

**PENEMUAN POLA TRANSAKSI PENJUALAN
DENGAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN*
STUDI KASUS UD. JATI MULYA**

SKRIPSI

**Untuk memenuhi sebagian
persyaratan untuk mencapai gelar Sarjana Komputer**



Disusun oleh :

AYU ELOK RACHMAWATI

NIM. 0810960036

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
MALANG**

2015

LEMBAR PERSETUJUAN

**PENEMUAN POLA TRANSAKSI PENJUALAN
DENGAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN*
STUDI KASUS UD. JATI MULYA**

SKRIPSI



**Disusun oleh :
AYU ELOK RACHMAWATI
NIM. 0810960036**

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I,

Dosen Pembimbing II,

**Dian Eka Ratnawati, S.Si., M.Kom
NIP. 19730619 200212 2 001**

**Budi Darma S., S.Kom, M.Cs
NIP. 19841015 201404 1 002**

LEMBAR PENGESAHAN

PENEMUAN POLA TRANSAKSI PENJUALAN DENGAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN* STUDI KASUS UD. JATI MULYA

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh:

AYU ELOK RACHMAWATI

NIM. 0810960036

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 14 Agustus 2015

Penguji I

Penguji II

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

Wayan Firdaus Mahmudy, S.Si.,MT.Ph.D

NIP. 19720919 199702 1 001

Penguji III

Randy Cahya W, S.ST.,M.Kom

NIK. 201405 880206 1 1 001

Mengetahui

Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 5 Agustus 2015

Mahasiswa,

Ayu Elok Rachmawati

NIM. 0810960036

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, Skripsi yang berjudul **“PENEMUAN POLA TRANSAKSI PENJUALAN DENGAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN* STUDI KASUS UD. JATI MULYA”** ini dapat disusun dengan baik. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dalam penyelesaian skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Dian Eka Ratnawati, S.Si., M.Kom., selaku dosen pembimbing utama yang telah meluangkan waktu untuk memberikan pengarahan dan masukan bagi penulis.
2. Budi Darma S., S.Kom, M.Cs., selaku pembimbing kedua yang telah meluangkan waktu dalam memberikan bimbingan penulisan skripsi ini.
3. Drs. Marji, MT., selaku Ketua Program Studi Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Bondan Sapta Prakoso, S.T, selaku Dosen Penasehat Akademik.
5. Ir. Sutrisno, MT, selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
6. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
7. Segenap staf dan karyawan di Program Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
8. Ibu saya Endang Suhartini, ayah saya Slamet Abdul Rachman, dan kakak saya Rachman Subekti, Roni Cahyadi dan Yeni Rachmawati, serta keluarga besar

saya tercinta, terima kasih atas semua doa, kasih sayang dan perhatian yang tulus serta dukungan yang telah diberikan.

9. Sahabat-sahabat saya tercinta yang bersama-sama masih mengerjakan skripsi di semester akhir ini Amirul Mu'minin, Tri Wahono Sadewa, Adityo Nugroho, dan rekan-rekan yang masih bersama-sama mengerjakan skripsi semester akhir di Lab. KC terima kasih atas kenangan, perjuangan dan kekeluargaan selama ini.
10. Sahabat saya Fahrur Rozi yang telah membantu saya dalam membuat dan menyelesaikan program GSP ini.
11. Sahabat-sahabat saya yang telah terlebih dahulu menyelesaikan skripsi, terima kasih atas kenangan dan support yang tak pernah berhenti serta rekan-rekan Program Studi Ilmu Komputer angkatan 2008 yang telah memberikan dukungannya kepada penulis.
12. Nanin Nurhayati, Ach. Faisal Kurniawan, Dwi Handoko, dan keluarga besar Niagara Production, terima kasih atas support, pelajaran hidup dan gemblengan yang telah diberikan.
13. Semua pihak yang telah terlibat baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa skripsi ini tentunya tidak terlepas dari berbagai kekurangan dan kesalahan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat penulis harapkan dari berbagai pihak demi penyempurnaan penulisan skripsi ini.

Akhirnya penulis berharap agar skripsi ini dapat memberikan sumbangan dan manfaat bagi semua pihak yang berkepentingan.

Malang, 5 Agustus 2015

AYU ELOK RACHMAWATI
NIM. 0810960036

**PENEMUAN POLA TRANSAKSI PENJUALAN
DENGAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN*
STUDI KASUS UD. JATI MULYA**

ABSTRAK

Perkembangan teknologi yang semakin pesat menyebabkan berkembang pula aktifitas bisnis yang terkait dengan penjualan dan penyedia layanan kepada konsumen. Strategi bisnis yang baik merupakan suatu kunci bagi perusahaan untuk dapat meningkatkan penjualan maupun memikat konsumen. Salah satunya dengan pemanfaatan teknologi informasi yaitu dengan menganalisa data transaksi penjualan sehingga diperoleh informasi tersembunyi atau tidak diketahui sebelumnya. Penerapan penggalian data untuk mendapatkan *knowledge* tentang asosiasi antar jenis *item*, dimana jenis *item* yang muncul bersamaan pada setiap transaksi, dapat menjadi masukan penting dalam melakukan usaha peningkatan penjualan. Dalam penelitian ini penggalian data dilakukan terhadap data transaksi penjualan di UD. Jati Mulya dengan algoritma *Generalized Sequential Pattern* (GSP). Penggalian data dilakukan untuk mendapatkan pola sekuensial maupun non-sekuensial dengan rentang bulan data transaksi yang bervariasi, yaitu per tiga bulan, enam bulan, satu tahun dan dua tahun. Sistem akan melakukan pencarian aturan asosiasi dengan tahap awal adalah mencari *frequent itemset*. Kemudian dari *frequent itemset* tersebut akan dilakukan pencarian pola sekuensialnya dengan menggunakan algoritma GSP. Pada penelitian ini didapatkan bahwa didapat bahwa dengan nilai *minimum support* terbaik sebesar 10% dan *minimum confidence* terbaik sebesar 60% dan nilai rata-rata akurasi terbaik yang didapatkan sebesar 73.57%.

Kata kunci: Strategi bisnis, Informasi, Penggalian Data, *Sequential Pattern Rule*, *Generalized Sequential Pattern*.

FINDING SALES TRANSACTIONAL PATTERN USING GENERALIZED SEQUENTIAL PATTERN ALGORITHM CASE STUDY AT UD. JATI MULYA

ABSTRACT

Increasingly rapid technological developments also led to growing business activities associated with the sale and service providers to consumers. Good business strategy is a key for the company to be able to increase sales and attract customers. One of them is the use of information technology to analyze the sales transaction data in order to obtain the information hidden or previously unknown. The application of data mining to gain knowledge about the association between the type of *item*, wherein the types of *items* that appear simultaneously on each transaction, can be an important input in making efforts to increase sales. In this research, data mining conducted on the sales transaction data at UD. Jati Mulya using Generalized Sequential Pattern (GSP) algorithm. Data mining is done to get a sequential pattern *rule* and non-sequential with a range of transaction data in a varied, ie per three months, six months, one year and two years. The system will search the *rules* of the association with early stage is to find frequent *itemset*. Then from frequent *itemset* will search the sequential pattern *rule* using GSP algorithm. In this study it was found that the best *minimum support* of 10% and the best *minimum confidence* of 60% and the best average accuracy value is obtained by 73.57%.

Keywords: Business strategy, Information, Data mining, Sequential Pattern Rule, Generalized Sequential Pattern.

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS SKRIPSI	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR <i>SOURCE CODE</i>	xii
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 SistematikaPenulisan	3
 BAB II TINJAUAN PUSTAKA	 5
2.1 Penelitian GSP Terdahulu	5
2.2 Data Mining.....	5
2.3 Association Rule	6
2.4 Sequential Pattern Mining.....	8
2.5 Generalized Sequential Pattern.....	10
2.6 Akurasi Rule	11
 BAB III METODE PENELITIAN DAN PERANCANGAN.....	 12
3.1 Analisa Umum.....	13
3.1.1. Deskripsi Umum Sistem.....	13
3.1.2. Data Penelitian	13
3.2 Perancangan.....	14

3.2.1. Perancangan Sistem.....	14
3.2.2. Perancangan Database	18
3.2.3. Perancangan Antar Muka	20
3.2.2. Perancangan Uji Coba	23
3.3 Perhitungan Manual	23
BAB IV IMPLEMENTASI.....	31
4.1 Lingkungan Implementasi.....	31
4.1.1 Lingkungan Perangkat Keras.....	31
4.1.2 Lingkungan Perangkat Lunak.....	31
4.2 Implementasi Program	31
4.2.1 Implementasi Tahap Pembentukan Sekuen	32
4.2.2 Implementasi Tahap Large 1-itemset	33
4.2.3 Implementasi Pembentukan Kandidat.....	34
4.2.4 Implementasi Tahap Pembentukan <i>Rule</i>	41
4.2.5 Implementasi Pengujian Akurasi <i>Rule</i>	43
4.3 Implementasi Antarmuka	45
BAB V PENGUJIAN DAN ANALISIS.....	49
5.1 Implementasi Uji Coba	49
5.2 Implementasi Pengujian dan Pembahasan	49
5.2.1 Pengaruh Nilai <i>Minimum Support</i> dan Nilai <i>Minimum Confidence</i> terhadap Jumlah <i>Rule</i>	49
5.2.2 Uji Akurasi <i>Rule</i>	63
BAB VI PENUTUP.....	68
6.1 Kesimpulan.....	68
6.2 Saran	68
DAFTAR PUSTAKA.....	69
LAMPIRAN	71

DAFTAR GAMBAR

Gambar 3.1	Langkah-Langkah Penelitian	12
Gambar 3.2	<i>Flowchart</i> Pembentukan Pola	14
Gambar 3.3	<i>Flowchart Join</i>	15
Gambar 3.4	<i>Flowchart Pruning</i>	16
Gambar 3.5	<i>Flowchart Rule Generation</i>	17
Gambar 3.6	Relasi Antar Tabel	20
Gambar 3.7	Rancangan Antarmuka Data Master	20
Gambar 3.8	Rancangan Antarmuka GSP	21
Gambar 3.9	Rancangan Antarmuka Pengujian	22
Gambar 4.1	Tampilan Antarmuka Program	45
Gambar 4.2	Tampilan Data Transaksi	46
Gambar 4.3	Tampilan Tabel Sekuen Pelanggan	46
Gambar 4.4	Tampilan Tabel <i>Frequent Itemset</i>	47
Gambar 4.5	Tampilan Tabel <i>Association Rule</i>	47
Gambar 4.6	Tampilan Tabel <i>Rule</i> yang Terbentuk	48
Gambar 5.1	Grafik Jumlah <i>Rule</i> Data Uji 3 Bulan	55
Gambar 5.2	Grafik Jumlah <i>Rule</i> Data Uji 6 Bulan	56
Gambar 5.3	Grafik Jumlah <i>Rule</i> Data Uji 12 Bulan	57
Gambar 5.4	Grafik Jumlah <i>Rule</i> Data Uji 24 Bulan	58
Gambar 5.5	Grafik Jumlah <i>Rule</i> Rata-rata <i>Minimum Support</i> 10%	58
Gambar 5.6	Grafik Jumlah <i>Rule</i> Rata-rata <i>Minimum Support</i> 20%	59
Gambar 5.7	Grafik Jumlah <i>Rule</i> Rata-rata <i>Minimum Support</i> 30%	59
Gambar 5.8	Diagram Venn Hubungan <i>Rule</i> dan Jumlah <i>Rule</i> pada Tiap <i>Minimum Confidence</i>	63
Gambar 5.9	Grafik Akurasi <i>Rule</i> Data Uji 3 Bulan	65
Gambar 5.10	Grafik Akurasi <i>Rule</i> Data Uji 6 Bulan	65
Gambar 5.11	Grafik Akurasi <i>Rule</i> Data Uji 12 Bulan	66
Gambar 5.12	Grafik Rata-rata Akurasi	67

DAFTAR TABEL

Tabel 2.1	Tabel Transaksi yang Diurutkan Berdasarkan Cust_ID dan Tgl_Trx..	8
Tabel 2.2	Tabel Sekuen Pelanggan	9
Tabel 3.1	Tabel Transaksi	18
Tabel 3.2	Tabel Detil Transaksi.....	18
Tabel 3.3	Tabel Produk	19
Tabel 3.4	Tabel Sub-produk	19
Tabel 3.5	Tabel Jenis Kayu	19
Tabel 3.6	Tabel Konsumen.....	19
Tabel 3.7	Tabel Pengujian 1 : Pengaruh <i>Minimum Support</i> dan <i>Minimum Confidence</i> terhadap Jumlah <i>Rule</i>	23
Tabel 3.8	Tabel Pengujian 2 : Akurasi <i>Rule</i>	23
Tabel 3.9	Tabel Sampel Data Produk	24
Tabel 3.10	Tabel Sampel Data Transaksi.....	24
Tabel 3.11	Tabel Sekuen Pelanggan	25
Tabel 3.12	Tabel Kandidat 1- <i>itemset</i>	26
Tabel 3.13	Tabel <i>Large</i> 1- <i>itemset</i>	25
Tabel 3.14	Kandidat 2- <i>itemset</i>	27
Tabel 3.15	Tabel <i>Large</i> 2- <i>itemset</i>	29
Tabel 3.16	Tabel Kandidat 3- <i>itemset</i>	29
Tabel 3.17	Tabel Pola Frekuen Tinggi.....	29
Tabel 3.18	Tabel Aturan Asosiatif.....	30
Tabel 5.1	Tabel Hasil Uji Pengaruh Nilai <i>Minimum support</i> dan Nilai <i>Minimum confidence</i> terhadap Jumlah <i>Rule</i> yang Dihasilkan	50
Tabel 5.2	Tabel Jumlah <i>Rule</i> Rata-rata Data Uji 3 Bulan	55
Tabel 5.3	Tabel Jumlah <i>Rule</i> Rata-rata Data Uji 6 Bulan	56
Tabel 5.4	Tabel Jumlah <i>Rule</i> Rata-rata Data Uji 12 Bulan	57
Tabel 5.5	Tabel Hubungan <i>Rule</i> dan Jumlah <i>Rule</i> pada Tiap <i>Minimum Confidence</i>	60
Tabel 5.6	Tabel Akurasi	64
Tabel 5.7	Tabel Rata-rata Akurasi	66

DAFTAR SOURCE CODE

<i>Pseudocode</i> 2.1	Mencari Maksimum Sekuen.....	9
<i>Pseudocode</i> 2.2	Algoritma GSP.....	10
<i>Pseudocode</i> 2.3	Pembentukan Kandidat.....	11
<i>Source Code</i> 4.1	Proses Pembentukan Sekuen Pelanggan	32
<i>Source Code</i> 4.2	Proses Pembentukan Kandidat 1-itemset	33
<i>Source Code</i> 4.3	Proses Pembentukan Large 1-itemset.....	34
<i>Source Code</i> 4.4	Proses Pembentukan Large 2-itemset.....	34
<i>Source Code</i> 4.5	Proses Pembentukan Large 3-itemset.....	37
<i>Source Code</i> 4.6	Proses Pembentukan Rule.....	41
<i>Source Code</i> 4.7	Proses Pengujian Akurasi	43



BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang semakin pesat menyebabkan berkembang pula aktivitas bisnis yang terkait dengan penjualan dan penyedia layanan kepada konsumen. Tidak heran jika hal ini menimbulkan persaingan bisnis yang semakin ketat, sehingga mewajibkan perusahaan harus pandai dalam memahami perilaku konsumen.

Strategi bisnis yang baik merupakan suatu kunci bagi perusahaan untuk dapat meningkatkan penjualan maupun memikat konsumen, salah satunya dengan pemanfaatan teknologi informasi. Informasi merupakan hal yang memegang peranan penting dalam segala aspek kehidupan, tidak terkecuali bagi perusahaan. Informasi dapat diperoleh dari berbagai macam data yang ada di perusahaan, contohnya data transaksi penjualan. Dari data tersebut, perusahaan dapat melakukan pengelolaan bisnis, pengendalian produksi, dan analisa pasar.

Data transaksi penjualan pada perusahaan yang tidak dimanfaatkan makin lama akan semakin menumpuk dan menjadi sia-sia, seperti halnya pada UD. Jati Mulya – *Wood Processing Industry*. UD. Jati Mulya merupakan sebuah perusahaan berskala menengah yang secara total bergerak di bidang perkayuan, mulai dari produksi kusen, pintu panil, pintu lapis, jendela, rumah kayu, gasebo, *railing* tangga kayu dan lain-lain, hingga penyedia jasa layanan *finishing* maupun pemasangan. Saat ini pemanfaatan data-data yang dimiliki oleh UD. Jati Mulya belum maksimal, hanya sebatas untuk pembuatan laporan keuangan tiap bulan.

Hal ini menarik perhatian penulis untuk memanfaatkan data transaksi penjualan pada perusahaan tersebut dengan menerapkan *data mining* untuk mendapatkan *knowledge* tentang asosiasi antar jenis *item*, dimana jenis *item* yang muncul bersamaan pada setiap transaksi, dapat menjadi masukan penting dalam melakukan usaha peningkatan penjualan. Misalkan dalam melakukan promosi untuk pembelian satu jenis *item* dapat membeli jenis *item* lain dengan harga lebih murah, pengetahuan asosiasi antar jenis *item* ini dapat memberikan informasi mengenai kombinasi *item* yang harus tercakup dalam promosi tersebut.

Metode data mining yang digunakan dalam penelitian ini adalah metode *sequential pattern mining*. Berbeda dengan metode *association rule* yang mencari *intra-transaction* yaitu mencari informasi mengenai *item* apa saja yang muncul dalam satu transaksi, *sequential pattern rule* mencari *inter-transaction* yaitu mencari informasi mengenai kemunculan *item* diikuti *item* yang lain terurut berdasarkan waktu transaksi maupun sistem *taxonomies* [SRI-96].

Penelitian berbasis *sequential pattern rule* telah banyak dilakukan dengan berbagai metode, salah satunya yaitu algoritma *generalized sequential pattern* (GSP). GSP diperkenalkan oleh Agrawal dan Srikant (1996) yang merupakan perkembangan dari algoritma apriori [AGR-95] yang mampu menangani kasus *time constraint*, *sliding windows* ataupun permasalahan *taxonomies*. Selain itu, GSP mampu memproses dan menemukan semua pola sekuensial maupun non-sekuensial yang ada.

Pada penelitian lain, algoritma *generalized sequential pattern* pernah digunakan untuk menggali data sekuensial sirkulasi buku pada perpustakaan UK Petra [BUD-09], berdasarkan hasil survey terhadap para pengambil keputusan di perpustakaan UK Petra didapatkan nilai rata-rata 88,34% yang menyatakan bahwa aplikasi menggunakan algoritma ini layak untuk digunakan.

Berdasarkan latar belakang yang telah dipaparkan di atas, maka pada penelitian kali ini penulis akan merancang dan membuat sebuah sistem aplikasi dengan judul **“Penemuan Pola Transaksi Penjualan dengan Algoritma Generalized Sequential Pattern Studi Kasus UD. Jati Mulya”**.

1.2 Rumusan Masalah

Permasalahan yang dijadikan obyek penelitian pada skripsi ini adalah:

1. Bagaimana menerapkan algoritma *generalized sequential pattern* untuk menemukan aturan sekuensial pada transaksi penjualan.
2. Bagaimana pengaruh *minimum support* dan *minimum confidence* terhadap jumlah *rule*, rata-rata *confidence*, dan rata-rata akurasi yang dihasilkan
3. Berapa tingkat akurasi yang dihasilkan oleh algoritma *generalized sequential pattern*?

1.3 Batasan Masalah

Agar tidak terdapat penyimpangan tujuan penelitian dan meluasnya pembahasan, maka diberikan batasan masalah dalam penelitian ini sebagai berikut:

1. Data *training* yang digunakan dalam penelitian ini yaitu data transaksi penjualan pada UD. Jati Mulya.
2. Program yang dibuat menghasilkan keluaran berupa *rule-rule* yang berhasil ditemukan setelah dilakukan proses *sequential pattern mining* dengan algoritma *generalized sequential pattern*.

1.4 Tujuan

Tujuan dari penulisan skripsi ini adalah :

1. Menerapkan algoritma *generalized sequential pattern* untuk menemukan aturan sekuensial pada penjualan
2. Mengetahui pengaruh *minimum support* dan *minimum confidence* terhadap jumlah *rule*, rata-rata *confidence*, dan rata-rata akurasi yang dihasilkan,
3. Mengetahui tingkat akurasi yang dihasilkan oleh algoritma *generalized sequential pattern*.

1.5 Manfaat

Manfaat yang diperoleh dari penulisan skripsi ini adalah untuk mengetahui pola asosiasi sekuensial *item-item* barang, dimana informasi yang terbentuk dapat menjadi masukan penting bagi perusahaan dalam melakukan strategi bisnis guna memperoleh keuntungan yang lebih besar.

1.6 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang penulisan skripsi, rumusan masalah, batasan masalah, tujuan penulisan, manfaat penulisan, metodologi yang digunakan serta sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang teori-teori, algoritma, dan konsep yang dipakai penulis dalam pembuatan skripsi.

3. BAB III METODOLOGI DAN PERANCANGAN

Bab ini berisi tentang langkah-langkah yang dilakukan dalam merancang aplikasi data mining.

4. BAB IV IMPLEMENTASI

Bab ini berisi penjelasan implementasi dan rancangan yang telah diuraikan pada Bab III.

5. BAB V PENGUJIAN DAN ANALISIS

Bab ini berisi tentang prosedur pengujian, hasil pengujian dan analisis data hasil pengujian.

6. BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran yang diperoleh dari pembahasan materi dan pengujian aplikasi yang diharapkan bermanfaat untuk pengembangan penelitian lebih lanjut.



BAB II

TINJAUAN PUSTAKA

2.1 Penelitian GSP Terdahulu

Penelitian berbasis *sequential pattern rule* khususnya menggunakan algoritma *generalized sequential pattern* (GSP) pernah dilakukan oleh Agrawal dan Srikant (1996) sebagai jawaban dari penelitian sebelumnya yang menggunakan algoritma apriori [AGR-95]. Dalam penelitian [SRI-96], algoritma GSP telah mampu menangani kasus *time constraint*, *sliding windows* ataupun permasalahan *taxonomies*. Selain itu, GSP mampu memproses dan menemukan semua pola dengan waktu eksekusi yang lebih cepat.

Pada penelitian lain, algoritma *generalized sequential pattern* pernah digunakan untuk menggali data sekuensial sirkulasi buku pada perpustakaan UK Petra [BUD-09], berdasarkan hasil survey terhadap para pengambil keputusan di perpustakaan UK Petra didapatkan nilai rata-rata 88.34% yang menyatakan bahwa aplikasi menggunakan algoritma ini layak untuk digunakan.

2.2 Data Mining

Munculnya *data mining* pertama kali diawali oleh kebutuhan terhadap informasi dari data yang melimpah yang dapat diguakan sebagai pendukung keputusan untuk solusi bisnis dan dukungan infrastruktur dalam bidang teknologi informasi [MOE-02].

Menurut Rennols (2004), *data mining* adalah salah satu teknik dalam ilmu komputer yang digunakan untuk menggali dan mengambil suatu informasi pada banyak data. Sedangkan menurut Han dan Kamber (2001), *data mining* adalah mencocokkan data dalam suatu model untuk menemukan informasi yang tersembunyi.

Data mining atau dalam bahasa Indonesia disebut dengan penggalian data, merupakan suatu proses dalam mencari hubungan, pola, atau tren dari sebuah data berukuran besar dengan menggunakan teknik statistika ataupun matematika [LAR-05]. *Data mining* juga didefinisikan sebagai serangkaian proses penggalian nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual [PRA-07].

Berdasarkan definisi *data mining* yang luas, maka terdapat banyak teknik analisa yang dapat digolongkan dalam *data mining*. Berikut adalah beberapa teknik populer yang sering disebut-sebut dalam literatur *data mining*, antara lain: *clustering* (penggugusan), *classification* (penggolongan), dan *association rule mining* (pencarian pola) [PRA-07].

2.3 Association Rule

Association rule mining atau analisis asosiasi, merupakan suatu teknik untuk mencari hubungan antar-item dalam suatu kumpulan data yang telah ditentukan [HAN-01]. Dalam *association rule*, suatu kelompok *item* dinamakan *itemset* [RUL-08]. Contoh dari aturan asosiatif dari pembelian di suatu pasar swalayan adalah mengetahui besarnya kemungkinan seorang pelanggan untuk membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut, pemilik pasar swalayan bisa mengatur penempatan barangnya atau merancang kampanye pemasaran menggunakan kupon diskon untuk kombinasi barang tertentu. Karena teknik analisis asosiasi terkenal dengan aplikasinya untuk menganalisa isi keranjang belanja konsumen di pasar swalayan, maka teknik ini sering disebut dengan istilah *market basket analysis* [BER-04].

Dalam pencarian *association rule*, terdapat dua tahap yaitu analisis pola frekuensi tinggi (*frequent pattern mining*) dan pembentukan aturan asosiasi [KUS-01]. Dalam tahapan-tahapan tersebut, dibutuhkan suatu variabel ukuran yang dapat ditentukan oleh pengguna untuk mengatur batasan sejauh mana dan sebanyak apa hasil keluaran yang diinginkan. *Support* dan *confidence* adalah dua ukuran kepercayaan yang menunjukkan kepastian dan tingkat kegunaan suatu *rule* yang ditemukan. Pada umumnya *association rule* yang ditemukan di katakan menarik apabila aturan tersebut memenuhi baik *minimum support* maupun *minimum confidence* yang telah di tentukan oleh pengguna [HAN-04].

Support (nilai penunjang) yaitu prosentase kombinasi *item* dalam basis data atau ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan transaksi [HAN-01]. Nilai *support* digunakan untuk menentukan *frequent itemset*. *Itemset* yang nilai *support*-nya memenuhi parameter *minimum support* (min_sup) akan masuk dalam *frequent itemset*. Min_sup digunakan sebagai batasan frekuensi kejadian atau jumlah *support* yang harus dipenuhi suatu

kelompok data yang dijadikan aturan [YUL-02]. Untuk mencari nilai *support* sebuah *item* ditunjukkan dalam Persamaan 2.1, sedangkan untuk mencari nilai *support* dari dua buah *item* ditunjukkan dalam Persamaan 2.2.

$$Support(A) = \frac{\sum \text{transaksi mengandung } A}{\sum \text{transaksi}} \dots\dots\dots (2.1)$$

$$\begin{aligned} Support(A, B) &= P(A \cap B) \\ &= \frac{\sum \text{transaksi mengandung } A \text{ dan } B}{\sum \text{transaksi}} \dots\dots\dots (2.2) \end{aligned}$$

Confidence (nilai kepercayaan) yaitu suatu ukuran yang menunjukkan hubungan antar dua atau lebih *item* dalam aturan assosiatif [HAN-01]. Nilai *confidence* digunakan untuk menentukan *strong association rule*, yaitu *association rule* yang nilai *confidence*-nya memenuhi parameter *minimum confidence* (min_conf) [YUL-02]. Untuk mencari nilai *confidence* dari aturan $A \rightarrow B$ ditunjukkan dalam Persamaan 2.3.

$$\begin{aligned} Confidence &= P(A|B) \\ Confidence &= \frac{\sum \text{transaksi mengandung } A \text{ dan } B}{\sum \text{transaksi mengandung } A} \dots\dots\dots (2.3) \end{aligned}$$

Selain menggunakan *confidence* untuk mengukur tingkat kebenaran, dikenal juga *lift ratio* untuk melihat kuat tidaknya aturan asosiasi dengan membandingkan dengan nilai *benchmark confidence*. *Lift ratio* merupakan sebuah angka yang menunjukkan berapa banyak kemungkinan menemukan sebuah atribut muncul bersama dengan atribut lainnya dibandingkan dengan seluruh kejadian adanya atribut yang terpenuhi. *Lift* didefinisikan dalam Persamaan 2.4 [AMI-11].

$$Lift = \frac{Confidence}{Expected\ Confidence} \dots\dots\dots (2.4)$$

Dimana,

$$Expected\ confidence = \frac{\sum \text{transaksi mengandung item consequent}}{\sum \text{transaksi}} \dots\dots\dots (2.5)$$

Atau dengan cara:

$$Lift = \frac{Pr(A|C)}{Pr(C)} \dots\dots\dots (2.6)$$

Ketika nilai *lift* sama dengan 1 maka A dan B adalah independent karena $Pr(C|A)=Pr(C)$. Ketika probabilitas C terjadi dipengaruhi oleh terjadinya A, maka *lift* lebih besar dari 1. Ketetapan *lift ratio* adalah apabila hasil perhitungan berada di bawah 1, maka *item-item* tersebut tidak menunjukkan adanya saling keterkaitan antara *antecedent* dan *consequent* [AMI-11].

2.4 Sequential Pattern Mining

Sequential Pattern Mining diperkenalkan pertama kali oleh Agrawal dan Srikant (1995) sebagai teknik yang digunakan untuk mencari *frequent subsequence* sebagai pola dalam suatu data sekuensial [PEI-04].

Pada prinsipnya, *sequential pattern rule* dan *association rule* sama-sama digunakan untuk mencari hubungan antar *item* dalam basis data, akan tetapi memiliki perbedaan yang cukup signifikan. Berbeda dengan metode *association rule* yang mencari *intra-transaction* yaitu mencari informasi mengenai *item* apa saja yang muncul dalam satu transaksi, *sequential pattern rule* mencari *inter-transaction* yaitu mencari informasi mengenai kemunculan *item* diikuti *item* yang lain terurut berdasarkan waktu transaksi maupun sistem *taxonomies* [SRI-96].

Selain itu, data yang digunakan pada *sequential pattern mining* berupa data yang bersifat sekuensial. Data sekuensial merupakan data yang berupa urutan dari elemen-elemen (transaksi), dimana tiap elemen tersebut merupakan kumpulan dari kejadian-kejadian (*item*). Contohnya yaitu pada data transaksi berikut [AGR-95]:

Tabel 2.1 Tabel Transaksi yang Diurutkan Berdasarkan Cust_ID dan Tgl_Trx

Cust_ID	Tgl_Trx	Item
1	25 Juni 1993	30
1	30 Juni 1993	90
2	10 Juni 1993	10, 20
2	15 Juni 1993	30
2	20 Juni 1993	40, 50, 70
3	25 Juni 1993	30, 50, 70
4	25 Juni 1993	30
4	30 Juni 1993	40, 70
4	25 Juli 1993	90
5	12 Juni 1993	90

Selanjutnya data dari Tabel 2.1 dibentuk menjadi tabel sekuensial, dan ditunjukkan pada Tabel 2.2:

Tabel 2.2 Tabel Sekuen Pelanggan

Cust_ID	Sekuen
1	(30) (90)
2	(10, 20) (30) (40, 50, 70)
3	(30, 50, 70)
4	(30), (40, 70) (90)
5	(90)

Permasalahan dalam *sequential pattern mining* dapat dijabarkan dalam beberapa tahap, yaitu [AGR-95]:

1. *Sort Phase*

Pada tahap ini *database* (D) diurutkan berdasarkan CustomerID sebagai kunci mayor dan TransactionTime sebagai kunci minor. Secara tidak langsung, kita mengubah data transaksi yang asli menjadi data *customer-sequences*.

2. *Large-itemset Phase*

Pada tahap ini dicari semua *large itemset* (*itemset*) L , selain itu juga dicari himpunan *large 1-sequences* karena $\{(l) | l \in L\}$.

3. *Transformation Phase*

Pada tahap ini, data asli dirubah variabelnya. Maksudnya kita mengubah tiap *customer-sequence* menjadi representasi alternatif lainnya. Kita dapat menyebut *customer-sequence* yang baru menjadi $\{l_1, l_2, \dots, l_n\}$ dimana l_i adalah *itemset*.

4. *Sequence Phase*

Pada tahap inilah diterapkannya algoritma, yaitu dengan memanfaatkan *itemset* untuk menemukan sekuen yang diinginkan.

5. *Maximal Phase*

Pada tahap ini dicari maksimum sekuen dari himpunan *large sequences*. Setelah menemukan himpunan semua *large sequence* S dalam sequence phase. Algoritma untuk mencari maksimum sekuen ditunjukkan pada *Pseudocode* 2.1.

Input	: All of Large sequence S
Output	: Maximal sequences di $U_k L_k$
1	$n =$ the length of longest sequence
2	for $(k = n; k > 1; k --)$ do
3	Foreach k-sequence s_k do
	Delete from S all subsequences of s_k

Pseudocode 2.1 Mencari Maksimum Sekuen [AGR-95]

2.5 Generalized Sequential Pattern

Salah satu algoritma yang menerapkan metode sequential pattern mining yaitu algoritma *Generalized Sequential Pattern* (GSP). GSP diperkenalkan pertama kali oleh Agrawal dan Srikant (1996) yang merupakan perkembangan dari algoritma apriori [AGR-95]. GSP mampu memproses dan menemukan semua pola sekuensial maupun non-sekuensial yang ada.

Algoritma GSP melakukan *multiple passes* pada data yang ada. Fase pertama yaitu menentukan *support* dari masing-masing *item*. Pada akhir fase pertama, algoritma ini akan mendapatkan *item* mana yang akan menjadi *frequent*, yaitu *item-item* yang memenuhi nilai *min_sup*. Masing-masing *item* menghasilkan sebuah *frequent sequence* yang pertama yang terdiri atas *item* tersebut (*frequent 1-sequence*). Fase selanjutnya, masing-masing *sub-sequence* pada setiap fase pada awalnya dimulai dengan sekumpulan calon *candidate* (suatu *frequent sequence* yang ditemukan pada fase sebelumnya). Sekumpulan calon *candidate* tersebut digunakan untuk menghasilkan *frequent sequence* baru yang berpotensi (*candidate sequence*). Masing-masing *candidate sequence* memiliki lebih dari satu *item* calon *sequence*. Pada akhir fase tersebut, algoritma akan menghasilkan mana saja *candidate sequences* yang termasuk dalam *frequent*, dimana *frequent candidate* tersebut menjadi calon *candidate* untuk fase selanjutnya. Algoritma ini akan berakhir ketika tidak ditemukan lagi *frequent sequences* atau tidak ada lagi *candidate sequences* yang dihasilkan [SRI-96].

Prosedur algoritma GSP berdasarkan dapat dilihat pada *Pseudocode 2.2*.

Input	: L_1 {large 1-sequence}
Output	: Maximal sequences di $U_k L_k$
1	for $\{k = 2; L_{k-1} \neq \emptyset; k++\}$ do
2	begin
3	$C_k =$ new candidate generated from L_{k-1}
4	Foreach customer-sequences c in the database

```

do
5   Increment the count of all candidate in  $C_k$ 
    that are contained in c
6    $L_{k-1}$  = Candidate in  $C_k$  with minimum support
7   end

```

Pseudocode 2.2 Algoritma GSP [AGR-95]

Terdapat dua kunci penting dalam algoritma GSP, yaitu:

1. *Candidate Generation*, yaitu tahap pembentukan kandidat sebelum fase berlangsung.
2. *Counting Candidate*, yaitu tahap perhitungan *support* dari *candidate sequence* yang terbentuk. Prosedur pembentukan kandidat dapat dilihat pada Pseudocode 2.3.

Input	: L_{k-1} {the set of all large (k-1)-sequence}
Output	: New candidate C_k
1	Join L_{k-1} with L_{k-1}
2	Insert into C_k
3	Select $p.litemset_1, \dots, p.litemset_{k-1},$ $q.litemset_{k-1}$
4	From $L_{k-1} p, L_{k-1} q$
5	Where $p.litemset_1 = q.litemset_1, \dots,$ $p.litemset_{k-2} = q.litemset_{k-2}$
6	Delete all sequence $c \in C_k$ such that some (k-1)- sequence of c is not in L_{k-1}

Pseudocode 2.3 Pembentukan Kandidat [SRI-96]

2.6 Akurasi Rule

Kualitas dari aturan asosiasi dapat dievaluasi menggunakan rumus akurasi. Akurasi dapat dihitung berdasarkan prosentase *error* yang terjadi. Akurasi didefinisikan dalam Persamaan 2.7.

$$\text{Prosentase Akurasi} = 100\% - \text{error} \dots \dots \dots (2.7)$$

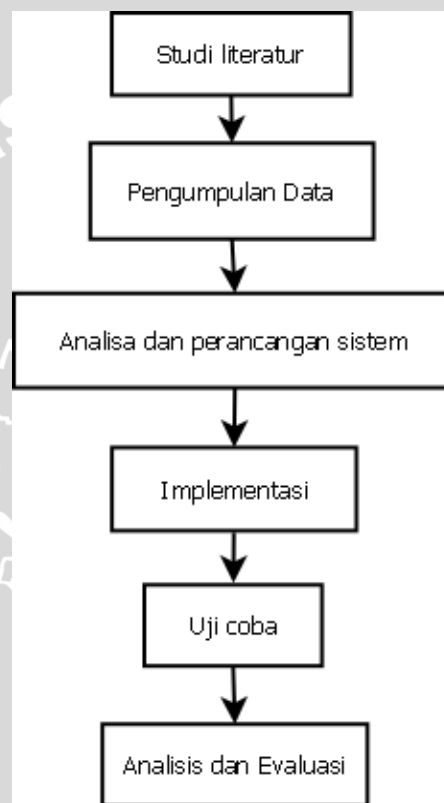
Dimana,

$$\text{Error} = \frac{\text{Prediksi salah}}{\text{Total prediksi}} \times 100\% \dots \dots \dots (2.8)$$

BAB III

METODOLOGI DAN PERANCANGAN

Pada bab ini dibahas mengenai metode yang digunakan dalam pembuatan perangkat lunak untuk menemukan pola transaksi penjualan dengan metode *sequential pattern mining* menggunakan algoritma *generalized sequential pattern*. Langkah-langkah yang dilakukan dalam penelitian ini ditunjukkan dalam Gambar 3.1.



Gambar 3.1 Langkah-langkah Penelitian

Berdasarkan Gambar 3.1, langkah-langkah penelitian tersebut dapat dijabarkan sebagai berikut:

1. Melakukan studi literatur mengenai *sequential pattern mining* dan algoritma *generalized sequential pattern*.
2. Melakukan pengumpulan data-data yang diperlukan dalam penelitian ini, yaitu data transaksi penjualan.
3. Menganalisa dan merancang sistem.

4. Mengimplementasikan rancangan menjadi sebuah perangkat lunak sesuai dengan analisa dan perancangan tahap sebelumnya..
5. Melakukan pengujian terhadap perangkat lunak menggunakan data yang telah dikumpulkan.
6. Menganalisis dan mengevaluasi hasil yang diperoleh dari uji coba pada tahap sebelumnya.

3.1 Analisis Umum

3.1.1 Deskripsi Umum Sistem

Sistem yang dibuat merupakan sistem yang dikembangkan untuk melakukan analisa terhadap data transaksi penjualan yang bertujuan untuk mengetahui pola penjualan dan menemukan aturan-aturan penting yang tersembunyi. Dalam skripsi ini terdapat tiga parameter yang digunakan dalam proses analisa, yaitu nilai *minimum support*, *minimum confidence* dan *lift ratio*. Hasil keluaran dari sistem berupa pola *sequential rule* yang menunjukkan keterkaitan antar produk satu dengan yang lain. Dari aturan-aturan yang ditemukan, maka dapat diketahui pola penjualan perusahaan yang kemudian dapat dimanfaatkan perusahaan sebagai masukan dalam melakukan strategi bisnis guna memperoleh keuntungan yang lebih besar.

Metode yang digunakan dalam skripsi ini adalah metode *sequential pattern rule* untuk melakukan *frequent pattern mining*, sedangkan untuk mencari *frequent itemset* digunakan algoritma GSP. Dengan menggunakan algoritma GSP, maka ditemukan *frequent set*, yaitu data yang telah memenuhi *minimum support*. Dari *frequent itemset* ini selanjutnya dihitung nilai *confidence*-nya. Jika memenuhi syarat *minimum confidence*, maka dihasilkan *association rule*-nya. Dari *rule* yang telah didapatkan, selanjutnya di uji kekuatan *rule* tersebut menggunakan *lift ratio*.

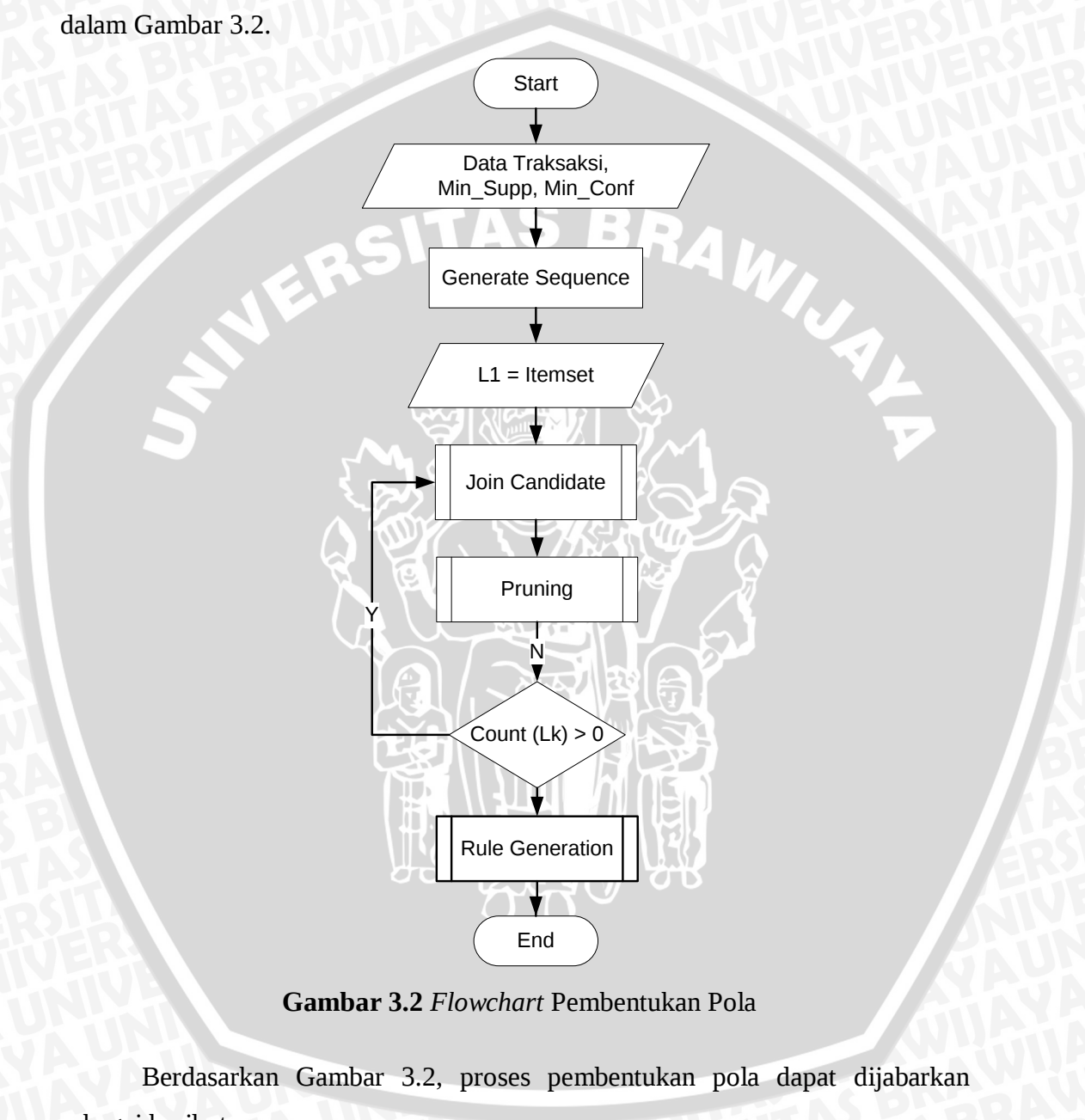
3.1.2 Data Penelitian

Pengumpulan data dilakukan dengan metode pengumpulan data primer. Data yang dimaksud dalam skripsi ini merupakan data transaksi penjualan pada UD. Jati Mulya bulan Januari 2011 hingga Desember 2012.

3.2 Perancangan

3.2.1 Perancangan Sistem

Dalam melakukan analisa pencarian pola terhadap data transaksi penjualan perlu dilakukan beberapa tahapan proses. Tahapan-tahapan tersebut ditunjukkan dalam Gambar 3.2.

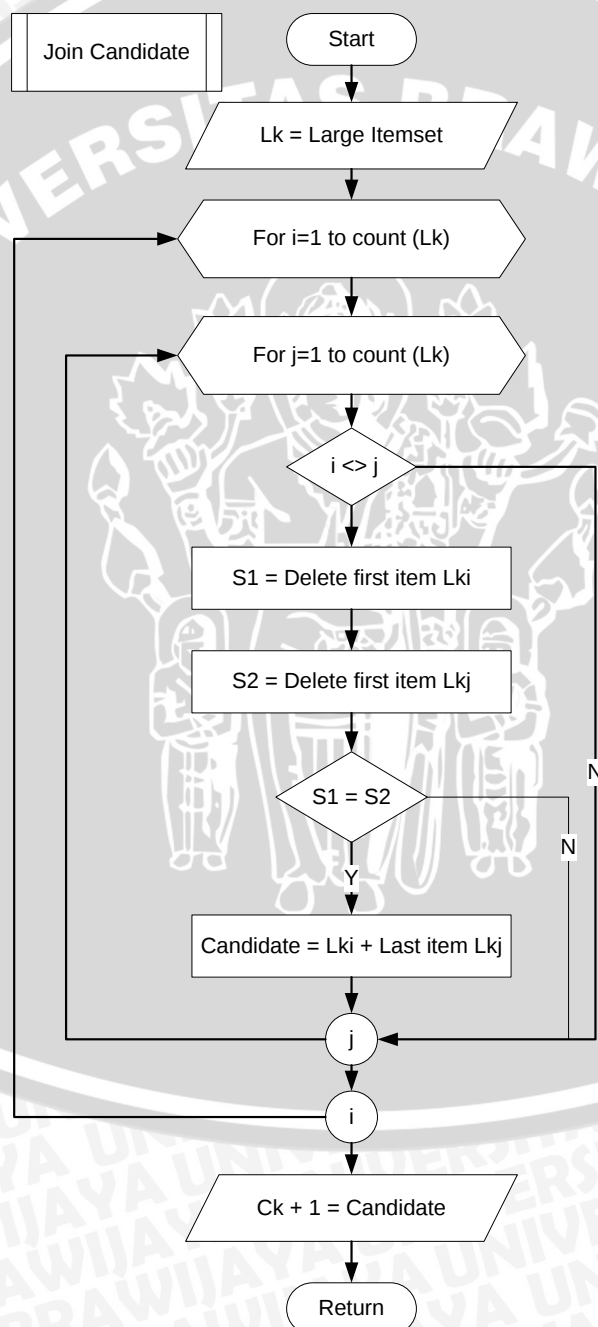


Gambar 3.2 Flowchart Pembentukan Pola

Berdasarkan Gambar 3.2, proses pembentukan pola dapat dijabarkan sebagai berikut:

1. Sistem menerima masukan berupa data transaksi penjualan, nilai *minimum support* (*min_sup*), dan nilai *minimum confidence* (*min_conf*), kemudian sistem akan melakukan proses *generate sequence*.

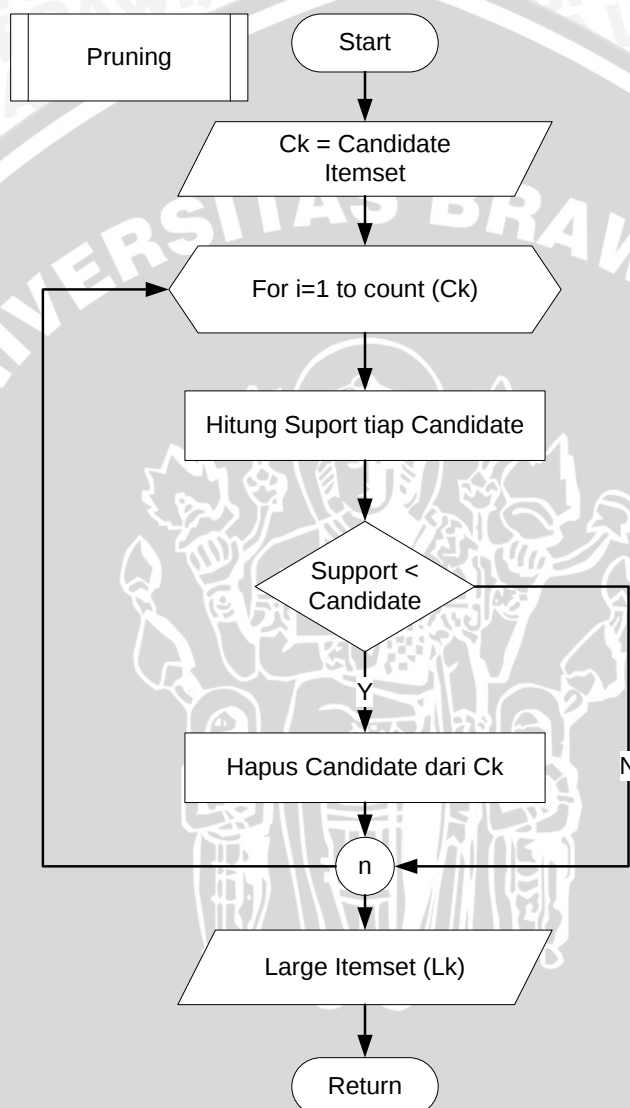
2. Proses *generate sequence* merupakan tahapan pembentukan tabel sekuen pelanggan yaitu membentuk *data training* menjadi sebuah urutan yang kemudian menghasilkan L_1 (*Large 1-itemset*).
3. Proses selanjutnya, L_1 digunakan dalam pembentukan kandidat. Proses ini terdiri atas beberapa tahapan, yaitu proses *join* dan *pruning*. Proses *join* dan *pruning* akan terus berlangsung sampai tidak ditemukan L_k .
4. Proses terakhir yaitu proses pembentukan aturan dari seluruh kandidat yang telah diperoleh.



Gambar 3.3 Flowchart Join

Proses *join* digunakan untuk membentuk kandidat berdasarkan *itemset* yang frekuen. Berdasarkan Gambar 3.3, tahapan dari proses *join* meliputi:

1. Pembentukan *subsequence* dengan menghapus *item* pertama pada S_1 .
2. Pembentukan *subsequence* dengan menghapus *item* terakhir pada S_2 .
3. Jika *subsequence* S_1 sama dengan *subsequence* S_2 , maka dilakukan *join*.
4. Proses ini berlangsung sampai sejumlah L_k .

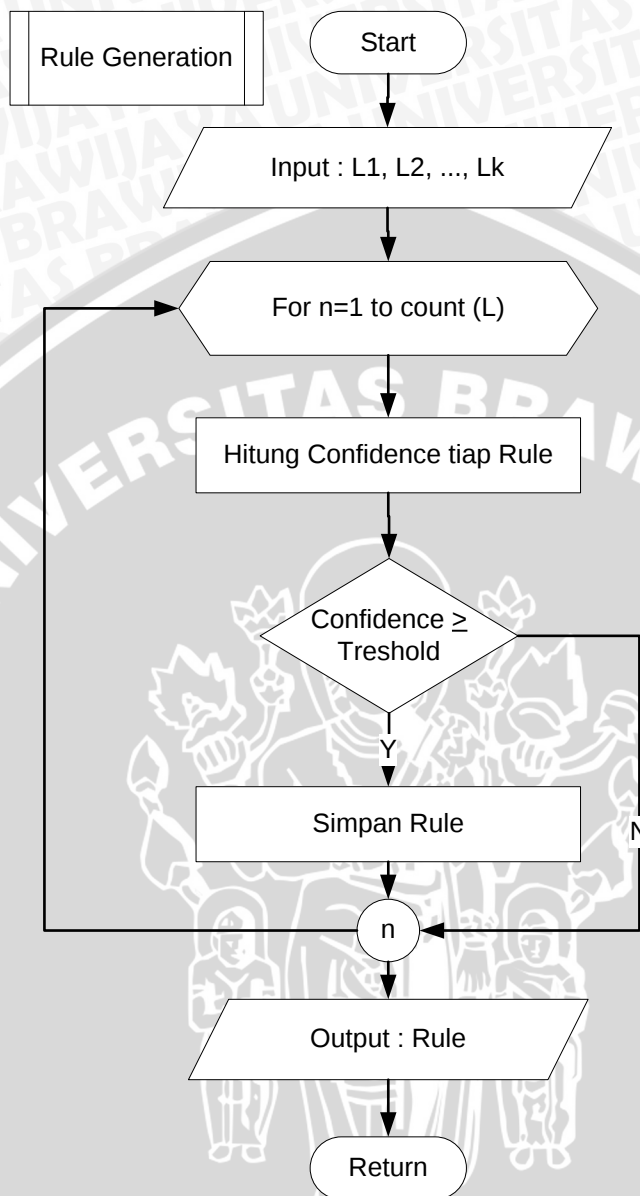


Gambar 3.4 Flowchart Pruning

Proses *pruning* digunakan untuk pemangkasan kandidat yang tidak memenuhi syarat *min_sup* dan *min_conf*. Tahapan proses ini ditunjukkan dalam Gambar 3.4, yang meliputi:

1. Menghitung nilai *support* dari tiap-tiap kandidat.
2. Menentukan kandidat yang memiliki nilai *support* lebih dari *min_sup*.

3. Menghapus kandidat yang nilai *support*-nya kurang dari *min_sup*.
4. Proses ini berulang sebanyak C_k (kandidat *itemset*).



Gambar 3.5 Flowchart Rule Generation

Tahapan dari proses *rule generation* (pembangkitan aturan) ditunjukkan dalam Gambar 3.5, yaitu dengan menghitung nilai *confidence* dari setiap aturan yang terbentuk. Apabila nilai *confidence* dari aturan lebih besar atau sama dengan nilai *min_conf*, maka aturan tersebut disimpan dalam Tabel *Rule*.

3.2.2 Perancangan Database

Untuk menyimpan data yang dibutuhkan dalam sistem ini, maka dibutuhkan suatu basis data. Basis data tersebut terdiri atas enam tabel, yaitu:

1. Tabel Transaksi (Tb_Trx)
2. Tabel Detil Transaksi (Tb_Detil)
3. Tabel Produk (Tb_Prod)
4. Tabel Sub-produk (Tb_Sub)
5. Tabel Jenis Kayu (Tb_JK)
6. Tabel Pelanggan (Tb_Cust)

Tabel transaksi, tabel detil transaksi, tabel produk, tabel sub-produk, tabel jenis kayu dan tabel konsumen digunakan untuk menyimpan data yang dibutuhkan dalam proses sistem. Penjelasan masing-masing tabel dijelaskan sebagai berikut:

1. Tabel Transaksi (Tb_Trx)

Tabel Tb_Trx digunakan untuk menyimpan data-data transaksi penjualan. Tabel Tb_Trx ditunjukkan dalam Tabel 3.1.

Tabel 3.1 Tabel Transaksi

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
ID_Trx	Number	Kode transaksi berupa <i>primary key</i>
Cust_ID	Varchar(32)	Kode pelanggan
Tgl_Trx	Date	Tanggal transaksi
Item	Varchar(32)	Produk-produk terjual diisi berdasarkan kode produk

2. Tabel Detil Transaksi (Tb_Detil)

Tabel Tb_Detil digunakan untuk menyimpan detil transaksi. Data yang ada pada tabel ini yaitu ID_Trx dan ID_Prod. Tabel Tb_Detil ditunjukkan dalam Tabel 3.2.

Tabel 3.2 Tabel Detil Transaksi

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
ID_Trx	Varchar(32)	Kode transaksi
ID_Prod	Varchar(32)	Kode produk

3. Tabel Produk (Tb_Prod)

Tabel Tb_Prod digunakan untuk menyimpan detil data produk, yaitu terdiri atas ID_Prod, ID_Kayu, ID_Sub dan Nama_Prod. Tb_Prod ditunjukkan dalam Tabel 3.3.

Tabel 3.3 Tabel Produk

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
<u>ID_Prod</u>	Varchar(32)	Kode produk berupa <i>primary key</i>
ID_Kayu	Varchar(32)	Kode jenis kayu
ID_Sub	Varchar(32)	Kode sub-produk
Nama_Prod	Varchar(32)	Nama produk

4. Tabel Sub-produk (Tb_Sub)

Tabel Tb_Sub digunakan untuk menyimpan sub kategori dari produk. Tabel ini berisi ID_Sub dan Nama_Sub. Tb_Sub ditunjukkan dalam Tabel 3.4.

Tabel 3.4 Tabel Sub-produk

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
<u>ID_Sub</u>	Varchar(32)	Kode sub-produk berupa <i>primary key</i>
Nama_Sub	Varchar(32)	Nama sub-produk

5. Tabel Jenis Kayu (Tb_JK)

Tabel Tb_JK digunakan untuk menyimpan jenis kayu dari produk. Tabel ini berisi ID_JK dan Nama_Kayu. Tb_JK ditunjukkan dalam Tabel 3.5.

Tabel 3.5 Tabel Jenis Kayu

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
<u>ID_Kayu</u>	Varchar(32)	Kode jenis kayu berupa <i>primary key</i>
Nama_Kayu	Varchar(32)	Nama jenis kayu

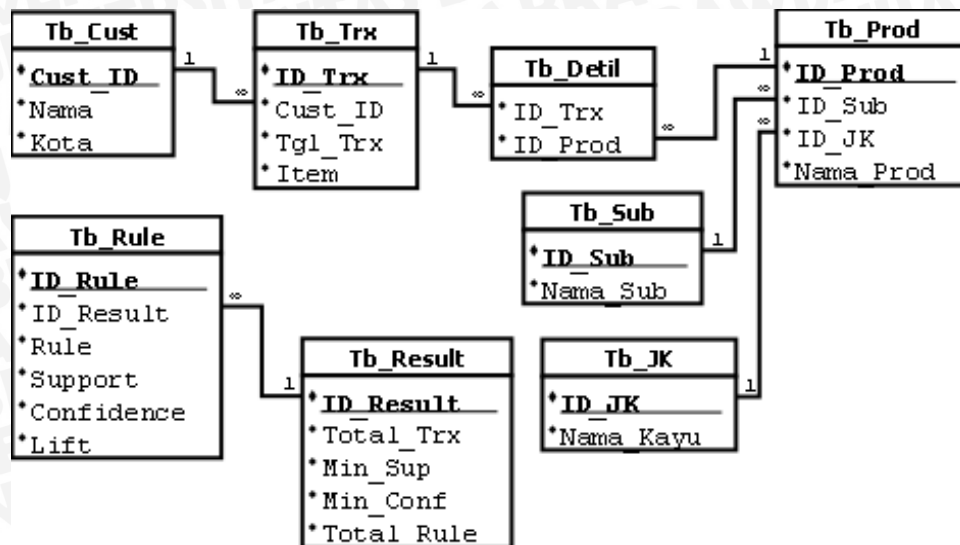
6. Tabel Pelanggan (Tb_Cust)

Tabel Tb_Cust digunakan untuk menyimpan data pelanggan. Setiap pelanggan dapat melakukan lebih dari satu transaksi. Data yang disimpan dalam tabel ini yaitu Cust_ID, Nama dan Kota. Tb_Cust ditunjukkan dalam Tabel 3.6.

Tabel 3.6 Tabel Konsumen

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
<u>Cust_ID</u>	Varchar(32)	Kode pelanggan berupa <i>primary key</i>
Nama	Varchar(32)	Nama pelanggan
Kota	Varchar(32)	Alamat pelanggan diwakili dengan nama kota

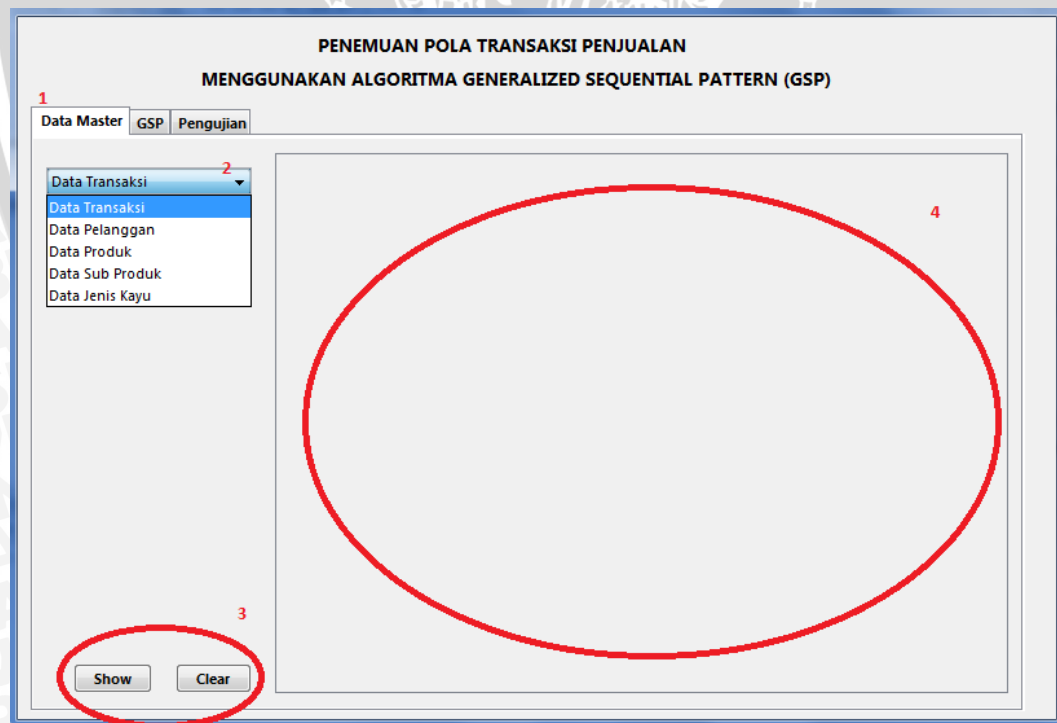
Dari tabel-tabel yang telah dijelaskan, maka dapat dibentuk sebuah relasi yang menghubungkan tabel satu dengan tabel yang lain. Relasi antar tabel tersebut dapat dilihat pada Gambar 3.6.



Gambar 3.6 Relasi Antar Tabel

3.2.3 Perancangan Antarmuka

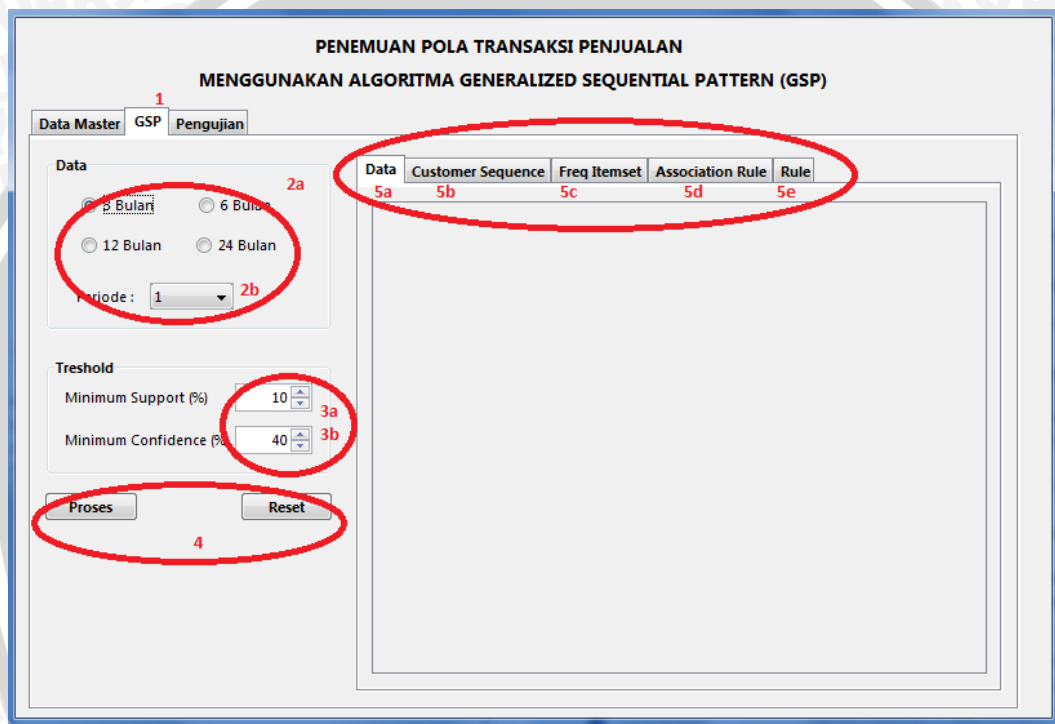
Dalam sistem yang dibuat ini dibutuhkan antarmuka untuk memudahkan pengguna dalam melakukan analisis. Rancangan antarmuka terdiri atas tiga bagian, yaitu Data Master, GSP, dan Pengujian. Rancangan antar muka Data Master dapat dilihat pada Gambar 3.7.



Gambar 3.7 Rancangan Antarmuka Data Master

Berdasarkan Gambar 3.7, antarmuka Data Master terdiri atas beberapa komponen, yaitu:

1. Tab panel Data master, digunakan untuk menampilkan data-data yang dibutuhkan sistem.
2. Combo box Data Master, berisi list data apa saja yang dapat ditampilkan.
3. Tombol Show dan Clear, digunakan untuk menampilkan dan membersihkan tabel area.
4. Tabel area, digunakan untuk menampilkan data yang ingin ditampilkan.



Gambar 3.8 Rancangan Antarmuka GSP

Rancangan antarmuka GSP dapat dilihat pada Gambar 3.8, yang meliputi beberapa bagian, yaitu:

1. Tab panel GSP, digunakan untuk melihat proses pencarian *frequent itemset*, proses *association aturan* dan pengujian data.
2. Panel Data, digunakan untuk memilih data yang akan diuji berdasarkan pilihan rentang bulan 3 bulan, 6 bulan, 12 bulan dan 24 bulan (2a) serta berdasarkan periode tertentu.(2b).
3. Panel Threshold, digunakan untuk memasukkan nilai *minimum support* (3a) dan nilai *minimum confidence* (3b)

4. Tombol Proses dan Reset, digunakan untuk memproses dan mereset proses pencarian pola.
5. Tab Panel yang terdiri atas lima sub, yaitu tab panel Data (5a), tab panel Customer Sequence (5b), tab panel Frequent *Itemset* (5c), tab panel Association *Rule* (5d), tab panel *Rule* (5e)

Gambar 3.9 Rancangan Antarmuka Pengujian

Rancangan antarmuka Pengujian sistem dapat dilihat pada Gambar 3.9, yang meliputi beberapa bagian, yaitu:

1. Tab panel Pengujian, digunakan untuk melihat proses pencarian nilai akurasi dari *rule-rule* yang dihasilkan pada tab GSP.
2. Panel Data, digunakan untuk memilih data yang akan diuji berdasarkan pilihan rentang bulan 3 bulan, 6 bulan, 12 bulan dan 24 bulan (2a) serta berdasarkan periode tertentu.(2b).
3. Tombol Pengujian, digunakan untuk menjalankan proses pencarian nilai akurasi.
4. Field Akurasi, digunakan untuk menampilkan nilai akurasi *rule* dalam persen (%).
5. Tabel Area, digunakan untuk menampilkan data yang diuji berdasarkan pilihan pengguna pada table panel data.

3.2.4 Perancangan Uji Coba

Uji coba sistem untuk analisa pola transaksi penjualan ini melakukan evaluasi terhadap hasil analisa yang dihasilkan oleh sistem. Berdasarkan rumusan masalah yang telah dijelaskan di bab sebelumnya, maka pada penelitian ini dilakukan dua pengujian. Pengujian pertama bertujuan untuk mengetahui pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan. Sedangkan pengujian kedua bertujuan untuk mengetahui nilai akurasi yang dihasilkan oleh algoritma GSP.

Pengujian dilakukan beberapa kali untuk mencari *rule* berdasarkan periode waktu. Rentang bulan untuk data transaksi yang akan digunakan dalam kedua pengujian ini adalah per 3 bulan, 6 bulan, per 12 bulan (1 tahun) dan per 24 bulan (2 tahun). Nilai *minimum support* yang digunakan pada uji pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah aturan yang dihasilkan adalah 10% sampai 30% tiap periode dengan nilai *minimum confidence* 40% sampai 60% tiap nilai *minimum support*. Tabel uji pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah aturan yang dihasilkan ditunjukkan pada Tabel 3.7.

Tabel 3.7 Tabel Pengujian 1 : Pengaruh *Minimum Support* dan *Minimum Confidence* terhadap Jumlah *Rule*

Rentang Bulan	Periode	Min Sup	Min Conf	Jumlah Rule

Untuk pengujian kedua, tabel pengujian nilai akurasi *rule* ditunjukkan pada Tabel 3.8.

Tabel 3.8 Tabel Pengujian 2 : Akurasi *Rule*

Rentang Bulan	Periode	Akurasi

3.3 Perhitungan Manual

Perhitungan manual dilakukan pada sampel data yang diambil dari data transaksi penjualan pada bulan Januari 2011 sebanyak 33 transaksi. Sampel data produk ditunjukkan pada Tabel 3.9.

Tabel 3.9 Tabel Sampel Data Produk

ID_Prod	Nama Produk
104	Partisi Akasia
108	Rak Akasia
113	Frame/kusen Bangkirai
118	Partisi Bangkirai
123	Papan kayu Bangkirai
126	Pintu Bangkirai
128	Sekat kayu Bangkirai
134	Frame/kusen Jati
145	Pintu Jati
150	Partisi Kamper
158	Flooring Lapis
161	Handrailling Lapis
171	Pintu Lapis
188	Pintu Merbau

Pada perhitungan manual ini, nilai *minimum support* yang digunakan adalah 20%, sedangkan nilai *minimum confidence* yang digunakan adalah 40%. Sampel data yang digunakan dapat dilihat pada Tabel 3.10.

Tabel 3.10 Tabel Sampel Data Transaksi

ID_Trx	ID_Cust	Item
1101007	1004	126, 161
1101033	1004	104, 113
1101005	1007	108, 145
1101019	1007	158, 161, 171, 188
1101001	1010	118, 134, 145
1101008	1010	113
1101017	1010	134
1101029	1010	104, 108, 188
1101009	1020	113, 126
1101014	1020	166
1101020	1020	171
1101030	1020	188
1101002	1021	113, 134
1101006	1021	145

ID_Trx	ID_Cust	Item
1101011	1021	188
1101016	1021	118, 161
1101024	1021	134, 158
1101031	1021	104, 126, 128
1101010	1025	188
1101013	1025	104, 108
1101018	1035	123, 161
1101025	1040	134, 145
1101003	1042	104
1101021	1042	134, 158
1101022	1043	104, 171
1101026	1043	145, 150, 171
1101015	1064	126, 158, 188
1101023	1064	145, 171
1101032	1064	108, 113
1101027	1066	113, 126
1101004	1079	108, 188
1101012	1079	145, 171
1101028	1079	104, 188

Langkah selanjutnya, data diurutkan berdasarkan ID_Cust. Kemudian dari data yang telah terurut berdasarkan ID_Cust dibuat tabel baru yaitu tabel sekuen pelanggan yang ditunjukkan pada Tabel 3.11.

Tabel 3.11 Tabel Sekuen Pelanggan

Seq_ID	ID_Cust	Sekuen
1	1004	(126, 161) (104, 113)
2	1007	(108, 145) (158, 161, 171, 188)
3	1010	(118, 134, 145) (113) (134) (104, 108, 188)
4	1020	(113, 126) (166) (171) (188)
5	1021	(113, 134) (145) (188) (118, 161) (134, 158) (104, 126, 128)
6	1025	(188) (104, 108)
7	1035	(123, 161)
8	1042	(104) (134, 158)
9	1043	(104, 171) (145, 150, 171)
10	1040	(134, 145)

Seq_ID	ID_Cust	Sekuen
11	1064	(126, 158, 188) (145, 171) (108, 113)
12	1066	(113, 126)
13	1079	(108, 188) (145, 171) (104, 188)

Berdasarkan Tabel 3.11, diketahui urutan *item-item* barang yang dibeli. Kemudian dari sekuen tersebut dicari kandidat 1-*itemset* berdasarkan nilai batasan yang telah diberikan. Untuk mencari kandidat 1-*itemset*, dicari dahulu support dari masing-masing *item*.

Misalkan untuk menentukan nilai *support* untuk *item* (104), maka dicari dahulu jumlah kemunculan *item* tersebut dalam tabel sekuen pelanggan yaitu sebanyak 7 kali. Selanjutnya dari nilai tersebut dibagi dengan banyaknya transaksi (sekuen pelanggan) yaitu sebanyak 13 sekuen transaksi. Untuk menentukan nilai *support* dari *itemset* telah dijelaskan pada Bab II menggunakan Persamaan 2.1. Maka untuk *item* (104) dihasilkan nilai *support* adalah sebesar 54%. Tabel Kandidat 1-*itemset* dapat ditunjukkan pada Tabel 3.12.

Tabel 3.12 Tabel Kandidat 1-*itemset*

1- <i>itemset</i>	Count	Support
104	7	54%
108	5	38%
113	6	46%
118	2	15%
123	1	8%
126	4	31%
128	1	8%
134	4	31%
145	7	54%
150	1	8%
158	4	31%
161	4	31%
171	5	38%
188	7	54%

Dari Tabel 3.12, diketahui nilai *support* tiap-tiap kandidat. Kandidat yang tidak memenuhi min_sup akan dihapus sedangkan kandidat yang memenuhi

min_sup dimasukkan dalam tabel *Large 1-itemset*. Tabel *Large 1-itemset* ditunjukkan pada Tabel 3.13, yang selanjutnya *itemset* dari tabel tersebut digunakan untuk mencari kandidat *2-itemset*.

Tabel 3.13 Tabel *Large 1-itemset*

Large 1-itemset	Count	Support
104	7	54%
108	5	38%
113	6	46%
126	4	31%
134	4	31%
145	7	54%
158	4	31%
161	4	31%
171	5	38%
188	7	54%

Pada saat proses *join*, harus diperhatikan apakah *kandidat* tersebut memenuhi persyaratan *subsequence* dari *s* atau tidak. Apabila tidak memenuhi maka tidak termasuk dalam kandidat *sequence*. Hasil *kandidat 2-itemset* yang terbentuk ditunjukkan pada Tabel 3.14.

Tabel 3.14 Kandidat *2-itemset*

2-itemset	Σ	Supp	2-itemset	Σ	Supp	2-itemset	Σ	Supp
(104, 108)	2	15%	(104) (108)	0	0%	(145) (104)	3	23%
(104, 113)	1	8%	(104) (113)	0	0%	(145) (108)	2	15%
(104, 126)	1	8%	(104) (126)	0	0%	(145) (113)	1	8%
(104, 134)	0	0%	(104) (134)	1	8%	(145) (126)	1	8%
(104, 145)	0	0%	(104) (145)	1	8%	(145) (134)	2	15%
(104, 158)	0	0%	(104) (158)	1	8%	(145) (158)	2	15%
(104, 161)	0	0%	(104) (161)	0	0%	(145) (161)	2	15%
(104, 171)	0	0%	(104) (171)	1	8%	(145) (171)	1	8%
(104, 188)	2	15%	(104) (188)	0	0%	(145) (188)	4	31%
(108, 113)	1	8%	(108) (104)	1	8%	(158) (104)	1	8%
(108, 126)	0	0%	(108) (113)	0	0%	(158) (108)	1	8%
(108, 134)	0	0%	(108) (126)	0	0%	(158) (113)	1	8%
(108, 145)	1	8%	(108) (134)	0	0%	(158) (126)	1	8%

2-itemset	Σ	Supp	2-itemset	Σ	Supp	2-itemset	Σ	Supp
(108, 158)	0	0%	(108) (145)	1	8%	(158) (134)	0	0%
(108, 161)	0	0%	(108) (158)	1	8%	(158) (145)	1	8%
(108, 171)	0	0%	(108) (161)	1	8%	(158) (161)	0	0%
(108, 188)	2	15%	(108) (171)	2	15%	(158) (171)	1	8%
(113, 126)	2	15%	(108) (188)	2	15%	(158) (188)	0	0%
(113, 134)	1	8%	(113) (104)	2	15%	(161) (104)	2	15%
(113, 145)	0	0%	(113) (108)	1	8%	(161) (108)	0	0%
(113, 158)	0	0%	(113) (126)	1	8%	(161) (113)	1	8%
(113, 161)	0	0%	(113) (134)	2	15%	(161) (126)	1	8%
(113, 171)	0	0%	(113) (145)	1	8%	(161) (134)	1	8%
(113, 188)	0	0%	(113) (158)	1	8%	(161) (145)	0	0%
(126, 134)	0	0%	(113) (161)	1	8%	(161) (158)	1	8%
(126, 145)	0	0%	(113) (171)	1	8%	(161) (171)	0	0%
(126, 158)	1	8%	(113) (188)	3	23%	(161) (188)	0	0%
(126, 161)	2	15%	(126) (104)	1	8%	(171) (104)	1	8%
(126, 171)	0	0%	(126) (108)	1	8%	(171) (108)	1	8%
(126, 188)	1	8%	(126) (113)	2	15%	(171) (113)	1	8%
(134, 145)	2	15%	(126) (134)	0	0%	(171) (126)	0	0%
(134, 158)	2	15%	(126) (145)	1	8%	(171) (134)	0	0%
(134, 161)	0	0%	(126) (158)	0	0%	(171) (145)	1	8%
(134, 171)	0	0%	(126) (161)	0	0%	(171) (158)	0	0%
(134, 188)	0	0%	(126) (171)	2	15%	(171) (161)	0	0%
(145, 158)	0	0%	(126) (188)	1	8%	(171) (188)	2	15%
(145, 161)	0	0%	(134) (104)	2	15%	(188) (104)	2	15%
(145, 171)	3	23%	(134) (108)	1	8%	(188) (108)	1	8%
(145, 188)	0	0%	(134) (113)	1	8%	(188) (113)	1	8%
(158, 161)	1	8%	(134) (126)	1	8%	(188) (126)	1	8%
(158, 171)	1	8%	(134) (145)	1	8%	(188) (134)	1	8%
(158, 188)	2	15%	(134) (158)	1	8%	(188) (145)	2	15%
(161, 171)	1	8%	(134) (161)	1	8%	(188) (158)	1	8%
(161, 188)	1	8%	(134) (171)	0	0%	(188) (161)	1	8%
(171, 188)	1	8%	(134) (188)	2	15%	(188) (171)	2	15%

Kemudian dari kandidat 2-itemset didapatkan *large 2-itemset* dengan menghapus kandidat dengan nilai *support* kurang dari *min_sup*. Hasil *large 2-itemset* ditunjukkan pada Tabel 3.15.

Tabel 3.15 Tabel *Large 2-itemset*

Large 2-itemset	Count	Support
(145, 171)	3	23%
(113) (188)	3	23%
(145) (104)	3	23%
(145) (188)	4	31%

Selanjutnya, dilakukan proses *join* untuk mendapatkan kandidat 3-itemset. Hasil proses *join* ditunjukkan pada Tabel 3.16.

Tabel 3.16 Tabel Kandidat 3-itemset

3-itemset	Count	Support
(145, 171)(104)	1	8%
(145, 171)(188)	1	8%
(113)(145)(188)	1	8%
(145)(113)(188)	1	8%

Dari tabel 3.16 tidak ditemukan aturan yang memenuhi *min_sup*, maka kandidat 3-itemset tersebut tidak dapat dimasukkan sebagai pola frekuensi tinggi. Setelah seluruh pola ditemukan, selanjutnya dicari aturan asosiatif yang memenuhi syarat *min_conf*. Aturan asosiatif yang tidak memenuhi *min_conf* (40%) akan dihapus.

Menghitung nilai *minimum confidence* telah dijelaskan pada Bab II menggunakan Persamaan 2.3. Misalkan untuk mencari *confidence* untuk pola (145, 171), pertama dicari banyaknya kemunculan *item* (145) dan *item* (171) dalam satu transaksi, yaitu sebanyak 3 kali. Selanjutnya dibagi dengan banyaknya transaksi mengandung *item* (145), yaitu sebanyak 7 kali kemudian dikalikan dengan 100%. Didapatkan nilai *confidence* untuk pola (145, 171) adalah sebesar 43%. Dari tahapan-tahapan diatas, ditemukan pola frekuensi tinggi yang terbentuk ditunjukkan pada Tabel 3.17.

Tabel 3.17 Tabel Pola Frekuen Tinggi

Pola Frekuen Tinggi	Conf
(145, 171)	43%
(171, 145)	60%

Pola Frekuen Tinggi	Conf
(113) (188)	50%
(145) (104)	43%

Berdasarkan Tabel 3.17, maka didapatkan aturan asosiatif yang telah memenuhi syarat min_conf. Aturan-aturan yang terbentuk ditunjukkan pada Tabel 3.18.

Tabel 3.18 Tabel Aturan Asosiatif

Aturan	Keterangan
(145, 171)	Jika membeli Pintu Jati, maka membeli Pintu Lapis
(113) (188)	Jika membeli Frame/kusen Bangkirai, maka pada transaksi selanjutnya akan membeli Pintu Merbau
(145) (104)	Jika membeli Pintu Jati, maka pada transaksi selanjutnya akan membeli Partisi Akasia
(145) (188)	Jika membeli Pintu Jati, maka pada transaksi selanjutnya akan membeli Pintu Merbau

BAB IV

IMPLEMENTASI

4.1 Lingkungan Implementasi

Implementasi dilakukan untuk proses transformasi representasi rancangan ke dalam bahasa pemrograman sehingga dapat dimengerti oleh komputer. Lingkungan implementasi yang dijelaskan meliputi lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan pada penelitian ini adalah sebagai berikut:

1. Processor Intel® Core i3 2.10GHz
2. Memori 2GB
3. Harddisk dengan kapasitas 320 GB
4. Monitor 14.0"
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut:

1. Sistem operasi Microsoft Windows 7 Ultimate
2. Netbeans IDE 6.9.1 untuk menulis dan meng-*compile* program yang ditulis dengan bahasa pemrograman Java.
3. JDK 1.6, yaitu *development kit* untuk menulis program java.
4. MySQL, yaitu perangkat lunak untuk manajemen basis data SQL
5. PhpMyAdmin untuk mengelola administrasi MySQL.

4.2 Implementasi Program

Berdasarkan rancangan pembuatan sistem pada bab 3, maka pada subbab ini dijelaskan implementasi proses-proses pencarian pola menggunakan Algoritma *Generalized Sequential Pattern* tersebut. Secara garis besar proses ini dikelompokkan menjadi 4 tahap, yaitu tahap pembentukan sekuen, tahap *large 1-itemset*, tahap pembentukan kandidat dan tahap pembentukan *rule*.

4.2.1 Implementasi Tahap Pembentukan Sekuen

Proses awal yang dilakukan pada tahap ini adalah dengan melakukan pembacaan data transaksi dalam periode waktu tertentu. Selanjutnya data transaksi tersebut diurutkan berdasarkan pelanggan untuk kemudian dibentuk tabel sekuen pelanggan. Implementasi dari proses ini ditunjukkan dalam *source code* 4.1.

```

1 private LinkedHashMap getCustomerSeq(ArrayList distinct){
    LinkedHashMap seq=new LinkedHashMap();
2     for (int i = 0; i < distinct.size(); i++) {
3         ArrayList date=db.getAllData("SELECT
            distinct(Tgl_Trx) FROM tb_trx WHERE id_Cust =
            '"+distinct.get(i)+"' AND
            month(tgl_trx)>="+lower+" AND
            month(tgl_trx)<="+upper+" AND
            year(tgl_trx)>="+ltahun+" AND
            year(tgl_trx)<="+utahun+" ORDER BY (Tgl_Trx) ASC",
            "Tgl_Trx");
4         ArrayList temp=new ArrayList();
5         ArrayList cek=new ArrayList();
6         for (int j = 0; j < date.size(); j++) {
7             if(!cek.contains(date.get(j))){
8                 cek.add(date.get(j));
9                 ArrayList temp2=db.getAllData("SELECT
                    id_prod FROM tb_detil a, tb_trx b WHERE
                    id_cust = '"+distinct.get(i)+"' AND
                    tgl_trx = '"+date.get(j)+"' AND a.id_trx =
                    b.id_trx AND month(tgl_trx)>="+lower+" AND
                    month(tgl_trx)<="+upper+" AND
                    year(tgl_trx)>="+ltahun+" AND
                    year(tgl_trx)<="+utahun+" ORDER BY
                    (id_prod)", "id_Prod");
10                temp.add(temp2);
11            }
12        }
13        tbCQ.addRow(new Object[]{distinct.get(i),temp});
14        seq.put(distinct.get(i), temp);
15    }

```

16	return seq;
17	}

Source Code 4.1 Proses Pembentukan Sekuen Pelanggan

Source code pembacaan data transaksi dari basis data dalam periode waktu tertentu ditunjukkan pada baris 3, kemudian hasil dari pembacaan tersebut disimpan dalam *ArrayList*. Selanjutnya data transaksi tersebut diurutkan berdasarkan pelanggan untuk kemudian disimpan dalam tabel sekuen pelanggan yang ditunjukkan pada baris 13.

4.2.2 Implementasi Tahap Large 1-itemset

Pada tahap ini, tiap-tiap *itemset* dihitung frekuensi kemunculan dan nilai *support*-nya. Setelah diketahui nilai *support* tiap-tiap *itemset*, bagi *itemset* yang tidak memenuhi *min_sup* akan dihapus, sedangkan *itemset* yang memenuhi *min_sup* akan masuk dalam tabel *large 1-itemset*. Pada proses pembentukan *Large 1-itemset*, diawali dengan pembentukan kandidat *itemset* yang ditunjukkan pada *Source Code 4.2*.

1	private LinkedHashMap getCandidate1(ArrayList distinct,
	ArrayList idProd, LinkedHashMap customerSeq) {
2	LinkedHashMap candidate = new LinkedHashMap();
3	int nSequence=0;
4	for (int i = 0; i < idProd.size(); i++) {
5	int sumTransact = getSupport(idProd.get(i),
	customerSeq);
6	double support = ((double)sumTransact / (double)
	distinct.size()) * 100;
7	candidate.put(idProd.get(i), support);
8	}
9	return candidate;
10	}

Source Code 4.2 Proses Pembentukan Kandidat 1-itemset

Kandidat yang dibentuk akan disimpan dalam *LinkedHashMap*. Dari tiap-tiap *itemset* dihitung frekuensi kemunculan dan nilai *support*-nya. *Source code* untuk mendapatkan frekuensi kemunculan tiap-tiap *item* produk ditunjukkan pada baris 5 dan untuk mencari nilai *support* ditunjukkan pada baris 6.

Dari kandidat yang telah dibentuk, selanjutnya akan diproses pada method `eliminateMinSupport`, yaitu method untuk mengeliminasi kandidat-kandidat yang tidak memenuhi *minimum support*. Jika terdapat *itemset* yang tidak memenuhi syarat `min_sup`, maka *item* tersebut akan dihapus, dan *itemset* yang memenuhi `min_sup` akan masuk dalam tabel *large 1-itemset*. Proses ini ditunjukkan dalam Source Code 4.3.

```

1 private LinkedHashMap eliminateMinSupport(LinkedHashMap
  candidate1, ArrayList idProd, double support) {
2     LinkedHashMap candidate = new LinkedHashMap();
3     for (int i = 0; i < idProd.size(); i++) {
4         double supportCand = Double.parseDouble
          (candidate1.get(idProd.get(i)).toString());
5         if (supportCand >= support) {
6             tblItemset.addRow(new Object[]{idProd.get(i),
              bulat(supportCand)});
7             candidate.put(idProd.get(i), supportCand);
8         }
9     }
10    return candidate;
11 }

```

Source Code 4.3 Proses Pembentukan *Large 1-itemset*

4.2.3 Implementasi Pembentukan Kandidat

Yang dimaksud pembentukan kandidat pada tahapan ini adalah proses *join* dan *pruning* dari *large 1-itemset* untuk menghasilkan kombinasi *itemset* baru hingga ditemukan *large n-itemset*. Pada penelitian kali ini proses pembentukan kandidat hingga kedalaman *3-itemset*.

Pembentukan *Large 2-itemset* didapatkan dari proses *join large 1-itemset*, proses ini ditunjukkan dalam Source Code 4.4.

```

1 private ArrayList<Pair>getPair(LinkedHashMap elmi){
2     ArrayList item=new ArrayList();
3     for(Object key:elmi.keySet())
4         item.add(key);
5     ArrayList<Pair>itemset=new ArrayList<Pair>();
6     for (int i = 0; i < item.size()-1; i++) {
7         for (int j = i+1; j < item.size(); j++) {

```

```

8      Pair p=new Pair();
9      p.setPair(item.get(i), item.get(j));
10     itemset.add(p);
11 }
12 }
13 return itemset;
14 }
15 private ArrayList<Pair> sameTransact(ArrayList<Pair>itemset,
LinkedHashMap cq, double support){
16     ArrayList<Pair> sup=new ArrayList<Pair>();
17     for (int p = 0; p < itemset.size(); p++) {
18         int countIn=countSame(itemset.get(p), cq);
19         double support1 = ((double)countIn /
(double)cq.size())*100;
20         if(support1>support){
21             itemset.get(p).setFrekuensi(countIn);
22             sup.add(itemset.get(p));
23             String pairBaru=transactPair("same",
itemset.get(p));
24             System.out.println(pairBaru+" = "+support1);
25             tb2Itemset.addRow(new Object[]{pairBaru,
bulat(support1)});
26         }
27     }
28     return sup;
29 }
30 private int countSame(Pair p,LinkedHashMap cq){
31     int countIn=0;
32     for(Object key:cq.keySet()){
33         ArrayList temp=(ArrayList) cq.get(key);
34         for (int i = 0; i < temp.size(); i++) {
35             ArrayList temp2=(ArrayList) temp.get(i);
36             if(temp2.contains(p.getObject1()) &&
temp2.contains(p.getObject2()))
37                 countIn++;
38         }
39     }
40     return countIn;
41 }
42 }

```

```
43 private ArrayList<Pair> diffTransact (ArrayList<Pair> itemset,  
    LinkedHashMap cq, double support){  
44     ArrayList<Pair> sup=new ArrayList<Pair>();  
45     for (int p = 0; p < itemset.size(); p++) {  
46         int countOut=countDiff(itemset.get(p), cq);  
47         double support2 = ((double)countOut /  
            (double)cq.size()) * 100;  
48         if(support2>support){  
49             itemset.get(p).setFrekuensi(countOut);  
50             sup.add(itemset.get(p));  
51             String pairBaru=transactPair("diff",  
                itemset.get(p));  
52             tb2Itemset.addRow(new Object[] {pairBaru,  
                bulat(support2)});  
53         }  
54     }  
55     return sup;  
56 }  
57 private int countDiff(Pair p, LinkedHashMap cq){  
58     int countOut=0;  
59     for(Object key:cq.keySet()){  
60         int object1=0;  
61         int object2=0;  
62         ArrayList temp=(ArrayList) cq.get(key);  
63         for (int i = 0; i < temp.size(); i++) {  
64             ArrayList temp2=(ArrayList) temp.get(i);  
65             if(temp2.contains(p.getObject1()))  
66                 object1++;  
67             if(temp2.contains(p.getObject2()))  
68                 object2++;  
69         }  
70         int cek=cekMin(object1, object2);  
71         if(cek==1){  
72             ArrayList con1=new ArrayList();  
73             ArrayList con2=new ArrayList();  
74             for (int i = 0; i < temp.size(); i++) {  
75                 ArrayList temp2=(ArrayList) temp.get(i);  
76                 if(temp2.contains(p.getObject1()))  
77                     con1.add(i);
```

```

78         if(temp2.contains(p.getObject2()))
79             con2.add(i);
80     }
81     if(con1.size() == con2.size() &&
        con1.containsAll(con2))
82         cek=0;
83     }    countOut+=cek;
84 }
85 return countOut;
86 }

```

Source Code 4.4 Proses Pembentukan *Large 2-itemset*

Setelah dilakukan proses pembentukan *large 2-itemset*, selanjutnya pembentukan *large 3-itemset*. Proses ini ditunjukkan dalam Source Code 4.5.

```

1  private LinkedHashMap<Object,ArrayList<Triple>>
   getLarge3Itemset(ArrayList<Pair> same,ArrayList<Pair> diff){
2      LinkedHashMap<Object,ArrayList<Triple>> type=new
        LinkedHashMap<Object, ArrayList<Triple>>();
3      type.put("same-same", getSameJoin(same));
4      type.put("diff-diff", getSameJoin(diff));
5      type.put("same-diff", getDiffJoin(same, diff));
6      type.put("diff-same", getDiffJoin(diff, same));
7      return type;
8  }
9  private ArrayList<Triple>getSameJoin(ArrayList<Pair>same){
10     ArrayList<Triple>array=new ArrayList<Triple>();
11     for (int i = 0; i < same.size()-1; i++) {
12         for (int j = i+1; j < same.size(); j++) {
13             if(same.get(i). getObject2().
14                 equals(same.get(j). getObject1())){
15                 Triple temp=new Triple();
16                 temp.setTriple(same.get(i).getObject1(),
17                     same.get(i).getObject2(),
18                     same.get(j).getObject2());
19                 array.add(temp);
20             }
21         }
22     }
23     return array;

```

```
23     }
24     private ArrayList<Triple>getDiffJoin(ArrayList<Pair>pair1,
    ArrayList<Pair> pair2){
25         ArrayList<Triple>array=new ArrayList<Triple>();
26         for (int i = 0; i < pair1.size(); i++) {
27             for (int j = 0; j < pair2.size(); j++) {
28                 if(pair1.get(i). getObject2().
                    equals(pair2.get(j). getObject1())){
29                     Triple temp=new Triple();
30                     temp.setTriple(pair1.get(i).getObject1(),
31                     pair1.get(i).getObject2(),
32                     pair2.get(j).getObject2());
33                     array.add(temp);
34                 }
35             }
36         }
37         return array;
38     }
39     Private LinkedHashMap<Object,ArrayList<Triple>>
    large3Support(LinkedHashMap<Object,ArrayList<Triple>>itemset,
    LinkedHashMap cq,double support){
40         System.out.println("ITEMSET 3 SUPPORT");
41         LinkedHashMap<Object,ArrayList<Triple>>lgSup=new
    LinkedHashMap<Object, ArrayList<Triple>>();
42         for(Object key:itemset.keySet()){
43             ArrayList<Triple>arr=itemset.get(key);
44             ArrayList<Triple>baru=new ArrayList<Triple>();
45             for (int i = 0; i < arr.size(); i++) {
46                 int count=0;
47                 for(Object id:cq.keySet()){
48                     ArrayList cek1=new ArrayList();
49                     ArrayList cek2=new ArrayList();
50                     ArrayList cek3=new ArrayList();
51                     ArrayList item=(ArrayList) cq.get(id);
52                     for (int j = 0; j < item.size(); j++) {
53                         ArrayList x=(ArrayList) item.get(j);
54                         Object obj1=arr.get(i).getObject1();
55                         Object obj2=arr.get(i).getObject2();
56                         Object obj3=arr.get(i).getObject3();
```

```
54         if(key.equals("same-same")){
55             if(x.contains(obj1) &&
56                 x.contains(obj2) &&
57                 x.contains(obj3))
58                 count++;
59         }
60     else if(key.equals("diff-diff")){
61         if(x.contains(obj1))
62             cek1.add(j);
63         if(x.contains(obj2))
64             cek2.add(j);
65         if(x.contains(obj3))
66             cek3.add(j);
67     }
68     else if(key.equals("same-diff")){
69         if(x.contains(obj1)&&x.contains(obj2))
70             cek1.add(j);
71         if(x.contains(obj3))
72             cek2.add(j);
73     }
74     else if(key.equals("diff-same")){
75         if(x.contains(obj2)&&x.contains(obj3))
76             cek1.add(j);
77         if(x.contains(obj1))
78             cek2.add(j);
79     }
80 }
81 if(key.equals("diff-diff")){
82     int c1 = countSingleDeep(arr.get(i).
83         getObject1(), (ArrayList) cq.get(id));
84     int c2 = countSingleDeep(arr.get(i).
85         getObject2(), (ArrayList) cq.get(id));
86     int c3 = countSingleDeep(arr.get(i).
87         getObject3(), (ArrayList) cq.get(id));
88     int min3=cekMinTriple(c1, c2, c3);
89     int cx=count3(cek1, cek2, cek3);
90     count+=cekMin(min3, cx);
91 }
92 else if(key.equals("same-diff")){
```

```

88         int c1 =
            countSameDeep(aPair(arr.get(i)),
                (ArrayList) cq.get(id));
89         int c2 =
            countSingleDeep(arr.get(i).getObject3(
                ), (ArrayList) cq.get(id));
90         int min3=cekMin(c1, c2);
91         int cx=count2(cek1, cek2);
92         count+=cekMin(min3, cx);
93     }
94     else if(key.equals("diff-same")){
95         int c1 =
            countSameDeep(bPair(arr.get(i)),
                (ArrayList) cq.get(id));
96         int c2 =
            countSingleDeep(arr.get(i).getObject1(
                ), (ArrayList) cq.get(id));
97         int min3=cekMin(c1, c2);
98         int cx=count2(cek1, cek2);
99         count+=cekMin(min3, cx);
100    }
101    }
102    Double sup=((double)count /
        (double)cq.size())*100;
103    if(sup>=support){
104        arr.get(i).setFrekuensi(count);
105        baru.add(arr.get(i));
106        String tripleBaru=transactTriple((String)
            key, arr.get(i));
107        tb3Itemset.addRow(new
            Object[]{tripleBaru,bulat(sup)});
108    }
109 }
110 lgSup.put(key, baru);
111 }
112 return lgSup;
113 }

```

Source Code 4.5 Proses Pembentukan *Large 3-itemset*

4.2.4 Implementasi Tahap Pembentukan *Rule*

Pada tahapan ini dilakukan proses pembentukan *rule* dari seluruh kandidat yang telah diperoleh. Tahapan dari proses ini yaitu dengan menghitung nilai *confidence* dari setiap *rule* yang terbentuk. Apabila nilai *confidence* dari *rule* lebih besar atau sama dengan nilai batasan, maka dilakukan penyimpanan *rule* pada *tbAsso* yang ditunjukkan pada baris 13, baris 22, dan baris 42. Selanjutnya *rule* yang telah disimpan akan ditampilkan pada tab panel *rule*. Proses pembentukan *rule* ditunjukkan pada Source Code 4.6.

```

1 public void asosiatifRule(LinkedHashMap<Object,
  ArrayList<Triple>> itemset3, ArrayList<Pair> same,
  ArrayList<Pair> diff, LinkedHashMap cq, double minConf) {
2     ArrayList rule = new ArrayList();
3     ArrayList rule2=new ArrayList();
4     String pairBaru = "";
5     for (int i = 0; i < same.size(); i++) {
6         int aFreq = aItemset(same.get(i), cq);
7         int bFreq = bItemset(same.get(i), cq);
8         double conf = confidence(same.get(i).
          getFrekuensi(), aFreq);
9         pairBaru = transactPair("same", same.get(i));
10        if (conf * 100 >= minConf) {
11            rule.add(samePairRule(same.get(i)));
12            rule2.add("("+(same.get(i).getObject1()+" "+sa
             me.get(i).getObject2()+"")");
13            tbAsso.addRow(new Object[]{pairBaru,
              bulat(conf * 100), bulat(lift)});
14        }
15    }
16    for (int i = 0; i < diff.size(); i++) {
17        int aFreq = aItemset(diff.get(i), cq);
18        int bFreq = bItemset(diff.get(i), cq);
19        double conf =
20            confidence(diff.get(i).getFrekuensi(), aFreq);
21        pairBaru = transactPair("diff", diff.get(i));
22        if (conf * 100 >= minConf) {
23            tbAsso.addRow(new Object[]{pairBaru,
              bulat(conf * 100), bulat(lift)});

```

```

23         rule.add(diffPairRule(diff.get(i)));
24         rule2.add("("+(diff.get(i).getObject1()+"|"+diff.get(i).getObject2()+"");
25     }
26 }
27 for (Object key : itemset3.keySet()) {
28     ArrayList<Triple> arr = itemset3.get(key);
29     int backFreq = 0;
30     for (int i = 0; i < arr.size(); i++) {
31         Pair back = bPair(arr.get(i));
32         if (key.equals("same-same") ||
33             key.equals("diff-same")) {
34             backFreq = countSame(back, cq);
35         } else if (key.equals("diff-diff") ||
36             key.equals("same-diff")) {
37             backFreq = countDiff(back, cq);
38         }
39         Pair front = aPair(arr.get(i));
40         int frontFreq = aItemset(front, cq);
41         double conf = confidence(arr.get(i).
42             getFrekuensi(), frontFreq);
43         String tripleBaru = transactTriple((String)
44             key, arr.get(i));
45         if (conf * 100 >= minConf) {
46             tbAsso.addRow(new Object[]{tripleBaru,
47                 bulat(conf * 100), bulat.lift)});
48             Triple tp=arr.get(i);
49             if (key.equals("same-same")) {
50                 rule2.add("("+tp.getObject1()+","+tp.g
51                     etObject2()+","+tp.getObject3()+");
52             } else if (key.equals("diff-diff")) {
53                 rule2.add("("+tp.getObject1()+"|"+tp.g
54                     etObject2()+"|"+tp.getObject3()+");
55             } else if (key.equals("same-diff")) {
56                 rule2.add("("+tp.getObject1()+","+tp.g
57                     etObject2()+"|"+tp.getObject3()+");
58             } else {
59                 rule2.add("("+tp.getObject1()+"|"+tp.g
60                     etObject2()+","+tp.getObject3()+");

```

```

52         }
53         rule.add(tripleRule(key, arr.get(i)));
54     }
55 }
56 }
57 for (int i = 0; i < rule.size(); i++){
58     tbRule.addRow(new Object[]{i+1, rule.get(i),
59         rule2.get(i)});
60 }
61 }
62 }

```

Source Code 4.6 Proses Pembentukan *Rule*

4.2.5 Implementasi Pengujian Akurasi *Rule*

Pada tahapan ini dilakukan proses perhitungan nilai akurasi *rule* dari seluruh *rule* yang telah diperoleh. Tahapan dari proses ini yaitu dengan menghitung nilai prosentase akurasi sesuai dengan Persamaan 2.7. Proses pengujian akurasi *rule* ditunjukkan pada Source Code 4.7.

```

1  private void getAkurasi(ArrayList rule, LinkedHashMap
   customerSeq) {
2      double nError;
3      int nBenar=0; int nBenar1=0; int nBenar2=0;
4      int nBenar3=0; int nBenar4=0;
5      for(Object cust:customerSeq.keySet()){
6          ArrayList temp=(ArrayList) customerSeq.get(cust);
7          for (int i = 0; i < temp.size(); i++) {
8              boolean cek=false;
9              ArrayList temp2=(ArrayList) temp.get(i);
10             for(Object str:rule){
11                 ArrayList temp3=new ArrayList();
12                 String strx=(String) str;
13                 if(!strx.contains("|")){
14                     String[]tempArr;
15                     tempArr=strx.split("\\s+|\\.|,|+|\\[+|\\]|
16                     +|\\(|+|\\)|+|\\|");
17                     for (int j = 0; j < tempArr.length; j++)
18                 {
19                     if(!tempArr[j].equals("")){

```

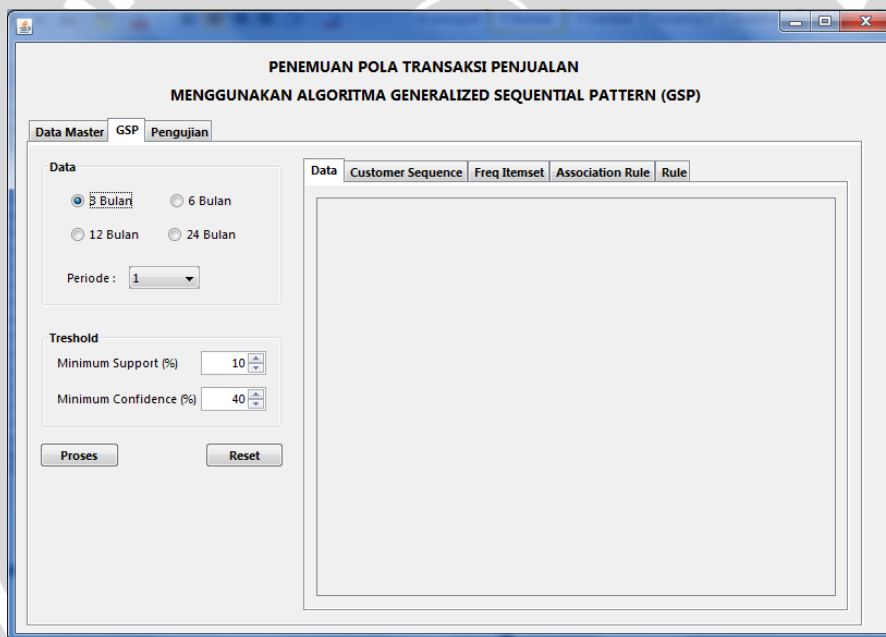
Source Code 4.7 Proses Pengujian Akurasi

4.3 Implementasi Antarmuka

Implementasi antarmuka sistem seperti yang telah dijelaskan dalam rancangan antarmuka pada subbab 3.2, terdiri atas beberapa bagian, yaitu: bagian input, output, tombol proses dan reset. Bagian input dibagi menjadi dua bagian, yaitu panel data dan panel *threshhold*.

Pada panel data, terdapat pilihan bulan berupa RadioButton. Pilihan bulan terdiri atas pilihan per 3 bulan, per 6 bulan, per 12 bulan dan per 24 bulan. Pilihan bulan yang dipilih mengubah isi pilihan pada ComboBox periode bulan.

Pada panel *threshold*, terdapat dua parameter inputan berupa *spinner* yang harus diisi. Dua parameter tersebut yaitu *minimum support* dan *minimum confidence*. Tampilan antarmuka yang telah dijelaskan diilustrasikan pada Gambar 4.1.



Gambar 4.1 Tampilan Antarmuka Program

Setelah data inputan terisi dengan benar, maka ketika tombol Proses ditekan maka akan muncul data transaksi pada panel output data seperti yang diilustrasikan pada Gambar 4.2. Selain itu, pada tab selanjutnya akan ditampilkan tabel *customer sequence*, tabel *frequent itemset*, tabel *association rule* dan tabel *rule* yang dihasilkan.

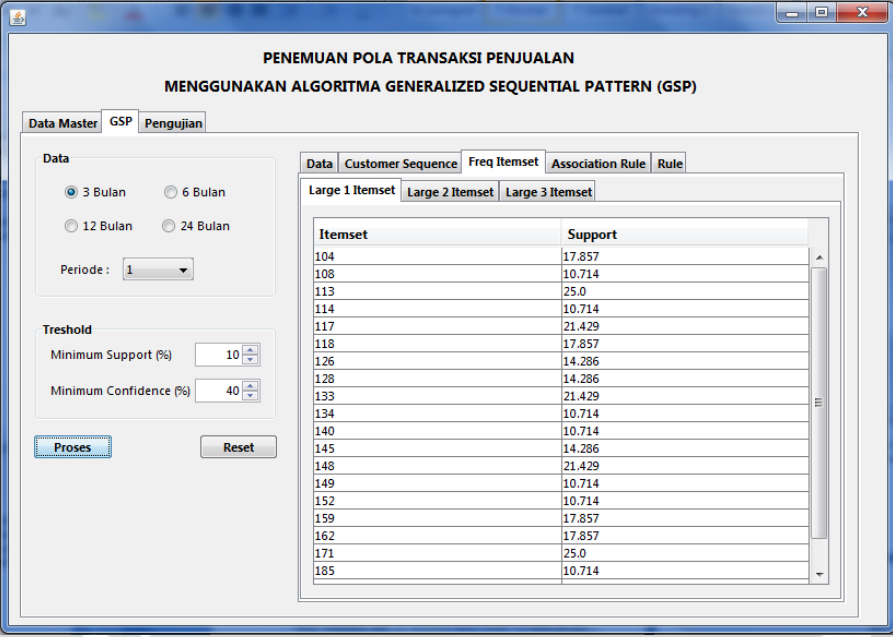
Id Transaksi	Tanggal Transaksi	Id Customer	Item yang Dibeli
1101001	2011-01-03	1010	[145, 137, 118]
1101002	2011-01-03	1021	[134, 113]
1101003	2011-01-04	1042	[104]
1101004	2011-01-05	1079	[108, 185]
1101005	2011-01-06	1007	[117, 143]
1101006	2011-01-07	1021	[145]
1101007	2011-01-10	1004	[125, 161]
1101008	2011-01-11	1010	[114]
1101009	2011-01-11	1020	[114, 127]
1101010	2011-01-12	1092	[187]
1101011	2011-01-13	1021	[188]
1101012	2011-01-15	1079	[155, 154]
1101013	2011-01-17	1025	[105, 108]
1101014	2011-01-18	1020	[166]
1101015	2011-01-18	1064	[159, 126, 188]
1101016	2011-01-19	1021	[118, 164]
1101017	2011-01-20	1010	[134]
1101018	2011-01-20	1035	[123, 161]
1101019	2011-01-21	1007	[159, 171, 162, 188]
1101020	2011-01-24	1020	[154]
1101021	2011-01-24	1042	[133, 158]

Gambar 4.2 Tampilan Data Transaksi

Tabel sekuen pelanggan ditunjukkan dalam Gambar 4.3, yaitu pada Tab *Customer Sequence* yang berisi urutan transaksi yang dilakukan oleh pelanggan dalam satu periode tertentu. Selanjutnya, dalam tabel *frequent itemset* yang ditunjukkan dalam Gambar 4.4, berisi *itemset-itemset* yang telah memenuhi syarat min_sup dan min_conf. *Itemset-itemset* tersebut terbagi atas tiga tabel, yaitu tabel *Large 1-itemset*, tabel *Large 2-itemset* dan tabel *Large 3-itemset*.

No	Id Pelanggan	Sequence
1	1004	[[125, 161]]
2	1007	[[117, 143], [159, 162, 171, 188]]
3	1010	[[118, 137, 145], [114], [134], [105, 108, 185]]
4	1018	[[128]]
5	1019	[[128]]
6	1020	[[114, 127], [166], [154], [187], [159, 162, 171], [118], [128], [113, 117...]]
7	1021	[[113, 134], [145], [188], [118, 164], [133, 158], [104, 126, 128]]
8	1023	[[117, 133, 140], [126], [113, 117, 120, 133]]
9	1025	[[105, 108]]
10	1028	[[117, 133, 140]]
11	1033	[[157, 176], [169, 176], [113]]
12	1035	[[123, 161], [159, 162, 171, 188], [159]]
13	1042	[[104], [133, 158]]
14	1043	[[104, 151], [148, 150, 171]]
15	1050	[[137, 145]]
16	1056	[[118, 123], [115, 129]]
17	1064	[[126, 159, 188], [148, 171], [111, 115], [104], [111, 112, 114, 118, 12...]]
18	1066	[[113, 126]]
19	1070	[[104, 111, 117], [111]]
20	1077	[[113, 185], [148]]
21	1079	[[108, 185], [154, 155], [101, 107]]

Gambar 4.3 Tampilan Tabel Sekuen Pelanggan



PENEMUAN POLA TRANSAKSI PENJUALAN
MENGUNAKAN ALGORITMA GENERALIZED SEQUENTIAL PATTERN (GSP)

Data Master | **GSP** | Pengujian

Data: ☒ 3 Bulan ☐ 6 Bulan
☐ 12 Bulan ☐ 24 Bulan
 Periode: 1

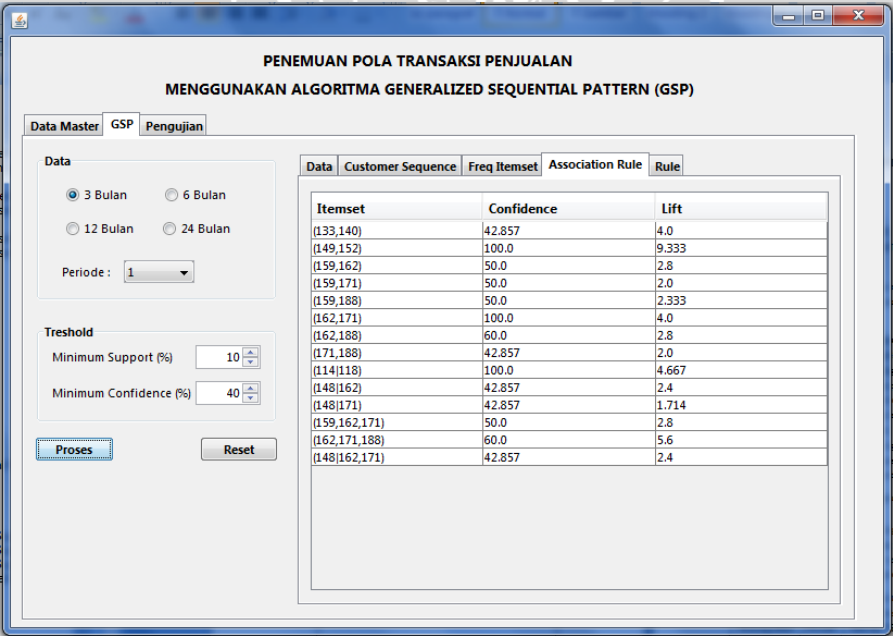
Threshold:
 Minimum Support (%) 10
 Minimum Confidence (%) 40

Proses Reset

Itemset	Support
104	17.857
108	10.714
113	25.0
114	10.714
117	21.429
118	17.857
126	14.286
128	14.286
133	21.429
134	10.714
140	10.714
145	14.286
148	21.429
149	10.714
152	10.714
159	17.857
162	17.857
171	25.0
185	10.714

Gambar 4.4 Tampilan Tabel Frequent *Itemset*

Tabel *association rule* ditunjukkan dalam Gambar 4.5, berisi *itemset-itemset* yang memenuhi syarat *min_sup* disertai dengan nilai *confidence* dan nilai *lift*-nya. Selanjutnya, tabel *rule* ditunjukkan dalam Gambar 4.6, berisi *rule-rule* yang telah memenuhi syarat *min_conf*.



PENEMUAN POLA TRANSAKSI PENJUALAN
MENGUNAKAN ALGORITMA GENERALIZED SEQUENTIAL PATTERN (GSP)

Data Master | GSP | **Pengujian**

Data: ☒ 3 Bulan ☐ 6 Bulan
☐ 12 Bulan ☐ 24 Bulan
 Periode: 1

Threshold:
 Minimum Support (%) 10
 Minimum Confidence (%) 40

Proses Reset

Itemset	Confidence	Lift
{133,140}	42.857	4.0
{149,152}	100.0	9.333
{159,162}	50.0	2.8
{159,171}	50.0	2.0
{159,188}	50.0	2.333
{162,171}	100.0	4.0
{162,188}	60.0	2.8
{171,188}	42.857	2.0
{114,118}	100.0	4.667
{148,162}	42.857	2.4
{148,171}	42.857	1.714
{159,162,171}	50.0	2.8
{162,171,188}	60.0	5.6
{148,162,171}	42.857	2.4

Gambar 4.5 Tampilan Tabel *Association Rule*

PENEMUAN POLA TRANSAKSI PENJUALAN
MENGUNAKAN ALGORITMA GENERALIZED SEQUENTIAL PATTERN (GSP)

Data Master **GSP** Pengujian

Data

☒ 3 Bulan ☐ 6 Bulan
☐ 12 Bulan ☐ 24 Bulan

Periode : 1

Threshold

Minimum Support (%) 10
Minimum Confidence (%) 40

Data Customer Sequence Freq Itemset Association Rule Rule

Nomor	Itemset	Rule
1	Jika membeli Flooring Jati, maka pembelian selanjutnya akan me...	{133,140}
2	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan ...	{149,152}
3	Jika membeli Frame/Kusen Merbau, maka pembelian selanjutnya ...	{159,162}
4	Jika membeli Frame/Kusen Merbau, maka pembelian selanjutnya ...	{159,171}
5	Jika membeli Frame/Kusen Merbau, maka pembelian selanjutnya ...	{159,188}
6	Jika membeli Jendela Merbau, maka pembelian selanjutnya akan ...	{162,171}
7	Jika membeli Jendela Merbau, maka pembelian selanjutnya akan ...	{162,188}
8	Jika membeli Pintu Merbau, maka pembelian selanjutnya akan m...	{171,188}
9	Jika membeli Gezebo Bangkirai, maka pembelian selanjutnya aka...	{114,118}
10	Jika membeli Frame/Kusen Kamper, maka pembelian selanjutnya ...	{148,162}
11	Jika membeli Frame/Kusen Kamper, maka pembelian selanjutnya ...	{148,171}
12	Jika membeli Frame/Kusen Merbau, maka pembelian selanjutnya ...	{159,162,171}
13	Jika membeli Jendela Merbau, maka pembelian selanjutnya akan ...	{162,171,188}
14	Jika membeli Frame/Kusen Kamper, maka pembelian selanjutnya ...	{148,162,171}

Gambar 4.6 Tampilan Tabel *Rule* yang Terbentuk

BAB V

PENGUJIAN DAN ANALISIS

5.1 Implementasi Uji Coba

Pada sub bab ini dijelaskan mengenai hasil uji coba yang telah dilakukan. Uji coba yang dilakukan sesuai dengan sistematika yang telah dipaparkan sebelumnya.

5.2 Implementasi Pengujian dan Pembahasan

Pengujian sistem dengan data penelitian 2 tahun yang telah dijelaskan sebelumnya dilakukan untuk masing-masing pilihan bulan, yaitu per tiga bulan, per enam bulan, per dua belas bulan dan per dua puluh empat bulan. Masing-masing pilihan bulan memiliki jumlah periode yang berbeda, yaitu rentang data per tiga bulan terdiri atas 8 periode (Bulan Januari-Maret, Bulan April-Juni, Bulan Juli-September, Bulan Oktober-Desember), data per enam bulan terdiri atas 4 periode (Bulan Januari-April, Bulan Mei-Agustus, Bulan September-Desember), data per dua belas bulan terdiri atas 2 periode (Bulan Januari-Juni dan Bulan Juli-Desember) dan data per dua puluh empat bulan terdiri atas 1 periode.

5.2.1 Pengaruh Nilai *Minimum Support* dan Nilai *Minimum Confidence* terhadap Jumlah *Rule*

Proses pengujian ini bertujuan untuk mengetahui pengaruh nilai *minimum support* dan *minimum confidence* terhadap jumlah *rule* yang dihasilkan. Selain itu, pengujian ini juga bertujuan untuk mengetahui berapa *minimum support* dan nilai *minimum confidence* (nilai kepercayaan) terbaik atau nilai hubungan antar dua *item* atau lebih dari *rule* yang dihasilkan.

Pada pengujian ini, nilai *min_sup* yang digunakan sebesar 10%, 20%, 30% dan 40%. Sedangkan nilai *min_conf* yang digunakan sebesar 40%, 50%, dan 60%. Hasil uji pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan dapat dilihat pada Tabel 5.1.

Tabel 5.1 Tabel Hasil Uji Pengaruh Nilai *Minimum support* dan Nilai *Minimum confidence* terhadap Jumlah *Rule* yang Dihasilkan

Rentang Bulan	Periode	Min Sup (%)	Min Conf (%)	Jumlah <i>Rule</i>
Per Tiga Bulan	1	10	40	14
			50	9
			60	6
		20	40	0
			50	0
			60	0
		30	40	0
			50	0
			60	0
	2	10	40	0
			50	0
			60	0
		20	40	0
			50	0
			60	0
		30	40	0
			50	0
			60	0
	3	10	40	5
			50	1
			60	1
		20	40	0
			50	0
			60	0
		30	40	0
			50	0
			60	0
	4	10	40	10

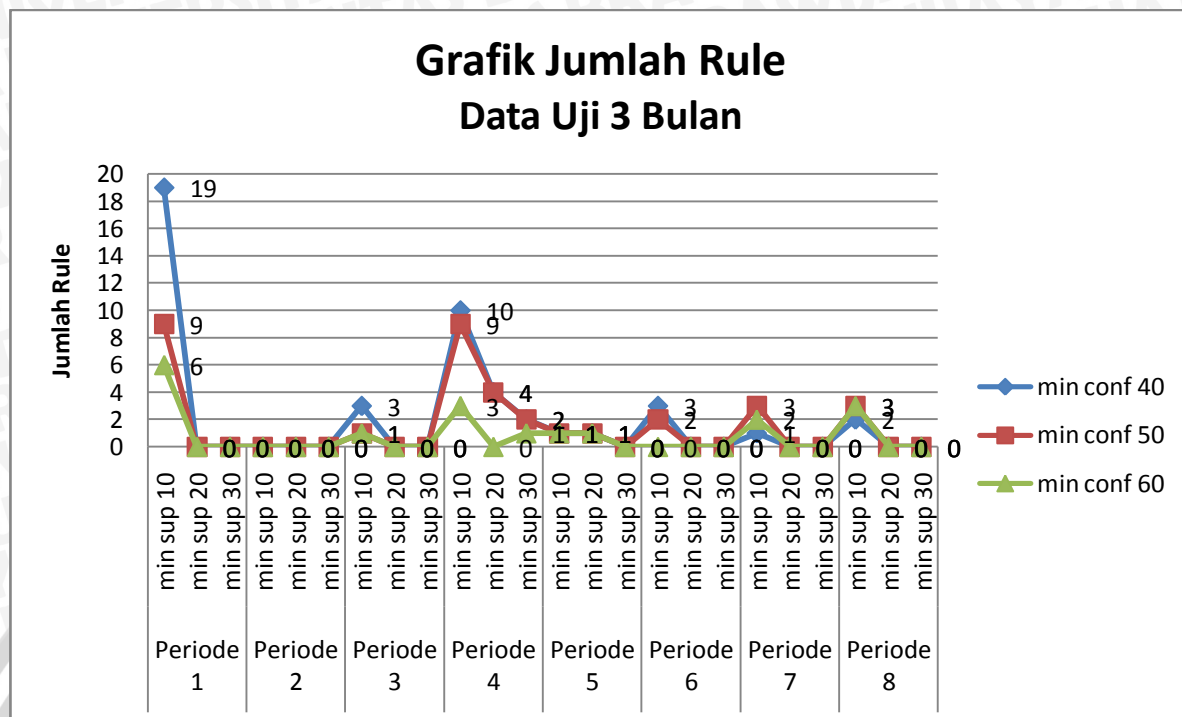
Rentang Bulan	Periode	Min Sup (%)	Min Conf (%)	Jumlah Rule
		20	50	9
			60	3
			40	4
		30	50	4
			60	0
			40	2
	5	10	50	2
			60	1
			40	2
		20	50	1
			60	1
			40	1
	6	30	50	1
			60	0
			40	0
		10	50	0
			60	0
			40	3
	7	10	40	3
			50	3
			60	2
		20	40	0
			50	0
			60	0
		30	40	0
			50	0
			60	0

Rentang Bulan	Periode	Min Sup (%)	Min Conf (%)	Jumlah Rule
		30	50	0
			60	0
			40	0
			50	0
			60	0
			40	3
	8	20	50	3
			60	3
			40	0
			50	0
			60	0
		30	40	0
			50	0
			60	0
Per Enam Bulan	1	10	40	4
			50	3
			60	2
		20	40	0
			50	0
			60	0
		30	40	0
			50	0
			60	0
	2	10	40	20
			50	16
			60	7
		20	40	6
			50	5
			60	1
		30	40	2

Rentang Bulan	Periode	Min Sup (%)	Min Conf (%)	Jumlah Rule
	3	10	50	2
			60	1
			40	13
		20	50	7
			60	4
			40	3
	4	10	50	2
			60	2
			40	0
		20	50	0
			60	0
			40	0
Per Satu Tahun	1	10	40	5
			50	5
			60	4
		20	40	0
			50	0
			60	0
	2	10	40	0
			50	0
			60	0
		20	40	46
			50	34
			60	17
	1	10	40	4
			50	4
			60	4
		20	40	2
			50	2
			60	2
	2	10	40	19

Rentang Bulan	Periode	Min Sup (%)	Min Conf (%)	Jumlah Rule
		20	50	10
			60	6
			40	6
		30	50	5
			60	3
			40	3
		40	50	3
			60	3
			40	124
Per Dua Tahun	1	10	50	84
			60	48
			40	23
		20	50	19
			60	15
			40	10
		30	50	7
			60	7
			40	7

Berdasarkan perhitungan pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan pada Tabel 5.1, dapat ditunjukkan grafik jumlah *rule* berdasarkan pilihan rentang bulan. Grafik jumlah *rule* dengan data uji 3 bulan ditunjukkan pada Gambar 5.1.



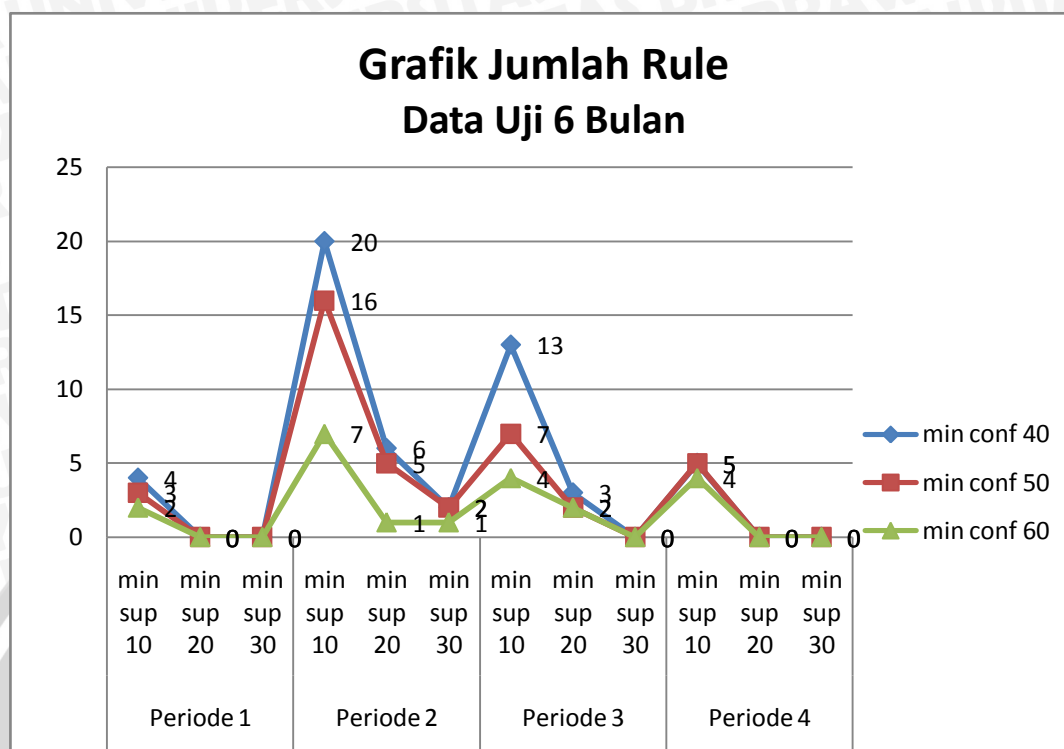
Gambar 5.1 Grafik Jumlah *Rule* Data Uji 3 Bulan

Berdasarkan Gambar 5.1, diketahui nilai rata-rata jumlah *rule* yang dihasilkan pada rentang waktu 3 bulan, yang ditunjukkan pada Tabel 5.2.

Tabel 5.2 Tabel Jumlah *Rule* Rata-rata Data Uji 3 Bulan

Data Uji 3 Bulan	Min Sup 10	Min Sup 20	Min Sup 30
Min Conf 40	5	1	0
Min Conf 50	4	1	0
Min Conf 60	2	0	0

Sedangkan berdasarkan perhitungan pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan pada Tabel 5.1, dapat ditunjukkan grafik jumlah *rule* berdasarkan pilihan rentang bulan dengan data uji 6 bulan pada Gambar 5.2.



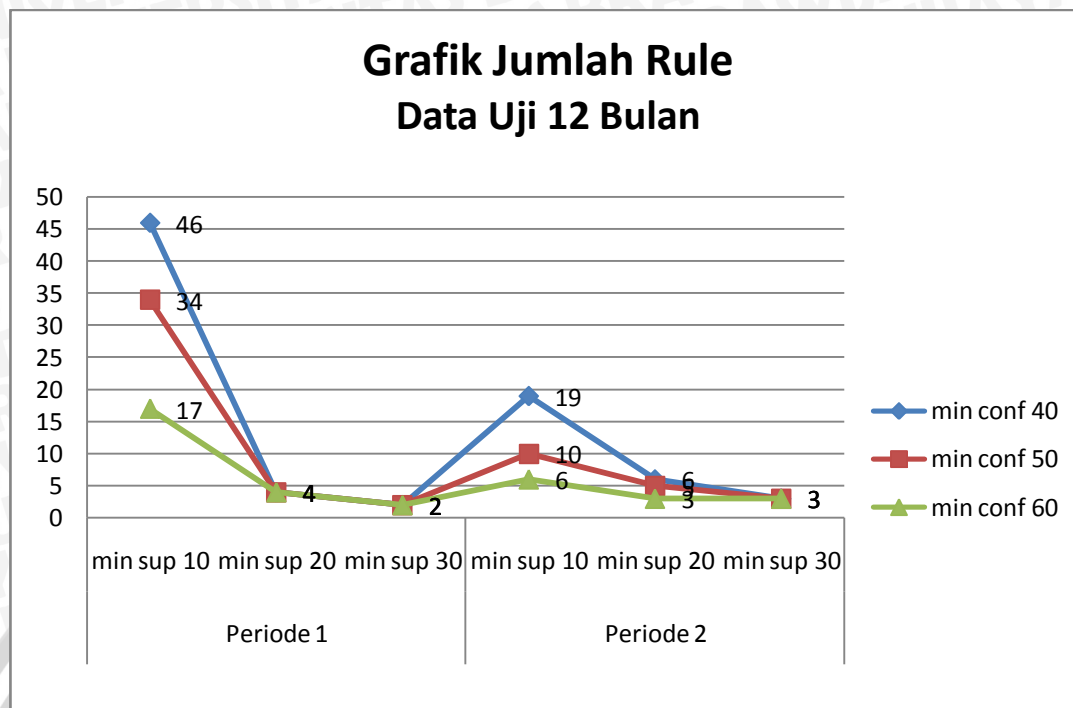
Gambar 5.2 Grafik Jumlah *Rule* Data Uji 6 Bulan

Berdasarkan Gambar 5.2, diketahui nilai rata-rata jumlah *rule* yang dihasilkan pada rentang waktu 6 bulan. Jumlah *rule* rata-rata yang dihasilkan oleh data uji 6 bulan ditunjukkan pada Tabel 5.3.

Tabel 5.3 Tabel Jumlah *Rule* Rata-rata Data Uji 6 Bulan

Data Uji 6 Bulan	Min Sup 10	Min Sup 20	Min Sup 30
Min Conf 40	11	2	1
Min Conf 50	8	2	1
Min Conf 60	4	1	0

berdasarkan perhitungan pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan pada Tabel 5.1, dapat ditunjukkan grafik jumlah *rule* berdasarkan pilihan rentang bulan dengan data uji 12 bulan pada Gambar 5.3.



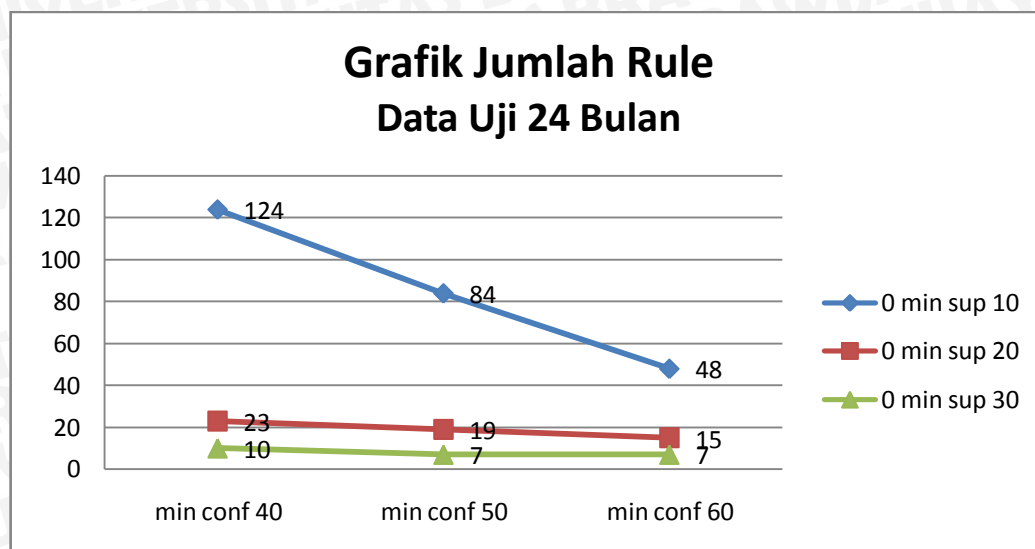
Gambar 5.3 Grafik Jumlah *Rule* Data Uji 12 Bulan

Berdasarkan Gambar 5.3, diketahui nilai rata-rata jumlah *rule* yang dihasilkan pada rentang waktu 12 bulan. Jumlah *rule* rata-rata yang dihasilkan oleh data uji 12 bulan ditunjukkan pada Tabel 5.4.

Tabel 5.4 Tabel Jumlah *Rule* Rata-rata Data Uji 12 Bulan

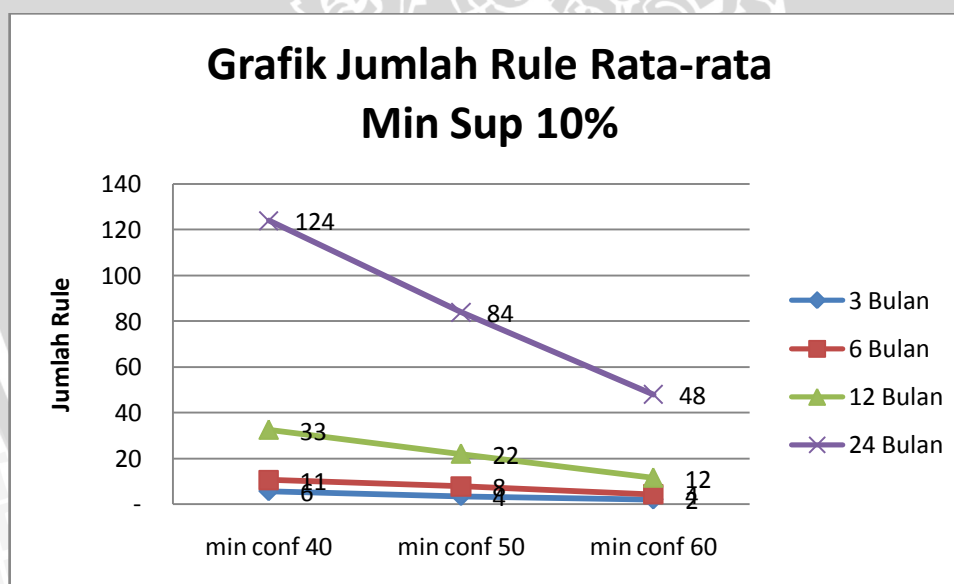
Data Uji 12 Bulan	Min Sup 10	Min Sup 20	Min Sup 30
Min Conf 40	33	5	3
Min Conf 50	22	5	3
Min Conf 60	12	4	3

Sedangkan grafik jumlah *rule* dengan data uji 24 bulan ditunjukkan pada Gambar 5.4



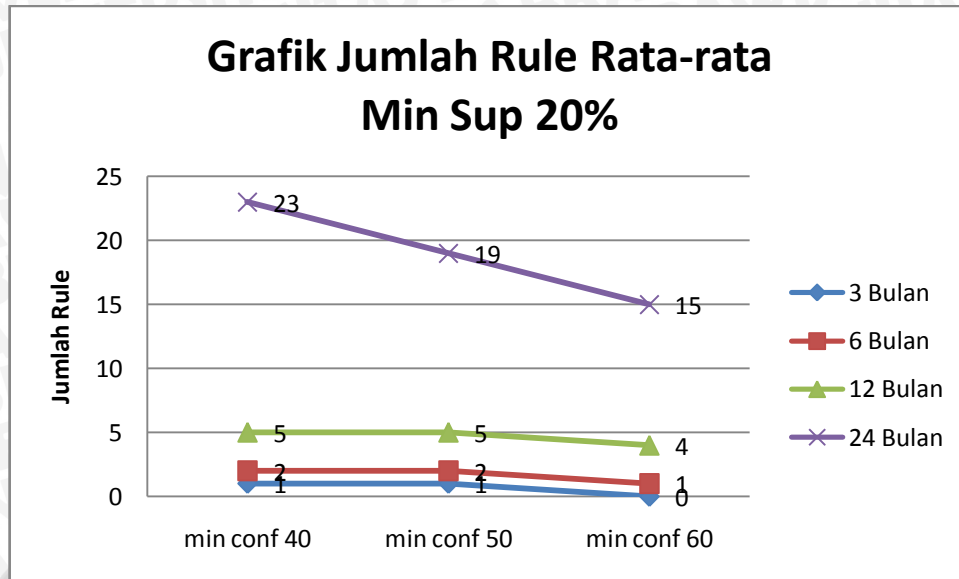
Gambar 5.4 Grafik Jumlah *Rule* Data Uji 24 Bulan

Jumlah *rule* rata-rata yang dihasilkan pada rentang waktu 3 bulan, 6 bulan, 12 bulan dan 24 bulan untuk *minimum confidence* untuk tiap-tiap pengujian berdasarkan *minimum support* 10% ditunjukkan pada Gambar 5.5.



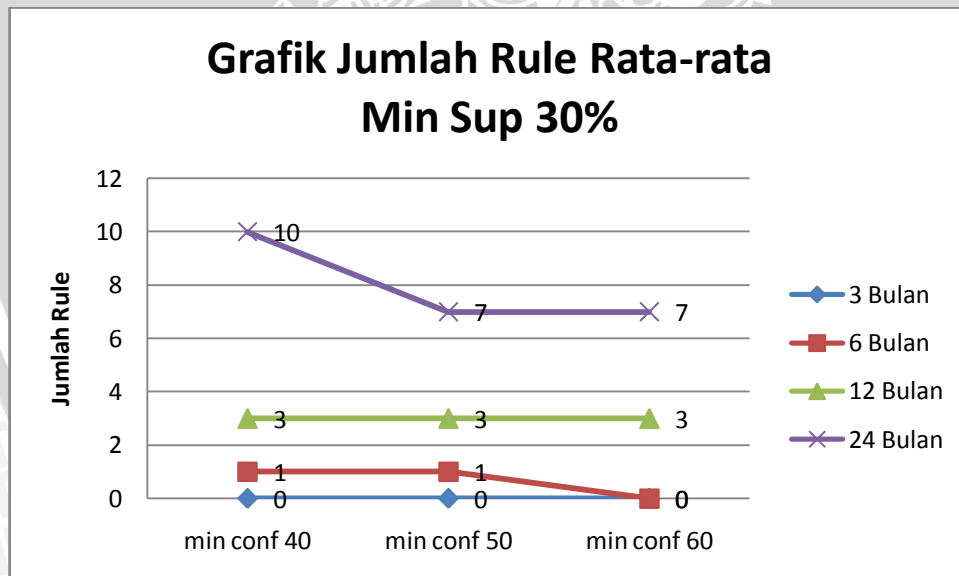
Gambar 5.5 Grafik Jumlah *Rule* Rata-rata *Minimum Support* 10%

Jumlah *rule* rata-rata yang dihasilkan pada rentang waktu 3 bulan, 6 bulan, 12 bulan dan 24 bulan untuk *minimum confidence* untuk tiap-tiap pengujian berdasarkan *minimum support* 20% ditunjukkan pada Gambar 5.6.



Gambar 5.6 Grafik Jumlah Rule Rata-rata *Minimum Support* 20%

Jumlah *rule* rata-rata yang dihasilkan pada rentang waktu 3 bulan, 6 bulan, 12 bulan dan 24 bulan untuk *minimum confidence* untuk tiap-tiap pengujian berdasarkan *minimum support* 30% ditunjukkan pada Gambar 5.7.



Gambar 5.7 Grafik Jumlah Rule Rata-rata *Minimum Support* 30%

Berdasarkan hasil pengujian pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan, didapatkan jumlah *rule* terbanyak pada rentang bulan per 3 bulan periode 1 sejumlah 14 *rule* dengan *minimum support* 10% dan *minimum confidence* 40%. Pada rentang bulan per 6 bulan periode 2 sebanyak 20 *rule* dengan *minimum support* 10% dan *minimum*

confidence 40%. Pada rentang bulan per 12 bulan periode 1 sebanyak 46 *rule* dengan *minimum support* 10% dan *minimum confidence* 40% dan ada rentang bulan per 24 bulan sebanyak 124 *rule*. Namun pada uji coba tersebut juga terdapat hasil pengujian yang tidak menghasilkan *rule* yaitu pada rentang bulan per 3 bulan periode 2 untuk setiap *minimum support* dan *minimum confidence*. Hal ini menjelaskan bahwa pada periode tersebut tidak ada *rule* yang memiliki nilai *support* lebih besar atau sama dengan 10%.

Pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan berdasarkan hasil pengujian adalah berbanding terbalik. Hal ini berarti semakin tinggi nilai *minimum support* dan nilai *minimum confidence* yang digunakan maka akan semakin sedikit jumlah *rule* yang dihasilkan seperti yang digambarkan pada Gambar 5.6.

Berbanding terbaliknya nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* tersebut dikarenakan semakin tinggi nilai *minimum support* yang digunakan berarti semakin tinggi nilai batas *support* yang harus dicapai kandidat *frequent itemset* untuk menjadi *frequent itemset* sehingga jumlah *frequent itemset* semakin sedikit dan bila semakin tinggi nilai *minimum confidence* yang digunakan maka semakin tinggi nilai batas *confidence* yang harus dicapai *frequent itemset*, yang merupakan kandidat *rule* untuk menjadi *rule* sehingga semakin sedikit jumlah *rule* yang dihasilkan.

Selain itu, pada pengujian ini dapat diketahui hubungan *rule* dan jumlah *rule* untuk tiap-tiap hasil uji berdasarkan *minimum confidence*. Seperti pada *rule* yang didapatkan dari percobaan data uji 3 bulan periode 1 dengan *minimum support* 10% dan *minimum confidence* 40%, 50% dan 60%. Hubungan jumlah *rule* dan *rule* yang dihasilkan ditunjukkan pada Tabel 5.5.

Tabel 5.5 Tabel Hubungan *Rule* dan Jumlah *Rule* pada Tiap *Minimum Confidence*

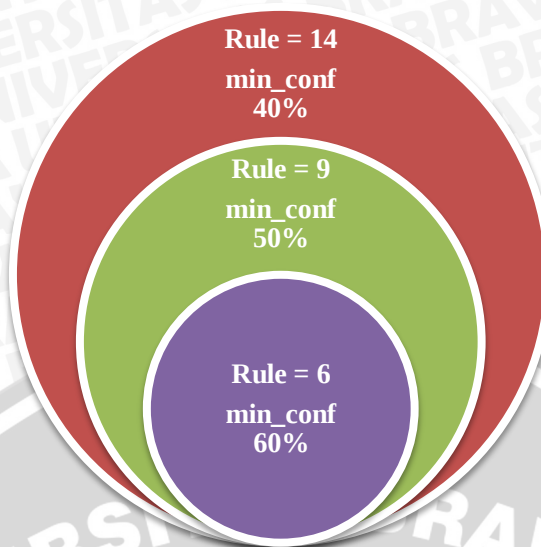
Min Conf	Rule	Conf (%)	Keterangan
40%	(133, 14)	42,857	Jika membeli Flooring Jati, maka juga membeli Lambrisering Jati
	(149 , 152)	100.0	Jika membeli Jendela Kamper, maka juga membeli Pintu Kamper

Min Conf	Rule	Conf (%)	Keterangan
	(159, 162)	50.0	Jika membeli Frame/Kusen Merbau, maka juga membeli Jendela Merbau
	(159, 171)	50.0	Jika membeli Frame/Kusen Merbau, maka juga membeli Pintu Merbau
	(159, 188)	50.0	Jika membeli Frame/Kusen Merbau, maka juga membeli Pintu Lapis
	(162, 171)	100.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Merbau
	(162, 188)	60.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Lapis
	(171, 188)	42,857	Jika membeli Pintu Merbau, maka juga membeli Pintu Lapis
	(114 118)	100.0	Jika membeli Gezebo Bangkirai, maka pembelian selanjutnya akan membeli Kayu Partisi Bangkirai
	(148 162)	42,857	Jika membeli Frame/Kusen Kamper, maka pembelian selanjutnya akan membeli Jendela Merbau
	(148 171)	42,857	Jika membeli Frame/Kusen Kamper, maka pembelian selanjutnya akan membeli Pintu Merbau
	(159,162,171)	50.0	Jika membeli Frame/Kusen Merbau, maka pembelian selanjutnya akan membeli Jendela Merbau dan Pintu Merbau
	(162,171,188)	60.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Merbau dan Pintu Lapis
	(148 162,171)	42,857	Jika membeli Frame/Kusen Kamper, maka pembelian selanjutnya akan membeli Jendela Merbau dan Pintu Merbau
50%	(149 , 152)	100.0	Jika membeli Jendela Kamper, maka juga membeli Pintu Kamper
	(159, 162)	50.0	Jika membeli Frame/Kusen Merbau, maka juga membeli Jendela Merbau
	(159, 171)	50.0	Jika membeli Frame/Kusen Merbau, maka juga membeli Pintu Merbau
	(159, 188)	50.0	Jika membeli Frame/Kusen Merbau, maka juga membeli Pintu Lapis
	(162, 171)	100.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Merbau
	(162, 188)	60.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Lapis
	(114 118)	100.0	Jika membeli Gezebo Bangkirai, maka pembelian selanjutnya akan membeli Kayu Partisi Bangkirai

Min Conf	Rule	Conf (%)	Keterangan
	(159,162,171)	50.0	Jika membeli Frame/Kusen Merbau, maka pembelian selanjutnya akan membeli Jendela Merbau dan Pintu Merbau
	(162,171,188)	60.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Merbau dan Pintu Lapis
60%	(149, 152)	100.0	Jika membeli Jendela Kamper, maka juga membeli Pintu Kamper
	(162, 171)	100.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Merbau
	(162, 188)	60.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Lapis
	(114 118)	100.0	Jika membeli Gezebo Bangkirai, maka pembelian selanjutnya akan membeli Kayu Partisi Bangkirai
	(162,171,188)	60.0	Jika membeli Jendela Merbau, maka juga membeli Pintu Merbau dan Pintu Lapis

Berdasarkan Tabel 5.5 diketahui bahwa terdapat lima *rule* yang dihasilkan dengan nilai *minimum confidence* 60%, *Rule-rule* tersebut juga termasuk dalam *rule* hasil pengujian dengan nilai *minimum confidence* sebesar 50%. Begitupula sembilan *rule* yang dihasilkan dari pengujian dengan nilai *minimum confidence* 50% termasuk dalam *rule* hasil pengujian dengan nilai *minimum confidence* 40%.. Atau dapat dikatakan bahwa *rule* yang dihasilkan pada *minimum confidence* 60% merupakan irisan dari *rule* yang dihasilkan pada *minimum confidence* 50%, dan *rule* yang dihasilkan pada *minimum confidence* 50% merupakan irisan dari *rule* yang dihasilkan pada *minimum confidence* 40%.

Hal ini dikarenakan semakin kecil nilai *mimimum confidence* yang diberikan maka akan semakin banyak varian *rule* yang dihasilkan, sebaliknya akan semakin sedikit varian *rule* yang dihasilkan apabila *minimum confidence* semakin besar. Diagram venn hubungan *rule* dan jumlah *rule* pada tiap *minimum confidence* ditunjukkan pada Gambar 5.8



Gambar 5.8 Diagram Venn Hubungan *Rule* dan Jumlah *Rule* pada Tiap *Minimum Confidence*

Berdasarkan tabel 5.5 juga dapat diketahui bahwa *rule* dengan nilai *min_conf* 60% mempunyai nilai *confidence* yang besar. Dari 5 *rule* yang terbentuk diketahui terdapat 3 *rule* yang memiliki nilai *confidence* hingga 100%. Dapat dikatakan bahwa *rule* dengan nilai *confidence* semakin besar maka *rule* tersebut semakin baik..

Berdasarkan pengujian yang telah dilakukan, dapat disimpulkan bahwa nilai *min_sup* terbaik adalah nilai *min_sup* yang paling kecil. Karena dengan nilai *min_sup* kecil, maka *frequent itemset* yang ditemukan akan semakin banyak. Sedangkan nilai *min_conf* terbaik adalah nilai *min_conf* yang semakin besar. Hal ini dikarenakan nilai *confidence* yang semakin besar menentukan kuat tidaknya hubungan antar dua *item* atau lebih yang saling berasosiasi. Pada pengujian ini dapat disimpulkan bahwa nilai *min_sup* terbaik sebesar 10% dan nilai *min_conf* terbaik sebesar 60%.

5.2.2 Uji Akurasi *Rule*

Pengujian kedua merupakan pengujian mengenai akurasi *rule*. Pengujian ini bertujuan untuk mengetahui prosentase penerapan suatu *rule* pada data uji dari keseluruhan jumlah data transaksi pada data uji.

Pada skenario pengujian ini, *rule* yang digunakan adalah *rule* yang diperoleh dari hasil *training* data dengan rentang bulan 24 bulan dengan nilai

min_sup dan nilai min_conf terbaik yang didapatkan di pengujian I. Prosedur pengujiannya adalah sebagai berikut:

1. Dibangkitkan rule dari data latih 24 bulan dengan nilai min_sup sebesar 10% dan nilai min_conf 60%.
2. Rule yang telah ditemukan diuji dengan data transaksi dengan rentang bulan per tiga bulan, per enam bulan, per dua belas bulan dan per dua puluh empat bulan dan tiap-tiap periodenya.
3. Dihitung nilai prosentase akurasi rule nya dengan Persamaan 2.7, yaitu dengan menghitung nilai *error* dengan menghitung banyaknya transaksi dikurangi dengan nilai rule yang benar dikali dengan 100%.

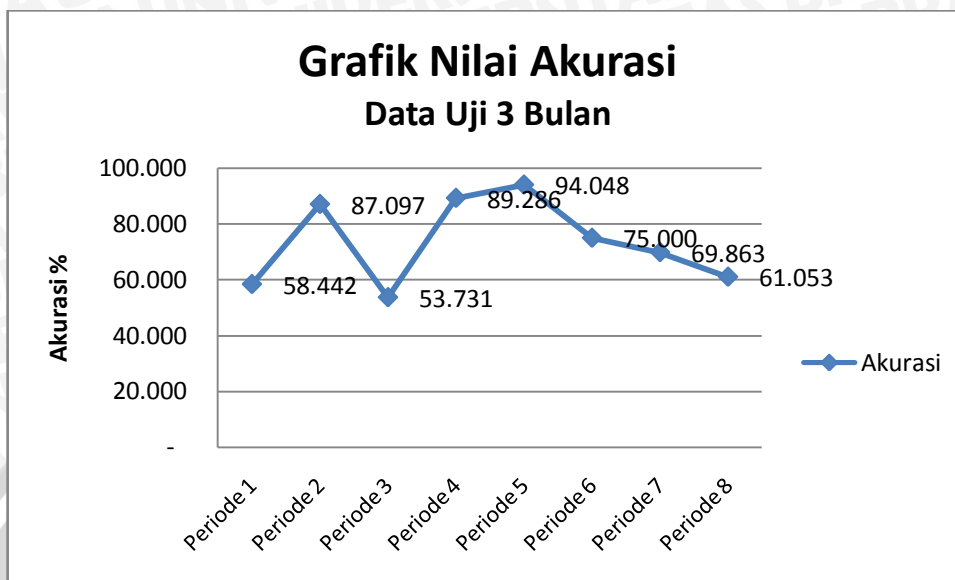
Tabel hasil pengujian akurasi rule pada skenario pengujian dengan data 24 bulan dengan nilai min_sup 10% dan nilai min_conf 60%, yaitu dengan jumlah *rule* yang diuji sebanyak 54 *rule* (Tabel *rule* dapat dilihat di Lampiran 2). Tabel akurasi ditunjukkan pada Tabel 5.6.

Tabel 5.6 Tabel Akurasi

Rentang Bulan	Periode	Nilai Akurasi (%)
3 Bulan	Periode 1	58.442
	Periode 2	87.097
	Periode 3	53.731
	Periode 4	89.286
	Periode 5	94.048
	Periode 6	75.000
	Periode 7	69.863
	Periode 8	61.053
6 Bulan	Periode 1	68.235
	Periode 2	74.396
	Periode 3	76.563
	Periode 4	66.667
12 Bulan	Periode 1	64.721
	Periode 2	66.389
24 Bulan	Periode 1	59.159

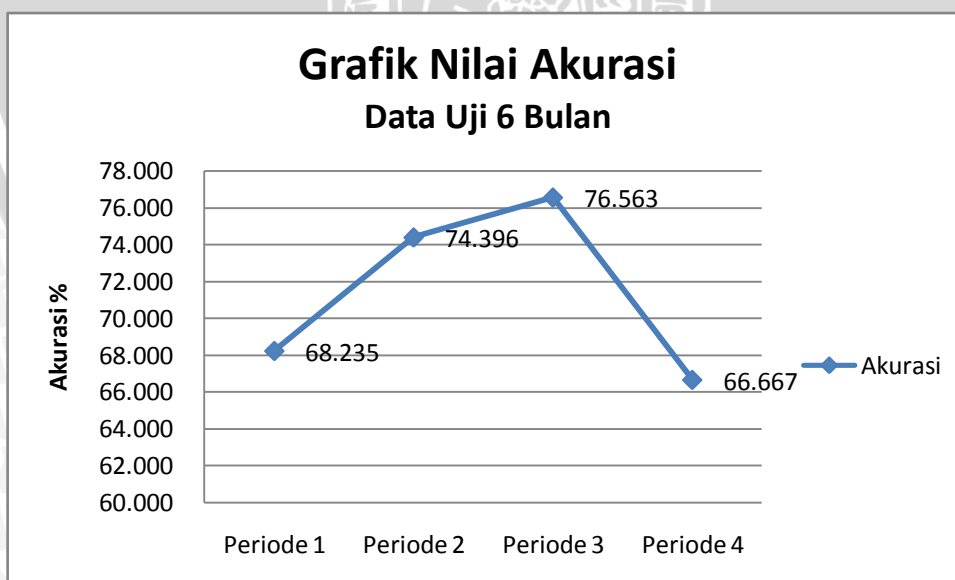
Sebagaimana ditunjukkan pada Tabel 5.5, nilai akurasi *rule* tertinggi untuk data uji 3 bulan sebesar 94.048% pada periode 5 dan nilai akurasi paling rendah

sebesar 53.731% pada periode 3. Grafik nilai akurasi untuk data uji 3 bulan ditunjukkan pada Gambar 5.9.



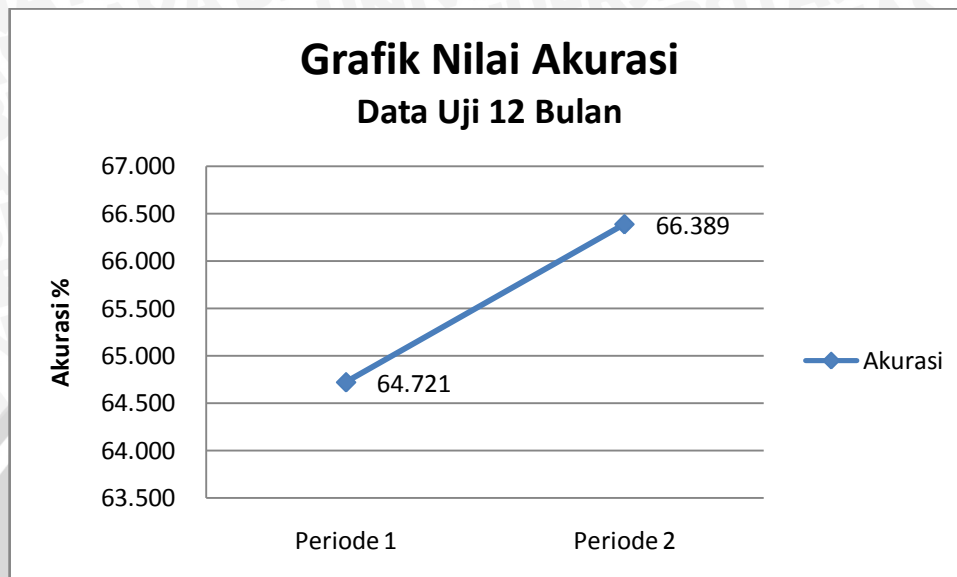
Gambar 5.9 Grafik Akurasi *Rule* Data Uji 3 Bulan

Berdasarkan Tabel 5.5, nilai akurasi *rule* tertinggi untuk data uji 6 bulan adalah sebesar 76.563% pada periode 3 dan nilai akurasi terendah sebesar 66.667% pada periode 4. Grafik nilai akurasi untuk data uji 6 bulan ditunjukkan pada Gambar 5.10.



Gambar 5.10 Grafik Akurasi *Rule* Data Uji 6 Bulan

Berdasarkan Tabel 5.5, nilai akurasi *rule* tertinggi untuk data uji 12 bulan yaitu sebesar 66.389% pada periode 2. Grafik akurasi untuk data uji 12 bulan ditunjukkan pada Gambar 5.11.



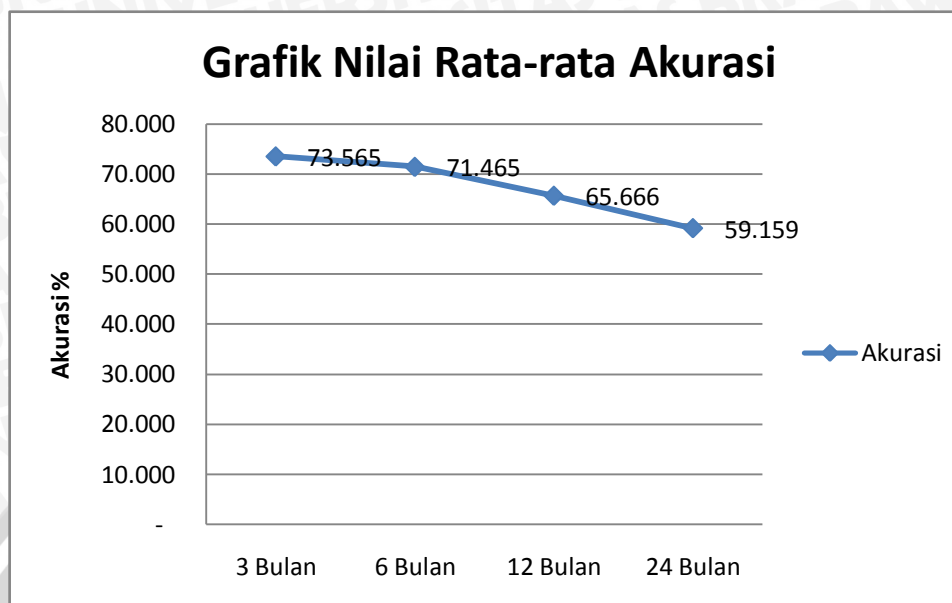
Gambar 5.11 Grafik Akurasi *Rule* Data Uji 12 Bulan

Sebagaimana yang ditunjukkan pada Tabel 5.5, dapat diketahui nilai rata-rata akurasi yang dihasilkan pada rentang waktu 3 bulan, 6 bulan, 12 bulan dan 24 bulan. Nilai rata-rata akurasi ditunjukkan pada Tabel 5.7.

Tabel 5.7 Tabel Rata-rata Akurasi

Bulan	Rata-Rata Akurasi (%)
3 Bulan	73.565
6 Bulan	71.465
12 Bulan	65.666
24 Bulan	59.159

Berdasarkan Tabel 5.6 dapat ditunjukkan grafik rata-rata akurasi *rule* pada Gambar 5.12.



Gambar 5.12 Grafik Rata-rata Akurasi

Berdasarkan Gambar 5.12 diketahui nilai rata-rata akurasi *rule* tertinggi yang dihasilkan adalah sebesar 73.565% pada rentang bulan 3 bulan, dan nilai rata-rata akurasi terendah yang dihasilkan sebesar 59.159% pada rentang bulan 24 bulan. Nilai rata-rata akurasi mengalami penurunan dari rentang 3 bulan hingga 24 bulan.

Selain itu berdasarkan Tabel 5.6, nilai akurasi akan cenderung tinggi ketika data *training* 24 bulan dibandingkan dengan data uji 3 bulan, 6 bulan, dan 12 bulan atau banyak data *training* lebih besar dibandingkan data uji. Hal ini dikarenakan jumlah data yang dibandingkan relatif sedikit, sehingga nilai pembandingnya semakin sedikit dan *rule* yang dibandingkan banyak ditemukan di data transaksi tersebut. Ketika data *training* dibandingkan dengan data uji 24 bulan atau data *training* sama banyak dengan data uji, nilai rata-rata akurasi cenderung kecil.

Sehingga dapat dikatakan bahwa besar nilai rata-rata akurasi berbanding terbalik dengan banyaknya data uji. Semakin sedikit data uji maka nilai rata-rata akurasi akan semakin besar, dan sebaliknya akan semakin kecil nilai rata-rata akurasi apabila data uji semakin besar.

BAB VI PENUTUP

6.1 Kesimpulan

Menjawab pertanyaan di rumusan masalah, maka setelah melakukan penelitian ini dapat disimpulkan bahwa :

1. Algoritma *Generalized Sequential Pattern* (GSP) dapat diterapkan untuk mencari pola keterkaitan antar *item* dalam sebuah transaksi penjualan di UD. Jati Mulya. Untuk mencari pola dengan algoritma ini adalah dengan membuat *sequence* transaksi, mencari *rule* berdasarkan *minimum support* dan *minimum confidence*. Contoh pola yang dihasilkan antara lain:
 - a. Pola ((134,145)|(171)), jika membeli jendela jati dan pintu jati, maka pembelian selanjutnya akan membeli pintu merbau, dengan nilai *confidence rule* sebesar 68% dan *lift* 2,782.
 - b. Pola ((149,152)|(171)), jika membeli jendela kamper dan pintu kamper, maka pembelian selanjutnya akan membeli pintu merbau, dengan nilai *confidence rule* sebesar 64% dan *lift* 3,416.
2. Besar nilai *minimum support* dan *minimum confidence* sangat berpengaruh terhadap jumlah *rule*. Pada uji coba tersebut, didapat bahwa dengan nilai *minimum support* 10% dan *minimum confidence* 40% *rule* yang dihasilkan lebih banyak. Dan didapatkan nilai *minimum support* terbaik sebesar 10% dan nilai *minimum confidence* terbaik sebesar 60%.
3. Nilai rata-rata akurasi terbaik yang didapatkan adalah sebesar 73.57 %. Nilai rata-rata akurasi nilainya akan besar apabila data uji yang digunakan lebih besar dari data *training*, begitupula sebaliknya nilai rata-rata akurasi akan semakin kecil apabila data uji yang digunakan lebih besar atau sama dengan data *training*.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian ini yaitu untuk dapat mengetahui bahwa penggunaan GSP memiliki nilai akurasi dan pembentukan *rule* yang lebih baik maka diharapkan penelitian selanjutnya dapat membandingkan dengan algoritma lain.

DAFTAR PUSTAKA

- [AGR-95] Agrawal, R dan Srikant, R. 1995. *Mining Sequential Pattern*. International Conference of Data Engineering
- [AMI-11] Amiruddin, Purnama, I K. E., & Purnomo, M. H. 2011. *Penerapan Association Rule Mining Pada Data Nomor Unik Pendidik dan Tenaga Kependidikan Untuk Menemukan Pola Sertifikasi Guru*. Institut Teknologi Surabaya, Surabaya.
- [BER-04] Berry, Michael J.A dan Linoff, Gordon S. 2004. *Data mining Techniques for Marketing, Sales, Customer Relationship Management Second Edition*. Wiley Publishing, Inc.
- [BUD-09] Budhi, Gregorius Satia., Handojo, A & Wirawan, Christine Oktavina. 2009. *Algoritma Generalized Sequential Pattern untuk Menggali Data Sekuensial Sirkulasi Buku pada Perpustakaan UK Petra*. Seminar Nasional Aplikasi Teknologi Informasi 2009 (SNATI 2009). Jurusan Teknik Informatika Universitas Kristen Petra, Surabaya.
- [HAN-01] Han, J dan Kamber, M. 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Fransisco.
- [HAN-04] Handojo, A dan Budhi, GS. 2004. *Aplikasi Data Mining untuk Meneliti Asosiasi Pembelian Item Barang di Supermaret dengan Metode Market Basket Analysis*. Seminar Nasional Teknologi Informasi. Jurusan Teknik Informatika Universitas Kristen Petra, Surabaya.
- [KUS-07] Kusrini. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Penerbit ANDI, Yogyakarta.
- [LAR-05] Larose, Daniel T. Discovering. 2005. *Knowledge in Data: An Introduction to Data Mining*. John Wiley & Sons. Inc., New Jersey.

- [MOE-02] Moertini, Veronika S. 2002. *Data Mining sebagai Solusi Bisnis*. Integral Majalah Ilmiah Matematika dan Ilmu Pengetahuan Alam, Volume 7 No.1. Jurusan Ilmu Komputer Universitas Katolik Parahyangan, Bandung.
- [PEI-04] Pei, J dan Han, J.2004. *Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach*. dalam IEEE Transaction on Knowledge and Data Engineering, Vol. 16 No. 11. Hal. 1424-1440
- [PRA-07] Pramudiono, Iko. 2003. Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data. <http://www.ilmukomputer.com>, tanggal akses 13 Maret 2013.
- [RUL-08] Ruldeviyani, Yova dan Muhammad Fahrian. 2008. *Implementasi Algoritma – Algoritma Association Rules sebagai Bagian dari Pengembangan Data Mining Algorithms Collection*. Fakultas Ilmu Komputer Universitas Indonesia. Jakarta.
- [SAN-07] Santosa, Budi. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Graha Ilmu, Yogyakarta.
- [SRI-96] Srikant, R dan Agrawal, R. 1996. *Mining Sequential Pattern: Generalizations and Performance Improvement*. International Conference on Extending Database Technology.
- [YUL-02] Yulita, Marsela dan Veronica S. Moertini. 2002. *Analisis Keranjang Pasar dengan Algoritma Hash-Based pada Transaksi Penjualan di Apotek*. Integral Majalah Ilmiah Matematika dan Ilmu Pengetahuan Alam, Volume 9 No. 3. Jurusan Ilmu Komputer Universitas Katolik Parahyangan. Bandung.
- [ZAK-97] Zaki, Mohammed J. 1997. *Fast Mining of Sequential Pattern in Very Large Databases*. The University of Rochester Computer Science Department Rochester, New York 14627. Technical Report 668.

LAMPIRAN

Lampiran 1 Tabel Sekuen Pelanggan Data Uji 24 Bulan

Seq_ID	ID_Cust	Sequence
1	1001	[[164]]
2	1002	[[162, 171], [159]]
3	1003	[[134, 137, 145, 152], [113, 134, 159], [116, 126, 188], [118, 120, 126, 145]]
4	1004	[[125, 161], [153, 153], [134, 140], [171], [159], [134]]
5	1005	[[111, 111, 115], [171]]
6	1006	[[159, 171], [112], [131, 188], [134, 145, 188], [164, 171], [162, 171], [115, 129, 140, 147], [162, 171], [159, 164], [159], [162, 171], [154, 166, 171], [159], [159], [159, 162, 171], [159], [159], [121, 162], [154, 159, 162], [121, 154, 162], [121, 154, 159, 171]]
7	1007	[[117, 143], [159, 162, 171, 188], [111, 126, 152], [115, 128], [101], [116, 116, 126, 126], [113, 129], [113], [113, 129], [113], [113]]
8	1008	[[157, 164], [121]]
9	1009	[[113, 116, 126], [159, 171]]
10	1010	[[118, 137, 145], [114], [134], [105, 108, 185], [152, 159, 188], [134, 171], [133, 149, 158], [164]]
11	1011	[[121, 132, 155], [111, 132, 133], [159, 171], [159, 162, 171, 188]]
12	1012	[[116, 126, 171], [171], [154, 156, 163], [137, 145], [121, 161], [154, 159], [159], [157], [162, 171], [153, 159], [153, 159, 164], [162, 171], [154, 159, 168, 171], [110, 123], [159, 164], [164], [110], [154, 157, 164], [130, 139], [154, 169], [154, 157], [154, 169], [171], [121, 154, 159, 162, 171], [153, 158], [121, 158]]
13	1013	[[111, 117], [133, 159], [162, 171], [161, 166, 174], [174], [174], [108], [154, 159, 171], [159], [121], [108, 162], [171], [134, 145], [134], [159, 171]]

Seq_ID	ID_Cust	Sequence
14	1014	[[159], [159]]
15	1015	[[111, 133], [162, 171], [171], [162, 171], [162, 171], [162, 171], [162]]
16	1016	[[150], [164, 183], [164, 183], [164], [150, 151], [164]]
17	1017	[[128, 164, 167], [134, 142, 182], [110, 118], [110, 118, 127], [110, 118, 127, 132], [118, 132]]
18	1018	[[128], [115, 133, 160], [162, 171]]
19	1019	[[128], [159, 162, 171], [164], [121], [162, 171], [162, 164, 171]]
20	1020	[[114, 127], [166], [154], [187], [159, 162, 171], [118], [128], [113, 117, 133, 140, 148], [117, 149, 152], [113, 134, 149, 159], [134, 149, 171], [152, 159, 185], [110, 133, 156], [156, 175], [113, 118], [134, 137, 145], [104, 118, 128], [159], [118, 126, 164, 171], [159, 164, 171], [159, 162, 171, 189], [121], [134], [171], [118, 128], [162, 171], [118, 128, 188], [121, 159], [164], [164], [188], [159], [156, 159, 164], [101, 101, 104, 104, 121, 161, 164, 164, 171], [154], [104, 107, 128, 188], [156], [107, 188]]
21	1021	[[113, 134], [145], [188], [118, 164], [133, 158], [104, 126, 128], [159, 159], [111, 115, 123, 129, 150, 158], [162, 171], [162, 171], [159]]
22	1022	[[121, 154, 164], [166], [129], [129, 132], [171], [115]]
23	1023	[[117, 133, 140], [126], [113, 117, 120, 133], [159], [171, 188]]
24	1024	[[173], [162, 171]]
25	1025	[[105, 108], [111, 149, 164], [134, 149, 171], [162, 171], [137, 145], [130, 137, 145]]
26	1026	[[149, 152, 169], [159], [159], [164], [180, 183]]
27	1027	[[171], [108], [162, 164, 171], [159, 171], [109], [119, 122], [108], [164]]
28	1028	[[117, 133, 140], [115, 134, 188], [111, 138, 171], [137, 138, 145], [149, 152], [159, 164], [113, 167], [148, 159], [154], [152, 188], [152], [152, 188], [187], [162], [188], [187], [120, 161], [187], [186], [152, 188], [187]]
29	1029	[[110, 121, 152], [164], [177], [178]]

Seq_ID	ID_Cust	Sequence
30	1030	[[152, 188], [154], [115], [116], [164]]
31	1031	[[117, 163], [113]]
32	1032	[[171], [130, 145], [120, 129], [121, 171], [171]]
33	1033	[[157, 176], [169, 176], [113], [161, 170, 174]]
34	1034	[[134, 137, 145], [128, 134, 141, 188], [190], [108]]
35	1035	[[123, 161], [159, 162, 171, 188], [159], [111, 171], [113, 134], [126, 138], [133, 149, 160], [154, 159, 171], [150], [107, 121, 158, 188], [159], [137, 145], [149, 152], [161, 174], [173]]
36	1036	[[152, 188], [121, 164, 169], [162, 171]]
37	1037	[[171, 188], [154, 166], [159, 171], [106, 108, 118], [115], [164]]
38	1038	[[109, 186], [185]]
39	1039	[[159], [154, 161, 166], [153, 171], [121, 153, 153, 154, 161, 171]]
40	1040	[[149, 152], [116, 126, 133], [164, 171], [112, 176], [162, 171], [171], [162, 171, 190], [164]]
41	1041	[[116, 126], [159], [181], [121, 153], [107], [154]]
42	1042	[[104], [133, 158], [148, 149, 152], [164, 174], [152, 187, 188, 1448]]
43	1043	[[104, 151], [148, 150, 171], [145, 160, 171], [162, 168, 171, 188], [148, 152], [159, 171], [159, 171], [159, 188], [111, 140, 152, 163, 188], [114], [113], [113], [113], [113]]
44	1044	[[115, 126, 133], [133, 145], [111, 159, 188], [145, 188], [113, 126], [149, 159], [153], [153]]
45	1045	[[133], [121, 158], [139, 141], [172], [166], [121]]
46	1046	[[115, 152, 188], [106, 108, 189], [138], [164], [138], [164], [164], [104]]
47	1047	[[145], [138]]
48	1048	[[133, 163], [162, 171, 188]]

Seq_ID	ID_Cust	Sequence
49	1049	[[138], [150]]
50	1050	[[137, 145], [157, 164], [159, 171], [188], [187], [162, 171]]
51	1051	[[154, 159, 171], [171], [164], [166]]
52	1052	[[148], [134]]
53	1053	[[113, 120, 187]]
54	1054	[[164], [164], [137], [159], [134, 137], [164], [162, 171]]
55	1055	[[159, 171]]
56	1056	[[118, 123], [115, 129], [115, 133, 159], [107, 126, 171], [107, 110, 126, 137], [133, 156], [111, 113, 188], [150], [178], [121, 185], [115, 115], [134], [117, 133], [145], [118], [118], [134], [137, 145], [115], [134], [137, 145], [129, 140], [129], [118]]
57	1057	[[104], [109]]
58	1058	[[171]]
59	1059	[[115, 159], [133], [164], [162, 171], [111, 121, 124, 126, 163], [111, 117, 121, 124, 127], [113, 126, 128, 187], [157, 162, 164], [149, 153, 188], [118], [118], [159, 171], [124, 159, 171]]
60	1060	[[133, 138]]
61	1061	[[145, 171], [134], [156]]
62	1062	[[159, 162, 171], [188]]
63	1063	[[164]]
64	1064	[[126, 159, 188], [148, 171], [111, 115], [104], [111, 112, 114, 118, 125], [111, 114, 118], [133, 147, 156], [138, 168], [133, 156], [117, 156, 163, 175], [121], [134, 159], [175], [175]]
65	1065	[[161, 166], [164], [148, 152]]
66	1066	[[113, 126], [131, 134], [121], [121], [121]]

Seq_ID	ID_Cust	Sequence
67	1067	[[162, 171], [164], [164], [181], [181], [164]]
68	1068	[[134, 158], [121, 155], [158, 161, 170, 174], [159, 171, 188], [104, 113, 124, 126], [121], [159, 171], [179], [132]]
69	1069	[[138]]
70	1070	[[104, 111, 117], [111], [121, 155], [145, 188], [161, 174], [115, 129, 144]]
71	1071	[[118], [153, 162, 171, 188], [111, 117], [118, 133]]
72	1072	[[101, 104, 164]]
73	1073	[[138], [156], [159, 171]]
74	1074	[[126, 134, 159], [153, 159, 162, 171], [152, 153], [137, 145, 185]]
75	1075	[[159], [136, 171], [113, 116, 126]]
76	1076	[[171]]
77	1077	[[113, 185], [148], [154, 159, 171], [159], [162], [164], [156], [121, 126], [121], [164], [162, 171]]
78	1078	[[111, 132, 133], [112, 115], [175], [175], [164, 171], [108, 186], [164], [129, 158, 159, 162, 164, 188]]
79	1079	[[108, 185], [154, 155], [101, 107], [164], [159, 162, 171]]
80	1080	[[148, 152], [117, 133], [160], [106, 111, 140, 185, 189], [133]]
81	1081	[[117, 133], [134], [125, 128, 132], [121, 149], [138, 148, 159], [121, 161, 174], [134, 145], [148], [121], [137, 145], [159], [121], [121], [159, 162, 171], [171, 188], [149, 152], [159, 171], [136, 171], [159], [162, 171], [134], [137, 145], [121], [117, 142, 187], [148], [117, 133, 142, 144], [145], [134], [148], [121], [137, 145], [121], [137, 145]]
82	1082	[[115, 126, 134], [145, 152], [162, 171], [174], [171], [159], [166], [158, 164], [161, 164], [171], [159], [174], [149, 152], [188], [166], [158, 164], [157, 161, 168]]
83	1083	[[150, 182, 187], [168], [168], [168], [164, 164], [151, 181], [164], [164, 188], [111, 171, 182], [164, 168], [105, 168], [164], [164, 168], [168], [164, 168], [164], [164, 168], [164], [158, 168], [164], [162], [164], [164], [164], [164], [164], [164], [164], [164], [168]]

Seq_ID	ID_Cust	Sequence
84	1084	[[113], [115, 134], [149, 185], [171, 184], [170, 182], [174, 180, 183], [188], [121, 154, 164], [159], [159], [153, 159], [171, 184], [152, 162, 171, 184], [170, 181], [171, 188]]
85	1085	[[159, 183], [126, 133], [126, 183], [113], [159, 162, 171], [154, 159, 171], [113, 126, 127], [159, 171], [154, 159, 171]]
86	1086	[[159, 162, 164, 171, 174], [159], [159]]
87	1087	[[142, 145], [142, 145], [145], [137, 145]]
88	1088	[[164, 168]]
89	1089	[[164], [114, 127], [168, 181], [164]]
90	1090	[[171]]
91	1091	[[159, 162], [148]]
92	1092	[[187], [148], [162, 171, 188], [127], [187, 188], [115, 159], [121, 140], [159, 188], [162, 171], [162, 171], [164], [164], [148], [148, 148], [188], [162, 171, 188], [148, 150], [148], [162, 171], [154, 188], [115, 120], [164], [188], [188], [121, 161], [182], [185, 186, 189]]
93	1093	[[145], [162, 171], [148, 149, 152], [148], [145], [121, 125, 161, 164]]
94	1094	[[164]]
95	1095	[[166, 170, 174], [170, 174], [164]]
96	1096	[[118, 148, 159], [151, 168], [168], [164], [164], [108, 141], [108]]
97	1097	[[149, 152], [133, 162, 171], [121, 159, 162, 171], [148], [121], [134], [147], [159], [159], [162, 171, 188], [111, 118, 126, 145], [159, 171], [124, 125], [103, 130, 153, 188], [162, 171], [134], [156, 159, 171, 174], [111, 171]]
98	1098	[[126, 131, 145], [136, 145, 175], [164]]
99	1099	[[133, 156], [118, 138, 148, 159], [112, 118, 157, 164, 166], [104, 107, 188], [171], [171]]
100	1100	[[112, 115], [139, 141, 154, 159]]

Seq_ID	ID_Cust	Sequence
101	1101	[[132, 152], [101, 108], [121, 161], [149, 152], [116, 126, 188], [159, 171], [118], [116, 126], [116, 116, 126, 126], [113, 185], [116, 126, 188], [131, 140, 145]]
102	1102	[[111, 138], [126, 171], [159, 162], [166], [166], [113, 166], [121], [162, 171], [137, 145], [159], [127], [118, 125], [159], [121, 161, 171, 174], [159], [162, 171], [162, 171], [120, 161], [164], [157, 162, 164], [168, 188], [104], [164], [104], [149, 152]]



Lampiran 2 Tabel *Rule* Data Uji 24 Bulan, *Minimum Support* 10% dan *Minimum Confidence* 60%

No.	Rule	Keterangan
1	(154, 159)	Jika membeli Architrave Merbau, maka juga membeli Frame/Kusen Merbau
2	(154, 171)	Jika membeli Architrave Merbau, maka juga membeli Pintu Merbau
3	(159, 171)	Jika membeli Frame/Kusen Merbau, maka juga membeli Pintu Merbau
4	(162, 171)	Jika membeli Jendela Merbau, maka juga membeli Pintu Merbau
5	(104 188)	Jika membeli Kayu Partisi Akasia, maka pembelian selanjutnya akan membeli Pintu Lapis
6	(111 159)	Jika membeli Decking Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
7	(111 162)	Jika membeli Decking Bangkirai, maka pembelian selanjutnya akan membeli Jendela Merbau
8	(111 171)	Jika membeli Decking Bangkirai, maka pembelian selanjutnya akan membeli Pintu Merbau
9	(111 188)	Jika membeli Decking Bangkirai, maka pembelian selanjutnya akan membeli Pintu Lapis
10	(113 159)	Jika membeli Frame/Kusen Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
11	(113 171)	Jika membeli Frame/Kusen Bangkirai, maka pembelian selanjutnya akan membeli Pintu Merbau
12	(115 159)	Jika membeli Hand Railling Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
13	(115 171)	Jika membeli Hand Railling Bangkirai, maka pembelian selanjutnya akan membeli Pintu Merbau
14	(115 188)	Jika membeli Hand Railling Bangkirai, maka pembelian selanjutnya akan membeli Pintu Lapis
15	(117 133)	Jika membeli Joist Bangkirai, maka pembelian selanjutnya akan membeli Flooring Jati
16	(117 159)	Jika membeli Joist Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
17	(117 171)	Jika membeli Joist Bangkirai, maka pembelian selanjutnya akan membeli Pintu Merbau
18	(117 188)	Jika membeli Joist Bangkirai, maka pembelian selanjutnya akan membeli Pintu Lapis
19	(118 159)	Jika membeli Kayu Partisi Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau

No.	Rule	Keterangan
20	(118 171)	Jika membeli Kayu Partisi Bangkirai, maka pembelian selanjutnya akan membeli Pintu Merbau
21	(121 159)	Jika membeli List/Banpile Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
22	(121 171)	Jika membeli List/Banpile Bangkirai, maka pembelian selanjutnya akan membeli Pintu Merbau
23	(126 159)	Jika membeli Pintu Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
24	(126 171)	Jika membeli Pintu Bangkirai, maka pembelian selanjutnya akan membeli Pintu Merbau
25	(133 159)	Jika membeli Flooring Jati, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
26	(133 171)	Jika membeli Flooring Jati, maka pembelian selanjutnya akan membeli Pintu Merbau
27	(134 159)	Jika membeli Frame/Kusen Jati, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
28	(134 171)	Jika membeli Frame/Kusen Jati, maka pembelian selanjutnya akan membeli Pintu Merbau
29	(137 159)	Jika membeli Jendela Jati, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
30	(137 171)	Jika membeli Jendela Jati, maka pembelian selanjutnya akan membeli Pintu Merbau
31	(148 171)	Jika membeli Frame/Kusen Kamper, maka pembelian selanjutnya akan membeli Pintu Merbau
32	(149 159)	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
33	(149 162)	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan membeli Jendela Merbau
34	(149 164)	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan membeli Kayu Partisi Merbau
35	(149 171)	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan membeli Pintu Merbau
36	(149 188)	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan membeli Pintu Lapis
37	(152 188)	Jika membeli Pintu Kamper, maka pembelian selanjutnya akan membeli Pintu Lapis
38	(154 159)	Jika membeli Architrave Merbau, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
39	(154 171)	Jika membeli Architrave Merbau, maka pembelian selanjutnya akan membeli Pintu Merbau
40	(159 171)	Jika membeli Frame/Kusen Merbau, maka pembelian selanjutnya akan membeli Pintu Merbau

No.	Rule	Keterangan
41	(161 171)	Jika membeli Hand Railling Merbau, maka pembelian selanjutnya akan membeli Pintu Merbau
42	(162 171)	Jika membeli Jendela Merbau, maka pembelian selanjutnya akan membeli Pintu Merbau
43	(166 171)	Jika membeli Lambrisering Merbau, maka pembelian selanjutnya akan membeli Pintu Merbau
44	(117 159 171)	Jika membeli Joist Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau dan pembelian selanjutnya akan membeli Pintu Merbau
45	(121 159 171)	Jika membeli List/Banpile Bangkirai, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau dan pembelian selanjutnya akan membeli Pintu Merbau
46	(134 159 171)	Jika membeli Frame/Kusen Jati, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau dan pembelian selanjutnya akan membeli Pintu Merbau
47	(137 159 171)	Jika membeli Jendela Jati, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau dan pembelian selanjutnya akan membeli Pintu Merbau
48	(149 159 171)	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau dan pembelian selanjutnya akan membeli Pintu Merbau
49	(149 162 171)	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan membeli Jendela Merbau dan pembelian selanjutnya akan membeli Pintu Merbau
50	(154 159 171)	Jika membeli Architrave Merbau, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau dan pembelian selanjutnya akan membeli Pintu Merbau
51	(137,145 159)	Jika membeli Jendela Jati dan Pintu Jati, maka pembelian selanjutnya akan membeli Frame/Kusen Merbau
52	(137,145 171)	Jika membeli Jendela Jati dan Pintu Jati, maka pembelian selanjutnya akan membeli Pintu Merbau
53	(149,152 171)	Jika membeli Jendela Kamper dan Pintu Kamper, maka pembelian selanjutnya akan membeli Pintu Merbau
54	(149 162,171)	Jika membeli Jendela Kamper, maka pembelian selanjutnya akan membeli Jendela Merbau dan Pintu Merbau

