

Scalability Analysis of Frequent Closed High Utility Itemset Mining on Multi-Year Retail Transaction Data

Kinana Syah Sulanjari ^{1*)} and Chastine Fatichah ²⁾

¹Institut Teknologi Sepuluh November, Surabaya, Indonesia
kinana.syahsulanjari@gmail.com¹⁾ and chastine@its.ac.id²⁾

ABSTRACT

Frequent Closed High Utility Itemset Mining (FCHUIM) is a vital approach for discovering high-value patterns from transactional data. However, a major challenge arises as historical data volume grows substantially over time, particularly in dynamic retail domains. This study aims to analyze the scalability of the Closed-FHUIM algorithm with respect to increasing volumes of multi-year retail cooperative transaction data, spanning from one to five years. The evaluation focuses on four key performance metrics: execution time, memory usage, number of discovered patterns, and pattern growth rate. Experiments were conducted incrementally using annual transaction datasets. The results show that execution time grows exponentially with data volume, while the number of patterns increases significantly in the early years and plateaus in later periods. Memory usage exhibits fluctuating behavior influenced by transaction structures, and the pattern growth rate gradually declines as the data span widens. These findings suggest that although Closed-FHUIM is effective for high-utility pattern discovery, further optimization is required for deployment in large-scale and longitudinal retail scenarios.

Keywords: FCHUIM, Scalability Evaluation, Multi Year Transaction, Retail Data

ABSTRAK

Penambangan *frequent closed high utility itemset* (FCHUI) merupakan pendekatan penting dalam menemukan pola-pola bernilai tinggi dari data transaksi. Namun, tantangan utama muncul ketika volume data historis meningkat secara signifikan dari tahun ke tahun, khususnya dalam domain ritel yang bersifat dinamis. Penelitian ini bertujuan untuk menganalisis skalabilitas algoritma Closed-FHUIM terhadap pertumbuhan data transaksi tahunan koperasi ritel dari satu hingga lima tahun. Evaluasi dilakukan terhadap empat metrik utama, yaitu waktu eksekusi, penggunaan memori, jumlah pola yang ditemukan, dan laju pertumbuhan pola (*growth rate*). Eksperimen dilakukan secara bertahap menggunakan *dataset* transaksi tahunan, dan menunjukkan bahwa waktu eksekusi meningkat secara eksponensial, sementara jumlah pola mengalami pertumbuhan tinggi pada tahun-tahun awal namun melandai setelahnya. Penggunaan memori menunjukkan pola fluktuatif yang dipengaruhi oleh karakteristik transaksi, sedangkan *growth rate* pola menurun secara progresif seiring bertambahnya cakupan data. Temuan ini menunjukkan bahwa Closed-FHUIM efektif dalam mengidentifikasi pola-pola bernilai tinggi, namun memerlukan optimasi lebih lanjut untuk diterapkan dalam skenario berskala besar.

Kata kunci: FCHUIM, Evaluasi Skalabilitas, Transaksi Multi Tahun, Data Ritel

PENDAHULUAN

Industri ritel modern menghasilkan data transaksi dalam jumlah besar setiap harinya, dan volume tersebut terus meningkat seiring perkembangan sistem *Point of*

Sales (POS) dan platform *e-commerce* [1]. Akumulasi data historis ini membuka peluang besar untuk menggali wawasan yang bernilai melalui proses penambangan pola (*pattern mining*), khususnya dalam mengidentifikasi perilaku belanja konsumen yang berdampak langsung terhadap strategi pemasaran dan pengambilan keputusan bisnis [11].

Salah satu pendekatan awal yang populer dalam analisis pola transaksi adalah *Frequent Itemset Mining* (FIM), yang bertujuan untuk menemukan *itemset* yang sering muncul di dalam *dataset*. Namun, FIM tidak memperhitungkan kontribusi ekonomi dari setiap item, sehingga pola bernilai tinggi secara finansial dapat terlewat [1]. Untuk mengatasi keterbatasan tersebut, *High Utility Itemset Mining* (HUIM) dikembangkan sebagai metode yang mempertimbangkan tidak hanya frekuensi kemunculan, tetapi juga utilitas dari suatu item, seperti keuntungan, volume pembelian, atau nilai transaksi [7] [8]

Salah satu varian HUIM yang penting adalah *Frequent Closed High Utility Itemset Mining* (FCHUIM), yang menggabungkan efisiensi penambangan *itemset* tertutup dengan fokus pada nilai utilitas untuk mengurangi redundansi pola tanpa kehilangan informasi penting [5][7]. FCHUIM telah terbukti efektif dalam mengekstraksi pola bernilai tinggi yang relevan secara bisnis, sekaligus menyederhanakan hasil analisis.

Namun demikian, studi terkait kemampuan algoritma FCHUIM dalam menangani *dataset* berskala besar dan data historis jangka panjang masih tergolong terbatas. Hal ini menjadi krusial mengingat organisasi ritel seperti koperasi dan toko jaringan cenderung menyimpan data transaksi secara bertahun-tahun. Dengan meningkatnya volume data dari waktu ke waktu, efisiensi komputasi, konsumsi memori, dan kualitas pola hasil algoritma perlu dievaluasi untuk memastikan penerapannya tetap optimal dalam konteks nyata [9].

Oleh karena itu, penelitian ini berfokus pada evaluasi skalabilitas algoritma *Frequent Closed High Utility Itemset Mining* (FCHUIM) dengan menggunakan data transaksi ritel dari koperasi selama periode tiga tahun. Fokus utama adalah mengukur kinerja algoritma dalam hal waktu proses, efisiensi memori, dan kualitas pola hasil seiring peningkatan volume data. Diharapkan, hasil penelitian ini dapat memberikan kontribusi praktis terhadap penerapan algoritma FCHUIM secara berkelanjutan dalam sistem analitik data ritel yang dinamis dan berkembang.

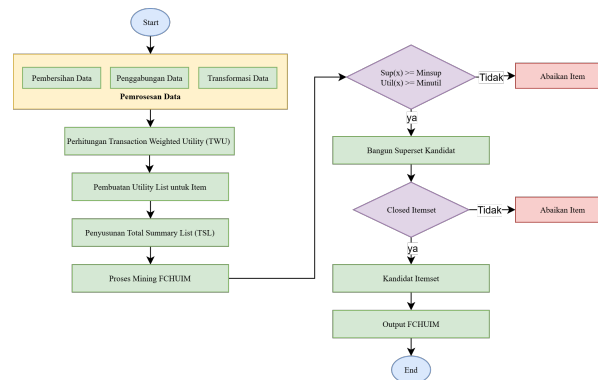
METODE PENELITIAN

Bagian ini menjelaskan tahapan metodologi yang digunakan dalam penelitian, mulai dari desain eksperimen, pra proses data, penerapan algoritma FCHUIM, hingga pengujian skalabilitas berdasarkan pertumbuhan volume data. Penelitian ini dilakukan secara eksperimental untuk mengevaluasi kinerja algoritma FCHUIM terhadap data transaksi ritel multi-tahun.

Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental yang bertujuan untuk menganalisis kemampuan skalabilitas algoritma *Frequent Closed High Utility Itemset Mining* (FCHUIM) dalam menghadapi peningkatan volume data

transaksi dari waktu ke waktu. Fokus utama adalah melihat dampak pertumbuhan data terhadap performa algoritma dari segi waktu proses, konsumsi memori, dan jumlah pola yang dihasilkan.



Gambar 1. Flowchart of the Closed-FHUIM Algorithm with OSR, OWL and MSU Pruning

HASIL DAN PEMBAHASAN

Pemrosesan Data

Tahap pra proses data merupakan langkah krusial dalam menyiapkan *dataset* agar sesuai dengan struktur dan format yang dapat diproses secara efisien oleh algoritma *Frequent Closed High Utility Itemset Mining* (FCHUIM). Proses ini bertujuan untuk memastikan bahwa data yang digunakan bersih, konsisten, dan relevan, sehingga hasil *mining* yang diperoleh valid dan bermakna secara bisnis [21]. Adapun tahapan pra proses data dalam penelitian ini mencakup tiga bagian utama: pembersihan data, integrasi data, dan transformasi data.

1. Pembersihan Data (*Data Cleaning*)

Langkah pertama adalah membersihkan *dataset* dari entri yang tidak valid. Proses ini meliputi:

- Penghapusan duplikasi: Transaksi yang tercatat lebih dari satu kali dengan informasi yang sama akan dihapus untuk menghindari bias pada hasil *mining* [12].
- Eliminasi data tidak lengkap: Transaksi yang tidak mencantumkan ID item, kuantitas, atau harga unit akan dihapus karena tidak dapat digunakan dalam perhitungan utilitas.
- Penyaringan atribut tidak relevan: Atribut seperti metode pembayaran, nama kasir, atau lokasi toko tidak digunakan dalam algoritma FCHUIM dan dihapus dari *dataset* untuk mengurangi kompleksitas dan meningkatkan efisiensi analisis [10].

2. Integrasi Data (*Data Integration*)

Tahap ini menggabungkan dua sumber data utama, yaitu:

- Data transaksi penjualan, yang berisi ID transaksi, ID produk, kuantitas pembelian, dan tanggal transaksi.

- Data master produk, yang mencakup ID produk, nama produk, harga satuan (unit profit), dan kategori produk.

Melalui proses *join* berdasarkan ID produk, setiap transaksi diperkaya dengan informasi harga satuan yang diperlukan untuk menghitung nilai utilitas. Integrasi ini penting agar data dari berbagai sistem dapat digunakan secara konsisten dan lengkap dalam proses *mining* [13].

3. Transformasi Data (*Data Transformation*)

Setelah data dibersihkan dan digabungkan, tahap transformasi dilakukan untuk mengubah struktur data menjadi format yang sesuai dengan *input* FCHUIM. Tahap ini mencakup:

- Pembentukan format transaksi standar, di mana setiap baris mencerminkan satu transaksi yang terdiri dari:
 - ID transaksi
 - ID item
 - Kuantitas pembelian
 - Harga satuan
 - Nilai utilitas (kuantitas × harga satuan)
- Perhitungan nilai transaksi total (*Transaction Utility*): Untuk setiap transaksi, dihitung jumlah total utilitas dari semua item di dalamnya. Nilai ini penting untuk proses penyaringan berdasarkan ambang utilitas.
- Pengurutan item berdasarkan TWU (*Transaction-Weighted Utility*): Item disusun dalam urutan menurun berdasarkan kontribusi utilitas totalnya, untuk mendukung proses pencarian pola secara efisien oleh algoritma.

Hasil akhir dari tahapan pra proses ini adalah *dataset* terstruktur dalam bentuk *utility table* dan *transaction utility list*, yang masing-masing menyimpan informasi detail tentang utilitas item dan total utilitas per transaksi. *Dataset* ini kemudian digunakan sebagai *input* utama dalam proses eksplorasi pola dengan algoritma FCHUIM.

FCHUIM Algorithm

Closed-Frequent and High Utility Itemset Mining (Closed-FHUIM) merupakan algoritma yang digunakan untuk menemukan pola *itemset* yang bersifat tertutup serta memiliki nilai utilitas tinggi dari data transaksi ritel. Proses algoritma ini dilaksanakan melalui beberapa tahapan utama berikut:

- **Konstruksi *Utility List***

Setiap item tunggal dalam *dataset* dibentuk menjadi struktur *utility list* yang berisi tiga komponen utama:

- TID (*Transaction ID*): Identifikasi transaksi tempat item muncul.
- IUtil (*Internal Utility*): Nilai utilitas item dalam suatu transaksi, dihitung dari kuantitas × profit per unit.

- RUtil (*Remaining Utility*): Estimasi utilitas item lain yang muncul setelah item tersebut dalam urutan transaksi.

Struktur ini memungkinkan perhitungan efisien selama proses pencarian tanpa penghitungan ulang utilitas.

- **Filtering Berdasarkan Threshold**

Setiap item diperiksa nilai *support* dan *utility*-nya. Item yang tidak memenuhi nilai minimum *support* (*minsup*) atau minimum *utility* (*minutil*) langsung dieliminasi dari proses lebih lanjut.

- **Ekstensi Itemset dengan Depth-First Search (DFS)**

Item yang lolos dari tahap awal diperluas secara rekursif menggunakan metode DFS. Kandidat *itemset* dibentuk berdasarkan kombinasi item yang tersisa dalam *utility list*.

- **Evaluasi Closedness**

Pada setiap tahap eksplorasi, sistem mengecek apakah *itemset* yang dihasilkan merupakan *closed*, yakni tidak ada *superset* dengan nilai *support* dan utilitas yang sama. *Itemset* yang tidak tertutup akan diabaikan untuk menghindari duplikasi pola.

- **Penyimpanan Pola Valid**

Itemset yang memenuhi kriteria sebagai *frequent*, *high utility*, dan *closed* disimpan sebagai hasil akhir. Hanya pola yang unik, bernilai tinggi, dan representatif yang dicatat.

Eksperimen Skalabilitas

Eksperimen dilakukan menggunakan data transaksi ritel dari koperasi selama periode lima tahun berturut-turut. Untuk mengamati dampak pertumbuhan data historis terhadap performa algoritma, *dataset* digunakan secara bertahap dan kumulatif melalui empat skenario eksperimental:

- Skenario 1: Data tahun ke-1 hingga ke-2 (2 tahun)
- Skenario 2: Data tahun ke-1 hingga ke-3 (3 tahun)
- Skenario 3: Data tahun ke-1 hingga ke-4 (4 tahun)
- Skenario 4: Data tahun ke-1 hingga ke-5 (5 tahun)

Pendekatan bertahap ini memungkinkan observasi yang terstruktur dan sistematis terhadap penurunan performa, peningkatan kompleksitas komputasi, dan potensi perubahan dalam pola hasil seiring dengan bertambahnya volume data historis. Setiap eksperimen dijalankan menggunakan parameter minimum *support* dan minimum *utility* yang konsisten, guna menjaga validitas perbandingan antar skenario.

Tujuan dari eksperimen ini adalah untuk mengevaluasi sejauh mana algoritma FCHUIM dasar mampu mempertahankan efisiensi dan efektivitasnya saat dihadapkan pada pertumbuhan data berskala waktu nyata, sebagaimana umum terjadi dalam sistem informasi koperasi dan ritel.

Evaluasi dan Analisis

Untuk mengukur skalabilitas, digunakan tiga metrik evaluasi:

- Waktu Eksekusi (detik): Lama waktu proses dari awal hingga akhir.
- Penggunaan Memori (MB): Puncak memori yang dikonsumsi selama proses *mining*.
- Jumlah *Itemset* yang Dihasilkan: Total pola FCHUIM yang ditemukan pada setiap skenario.

Hasil dari ketiga metrik ini dibandingkan antar skenario *dataset* untuk mengidentifikasi tren performa terhadap volume data.

Experiment and Discussion

Bab ini menyajikan hasil eksperimen yang dilakukan untuk mengevaluasi kinerja algoritma Closed-FHUIM dalam konteks data transaksi koperasi ritel. Evaluasi dilakukan terhadap berbagai aspek, termasuk waktu eksekusi, penggunaan memori, jumlah pola yang ditemukan serta tingkat pertumbuhan pola dari tahun ke tahun (*growth rate*) Eksperimen dilakukan secara bertahap dengan menambahkan data transaksi dari satu hingga lima tahun guna menilai skalabilitas dan stabilitas performa algoritma terhadap peningkatan volume data historis.

Experimental Setup

Seluruh eksperimen dalam penelitian ini dijalankan menggunakan perangkat komputer dengan spesifikasi sebagai berikut: prosesor Intel Core i7-1165G7, memori utama (RAM) sebesar 16 GB, dan sistem operasi Windows 11 Pro 64-bit. Algoritma Closed-FHUIM diimplementasikan dalam bahasa pemrograman Python versi 3.10, dan dijalankan di lingkungan Jupyter Notebook yang memungkinkan eksplorasi interaktif, visualisasi hasil, serta fleksibilitas dalam pengaturan parameter dan pemantauan performa.

Eksperimen dilakukan untuk mengevaluasi skalabilitas algoritma terhadap penambahan volume data historis. *Dataset* yang digunakan merupakan data transaksi koperasi ritel selama lima tahun berturut-turut, dari Januari 2020 hingga Desember 2024. Evaluasi dilakukan secara bertahap berdasarkan cakupan tahunan, dimulai dari satu tahun hingga lima tahun data, untuk mengamati dampak pertumbuhan data terhadap waktu eksekusi, penggunaan memori, dan jumlah pola yang dihasilkan.

Tabel 1 Experimental Parameter Configuration

Parameter	Value Used
Minimum support (minsup)	20
Minimum utility (minutil)	50.000
Dataset	1-5 Tahun (berjenjang)
Evaluation Metrics	Execution Time, Memory Usage, Number of Itemsets

Tabel 2. Dataset

Dataset ke-	Periode Tahun	Jumlah Transaksi
1 Tahun	2024	1650
2 Tahun	2023 – 2024	3435
3 Tahun	2022 – 2024	5105
4 Tahun	2021 – 2024	6694, 5255
5 Tahun	2020 – 2024	8875, 5646

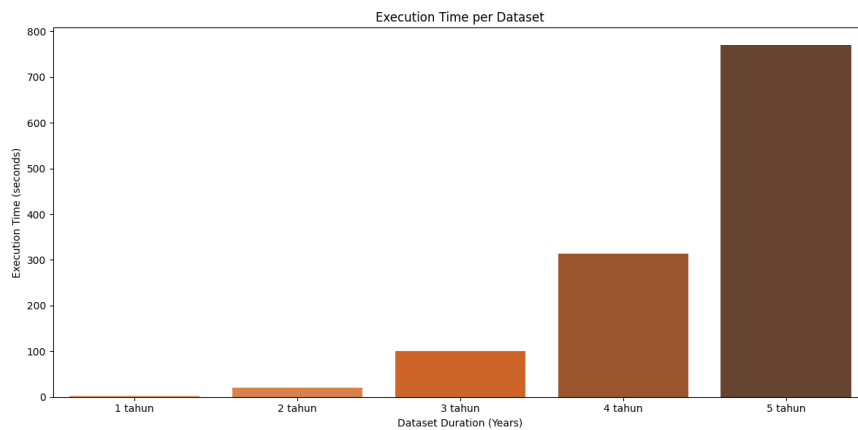
Melalui desain eksperimen berjenjang ini, dapat dianalisis bagaimana kinerja algoritma Closed-FHUIM berubah seiring pertambahan data historis. Evaluasi mencakup empat metrik utama: waktu eksekusi (detik), penggunaan memori (MB), jumlah *itemset* yang ditemukan, laju pertumbuhan jumlah pola (*growth rate*), serta rata-rata utilitas per pola yang dihasilkan. Pendekatan ini memberikan gambaran menyeluruh terhadap efisiensi algoritma dalam skenario data berskala besar dan longitudinal.

Evaluasi *Scalability*

Evaluasi skalabilitas dilakukan untuk mengukur kinerja algoritma FCHUIM terhadap peningkatan jumlah data historis yang digunakan dalam proses penambangan pola. Pengujian dilakukan secara bertahap menggunakan *dataset* transaksi koperasi selama lima tahun, dimulai dari satu tahun hingga lima tahun, guna mengamati bagaimana waktu eksekusi, penggunaan memori, dan jumlah pola yang dihasilkan berubah seiring bertambahnya volume data. Evaluasi ini penting untuk menilai sejauh mana algoritma dapat diterapkan secara efisien dalam konteks data ritel jangka panjang yang bersifat dinamis dan terus berkembang.

Tabel 3. Waktu Eksekusi

Dataset ke-	Waktu Eksekusi (Sec)
1 Tahun	2.82
2 Tahun	20.77
3 Tahun	100.78
4 Tahun	313.75
5 Tahun	769.51

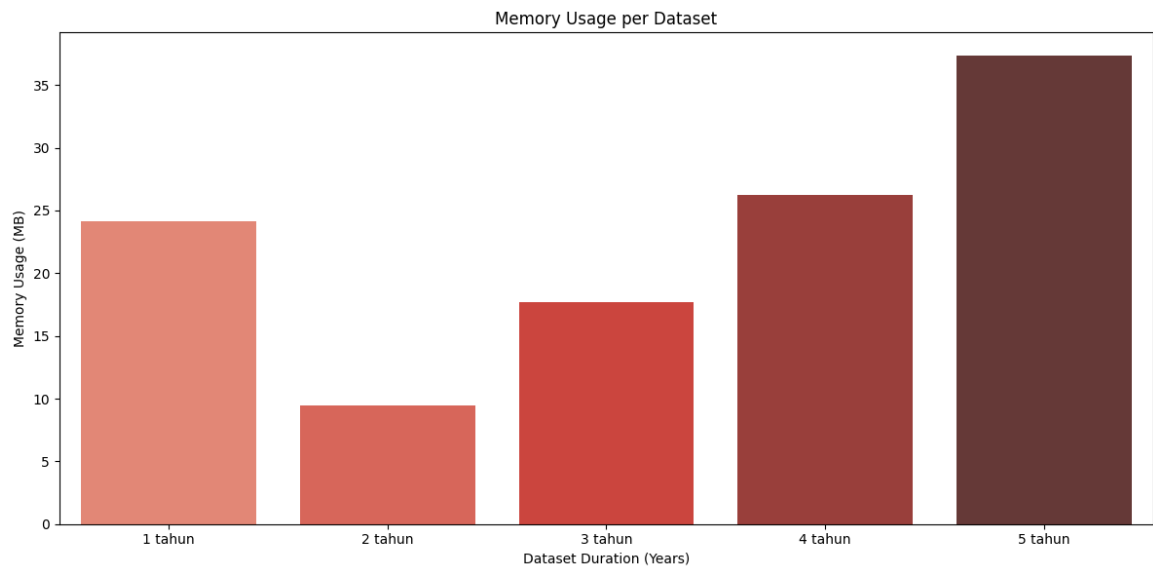


Gambar 2. Grafik Dataset

Waktu eksekusi algoritma Closed-FHUIM meningkat secara eksponensial seiring dengan bertambahnya jumlah data historis yang digunakan. Untuk *dataset* satu tahun, waktu eksekusi relatif rendah, yakni 2,82 detik. Namun, saat data diperluas menjadi dua tahun, waktu melonjak tajam menjadi 20,77 detik. Pertumbuhan ini berlanjut dengan signifikan pada tiga tahun (100,78 detik), empat tahun (313,75 detik), hingga mencapai 769,51 detik pada *dataset* lima tahun.

Kenaikan waktu ini menunjukkan bahwa algoritma Closed-FHUIM memiliki sensitivitas tinggi terhadap jumlah transaksi yang dianalisis. Hal ini dapat dijelaskan oleh peningkatan jumlah kombinasi item yang mungkin terbentuk dalam *dataset* yang lebih besar, serta tingginya kompleksitas perhitungan *utility* dan evaluasi pola tertutup untuk setiap kandidat *itemset*. Proses evaluasi pola yang dilakukan secara rekursif dan eksploratif juga turut berkontribusi terhadap waktu komputasi yang semakin besar.

Meskipun waktu eksekusi meningkat secara signifikan, performa algoritma masih dalam rentang yang dapat diterima untuk eksperimen skala menengah. Namun demikian, untuk penggunaan di lingkungan produksi atau skenario *big data*, optimasi lanjutan seperti penerapan teknik paralelisme atau *pruning* berbasis heuristik tetap direkomendasikan agar efisiensi waktu dapat dijaga tanpa mengorbankan kualitas pola.



Gambar 3. Grafik Penggunaan Memori

Tabel 4. Penggunaan Memori

Dataset ke-	Penggunaan Memori (MB)
1 Tahun	24.17
2 Tahun	9.45
3 Tahun	17.72
4 Tahun	26.24
5 Tahun	37.32

Penggunaan memori menunjukkan tren yang lebih fluktuatif dibandingkan waktu eksekusi. Pada *dataset* satu tahun, penggunaan memori mencapai 24,17 MB, namun menurun menjadi 9,45 MB pada dua tahun, kemudian meningkat kembali pada tiga tahun (17,72 MB), empat tahun (26,24 MB), dan lima tahun (37,32 MB).

Variasi ini mengindikasikan bahwa konsumsi memori algoritma tidak hanya dipengaruhi oleh volume data, tetapi juga oleh struktur dan karakteristik distribusi transaksi di dalam *dataset*. Misalnya, jika dalam *dataset* dua tahun terdapat banyak transaksi dengan item serupa atau repetitif, struktur *utility list* dan evaluasi kandidat bisa menjadi lebih terkompresi, sehingga mengurangi kebutuhan memori sementara.

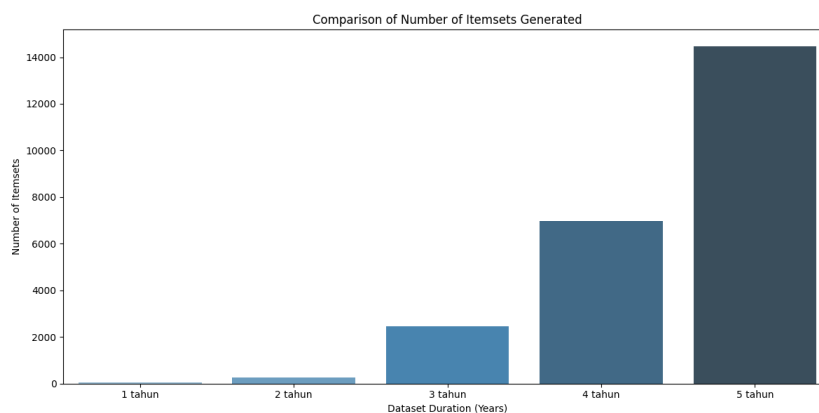
Namun, secara umum terdapat kecenderungan peningkatan memori seiring waktu, yang menggambarkan bertambahnya jumlah *utility list*, kombinasi *itemset*, dan informasi *intermediate* yang perlu disimpan selama proses penambangan. Lonjakan pada *dataset* lima tahun menunjukkan bahwa saat data historis bertambah panjang dan kompleks, struktur data yang dikelola oleh algoritma juga tumbuh secara signifikan.

Fluktuasi ini juga menjadi pertimbangan penting dalam implementasi algoritma pada sistem nyata, khususnya pada sistem dengan keterbatasan sumber daya. Oleh karena itu, teknik pengelolaan memori seperti *lazy loading*, *buffering*, dan

memory-efficient pruning dapat menjadi solusi untuk memastikan skalabilitas algoritma dalam jangka panjang.

Tabel 5. Jumlah Pola Ditemukan

Dataset ke-	Jumlah Pola Ditemukan
1 Tahun	56
2 Tahun	258
3 Tahun	2458
4 Tahun	6968
5 Tahun	14456



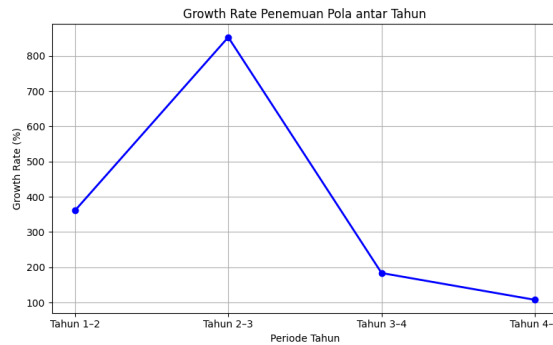
Gambar 4. Grafik Jumlah Pola Ditemukan

Jumlah pola *frequent closed high utility itemset* (FCHUI) yang ditemukan mengalami peningkatan signifikan seiring dengan bertambahnya data historis yang digunakan dalam eksperimen. Pada *dataset* satu tahun, algoritma menghasilkan 56 pola. Jumlah ini meningkat drastis menjadi 258 pola untuk dua tahun, 2.458 pola untuk tiga tahun, 6.968 pola untuk empat tahun, dan mencapai 14.456 pola untuk lima tahun.

Peningkatan jumlah pola ini secara umum menggambarkan bahwa semakin panjang rentang waktu data yang diamati, semakin kompleks dan variatif pula hubungan antar item dalam transaksi ritel. Hal ini dapat dimaknai sebagai indikasi bahwa pola pembelian konsumen berkembang dan menjadi lebih kaya dalam jangka panjang, serta adanya akumulasi variasi produk, promosi, dan perilaku pembelian.

Namun demikian, perlu dicermati bahwa konfigurasi nilai *minimum support* (*minsup*) dan *minimum utility* (*minutil*) yang digunakan dalam eksperimen bersifat tetap dan tidak disesuaikan terhadap pertumbuhan volume data. Akibatnya, ketika jumlah transaksi meningkat, peluang *itemset* untuk melampaui ambang dukungan (*support*) dan utilitas minimum juga meningkat secara alami, sehingga menghasilkan lebih banyak pola. Dengan kata lain, banyaknya pola yang ditemukan tidak hanya mencerminkan kompleksitas data, tetapi juga dipengaruhi oleh ketetapan parameter yang mungkin kurang adaptif terhadap skala data.

Situasi ini membuka ruang untuk evaluasi lebih lanjut terkait optimalisasi parameter *minsup* dan *minutil* berbasis dinamika *dataset*. Penyesuaian adaptif terhadap ukuran *dataset* dapat membantu dalam menghasilkan jumlah pola yang lebih representatif dan relevan, serta menghindari *overfitting* informasi yang tidak signifikan secara bisnis.



Gambar 5. Growth Rate of Pattern Discovery

Pertumbuhan jumlah pola yang ditemukan antar tahun menunjukkan dinamika yang signifikan seiring bertambahnya cakupan data historis. Dari Tahun 1 ke Tahun 2, terjadi peningkatan sebesar 360,71%, yang mencerminkan efek awal dari penambahan data dalam memperluas kombinasi item yang memenuhi ambang *support* dan *utility*. Laju pertumbuhan ini melonjak tajam pada periode Tahun 2 ke Tahun 3, mencapai 852,71%. Lonjakan ini menunjukkan bahwa pada fase pertengahan, penambahan satu tahun data baru mampu mengungkap banyak pola tersembunyi yang sebelumnya tidak dapat dideteksi karena keterbatasan cakupan transaksi.

Namun, mulai Tahun 3 ke Tahun 4, tren pertumbuhan mulai menurun secara signifikan menjadi 183,48%, dan kembali menurun menjadi 107,46% pada transisi Tahun 4 ke Tahun 5. Penurunan ini mengindikasikan bahwa proses penemuan pola mulai mengalami kejenuhan—di mana sebagian besar pola bernilai tinggi telah ditemukan pada tahun-tahun sebelumnya. Meskipun data baru terus masuk, kontribusinya terhadap jumlah pola baru yang ditemukan tidak lagi setajam pada fase awal.

Fenomena ini menunjukkan bahwa penambahan data historis secara bertahap memberikan dampak eksponensial di awal, namun menjadi lebih terbatas pada periode selanjutnya. Visualisasi dalam bentuk *line chart* memperjelas tren ini, dengan kurva yang menanjak tajam pada awal, lalu melandai setelah memasuki tahun keempat dan kelima. Pola ini penting untuk dipertimbangkan dalam implementasi algoritma Closed-FHUIM di lingkungan produksi, agar evaluasi terhadap efisiensi dan kebutuhan komputasi tetap proporsional terhadap nilai tambah pola yang dihasilkan.

KESIMPULAN

Penelitian ini menunjukkan bahwa algoritma Closed-FHUIM mampu menambang pola *frequent closed high utility itemset* secara efektif pada data transaksi koperasi ritel berskala tahunan hingga lima tahun. Evaluasi terhadap aspek waktu

eksekusi, penggunaan memori, jumlah pola ditemukan, *growth rate*, dan utilitas per pola menunjukkan bahwa algoritma ini dapat menangani peningkatan volume data historis secara progresif, meskipun dengan peningkatan kompleksitas komputasi.

Eksperimen membuktikan bahwa waktu eksekusi meningkat secara eksponensial seiring bertambahnya cakupan data, sementara penggunaan memori menunjukkan tren fluktuatif yang dipengaruhi oleh struktur transaksi. Jumlah pola yang ditemukan tumbuh sangat signifikan terutama pada penambahan data awal (1 ke 3 tahun), namun pertumbuhannya mulai melambat setelahnya, mengindikasikan fase kejenuhan pola.

Kinerja algoritma secara keseluruhan masih berada dalam batas wajar untuk skenario eksploratif, namun untuk penggunaan skala produksi atau big data, diperlukan pengembangan lebih lanjut seperti adaptasi parameter (*minsup*, *minutil*) secara dinamis serta penerapan teknik optimasi memori dan eksekusi paralel.

DAFTAR PUSTAKA

- Ashraf, M. W., Naeem, M. A., & Lee, H. J. (2024). A robust technique for closed frequent and high utility itemsets mining: Closed-FHUI. *IEEE Access*, 12, 196517–196532. <https://doi.org/10.1109/ACCESS.2024.3520619> [5]
- Borah, A., & Nath, B. (2021). Comparative evaluation of pattern mining techniques: An empirical study. *Complex & Intelligent Systems*, 7(2), 589–619. <https://doi.org/10.1007/s40747-020-00226-4> [8]
- Dekimpe, M. G. (2020). Retailing and retailing research in the age of big data analytics. *International Journal of Research in Marketing*, 37(1), 3–14. <https://doi.org/10.1016/j.ijresmar.2019.09.001> [1]
- Duong, H., Pham, H., Truong, T., & Fournier-Viger, P. (2024). Efficient algorithms to mine concise representations of frequent high utility occupancy patterns. *Applied Intelligence*, 54(5), 4012–4042. <https://doi.org/10.1007/s10489-024-05296-2> [6]
- Gajera, R., Patel, S., Madhani, K., & Solanki, A. (2024). An efficient join operations for utility list-based high-utility mining approaches using hybrid search technique. *International Journal of Data Science and Analytics*. <https://doi.org/10.1007/s41060-024-00538-5> [9]
- Gupta, S., & Ramachandran, D. (2021). Emerging market retail: Transitioning from a product-centric to a customer-centric approach. *Journal of Retailing*, 97(4), 597–620. <https://doi.org/10.1016/j.jretai.2021.01.008> [2]
- Kumar, S., Yadav, R. K., & Singh, C. (2024). Fuzzy set-based inventory model for TPD demand under effect of inflation and carbon emissions with partial backordering. *Journal of Computational Analysis and Applications*, 33(8). [4]
- Malik, M. H., Ghous, H., Ismail, M., Jamshaid, S., & Altaf, J. (n.d.). Market basket analysis for next basket item prediction using data mining and machine learning. *Biomedical Informatics*. [10]

- Pratama, I. W. P. (2024). Exploring the depths of market basket analysis: A comprehensive guide to transaction analysis with FP-growth and Apriori algorithms. *INVOTEK*, 23(2), 109–118. <https://doi.org/10.24036/invotek.v23i2.1094> [11]
- Hermawan, A., Wicaksono, B., Ahmadjayadi, T., Prakasa, B. S., & Aruan, J. D. (2024). Implementasi algoritma Apriori pada market basket analysis terhadap data penjualan produk supermarket. *Algoritma*, 2(5), 95–105. <https://doi.org/10.62383/algoritma.v2i5.137> [12]
- Rao, K. M., Saikrishna, G., & Supriya, K. (2023). Data preprocessing techniques: Emergence and selection towards machine learning models – A practical review using HPA dataset. *Multimedia Tools and Applications*, 82(24), 37177–37196. <https://doi.org/10.1007/s11042-023-15087-5> [13]
- Tung, N. T., Nguyen, L. T. T., Nguyen, T. D. D., & Vo, B. (2022). An efficient method for mining multi-level high utility itemsets. *Applied Intelligence*, 52(5), 5475–5496. <https://doi.org/10.1007/s10489-021-02681-z> [3]
- Zhao, X., Zhong, X., & Han, B. (2024). Frequent closed high-utility itemset mining algorithm based on Leiden community detection and compact genetic algorithm. *IEEE Access*, 12, 84763–84773. <https://doi.org/10.1109/ACCESS.2024.3413966> [7]