkan beberapa kesimpulan yaitu sebagai berikut :

1. Semakin kecil nilai *support* maka dihasilkan jumlah itemset semakin banyak.
2. Jumlah *itemset* N=3 hanya ditemukan pada perhitungan dengan nilai *support* < 0.009. Jumlah itemset yang ditemukan maksimal 276 *itemset*.
3. Jumlah *itemset* N=2 hanya ditemukan pada perhitungan dengan nilai *support* < 0.030. Jumlah itemset yang ditemukan maksimal 546 *itemset*.
4. Jumlah *itemset* N=1 dapat ditemukan pada perhitungan dengan nilai *support* < 0.080.

Dari Tabel 4.1. Hasil perhitungan nilai *support* kita pilih nilai *support* 0.01 dengan nilai N=2 untuk menampilkan aturan asosiasi yang terbentuk.

4.2.2. Pembentukan Aturan Asosiatif

4.2.2.1 Pembentukan Aturan Asosiatif Algoritma Apriori

Syntax python untuk melakukan pembentukan aturan asosiatif pada algoritma apriori adalah sebagai berikut :

```
from mxtend.frequent_patterns import association_rules
from mxtend.frequent_patterns import apriori, fpmax,
fpgrowth

freuent_itemsets=apriori(df.astype('bool'),min_support=0.01,
use_colnames=True)
df_ar=association_rules(freuent_itemsets,
metric="confidence", min_threshold=0.1)
df_ar
```

Menghasilkan data aturan asosiatif seperti pada tabel berikut :

Tabel 4.2 Hasil aturan asosiatif algoritma apriori

	antecedents	consequents	support	confidence	lift	leverage	conviction
0	(E86)	(A099)	0.013154	0.476543	14.876260	0.012270	1.849.181
1	(A099)	(E86)	0.013154	0.410638	14.876260	0.012270	1.649.914
2	(C20)	(Z511)	0.012745	0.745020	21.731476	0.012159	3.787.421
3	(Z511)	(C20)	0.012745	0.371769	21.731476	0.012159	1.564.541
4	(D649)	(E880)	0.018130	0.228326	3.303749	0.012642	1.206.324
5	(E880)	(D649)	0.018130	0.262327	3.303749	0.012642	1.247.975
6	(E870)	(J969)	0.010087	0.698113	22.074821	0.009630	3.207.743
7	(J969)	(E870)	0.010087	0.318966	22.074821	0.009630	1.447.138
8	(E878)	(J969)	0.014517	0.560526	17.724229	0.013698	2.203.488
9	(J969)	(E878)	0.014517	0.459052	17.724229	0.013698	1.800.727
10	(I10)	(I639)	0.012541	0.327402	10.397498	0.011335	1.439.956
11	(I639)	(I10)	0.012541	0.398268	10.397498	0.011335	1.598.214
12	(N179)	(J969)	0.011791	0.298791	9.447978	0.010543	1.381.008
13	(J969)	(N179)	0.011791	0.372845	9.447978	0.010543	1.531.578
14	(O429)	(Z370)	0.010564	0.856354	18.078302	0.009980	6.631.776
15	(Z370)	(O429)	0.010564	0.223022	18.078302	0.009980	1.271.160
16	(Z370)	(O809)	0.013768	0.290647	20.114999	0.013083	1.389.367
17	(O809)	(Z370)	0.013768	0.952830	20.114999	0.013083	20.195.774
18	(O821)	(Z370)	0.027058	0.990025	20.900210	0.025764	95.501.244
19	(Z370)	(O821)	0.027058	0.571223	20.900210	0.025764	2.268.473
20	(P071)	(P599)	0.010019	0.604938	30.605705	0.009692	2.481.218
21	(P599)	(P071)	0.010019	0.506897	30.605705	0.009692	1.994.384
22	(P929)	(P599)	0.010087	0.711538	35.998939	0.009807	3.398.146
23	(P599)	(P929)	0.010087	0.510345	35.998939	0.009807	2.013.301

Dari Tabel 4.2 Hasil aturan asosiatif algoritma apriori hasil pembentukan aturan asosiasi algoritma apriori menggunakan nilai minimum *support* = 0.01 didapat 24 aturan asosiasi dengan rentang nilai *confidence* antara 0.223022 sampai dengan 0.990025.

Berdasarkan Tabel 4.2 Hasil aturan asosiatif algoritma apriori, pada index data ke 0 dengan *antecedents* E86 dan *consequents* A099 didapat nilai *confidence* = 0.476543, diperoleh dari perhitungan sebagai berikut :

Diketahui :

Antecedents (A) = E86, *Consequents* (C) = A099, Jumlah transaksi *Antecedents* (A) = 405, dan frekuensi kemunculan (A → C) = 193

Dengan menggunakan rumus persamaan 2.3, maka :

$$\text{Confidence} = P(A \rightarrow C) = \frac{\sum \text{Transaksi } (A \rightarrow C)}{\sum \text{Transaksi } A}$$

$$\text{Confidence} = \frac{193}{405} = 0.476543$$

Untuk mendapatkan nilai *confidence* terbaik, kita menentukan minimum *confidence* yang akan dipilih adalah nilai *confidence* > 0.6, dengan menggunakan *syntax* python adalah sebagai berikut :

```
df_ar[(df_ar.support > 0.01) & (df_ar.confidence > 0.6)].sort_values("confidence", ascending = False)
```

Hasilnya adalah seperti pada tabel berikut :

Tabel 4.3. Hasil aturan asosiasi algoritma apriori nilai *confidence* > 0.6

	antece dents	conseq uents	support	confidence	lift	leverage	conviction
18	(O821)	(Z370)	0.027058	0.990025	20.900210	0.025764	95.501.244
17	(O809)	(Z370)	0.013768	0.952830	20.114999	0.013083	20.195.774
14	(O429)	(Z370)	0.010564	0.856354	18.078302	0.009980	6.631.776
3	(C20)	(Z511)	0.012745	0.745020	21.731476	0.012159	3.787.421
23	(P929)	(P599)	0.010087	0.711538	35.998939	0.009807	3.398.146
7	(E870)	(J969)	0.010087	0.698113	22.074821	0.009630	3.207.743
21	(P071)	(P599)	0.010019	0.604938	30.605705	0.009692	2.481.218

Dari Tabel 4.3. Hasil aturan asosiasi algoritma apriori nilai *confidence* > 0.6, maka diperoleh 7 aturan asosiasi pada algoritma ECLAT dengan nilai *support* > 0.01 dan nilai *confidence* > 0.6 dengan nilai *lift ratio* tertinggi 35.998939 pada

index data ke 23 yaitu *itemset* P929 dengan P599. Nilai *lift ratio* tersebut diperoleh dari perhitungan sebagai berikut :

Diketahui :

Antecedents (A) support P929 = 0.014177, *Consequents (C) support* P599 = 0.019766, *Support* A→C = 0.010087, dengan menggunakan rumus persamaan 2.7, maka :

$$\text{Lift Ratio} = \frac{\text{Support}(A \rightarrow C)}{(\text{Support}(A) * \text{Support}(C))}$$

$$\text{Lift Ratio} = \frac{0.010087}{(0.014177 * 0.019766)} = 35,99$$

Pada Tabel 4.3. Hasil aturan asosiasi algoritma apriori nilai *confidence* > 0.6, ditemukan nilai *confidence* tertinggi yaitu 0.990025 (mendekati 1) pada iterasi ke 18 bisa disimpulkan *rules* yang terbentuk dengan *Antecedent* (O821) dan *Consequents* (Z370) adalah akurat atau pasti. Nilai *Conviction* pada iterasi ke 18 sebesar 95.501, didapat dari perhitungan rumus persamaan 2.4, sebagai berikut :

$$\text{Conviction}(A \rightarrow B) = \frac{1 - \text{Support}(B)}{1 - \text{Confidence}(A \rightarrow B)} = \frac{1 - 0.047369}{1 - 0.990025} = 95.501$$

Nilai *Conviction* 95.501 menyatakan bahwa nilai *rule* > 1 bahkan semakin menjauh dari 1, maka *rule* tersebut semakin akurat.

4.2.2.2 Pembentukan Aturan Asosiatif Algoritma FP-Growth

Syntax python untuk melakukan pembentukan aturan asosiatif pada algoritma FP-Growth adalah sebagai berikut :

```
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, fpmax, fpgrowth
freuent_itemsets=fpgrowth(df.astype('bool'),min_support=0.01,
use_colnames=True)
df_ar=association_rules(freuent_itemsets,metric="confidence",
min_threshold=0.1) df_ar
```

Menghasilkan data aturan asosiatif seperti pada tabel berikut :

Tabel 4.4. Hasil aturan asosiatif algoritma fp-growth

	<i>antecedents</i>	<i>consequents</i>	<i>support</i>	<i>confidence</i>	<i>lift</i>	<i>leverage</i>	<i>conviction</i>
0	(P071)	(P599)	0.010019	0.604938	30.605705	0.009692	2.481.218
1	(P599)	(P071)	0.010019	0.506897	30.605705	0.009692	1.994.384
2	(P929)	(P599)	0.010087	0.711538	35.998939	0.009807	3.398.146
3	(P599)	(P929)	0.010087	0.510345	35.998939	0.009807	2.013.301
4	(N179)	(J969)	0.011791	0.298791	9.447978	0.010543	1.381.008
5	(J969)	(N179)	0.011791	0.372845	9.447978	0.010543	1.531.578
6	(I639)	(I10)	0.012541	0.398268	10.397498	0.011335	1.598.214
7	(I10)	(I639)	0.012541	0.327402	10.397498	0.011335	1.439.956
8	(E878)	(J969)	0.014517	0.560526	17.724229	0.013698	2.203.488
9	(J969)	(E878)	0.014517	0.459052	17.724229	0.013698	1.800.727
10	(E870)	(J969)	0.010087	0.698113	22.074821	0.009630	3.207.743
11	(J969)	(E870)	0.010087	0.318966	22.074821	0.009630	1.447.138
12	(Z370)	(O809)	0.013768	0.290647	20.114999	0.013083	1.389.367
13	(O809)	(Z370)	0.013768	0.952830	20.114999	0.013083	20.195.774
14	(D649)	(E880)	0.018130	0.228326	3.303749	0.012642	1.206.324
15	(E880)	(D649)	0.018130	0.262327	3.303749	0.012642	1.247.975
16	(Z370)	(O429)	0.010564	0.223022	18.078302	0.009980	1.271.160
17	(O429)	(Z370)	0.010564	0.856354	18.078302	0.009980	6.631.776
18	(E86)	(A099)	0.013154	0.476543	14.876260	0.012270	1.849.181
19	(A099)	(E86)	0.013154	0.410638	14.876260	0.012270	1.649.914
20	(Z370)	(O821)	0.027058	0.571223	20.900210	0.025764	2.268.473
21	(O821)	(Z370)	0.027058	0.990025	20.900210	0.025764	95.501.244
22	(Z511)	(C20)	0.012745	0.371769	21.731476	0.012159	1.564.541
23	(C20)	(Z511)	0.012745	0.745020	21.731476	0.012159	3.787.421

Dari Tabel 4.4. Hasil aturan asosiatif algoritma fp-growth, hasil pembentukan aturan asosiasi algoritma FP-Growth menggunakan nilai minimum *support* = 0.01 didapat 24 aturan asosiasi dengan rentang nilai *confidence* antara 0.223022 sampai dengan 0.990025.

Untuk mendapatkan nilai *confidence* terbaik, kita menentukan minimum *confidence* yang akan dipilih adalah nilai *confidence* > 0.6, dengan menggunakan *syntax* python adalah sebagai berikut :

```
df_ar[(df_ar.support > 0.01) & (df_ar.confidence > 0.6)].sort_values("confidence", ascending = False)
```

Hasilnya adalah seperti pada tabel berikut :

Tabel 4.5. Hasil aturan asosiatif algoritma fp-growth nilai *confidence* > 0.6

	<i>antecedents</i>	<i>consequents</i>	<i>support</i>	<i>confidence</i>	<i>lift</i>	<i>leverage</i>	<i>conviction</i>
21	(O821)	(Z370)	0.027058	0.990025	20.900210	0.025764	95.501.244
13	(O809)	(Z370)	0.013768	0.952830	20.114999	0.013083	20.195.774
17	(O429)	(Z370)	0.010564	0.856354	18.078302	0.009980	6.631.776
23	(C20)	(Z511)	0.012745	0.745020	21.731476	0.012159	3.787.421
2	(P929)	(P599)	0.010087	0.711538	35.998939	0.009807	3.398.146
10	(E870)	(P969)	0.010087	0.698113	22.074821	0.009630	3.207.743
0	(P071)	(P599)	0.010019	0.604938	30.605705	0.009692	2.481.218

Dari Tabel 4.5. Hasil aturan asosiatif algoritma fp-growth nilai *confidence* > 0.6, maka diperoleh 7 aturan asosiasi pada algoritma FP-Growth dengan nilai *support* > 0.01 dan nilai *confidence* > 0.6 dengan nilai *lift ratio* tertinggi 35.998939 yaitu pada *itemset* P929 dengan P599 pada index data ke 2 yaitu *itemset* P929 dengan P599. Nilai *lift ratio* tersebut diperoleh dari perhitungan sebagai berikut :

Diketahui :

Antecedents (A) *support* P929 = 0.014177, *Consequents* (C) *support* P599 = 0.019766, *Support* A→C = 0.010087, dengan menggunakan rumus persamaan 2.7, maka :

$$\text{Lift Ratio} = \frac{\text{Support}(A \rightarrow C)}{(\text{Support}(A) * \text{Support}(C))}$$

$$\text{Lift Ratio} = \frac{0.010087}{(0.014177 * 0.019766)} = 35,99$$

Pada Tabel 4.5. Hasil aturan asosiatif algoritma fp-growth nilai *confidence* > 0.6, ditemukan nilai *confidence* tertinggi yaitu 0.990025 (mendekati 1) pada iterasi ke 21 bisa disimpulkan rules yang terbentuk dengan *Antecedent* (O821) dan

Consequents (Z370) adalah akurat atau pasti. Nilai *Conviction* pada iterasi ke 21 sebesar 95.501, didapat dari perhitungan rumus persamaan 2.4, sebagai berikut :

$$\text{Conviction}(A \rightarrow B) = \frac{1 - \text{Support}(B)}{1 - \text{Confidence}(A \rightarrow B)} = \frac{1 - 0.047369}{1 - 0.990025} = 95.501$$

Nilai *Conviction* 95.501 menyatakan bahwa nilai *rule* > 1 bahkan semakin menjauh dari 1, maka *rule* tersebut semakin akurat.

4.2.2.2 Pembentukan Aturan Asosiatif Algoritma ECLAT

Syntax python untuk melakukan pembentukan aturan asosiatif pada algoritma ECLAT adalah sebagai berikut :

```
from pyECLAT import ECLAT
import pandas as pd
from mlxtend.frequent_patterns import association_rules

dataset = pd.read_csv("/content/drive/MyDrive/ADataset/insanbg-ranap-
2021.csv", header = None)
dataset.dropna()
dataset = dataset.applymap(str)
eclat_instance = ECLAT(data=dataset, verbose=True)
get_ECLAT_indexes, get_ECLAT_supports =
eclat_instance.fit(min_support=0.01, min_combination=1, max_combination=
3, separator=' + ', verbose=True)
df = pd.DataFrame(get_ECLAT_supports.items(), columns=['itemsets',
'support'])
new_column = []
for row in df['itemsets']:
    r = row.split(' + ')
    r = tuple(i.strip() for i in r)
    new_column.append(r)
df['itemsets'] = pd.Series(new_column)
df_ar=association_rules(df, metric="confidence", min_threshold=0.1)
df_ar
```

Menghasilkan data aturan asosiatif seperti pada tabel berikut :

Tabel 4.6. Hasil aturan asosiatif algoritma ECLAT

	<i>antecedents</i>	<i>consequents</i>	<i>support</i>	<i>confidence</i>	<i>lift</i>	<i>leverage</i>	<i>conviction</i>
0	(P929)	(P599)	0.010087	0.711538	35.998939	0.009807	3.398.146
1	(P599)	(P929)	0.010087	0.510345	35.998939	0.009807	2.013.301
2	(E880)	(D649)	0.018130	0.262327	3.303749	0.012642	1.247.975
3	(D649)	(E880)	0.018130	0.228326	3.303749	0.012642	1.206.324
4	(Z370)	(O821)	0.027058	0.571223	20.900210	0.025764	2.268.473
5	(O821)	(Z370)	0.027058	0.990025	20.900210	0.025764	95.501.244
6	(C20)	(Z511)	0.012745	0.745020	21.731476	0.012159	3.787.421
7	(Z511)	(C20)	0.012745	0.371769	21.731476	0.012159	1.564.541
8	(P599)	(P071)	0.010019	0.506897	30.605705	0.009692	1.994.384
9	(P071)	(P599)	0.010019	0.604938	30.605705	0.009692	2.481.218
10	(I10)	(I639)	0.012541	0.327402	10.397498	0.011335	1.439.956
11	(I639)	(I10)	0.012541	0.398268	10.397498	0.011335	1.598.214
12	(J969)	(E870)	0.010087	0.318966	22.074821	0.009630	1.447.138
13	(E870)	(J969)	0.010087	0.698113	22.074821	0.009630	3.207.743
14	(Z370)	(O429)	0.010564	0.223022	18.078302	0.009980	1.271.160
15	(O429)	(Z370)	0.010564	0.856354	18.078302	0.009980	6.631.776
16	(O809)	(Z370)	0.013768	0.952830	20.114999	0.013083	20.195.774
17	(Z370)	(O809)	0.013768	0.290647	20.114999	0.013083	1.389.367
18	(E86)	(A099)	0.013154	0.476543	14.876260	0.012270	1.849.181
19	(A099)	(E86)	0.013154	0.410638	14.876260	0.012270	1.649.914
20	(N179)	(J969)	0.011791	0.298791	9.447978	0.010543	1.381.008
21	(J969)	(N179)	0.011791	0.372845	9.447978	0.010543	1.531.578
22	(J969)	(E878)	0.014517	0.459052	17.724229	0.013698	1.800.727
23	(E878)	(J969)	0.014517	0.560526	17.724229	0.013698	2.203.488

Tabel 4.6 Hasil aturan asosiatif algoritma ECLAT, hasil pembentukan aturan asosiasi algoritma ECLAT menggunakan nilai minimum *support* = 0.01 didapat 24 aturan asosiasi dengan rentang nilai *confidence* antara 0.223022 sampai dengan 0.990025.

Tabel 4.6 Hasil aturan asosiatif algoritma ECLAT, pada *index* data ke 0 dengan *antecedents* P929 dan *consequents* P599 didapat nilai *confidence* = 0.711538, diperoleh dari perhitungan sebagai berikut :

Diketahui :

Antecedents (A) = P929, *Consequents* (C) = P599, Jumlah transaksi *Antecedents*

(A) = 208, dan Frekuensi kemunculan (A → C) = 148

Dengan menggunakan rumus 2.3, maka :

$$\text{Confidence} = P(A \rightarrow C) = \frac{\sum \text{Transaksi } (A \rightarrow C)}{\sum \text{Transaksi } A}$$

$$\text{Confidence} = \frac{148}{208} = 0.711538$$

Untuk mendapatkan nilai *confidence* terbaik, kita menentukan minimum *confidence* yang akan dipilih adalah nilai *confidence* > 0.6, dengan menggunakan *syntax* python adalah sebagai berikut :

```
df_ar[(df_ar.support > 0.01) & (df_ar.confidence > 0.6)].sort_values("confidence", ascending = False)
```

Hasilnya adalah seperti pada tabel berikut :

Tabel 4.7. Hasil aturan asosiatif algoritma ECLAT nilai *confidence* > 0.6

	<i>antecedents</i>	<i>consequents</i>	<i>support</i>	<i>confidence</i>	<i>lift</i>	<i>leverage</i>	<i>conviction</i>
5	(O821)	(Z370)	0.027058	0.990025	20.900210	0.025764	95.501.244
16	(O809)	(Z370)	0.013768	0.952830	20.114999	0.013083	20.195.774
15	(O429)	(Z370)	0.010564	0.856354	18.078302	0.009980	6.631.776
6	(C20)	(Z511)	0.012745	0.745020	21.731476	0.012159	3.787.421
0	(P929)	(P599)	0.010087	0.711538	35.998939	0.009807	3.398.146
13	(E870)	(J969)	0.010087	0.698113	22.074821	0.009630	3.207.743
9	(P071)	(P599)	0.010019	0.604938	30.605705	0.009692	2.481.218

Dari Tabel 4.7. Hasil aturan asosiatif algoritma ECLAT nilai *confidence* > 0.6, maka diperoleh 7 aturan asosiasi pada algoritma ECLAT dengan nilai *support* > 0.01 dan nilai *confidence* > 0.6 dengan nilai *lift ratio* tertinggi 35.998939 pada

index data ke 0 yaitu *itemset* P929 dengan P599. Nilai *lift ratio* tersebut diperoleh dari perhitungan sebagai berikut :

Diketahui :

Antecedents (A) *Support* P929 = 0.014177, *Consequents* (C) *support* P599 = 0.019766, *Support* A→C = 0.010087, dengan menggunakan rumus persamaan 2.7, maka :

$$\text{Lift Ratio} = \frac{\text{Support}(A \rightarrow C)}{(\text{Support}(A) * \text{Support}(C))}$$

$$\text{Lift Ratio} = \frac{0.010087}{(0.014177 * 0.019766)} = 35,99$$

Pada Tabel 4.7. Hasil aturan asosiatif algoritma ECLAT nilai *confidence* > 0.6. ditemukan nilai *confidence* tertinggi yaitu 0.990025 (mendekati 1) pada iterasi ke 5 bisa disimpulkan *rules* yang terbentuk dengan *Antecedent* (O821) dan *Consequents* (Z370) adalah akurat atau pasti. Nilai *Conviction* pada iterasi ke 5 sebesar 95.501, didapat dari perhitungan rumus persamaan 2.4, sebagai berikut :

$$\text{Conviction}(A \rightarrow B) = \frac{1 - \text{Support}(B)}{1 - \text{Confidence}(A \rightarrow B)} = \frac{1 - 0.047369}{1 - 0.990025} = 95,501$$

Nilai *Conviction* 95.501 menyatakan bahwa nilai *rule* > 1 bahkan semakin menjauh dari 1, maka *rule* tersebut semakin akurat.

Hasil dari perhitungan aturan asosiatif ketiga algoritma menemukan *itemset* yang sama, hanya berbeda pada proses iterasi. Jika kita urutkan *itemset* berdasarkan nilai *conviction* tertinggi maka data aturan asosiatif ketiga algoritma tersebut seperti tabel berikut ini :

Tabel 4.8. Hasil aturan asosiatif ketiga algoritma

No	antecedents	consequents	support	confidence	lift	leverage	conviction
1	(O821)	(Z370)	0.027058	0.990025	20.900.210	0.025764	95.501.244
2	(O809)	(Z370)	0.013768	0.952830	20.114.999	0.013083	20.195.774
3	(O429)	(Z370)	0.010564	0.856354	18.078.302	0.009980	6.631.776
4	(C20)	(Z511)	0.012745	0.745020	21.731.476	0.012159	3.787.421
5	(P929)	(P599)	0.010087	0.711538	35.998.939	0.009807	3.398.146
6	(E870)	(J969)	0.010087	0.698113	22.074.821	0.009630	3.207.743
7	(P071)	(P599)	0.010019	0.604938	30.605.705	0.009692	2.481.218

Pada ketiga algoritma tersebut juga dilakukan pencarian aturan asosiasi dengan nilai *support* > 0.02 dan nilai *confidence* > 0.6 dihasilkan *itemset* sebagai berikut :

Tabel 4.9. Hasil aturan asosiatif algoritma nilai *support*>0.02 dan nilai *confidence* > 0.6

	antecedents	consequents	support	confidence	lift	leverage	conviction
0	(O821)	(Z370)	0.027058	0.990025	20.90021	0.025764	95.501244

Dari Tabel 4.9. Hasil aturan asosiatif algoritma nilai *support*>0.02 dan nilai *confidence* > 0.6, dinyatakan bahwa *itemset* O821 dengan Z370 memiliki nilai *confidence* yang sangat tinggi yaitu 0.990025, nilai *lift ratio* 20.90021 dan nilai *conviction* 95.501. hubungan langsung antara *itemset* anteseden dan konsekuen tersebut sangat kuat.

4.2.3. Analisa Aturan Asosiatif

Setelah hasil aturan asosiasi yang terbentuk selanjutnya mendeskripsikan nilai antecedent dan consequent dalam bentuk kode ICD menjadi deskripsi diagnosa sesuai dengan Buku Kode Klasifikasi Diagnosa Penyakit ICD 10. menjadi sebagai berikut :

1. O821 : Delivery by emergency caesarean section
2. Z370 : Single live birth
3. O809 : Single spontaneous delivery
4. O429 : Premature rupture of membranes
5. C20 : Malignant neoplasm of rectum
6. Z511 : Encounter for antineoplastic chemotherapy and immunotherapy
7. P929 : Feeding problem of newborn
8. P599 : Neonatal jaundice
9. E870 : Hyperosmolality and hypernatraemia
10. J969 : Respiratory failure
11. P071 : Other low birth weight

Berdasarkan tabel hasil perhitungan aturan asosiasi ketiga algoritma diatas, maka didapatkan aturan asosiasi dengan nilai *support* 0.01 dan nilai *confidence* > 0.6, sebagai berikut :

1. *Itemset* (O821) dan (Z370) dengan nilai *confidence* 0.99 dan nilai *lift ratio* 20.90
Jika pasien melahirkan dengan tindakan operasi caesar sebagian besar melahirkan 1 bayi
2. *Itemset* (O809) dan (Z370) dengan nilai *confidence* 0.95 dan nilai *lift ratio* 20.11
Jika pasien melahirkan dengan normal sebagian besar melahirkan 1 bayi
3. *Itemset* (O429) dan (Z370) dengan nilai *confidence* 0.85 dan nilai *lift ratio* 18.07
Jika pasien melahirkan bayi premature sebagian besar melahirkan 1 bayi
4. *Itemset* (C20) dan (Z511) dengan nilai *confidence* 0.74 dan nilai *lift ratio* 21.73
Jika pasien didiagnosa tumor sebagian besar dilakukan kemoterapi
5. *Itemset* (P929) dan (P599) dengan nilai *confidence* 0.71 dan nilai *lift ratio* 35.99
Jika pasien bayi baru lahir susah menelan sebagian besar terkena juga penyakit kuning.
6. *Itemset* (E870) dan (J969) dengan nilai *confidence* 0.69 dan nilai *lift ratio* 22.07
Jika pasien didiagnosa Hiperosmolalitas dan hypernatremia biasanya mengalami gagal nafas.

7. *Itemset* (P071) dan (P599) dengan nilai *confidence* 0.60 dan nilai *lift ratio* 30.60

Jika pasien bayi lahir dengan berat badan rendah sebagian besar terkena juga penyakit kuning.

Aturan asosiasi yang dihasilkan dari ketiga algoritma dengan nilai *support*

0.02 dan nilai *confidence* > 0.6 adalah sebagai berikut :

1. *Itemset* (O821) dan (Z370) dengan nilai *confidence* 0.99 dan nilai *lift ratio* 20.90

Jika pasien melahirkan dengan tindakan operasi caesar sebagian besar melahirkan 1 bayi. *Itemset* ini sangat kuat hubungannya dan sering terjadi pada data klaim INACBG.

4.3. Analisis Perbandingan Kinerja Algoritma

4.3.1. Perbandingan Waktu Proses

Untuk mendapatkan perbandingan dari kinerja ketiga algoritma yang digunakan, maka peneliti menghitung waktu proses dan kapasitas memori yang digunakan untuk melakukan perhitungan dalam pembentukan aturan asosiasi.

Perbandingan waktu proses dilakukan menggunakan *library time* pada python, proses yang dilakukan adalah mencari aturan asosiasi dengan nilai *support* 0.01 dan *confidence* 0.6 didapatkan hasil sebagai berikut :

4.3.1.1. Waktu Proses Algoritma Apriori

Syntax untuk menghitung waktu proses pada algoritma apriori adalah sebagai berikut :

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import association_rules
from mlxtend.frequent_patterns import apriori, fpm_max, fpgrowth
import time
awal = time.time()
dataset = pd.read_csv("/content/drive/MyDrive/ADataset/inacbg-ranap-2022.csv")
all_transactions = dataset['DIAGLIST']
data = list(all_transactions.apply(lambda x:x.split(',')))
```



```

te = TransactionEncoder()
te_ary = te.fit(data).transform(data)
df = pd.DataFrame(te_ary, columns=te.columns_)
df

apriori(df, min_support=0.01, use_colnames=True)
tampil=apriori(df, min_support=0.01, use_colnames=True)

frequent_itemsets=apriori(df.astype('bool'),min_support=0.01,
use_colnames=True)
df_ar=association_rules(frequent_itemsets, metric="confidence",
min_threshold=0.1)
df_ar

df_ar[(df_ar.support > 0.01) & (df_ar.confidence >
0,6)].sort_values("confidence", ascending = False)
akhir = time.time()
print ("Total Waktu Proses ", akhir-awal, " Detik." )

```

Pengujian dilakukan 6 kali dengan hasil pada tabel berikut :

Tabel 4.10. Hasil perhitungan waktu proses algoritma apriori

Keterangan	Waktu Proses
Pengujian ke 1	1.5409400463104248 Detik
Pengujian ke 2	1.0542566776275635 Detik
Pengujian ke 3	1.6726973056793213 Detik
Pengujian ke 4	1.7362923622131348 Detik
Pengujian ke 5	2.1141762733459473 Detik
Pengujian ke 6	0.9100010395050049 Detik
Rata-rata Waktu Proses	1,504727284 Detik

4.3.1.2. Waktu Proses Algoritma FP-Growth

Syntax untuk menghitung waktu proses pada algoritma apriori adalah sebagai berikut :

```
import pandas as pd
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, fpgrowth

import time
awal = time.time()

dataset = pd.read_csv("/content/drive/MyDrive/ADataset/inacbg-ranap-2022.csv")
all_transactions = dataset['DIAGLIST']
data = list(all_transactions.apply(lambda x:x.split(',','or'.)))
te = TransactionEncoder()
te_ary = te.fit(data).transform(data)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = fpgrowth(df, min_support=0.01,
use_colnames=True)
frequent_itemsets['length'] =
frequent_itemsets['itemsets'].apply(lambda x: len(x))
freuent_itemsets=fpgrowth(df.astype('bool'),min_support=0.01,
use_colnames=True)
df_ar=association_rules(freuent_itemsets, metric="confidence",
min_threshold=0.6)
df_ar[(df_ar.support > 0.01) & (df_ar.confidence >
0.6)].sort_values("confidence", ascending = False)
akhir = time.time()
print ("Total Waktu Proses ", akhir- awal, " Detik." )
```

Pengujian dilakukan 6 kali dengan hasil pada tabel berikut :

Tabel 4.11. Hasil perhitungan waktu proses algoritma fp-growth

Keterangan	Waktu Proses
Pengujian ke 1	1.3258388042449951 Detik
Pengujian ke 2	0.7631182670593262 Detik

Tabel 4.11. Hasil perhitungan waktu proses algoritma fp-growth (Lanjutan)

Keterangan	Waktu Proses
Pengujian ke 3	0.46314048767089844 Detik
Pengujian ke 4	0.4280216693878174 Detik
Pengujian ke 5	0.530996561050415 Detik
Pengujian ke 6	0.6392478942871094 Detik
Rata-rata Waktu Proses	0,691727281 Detik

4.3.1.3. Waktu Proses Algoritma ECLAT

Syntax untuk menghitung waktu proses pada algoritma ECLAT adalah sebagai berikut :

```

from pyECLAT import ECLAT
import pandas as pd
import time
awal = time.time()
dataset = pd.read_csv('inacbg-ranap-2022.csv', header=None)
dataset.dropna()
dataset = dataset.applymap(str)
eclat_instance = ECLAT(data=dataset, verbose=True)
get_ECLAT_indexes, get_ECLAT_supports =
eclat_instance.fit(min_support=0.01, min_combination=1, max_combination=2,
separator=' ', verbose=True)
df = pd.DataFrame(get_ECLAT_supports.items(), columns=['itemsets',
'support'])
new_column = []
for row in df['itemsets']:
    r = row.split(' ')
    r = tuple(i.strip() for i in r)
    new_column.append(r)
df['itemsets'] = pd.Series(new_column)
from mixtend.frequent_patterns import association_rules
df_ar=association_rules(df, metric="confidence", min_threshold=0.6)
df_ar[(df_ar.support > 0.01) & (df_ar.confidence >
0.6)].sort_values("confidence", ascending = False)
akhir = time.time()
print ("Total Waktu Proses ", akhir- awal, " Detik." )

```

Pengujian dilakukan 6 kali dengan hasil pada tabel berikut :

Tabel 4.12. Hasil perhitungan waktu proses algoritma ECLAT

Keterangan	Waktu Proses
Pengujian ke 1	195.92988896369934 Detik
Pengujian ke 2	201.40767407417297 Detik
Pengujian ke 3	177.6650094985962 Detik
Pengujian ke 4	174.96657919883728 Detik
Pengujian ke 5	180.2909107208252 Detik
Pengujian ke 6	177.34695601463318 Detik.
Rata-rata Waktu Proses	186,0520125 Detik

Dari hasil perhitungan waktu proses ketiga algoritma tersebut maka dihasilkan perbandingan waktu proses seperti Gambar 4.11. Grafik rata-rata waktu proses algoritma dibawah ini :



Gambar 4.11. Grafik rata-rata waktu proses algoritma

Pada Gambar 4.11. Grafik rata-rata waktu proses algoritma dapat disimpulkan bahwa untuk waktu proses perhitungan pembentukan aturan asosiasi dengan menggunakan algoritma Eclat membutuhkan waktu paling lama rata-rata yaitu 186,0520125 detik, sedangkan FP-Growth adalah algoritma yang paling cepat rata-rata waktu prosesnya yaitu 0.691727281 detik.

Penulis melakukan pengujian menggunakan beberapa variasi dataset dengan tujuan menemukan penyebab salah satu algoritma membutuhkan waktu yang cukup lama, hasil pengujian tersebut dapat dilihat pada tabel berikut ini :

Tabel 4.13. Tabel pengujian algoritma

Ke	Record	Kolom	Diagnosa	Algoritma					
				Apriori		FP-Growth		ECLAT	
				Time	Memory	Time	Memory	Time	Memory
1	1000	5	5	0,10	938.804,58	0,12	959.983,38	0,03	305.249,94
2	7000	5	5	0,36	5.109.896,16	0,45	5.211.527,41	0,12	2.924.425,33
3	14672	5	5	0,80	10.287.302,93	0,84	10.275.577,12	0,23	7.544.707,74
4	1000	5	200	0,59	1.716.392,12	0,75	1.752.812,18	0,43	1.534.664,18
5	7000	5	200	0,76	8.148.146,60	0,33	8.170.647,82	0,62	4.561.796,58
6	14672	5	200	1,90	16.836.040,13	0,81	16.911.565,18	1,29	13.182.166,17
7	1000	5	2151	0,50	1.887.874,88	0,51	2.062.826,21	0,23	724.732,12
8	7000	5	2151	0,84	17.241.152,46	0,82	17.665.398,18	7,19	5.119.434,71
9	14672	5	2151	1,31	43.756.220,10	1,43	44.428.564,45	25,60	10.683.676,14
10	1000	16	200	0,14	729.196,11	0,12	736.578,85	0,05	306.988,52
11	7000	16	200	1,07	13.951.107,17	0,93	13.997.958,14	103,66	28.287.098,60
12	14672	16	200	3,56	26.316.454,36	1,67	26.298.502,30	107,83	54.149.617,12
13	1000	16	2151	0,53	1.754.330,78	0,61	1.992.811,20	1,00	1.352.129,19
14	7000	16	2151	0,94	18.353.267,51	0,65	18.913.964,19	229,49	110.565.635,29
15	14672	16	2151	1,72	46.298.600,11	1,64	47.263.634,47	349,64	272.918.182,78

Dari Tabel 4.13. Tabel pengujian algoritma di atas, maka dapat ditemukan beberapa temuan diantaranya adalah :

1. Pada pengujian ke 1 sampai dengan ke 3 dinyatakan bahwa perhitungan asosiasi dari dataset dengan jumlah record 1000, 7000 dan 15672, jumlah kolom 5 dan jumlah variabel diagnose 5, waktu yang diperlukan algoritma ECLAT paling sedikit atau paling cepat dibandingkan dengan algoritma

Apriori dan FP Growth. Pengujian ini menyatakan bahwa pada dataset dengan jumlah kolom dan jumlah variasi diagnose sedikit Algoritma ECLAT memerlukan waktu proses lebih cepat daripada algoritma Apriori dan FP-Growth.

2. Pada pengujian ke 7 pada dataset dengan jumlah record 1000, jumlah kolom 5 dan jumlah variabel diagnose 2151, algoritma ECLAT memerlukan waktu yang paling cepat dibandingkan algoritma Apriori dan FP-Growth, tetapi pada pengujian ke 8 dan ke 9 yaitu dataset dengan jumlah record 7000 dan 14672 terjadi bahwa algoritma ECLAT lebih lambat daripada algoritma Apriori dan FP-Growth. Pengujian ini menyatakan bahwa jumlah record sangat berpengaruh pada kecepatan proses dari Algoritma ECLAT.
3. Pada pengujian ke 9 menggunakan dataset yang memiliki jumlah record 14672, jumlah kolom 5, dan jumlah variabel diagnose 2151, ECLAT memperoleh kecepatan 25,60 detik, sedangkan pada pengujian 12 menggunakan dataset yang memiliki jumlah record 14672, jumlah kolom 16, dan jumlah variabel diagnose 200, ECLAT memperoleh kecepatan 107,83 detik. Hal ini dapat diartikan bahwa jumlah kolom berpengaruh terhadap kecepatan pada algoritma ECLAT.
4. Pada perbandingan hasil kecepatan algoritma ECLAT pada pengujian ke 6 dan 9, ataupun pengujian ke 12 dan 15 terlihat jumlah variabel diagnose berpengaruh terhadap kecepatan algoritma ECLAT, sedangkan pada algoritma Apriori dan FP-Growth tidak berpengaruh.

4.3.2. Perbandingan Kapasitas Memori

Penggunaan memori dihitung dengan menggunakan *library tracemalloc* pada python, proses yang dilakukan adalah mencari aturan asosiasi dengan nilai support 0.01 dan confidence 0.6 setelah diuji maka didapatkan perbedaan kapasitas penggunaan memori dari ketiga algoritma sebagai berikut :

4.3.2.1. Kapasitas Memori Algoritma Apriori

Syntax untuk menghitung kapasitas memori yang diperlukan pada algoritma apriori adalah sebagai berikut :

```
import tracemalloc
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import association_rules
from mlxtend.frequent_patterns import apriori, fpmax, fpgrowth
def app():
    lt = []
    for i in range(0, 100000):
        lt.append(i)
tracemalloc.start()
dataset = pd.read_csv("/content/drive/MyDrive/ADataset/inacbg-ranap-2022.csv")
all_transactions = dataset['DIAGLIST']
data = list(all_transactions.apply(lambda x:x.split(',')))

te = TransactionEncoder()
te_ary = te.fit(data).transform(data)
df = pd.DataFrame(te_ary, columns=te.columns_)
apriori(df, min_support=0.01, use_colnames=True)
tampil=apriori(df, min_support=0.01, use_colnames=True)
frequent_itemsets=apriori(df.astype('bool'),min_support=0.01,
use_colnames=True)
df_ar=association_rules(frequent_itemsets, metric="confidence",
min_threshold=0.1)
df_ar

df_ar[(df_ar.support > 0.01) & (df_ar.confidence >
0.6)].sort_values("confidence", ascending = False)
# displaying the memory
print(tracemalloc.get_traced_memory())
# stopping the library
Tracemalloc.stop()
```

Pengujian dilakukan 6 kali dengan hasil pada tabel berikut :

Tabel 4.14. Hasil perhitungan kapasitas memori algoritma apriori

Keterangan	Kapasitas Memori (Byte)	Kapasitas Memori (MB)
Pengujian ke 1	36808029,1363361 Byte	36,808029 MB
Pengujian ke 2	36806014,1363336 Byte	36,806014 MB
Pengujian ke 3	36804865,1363331 Byte	36,804865 MB
Pengujian ke 4	36806440,1363344 Byte	36,806440 MB
Pengujian ke 5	36807671,1363335 Byte	36,807671 MB
Pengujian ke 6	36807540,1363335 Byte	36,807540 MB
Rata-rata Kapasitas Memori		36,806760 MB

4.3.2.2. Kapasitas Memori Algoritma FP-Growth

Syntax untuk menghitung kapasitas memori yang diperlukan pada algoritma

FP-Growth adalah sebagai berikut :

```
import tracemalloc
import pandas as pd
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, fpmax, fpgrowth
# code or function for which memory
# has to be monitored
def app():
    lt = []
    for i in range(0, 100000):
        lt.append(i)
# starting the monitoring
tracemalloc.start()
from mlxtend.preprocessing import TransactionEncoder
dataset = pd.read_csv("/content/drive/MyDrive/ADataset/inacbg-
ranap-2022.csv")
all_transactions = dataset['DIAGLIST']
data = list(all_transactions.apply(lambda
x:x.split(', 'or'. ')))
```

```

te = TransactionEncoder()
te_ary = te.fit(data).transform(data)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = fpgrowth(df, min_support=0.01,
use_colnames=True)
frequent_itemsets['length'] =
frequent_itemsets['itemsets'].apply(lambda x: len(x))
freuent_itemsets=fpgrowth(df.astype('bool'),min_support=0.01,
use_colnames=True)
df_ar=association_rules(freuent_itemsets,
metric="confidence", min_threshold=0.6)
df_ar[(df_ar.support > 0.01) & (df_ar.confidence >
0.6)].sort_values("confidence", ascending = False)

```

Pengujian dilakukan 6 kali dengan hasil pada tabel berikut :

Tabel 4.15. Hasil perhitungan kapasitas memori algoritma fp-growth

Keterangan	Kapasitas Memori (Byte)	Kapasitas Memori (Mega Byte)
Pengujian ke 1	38747930,7050405 Byte	38,747930 MB
Pengujian ke 2	38747841,7050449 Byte	38,747841 MB
Pengujian ke 3	38768699,7052671 Byte	38,768699 MB
Pengujian ke 4	38744718,70501 Byte	38,744718 MB
Pengujian ke 5	38749820,7050697 Byte	38,749820 MB
Pengujian ke 6	38766556,7052398 Byte	38,766556 MB
Rata-rata Kapasitas Memori		38,754261 MB

4.3.2.3. Kapasitas Memori Algoritma ECLAT

Syntax untuk menghitung kapasitas memori yang diperlukan pada algoritma

ECLAT adalah sebagai berikut :


```

import tracemalloc
from pyECLAT import ECLAT
import pandas as pd
from mlxtend.frequent_patterns import association_rules
def app():
    lt = []
    for i in range(0, 100000):
        lt.append(i)
    tracemalloc.start()
    dataset = pd.read_csv('inacbg-ranap-2022.csv', header=None)
    dataset.dropna()
    dataset = dataset.applymap(str)
    eclat_instance = ECLAT(data=dataset, verbose=True)
    get_ECLAT_indexes, get_ECLAT_supports =
    eclat_instance.fit(min_support=0.01, min_combination=1, max_combination=2, separator=' & ', verbose=True)
    df =
    pd.DataFrame(get_ECLAT_supports.items(), columns=['itemsets',
    'support'])
    new_column = []
    for row in df['itemsets']:
        r = row.split('&')
        r = tuple(i.strip() for i in r)
        new_column.append(r)
    df['itemsets'] = pd.Series(new_column)
    df_ar = association_rules(df, metric="confidence",
    min_threshold=0.6)
    df_ar[(df_ar.support > 0.01) & (df_ar.confidence >
    0.6)].sort_values("confidence", ascending = False)
    akhir = time.time()

```

Pengujian dilakukan 6 kali dengan hasil pada tabel berikut :

Tabel 4.16. Hasil perhitungan kapasitas memori algoritma ECLAT

Keterangan	Kapasitas Memori (Byte)	Kapasitas Memori (MB)
Pengujian ke 1	271358325,780979 Byte	271,358325 MB
Pengujian ke 2	271508687,78112 Byte	271,508687 MB
Pengujian ke 3	271314291,780972 Byte	271,314291 MB
Pengujian ke 4	271418841,780975 Byte	271,418841 MB
Pengujian ke 5	271463766,78097 Byte	271,463766 MB
Pengujian ke 6	271714826,78112 Byte	271,714826 MB
Rata-rata Kapasitas Memori		271,463123 MB

Dari hasil perhitungan kapasitas memori ketiga algoritma tersebut maka dihasilkan perbandingan kapasitas memori seperti gambar dibawah ini :



Gambar 4.12. Grafik rata-rata penggunaan memori pada proses algoritma

Pada Gambar 4.12. Grafik rata-rata penggunaan memori pada proses algoritma dapat dilihat bahwa algoritma ECLAT membutuhkan kapasitas memori yang paling besar jika dibandingkan dengan kapasitas memori yang dibutuhkan algoritma apriori dan FP Growth untuk melakukan proses perhitungan pembentukan aturan asosiasi tersebut.

4.4. Implementasi Algoritma Asosiasi di Bidang Medis

Memilih algoritma asosiasi yang tepat untuk analisis data medis merupakan suatu hal yang krusial, mengingat presisi tinggi yang dibutuhkan dalam bidang ini. Kesalahan dalam memilih algoritma dapat berakibat pada hasil analisis yang tidak akurat, bahkan berpotensi meningkatkan risiko salah diagnosa. Berikut beberapa tantangan yang dihadapi dalam implementasi algoritma asosiasi di bidang medis :

a. Kualitas Data

Data medis seringkali tidak lengkap, tidak konsisten, dan mengandung noise. Hal ini dapat memengaruhi performa algoritma asosiasi dan menghasilkan pola yang tidak relevan.

Standarisasi data dan teknik pembersihan data yang tepat menjadi kunci untuk meningkatkan kualitas data dan meminimalisir dampak negatifnya terhadap analisis.

b. Dimensi Tinggi

Data medis umumnya memiliki dimensi yang tinggi, dengan banyak variabel dan atribut. Hal ini dapat menyebabkan masalah "curse of dimensionality", di mana algoritma kesulitan menemukan pola yang relevan karena banyaknya kemungkinan kombinasi variabel.

Teknik reduksi dimensi dan seleksi fitur yang tepat dapat membantu mengatasi masalah ini dan mempermudah analisis data.

c. Interpretasi Hasil

Algoritma asosiasi menghasilkan pola yang kompleks dan terkadang sulit untuk diinterpretasikan oleh manusia. Hal ini dapat membingungkan dokter dan ahli medis dalam memahami makna pola dan implikasinya terhadap diagnosis dan pengobatan.

Visualisasi data dan teknik interpretasi pola yang efektif dapat membantu dokter dan ahli medis dalam memahami hasil analisis dan membuat keputusan yang tepat.

d. Etika dan Privasi

Data medis bersifat sensitif dan mengandung informasi pribadi pasien. Penerapan algoritma asosiasi harus mempertimbangkan aspek etika dan privasi data pasien.

Anonimisasi data dan teknik privasi data yang tepat harus diterapkan untuk melindungi privasi pasien dan mematuhi peraturan yang berlaku.

e. Tantangan Implementasi Ke Depan

1. Pengembangan Algoritma yang Lebih Robust

- 1) Algoritma asosiasi yang lebih robust terhadap data yang tidak lengkap, tidak konsisten, dan bervolume tinggi perlu dikembangkan.
- 2) Algoritma yang mampu mengatasi masalah "curse of dimensionality" dan menghasilkan pola yang lebih mudah diinterpretasikan juga diperlukan.

2. Integrasi dengan Sistem Medis

- 1) Algoritma asosiasi perlu diintegrasikan dengan sistem medis yang ada untuk memudahkan akses dan penggunaan oleh dokter dan ahli medis.
- 2) Antarmuka yang intuitif dan mudah digunakan perlu dikembangkan untuk membantu dokter dalam memahami hasil analisis dan membuat keputusan yang tepat.

3. Penelitian dan Validasi

- 1) Penelitian lebih lanjut diperlukan untuk mengevaluasi performa algoritma asosiasi yang berbeda dalam konteks medis.
- 2) Validasi klinis dan penerapan algoritma dalam studi nyata perlu dilakukan untuk memastikan keefektifannya dalam meningkatkan diagnosis dan pengobatan.

4. Etika dan Privasi Data

- 1) Pedoman etika dan privasi data yang jelas perlu dirumuskan untuk memastikan penggunaan algoritma asosiasi yang bertanggung jawab dan sesuai dengan peraturan yang berlaku.
- 2) Peningkatan kesadaran dan edukasi tentang etika dan privasi data bagi para profesional medis dan pengembang algoritma juga diperlukan.

BAB V PENUTUP

5.1. Kesimpulan

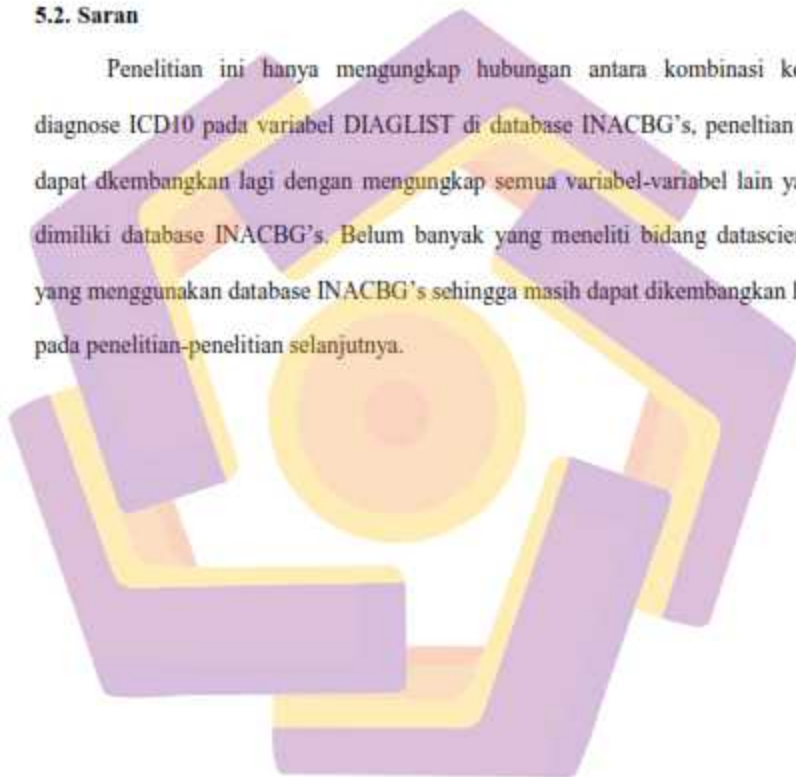
Berdasarkan hasil pengujian dan analisis yang telah dilakukan terhadap perhitungan frekuensi itemset dan aturan asosiatif data INACBG's dapat disimpulkan sebagai berikut :

1. Nilai confidence tertinggi sebesar 0.990025 dan nilai Conviction tertinggi sebesar 95.501 ditemukan pada Algoritma Apriori pada iterasi ke 18, sedangkan Algoritma Fp-Growth pada iterasi ke 21 dan Algoritma Eclat pada iterasi ke 5, pada itemset yang memiliki nilai *antecedents* (O821) dan nilai *consequents* (Z370) yaitu “Jika pasien melahirkan dengan tindakan operasi caesar sebagian besar melahirkan 1 bayi”. Hal ini menunjukkan hubungan yang sangat kuat dan akurat antara *antecedents* dan *consequents* tersebut.
2. Hasil perbandingan dari waktu proses perhitungan pembentukan aturan asosiasi dengan menggunakan algoritma Eclat membutuhkan rata-rata waktu paling lama yaitu 186,0520125 detik, sedangkan FP-Growth adalah algoritma yang paling cepat rata-rata waktu prosesnya yaitu 0,691727281 detik. Dalam proses pembentukan aturan asosiasi Algoritma ECLAT membutuhkan kapasitas memori yang paling besar yaitu 271,463123 MB sedangkan kapasitas memori yang dibutuhkan Algoritma Apriori sebesar 36,806760 MB dan FP-Growth membutuhkan memori sebesar 38,754261 MB untuk melakukan proses perhitungan pembentukan aturan asosiasi tersebut. Maka

dari hasil analisis ketiga perbandingan tadi peneliti menyimpulkan bahwa Algoritma FP-Growth adalah algoritma yang paling cocok dan efisien dalam melakukan proses pembentukan aturan asosiasi untuk dataset DIAGLIST pada database INACBG's.

5.2. Saran

Penelitian ini hanya mengungkap hubungan antara kombinasi kode diagnose ICD10 pada variabel DIAGLIST di database INACBG's, penelitian ini dapat dikembangkan lagi dengan mengungkap semua variabel-variabel lain yang dimiliki database INACBG's. Belum banyak yang meneliti bidang datascience yang menggunakan database INACBG's sehingga masih dapat dikembangkan lagi pada penelitian-penelitian selanjutnya.



DAFTAR PUSTAKA

PUSTAKA BUKU

- B. Santosa, 2007, "Data Mining : Teknik Pemanfaatan Data untuk Keperluan Bisnis", Yogyakarta: Graha Ilmu.
- Daniel T. Larose, 2005, *Discovering Knowledge in Data :An Introduction to Data mining*, JohnWiley's & Sons, Inc.
- Jiawei Han, Micheline Kamber, Jian Pei 2006, "Data Mining: Concepts and Techniques", Sanfrancisco: Morgan Kaufmann Publishers.
- Joko Suntoro, 2019, "Data Mining : Algoritma dan Implementasi dengan Pemrograman PHP". PT Elex Media Kompiutindo.
- Kusrini, luthfi taufiq Emha, 2009, "Algoritma Data Mining", Penerbit Andi, Yogyakarta.
- P.-N. Tan, M. Steinbach, and V. Kumar, 2006, "Introduction to Data Mining - Instructor's Solution Manual".
- Peraturan Presiden Republik Indonesia Nomor 111 Tahun 2013 tentang Jaminan Kesehatan.
- Peraturan Menteri Kesehatan Nomor 69 Tahun 2013 Tentang Standar Tarif Pelayanan Kesehatan Pada Fasilitas Kesehatan Tingkat Pertama dan Fasilitas Kesehatan Tingkat Lanjutan dalam Penyelenggaraan Program Jaminan Kesehatan.

PUSTAKA MAJALAN, JURNAL ATAU PROSIDING

- A. Ulumuddin and S. Juanita, 2018, "Implementasi Data Mining Dengan Metode Association Rule Pada Aplikasi Business Analytic Data Penjualan Menggunakan Algoritma Apriori," SKANIKA, vol. 1, no. 3, pp. 1212-1218.
- Agrawal, R. Srikant, R, 1994, "Fast Algorithm for Mining Association Rules. In Proceedings of the International Conference on Very Large Data Bases". IBM Almaden Research Center 650 Harry Road, San Jose, CA 95120.
- Andi Ilhamsyah Idris, Eliyah A M Sampetoding, Valian Yoga Pudya Ardhana, Irene Maritsa, Adrisumatri Sakri, Hidayatullah Ruslan, Esther Sanda Manapa, 2022, "Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores", "Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores", Hasanuddin University, Makassar, Indonesia - Universitas Qamarul Huda Badaruddin, Lombok, Indonesia, IJICS

(International Journal of Informatics and Computer Science), DOI 10.30865/ijics.v6i2.4535.

- D. Samuel, 2008, "Penerapan Stuktur FP-Tree dan Algoritma FP-Growth dalam Optimasi Penentuan Frequent Itemset", Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- Dr. V Srinadh, 2022, "Evaluation of Apriori, FP growth and Eclat association rule mining algorithms", International Journal of Health Sciences, DOI:<https://doi.org/10.53730/ijhs.v6nS2.6729>.
- Karina Intan Indradewi, Tia Atika Putri, 2019, "Penerapan Association Rule Dalam Bidang Kesehatan Pada Rumah Sakit X", Universitas Islam Indonesia, Seminar Nasional IENACO – 2019 ISSN : 2337 - 4349.
- M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, 1997, "New Algorithms for Fast Discovery of Association Rule". The University of Rochester Computer Science Department Rochester, New York 14627
- Moch Syahrir, Lalu Zazuli Azhar Mardedi, 2023, "Determination of the best rule-based analysis results from the comparison of the Fp-Growth, Apriori, and TPQ-Apriori Algorithms for recommendation systems". Bumigora University Indonesia, Jurnal Manajemen Teknologi dan Informatika, DOI:<http://dx.doi.org/10.31940/matrix.v13i2.52-67>.
- Nola Ritha, Eka Suswaini Wisnu Pebriadi, 2021, "Penerapan Association Rule Menggunakan Algoritma Apriori Pada Poliklinik Penyakit Dalam (Studi Kasus: Rumah Sakit Umum Daerah Bintan)", Jurnal Sains dan Informatika, ISSN e-ISSN: 2598-5841.
- Putthiporn Thanathamthe, Siriporn Sawangarreearak, 2022, "Discovering Future Earnings Patterns through FP-Growth and ECLAT Algorithms with Optimized Discretization", Emerging Science Journal, ISSN 2610-9182
- R. Mustofa and Irhamah, 2019, "Topic Discovery pada Jurnal-jurnal di IEEE Explore menggunakan Association Rule Mining dengan Pendekatan Closed Frequent Itemset," Ejournal.lts.Ac.Id, vol. 8, no. 2.

PUSTAKA ELEKTRONIK

- trivusi.web.id, 2022, "Algoritma Apriori: Pengertian, Cara Kerja, Kelebihan, dan Kekurangannya", <https://www.trivusi.web.id/2022/08/algoritma-apriori.html>
- who.int, 2023, "ICD-10 Version:2010", <https://icd.who.int/browse10/2010/en>