

LEMBAR PERSETUJUAN
PENERAPAN METODE ASSOCIATION RULE DENGAN ALGORITMA
FP-GROWTH UNTUK MENGETAHUI KARAKTERISTIK PEMBELI
MOBIL

SKRIPSI



Disusun oleh:

Kadek Andriana Putra

NIM.0710963031

Telah diperiksa dan disetujui oleh :

Pembimbing I,

Pembimbing II,

Drs. Marji, M.T.

NIP. 197508191999031001

Lailil Muflikhah, S.Kom, M.Sc

NIP. 197411132005012001

LEMBAR PENGESAHAN
PENERAPAN METODE ASSOCIATION RULE DENGAN ALGORITMA
FP-GROWTH UNTUK MENGETAHUI KARAKTERISTIK PEMBELI
MOBIL

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh:

Kadek Andriana Putra

NIM.0710963031

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 28 Desember 2012

Penguji I

Penguji II

Drs. Achmad Ridok, M.Kom
NIP. 196808251994031002

Rekyan Regasari MP, S.T., M.T.
NIK. 770414 06120253

Penguji III

Budi Darma Setiawan, S.Kom., M.Cs
NIK. 84101506110090

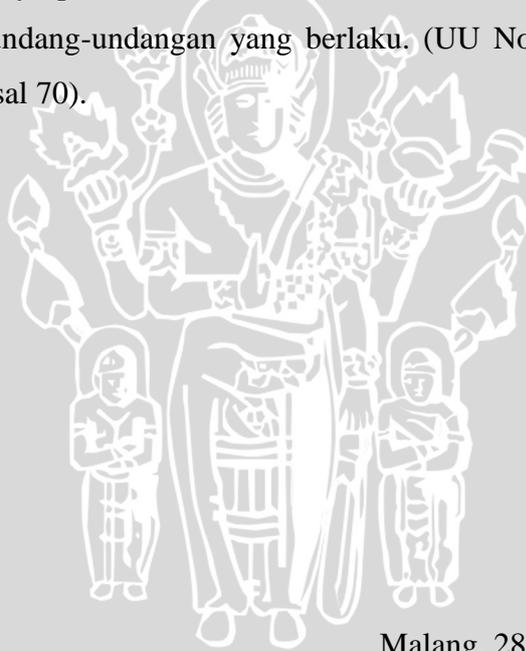
Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, M.T.
NIP. 196708011992031001

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, 28 Desember 2012

Mahasiswa,

Kadek Andriana Putra
NIM. 0710963031

KATA PENGANTAR

Puji Syukur ke hadirat Tuhan atas berkat, rahmat dan penyertaan-Nya, sehingga penulis mampu menyelesaikan skripsi yang berjudul “**Penerapan Metode Association Rule Dengan Algoritma FP-Growth Untuk Mengetahui Karakteristik Pembeli Mobil**”.

Penulis mengucapkan terima kasih pada semua pihak yang telah membantu penyelesaian skripsi ini, terutama kepada:

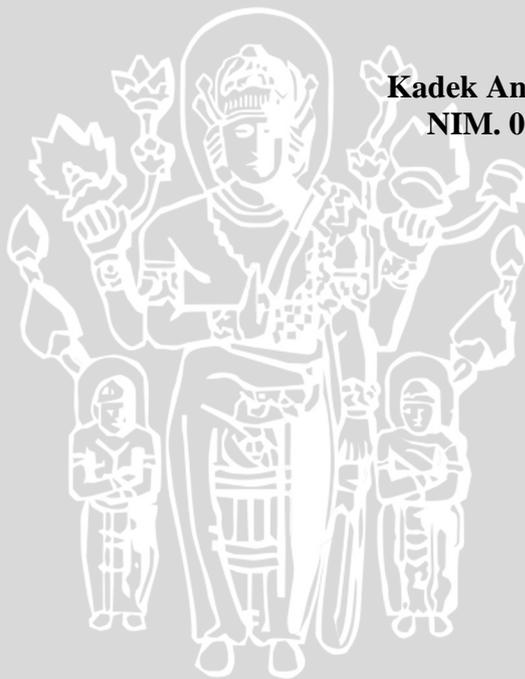
1. Drs. Marji, MT., selaku dosen pembimbing I dan Ketua Program Studi Ilmu Komputer Jurusan Matematika Universitas Brawijaya, atas kebersediaan meluangkan waktu untuk memberikan bimbingan, pengarahan, saran, dan dukungan selama penyusunan skripsi ini.
2. Lailil Muflikhah, S.Kom, M.Sc selaku dosen pembimbing II, atas kebersediaan meluangkan waktu untuk memberikan bimbingan, pengarahan, saran, dan dukungan selama penyusunan skripsi ini.
3. Ir. Sutrisno, MT, selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Bapak dan Ibu Dosen Ilmu Komputer atas didikan dan bimbingan selama kuliah.
5. Kedua orangtua penulis yang tersayang dan tercinta, atas kesabaran dan keikhlasan yang selalu mengiringi langkah hidupku, dukungan berupa materi, nasehat, motivasi, dan kepercayaan untukku menuntut ilmu disini.
6. Saudara-saudaraku dan semua keluarga atas semua kasih, doa dan dukungan selama hidup.
7. Sahabat-sahabatku yang selalu memberikan dukungan, motivasi dan hiburan selama skripsi.
8. Teman-teman Prodi Ilmu Komputer, terutama Ilkom B 2007 atas kerjasama dan persahabatannya. Marisa Hamnon, S.Kom selaku penasehat dan pendamping skripsi ini, yang menjadi ‘guru’ selama ini. Teman-teman yang telah meluangkan waktu untuk diskusi bersama.

9. Dan semua pihak yang terlibat baik secara langsung maupun tidak langsung yang tidak dapat disebutkan satu per satu. Terima kasih atas semua bantuan yang telah diberikan.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu penulis mengharap kritik dan saran yang membangun dari pembaca dan semoga skripsi ini dapat bermanfaat bagi penulis dan pembaca.

Malang, 28 Desember 2012

Kadek Andriana Putra
NIM. 071093031



ABSTRAK

Persaingan didalam bisnis membutuhkan suatu strategi agar dapat meningkatkan penjualan. Strategi bisnis dapat dilakukan dengan segmentasi pasar berdasarkan aspek demografis dengan menganalisa karakteristik pembeli mobil berdasarkan parameter jenis kelamin dan rentang umur pada data transaksi penjualan sehingga diperoleh informasi sesuai dengan permintaan pasar. Data transaksi penjualan merupakan data yang berskala besar sehingga dibutuhkan teknik khusus yang disebut penggalian data. Penggalian data cukup efektif untuk mengetahui adanya serangkaian pola informasi dari data dalam jumlah besar. Dalam penelitian ini, penggalian dilakukan terhadap data transaksi penjualan *dealer* mobil. Penggalian data dilakukan untuk mendapatkan pola hubungan antara jenis kelamin dan rentang umur terhadap jenis mobil yang dibeli. Sistem akan melakukan pencarian aturan asosiasi dengan tahap awal adalah mencari frequent itemset dengan menerapkan algoritma *fp-growth*. Kemudian dari *frequent itemset* tersebut akan dilakukan pencarian pola asosiasi dengan menggunakan teknik *association rule*. Pada tahap akhir, sistem akan menghitung kekuatan pola asosiasi yang terbentuk dengan menggunakan *lift ratio*. Pada penelitian ini didapatkan hasil pengujian nilai kekuatan (*lift ratio*) *rule*, jumlah *rule* yang dihasilkan tiap *dealer* adalah *dealer* Suzuki 3 *rule*, *dealer* Honda 5 *rule*, *dealer* Toyota 4 *rule*, dan *dealer* Daihatsu 4 *rule*. Dan didapatkan nilai rata – rata *lift ratio* dari *rule* yang dihasilkan dengan minimum confidence 50% sebesar 1.15. Nilai *lift ratio* tertinggi sebesar 2.19 pada *dealer* Honda dan nilai *lift ratio* terendah sebesar 0.90 pada *dealer* Toyota.

Kata kunci: penggalian data, aturan asosiasi, *fp-growth*.

ABSTRACT

Competition in business requires a strategy in order to increase sales. Business strategy can be done by segmenting the market based on demographic aspects by analyzing the characteristics of car buyers based on the parameters gender and age range of the sales transaction data to obtain information which fit with market demand. Sales transaction data is a large-scale data so that it takes a special technique called data mining. Data mining is quite effective for the presence of a series of pattern information from large amounts of data. In this research, mining of data carried out on the car dealer sales transaction data. The data, performed to obtain the relationship between gender and age range of the type of car purchased. The system will do a search with the early stages of association rules is to find frequent itemset by applying fp-growth algorithm. Then association pattern searches will be conducted by using association rule after frequent itemset found. In the final stage, the system will calculate the power of association patterns with lift ratio equation. In this research, the results of testing the value of force (lift ratio) rule, the number of rules generated by the dealer is the dealer of Suzuki is 3 rules, dealer of Honda is 5 rules, dealer of Toyota is 4 rules, and dealer of Daihatsu is 4 rules. And average value earned on lift ratio of generated rule is 50% and minimum confidence of 1.15. Highest lift value ratio of 2.19 at dealer of Honda and the lowest value ratio of 0.90 at dealer of Toyota.

Keywords: data mining, association rules, fp-growth.

DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS SKRIPSI.....	iii
KATA PENGANTAR.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiv
DAFTAR SOURCECODE.....	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	4
1.4. Tujuan.....	4
1.5. Manfaat.....	4
1.6. Metodologi Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1. Bisnis Proses <i>Dealer</i> Mobil.....	6
2.2. Segmentasi Pasar	6
2.2.1. Manfaat Segmentasi Pasar	7
2.2.2. Aspek Segmentasi Pasar	7
2.3. Data	8
2.4. Data Transaksi	8
2.5. Data Mining.....	9
2.5.1. Pengertian Data Mining	9
2.5.2. Proses Data Mining	10
2.5.3. Fungsionalitas Data Mining.....	12
2.5.4. Teknik Data Mining	13



2.6.	Association Rule.....	14
2.6.1.	Support	15
2.6.2.	Confidence	15
2.7.	FP-Tree.....	16
2.8.	Algoritma <i>FP-Growth</i>	18
2.9.	Lift Ratio	20
2.10.	Pengujian Akurasi <i>Association Rule</i>	21
BAB III METODOLOGI DAN PERANCANGAN		22
3.1.	Pengumpulan Data	23
3.1.1.	Studi Literatur	23
3.1.2.	Pengumpulan data lapangan.....	23
3.2.	Analisis dan Perancangan Sistem.....	24
3.2.1.	Deskripsi Sistem	24
3.2.2.	Rancangan Pembuatan Sistem	25
3.3.	Perhitungan Manual	43
3.4.	Rancangan Antar Muka.....	70
3.5.	Rancangan Uji Coba.....	71
BAB IV IMPLEMENTASI DAN PEMBAHASAN		72
4.1.	Lingkungan Implementasi.....	72
4.1.1.	Lingkungan Perangkat Keras	72
4.1.2.	Lingkungan Perangkat Lunak	72
4.2.	Implementasi Program	72
4.2.1.	Implementasi Tahap <i>Frequent 1-itemset</i>	76
4.2.2.	Implementasi Tahap <i>Frequent Itemset</i>	84
4.2.3.	Implementasi Tahap <i>Generate Rule</i>	91
4.2.4.	Implementasi Tahap <i>Lift Ratio</i>	94
4.3.	Implementasi Antarmuka	95
4.4.	Pengujian Sistem	102
4.5.	Analisa Hasil	108
BAB V PENUTUP		119
5.1.	Kesimpulan.....	119
5.2.	Saran.....	120



DAFTAR GAMBAR

Gambar 2. 1 Tahap – tahap Proses Knowledge Discovery	11
Gambar 2. 2 Proses Data Mining	12
Gambar 2. 3 Algoritma FP-Tree [HAN-00].....	17
Gambar 2. 4 <i>FP-Tree</i> [YIA-06].	18
Gambar 3. 1 Langkah – Langkah Penelitian.....	22
Gambar 3. 2 Flowchart Gambaran Umum Sistem.....	25
Gambar 3. 3 Flowchart proses association rule	26
Gambar 3. 4 <i>Flowchart</i> proses frequent 1-itemset.....	27
Gambar 3. 5 Flowchart proses koleksi jenis mobil.....	28
Gambar 3. 6 <i>Flowchart</i> proses koleksi kombinasi antara jenis kelamin dan rentang umur	30
Gambar 3. 7 <i>Flowchart</i> proses pencarian jenis mobil yang frequent	31
Gambar 3. 8 <i>Flowchart</i> proses pencarian kombinasi yang <i>frequent</i>	32
Gambar 3. 9 <i>Flowchart</i> proses seleksi data transaksi	33
Gambar 3. 10 <i>Flowchart</i> proses pembentukan <i>FP-tree</i>	34
Gambar 3. 11 Flowchart proses insert tree	35
Gambar 3. 12 Flowchart proses algoritma FP-Growth.....	35
Gambar 3. 13 <i>Flowchart</i> proses conditional <i>pattern base</i>	37
Gambar 3. 14 <i>Flowchart</i> proses conditional <i>FP-tree</i>	38
Gambar 3. 15 <i>Flowchart</i> proses koleksi <i>frequent itemset</i>	39
Gambar 3. 16 <i>Flowchart</i> proses perhitungan <i>confidence</i>	41
Gambar 3. 17 <i>Flowchart</i> proses perhitungan <i>lift ratio</i>	42
Gambar 3. 18 <i>Flowchart</i> proses seleksi <i>rule</i>	43
Gambar 3. 19 <i>FP-Tree</i> Awal.....	50
Gambar 3. 20 <i>FP-Tree</i> Pembacaan id Transaksi 1	50
Gambar 3. 21 <i>FP-Tree</i> Pembacaan id Transaksi 2	51
Gambar 3. 22 <i>FP-Tree</i> Pembacaan id Transaksi 3	52
Gambar 3. 23 <i>FP-Tree</i> Pembacaan id Transaksi 5	52
Gambar 3. 24 <i>FP-Tree</i> Pembacaan id Transaksi 6	53

Gambar 3. 25 <i>FP-Tree</i> Pembacaan id Transaksi 7	53
Gambar 3. 26 <i>FP-Tree</i> Pembacaan id Transaksi 8	54
Gambar 3. 27 <i>FP-Tree</i> Pembacaan id Transaksi 9	55
Gambar 3. 28 <i>FP-Tree</i> Pembacaan id Transaksi 10	55
Gambar 3. 29 <i>FP-Tree</i> Pembacaan id Transaksi 11	56
Gambar 3. 30 <i>FP-Tree</i> Pembacaan id Transaksi 12	57
Gambar 3. 31 <i>FP-Tree</i> Pembacaan id Transaksi 13	58
Gambar 3. 32 <i>FP-Tree</i> Pembacaan id Transaksi 14	59
Gambar 3. 33 <i>FP-Tree</i> Pembacaan id Transaksi 15	60
Gambar 3. 34 <i>FP-Tree</i> Pembacaan id Transaksi 16	61
Gambar 3. 35 <i>FP-Tree</i> Pembacaan id Transaksi 17	62
Gambar 3. 36 <i>FP-Tree</i> Pembacaan id Transaksi 18	62
Gambar 3. 37 <i>FP-Tree</i> Pembacaan id Transaksi 19	64
Gambar 3. 38 <i>FP-Tree</i> Pembacaan id Transaksi 20	64
Gambar 3. 39 Lintasan yang memiliki suffix A	65
Gambar 3. 40 Lintasan yang memiliki suffix C	66
Gambar 3. 41 Conditional <i>FP-Tree</i> untuk suffix C	66
Gambar 3. 42 Lintasan yang memiliki suffix D	66
Gambar 3. 43 Conditional <i>FP-Tree</i> untuk suffix D	66
Gambar 3. 44 Lintasan yang memiliki suffix E	67
Gambar 3. 45 Conditional <i>FP-Tree</i> untuk suffix E	67
Gambar 3. 46 Lintasan yang memiliki suffix F	67
Gambar 3. 47 Lintasan yang memiliki suffix F	67
Gambar 3. 48 Conditional <i>FP-Tree</i> untuk suffix F	68
Gambar 3. 49 Rancangan Antar Muka Secara Umum	71
Gambar 4. 1 Tampilan Data Transaksi	96
Gambar 4. 2 Tampilan Dialog Box Peringatan	97
Gambar 4. 3 Tampilan Submenu One Itemset	98
Gambar 4. 4 Tampilan Submenu Frequent 1-Itemset	99
Gambar 4. 5 Tampilan Submenu Frequent Itemset	99
Gambar 4. 6 Tampilan Submenu Rule	100
Gambar 4. 7 Tampilan Submenu Akurasi	101

Gambar 4. 8 Tampilan Tabel Rule Setelah Uji Akurasi	101
Gambar 4. 9 Tampilan Keterangan Export Data Output	102
Gambar 4. 10 Grafik Confidence dan Lift Ratio Rule Dealer Daihatsu	110
Gambar 4. 11 Grafik Confidence dan Lift Ratio Rule Dealer Toyota.....	111
Gambar 4. 12 Grafik Confidence dan Lift Ratio Rule Dealer Honda.....	112
Gambar 4. 13 Grafik Confidence dan Lift Ratio Rule Dealer Suzuki	112
Gambar 4. 14 Grafik Akurasi Association Rule yang terbentuk dari dealer Daihatsu	115
Gambar 4. 15 Grafik Akurasi Association Rule yang Terbentuk dari Dealer Toyota.....	116
Gambar 4. 16 Grafik Akurasi Association Rule yang Terbentuk dari Dealer Suzuki	117
Gambar 4. 17 Grafik Akurasi Association Rule yang Terbentuk dari Dealer Honda	117



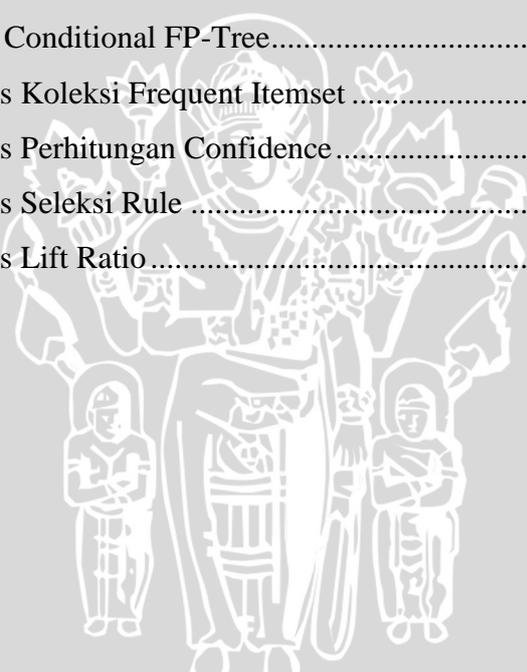
DAFTAR TABEL

Tabel 2. 1 Tabel Contoh Data Transaksi pada Suatu Dealer Mobil.....	6
Tabel 2. 2 Tabel Contoh Data Transaksi [YIA-06].....	18
Tabel 2. 3 Hasil Fase Proses Penggalian <i>FP-growth</i>	20
Tabel 3. 1 Sampel Data Transaksi Penjualan Mobil Hasil <i>Preprocessing</i>	44
Tabel 3. 2 Frekuensi Kemunculan Tiap Jenis Mobil	45
Tabel 3. 3 Frekuensi Kemunculan Tiap Kombinasi Antara Jenis Kelamin dengan Rentang Umur	45
Tabel 3. 4 Support Kandidat <i>Frequent1 -Itemset</i> Jenis Mobil	46
Tabel 3. 5 <i>Support</i> Kandidat <i>Frequent1 -Itemset</i> Kombinasi Antara Jenis Kelamin dengan Rentang Umur.....	47
Tabel 3. 6 <i>Frequent 1-Itemset</i> Jenis Mobil	47
Tabel 3. 7 <i>Frequent 1-Itemset</i> Jenis Kelamin dan Rentang Umur.....	48
Tabel 3. 8 Sampel Data Transaksi Hasil Proses Seleksi	48
Tabel 3. 9 Tabel Transformasi Data Transaksi	49
Tabel 3. 10 Conditional Pattern Base dari Sampel Data.....	65
Tabel 3. 11 Hasil Frequent Itemset Sampel Data.....	68
Tabel 3. 12 Confidence Frequent Itemset Sampel Data.....	69
Tabel 3. 13 Rule yang Terbentuk.....	69
Tabel 3. 14 Lift Ratio Rules	70
Tabel 3. 15 Tabel Rancangan Uji Coba Lift Ratio Rule.....	71
Tabel 3. 16 Tabel Rancangan Uji Coba Akurasi Rule	71
Tabel 4. 1 Class Pada Implementasi Program dan Fungsinya	73
Tabel 4. 2 Hasil Uji Tingkat Kekuatan (Lift Ratio) Rule yang Dihasilkan.	102
Tabel 4. 3 Keterangan Rule Hasil Uji Kekuatan Lift Ratio.	106
Tabel 4. 4 Akurasi Rule Hasil Uji Kekuatan Lift Ratio.....	107
Tabel 4. 5 Rule yang Bermanfaat.....	114
Tabel 4. 6 Detail Informasi Rule yang Bermanfaat	115



DAFTAR SOURCECODE

Sourcecode 4. 1 Proses Pembacaan Data Transaksi	77
Sourcecode 4. 2 Proses Koleksi Jenis Mobil	79
Sourcecode 4. 3 Proses Koleksi Kombinasi.....	81
Sourcecode 4. 4 Proses Pencarian Jenis Mobil yang Frequent.....	82
Sourcecode 4. 5 Proses Pencarian Kombinasi yang Frequent	83
Sourcecode 4. 6 Proses Seleksi Data Transaksi	85
Sourcecode 4. 7 Proses Pembentukan FP-Tree.....	87
Sourcecode 4. 8 Proses Conditional Pattern Base.....	88
Sourcecode 4. 9 Proses Conditional FP-Tree.....	89
Sourcecode 4. 10 Proses Koleksi Frequent Itemset	90
Sourcecode 4. 11 Proses Perhitungan Confidence.....	92
Sourcecode 4. 12 Proses Seleksi Rule	94
Sourcecode 4. 13 Proses Lift Ratio	95



BAB I PENDAHULUAN

1.1. Latar Belakang

Banyaknya *dealer* mobil saat ini, membuat setiap *dealer* mobil harus berupaya untuk bersaing dalam meningkatkan penjualan. Untuk tetap dapat bersaing dalam pangsa pasar mobil teratas maka dibutuhkan informasi untuk mendukung pembuatan strategi pemasaran dimana salah satunya adalah dengan melakukan segmentasi pasar. Segmentasi pasar menurut [ARM-04] adalah membagi sebuah pasar ke dalam kelompok-kelompok pembeli yang khas berdasarkan kebutuhan, karakteristik, atau perilaku yang mungkin membutuhkan produk atau bauran pemasaran yang terpisah. Segmentasi pasar terdiri dari 4 aspek, yaitu aspek geografis, demografis, psikografis dan perilaku. Pada penelitian ini, aspek segmentasi pasar yang digunakan adalah aspek demografis yang bertujuan untuk analisa karakteristik pembeli mobil berdasarkan parameter demografis (jenis kelamin, umur, status, perkawinan, pendapatan, lokasi geografi, pendidikan). Parameter dari aspek demografis yang digunakan pada penelitian ini yaitu, jenis kelamin dan umur. Melalui analisa karakteristik pembeli mobil, suatu *dealer* mobil memiliki informasi tentang target pembeli mobil untuk tiap jenis mobil sehingga *dealer* mobil dapat mempertahankan bahkan meningkatkan tingkat penjualan serta memberikan dampak jangka panjang yang lebih kompetitif. Selain itu juga, suatu *dealer* dapat mendeteksi secara dini dan tepat mengenai kecenderungan-kecenderungan dalam pasar yang senantiasa berubah dan mampu merancang produk yang benar-benar sesuai dengan permintaan pasar. Informasi yang didapatkan antara lain pola hubungan antara jenis kelamin dan rentang umur terhadap jenis mobil yang dibeli.

Sebuah *dealer* mobil memiliki jumlah data transaksi penjualan yang banyak yang terdiri dari tanggal transaksi, nama pembeli, jenis kelamin, usia, jenis mobil dan warna mobil. Suatu data transaksi memiliki pola hubungan antara jenis kelamin, usia, jenis mobil dan warna mobil dan pola tersebut dapat terjadi kembali di transaksi selanjutnya sehingga tiap pola hubungan memiliki frekuensi kejadian.

Maka dari itu diperlukan proses *data mining* yang merupakan salah satu teknik efektif untuk mengetahui pola informasi segmen-segmen pelanggan potensial dengan kesamaan karakteristik *customer* mobil dari sejumlah besar data yang ada. Dalam *data mining* terdapat beberapa teknik dimana penggunaannya bergantung terhadap permasalahan yang dihadapi. Pada permasalahan segmentasi pasar untuk mengetahui karakteristik pembeli mobil ini digunakan teknik *association rule*. Teknik *association rule* adalah teknik *data mining* yang dapat mencari korelasi antara *item* yang satu dengan *item* yang lain dengan mengetahui pola asosiasinya.

Teknik *association rule* terdiri dari dua tahap, antara lain tahap pencarian kombinasi *item* yang paling sering muncul (pencarian *frequent itemset*) dan tahap pembangkitan *rule* yang telah terbentuk dari *frequent itemset* [HAN-01]. Algoritma *association rule* yang paling sering digunakan sampai saat ini adalah algoritma apriori. Proses dasar apriori ini adalah dengan mencari himpunan kandidat dengan panjang $(k+1)$ dari sekumpulan pola *frequent* dengan panjang k , lalu mencocokkan jumlah kemunculan pola tersebut dengan informasi yang terdapat dalam *database*. Adapun hal ini akan mengakibatkan algoritma apriori akan melakukan *scanning database* yang berulang-ulang, apalagi jika jumlah data cukup besar. Berbeda dengan algoritma *FP-Growth* yang hanya memerlukan dua kali *scanning database* untuk menentukan *frequent itemset*, yaitu *scanning* pertama pada saat perhitungan *support frequent 1-itemset* dan *scanning* kedua adalah sebelum pembentukan *fp-tree*. Maka salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data adalah *FP-Growth* [SAM-08]. Pencarian *frequent itemset* dengan algoritma *FP-Growth* tidak dengan membangkitkan kandidat melainkan dengan menggunakan konsep pembangunan *tree* [ERW-09]. Untuk dapat membangun *tree*, algoritma *FP-Growth* mencari item yang *frequent* (*frequent 1-itemset*) dengan cara membandingkan frekuensi kemunculannya dengan nilai minimum *support* yang telah ditentukan. Item yang *frequent* selanjutnya akan dibentuk kombinasinya sesuai dengan data transaksi yang ada sehingga untuk item yang tidak *frequent* telah dipangkas terlebih dahulu. Hasil kombinasi tersebut digunakan untuk membangun *tree* dan dianggap *frequent*

itemset jika frekuensi kemunculannya lebih besar atau sama dengan nilai minimum *support* yang telah ditentukan. Data input yang dibutuhkan oleh algoritma *FP-Growth* harus memiliki *itemset* dengan kejadian yang berulang – ulang.

Dari penelitian sebelumnya, yang dilakukan oleh [ERW-09] tentang *market basket analysis*, didapatkan hasil bahwa algoritma *FP – Growth* dapat memampatkan data transaksi barang pada *supermarket* yang memiliki *item* yang sama sehingga penggunaan memori komputer lebih sedikit dan proses pencarian *frequent itemset* menjadi lebih cepat karena menggunakan struktur data *FP – Tree*. Kelebihan dari algoritma *FP-Growth* juga dijelaskan oleh [VER-08] antara lain hanya membutuhkan dua kali proses pembacaan *dataset*, tidak ada pembangkitan kandidat (*candidate generation*), lebih cepat daripada *Apriori*, dan memampatkan *dataset*. Berdasarkan latar belakang yang telah dijelaskan tersebut, maka teknik *data mining* yang dipilih pada penelitian ini yaitu *association rule* dengan menggunakan algoritma *FP – Growth* untuk mengetahui karakteristik pembeli mobil, dengan judul penelitian “**Penerapan metode *association rule* dengan algoritma *FP – Growth* untuk mengetahui karakteristik pembeli mobil**”.

1.2. Rumusan Masalah

Permasalahan yang dijadikan subyek penelitian dari skripsi ini adalah sebagai berikut:

1. Bagaimana menerapkan metode *association rule* dengan algoritma *FP-Growth* untuk mengetahui karakteristik *customer* mobil berdasarkan umur, jenis kelamin dari data transaksi penjualan *dealer* mobil.
2. Bagaimana kekuatan (*lift ratio*) *rule* hasil penerapan algoritma *fp-growth*.
3. Bagaimana hasil akurasi *rule* hasil penerapan algoritma *fp-growth* terhadap data uji.

1.3. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1. Penelitian dilakukan dengan menggunakan data transaksi penjualan mobil di empat *dealer* resmi di wilayah Mataram yaitu *dealer* mobil Suzuki, Toyota, Daihatsu dan Honda dari bulan Juni sampai Desember 2011.
2. Data transaksi penjualan mobil untuk setiap *dealer* resmi akan dibagi menjadi dua bagian, yaitu data latih dan data uji.
3. *Rule* yang dihasilkan berupa hubungan antara parameter jenis kelamin dan rentang umur terhadap jenis mobil yang dibeli.

1.4. Tujuan

Tujuan dari skripsi ini adalah:

1. Mengetahui karakteristik *customer* mobil berdasarkan umur dan jenis kelamin dari data transaksi penjualan mobil dengan menerapkan metode association rule menggunakan algoritma FP-Growth.
2. Mengukur kekuatan *rule* (*lift ratio*) hasil penerapan algoritma *fp-growth*.
3. Mengukur akurasi *rule* hasil penerapan algoritma *fp-growth* terhadap data uji.

1.5. Manfaat

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Memberikan informasi mengenai karakteristik *customer* mobil tiap *dealer* di wilayah Mataram.
2. Untuk pemanfaatan lebih lanjut, informasi mengenai karakteristik *customer* dapat mendukung keputusan dalam pembuatan strategi untuk promosi mobil, tepatnya kepada siapa mobil ini harus ditawarkan berdasarkan rentang umur dan jenis kelamin.

1.6. Metodologi Penulisan

Sistematika makalah penelitian skripsi ini dibagi menjadi lima bab dengan masing-masing bab diuraikan sebagai berikut :

1. BAB I PENDAHULUAN

Berisi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat, metode penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori berbagai pustaka yang menunjang penelitian dalam penulisan skripsi yaitu mengenai definisi dan konsep *Data mining*, *Association Rule*, *FP-Tree*, *FP-Growth*.

3. BAB III METODOLOGI DAN PERANCANGAN

Bab ini berisi tentang perancangan perangkat lunak yang dibangun, meliputi perancangan proses , perancangan tabel, dan perancangan uji coba.

4. BAB IV IMPLEMENTASI DAN PEMBAHASAN

Bab ini berisi hasil dari implementasi perangkat lunak yang digunakan untuk mengetahui karakteristik pembeli mobil menggunakan algoritma *FP-Growth*.

5. BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari hasil penelitian dan saran sebagai pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1. Bisnis Proses Dealer Mobil

Dalam proses bisnis *dealer* penjualan mobil harus melengkapi beberapa syarat yang harus dipenuhi, dimana setiap pembeli yang melakukan transaksi pembelian dilakukan pengarsipan atau pencatatan data transaksi yang terdiri dari informasi data pembeli dan data mobil. Pada setiap data pembeli terdiri dari nama pembeli, jenis kelamin, tanggal lahir, umur, alamat, pekerjaan, dan status. Sedangkan untuk setiap data mobil terdiri dari type mobil, jenis mobil, tahun pembuatan, warna mobil, nomor mesin dan nomor rangka. Contoh data transaksi dapat dilihat pada tabel 2.1.

Tabel 2. 1 Tabel Contoh Data Transaksi pada Suatu Dealer Mobil.

Tipe Kend.	Nama Pembeli	Tgl. Lahir	Jenis Kelamin	Tgl. Transaksi	Umur
JAZZ GE8 E MT	ROHINI AZIM	11/28/1984 12:00:00 AM	W	7/4/2011 11:00	27
JAZZ GE8 E MT	H.HASANUDDIN. SIP	1/1/1970 0:00	P	7/4/2011 16:04	41
JAZZ GE8 E AT	A UKKI G HARTANTO	6/9/1975 0:00	P	7/12/2011 7:37	36
CITY GM2 1.5 E AT	I PUTU DEDI SAPUTRA,ST.	12/31/1954 12:00:00 AM	P	7/12/2011 14:00	57

2.2. Segmentasi Pasar

Segmentasi pasar adalah suatu proses membagi pasar ke dalam segmen-segmen pelanggan potensial dengan kesamaan karakteristik yang menunjukkan adanya kesamaan perilaku pembeli. Segmentasi pasar terdiri dari pembeli dan pembeli berbeda-beda dalam berbagai hal yang bisa membeli dalam keinginan,

sumber daya, lokasi, sikap membeli, dan kebiasaan membeli. Karena masing-masing memiliki kebutuhan dan keinginan yang unik, masing-masing pembeli merupakan pasar potensial tersendiri. Oleh sebab itu penjual idealnya mendisain program pemasarannya tersendiri bagi masing-masing pembeli [ARM-04].

2.2.1. Manfaat Segmentasi Pasar

Menurut [ARM-04] menyatakan manfaat segmentasi pasar ini, sebagai berikut:

1. Dapat membedakan antara segmen yang satu dengan segmen lainnya.
2. Dapat digunakan untuk mengetahui sifat masing-masing segmen.
3. Dapat digunakan untuk mencari segmen mana yang potensinya paling besar.
4. Dapat digunakan untuk memilih segmen mana yang akan dijadikan pasar sasaran.

2.2.2. Aspek Segmentasi Pasar

Pengusaha yang melakukan segmentasi pasar akan berusaha mengelompokkan konsumen kedalam beberapa segmen yang secara relatif memiliki sifat-sifat homogen dan kemudian memperlakukan masing-masing segmen dengan cara atau pelayanan yang berbeda.

Seberapa jauh pengelompokkan itu harus dilakukan, nampaknya banyak faktor yang terlebih dahulu perlu dicermati. Faktor-faktor tersebut antara lain sebagai berikut:

1. Segmentasi Geografi.

Segmentasi ini membagi pasar menjadi unit-unit geografi yang berbeda, seperti negara, propinsi, kabupaten, kota, wilayah, daerah atau kawasan. Jadi dengan segmentasi ini, pemasar memperoleh kepastian kemana atau dimana produk ini harus dipasarkan.

2. Segmentasi Demografi.

Segmentasi ini memberikan gambaran bagi pemasar kepada siapa produk ini harus ditawarkan. Jawaban atas pertanyaan kepada siapa dapat berkonotasi pada umur, jenis kelamin, jumlah anggota keluarga, siklus kehidupan keluarga seperti anak-anak, remaja, dewasa, kawin/ belum kawin, keluarga

muda dengan satu anak, keluarga dengan dua anak, keluarga yang anak-anaknya sudah bekerja dan seterusnya. Dapat pula berkonotasi pada tingkat penghasilan, pendidikan, jenis pekerjaan, pengalaman, agama dan keturunan.

3. Segmentasi Psikografi.

Segmentasi psikografi membagi pemebel menjadi kelompok berbeda berdasarkan pada karakteristik kelas social, gaya hidup, atau kepribadian. Dalam kelompok demografi, orang yang berbeda dapat mempunyai ciri psikografi yang berbeda

4. Segmentasi Tingkah Laku.

Segmentasi tingkah laku mengelompokkan pembeli berdasarkan pada pengetahuan, sikap, penggunaan atau reaksi mereka terhadap suatu produk. Banyak pemasar yakin bahwa variabel tingkah laku merupakan awal paling baik untuk membentuk segmen pasar. Pada penelitian ini, faktor segmentasi pasar yang digunakan adalah segmentasi demografi.

2.3. Data

Definisi data adalah nilai yang mewakili sebuah fakta atau gambaran dari objek atau kejadian. Fakta dari transaksi dalam perusahaan di bidang perdagangan misalnya, dapat berupa transaksi penjualan yang meliputi waktu transaksi, pelaku transaksi, barang yang ditransaksikan beserta jumlah dan harganya. Dalam menyatakan data dapat berupa nilai yang berbentuk angka, deretan karakter, atau symbol [KUS-07].

2.4. Data Transaksi

Transaksi yang terjadi dalam suatu perusahaan dapat dibagi menjadi dua kelompok, antara lain [KOT-07] :

1. Transaksi ekstern, yaitu transaksi yang terjadi dengan pihak luar perusahaan, seperti penjualan, pembelian, pengeluaran, dan penerimaan uang.
2. Transaksi intern, yaitu pembagian kembali biaya – biaya dalam perusahaan, seperti depresiasi aktiva tetap, pemakaian bahan baku untuk produksi, transfer dari barang dalam proses ke barang jadi.

Transaksi ekstern akan lebih mudah dianalisis daripada transaksi intern karena sebagian besar transaksi tersebut mengakibatkan penambahan dan pengurangan data perusahaan. Fungsi dari data transaksi adalah sebagai bahan dasar obyektif (relatif) di dalam proses penyusunan kebijaksanaan dan keputusan oleh pimpinan organisasi.

2.5. Data Mining

2.5.1. Pengertian Data Mining

Beberapa pengertian tentang *data mining*, antara lain sebagai berikut :

1. *Data mining* adalah suatu proses dalam mencari korelasi, pola, atau tren yang bermanfaat dari sebuah data yang berukuran besar dengan menggunakan teknik statistika ataupun matematika [LAR-05].
2. *Data mining* adalah salah satu teknik dalam ilmu komputer yang digunakan untuk menggali dan mengambil suatu informasi pada banyak data [REN-04].
3. *Data mining* adalah mencocokkan data dalam suatu model untuk menemukan informasi yang tersembunyi dalam basis data [HAN-01].
4. *Data mining* adalah seangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual [PRA-03].
5. *Data mining* Proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning*, untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait di berbagai *database* [TUR-05].

Berdasarkan beberapa pengertian diatas dapat disimpulkan bahwa penggalian atau penambangan data (*data mining*) adalah proses pencarian otomatis terhadap pola dalam data dalam jumlah besar untuk melihat informasi yang tersembunyi dengan menggunakan perangkat seperti klasifikasi, penggugusan (*clustering*), dan *association rule*.

2.5.2. Proses Data Mining

Data mining dan *knowledge discovery in database* (KDD) merupakan istilah yang memiliki konsep berbeda akan tetapi saling berkaitan karena *data mining* adalah bagian dalam proses *knowledge discovery in database* (KDD) [KUS-07]. Proses *knowledge discovery in database* (KDD) secara umum adalah sebagai berikut :

1. *Data Selection.*

Merupakan tahap seleksi data, yang akan digunakan dalam proses *data mining*, dari sejumlah besar data operasional. Hasil dari seleksi data disimpan dalam suatu berkas terpisah dari *database* operasional.

2. *Preprocessing atau Cleaning*

Pada tahap ini, dilakukan pembuangan duplikasi data, pemeriksaan data yang tidak konsisten, dan memperbaiki kesalahan pada data, serta memperkaya data yang sudah ada dengan data atau informasi eksternal.

3. *Transformation*

Tahap transformasi sangat bergantung pada jenis atau pola informasi yang akan dicari dalam basis data. Menurut Huda (2010), transformasi merupakan proses pengubahan sesuai format yang dibutuhkan untuk digunakan dalam proses *data mining*.

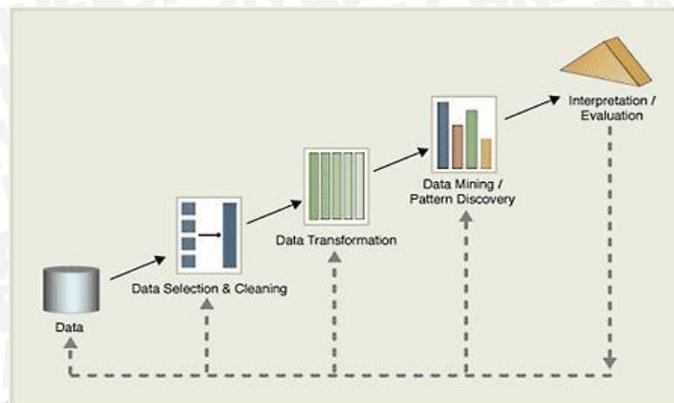
4. *Data Mining*

Merupakan tahap pencarian pola atau informasi dari data yang terpilih dengan menggunakan metode atau teknik tertentu. Ketepatan metode atau teknik yang dipilih sangat bergantung pada tujuan dari proses *knowledge discovery in database* (KDD) secara keseluruhan.

5. *Interpretation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

Keseluruhan dari proses *knowledge discovery in database* (KDD) diilustrasikan pada Gambar 2.1.



Gambar 2. 1 Tahap – tahap Proses Knowledge Discovery
in Database [HAN-01]

Menurut [KAN-03], prosedur umum untuk menyelesaikan permasalahan *data mining* adalah sebagai berikut (seperti yang diilustrasikan pada Gambar 2.2) :

1. Merumuskan permasalahan

Pada tahap ini, ditetapkan sebuah rumusan masalah serta variable – variabel yang terlibat.

2. Mengumpulkan data

Pada prosedur ini, konsentrasi ditujukan pada proses pembuatan atau pengumpulan data.

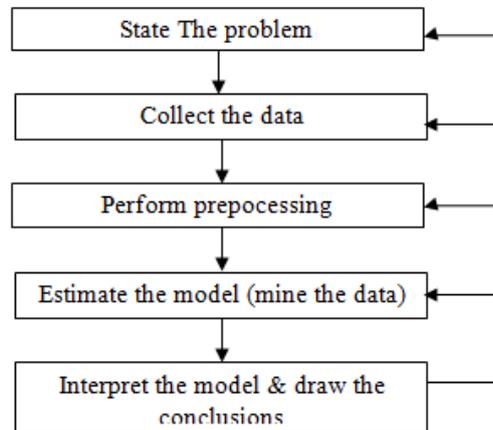
3. Preprocessing data

Preprocessing data digunakan untuk menyeleksi data yang akan digunakan dalam proses.

4. Estimasi model

Disebut juga proses utama dari prosedur ini sebab implementasi dari teknik *data mining* dilakukan pada proses ini.

5. Menafsirkan informasi yang dihasilkan dari proses Sebelumnya



Gambar 2. 2 Proses Data Mining

2.5.3. Fungsionalitas Data Mining

Data mining dapat diklasifikasikan berdasarkan fungsi yang dilakukan atau berdasarkan kelas aplikasi yang digunakan [HAN-01] :

1. *Characterization*

Memberikan ringkasan mengenai karakteristik objek-objek dalam satu kelas tertentu.

2. *Discrimination*

Membandingkan karakteristik objek-objek pada kelas tertentu dengan kelas lainnya

3. *Association/Asosiasi*

Mempelajari frekuensi item-item yang terjadi secara bersamaan dalam transaksi *database*.

4. *Prediction*

Meramalkan nilai atribut yang hilang atau tidak diketahui berdasarkan informasi lain. Tipe datanya bersifat kontinyu.

5. *Classification/Klasifikasi*

Mengklasifikasikan data ke dalam kelas yang diberikan berdasarkan nilai atribut (*supervised classification*). Tipe datanya bersifat kategorik.

6. *Clustering*/Klasterisasi

Klusterisasi merupakan *unsupervised classification*. Dalam klusterisasi, kelas-kelas tidak didefinisikan terlebih dahulu.

7. *Outlier Analysis*

Mengidentifikasi dan menjelaskan *noise* dan *outlier*. *Outlier* adalah objek data yang tidak memenuhi model dan persyaratan secara umum, yang berbeda dan tidak konsisten dengan data set yang ada.

8. *Time-series analysis*

Menganalisa tren, deviasi, regresi, *sequential pattern* dan *similar sequence*.

2.5.4. Teknik Data Mining

Menurut [PRA-03], teknik – teknik *data mining* yang paling populer antara lain :

a. ***Association Rule Mining***

Association Rule Mining adalah teknik *data mining* untuk menemukan aturan asosiasi antara suatu kombinasi *item*.

b. ***Classification***

Classification adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data dengan tujuan untuk dapat memperkirakan kelas dari suatu obyek yang labelnya tidak diketahui.

c. ***Clustering***

Clustering adalah proses pengelompokkan data tanpa berdasarkan kelas data tertentu. *Clustering* dapat dipakai untuk memberikan label pada kelas data yang belum diketahui.

Menurut [KAN-03], teknik – teknik dalam *data mining* terdiri dari :

1. **Klasifikasi**

Penemuan suatu fungsi yang dapat mengklasifikasikan data ke dalam salah satu kelas dari beberapa kelas yang telah ditentukan sebelumnya.

2. **Regresi**

Penentuan suatu fungsi yang dapat memetakan data ke suatu nilai real dan variabel perkiraan.

3. *Clustering*

Tugas deskriptif yang umumnya digunakan untuk menentukan himpunan berhingga dari kategori atau *cluster* untuk mendeskripsikan data.

4. *Summarization (Ringkasan)*

Tugas deskripsi tambahan yang meliputi metode untuk menemukan deskripsi dari kumpulan data atau bagian dari kumpulan data.

5. *Dependency Modeling (Pemodelan Keterkaitan Antar Data)*

Menemukan suatu model yang mendeskripsikan keterkaitan atau hubungan yang signifikan antara variabel atau nilai dari fitur – fitur dalam kumpulan data atau bagian dari kumpulan data.

6. *Change and Deviation Detection (Penemuan Perubahan dan Deviasi atau Simpangan)*

Menemukan perubahan yang paling signifikan dalam kumpulan data.

2.6. Association Rule

Association Rule adalah teknik data mining untuk menemukan aturan hubungan asosiatif antara suatu kombinasi item dengan tujuan akhir mendapatkan kombinasi item yang dikatakan menarik (*interesting association rules*). *Association rule* meliputi dua tahap [GRE-05] :

1. Mencari kombinasi yang paling sering terjadi dari suatu *itemset*.
2. Mendefinisikan *Condition* dan *Result* (untuk *conditional association rule*).

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter [SAN-07], yaitu:

1. *Support*, suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item* atau *itemset* dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu *item* atau *itemset* layak untuk dicari *confidence factor*-nya.
2. *Confidence*, suatu ukuran yang menunjukkan hubungan antar dua atau lebih *item* secara *conditional* .

Kedua ukuran ini nantinya berguna dalam menentukan *interesting association rules*, yaitu nilai suatu *item* dengan melihat peluang munculnya *item* berdasarkan support dari item tersebut dan kemudian dibandingkan dengan batasan (*threshold*)

yang ditentukan oleh *user*. Batasan tersebut umumnya terdiri dari *minimum support* dan *minimum confidence*.

2.6.1. Support

Support adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan transaksi [HAN-01]. Menurut [ROC-10], *support* dari aturan asosiasi adalah proporsi transaksi yang mengandung *antecedent* dan

consequent. *Support* adalah probabilitas pelanggan membeli beberapa produk secara bersamaan dari jumlah seluruh transaksi.

Support untuk aturan “ $X \rightarrow Y$ ” adalah probabilitas atribut atau kumpulan atribut X dan Y yang terjadi bersamaan dalam suatu transaksi [YUL-02]. Nilai *support* untuk suatu item dapat diperoleh dengan rumus pada Persamaan 2.1, sedangkan nilai *support* untuk kombinasi item *antecedent* dan *consequent* dapat diperoleh dengan rumus pada Persamaan 2.2 [ROC-10].

$$\text{Support}(A) = \frac{\text{Jumlah transaksi yang mengandung } A}{\text{Total transaksi}} \quad (2.1)$$

$$\begin{aligned} \text{Support}(A, C) &= P(A \cap C) \\ &= \frac{\text{Jumlah transaksi yang mengandung } A \text{ dan } C}{\text{Total transaksi}} \end{aligned} \quad (2.2)$$

Nilai *support* digunakan untuk menentukan *frequent itemset*. *Itemset* yang nilai *support*-nya memenuhi parameter *minimum support* (*min_sup*) masuk dalam *frequent itemset*. *Minimum support* merupakan parameter yang digunakan sebagai batasan frekuensi kejadian atau jumlah *support* yang harus dipenuhi suatu kelompok data yang dijadikan aturan [YUL-02].

2.6.2. Confidence

Confidence adalah suatu ukuran yang menunjukkan hubungan antara dua atau lebih item secara *conditional* [HAN-01]. Menurut [ROC-10], *confidence* adalah *ratio* antara jumlah transaksi yang meliputi semua item dalam *antecedent*

dan *consequent* dengan jumlah transaksi meliputi semua *item* dalam *antecedent*. *Confidence* atau tingkat kepercayaan merupakan probabilitas kejadian beberapa produk yang dibeli secara bersamaan dimana salah satu produk sudah pasti dibeli, misalnya jika terdapat n transaksi dimana X yang dibeli dan m transaksi dimana X dan Y yang dibeli bersamaan, maka *confidence* dari aturan $X \rightarrow Y$ adalah m/n [YUL-02]. Nilai *confidence* dapat diperoleh dengan rumus pada Persamaan 2.3 dan dijabarkan lebih khusus pada Persamaan 2.4 [ROC-10].

$$\text{Confidence } (A \rightarrow B) = \frac{\text{Support } A \cap B}{\text{Support } A} \quad (2.3)$$

$$\begin{aligned} \text{Confidence } (A \rightarrow B) &= P(B|A) \\ &= \frac{\text{Jumlah transaksi yang mengandung } A \& B}{\text{Jumlah transaksi yang mengandung } A} \end{aligned} \quad (2.4)$$

Nilai *confidence* digunakan dalam menentukan *strong association rule*. *Association rule* yang nilai *confidence*-nya memenuhi parameter *threshold* minimum *confidence* (*min_conf*) termasuk dalam *strong association rule*. Minimum *confidence* adalah parameter yang mendefinisikan minimum *level* dari *confidence* yang dipenuhi oleh aturan yang berkualitas [YUL-02].

2.7. FP-Tree.

FP-tree adalah struktur data berbentuk pohon (*tree*), yang terdiri dari : satu tabel *header*, satu *root* yang diberi label *null*, dan satu himpunan item *prefix subtree* sebagai node dari anak *root*. Masing-masing *entry* dalam tabel *header* adalah *frequent itemset*, dan setiap *record* terdiri dari dua atribut yaitu nama *item* (*item_name*) dan *node* yang menjadi *head of node-link*. Sedangkan setiap node anak terdiri dari informasi nama *item*, *count*, dan *node-link*. Dimana *count* menunjukkan jumlah transaksi direpresentasikan oleh cabang yang mengandung node tersebut, dan *node-link* menghubungkan node ke node berikutnya pada tree tersebut yang mempunyai nama item yang sama atau *null* jika tidak ada [MAR-06].

Contoh penerapan algoritma pembuatan *FP-tree* adalah dimisalkan terdapat data transaksi pada Tabel 2.1 dengan *minimumsupport* adalah 2 maka hasil *frequent 1-itemset* adalah $F = \{\{I2 :7\}, \{I1 :6\}, \{I3 :6\}, \{I4 :2\}, \{I5 :2\}\}$ karena item – item tersebut memiliki *support* lebih besar atau sama dengan *minimum support*. Selanjutnya, pembentukan cabang pertama dilakukan untuk transaksi pertama yang terdiri dari tiga node yaitu $\langle I2 :1 \rangle$, $\langle I1 :1 \rangle$, dan $\langle I5 :1 \rangle$. Node I2 dihubungkan ke *root*, Node I1 menjadi anak dari node I2, dan I3 menjadi anak dari node I1. Transaksi selanjutnya merupakan pembentukan cabang kedua dari *FP-tree* yang terdiri dari dua node yaitu I2 dan I4. Karena node I2 sudah terbentuk pada *FPtree* maka tidak perlu dibentuk node yang sama melainkan 17 memampatkan pada node yang telah ada dengan menambahkan nilai *support* pada node I2 sehingga node I2 menjadi $\langle I2 :2 \rangle$ dan menciptakan node baru $\langle I4 :1 \rangle$ menjadi anak dari node $\langle I2 :2 \rangle$. Hasil *FP-tree* setelah pembacaan semua transaksi dari Tabel 2.2 dapat dilihat pada Gambar 2.3 [YIA-06].

Pembentukan **FP-Tree**

Input : Basis Transaksi (D) dan nilai *minimum support* (minsup);

Output : Ifrequent pattern tree, FP-tree;

Method : FP-tree dibentuk dengan cara berikut.

1. Transaksi di *database* (DB) ditelusuri sekali agar didapat himpunan *frequent item* (F) dan nilai support-nya. Lalu F diurutkan mulai dari yang mempunyai nilai *support* terbesar sampai terkecil, dan hasil pengurutannya dilambangkan dengan L (daftar *frequent item*).
2. Buat *root* dari FP-tree, T, dan diberi nilai *null*. **Foreach** transaksi *Trans* di DB lakukan langkah berikut.

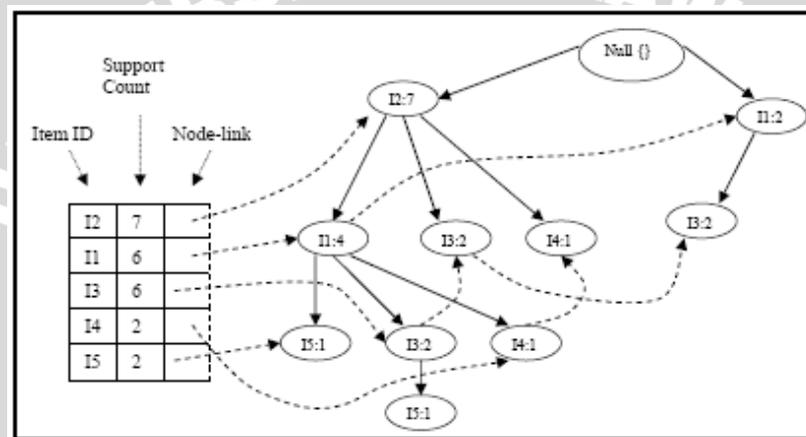
Pilih dan diurutkan *frequent items* dalam *Trans* (disebut $[p|P]$) mengacu pada L, dimana *p* adalah elemen pertama dan P adalah *item* berikutnya. Panggil *insert_tree*($[p|P],T$).

Fungsi *insert_tree*($[p|P],T$) terdiri dari langkah-langkah berikut. **If** T punya anak N dimana $N.nama-item=p.nama-item$, maka nilai N bertambah 1; **else** buat node baru N, beri nilai 1, $N.parent.link = T$, dan $N.link =$ node yang sama nama itemnya dihubungkan oleh struktur *node-link*. **If** P tidak kosong, **then** panggil *insert_tree* (P,N) secara rekursif.

Gambar 2. 3 Algoritma FP-Tree [HAN-00].

Tabel 2. 2 Tabel Contoh Data Transaksi [YIA-06].

TID	Items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Gambar 2. 4 FP-Tree [YIA-06].

2.8. Algoritma FP-Growth.

Algoritma *FP-Growth* merupakan pengembangan dari algoritma *Apriori*. *Frequent Patten Growth (FP-Growth)* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) [SAM-08]. Pada algoritma *FP-Growth* tidak dilakukan *generate candidate* seperti yang diperukan pada algoritma *Apriori* karena *FP-Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset*. Oleh sebab itu, algoritma *FP-Growth* memiliki waktu kompilasi yang lebih cepat dari algoritma *Apriori* [ERW-09].

FP-Growth menggunakan pendekatan yang berbeda dari paradigma yang selama ini sering digunakan, yaitu paradigma *Apriori*. Algoritma *Apriori* melakukan pemindaian basis data secara berulang – ulang untuk menentukan

frequent itemset, sedangkan algoritma *FP-Growth* hanya memerlukan dua kali pemindaian basis data [SAM-08].

Karakteristik algoritma *FP - Growth* adalah struktur data yang digunakan adalah *tree*, yaitu disebut dengan *FP - tree*. Penggunaan *FP - tree* membuat algoritma *FP - Growth* dapat langsung mengekstrak *frequent itemset* [ERW-09].

Algoritma *FP - Growth* menggunakan *divide* dan *conquer*, dimana membagi suatu permasalahan besar menjadi permasalahan – permasalahan yang lebih kecil (*divide*) dan pembagian dilakukan terus sampai ditemukan bagian masalah kecil yang mudah untuk dipecahkan (*conquer*) [FEB-09].

Fase dari algoritma *FP-growth* terdiri dari dua fase yaitu fase pembentukan *FP-tree* dan fase proses penggalian *FP-growth* [YIA-06]. Fase pembentukan *FP-tree* telah dijelaskan pada subbab *FP-tree*. Sedangkan fase dari proses penggalian *FP-growth* dijelaskan pada *Pseudocode 2.1*.

<p>Masukan : <i>FP-tree</i> yang telah dibentuk sebelumnya pada langkah awal.</p> <p>Keluaran : Sekumpulan dari pola <i>frequent</i></p> <p>Fungsi : panggil fungsi <i>fpGrowth(FP-Tree, null)</i></p> <p>Prosedur : <i>FP-Growth(Tree, α)</i></p> <pre> { (1) jika Tree mengandung lintasan(P) tunggal; (2) maka untuk setiap kombinasi(β) dari node-node dalam lintasan P lakukan (3) pembangkitan pola β yang mengandung item α dengan support = minsup dari node - node dalam kombinasi(β); (4) jika tidak untuk setiap a_i pada header Tree lakukan{ (5) pembangkitan pola kombinasi(β) = a_i yang mengandung item α dengan support = a_i.support; (6) bentuk conditional pattern base dari β; (7) kemudian bentuk conditional FP-tree dari β ($Tree_\beta$); (8) jika $Tree_\beta \neq \emptyset$ maka panggil <i>FP-Growth(Tree, β)</i> } }</pre>

Pseudocode 2. 1 Algoritma *FP - Growth* [YIA-06].

Algoritma *FP-growth* dimulai dengan dilakukan operasi pola *suffix* pada pola *frequent 1-itemset* dimana untuk masing – masing *suffix* dilakukan pembuatan *conditional pattern base*. *Conditional pattern base* merupakan suatu

himpunan yang terdiri dari node yang membentuk lintasan awal (*prefix path*) yang berakhir pada *suffix* tertentu. Dari *conditional pattern base* tersebut akhirnya dibangun *conditional FP-tree* untuk melakukan penambangan secara rekursif. Pola yang dihasilkan didapatkan melalui kombinasi dari pola akhiran (*suffix*) dengan pola *frequent* yang dibangkitkan dari setiap *conditional FP-tree* dengan lintasan tunggal [YIA-06].

Pada permasalahan data transaksi Tabel 2.1, maka operasi pola *suffix* dimulai dengan item yang berada pada urutan terakhir dari pola *frequent 1-itemset*, yaitu I5. Berdasarkan *FP-tree* yang telah terbangun pada Gambar 2.4 maka pola dengan *suffix* I5 ditemukan pada dua cabang *FP-tree* yaitu lintasan $\langle I2, I1, I5 : 1 \rangle$ dan lintasan $\langle I2, I1, I3, I5 : 1 \rangle$ sehingga lintasan awal $\langle I2, I1 : 1 \rangle$ dan $\langle I2, I1, I3 : 1 \rangle$ merupakan *conditional pattern base* dari I5. Kemudian *conditional FP-tree* yang terbentuk adalah lintasan $\langle I2, I1 : 2 \rangle$ karena I2 dan I1 yang nilai *support*-nya memenuhi *minimum support* sedangkan I3 tidak memenuhi *minimum support*. Hasil penambangan yang dilakukan pada *conditional FP-tree* tersebut menghasilkan pola *frequent* $\{\{I2, I5 : 2\}, \{I1, I5 : 2\}, \{I2, I1, I5 : 2\}\}$. Hasil dari fase proses penggalian *FP-growth* keseluruhan dari permasalahan ini dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Hasil Fase Proses Penggalian *FP-growth*.

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Pattern yang Dihasilkan
I5	$\{\{I2, I1 : 1\}, \{I2, I1, I3 : 1\}\}$	$\langle I2 : 2, I1 : 2 \rangle$	$\{I2, I5 : 2\}, \{I1, I5 : 2\}, \{I2, I1, I5 : 2\}$
I4	$\{\{I2, I1 : 1\}, \{I2 : 1\}\}$	$\langle I2 : 2 \rangle$	$\{I2, I4 : 2\}$
I3	$\{I2, I1 : 2\}, \{I2 : 2\}, \{I1 : 2\}\}$	$\langle I2 : 4, I1 : 2 \rangle, \langle I1 : 2 \rangle$	$\{I2, I3 : 4\}, \{I1, I3 : 4\}, \{I2, I1, I3 : 2\}$
I1	$\{\{I2 : 4\}\}$	$\langle I2 : 4 \rangle$	$\{I2, I1 : 4\}$

2.9. Lift Ratio

Lift ratio digunakan untuk mengevaluasi kuat tidaknya sebuah aturan asosiasi. *Lift ratio* adalah perbandingan antara *confidence* sebuah aturan dengan nilai *benchmark confidence*. *Benchmark confidence* adalah perbandingan antara

jumlah semua *item* yang menjadi *consequent* terhadap total jumlah transaksi [SAN-07]. Rumus *benchmark confidence* dapat dilihat pada Persamaan 2.5, sedangkan rumus *lift ratio* dapat dilihat pada Persamaan 2.6 [SAN-07].

$$\text{Benchmark Confidence} = \frac{N_c}{N} \quad (2.5)$$

Keterangan rumus 2.5 :

N_c = jumlah transaksi dengan *item* yang menjadi *consequent*

N = jumlah transaksi basis data

$$\text{Lift Ratio} = \frac{\text{Confidence}(A,C)}{\text{Benchmark Confidence}(A,C)} \quad (2.6)$$

Apabila nilai *lift ratio* lebih besar dari 1, maka menunjukkan adanya manfaat dari aturan tersebut. Lebih tinggi nilai *lift ratio* maka lebih besar kekuatan asosiasinya [SAN-07].

Lift ratio untuk aturan “ $X \rightarrow Y$ ” adalah perbandingan *confidence*(X,Y) dengan *benchmark confidence*(X,Y) yang didapatkan dari hasil perbandingan dari jumlah transaksi yang mengandung item Y terhadap jumlah transaksi pada basis data atau disebut juga *support*(Y).

2.10. Pengujian Akurasi Association Rule

Pengujian akurasi *association rule* yang terbentuk yaitu dengan membandingkan *association rule* hasil proses data mining pada data latih dengan data hasil pada data uji. Penghitungan akurasi *association rule* dilakukan dengan menggunakan persamaan 2.7 [SAN-07].

$$\text{Error} = \frac{|\text{support data uji} - \text{support data latih}|}{\text{support data latih}} \times 100\% \quad (2.7)$$

$$\text{Akurasi} = 100\% - \text{error} \quad (2.8)$$

BAB III

METODOLOGI DAN PERANCANGAN

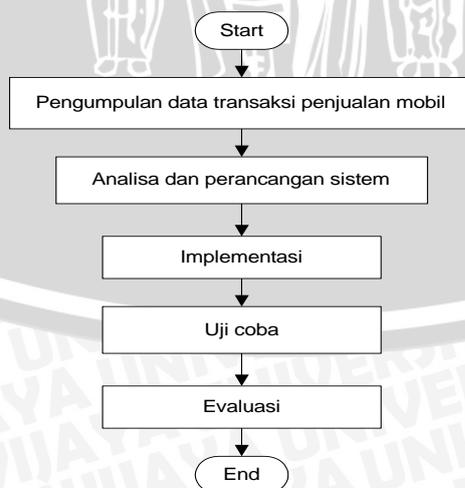
Pembahasan pada bab ini meliputi metode dan langkah – langkah perancangan yang dilakukan dalam penelitian untuk penggalian data transaksi penjualan mobil dengan menggunakan algoritma *FP-Growth*.

Langkah – langkah yang dilakukan dalam penelitian ini meliputi :

1. Melakukan studi literatur mengenai *association rule* dan algoritma *FP-Growth*.
2. Mengumpulkan data – data yang diperlukan dalam penelitian ini yaitu data transaksi penjualan mobil.
3. Menganalisa dan melakukan perancangan sistem untuk menggali *association rule* dari data transaksi penjualan mobil.
4. Mengimplementasikan rancangan yang dilakukan pada tahap sebelumnya menjadi sebuah perangkat lunak untuk analisa pola *association rule* dari data transaksi penjualan mobil.
5. Melakukan uji coba terhadap perangkat lunak menggunakan data transaksi penjualan mobil yang sudah disimpan dalam *database*.
6. Mengevaluasi hasil analisa yang dilakukan oleh sistem.

Langkah – langkah penelitian ini dapat digambarkan seperti pada Gambar

3.1.



Gambar 3. 1 Langkah – Langkah Penelitian

3.1. Pengumpulan Data

Pengumpulan data merupakan usaha untuk memperoleh data atau dokumen yang dibutuhkan dalam penelitian untuk diproses sesuai dengan kebutuhan. Pengumpulan data terdiri dari pengumpulan data yang digunakan sebagai bahan untuk studi literatur dan pengumpulan data yang akan digunakan sebagai studi kasus dalam penelitian ini. Oleh karena itu, metode pengumpulan data yang digunakan pada penelitian ini adalah studi literatur dan pengumpulan data lapangan.

3.1.1. Studi Literatur

Studi literatur merupakan cara pengumpulan data yang diperoleh dengan mengumpulkan berbagai sumber kepustakaan, baik berupa buku-buku, jurnal, laporan penelitian, dan lain sebagainya untuk ditelaah lebih lanjut sebagai bahan pendukung penelitian. Dalam penelitian ini, data yang dikumpulkan sebagai bahan studi literatur adalah data yang berisi literatur tentang *data* mining, metode *association rule*, dan algoritma *FP-Growth*.

3.1.2. Pengumpulan data lapangan

Pengumpulan data lapangan merupakan metode yang digunakan dalam penelitian ini. Metode ini digunakan untuk mendapatkan data yang akan digunakan sebagai studi kasus dalam penelitian ini. Data yang dimaksud adalah data transaksi penjualan mobil. Sumber data yang digunakan adalah laporan transaksi penjualan mobil di empat *dealer* resmi di wilayah Mataram (*dealer* Toyota, Honda, Suzuki, Daihatsu). Data yang diperoleh adalah data transaksi penjualan dari bulan Juni sampai bulan Desember tahun 2011. *Dataset* yang didapat terdiri dari 5 atribut, yaitu tanggal transaksi, nama pembeli, jenis kelamin, usia, dan jenis mobil yang dibeli. Jumlah data transaksi yang didapatkan dari masing – masing *dealer* tidak sama, yaitu :

- *Dealer* Toyota 339 data transaksi
- *Dealer* Honda 66 data transaksi
- *Dealer* Suzuki 713 data transaksi

- Dealer Daihatsu 409 data transaksi

3.2. Analisis dan Perancangan Sistem

3.2.1. Deskripsi Sistem

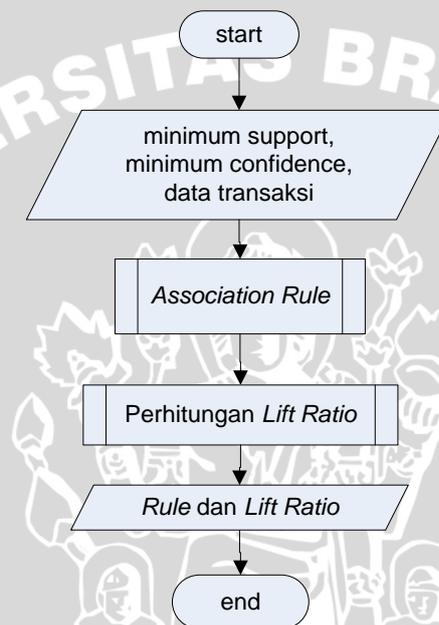
Sistem yang akan dibuat merupakan sistem yang dikembangkan untuk melakukan analisa terhadap data transaksi penjualan mobil untuk mengetahui pola hubungan antara jenis kelamin dan rentang umur terhadap jenis mobil yang dibeli. Parameter yang digunakan dalam analisa adalah nilai *minimum support*, nilai *minimum confidence*, usia, jenis kelamin dan jenis mobil.

Metode yang digunakan dalam sistem yang akan dibuat adalah metode *association rule* untuk melakukan *frequent itemset mining*. Sedangkan untuk mencari *frequent itemset* dan menentukan nilai *support* dalam sistem yang akan dibuat digunakan algoritma *FP-Growth*.

Dalam sistem yang akan dibuat, *itemset* berupa himpunan data transaksi yaitu kombinasi jenis kelamin, rentang umur dan jenis mobil sehingga *frequent itemset*-nya berupa himpunan data transaksi yang memiliki frekuensi kemunculan tinggi yaitu diperoleh dari hasil pemindaian basis data dan yang memenuhi nilai *minimum support* yang menjadi masukan dalam sistem. Setelah didapatkan *frequent itemset* dengan menerapkan algoritma *FP-Growth*, selanjutnya dalam sistem akan dilakukan perhitungan nilai *confidence* untuk setiap *frequent itemset*. Pada tahap ini, dilakukan proses pembangkitan *frequent itemset* menjadi *rule* dengan mengimplementasikan metode *association rule* pada sistem. Suatu *frequent itemset* akan dibangkitkan menjadi *rule* apabila memiliki nilai *confidence* yang memenuhi nilai *minimum confidence* (lebih besar atau sama dengan *minimum confidence*) yang menjadi masukan juga dalam sistem. *Frequent itemset* yang terbentuk menjadi *rule* akan ditampilkan sebagai hasil keluaran dalam bentuk kode jenis kelamin, kode rentang umur dan kode jenis mobil maupun keterangan jenis kelamin, rentang umur dan nama jenis mobil beserta nilai *confidence* dan nilai *lift ratio* tiap *rule* tersebut. *Rule* yang dihasilkan selanjutnya akan disimpan dalam file berformat xls untuk mempermudah melakukan proses analisis hasil.

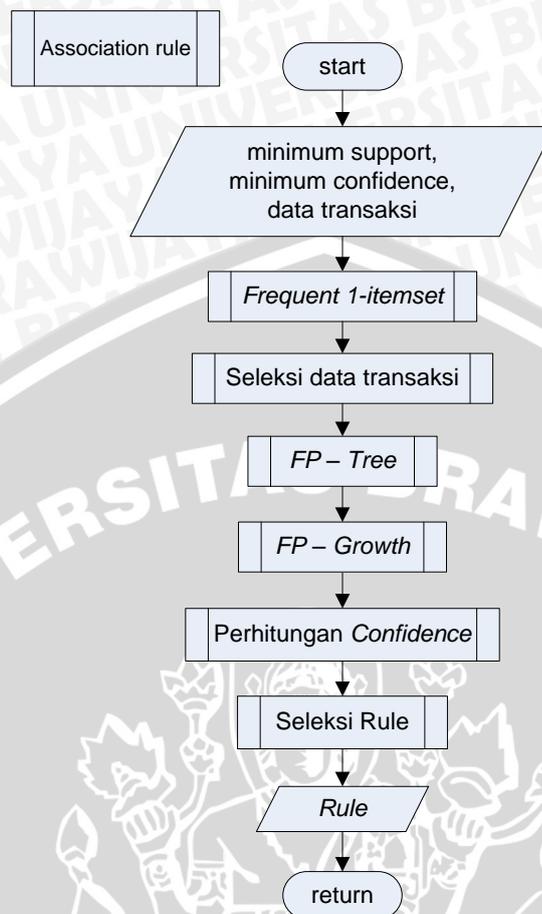
3.2.2. Rancangan Pembuatan Sistem

Dalam proses penelitian ini akan diimplementasikan algoritma *FP-Growth*. Proses yang dilakukan digambarkan dengan *flowchart* pada Gambar 3.2. Secara umum, input yang dibutuhkan sistem adalah minimum *suport*, minimum *confidence* dan data transaksi *dealer* yang ingin dilakukan *mining*. Proses awal dalam sistem adalah proses *association rule*. Selanjutnya dilakukan proses perhitungan *lift ratio* sehingga dihasilkan keluaran *rule* dan nilai *lift ratio rule*.



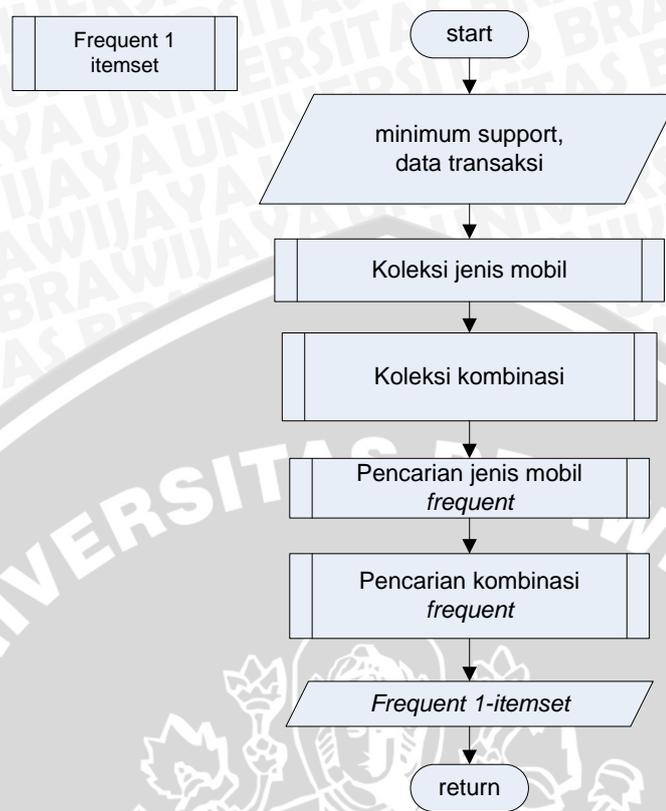
Gambar 3. 2 Flowchart Gambaran Umum Sistem

Proses *association rule* merupakan proses membangkitkan *rule* dari himpunan *frequent itemset* yang didapat. Himpunan *frequent itemset* yang dibangkitkan menjadi *rule* adalah *frequent itemset* yang memiliki nilai *confidence* lebih dari atau sama dengan nilai minimum *confidence*. Pada proses *association rule* sendiri membutuhkan data input berupa minimum *support*, minimum *confidence*, dan data transaksi *dealer* tertentu yang akan diproses melalui proses – proses yang terdiri dari proses *frequent 1-itemset*, proses seleksi data transaksi, proses *FP-Tree*, proses *FP-Growth*, proses perhitungan *confidence*, dan proses seleksi *rule*. Keluaran dari proses ini adalah *rule*. Keseluruhan proses *association rule* ini dapat dilihat pada Gambar 3.3.



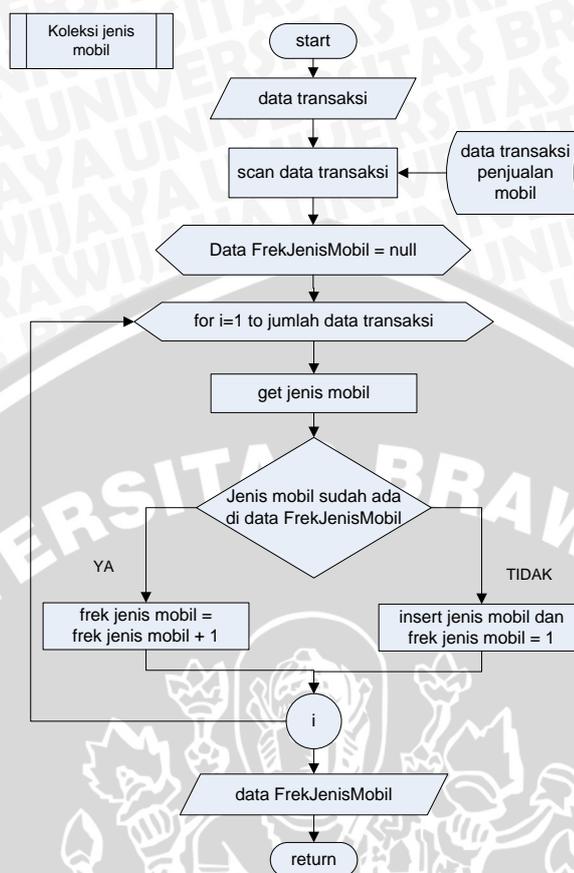
Gambar 3. 3 Flowchart proses association rule

Proses *frequent 1-itemset* merupakan proses untuk mendapatkan himpunan *frequent 1-itemset*. Himpunan *frequent 1-itemset* didapatkan dari item yang memiliki nilai *support* lebih dari atau sama dengan nilai minimum *support*. Item disini merupakan jenis mobil dan kombinasi antara jenis kelamin dengan rentang umur. Pada proses *frequent 1-itemset* sendiri membutuhkan data input berupa minimum *support* dan data transaksi *dealer* tertentu yang akan diproses melalui proses-proses yang terdiri dari proses pengkoleksian jenis mobil dan frekuensinya, proses pengkoleksian kombinasi jenis kelamin dan rentang umur beserta frekuensinya, proses pencarian jenis mobil yang *frequent*, dan proses pencarian jenis kelamin dan rentang umur yang *frequent*. Keluaran dari proses ini adalah *frequent 1-itemset*. Keseluruhan proses *frequent 1-itemset* ini dapat dilihat pada Gambar 3.4.



Gambar 3. 4 *Flowchart* proses frequent 1-itemset

Proses koleksi jenis mobil merupakan proses untuk mendapatkan jenis mobil dan frekuensi kemunculannya pada data transaksi. Pada proses ini dibutuhkan data input berupa data transaksi *dealer* tertentu. Proses dilakukan dengan melakukan *scan* data transaksi. Selanjutnya dilakukan inisialisasi *FrekJenisMobil* yang akan menyimpan data nama jenis mobil dan frekuensi jenis mobil. Setiap pembacaan data transaksi akan dilakukan pengecekan apakah data jenis mobil tersebut sudah ada pada data *FrekJenisMobil*, jika ada maka frekuensi ditambahkan 1 nilainya, sedangkan jika belum ada maka disimpan nama jenis mobil tersebut dan frekuensi diberi nilai 1. Keluaran dari proses ini adalah data *FrekJenisMobil*. Keseluruhan proses koleksi jenis mobil ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Flowchart proses koleksi jenis mobil

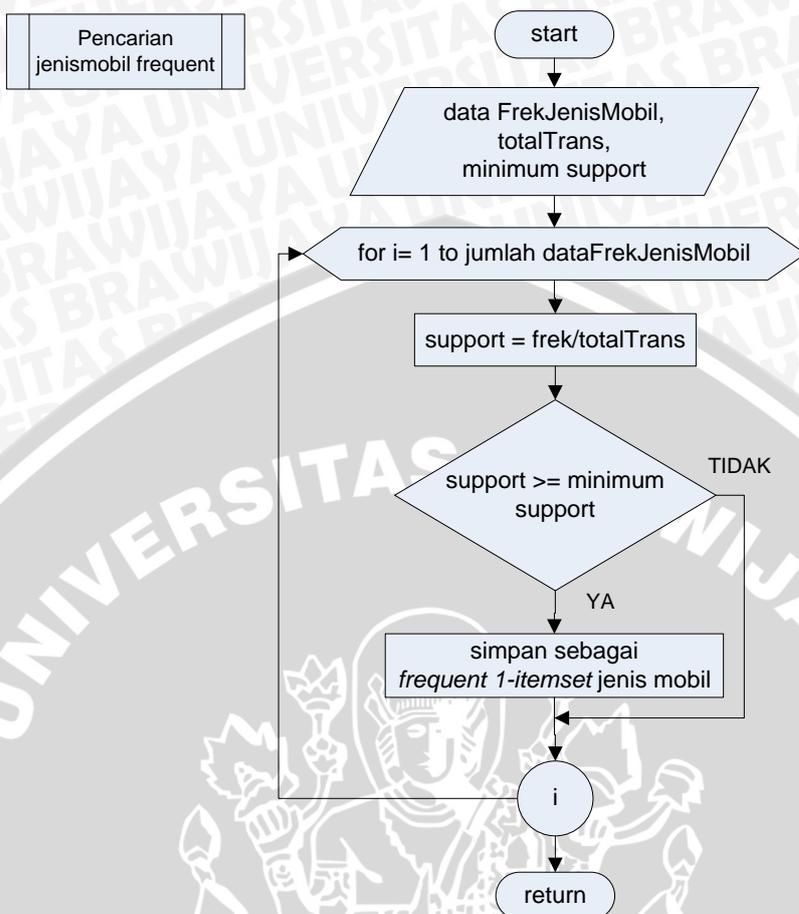
Proses koleksi kombinasi jenis kelamin dan rentang umur merupakan proses untuk mendapatkan kombinasi antara jenis kelamin dengan rentang umur dan frekuensi kemunculannya pada data transaksi. Pada proses ini dibutuhkan data input berupa data transaksi *dealer* tertentu. Proses dilakukan dengan melakukan *scan* data transaksi. Selanjutnya dilakukan inialisasi *FrekKombinasi* yang akan menyimpan data nama kombinasi dan frekuensi kombinasi. Setiap pembacaan data transaksi akan dilakukan pengecekan apakah data kombinasi tersebut sudah ada pada data *FrekKombinasi*, jika ada maka frekuensi ditambahkan 1 nilainya, sedangkan jika belum ada maka disimpan nama jenis kombinasi tersebut dan frekuensi diberi nilai 1. Keluaran dari proses ini adalah data *FrekKombinasi*. Keseluruhan proses koleksi kombinasi jenis kelamin dan rentang umur ini dapat dilihat pada Gambar 3.6.

Proses pencarian jenis mobil yang *frequent* merupakan proses untuk menyeleksi data jenis mobil yang memenuhi nilai minimum *support*. Pada proses ini dibutuhkan data input berupa data FrekJenisMobil, total transaksi, minimum *support*. Proses dilakukan dengan melakukan pembacaan setiap data FrekJenisMobil. Setiap pembacaan data FrekJenisMobil akan dilakukan perhitungan nilai *support* yang didapat dari nilai frekuensi dibagi dengan total transaksi. Kemudian dilakukan pengecekan apakah nilai *support* lebih dari atau sama dengan nilai minimum *support*, jika lebih dari atau sama dengan maka disimpan sebagai *frequent 1-itemset* jenis mobil. Keseluruhan proses pencarian jenis mobil yang *frequent* ini dapat dilihat pada Gambar 3.7.

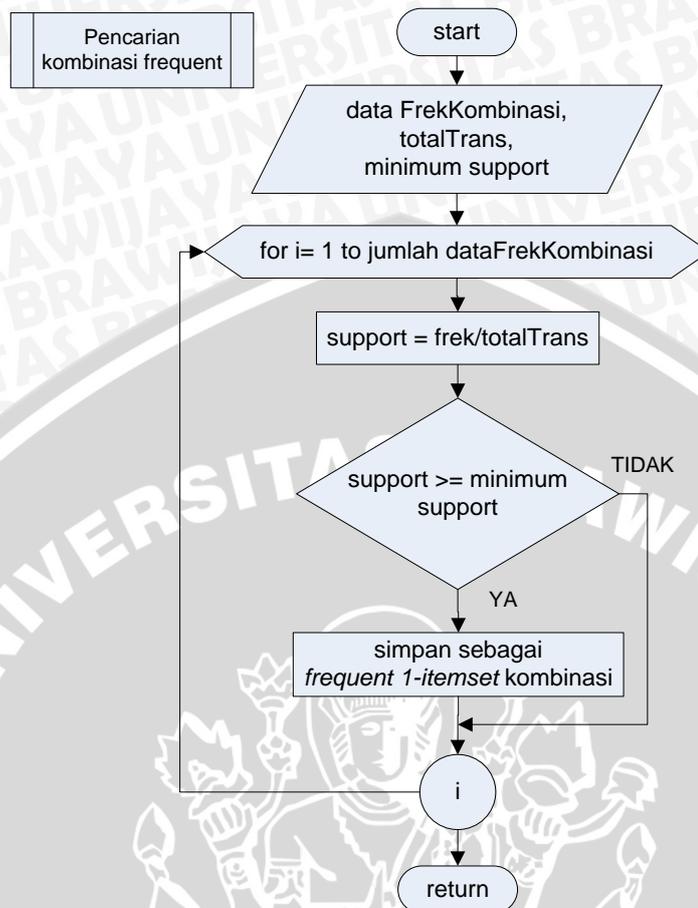
Proses pencarian kombinasi yang *frequent* merupakan proses untuk menyeleksi data kombinasi antara jenis kelamin dan rentang umur yang memenuhi nilai minimum *support*. Pada proses ini dibutuhkan data input berupa data FrekKombinasi, total transaksi, minimum *support*. Proses dilakukan dengan melakukan pembacaan setiap data FrekKombinasi. Setiap pembacaan data FrekKombinasi akan dilakukan perhitungan nilai *support* yang didapat dari nilai frekuensi dibagi dengan total transaksi. Kemudian dilakukan pengecekan apakah nilai *support* lebih dari atau sama dengan nilai minimum *support*, jika lebih dari atau sama dengan maka disimpan sebagai *frequent 1-itemset* kombinasi. Keseluruhan proses pencarian kombinasi yang *frequent* ini dapat dilihat pada Gambar 3.8.



Gambar 3. 6 Flowchart proses koleksi kombinasi antara jenis kelamin dan rentang umur

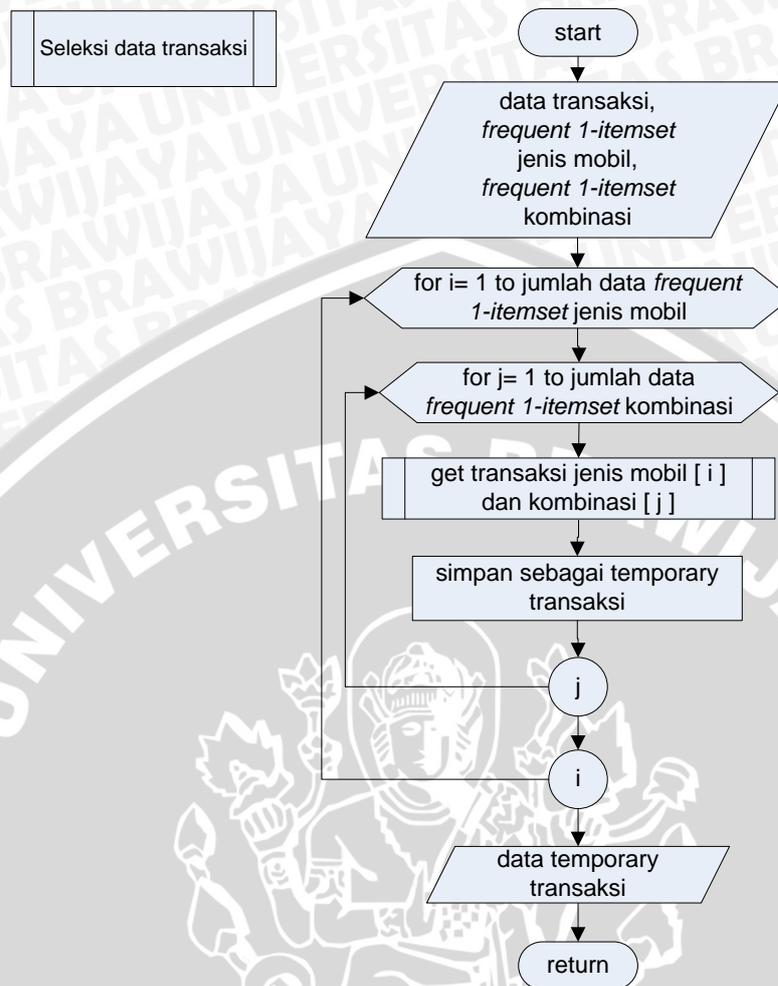


Gambar 3. 7 Flowchart proses pencarian jenis mobil yang frequent



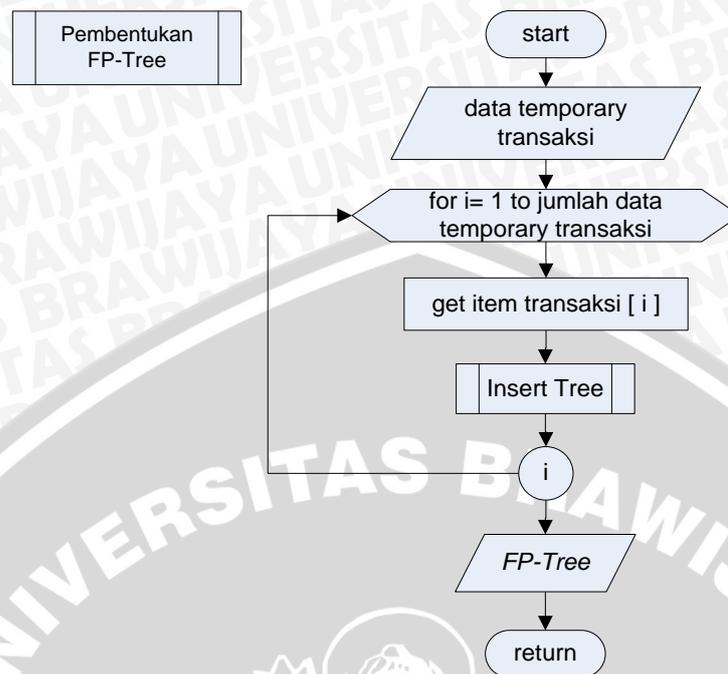
Gambar 3. 8 Flowchart proses pencarian kombinasi yang frequent.

Proses seleksi data transaksi merupakan proses untuk menyeleksi data transaksi yang memiliki jenis mobil dan kombinasi jenis kelamin dan rentang umur yang menjadi *frequent 1-itemset*. Pada proses ini dibutuhkan data input berupa data transaksi, *frequent 1-itemset* jenis mobil, *frequent 1-itemset* kombinasi. Proses dilakukan dengan melakukan pembacaan setiap data *frequent 1-itemset* jenis mobil sebagai *i* dan *frequent 1-itemset* kombinasi sebagai *j*. Kemudian untuk setiap *i* dan *j* dilakukan pencarian pada data transaksi yang memiliki item – item tersebut. Transaksi yang memiliki kedua item *i* dan *j* tersebut disimpan sebagai *temporary* transaksi. Keluaran yang dihasilkan dari proses ini adalah data *temporary* transaksi. Keseluruhan proses seleksi data transaksi ini dapat dilihat pada Gambar 3.9.



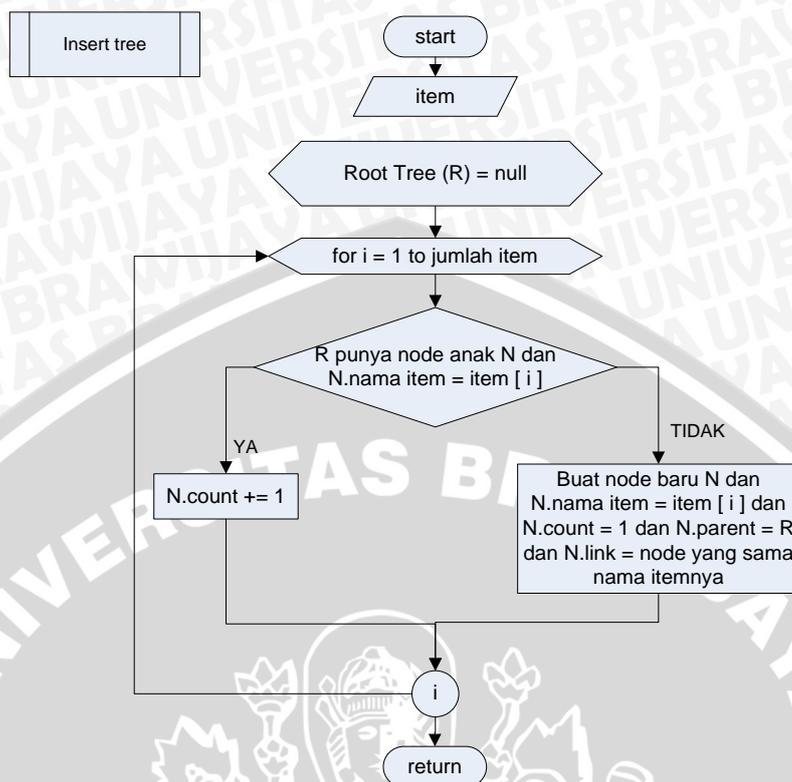
Gambar 3. 9 Flowchart proses seleksi data transaksi

Proses pembentukan *FP-Tree* membutuhkan data input berupa data *temporary* transaksi. Proses dilakukan dengan cara item yang berupa kombinasi jenis kelamin dengan rentang buku, dan jenis mobil pada data *temporary* transaksi yang dibaca dimasukkan ke dalam *FP-Tree* melalui proses *insert tree*. Keluaran yang dihasilkan dari proses ini adalah *FP-Tree*. Keseluruhan proses pembentukan *FP-Tree* ini digambarkan pada Gambar 3.10.



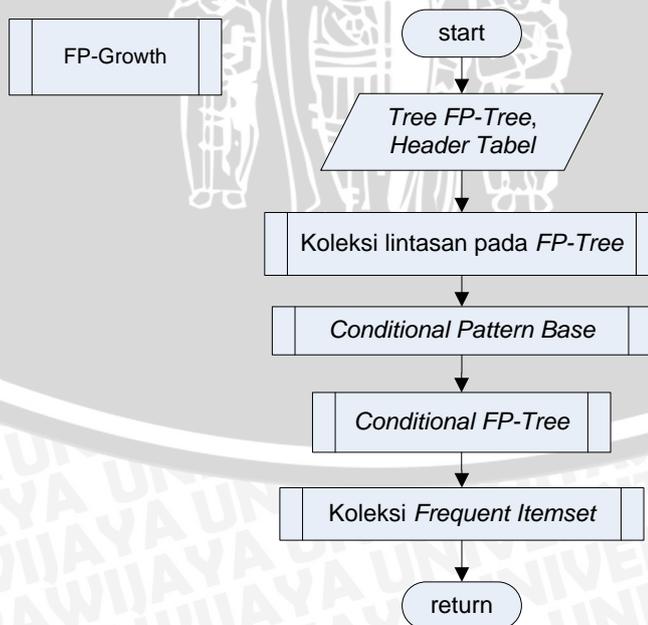
Gambar 3. 10 Flowchart proses pembentukan *FP-tree*

Proses *insert tree* merupakan proses yang digunakan untuk memasukkan item berupa kombinasi antara jenis kelamin dengan rentang umur, dan jenis mobil ke dalam *FP-Tree*. Pada proses ini dibutuhkan data input berupa item yang akan dimasukkan. Proses dilakukan dengan melakukan inisialisasi awal *root tree* adalah *null*. Setiap pembacaan data item dilakukan pengecekan apakah *root* mempunyai *node* anak dengan nama item yang sama dengan item yang sedang dibaca. Jika ada maka *count* pada *node* tersebut ditambahkan 1 nilainya, sedangkan jika tidak maka dibuat *node* baru dengan nama item adalah item yang dibaca dan *parent node* tersebut adalah *root* saat ini, sedangkan *link node* adalah *node* lain yang sama nama itemnya. Keseluruhan proses *insert tree* ini dapat dilihat pada Gambar 3.11.



Gambar 3. 11 Flowchart proses insert tree

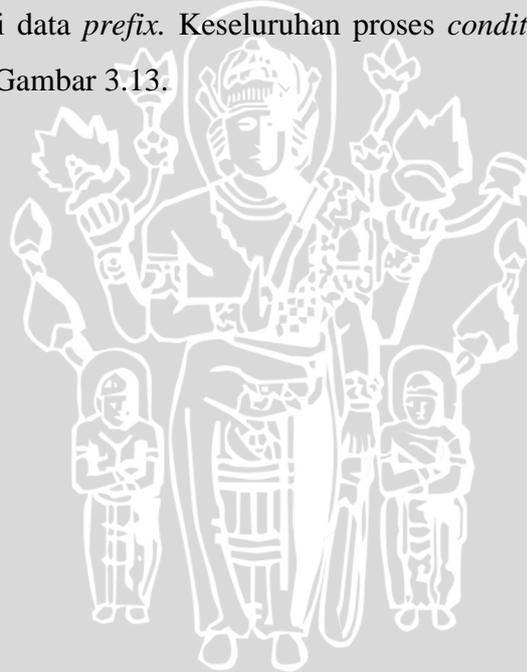
proses ini berupa *FP-Tree* dan *Header Tabel*. Keseluruhan proses *FP-Growth* ini dapat dilihat pada Gambar 3.12.

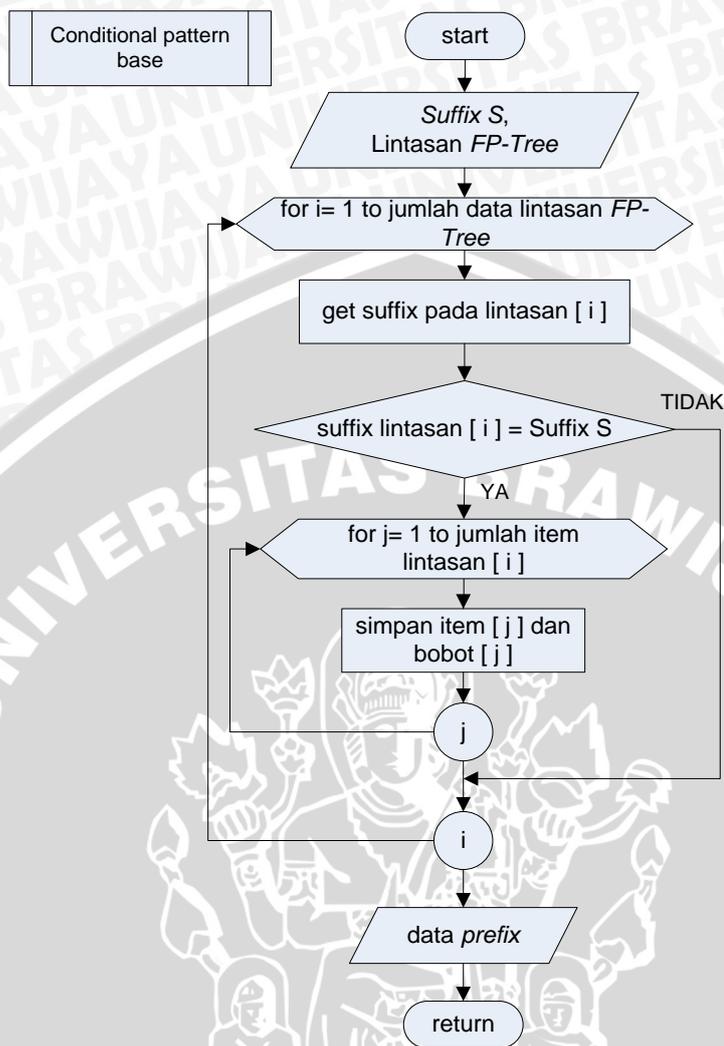


Gambar 3. 12 Flowchart proses algoritma FP-Growth

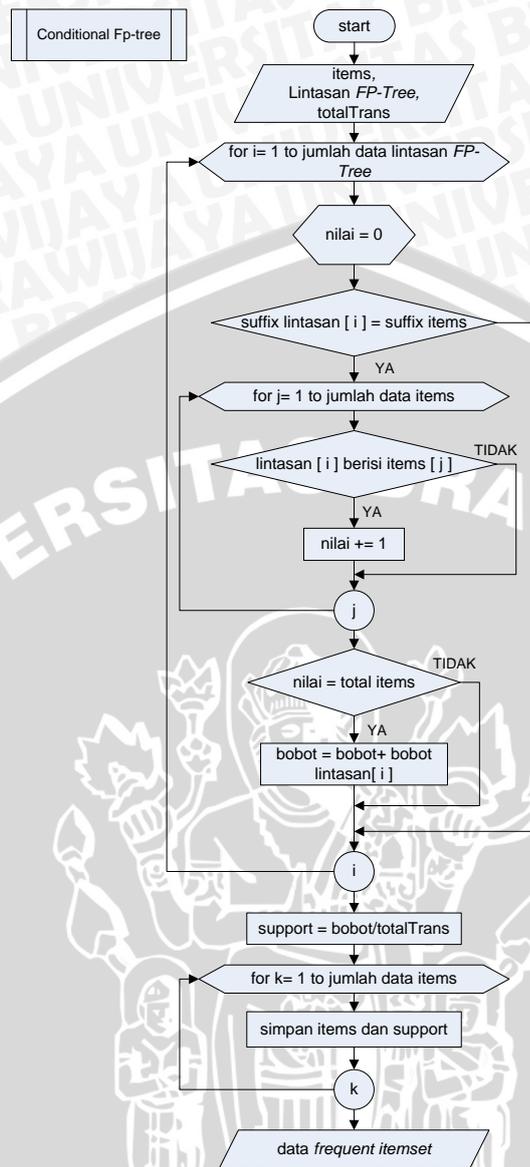
Proses *FP-Growth* merupakan proses yang digunakan untuk *mining frequent itemset*. Pada proses ini akan dilakukan proses – proses yang terdiri dari proses koleksi lintasan pada *FP-Tree*, proses *conditional pattern base*, proses *conditional FP-Tree*, dan proses koleksi *frequent itemset*. Data input yang dibutuhkan pada

Proses *conditional pattern base* merupakan proses yang digunakan untuk koleksi lintasan tiap *suffix* dimana item yang menjadi *suffix* adalah jenis mobil. Pada proses ini dibutuhkan data input berupa *suffix S* dan lintasan *FP-Tree*. Proses dilakukan dengan melakukan pembacaan pada setiap data lintasan *FP-Tree* yang selanjutnya dilakukan pengecekan apakah *suffix* pada lintasan yang dibaca sama dengan *suffix S*. Jika sama maka item dan bobot pada lintasan *FP-Tree* tersebut akan disimpan sebagai data *prefix*. Keseluruhan proses *conditional pattern base* ini digambarkan pada Gambar 3.13.



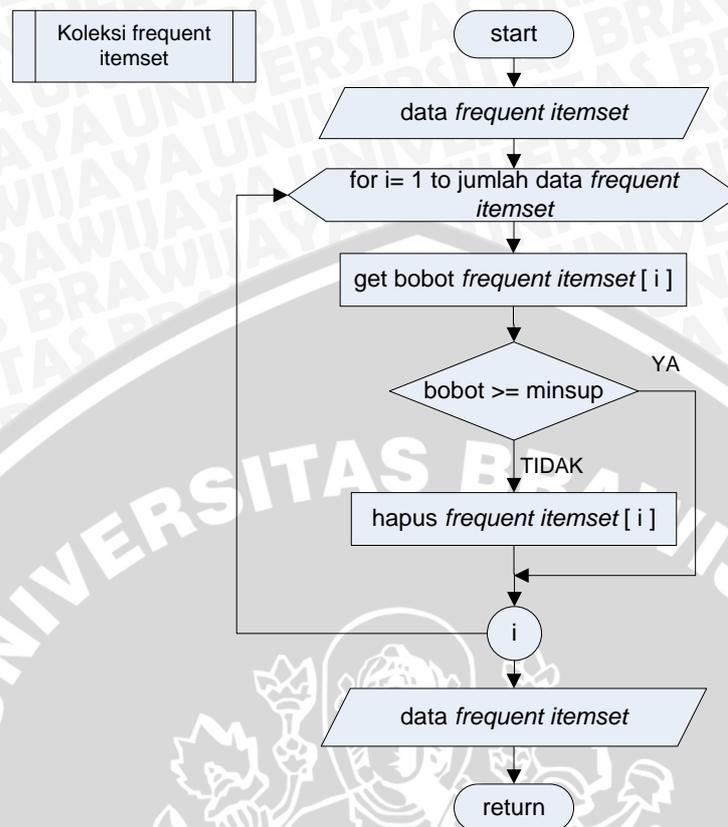


Gambar 3. 13 Flowchart proses conditional pattern base.



Gambar 3. 14 Flowchart proses conditional *FP-tree*

Proses *conditional FP-Tree* merupakan proses untuk mengetahui *support* tiap item dengan lintasan yang sama. Pada proses ini dibutuhkan data input berupa items, lintasan *FP-Tree*, dan total transaksi. Keluaran dari proses ini adalah data *frequent itemset*. Keseluruhan proses *conditional FP-Tree* dijelaskan pada Gambar 3.14.



Gambar 3. 15 Flowchart proses koleksi *frequent itemset*

Proses koleksi *frequent itemset* merupakan proses yang digunakan untuk mengkoleksi *frequent itemset* yang memenuhi nilai minimum *support*. Pada proses ini dibutuhkan data input berupa data *frequent itemset*. Proses dilakukan dengan melakukan pembacaan pada data *frequent itemset*. Setiap pembacaan data *frequent itemset* dilakukan pengecekan apakah bobot *frequent itemset* lebih dari atau sama dengan minimum *support*. Jika tidak maka dihapus sebagai *frequent itemset*. Keseluruhan proses ini dapat dilihat pada Gambar 3.15.

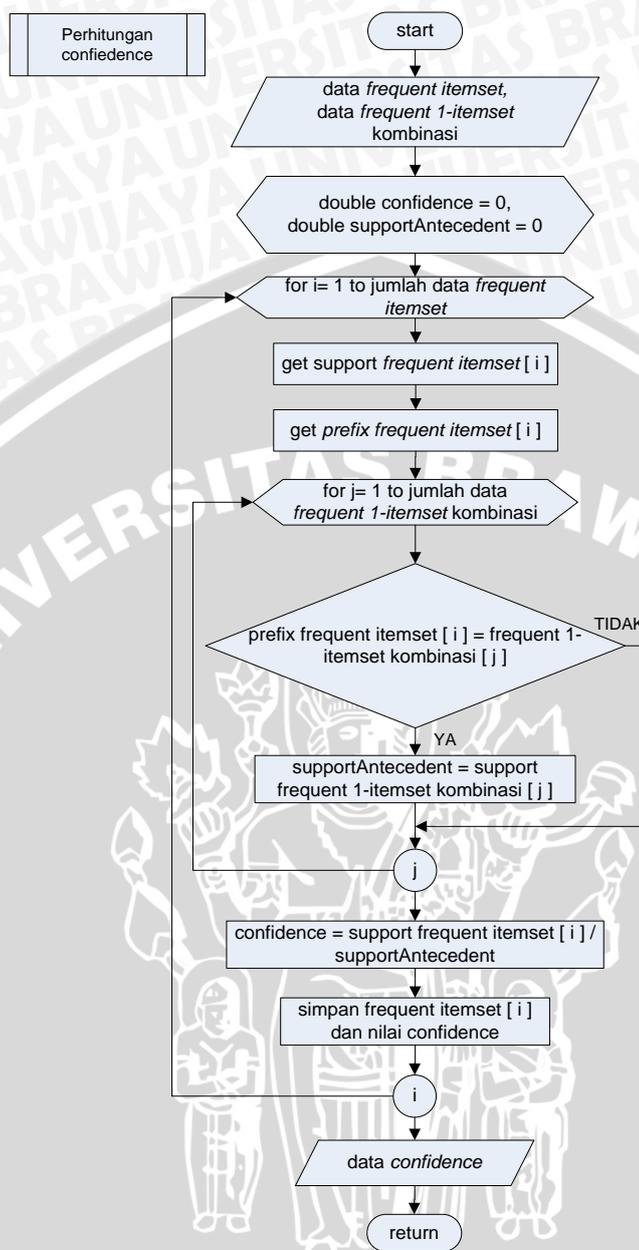
Proses perhitungan *confidence* merupakan proses yang digunakan untuk menghitung *confidence frequent itemset*. Pada proses ini dibutuhkan data input data *frequent itemset* dan data *frequent 1-itemset* kombinasi. Proses dilakukan dengan mendapatkan terlebih dahulu nilai *support antecedent*. *Support antecedent* merupakan *support frequent 1-itemset* kombinasi yang sama dengan *prefix frequent itemset* yang dibaca. Kemudian nilai *confidence* didapatkan dari nilai *support frequent itemset* tersebut dibagi dengan nilai *support antecedent*.

Keluaran yang dihasilkan dari proses ini adalah data *confidence*. Keseluruhan proses perhitungan *confidence* ini dapat dilihat pada Gambar 3.16.

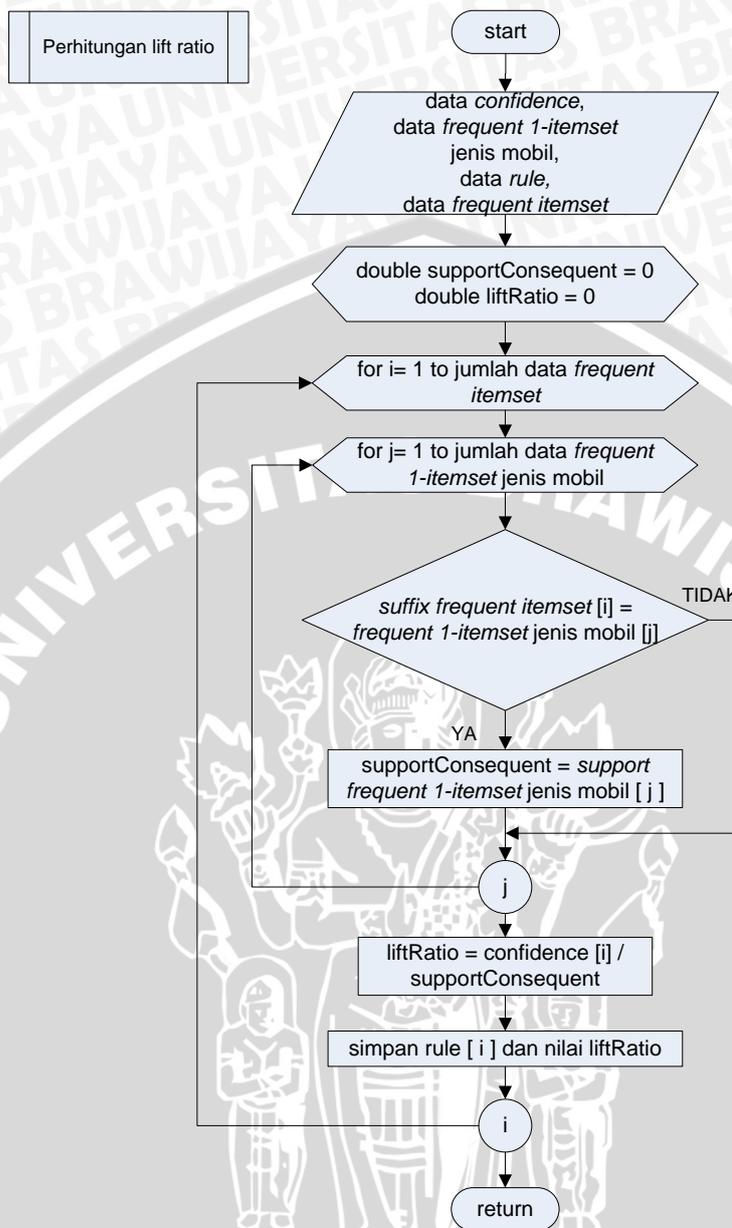
Proses perhitungan *lift ratio* merupakan proses yang digunakan untuk menghitung nilai *lift ratio* tiap *rule*. Pada proses ini dibutuhkan data input berupa data *confidence*, data *frequent itemset*, data *rule*, dan data *frequent 1-itemset* jenis mobil. Proses dilakukan dengan mendapatkan nilai *support consequent* terlebih dahulu. *Support consequent* didapatkan dari *frequent 1-itemset* jenis mobil yang sama dengan *suffix frequent itemset* yang sedang dibaca. Kemudian dilakukan perhitungan nilai *lift ratio* yang didapatkan dari nilai *confidence frequent itemset* dibagi dengan nilai *support consequent*. Keseluruhan proses perhitungan *lift ratio* ini dapat dilihat pada Gambar 3.17.

Proses seleksi *rule* merupakan proses yang digunakan untuk menyeleksi *rule* yang memenuhi minimum *confidence*. Pada proses ini dibutuhkan data input berupa data *confidence* dan data minimum *confidence*. Proses dilakukan dengan melakukan pembacaan setiap data *confidence* yang selanjutnya dilakukan pengecekan apakah data *confidence* yang dibaca lebih dari atau sama dengan nilai minimum *confidence*. Jika ya maka disimpan sebagai *rule*. Keseluruhan proses seleksi *rule* ini dapat dilihat pada Gambar 3.18.

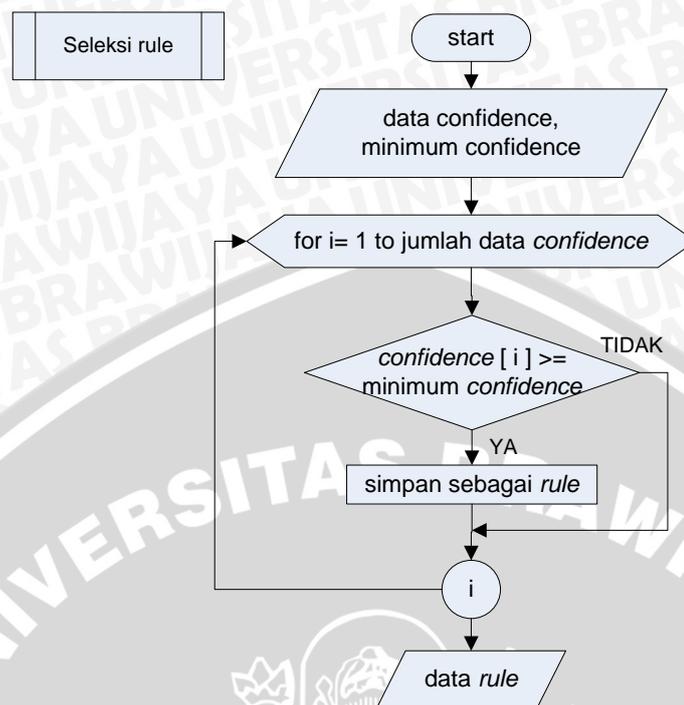




Gambar 3. 16 Flowchart proses perhitungan confidence



Gambar 3. 17 Flowchart proses perhitungan lift ratio



Gambar 3. 18 Flowchart proses seleksi rule

3.3. Perhitungan Manual

Perhitungan manual dilakukan pada sampel data yang diambil acak dari data transaksi penjualan mobil *dealer* resmi Daihatsu sebanyak 20 transaksi. Nilai minimum *support* yang digunakan pada proses perhitungan manual adalah 10% sedangkan nilai minimum *confidence* yang digunakan adalah 50%.

Dalam melakukan perhitungan manual ini, sebelum dilakukan penerapan *FP-growth*, sampel data diproses terlebih dahulu sampai bentuk data sesuai dengan yang dibutuhkan. Proses tersebut terdiri dari proses seleksi, proses *preprocessing*, dan proses transformasi. Proses seleksi telah dilakukan pada pemilihan sampel data secara acak, sehingga dalam perhitungan manual ini proses yang dijelaskan dimulai dari proses *preprocessing*.

Langkah awal dari proses penyesuaian data adalah tahap *preprocessing*. Tahap *preprocessing* dilakukan dengan menghapus kolom nama *customer*, tanggal transaksi dan tanggal lahir. Sampel data yang telah melalui tahap *preprocessing* data dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Sampel Data Transaksi Penjualan Mobil Hasil *Preprocessing*

Id	Jenis mobil	Jenis kelamin	Rentang umur pembeli			
			20-30	31-40	41-50	> 50
1	LUXIO 1.5 D M/T	W	1			
2	LUXIO 1.5 D M/T	P		1		
3	LUXIO 1.5 M M/T	W	1			
4	SIRION 1.3 FMC MT	P			1	
5	SIRION 1.3 FMC MT	W			1	
6	SIRION 1.3 FMC MT	W			1	
7	TERIOS TX MT ADV MC	P	1			
8	TERIOS TX MT ADV MC	P	1			
9	TERIOS TX MT ADV MC	P	1			
10	TERIOS TX MT ADV MC	P	1			
11	TERIOS TX MT ADV MC	P				1
12	XENIA FMC R MT 1.3 F	P			1	
13	XENIA FMC R MT 1.3 F	W			1	
14	XENIA FMC R MT 1.3 F	P			1	
15	XENIA FMC R MT 1.3 F	P			1	
16	XENIA FMC R MT 1.3 F	P			1	
17	XENIA FMC R MT 1.3 S	P		1		
18	XENIA FMC R MT 1.3 S	P				1
19	XENIA FMC R MT 1.3 S	P		1		
20	XENIA FMC R MT 1.3 S	W			1	

Langkah kedua, dilakukan pembacaan pada data transaksi yang telah dilakukan *preprocessing* untuk menghitung frekuensi kemunculan tiap jenis mobil dan kombinasi jenis kelamin dengan rentang umur seperti pada Tabel 3.2. dan Tabel 3.3. Jenis kelamin dengan rentang umur dikombinasi atau dianggap satu kesatuan yang harus muncul bersamaan dalam kasus ini karena target aturan yang ingin didapatkan adalah asosiasi jenis kelamin dan rentang umur terhadap jenis mobil yang dibeli.

Tabel 3. 2 Frekuensi Kemunculan Tiap Jenis Mobil

Jenis mobil	Kode Jenis Mobil	Frekuensi
LUXIO 1.5 D M/T	A	2
LUXIO 1.5 M M/T	B	1
SIRION 1.3 FMC MT	C	3
TERIOS TX MT ADV MC	D	5
XENIA FMC R MT 1.3 F	E	5
XENIA FMC R MT 1.3 S	F	4

Tabel 3. 3 Frekuensi Kemunculan Tiap Kombinasi Antara Jenis Kelamin dengan Rentang Umur

Jenis Kelamin	Rentang Umur	Kode Jenis Kelamin dan Rentang Umur	Frekuensi
Pria	20-30	P1	4
Pria	31-40	P2	3
Pria	41-50	P3	5
Pria	> 50	P4	2
Wanita	20-30	W1	2
Wanita	31-40	W2	0
Wanita	41-50	W3	4
Wanita	> 50	W4	0

Langkah ketiga adalah perhitungan *support* pada kandidat *frequent 1-itemset* yang dapat dilihat pada Tabel 3.4 untuk *support* kandidat *frequent 1-itemset* jenis mobil dan Tabel 3.5 untuk *support* kandidat *frequent 1-itemset* kombinasi jenis kelamin dengan rentang umur.

$$\begin{aligned}
 \text{Support } \{A\} &= \frac{2}{20} = 0.10 \\
 \text{Support } \{B\} &= \frac{1}{20} = 0.05 \\
 \text{Support } \{C\} &= \frac{3}{20} = 0.15 \\
 \text{Support } \{D\} &= \frac{5}{20} = 0.25 \\
 \text{Support } \{E\} &= \frac{4}{20} = 0.20 \\
 \text{Support } \{F\} &= \frac{5}{20} = 0.25 \\
 \text{Support } \{P1\} &= \frac{4}{20} = 0.20 \\
 \text{Support } \{P2\} &= \frac{3}{20} = 0.15 \\
 \text{Support } \{P3\} &= \frac{5}{20} = 0.25 \\
 \text{Support } \{P4\} &= \frac{2}{20} = 0.10 \\
 \text{Support } \{W1\} &= \frac{2}{20} = 0.10 \\
 \text{Support } \{W2\} &= \frac{0}{20} = 0.00 \\
 \text{Support } \{W3\} &= \frac{4}{20} = 0.20 \\
 \text{Support } \{W4\} &= \frac{0}{20} = 0.00
 \end{aligned}$$

Tabel 3. 4 Support Kandidat *Frequent 1-Itemset* Jenis Mobil

Jenis mobil	Kode Jenis Mobil	Frekuensi	Support
LUXIO 1.5 D M/T	A	2	0.10
LUXIO 1.5 M M/T	B	1	0.05
SIRION 1.3 FMC MT	C	3	0.15
TERIOS TX MT ADV MC	D	5	0.25
XENIA FMC R MT 1.3 F	E	4	0.20
XENIA FMC R MT 1.3 S	F	5	0.25

Tabel 3. 5 *Support* Kandidat *Frequent 1-Itemset* Kombinasi Antara Jenis Kelamin dengan Rentang Umur

Jenis Kelamin	Rentang Umur	Kode Jenis Kelamin dan Rentang Umur	Frekuensi	Support
Pria	20-30	P1	4	0.20
Pria	31-40	P2	3	0.15
Pria	41-50	P3	5	0.25
Pria	> 50	P4	2	0.10
Wanita	20-30	W1	2	0.10
Wanita	31-40	W2	0	0.00
Wanita	41-50	W3	4	0.20
Wanita	> 50	W4	0	0.00

Langkah keempat adalah pencarian *frequent 1-itemset* dari kandidat *frequent 1-itemset* baik jenis mobil maupun jenis kelamin dan rentang umur. Pada kandidat *frequent 1-itemset* jenis mobil, kode jenis mobil B tidak *frequent* karena nilai *support* yang dimiliki kode jenis mobil B kurang dari nilai minimum *support*. Sedangkan pada kandidat *frequent 1-itemset* kode jenis kelamin dan rentang umur W2, W4 yang tidak *frequent*. Kode jenis mobil A,C,D,E,F merupakan kandidat jenis mobil yang *frequent* atau terpilih menjadi *frequent 1-itemset* dan dapat dilihat pada Tabel 3.6, sedangkan P1, P2, P3, P4, W1, W3 merupakan kandidat kode jenis kelamin dan rentang umur yang *frequent* atau terpilih menjadi *frequent 1-itemset* dan dapat dilihat pada Tabel 3.7.

Tabel 3. 6 *Frequent 1-Itemset* Jenis Mobil

<i>Frequent 1-Itemset</i>	<i>support</i>
A	0.10
C	0.15
D	0.25
E	0.20
F	0.25

Tabel 3. 7 *Frequent 1-Itemset* Jenis Kelamin dan Rentang Umur.

Frequent 1-itemset	Support
P1	0.20
P2	0.15
P3	0.25
P4	0.10
W1	0.10
W3	0.20

Langkah kelima adalah proses seleksi data transaksi yang memiliki baik *frequent 1-itemset* jenis mobil maupun *frequent 1-itemset* jenis kelamin dan rentang umur. Perubahan pada sampel data transaksi dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Sampel Data Transaksi Hasil Proses Seleksi

Id	Jenis mobil	Jenis kelamin	Rentang umur			
			20-30	31-40	41-50	> 50
1	LUXIO 1.5 D M/T	W	1			
2	LUXIO 1.5 D M/T	P		1		
4	SIRION 1.3 FMC MT	P			1	
5	SIRION 1.3 FMC MT	W			1	
6	SIRION 1.3 FMC MT	W			1	
7	TERIOS TX MT ADV MC	P	1			
8	TERIOS TX MT ADV MC	P	1			
9	TERIOS TX MT ADV MC	P	1			
10	TERIOS TX MT ADV MC	P	1			
11	TERIOS TX MT ADV MC	P				1
12	XENIA FMC R MT 1.3 S	W			1	
13	XENIA FMC R MT 1.3 F	P			1	
14	XENIA FMC R MT 1.3 F	P			1	
15	XENIA FMC R MT 1.3 F	P			1	
16	XENIA FMC R MT 1.3 F	P			1	
17	XENIA FMC R MT 1.3 S	P		1		
18	XENIA FMC R MT 1.3 S	P				1
19	XENIA FMC R MT 1.3 S	P		1		
20	XENIA FMC R MT 1.3 S	W			1	

Langkah keenam adalah tahap transformasi data dimana data hasil *preprocessing* data diubah bentuk sesuai dengan kebutuhan. Tahap transformasi

yang dilakukan adalah mengubah bentuk tabel transaksi sehingga memudahkan proses *data mining*. Tabel diubah menjadi terdiri dua kolom yaitu id transaksi dan transaksi. Kolom id transaksi menyimpan informasi nomor id transaksi, sedangkan kolom transaksi menyimpan informasi transaksi yang berupa kombinasi jenis kelamin pembeli dengan rentang umur pembeli dan kode jenis mobil. Pengurutan item transaksi jenis mobil di belakang disebabkan karena target yang ingin diketahui adalah jenis mobil apa yang dibeli dengan karakteristik pembeli berdasarkan jenis kelamin pembeli dan rentang umur pembeli. Tabel hasil transformasi dapat dilihat pada Tabel 3.9.

Tabel 3. 9 Tabel Transformasi Data Transaksi

Id	Transaksi
1	P2, A
2	W1, A
4	P3, C
5	W3, C
6	W3, C
7	P1, D
8	P1, D
9	P1, D
10	P1, D
11	P4, D
12	W3, F
13	P3, E
14	P3, E
15	P3, E
16	P3, E
17	P2, F
18	P2, F
19	P4, F
20	W3, F

Langkah ketujuh adalah setiap transaksi dari sampel data transaksi dibuat *FP-tree* yang disajikan pada Gambar 3.19 sampai Gambar 3.27. Pada pembuatan awal *root* diberikan nilai *null*. Pada *header table*, *head of table link* untuk semua kode item bernilai *null*.

Itemset	Head of table link
P1	<i>null</i>
P2	<i>null</i>
P3	<i>null</i>
P4	<i>null</i>
W1	<i>null</i>
W3	<i>null</i>
A	<i>null</i>
C	<i>null</i>
D	<i>null</i>
E	<i>null</i>
F	<i>null</i>

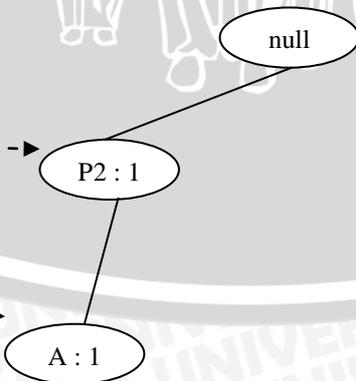


Gambar 3. 19 FP-Tree Awal

Pembacaan pertama yaitu untuk transaksi dengan id transaksi 1, transaksi yang terjadi adalah {P2, A}. P2 masuk terlebih dahulu ke dalam *FP-Tree* sehingga dibuat *node* baru dengan label P2 yang merupakan *node* anak dari *root* dan *count support* diberi nilai 1. Selain itu *itemset* P2 pada *header* tabel dibuat *link* yang menunjukkan *node* dimana informasi mengenai P2 disimpan. Selanjutnya *item* transaksi A masuk ke dalam *FP-Tree*, karena *node* P2 belum memiliki *node* anak berlabel A maka dibuat *node* baru dengan label A dan *count support node* A diberi nilai 1. Begitu pula dengan *itemset* A pada *header* tabel dibuat *link* yang merujuk ke *node* A tersebut. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 1 dapat dilihat pada Gambar 3.20.

Header Tabel

Itemset	Head of table link
P1	<i>null</i>
P2	<i>null</i>
P3	<i>null</i>
P4	<i>null</i>
W1	<i>null</i>
W3	<i>null</i>
A	<i>null</i>
C	<i>null</i>
D	<i>null</i>
E	<i>null</i>
F	<i>null</i>

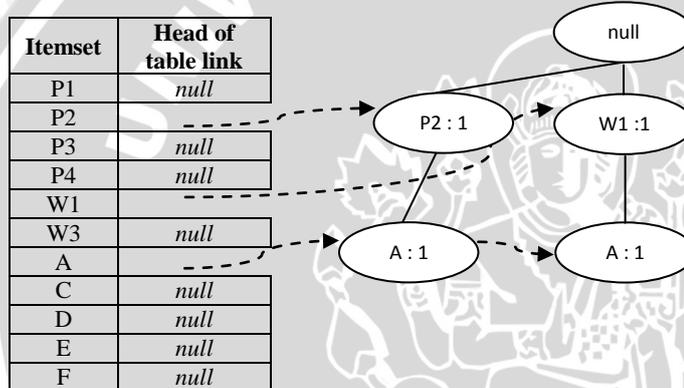


Gambar 3. 20 FP-Tree Pembacaan id Transaksi 1

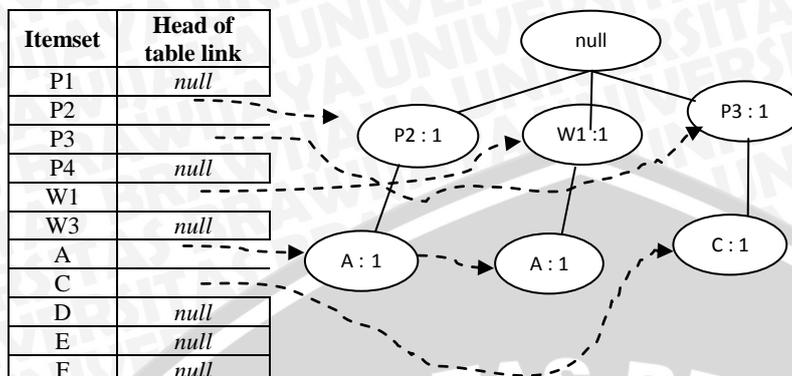


Pembacaan kedua yaitu untuk transaksi dengan id transaksi 2, transaksi yang terjadi adalah {W1, A}. Dimana W1 masuk terlebih dahulu ke dalam *FP-Tree* sehingga dibuat *node* baru dengan label W1 yang merupakan *node* anak dari *root* dan *count support* diberi nilai 1. Setelah itu *itemset* W1 pada *header* tabel dibuat *link* yang menunjukkan *node* dimana informasi mengenai W1 disimpan. Selanjutnya *item* transaksi A masuk ke dalam *FP-Tree*, karena *node* W1 belum memiliki *node* anak berlabel A maka dibuat *node* baru dengan label A dan *count support node* A diberi nilai 1. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 2 dapat dilihat pada Gambar 3.21.

Header Tabel

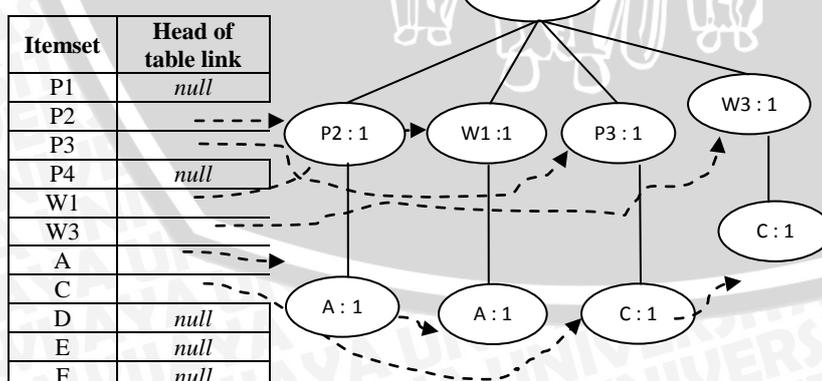
Gambar 3. 21 *FP-Tree* Pembacaan id Transaksi 2

Pembacaan ketiga yaitu untuk transaksi dengan id transaksi 4, transaksi yang terjadi adalah {P3, C}. P3 masuk terlebih dahulu ke dalam *FP-Tree* sehingga dibuat *node* baru dengan label P3 yang merupakan *node* anak dari *root* dan *count support* diberi nilai 1. Selain itu *itemset* P3 pada *header* tabel dibuat *link* yang menunjukkan *node* dimana informasi mengenai P3 disimpan. Selanjutnya *item* transaksi C masuk ke dalam *FP-Tree*, karena *node* P3 belum memiliki *node* anak berlabel C maka dibuat *node* baru dengan label C dan *count support node* C diberi nilai 1. Begitu pula dengan *itemset* C pada *header* tabel dibuat *link* yang merujuk ke *node* C tersebut. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 4 dapat dilihat pada Gambar 3.22.

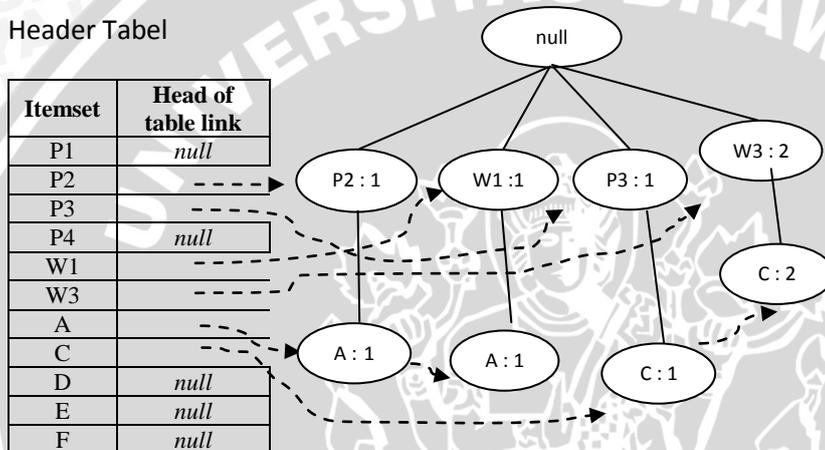
Gambar 3. 22 *FP-Tree* Pembacaan id Transaksi 3

Pembacaan keempat yaitu untuk transaksi dengan id transaksi 5, transaksi yang terjadi adalah {W3, C}. W3 masuk terlebih dahulu ke dalam *FP-Tree* sehingga dibuat *node* baru dengan label W3 yang merupakan *node* anak dari *root* dan *count support* diberi nilai 1. Setelah itu *itemset* W3 pada *header* tabel dibuat *link* yang menunjukkan *node* dimana informasi mengenai W3 disimpan. Selanjutnya *item* transaksi C masuk ke dalam *FP-Tree*, karena *node* W3 belum memiliki *node* anak berlabel C maka dibuat *node* baru dengan label C dan *count support node* C diberi nilai 1. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 5 dapat dilihat pada Gambar 3.23.

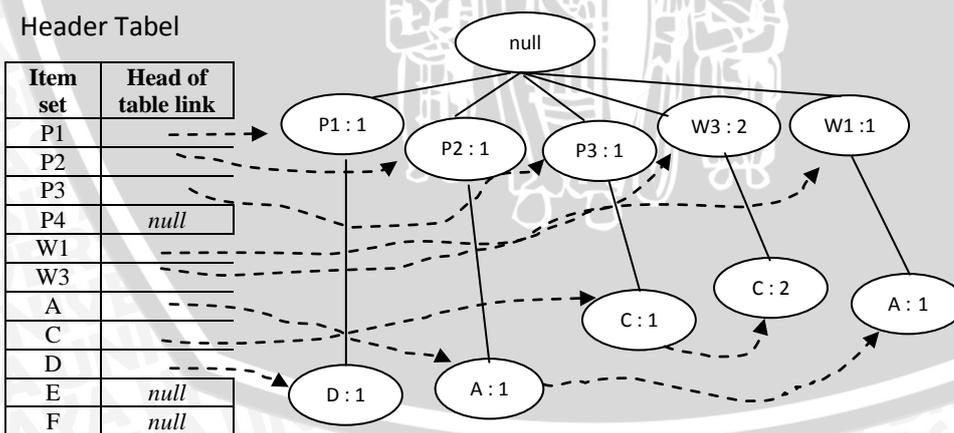
Header Tabel

Gambar 3. 23 *FP-Tree* Pembacaan id Transaksi 5

Pembacaan kelima yaitu untuk transaksi dengan id transaksi 6, transaksi ini sama dengan transaksi sebelumnya yaitu {W3, C}. Sehingga tidak dibuat simpul anak baru dari root. Itemset W3 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 ($count\ support = 2$). Selanjutnya *item* transaksi C masuk ke dalam *FP-Tree*, karena *node* W3 sudah memiliki *node* anak berlabel C maka dimasukkan ke *node* yang sama dengan label C dan *count support node* C ditambahkan 1 ($count\ support = 2$). *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 6 dapat dilihat pada Gambar 3.24.



Gambar 3. 24 *FP-Tree* Pembacaan id Transaksi 6



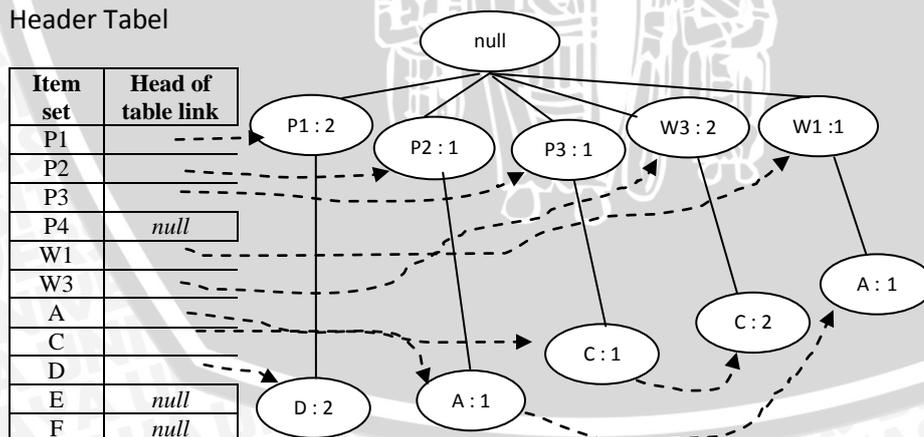
Gambar 3. 25 *FP-Tree* Pembacaan id Transaksi 7



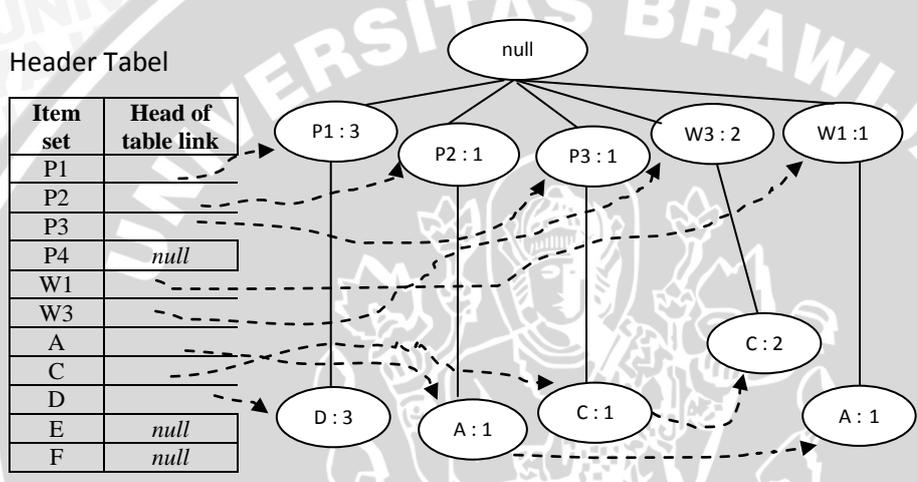
Pembacaan keenam yaitu untuk transaksi dengan id transaksi 7, transaksi yang terjadi adalah {P1, D}. P1 masuk terlebih dahulu ke dalam *FP-Tree* sehingga dibuat *node* baru dengan label P1 yang merupakan *node* anak dari *root* dan *count support* diberi nilai 1. Selain itu *itemset* P1 pada *header* tabel dibuat *link* yang menunjukkan *node* dimana informasi mengenai P1 disimpan. Selanjutnya *item* transaksi D masuk ke dalam *FP-Tree*, karena *node* P1 belum memiliki *node* anak berlabel D maka dibuat *node* baru dengan label D dan *count support node* D diberi nilai 1. Begitu pula dengan *itemset* D pada *header* tabel dibuat *link* yang merujuk ke *node* D tersebut. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 7 dapat dilihat pada Gambar 3.25.

Pembacaan ketujuh yaitu untuk transaksi dengan id transaksi 8, transaksi ini sama dengan transaksi sebelumnya yaitu {P1, D}. Sehingga tidak dibuat simpul anak baru dari *root*. *Itemset* P1 dimasukkan ke dalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 2). Selanjutnya *item* transaksi D masuk ke dalam *FP-Tree*, karena *node* P1 sudah memiliki *node* anak berlabel D maka dimasukkan ke *node* yang sama dengan label D dan *count support node* D ditambahkan 1 (*count support* = 2). *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 8 dapat dilihat pada Gambar 3.26.

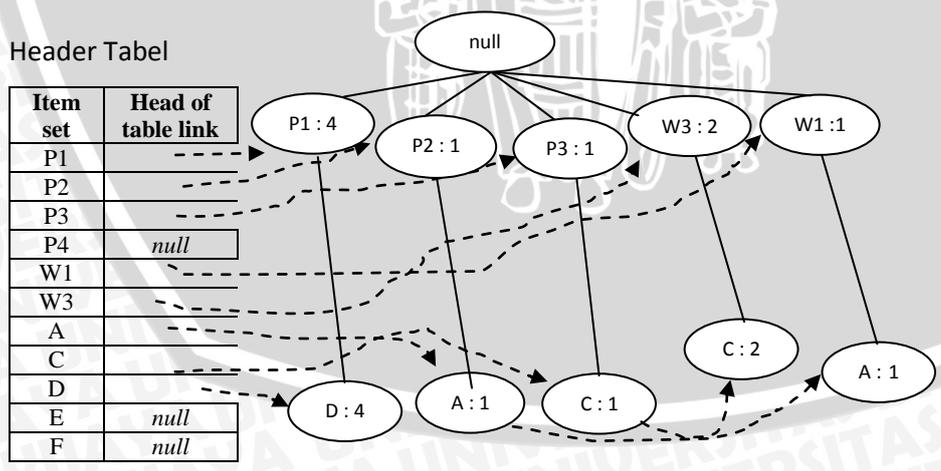
Header Tabel

Gambar 3. 26 *FP-Tree* Pembacaan id Transaksi 8

Pembacaan kedelapan yaitu untuk transaksi dengan id transaksi 9, transaksi ini juga sama dengan transaksi sebelumnya yaitu {P1, D}. Sehingga tidak dibuat simpul anak baru dari root. Itemset P1 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 3). Selanjutnya *item* transaksi D masuk ke dalam *FP-Tree*, karena *node* P1 sudah memiliki *node* anak berlabel D maka dimasukkan ke *node* yang sama dengan label D dan *count support node* D ditambahkan 1 (*count support* = 3). *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 9 dapat dilihat pada Gambar 3.27.



Gambar 3. 27 FP-Tree Pembacaan id Transaksi 9

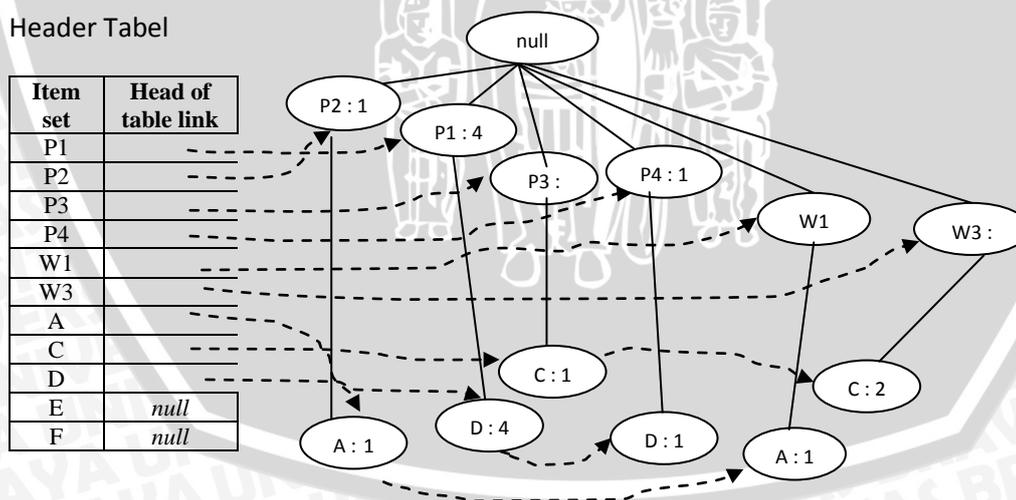


Gambar 3. 28 FP-Tree Pembacaan id Transaksi 10

Pembacaan kesembilan yaitu untuk transaksi dengan id transaksi 10, transaksi ini juga sama dengan transaksi sebelumnya yaitu {P1, D}. Sehingga tidak dibuat simpul anak baru dari root. Itemset P1 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 4). Selanjutnya *item* transaksi D masuk ke dalam *FP-Tree*, karena *node* P1 sudah memiliki *node* anak berlabel D maka dimasukkan ke *node* yang sama dengan label D dan *count support node* D ditambahkan 1 (*count support* = 4). *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 10 dapat dilihat pada Gambar 3.28.

Pembacaan kesepuluh yaitu untuk transaksi dengan id transaksi 11, transaksi yang terjadi adalah {P4, D}. Dimana P4 masuk terlebih dahulu ke dalam *FP-Tree* sehingga dibuat *node* baru dengan label P4 yang merupakan *node* anak dari *root* dan *count support* diberi nilai 1. Selain itu *itemset* P4 pada *header* tabel dibuat *link* yang menunjukkan *node* dimana informasi mengenai P4 disimpan. Selanjutnya *item* transaksi D masuk ke dalam *FP-Tree*, karena *node* P4 belum memiliki *node* anak berlabel D maka dibuat *node* baru dengan label D dan *count support node* D diberi nilai 1. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 11 dapat dilihat pada Gambar 3.29.

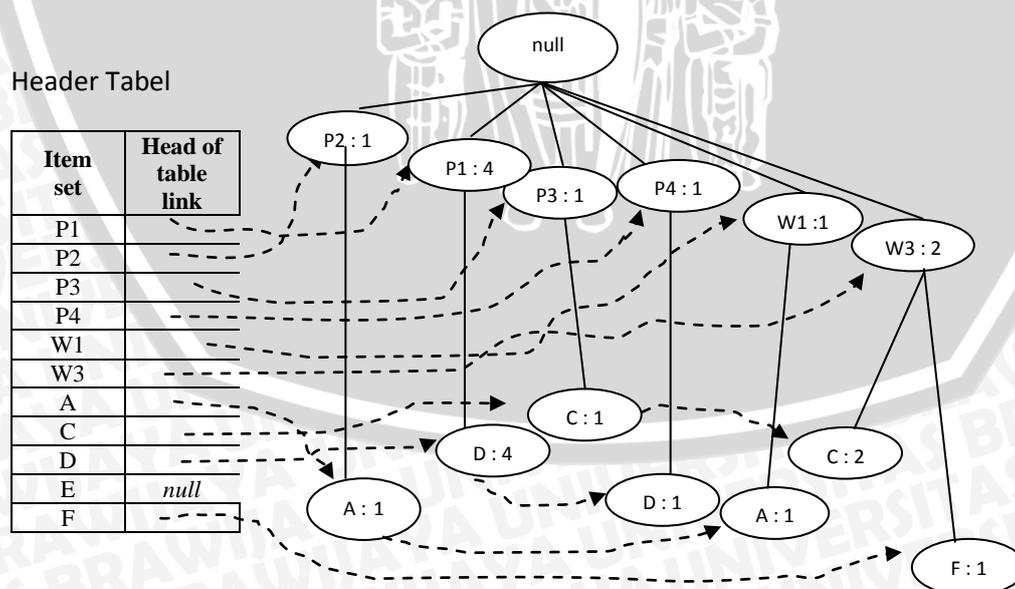
Header Tabel



Gambar 3. 29 FP-Tree Pembacaan id Transaksi 11

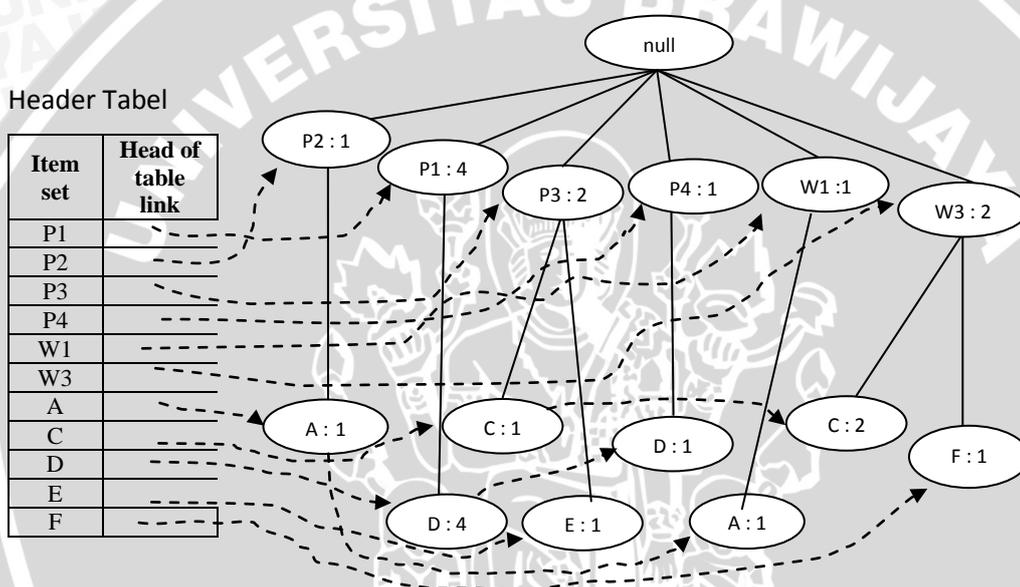
Pembacaan kesebelas yaitu untuk transaksi dengan id transaksi 12, transaksi yang terjadi adalah {W3, F}. Karena itemset W3 sudah mempunyai *node* dari akar *root* pada transaksi sebelumnya sehingga tidak dibuat simpul anak baru dari *root*. Itemset W3 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 3). Selanjutnya *item* transaksi F masuk ke dalam *FP-Tree*, karena *node* W3 belum memiliki *node* anak berlabel F maka dibuat *node* baru dengan label F dan *count support node* F diberi nilai 1. Begitu pula dengan *itemset* F pada *header* tabel dibuat *link* yang merujuk ke *node* F tersebut. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 12 dapat dilihat pada Gambar 3.30.

Pembacaan kedua belas yaitu untuk transaksi dengan id transaksi 13, transaksi yang terjadi adalah {P3, E}. Karena itemset P3 sudah mempunyai *node* dari akar *root* pada transaksi sebelumnya sehingga tidak dibuat simpul anak baru dari *root*. Itemset P3 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 2). Selanjutnya *item* transaksi E masuk ke dalam *FP-Tree*, karena *node* P3 belum memiliki *node* anak berlabel E maka dibuat *node* baru dengan label E dan *count support node* E diberi nilai 1. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 13 dapat dilihat pada Gambar 3.31.

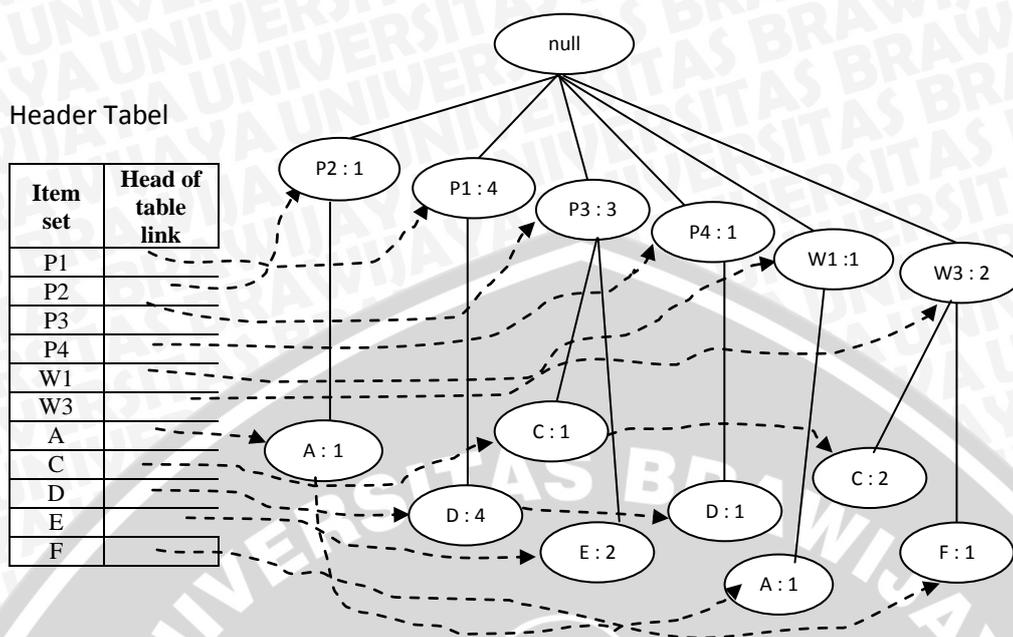


Gambar 3. 30 FP-Tree Pembacaan id Transaksi 12

Pembacaan ketiga belas yaitu untuk transaksi dengan id transaksi 14, transaksi ini sama dengan transaksi sebelumnya yaitu {P3, E}. Sehingga tidak dibuat simpul anak baru dari root. Itemset P3 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 3). Selanjutnya *item* transaksi E masuk ke dalam *FP-Tree*, karena *node* P3 sudah memiliki *node* anak berlabel E maka dimasukkan ke *node* yang sama dengan label E dan *count support node* E ditambahkan 1 (*count support* = 2). *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 14 dapat dilihat pada Gambar 3.32.

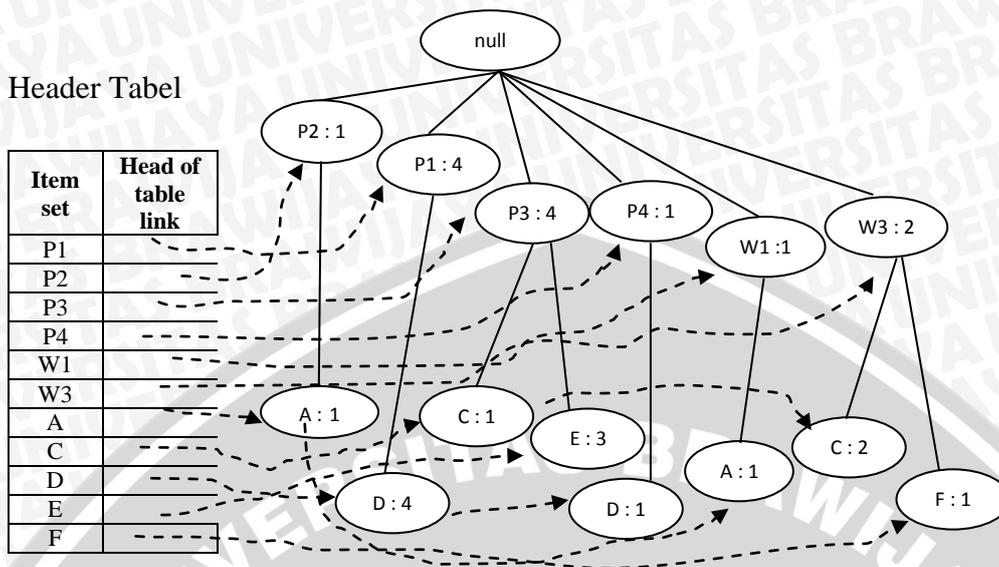


Gambar 3. 31 FP-Tree Pembacaan id Transaksi 13



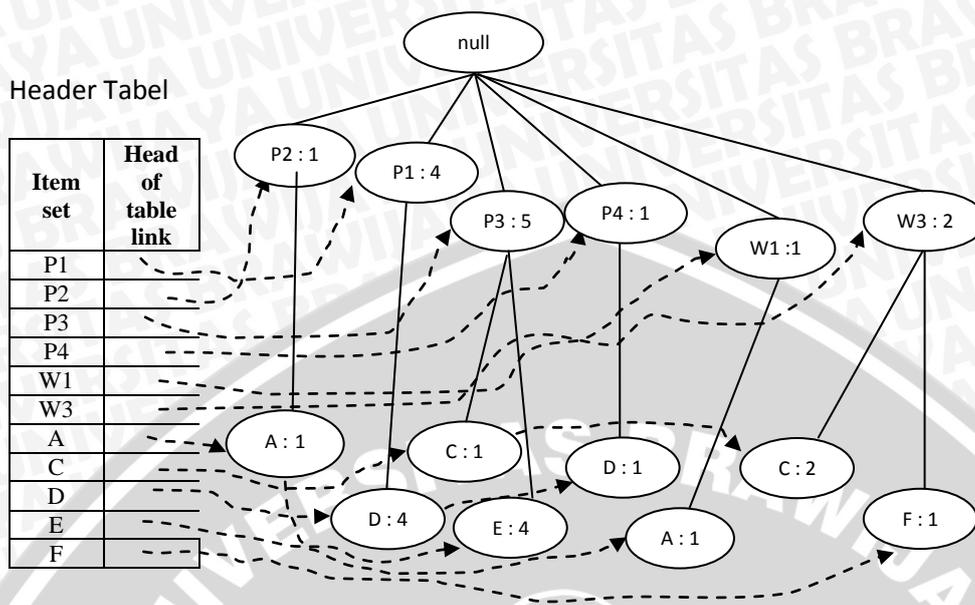
Gambar 3. 32 FP-Tree Pembacaan id Transaksi 14

Pembacaan keempat belas yaitu untuk transaksi dengan id transaksi 15, transaksi ini sama dengan transaksi sebelumnya yaitu {P3, E}. Sehingga tidak dibuat simpul anak baru dari root. Itemset P3 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 4). Selanjutnya *item* transaksi E masuk ke dalam *FP-Tree*, karena *node* P3 sudah memiliki *node* anak berlabel E maka dimasukkan ke *node* yang sama dengan label E dan *count support node* E ditambahkan 1 (*count support* = 3). *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 15 dapat dilihat pada Gambar 3.33.



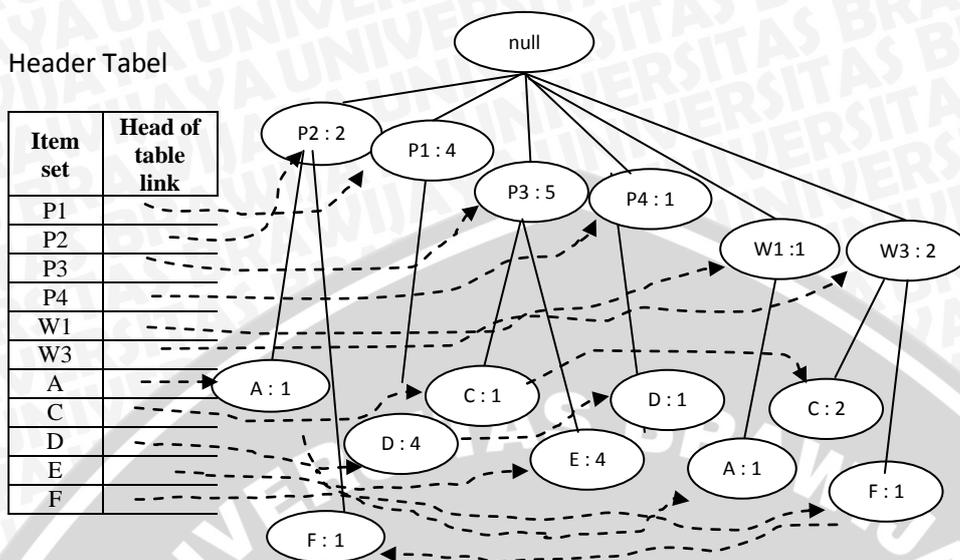
Gambar 3. 33 FP-Tree Pembacaan id Transaksi 15

Pembacaan kelima belas yaitu untuk transaksi dengan id transaksi 16, transaksi ini sama dengan transaksi sebelumnya yaitu {P3, E}. Sehingga tidak dibuat simpul anak baru dari root. Itemset P3 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 5). Selanjutnya *item* transaksi E masuk ke dalam *FP-Tree*, karena *node* P3 sudah memiliki *node* anak berlabel E maka dimasukkan ke *node* yang sama dengan label E dan *count support node* E ditambahkan 1 (*count support* = 4). *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 16 dapat dilihat pada Gambar 3.34.

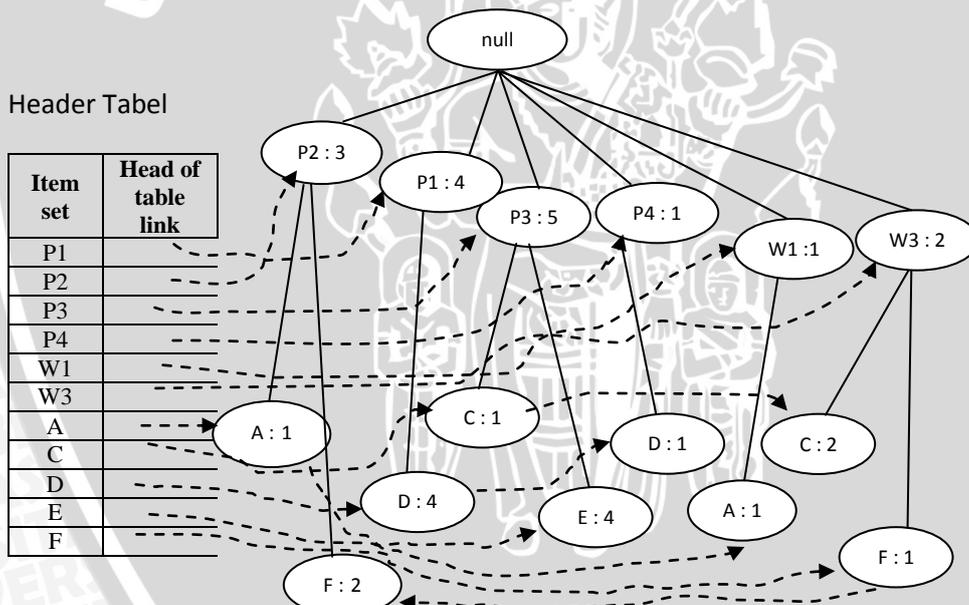


Gambar 3. 34 FP-Tree Pembacaan id Transaksi 16

Pembacaan keenam belas yaitu untuk transaksi dengan id transaksi 17, transaksi yang terjadi adalah {P2, F}. Karena itemset P2 sudah mempunyai *node* dari akar *root* pada transaksi sebelumnya sehingga tidak dibuat simpul anak baru dari *root*. Itemset P2 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 2). Selanjutnya *item* transaksi F masuk ke dalam *FP-Tree*, karena *node* P2 belum memiliki *node* anak berlabel F maka dibuat *node* baru dengan label F dan *count support node* F diberi nilai 1. Begitu pula dengan *itemset* F pada *header* tabel dibuat *link* yang merujuk ke *node* F tersebut. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 17 dapat dilihat pada Gambar 3.35.



Gambar 3. 35 FP-Tree Pembacaan id Transaksi 17



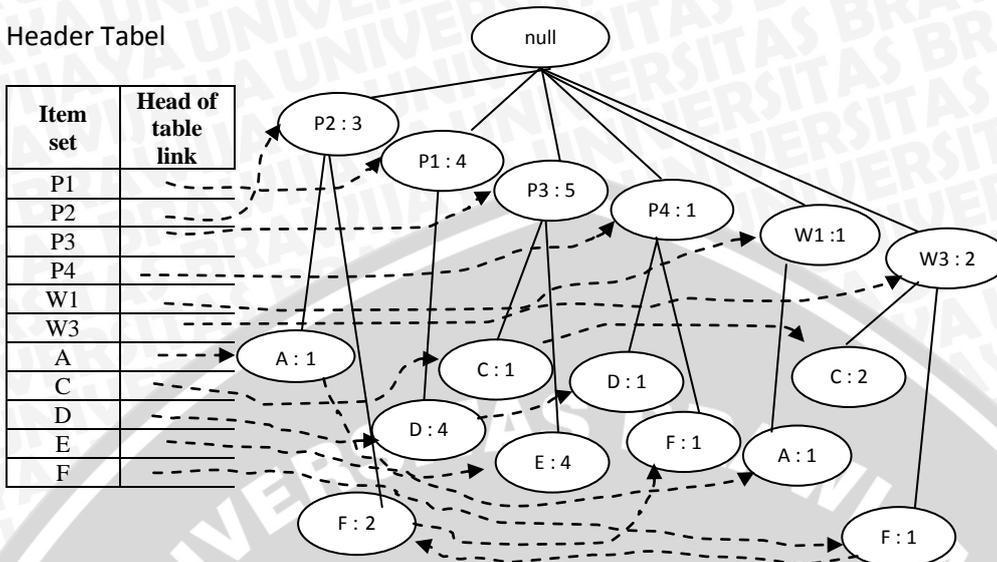
Gambar 3. 36 FP-Tree Pembacaan id Transaksi 18

Pembacaan ketujuh belas yaitu untuk transaksi dengan id transaksi 18, transaksi ini sama dengan transaksi sebelumnya yaitu {P2, F}. Sehingga tidak dibuat simpul anak baru dari root. Itemset P2 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 3). Selanjutnya *item* transaksi F masuk ke dalam *FP-Tree*, karena *node* P2 sudah memiliki *node* anak berlabel F maka dimasukkan ke *node* yang sama dengan label F dan *count support*

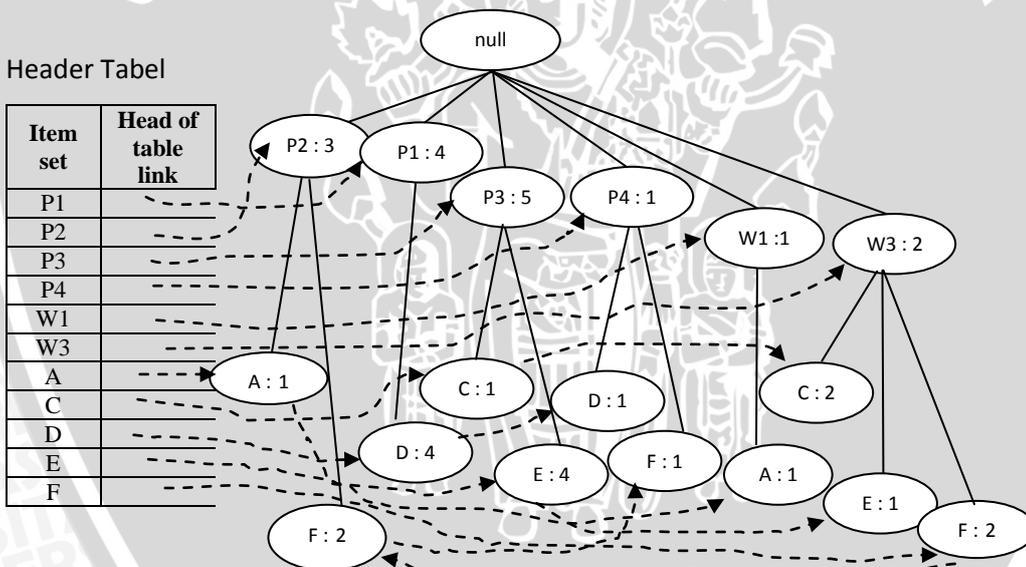
node F ditambahkan 1 (*count support* = 2). *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 18 dapat dilihat pada Gambar 3.36.

Pembacaan kedelapan belas yaitu untuk transaksi dengan id transaksi 19, transaksi yang terjadi adalah {P4, F}. Karena itemset P4 sudah mempunyai *node* dari akar *root* pada transaksi sebelumnya sehingga tidak dibuat simpul anak baru dari *root*. Itemset P4 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 2). Selanjutnya *item* transaksi F masuk ke dalam *FP-Tree*, karena *node* P2 belum memiliki *node* anak berlabel F maka dibuat *node* baru dengan label F dan *count support node* F diberi nilai 1. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 19 dapat dilihat pada Gambar 3.37.

Pembacaan kesembilan belas yaitu untuk transaksi dengan id transaksi 20, transaksi yang terjadi adalah {W3, F}. Karena itemset W3 sudah mempunyai *node* dari akar *root* pada transaksi sebelumnya sehingga tidak dibuat simpul anak baru dari *root*. Itemset W3 dimasukkan kedalam *node* yang sama dengan *count support* bertambah 1 (*count support* = 2). Selanjutnya *item* transaksi F masuk ke dalam *FP-Tree*, karena *node* W3 belum memiliki *node* anak berlabel F maka dibuat *node* baru dengan label F dan *count support node* F diberi nilai 1. *FP-Tree* yang terbentuk untuk pembacaan data transaksi dengan id transaksi 20 dapat dilihat pada Gambar 3.38.



Gambar 3. 37 FP-Tree Pembacaan id Transaksi 19



Gambar 3. 38 FP-Tree Pembacaan id Transaksi 20

Langkah ketujuh adalah *conditional pattern base* yaitu mencari *suffix pattern* dan *prefix path* pada *FP-tree* yang telah terbentuk seperti yang dapat dilihat pada Tabel 3.10.

Tabel 3. 10 Conditional Pattern Base dari Sampel Data

Suffix	Conditional Pattern Base
{A}	{P2 : 1}, {W1 : 1}
{C}	{P3 : 1}, {W3 : 2}
{D}	{P1 : 4}, {P4 : 1}
{E}	{P3 : 4}
{F}	{P2 : 2}, {W3 : 2}, {P4 : 1}

Langkah kedelapan adalah tahap *conditional FP-tree* yaitu *support count* setiap itemset *prefix path* setiap *suffix* dijumlahkan. Item yang memiliki jumlah *support count* lebih besar atau sama dengan *minimum support* akan dibangkitkan dengan *conditional FP-tree*.

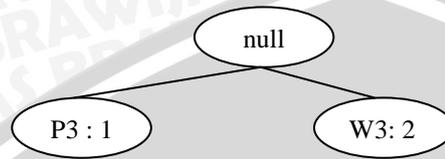
$$\begin{aligned}
 \text{Suffix \{A\} :} & \quad \text{Support \{P2\}} = 1/20 = 0.05 \\
 & \quad \text{Support \{W1\}} = 1/20 = 0.05 \\
 \text{Suffix \{C\} :} & \quad \text{Support \{P3\}} = 1/20 = 0.05 \\
 & \quad \text{Support \{W3\}} = 2/20 = 0.10 \\
 \text{Suffix \{D\} :} & \quad \text{Support \{P1\}} = 4/20 = 0.20 \\
 & \quad \text{Support \{P4\}} = 1/20 = 0.05 \\
 \text{Suffix \{E\} :} & \quad \text{Support \{P3\}} = 4/20 = 0.20 \\
 \text{Suffix \{F\} :} & \quad \text{Support \{P2\}} = 2/20 = 0.10 \\
 & \quad \text{Support \{W3\}} = 2/20 = 0.10 \\
 & \quad \text{Support \{P4\}} = 1/20 = 0.05
 \end{aligned}$$

Pada Gambar 3.39 merupakan lintasan pada *FP-Tree* yang memiliki *suffix* A, sedangkan *conditional FP-Tree* untuk *suffix* A tidak dibangun karena *support* P2 dan W1 tidak memenuhi nilai *minimum support*.

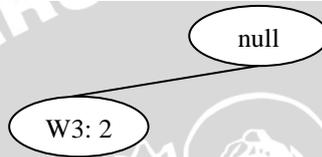


Gambar 3. 39 Lintasan yang memiliki *suffix* A

Pada Gambar 3.40 merupakan lintasan pada *FP-Tree* yang memiliki *suffix* C, sedangkan *conditional FP-Tree* untuk *suffix* C dapat dilihat pada Gambar 3.41. Lintasan {P3, C} tidak dibangun *conditional FP-Tree* karena *support* P3 tidak memenuhi nilai minimum *support*.



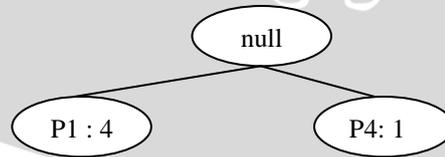
Gambar 3. 40 Lintasan yang memiliki suffix C



Gambar 3. 41 Conditional FP-Tree untuk suffix C

Berdasarkan Gambar 3.41, karena lintasan yang terbentuk berupa lintasan tunggal maka *frequent itemset* yang didapat dari *suffix* C adalah {W3, C} dengan frekuensi 2.

Pada Gambar 3.42 merupakan lintasan pada *FP-Tree* yang memiliki *suffix* D, sedangkan *conditional FP-Tree* untuk *suffix* D dapat dilihat pada Gambar 3.43. Lintasan {P4, D} tidak dibangun *conditional FP-Tree* karena *support* P4 tidak memenuhi nilai minimum *support*.



Gambar 3. 42 Lintasan yang memiliki suffix D



Gambar 3. 43 Conditional FP-Tree untuk suffix D

Berdasarkan Gambar 3.43, karena lintasan yang terbentuk berupa lintasan tunggal maka *frequent itemset* yang didapat dari *suffix* D adalah {P1, D} dengan frekuensi 4.

Pada Gambar 3.44 merupakan lintasan pada *FP-Tree* yang memiliki *suffix* E, sedangkan *conditional FP-Tree* untuk *suffix* E dapat dilihat pada Gambar 3.45. Lintasan {W3, E} tidak dibangun *conditional FP-Tree* karena *support* W3 tidak memenuhi nilai minimum *support*.



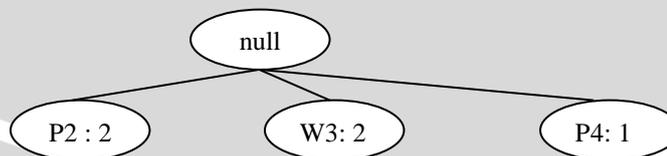
Gambar 3. 44 Lintasan yang memiliki suffix E



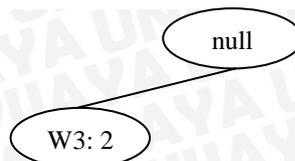
Gambar 3. 45 Conditional FP-Tree untuk suffix E

Berdasarkan Gambar 3.45, karena lintasan yang terbentuk berupa lintasan tunggal maka *frequent itemset* yang didapat dari *suffix* E adalah {P3, E} dengan frekuensi 4.

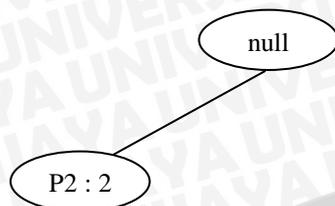
Pada Gambar 3.46 merupakan lintasan pada *FP-Tree* yang memiliki *suffix* F, sedangkan *conditional FP-Tree* untuk *suffix* F dapat dilihat pada Gambar 3.47. Lintasan {P4, F} tidak dibangun *conditional FP-Tree* karena *support* P4 tidak memenuhi nilai minimum *support*.



Gambar 3. 46 Lintasan yang memiliki suffix F



Gambar 3. 47 Lintasan yang memiliki suffix F



Gambar 3. 48 Conditional FP-Tree untuk suffix F

Berdasarkan Gambar 3.47, karena lintasan yang terbentuk berupa lintasan tunggal maka *frequent itemset* yang didapat dari *suffix F* adalah {P2, F} dengan frekuensi 2 dan {W3, F} dengan frekuensi 2.

Hasil *frequent itemset* dari setiap proses *conditional FP-tree* digabungkan menjadi satu dalam suatu tabel yaitu tabel *Frequent Itemset* seperti pada Tabel 3.11.

Tabel 3. 11 Hasil Frequent Itemset Sampel Data

<i>Frequent Itemset</i>	<i>Frekuensi</i>	<i>Support</i>
W3 → C	2	0.10
P1 → D	4	0.20
P3 → E	4	0.20
P2 → F	2	0.10
W3 → F	2	0.10

Langkah kesembilan adalah *frequent itemset* yang didapat selanjutnya dilakukan perhitungan nilai *confidence*. Hasil perhitungan nilai *confidence frequent itemset* dapat dilihat pada Tabel 3.12. Nilai *confidence* ini akan menentukan apakah *frequent itemset* tersebut dapat dibangkitkan menjadi *rule*. Bila nilai *confidence* lebih besar atau sama dengan nilai minimum *confidence* yang telah ditentukan maka *frequent itemset* tersebut dapat dibangkitkan untuk menjadi *rule*, sedangkan bila nilai *confidence* lebih kecil dari nilai minimum *confidence* maka *frequent itemset* tersebut tidak dapat dibangkitkan menjadi *rule*.

$$\begin{aligned}
 \text{Confidence } \{W3\} \rightarrow \{C\} &= 0.1/0.20 = 0.50 \\
 \text{Confidence } \{P1\} \rightarrow \{D\} &= 0.2/0.20 = 1 \\
 \text{Confidence } \{P3\} \rightarrow \{E\} &= 0.2/0.25 = 0.80 \\
 \text{Confidence } \{P2\} \rightarrow \{F\} &= 0.1/0.25 = 0.40 \\
 \text{Confidence } \{W3\} \rightarrow \{F\} &= 0.1/0.25 = 0.40
 \end{aligned}$$



Tabel 3. 12 Confidence Frequent Itemset Sampel Data

<i>Frequent Itemset</i>	<i>Support (A,B)</i>	<i>Support (A)</i>	<i>Confidence</i>
W3 → C	0.10	0.20	0.50
P1 → D	0.20	0.20	1.00
P3 → E	0.20	0.25	0.80
P2 → F	0.10	0.25	0.40
W3 → F	0.10	0.20	0.50

Berdasarkan Tabel 3.12, *Frequent itemset* yang terpilih untuk dibangkitkan menjadi *rule* adalah $W3 \rightarrow C$, $P1 \rightarrow D$, $P3 \rightarrow E$, dan $W3 \rightarrow F$ karena nilai *confidence* masing – masing *frequent itemset* tersebut lebih besar atau sama dengan nilai *minimum confidence* yang telah ditentukan. *Rule* yang terbentuk dapat dilihat pada Tabel 3.13.

Tabel 3. 13 Rule yang Terbentuk

<i>Rules</i>	<i>Keterangan</i>
$W3 \rightarrow C$	Jika jenis kelamin Wanita dan rentang umur 41-50 tahun maka jenis mobil yang dibeli SIRION 1.3 FMC MT
$P1 \rightarrow D$	Jika jenis kelamin Pria dan rentang umur 20-30 tahun maka jenis mobil yang dibeli TERIOS TX MT ADV MC
$P3 \rightarrow E$	Jika jenis kelamin Pria dan rentang umur 41-50 tahun maka jenis mobil yang dibeli XENIA FMC R MT 1.3 F
$W3 \rightarrow F$	Jika jenis kelamin Wanita dan rentang umur 41-50 tahun maka jenis mobil yang dibeli XENIA FMC R MT 1.3 S

Selanjutnya *rule* yang terdapat pada Tabel 3.13 dilakukan uji kekuatan *rule* dengan menggunakan *lift ratio*. *Lift ratio* didapatkan dari hasil perbandingan antara *confidence* dengan *benchmark confidence*. Hasil perhitungan nilai *lift ratio* dapat dilihat pada Tabel 3.14

$$\begin{aligned}
 \text{Lift Ratio } \{W3\} \rightarrow \{C\} &= \frac{0.50}{0.10} = 3.33 \\
 \text{Lift Ratio } \{P1\} \rightarrow \{D\} &= \frac{1.00}{0.25} = 4.00 \\
 \text{Lift Ratio } \{P3\} \rightarrow \{E\} &= \frac{0.80}{0.20} = 4.00 \\
 \text{Lift Ratio } \{W3\} \rightarrow \{F\} &= \frac{0.50}{0.10} = 5.00
 \end{aligned}$$

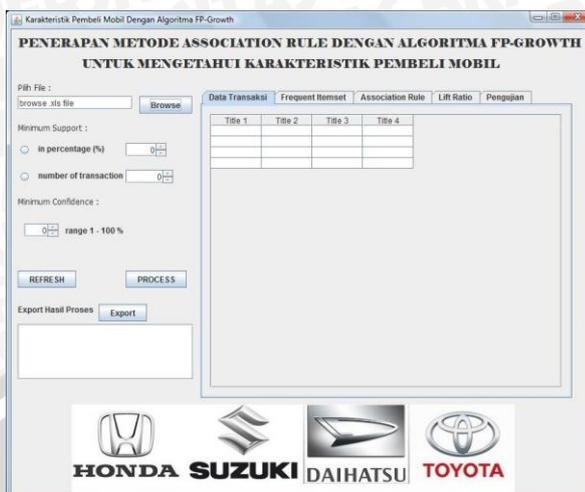
Tabel 3. 14 Lift Ratio Rules

<i>Rules</i>	<i>Confidence</i>	<i>Frekuensi Item Consequent</i>	<i>Benchmark Confidence</i>	<i>Lift Ratio</i>
W3 → C	0.50	2	0.15 (support C)	5.00
P1 → D	1.00	4	0.25 (support D)	5.00
P3 → E	0.80	4	0.20 (support E)	4.00
W3 → F	0.50	2	0.25 (support F)	2.00

Berdasarkan hasil perhitungan *lift ratio* pada Tabel 3.14, dapat dilihat bahwa semua *rule* yang dihasilkan memiliki kekuatan yang bagus karena nilai *lift ratio* semua *rule* lebih besar dari 1. *Rule* yang memiliki nilai *lift ratio* paling besar adalah P1 → D dan P3 → E yaitu sebesar 5.00, dan artinya *rule* P1 → D dan P3 → E memiliki kekuatan asosiasi yang tinggi sehingga dapat menunjukkan adanya manfaat.

3.4. Rancangan Antar Muka

Rancangan antar muka secara umum terdiri dari bagian *input*, bagian *output*, tombol proses, tombol *refresh*, dan tombol *export*. Bagian input terdiri dari tiga data *input*, yaitu data *input browse file*, data *input minimum support*, dan data *input minimum confidence*. Bagian output terdiri dari empat submenu yaitu submenu Data Transaksi, submenu *Frequent Itemset*, submenu *Association Rule*, dan submenu *Lift Ratio*. Tampilan rancangan antar muka secara umum dapat dilihat pada Gambar 3.48.



Gambar 3. 49 Rancangan Antar Muka Secara Umum

3.5. Rancangan Uji Coba

Uji coba sistem untuk analisa karakteristik pembeli mobil ini adalah dilakukan evaluasi terhadap hasil analisa yang dihasilkan oleh sistem. Tujuan dari uji coba ini adalah untuk mengetahui tingkat kekuatan (*lift ratio*) dari *rule* yang dihasilkan. Selain itu, uji coba ini dilakukan juga untuk mengetahui akurasi *rule* yang dihasilkan terhadap data uji.

Pada penelitian ini digunakan data transaksi penjualan selama enam bulan di empat *dealer* kota Mataram. Rancangan uji coba dari penelitian ini ditunjukkan pada Tabel 3.15 untuk uji tingkat kekuatan (*lift ratio*) *rule* yang dihasilkan dan pada Tabel 3.16 untuk uji tingkat akurasi *rule* yang dihasilkan terhadap data uji.

Tabel 3. 15 Tabel Rancangan Uji Coba Lift Ratio Rule

<i>Dealer</i>	<i>Minimum Support (%)</i>	<i>Minimum Confidence (%)</i>	<i>Rule</i>	<i>Lift Ratio</i>

Tabel 3. 16 Tabel Rancangan Uji Coba Akurasi Rule

<i>Rule</i>	<i>Frek Rule Di Data Uji</i>	<i>Support Data uji</i>	<i>Support Data Latih</i>	<i>Error</i>	<i>Akurasi</i>

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1. Lingkungan Implementasi

Implementasi dilakukan untuk proses transformasi representasi rancangan ke dalam bahasa pemrograman sehingga dapat dimengerti oleh komputer. Lingkungan implementasi yang akan 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 sebuah PC (*Personal Computer*) dengan spesifikasi sebagai berikut :

1. *Processor* AMD Phenom(tm) II X4 925 2.80 GHz
2. Memori 2GB
3. *Harddisk* dengan kapasitas 500 GB
4. Monitor 17.0"
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 Vista™ Home Premium Service Pack 1*.
2. *Netbeans IDE 7.1.1* dan *JDK 1.6* dalam mengembangkan aplikasi untuk analisis pola penjualan buku.
3. *Jxl.jar*.

4.2. Implementasi Program

Berdasarkan rancangan pembuatan sistem pada bab 3, maka pada subbab ini akan dijelaskan implementasi proses – proses tersebut. Secara garis besar proses dikelompokkan menjadi 5 tahap, yaitu tahap *frequent 1-itemset*, tahap

frequent itemset, tahap *generate rule*, tahap *lift ratio* dan tahap *akurasi rule*. Implementasi program terdiri dari 18 *class* yang dijelaskan pada Tabel 4.1 beserta fungsinya.

Tabel 4. 1 Class Pada Implementasi Program dan Fungsinya

Nama Class	Fungsi
Class Data	Menyimpan data parameter yang digunakan (jenis mobil, jenis kelamin, dan umur) dari pembacaan file excel ke Arraylist. Terdiri dari <i>method</i> sebagai berikut : <ol style="list-style-type: none"> 1. Data 2. inputJMobil 3. getJMobil 4. inputJKelamin 5. getJKelamin 6. cekUmur 7. inputUmur 8. getUmur
Class MyCustomFilter	Filter file yang dapat dipilih pada komponen <i>filechooser</i> . Terdiri dari <i>method</i> , sebagai berikut : <ol style="list-style-type: none"> 1. accept 2. getDescription 3. cekFile
Class FrekJenisMobil	Menyimpan data nama, frekuensi, dan support setiap jenis mobil yang terdapat pada transaksi. Terdiri dari <i>method</i> sebagai berikut : <ol style="list-style-type: none"> 1. FrekJenisMobil 2. getFrek 3. hitungSupport
Class FrekKombinasi	Menyimpan data kombinasi umur dan jenis kelamin yang terdapat pada transaksi beserta frekuensi dan supportnya. Terdiri dari <i>method</i> sebagai berikut : <ol style="list-style-type: none"> 1. FrekKombinasi 2. kombinasi 3. KombinasiTransaksi 4. ItungFrekKomb 5. itungSupportKomb
Class frequentItemsetJenisMobil	Menyimpan data nama jenis mobil dan nilai supportnya untuk setiap jenis mobil yang lolos minimum <i>support</i> . Terdiri dari <i>method</i> <i>frequentItemset JenisMobil</i>
Class	Menyimpan data nama kombinasi umur dan

frequent1ItemsetKombinasi	jenis kelamin, dan nilai supportnya untuk setiap kombinasi yang lolos minimum <i>support</i> . Terdiri dari <i>method</i> frequent1ItemsetKombinasi
Class OneItemset	Melakukan seleksi kandidat 1 itemset yang lolos minimum <i>support</i> untuk menjadi <i>frequent 1-itemset</i> . Terdiri dari <i>method</i> sebagai berikut : 1. OneItemsetKomb 2. OneItemsetJM
Class TemporaryTrans	Melakukan kombinasi data transaksi jenis mobil, jenis kelamin, dan umur. Terdiri dari <i>method</i> sebagai berikut : 1. TemporaryTrans 2. seleksiTrans
Class FPTree	Melakukan pembentukan struktur data <i>FP-Tree</i> . Terdiri dari <i>method</i> sebagai berikut : 1. FPTree 2. insert 3. koleksiLintasan 4. simpanPath
Class FPTreeNode	Menyimpan data informasi <i>node</i> pada <i>FP-Tree</i> . Terdiri dari <i>method</i> : 1. FPTreeNode 2. displayNode 3. NodeNext 4. NodeSibling 5. NodeParent
Class FPTreeHeaderEntry	Sebagai <i>header table</i> yang menyimpan informasi total bobot tiap item yang menjadi <i>frequent 1-itemset</i> yang telah masuk ke dalam <i>FP-Tree</i> . Terdiri dari <i>method</i> FPTreeHeaderEntry dan DisplayHeaderEntry.
Class Lintasan	Menyimpan data lintasan. Terdiri dari <i>method</i> Lintasan.
Class Prefix	Menyimpan data <i>prefix</i> dari semua <i>suffix</i> (proses <i>conditional pattern base</i>). Terdiri dari <i>method</i> Prefix dan condPatternBase.
Class FrequentItemset	Melakukan proses <i>conditional FP-Tree</i> , melakukan proses koleksi <i>frequent itemset</i> dan menyimpan data <i>frequent itemset</i> . Terdiri dari <i>method</i> sebagai berikut : 1. FrequentItemset 2. condFPTree 3. koleksiFreqItemset 4. PenulisanFreqit
Class Confidence	Menyimpan data nilai <i>confidence</i> . Terdiri dari <i>methodr</i> Confidence.

<i>Class Rule</i>	Menyimpan data <i>rule</i> . Terdiri dari <i>method Rule</i> .
<i>Class LiftRatio</i>	Menyimpan data nilai <i>lift ratio</i> . Terdiri dari <i>method LiftRatio</i> .
<i>Class ProsesRule</i>	Melakukan perhitungan <i>confidence</i> tiap <i>frequent itemset</i> , melakukan seleksi <i>rule</i> , dan perhitungan <i>lift ratio</i> tiap <i>rule</i> yang dihasilkan. Terdiri dari <i>method</i> sebagai berikut : <ol style="list-style-type: none"> 1. ConfidenceFI 2. SeleksiRule
	<ol style="list-style-type: none"> 3. PenulisanRule 4. hitungLift
<i>Class InfoDataUji</i>	Menyimpan informasi dari hasil perhitungan akurasi data uji. Terdiri dari <i>method InfoDataUji</i>
<i>Class TransDataUji</i>	Menyimpan parameter yang dibutuhkan (jenis mobil, kombinasi umur dan jenis kelamin) dari data uji. Terdiri dari <i>method TransDataUji</i>
<i>Class Akurasi</i>	Melakukan proses perhitungan akurasi. Terdiri dari <i>method</i> sebagai berikut : <ol style="list-style-type: none"> 1. Akurasi 2. KombinasiUmurJKUji 3. SetData 4. prosesAkurasi
<i>Class exportHasil</i>	Melakukan ekspor data output ke file .xls Terdiri dari <i>method exportTable</i>
<i>Class Awal</i>	Mengatur tampilan GUI dan semua proses sistem. Terdiri dari <i>method</i> sebagai berikut : <ol style="list-style-type: none"> 1. Awal 2. bacaCell 3. bacaExcel 4. setTabelData 5. cekArrayList 6. jButton1ActionPerformed 7. OneItemsetMobil 8. OneItemsetKombinasi 9. FreqOneItemset 10. FPTree 11. conditionalPatternBase 12. conditionalFPTree 13. hitungConfidence 14. seleksiRule
	<ol style="list-style-type: none"> 15. jButton3ActionPerformed 16. jButton1MouseClicked 17. jButton1StateChanged 18. jButton2MouseClicked 19. jButton2StateChanged

	20. jSpinner3StateChanged 21. jButton2ActionPerformed 22. jButton4ActionPerformed 23. setDataUji 24. hitungAkurasi 25. jButton5ActionPerformed 26. jButton6ActionPerformed 27. RefreshAkurasi 28. jButton7ActionPerformed 29. RefreshTabelData 30. RefreshJenisMobil 31. RefreshKombinasi 32. RefreshOneItemset 33. RefreshFrequentItemset 34. RefreshKandidatRule 35. RefreshRule 36. RefreshKetRule
Class Main	Main program atau menjalankan program. Terdiri dari <i>method</i> main

4.2.1. Implementasi Tahap *Frequent 1-itemset*

Pada tahap ini, dilakukan pengkoleksian tiap item yang memiliki frekuensi kemunculannya pada data transaksi dalam periode waktu tertentu lebih besar atau sama dengan minimum *support*. Dimana terdiri dari 2 item, yaitu item jenis mobil dan item kombinasi antara jenis kelamin dan rentang umur.

Tahap *frequent 1-itemset* terdiri dari 5 proses, yaitu proses pembacaan data transaksi, proses koleksi jenis mobil, proses koleksi kombinasi antara jenis kelamin dan rentang umur, proses pencarian jenis mobil yang *frequent*, dan proses pencarian kombinasi yang *frequent*.

4.2.1.1. Implementasi Pembacaan Data Transaksi

Proses awal yang dilakukan adalah pembacaan data transaksi dari file excel berekstensi .xls. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.1.

```
public void bacaExcel(File dataFile){
    try{
        WorkbookSettings ws = new WorkbookSettings();
        ws.setLocale(new Locale("en", "EN"));
    }
}
```

```
Workbook wb = null;
try{
    wb = Workbook.getWorkbook(dataFile, ws);
}catch (Exception ex) {
    ex.printStackTrace();
}
Sheet s = wb.getSheet(0);
jumlahdata = s.getRows();
}catch (Exception e) {
    e.printStackTrace();
}
}

void setTabelData(Sheet s) {
    Object[] kolom = new Object[s.getColumns()+1];
    kolom[0] = "No.";
    for (int i = 0; i < s.getColumns(); i++){
        Cell c = s.getCell(i, 0);
        kolom[i+1] = c.getContents();
    }
    dataKolom = new String[s.getColumns()];
    for (int i = 0; i < s.getColumns(); i++) {
        Cell c = s.getCell(i, 0);
        dataKolom[i] = c.getContents();
    }
    data = new String[s.getColumns()][s.getRows()];
    modeldata = new DefaultTableModel(null, kolom);
    Object[] tampil = new Object[kolom.length];
    for (int i = 0; i < s.getRows() - 1; i++) {
        tampil[0] = (i + 1);
        for (int j = 0; j < s.getColumns(); j++) {
            tampil[j + 1] = bacaCell(s, j, i+1);
            data[j][i] = bacaCell(s, j, i+1);
        }
        jTable1.setModel(modeldata);
        modeldata.addRow(tampil);
    }
}
```

Sourcecode 4. 1 Proses Pembacaan Data Transaksi

Fungsi `bacaExcel` memiliki input parameter `dataFile` yang bertipe `File` . Fungsi ini berfungsi untuk mendapatkan data excel dari `workbook` `dataFile` , menentukan `sheet` dari file `excel` `dataFile` yang akan dibaca adalah `sheet` pertama, dan mengambil nilai jumlah baris data dari `dataFile` . Untuk mendapatkan data `excel` dari `workbook` `dataFile` menggunakan `method` `getWorkbook` yang kemudian disimpan pada `object class` `workbook` yaitu `wb` . Menentukan `sheet` dari file `excel` `dataFile` yaitu menggunakan `method` `getSheet(0)` yang kemudian disimpan pada `object class` `sheet` yaitu `s` . Untuk mengambil nilai jumlah baris data digunakan `method` `getRows()` yang selanjutnya disimpan pada variabel `public` `jumlahdata` yang bertipe `integer` .

Fungsi dari `setTabelData` berfungsi untuk menampilkan data dari file `excel` ke tabel `jTable1` . Selain itu juga menyimpan tiap baris data transaksi ke dalam struktur data `array` dua dimensi `data[j][i]` dimana `j` merupakan `index` kolom dan `i` merupakan `index` baris. Parameter inputan dari fungsi ini yaitu `Sheet` `s` . Pada fungsi ini digunakan fungsi `getColumn` untuk mendapatkan nilai jumlah kolom yang ada di file `excel` , `getRows` untuk mendapatkan nilai jumlah baris yang ada di file `excel` , `getContent` untuk mendapatkan isi `cell` pada file `excel` , `setModel` untuk mengatur tampilan pada `jTable1` agar sesuai dengan model tabel yang telah ditentukan, dan `addRow` untuk menambahkan baris data pada `jTable1` .

4.2.1.2. Implementasi Koleksi Jenis Mobil

Proses selanjutnya adalah mengkoleksi jenis mobil yang ada pada data transaksi dan menghitung frekuensi kemunculan tiap jenis mobil. Implementasi dari proses ini dapat ditunjukkan pada `source code` 4.2.

Pada fungsi `getFrek` , parameter inputan adalah data transaksi `Trans` yang bertipe `ArrayList<String>` . Proses pada fungsi ini dilakukan dengan pembacaan setiap data transaksi dan diambil jenis mobil pada `index` data transaksi yang sedang dibaca dengan perintah `Trans.get(i)` . Setiap `index` data transaksi, dilakukan pengecekan pada `object` `F` , yang merupakan deklarasi dari `class` `FrekJenisMobil` . Apabila jenis mobil belum ada di `ArrayList<String>` `ItemJMobil` pada `object` `F` maka jenis mobil tersebut disimpan dalam `ArrayList<String>` `ItemJMobil` dengan perintah `F.ItemJMobil.add(Trans.get(i))` dan

`ArrayList<Integer> frekMobil` yang merupakan nilai frekuensi jenis mobil tersebut diberi nilai 1 dengan perintah `F.frekMobil.add(1)`

Untuk jenis mobil yang sudah ada pada `F.ItemJMobil` maka index dari `F.ItemJMobil` yang mengandung jenis mobil tersebut, disimpan dalam variabel `indexArrFrek` yang bertipe *integer*. Kemudian frekuensi jenis mobil tersebut diambil pada `F.frekMobil` dengan index `indexArrFrek` dan disimpan pada `totFrekAwal` yang bertipe *integer*. Nilai frekuensi pada `F.frekMobil` index `indexArrFrek` dihapus dengan perintah `F.frekMobil.remove(indexArrFrek)` kemudian pada index `indexArrFrek` tersebut ditambahkan nilai frekuensi baru yaitu `totFrekAwal+1`. Fungsi ini selanjutnya mempunyai pengembalian nilai yaitu struktur data *class* `FrekJenisMobil`.

```
FrekJenisMobil getFrek(ArrayList<String> Trans)
{
    FrekJenisMobil F = new FrekJenisMobil();
    for(int i = 0; i< Trans.size(); i++)
    {
        if (F.ItemJMobil.contains(Trans.get(i)))
        {
            int indexArrFrek = F.ItemJMobil.indexOf(Trans.get(i));
            int totFrekAwal = F.frekMobil.get(indexArrFrek);
            F.frekMobil.remove(indexArrFrek);
            F.frekMobil.add(indexArrFrek,totFrekAwal+1);
        }
        else
        {
            F.ItemJMobil.add(Trans.get(i));
            F.frekMobil.add(1);
        }
    }
    return F;
}
```

Sourcecode 4. 2 Proses Koleksi Jenis Mobil

4.2.1.3. Implementasi Koleksi Kombinasi

Pada proses ini, dilakukan pengkoleksian kombinasi antara jenis kelamin dan umur yang ada pada data transaksi dan menghitung frekuensi kemunculan tiap kombinasi jenis kelamin dan umur. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.3. Proses dilakukan dengan mengkombinasi terlebih dahulu macam jenis kelamin dan kelompok umur yang digunakan, yang dilakukan pada fungsi kombinasi. Fungsi ini memiliki parameter input simbolUmur yang merupakan *array* bertipe data *String*. Fungsi ini dilakukan dengan melakukan pembacaan data simbolUmur. Untuk setiap pembacaan data simbolUmur dilakukan pembacaan data JK yang menyimpan macam jenis kelamin. Selanjutnya data JK *index* tersebut dan data simbolUmur *index* tersebut disimpan sementara pada titip yang merupakan struktur data *arraylist* bertipe *String*. Kemudian data titip tersebut disimpan pada Komb dan frekKomb untuk *index* yang sama diberi nilai 0. Nilai pengembalian dari fungsi ini adalah struktur data *arraylist* yang berisi *arraylist* dengan tipe data *String*.

Fungsi ItungFrekKomb digunakan untuk melakukan perhitungan frekuensi tiap kombinasi antara jenis kelamin dan umur pada data transaksi. Parameter input adalah ListKomb atau *arraylist* yang menyimpan list kombinasi antara jenis kelamin dan umur yaitu nilai pengembalian dari fungsi kombinasi, dan TransKomb atau *arraylist* yang menyimpan data transaksi yang telah ditransformasi hanya berisi parameter yang dibutuhkan yaitu jenis kelamin, umur dan jenis mobil.

```
ArrayList<ArrayList<String>> kombinasi(String[] simbolUmur)
{
    for(int i=0; i<simbolUmur.length; i++)
    {
        for(int j=0; j<JK.length; j++)
        {
            ArrayList<String> titip = new ArrayList<String>();
            titip.add(JK[j]);
            titip.add(simbolUmur[i]);
            Komb.add(titip);
            frekKomb.add(0);
        }
    }
}
```

```
    }  
    }  
    return Komb;  
} return hasil;  
}  
  
ArrayList<Integer> ItungFrekKomb  
    (ArrayList<ArrayList<String>> ListKomb,  
     ArrayList<ArrayList<String>> TransKomb){  
    for(int i=0; i<TransKomb.size(); i++){  
        for(int j=0; j<ListKomb.size(); j++){  
            {  
  
if(ListKomb.get(j).toString().equalsIgnoreCase(TransKomb.get(i).toString  
( )))  
            {  
                int index =  
ListKomb.indexOf(TransKomb.get(i));  
                int frek = frekKomb.get(index);  
                frek = frek+1;  
                frekKomb.remove(index);  
                frekKomb.add(index, frek);  
            }  
        }  
    }  
    return frekKomb;  
}
```

Sourcecode 4. 3 Proses Koleksi Kombinasi

Proses perhitungan frekuensi dilakukan dengan melakukan pembacaan tiap baris data pada TransKomb, kemudian untuk setiap *index* TransKomb, dilakukan pembacaan tiap baris data ListKomb. Apabila kombinasi jenis kelamin dan umur yang dibaca pada data TransKomb sama dengan data kombinasi jenis kelamin dan umur pada data ListKomb maka *index* ListKomb tersebut disimpan dan nilai frekuensi pada *arraylist* frekKomb pada *index* tersebut diambil dan disimpan sementara pada variabel *frek* yang selanjutnya ditambahkan 1 nilai *frek* tersebut, sedangkan nilai frekuensi pada *arraylist* frekKomb *index* tersebut dihapus `frekKomb.remove(index)` dan diisikan dengan nilai *frek*, `frekKomb.add(index,`

frek). Nilai pengembalian dari fungsi ini adalah *arraylist* dengan tipe data *Integer* yaitu *frekKomb*.

4.2.1.4. Implementasi Pencarian Jenis Mobil yang Frequent

Pada proses ini, dilakukan seleksi jenis mobil yang *frequent* yaitu yang memiliki *support* lebih dari atau sama dengan nilai minimum *support*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.4.

```
frequent1ItemsetJenisMobil OneItemsetJM
    (ArrayList<String> ItemJMobil,
     ArrayList<Double> supMobil, int minsup)
{
    frequent1ItemsetJenisMobil f = new
                                     frequent1ItemsetJenisMobil();
    double batasSup = (double)minsup/100;
    for(int i=0; i<supMobil.size();i++)
    {
        if(supMobil.get(i)>= batasSup)
        {
            f.OneJM.add(ItemJMobil.get(i));
            f.support.add(supMobil.get(i));
        }
    }
    return f;
}
```

Sourcecode 4. 4 Proses Pencarian Jenis Mobil yang Frequent

Pada fungsi *OneItemsetJM* dilakukan proses seleksi jenis mobil yang memenuhi nilai minimum *support* yang telah ditentukan. Parameter input yang dibutuhkan dari fungsi ini adalah *arraylist* bertipe data *String* yang menyimpan list jenis mobil *ItemJMobil* dan *arraylist* bertipe data *Double* yang menyimpan nilai *support* tiap jenis mobil *supMobil* serta prosentase nilai minimum *support* *minsup*. Proses dilakukan dengan melakukan pembacaan tiap baris data *arraylist* *supMobil* yang selanjutnya dibandingkan terhadap nilai *batasSup*. Nilai *batasSup* merupakan bentuk desimal dari nilai *minsup*. Apabila nilai *supMobil* *index* ke *i* memiliki nilai yang lebih besar dari atau sama dengan nilai *batasSup* maka *ItemJMobil* *index* *i* disimpan ke dalam *arraylist* *OneJM* pada *object class*

frequentItemsetJenisMobil *f* dan nilai *supMobil* *index* ke *i* disimpan ke dalam *arraylist* *support* pada *object class* *frequentItemsetJenisMobil* *f*. Nilai pengembalian dari fungsi ini adalah struktur data *class* *frequentItemsetJenisMobil*.

4.2.1.5. Implementasi Pencarian Kombinasi yang Frequent

Pada proses ini, dilakukan seleksi kombinasi jenis kelamin dan umur yang *frequent* yaitu yang memiliki *support* lebih dari atau sama dengan nilai minimum *support*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.5.

```
frequentItemsetKombinasi OneItemsetKomb
    (ArrayList<ArrayList<String>> Komb,
     ArrayList<Double> supKomb, int minsup)
{
    frequentItemsetKombinasi f = new frequentItemsetKombinasi();
    double batasSup = (double)minsup/100;
    for(int i=0; i<supKomb.size();i++)
    {
        if(supKomb.get(i)>= batasSup)
        {
            f.OneKomb.add(Komb.get(i));
            f.support.add(supKomb.get(i));
        }
    }
    return f;
}
```

Sourcecode 4. 5 Proses Pencarian Kombinasi yang Frequent

Dalam fungsi *OneItemsetKomb* dibutuhkan parameter input berupa *arraylist* berisi *arraylist* bertipe data *String* *Komb* yang menyimpan list kombinasi antara jenis kelamin dan umur dan *arraylist* bertipe data *Double* *supKomb* yang menyimpan nilai *support* tiap kombinasi jenis kelamin dan umur. Proses dilakukan dengan melakukan pembacaan tiap baris data *arraylist* *supKomb* yang selanjutnya dibandingkan terhadap nilai *batasSup*. Nilai *batasSup* merupakan bentuk desimal dari nilai *minsup*. Apabila nilai *supKomb* *index* ke *i* memiliki nilai yang lebih besar dari atau sama dengan nilai *batasSup* maka *Komb* *index* *i*

disimpan ke dalam *arraylist* *OneKomb* pada *object class* *frequentItemsetKombinasi* *f* dan nilai *supKomb index* ke *i* disimpan ke dalam *arraylist* *support* pada *object class* *frequentItemsetKombinasi* *f*. Nilai pengembalian dari fungsi ini adalah struktur data *class* *frequentItemset* *Kombinasi*.

4.2.2. Implementasi Tahap *Frequent Itemset*

Pada tahap ini, dilakukan kombinasi item yang menjadi *frequent 1-itemset* dan kemudian dilakukan perhitungan *support* untuk masing – masing kombinasi. Kombinasi yang memiliki nilai *support* lebih besar atau sama dengan nilai minimum *support* yang telah ditentukan, akan menjadi *frequent itemset*. Tahap *frequent itemset* terdiri dari 5 proses, yaitu proses seleksi data transaksi, proses pembentukan *FP-tree*, proses *conditional pattern base*, proses *conditional FP-tree*, dan proses koleksi *frequent itemset*.

4.2.2.1. Implementasi Seleksi Data Transaksi

Proses awal yang dilakukan adalah seleksi data transaksi. Proses ini bertujuan untuk menyimpan data transaksi yang memiliki jenis mobil yang *frequent* dan kombinasi antara jenis kelamin dan umur yang *frequent*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.6.

Fungsi *seleksiTrans* memiliki parameter input berupa *arraylist* yang berisi *arraylist* bertipe data *String* *TransKomb* yang menyimpan data transaksi berupa kombinasi antara jenis kelamin dan umur, *arraylist* bertipe data *String* *JMobil* yang menyimpan data transaksi berupa jenis mobil, *arraylist* berisi *arraylist* bertipe data *String* *OneKomb* yang menyimpan list kombinasi jenis kelamin dan umur yang *frequent*, dan *arraylist* bertipe data *String* *OneJM* yang menyimpan list jenis mobil yang *frequent*.

Proses dalam fungsi ini dilakukan dengan pembacaan tiap baris data *TransKomb*, untuk tiap baris data *TransKomb* dilakukan pembacaan tiap baris data *OneKomb*. Kemudian dilakukan pengecekan pada *OneKomb index* ke *j* apakah semua itemnya berisi sama dengan *TransKomb index* ke *i*. Jika sama maka dilakukan pembacaan tiap baris data pada *OneJM*. Setiap baris data *OneJM* yang

dibaca, dilakukan pengecekan apakah *OneJM index* k berisi jenis mobil yang sama dengan *JMobil* ke i. Apabila sama, maka dilakukan penyimpanan data *TransKomb index* ke i, data *JMobil index* ke i ke *arraylist* bertipe *String* titip. Nilai pengembalian dari fungsi ini adalah *arraylist* berisi *arraylist* bertipe data *String* tempTrans pada *object class* TemporaryTrans.

```

ArrayList<ArrayList<String>> seleksiTrans
    (ArrayList<ArrayList<String>> TransKomb, ArrayList<String>
    JMobil,
    ArrayList<ArrayList<String>> OneKomb, ArrayList<String>
    OneJM)
{
    TemporaryTrans t = new TemporaryTrans();
    for(int i=0;i<TransKomb.size();i++){
        for(int j=0;j<OneKomb.size();j++){
            if(OneKomb.get(j).containsAll(TransKomb.get(i))){
                for(int k=0;k<OneJM.size();k++){
                    if(OneJM.get(k).contains(JMobil.get(i))){
                        ArrayList<String> titip = new
ArrayList<String>();
                        titip.add(TransKomb.get(i).toString());
                        titip.add(JMobil.get(i));
                        t.tempTrans.add(titip);
                    }
                }
            }
        }
    }
    return t.tempTrans;
}

```

Sourcecode 4. 6 Proses Seleksi Data Transaksi

4.2.2.2.Implementasi Pembentukan *FP-tree*

Proses ini bertujuan untuk menyimpan data transaksi sehingga tidak perlu dilakukan pembacaan data transaksi berulang kali. Pada proses ini kombinasi jenis kelamin dan umur, dan jenis mobil setiap transaksi dimasukkan ke dalam *tree* sebagai satu lintasan. Node dari *FP-tree* menyimpan informasi nama item, bobot, kedalaman, indeks pada *header table*, indeks node, node *parent*, node *sibling*, dan node *link*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.7.

Pembentukan *FP-tree* diimplementasikan pada *class FPTree* yaitu pada fungsi *insert*. Pada fungsi ini, parameter inputan berupa `ArrayList<String>` items yang merupakan kombinasi antara jenis kelamin dan umur, dan jenis mobil yang dimiliki pada satu baris data transaksi dan `int count` yang merupakan jumlah bobot tiap data items tersebut. Proses awal adalah inisialisasi *node root*. Kemudian dilakukan pembacaan tiap index data pada items, yang berarti kombinasi antara jenis kelamin dan umur, dan jenis mobil pada index tersebut dimasukkan ke dalam struktur *FP-tree*. Kombinasi antara jenis kelamin dan umur selalu berada pada *index* pertama dalam items, *index* selanjutnya diikuti dengan jenis mobil.

```
void insert(ArrayList<String> items, int count)
{
    FPTreeNode current_node = root;

    for (int index = 0; index < items.size(); index++)
    {
        int entry_index = ((Integer) item2index.get ((items.get
            (index)))));
        header[entry_index].count += count;
        FPTreeNode walker = current_node.child;

        for ( ; walker != null; walker = walker.sibling)
            if (walker.item.equalsIgnoreCase(items.get(index)))
                break;
        if (walker == null)
        {
            if (current_node.child != null)
                hasMultiplePaths = true;
            count_nodes++;
            FPTreeNode new_node = new FPTreeNode (items.get
                (index), count,
                index + 1, entry_index, count_nodes,
                current_node, current_node.child,
                header[entry_index].head);
            header[entry_index].head = new_node;
            current_node.child = new_node;
            current_node = new_node;
            n.displayNode(current_node);
        }
    }
}
```

```

he.DisplayHeaderEntry((Integer)item2index.get(items.get(index)),
                      header[entry_index].item,header[entry_index].count,
                      header[entry_index].head);
    }
    else
    {
        walker.count += count;
        n.displayNode(walker);
    }
he.DisplayHeaderEntry((Integer)item2index.get(items.get(index)),
                      walker.item,header[entry_index].count,walker);
    current_node = walker;
}
}
}

```

Sourcecode 4. 7 Proses Pembentukan FP-Tree.

4.2.2.3. Implementasi *Conditional Pattern Base*

Pada proses ini, dilakukan pengkoleksian lintasan yang dimiliki tiap *suffix*, dimana *suffix* merupakan jenis mobil yang menjadi *frequent 1-itemset*. Selain itu, proses ini juga menyimpan bobot tiap node pada lintasan tersebut dengan tujuan untuk digunakan dalam menghitung nilai *support* tiap kombinasi antara jenis kelamin dan umur yang merupakan *prefix* pada tiap lintasan. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.8.

```

Prefix condPatternBase(String suffix, ArrayList<Lintasan> p)
{
    Prefix hasil = new Prefix();
    ArrayList itemPref = new ArrayList<String>();
    ArrayList itemSuf = new ArrayList<String>();
    ArrayList bobotPref = new ArrayList<Integer>();
    for(int i=0;i<(p.size());i++){
        if((p.get(i).itemPath.get(p.get(i).itemPath.size()-
1)).equalsIgnoreCase(suffix))
        {
            itemSuf.add(suffix);
            for(int j=0;j<p.get(i).itemPath.size()-1;j++)
            {

```

```

        itemPref.add(p.get(i).itemPath.get(j));
        bobotPref.add(p.get(i).bobot.get(j));
    }
    hasil = new Prefix(itemPref,itemSuf,bobotPref);
}
} return hasil;
}

```

Sourcecode 4. 8 Proses Conditional Pattern Base

Fungsi `condPatternBase` memiliki parameter inputan berupa suffix yang bertipe data *String*, dan *arraylist* yang menyimpan data lintasan *p*. Proses dilakukan dengan melakukan pembacaan tiap baris data lintasan *p*. Untuk setiap *index* data lintasan *p* dilakukan pengecekan apakah `itemPath` terakhir dari lintasan *p* *index* tersebut sama dengan suffix. Apabila sama maka suffix disimpan ke dalam *arraylist* bertipe data *String* `itemSuf` dan item pada `itemPath` selain suffix disimpan ke dalam *arraylist* bertipe data `itemPref` yang berarti merupakan *prefix* dari lintasan *index* tersebut dan bobot dari lintasan tersebut disimpan ke dalam *arraylist* bertipe data *Integer* `bobotPref`. Nilai pengembalian dari fungsi ini adalah struktur data *class* `Prefix` dengan nama *object* `hasil`.

4.2.2.4. Implementasi Conditional FP-Tree

Proses ini bertujuan untuk mengetahui nilai *support* tiap lintasan *prefix* tiap *suffix*. Pada proses ini, dilakukan pengecekan nilai *support* setiap lintasan *prefix* tiap *suffix* terhadap nilai minimum *support* yang telah ditentukan. Lintasan *prefix* yang memiliki nilai *support* lebih besar atau sama dengan nilai minimum *support* akan menjadi *frequent itemset*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.9.

Fungsi `condFPtree` memiliki parameter inputan berupa kombinasi jenis kelamin dan umur, dan jenis mobil yang dideklarasikan sebagai `ArrayList<Integer>` dengan nama variabel `items` dan data lintasan `ArrayList<Path>` `pa`. Proses dilakukan dengan mengecek pada setiap data lintasan sampai ditemukan data lintasan yang terdiri dari kombinasi antara jenis kelamin dan umur, dan jenis mobil yang sama pada `items`. Setiap data lintasan yang sesuai

maka nilai bobotnya disimpan pada variabel bobot yang bertipe *double*. Setelah pengecekan dilakukan pada semua data lintasan, maka dihitung nilai *support* hubungan antara kombinasi antara jenis kelamin dan umur dengan jenis mobil tersebut dengan cara total bobot tersebut yang disimpan dalam variabel bobot dibagi dengan jumlah transaksi yang disimpan dalam variabel jumlahData. Kemudian kombinasi antara jenis kelamin dan umur disimpan dalam `ArrayList<String>` itemset, jenis mobil disimpan dalam `ArrayList<String>` suffix dan nilai *support* hubungan keduanya disimpan dalam `ArrayList<Double>` bobot. Nilai pengembalian dari fungsi ini adalah struktur data *class* `FrequentItemset`.

```
Itemset condFPtree(ArrayList<String> items,
                  int jumlahData, ArrayList<Lintasan> p)

    double bobot = 0;
    FrequentItemset f= new FrequentItemset();
    for(int i=0; i<p.size();i++)
    {
        int nilai = 0;
        if(p.get(i).itemPath.get(1).equalsIgnoreCase(items.get(1))){
            if(p.get(i).itemPath.get(0).equalsIgnoreCase(items.get(0))){
                nilai+=1;
                if(nilai == items.size()-1){
                    bobot = p.get(i).bobot.get(1);
                }
            }
        }
        f.frek.add(bobot);
        bobot = (bobot / (double)jumlahData);
        f.itemset.add(items.get(0));
        f.suffix.add(items.get(items.size()-1));
        f.bobot.add(bobot);
    }
    return f;
```

Sourcecode 4. 9 Proses Conditional FP-Tree

4.2.2.5. Implementasi Koleksi *Frequent Itemset*

Pada proses ini, dilakukan pengecekan nilai *support* setiap lintasan *prefix* tiap *suffix* terhadap nilai minimum *support* yang telah ditentukan. Lintasan *prefix* yang memiliki nilai *support* lebih besar atau sama dengan nilai minimum *support* akan menjadi *frequent itemset*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.10.

Fungsi koleksiFreqItemset berfungsi untuk mengkoleksi hasil dari proses *Conditional FP-tree* yang memiliki nilai *support* lebih besar atau sama dengan nilai *minimum support* yang telah ditentukan. Proses dilakukan dengan pembacaan tiap baris data dari data FI yang dideklarasikan `ArrayList<FrequentItemset>`. Kemudian bobot tiap baris data FI tersebut dibandingkan terhadap nilai minimum *support*. Nilai bobot data FI yang memiliki nilai lebih besar atau sama dengan dari nilai minimum *support* disimpan ke dalam *object class* `FrequentItemset` titip. Data *prefix* disimpan pada *arraylist* itemset, data *suffix* disimpan pada *arraylist* suffix, data *support* disimpan pada *arraylist* bobot, dan data frekuensi disimpan pada *arraylist* frek pada *object class* `FrequentItemset` titip. Nilai pengembalian dari fungsi ini adalah `FrequentItemset`.

```

FrequentItemset koleksiFreqItemset(ArrayList<FrequentItemset> FI, int
minsup)
{
    FrequentItemset titip = new FrequentItemset();
    double batasSup = (double)minsup/100;
    for(int i=0;i<FI.size();i++)
    {
        if(FI.get(i).bobot.get(0)>=batasSup)
        {
            titip.itemset.add(FI.get(i).itemset.get(0));
            titip.suffix.add(FI.get(i).suffix.get(0));
            titip.bobot.add(FI.get(i).bobot.get(0));
            titip.frek.add(FI.get(i).frek.get(0));
        }
    } return titip;
}

```

Sourcecode 4. 10 Proses Koleksi Frequent Itemset

4.2.3. Implementasi Tahap *Generate Rule*

Pada tahap ini, dilakukan perhitungan nilai *confidence* pada setiap *frequent itemset*. Nilai *confidence* didapatkan dari nilai *support frequent itemset* dibagi dengan nilai *support* bagian *antecedent*. *Frequent Itemset* yang memiliki nilai *confidence* lebih besar atau sama dengan nilai minimum *confidence* yang telah ditentukan akan menjadi *rule*. Tahap *generate rule* terdiri dari 2 proses, yaitu perhitungan *confidence* dan proses seleksi *rule*.

4.2.3.1. Implementasi Perhitungan *Confidence*

Proses ini merupakan proses awal dari tahap *generate rule*. Pada proses ini, tiap *frequent itemset* dihitung nilai *confidence*-nya. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.11. Fungsi *ConfidenceFI* berfungsi untuk menghitung nilai *confidence* untuk setiap data *frequent itemset*. Parameter inputan yang dibutuhkan pada fungsi ini adalah data *frequent itemset* sebagai variabel hasil, data *support frequent 1-itemset* kombinasi antara jenis kelamin dan umur sebagai variabel *f1Komb*.

Setiap data *frequent itemset* diambil *prefix*-nya yang disimpan sementara pada `ArrayList<String>` *itemset* pada data hasil. *Support prefix* dari *frequent itemset* ini yang akan menjadi *support antecedent* dalam perhitungan nilai *confidence* suatu *frequent itemset* sehingga dilakukan pencarian terhadap data *frequent 1-itemset* kombinasi antara jenis kelamin dan umur sebagai variabel *f1Komb* pada `arraylist OneKomb` yang memiliki item yang sama dengan *prefix* dari *frequent itemset*. Hasil pencarian tersebut berupa nilai *support* dari *frequent 1-itemset* yang disimpan dalam *supAntecedent* dan untuk selanjutnya digunakan untuk perhitungan nilai *confidence* yaitu sebagai pembagi dari nilai bobot *frequent itemset confidence* = $\text{supFI}/\text{supAntecedent}$.

Hasil perhitungan selanjutnya disimpan ke dalam struktur data *class Confidence* dengan nama *object* *hasilCo* menggunakan *method* *add*. Nilai pengembalian dari fungsi ini adalah struktur data *class Confidence*.

```

Confidence ConfidenceFI(FrequentItemset hasil, frequentItemsetKombinasi
f1Komb){
    Confidence hasilCo = new Confidence();
    double confidence= 0;
    double supAntecedent = 0;
    double supFI = 0;
    for(int i=0;i<hasil.itemset.size();i++){
        supFI = hasil.bobot.get(i);
        for(int j=0; j<f1Komb.OneKomb.size();j++){
            if(f1Komb.OneKomb.get(j).toString().equalsIgnoreCase(hasil.itemset.get(i)
            )){
                supAntecedent = f1Komb.support.get(j);
                confidence = supFI/supAntecedent;
                hasilCo.prefix.add(hasil.itemset.get(i));
                hasilCo.suffix.add(hasil.suffix.get(i));
                hasilCo.confidence.add(confidence);
                hasilCo.support.add(supFI);
                hasilCo.supAntecedent.add(supAntecedent);
            }
        }
    }return hasilCo;
}

```

Sourcecode 4. 11 Proses Perhitungan Confidence

4.2.3.2. Implementasi Seleksi Rule

Pada proses ini, *frequent itemset* yang memiliki nilai *confidence* lebih besar atau sama dengan minimum *confidence* dibangkitkan sebagai *rule*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.12.

Fungsi SeleksiRule terdiri dari parameter inputan data Confidence hasilCo, dan prosentase minimum *confidence* minconf. Proses dalam fungsi ini dilakukan dengan melakukan pembacaan nilai *confidence* data hasilCo yang disimpan dalam *arraylist* confidence. Selanjutnya, nilai *confidence* tersebut akan dibandingkan terhadap nilai minimum *confidence* batasConf. Jika nilai confidence *index* i lebih besar atau sama dengan nilai batasConf maka data hasilCo akan disimpan pada *object class* R dari *class* Rule.

Selanjutnya untuk *arraylist* freqitem dan keterangan pada fungsi ini akan diisi pada fungsi PenulisanRule karena *prefix* dan *suffix* akan ditulis dalam bentuk

aturan *prefix* -> *suffix* yang selanjutnya disimpan dalam *arraylist* freqitem dan untuk keterangan akan diisi dengan keterangan *rule* dari penyimbolan kombinasi jenis kelamin dan umur, seperti [P, 1] yang berarti jenis kelamin Pria dan kelompok umur 1 yaitu 20 tahun sampai 30 tahun. Nilai pengembalian dari fungsi ini adalah *object* dari struktur data *class Rule* yaitu R.

```
Rule SeleksiRule(Confidence hasilCo, double minconf){
    Rule R = new Rule();
    batasConf = minconf/100;
    for(int i=0;i<hasilCo.confidence.size();i++){
        if(hasilCo.confidence.get(i)>=batasConf){
            R.suffix.add(hasilCo.suffix.get(i));
            R.prefix.add(hasilCo.prefix.get(i));
            R.confidence.add(hasilCo.confidence.get(i));
            R.freqitem.add("");
            R.keterangan.add("");
        }
    }
    return R;
}

Rule PenulisanRule(Rule R){
    for(int i=0;i<R.prefix.size();i++){
        String Freqit = R.prefix.get(i)+"->"+R.suffix.get(i);
        R.freqitem.remove(i);
        R.freqitem.add(i, Freqit);
        String ket= "";
        if(R.prefix.get(i).toString().equalsIgnoreCase("[P, 1]")){
            ket = "Jika Pembeli PRIA dan UMUR 20-30 th" +
                " Maka Jenis Mobil "+R.suffix.get(i);
            R.keterangan.remove(i);
            R.keterangan.add(i, ket);
        }else if(R.prefix.get(i).toString().equalsIgnoreCase("[P,
2]")){
            ket = "Jika Pembeli PRIA dan UMUR 31-40 th" +
                " Maka Jenis Mobil "+R.suffix.get(i);
            R.keterangan.remove(i);
            R.keterangan.add(i, ket);
        }else if(R.prefix.get(i).toString().equalsIgnoreCase("[P,
3]")){
            ket = "Jika Pembeli PRIA dan UMUR 41-50 th" +
                " Maka Jenis Mobil "+R.suffix.get(i);
```

```

R.keterangan.remove(i);
R.keterangan.add(i, ket);
}else if(R.prefix.get(i).toString().equalsIgnoreCase("[P,
4]")){
    ket = "Jika Pembeli PRIA dan UMUR diatas 50 th" +
        " Maka Jenis Mobil "+R.suffix.get(i);
    R.keterangan.remove(i);
    R.keterangan.add(i, ket);
}else if(R.prefix.get(i).toString().equalsIgnoreCase("[W,
1]")){
    ket = "Jika Pembeli WANITA dan UMUR 20-30 th" +
        " Maka Jenis Mobil "+R.suffix.get(i);
    R.keterangan.remove(i);
    R.keterangan.add(i, ket);
}else if(R.prefix.get(i).toString().equalsIgnoreCase("[W,
2]")){
    ket = "Jika Pembeli WANITA dan UMUR 31-40 th" +
        " Maka Jenis Mobil "+R.suffix.get(i);
    R.keterangan.remove(i);
    R.keterangan.add(i, ket);
}else if(R.prefix.get(i).toString().equalsIgnoreCase("[W,
3]")){
    ket = "Jika Pembeli WANITA dan UMUR 41-50 th" +
        " Maka Jenis Mobil "+R.suffix.get(i);
    R.keterangan.remove(i);
    R.keterangan.add(i, ket);
}else if(R.prefix.get(i).toString().equalsIgnoreCase("[W,
4]")){
    ket = "Jika Pembeli WANITA dan UMUR diatas 50th" +
        " Maka Jenis Mobil "+R.suffix.get(i);
    R.keterangan.remove(i);
    R.keterangan.add(i, ket);
}
}
return R;
}

```

Sourcecode 4. 12 Proses Seleksi Rule

4.2.4. Implementasi Tahap *Lift Ratio*

Tahap *lift ratio* terdiri dari satu proses yaitu dilakukan perhitungan nilai *lift ratio* pada *frequent itemset* yang menjadi *rule*. Nilai *lift ratio* didapatkan dari

hasil pembagian dari nilai *confidence* terhadap nilai *support* item yang menjadi *consequent* (bagian maka). Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.13.

```

LiftRatio hitungLift(Rule R, frequentItemsetJenisMobil flJM){
    LiftRatio LR = new LiftRatio();
    double lift= 0;
    double supConsequent = 0;
    for(int i=0; i<R.freqitem.size();i++){
        for(int j=0; j<flJM.OneJM.size();j++){
            if(flJM.OneJM.get(j).toString().equalsIgnoreCase(R.suffix.get(i))){
                supConsequent = flJM.support.get(j);
                lift = R.confidence.get(i)/supConsequent;
                LR.freqitem.add(R.freqitem.get(i));
                LR.lift.add(lift);
            }
        }
    }
    return LR;
}

```

Sourcecode 4. 13 Proses Lift Ratio

Fungsi *hitungLift* dimulai dengan mencari nilai *support* item yang menjadi *consequent* (bagian maka) dari index data *confidence* yaitu *support* dari jenis mobil. Nilai *support* didapatkan dari data *support frequent 1-itemset* jenis mobil yang dideklarasikan sebagai parameter inputan yaitu *frequentItemsetJenisMobil flJM*. Nilai *lift ratio* didapatkan dari hasil pembagian dari nilai *confidence frequent itemset* tersebut dengan nilai *support consequent* dari *frequent itemset* tersebut. Nilai *lift ratio* disimpan dalam *ArrayList<Double> lift*, sedangkan *frequent itemset* disimpan dalam *ArrayList<String> freqitem* pada *object LR* dari struktur data class *LiftRatio*.

4.3. Implementasi Antarmuka

Implementasi antarmuka sistem seperti yang telah dijelaskan dalam rancangan antarmuka pada subbab 3.4, terdiri dari bagian *input*, bagian *output*, tombol *browse*, tombol proses, tombol *refresh*, tombol *export*. Bagian input terdiri

dari tiga data *input*, yaitu data *input file* data transaksi bertipe xls, data *input* minimum *support*, dan data *input* minimum *confidence*. Bagian output terdiri dari *text area* untuk keterangan nama file hasil *export output*, dan enam submenu yaitu submenu Data Transaksi, submenu *One Itemset*, submenu *Frequent 1-itemset*, submenu *Frequent Itemset*, submenu *Association Rule*, dan submenu Akurasi.

Bagian *input* yang pertama adalah *textfield* yang berisi *file* data transaksi yang dipilih. *Input* yang kedua adalah pilihan *radio button* untuk minimum *support*. *Radio button* pertama adalah untuk *input* minimum *support* dalam bentuk prosentase, sedangkan *radio button* kedua untuk *input* minimum *support* dalam bentuk jumlah minimal frekuensi dalam transaksi. *Input* yang ketiga adalah *input* nilai minimum *confidence* dalam bentuk prosentase. Bagian *ouput* submenu Data Transaksi muncul tabel transaksi yang berisi data transaksi untuk data *file* yang telah dipilih seperti yang diilustrasikan pada Gambar 4.1.

**PENERAPAN METODE ASSOCIATION RULE DENGAN ALGORITMA FP-GROWTH
UNTUK MENGETAHUI KARAKTERISTIK PEMBELI MOBIL**

Pilih File :
mobilDaihatsuxts

Minimum Support :
 in percentage (%)
 number of transaction

Minimum Confidence :
 range 1 - 100 %

Export Hasil Proses

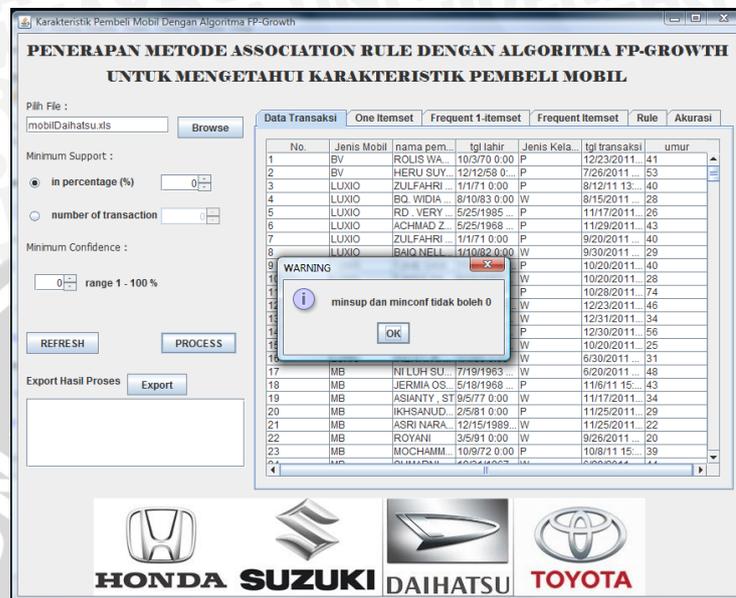
No.	Jenis Mobil	nama pem.	tgl lahir	Jenis Kela.	tgl transaksi	umur
1	BV	ROLIS WA.	10/3/70 0 00	P	12/2/2011	41
2	BV	HERU SUY.	12/12/58 0	P	7/28/2011	53
3	LUXIO	ZULFAHRI	1/1/71 0 00	P	8/12/11 13	40
4	LUXIO	BO. WIDIA	8/10/83 0 00	W	8/15/2011	28
5	LUXIO	RD. VERY	5/25/1985	P	11/17/2011	26
6	LUXIO	ACHMAD Z.	5/25/1988	P	11/29/2011	23
7	LUXIO	ZULFAHRI	1/1/71 0 00	P	9/20/2011	40
8	LUXIO	BAIQ NELL.	1/10/82 0 00	W	9/30/2011	29
9	LUXIO	TJHIE DAVI.	11/25/1971	P	10/20/2011	40
10	LUXIO	TJIANG FA.	8/28/1983	W	10/20/2011	28
11	LUXIO	I NENGARI	8/7/37 0 00	P	10/28/2011	74
12	LUXIO	HJ. BAIQ M.	12/31/1965	W	12/2/2011	46
13	LUXIO	JOICE JOH.	6/13/1977	W	12/31/2011	34
14	LUXIO	DRS. IDA	2/20/1955	P	12/30/2011	56
15	LUXIO	IKA RIZKY	2/15/1988	W	10/20/2011	25
16	LUXIO	IRENI R.	1/18/0 0 00	W	9/30/2011	31
17	MB	NILUH SU.	7/19/1963	W	9/20/2011	48
18	MB	JERMA OS.	5/18/1968	P	11/8/11 15	43
19	MB	ASIANTY.	ST 9/5/77 0 00	W	11/17/2011	34
20	MB	IKHSANUD.	2/5/81 0 00	P	11/25/2011	29
21	MB	ASRI NARA.	12/15/1989	W	11/25/2011	22
22	MB	ROYANI	3/5/91 0 00	W	9/26/2011	20
23	MB	MOCHAMM.	10/9/72 0 00	P	10/8/11 15	39
24	MB	SUMARNI	10/24/1967	W	6/28/11 15	44

HONDA SUZUKI DAIHATSU TOYOTA

Gambar 4. 1 Tampilan Data Transaksi

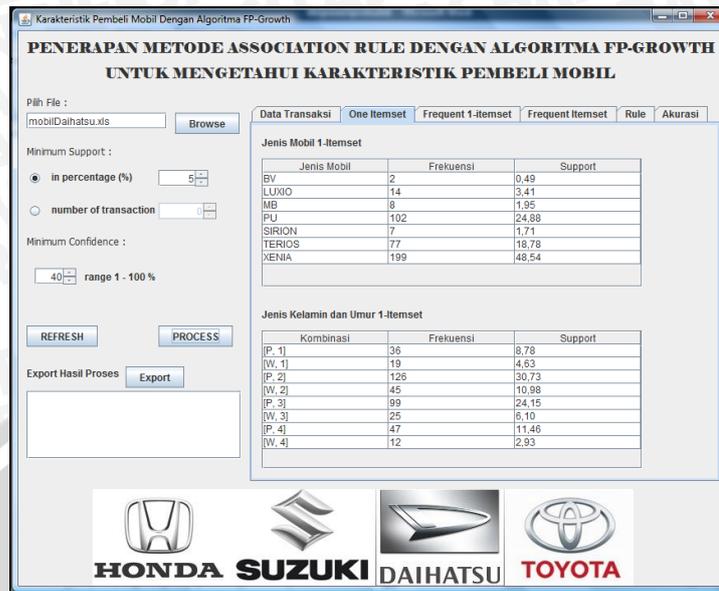
Aturan untuk data inputan minimum *support* dalam bentuk prosentase harus diisi minimal 1%, sedangkan data inputan minimum *support* dalam bentuk jumlah transaksi harus diisi minimal 1% dari total transaksi. Begitu pula untuk inputan minimum *confidence* harus diisi minimal 1%. Jika inputan tersebut kurang dari

1% maka akan keluar dialog *box* peringatan bila tombol proses ditekan seperti yang diilustrasikan pada Gambar 4.2.



Gambar 4. 2 Tampilan Dialog Box Peringatan

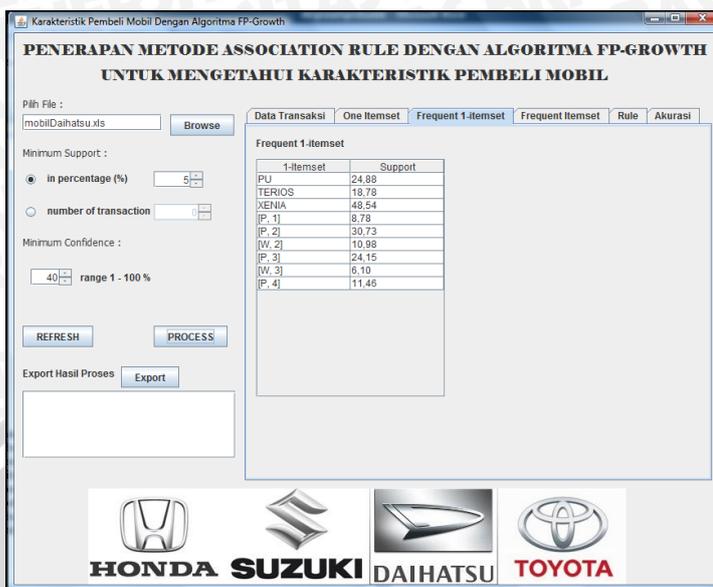
Setelah data inputan terisi dengan benar, maka ketika tombol proses ditekan maka data *output* pada submenu *one itemset*, *frequent 1-itemset*, *frequent itemset*, *rule*, dan akurasi akan ditampilkan pada masing – masing tabel. Submenu *One Itemset* terdiri dari dua tabel, tabel yang pertama adalah tabel *one itemset* untuk jenis mobil, sedangkan tabel yang kedua adalah tabel *one itemset* untuk kombinasi antara jenis kelamin dan umur. Tampilan submenu *one itemset* diilustrasikan pada Gambar 4.3. Tabel jenis mobil 1-itemset merupakan tabel yang menampilkan frekuensi dan *support* tiap jenis mobil, sedangkan tabel jenis kelamin dan umur 1-itemset merupakan tabel yang menampilkan frekuensi dan *support* tiap kombinasi antara jenis kelamin dan umur.



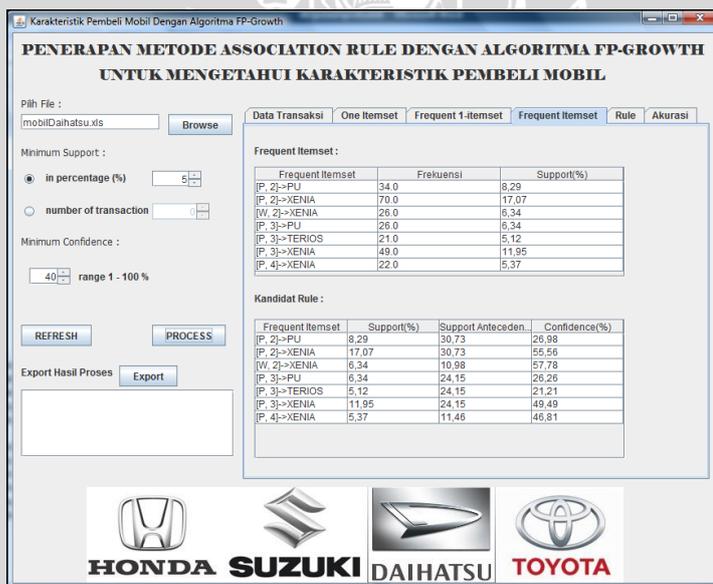
Gambar 4. 3 Tampilan Submenu One Itemset

Submenu *Frequent 1-itemset* terdiri dari tabel *frequent 1-itemset*. Tampilan submenu *frequent 1-itemset* diilustrasikan pada Gambar 4.4. Tabel *frequent 1-itemset* ini menampilkan jenis mobil atau kombinasi antara jenis kelamin dan umur yang lolos minimum *support*, yaitu yang memiliki nilai *support* lebih besar dari atau sama dengan minimum *support*.

Submenu *Frequent Itemset* terdiri dari tabel *frequent itemset* dan tabel kandidat rule. Tampilan submenu *frequent itemset* diilustrasikan pada Gambar 4.5. Tabel *frequent itemset* ini menampilkan kombinasi antara jenis kelamin dan umur, dan jenis mobil yang lolos minimum *support*, yaitu yang memiliki nilai *support* lebih besar dari atau sama dengan minimum *support*. Tabel kandidat *rule* menampilkan hasil perhitungan nilai *confidence* setiap *frequent itemset*.



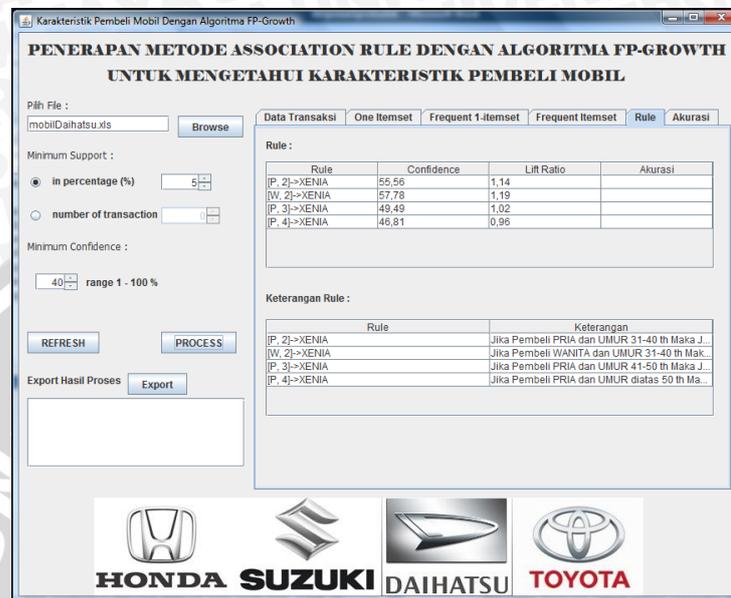
Gambar 4. 4 Tampilan Submenu Frequent 1-Itemset



Gambar 4. 5 Tampilan Submenu Frequent Itemset

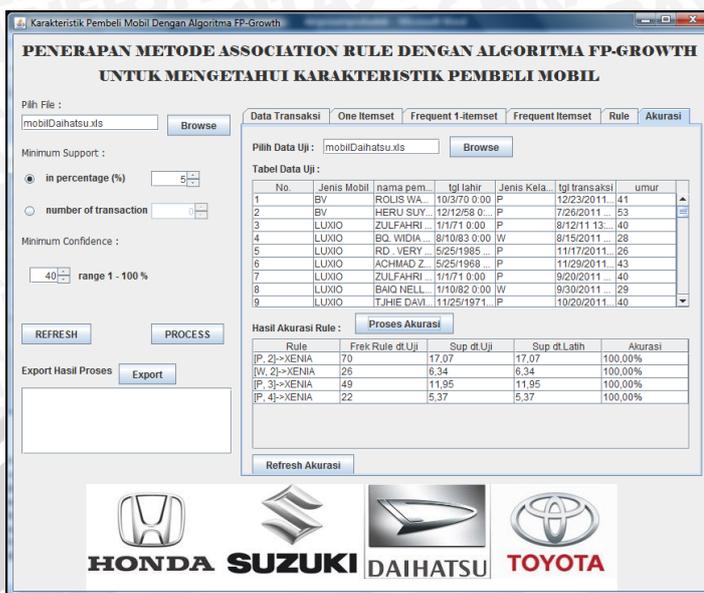
Submenu *Rule* terdiri dari tabel *rule* dan tabel keterangan *rule*. Tampilan submenu *rule* diilustrasikan pada Gambar 4.6. Tabel *rule* ini menampilkan *frequent itemset* yang lolos *minimum confidence*, yaitu yang memiliki nilai *confidence* lebih besar dari atau sama dengan *minimum confidence*. Pada tabel *rule* kolom akurasi kosong selama proses uji akurasi belum dilakukan. Proses uji

akurasi terdapat pada submenu akurasi. Tabel keterangan *rule* menampilkan deskripsi dari simbolik *rule*.

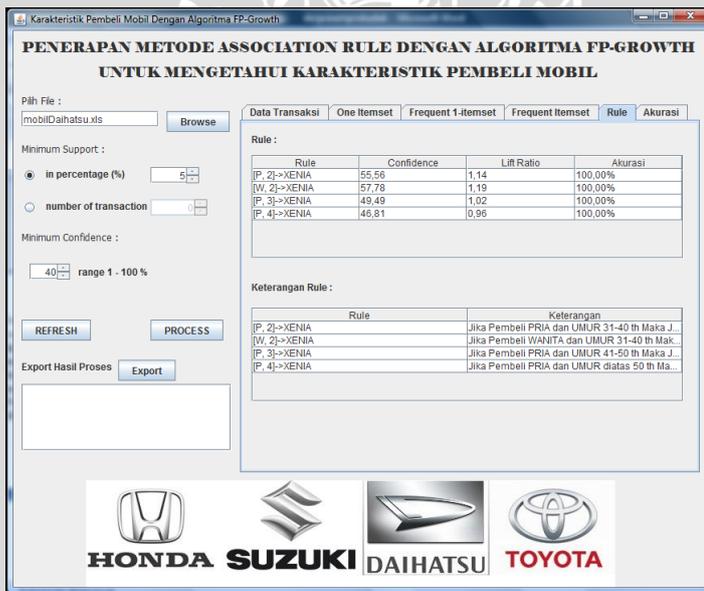


Gambar 4. 6 Tampilan Submenu Rule

Submenu akurasi terdiri dari data *input file* data uji, tombol *browse*, tombol proses akurasi dan tombol *refresh*. Selain itu terdapat *output* berupa tabel data uji dan tabel akurasi. Tampilan submenu akurasi diilustrasikan pada Gambar 4.7. Tabel data uji ini menampilkan data transaksi sebagai data uji yang telah dipilih dengan menggunakan tombol *browse*, sedangkan tabel akurasi menampilkan hasil perhitungan akurasi *rule* yang berdasarkan frekuensi kemunculan *rule* terhadap data uji setelah tombol proses akurasi ditekan. Tombol proses akurasi ini juga memberikan perubahan pada tabel *rule* yang ada pada submenu *rule*, yaitu kolom akurasi yang sebelumnya kosong menjadi terisikan data akurasi seperti yang diilustrasikan pada Gambar 4.8.



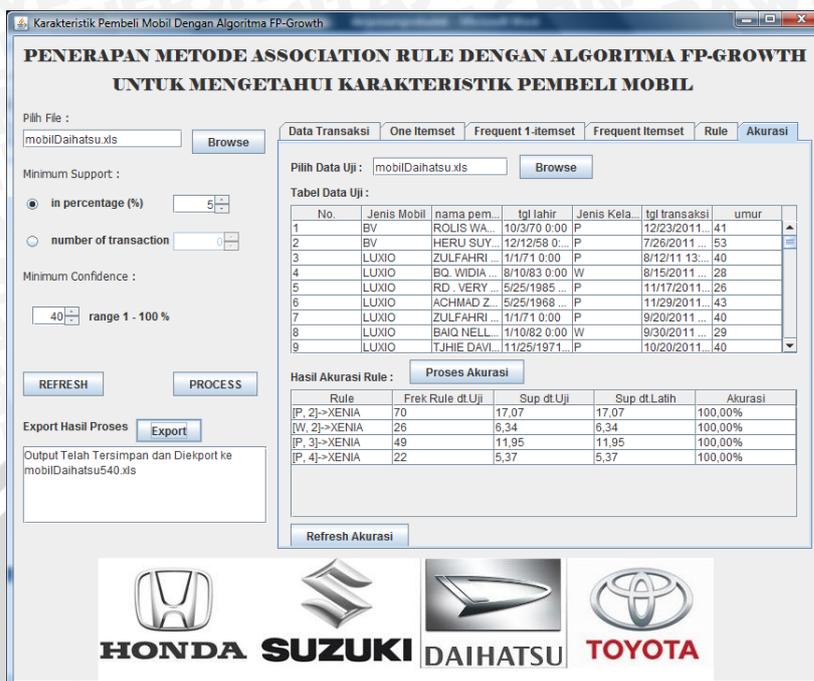
Gambar 4. 7 Tampilan Submenu Akurasi



Gambar 4. 8 Tampilan Tabel Rule Setelah Uji Akurasi

Selanjutnya, setelah semua proses dilakukan dan semua *output* ditampilkan, dilakukan proses *export* data *output* ke dalam *file excel* yaitu dengan menekan tombol *export*. Setelah tombol *export* ditekan maka akan tampil keterangan pada *textarea* bahwa data *output* telah di *export* dengan nama *file* yang tertera pada keterangan di *textarea* seperti yang ditampilkan pada Gambar 4.9.





Gambar 4. 9 Tampilan Keterangan Export Data Output

4.4. Pengujian Sistem

Pengujian sistem dilakukan untuk masing – masing dealer, yaitu dealer Daihatsu, dealer Toyota, dealer Suzuki dan dealer Honda. Masing – masing dealer terdiri dari 3 macam pengujian support, yaitu 5%, 7%, dan 9%. Selain itu, terdiri 3 macam pengujian confidence, yaitu 40%, 45%, dan 50%.

Pengujian pertama dilakukan untuk mengetahui tingkat kekuatan (lift ratio) rule yang dihasilkan. Hasil uji tingkat kekuatan (lift ratio) rule yang dihasilkan dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Hasil Uji Tingkat Kekuatan (Lift Ratio) Rule yang Dihasilkan.

Dealer	MinSup (%)	Min Conf (%)	Rule	Conf (%)	Lift Ratio
Daihatsu	5	40	[P, 2]>XENIA	47.29	0.97
			[W, 2]->XENIA	53.66	1.11
			[P, 3]->XENIA	44.94	0.93
			[P, 4]->XENIA	65.22	1.34
	45	45	[P, 2]->XENIA	47.29	0.97
			[W, 2]->XENIA	53.66	1.11
			[P, 4]->XENIA	65.22	1.34
	50	50	[W, 2]->XENIA	53.66	1.11



			[P, 4]->XENIA	65.22	1.34
	7	40	[P, 2]->XENIA	47.29	0.97
			[P, 3]->XENIA	44.94	0.93
			[P, 4]->XENIA	65.22	1.34
			[P, 2]->XENIA	47.29	0.97
		45	[P, 4]->XENIA	65.22	1.34
			[P, 4]->XENIA	65.22	1.34
	9	40	[P, 2]->XENIA	47.29	0.97
			[P, 3]->XENIA	44.94	0.93
		45	[P, 2]->XENIA	47.29	0.97
		50	-	-	-
Toyota	5	40	[P, 2]->AVANZA 1300 CC	53.95	1.01
			[P, 3]->AVANZA 1300 CC	54.88	1.03
			[W, 3]->AVANZA 1300 CC	59.38	1.12
			[P, 4]->AVANZA 1300 CC	47.83	0.90
		45	[P, 2]->AVANZA 1300 CC	53.95	1.01
			[P, 3]->AVANZA 1300 CC	54.88	1.03
			[W, 3]->AVANZA 1300 CC	59.38	1.12
			[P, 4]->AVANZA 1300 CC	47.83	0.90
		50	[P, 2]->AVANZA 1300 CC	53.95	1.01
			[P, 3]->AVANZA 1300 CC	54.88	1.03
			[W, 3]->AVANZA 1300 CC	59.38	1.12
			CC		
	7	40	[P, 2]->AVANZA 1300 CC	53.95	1.01
			[P, 3]->AVANZA 1300 CC	54.88	1.03
			CC		
		45	[P, 2]->AVANZA 1300 CC	53.95	1.01
			[P, 3]->AVANZA 1300 CC	54.88	1.03
			CC		
	50	[P, 2]->AVANZA 1300 CC	53.95	1.01	
		[P, 3]->AVANZA 1300 CC	54.88	1.03	
		CC			
9	40	[P, 2]->AVANZA 1300	53.95	1.01	

			CC		
			[P, 3]->AVANZA 1300	54.88	1.03
			CC		
		45	[P, 2]->AVANZA 1300	53.95	1.01
			CC		
			[P, 3]->AVANZA 1300	54.88	1.03
			CC		
		50	[P, 2]->AVANZA 1300	53.95	1.01
			CC		
			[P, 3]->AVANZA 1300	54.88	1.03
			CC		
Suzuki	5	40	[P, 2]->PICK UP	46.09	0.99
			[P, 3]->PICK UP	48.15	1.04
			[P, 4]->PICK UP	56.25	1.21
		45	[P, 2]->PICK UP	46.09	0.99
			[P, 3]->PICK UP	48.15	1.04
			[P, 4]->PICK UP	56.25	1.21
		50	[P, 4]->PICK UP	56.25	1.21
	7	40	[P, 2]->PICK UP	46.09	0.99
			[P, 3]->PICK UP	48.15	1.04
		45	[P, 2]->PICK UP	46.09	0.99
			[P, 3]->PICK UP	48.15	1.04
		50	-	-	-
	9	40	[P, 2]->PICK UP	46.09	0.99
			[P, 3]->PICK UP	48.15	1.04
		45	[P, 2]->PICK UP	46.09	0.99
			[P, 3]->PICK UP	48.15	1.04
		50	-	-	-
Honda	5	40	[W, 1]->JAZZ GE8 E MT	80.00	1.45
			[P, 2]->JAZZ GE8 E MT	66.67	1.21
			[W, 2]->JAZZ GE8 E MT	64.29	1.16
			[P, 3]->JAZZ GE8 E MT	61.54	1.11
			[W, 3]->JAZZ GE8 E AT	55.56	2.19
		45	[W, 1]->JAZZ GE8 E MT	80.00	1.45
			[P, 2]->JAZZ GE8 E MT	66.67	1.21
			[W, 2]->JAZZ GE8 E MT	64.29	1.16
			[P, 3]->JAZZ GE8 E MT	61.54	1.11
			[W, 3]->JAZZ GE8 E AT	55.56	2.19
		50	[W, 1]->JAZZ GE8 E MT	80.00	1.45
			[P, 2]->JAZZ GE8 E MT	66.67	1.21
			[W, 2]->JAZZ GE8 E	64.29	1.16

			MT			
			[P, 3]->JAZZ GE8 E MT	61.54	1.11	
			[W, 3]->JAZZ GE8 E AT	55.56	2.19	
7	40		[P, 2]->JAZZ GE8 E MT	66.67	1.21	
			[W, 2]->JAZZ GE8 E MT	64.29	1.16	
			[P, 3]->JAZZ GE8 E MT	61.54	1.11	
			[W, 3]->JAZZ GE8 E AT	55.56	2.19	
		45		[P, 2]->JAZZ GE8 E MT	66.67	1.21
				[W, 2]->JAZZ GE8 E MT	64.29	1.16
			[P, 3]->JAZZ GE8 E MT	61.54	1.11	
	50		[W, 3]->JAZZ GE8 E AT	55.56	2.19	
			[P, 2]->JAZZ GE8 E MT	66.67	1.21	
			[W, 2]->JAZZ GE8 E MT	64.29	1.16	
			[P, 3]->JAZZ GE8 E MT	61.54	1.11	
			[W, 3]->JAZZ GE8 E AT	55.56	2.19	
		[P, 2]->JAZZ GE8 E MT	66.67	1.21		
9	40		[W, 2]->JAZZ GE8 E MT	64.29	1.16	
			[P, 3]->JAZZ GE8 E MT	61.54	1.11	
		45		[P, 2]->JAZZ GE8 E MT	66.67	1.21
				[W, 2]->JAZZ GE8 E MT	64.29	1.16
				[P, 3]->JAZZ GE8 E MT	61.54	1.11
		50		[P, 2]->JAZZ GE8 E MT	66.67	1.21
			[W, 2]->JAZZ GE8 E MT	64.29	1.16	
			[P, 3]->JAZZ GE8 E MT	61.54	1.11	

Berdasarkan pada pengujian pertama maka didapatkan informasi tingkat kekuatan (*lift ratio*) rule yang dihasilkan yaitu nilai *lift ratio* dari rule tiap masing-masing *dealer*. Pada *dealer* pertama yaitu *dealer* Daihatsu didapatkan nilai *lift ratio* tertinggi adalah 1.34 sedangkan nilai *lift ratio* terendah adalah 0.93 dan rata-rata nilai *lift ratio* dari *dealer* Daihatsu adalah 1.09. Yang kedua untuk *dealer* Toyota didapatkan nilai *lift ratio* tertingginya adalah 1.12 sedangkan nilai *lift ratio* terendah adalah 0.90 dan rata-ratanya adalah 1.01 . Yang ketiga untuk *dealer* Suzuki didapatkan nilai *lift ratio* tertinggi yaitu 1.21 sedangkan nilai *lift ratio* terendah yaitu 0.99 dan rata-ratanya adalah 1.08. Kemudian yang terakhir untuk *dealer* Honda didapatkan nilai *lift ratio* tertinggi adalah 2.19 sedangkan nilai *lift*

ratio terendah adalah 1.11 dan rata-ratanya adalah 1.42. Berdasarkan nilai *lift ratio* keseluruhan diperoleh nilai rata-rata dari keempat *dealer* yaitu 1.15. Untuk nilai tertinggi dari keseluruhan nilai *lift ratio* dimiliki oleh *dealer* Honda yaitu 2.19 dan yang terendah dimiliki oleh *dealer* Toyota yaitu 0.90. Untuk keterangan rule hasil dari pengujian dapat dilihat pada Tabel 4.3

Tabel 4. 3 Keterangan Rule Hasil Uji Kekuatan Lift Ratio.

Rule	Keterangan
[P,2]->XENIA	Jika Pembeli PRIA dan UMUR 31-40 th Maka Jenis Mobil XENIA
[P,3]->XENIA	Jika Pembeli PRIA dan UMUR 41-50 th Maka Jenis Mobil XENIA
[P,4]->XENIA	Jika Pembeli PRIA dan UMUR diatas 50 th Maka Jenis Mobil XENIA
[W,2]->XENIA	Jika Pembeli WANITA dan UMUR 31-40 th Maka Jenis Mobil XENIA
[P,2]->AVANZA 1300 CC	Jika Pembeli PRIA dan UMUR 31-40 th Maka Jenis Mobil AVANZA 1300 CC
[P,3]->AVANZA 1300 CC	Jika Pembeli PRIA dan UMUR 41-50 th Maka Jenis Mobil AVANZA 1300 CC
[P,4]->AVANZA 1300 CC	Jika Pembeli PRIA dan UMUR diatas 50 th Maka Jenis Mobil AVANZA 1300 CC
[W,3]->AVANZA 1300 CC	Jika Pembeli WANITA dan UMUR 41-50 th Maka Jenis Mobil AVANZA 1300 CC
[P, 2]->PICK UP	Jika Pembeli PRIA dan UMUR 31-40 th Maka Jenis Mobil PICK UP
[P, 3]->PICK UP	Jika Pembeli PRIA dan UMUR 41-50 th Maka Jenis Mobil PICK UP
[P, 4]->PICK UP	Jika Pembeli PRIA dan UMUR diatas 50 th Maka Jenis Mobil PICK UP
[P, 2]->JAZZ GE8 E MT	Jika Pembeli PRIA dan UMUR 31-40 th Maka Jenis Mobil JAZZ GE8 E MT
[P, 3]->JAZZ GE8 E MT	Jika Pembeli PRIA dan UMUR 41-50 th Maka Jenis Mobil JAZZ GE8 E MT
[W, 1]->JAZZ GE8 E MT	Jika Pembeli WANITA dan UMUR 20-30 th Maka Jenis Mobil JAZZ GE8 E MT
[W, 2]->JAZZ GE8 E MT	Jika Pembeli WANITA dan UMUR 31-40 th Maka Jenis Mobil JAZZ GE8 E MT
[W, 3]->JAZZ GE8 E AT	Jika Pembeli WANITA dan UMUR 41-50 th Maka Jenis Mobil JAZZ GE8 E AT

Tabel 4. 4 Akurasi Rule Hasil Uji Kekuatan Lift Ratio.

<i>Dealer</i>	Rule	Support Data uji	Support Data Latih	Error	Akurasi
Daihatsu	[P,2]->XENIA	10.73	14.88	27.87%	72.13%
	[P,3]->XENIA	9.76	9.76	0%	100.00%
	[P,4]->XENIA	7.80	7.32	6.67%	93.33%
	[W,2]>XENIA	4.88	5.37	9.09%	90.91%
Toyota	[P,2]->AVANZA 1300 CC	11.63	12.06	5.57%	96.43%
	[P,3]->AVANZA 1300 CC	12.21	13.24	7.75%	92.25%
	[W,3]>AVANZA 1300 CC	5.23	5.59	6.36%	93.64%
	[P,4]->AVANZA 1300 CC	9.30	6.47	43.76%	56.24%
Suzuki	[P, 2]->PICK UP	14.53	14.85	2.16%	97.84%
	[P, 3]->PICK UP	10.06	10.92	7.95%	92.05%
	[P, 4]->PICK UP	7.82	6.30	24.10%	75.90%
Honda	[P, 2]->JAZZ GE8 E MT	11.43	14.93	23.43%	76.57%
	[P, 3]->JAZZ GE8 E MT	14.29	11.94	19.64%	80.36%
	[W, 1]->JAZZ GE8 E MT	2.86	5.97	52.14%	47.86%
	[W, 2]->JAZZ GE8 E MT	14.29	13.43	6.35%	93.65%
	[W, 3]->JAZZ GE8 E AT	8.57	7.46	14.86%	85.14%

Pengujian kedua dilakukan untuk mengetahui tingkat akurasi dari rule yang dihasilkan masing-masing *dealer*. Pada pengujian ini setiap rule diuji akurasinya dengan menggunakan data uji. Data uji diambil dari 50% dari data latih yang kemudian dilakukan perhitungan untuk mendapatkan nilai akurasi dari masing-masing rule.

Pada *dealer* pertama yaitu *dealer* Daihatsu didapatkan nilai akurasi tertinggi adalah 100% sedangkan nilai akurasi terendah adalah 72.13% dan rata-rata nilai akurasinya adalah 89.09%. *Dealer* kedua untuk *dealer* Toyota didapatkan nilai akurasi tertingginya adalah 96.43% sedangkan nilai akurasi terendah adalah 56.24% dan rata-ratanya adalah 76.33% . *Dealer* ketiga untuk *dealer* Suzuki didapatkan nilai akurasi tertinggi yaitu 97.84% sedangkan nilai akurasi terendah yaitu 75.90% dan rata-ratanya adalah 88.60%. Selanjutnya untuk *dealer* Honda didapatkan nilai akurasi tertinggi adalah 93.65% sedangkan nilai akurasi terendah adalah 47.86% dan rata-ratanya adalah 76.71%. Berdasarkan nilai akurasi keseluruhan diperoleh nilai akurasi rata-rata dari keempat *dealer* yaitu 82.68%.

Untuk nilai tertinggi dari keseluruhan nilai akurasi dimiliki oleh *dealer* Daihatsu yaitu 100% dan yang terendah dimiliki oleh *dealer* Honda yaitu 47.86%.

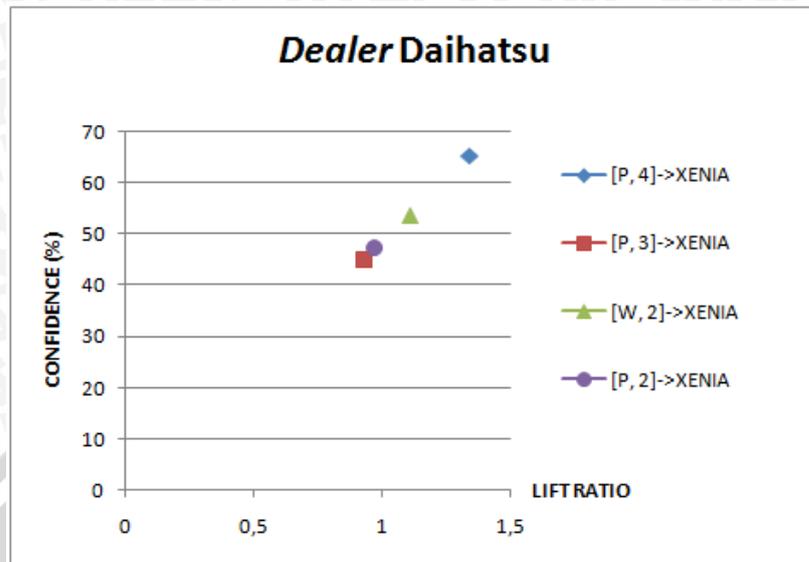
4.5. Analisa Hasil

Berdasarkan hasil pengujian nilai kekuatan (*lift ratio*) *rule*, jumlah *rule* yang dihasilkan tiap *dealer* adalah sebagai *dealer* Suzuki 3 *rule*, *dealer* Honda 5 *rule*, *dealer* Toyota 4 *rule*, dan *dealer* Daihatsu 4 *rule*. Rata – rata nilai *lift ratio* yang didapatkan dari keseluruhan *rule* yang dihasilkan adalah 1.15. Nilai *lift ratio* tertinggi adalah 2.19 yang didapatkan dari *rule* hasil pengujian *dealer* Honda yaitu [W, 3]->JAZZ GE8 E AT atau **jika pembeli wanita dan umur 41 sampai 50 tahun maka jenis mobil yang dibeli JAZZ GE8 E AT** dengan tingkat kepercayaan sebesar 55.56%. Nilai *lift ratio* terendah adalah 0.90 yang didapatkan dari *rule* hasil pengujian *dealer* Toyota yaitu [P, 4]->AVANZA 1300 CC atau **jika pembeli pria dan umur diatas 50 tahun maka jenis mobil yang dibeli AVANZA 1300cc** dengan tingkat kepercayaan sebesar 47.83%.

Pada hasil pengujian nilai kekuatan (*lift ratio*) *rule* ini, terdapat hasil pengujian yang tidak menghasilkan *rule* yaitu pada pengujian *dealer* Daihatsu dengan minimum *support* 9% dan minimum *confidence* 50%, sedangkan pada minimum *support* 5% untuk minimum *confidence* 50% dihasilkan *rule* [W, 2]->XENIA atau **jika pembeli wanita dan umur 31 sampai 40 tahun maka jenis mobil yang dibeli XENIA**, dan pada minimum *support* 7% untuk minimum *confidence* 50% *rule* [P, 4]->XENIA atau **jika pembeli pria dan umur diatas 50 tahun maka jenis mobil yang dibeli XENIA**. Hal ini menunjukkan bahwa kandidat *frequent itemset* dengan tingkat kepercayaan minimal 50% pada *dealer* Daihatsu hanya mampu mencapai frekuensi kemunculan 22 data dari 409 data transaksi penjualan, yaitu [W, 2]->XENIA atau **jika pembeli wanita dan umur 31 sampai 40 tahun maka jenis mobil yang dibeli XENIA** dan hanya mampu mencapai frekuensi kemunculan 30 data dari 409 data transaksi penjualan [P, 4]->XENIA atau **jika pembeli pria dan umur diatas 50 tahun maka jenis mobil yang dibeli XENIA** sehingga dapat diambil kesimpulan bahwa tidak dihasilkannya *rule* adalah tidak ada kandidat *frequent itemset* yang frekuensi kemunculannya mencapai minimal 36 data dari 409 data transaksi penjualan dan tingkat kepercayaan minimal 50%

Selain itu, pengujian yang tidak menghasilkan *rule* juga terjadi pada pengujian *dealer* Suzuki dengan minimum *support* 7% dan 9% serta minimum *confidence* 50%, sedangkan pada minimum *support* 5% untuk minimum *confidence* 50% masih dihasilkan *rule* yaitu **[P, 4]->PICK UP** atau **jika pembeli pria dan umur diatas 50 tahun maka jenis mobil yang dibeli PICK UP**. Hal ini menunjukkan bahwa kandidat *frequent itemset* dengan tingkat kepercayaan minimal 50% pada *dealer* Suzuki hanya mampu mencapai frekuensi kemunculan 35 data dari 713 data transaksi penjualan, yaitu **[P, 4]->PICK UP** atau **jika pembeli pria dan umur diatas 50 tahun maka jenis mobil yang dibeli PICK UP** sehingga dapat diambil kesimpulan bahwa tidak dihasilkannya *rule* adalah tidak ada kandidat *frequent itemset* yang frekuensi kemunculannya mencapai minimal 49 data atau 64 data dari 713 data transaksi penjualan dan tingkat kepercayaan minimal 50%.

Pada Gambar 4.10 dapat dilihat grafik *confidence* dan *lift ratio* untuk *rule* yang dihasilkan dari pengujian *dealer* Daihatsu. Berdasarkan grafik tersebut, dari 4 macam *rule* yang dihasilkan *rule* yang memiliki nilai *lift ratio* lebih dari 1 adalah **[P, 4]->XENIA** dan **[W, 2]->XENIA**. Untuk *rule* **[P, 4]->XENIA**, hal ini menjelaskan bahwa dari keseluruhan jumlah pembeli pria yang berumur diatas 50 tahun sebesar 65.22% yang membeli jenis mobil Xenia dan tingkat kepercayaan pembeli pria yang berumur diatas 50 tahun masih lebih tinggi dari keseluruhan jumlah jenis mobil Xenia yang terjual. Untuk *rule* **[W, 2]->XENIA**, hal ini menjelaskan bahwa dari keseluruhan jumlah pembeli wanita yang berumur 31 sampai 40 tahun sebesar 53.66% yang membeli jenis mobil Xenia yang terjual dan tingkat kepercayaan pembeli pria yang berumur diatas 50 tahun masih lebih tinggi dari keseluruhan jumlah jenis mobil Xenia yang terjual.



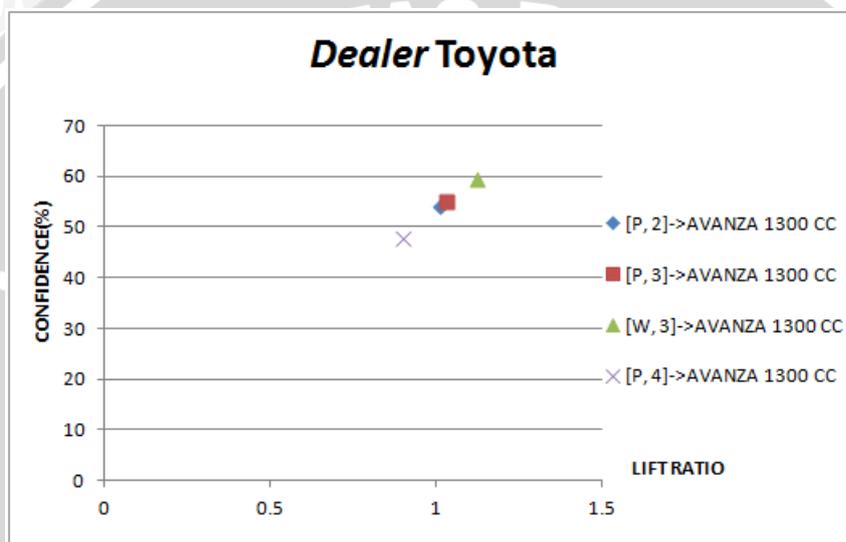
Gambar 4. 10 Grafik Confidence dan Lift Ratio Rule Dealer Daihatsu

Selain itu, berdasarkan grafik Gambar 4.10 dapat disimpulkan bahwa semakin tinggi nilai *confidence* suatu *rule* maka semakin tinggi pula nilai *lift ratio rule* tersebut, yang berarti bahwa semakin tinggi tingkat kepercayaan suatu *rule* maka nilai kekuatan *rule* tersebut juga semakin tinggi.

Pada Gambar 4.11 dapat dilihat grafik *confidence* dan *lift ratio* untuk *rule* yang dihasilkan dari pengujian *dealer* Toyota. Berdasarkan grafik tersebut, dari 3 macam *rule* yang dihasilkan *rule* yang memiliki nilai *lift ratio* lebih dari 1 [P, 2]->AVANZA 1300cc, [P, 3]->AVANZA 1300cc, dan [W, 3]->AVANZA 1300cc. Untuk *rule* [P, 2]->AVANZA 1300cc, hal ini menjelaskan bahwa dari keseluruhan jumlah pembeli pria yang berumur 31 sampai 40 tahun sebesar 53.95% yang membeli jenis mobil Avanza 1300cc dan tingkat kepercayaan pembeli pria yang berumur 31 sampai 40 tahun masih lebih tinggi dari keseluruhan jumlah jenis mobil Avanza 1300cc yang terjual . Untuk *rule* [P, 3]->Avanza 1300cc, hal ini menjelaskan bahwa dari keseluruhan jumlah pembeli pria yang berumur 41 sampai 50 tahun sebesar 54.88% yang membeli jenis mobil Avanza 1300cc masih lebih tinggi dari keseluruhan jumlah jenis mobil Avanza 1300cc yang terjual. Untuk *rule* [W, 3]->Avanza 1300cc, hal ini menjelaskan bahwa dari keseluruhan jumlah pembeli wanita yang berumur 41 sampai 50 tahun sebesar 59.38% yang membeli Avanza 1300cc dan tingkat kepercayaan wanita

yang berumur 41 sampai 50 tahun yang membeli jenis mobil Avanza 1300cc masih lebih tinggi dari keseluruhan jumlah jenis mobil Avanza 1300cc yang terjual.

Selain itu, berdasarkan grafik Gambar 4.11 dapat disimpulkan bahwa semakin tinggi nilai *confidence* suatu *rule* maka semakin tinggi pula nilai *lift ratio* *rule* tersebut, yang berarti bahwa semakin tinggi tingkat kepercayaan suatu *rule* maka nilai kekuatan *rule* tersebut juga semakin tinggi.

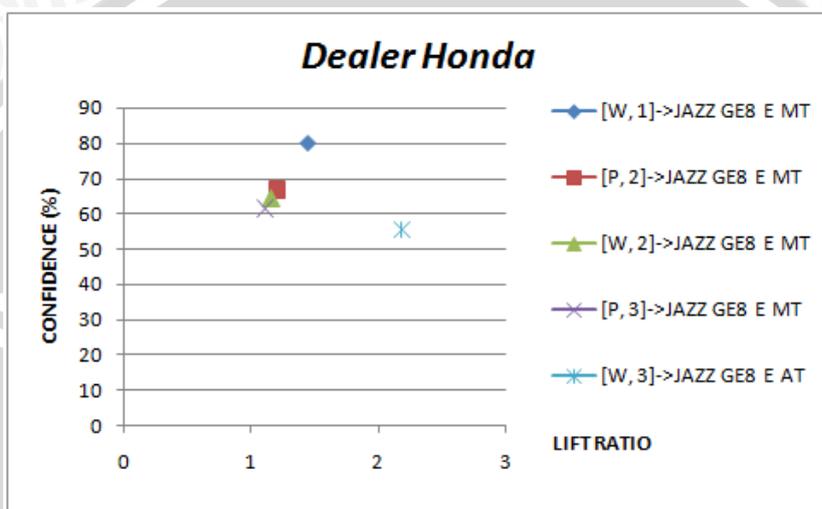


Gambar 4. 11 Grafik Confidence dan Lift Ratio Rule Dealer Toyota

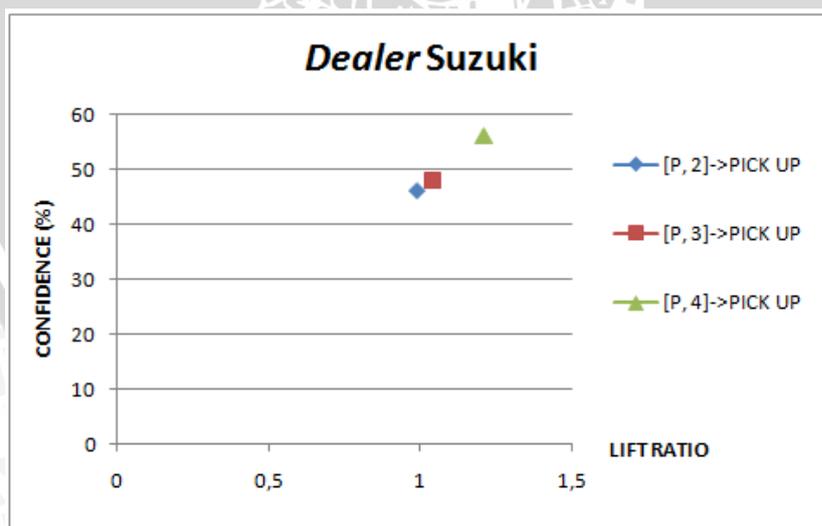
Gambar 4.12 merupakan grafik *confidence* dan *lift ratio* untuk *rule* yang dihasilkan dari pengujian *dealer* Honda. Berdasarkan grafik tersebut, dari 5 macam *rule* yang dihasilkan *rule* yang memiliki nilai *lift ratio* lebih dari 1 adalah semua *rule*. Hal ini berarti tingkat kepercayaan pembeli wanita berumur 20 sampai 30 tahun dan 31 sampai 40 tahun yang membeli Jazz GE8 MT serta tingkat kepercayaan pembeli pria berumur 31 sampai 40 tahun dan 41 sampai 50 tahun yang membeli Jazz GE8 MT masih lebih tinggi daripada keseluruhan jumlah Jazz GE8 MT yang terjual.

Selain itu, berdasarkan grafik Gambar 4.12 terdapat *rule* yang nilai *confidence*-nya tidak terlalu tinggi dibandingkan nilai *confidence* *rule* lain namun nilai *lift ratio* yang dimiliki justru sangat tinggi dibandingkan nilai *lift ratio* *rule*

yang lain. Hal ini dijelaskan pada *rule* [W, 3]->JAZZ GE8 E AT dan menunjukkan bahwa frekuensi penjualan jenis mobil Jazz GE8 E AT yang tidak terlalu tinggi sehingga dengan tingkat kepercayaan yang tidak terlalu tinggi dapat menghasilkan kekuatan *rule* yang tinggi, namun tingkat kepercayaan *rule* tersebut masih tetap lebih tinggi daripada keseluruhan jumlah jenis mobil Jazz GE8 E AT yang terjual.



Gambar 4. 12 Grafik Confidence dan Lift Ratio Rule Dealer Honda



Gambar 4. 13 Grafik Confidence dan Lift Ratio Rule Dealer Suzuki

Pada Gambar 4.13 dapat dilihat grafik *confidence* dan *lift ratio* untuk *rule* yang dihasilkan dari pengujian *dealer* Suzuki. Berdasarkan grafik tersebut, dari 3 macam *rule* yang dihasilkan *rule* yang memiliki nilai *lift ratio* lebih dari 1 adalah [P, 3]->PICK UP dan [P, 4]->PICK UP. Untuk *rule* [P, 3]->PICK UP, hal ini menjelaskan bahwa dari keseluruhan jumlah pembeli pria yang berumur 41 sampai 50 tahun sebesar 48.15% yang membeli jenis mobil Pick Up dan tingkat kepercayaan pembeli pria yang berumur berumur 41 sampai 50 tahun masih lebih tinggi dari keseluruhan jumlah jenis mobil Pick Up yang terjual. Untuk *rule* [P, 4]->PICK UP, hal ini menjelaskan bahwa dari keseluruhan jumlah pembeli wanita yang berumur diatas 50 tahun sebesar 56.25% yang membeli jenis mobil Pick Up yang terjual dan tingkat kepercayaan pembeli pria yang berumur diatas 50 tahun masih lebih tinggi dari keseluruhan jumlah jenis mobil Pick Up yang terjual.

Selain itu, berdasarkan grafik Gambar 4.13 dapat disimpulkan bahwa semakin tinggi nilai *confidence* suatu *rule* maka semakin tinggi pula nilai *lift ratio rule* tersebut, yang berarti bahwa semakin tinggi tingkat kepercayaan suatu *rule* maka nilai kekuatan *rule* tersebut juga semakin tinggi.

Berdasarkan nilai *confidence* dan nilai *lift ratio rule* yang dihasilkan dalam pengujian ini, semakin tinggi nilai *confidence* suatu *rule* maka nilai *lift ratio rule* tersebut semakin tinggi pula dan semakin tinggi nilai frekuensi kemunculan item *consequent* suatu *rule* maka semakin kecil nilai *lift ratio rule* tersebut. *Rule* yang memiliki nilai *confidence* lebih tinggi daripada *rule* lain belum tentu memiliki nilai *lift ratio* yang lebih tinggi atau belum tentu memiliki nilai *lift ratio* lebih dari 1. Hal ini menjelaskan bahwa meskipun nilai kepercayaan (*confidence*) dari *rule* tersebut tinggi namun belum tentu *rule* tersebut bermanfaat karena nilai *confidence* menggambarkan tingkat kepercayaan item dalam *rule* tersebut muncul secara bersamaan terhadap jumlah keseluruhan item *antecedent* yang muncul sedangkan nilai *lift ratio* menggambarkan keseimbangan antara frekuensi kemunculan tiap item baik yang menjadi *antecedent* maupun *consequent* dalam *rule* tersebut secara independen dengan frekuensi kemunculannya secara bersamaan sehingga nilai *lift ratio* lebih dapat menunjukkan manfaat dari *rule* tersebut. Akan tetapi nilai minimum *support* yang digunakan juga mempengaruhi

untuk mendapatkan *rule* yang benar – benar bermanfaat. *Rule* yang dapat dikatakan bermanfaat dari hasil pengujian ini dapat dilihat pada Tabel 4.5 dan detail informasi *rule* dapat dilihat pada Tabel 4.6

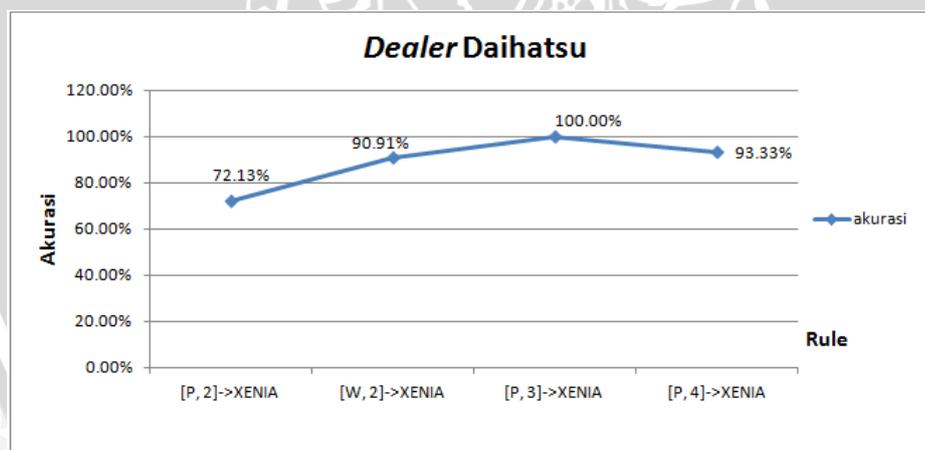
Tabel 4. 5 Rule yang Bermanfaat

Dealer	Rule	Keterangan
Daihatsu	[W, 2]->XENIA	Jika Pembeli WANITA dan UMUR 31-40 th Maka Jenis Mobil XENIA
	[P, 4]->XENIA	Jika Pembeli PRIA dan UMUR DIATAS 50 th Maka Jenis Mobil XENIA
Toyota	[P, 2]->AVANZA 1300 CC	Jika Pembeli PRIA dan UMUR 31-40 th Maka Jenis Mobil AVANZA 1300 CC
	[P, 3]->AVANZA 1300 CC	Jika Pembeli PRIA dan UMUR 41-50 th Maka Jenis Mobil AVANZA 1300 CC
	[W, 3]->AVANZA 1300 CC	Jika Pembeli WANITA dan UMUR 41-50 th Maka Jenis Mobil AVANZA 1300 CC
Suzuki	[P, 4]->PICK UP	Jika Pembeli PRIA dan UMUR diatas 50th Maka Jenis Mobil PICK UP
Honda	[W, 1]->JAZZ GE8 E MT	Jika Pembeli WANITA dan UMUR 20-30 th Maka Jenis Mobil JAZZ GE8 E MT
	[P, 2]->JAZZ GE8 E MT	Jika Pembeli PRIA dan UMUR 31-40 th Maka Jenis Mobil JAZZ GE8 E MT
	[W, 2]->JAZZ GE8 E MT	Jika Pembeli WANITA dan UMUR 31-40 th Maka Jenis Mobil JAZZ GE8 E MT
	[P, 3]->JAZZ GE8 E MT	Jika Pembeli PRIA dan UMUR 41-50 th Maka Jenis Mobil JAZZ GE8 E MT
	[W, 3]->JAZZ GE8 E AT	Jika Pembeli WANITA dan UMUR 41-50 th Maka Jenis Mobil JAZZ GE8 E AT

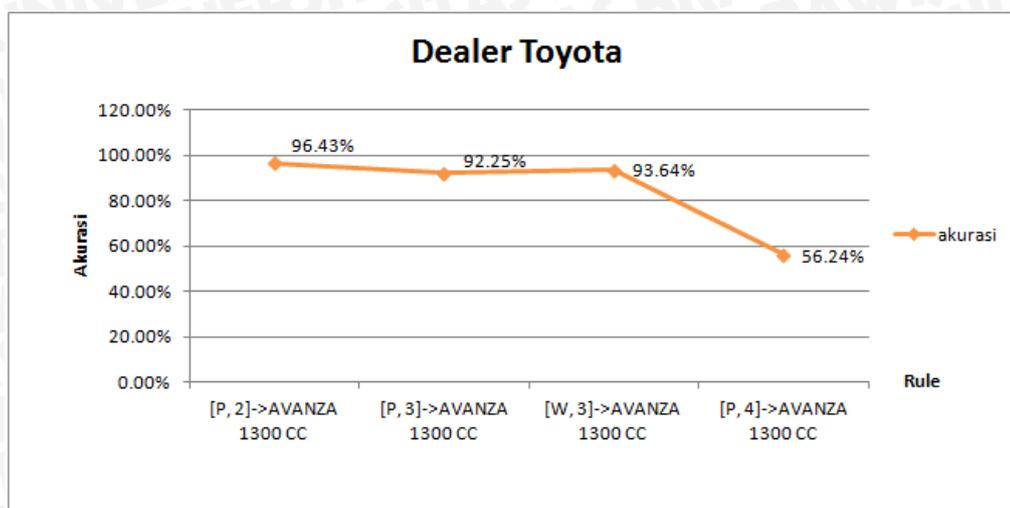
Tabel 4. 6 Detail Informasi Rule yang Bermanfaat

Dealer	Rule	Support	Confidence	Lift Ratio
Daihatsu	[W, 2]->XENIA	5.37	53.66	1.11
	[P, 4]->XENIA	7.32	65.22	1.34
Toyota	[P, 2]->AVANZA 1300 CC	12.06	53.95	1.01
	[P, 3]->AVANZA 1300 CC	13.24	54.88	1.03
	[W, 3]->AVANZA 1300 CC	5.59	59.38	1.12
Suzuki	[P, 4]->PICK UP	6.3	56.25	1.21
Honda	[W, 1]->JAZZ GE8 E MT	5.97	80.00	1.45
	[P, 2]->JAZZ GE8 E MT	14.93	66.67	1.21
	[W, 2]->JAZZ GE8 E MT	13.43	64.29	1.16
	[P, 3]->JAZZ GE8 E MT	11.94	61.54	1.11
	[W, 3]->JAZZ GE8 E AT	7.46	55.56	2.19

Hasil pengujian akurasi *Association Rule* yang dihasilkan dari *dealer* Daihatsu, *dealer* Toyota, *dealer* Suzuki dan *dealer* Honda dapat dilihat melalui grafik akurasi rule pada Gambar 4.14, Gambar 4.15, Gambar 4.16 dan Gambar 4.17.



Gambar 4. 14 Grafik Akurasi Association Rule yang terbentuk dari dealer Daihatsu



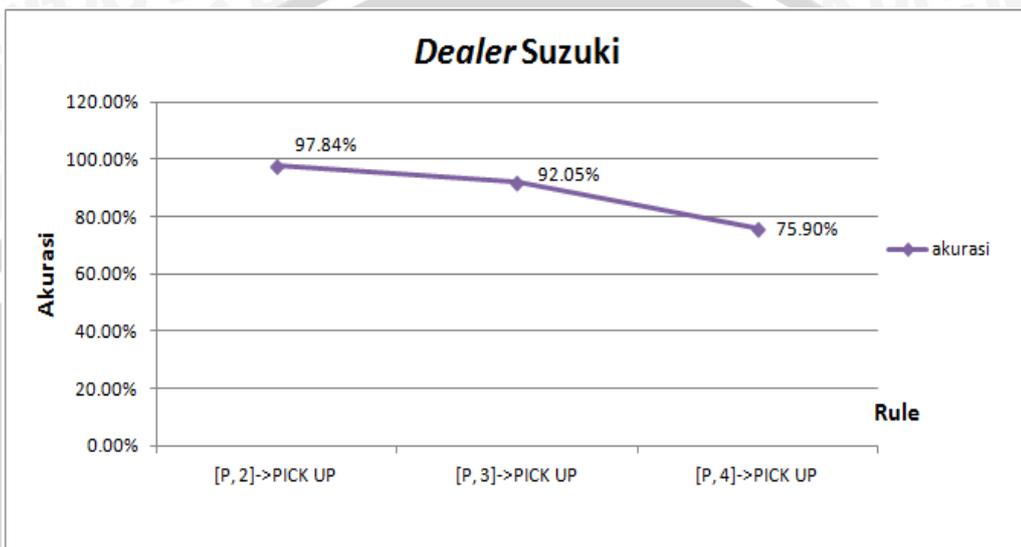
Gambar 4. 15 Grafik Akurasi Association Rule yang Terbentuk dari Dealer Toyota

Berdasarkan pada grafik Gambar 4.14 akurasi rule yang terbentuk dari dealer Daihatsu didapatkan bahwa rule [P,3]->XENIA memiliki akurasi paling tinggi, yaitu 100%. Hal ini menunjukkan bahwa pada data uji dealer Daihatsu sebanyak 204 data transaksi penjualan terdapat 20 data transaksi dengan jenis kelamin pria dan umur 41 sampai 50 tahun yang membeli jenis mobil Xenia dibandingkan dengan data latih sebanyak 409 data transaksi penjualan terdapat 40 data transaksi dengan jenis kelamin pria dan umur 41 sampai 50 tahun yang membeli jenis mobil Xenia.

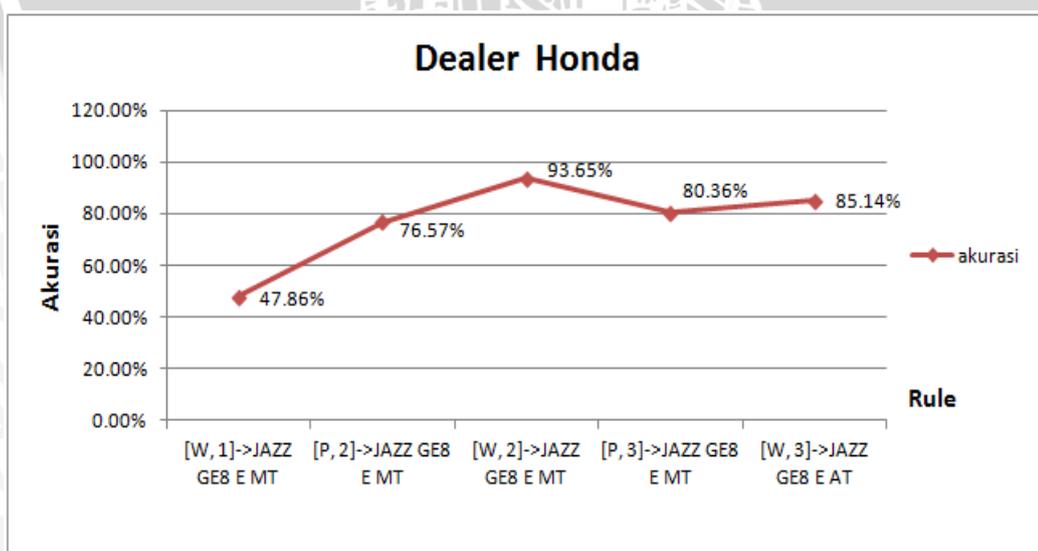
Berdasarkan pada grafik Gambar 4.15 akurasi rule yang terbentuk dari dealer Toyota didapatkan bahwa rule [P,2]-> AVANZA 1300 CC memiliki akurasi paling tinggi, yaitu 96.43%. Hal ini menunjukkan bahwa pada data uji dealer Toyota sebanyak 170 data transaksi penjualan terdapat 20 data transaksi dengan jenis kelamin pria dan umur 31 sampai 40 tahun yang membeli jenis mobil AVANZA 1300 CC dibandingkan dengan data latih sebanyak 340 data transaksi penjualan terdapat 41 data transaksi dengan jenis kelamin pria dan umur 31 sampai 40 tahun yang membeli jenis mobil AVANZA 1300 CC.

Berdasarkan pada grafik Gambar 4.16 akurasi rule yang terbentuk dari dealer Suzuki didapatkan bahwa rule [P,2]->PICK UP memiliki akurasi paling tinggi, yaitu 97.84%. Hal ini menunjukkan bahwa pada data uji dealer Suzuki

sebanyak 357 data transaksi penjualan terdapat 52 data transaksi dengan jenis kelamin pria dan umur 31 sampai 40 tahun yang membeli jenis mobil PICK UP dibandingkan dengan data latih sebanyak 714 data transaksi penjualan terdapat 106 data transaksi dengan jenis kelamin pria dan umur 31 sampai 40 tahun yang membeli jenis mobil PICK UP.



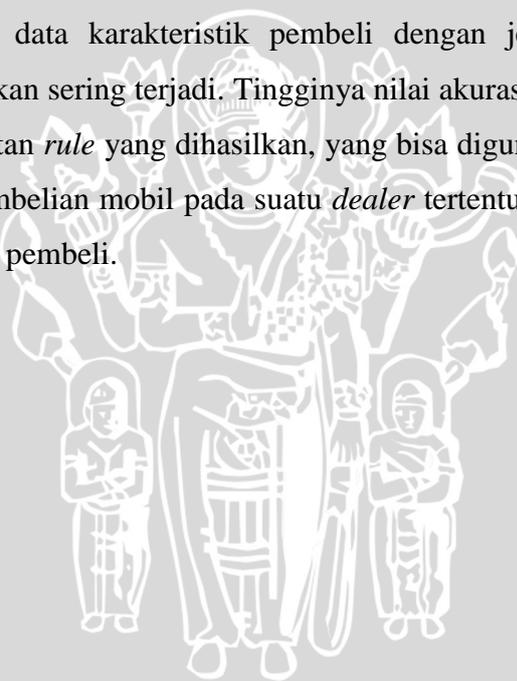
Gambar 4. 16 Grafik Akurasi Association Rule yang Terbentuk dari Dealer Suzuki



Gambar 4. 17 Grafik Akurasi Association Rule yang Terbentuk dari Dealer Honda

Berdasarkan pada grafik Gambar 4.17 akurasi rule yang terbentuk dari *dealer* Honda didapatkan bahwa rule [W, 2]->JAZZ GE8 E MT memiliki akurasi paling tinggi, yaitu 93.65%. Hal ini menunjukkan bahwa pada data uji *dealer* Honda sebanyak 66 data transaksi penjualan terdapat 4 data transaksi dengan jenis kelamin wanita dan umur 31 sampai 40 tahun yang membeli jenis mobil [W, 2]->JAZZ GE8 E MT dibandingkan dengan data latih sebanyak 66 data transaksi penjualan terdapat 9 data transaksi dengan jenis kelamin pria dan umur 31 sampai 40 tahun yang membeli jenis mobil [W, 2]->JAZZ GE8 E MT.

Tingginya nilai akurasi *rule* menunjukkan bahwa tingginya penerapan *association rule* yang terbentuk pada data latih pada data uji. Semakin tinggi nilai akurasi rule maka semakin tinggi pula nilai kepercayaan bahwa rule tersebut menjadi pola asosiasi data karakteristik pembeli dengan jenis kelamin dan kelompok umur yang akan sering terjadi. Tingginya nilai akurasi rule memberikan informasi terkait kekuatan *rule* yang dihasilkan, yang bisa digunakan untuk bahan dasar prediksi pada pembelian mobil pada suatu *dealer* tertentu yang berdasarkan jenis kelamin dan umur pembeli.



BAB V PENUTUP

5.1. Kesimpulan

Berdasarkan hasil analisis uji coba dapat diambil kesimpulan:

1. Dengan menerapkan algoritma *FP-Growth* pada data transaksi penjualan *dealer* mobil dengan karakteristik jenis kelamin dan jenis umur untuk jenis mobil tertentu di wilayah Mataram, didapatkan rule sebagai berikut :
 - a) Jenis Mobil Xenia *dealer* Daihatsu diminati oleh pembeli berjenis kelamin pria berumur diatas 50 tahun dan pembeli berjenis kelamin wanita berumur 31 sampai 40 tahun.
 - b) Jenis Mobil Avanza *dealer* Toyota diminati oleh pembeli berjenis kelamin pria berumur 31 sampai 40 tahun dan pembeli berjenis kelamin wanita berumur 31 sampai 40 tahun.
 - c) Jenis Mobil Pick Up *dealer* Suzuki diminati oleh pembeli berjenis kelamin pria berumur diatas 50 tahun.
 - d) Jenis Mobil Jazz GE8 E MT *dealer* Honda diminati oleh pembeli berjenis kelamin pria berumur 31 sampai 50 tahun dan pembeli berjenis kelamin wanita berumur 20 sampai 40 tahun.
 - e) Jenis Mobil Jazz GE8 E AT *dealer* Honda diminati oleh pembeli berjenis kelamin wanita berumur 41 sampai 50 tahun.
2. *Rule* yang dihasilkan dari penerapan algoritma *FP-Growth* dalam permasalahan data transaksi penjualan *dealer* mobil di wilayah Mataram ini memiliki rata – rata tingkat kekuatan (*lift ratio*) sebesar 1.31 dengan tingkat kekuatan (*lift ratio*) *association rule* terbesar yang dihasilkan adalah 2.19 dan terkecil adalah 1.04 untuk *rule* dengan tingkat kepercayaan *rule* minimal 50%. Berdasarkan tingkat kekuatan (*lift ratio*) yang dihasilkan, dapat dikatakan bahwa *rule* yang bermanfaat tidak hanya dilihat berdasarkan nilai *confidence*, akan tetapi

berdasarkan tiga parameter penting yaitu nilai *confidence*, nilai *lift ratio*, dan nilai minimum *support* yang digunakan.

3. Akurasi *rule* yang dihasilkan dari penerapan *rule* hasil algoritma *FP-Growth* terhadap data uji, memiliki nilai rata – rata sebesar 83.28% dengan akurasi tertinggi sebesar 100% dan terkecil sebesar 47.86%
Nilai akurasi menunjukkan prosentase penerapan suatu *rule* pada data uji dari keseluruhan jumlah data transaksi pada data uji.

5.2. Saran

Saran yang dapat dijadikan bahan pertimbangan untuk penelitian selanjutnya adalah dengan penambahan parameter seperti warna mobil, pekerjaan pembeli dan wilayah tempat tinggal pembeli. Selain itu, dapat dilakukan pembuatan sistem yang lebih aplikatif dengan memberikan sistem prediksi.



DAFTAR PUSTAKA

- [ARM-04] Armstrong, Garry & Philip Kotler. 2004. *Dasar-Dasar Pemasaran: Jilid 1 Edisi Kesembilan*. Jakarta: PT.Indeks.
- [ERW-09] Erwin. 2009. *Analisis Market Basket dengan Algoritma Apriori dan FP – Growth*. Jurnal Generic Volume 4 No. 2 Fakultas Ilmu Komputer. Universitas Sriwijaya. Sumatera Selatan.
- [FEB-09] Febriyana, Fatma Rika. 2009. *Perbandingan Kecepatan dalam Pencarian Frequent Itemset antara Algoritma FP – Growth dan Cut Both Ways*. Universitas Brawijaya. Malang
- [GAI-05] [GAIKINDO] Gabungan Industri Kendaraan Bermotor Indonesia, Laporan Data periode 2005
- [GIL-09] Gilbert Mc Devinn. 2009. *Showroom Definition*, Edisi 154. Penerbit Kompas Gramedia Building Unit 2 Lt 1, Jakarta,
- [GRE-05] Gregorius S. Budhi, Leo W. Santoso, Karel R. Harisaputra. 2005. *Penggunaan Metode Data Mining CLARANS dan Naive Bayes Classification untuk Sistem Analisa Cluster dan Klasifikasi Data Terintegrasi*. Prosiding Sem Nas The Application of Technology Toward a Better Life 2005 buku 10.
- [HAN-01] Han , J dan Kamber , M . 2001. *Data Mining : Concepts and Techniques*. Morgan Kaufman. San Fransisco.
- [HAN-00] Han, J., Pie, J., Yin, Y. 2000. *Mining Frequent Patterns without Candidate Generation*, School of Computing Science Simon Fraser University.

- [HUD-10] Huda, Nuqson Masykur. 2010. *Aplikasi Data Mining untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa*. Semarang : Universitas Diponegoro.
- [KAN-03] Kantardzic, Mehmed. 2003 . *Data mining : Concepts, Models, Methods, and Algorithms*. John Wiley & Sons. New Jersey.
- [KOT-07] Kotler, Philip dan Kevin Lane Keller. 2007. *Manajemen Pemasaran*. PT. Indeks. Jakarta.
- [KUS-07] Kusriani, M.Kom. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*. CV Andi Offset. Yogyakarta.
- [LAR-05] Larose, Danniell T. Discovering.2005. *Knowledge in data : An Introduction to Data Mining*. John Wiley & Sons Inc. New Jersey.
- [MAR-06] Maryeti , Sri. 2006. *Analisis Perbandingan Algoritma FP-Growth dan algoritma Tree Projection dalam Pembangkitan Frequent Pattern*. STT Telkom Bandung.
- [PRA-03] Pramudiono, Iko. 2003. *Pengantar Data Mining : Menambang Permata Pengetahuan di Gudang Data*. <http://www.ilmukomputer.com>, tanggal akses 15 Maret 2011.
- [REN-04] Rennolls, K. 2004. *Introduction to Data Mining – Chapter 2*.
- [ROC-10] Rochmah, Affriantari. 2010. *Perancangan Fitur Rekomendasi Film Website Solo Movie dengan Menggunakan Metode Algoritma Apriori*. Universitas Sebelas Maret. Surakarta.

- [SAM-08] Samuel, David. 2008. *Penerapan Struktur FP – Tree dan Algoritma FP – Growth dalam Optimasi Penentuan Frequent Itemset*. Institut Teknologi Bandung.
- [SAN-07] Santosa, Budi. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Graha Ilmu. Yogyakarta.
- [TUR-05] Turban, E . 2005. *Decision Support System and Itelegence System*.Yogyakarta. Andi Ofset.
- [VER-08] Verhein, Florian. 2008. *Frequent Pattern Growth (FP – Growth) Algorithm*. School of Information Technologies The University of Sydney. Australia.
- [YIA-06] Yiapanis, Paraskevas. 2006. *Parallel Mining of Minimal Sample Unique Itemsets*. School of Computer Science, University of Manchester.
- [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.