

**PENERAPAN METODE ASSOCIATION RULE  
DENGAN ALGORITMA FP-GROWTH  
PADA DATA TRANSAKSI PENJUALAN BUKU**

**SKRIPSI**

Oleh :  
**MARISSA HAMNON**  
**0710960016 - 96**



**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2012**

UNIVERSITAS BRAWIJAYA



**PENERAPAN METODE ASSOCIATION RULE  
DENGAN ALGORITMA FP-GROWTH  
PADA DATA TRANSAKSI PENJUALAN BUKU**

**SKRIPSI**

Sebagai salah satu syarat untuk memperoleh gelar  
Sarjana dalam bidang Ilmu Komputer

Oleh :  
**MARISSA HAMNON**  
**0710960016 - 96**



HALAMAN JUDUL

**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2012**

UNIVERSITAS BRAWIJAYA



**LEMBAR PENGESAHAN SKRIPSI**  
**PENERAPAN METODE ASSOCIATION RULE**  
**DENGAN ALGORITMA FP-GROWTH**  
**PADA DATA TRANSAKSI PENJUALAN BUKU**

Oleh :  
**MARISSA HAMNON**  
**0710960016 – 96**

**Setelah dipertahankan di depan Majelis Penguji**  
**Pada tanggal 12 Januari 2012**  
**dan dinyatakan memenuhi syarat untuk memperoleh gelar**  
**Sarjana dalam bidang Ilmu Komputer**

**Pembimbing I**

**Pembimbing II**

**Laili Muflikhah, S.Kom., M.Sc**  
**NIP. 19741113200501 2 001**

**Drs. M. Arif Rahman, M.Kom**  
**NIP. 19660423199111 1 001**

**Mengetahui,**  
**Ketua Jurusan Matematika Fakultas MIPA**  
**Universitas Brawijaya Malang**

**Dr. Abdul Rouf Alghofari, M.Sc**  
**NIP. 19670907199203 1 001**

UNIVERSITAS BRAWIJAYA



## LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Marissa Hamnon  
NIM : 0710960016 - 96  
Jurusan : Matematika  
Penulis Skripsi Berjudul : Penerapan Metode *Association Rule* dengan Algoritma *FP-Growth* pada Data Transaksi Penjualan Buku

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar – benar karya sendiri dan tidak menjiplak karya orang lain, selain nama – nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila di kemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 12 Januari 2012

Yang menyatakan,

Marissa Hamnon  
NIM. 0710960016

UNIVERSITAS BRAWIJAYA



# PENERAPAN METODE ASSOCIATION RULE DENGAN ALGORITMA *FP-GROWTH* PADA DATA TRANSAKSI PENJUALAN BUKU

## ABSTRAK

Dalam mengatasi persaingan bisnis, dibutuhkan suatu strategi bisnis agar dapat meningkatkan penjualan. Strategi bisnis dapat dilakukan dengan menganalisa data transaksi penjualan sehingga diperoleh informasi tersembunyi atau tidak diketahui sebelumnya. 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 tersembunyi dari data dalam jumlah besar. Dalam penelitian ini, penggalian dilakukan terhadap data transaksi penjualan buku toko buku Togamas. Penggalian data dilakukan untuk mendapatkan pola asosiasi buku berdasarkan golongan buku dengan rentang bulan data transaksi yang bervariasi, yaitu per bulan, per dua bulan, triwulan, dan kwartal. 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 nilai rata – rata *lift ratio* dari *rule* yang dihasilkan dengan minimum *confidence* 50% sebesar 0,95. Nilai *lift ratio* tertinggi sebesar 1,02 pada periode bulan Agustus dan nilai *lift ratio* terendah sebesar 0,85 pada periode bulan Mei rentang bulan per bulan.

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

UNIVERSITAS BRAWIJAYA



# APPLICATION OF ASSOCIATION RULE METHOD WITH FP-GROWTH ALGORITHM ON BOOK SALES TRANSACTION DATA

## ABSTRACT

In order to overcome business competition, it took a business strategy to increase sales. Business strategies can be done by analyzing the book sales transaction data so that previously hidden or unknown informations can be obtained. Book sales transaction data is a large-scaled data in which a special technique called data mining is needed. Data mining is effective enough to find a set of hidden information patterns from a large amounts of data. In this research, minings were carried out on the book sales transaction data of Togamas bookstore. The mining of data were performed to obtain the association pattern of books based on a book group with various month range of transaction data i.e within a month, two-months, three-months, and quarter months. The system will perform a search of association rules with the early step is to find frequent itemsets by applying the fp-growth algorithm. From then, the association patterns search will be performed by using association rule technique. At the final stage, the system will calculate the strength of association patterns which are formed by using lift ratio. This research have resulted 0,95 in average value of lift ratio from the rules generated, with the minimum confidence 50%. The value of lift ratio reached the highest value on August at 1,02 and the lowest values on May at 0,85 with monthly range.

Keywords : *data mining, association rule, fp-growth, lift ratio.*

UNIVERSITAS BRAWIJAYA



## KATA PENGANTAR

*Bismillahirrahmanirrohim*

*Alhamdulillah*, puji dan syukur kehadirat Allah *Subhanahu Wa Ta'ala*, yang selalu melimpahkan rahmat, taufik, hidayah, ilmu, petunjuk dikala bimbang, cahaya dikala hati gelap, kesabaran dan keikhlasan dikala hati mengeluh, ketenangan dikala rasa khawatir, kekuatan dikala rapuh kepada penulis sehingga dapat menyelesaikan skripsi yang berjudul **Penerapan Metode Association Rule Dengan Algoritma FP-Growth Pada Data Transaksi Penjualan Buku**. Tak lupa, shalawat dan salam kepada cahaya segala cahaya, obat segala obat, rahasia segala rahasia, dan kunci segala kemudahan, yaitu junjungan kita, Nabi Muhammad *Shalallahu Alaihi Wasallam*.

Penyusunan skripsi ini juga tak lepas dari bantuan banyak pihak, baik itu berupa bimbingan, kritik, saran, dukungan, motivasi maupun doa. Oleh karena itu, ucapan terima kasih ingin penulis sampaikan kepada :

1. Lailil Muflikhah, S.Kom, M.Sc selaku dosen pembimbing I, atas kebersediaan meluangkan waktu untuk memberikan bimbingan, pengarahan, saran, dan dukungan selama penyusunan skripsi ini.
2. Drs. Muhammad Arif Rahman, M.kom selaku dosen pembimbing II, atas kebersediaan meluangkan waktu untuk memberikan bimbingan, pengarahan, saran, dan dukungan selama penyusunan skripsi ini.
3. Dr. Abdul Rouf Alghofari, M.Sc selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang.
4. Mardji, MT selaku Ketua Program Studi Ilmu Komputer Jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang.
5. Dany Primanita Kartikasari, ST selaku dosen pembimbing akademik, atas nasehat, bimbingan, saran, dukungan dan kenangan yang tidak terlupakan yang diberikan selama penulis menuntut ilmu di bangku perkuliahan.
6. Seluruh Bapak dan Ibu Dosen, khususnya Dosen Program Studi Ilmu Komputer yang telah memberikan ilmu – ilmu yang bermanfaat dengan sabar dan ikhlas kepada penulis.
7. Kedua orangtua penulis yang tersayang dan tercinta, atas ayat – ayat cinta, kesabaran dan keikhlasan yang selalu mengiringi

langkah hidupku, dukungan berupa materi, nasehat, motivasi, dan kepercayaan untukku menuntut ilmu disini.

8. Kakak – kakak tercinta dan tersayang Mafkari Hamnon, Mafulah Hamnon, dan Maya Masyitah Hamnon.
9. Para pegawai dan staf Jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang.
10. Nanang Akhmad Choirul Anam, atas ilmu kehidupan yang telah banyak mengubah cara pandangku terhadap kehidupan dan ilmu yang tak pernah kudapat dalam perkuliahan.
11. Keluarga keduaku, yaitu Yudi Ariesta Sanjaya sekeluarga dan Susilawati sekeluarga atas kasih sayang, dukungan, nasehat, motivasi, dan semua bantuannya selama ini.
12. Saudara – saudaraku satu kos Mbak Desthi, Adinda, Zam Zami, Anya dan Nela.
13. Sahabatku tercinta dan tersayang, Novi, Nova, Tebe, Adityo Nugroho, Susan, Inunk, Cink, Betha Nurina Sari, Ary Alfrian, Bapak Supardi, Johan, Mustika Rahayu, Aini, Citra, Raafi, Dita, Nurul, Betha Noor Indahsari, Intan, Erin dan Windi.
14. Teman–teman Program Studi Ilmu Komputer Kelas B 2007 Jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang atas segala kepercayaan dan motivasi bahwa aku bisa.
15. Semua kakak tingkat Program Studi Ilmu Komputer yang telah berbagi kisah, saran, kritik, dan motivasi.
16. 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.

Semoga penulisan laporan tugas akhir ini bermanfaat bagi pembaca dan untuk pengembangan selanjutnya. Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan disebabkan oleh keterbatasan kemampuan dan pengalaman, dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca.

Malang, Januari 2012

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iii</b>
<b>LEMBAR PERNYATAAN</b> .....	<b>v</b>
<b>ABSTRAK</b> .....	<b>vii</b>
<b>ABSTRACT</b> .....	<b>ix</b>
<b>KATA PENGANTAR</b> .....	<b>xi</b>
<b>DAFTAR ISI</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvii</b>
<b>DAFTAR TABEL</b> .....	<b>xxi</b>
<b>DAFTAR PSEUDOCODE</b> .....	<b>xxiii</b>
<b>DAFTAR PERSAMAAN</b> .....	<b>xxv</b>
<b>DAFTAR SOURCE CODE</b> .....	<b>xxvii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	4
1.4 Tujuan .....	4
1.5 Manfaat .....	4
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1. Data .....	5
2.2. Data Transaksi.....	5
2.3. Informasi .....	5
2.4. <i>Data Mining</i> .....	6
2.4.1. Pengertian <i>Data Mining</i> .....	6
2.4.2. Tugas <i>Data Mining</i> .....	6
2.4.3. Tahap – Tahap <i>Data Mining</i> .....	7
2.4.4. Teknik <i>Data Mining</i> .....	10
2.5. <i>Association Rule</i> .....	11
2.5.1. <i>Support</i> .....	12
2.5.2. <i>Confidence</i> .....	13
2.5.3. Algoritma <i>Apriori</i> .....	14
2.5.4. <i>FP-tree</i> .....	14
2.5.5. Algoritma <i>FP-Growth</i> .....	18
2.6. <i>Lift Ratio</i> .....	20

<b>BAB III METODOLOGI DAN PERANCANGAN.....</b>	<b>23</b>
3.1. Pengumpulan Data .....	24
3.2. Analisis dan Perancangan Sistem.....	25
3.2.1. Deskripsi Sistem.....	25
3.2.2. Rancangan Pembuatan Sistem.....	25
3.2.3. Rancangan Basis Data .....	43
3.3. Perhitungan Manual .....	44
3.4. Rancangan Antar Muka.....	65
3.5. Rancangan Uji Coba.....	70
<b>BAB IV IMPLEMENTASI DAN PEMBAHASAN.....</b>	<b>73</b>
4.1. Lingkungan Implementasi.....	73
4.1.1. Lingkungan Perangkat Keras.....	73
4.1.2. Lingkungan Perangkat Lunak.....	73
4.2. Implementasi Program .....	73
4.2.1. Implementasi Tahap <i>Frequent 1-itemset</i> .....	76
4.2.1.1. Implementasi Pembacaan Data Transaksi.....	76
4.2.1.2. Implementasi Pengkoleksian Golongan Buku dan Frekuensinya.....	78
4.2.1.3. Implementasi Penyortiran Kode Golongan Buku.....	79
4.2.1.4. Implementasi Pencarian Kode Golongan Buku yang Frequent .....	80
4.2.1.5. Implementasi Pengkoleksian Transaksi dengan Kode Golongan Buku yang Frequent .....	82
4.2.2. Implementasi Tahap <i>Frequent Itemset</i> .....	83
4.2.2.1. Implementasi Pembentukan FP-tree.....	83
4.2.2.2. Implementasi Conditional Pattern Base .....	85
4.2.2.3. Implementasi Conditional FP-tree.....	86
4.2.2.4. Implementasi Koleksi Frequent Itemset.....	87
4.2.3. Implementasi Tahap <i>Generate Rule</i> .....	87
4.2.3.1. Implementasi Hitung Confidence.....	87
4.2.3.2. Implementasi Koleksi Rule .....	89
4.2.4. Implementasi Tahap <i>Lift Ratio</i> .....	90
4.3. Implementasi Antarmuka .....	91
4.4. Pengujian Sistem.....	96
4.5. Analisis Hasil .....	108
<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>119</b>

5.1. Kesimpulan .....	119
5.2. Saran .....	119
<b>DAFTAR PUSTAKA .....</b>	<b>121</b>
<b>LAMPIRAN.....</b>	<b>124</b>

**UNIVERSITAS BRAWIJAYA**



UNIVERSITAS BRAWIJAYA



## DAFTAR GAMBAR

Gambar 2.1	Tahap – tahap Proses <i>Knowledge Discovery in Database</i> .....	8
Gambar 2.2	Prosedur Umum <i>Data Mining</i> .....	9
Gambar 2.3	<i>FP-tree</i> .....	17
Gambar 3.1	Langkah – langkah Penelitian .....	23
Gambar 3.2	<i>Flowchart</i> Sistem Secara Umum .....	29
Gambar 3.3	<i>Flowchart Association Rule</i> .....	30
Gambar 3.4	<i>Flowchart</i> Algoritma <i>FP-Growth</i> .....	31
Gambar 3.5	<i>Flowchart</i> Pembentukan <i>FP-tree</i> .....	32
Gambar 3.6	<i>Flowchart Frequent 1-itemset</i> .....	33
Gambar 3.7	<i>Flowchart</i> Pengkoleksian Golongan Buku dan Frekuensinya .....	34
Gambar 3.8	<i>Flowchart</i> Penyortiran Golongan Buku Berdasarkan Frekuensinya .....	35
Gambar 3.9	<i>Flowchart</i> Pencarian Kode Golongan Buku yang <i>Frequent</i> .....	36
Gambar 3.10	<i>Flowchart</i> Pengkoleksian Transaksi yang Mengandung Kode Golongan Buku yang <i>frequent</i> .....	37
Gambar 3.11	<i>Flowchart Insert Tree</i> .....	38
Gambar 3.12	<i>Flowchart Conditional Pattern Base</i> .....	39
Gambar 3.13	<i>Flowchart Conditional FP-tree</i> .....	40
Gambar 3.14	<i>Flowchart</i> Koleksi <i>Frequent Itemset</i> .....	41
Gambar 3.15	<i>Flowchart</i> Hitung <i>Lift Ratio</i> .....	42
Gambar 3.16	Diagram Relasi Basis Data .....	44
Gambar 3.17	<i>FP-tree</i> Awal .....	49
Gambar 3.18	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650LEN-10010300149 .....	50
Gambar 3.19	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650LEN-10010300150 .....	51
Gambar 3.20	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650MAR-10010400068 .....	51
Gambar 3.21	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650LEN-10010500071 .....	52
Gambar 3.22	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650LEN-10010500076 .....	53

Gambar 3.23	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650LEN-10010600049 .....	53
Gambar 3.24	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650MAR-10010700143 .....	54
Gambar 3.25	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650MAR-10010700155 .....	55
Gambar 3.26	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650RED-10010700115 .....	56
Gambar 3.27	<i>FP-tree</i> Setelah Pembacaan Kode Transaksi 650KMU-10010800026 .....	57
Gambar 3.28	Lintasan yang Memiliki <i>Suffix</i> {7} .....	58
Gambar 3.29	<i>Conditional FP-tree</i> untuk <i>Suffix</i> {7} .....	59
Gambar 3.30	Lintasan yang Memiliki <i>Suffix</i> {5} .....	59
Gambar 3.31	<i>Conditional FP-tree</i> untuk <i>Suffix</i> {5} .....	60
Gambar 3.32	<i>Conditional FP-tree</i> untuk <i>Suffix</i> {1,5} .....	60
Gambar 3.33	<i>Conditional FP-tree</i> untuk <i>Suffix</i> {6,5} .....	60
Gambar 3.34	Lintasan yang Memiliki <i>Suffix</i> {1} .....	61
Gambar 3.35	<i>Conditional FP-tree</i> untuk <i>Suffix</i> {1} .....	61
Gambar 3.36	Lintasan yang Memiliki <i>Suffix</i> {3} .....	61
Gambar 3.37	<i>Conditional FP-tree</i> untuk <i>Suffix</i> {3} .....	62
Gambar 3.38	Lintasan yang Memiliki <i>Suffix</i> {6} .....	62
Gambar 3.39	<i>Conditional FP-tree</i> untuk <i>Suffix</i> {6} .....	62
Gambar 3.40	Rancangan Antar Muka Secara Umum .....	66
Gambar 3.41	Rancangan Antar Muka Data Transaksi .....	67
Gambar 3.42	Rancangan Antar Muka Submenu <i>Frequent Itemset</i> .....	68
Gambar 3.43	Rancangan Antar Muka Submenu <i>Association Rule</i> .....	69
Gambar 3.44	Rancangan Antar Muka Submenu <i>Lift Ratio</i> .....	69
Gambar 4.1	Tampilan Data Transaksi .....	92
Gambar 4.2	Tampilan Dialog <i>Box</i> Peringatan .....	93
Gambar 4.3	Tampilan Dialog <i>Box</i> Data Telah Tersimpan Pada File Format xls .....	93
Gambar 4.4	Tampilan Submenu <i>Frequent Itemset</i> .....	94
Gambar 4.5	Tampilan Submenu <i>Association Rule</i> .....	95
Gambar 4.6	Tampilan Submenu <i>Lift Ratio</i> .....	96

Gambar 4.7	Grafik Pengaruh Nilai Minimum <i>Support</i> dan Nilai Minimum <i>Confidence</i> terhadap Jumlah <i>Rule</i> .....	109
Gambar 4.8	Grafik Nilai <i>Confidence Rule</i> Hasil Uji Per Bulan .....	110
Gambar 4.9	Grafik Nilai <i>Confidence Rule</i> Hasil Uji Per Dua Bulan.....	111
Gambar 4.10	Grafik Nilai <i>Confidence Rule</i> Hasil Uji Per Triwulan .....	112
Gambar 4.11	Grafik Nilai <i>Confidence Rule</i> Hasil Uji Per Kwartal .....	113
Gambar 4.12	Grafik Nilai <i>Lift Ratio</i> Hasil Uji Per Bulan .....	114
Gambar 4.13	Grafik Nilai <i>Lift Ratio</i> Hasil Uji Per Dua Bulan.....	115
Gambar 4.14	Grafik Nilai <i>Lift Ratio</i> Hasil Uji Per Triwulan .....	116
Gambar 4.15	Grafik Nilai <i>Lift Ratio</i> Hasil Uji Per Kwartal.....	116



UNIVERSITAS BRAWIJAYA



## DAFTAR TABEL

Tabel 2.1	Contoh Data Transaksi .....	17
Tabel 2.2	Hasil Fase Proses Penggalian <i>FP-growth</i> .....	20
Tabel 3.1	Tabel trans .....	43
Tabel 3.2	Tabel golonganbuku.....	43
Tabel 3.3	Sampel Data Transaksi Penjualan Buku Hasil <i>Preprocessing</i> .....	45
Tabel 3.4	Sampel Data Golongan Buku.....	46
Tabel 3.5	Sampel Data Transaksi Penjualan Buku Hasil Transformasi .....	46
Tabel 3.6	Frekuensi Kemunculan Tiap Jenis Buku.....	47
Tabel 3.7	Frekuensi Kemunculan Tiap Jenis Buku Hasil Penyortiran .....	47
Tabel 3.8	<i>Frequent 1-Itemset</i> yang Memenuhi Minimum <i>Support</i> .....	48
Tabel 3.9	Sampel Data Transaksi Setelah Penghapusan Kode Jenis Buku Tidak <i>Frequent</i> dan Penyortiran.....	49
Tabel 3.10	<i>Conditional Pattern Base</i> dari Sampel Data .....	57
Tabel 3.11	Hasil <i>Frequent Itemset</i> Sampel Data.....	63
Tabel 3.12	<i>Confidence Frequent Itemset</i> Sampel Data .....	63
Tabel 3.13	<i>Rule</i> yang Terbentuk .....	64
Tabel 3.14	<i>Lift Ratio Rules</i> .....	65
Tabel 3.15	Tabel Uji Pengaruh Nilai Minimum <i>Support</i> dan Nilai <i>Confidence</i> terhadap Jumlah <i>Rule</i> yang Dihasilkan .....	70
Tabel 3.16	Tabel Uji <i>Lift Ratio Association Rule</i> .....	71
Tabel 4.1	<i>Class</i> Pada Implementasi Program dan Fungsinya .....	74
Tabel 4.2	Hasil Uji Pengaruh Nilai Minimum <i>Support</i> dan Nilai Minimum <i>Confidence</i> terhadap Jumlah <i>Rule</i> yang Dihasilkan.....	97
Tabel 4.3	Hasil Uji Kekuatan ( <i>Lift Ratio</i> ) <i>Rule</i> yang Dihasilkan .....	106
Tabel 4.4	Keterangan <i>Rule</i> Hasil Uji Kekuatan <i>Lift Ratio</i> .....	107
Tabel 4.5	<i>Rule</i> yang Bermanfaat .....	117
Tabel 4.6	Detail Informasi <i>Rule</i> yang Bermanfaat.....	117

UNIVERSITAS BRAWIJAYA



## DAFTAR PSEUDOCODE

Pseudocode 2.1	Pembentukan <i>FP-tree</i> .....	16
Pseudocode 2.2	Algoritma <i>FP-Growth</i> .....	19

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



## DAFTAR PERSAMAAN

Persamaan 2.1	<i>Support (A)</i> .....	12
Persamaan 2.2	<i>Support (A,C)</i> .....	12
Persamaan 2.3	<i>Confidence (A→B)</i> .....	13
Persamaan 2.4	<i>Confidence (A→B)</i> .....	13
Persamaan 2.5	<i>Benchmark Confidence</i> .....	20
Persamaan 2.6	<i>Lift Ratio</i> .....	20

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



## DAFTAR SOURCE CODE

Source Code 4.1	Proses Pembacaan Data Transaksi .....	77
Source Code 4.2	Proses Pengkoleksian Golongan Buku dan Frekuensinya.....	79
Source Code 4.3	Proses Penyortiran Kode Golongan Buku .....	80
Source Code 4.4	Proses Pencarian Kode Golongan Buku yang <i>Frequent</i> .....	81
Source Code 4.5	Proses Pengkoleksian Transaksi dengan Kode Golongan Buku yang <i>Frequent</i> .....	82
Source Code 4.6	Proses Pembentukan <i>FP-tree</i> .....	84
Source Code 4.7	Proses <i>Conditional Pattern Base</i> .....	85
Source Code 4.8	Proses <i>Conditional FP-tree</i> .....	86
Source Code 4.9	Proses Koleksi <i>Frequent Itemset</i> .....	87
Source Code 4.10	Proses Hitung <i>Confidence</i> .....	89
Source Code 4.11	Proses Koleksi <i>Rule</i> .....	90
Source Code 4.12	Proses <i>Lift Ratio</i> .....	91



UNIVERSITAS BRAWIJAYA



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Banyaknya toko buku saat ini, membuat tiap toko buku harus berupaya untuk bersaing dalam meningkatkan penjualan. Untuk dapat meningkatkan penjualan buku dibutuhkan sebuah strategi bisnis. Strategi bisnis tidak hanya dilakukan dengan menawarkan program diskon kepada konsumen tetapi juga dapat dilakukan dengan menggali data transaksi penjualan buku untuk mendapatkan pola penjualan buku. Sebuah toko buku memiliki jumlah data buku sangat banyak dan setiap harinya menangani transaksi penjualan buku dari ratusan hingga ribuan transaksi. Penggalan data merupakan salah satu cara yang cukup efektif untuk mengetahui adanya serangkaian pola informasi dari sejumlah besar data yang ada. Dengan mengetahui pola penjualan buku, maka akan didapatkan informasi yang berguna untuk pendukung keputusan (*decision support*) dalam meningkatkan pelayanan kepada konsumen dan meningkatkan penjualan buku. Informasi yang didapatkan misalnya informasi mengenai buku – buku yang berelasi sehingga dapat menginformasikan atau menyarankan konsumen buku – buku yang berelasi, menjaga ketersediaan stok baik buku yang sering dibeli maupun buku yang berelasi agar berimbang, dan peletakkan buku – buku yang berelasi pada rak buku yang berdekatan.

Data yang berskala besar akan mengalami kesulitan bila dianalisis dengan analisis data sederhana sehingga dibutuhkan suatu teknik khusus untuk dapat menganalisis data berskala besar, yaitu *data mining*. *Data mining* dapat dikatakan serangkaian proses untuk mendapatkan informasi yang tidak diketahui secara manual dari kumpulan data (Pramudiono, 2003). Ada berbagai macam teknik dalam *data mining*, dimana penggunaannya tergantung dari permasalahan yang dihadapi. Dalam permasalahan analisis penggalan data transaksi penjualan barang dibutuhkan teknik *data mining* yang dapat mencari korelasi antara barang – barang yang berbeda dengan mengetahui pola asosiasinya (Rochmah,2010). Teknik *data mining* tersebut dikenal dengan teknik *association rule* atau kaidah asosiasi.

Dalam prosesnya, *association rule* terdiri dari dua tahap, antara lain mencari kombinasi *item* yang paling sering terjadi

(mencari *frequent itemset*) dan membangkitkan *rule* yang telah terbentuk dari *frequent itemset* (Han dan Kamber, 2001). Salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data adalah *FP-Growth* (Samuel, 2008). Algoritma *FP-Growth* melakukan pencarian *frequent itemset* tidak dengan membangkitkan kandidat melainkan dengan menggunakan konsep pembangunan *tree* (Erwin, 2009). 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.

Pada data transaksi penjualan buku, *itemset* yang dimaksud adalah golongan buku yang dibeli. Tiap transaksi, pembeli dapat melakukan pembelian lebih dari satu buku dimana buku yang dibeli bisa jadi merupakan golongan buku yang sama atau bisa juga merupakan golongan buku yang berbeda. Pembelian buku yang sama dapat terjadi kembali pada transaksi berikutnya sehingga tiap buku memiliki frekuensi kejadian. Begitu juga dengan golongan buku yang dapat dimodelkan sebagai berikut, misalnya transaksi ( $T$ ) dengan nomor transaksi ke  $i$  ( $T_i$ ) dan dengan kode golongan buku yang dibeli  $j$  ( $T_{ij}$ ) sehingga  $T_{11}, T_{12}, T_{13}$  merupakan himpunan transaksi dengan nomor transaksi 1 dan buku yang dibeli adalah buku dengan kode golongan buku 1, 2, dan 3. Pada transaksi berikutnya ( $T_{i+1}$ ) terjadi pembelian buku dengan kode golongan buku 1, 2, dan 5, maka dapat dituliskan  $T_{21}, T_{22}, T_{25}$ . Hal ini berarti golongan buku 1 dan 2 memiliki frekuensi kejadian lebih dari satu sehingga dengan model data transaksi tersebut dapat dilakukan penggalian data untuk menemukan keterkaitan antara golongan buku yang satu dengan golongan buku yang lain dengan menggunakan algoritma *FP-Growth*.

Dari penelitian sebelumnya, yang dilakukan oleh Erwin(2009) tentang *market basket analysis*, didapatkan hasil bahwa algoritma *FP – Growth* dapat memampatkan data transaksi 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 Verhein(2008) 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* dalam menentukan relasi antara golongan buku yang satu dengan golongan buku yang lain, dengan judul penelitian **“Penerapan metode *association rule* dengan algoritma *FP – Growth* pada data transaksi penjualan buku”**.

## **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 pola penjualan buku berdasarkan golongan buku dari data transaksi penjualan buku.
2. Bagaimana tingkat kekuatan (*lift-ratio*) *association rule* hasil dari penerapan algoritma *FP – Growth*.

### 1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut :

1. Penelitian dilakukan dengan menggunakan data transaksi penjualan buku di toko buku Togamas Malang pada bulan Januari sampai Desember tahun 2010.
2. Input yang dibutuhkan untuk sistem antara lain data transaksi penjualan buku toko buku Togamas Malang dengan rentang bulan dan periode data berbeda, besar minimum *support*, dan besar minimum *confidence*.
3. Output yang dihasilkan adalah aturan asosiasi (*association rule*).

### 1.4 Tujuan

Tujuan dari skripsi ini adalah :

1. Mengetahui pola penjualan buku berdasarkan golongan buku dari data transaksi penjualan buku dengan menerapkan metode *association rule* menggunakan algoritma *FP-Growth*.
2. Mengetahui tingkat kekuatan (*lift-ratio*) *association rule* yang dihasilkan dari penerapan algoritma *FP – Growth*.

### 1.5 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut :

1. Memberikan informasi mengenai pola penjualan buku di toko buku Togamas Malang
2. Untuk pemanfaatan lebih lanjut, informasi mengenai pola penjualan buku diharapkan dapat mendukung keputusan misalnya dalam peletakkan rak buku, dan penyediaan stok buku berdasarkan golongan buku.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. 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 simbol (Kusrini, 2007).

#### **2.2. Data Transaksi**

Transaksi yang terjadi dalam suatu perusahaan dapat dibagi menjadi dua kelompok, antara lain (Kotler dan Kevin, 2007) :

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 (Kotler dan Kevin, 2007).

#### **2.3. Informasi**

Data yang telah diproses dan diinterpretasikan menjadi sesuatu yang berguna disebut informasi. Informasi berguna dalam pengambilan keputusan karena memberikan tambahan pengetahuan sehingga meminimalkan ketidakpastian dari suatu masalah. Pemrosesan data menjadi informasi merupakan proses manipulasi atau pengubahan data dengan tujuan meningkatkan fungsi dari data itu sendiri (Kusrini, 2007).

## 2.4. *Data Mining*

### 2.4.1. Pengertian *Data Mining*

Lahirnya *data mining* diawali oleh kebutuhan terhadap informasi dari data yang melimpah sebagai pendukung keputusan untuk solusi bisnis dan dukungan infrastruktur bidang teknologi informasi (Moertini, 2002).

Kata *mining* sendiri memiliki arti usaha untuk memperoleh sesuatu yang berharga dari material dasar dalam jumlah besar. Berdasarkan pengertian kata *mining* maka *data mining* dapat diartikan usaha untuk memperoleh sesuatu yang berharga (informasi) dari data dalam jumlah besar (Pramudiono, 2003).

Menurut Han dan Kamber (2001), *data mining* merupakan solusi yang mampu menemukan kandungan informasi yang tersembunyi berupa pola dan aturan dari sekumpulan data yang besar agar mudah dipahami. Informasi yang tersembunyi ini sangat menguntungkan dari sudut pandang penelitian, bisnis dan lainnya.

“*Data mining* didefinisikan sebuah proses untuk menemukan hubungan, pola, dan tren baru yang bermakna dengan menyaring data yang sangat besar, yang tersimpan dalam penyimpanan, menggunakan teknik pengenalan pola seperti teknik statistik dan matematika” (Kusrini, 2007).

### 2.4.2. Tugas *Data Mining*

Fungsionalitas dari *data mining* dan pola pencarian yang dapat ditemukan dalam *data mining* adalah sebagai berikut (Han dan Kamber, 2001) :

#### 1. Deskripsi kelas

Data dapat diasosiasikan dengan kelas, yang berguna untuk menggambarkan kelas secara individual dan konsep secara tepat.

#### 2. Analisis *association*

Analisis *association* adalah penemuan *association rules* yang menunjukkan nilai kondisi dari atribut yang terjadi secara bersama – sama dan terus – menerus dalam membentuk sekumpulan data.

#### 3. Klasifikasi dan Prediksi

Klasifikasi adalah proses untuk menemukan sekumpulan model yang menggambarkan dan membedakan kelas data, yang bertujuan agar model tersebut dapat digunakan untuk memprediksi kelas suatu obyek yang belum diketahui kelasnya.

Klasifikasi dapat digunakan untuk memprediksi baik data nominal maupun data diskrit.

4. Analisis *Cluster*

Analisis *cluster* berbeda dengan klasifikasi. *Clustering* menganalisis obyek data dengan kelas yang belum diketahui, sedangkan klasifikasi digunakan untuk menganalisis obyek data dari kelas yang telah diketahui.

5. Analisis *Outlier*

*Outlier* dapat dideteksi menggunakan tes yang bersifat statistik yang mengambil sebuah distribusi dan probabilitas model untuk data atau menggunakan ukuran jarak, dimana obyek yang jaraknya jauh dari *cluster* yang lain dianggap sebagai *outlier*.

6. Analisis *Evolution*

Data analisis *evolution* ini mencari model atau obyek yang memiliki kebiasaan berubah setiap waktu. Analisis ini berkaitan dengan *time-series*. Analisis *evolution* dapat berupa *characterization, discrimination, association, classification, dan clustering*.

Fungsi *data mining* adalah untuk menspesifikasikan pola yang harus ditemukan. Secara umum tugas *data mining* dapat diklasifikasikan dalam dua kategori yaitu deskriptif dan prediktif. Tugas *data mining* secara deskriptif adalah untuk mengklasifikasikan sifat umum suatu data di dalam *database*. Tugas *data mining* secara prediktif adalah untuk mengambil kesimpulan terhadap data dalam membuat prediksi (Han dan Kamber, 2001).

### 2.4.3. Tahap – Tahap *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)* (Kusrini, 2007).

Proses *knowledge discovery in database (KDD)* secara umum adalah sebagai berikut (Kusrini, 2007) :

#### 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 perubahan sesuai format yang dibutuhkan untuk digunakan dalam proses *data mining*.

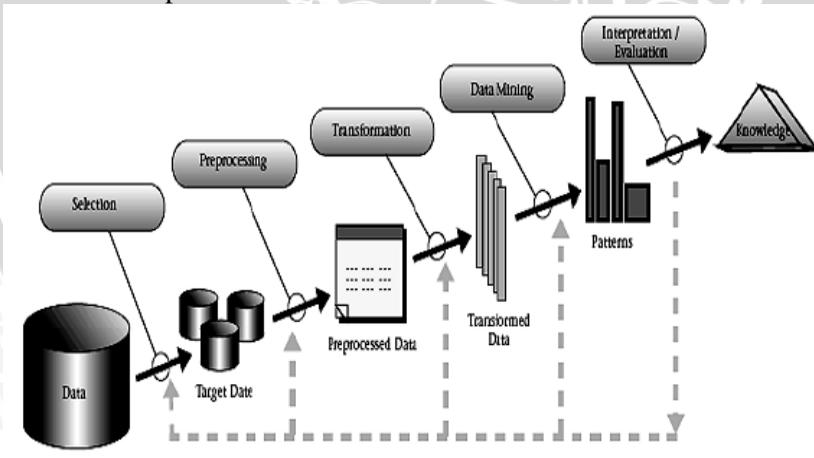
## 4. *Data Mining*

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* atau *Evaluation*

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 dan Kamber, 2001)

Menurut Kantardzic (2003), 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 variabel – variabel yang terlibat.

**2. Mengumpulkan data**

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

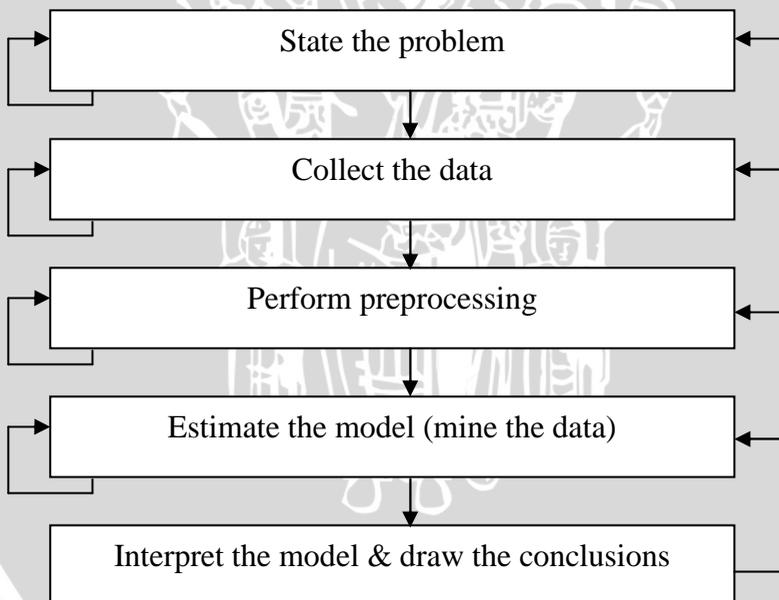
**3. Preprocessing data**

*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** Prosedur Umum *Data Mining* (Kantardzic, 2003)

#### 2.4.4. Teknik *Data Mining*

Menurut Pramudiono (2003), 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 pengelompokan data tanpa berdasarkan kelas data tertentu. *Clustering* dapat dipakai untuk memberikan label pada kelas data yang belum diketahui.

Menurut Kantardzic (2003), 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.

### 2.5. *Association Rule*

*Association rule* adalah suatu prosedur untuk mencari hubungan antar item dalam suatu kumpulan data yang ditentukan. Dalam menentukan suatu *association rule*, terdapat suatu ukuran kepercayaan yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu. *Association rule* memberikan informasi dalam bentuk "if – then" atau "jika – maka". Biasanya digunakan istilah *antecedent* untuk mewakili bagian "jika" dan *consequent* untuk mewakili bagian "maka" (Santosa, 2007).

Dalam *data mining*, *association rule* merupakan pekerjaan untuk menentukan atribut yang akan didapatkan bersamaan. Tugas dari *association rule* adalah mencari aturan yang tersembunyi untuk mengukur hubungan antara dua atau lebih atribut (Sholichah, 2009). Menurut Kantardzic (2003), *association rule* merupakan bentuk dari *data mining* yang menyerupai proses menggali sesuatu yang sangat berharga dari basis data yang besar, dimana sesuatu yang sangat berharga adalah *rule* yang nilai confidence-nya lebih besar dari nilai yang telah ditetapkan sebagai batas minimum. *Rule* tersebut menggambarkan isi dari basis data yang belum diketahui dan mungkin tidak bisa diungkapkan secara eksplisit. Metode ini mencari semua *interesting pattern* yang mungkin dari dalam basis data.

*Association rule* digunakan untuk menemukan hubungan diantara data atau bagaimana suatu kelompok data mempengaruhi suatu keberadaan data yang lain. Metode ini dapat membantu mengenali pola – pola tertentu di dalam kumpulan data yang besar. Dalam *association rule*, suatu kelompok *item* dinamakan *itemset* (Ruldeviyani dan Muhammad, 2008).

*Association rule* meliputi dua tahap, yaitu (Han dan Kamber, 2001):

1. Mencari kombinasi yang paling sering terjadi dari suatu *itemset*.
2. Membangkitkan *association rule* dari *frequent itemset* yang telah dibuat sebelumnya.

Umumnya dalam *association rule* terdapat dua ukuran, yaitu *support* dan *confidence* (Han dan Kamber, 2001). Sebuah *association rule* dengan *confidence* sama atau lebih dari *minimum confidence*, dapat dikatakan sebagai *valid association rule* (Ruldeviyani dan Muhammad, 2008). *Rule* yang memenuhi baik minimum *support* maupun minimum *confidence* disebut juga *strong rule* (Handojo dkk, 2004).

### 2.5.1. *Support*

*Support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan transaksi (Han dan Kamber, 2001). Menurut Rochmah (2010), *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 → Y” adalah probabilitas atribut atau kumpulan atribut X dan Y yang terjadi bersamaan dalam suatu transaksi (Yulita, 2002). 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 (Rochmah, 2010).

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

$$Support(A, C) = P(A \cap C) = \frac{\text{Jumlah Transaksi yang Mengandung A dan C}}{\text{Total Transaksi}} \quad (2.2)$$

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 (Yulita, 2002).

### 2.5.2. Confidence

*Confidence* adalah suatu ukuran yang menunjukkan hubungan antara dua atau lebih item secara *conditional* (Han dan Kamber, 2001). Menurut Rochmah (2010), *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  $m/n$  (Yulita, 2002). Nilai *confidence* dapat diperoleh dengan rumus pada Persamaan 2.3 dan dijabarkan lebih khusus pada Persamaan 2.4 (Rochmah, 2010).

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

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

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 (Yulita, 2002).

### 2.5.3. Algoritma Apriori

Menurut Erwin (2009), algoritma *apriori* menggunakan pengetahuan mengenai *frequent itemset* yang telah diketahui sebelumnya, untuk memproses informasi selanjutnya. Pada algoritma *apriori* untuk menentukan kandidat – kandidat yang mungkin muncul dengan cara memperhatikan minimum *support*. Adapun dua proses utama yang dilakukan dalam algoritma *apriori*, yaitu (Han dan Kamber, 2001) :

#### 1. Join (Penggabungan)

Pada proses ini, setiap *item* dikombinasikan dengan *item* yang lainnya sampai tidak terbentuk kombinasi lagi.

#### 2. Prune (Pemangkasan)

Pada proses ini, hasil dari *item* yang telah dikombinasikan tadi lalu dipangkas dengan menggunakan minimum *support* yang telah ditentukan.

Kekurangan yang dimiliki oleh algoritma *apriori* yaitu, untuk melakukan pencarian *frequent itemset* harus melakukan pemindaian basis data berulang kali untuk setiap kombinasi *item*. Hal tersebut menyebabkan banyaknya waktu yang dibutuhkan untuk melakukan pemindaian basis data. Selain itu, dibutuhkan *generate candidate* yang besar untuk mendapatkan kombinasi *item* dari basis data (Erwin, 2009).

### 2.5.4. FP-tree

*FP-tree* merupakan struktur penyimpanan data yang dimampatkan. *FP-tree* dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu sehingga bila ada transaksi yang memiliki *item* yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki *item* yang sama, maka proses pemampatan dengan struktur data *FP-tree* semakin efektif. Kelebihan dari *FP-tree* adalah hanya memerlukan dua kali pemindaian data transaksi yang terbukti sangat efisien (Samuel, 2008).

Algoritma *FP-Growth* mencari himpunan *item* yang sering muncul dengan cara membangun *FP-tree* secara rekursif dan menggabungkan *frequent itemset* yang ada pada *conditional FP-tree* secara berturut – turut. *FP-tree* adalah sebuah *tree* yang terdiri dari satu *header* tabel, satu *root* yang diberi label “null”, dan satu himpunan *item prefix subtree* sebagai node dari anak *root*. Masing –

masing *entry* dalam *header table* adalah *frequent item*, dan setiap *record* terdiri dari dua atribut yaitu nama *item* dan *head of node-link*. Sedangkan setiap node anak terdiri dari atribut : nama *item*, *count*, dan *node-link*. *Node-link* menghubungkan node ke node berikutnya pada *tree* tersebut yang mempunyai nama *item* yang sama atau *null* jika tidak ada (Febriyana, 2009).

Jika *tree* mempunyai lebih dari satu cabang maka satu *frequent pattern* dibangkitkan dan juga dibangun *conditional FP-tree* untuk *frequent pattern* tersebut. Proses pembangunan *conditional FP-tree* sama seperti proses pembangunan *FP-tree*. *Conditional FP-tree* dibangun dari pola dasar *conditional (conditional pattern base)*. *Conditional pattern base* *i* adalah himpunan *prefix path* dari node *i* yang diperoleh dari penelusuran *node-link* (Febriyana, 2009).

Menurut Han dan Kamber (2001), keuntungan dari penggunaan struktur data *FP-tree*, selain lebih padat, juga tetap lengkap. Informasi yang tidak relevan dikurangi, *item – item* yang tidak sering muncul (*infrequent*) dihilangkan, akan tetapi tetap menjaga kelengkapan informasi yang dibutuhkan untuk *frequent pattern mining*.

Langkah awal dari algoritma pembuatan *FP-tree* adalah dilakukan pembacaan basis data terlebih dahulu dan membuat kumpulan dari *frequent 1-itemset* beserta nilai *support* kemudian diurutkan secara *descending* berdasarkan nilai *support*. Pembuatan *FP-tree* dimulai dengan membuat *root* dan diberi nama *null*. Kemudian item dari tiap transaksi diurutkan sesuai dengan urutan *frequent 1-itemset* dan dibuat cabang pada *FP-tree* untuk setiap transaksi. Setiap cabang baru akan ditambahkan ke dalam *FP-tree* maka dilakukan penelusuran apakah node sudah ada. Jika node sudah ada maka dilakukan penambahan nilai *support* sebanyak 1. Dalam proses pembuatan *FP-tree* tidak hanya dilakukan pembuatan struktur pohon tetapi juga dilakukan pembuatan tabel untuk melacak dimana setiap item ditempatkan dalam *FP-tree*. Setiap item yang sama pada cabang yang berbeda dihubungkan dengan *link* yang dinamakan *nodelink* (Yiapanis, 2006). *Pseudocode* dari algoritma pembentukan *FP-tree* dapat dilihat pada *Pseudocode 2.1*.

**Masukan** : Basis data Transaksi(D), dan nilai minimum *support*(minsup)

**Keluaran**: FP-tree Tree;

**Fungsi** : FP-tree dibentuk dengan cara berikut

- (1) Transaksi di database (D) ditelusuri sekali agar didapat himpunan *frequent item*(F) dan nilai *support*nya.
- (2) Lalu F diurutkan mulai dari yang mempunyai nilai *support* yang terbesar sampai terkecil.
- (3) Buat *root* dari FP-tree(T) dan diberi nama null.
- (4) Untuk tiap transaksi **Trans** di D lakukan {
- (5) pilih dan dilakukan pengurutan *frequent item* dalam Trans(disebut [p|P]) mengacu pada F dimana p adalah elemen pertama dan P adalah *item* berikutnya.
- (6) Panggil **insert\_tree([p|P],T);**
- (7) Fungsi **insert\_tree([p|P],T)** terdiri dari langkah - langkah berikut :
  - (8) inialisasi N = anak dari T
  - (9) inialisasi nama\_item = nama dari item yang terdapat pada T
  - (10) **Jika** (N.nama\_item == p.nama\_item) {N.count ++;}
  - (11) **Jika tidak** {
    - buat node baru N;N.count = 1;N.parent\_link =T;
    - N\_link = node yang sama nama itemnya
    - dihubungkan oleh struktur *node\_link*;
  - (12) **Jika** P tidak kosong {insert\_tree(P,N)}

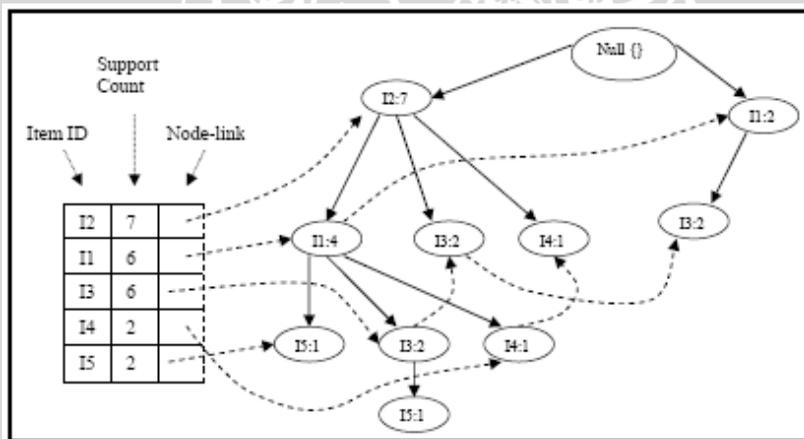
**Pseudocode 2.1** Pembentukan *FP-tree* (Yiapanis, 2006)

Contoh penerapan algoritma pembuatan *FP-tree* adalah dimisalkan terdapat data transaksi pada Tabel 2.1 dengan *minimum support* 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 <I2 :1>, <I1:1>, dan <I5 :1>. 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 *FP-tree* maka tidak perlu dibentuk node yang sama melainkan

memampatkan pada node yang telah ada dengan menambahkan nilai *support* pada node I2 sehingga node I2 menjadi <I2 :2> dan menciptakan node baru <I4 :1> menjadi anak dari node <I2 :2> . Hasil *FP-tree* setelah pembacaan semua transaksi dari Tabel 2.1 dapat dilihat pada Gambar 2.3 (Yiapanis, 2006).

**Tabel 2.1** Contoh Data Transaksi (Yiapanis, 2006)

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 2.3** *FP-tree* (Yiapanis, 2006)

### 2.5.5. 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*) (Samuel, 2008). 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* (Erwin, 2009).

*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 (Samuel, 2008).

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* (Erwin, 2009).

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*) (Febriyana, 2009).

Fase dari algoritma *FP-growth* terdiri dari dua fase yaitu fase pembentukan *FP-tree* dan fase proses penggalian *FP-growth* (Yiapanis, 2006). Fase pembentukan *FP-tree* telah dijelaskan pada subbab *FP-tree*. Sedangkan fase dari proses penggalian *FP-growth* dijelaskan pada *Pseudocode 2.2*.

```

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

```

**Pseudocode 2.2** Algoritma *FP – Growth* (Yiapanis, 2006)

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 (Yiapanis, 2006).

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 penambahan 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 2.2.

**Tabel 2.2** 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	$\{\{I2 : 4\}\}$	$\langle I2 : 4 \rangle$	$\{I2, I1 : 4\}$

## 2.6. 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 (Santosa, 2007). Rumus *benchmark confidence* dapat dilihat pada Persamaan 2.5, sedangkan rumus *lift ratio* dapat dilihat pada Persamaan 2.6.

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

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 (Santosa, 2007).

*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  $support(Y)$ .

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



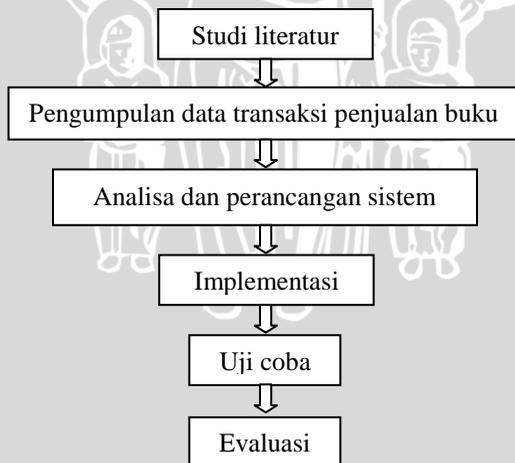
### BAB III METODOLOGI DAN PERANCANGAN

Pembahasan pada bab ini meliputi metode dan langkah – langkah perancangan yang dilakukan dalam penelitian untuk penggalian data transaksi penjualan buku 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 buku.
3. Menganalisa dan melakukan perancangan sistem untuk menggali *association rule* dari data transaksi penjualan buku.
4. Mengimplementasikan rancangan yang dilakukan pada tahap sebelumnya menjadi sebuah perangkat lunak untuk analisa pola *association rule* dari data transaksi penjualan buku.
5. Melakukan uji coba terhadap perangkat lunak menggunakan data transaksi penjualan buku 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 dilakukan dengan metode pengumpulan data lapangan. Data yang dimaksud adalah data transaksi penjualan buku. Sumber data yang digunakan adalah laporan transaksi penjualan buku dari toko buku Togamas Malang. Data yang diperoleh adalah data transaksi penjualan dari bulan Januari sampai bulan Desember tahun 2010 sehingga *dataset* terdiri dari 12 data dengan rincian sebagai berikut :

1. Data transaksi bulan Januari terdiri dari 65530 *record* data.
2. Data transaksi bulan Februari terdiri dari 65530 *record* data.
3. Data transaksi bulan Maret terdiri dari 65530 *record* data.
4. Data transaksi bulan April terdiri dari 64090 *record* data.
5. Data transaksi bulan Mei terdiri dari 61897 *record* data.
6. Data transaksi bulan Juni terdiri dari 65530 *record* data.
7. Data transaksi bulan Juli terdiri dari 65530 *record* data.
8. Data transaksi bulan Agustus terdiri dari 65530 *record* data.
9. Data transaksi bulan September terdiri dari 65530 *record* data.
10. Data transaksi bulan Oktober terdiri dari 65530 *record* data.
11. Data transaksi bulan November terdiri dari 65530 *record* data.
12. Data transaksi bulan Desember terdiri dari 65530 *record* data.

Rincian data transaksi penjualan tersebut tidak hanya terdiri dari data penjualan buku saja tetapi juga terdiri dari data penjualan alat tulis.

*Dataset* yang didapat terdiri dari 5 atribut, yaitu kode transaksi, tanggal transaksi, nama buku, penerbit buku dan golongan buku.

Kode transaksi terdiri dari informasi kode kasir, tahun transaksi(*yy*), bulan transaksi(*mm*), tanggal transaksi(*dd*), dan nomor urut transaksi pada kasir dan tanggal bersangkutan . Format tanggal transaksi adalah tanggal, bulan, dan tahun transaksi (*dd/mm/yyyy*).

Berdasarkan hasil pengamatan dari data transaksi penjualan buku yang terkumpul, jenis buku yang ada pada toko buku Togamas Malang adalah sebanyak 84 jenis buku dan terbagi menjadi 13 golongan buku.

## 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 buku untuk mengetahui pola penjualan buku berdasarkan golongan buku. Parameter yang digunakan dalam analisa adalah nilai *minimum support*, nilai *minimum confidence*, dan periode data transaksi. Periode data yang disediakan adalah per bulan, per dua bulan, triwulan, dan kwartal. Hasil keluaran dari sistem berupa pola *association rule* yang menunjukkan keterkaitan antara golongan buku yang satu dengan golongan buku yang lain.

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 golongan buku sehingga *frequent itemset*-nya berupa himpunan golongan buku yang sering muncul, yang diperoleh dari hasil pemindaian basis data dan yang memenuhi nilai *minimum support* yang menjadi masukan dalam sistem. *Frequent itemset* yang dihasilkan oleh algoritma *FP-Growth* selanjutnya dalam sistem akan dilakukan perhitungan nilai *confidence*-nya. Pada tahap ini, metode yang diimplementasikan pada sistem adalah *association rule*, yaitu proses pembangkitan *frequent itemset* menjadi *rule* untuk *frequent itemset* yang memiliki nilai *confidence* yang memenuhi nilai *minimum confidence* (lebih besar atau sama dengan *minimum confidence*).

*Rule – rule* yang terbentuk akan ditampilkan sebagai hasil keluaran dalam bentuk kode golongan buku maupun nama golongan buku beserta nilai *confidence* dan nilai *lift ratio rule* tersebut. Sistem akan menyimpan hasil keluaran hanya dalam file berformat xls.

### 3.2.2. Rancangan Pembuatan Sistem

Dalam Dalam proses penelitian ini, metode *data mining* yang digunakan adalah *association rule* dengan menggunakan algoritma *FP – Growth*. Metode ini diterapkan untuk menganalisa frekuensi kemunculan masing – masing golongan buku dengan menggunakan parameter waktu transaksi.

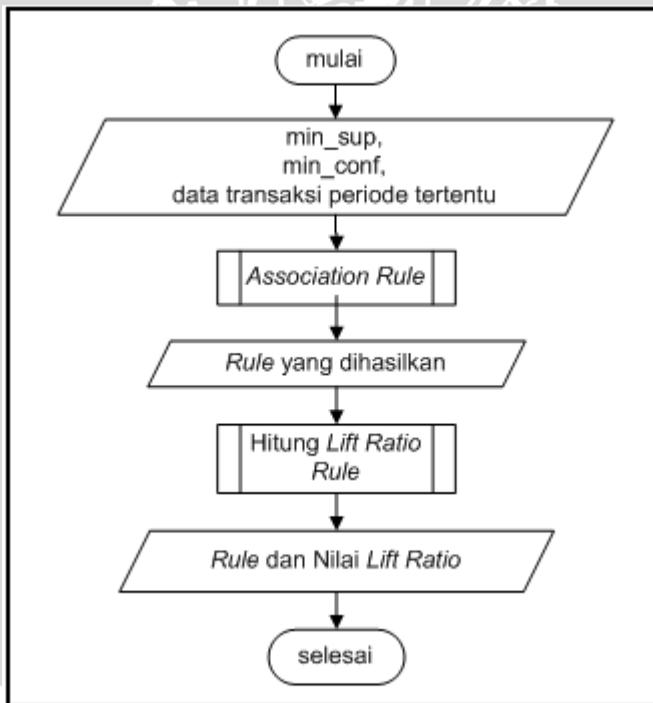
Untuk melakukan analisa terhadap data transaksi penjualan buku, perlu dilakukan proses yang meliputi :

1. Sistem menerima masukan berupa *minimum support* (minsup), *minimum confidence* (minconf), dan data transaksi penjualan buku rentang bulan tertentu dan periode tertentu. Kemudian sistem akan melakukan proses *association rule*. Hasil keluaran sistem adalah *rule* yang dihasilkan dari proses *association rule*. *Rule* yang dihasilkan selanjutnya dihitung nilai *lift ratio* sehingga dihasilkan keluaran *rule* dan nilai *lift ratio*. Proses ini merupakan gambaran secara umum sistem dan dapat dilihat pada Gambar 3.2.
2. Dalam proses *association rule*, dilakukan proses pembentukan *FP-tree* dan dilanjutkan proses *FP-Growth* untuk mendapatkan *frequent itemset*. Setiap *frequent itemset* dihitung nilai *confidence*-nya. Nilai *confidence* dari tiap *frequent itemset* tersebut dibandingkan dengan nilai *minimum confidence* (minconf). Jika nilai *confidence* lebih besar atau sama dengan nilai *minimum confidence* (minconf), maka *frequent item* tersebut disimpan dalam tabel *rule*, sedangkan bila tidak maka *frequent item* tersebut dilewati. Diagram alir proses pada tahap ini dapat dilihat pada Gambar 3.3.
3. Proses pertama dalam *FP-Growth* untuk mendapatkan *frequent itemset* adalah dilakukan pengkoleksian lintasan yang terbentuk pada *tree* hasil *FP-tree*. Kemudian dilanjutkan dengan proses *conditional pattern base* dan *conditional FP-tree*. Selanjutnya, hasil dari proses *conditional FP-tree* dikoleksi yang menjadi *frequent itemset*. Diagram alir proses pada tahap ini dapat dilihat pada Gambar 3.4.
4. Pembentukan *FP-tree* ditunjukkan pada Gambar 3.5. Langkah pertama adalah dilakukan pembacaan data transaksi kemudian dilanjutkan dengan proses pencarian *frequent 1-itemset*. Proses pencarian *frequent 1-itemset* menghasilkan data transaksi dengan kode golongan buku yang *frequent* yang tersortir secara *descending* berdasarkan nilai frekuensi. Setiap data transaksi tersebut selanjutnya dipindai untuk dilakukan proses *insert tree*.
5. Proses pencarian *frequent 1-itemset* ditunjukkan pada Gambar 3.6. Langkah awal dari proses ini adalah mengkoleksi kode golongan buku yang muncul pada data transaksi dan frekuensinya. Hasil dari proses pengkoleksian kode golongan

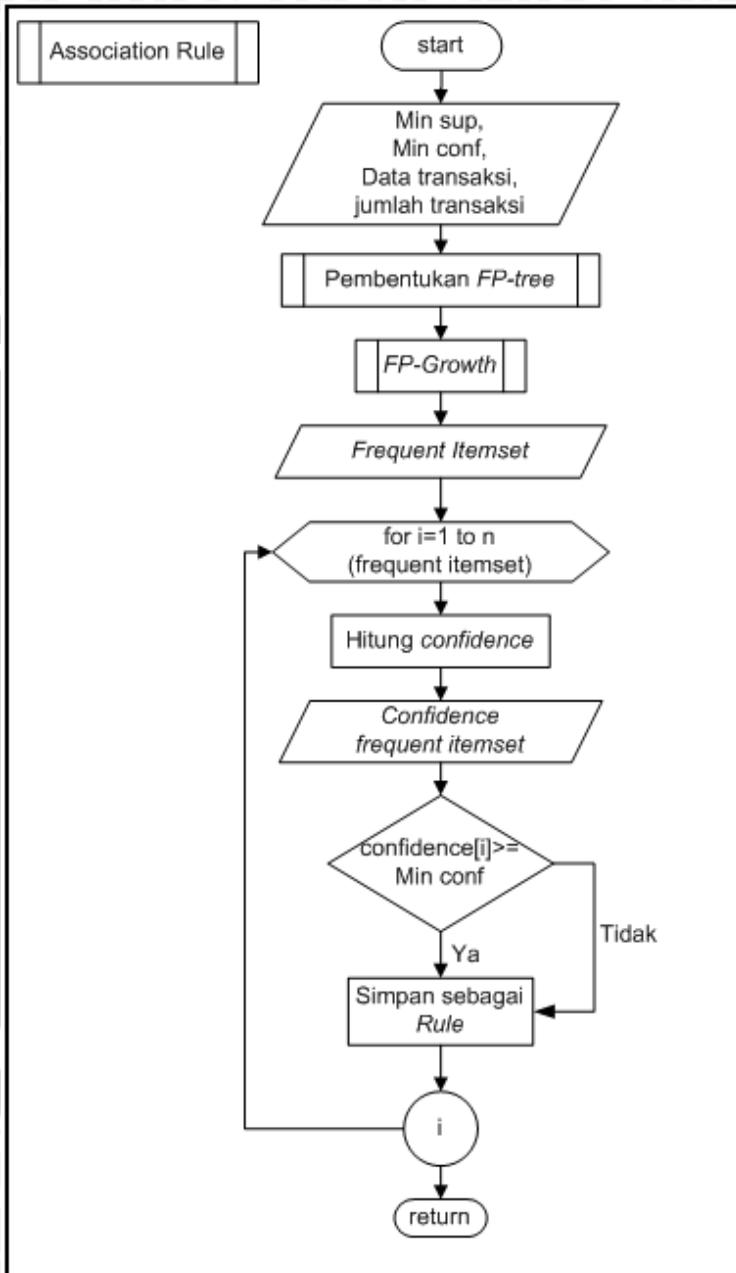
- buku dan frekuensinya tersebut dilanjutkan dengan proses penyortiran kode golongan buku berdasarkan frekuensinya dari terbesar ke terkecil. Proses dilanjutkan dengan pencarian kode golongan buku yang *frequent*. Kode golongan buku yang *frequent* tersebut kemudian digunakan untuk mengkoleksi data transaksi yang memiliki kode golongan tersebut.
6. Pengkoleksian kode golongan buku dan frekuensinya ditunjukkan pada Gambar 3.7 dimana dilakukan pemindaian pada data transaksi penjualan buku dan dilakukan inisialisasi data frek untuk menyimpan data frekuensi tiap kode golongan buku. Kemudian pada setiap data transaksi, akan diambil kode golongan bukunya. Jika kode golongan buku tersebut sudah ada pada data frek maka frekuensi kode golongan buku tersebut ditambah 1 nilainya, sedangkan jika belum ada maka kode golongan buku disimpan dalam data frek beserta nilai frekuensinya sebanyak 1. Keluaran dari proses ini adalah data frek yang berisi kode golongan buku dan nilai frekuensinya.
  7. Proses selanjutnya adalah penyortiran kode golongan buku berdasarkan frekuensi kode golongan buku yang ditunjukkan pada Gambar 3.8. Penyortiran dilakukan secara *descending* atau terurut dari nilai frekuensi yang terbesar ke terkecil. Setiap data frek dibaca dan dibandingkan nilai frekuensinya terhadap data frek yang lain. Bila nilai frekuensi kode golongan buku yang sedang dibaca lebih kecil dari nilai frekuensi kode golongan buku yang dibandingkan maka isi data frek akan ditukar.
  8. Selanjutnya dilakukan proses pencarian kode golongan buku yang *frequent*. Proses ini ditunjukkan pada Gambar 3.9. Setiap nilai frekuensi kode golongan buku pada data frek dihitung nilai *support*. Kemudian nilai *support* tersebut dibandingkan dengan nilai minimum *support*. Jika nilai *support* lebih besar atau sama dengan nilai minimum *support* maka data kode golongan buku dan nilai *support* disimpan sebagai *frequent 1-itemset*.
  9. Proses terakhir yang merupakan *breaking down* dari proses *frequent 1-itemset* adalah proses pengkoleksian transaksi dengan kode golongan buku yang *frequent* yang ditunjukkan pada Gambar 3.10. Pada proses ini, setiap data *frequent 1-itemset* diambil data golongan bukunya. Kemudian dilakukan pencarian transaksi yang mengandung kode golongan buku tersebut. Selanjutnya, dilakukan pengecekan terhadap *key* dari *hashmap*

- H untuk setiap data transaksi yang mengandung kode golongan buku tersebut. Jika tidak terdapat *key* yang merupakan kode transaksi tersebut maka kode transaksi disimpan sebagai *key* dan kode golongan buku disimpan sebagai *value*. Sedangkan jika terdapat *key* yang merupakan kode transaksi tersebut maka pada *key* tersebut, *value* ditambahkan kode golongan buku.
10. Proses selanjutnya adalah *insert tree* yang merupakan *breaking down* dari proses *insert tree* pada proses pembentukan *FP-tree* pada Gambar 3.5. Proses *insert tree* ini ditunjukkan pada Gambar 3.11. Pada proses *insert tree* diinisialisasi terlebih dahulu  $root = null$ . Kemudian untuk setiap item yang dimasukkan, dilakukan pengecekan terhadap *root* apakah *root* memiliki simpul anak dengan nama *item* yang sama dengan item yang dimasukkan. Apabila ada yang sama maka *N.count* ditambahkan nilainya sebanyak 1. Bila tidak maka buat simpul anak dengan nama *item* sama dengan nama item tersebut, *N.parent* adalah *root*, *N.count* = 1, dan *N.link* menghubungkan dengan node yang memiliki nama *item* yang sama.
  11. Bila semua kode golongan buku pada baris data transaksi telah dimasukkan pada *FP-tree* maka dilakukan proses *Conditional Pattern Base*. Pada tahap ini pada setiap *path* (lintasan) dilakukan pencarian *prefix* (kode golongan buku yang menyusun lintasan) untuk setiap *suffix* dimana yang menjadi *suffix* adalah setiap kode golongan buku pada *frequent 1-itemset*. Proses ini ditunjukkan pada Gambar 3.12.
  12. Tahap selanjutnya adalah *conditional FP-tree*, yaitu menghitung *support count* dari setiap kombinasi kode golongan buku yang menjadi *prefix*, yang memiliki nilai *support* diatas atau sama dengan nilai minimum *support*, dengan *suffix* hasil dari proses *conditional pattern base*. Proses dilakukan dengan mengecek pada setiap data lintasan (*path*) apakah memiliki *suffix* yang sama dengan *suffix* kombinasi. Jika sama maka dilakukan pengecekan terlebih dahulu apakah kode golongan buku penyusun data lintasan tersebut memiliki semua kode golongan buku yang menjadi *prefix* kombinasi. Jika semua kode golongan buku dimiliki data lintasan maka bobot data lintasan tersebut dijumlahkan. Kemudian nilai bobot tersebut dibagi dengan jumlah transaksi lalu disimpan sebagai data *frequent itemset*. Proses ini ditunjukkan pada Gambar 3.13.

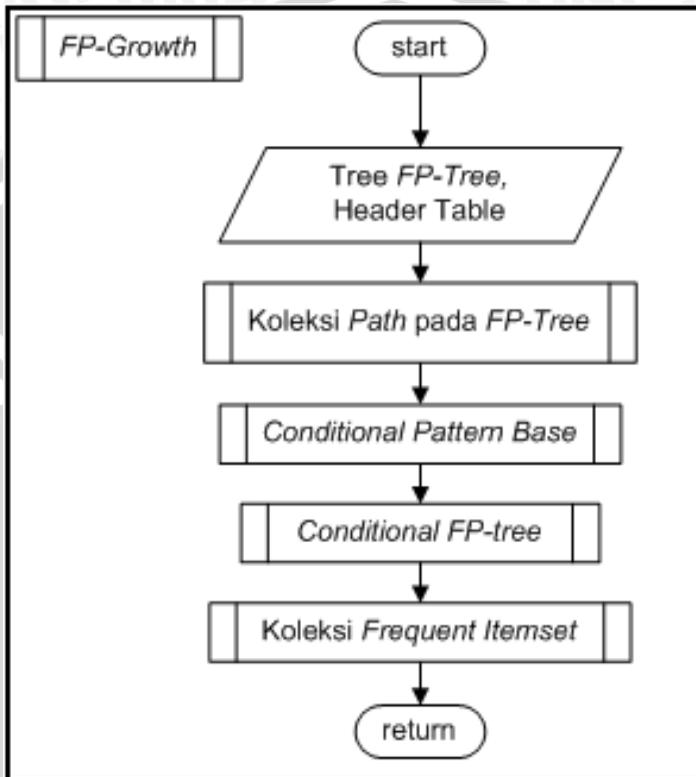
13. Proses terakhir yang merupakan *breaking down* dari proses *fp-growth* adalah proses koleksi *frequent itemset* yang ditunjukkan pada Gambar 3.14. Pada proses ini setiap baris data *frequent itemset* dilakukan pengecekan apakah nilai bobotnya kurang dari minimum *support*. Jika nilai bobotnya kurang maka data tersebut dihapus dari data *frequent itemset*.
14. Proses terakhir dalam sistem adalah perhitungan *lift ratio* yang ditunjukkan pada Gambar 3.15. Pada proses ini setiap data *frequent itemset* yang menjadi *rule* dihitung nilai *lift ratio* dengan cara setiap *suffix* data *frequent itemset* yang menjadi *rule* dicari nilai *support*-nya pada data *frequent 1-itemset* disimpan sebagai *supConsequent*. Kemudian nilai *confidence* dari *frequent itemset* yang menjadi *rule* dibagi dengan nilai *supConsequent* lalu disimpan *rule* dan nilai *lift ratio rule* tersebut.



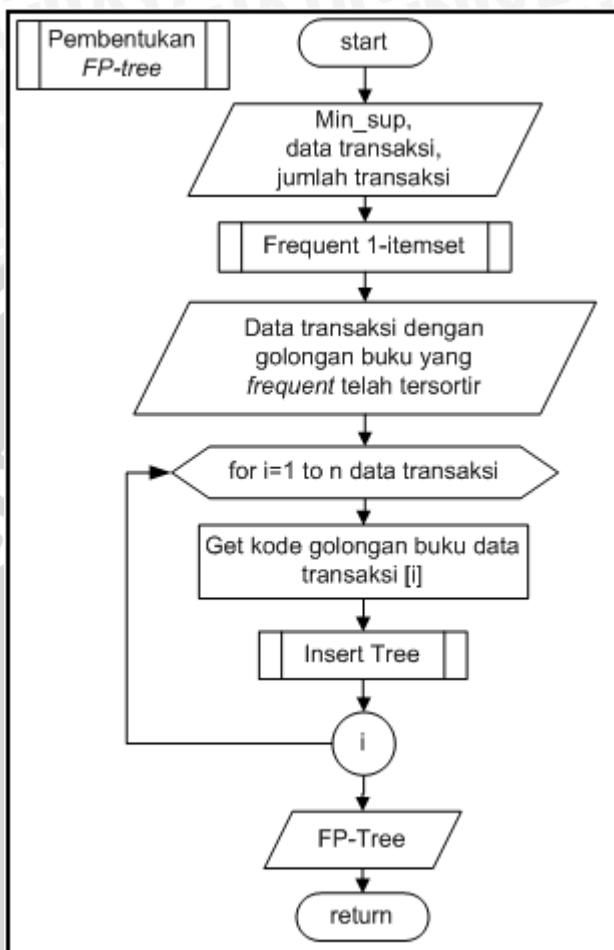
**Gambar 3.2** Flowchart Sistem Secara Umum



**Gambar 3.3** Flowchart Association Rule



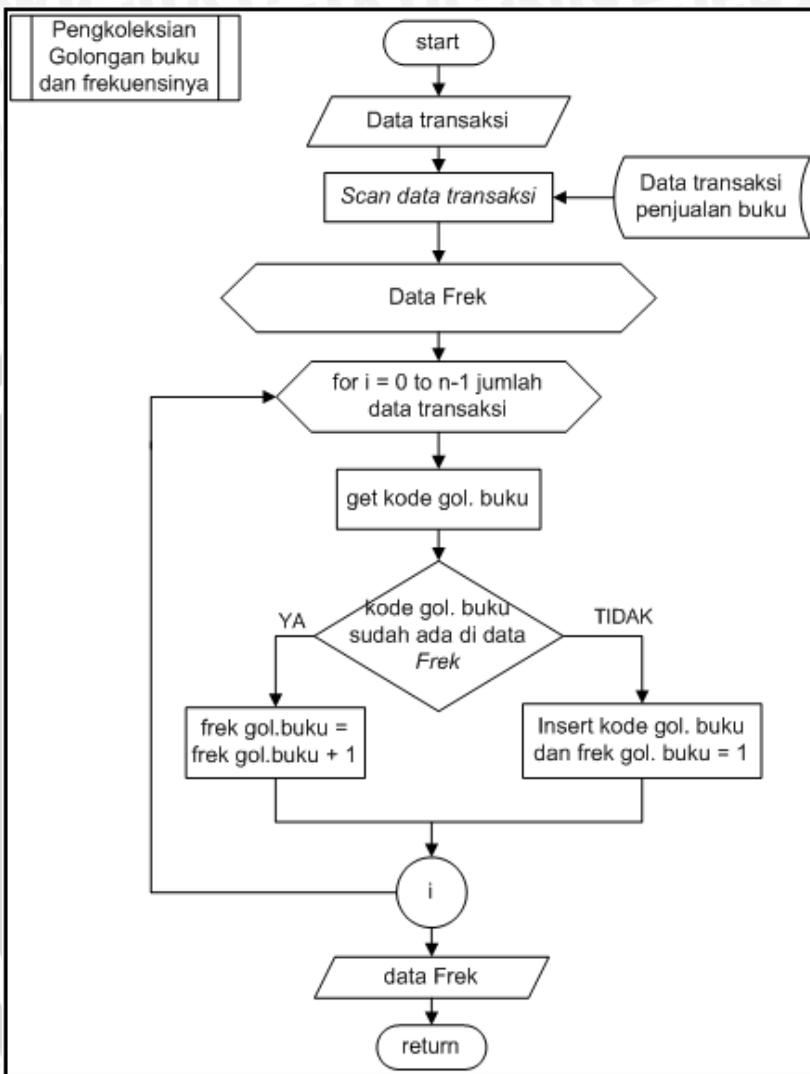
**Gambar 3.4** *Flowchart* Algoritma *FP-Growth*



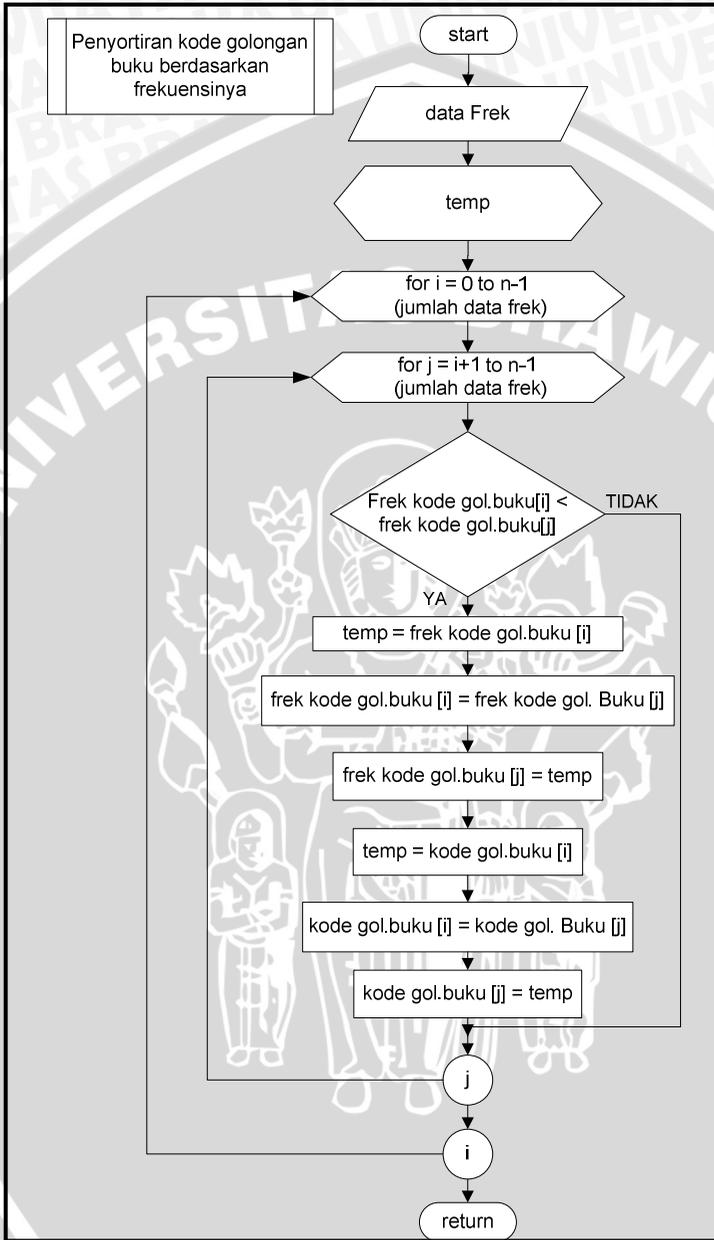
**Gambar 3.5** Flowchart Pembentukan *FP-tree*



**Gambar 3.6** *Flowchart Frequent 1-itemset*

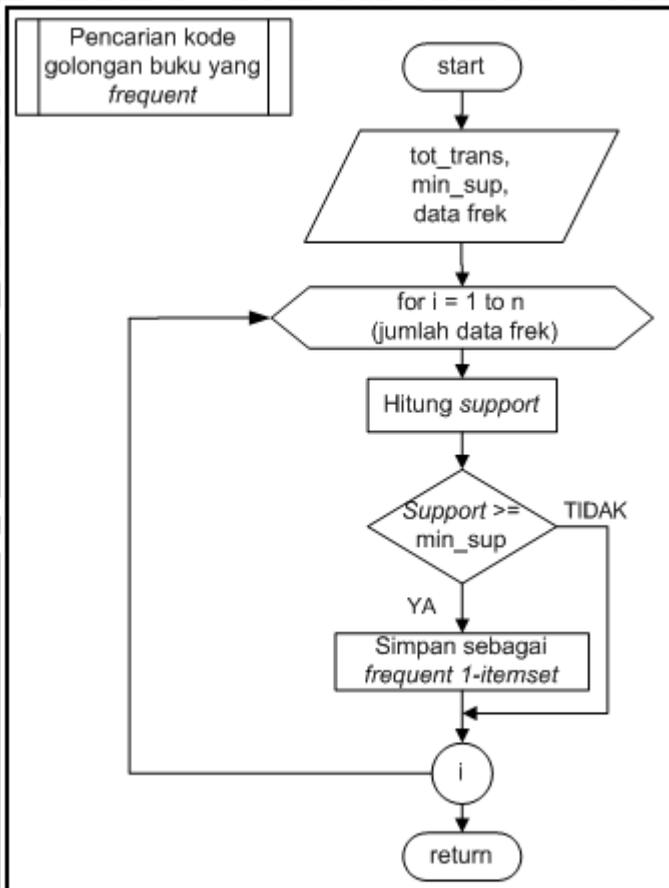


**Gambar 3.7** Flowchart Pengoleksian Golongan Buku dan Frekuensinya

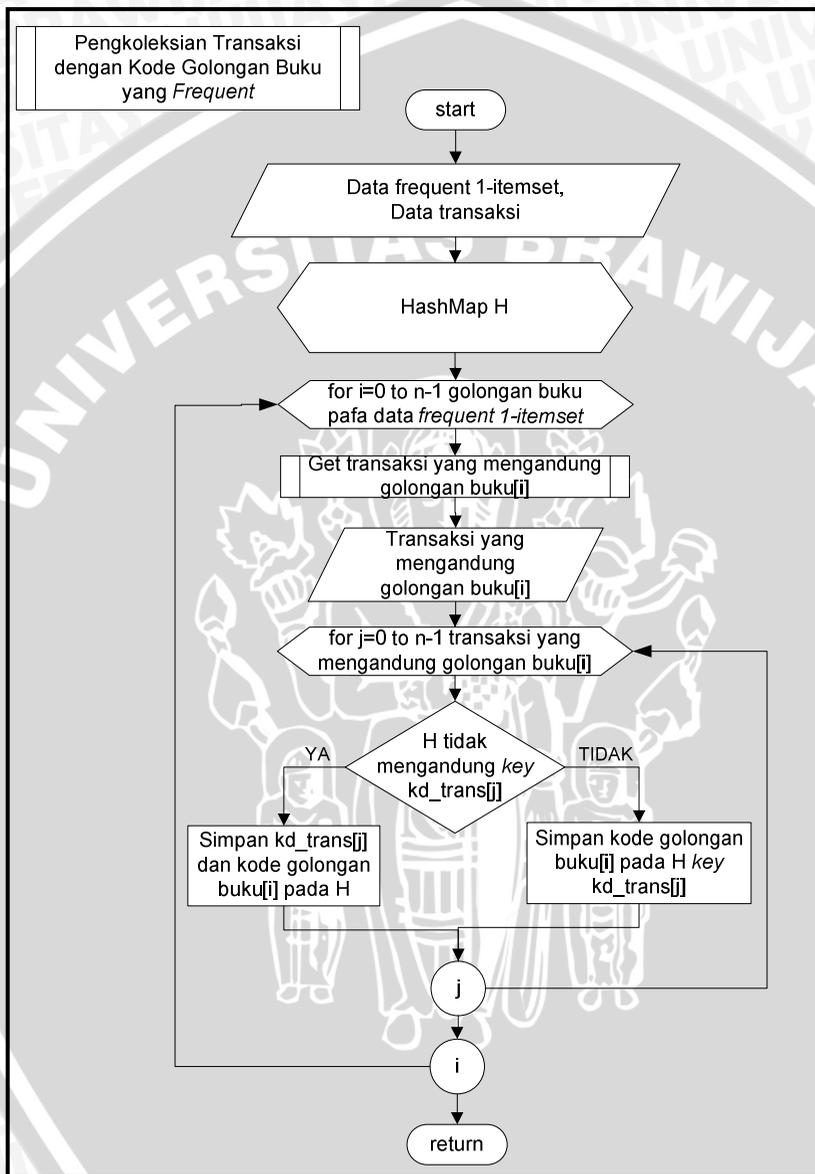


Penyortiran kode golongan buku berdasarkan frekuensinya

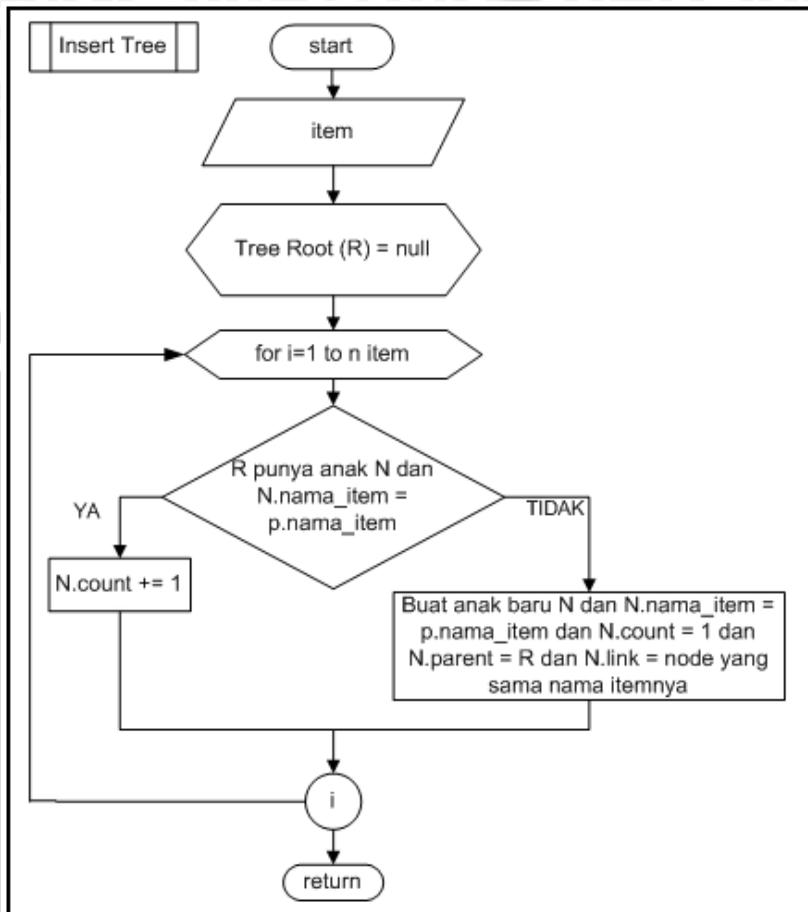
**Gambar 3.8** Flowchart Penyortiran Golongan Buku Berdasarkan Frekuensinya



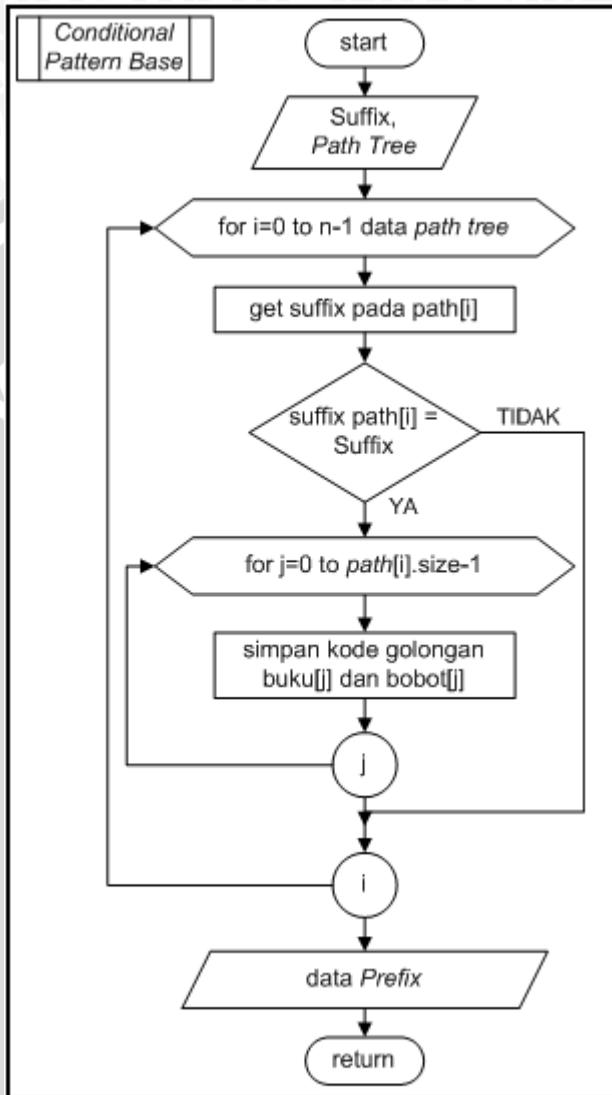
**Gambar 3.9** Flowchart Pencarian Kode Golongan Buku yang *Frequent*



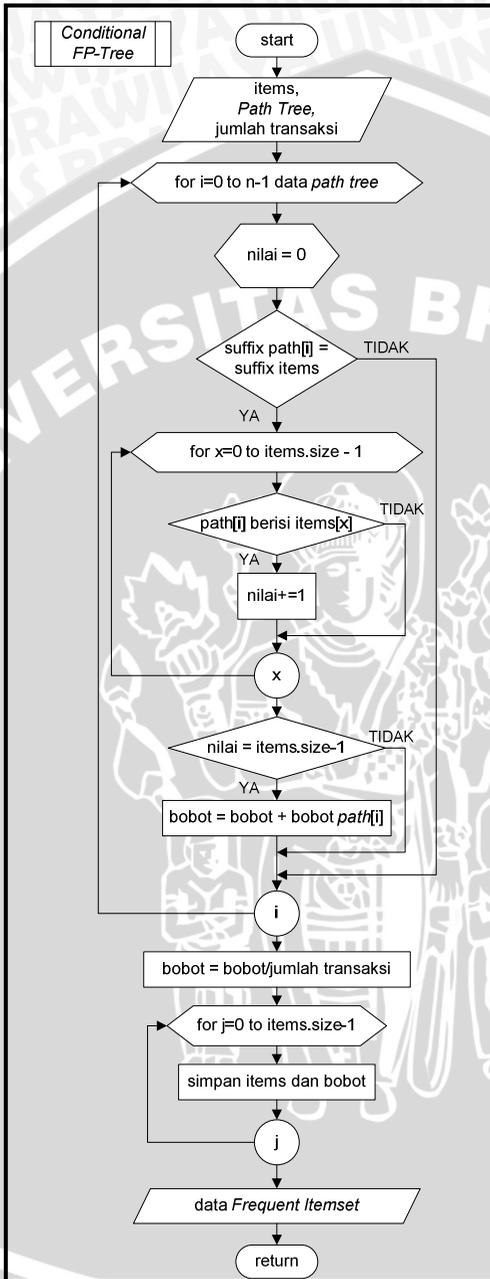
**Gambar 3.10** Flowchart Pengkoleksian Transaksi yang Mengandung Kode Golongan Buku yang *Frequent*



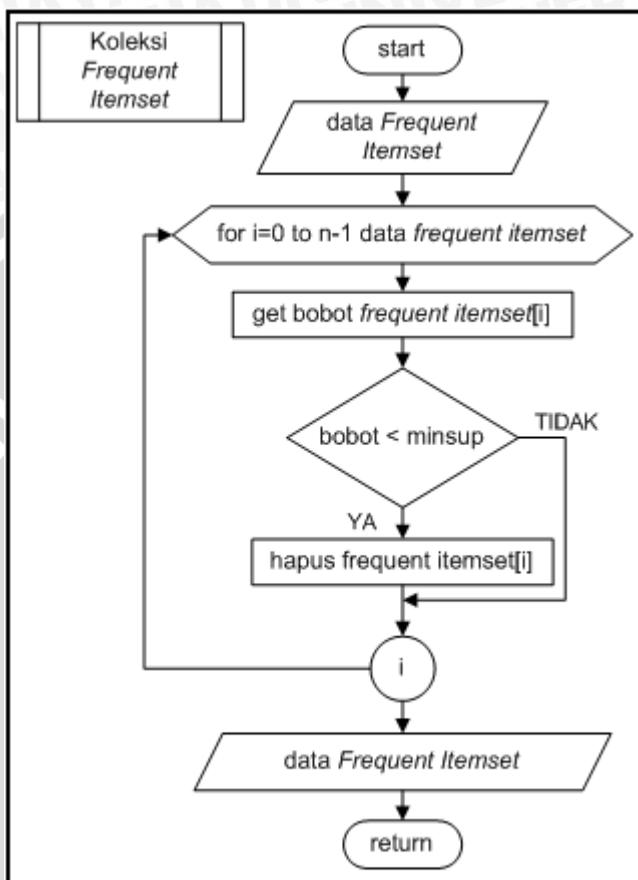
**Gambar 3.11** *Flowchart Insert Tree*



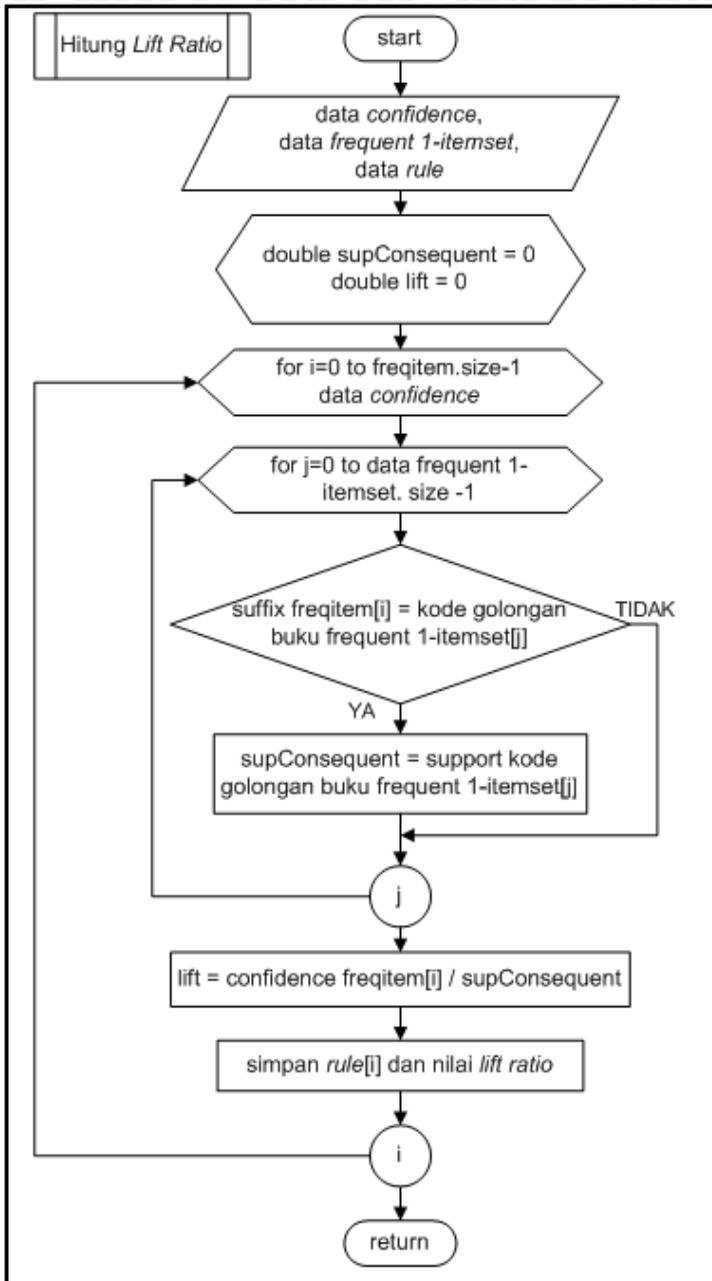
**Gambar 3.12** Flowchart Conditional Pattern Base



Gambar 3.13 Flowchart Conditional FP-tree



**Gambar 3.14** Flowchart Koleksi Frequent Itemset



**Gambar 3.15** Flowchart Hitung Lift Ratio

### 3.2.3. Rancangan Basis Data

Dalam penelitian ini, dibutuhkan data kode transaksi, tanggal transaksi, kode golongan buku, dan golongan buku. Oleh karena itu, *dataset* dari hasil pengumpulan data diproses terlebih dahulu. Proses yang dilakukan adalah seleksi data, *preprocessing* data dan transformasi data. Karena data transaksi penjualan terdiri dari data transaksi penjualan buku dan data transaksi penjualan alat tulis maka dilakukan seleksi data untuk menyeleksi data transaksi penjualan buku dari keseluruhan data yang ada. Selanjutnya hasil dari seleksi data tersebut dilakukan *preprocessing* data dengan melakukan pembersihan data dari *field* yang tidak dibutuhkan seperti *field* nama buku, tanggal transaksi dan nama penerbit, pembersihan dari data dengan nilai yang hilang (*missing value*), dan memperbaiki kesalahan pada data seperti pengisian nilai pada *field* yang tidak tepat. Hasil dari *preprocessing* data selanjutnya di transformasi ke dalam format data yang dibutuhkan sesuai dengan pola informasi yang akan dicari. Keseluruhan proses tersebut dilakukan diluar sistem yang akan dibuat.

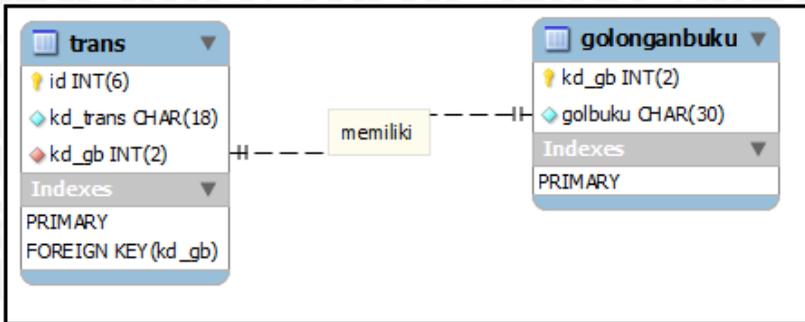
Pada penelitian ini, pola informasi yang akan dicari adalah asosiasi antar golongan buku sehingga *dataset* ditransformasi menjadi dua tabel yaitu tabel trans dan tabel golonganbuku. Tabel trans terdiri dari atribut id, kode transaksi, dan kode golongan buku seperti pada Tabel 3.1, sedangkan tabel golonganbuku terdiri dari atribut kode golongan buku dan golongan buku seperti pada Tabel 3.2.

**Tabel 3.1** Tabel trans

Atribut	Tipe Data	Keterangan
<u>id</u>	char(6)	Id
kd_trans	char(18)	kode transaksi
kd_gb	char(2)	kode golongan buku

**Tabel 3.2** Tabel golonganbuku

Atribut	Tipe Data	Keterangan
<u>kd_gb</u>	char(2)	kode golongan buku
golbuku	char(41)	nama golongan buku



**Gambar 3.16** Diagram Relasi Basis Data

Berdasarkan diagram relasi basis data, tabel yang berelasi antara lain tabel transaksi dengan tabel golongan buku. Tabel transaksi terhadap tabel golongan buku memiliki kardinalitas relasi satu ke satu. Meskipun pada sebenarnya tiap transaksi dapat melakukan pembelian banyak golongan buku, namun dalam basis data pembuatan tabel – tabel tersebut bertujuan untuk memetakan.

### 3.3. Perhitungan Manual

Perhitungan manual dilakukan pada sampel data yang diambil acak dari data transaksi penjualan buku toko buku Togamas pada bulan Januari sebanyak 10 transaksi. Nilai minimum *support* yang digunakan pada proses perhitungan manual adalah 20% sedangkan nilai minimum *confidence* yang digunakan adalah 50%. Sampel data yang digunakan dapat dilihat pada lampiran A.

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 tanggal transaksi, nama buku dan penerbit. Selain itu, untuk golongan buku yang sama yang dibeli pada satu transaksi cukup dituliskan satu kali saja pada transaksi tersebut. Sampel data yang telah melalui tahap *preprocessing* data dapat dilihat pada Tabel 3.3.

Langkah kedua adalah tahap transformasi data dimana data hasil *preprocessing* data diubah bentuk sesuai dengan kebutuhan. Tahap transformasi yang dilakukan adalah mengubah menjadi dua tabel yaitu tabel golongan buku dan tabel transaksi. Tabel golongan buku digunakan untuk menyimpan informasi kode golongan buku dan golongan buku, sedangkan tabel transaksi digunakan untuk menyimpan informasi kode transaksi dan kode golongan buku. Hasil dari tahap transformasi sampel data dapat dilihat pada Tabel 3.4 dan Tabel 3.5.

**Tabel 3.3** Sampel Data Transaksi Penjualan Buku Hasil *Preprocessing*

<b>kode transaksi</b>	<b>golongan buku</b>
650LEN-10010300149	Media
	Umum
650LEN-10010300150	Agama
	Media
	Umum
650MAR-10010400068	Media
	Sosial
	Umum
650LEN-10010500071	Bisnis
	Umum
650LEN-10010500076	Media
	Umum
650LEN-10010600049	Media
	Umum
650MAR-10010700143	Agama
	Umum
	Teknik
650MAR-10010700155	Agama
	Buku Anak
	Bisnis
	Umum
	Bahasa dan Sastra

650RED-10010700115	Media
	Umum
650KMU-10010800026	Agama
	Bisnis
	Bahasa dan Sastra

**Tabel 3.4** Sampel Data Golongan Buku

<b>kode golongan buku</b>	<b>golongan buku</b>
1	Agama
2	Buku Anak
3	Media
4	Sosial
5	Bisnis
6	Umum
7	Bahasa dan Sastra
8	Teknik

**Tabel 3.5** Sampel Data Transaksi Penjualan Buku Hasil Transformasi

<b>kode transaksi</b>	<b>kode golongan buku</b>
650LEN-10010300149	3, 6
650LEN-10010300150	1, 3, 6
650MAR-10010400068	3, 4, 6
650LEN-10010500071	5, 6
650LEN-10010500076	3, 6
650LEN-10010600049	3, 6
650MAR-10010700143	1, 6, 8
650MAR-10010700155	1, 2, 5, 6, 7
650RED-10010700115	3, 6
650KMU-10010800026	1, 5, 7

Langkah ketiga, dilakukan pembacaan pada data transaksi yang telah di transformasi untuk menghitung frekuensi kemunculan tiap golongan buku seperti pada Tabel 3.6. Kemudian dilakukan penyortiran berdasarkan frekuensi kemunculannya dari yang terbesar ke yang terkecil seperti pada Tabel 3.7.

**Tabel 3.6** Frekuensi Kemunculan Tiap Golongan Buku

<b>kode golonga buku</b>	<b><i>frekuensi</i></b>
1	4
2	1
3	6
4	1
5	3
6	9
7	2
8	1

**Tabel 3.7** Frekuensi Kemunculan Tiap Golongan Buku Hasil Penyortiran

<b>kode golongan buku</b>	<b><i>frekuensi</i></b>
6	9
3	6
1	4
5	3
7	2
2	1
4	1
8	1

Langkah keempat adalah perhitungan *support* pada *frequent 1-itemset* yang dapat dilihat pada Tabel 3.8.

$$\begin{aligned}
 \text{Support } \{6\} &= \frac{9}{10} = 0.9 \\
 \text{Support } \{3\} &= \frac{6}{10} = 0.6 \\
 \text{Support } \{1\} &= \frac{4}{10} = 0.4 \\
 \text{Support } \{5\} &= \frac{3}{10} = 0.3 \\
 \text{Support } \{7\} &= \frac{2}{10} = 0.2 \\
 \text{Support } \{2\} &= \frac{1}{10} = 0.1 \\
 \text{Support } \{4\} &= \frac{1}{10} = 0.1 \\
 \text{Support } \{8\} &= \frac{1}{10} = 0.1
 \end{aligned}$$

Kode golongan buku 2, 4, dan 8 tidak *frequent* karena nilai *support* masing – masing kode golongan buku tersebut lebih kecil dari nilai *minimum support*. Sedangkan *item* kode golongan buku 6, 3, 1, 5, dan 7 terpilih menjadi *frequent 1-itemset* karena nilai *support*-nya lebih besar atau sama dengan nilai *minimum support*.

**Tabel 3.8** *Frequent 1-Itemset* yang Memenuhi Minimum *Support*

<b>kode golongan buku</b>	<b><i>support</i></b>
6	0.9
3	0.6
1	0.4
5	0.3
7	0.2

Langkah kelima adalah proses penghapusan kode golongan buku yang tidak *frequent* dari data transaksi yang dipilih dan dilakukan penyortiran kode golongan buku pada tiap transaksi berdasarkan nilai *support* tertinggi. Perubahan pada sampel data transaksi dapat dilihat pada Tabel 3.9.

**Tabel 3.9** Sampel Data Transaksi Setelah Penghapusan Kode Golongan Buku Tidak *Frequent* dan Penyortiran.

<b>kode transaksi</b>	<b>kode golongan buku</b>
650LEN-10010300149	6, 3
650LEN-10010300150	6, 3, 1
650MAR-10010400068	6, 3
650LEN-10010500071	6, 5
650LEN-10010500076	6, 3
650LEN-10010600049	6, 3
650MAR-10010700143	6, 1
650MAR-10010700155	6, 1, 5, 7
650RED-10010700115	6, 3
650KMU-10010800026	1, 5, 7

Langkah keenam adalah setiap transaksi dari sampel data transaksi dibuat *FP-tree* yang disajikan pada Gambar 3.17 sampai Gambar 3.27. Pada pembuatan awal *root* diberikan nilai *null*. Pada *header table*, *head of table link* untuk semua kode golongan buku bernilai *null*.

Header Table

<b>kode golongan buku</b>	<b>Head of table link</b>
6	null
3	null
1	null
5	null
7	null

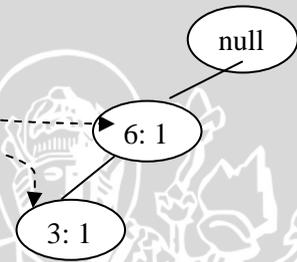
null

**Gambar 3.17** *FP-tree* Awal

Transaksi pertama pada sampel data, yaitu transaksi dengan kode transaksi 650LEN-10010300149 yang dapat dilihat pada Gambar 3.18. Urutan kode golongan buku yang dibeli adalah 6 dan 3. Kode golongan buku 6 merupakan *prefix* sehingga simpul anak *root* berisi kode golongan buku 6 dan *count support* bernilai 1 yang artinya kode golongan buku 6 yang melewati lintasan tersebut adalah sebanyak 1. Selanjutnya kode golongan buku 3 menjadi simpul anak dari simpul berlabel 6. *Head of table link* menunjukkan pada simpul yang mana kode golongan buku tersebut berada dan disimbolkan tanda panah dengan garis panah putus – putus.

Header Table

kode golongan buku	Head of table link
6	
3	
1	null
5	null
7	null

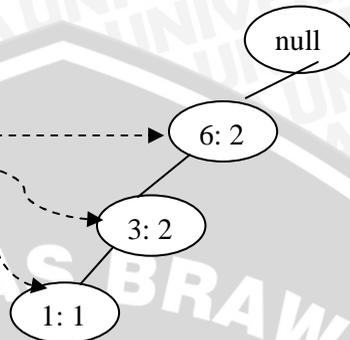


**Gambar 3.18** *FP-tree* Setelah Pembacaan Kode Transaksi 650LEN-10010300149

Transaksi yang dibaca selanjutnya adalah transaksi dengan kode transaksi 650LEN-10010300150 yang disajikan pada Gambar 3.19. Pada transaksi ini, urutan kode golongan buku yang dibeli adalah 6, 3, dan 1. *Prefix* dari transaksi ini sama dengan transaksi sebelumnya yaitu kode golongan buku 6 sehingga tidak dibuat simpul anak baru dari *root*. *Prefix* tersebut ditimpakan pada simpul berlabel 6 sehingga pada simpul tersebut *count support* bertambah 1 (*count support* = 2). Kode golongan selanjutnya yang dimasukkan ke dalam *FP-tree* adalah kode golongan buku 3. Pada simpul berlabel 6 ditelusuri simpul anaknya. Karena simpul anak dari simpul 6 adalah simpul berlabel kode golongan buku 3 maka kode golongan buku 3 pada transaksi ini ditimpakan juga pada simpul berlabel 3 yang telah ada dan *count support* pada simpul berlabel 3 ditambahkan 1 (*count support* = 2). Kemudian kode golongan buku 1 dimasukkan ke dalam *FP-tree* menjadi simpul anak dari simpul 3. Karena pada simpul 3 pada pembacaan transaksi sebelumnya tidak terbentuk simpul anak, maka simpul 1 dibuat dengan *count support* 1.

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	null
7	null

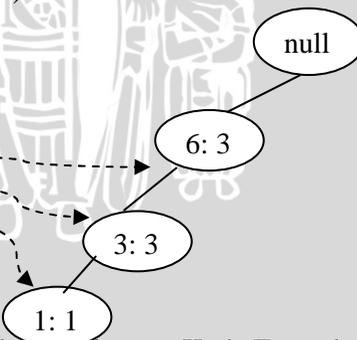


**Gambar 3.19** FP-tree Setelah Pembacaan Kode Transaksi 650LEN-10010300150

Pada Gambar 3.20, transaksi yang dibaca adalah kode transaksi 650MAR-10010400068 dengan urutan kode golongan buku yang dibeli adalah 6, dan 3. *Prefix* dari transaksi ini yaitu kode golongan buku 6 sama dengan transaksi sebelumnya sehingga ditimpakan pada simpul anak *root* yang berlabel 6 sehingga pada simpul tersebut *count support* bertambah 1 (*count support* = 3). Kode golongan buku selanjutnya adalah kode golongan buku 3. Pada simpul berlabel 6 ditelusuri simpul anaknya. Karena simpul anak dari simpul 6 adalah simpul berlabel kode golongan buku 3 maka kode golongan buku 3 pada transaksi ini ditimpakan juga pada simpul berlabel 3 yang telah ada dan *count support* pada simpul berlabel 3 ditambahkan 1 (*count support* = 3).

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	Null
7	Null

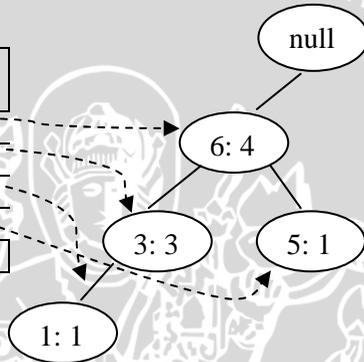


**Gambar 3.20** FP-tree Setelah Pembacaan Kode Transaksi 650MAR-10010400068

Pada Gambar 3.21, transaksi yang dibaca adalah kode transaksi 650LEN-10010500071 dengan urutan kode golongan buku yang dibeli adalah 6 dan 5. *Prefix* dari transaksi ini yaitu kode golongan buku 6 sama dengan transaksi sebelumnya sehingga ditimpakan pada simpul anak *root* yang berlabel 6 sehingga pada simpul tersebut *count support* bertambah 1 (*count support* = 4). Selanjutnya kode golongan buku 5 dimasukkan ke dalam *FP-tree*. Pada simpul berlabel 6 ditelusuri simpul anaknya. Karena simpul anak dari simpul 6 tidak ada simpul anak berlabel 5 maka dibuatkan simpul anak baru dari simpul 6 dengan label 5 dan *count support* = 1.

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	
7	null

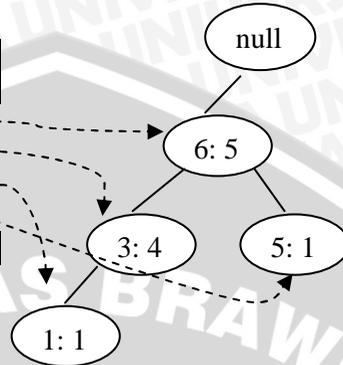


**Gambar 3.21** *FP-tree* Setelah Pembacaan Kode Transaksi 650LEN-10010500071

Pada Gambar 3.22, transaksi yang dibaca adalah kode transaksi 650LEN-10010500076 dengan urutan kode golongan buku yang dibeli adalah 6 dan 3. *Prefix* dari transaksi ini yaitu kode golongan buku 6 sama dengan transaksi sebelumnya sehingga ditimpakan pada simpul anak *root* yang berlabel 6 sehingga pada simpul tersebut *count support* bertambah 1 (*count support* = 5). Selanjutnya kode golongan buku 3 dimasukkan ke dalam *FP-tree*. Pada simpul berlabel 6 ditelusuri simpul anaknya. Karena terdapat simpul anak berlabel 3 maka kode golongan buku 3 ditimpakan pada simpul berlabel 3 tersebut dengan ditambahkan nilai *count support* sebanyak 1 (*count support* = 4).

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	
7	null

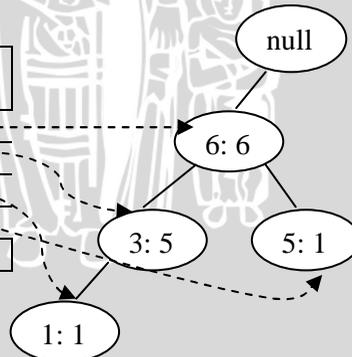


**Gambar 3.22** *FP-tree* Setelah Pembacaan Kode Transaksi 650LEN-10010500076

Pada Gambar 3.23, transaksi yang dibaca adalah kode transaksi 650LEN-10010600049 dengan urutan kode golongan buku yang dibeli adalah 6 dan 3. *Prefix* dari transaksi ini yaitu kode golongan buku 6 sama dengan transaksi sebelumnya sehingga ditimpakan pada simpul anak *root* yang berlabel 6 sehingga pada simpul tersebut *count support* bertambah 1 (*count support* = 6). Selanjutnya kode golongan buku 3 dimasukkan ke dalam *FP-tree*. Pada simpul berlabel 6 ditelusuri simpul anaknya. Karena simpul 6 memiliki simpul anak berlabel 3 maka *count support* simpul 3 ditambahkan 1 (*count support* = 5).

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	
7	null

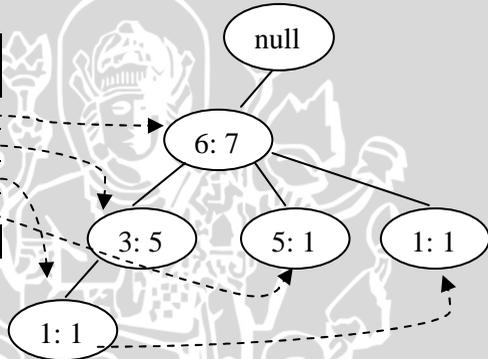


**Gambar 3.23** *FP-tree* Setelah Pembacaan Kode Transaksi 650LEN-10010600049

Transaksi yang dibaca pada gambar 3.24 adalah kode transaksi 650MAR-10010700143 dengan urutan kode golongan buku yang dibeli adalah 6 dan 1. *Prefix* dari transaksi ini yaitu kode golongan buku 6 sama dengan transaksi sebelumnya sehingga ditimpakan pada simpul anak *root* yang berlabel 6 sehingga pada simpul tersebut *count support* bertambah 1 (*count support* = 7). Kode golongan buku selanjutnya yang dimasukkan ke dalam *FP-tree* adalah kode golongan buku 1. Pada simpul berlabel 6 ditelusuri simpul anaknya. Karena simpul 6 tidak memiliki simpul anak berlabel 1 maka dibuatkan simpul anak berlabel 1 dengan *count support* = 1 dan dihubungkan dengan simpul berlabel 1 pada lintasan lain.

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	
7	null



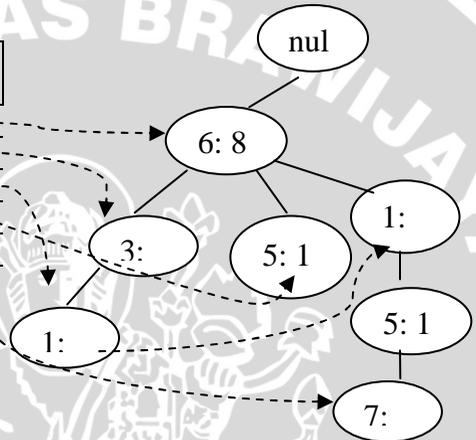
**Gambar 3.24** *FP-tree* Setelah Pembacaan Kode Transaksi 650MAR-10010700143

Transaksi selanjutnya adalah kode transaksi 650MAR-10010700155 dengan urutan kode golongan buku yang dibeli adalah 6, 1, 5, dan 7. Seperti yang dapat dilihat pada gambar 3.25, *prefix* dari transaksi ini yaitu kode golongan buku 6 sama dengan transaksi sebelumnya sehingga ditimpakan pada simpul anak *root* yang berlabel 6 sehingga pada simpul tersebut *count support* bertambah 1 (*count support* = 8). Selanjutnya kode golongan buku 1 dimasukkan ke dalam *FP-tree* maka simpul berlabel 6 ditelusuri simpul anaknya. Simpul 6 memiliki simpul anak berlabel 1 sehingga *count support* simpul berlabel 1 dari *root* simpul berlabel 6 ditambahkan 1 (*count support* = 2). Kode golongan buku 5 kemudian dimasukkan ke dalam *FP-tree* sehingga simpul 1 pada lintasan tersebut ditelusuri simpul

anaknya. Karena tidak ada simpul anak pada simpul berlabel 1 maka dibuat simpul anak baru berlabel 5 dengan *count support* = 1. Kemudian dilanjutkan dengan kode golongan buku 7 yang dimasukkan ke dalam *FP-tree* sehingga simpul 5 yang telah dibuat tersebut ditelusuri simpul anaknya. Karena tidak memiliki simpul anak maka dibuatkan simpul anak baru berlabel 7 dari simpul berlabel 5 dengan *count support* = 1.

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	
7	

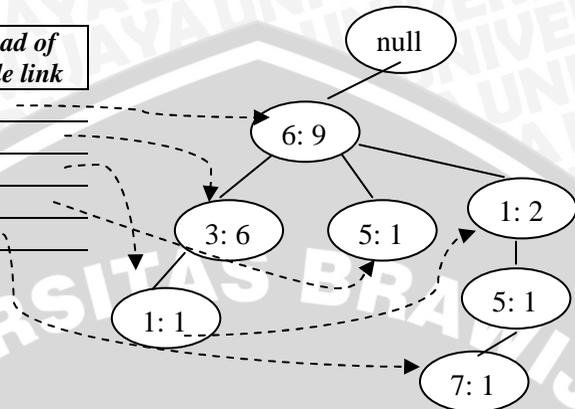


**Gambar 3.25** *FP-tree* Setelah Pembacaan Kode Transaksi 650MAR-10010700155

Pada Gambar 3.26, transaksi yang dibaca adalah kode transaksi 650RED-10010700115 dengan urutan kode golongan buku yang dibeli adalah 6 dan 3. *Prefix* dari transaksi ini yaitu kode golongan buku 6 sehingga ditimpakan pada simpul anak *root* yang berlabel 6 sehingga pada simpul tersebut *count support* bertambah 1 (*count support* = 9). Selanjutnya kode golongan buku 3 dimasukkan ke dalam *FP-tree*. Pada simpul berlabel 6 ditelusuri simpul anaknya. Karena simpul 6 memiliki simpul anak berlabel 3 maka *count support* simpul 3 ditambahkan 1 (*count support* = 6).

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	
7	

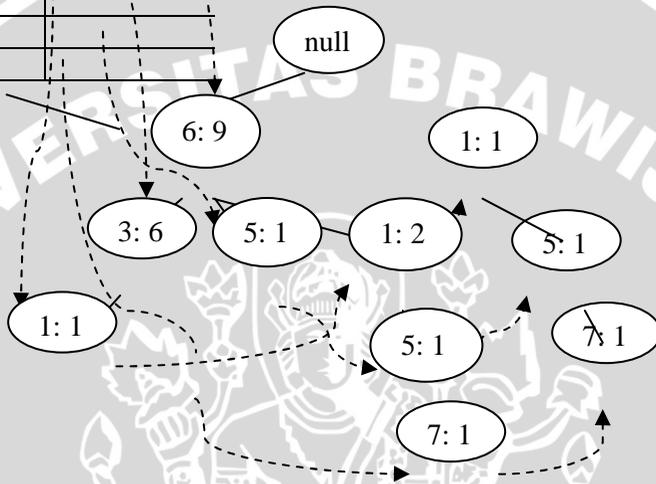


**Gambar 3.26** FP-tree Setelah Pembacaan Kode Transaksi 650RED-10010700115

Pada Gambar 3.27, transaksi yang dibaca adalah kode transaksi 650KMU-10010800026 dengan urutan kode golongan buku yang dibeli adalah 1, 5 dan 7. *Prefix* dari transaksi ini yaitu kode golongan buku 1. Pada *root* ditelusuri simpul anak berlabel 1. Karena simpul anak dari *root* hanya ada simpul anak berlabel 6 maka dibuat simpul anak berlabel 1 dengan *count support* = 1. Selanjutnya kode golongan buku 5 dimasukkan ke dalam FP-tree. Pada simpul berlabel 1 yang telah dibuat ditelusuri simpul anaknya. Karena simpul 1 tidak memiliki simpul anak maka dibuat simpul anak berlabel 5 dari simpul 1 dengan *count support* = 1. Kode golongan buku berikutnya adalah kode golongan buku 7. Pada simpul berlabel 5 dalam lintasan yang sama ditelusuri simpul anaknya. Karena simpul 5 tidak memiliki anak maka dibuat simpul anak berlabel 7 dari simpul 5 dengan *count support* = 1.

Header Table

kode golongan buku	Head of table link
6	
3	
1	
5	
7	



**Gambar 3.27** FP-tree Setelah Pembacaan Kode Transaksi 650KMU-10010800026

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.

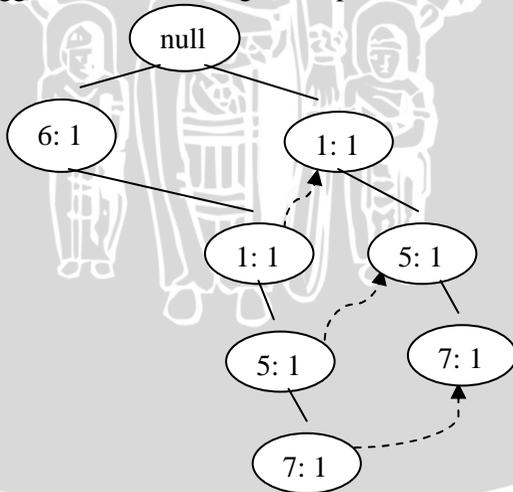
**Tabel 3.10** Conditional Pattern Base dari Sampel Data

Suffix	Conditional Pattern Base
{7}	{6 : 1, 1 : 1, 5 : 1}, {1 : 1, 5 : 1}
{5}	{6 : 1}, {1 : 1}, {6 : 1, 1 : 1}
{1}	{6 : 1, 3 : 1}, {6 : 2}
{3}	{6 : 6}
{6}	<i>null</i>

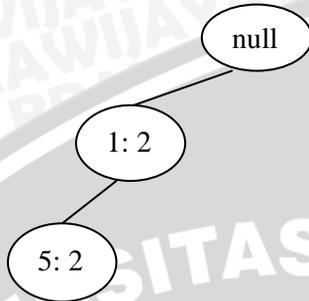
Langkah kedelapan adalah tahap *conditional FP-tree* yaitu *support count* setiap kode golongan buku dalam *prefix path* setiap *suffix* dijumlahkan. Kode golongan buku yang memiliki jumlah *support count* lebih besar atau sama dengan *minimum support* akan dibangkitkan dengan *conditional FP-tree*.

$$\begin{aligned}
 \text{Suffix } \{7\} : \quad & \text{Support } \{6\} &= & 1/10 &= & 0.1 \\
 & \text{Support } \{1\} &= & 2/10 &= & 0.2 \\
 & \text{Support } \{5\} &= & 2/10 &= & 0.2 \\
 \text{Suffix } \{5\} : \quad & \text{Support } \{6\} &= & 2/10 &= & 0.2 \\
 & \text{Support } \{1\} &= & 2/10 &= & 0.2 \\
 \text{Suffix } \{1\} : \quad & \text{Support } \{6\} &= & 2/10 &= & 0.2 \\
 & \text{Support } \{3\} &= & 1/10 &= & 0.1 \\
 \text{Suffix } \{3\} : \quad & \text{Support } \{6\} &= & 6/10 &= & 0.6
 \end{aligned}$$

Pada Gambar 3.28 menggambarkan lintasan yang memiliki *suffix* kode golongan buku 7, sedangkan *conditional FP-tree* yang dibangkitkan untuk *suffix* kode golongan buku 7 dapat dilihat pada Gambar 3.29. Kode golongan buku 6 memiliki *support* dibawah *minimum support* sehingga tidak ikut dibangkitkan pada *conditional FP-tree*.



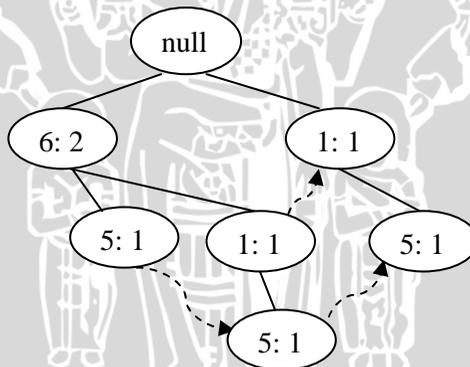
**Gambar 3.28** Lintasan yang Memiliki Suffix {7}



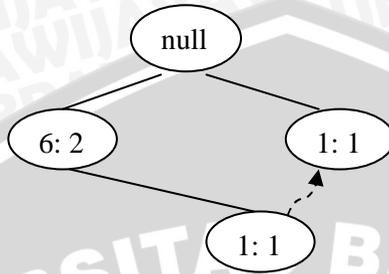
**Gambar 3.29** Conditional FP-tree untuk Suffix {7}

Berdasarkan Gambar 3.29, karena *path* yang terbentuk berupa *single path* maka *frequent itemset* yang didapat dari *suffix* kode golongan buku 7 adalah {1,7}, {5,7}, dan {1,5,7} dengan masing – masing frekuensi sebanyak 2.

Pada Gambar 3.30 menggambarkan lintasan yang memiliki *suffix* kode golongan buku 5. Conditional FP-tree yang dibangkitkan untuk *suffix* kode golongan buku 5 dapat dilihat pada Gambar 3.31.



**Gambar 3.30** Lintasan yang Memiliki Suffix {5}



**Gambar 3.31** Conditional FP-tree untuk Suffix {5}

Path yang terbentuk bukan *single path* maka dilakukan *generate conditional FP-tree* kembali dengan *suffix* kode golongan buku 1 dan 5 yang dapat dilihat pada Gambar 3.32.



**Gambar 3.32** Conditional FP-tree untuk Suffix {1,5}

Hasil *conditional FP-tree* untuk *suffix* {1,5} adalah *null*, karena *support* dari {6,1,5} kurang dari *minimum support*. Berdasarkan Gambar 3.31, *path* yang dihasilkan adalah *single path* sehingga *frequent itemset* yang didapat dari *suffix* {1,5} adalah {1,5} dengan frekuensi sebanyak 2.

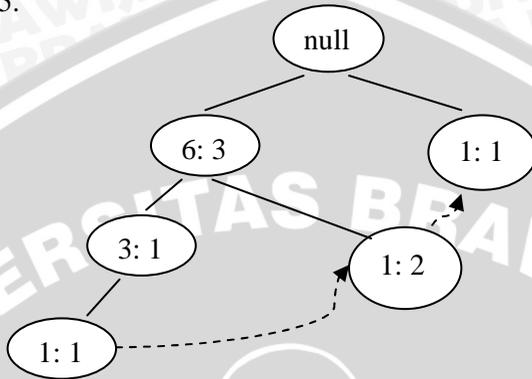
Kemudian dilakukan *generate conditional FP-tree* kembali dengan *suffix* kode golongan buku 6 dan 5 yang dapat dilihat pada Gambar 3.33.



**Gambar 3.33** Conditional FP-tree untuk Suffix {6,5}

Hasil *conditional FP-tree* untuk *suffix* {6,5} merupakan *single path*, sehingga *frequent itemset* yang didapat dari *suffix*{6,5} adalah {6,5} dengan frekuensi sebanyak 2.

Pada Gambar 3.34 menggambarkan lintasan yang memiliki *suffix* {1}. *Conditional FP-tree* untuk *suffix* {1} dapat dilihat pada Gambar 3.35.



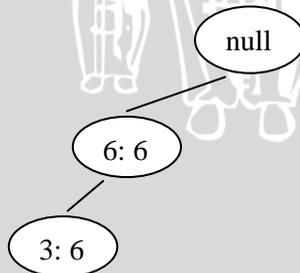
**Gambar 3.34** Lintasan yang Memiliki *Suffix* {1}



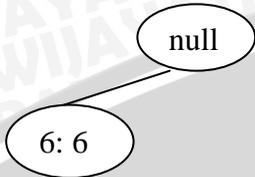
**Gambar 3.35** *Conditional FP-tree* untuk *Suffix* {1}

Berdasarkan Gambar 3.35, *path* dari *conditional FP-tree* merupakan *single path* sehingga *frequent itemset* yang didapat untuk *suffix* {1} adalah {6,1} dengan frekuensi sebanyak 3.

Pada Gambar 3.36 menggambarkan lintasan yang memiliki *suffix* {3}. *Conditional FP-tree* untuk *suffix* {3} dapat dilihat pada Gambar 3.37.



**Gambar 3.36** Lintasan yang Memiliki *Suffix* {3}



**Gambar 3.37** *Conditional FP-tree* untuk *Suffix* {3}

*Path* yang terbentuk pada Gambar 3.37 berupa *single path* maka *frequent itemset* yang didapatkan untuk *suffix* {3} adalah {6,3} dengan frekuensi sebanyak 6.

Gambar 3.38 menggambarkan lintasan yang memiliki *suffix* {6}, sedangkan Gambar 3.39 menggambarkan *conditional FP-tree* untuk *suffix* {6}.



**Gambar 3.38** Lintasan yang Memiliki *Suffix* {6}



**Gambar 3.39** *Conditional FP-tree* untuk *Suffix* {6}

Berdasarkan Gambar 3.39, *path* yang dihasilkan dari *conditional FP-tree* maka *frequent itemset* yang didapatkan tidak ada (*null*).

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>	<b>Frekuensi</b>	<b>Support</b>
5 → 7	2	0.2
1 → 7	2	0.2
1, 5 → 7	2	0.2
6 → 5	2	0.2
1 → 5	2	0.2
6 → 1	3	0.3
6 → 3	6	0.6

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*.

$$\begin{aligned}
 \text{Confidence } \{6\} \rightarrow \{3\} &= 0.6/0.9 = 0.67 \\
 \text{Confidence } \{6\} \rightarrow \{1\} &= 0.3/0.9 = 0.33 \\
 \text{Confidence } \{6\} \rightarrow \{5\} &= 0.2/0.9 = 0.22 \\
 \text{Confidence } \{1\} \rightarrow \{5\} &= 0.2/0.4 = 0.5 \\
 \text{Confidence } \{5\} \rightarrow \{7\} &= 0.2/0.3 = 0.67 \\
 \text{Confidence } \{1\} \rightarrow \{7\} &= 0.2/0.4 = 0.5 \\
 \text{Confidence } \{1,5\} \rightarrow \{7\} &= 0.2/0.2 = 1
 \end{aligned}$$

**Tabel 3.12** *Confidence Frequent Itemset* Sampel Data

<i>Frequent Itemset</i>	<b>Support (A,B)</b>	<b>Support (A)</b>	<b>Confidence</b>
6 → 3	0.6	0.9	0.67
6 → 1	0.3	0.9	0.33
6 → 5	0.2	0.9	0.22
1 → 5	0.2	0.4	0.5
5 → 7	0.2	0.3	0.67
1 → 7	0.2	0.4	0.5
1, 5 → 7	0.2	0.2	1

Berdasarkan Tabel 3.12, *Frequent itemset* yang terpilih untuk dibangkitkan menjadi *rule* adalah  $6 \rightarrow 3$ ,  $1 \rightarrow 5$ ,  $5 \rightarrow 7$ ,  $1 \rightarrow 7$ , dan  $1,5 \rightarrow 7$  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

<b>Rules</b>	<b>Keterangan</b>
$6 \rightarrow 3$	Jika membeli buku golongan Umum maka membeli buku golongan Media
$1 \rightarrow 5$	Jika membeli buku golongan Agama maka membeli buku golongan Bisnis
$5 \rightarrow 7$	Jika membeli buku golongan Bisnis maka membeli buku golongan Bahasa dan Sastra
$1 \rightarrow 7$	Jika membeli buku golongan Agama maka membeli buku golongan Bahasa dan Sastra
$1,5 \rightarrow 7$	Jika membeli buku golongan Agama dan Bisnis maka membeli buku golongan Bahasa dan Sastra

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

$$\begin{aligned}
 \text{Lift Ratio } \{6\} \rightarrow \{3\} &= \frac{0.67}{0.6} = 1.12 \\
 \text{Lift Ratio } \{1\} \rightarrow \{5\} &= \frac{0.5}{0.3} = 1.67 \\
 \text{Lift Ratio } \{5\} \rightarrow \{7\} &= \frac{0.67}{0.2} = 3.35 \\
 \text{Lift Ratio } \{1\} \rightarrow \{7\} &= \frac{0.5}{0.2} = 2.5 \\
 \text{Lift Ratio } \{1,5\} \rightarrow \{7\} &= \frac{1}{0.2} = 5
 \end{aligned}$$

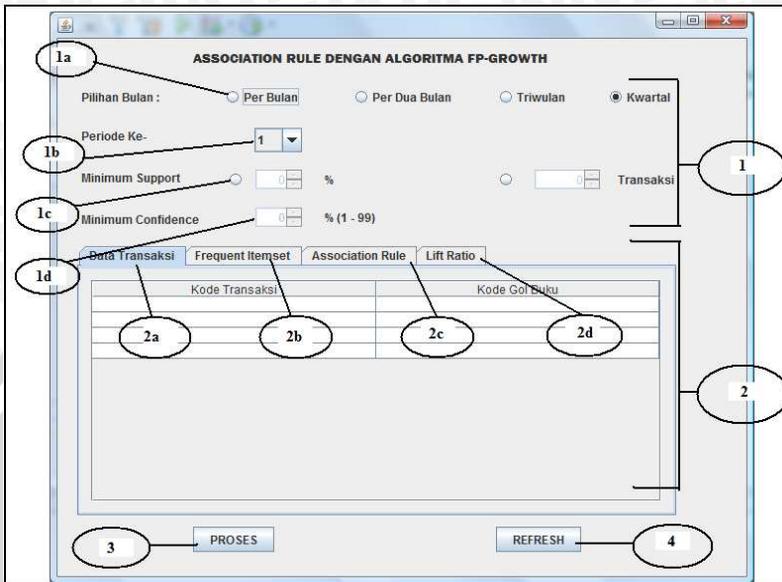
**Tabel 3.14** *Lift Ratio Rules*

<i>Rules</i>	<i>Confidence</i>	<i>Frekuensi Item Consequent</i>	<i>Benchmark Confidence</i>	<i>Lift Ratio</i>
6 → 3	0.67	6	0.6	1.12
1 → 5	0.5	3	0.3	1.67
5 → 7	0.67	2	0.2	3.35
1 → 7	0.5	2	0.2	2.5
1,5 → 7	1	2	0.2	5

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 1,5 → 7 yaitu sebesar 5, dan artinya *rule* 1,5 → 7 memiliki kekuatan asosiasi yang tinggi sehingga dapat menunjukkan adanya manfaat. Selain itu, dapat dilihat juga bahwa nilai *confidence* yang tinggi belum dapat menjadi acuan bahwa *rule* tersebut memiliki kekuatan asosiasi yang tinggi karena dengan *lift ratio*, peluang munculnya kode golongan buku tersebut bersamaan akan dibandingkan dengan peluang munculnya masing – masing kode golongan buku tersebut muncul secara independen. Sehingga apabila nilai *confidence* dari suatu *rule* tinggi sedangkan nilai peluang munculnya masing – masing kode golongan buku tersebut muncul secara independen juga tinggi maka kekuatan dari *rule* tersebut menjadi rendah.

### 3.4. Rancangan Antar Muka

Rancangan antar muka secara umum terdiri dari bagian *input*, bagian *output*, tombol proses dan tombol *refresh*. Bagian *input* terdiri dari tiga data *input*, yaitu data *input* pilihan bulan, data *input* periode bulan, 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 3.40.



**Gambar 3.40** Rancangan Antar Muka Secara Umum

Keterangan :

1. Bagian *Input*

- 1a. Data input pilihan bulan : pilihan bulan untuk pembacaan data transaksi yang akan digunakan. Pilihan yang diberikan terdiri dari per bulan, per dua bulan, triwulan, dan kwartal.
- 1b. Data input periode bulan : periode bulan data transaksi yang akan digunakan. Pilihan yang diberikan untuk periode bergantung dari pilihan bulan yang dipilih.
- 1c. Data input minimum *support* : untuk batas terendah nilai *support* yang harus dimiliki tiap golongan buku maupun kombinasinya. Input terdiri dari dua pilihan yaitu minimum *support* menggunakan prosentase dan minimum *support* menggunakan *support count*.
- 1d. Data input minimum *confidence* : untuk batas prosentase terendah probabilitas kemunculan secara bersama.

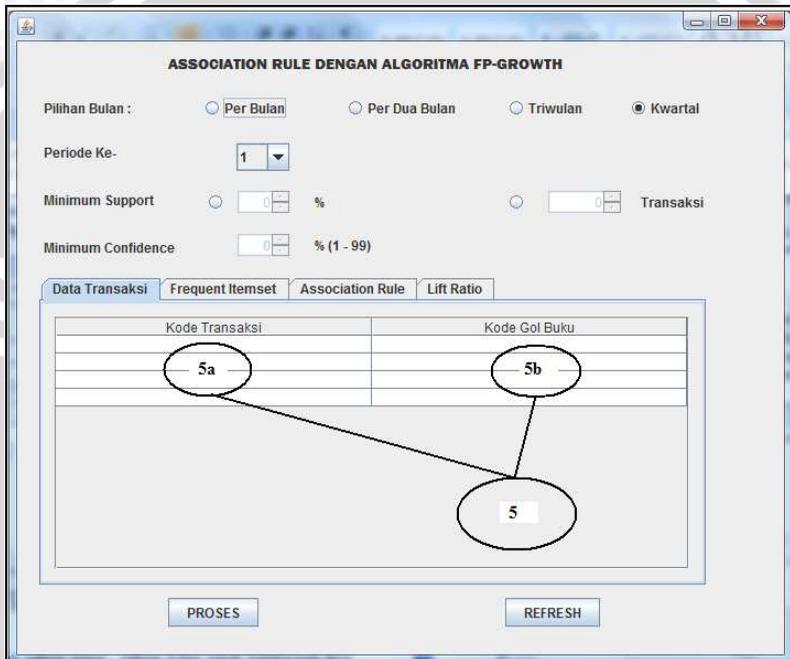
2. Bagian *Output*

- 2a. Submenu data transaksi : menampilkan data transaksi berdasarkan periode bulan yang dipilih.
- 2b. Submenu *frequent itemset* : menampilkan hasil *frequent 1-itemset* dan *frequent itemset*.
- 2c. Submenu *association rule* : menampilkan *rule* yang dihasilkan.
- 2d. Submenu *lift ratio* : menampilkan hasil perhitungan *lift ratio rule*.

3. Tombol *Proses*

4. Tombol *Refresh*

Pada bagian *output*, submenu data transaksi digunakan untuk menampilkan tabel data transaksi berdasarkan periode bulan yang dipilih. Tabel data terdiri dari kolom kode transaksi dan kolom kode golongan buku. Tampilan rancangan antar muka submenu data transaksi dapat dilihat pada Gambar 3.41.

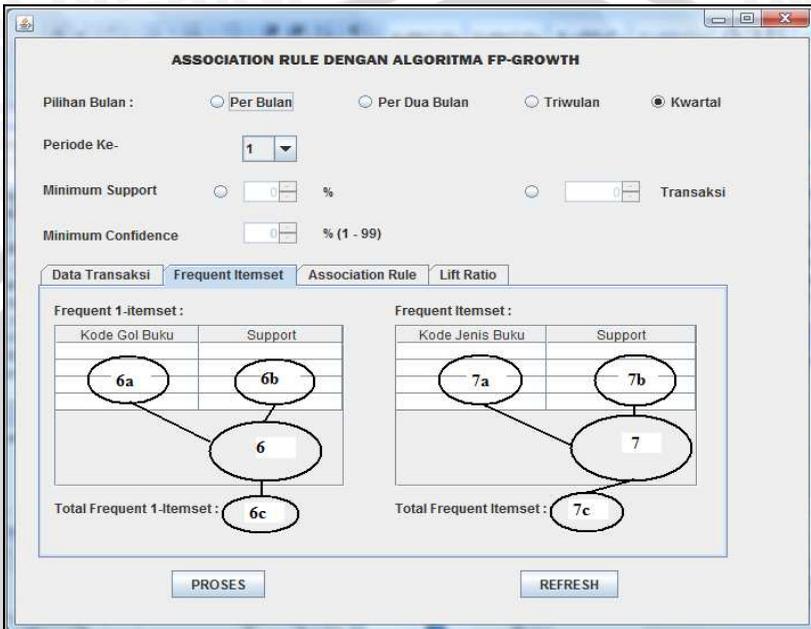


**Gambar 3.41** Rancangan Antar Muka Submenu Data Transaksi  
Keterangan :

5. Tabel data transaksi, terdiri dari kolom kode transaksi (5a) dan kolom kode golongan buku (5b).

Pada bagian *output*, submenu *frequent itemset* digunakan untuk menampilkan tabel *frequent 1-itemset* dan tabel *frequent itemset*. Tabel *frequent 1-itemset* merupakan tabel yang menampilkan data golongan buku yang nilai *support* secara independennya memenuhi *minimum support*. Tabel *frequent 1-itemset* terdiri dari kolom kode golongan buku dan *support*. Tabel *frequent itemset* merupakan tabel yang menampilkan data kombinasi golongan buku yang nilai *support* secara bersamanya memenuhi *minimum support*. Tabel *frequent itemset* terdiri dari kolom kode

golongan buku dan kolom *support*. Tampilan rancangan antar muka submenu *frequent itemset* dapat dilihat pada Gambar 3.42.

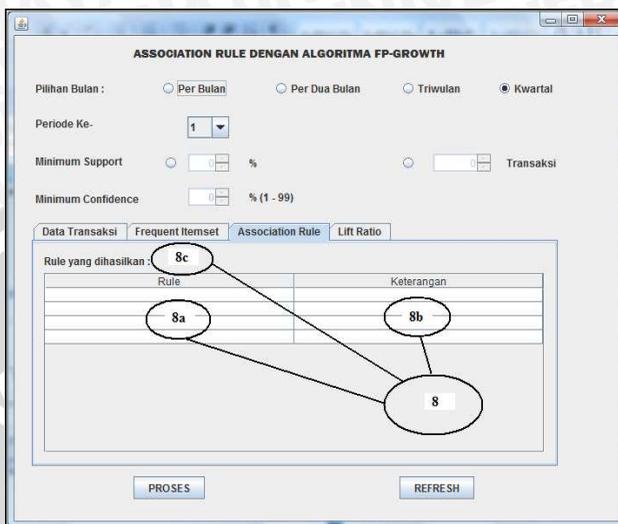


**Gambar 3.42** Rancangan Antar Muka Submenu *Frequent Itemset*

Keterangan :

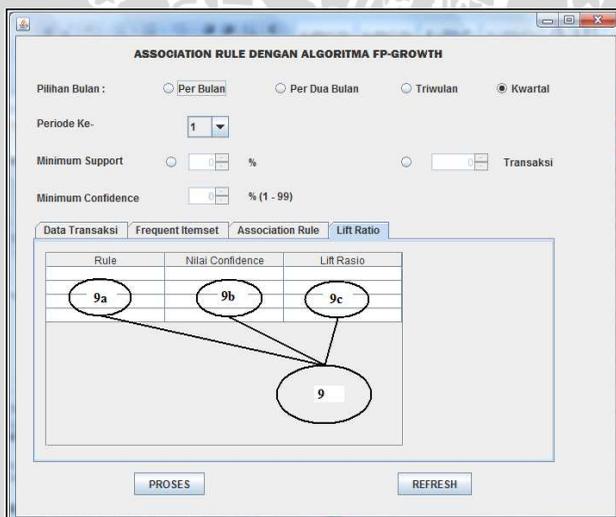
6. Tabel *frequent 1-itemset*, terdiri dari kolom kode golongan buku (6a) dan kolom *support* (6b) serta informasi jumlah *frequent 1-itemset* (6c).
7. Tabel *frequent itemset*, terdiri dari kolom kode golongan buku (7a) dan kolom *support* (7b) serta informasi jumlah *frequent itemset* (7c).

Pada bagian *output*, submenu *association rule* digunakan untuk menampilkan tabel *rule – rule* yang dihasilkan, yaitu *frequent itemset* yang nilai *confidence*-nya memenuhi *minimum confidence*. Tabel *rule* terdiri dari kolom *rule* dan kolom keterangan. Tampilan rancangan antar muka submenu *association rule* dapat dilihat pada Gambar 3.43. Pada bagian *output*, submenu *lift ratio* digunakan untuk menampilkan tabel hasil perhitungan *lift ratio* setiap *rule*. Tabel *lift ratio* terdiri dari kolom *rule*, kolom nilai *confidence* dan kolom *lift ratio*. Tampilan rancangan antar muka submenu *lift ratio* dapat dilihat pada Gambar 3.44.



**Gambar 3.43** Rancangan Antar Muka Submenu *Association Rule*  
Keterangan :

8. Tabel *rule*, terdiri dari kolom *rule* (8a) dan kolom *keterangan* (8b) serta informasi jumlah *rule* yang dihasilkan (8c).



**Gambar 3.44** Rancangan Antar Muka Submenu *Lift Ratio*

Keterangan :

9. Tabel *lift ratio*, terdiri dari kolom *rule* (9a), kolom *nilai confidence* (9b) dan kolom *lift ratio* (9c).

### 3.5. Rancangan Uji Coba

Uji coba sistem untuk analisa pola konsumsi buku ini akan melakukan evaluasi terhadap hasil analisa yang dihasilkan oleh sistem. Tujuan dari uji coba ini adalah untuk mengetahui pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan. Selain itu, uji coba ini juga bertujuan untuk mengetahui tingkat kekuatan dari *rule* yang dihasilkan.

Pengujian dilakukan pada data transaksi dengan aturan *preprocessing* yaitu memiliki kode golongan buku minimal 4. Uji pengaruh nilai *minimum support* dan nilai *minimum confidence* dilakukan pada sejumlah data baik dengan rentang bulan dan periode data yang sama maupun berbeda. Tabel uji untuk pengaruh nilai *minimum support* dan nilai *minimum confidence* dapat dilihat pada Tabel 3.15. Selanjutnya *rule* yang dihasilkan dari uji coba tersebut dilakukan uji kekuatan *rule* dengan menggunakan *lift ratio* pada Persamaan 2.6. Tabel uji untuk uji kekuatan *rule* dapat dilihat pada Tabel 3.16.

Rentang bulan data transaksi yang digunakan dalam uji pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan adalah per bulan, per dua bulan, triwulan, dan kwartal. Begitu juga rentang bulan data transaksi dalam uji kekuatan *rule* dengan menggunakan *lift ratio*. Nilai *minimum support* yang digunakan pada uji pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan adalah 10% sampai 40% tiap periode dengan nilai *minimum confidence* 40% sampai 60% tiap nilai *minimum support*.

**Tabel 3.15** Tabel Uji Pengaruh Nilai *Minimum Support* dan Nilai *Confidence* terhadap Jumlah *Rule* yang Dihasilkan

Rentang Bulan	Periode	<i>Minimum Support</i> (%)	<i>Minimum Confidence</i> (%)	Jumlah Rule

**Tabel 3.16** Tabel Uji *Lift Ratio Association Rule*

Rentang Bulan	Periode	Aturan (Rule)	Confidence	Lift Ratio

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



## 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* Intel® Core2 Duo 2.10GHz
2. Memori 2GB
3. *Harddisk* dengan kapasitas 320 GB
4. Monitor 14.0"
5. *Keyboard*
6. *Mouse*

#### 4.1.2. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut :

1. Sistem operasi *Microsoft Windows Vista™ Home Premium Service Pack 1*.
2. *Netbeans IDE 6.8* dan *JDK 1.6* dalam mengembangkan aplikasi untuk analisis pola penjualan buku.
3. *MySQL* dan *JDBC*

### 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 4 tahap, yaitu tahap *frequent 1-itemset*, tahap *frequent itemset*, tahap *generate rule*, dan tahap *lift ratio*. Implementasi program terdiri dari 18 *class* yang dijelaskan pada Tabel 4.1 beserta fungsinya.

**Tabel 4.1** Class Pada Implementasi Program dan Fungsinya

<b>Nama Class</b>	<b>Fungsi</b>
<i>Class</i> HapusRedundansi	Menghapus kode golongan buku yang sama pada kode transaksi yang sama untuk proses <i>preprocessing</i> . Terdiri dari <i>method test</i> dan <i>method main</i> .
<i>Class</i> Data	Membaca data transaksi, membaca data golongan buku, dan proses <i>frequent 1-itemset</i> . Terdiri dari <i>method stat</i> , <i>method connect</i> , <i>method Transaksi</i> , <i>method getTrans</i> , <i>method getFrek</i> , <i>method urutFrek</i> , <i>method jumlahTrans</i> , <i>method oneItemset</i> , <i>method jumlahOneItemset</i> , <i>method transMinsup</i> , <i>method getJB</i> , <i>method maxSupport</i> , <i>method itungSupport</i> .
<i>Class</i> Frek	Menyimpan data frekuensi semua kode golongan buku. Terdapat dari <i>constructor</i> Frek.
<i>Class</i> JenisBuku	Menyimpan data golongan buku. Terdapat dari <i>constructor</i> JenisBuku.
<i>Class</i> Support	Menyimpan data nilai <i>support</i> semua kode golongan buku. Terdiri dari <i>constructor</i> Support.
<i>Class</i> Trans	Menyimpan data transaksi berupa kode transaksi dan kode golongan buku yang dimiliki tiap kode transaksi. Terdiri dari <i>constructor</i> Trans
<i>Class</i> Temporary	Menyimpan data kode transaksi. Terdiri dari <i>constructor</i> Temporary
<i>Class</i> Path	Menyimpan data lintasan. Terdiri dari <i>constructor</i> Path.
<i>Class</i> Prefix	Menyimpan data <i>prefix</i> dari semua <i>suffix</i> . Terdapat dari <i>constructor</i> Prefix.
<i>Class</i> FrequentItemset	Menyimpan data <i>frequent itemset</i> . Terdiri dari <i>constructor</i> FrequentItemset.

Nama Class	Fungsi
Class Confidence	Menyimpan data nilai <i>confidence</i> . Terdiri dari <i>constructor</i> Confidence.
Class Rule	Menyimpan data <i>rule</i> . Terdiri dari <i>constructor</i> Rule.
Class LiftRatio	Menyimpan data nilai <i>lift ratio</i> . Terdiri dari <i>constructor</i> LiftRatio.
Class FPTreeNode	Menyimpan data informasi <i>node</i> pada <i>FP-Tree</i> . Terdiri dari <i>constructor</i> FPTreeNode, <i>method</i> displayNode, <i>method</i> NodeNext, <i>method</i> NodeSibling, <i>method</i> NodeParent.
Class FPTreeHeaderEntry	Sebagai <i>header table</i> yang menyimpan informasi total bobot tiap kode golongan buku yang menjadi <i>frequent 1-itemset</i> yang telah masuk ke dalam <i>FP-Tree</i> . Terdiri dari <i>constructor</i> FPTreeHeaderEntry, <i>method</i> DisplayHeaderEntry.
Class FPTree	Implementasi proses untuk tahap <i>frequent itemset</i> , proses untuk tahap koleksi <i>rule</i> , dan proses untuk tahap <i>lift ratio</i> . Terdiri dari <i>constructor</i> FPTree <i>method</i> insert, <i>method</i> cari, <i>method</i> simpanPath, <i>method</i> cariPrefix, <i>method</i> SupportPref, <i>method</i> ConditionalFPTree, <i>method</i> urutFI, <i>method</i> cekSupport, <i>method</i> FreqIt, <i>method</i> jumlahFrequentItemset, <i>method</i> ConfidenceFI, <i>method</i> JumlahRule, <i>method</i> RuleFI, <i>method</i> hitungLift, <i>method</i> LiftRatio, <i>method</i> RuleConf,
Class Interface	Main program dan mengatur tampilan GUI. Terdiri dari <i>constructor</i> Interface, <i>method</i> tabelData, <i>method</i> oneItemset, <i>method</i> FrequentItemset, <i>method</i> Rule,

Nama Class	Fungsi
Class Interface	<i>method</i> LiftRatio, <i>method</i> jComboBox1ActionPerformed, <i>method</i> refreshFreqOne, <i>method</i> refreshFreqItem, <i>method</i> refreshRule, <i>method</i> refreshLift, <i>method</i> jButton2ActionPerformed, <i>method</i> jButton1ActionPerformed, <i>method</i> jSpinner1StateChanged, <i>method</i> jSpinner2StateChanged, <i>method</i> jSpinner3StateChanged, <i>method</i> jRadioButton1MouseClicked, <i>method</i> jRadioButton2MouseClicked, <i>method</i> jRadioButton3MouseClicked, <i>method</i> jRadioButton4MouseClicked, <i>method</i> jRadioButton5MouseClicked, <i>method</i> jRadioButton6MouseClicked, <i>method</i> main.
Class ExportData	Mengekspor data dari <i>jTable</i> ke <i>excel</i> . Terdiri dari <i>method</i> exportTable.

#### 4.2.1. Implementasi Tahap *Frequent 1-itemset*

Pada tahap ini, dilakukan pengkoleksian tiap golongan buku yang memiliki frekuensi kemunculannya pada data transaksi dalam periode waktu tertentu lebih besar atau sama dengan minimum *support*. Tahap *frequent 1-itemset* terdiri dari 4 proses, yaitu proses pembacaan data transaksi, proses pengkoleksian golongan buku dan frekuensinya, proses penyortiran kode golongan buku, proses pencarian kode golongan buku yang *frequent*, dan proses pengkoleksian transaksi dengan kode golongan buku yang *frequent*.

##### 4.2.1.1. Implementasi Pembacaan Data Transaksi

Proses awal yang dilakukan adalah pembacaan data transaksi dalam periode waktu tertentu dan *load* data transaksi yang memiliki pembelian golongan buku lebih dari 3. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.1.

```

Temporary Transaksi(int periode, int bulan){
    Temporary Save = new Temporary();
    String[] siBulan = {"01","02","03","04","05","06","07"
                      ,"08","09","10","11","12"};
    int[] mulai = {0,1,2,3,4,5,6,7,8,9,10,11};
    bulan -= 1;
    int start = periode*bulan;
    int selesai = start + periode;
    try{
        String sql = "";
        for(int i=mulai[start];i<selesai;i++){
            if(i==mulai[start]){
                sql = "SELECT * FROM trans WHERE (kd_trans
                    LIKE '65%-10"+siBulan[i]+"%';
            }else{
                sql = sql+" OR kd_trans LIKE
                    '65%-10"+siBulan[i]+"%';
            }
        }
        sql = sql+")"+"GROUP BY kd_trans
            HAVING count(kd_trans)>3";
        Save.kd_trans = stat(sql,"kd_trans");
    }catch (Exception e){
        e.printStackTrace();
    }
    Save = new Temporary(Save.kd_trans);
    return Save;
}

Trans Transaksi(Temporary temp){
    Trans T = new Trans();
    Object[] siKode = temp.kd_trans.toArray();
    String sql="";
    try{
        sql = "SELECT * FROM trans WHERE
            kd_trans ='"+siKode[0]+"'";
        for (int i=1; i<siKode.length;i++){
            sql = sql+" OR kd_trans ='"+siKode[i]+"'";
        }
        T.kd_trans = stat(sql,"kd_trans");
        T.kd_jb = stat(sql,"kd_jb");
    }catch(Exception e){
        e.printStackTrace();
    }
    T = new Trans(T.kd_trans, T.kd_jb);
    return T;
}

```

**Source Code 4.1** Proses Pembacaan Data Transaksi

Fungsi Temporary Transaksi berfungsi untuk mengkoleksi kode transaksi dari tabel trans pada *database*, yang memiliki golongan buku minimal 4 golongan buku pada rentang bulan dan periode yang telah ditentukan. Kode transaksi hasil koleksi disimpan

dalam `ArrayList<String> kd_trans` pada *class* `Temporary` yang dideklarasikan dengan nama *object* `Save`.

Fungsi dari `Trans Transaksi` berfungsi untuk mengkoleksi informasi kode golongan buku dari kode transaksi hasil koleksi fungsi `Temporary Transaksi` yang menjadi parameter inputan dari fungsi ini yaitu `Temporary temp`. Hasil koleksi berupa kode transaksi dan kode golongan buku. Kode transaksi disimpan dalam `ArrayList<String> kd_trans` sedangkan kode golongan buku disimpan dalam `ArrayList<Integer> kd_jb` pada *class* `Trans` yang dideklarasikan dengan nama *object* `T`.

Pada kedua fungsi tersebut dilakukan pemanggilan fungsi `stat` yang berfungsi untuk mengeksekusi perintah SQL. Parameter dari fungsi `stat` adalah perintah SQL bertipe *String* dan nama kolom yang ingin diambil dari hasil eksekusi perintah SQL.

#### 4.2.1.2. Implementasi Pengkoleksian Golongan Buku dan Frekuensinya

Proses selanjutnya adalah mengkoleksi kode golongan buku yang ada pada data transaksi dan menghitung frekuensi kemunculan tiap – tiap kode golongan buku. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.2.

Pada fungsi `getFrek`, parameter inputan adalah data transaksi `Trans T`. Proses pada fungsi ini dilakukan dengan pembacaan setiap data transaksi dan diambil kode golongan bukunya pada index data transaksi yang sedang dibaca dengan perintah `T.kd_jb.get(i)`. Setiap index data transaksi, dilakukan pengecekan pada *object* `F`, yang merupakan deklarasi dari *class* `Frek`. Apabila kode golongan buku belum ada di `ArrayList<String> kodeBukunya` pada *object* `F` maka kode golongan buku tersebut disimpan dalam `ArrayList<String> kodeBukunya` dengan perintah `F.kodeBukunya.add` dan `ArrayList<Integer> jumlahBukunya` yang merupakan nilai frekuensi golongan buku tersebut diberi nilai 1 dengan perintah `F.jumlahBukunya.add(1)`

Kode golongan buku yang sudah ada pada `F.kodeBukunya` maka index dari `F.kodeBukunya` yang mengandung kode golongan buku tersebut disimpan dalam variabel `pos`. Kemudian frekuensi golongan buku tersebut diambil pada `F.jumlahBukunya` dengan index `pos` dan disimpan pada `temp`. Nilai frekuensi pada `F.jumlahBukunya` index `pos` dihapus kemudian pada index `pos` tersebut ditambahkan nilai frekuensi baru yaitu `temp+1`.

```

Frek getFrek(Trans T){
    Frek F = new Frek();
    for(int i =0; i<T.kd_trans.size();i++){
        if (F.kodeBukunya.contains
            (String.valueOf(T.kd_jb.get(i)))){
            int pos = F.kodeBukunya.indexOf(T.kd_jb.get(i));
            int temp = F.jumlahBukunya.get(pos);
            F.jumlahBukunya.remove(pos);
            F.jumlahBukunya.add(pos , temp+1);
        }else{
            F.kodeBukunya.add(String.valueOf(T.kd_jb.get(i)));
            F.jumlahBukunya.add(1);
        }
    }
    return F;
}

```

**Source Code 4.2** Proses Pengkoleksian Golongan Buku dan Frekuensinya

#### 4.2.1.3. Implementasi Penyortiran Kode Golongan Buku

Pada proses ini, dilakukan sortir kode golongan buku berdasarkan nilai frekuensinya dari yang tertinggi ke terendah pada data `Frek`. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.3. Proses dilakukan dengan mengkonversi `kodeBukunya` dan `jumlahBukunya` dari *class* `Frek` ke dalam bentuk *array*. Kode golongan buku disimpan dalam `Object[] KB` sedangkan nilai frekuensi disimpan dalam `Object[] JB`. Kemudian dilakukan pengecekan nilai frekuensi untuk kode golongan buku pada index `i` terhadap nilai frekuensi kode golongan buku pada index `i+1` sampai banyaknya data pada `KB`.

Ketika kondisi pengecekan tersebut, apabila nilai frekuensi kode golongan buku pada index `i` lebih kecil daripada nilai frekuensi kode golongan buku pada index `i+1` maka nilai frekuensi kode golongan buku pada index `i` disimpan sementara pada `Object temp` sedangkan nilai frekuensi kode golongan buku pada index `i+1` disimpan ke `JB` index ke `i`. Begitu juga untuk kode golongan buku pada index `i` dan index `i+1`.

Setelah pengecekan dilakukan terhadap semua index data `Frek`, data kode golongan buku dan nilai frekuensi kembali disimpan ke dalam *arraylist*, yaitu `ArrayList<String> k` untuk kode golongan buku yang telah tersortir dan `ArrayList<Integer> j` untuk nilai frekuensi yang telah tersortir, pada *class* `Frek` dengan nama *object* `hasil` yang merupakan nilai pengembalian dari fungsi `urutFrek`.

```

Frek urutFrek(Frek asli){
    Object[] KB = asli.kodeBukunya.toArray();
    Object[] JB = asli.jumlahBukunya.toArray();
    Object temp = JB[0];
    for (int i =0;i<KB.length;i++){
        for(int j=i+1;j<KB.length;j++){
            if ((Integer) JB[i]<(Integer)JB[j]){
                temp = JB[i];
                JB[i] = JB[j];
                JB[j] = temp;
                temp = KB[i];
                KB[i] = KB[j];
                KB[j] = temp;
            }
        }
    }
    ArrayList<String> k = new ArrayList<String>();
    ArrayList<Integer> j = new ArrayList<Integer>();

    for (int i =0;i<KB.length ;i++){
        k.add((String) KB[i]);
        j.add((Integer)JB[i]);
    }
    Frek hasil = new Frek(k, j);
    return hasil;
}

```

**Source Code 4.3** Proses Penyortiran Kode Golongan Buku

#### 4.2.1.4. Implementasi Pencarian Kode Golongan Buku yang *Frequent*

Pada proses ini, dilakukan perhitungan nilai *support* tiap kode golongan buku dari nilai frekuensi kemunculannya dibagi dengan total transaksi. Kemudian nilai *support* tiap kode golongan buku dibandingkan dengan nilai minimum *support* yang telah ditentukan. Golongan buku yang memiliki nilai *support* lebih dari atau sama dengan minimum *support* yang akan menjadi *frequent 1-itemset*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.4.

```

Support itungSupport(Frek asli, int bulan)
{
    Support S = new Support();
    double jml = jumlahTrans(bulan);
    for(int i=0;i<asli.kodeBukunya.size();i++){
        S.bobot.add(i, (double)
            (asli.jumlahBukunya.get(i) / jml));
    }
    S.kodeBukunya = asli.kodeBukunya;
}

```

```

    return S;
}
Support oneItemset(Support s, double persen){
    Object[] sup = s.bobot.toArray();
    Object[] kd = s.kodeBukunya.toArray();
    double minsup = persen/100;
    ArrayList<String> k = new ArrayList<String>();
    ArrayList<Double> j = new ArrayList<Double>();
    for(int i=0;i<kd.length;i++)
    {
        if((Double)sup[i]>=minsup)
        {
            k.add((String)kd[i]);
            j.add((Double)sup[i]);
        }
    }
    Support hasil = new Support(k,j);
    return hasil;
}

```

**Source Code 4.4** Proses Pencarian Kode Golongan Buku yang *Frequent*

Pada fungsi `itungSupport` dilakukan proses perhitungan *support* dari data frekuensi yang telah tersortir `Frek asli`. Untuk menghitung nilai *support* dibutuhkan total transaksi sehingga dilakukan pemanggilan fungsi `jumlahTrans(bulan)` yang disimpan dalam variabel `jm1`. Pembacaan data `Frek asli` dilakukan setiap index data yang ada. Setiap index selanjutnya dilakukan perhitungan nilai frekuensinya `asli.jumlahBukunya.get(i)` dibagi dengan total transaksi `jm1`. Hasil dari perhitungan nilai *support* tersebut disimpan dalam `ArrayList<Double> bobot` sedangkan golongan buku dari `asli.kodeBukunya` disimpan ke dalam `ArrayList<String> kodeBukunya` pada class `Support` yang dideklarasikan dengan nama *object* `s`. Fungsi `oneItemset` berfungsi untuk mengkoleksi kode golongan buku yang memiliki nilai *support* lebih dari atau sama dengan nilai minimum *support*. Data nilai *support* dan data kode golongan buku yang merupakan *arraylist* pada *object class* `Support s` dikonversi ke bentuk *array*. Nilai minimum *support* dihitung terlebih dahulu karena parameter inputan `double persen` merupakan minimum *support* yang diberikan *user* dalam bentuk prosentase. Kode golongan buku yang memiliki nilai *support* lebih dari atau sama dengan minimum *support* `minsup`, disimpan ke dalam `ArrayList<String> k` untuk kode golongan bukunya dan ke dalam `ArrayList<Double> j` untuk nilai *support* kode golongan tersebut. Selanjutnya `ArrayList<String> k` dan `ArrayList<Double> j` disimpan ke dalam class `Support` dengan nama *object* `hasil`.

#### 4.2.1.5. Implementasi Pengkoleksian Transaksi dengan Kode Golongan Buku yang *Frequent*

Pada proses ini, dilakukan pengkoleksian kode transaksi dengan kode golongan buku yang *frequent*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.5.

```
HashMap transMinsup(Support S, Trans T){
    ArrayList<String> kd_trans = new ArrayList<String>();
    ArrayList<Integer> kd_jb;
    HashMap<String, ArrayList<Integer>> H =
        new HashMap<String, ArrayList<Integer>>();
    for(int i=0;i<S.kodeBukunya.size();i++){
        kd_trans = getTrans(T, S.kodeBukunya.get(i));
        for(int j = 0 ; j<kd_trans.size();j++){
            kd_jb = new ArrayList<Integer>();

            if(!H.containsKey(String.valueOf(kd_trans.get(j)))){
                kd_jb.add(Integer.parseInt(S.kodeBukunya.get(i)));
                H.put(kd_trans.get(j), kd_jb);
            }else{
                H.get(kd_trans.get(j)).add(Integer.parseInt
                    (S.kodeBukunya.get(i)));
            }
        }
    }
    return H;
}
```

**Source Code 4.5** Proses Pengkoleksian Transaksi dengan Kode Golongan Buku yang *Frequent*

Dalam fungsi `transMinsup` proses dilakukan dengan tahap setiap kode golongan buku yang lolos minimum *support* yaitu yang disimpan dalam `S.kodeBukunya` dicari kode transaksi pada `Trans T` yang memiliki kode golongan buku tersebut dengan memanggil fungsi `getTrans(T, S.kodeBukunya.get(i))` dimana parameter input dari fungsi ini adalah data transaksi `Trans` dan kode golongan buku yang ingin dicari `String kd_jb`. Nilai pengembalian dari fungsi `getTrans` adalah `ArrayList<String>` yang pada proses ini disimpan dalam variabel `kd_trans`.

Kemudian dilakukan pengecekan pada `HashMap H`, yang merupakan struktur data untuk menyimpan kode transaksi bertipe `String` sebagai *key* dan kode golongan buku yang *frequent* yang dimiliki kode transaksi tersebut bertipe `ArrayList<Integer>` sebagai *value*, untuk setiap index pada `kd_trans`. Apabila tidak ada *key* pada `HashMap H` yang berisi `kd_trans` pada index `j` maka kode golongan

buku index  $i$  pada  $S.kodeBukunya$  disimpan terlebih dahulu ke dalam `ArrayList<Integer> kd_jb` kemudian pada `HashMap H` dilakukan penyimpanan dimana *key* adalah `kd_trans` pada index  $j$  dan *value* adalah `kd_jb`. Sedangkan, apabila pada `HashMap H` terdapat *key* yang berisi `kd_trans` pada index  $j$  maka data pada `HashMap H` dengan *key* `kd_trans.get(j)` diambil untuk ditambahkan nilai *value*-nya dengan kode golongan buku index  $i$  pada  $S.kodeBukunya$ .

#### 4.2.2. Implementasi Tahap *Frequent Itemset*

Pada tahap ini, dilakukan kombinasi kode golongan buku 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 4 proses, yaitu proses pembentukan *FP-tree*, proses *conditional pattern base*, proses *conditional FP-tree*, dan proses koleksi *frequent itemset*.

##### 4.2.2.1. Implementasi Pembentukan *FP-tree*

Proses awal yang dilakukan adalah pembentukan *FP-tree*. Proses ini bertujuan untuk menyimpan data transaksi sehingga tidak perlu dilakukan pembacaan data transaksi berulang kali. Pada proses ini kode golongan buku setiap transaksi dimasukkan ke dalam *tree* sebagai satu lintasan. Node dari *FP-tree* menyimpan informasi kode golongan buku, bobot, kedalaman, indeks pada *header table*, indeks node, node *parent*, node *sibling*, dan node *link*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.6.

Pembentukan *FP-tree* diimplementasikan pada *class FPTree* yaitu pada fungsi *insert*. Pada fungsi ini, parameter inputan berupa `ArrayList<Integer> items` yang merupakan semua kode golongan buku yang dimiliki pada suatu kode transaksi dan `int count` yang merupakan jumlah bobot kode golongan buku. Proses awal adalah inisialisasi *node root*. Kemudian dilakukan pembacaan tiap index data kode golongan buku pada `items`, yang berarti kode golongan buku pada index tersebut dimasukkan ke dalam struktur *FP-tree*.

```
void insert(ArrayList<Integer> items,int count){
    FPTreeNode current_node = root;
    for (int index=0; index<items.size();
        index++){
        int entry_index=
```

```

((Integer)item2index.get(new
Integer(items.get(index))).intValue());
header[entry_index].count += count;
FPTreeNode walker = current_node.child;
for(;walker!=null;walker= walker.sibling)
    if (walker.item == 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;
}else{
    walker.count += count;
    current_node = walker;}
}
}

```

**Source Code 4.6** Proses Pembentukan *FP-tree*

Kode golongan buku yang dimasukkan pada struktur *FP-tree*, diambil indexnya pada *item2index* yang menyimpan index kode golongan buku pada *header table*. *Header table* merupakan tabel informasi mengenai node kode golongan buku yang sama yaitu yang menyimpan index node tiap kode golongan buku, kode golongan bukunya, total frekuensi yang telah masuk ke dalam *FP-tree* tiap kode golongan buku, dan *head node* dari tiap kode golongan buku.

Setelah index dari kode golongan buku didapatkan, maka pada *header table* data pada index tersebut ditambahkan 1 nilai *count*-nya. Proses selanjutnya adalah inisialisasi *node* pointer dengan nama *walker* dimana *walker* merupakan anak dari *node* sekarang. Selama *node* sekarang memiliki anak maka pointer akan menelusuri semua saudara dari anak *node* sekarang. Selama penelusuran, dilakukan pengecekan kode golongan buku pada *node* tersebut sama atau tidak dengan kode golongan buku yang dimasukkan ke dalam *FP-tree*. Jika ada yang sama maka posisi pointer berada pada *node* tersebut, sedangkan bila tidak ada yang sama berarti belum pernah terbentuk *node* dengan kode golongan buku yang dimasukkan ke dalam *FP-tree* yang menjadi anak dari *node* sekarang atau *walker* bernilai *null*.

Jika `walker` bernilai `null` maka diciptakan *node* baru yang dideklarasikan dengan nama `new_node` yang menjadi anak dari *node* sekarang dan *head node* pada *header table* untuk kode golongan buku tersebut. Kemudian *node* sekarang adalah *node* baru. Sedangkan, untuk `walker` yang tidak bernilai `null` maka bobot pada *node* yang dirujuk oleh pointer `walker` ditambahkan 1 dan *node* sekarang adalah *node* yang dirujuk oleh pointer `walker`.

#### 4.2.2.2. Implementasi Conditional Pattern Base

Pada proses ini, dilakukan pengkoleksian lintasan yang dimiliki tiap *suffix*, dimana *suffix* merupakan kode golongan buku 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 kode golongan buku yang menjadi *prefix* pada tiap *suffix*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.7.

Fungsi `cariPrefix` memiliki parameter inputan kode golongan buku yang menjadi *suffix* `int suffix` dan data lintasan `ArrayList<Path> p`. Data lintasan yang mengandung *suffix* akan disimpan ke dalam `ArrayList itemPref` dan bobot *prefix* pada lintasan tersebut disimpan dalam `ArrayList bobotPref`. Kode golongan buku yang menjadi *suffix* juga disimpan dalam `ArrayList itemSuf`.

```
Prefix cariPrefix(int suffix, ArrayList<Path> p){
    Prefix hasil = new Prefix();
    ArrayList itemPref = new ArrayList<Integer>();
    ArrayList itemSuf = new ArrayList<Integer>();
    ArrayList bobotPref = new ArrayList<Integer>();
    for(int i=0;i<p.size();i++){
        itemSuf.add(suffix);
        if((p.get(i).kd_jb.get
            (p.get(i).kd_jb.size()-1))==suffix){
            for(int j=0;j<p.get(i).kd_jb.size()-1;j++){
                itemPref.add(p.get(i).kd_jb.get(j));
                bobotPref.add(p.get(i).bobot.get(j));
            }
            hasil=new Prefix(itemPref,itemSuf,bobotPref);
        }
    }return hasil;
}
```

**Source Code 4.7** Proses Conditional Pattern Base

### 4.2.2.3. Implementasi *Conditional FP-tree*

Proses ini bertujuan untuk mengetahui nilai frekuensi tiap lintasan *prefix* tiap *suffix*. Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.8.

```
FrequentItemset ConditionalFPTree
(ArrayList<Integer> items, ArrayList<Path> pa){
    int nilai;
    double bobot = 0;
    ArrayList<Integer> fit = new ArrayList<Integer>();
    ArrayList<Double> fitBo = new ArrayList<Double>();
    FrequentItemset hasil = new FrequentItemset();
    for(int i=0;i<pa.size();i++){
        nilai = 0;
        if(pa.get(i).kd_jb.get
            (pa.get(i).kd_jb.size()-1)==
            items.get(items.size()-1)){
            for(int x=0;x<items.size()-1;x++){
                if(pa.get(i).kd_jb.contains(items.get(x))){
                    nilai+=1;}
            }
            if(nilai == items.size()-1){
                bobot=bobot+pa.get(i).bobot.get(0);}
        }
    }
    bobot = (bobot / total);
    for(int j = 0; j<items.size();j++){
        fit.add(items.get(j));
        fitBo.add(bobot);}
    hasil = new FrequentItemset(fit, fitBo);
    return hasil;}
```

**Source Code 4.8** Proses *Conditional FP-tree*

Fungsi `ConditionalFPTree` memiliki parameter inputan berupa kombinasi kode golongan buku yang menjadi *frequent 1-itemset* 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 kode golongan buku yang sama sejumlah kode golongan buku pada data kombinasi pada `items`. Setiap data lintasan yang sesuai maka nilai bobotnya dijumlahkan. Setelah pengecekan dilakukan pada semua data lintasan, maka dihitung nilai *support* kombinasi kode golongan buku tersebut dengan cara total bobot untuk kombinasi tersebut yang disimpan dalam variabel `bobot` dibagi dengan jumlah transaksi yang disimpan dalam variabel `total`. Kemudian kombinasi kode golongan buku

disimpan dalam `ArrayList<Integer>` `fit` dan nilai *support* kombinasi kode golongan tersebut disimpan dalam `ArrayList<Double>` `fitBo`.

#### 4.2.2.4. 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* 4.9.

Fungsi `cekSupport` 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 menghapus dari data *frequent itemset* yang dideklarasikan `ArrayList<FrequentItemset>` hasil untuk kandidat *frequent itemset* yang memiliki nilai *support* kurang dari minimum *support*.

```
ArrayList<FrequentItemset>
cekSupport(ArrayList<FrequentItemset> hasil){
    for(int i=0;i<hasil.size();i++){
        if((double)hasil.get(i).bobot.get(0)<minsip){
            hasil.remove(hasil.get(i));
            i=i-1;
        }
    }return hasil;
}
```

**Source Code 4.9** 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 hitung *confidence* dan proses koleksi *rule*.

##### 4.2.3.1. Implementasi Hitung *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* 4.10. 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* sebagai variabel `s`, dan nilai *minimum confidence* dalam bentuk prosentase sebagai variabel `min`.

Setiap data *frequent itemset* diambil *prefix*-nya yang disimpan sementara pada `ArrayList<Integer>` `titip`. Jika *prefix* dari index data *frequent itemset* tersebut terdiri lebih dari satu kode golongan buku maka untuk mendapatkan nilai *support* kode golongan buku yang menjadi *antecedent* (bagian jika) dilakukan pencarian terhadap data *frequent itemset* `hasil` yang terdiri dari kode golongan buku dalam `titip`. Hasil pencarian tersebut berupa nilai *support* yang disimpan dalam `supAntecedent` dan untuk selanjutnya digunakan untuk perhitungan nilai *confidence* yaitu sebagai pembagi dari nilai bobot *frequent itemset*.

Sedangkan untuk *prefix* index data *frequent itemset* yang hanya terdiri dari satu kode golongan buku, nilai *support* kode golongan buku tersebut didapatkan dari pembacaan data *support frequent 1-itemset*. *Frequent itemset* yang memiliki nilai *confidence* lebih dari atau sama dengan *minimum confidence* akan disimpan data *frequent itemset*-nya dalam `ArrayList<ArrayList<Integer>>` `f` dan nilai *confidence frequent itemset* tersebut disimpan dalam `ArrayList<Double>` `co`. Data yang disimpan sementara pada `titip` selanjutnya dihapus semuanya karena akan digunakan untuk menyimpan data *prefix* index data *frequent itemset* selanjutnya.

```
Confidence ConfidenceFI(ArrayList<FrequentItemset>
hasil, Support S, double min){
    ArrayList<Integer> titip = new ArrayList<Integer>();
    ArrayList<ArrayList<Integer>> f=
    new ArrayList<ArrayList<Integer>>();
    ArrayList<Double> co = new ArrayList<Double>();
    Confidence hasilCo = new Confidence();
    double confidence= 0;
    double supAntecedent = 0;
    minconf = (min/100);
    for(int i=0;i<hasil.size();i++){
        for(int x=0;x<hasil.get(i).prefix.size()-1;x++)
        {
            titip.add(hasil.get(i).prefix.get(x));
        }
    }
    if(titip.size(>)1){
```

```

for(int a=0;a<titip.size();a++)
{
    for(int y=0;y<hasil.size();y++){
        if(hasil.get(y).prefix.equals(titip))
        {
            supAntecedent=hasil.get(y).bobot.get(0);
        }
    }
}
confidence=
(hasil.get(i).bobot.get(0)/supAntecedent);
}else{
for(int y=0;y<S.kodeBukunya.size();y++){
    if((Integer.parseInt
(S.kodeBukunya.get(y))==(titip.get(0))){
        confidence=(hasil.get(i).bobot.get(0)
/S.bobot.get(0));
    }
}
if(confidence >= minconf){
    f.add(hasil.get(i).prefix);
    co.add(confidence);
    titip.removeAll(titip);
    hasilCo = new Confidence(f,co);
}return hasilCo;
}

```

**Source Code 4.10** Proses Hitung Confidence

#### 4.2.3.2. Implementasi Koleksi 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* 4.11.

Fungsi `RuleFI` terdiri dari parameter inputan data golongan buku yaitu `JenisBuku JB` dan data *frequent itemset* yang lolos nilai *confidence*-nya yaitu `ArrayList<Confidence> conf`. Proses dalam fungsi ini adalah mengubah format penulisan *rule* dari kode golongan buku menjadi nama golongan buku. *Rule* dengan notasi kode golongan buku disimpan dalam `ArrayList<String> r`, sedangkan *rule* dengan notasi nama golongan buku disimpan dalam `ArrayList<String> kt`.

```

Rule RuleFI(ArrayList<Confidence> conf, JenisBuku JB){
    Rule R= new Rule();
    String ket,rule;
    String titipl = "";
    ArrayList<String> r = new ArrayList<String>();
    ArrayList<String> kt = new ArrayList<String>();

```

```

ArrayList<Double> con = new ArrayList<Double>();
for(int i=0;i<conf.size();i++){
    for(int j=0;j<conf.get(i).freqitem.size();j++){
        rule = "";ket = "";
        for(int l=0;l<conf.get(i).
            freqitem.get(j).size();l++){
            if(l!=conf.get(i).freqitem.get(j).size()-1){
                for(int k=0;k<JB.kodeJenis.size();k++){
                    if(Integer.parseInt(JB.kodeJenis.get(k))==
                        conf.get(i).freqitem.
                            get(j).get(l)){
                        titipl=JB.namaJenis.get(k);}
                }
            }
            if(l==0){
                rule= conf.get(i).freqitem.
                    get(j).get(l).toString();
                ket = "jika "+titipl;
            }else{
                rule=rule+" "+conf.get(i).
                    freqitem.get(j).get(l).toString();
                ket = ket+" dan "+titipl;}
            }else{
                for(int k=0;k<JB.kodeJenis.size();k++){
                    if(Integer.parseInt(JB.kodeJenis.get(k))==
                        conf.get(i).freqitem.get(j).get(l)){
                        titipl = JB.namaJenis.get(k);}
                    }
                rule=rule+"->"+conf.get(i).
                    freqitem.get(j).get(l).toString();
                ket = ket+" maka "+titipl;}
            }
        con.add(conf.get(i).confidence.get(j));
        r.add(rule);
        kt.add(ket);
        R = new Rule(r,kt,con);
    }
}return R;
}

```

**Source Code 4.11** Proses Koleksi *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* golongan buku yang menjadi *consequent* (bagian maka). Implementasi dari proses ini dapat ditunjukkan pada *source code* 4.12.

```

LiftRatio hitungLift(ArrayList<Confidence> conf, Support S,
ArrayList<Rule> rul)
{

```

```

LiftRatio LR = new LiftRatio();
double lift= 0;
double supConsequent = 0;
String fi = "";
ArrayList<Double> L = new ArrayList<Double>();
ArrayList<String> FIT = new ArrayList<String>();
for(int i=0;i<conf.get(0).freqitem.size();i++)
{
    for(int j=0;j<S.kodeBukunya.size();j++)
    {
        if(conf.get(0).freqitem.get(i).
get(conf.get(0).freqitem.get(i).
size()-1)==(Integer.parseInt
(S.kodeBukunya.get(j)))){
            supConsequent = S.bobot.get(j);
        }
    }
    lift = conf.get(0).confidence.get(i)
/supConsequent;
    fi = rul.get(0).freqitem.get(i);
    L.add(lift);
    FIT.add(fi);
    LR = new LiftRatio(FIT,L);
}
return LR;
}
}

```

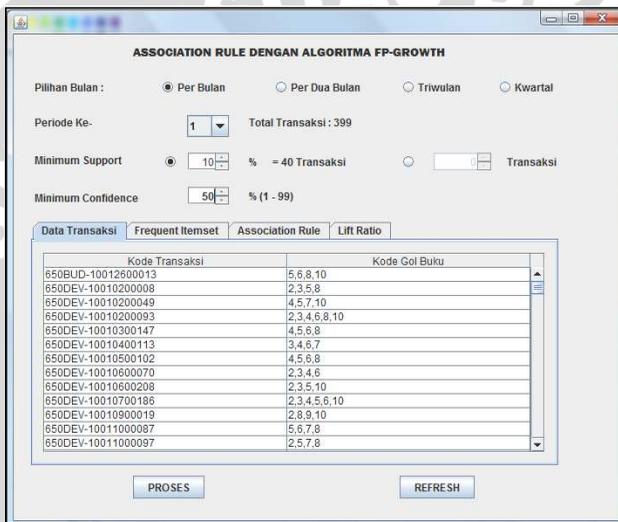
**Source Code 4.12** Proses Lift Ratio

Fungsi `hitungLift` dimulai dengan mencari nilai *support* kode golongan buku yang menjadi *consequent* (bagian maka) dari index data *confidence*. Nilai *support* didapatkan dari data *support frequent 1-itemset* yang dideklarasikan sebagai parameter inputan yaitu `Support s`. 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> L`, sedangkan *frequent itemset* disimpan dalam `ArrayList<String> FIT`.

### 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 proses dan tombol *refresh*. Bagian input terdiri dari tiga data *input*, yaitu data *input* pilihan bulan, data *input* periode bulan, 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*.

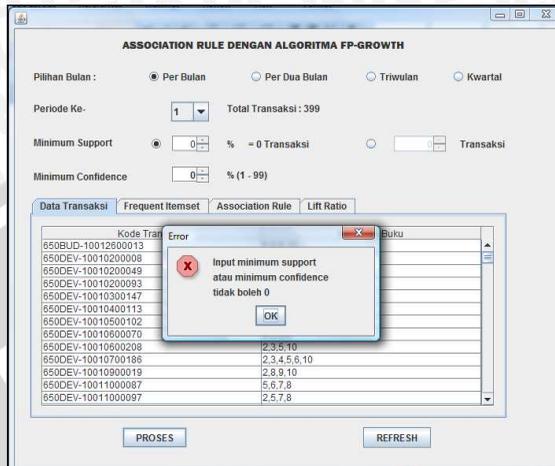
Bagian inputan yang aktif pertama adalah pilihan bulan yang berupa *RadioButton*. Pilihan bulan terdiri dari per bulan, per dua bulan, triwulan, dan kwartal. Pilihan bulan yang dipilih mengubah isi pilihan pada *ComboBox* periode bulan. Setelah periode bulan dipilih, informasi mengenai total transaksi ditampilkan, *RadioButton* minimum *support* menjadi aktif dan bagian *ouput* submenu Data Transaksi muncul tabel transaksi yang berisi data transaksi untuk periode bulan yang telah dipilih seperti yang diilustrasikan pada Gambar 4.1.



**Gambar 4.1** Tampilan Data Transaksi

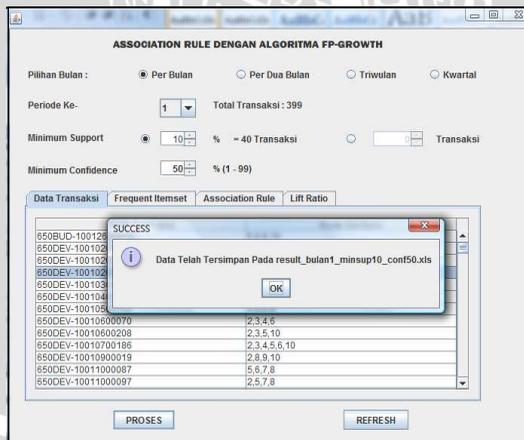
Data inputan untuk minimum *support* diberikan dua macam pilihan yaitu *RadioButton* pertama untuk inputan minimum *support* dalam bentuk prosentase dan *RadioButton* kedua untuk inputan minimum *support* dalam bentuk jumlah transaksi. Bagian inputan minimum *confidence* yang berupa *Spinner* menjadi aktif apabila salah satu *RadioButton* inputan minimum *support* telah dipilih.

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.

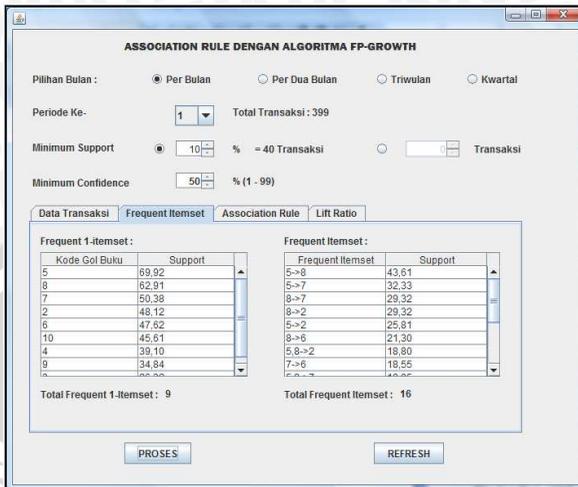


Gambar 4.2 Tampilan Dialog Box Peringatan

Setelah data inputan terisi dengan benar, maka ketika tombol proses ditekan akan keluar *dialog box* bahwa data telah tersimpan seperti yang diilustrasikan pada Gambar 4.3. Selain itu data *output* yaitu *frequent 1-itemset*, *frequent itemset*, *association rule*, dan hasil perhitungan *lift ratio* akan ditampilkan pada masing – masing tabel. Tabel *frequent 1-itemset* beserta jumlahnya dan tabel *frequent itemset* beserta jumlahnya ditampilkan pada submenu *Frequent Itemset* seperti yang diilustrasikan pada Gambar 4.4.



Gambar 4.3 Tampilan Dialog Box Data Telah Tersimpan Pada File Format xls

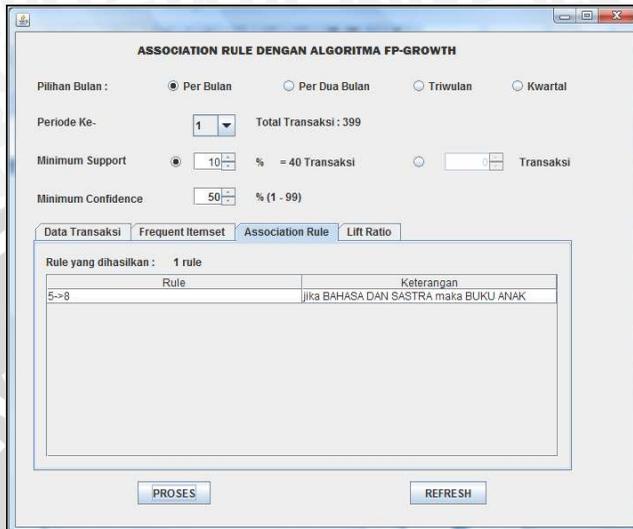


**Gambar 4.4** Tampilan Submenu *Frequent Itemset*

*Frequent 1-itemset* yang memiliki nilai *minimum support* lebih dari atau sama dengan *minimum support* dibentuk kombinasinya untuk menjadi kandidat *frequent itemset*. Pembentukan kombinasi memiliki aturan yaitu yang menjadi bagian *suffix* adalah golongan buku yang memiliki nilai *support* yang terkecil pada kombinasi tersebut.

*Frequent Itemset* yang memiliki nilai *support* lebih dari atau sama dengan *minimum support* akan menjadi kandidat *rule*. Kandidat *rule* yang menjadi *rule* adalah yang memiliki nilai *confidence* lebih dari atau sama dengan *minimum confidence*.

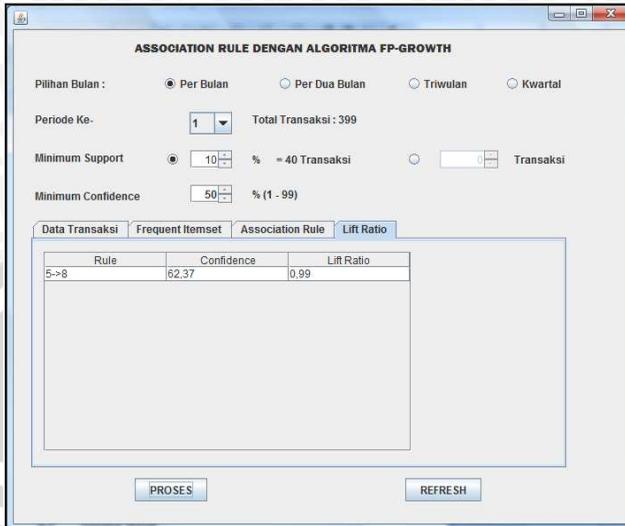
*Rule* yang dihasilkan ditampilkan pada tabel *Rule* yang terdapat pada submenu *Association Rule*. Tabel *Rule* terdiri dari dua kolom yaitu kolom *rule* yang berisi *rule* dalam bentuk kode dan kolom keterangan yang berisi keterangan dari kode *rule*. Pada submenu *Association Rule*, selain ditampilkan tabel *rule* juga ditampilkan informasi jumlah *rule* yang dihasilkan. Antarmuka submenu *Association Rule* ini diilustrasikan pada Gambar 4.5.



**Gambar 4.5** Tampilan Submenu *Association Rule*

*Rule* yang terbentuk kemudian dihitung nilai *lift ratio*. Nilai *lift ratio* ini yang menentukan kekuatan *rule* berdasarkan manfaatnya karena nilai *lift ratio* ini merupakan nilai keseimbangan antara frekuensi kemunculan golongan buku dalam *rule* tersebut muncul secara independen dan muncul secara bersama – sama.

Informasi nilai *confidence* dari *rule* yang dihasilkan dapat dilihat dalam tabel *lift ratio* pada submenu *Lift Ratio*. Hal ini dimaksudkan untuk mempermudah user melakukan analisis nilai *confidence* terhadap nilai *lift ratio* suatu *rule*. Tabel *lift ratio* terdiri dari kolom *rule* yang berisi kode *rule*, kolom *confidence* yang berisi nilai *confidence* untuk tiap *rule*, dan kolom *lift ratio* yang berisi nilai *lift ratio* untuk tiap *rule*. Antarmuka submenu *Lift Ratio* diilustrasikan pada Gambar 4.6.



**Gambar 4.6** Tampilan Submenu *Lift Ratio*

#### 4.4. Pengujian Sistem

Pengujian sistem dilakukan untuk masing – masing pilihan bulan, yaitu per bulan, per dua bulan, triwulan, dan kwartal. Masing – masing pilihan bulan memiliki jumlah periode yang berbeda, yaitu data per bulan terdiri dari 12 periode, data per dua bulan terdiri dari 6 periode, data per triwulan terdiri dari 4 periode, dan data kwartal terdiri dari 3 periode. Data transaksi yang digunakan dalam pengujian ini dibatasi hanya untuk transaksi yang memiliki pembelian golongan buku minimal 4 buah golongan buku.

Pengujian pertama dilakukan untuk mengetahui pengaruh nilai minimum *support* dan nilai minimum *confidence* yang diberikan terhadap jumlah *rule* yang dihasilkan. Pada pengujian pertama ini, minimum *support* yang digunakan adalah sejumlah 10%, 20%, 30%, dan 40%. Minimum *confidence* yang digunakan adalah 40%, 50% dan 60%. Hasil uji pengaruh nilai minimum *support* dan nilai minimum *confidence* terhadap jumlah *rule* yang dihasilkan dapat dilihat pada Tabel 4.2.

**Tabel 4.2** Hasil Uji Pengaruh Nilai Minimum *Support* dan Nilai Minimum *Confidence* terhadap Jumlah *Rule* yang Dihasilkan

	<b>Periode</b>	<b>Minimum Support(%)</b>	<b>Minimum Confidence(%)</b>	<b>Jumlah Rule</b>
<b>Per Bulan</b>	1	10	40	7
			50	1
			60	1
		20	40	4
			50	1
			60	1
		30	40	2
			50	1
			60	1
		40	40	1
			50	1
			60	1
	2	10	40	4
			50	1
			60	0
		20	40	2
			50	1
			60	0
		30	40	2
			50	1
			60	0
		40	40	1
			50	1
			60	0

	<b>Periode</b>	<b>Minimum Support(%)</b>	<b>Minimum Confidence(%)</b>	<b>Jumlah Rule</b>
<b>Per Bulan</b>	3	10	40	7
			50	2
			60	1
		20	40	4
			50	2
			60	1
		30	40	3
			50	2
			60	1
	40	40	1	
		50	1	
		60	1	
	4	10	40	7
			50	4
			60	1
		20	40	5
			50	4
			60	1
		30	40	4
			50	3
			60	1
40		40	1	
		50	1	
		60	1	
5	10	40	9	
		50	4	
		60	2	
	20	40	5	
		50	4	
		60	2	
	30	40	4	
		50	3	
		60	2	
40	40	2		
	50	2		
	60	2		

	Periode	Minimum Support(%)	Minimum Confidence(%)	Jumlah Rule
Per Bulan	6	10	40	5
			50	2
			60	0
		20	40	3
			50	2
			60	0
		30	40	2
			50	2
			60	0
	40	40	1	
		50	1	
		60	0	
	7	10	40	7
			50	1
			60	1
		20	40	4
			50	1
			60	1
		30	40	4
			50	1
			60	1
40		40	1	
		50	1	
		60	1	
8	10	40	9	
		50	4	
		60	0	
	20	40	7	
		50	4	
		60	0	
	30	40	4	
		50	2	
		60	0	
	40	40	2	
		50	2	
		60	0	

	<b>Periode</b>	<b>Minimum Support(%)</b>	<b>Minimum Confidence(%)</b>	<b>Jumlah Rule</b>
<b>Per Bulan</b>	9	10	40	6
			50	1
			60	0
		20	40	3
			50	1
			60	0
		30	40	2
			50	1
			60	0
	40	40	1	
		50	1	
		60	0	
	10	10	40	4
			50	2
			60	1
		20	40	4
			50	2
			60	1
		30	40	3
			50	2
			60	1
		40	40	1
			50	1
			60	1
	11	10	40	7
			50	2
			60	1
20		40	4	
		50	2	
		60	1	
30		40	3	
		50	2	
		60	1	
40	40	1		
	50	1		
	60	1		

	Periode	Minimum Support(%)	Minimum Confidence(%)	Jumlah Rule
<b>Per Bulan</b>	12	10	40	5
			50	1
			60	1
		20	40	4
			50	1
			60	1
		30	40	3
			50	1
			60	1
		40	40	1
			50	1
			60	1
<b>Per Dua Bulan</b>	1	10	40	2
			50	1
			60	0
		20	40	2
			50	1
			60	0
		30	40	2
			50	1
			60	0
	40	40	1	
		50	1	
		60	0	
	2	10	40	7
			50	3
			60	1
		20	40	5
			50	3
			60	1
30		40	4	
		50	3	
		60	1	
40	40	1		
	50	1		
	60	1		

	<b>Periode</b>	<b>Minimum Support(%)</b>	<b>Minimum Confidence(%)</b>	<b>Jumlah Rule</b>
<b>Per Dua Bulan</b>	3	10	40	6
			50	3
			60	1
		20	40	5
			50	3
			60	1
		30	40	4
			50	2
			60	1
	40	40	2	
		50	2	
		60	1	
	4	10	40	6
			50	2
			60	0
		20	40	4
			50	2
			60	0
		30	40	3
			50	2
			60	0
		40	40	1
			50	1
			60	0
5	10	40	7	
		50	1	
		60	0	
	20	40	3	
		50	1	
		60	0	
	30	40	2	
		50	1	
		60	0	
40	40	1		
	50	1		
	60	0		

	<b>Periode</b>	<b>Minimum Support(%)</b>	<b>Minimum Confidence(%)</b>	<b>Jumlah Rule</b>
<b>Per Dua Bulan</b>	6	10	40	6
			50	1
			60	1
		20	40	4
			50	1
			60	1
		30	40	3
			50	1
			60	1
		40	40	1
			50	1
			60	1
<b>Tri wulan</b>	1	10	40	4
			50	1
			60	0
		20	40	3
			50	1
			60	0
		30	40	2
			50	1
			60	0
	40	40	1	
		50	1	
		60	0	
	2	10	40	7
			50	4
			60	1
		20	40	5
			50	4
			60	1
30		40	4	
		50	3	
		60	1	
40	40	1		
	50	1		
	60	1		

	<b>Periode</b>	<b>Minimum Support(%)</b>	<b>Minimum Confidence(%)</b>	<b>Jumlah Rule</b>
<b>Tri wulan</b>	3	10	40	6
			50	1
			60	0
		20	40	2
			50	1
			60	0
		30	40	2
			50	1
			60	0
	40	40	1	
		50	1	
		60	0	
	4	10	40	7
			50	1
			60	1
		20	40	4
			50	1
			60	1
30		40	4	
		50	1	
		60	1	
40		40	1	
		50	1	
		60	1	
<b>Kwartal</b>	1	10	40	6
			50	1
			60	1
		20	40	4
			50	1
			60	1
		30	40	3
			50	1
			60	1
		40	40	1
			50	1
			60	1

	Periode	Minimum Support(%)	Minimum Confidence(%)	Jumlah Rule
Kwartal	2	10	40	6
			50	1
			60	1
		20	40	4
			50	1
			60	1
		30	40	3
			50	1
			60	1
	40	40	1	
		50	1	
		60	1	
	3	10	40	7
			50	1
			60	1
		20	40	4
			50	1
			60	1
30		40	3	
		50	1	
		60	1	
40		40	1	
		50	1	
		60	1	

Pengujian kedua dilakukan untuk mengetahui tingkat kekuatan (*lift ratio*) dari *rule* yang dihasilkan. Pada pengujian ini *rule* yang diuji adalah *rule* untuk minimum *confidence* 50%. Hasil pengujian kedua dapat dilihat pada Tabel 4.3 dan keterangan dari *rule* hasil pengujian dapat dilihat pada Tabel 4.4.

**Tabel 4.3 Hasil Uji Kekuatan (*Lift Ratio*) Rule yang Dihasilkan**

	Periode	Rule	Confidence (%)	Lift Ratio
<b>Per Bulan</b>	1	5 → 8	62.37	0.99
	2	5 → 8	53.69	0.94
	3	8 → 6	51.72	0.91
		5 → 8	62.64	0.95
	4	8 → 5	62.91	0.96
		8 → 6	54.93	0.95
		8,5 → 6	52.24	0.91
	5	5 → 6	52.11	0.9
		8 → 6	54.31	0.85
		5 → 8	65.48	0.97
	5	5 → 6	62.44	0.98
		5,8 → 6	56.59	0.89
	6	5 → 8	59.69	0.92
		5 → 6	52.04	0.99
	7	5 → 8	61.07	0.93
	8	5 → 9	59.63	1.02
		5 → 8	54.66	0.95
		5,9 → 2	52.08	1.01
	9	5 → 8	57.48	1
	10	5 → 8	60.63	1
5 → 6		55.12	0.94	
11	8 → 5	61.54	0.96	
	8 → 2	51.79	0.96	
12	5 → 8	61.57	0.96	
<b>Per Dua Bulan</b>	1	5 → 8	58.71	0.97
	2	8 → 5	63.28	0.96
		8 → 6	53.91	0.94
		5 → 6	50.78	0.89
	3	5 → 8	62.6	0.95
		5 → 6	57.25	0.99
		5,8 → 6	50	0.86
	4	5 → 9	54.52	1
5 → 8		57.74	0.94	

	Periode	Rule	Confidence (%)	Lift Ratio
<b>Per Dua Bulan</b>	5	5 → 8	58.65	1
	6	5 → 8	61.83	0.96
<b>Triwulan</b>	1	5 → 8	59.76	0.96
	2	8 → 6	51.69	0.89
		5 → 8	64.41	0.95
		5 → 6	56.95	0.98
		5,8 → 6	50.79	0.88
	3	5 → 8	57.63	0.96
	4	5 → 8	61.57	0.97
<b>Kwartal</b>	1	5 → 8	61.66	0.96
	2	5 → 8	60.46	0.94
	3	5 → 8	60.46	0.97

**Tabel 4.4** Keterangan *Rule* Hasil Uji Kekuatan *Lift Ratio*

<b>Rule</b>	<b>Keterangan</b>
5 → 6	Jika <b>bahasa dan sastra</b> maka <b>bisnis</b>
5 → 8	Jika <b>bahasa dan sastra</b> maka <b>buku anak</b>
5 → 9	Jika <b>bahasa dan sastra</b> maka <b>buku sekolah</b>
5,8 → 6	Jika <b>bahasa dan sastra</b> dan <b>buku anak</b> maka <b>bisnis</b>
5,9 → 2	Jika <b>bahasa dan sastra</b> dan <b>buku sekolah</b> maka <b>teknik</b>
8 → 2	Jika <b>buku anak</b> maka <b>teknik</b>
8 → 5	Jika <b>buku anak</b> maka <b>bahasa dan sastra</b>
8 → 6	Jika <b>buku anak</b> maka <b>bisnis</b>

Berdasarkan Tabel 4.3 didapatkan informasi bahwa rata – rata nilai *lift ratio rule* yang dihasilkan secara keseluruhan adalah 0.95. Rata – rata nilai *lift ratio rule* yang dihasilkan pada rentang per bulan adalah 0.95. *Rule* yang dihasilkan pada rentang per dua bulan memiliki rata – rata nilai *lift ratio* sebesar 0.95 sedangkan untuk *rule* yang dihasilkan pada rentang triwulan memiliki rata – rata nilai *lift ratio* sebesar 0.94. Rata – rata nilai *lift ratio rule* yang dihasilkan pada rentang kwartal adalah 0.95.

#### 4.5. Analisis Hasil

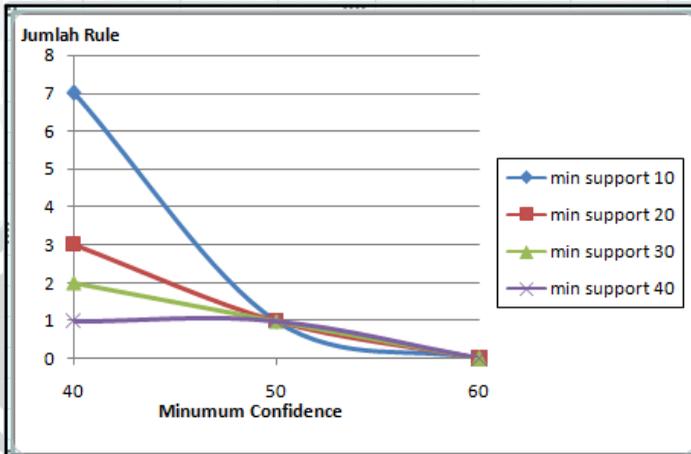
Berdasarkan hasil pengujian pengaruh nilai minimum *support* dan nilai minimum *confidence* terhadap jumlah *rule* yang dihasilkan, didapatkan jumlah *rule* terbanyak yang dihasilkan adalah 9 *rule* yaitu pada periode bulan Mei dan periode bulan Agustus rentang bulan per bulan dengan minimum *support* 10% dan minimum *confidence* 40%. Namun, terdapat hasil pengujian yang tidak menghasilkan *rule* yaitu

- Pada rentang bulan per bulan periode bulan Februari, Juni, Agustus dan September
- Pada rentang bulan per dua bulan periode Januari – Februari, periode Juli – Agustus, dan periode September – Oktober
- Pada rentang bulan triwulan periode Januari – Maret dan Juli – September

ketika nilai minimum *confidence* 60% untuk setiap minimum *support*. Hal ini menjelaskan bahwa pada periode tersebut tidak ada *rule* yang memiliki nilai *confidence* lebih besar atau sama dengan 60% dan batas maksimum tingkat kepercayaan yang dapat digunakan pada pengujian dengan batasan data transaksi memiliki minimal 4 kode golongan buku adalah 60%.

Pengaruh nilai minimum *support* dan nilai minimum *confidence* terhadap jumlah *rule* yang dihasilkan berdasarkan hasil pengujian adalah berbanding terbalik. Hal ini berarti semakin tinggi nilai minimum *support* dan nilai minimum *confidence* yang digunakan maka semakin sedikit jumlah *rule* yang dihasilkan seperti yang digambarkan pada Gambar 4.7.

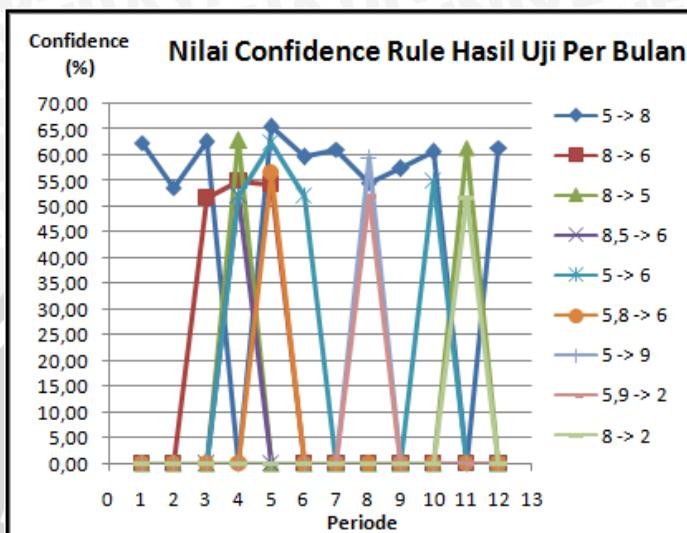
Berbanding terbaliknya nilai minimum *support* dan nilai minimum *confidence* terhadap jumlah *rule* tersebut dikarenakan semakin tinggi nilai minimum *support* yang digunakan berarti semakin tinggi nilai batas *support* yang harus dicapai kandidat *frequent itemset* untuk menjadi *frequent itemset*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



**Gambar 4.7** Grafik Pengaruh Nilai Minimum *Support* dan Nilai Minimum *Confidence* terhadap Jumlah *Rule*

Berdasarkan hasil pengujian kekuatan (*lift ratio*) *rule* yang dihasilkan dengan nilai minimum *confidence* 50% didapatkan rata – rata nilai *lift ratio* dari keseluruhan *rule* yang dihasilkan adalah 0.95. Nilai *lift ratio* tertinggi adalah *rule* jika membeli buku golongan **bahasa dan sastra** maka membeli buku golongan **buku sekolah** (5 → 9) sebesar 1.02 yang didapatkan pada rentang bulan per bulan periode bulan Agustus, sedangkan nilai *lift ratio* terendah adalah 0.85 yaitu pada *rule* jika membeli buku golongan **buku anak** maka membeli buku golongan **bisnis** (8 → 6) yang didapatkan pada rentang bulan per bulan periode bulan Mei.

Rentang bulan yang digunakan pada pengujian selain memberikan pengaruh terhadap variasi *rule* yang dihasilkan juga memberikan pengaruh terhadap nilai *confidence* dari *rule* yang dihasilkan. Hal ini digambarkan dengan grafik nilai *confidence rule* yang dihasilkan per rentang bulan yang digunakan yang dapat dilihat pada Gambar 4.8 sampai Gambar 4.11.

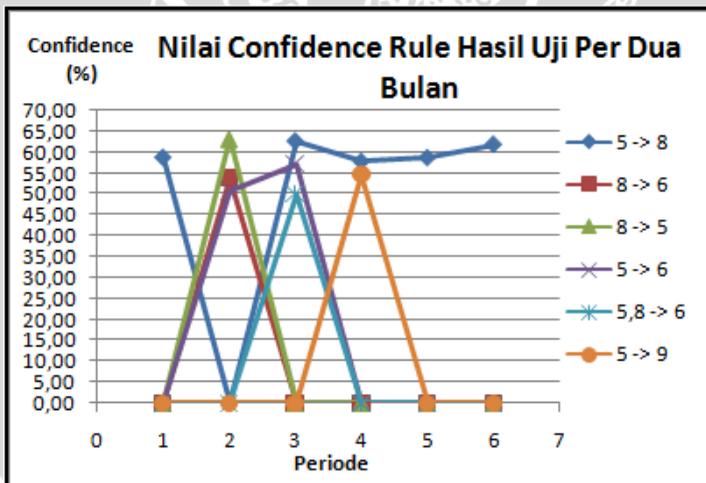


**Gambar 4.8** Grafik Nilai *Confidence Rule* Hasil Uji Per Bulan

Pada grafik Gambar 4.8, *rule* 5 → 8 terlihat memiliki nilai *confidence* yang selalu lebih tinggi dari *rule* lain tiap periode kecuali pada periode 4 (bulan April), periode 8 (bulan Agustus), dan periode 11 (bulan November). Hal ini dikarenakan pada periode 4 (bulan April) dan periode 11 (bulan November) tidak dihasilkan *rule* 5 → 8 tetapi dihasilkan bentuk baru dari *rule* tersebut yaitu 8 → 5 yang nilai *confidence*-nya juga lebih tinggi dari *rule* lain pada periode tersebut. Perubahan bentuk *rule* ini menjelaskan bahwa pada periode tersebut nilai frekuensi golongan buku **buku anak** menjadi lebih tinggi daripada nilai frekuensi golongan buku **bahasa dan sastra**, yang dapat diartikan penjualan golongan buku **buku anak** meningkat pada periode bulan April dan pada bulan November. Selain itu, dapat disimpulkan bahwa hubungan golongan buku **buku anak** dan **bahasa dan sastra** memiliki nilai *confidence* yang konstan yaitu diatas 50% yang berarti dari jumlah frekuensi golongan buku **bahasa dan sastra** yang terjual, 50% sampai 65% terjual bersama – sama dengan golongan buku **buku anak**. *Rule* yang menggambarkan hubungan kedua golongan buku tersebut selalu dihasilkan tiap periode.

Periode bulan Agustus dihasilkan *rule* lain yang memiliki nilai *confidence* lebih tinggi yaitu  $5 \rightarrow 9$ . Hal ini menjelaskan bahwa pada bulan Agustus tingkat penjualan golongan buku **bahasa dan sastra** dan **buku sekolah** terjual secara bersama – sama meningkat yaitu hampir mendekati 60% dari jumlah frekuensi golongan buku **bahasa dan sastra** yang terjual.

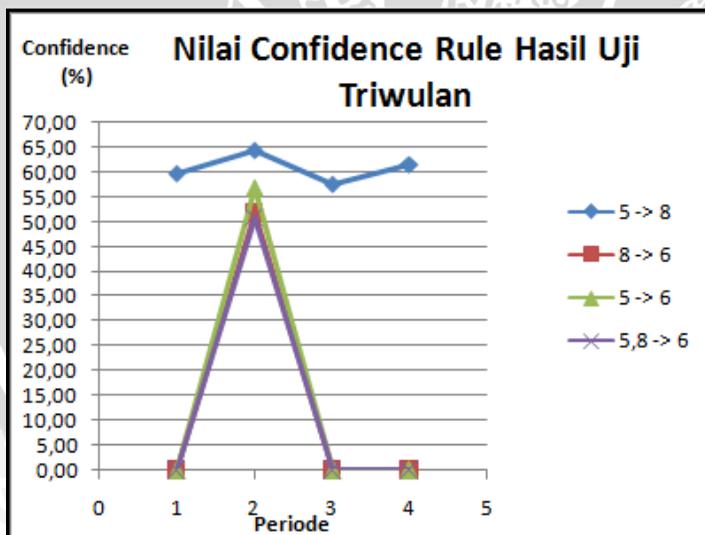
*Rule* lain meskipun memiliki nilai *confidence* yang tidak menonjol, tetapi dianggap menjadi informasi penting yang menjelaskan pola penjualan dan hubungan golongan buku. Seperti *rule*  $5 \rightarrow 6$  dan  $8 \rightarrow 6$ . Grafik *rule*  $5 \rightarrow 6$  menjelaskan bahwa pada bulan April, Mei, Juni dan Oktober tingkat penjualan golongan buku **bahasa dan sastra** dan **bisnis** dibeli secara bersama – sama meningkat dengan tingkat kepercayaan yang cukup tinggi yaitu berkisar antara 50% sampai 65% sehingga dapat diartikan dari jumlah penjualan golongan buku **bahasa dan sastra**, 50% sampai 65%-nya terjual bersama – sama dengan golongan buku **bisnis**. Grafik *rule*  $8 \rightarrow 6$  menjelaskan bahwa pada bulan Maret, April, dan Mei tingkat penjualan golongan buku **buku anak** dan **bisnis** terjual secara bersama – sama meningkat yaitu berkisar antara 50% - 55% dari jumlah frekuensi golongan buku **buku anak** yang terjual.



Gambar 4.9 Grafik Nilai *Confidence Rule* Hasil Uji Per Dua Bulan

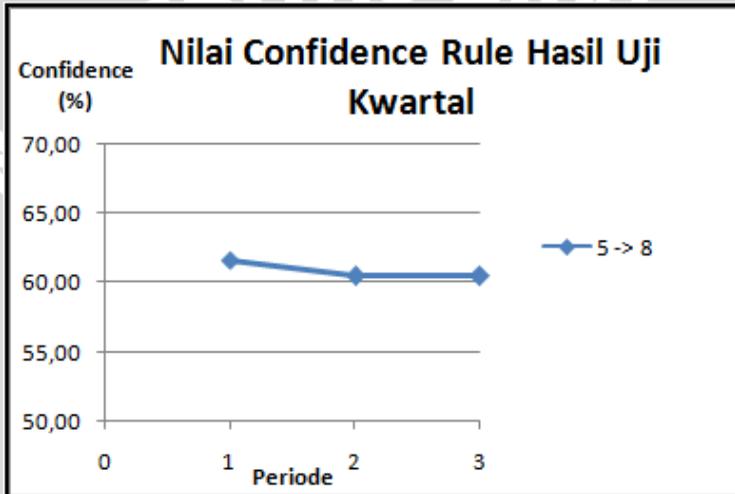
Grafik Gambar 4.9 memperlihatkan bahwa *rule* 5 → 8 tetap selalu memiliki nilai *confidence* tertinggi dari *rule* lainnya tiap periode kecuali pada periode 2 (Maret – April) karena pada periode tersebut dihasilkan *rule* bentuk baru untuk hubungan golongan buku **buku anak** dan **bahasa dan sastra** yaitu 8 → 5 dengan nilai *confidence* yang lebih tinggi juga dari *rule* lainnya. Hal ini menjelaskan bahwa pada periode bulan Maret – April, golongan buku **buku anak** memiliki nilai frekuensi kemunculannya pada data transaksi lebih tinggi daripada golongan buku **bahasa dan sastra**. Jika dihubungkan dengan Gambar 4.8 terlihat pada bulan Maret *rule* tersebut justru tidak dihasilkan pada Gambar 4.9 ketika data transaksi periode bulan Maret dan April digabungkan, nilai frekuensi golongan buku **buku anak** tetap tinggi maka dapat disimpulkan bahwa pada periode bulan Maret nilai frekuensi **buku anak** tidak jauh berbeda dengan nilai frekuensi **bahasa dan sastra**.

*Rule* yang lain seperti 8 → 6, 5 → 6, 5,8 → 6, dan 5 → 9 masih dihasilkan bila dianalisis per dua bulan, hal ini menunjukkan bahwa hubungan golongan – golongan buku tersebut tetap memiliki nilai kepercayaan cukup tinggi yaitu minimal 50% ketika data transaksi dianalisis secara dua bulan.



Gambar 4.10 Grafik Nilai *Confidence Rule* Hasil Uji Per Triwulan

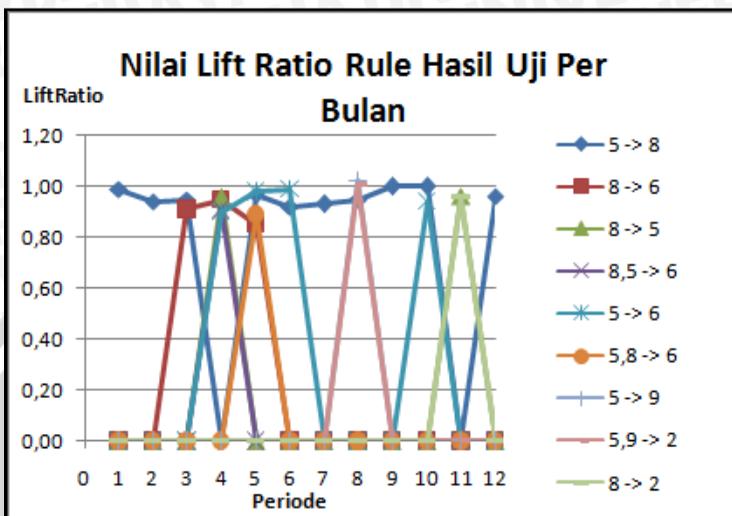
Gambar 4.10 memperlihatkan bahwa *rule* 5 → 8 terlihat memiliki nilai *confidence* yang menonjol dari *rule* lainnya meskipun data transaksi dianalisis per 3 bulan. *Rule* lain yang tetap dihasilkan adalah *rule* 8 → 6, 5 → 6, dan 5,8 → 6 yang semuanya dihasilkan pada periode yang sama yaitu periode 2 (periode bulan April – Juni). Hal ini menunjukkan pada periode bulan April sampai Juni tingkat kepercayaan penjualan golongan buku tersebut secara bersama – sama tetap tinggi.



**Gambar 4.11** Grafik Nilai *Confidence Rule* Hasil Uji Per Kwartal

Nilai *confidence rule* hasil uji rentang bulan kwartal lebih mudah diamati karena *rule* yang dihasilkan hanya *rule* 5 → 8 seperti yang dapat dilihat pada Gambar 4.11. Grafik tersebut menunjukkan bahwa hubungan golongan buku **bahasa dan sastra** dan **buku anak** tetap tinggi nilai *confidence*-nya ketika data transaksi dianalisis per 4 bulan dan antara 60% sampai 65% dari jumlah golongan buku **bahasa dan sastra** yang terjual, terjual bersama – sama dengan golongan buku **buku anak**.

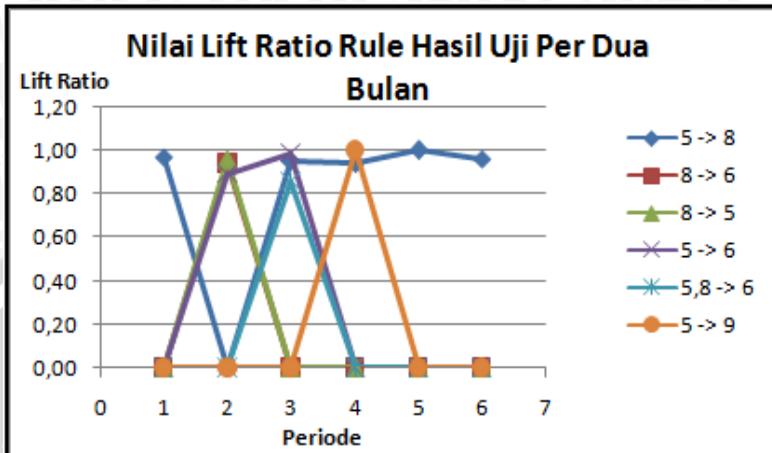
Rentang bulan yang digunakan pada pengujian juga memberikan pengaruh terhadap nilai *lift ratio* dari *rule* yang dihasilkan seperti yang dapat dilihat pada Gambar 4.12 sampai Gambar 4.15. Hal ini dikarenakan nilai *confidence* yang berubah – ubah akibat pengaruh dari rentang bulan yang digunakan.



**Gambar 4.12** Grafik Nilai *Lift Ratio* Hasil Uji Per Bulan

Grafik pada Gambar 4.12 menggambarkan bahwa nilai *lift ratio* dari *rule* yang dihasilkan berkisar antara 0,8 sampai 1,01. *Rule* yang memiliki nilai *lift ratio* minimal 1 adalah *rule* 5 → 8, *rule* 5 → 9, dan *rule* 5,9 → 2. *Rule* 5 → 8 memiliki nilai *lift ratio* minimal 1 pada periode 9 (bulan September) dan periode 10 (bulan Oktober). Hal ini menjelaskan bahwa pada bulan September dan Oktober golongan buku **bahasa dan sastra** dan **buku anak** terjual secara bersama – sama juga memiliki tingkat kepercayaan yang cukup tinggi terhadap jumlah **buku anak** yang terjual. Artinya, dari jumlah golongan **buku anak** yang terjual, jika dihubungkan dengan nilai *confidence* dari Gambar 4.8, maka antara 55% sampai 60% terjual bersama dengan golongan buku **bahasa dan sastra**.

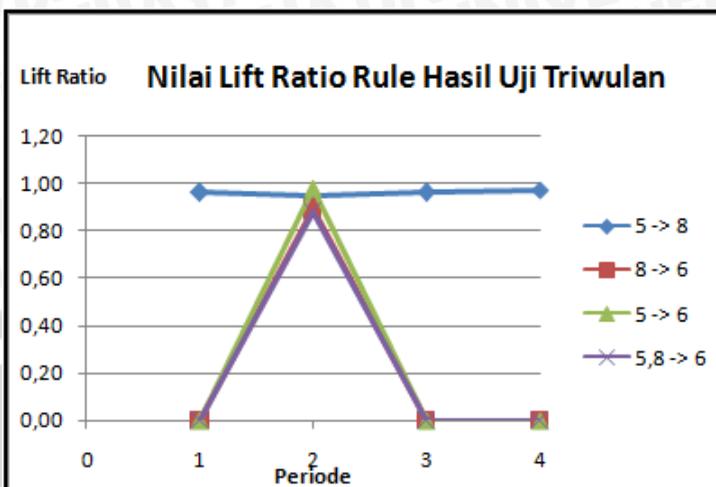
Begitu pula *rule* 5,9 → 2 memiliki nilai *lift ratio* lebih dari 1 pada periode 8 (bulan Agustus). Hal ini menjelaskan bahwa pada bulan Agustus golongan buku **bahasa dan sastra**, **buku sekolah**, dan **teknik** terjual secara bersama – sama juga memiliki tingkat kepercayaan yang cukup tinggi terhadap jumlah golongan buku **teknik** yang terjual. Artinya, dari jumlah golongan buku **teknik** yang terjual, jika dihubungkan dengan nilai *confidence* dari Gambar 4.8, maka antara 50% sampai 55% terjual bersama dengan golongan buku **bahasa dan sastra** dan **buku sekolah**.



**Gambar 4.13** Grafik Nilai *Lift Ratio* Hasil Uji Per Dua Bulan

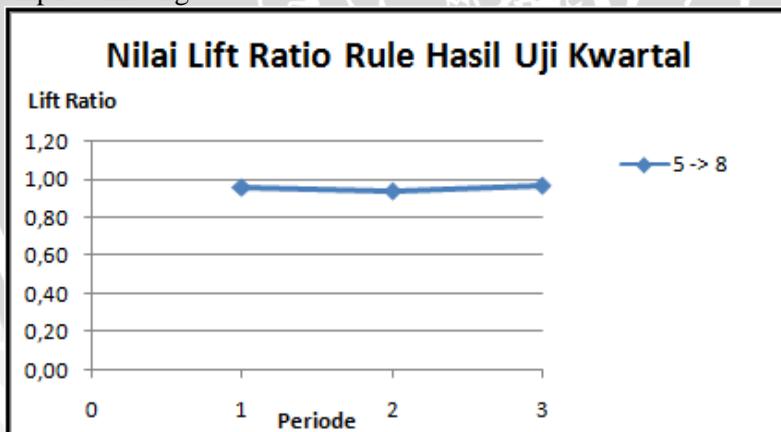
*Rule* pada grafik Gambar 4.13 yang memiliki nilai *lift ratio* minimal 1 adalah *rule* 5 → 8 dan *rule* 5 → 9. *Rule* lainnya memiliki nilai *lift ratio* berkisar antara 0.8 sampai 1. *Rule* 5 → 8 memiliki nilai *lift ratio* minimal 1 pada periode 5 (bulan September – Oktober), sedangkan *rule* 5 → 9 memiliki nilai *lift ratio* minimal 1 pada periode 4 (bulan Juli – Agustus). Hal ini menjelaskan bahwa pada bulan September – Oktober dari jumlah golongan buku **buku anak** yang terjual, jika dihubungkan dengan nilai *confidence* dari Gambar 4.9, hampir mendekati 60% yang terjual bersama golongan buku **bahasa dan sastra**. Sedangkan untuk *rule* 5 → 9, dari jumlah golongan buku **buku sekolah** yang terjual, 55% terjual bersama dengan golongan buku **bahasa dan sastra**.

Grafik pada Gambar 4.14 menggambarkan bahwa nilai *lift ratio* dari *rule* yang dihasilkan berkisar antara 0,8 sampai kurang dari 1. Tidak adanya *rule* yang memiliki nilai *lift ratio* minimal 1 dikarenakan nilai *confidence rule* terhadap nilai *support* golongan buku yang menjadi *consequent* (bagian maka) hampir seimbang. Artinya, misalkan pada *rule* 5 → 8 pada periode 1 (periode bulan Januari – Maret), dari jumlah golongan buku **buku anak** yang terjual, yang terjual secara bersama – sama dengan golongan buku **bahasa dan sastra** tidak dapat mencapai minimal 60% (kurang dari nilai *confidence* seperti pada Gambar 4.10).



**Gambar 4.14** Grafik Nilai *Lift Ratio* Hasil Uji Triwulan

Pada grafik Gambar 4.15 menggambarkan bahwa nilai *lift ratio* dari *rule* yang dihasilkan hampir mendekati 1. Sama pada penjelasan sebelumnya, tidak adanya *rule* yang memiliki nilai *lift ratio* minimal 1 dikarenakan nilai *confidence rule* terhadap nilai *support* golongan buku yang menjadi *consequent* (bagian maka) hampir seimbang.



**Gambar 4.15** Grafik Nilai *Lift Ratio* Hasil Uji Kwartal

Berdasarkan nilai *confidence* dan nilai *lift ratio rule* yang dihasilkan dalam pengujian ini, *rule* yang memiliki nilai *confidence* tinggi belum tentu memiliki nilai *lift ratio* yang tinggi (lebih dari atau sama dengan 1). Hal ini menjelaskan bahwa meskipun nilai kepercayaan dari *rule* tersebut tinggi namun belum tentu *rule* tersebut bermanfaat karena nilai *lift ratio* menggambarkan keseimbangan antara frekuensi kemunculan tiap golongan buku dalam *rule* tersebut secara independen dengan frekuensi kemunculannya secara bersamaan, sedangkan nilai *confidence* menggambarkan tingkat kepercayaan golongan buku dalam *rule* tersebut muncul secara bersamaan. 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

<b>Rule</b>	<b>Keterangan</b>
5 → 9	Jika membeli buku golongan <b>bahasa dan sastra</b> maka membeli buku golongan <b>buku sekolah</b>
5,9 → 2	Jika membeli buku golongan <b>bahasa dan sastra</b> dan <b>buku sekolah</b> maka membeli buku golongan <b>teknik</b>
5 → 8	Jika membeli buku golongan <b>bahasa dan sastra</b> maka membeli buku golongan <b>buku anak</b>

**Tabel 4.6** Detail Informasi *Rule* yang Bermanfaat

<b>Rule</b>	<b>Support</b>	<b>Confidence</b>	<b>Lift Ratio</b>
5 → 9	44.04	59.63	1.02
5,9 → 2	22,94	52.08	1.01
5 → 8	41.62	60.63	1

*Rule* jika membeli buku golongan **bahasa dan sastra** maka membeli buku golongan **buku sekolah** (5 → 9) dan *rule* jika membeli buku golongan **bahasa dan sastra** dan **buku sekolah** maka membeli buku golongan **teknik** (5,9 → 2) dihasilkan pada periode tertentu yaitu pada periode bulan Agustus. Hal ini mengartikan bahwa pada bulan Agustus tingkat penjualan buku golongan **bahasa dan sastra**, **buku sekolah**, dan **teknik** meningkat baik dibeli secara independen maupun secara bersama – sama. Sedangkan untuk *rule*

jika membeli buku golongan **bahasa dan sastra** maka membeli buku golongan **buku anak** ( $5 \rightarrow 8$ ) merupakan *rule* yang paling bermanfaat karena *rule* tersebut dihasilkan tiap periode meskipun nilai *lift ratio* tidak selalu tinggi. Hal ini mengartikan bahwa *rule* tersebut mendominasi pola penjualan buku pada toko buku Togamas.

UNIVERSITAS BRAWIJAYA



## BAB V KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Berdasarkan hasil analisis uji coba dapat diambil kesimpulan:

1. Dalam menerapkan algoritma *FP-Growth* untuk mengetahui pola penjualan buku dari data transaksi penjualan buku berdasarkan golongan buku, parameter inputan yang berpengaruh untuk mendapatkan *rule* dengan nilai *confidence* dan nilai *lift ratio* tinggi adalah penggunaan batasan jumlah golongan buku per transaksi, minimum *support*, dan penggunaan minimum *confidence*.
2. *Rule* yang dihasilkan dari penerapan algoritma *FP-Growth* dalam permasalahan data transaksi penjualan buku ini memiliki rata – rata tingkat kekuatan (*lift ratio*) sebesar 0.95 dengan tingkat kekuatan (*lift ratio*) *association rule* terbesar yang dihasilkan adalah 1.02 dan terkecil adalah 0.85 untuk *rule* dengan tingkat kepercayaan minimal 50% golongan buku tersebut dibeli secara bersamaan. 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.

### 5.2. Saran

Saran yang dapat dijadikan bahan pertimbangan untuk penelitian selanjutnya adalah dalam melakukan analisis pola menggunakan metode *association rule* terhadap data dengan model data seperti data transaksi penjualan buku , proses *preprocessing* data dan penggunaan nilai parameter seperti rentang bulan, nilai minimum *support*, dan nilai minimum *confidence* sangat berpengaruh penting terhadap jumlah *rule* dan kekuatan *rule* yang dihasilkan.

UNIVERSITAS BRAWIJAYA



## DAFTAR PUSTAKA

- Budhi, Gregorius Satia, Andreas Handojo, Christine Oktavina Wirawan. 2009. *Algoritma Generalized Sequential Pattern untuk Menggali Data Sekuensial Sirkulasi Buku pada Perpustakaan UK Petra*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI) Universitas UK Petra. Surabaya.
- Erwin. 2009. *Analisis Market Basket dengan Algoritma Apriori dan FP – Growth*. Jurnal Generic Volume 4 No. 2 Fakultas Ilmu Komputer. Universitas Sriwijaya. Sumatera Selatan.
- Febriyana, Fatma Rika. 2009. *Perbandingan Kecepatan dalam Pencarian Frequent Itemset antara Algoritma FP – Growth dan Cut Both Ways*. Universitas Brawijaya. Malang
- Han, Jiawei dan Micheline Kamber. 2001. *Data Mining : Concepts and Techniques*. Morgan Kaufmann, California
- Handojo, Andreas, Gregorius Satia Budhi, Hendra Rusly. 2004. *Aplikasi Data Mining untuk Meneliti Asosiasi Pembelian Item Barang di Supermarket dengan Metode Market Basket Analysis*. Seminar Nasional Teknologi Informasi Universitas Kristen Petra. Surabaya.
- Huda, Nuqson Masykur. 2010. *Aplikasi Data Mining untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa*. Semarang : Universitas Diponegoro.
- Kantardzic, Mehmed. 2003. *Data Mining : Concepts, Models, Methods, and Algorithms*. John Willey & Sons, Inc. New Jersey.
- Kotler, Philip dan Kevin Lane Keller. 2007. *Manajemen Pemasaran*. PT. Indeks. Jakarta.
- Kusrini, M.Kom. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*. CV Andi Offset. Yogyakarta.

- Moertini, Veronika S. 2002. *Data Mining sebagai Solusi Bisnis*. Integral Majalah Ilmiah Matematika dan Ilmu Pengetahuan Alam, Volume 7 No.1. Jurusan Ilmu Komputer Universitas Katolik Parahyangan. Bandung.
- Pramudiono, Iko. 2003. *Pengantar Data Mining : Menambang Permata Pengetahuan di Gudang Data*. <http://www.ilmukomputer.com>, tanggal akses 15 Maret 2011.
- Rochmah, Affriantari. 2010. *Perancangan Fitur Rekomendasi Film Website Solo Movie dengan Menggunakan Metode Algoritma Apriori*. Universitas Sebelas Maret. Surakarta.
- Ruldeviyani, Yova dan Muhammad Fahrhan. 2008. *Implementasi Algoritma – Algoritma Association Rules sebagai Bagian dari Pengembangan Data Mining Algorithms Collection*. Fakultas Ilmu Komputer Universitas Indonesia. Jakarta.
- Samuel, David. 2008. *Penerapan Struktur FP – Tree dan Algoritma FP – Growth dalam Optimasi Penentuan Frequent Itemset*. Institut Teknologi Bandung
- Santosa, Budi. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Graha Ilmu. Yogyakarta.
- Sholichah, Alfiyatus. 2009. *Data Mining untuk Pembiayaan Murabahah Menggunakan Association Rule*. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim. Malang.
- Verhein, Florian. 2008. *Frequent Pattern Growth (FP – Growth) Algorithm*. School of Information Technologies The University of Sydney. Australia.

Yiapanis, Paraskevas. 2006. *Parallel Mining of Minimal Sample Unique Itemsets*. School of Computer Science, University of Manchester.

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.



UNIVERSITAS BRAWIJAYA



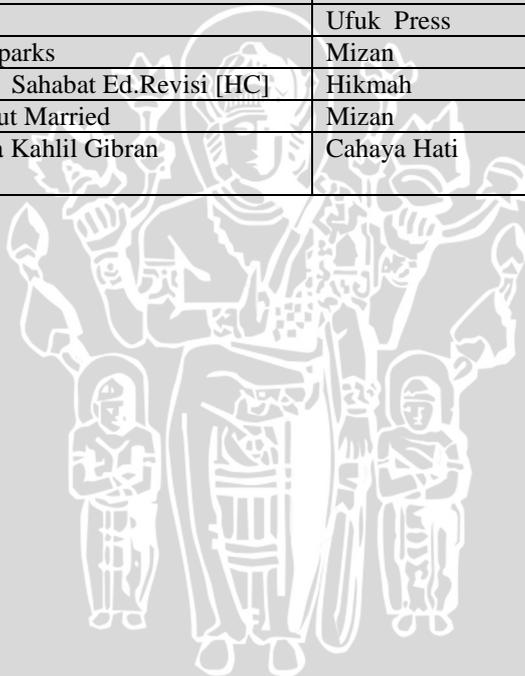
## LAMPIRAN A

**Tabel Sampel Data Transaksi Penjualan Buku**

<b>kode transaksi</b>	<b>tanggal transaksi</b>	<b>nama buku</b>	<b>penerbit</b>	<b>golongan buku</b>
650LEN-10010300149	03/01/2010	Nube Buku Ahli Roh 14	M&C	Media
		Nube Buku Ahli Roh 15	M&C	Media
		Detektif Conan Spesial 30	PT. Elex Media Komputindo	Media
		The Comic Club	PT. Elex Media Komputindo	Media
		Kamar 307	Juxtapose	Umum
650LEN-10010300150	03/01/2010	Menjadi Haji Tanpa Berhaji	Padang Makhsyar Press/ PADMA	Agama
		Avatar 3	Citra Sastra Media PT	Media
		Avatar 4	Citra Sastra Media PT	Media
		Avatar 5	Citra Sastra Media PT	Media
		Ceritakan Mimpi – Mimpimu Edisi Baru	PT. Gramedia Pustaka Utama	Umum
		Padang Bayang Kelabu (Memories of Midnight) Edisi Baru	PT. Gramedia Pustaka Utama	Umum
		Rahasia Sihir Hitam Mematikan (The Novice)	Mizan	Umum
		Ketika Cinta Bertasbih 1 – 2/Paket	Republika	Umum
		Misteri Kereta Api Biru (Cover Baru)	PT. Gramedia Pustaka Utama	Umum
650MAR-10010400068	04/01/2010	Teenage Days	M&C	Media
		Everlasting Portrait	M&C	Sosial
		Stra-wberry	Castle Books	Umum

650LEN-10010500071	05/01/2010	The Secret : Rahasia	PT. Gramedia Pustaka Utama	Bisnis
		Terapi Berpikir Positif	Penerbit Zaman	Bisnis
		Bassam	Diva Press	Umum
		The Legend of John Kepix	Osmo	Umum
		Negeri Van Oranje	Pustaka Populer	Umum
650LEN-10010500076	05/01/2010	Nakayoshi Ed.74/Nop 2009	PT. Elex Media Komputindo	Media
		Harlequin HR : Perayaan Cinta di Penghujung Tahun	PT. Gramedia Pustaka Utama	Umum
		Omiyage (Kisah Orang Biasa Menaklukkan Tanah Jepang)	Andi Offset	Umum
		Ketika Cinta Bertasbih 1-2/Paket	Republika	Umum
		Catatan Pernikahan	Lingkar Pena	Umum
650LEN-10010600049	06/01/2010	A Story of Heroes 29	PT.Elex Media Komputindo	Media
		LC : Rainbow 14	PT.Elex Media Komputindo	Media
		Harlequin KI : Wanita Pilihan Keir O'Connell	PT.Gramedia Pustaka Utama	Umum
		Izrail	Navila	Agama
650MAR-10010700143	07/01/2010	Super Stylish	Diwan	Umum
		Kopi	Harmoni	Teknik
		Al-Qur'an&Terj. Syaamil Cordova Reslt (KLF)	PT.Syaamil Pustaka Utama	Agama
650MAR-10010700155	07/01/2010	Pesan – Pesan Rahasia	Babul Hikmah	Buku Anak
		Hak untuk Malas	Jalasutra	Bisnis
		Harut dan Marut	Navila	Umum
		Layla Majnun	Navila	Umum

		The Witch of Portobello (Sang Penyihir dari Portobello)	PT. Gramedia Pustaka Utama	Umum
		Kalilah dan Dimnah	Navila	Bahasa dan Sastra
650RED-10010700115	07/01/2010	LC : Ergo Proxy 02	PT. Elex Media Komputindo	Media
		Pluto 06	M&C	Media
		Hush, Hush	Ufuk Press	Umum
		People of Sparks	Mizan	Umum
650KMU-10010800026	08/01/2010	Doa Ajaran Sahabat Ed.Revisi [HC]	Hikmah	Agama
		Jangan Takut Married	Mizan	Bisnis
		Fatwa Cinta Kahlil Gibran	Cahaya Hati	Bahasa dan Sastra



## LAMPIRAN B

### 1. Tabel Hasil Pengujian Rentang Per Bulan

Periode : Januari                      Min.support : 10%

#### Frequent 1-itemset

Kode golongan buku	Support
5	69,92
8	62,91
7	50,38
2	48,12
6	47,62
10	45,61
4	39,1
9	34,84
3	26,32

#### Frequent Itemset

Frequent Itemset	Support
5->8	43,61
5->7	32,33
8->7	29,32
8->2	29,32
5->2	25,81
8->6	21,3
5,8->2	18,8
7->6	18,55
5,8->7	18,05
5->6	17,29
8,7->6	13,53
5,8->6	13,53
7->2	13,03
8->10	11,78
8,7->2	10,03
7->10	10,03

Min.confidence : 40%

#### Rule

Rule	Keterangan
5,8->2	jika BAHASA DAN SASTRA dan BUKU ANAK maka TEKNIK
5,8->7	jika BAHASA DAN SASTRA dan BUKU ANAK maka AGAMA
5->7	jika BAHASA DAN SASTRA maka AGAMA
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
8,7->6	jika BUKU ANAK dan AGAMA maka BISNIS
8->2	jika BUKU ANAK maka TEKNIK
8->7	jika BUKU ANAK maka AGAMA

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5,8->2	43,1	0,9
5,8->7	41,38	0,82
5->7	46,24	0,92
5->8	62,37	0,99
8,7->6	46,15	0,97
8->2	41,94	0,87
8->7	41,94	0,83

Min.confidence : 50%

*Rule*

<i>Rule</i>	<i>Keterangan</i>
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5->8	62,37	0,99

Min.confidence : 60%

*Rule*

<i>Rule</i>	<i>Keterangan</i>
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5->8	62,37	0,99

**Periode : Februari**

**Min.support : 10%**

*Frequent 1-itemset*

<i>Kode golongan buku</i>	<i>Support</i>
5	76,03
8	57,3
6	55,06
10	49,81
2	47,19
7	40,07
4	40,07
9	32,96
3	25,09

*Frequent Itemset*

<i>Frequent Itemset</i>	<i>Support</i>
5->8	40,82
5->6	36,7
8->6	29,21
5->10	27,72

8->10	27,72
5,8->10	18,35
5,8->6	16,85
8->2	15,36
6->10	14,23
6->2	13,86
8->7	13,48
5->2	13,11
5->7	11,99
8,6->10	11,61
8,6->2	10,86

Min.confidence : 40%

Rule

Rule	Keterangan
5,8->10	jika BAHASA DAN SASTRA dan BUKU ANAK maka MEDIA
5,8->6	jika BAHASA DAN SASTRA dan BUKU ANAK maka BISNIS
5->6	jika BAHASA DAN SASTRA maka BISNIS
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

Lift Ratio

Rule	Confidence	Lift Ratio
5,8->10	44,95	0,9
5,8->6	41,28	0,75
5->6	48,28	0,88
5->8	53,69	0,94

Min.confidence : 50%

Rule

Rule	Keterangan
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

Lift Ratio

Rule	Confidence	Lift Ratio
5->8	53,69	0,94

Periode : Maret

Min.support : 10%

Frequent 1-itemset

Kode golongan buku	Support
5	66,92
8	65,77
6	56,54
2	48,08
10	44,23
7	41,15
4	39,62

3	28,08
9	28,08

*Frequent Itemset*

<i>Frequent Itemset</i>	<i>Support</i>
5->8	41,92
8->6	34,62
5->6	32,31
8->2	27,31
5->2	24,23
8->10	21,54
6->2	18,85
5,8->6	18,08
5,8->2	17,31
8->7	15,38
6->10	15,38
8,6->2	14,23
5->10	13,08
6->7	12,31
8,6->10	11,54
5,8->10	11,54
8->4	10,38
8,6->7	10
5->7	10

Min.confidence : 40%

*Rule*

<i>Rule</i>	<i>Keterangan</i>
5,8->2	jika BAHASA DAN SASTRA dan BUKU ANAK maka TEKNIK
5,8->6	jika BAHASA DAN SASTRA dan BUKU ANAK maka BISNIS
5->6	jika BAHASA DAN SASTRA maka BISNIS
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
8,6->2	jika BUKU ANAK dan BISNIS maka TEKNIK
8->2	jika BUKU ANAK maka TEKNIK
8->6	jika BUKU ANAK maka BISNIS

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5,8->2	41,28	0,86
5,8->6	43,12	0,76
5->6	48,28	0,85
5->8	62,64	0,95
8,6->2	41,11	0,86
8->2	40,8	0,85
8->6	51,72	0,91

Min.confidence : 50%

Rule

Rule	Keterangan
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
8->6	jika BUKU ANAK maka BISNIS

Lift Ratio

Rule	Confidence	Lift Ratio
5->8	62,64	0,95
8->6	51,72	0,91

Min.confidence : 60%

Rule

Rule	Keterangan
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

Lift Ratio

Rule	Confidence	Lift Ratio
5->8	62,64	0,95

Periode : April

Min.support : 10%

Frequent 1-itemset

Kode golongan buku	Support
8	71
5	65,67
6	57,67
2	53
10	49,33
7	47,67
4	34
9	24,33
3	23,67

Frequent Itemset

Frequent Itemset	Support
8->5	44,67
8->6	39
5->6	37
5->2	31
8->2	25,67
8,5->6	23,33
5->10	23
6->2	19
8,5->2	18,33
6->10	17,33
5->7	16

5,6->2	15,33
<i>Frequent Itemset</i>	<i>Support</i>
6->7	14,67
5,6->10	14,67
8->10	14,67
8->7	13
8,5->10	12,33
5,6->7	12,33
8,5->7	12

Min.confidence : 40%

Rule

Rule	Keterangan
5,6->2	jika BAHASA DAN SASTRA dan BISNIS maka TEKNIK
5->2	jika BAHASA DAN SASTRA maka TEKNIK
5->6	jika BAHASA DAN SASTRA maka BISNIS
8,5->2	jika BUKU ANAK dan BAHASA DAN SASTRA maka TEKNIK
8,5->6	jika BUKU ANAK dan BAHASA DAN SASTRA maka BISNIS
8->5	jika BUKU ANAK maka BAHASA DAN SASTRA
8->6	jika BUKU ANAK maka BISNIS

Lift Ratio

Rule	Confidence	Lift Ratio
5,6->2	41,44	0,78
5->2	43,66	0,82
5->6	52,11	0,9
8,5->2	41,04	0,77
8,5->6	52,24	0,91
8->5	62,91	0,96
8->6	54,93	0,95

Min.confidence : 50%

Rule

Rule	Keterangan
5->6	jika BAHASA DAN SASTRA maka BISNIS
8,5->6	jika BUKU ANAK dan BAHASA DAN SASTRA maka BISNIS
8->5	jika BUKU ANAK maka BAHASA DAN SASTRA
8->6	jika BUKU ANAK maka BISNIS

Lift Ratio

Rule	Confidence	Lift Ratio
5->6	52,11	0,9
8,5->6	52,24	0,91
8->5	62,91	0,96
8->6	54,93	0,95

Min.confidence : 60%

Rule

Rule	Keterangan
8->5	jika BUKU ANAK maka BAHASA DAN SASTRA

Lift Ratio

Rule	Confidence	Lift Ratio
8->5	62,91	0,96

Periode : Mei

Min.support : 10%

Frequent 1-itemset

Kode golongan buku	Support
5	73,51
8	67,54
6	63,81
10	51,49
7	42,16
2	41,79
4	34,7
9	27,61
3	24,25

Frequent Itemset

Frequent Itemset	Support
5->8	48,13
5->6	45,9
8->6	39,93
8->10	30,97
5,8->6	27,24
5->10	23,88
8->7	23,51
5->7	21,64
5,8->10	19,78
5,8->7	19,4
6->10	19,03
8,6->10	16,42
6->7	16,42
8,6->7	14,93
8->2	14,18
6->2	13,81
5,8,6->7	11,19
8,6->2	10,07

Min.confidence : 40%

Rule

Rule	Keterangan
5,8,6->7	jika BAHASA DAN SASTRA dan BUKU ANAK dan BISNIS maka AGAMA
5,8->10	jika BAHASA DAN SASTRA dan BUKU ANAK maka MEDIA
5,8->6	jika BAHASA DAN SASTRA dan BUKU ANAK maka BISNIS
5,8->7	jika BAHASA DAN SASTRA dan BUKU ANAK maka AGAMA
5->6	jika BAHASA DAN SASTRA maka BISNIS
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
8,6->10	jika BUKU ANAK dan BISNIS maka MEDIA
8->10	jika BUKU ANAK maka MEDIA
8->6	jika BUKU ANAK maka BISNIS

Lift Ratio

Rule	Confidence	Lift Ratio
5,8,6->7	41,1	0,97
5,8->10	41,09	0,8
5,8->6	56,59	0,89
5,8->7	40,31	0,96
5->6	62,44	0,98
5->8	65,48	0,97
8,6->10	41,12	0,8
8->10	42,13	0,82
8->6	54,31	0,85

Min.confidence : 50%

Rule

Rule	Keterangan
5,8->6	jika BAHASA DAN SASTRA dan BUKU ANAK maka BISNIS
5->6	jika BAHASA DAN SASTRA maka BISNIS
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
8->6	jika BUKU ANAK maka BISNIS

Lift Ratio

Rule	Confidence	Lift Ratio
5,8->6	56,59	0,89
5->6	62,44	0,98
5->8	65,48	0,97
8->6	54,31	0,85

Min.confidence : 60%

Rule

Rule	Keterangan
5->6	jika BAHASA DAN SASTRA maka BISNIS
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK



### Lift Ratio

Rule	Confidence	Lift Ratio
5,8->6	42,74	0,82
5->6	52,04	0,99
5->8	59,69	0,92
8,6->7	41,98	0,89
8->6	41,33	0,79

Min.confidence : 50%

### Rule

Rule	Keterangan
5->6	jika BAHASA DAN SASTRA maka BISNIS
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

### Lift Ratio

Rule	Confidence	Lift Ratio
5->6	52,04	0,99
5->8	59,69	0,92

Periode : Juli

Min.support : 10%

### Frequent 1-itemset

Kode golongan buku	Support
5	71,63
8	65,38
9	50,48
2	49,52
10	49,04
6	45,67
7	35,1
3	22,6
4	22,12
11	18,27

### Frequent Itemset

Frequent Itemset	Support
5->8	43,75
5->9	35,1
8->2	31,25
8->9	30,77
5->2	26,44
5,8->9	19,71
8->10	19,71
5,8->2	18,75
9->2	16,35
5->10	15,87
8->6	15,87

8,9->2	13,94
<i>Frequent Itemset</i>	<i>Support</i>
9->10	13,46
5,8->10	12,5
5->6	12,02
9->6	11,06
5,8->6	10,58
8,9->10	10,1

Min.confidence : 40%

Rule

Rule	Keterangan
5,8->2	jika BAHASA DAN SASTRA dan BUKU ANAK maka TEKNIK
5,8->9	jika BAHASA DAN SASTRA dan BUKU ANAK maka BUKU SEKOLAH
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
5->9	jika BAHASA DAN SASTRA maka BUKU SEKOLAH
8,9->2	jika BUKU ANAK dan BUKU SEKOLAH maka TEKNIK
8->2	jika BUKU ANAK maka TEKNIK
8->9	jika BUKU ANAK maka BUKU SEKOLAH

Lift Ratio

Rule	Confidence	Lift Ratio
5,8->2	42,86	0,87
5,8->9	45,05	0,89
5->8	61,07	0,93
5->9	48,99	0,97
8,9->2	45,31	0,92
8->2	43,62	0,88
8->9	42,95	0,85

Min.confidence : 50%

Rule

Rule	Keterangan
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

Lift Ratio

Rule	Confidence	Lift Ratio
5->8	61,07	0,93

Min.confidence : 60%

Rule

Rule	Keterangan
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

Lift Ratio

Rule	Confidence	Lift Ratio
5->8	61,07	0,93

Periode : Agustus Min.support : 10%

Frequent 1-itemset

Kode golongan buku	Support
5	73,85
9	58,26
8	57,34
2	51,38
10	50,46
6	42,2
7	39,45
4	33,03
3	19,27

Frequent Itemset

Frequent Itemset	Support
5->9	44,04
5->8	40,37
9->8	32,11
9->2	30,28
5->2	29,82
5,9->2	22,94
5,9->8	22,02
9->10	18,81
8->10	17,89
8->2	16,97
9,8->2	14,22
5->10	14,22
9,8->10	13,3
5,9->10	12,39
5->6	10,09
9->6	10,09
8->6	10,09
2->10	10,09

Min.confidence : 40%

Rule

Rule	Keterangan
5,9->2	jika BAHASA DAN SASTRA dan BUKU SEKOLAH maka TEKNIK
5,9->8	jika BAHASA DAN SASTRA dan BUKU SEKOLAH maka BUKU ANAK
5->2	jika BAHASA DAN SASTRA maka TEKNIK
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
5->9	jika BAHASA DAN SASTRA maka BUKU SEKOLAH
9,8->10	jika BUKU SEKOLAH dan BUKU ANAK maka MEDIA

9,8->2	jika BUKU SEKOLAH dan BUKU ANAK maka TEKNIK
9->2	jika BUKU SEKOLAH maka TEKNIK
9->8	jika BUKU SEKOLAH maka BUKU ANAK

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5,9->2	52,08	1,01
5,9->8	50	0,87
5->2	40,37	0,79
5->8	54,66	0,95
5->9	59,63	1,02
9,8->10	41,43	0,82
9,8->2	44,29	0,86
9->2	40,99	0,8
9->8	43,48	0,76

Min.confidence : 50%

*Rule*

<i>Rule</i>	<i>Keterangan</i>
5,9->2	jika BAHASA DAN SASTRA dan BUKU SEKOLAH maka TEKNIK
5,9->8	jika BAHASA DAN SASTRA dan BUKU SEKOLAH maka BUKU ANAK
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
5->9	jika BAHASA DAN SASTRA maka BUKU SEKOLAH

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5,9->2	52,08	1,01
5,9->8	50	0,87
5->8	54,66	0,95
5->9	59,63	1,02

**Periode : September**

**Min.support : 10%**

*Frequent 1-itemset*

<i>Kode golongan buku</i>	<i>Support</i>
5	71,33
8	57,67
2	55,67
6	49
10	47,33
7	45,67
9	39,33
4	35,33
3	22,33

*Frequent Itemset*

<i>Frequent Itemset</i>	<i>Support</i>
-------------------------	----------------

5->8	41
5->2	34,67
8->2	29,67
5->6	25,67
<i>Frequent Itemset</i>	<i>Support</i>
8->6	23,33
8->10	23,33
2->10	20
5->10	19,33
5,8->2	18
2->6	17,67
5,8->6	16,33
5,8->10	16,33
8,2->10	13,33
8->7	13,33
8,2->6	12,33
5->7	10,67
2->7	10,67

Min.confidence : 40%

Rule

Rule	Keterangan
5,8->2	jika BAHASA DAN SASTRA dan BUKU ANAK maka TEKNIK
5->2	jika BAHASA DAN SASTRA maka TEKNIK
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
8,2->10	jika BUKU ANAK dan TEKNIK maka MEDIA
8,2->6	jika BUKU ANAK dan TEKNIK maka BISNIS
8->2	jika BUKU ANAK maka TEKNIK

Lift Ratio

Rule	Confidence	Lift Ratio
5,8->2	43,9	0,79
5->2	48,6	0,87
5->8	57,48	1
8,2->10	44,94	0,95
8,2->6	41,57	0,85
8->2	41,59	0,75

Min.confidence : 50%

Rule

Rule	Keterangan
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

Lift Ratio

Rule	Confidence	Lift Ratio
5->8	57,48	1

Periode : Oktober

Min.support : 10%

Frequent 1-itemset

Kode golongan buku	Support
5	68,65
8	60,54
6	58,92
2	49,73
7	41,62
10	40,54
4	38,92
9	38,38
3	23,78

Frequent Itemset

Frequent Itemset	Support
5->8	41,62
5->6	37,84
8->6	32,97
8->2	25,95
5->2	21,08
5,8->6	20
8->7	17,3
6->2	17,3
5,8->2	14,59
5->7	14,59
6->7	14,05
8->10	12,97
6->4	12,43
5,8->7	12,43
8,6->2	11,89
8->4	10,81
6->10	10,27

Min.confidence : 40%

Rule

Rule	Keterangan
5,8->6	jika BAHASA DAN SASTRA dan BUKU ANAK maka BISNIS
5->6	jika BAHASA DAN SASTRA maka BISNIS
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
8->6	jika BUKU ANAK maka BISNIS

Lift Ratio

Rule	Confidence	Lift Ratio
5,8->6	48,05	0,82
5->6	55,12	0,94

5->8	60,63	1
8->6	48,03	0,82

Min.confidence : 50%

Rule

Rule	Keterangan
5->6	jika BAHASA DAN SASTRA maka BISNIS
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

Lift Ratio

Rule	Confidence	Lift Ratio
5->6	55,12	0,94
5->8	60,63	1

Min.confidence : 60%

Rule

Rule	Keterangan
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

Lift Ratio

Rule	Confidence	Lift Ratio
5->8	60,63	1

Periode : November

Min.support : 10%

Frequent 1-itemset

Kode golongan buku	Support
8	65
5	64,33
2	54
6	52
10	42,67
9	39
7	39
4	32
3	23,33
11	21

Frequent Itemset

Frequent Itemset	Support
8->5	40
8->2	33,67
5->2	31,33
5->6	29,33
8->6	22
5->10	21,67
2->6	18,67
8,5->2	18

2->10	17,33
8->10	16,33
8,5->6	15,67
<i>Frequent Itemset</i>	<i>Support</i>
5->9	15
5,2->6	14,33
8,5->10	14,33
5,2->10	13,67
8->9	10,67

Min.confidence : 40%

Rule

Rule	Keterangan
5,2->10	jika BAHASA DAN SASTRA dan TEKNIK maka MEDIA
5,2->6	jika BAHASA DAN SASTRA dan TEKNIK maka BISNIS
5->2	jika BAHASA DAN SASTRA maka TEKNIK
5->6	jika BAHASA DAN SASTRA maka BISNIS
8,5->2	jika BUKU ANAK dan BAHASA DAN SASTRA maka TEKNIK
8->2	jika BUKU ANAK maka TEKNIK
8->5	jika BUKU ANAK maka BAHASA DAN SASTRA

Lift Ratio

Rule	Confidence	Lift Ratio
5,2->10	43,62	1,02
5,2->6	45,74	0,88
5->2	48,21	0,89
5->6	45,13	0,87
8,5->2	45	0,83
8->2	51,79	0,96
8->5	61,54	0,96

Min.confidence : 50%

Rule

Rule	Keterangan
8->2	jika BUKU ANAK maka TEKNIK
8->5	jika BUKU ANAK maka BAHASA DAN SASTRA

Lift Ratio

Rule	Confidence	Lift Ratio
8->2	51,79	0,96
8->5	61,54	0,96

Min.confidence : 60%

Rule

Rule	Keterangan
8->5	jika BUKU ANAK maka BAHASA DAN SASTRA

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
8->5	61,54	0,96

**Periode : Desember Min.support : 10%**

*Frequent 1-itemset*

<i>Kode golongan buku</i>	<i>Support</i>
5	73,7
8	64,45
2	52,31
6	51,16
10	47,4
9	39,02
7	38,44
4	36,13
3	19,08
11	13,58

*Frequent Itemset*

<i>Frequent Itemset</i>	<i>Support</i>
5->8	45,38
5->2	35,26
8->2	32,37
8->6	28,9
5->6	26,88
5,8->2	20,52
8->10	20,23
5->10	18,21
5,8->6	17,92
2->6	17,63
5,8->10	14,16
8->9	14,16
8,2->6	13,87
2->9	12,72
2->10	11,85
8,2->9	11,27

**Min.confidence : 40%**

*Rule*

<i>Rule</i>	<i>Keterangan</i>
5,8->2	jika BAHASA DAN SASTRA dan BUKU ANAK maka TEKNIK
5->2	jika BAHASA DAN SASTRA maka TEKNIK
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK
8,2->6	jika BUKU ANAK dan TEKNIK maka BISNIS
8->2	jika BUKU ANAK maka TEKNIK

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5,8->2	45,22	0,86
5->2	47,84	0,91
5->8	61,57	0,96
<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
8,2->6	42,86	0,84
8->2	43,92	0,84

Min.confidence : 50%

*Rule*

<i>Rule</i>	<i>Keterangan</i>
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5->8	61,57	0,96

Min.confidence : 60%

*Rule*

<i>Rule</i>	<i>Keterangan</i>
5->8	jika BAHASA DAN SASTRA maka BUKU ANAK

*Lift Ratio*

<i>Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
5->8	61,57	0,96