PENGENALAN POLA TRANSAKSI SIRKULASI BUKU PADA *DATABASE* PERPUSTAKAAN MENGGUNAKAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN*

SKRIPSI

Disusun oleh :

SUPARDI

NIM. 0710963050

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2014

PENGENALAN POLA TRANSAKSI SIRKULASI BUKU PADA *DATABASE* PERPUSTAKAAN MENGGUNAKAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN*

SKRIPSI

Untuk memenuhi sebagian persyaratan
untuk mencapai gelar Sarjana Komputer

Disusun oleh :

SUPARDI

NIM. 0710963050

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2014

LEMBAR PERSETUJUAN

PENGENALAN POLA TRANSAKSI SIRKULASI BUKU PADA *DATABASE* PERPUSTAKAAN MENGGUNAKAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN*

SKRIPSI

Disusun oleh :

SUPARDI

NIM. 0710963050

Skripsi ini telah disetujui oleh :

Dosen Pembimbing I,	**Dosen Pembimbing II,**
Dian Eka Ratnawati, S.Si.,M.Kom	**Wayan Firdaus Mahmudy, S.Si.,MT.,Ph.D**
NIP. 19730619 200212 2 001	**NIP. 19720919 199702 1 001**

LEMBAR PENGESAHAN

PENGENALAN POLA TRANSAKSI SIRKULASI BUKU PADA *DATABASE* PERPUSTAKAAN MENGGUNAKAN ALGORITMA *GENERALIZED SEQUENTIAL PATTERN*

SKRIPSI

Untuk memenuhi sebagian persyaratan
untuk mencapai gelar Sarjana Komputer

Disusun oleh:

SUPARDI

NIM. 0710963050

Setelah dipertahankan di depan Majelis Penguji pada tanggal 11 Agustus 2014
dan dinyatakan memenuhi syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Penguji I,

Rekyan Regasari MP, ST., MT.
NIK. 770414 06 1 2 0253

Penguji II,

Suprapto, ST.,MT.
NIP. 19710727 199603 1 001

Penguji III,

M. Ali Fauzi, S.Kom., M.Kom

Mengetahui,
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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

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

Malang, Agustus 2014

Mahasiswa,

Supardi

NIM. 0710963050

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, hanya dengan rahmat dan karunia yang telah diberikan kepada penulis, sehingga dapat menyelesaikan skripsi yang berjudul "Pengenalan Pola Transaksi Sirkulasi Buku pada *Database* Perpustakaan Menggunakan Algoritma *Generalized Sequential Pattrens*". Skripsi ini merupakan salah satu syarat untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Dalam penyelesaian skripsi ini, penulis banyak mengalami hambatan karena keterbatasan ide tentang bagaimana menyusun sebuah laporan. Penulis bersyukur karena dapat menyelesaikan laporan skripsi dengan bantuan dari berbagai pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan ucapan terima kasih yang setinggi-tingginya kepada :

1. Dian Eka Ratnawati, S.Si., M.Kom., selaku dosen pembimbing I yang telah membimbing dengan bijaksana dan sabar dalam penyusunan skripsi ini.
2. Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., selaku dosen pembimbing II yang telah membimbing dengan bijaksana dan sabar dalam penyusunan skripsi ini.
3. Drs. Marji, M.T, selaku Ketua Program Studi Ilmu Komputer di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Issa Arwani, S.Kom., MSc., selaku Sekretaris Program Studi Ilmu Komputer di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Yusi Tyroni Mursityo, S.Kom., M.S., selaku dosen pembimbing akademik atas nasehat, bimbingan, saran, dukungan yang diberikan selama penulis menuntut ilmu di Program Studi Ilmu Komputer.
6. Ir. Sutrisno, M.T., selaku Ketua Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
7. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

8. Wiwin Lukitohadi, S.Psi., S.H., CHRM beserta staf di BKPK yang telah banyak memberikan motivasi, bimbingan dan arahan.
9. Segenap staf dan karyawan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya yang telah memberikan pelayanan dan banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
10. Staf PDIH Fakultas Hukum Universitas Brawijaya yang telah memberikan data untuk digunakan dalam skripsi ini.
11. Ibu, istri dan anakku serta keluarga besarku yang tersayang, terima kasih atas dukungan dan doanya.
12. Teman saya Mas Ardi yang sering memberikan bantuan dan motivasi sehingga Penulis dapat menyelesaikan skripsi ini.
13. Teman saya Yudi, Kukuh, Rhizma, Nova, Azhar, Fajar, Agham, Idham, Marissa, Anam, Rijal, Tri Prayudianto serta semua teman-teman Program Studi Ilmu Komputer 2007 Kelas B atas segala bantuan, motivasi dan doanya.
14. Seluruh pihak yang tidak dapat disebut secara langsung yang telah memberikan bantuan demi terselesaikannya skripsi ini.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan laporan ini yang disebabkan oleh keterbatasan kemampuan dan pengalaman. Oleh karena itu, Penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi penelitian ini.

Penulis berharap semoga laporan ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya untuk pengembangan selanjutnya.

Malang, Agustus 2014

Penulis

ABSTRAK

Supardi. 2014. Pengenalan Pola Transaksi Sirkulasi Buku Pada *Database* Perpustakaan Menggunakan Algoritma *Generalized Sequential Pattern*.
Dosen Pembimbing: Dian Eka Ratnawati, S.Si., M.Kom dan Wayan Firdaus Mahmudy, S.Si., M.T.,Ph.D.

Setiap perpustakaan memiliki data transaksi peminjaman buku. Di dalam tumpukan data tersebut banyak informasi yang bisa digali. Untuk menganalisa data yang sangat banyak akan sulit jika hanya dilakukan analisis data sederhana, sehingga dibutuhkan teknik khusus yaitu *data mining.* Dalam penelitian ini digunakan a*ssociation rule mining* dengan algoritma *Generalized Sequential Patterns* (GSP) yaitu mencari kombinasi kategori buku yang paling sering dipinjam. Ukuran kepercayaan yang digunakan adalah *support* dan *confidence.*

Proses untuk menemukan pola pada algoritma ini yaitu menentukan *candidates* yang merupakan kumpulan transaksi peminjaman buku. Dari kategori buku yang dipinjam akan dilakukan pembetukan *candidates* dan dikombinasikan (*join*), sehingga *candidate* yang memenuhi *minimum support* akan dikombinasikan lagi dengan *candidate* lainnya, sehingga membentuk *length*-3 *candidates*. Proses ini dilakukan sampai tidak ada lagi *candidate* atau *frequent sequence.*

Dari hasil pengujian pada periode 1 bulan, 3 bulan dan 5 bulan dengan jumlah data yang digunakan sebanyak 2006 *record* data didapatkan *candidate* kategori 345,346 dengan nilai *minimum support* sebesar 13,76% dan *minimum confidence* sebesar 35,44% sebagai nilai tertinggi. Sehingga disimpulkan jika meminjam buku kategori Hukum Pidana (345) maka juga meminjam Hukum Perdata (346).

Kata kunci : data mining, GSP, asosiasi

ABSTRACT

Supardi. 2014. Implementation The Pattern of Book Transaction Circulation at Database of Library Using Generalized Sequential Pattern Algorithm. Advisor: Dian Eka Ratnawati, S.Si., M.Kom and Wayan Firdaus Mahmudy, S.Si., M.T.,Ph.D.

Each library has lending transaction data. There are many informations can we find from those pile of data. It will be difficult for us to analyze so many data by using the simple data analysis, so it needs the specific technique that is data mining. This research uses the association rule mining with Generalized Sequential Patterns (GSP) algorithm as the tool to find the category of the most boorowed book. The measurement of trust used are support and confidence.

The process to find the pattern of this algorithm is by determining which candidates are the collection of book lending transaction. From the category of book lending, it will compose the candidates and combine it (joint), so that the candidate which fulfill the minimum support will be combined with another candidate, and it will compose length-3 candidates. This process is done until there is no candidate found nor frequent sequence.

From the test results of the experiment in the one month period, three months and five months period with 2006 record of data used, it gets 345,346 categorries of candidate with 13.76% minimum support value and of 35.44% minimum confidence as the highest value. Thus, it can be conclude if the borrowed book with Criminal Law category (345), so the borrowed book with the Civil Law category (346).

Keywords: data mining, GSP, association

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

SOURCE CODE

BAB I
PENDAHULUAN

1. 1 Latar Belakang

Hampir semua lembaga pendidikan menggunakan komputer sebagai sarana untuk mengelola data, tak terkecuali perpustakaan. Perpustakaan sebagaimana yang ada dan berkembang sekarang telah dipergunakan sebagai salah satu pusat informasi, sumber ilmu pengetahuan, penelitian, rekreasi, pelestarian khasanah budaya bangsa, serta memberikan berbagai layanan jasa lainnya (Lasa, 1998). Informasi yang diberikan oleh perpustakaan sangat membantu civitas akademika terutama mahasiswa yang membutuhkan literatur kuliah maupun tugas akhir. Transaksi yang dilakukan dalam sebuah perpustakaan adalah peminjaman buku atau literatur lainnya. Hal ini biasa disebut dengan sirkulasi buku. Pengertian pelayanan sirkulasi sebenarnya adalah mencakup semua bentuk kegiatan pencatatan yang berkaitan dengan pemanfaatan, penggunaan koleksi perpustakaan dengan tepat guna dan tepat waktu untuk kepentingan pengguna jasa perpustakaan (Lasa, 1998).

Saat ini kebanyakan perpustakaan memiliki *database* sirkulasi buku, baik yang sederhana maupun *database* dalam skala besar. Data sirkulasi buku pada perpustakaan sangatlah banyak apalagi jika perpustakaan tersebut memiliki banyak koleksi dan pengguna perpustakaan. Setiap hari perpustakaan melakukan puluhan bahkan ratusan transaksi. Dalam kumpulan data transaksi yang sangat banyak tersebut kemungkinan terdapat informasi signifikan yang dapat membantu dalam mengambil keputusan. Ketika seorang pengguna melakukan transaksi peminjaman buku, dimungkinkan buku-buku yang dipinjam merupakan buku yang saling terkait. Dengan mengetahui pola (*pattern*) sekuensial peminjaman buku pada perpustakaan, banyak putusan/kebijakan strategis yang dapat diambil oleh pengelola/pimpinan perpustakaan, misalnya: memberi informasi pada penggunanya tentang buku-buku yang berelasi, pengaturan peletakan buku-buku yang berelasi pada rak buku secara berdekatan, dan pengadaan/penambahan buku-buku yang berelasi.

Untuk menganalisa data dalam skala besar akan sulit jika hanya dilakukan analisis data sederhana, misalnya dengan menggunakan Microsoft Office Excel, dibutuhkan keahlian untuk menuliskan banyak rumus. Selain itu, waktu yang dibutuhkan juga akan sangat lama atau tidak efisien, padahal sering suatu hasil analisis harus segera diketahui untuk mengambil keputusan (*decision support*) yang cepat dan tepat. Alternatif untuk menangani masalah analisis data dalam skala besar ini adalah dibutuhkan teknik khusus yaitu *data mining*. *Data mining* ialah ekstraksi informasi atau pola yang penting atau menarik dari data yang ada di *database* yang besar (Han, 2001). Hubungan yang dicari dalam *data mining* dapat berupa hubungan antara dua atau lebih dalam satu dimensi.

Data mining sendiri, memiliki berbagai macam metode dimana penggunaannya bergantung pada permasalahan yang dihadapi. Setiap transaksi peminjaman buku, pengguna dapat meminjam lebih dari satu buku yang dimungkinkan buku yang dipinjam merupakan golongan buku yang sama atau golongan buku yang berbeda. Dalam permasalahan pencarian pola transaksi peminjaman buku ini dapat diselesaikan oleh *data mining* dengan metode *association rule.*

Pada penelitian sebelumnya, yang dilakukan oleh Gregorius Satia Budhi, Andreas Handojo, dan Christine Oktavina Wirawan (2009) dengan judul Algoritma *Generalized Sequential Pattern* untuk Menggali Data Sekuensial Sirkulasi Buku Pada Perpustakaan UK Petra diketahui bahwa kecepatan proses aplikasi penerapan algoritma GSP untuk data transaksi yang cukup banyak membutuhkan waktu kurang dari 10 menit. Hal ini membuktikan bahwa algoritma GSP cukup efisien dalam pengolahan data transaksi.

Berdasarkan latar belakang yang telah dijelaskan tersebut, maka teknik *data mining* yang dipilih pada penelitian ini yaitu *association rule* menggunakan algoritma GSP dalam pencarian pola peminjaman buku berdasarkan kategori/klasifikasi pada data transaksi peminjaman buku, dengan judul penelitian **"Pengenalan Pola Transaksi Sirkulasi Buku pada Database Perpustakaan Menggunakan Algoritma *Generalized Sequential Pattern*.**

1. 2 Rumusan Masalah

Permasalahan yang dijadikan subyek penelitian dari skripsi ini adalah bagaimana menerapkan metode *association rule* dengan algoritma GSP untuk mengetahui pola peminjaman buku berdasarkan kategori/klasifikasi buku dari data transaksi peminjaman buku.

1. 3 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

a. Penelitian dilakukan dengan menggunakan data transaksi peminjaman buku di perpustakaan Fakultas Hukum Universitas Brawijaya pada bulan Januari sampai dengan Desember tahun 2012.
b. Input yang dibutuhkan untuk sistem antara lain data transaksi peminjaman buku perpustakaan Fakultas Hukum Universitas Brawijaya dengan rentang bulan dan periode data berbeda, besar minimum *support*, dan besar minimum *confidence*.
c. Output yang dihasilkan adalah aturan assosiatif (*association rule)*.

1. 4 Tujuan Penelitian

Tujuan dari skripsi ini adalah untuk mengetahui pola peminjaman buku berdasarkan kategori buku dari data transaksi peminjaman buku, menerapkan metode *association rule* menggunakan algoritma GSP.

1. 5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah :

a. Memberikan informasi mengenai pola peminjaman buku berdasarkan kategori buku di perpustakaan Fakultas Hukum Universitas Brawijaya.
b. Untuk pemanfaatan lebih lanjut informasi mengenai pola peminjaman buku diharapkan dapat mendukung kepastian, misalnya dalam peletakan rak buku dan menyediakan stok buku berdasarkan kategori/klasifikasi buku.

1. 6 Sistematika Penulisan

Sistematika penulisan yang akan diuraikan dalam tugas akhir ini terbagi dalam enam bab dengan masing-masing bab diuraikan sebagai berikut:

BAB I PENDAHULUAN

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

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi teori-teori dari berbagai pustaka yang menunjang penelitian dalam penulisan tugas akhir. Adapun teori yang tercakup dalam bab ini yaitu definisi *data mining*, *association rule mining*, *market basket analysis*, dan algoritma GSP.

BAB III METODE PENELITIAN DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dan langkah-langkah yang harus dilakukan dalam penelitian ini.

BAB IV IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai lingkungan implementasi dan algoritma operasi yang diimplementasikan untuk pengembangan perangkat lunak.

BAB V PENGUJIAN DAN ANALISIS

Pada bab ini akan di jelaskan tentang prosedur uji dan hasil yang diharapkan serta analisis hasilnya. Pada bagian akhir dilakukan analisis hasil pengujian keseluruhan.

BAB VI PENUTUP

Berisi kesimpulan dari seluruh rangkaian penelitian dan saran yang diharapkan bermanfaat untuk pengembangan tugas akhir ini selanjutnya.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

Penelitian tentang pencarian pola (*mining sequential pattern*) akhir-akhir ini mengalami perkembangan yang sangat pesat, salah satunya dilakukan oleh Agrawal dan Srikant tahun 1994. Dilanjutkan dengan penelitian yang dilakukan oleh Srikant sendiri pada tahun 1996 sehingga memperoleh gelar Doktor bidang komputer. Srikant mengembangkan algoritma apriori untuk proses *mining association rule.*

Leo Willyanto Santoso (2003) melakukan penelitian data mining dengan judul ”Pembuatan perangkat lunak data mining untuk penggalian kaidah asosiasi menggunakan metode apriori”. Dari penelitian ini dihasilkan kesimpulan bahwa waktu komputasi untuk menghasilkan kaidah asosiasi dipengaruhi oleh jumlah transaksi dan penggunaan struktur data ”tidlist” pada algoritma apriori menyebabkan waktu komputasi yang dibutuhkan relatif berkurang karena hanya memerlukan pembacaan basis data sekali saja.

Beberapa peneliti berusaha untuk menyempurnakan metode apriori, sehingga ditemukan algoritma baru yang merupakan penyempurnaan dari algoritma apriori yaitu algoritma *Generalized Sequential Patterns* (GSP) atau dengan nama lain *apriori all.* Algoritma GSP merupakan suatu algoritma yang dapat memproses dan menemukan semua pola sekuensial dan non sekuensial yang ada (Zaki,1997). Penjelasan tentang algoritma GSP pada sub bab berikutnya.

Pada tahun 2009, Gregorius Satia Budhi, Andreas Handojo dan Christine Oktavina Wirawan dari Universitas Kristen Petra meneliti algoritma GSP untuk menggali data sekuensial sirkulasi buku pada perpustakaan UK Petra. Penelitian ini menghasilkan aplikasi yang bisa menemukan *association rule* dan *sequential pattern rules* pada transaksi peminjaman buku perpustakaan di UK Petra. Selain itu dilakukan pengujian kecepatan proses terhadap aplikasi yang dibuat dalam meng-*generate rule* dari beberapa macam data transaksi berbeda. Dari hasil pengujian kecepatan proses ini dapat diketahui bahwa semakin banyak jumlah

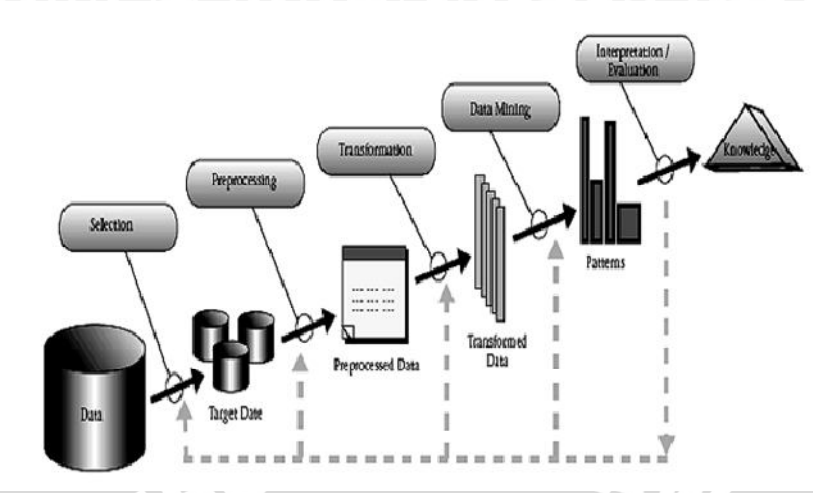
transaksi dan jumlah item yang diproses, waktu proses yang dibutuhkan semakin lama. Lama waktu proses ini masih kurang dari 10 menit untuk memproses transaksi yang cukup banyak, sehingga aplikasi yang dibuat masih dianggap wajar untuk dipakai sebagai *decision support* sistem diperpustakaan UK Petra.

2.2 *Data Mining*

Perkembangan *data mining* yang pesat tidak dapat lepas dari perkembangan teknologi informasi yang memungkinkan data dalam jumlah besar terakumulasi. Saat ini banyak kita jumpai pengertian *data mining.* Beberapa pengertian *data mining* yang terkenal diantaranya adalah :

1. Menurut Gartner Group, *data mining* adalah proses untuk menemukan korelasi baru, pola dan trend dengan seleksi pada sejumlah data yang besar dengan menggunakan teknologi pengenalan pola, statistik dan matematika (Larose, 2005).
2. *Data mining* adalah analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data dengan cara yang berbeda dengan sebelumnya, yang dapat dipahami dan bermanfaat bagi pemilik data. *Data mining* merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, *database*, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari *database* yang besar (Larose, 2005).
3. *Data mining* atau *Knowledge Discovery in Databases* (KDD) adalah pengambilan informasi yang tersembunyi, dimana informasi tersebut sebelumnya tidak dikenal dan berpotensi bermanfaat (Fayyad, 1996).

Data mining merupakan bagian dari proses *Knowledge Discovery in Databases* (KDD). Proses dari KDD dapat dilihat pada Gambar 2.1.

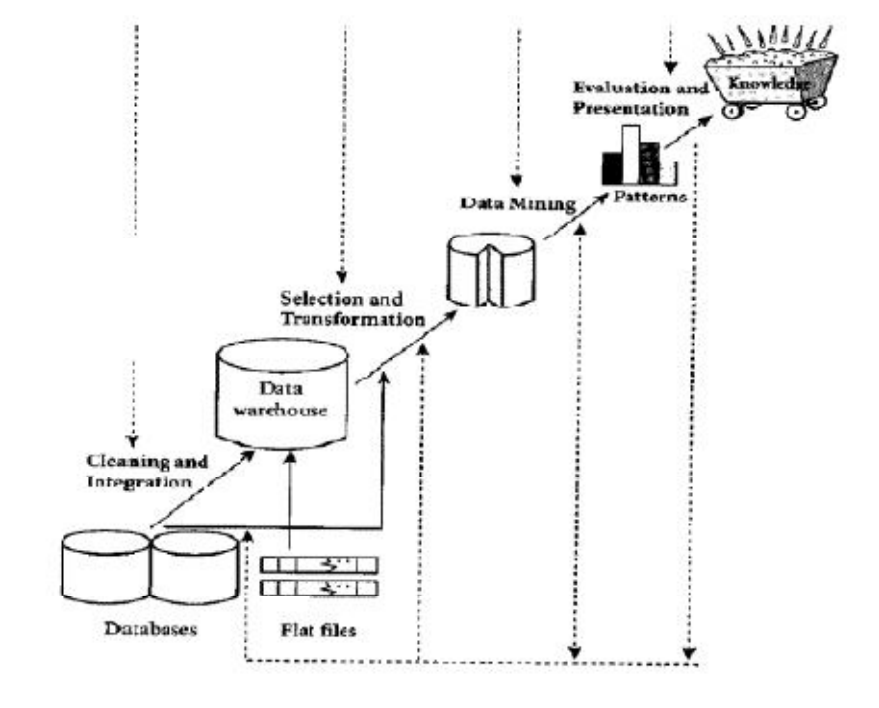


Gambar 2.1 Tahap-tahap dalam Proses KDD (Fayyad, 1996)

Dari definisi-definisi yang telah disampaikan, hal penting yang terkait dengan *data mining adalah* (Kusrini dan Emha, 2009) :

- *Data mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
- Data yang akan diproses berupa data yang sangat besar.
- Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

Data mining ketika diterapkan pada data dalam skala besar diperlukan metodologi sistematis tidak hanya ketika melakukan analisa saja tetapi juga ketika mempersiapkan data dan juga melakukan interpretasi dari hasilnya sehingga bermanfaat. *Data mining* seharusnya dipahami sebagai suatu proses, yang memiliki tahapan-tahapan tertentu dan juga ada umpan balik dari setiap tahapan ke tahapan sebelumnya. Pada umumnya proses *data mining* berjalan interaktif karena tidak jarang hasil *data mining* pada awalnya tidak sesuai dengan harapan analisnya sehingga perlu dilakukan desain ulang prosesnya. Tahapan pada *data mining* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Tahap-tahap Data Mining

Sumber : Seminar Nasional Teknologi (2007)

Tahap-tahap proses *data mining* dapat dijelaskan sebagai berikut :

1. Pembersihan data (*data cleaning*)

   Pembersihan data merupakan proses menghilangkan *noise* dan data yang tidak konsisten atau data tidak relevan. Pada umumnya data yang diperoleh, baik dari *database* suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesa *data mining* yang dimiliki. Data-data yang tidak relevan itu juga lebih baik dibuang. Pembersihan data juga akan mempengaruhi performasi dari teknik *data mining* karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

2. Integrasi data (*data integration*)

   Integrasi data merupakan penggabungan data dari berbagai *database* ke

dalam satu *database* baru. Tidak jarang data yang diperlukan untuk *data mining* tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database* atau file teks. Integrasi data dilakukan pada atribut-aribut yang mengidentifikasikan entitas-entitas yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

3. Seleksi Data (*Data Selection*)

   Data yang ada pada *database* sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*. Sebagai contoh, sebuah kasus yang meneliti faktor kecenderungan orang membeli dalam kasus *market basket analysis*, tidak perlu mengambil nama pelanggan, cukup dengan id pelanggan saja.

4. Transformasi data (*Data Transformation*)

   Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*. Beberapa metode *data mining* membutuhkan format data yang khusus sebelum bisa diaplikasikan. Sebagai contoh beberapa metode standar seperti analisis asosiasi dan *clustering* hanya bisa menerima *input* data kategorikal. Karenanya data berupa angka numerik yang berlanjut perlu dibagi-bagi menjadi beberapa interval. Proses ini sering disebut transformasi data.

5. Proses *mining*

   Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

6. Evaluasi pola (*pattern evaluation*)

   Untuk mengidentifikasi pola-pola menarik kedalam *knowledge based* yang ditemukan. Dalam tahap ini hasil dari teknik *data mining* berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai hipotesa ada beberapa alternatif yang dapat diambil seperti menjadikannya umpan balik untuk memperbaiki proses *data mining*, mencoba metode *data mining* lain yang lebih sesuai, atau menerima hasil ini sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

7. Presentasi pengetahuan (*knowledge presentation*),

   Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Tahap terakhir dari proses *data mining* adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisis yang didapat. Ada kalanya hal ini harus melibatkan orang-orang yang tidak memahami *data mining*. Karenanya presentasi hasil *data mining* dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses *data mining*. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan hasil *data mining* (Han, 2004).

Data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Perlu diingat bahwa kata *mining* sendiri berarti usaha untuk mendapatkan sedikit data berharga dari sejumlah besar data dasar. Karena itu *data mining* sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, statistik dan basis data. Beberapa teknik yang sering disebut dalam literatur *data mining* antara lain yaitu *association rule mining, clustering, klasifikasi, neural network, genetic algorithm* dan lain-lain.

2.2 Data *Preprocessing*

Sebelum data diolah dengan *data mining*, data perlu melalui tahap *preprocessing*. Tahap ini berhubungan dengan pemilihan dan pemindahan data yang tidak berguna (*data cleaning*), penggabungan sumber-sumber data (*data integration*), transformasi data dalam bentuk yang dapat mempermudah proses *(data transformation*), menampilkan data dalam jumlah yang lebih mudah dibaca (*data reduction*). Semuanya berasal dari data mentah (data transaksi) dan hasilnya akan menjadi data yang nantinya siap untuk diolah dengan data mining (Han, 2004).

2.3 *Association Rule Mining*

Association rule mining adalah teknik *mining* untuk menemukan aturan assosiatif antara suatu kombinasi item dalam suatu *data set* yang ditentukan (Han, Jiawei, Kamber, dan Micheline, 2004).

Association Rule Mining meliputi dua tahap (Kantardzic, 2003):

a. Mencari kombinasi yang paling sering terjadi dari suatu *itemset (frequent itemset)*.
b. *Generate Association Rule* dari *frequent itemset* yang telah dibuat sebelumnya.

Umumnya ada dua ukuran kepercayaan (*interestingness measure)* yang digunakan dalam menentukan suatu *association rule*, yaitu: (Han, 2004)

a. *Support*, suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item* atau *itemset* dari keseluruhan transaksi. Perhitungan nilai *support* ditunjukkan pada Persamaan (2.1).
b. *Confidence*, suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional*. Perhitungan nilai *confidence* ditunjukkan pada Persamaan (2.2).

2.4. Konsep Dasar *Association Rule Mining*

Ada sebuah transaksi D memiliki itemset J = {i1, i2, …, im}. Masing-masing transaksi memiliki identitas yaitu TID. Pada setiap TID yang dimiliki D, memiliki itemset T dimana $T \subseteq J$. A dikatakan sebuah transaksi D jika dan hanya jika $A \subseteq T$.

Association rule adalah bentuk implikasi A ⇒ B dimana A ⊂ J, B ⊂ J, dan A ∩ B= φ. Sebuah rule A ⇒ B memiliki *support* s, dimana s adalah persentase transaksi di D yang terdiri A ∪ B dengan probabilitas P(A ∪ B). Rule A ⇒ B memiliki *confidence* c, dimana c adalah persentase transaksi pada D yang meliputi A dan juga meliputi *frequent itemset* B (Han, Jiawei, Kamber, dan Micheline, 2004).

$$Support\ A \Rightarrow B = P(A \cup B) = \frac{jumlah\ transaksi\ mengandung\ A\ dan\ B}{total\ transaksi} \quad (2.1)$$

$$Confidence\ (A \Rightarrow B) = P(B \backslash A)) = \frac{jumlah\ transaksi\ mengandung\ A\ dan\ B}{Jumlah\ transaksi\ mengandung\ A} \quad (2.2)$$

Support merupakan suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item* atau *itemset* dari keseluruhan transaksi. Sedangkan *confidence* merupakan suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional*.

Aturan yang memenuhi kedua *minimum support threshold (min_sup) dan minimum confidence threshold (min_conf)* disebut *strong*. Dengan ketetapan, penulisan nilai *support* dan *confidence* antara 0% dan 100%.

Set of item dinamakan *itemset*. Sebuah *itemset* terdiri dari *k-items* adalah *k-itemset*. *Set* {komputer,*software*} adalah 2-*itemset*. *Itemset* memenuhi *minimum support* jika frekuensi kemunculan itemset lebih besar sama dengan perkalian *min_sup* dan total jumlah transaksi di D ≥ (min_sup* total transaksi)). Jumlah transaksi dibutuhkan *itemset* untuk memenuhi *minimum support* maka dari itu merujuk ke *minimum support count*. Jika *itemset* memenuhi *minimum support* maka disebut *frequent itemset*. Sekumpulan frequent *k-itemset* dilambangkan dengan Lk .

Setelah *frequent itemset* dari transaksi pada *database* D ditemukan, lanjut ke membuat *strong association rule* (*strong minimum support dan minimum confidence*), dimana kondisi probabilitas diekspresikan pada bagian *itemset*

support count seperti yang ditunjukkan pada Persamaan (2.2). (Han, Jiawei, Kamber, dan Micheline, 2004). Sebagai contoh, berdasarkan persamaan (2.2) *assosiation rule* yang dibentuk dari suatu itemset l={i1, i2, i5} adalah:

i1 /\ i2⇒ i5

i1 /\ i5⇒ i2

i2 /\ i5⇒ i1

i1⇒ i2 /\ i5

i2⇒ i1 /\ i5

i5⇒ i1 /\ i2

Karena *rules* dibentuk dari *frequent itemset*, masing-masing secara otomatis memenuhi *minimum support.*

2.5 *Market Basket Analysis*

Menurut Han Jiawei (2001), fungsi *Association Rules* seringkali disebut dengan "*Market Basket Analysis". Market basket analysis* adalah salah satu cara yang digunakan untuk menganalisis data penjualan dari suatu perusahaan. Proses ini menganalisis kebiasaan belanja konsumen dengan menemukan asosiasi antar *item-item* yang berbeda yang diletakkan konsumen dalam keranjang belanja. Hasil yang telah didapatkan ini nantinya dapat dimanfaatkan oleh perusahaan retail seperti toko atau swalayan untuk mengembangkan strategi pemasaran dengan melihat *item-item* mana saja yang sering dibeli secara bersamaan oleh konsumen.

Association rule yang didefinisikan pada basket data, digunakan untuk keperluan promosi, desain katalog, segmentasi konsumen dan target pemasaran. Secara tradisional, *association rule* digunakan untuk menemukan *trend* bisnis dengan menganalisa transaksi konsumen.

2.6 *Bottom-up* dan *Top-down*

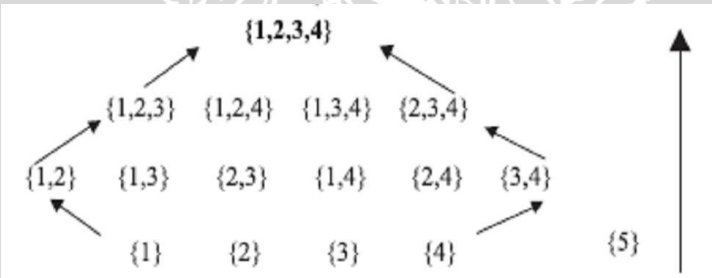
Pada umumnya untuk menemukan *frequent itemset* yang merupakan tahap awal dari *association rule*, dapat dilakukan dengan dua cara pendekatan, yaitu *bottom-up* dan *top-down search* (Lin,1998).

Tabel 2.1 Tabel Contoh Data

Transaksi	Itemset
20	1,2,3,4,5
50	1,3
100	1,2
200	1,2,3,4

Berdasarkan pada Table 2.1 akan diberikan gambaran mengenai cara kerja *Bottom-Up* dan *Top-Bottom* dengan nilai *min_support* = 2.

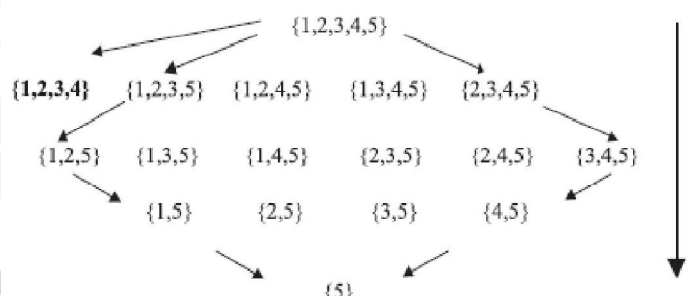
Bottom-Up Search



Gambar 2.3 *Bottom-Up Search* (Lin, 1998)

Metode *bottom-up* dimulai dari 1-*itemset* sampai dengan ke n-*itemset*. Metode ini akan menemukan *itemset* yang memenuhi *min_support* dan tidak meneruskan *itemset* yang tidak memenuhi *min_support*. Pada transaksi tersebut telah ditemukan *frequent itemset*nya adalah : {1}, {2}, {3}, {4}, {1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}, {1,2,3}, {1,2,4}, {1,3,4}, {2,3,4}, {1,2,3,4}.
Metode ini efektif apabila maksimal *itemset*nya relatif pendek.

Top-Down Search



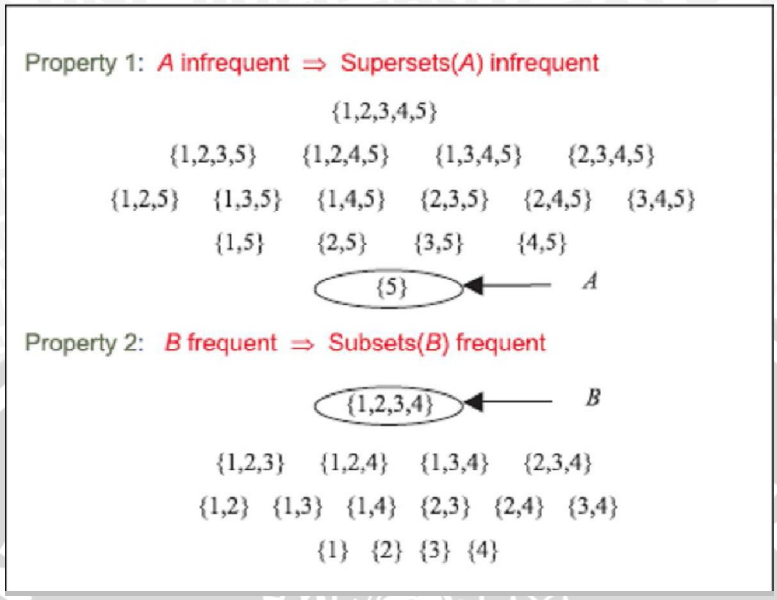
Gambar 2.4 *Top-Down Search* (Lin, 1998)

Metode *top-down* ini melakukan pencarian mulai dari n-*itemset* sampai ke 1-*itemset*. Ketika k-itemset ditemukan sebagai *infrequent*, maka semua *subset* (k-1) akan di-*generate* pada langkah selanjutnya. Sebaliknya, jika *k-itemset frequen*t, maka semua *subset* ini adalah *frequent* dan tidak perlu di-*generate* lagi sesuai dengan *property* 2. Kebalikan dari metode *bottom-up*, metode ini efektif dalam pencarian dari suatu maksimal *itemset* yang panjang. Dengan metode ini dapat menuruni banyak level hanya dengan sekali jalan. Baik *bottom-up* maupun *top-down search* merupakan metode *one-way search*.

2.7 *Downward Closure Property* dan *Upward Closure Property*

Kedua properti ini digunakan untuk melakukan *pruning candidate set* sehingga dapat mengurangi waktu penghitungan. Pendekatan untuk melakukan *prune* dengan cara seperti ini adalah sebagai berikut (Lin, 1998):

1. *Upward Closure Property* : Jika semua *infrequent itemset* ditemukan di *Bottom Up*, hasilnya dapat digunakan untuk mengeliminasi kandidat yang ditemukan pada arah *Top Down* yang telah ditemukan sejauh ini. Semua *superset* dari *itemset* ini pastilah *infrequent*.
2. *Downward Closure Property* : Jika semua *Maximal Frequent Itemset* ditemukan pada arah *Top Down*, maka *itemset* tersebut dapat digunakan untuk mengeliminasi banyak kandidat dari arah *Bottom Up*. Semua *subset* dari *itemset* ini pasti *frequent*.



Gambar 2.5 Dua Properti *Closure* (Lin, 1998)

Berdasarkan pada Tabel 2.1 dan Gambar 2.5 dapat dijelaskan sebagai berikut. Jika *itemset* {5} *infrequent*, maka *itemset* {2,5} juga *infrequent* karena nilai *support itemset* {2,5} sama dengan atau kurang dari *support itemset* {5}. Dan sebaliknya, jika *itemset* {1,2,3,4} *frequent*, maka *itemset* {1,2,3} juga *frequent*

2.6 Algoritma *Apriori*

Algoritma *apriori* (Agrawal, 1994) merupakan tipikal dari pendekatan *bottom up*. Algoritma ini menggunakan pendekatan iteratif, yang dikenal dengan *level-wise search*, dimana k-kelompok produk digunakan untuk mengeksplorasi (k+1)-kelompok produk atau (k+1)-*itemset*. Algoritma ini berdasarkan pada prinsip *Apriori*, yaitu jika suatu *itemset* merupakan *frequent itemset*, maka semua *subset*-nya akan berupa *frequent itemset*. Pembentukan *frequent itemset* dilakukan dengan mencari semua kombinasi item-item yang memiliki *support* di atas *min_support* yang telah ditentukan.

Proses pada algoritma ini membangkitkan *frequent itemset* per level, dimulai dari level 1-*itemset* sampai ke *longest itemset*, kandidat level yang baru

dibentuk dari *frequent itemset* yang ditemukan di level sebelumnya lalu menentukan nilai *support*-nya. Dibawah ini merupakan prosedur-prosedur yang digunakan dalam algoritma *Apriori*:

1. Algoritma *Apriori-gen*

1) Memanggil prosedur *join* untuk *generate* kandidat set

2) Memanggil prosedur *prune* untuk mendapatkan hasil akhir kandidat set

2. *Join* Prosedur

1) ***For*** *i from 1 to* $|L_k - 1|$

2) ***For*** *j from i+1 to* $|L_k|$

3) ***if*** $L_k.itemset_i$ *and* $L_k.itemset_j$ *have the same (k-1)-prefix*

4) $C_{k+1} := C_{k+1} \cup (L_k.itemset_i \cup L_k.itemset_j)$

5) ***else***

6) ***break***

Pada langkah ini dilakukan kombinasi dua *frequent k-itemset* dimana mempunyai *(k-1)prefix* yang sama untuk meng-*generate (k+1)-itemset* sebagai kandidat yang baru

3. *Prune* Prosedur

1) ***For all*** *itemsets c in* C_{k+1}

2) ***For all*** *k-subsets s of c*

3) ***If*** $s \in L_k$

4) ***Delete*** *c from* C_{k+1}

Pada langkah ini membuang *itemset* c dalam C_{k+1} yang beberapa *subset* ukuran *k* yang tidak berada dalam L_k. Dengan kata lain, *superset* dari *infrequent itemset* dihilangkan.

4. Algoritma *Apriori*

C_k: kandidat *itemset* dengan ukuran *k*

L_k: *frequent itemset* dengan ukuran *k*

D : data seluruh transaksi

1) $L_0 := \emptyset$; k:= 1;

2) C_1 := {{i} | i Ø I}

3) *Answer* := Ø

4) *while* $C_k \neq$ Ø

5) *read database and count supports for* C_k

6) L_k := *{frequent itemsets in* C_k*}*

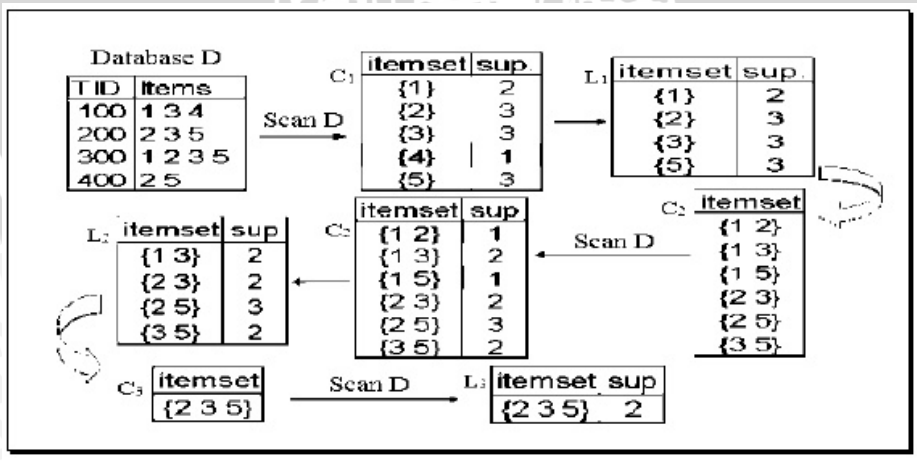
7) C_{k+1} := *Apriori-gen*(L_k)

8) *k* := *k*+1

9) *Answer* := *Answer* ∪ L_k

10) *return Answer*

Berdasarkan algoritma tersebut, yang dilakukan pertama kali yaitu menelusuri seluruh *record* di basis data transaksi dan menghitung *support* dari tiap *item*, yang merupakan kandidat *1-itemset*, C_1. Setelah itu L_1 dibangun dengan menyaring C_1 dengan *support count* yang lebih besar sama dengan *min_support* untuk dimasukkan ke dalam L_1. Untuk membangun L_2, algoritma *apriori* menggunakan prosedur *join* untuk menghasilkan C_2. Dari C_2, 2-*itemset* yang memiliki *support* yang lebih besar sama dengan *min_support* akan disimpan ke dalam L_2. Proses ini diulang sampai tidak ada lagi kemungkinan *k-itemset*. Contoh tahapan pembangkitan C_1, L_1, C_2, L_2, C_3, L_3 ditunjukkan pada Gambar 2.6.



Gambar 2.6 Ilustrasi Algoritma *Apriori* (Ikhsan, 2007)

Kelemahan dari algoritma *apriori* adalah proses *generate* kandidat yang sangat besar sekali sehingga membutuhkan waktu komputasi yang lama. Misal untuk ukuran 100 item data $\{a_1,a_2\ldots,a_{100}\}$, perlu dibangun $2^{100} \approx 10^{30}$kandidat.

2.8 *Sequential Pattern Mining*

Pada proses *mining* data yang dibutuhkan sangat besar, maka setiap obyeknya melakukan aksi yang berulang kali. Data yang dimasukkan merupakan sekumpulan data *sequences*. Dalam sebuah *sequence* merupakan sebuah transaksi. Di dalam satu kali transaksi terdiri dari beberapa item. Setiap transaksi berhubungan dengan waktu transaksi. Sebelum dilakukan proses mining, maka ditentukan periode transaksi dan *minimum support*. *Support* dari sebuah *sequential pattern* merupakan persentase dari *data sequences* yang mengandung suatu pola tertentu. *Sequential Pattern* adalah pola yang menggambarkan urutan waktu terjadinya peristiwa (Han, 2004). Dalam menentukan pola-pola tersebut dibutuhkan algoritma *data mining*.

2.9 Algoritma *Generalized Sequential Patterns*

Salah satu algoritma yang dapat memecahkan masalah *sequential pattern* adalah *Generalized Sequential Patterns* (GSP)**.** Algoritma GSP, atau dengan nama lain *apriori all*, adalah suatu algoritma yang dapat memproses dan menemukan semua pola sekuensial dan non sekuensial yang ada (Zaki,1997). Algoritma ini digunakan untuk membentuk aturan-aturan (*Association Rule dan Sequential Pattern Rule*) dari semua *frequent sequence pattern* yang telah ditemukan. Algoritma GSP didesain untuk data transaksi, dimana setiap pola merupakan kumpulan dari transaksi berupa *items* (Ahola, 2001). Algoritma ini bekerja menemukan semua pola sequensial yang sesuai dengan minimum *support* yang ditentukan, sehingga memakan waktu yang cukup besar dalam penggaliannya. Struktur dasar dari algoritma GSP sangat mirip dengan algoritma Apriori, kecuali bahwa GSP dengan urutan bukan *itemset* (Srikant, 1996).

Algoritma GSP dapat dilihat pada Gambar 2.7 dan Gambar 2.8

```
L1 := {frequent 1-item sequences};
k := 2; // k represents the pass number
while ( Lk-1≠Ø ) do begin
    Ck := New candidates of size k generated from Lk-1;
    forall sequences S ∈ D do begin
        Increment the count of all candidates in Ck that are
        contained in S.
    end
Lk := All candidates in Ck with minimum support.
k := k + 1;
end
        Answer := Uk Lk;
```

Gambar 2.7 Algoritma GSP (Srikant, 1996)

```
insert into Ck select p.litemset1, ..., p.litemsetk-1, q.litemsetk-1
from Lk-1 p, Lk-1 q
where p.litemset1 = q.litemset1, ...,
p.litemsetk-2 = q.litemsetk-2;
```

Gambar 2.8 Algoritma untuk meng-*generate* kandidat dari Lk-1 (Agrawal, 1995; Zaki, 1997)

Menemukan pola *sequential* adalah fungsi utama algoritma GSP. Pada saat proses menemukan pola, hal pertama yang dilakukan adalah menentukan kandidat-kandidat yang merupakan kumpulan *items* dari setiap *sequence*. Banyaknya *items* yang digunakan pada kandidat-kandidat akan bertambah setiap perulangan. Kandidat-kandidat akan berubah menjadi *frequent sequence* jika kandidat-kandidat tersebut memenuhi *minimum support*. Untuk menghitung *support* dari setiap kandidat-kandidat, maka dilakukan pengecekan setiap

sequence pada *database*. Jika pada satu *sequence* terdapat kandidat-kandidat, maka nilai *support* dari kandidat-kandidat tersebut akan bertambah. Kandidat yang lolos *minimum support* akan menjadi kandidat pada perulangan selanjutnya. Kerja algoritma GSP akan berhenti jika tidak ada lagi *frequent sequence* pada akhir fase atau tidak ada lagi kandidat untuk fase berikutnya. Sebagai contoh penggunaan algoritma GSP dapat dilihat pada Tabel 2.2 yaitu sebuah tabel transaksi.

Tabel 2.2 Tabel *Sequence Database* Transaksi Pembelian

Id	Sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)ab)(df)cb>
40	<eg(af)cbc>

Sumber : Jiawei & Kamber (2004)

Pada Tabel 2.2 dapat dijelaskan bahwa pada Id ke-10 terdapat *sequence* <a(abc)(ac)d(cf)>, sehingga *subsequence* untuk Id ke-10 adalah a, (abc), (ac), d, dan (cf). Ini berarti bahwa transaksi pertama adalah pembelian *item* a. Lalu pada transaksi selanjutnya membeli item a, b, dan c, lalu membeli *item* a dan c, selanjutnya membeli item d, dan transaksi terakhir membeli *item* c dan f. Sehingga dapat dikatakan bahwa setiap transaksi pembelian dapat disebut *subsequence*.

Jika ditentukan minimum support adalah 2, maka dapat dihitung *support* setiap *items* pada setiap *sequence*. Jika dalam satu *sequence* terdapat *item*, maka *support* dari *item* akan bertambah satu. Jika dalam satu *sequence* muncul *item* tertentu dua kali maka nilai *support* hanya ditambah satu. Hasil perhitungan nilai *support* untuk masing-masing *item* yang merupakan kandidat dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tabel Kandidat *Sequence*

Kandidat	Support
a	4
b	4
c	4
d	3
e	3
f	3
g	1

Pada Tabel 2.3 dapat dilihat kandidat dan *support* untuk masing-masing *item*. Proses tersebut dapat dikatakan pada *length*-1 kandidat. Karena telah ditentukan bahwa *minimum support* adalah 2, maka hanya ada enam kandidat yang didapatkan yaitu a, b, c, d, e, dan f. Untuk g tidak memenuhi kandidat karena hanya memiliki satu *support*.

Selanjutnya masing-masing kandidat yang memenuhi *minimum support* akan di–*join*-kan, sehingga akan menghasilkan kandidat baru. Proses ini dapat dikatakan pada *length*-2 kandidat karena satu kandidat terdapat dua *item*. Proses pembentukan kandidat baru pada *length*-2 kandidat dapat dilihat pada Tabel 2.4. dan Tabel 2.5.

Tabel 2.4 Tabel Kombinasi Kandidat

	a	b	c	d	e	f
a	aa	ab	ac	ad	ae	ae
b	ba	bb	bc	bd	be	bf
c	ca	cb	cc	cd	ce	cf
d	da	db	dc	dd	de	df
e	ea	eb	ec	ed	ee	ef
f	fa	fb	fc	fd	fe	ff

Tabel 2.5 Tabel Kombinasi Kandidat (disederhanakan)

	a	b	c	d	e	f
a		ab	ac	ad	ae	ae
b			bc	bd	be	bf
c				cd	ce	cf
d					de	df
e						ef

Pada Tabel 2.5 kandidat hasil kombinasi (*join*) yang muncul dua kali hanya dihitung satu kali. Kandidat yang memenuhi *minimum support* akan di-*join*-kan lagi dengan kandidat lainnya, sehingga membentuk *length*-3 kandidat. Proses ini dilakukan sampai tidak ada lagi kandidat atau *frequent sequence*. Hasil akhir dari proses algoritma GSP dapat dilihat pada Tabel 2.6.

Tabel 2.6 Hasil Akhir Algoritma GSP

Proses	Frequent Sequence
Length-5	<(bd)cba>
Length-4	<(bd)bc>…
Length-3	<abb><baa><aba>…
Length-2	<aa><ab><ba><bb>…<ff>
Length-1	<a><b><c><d><e><f>

Dari Tabel 2.6 dapat diketahui bahwa proses algoritma GSP mengalami perulangan sebanyak lima kali proses. Pada proses kelima telah ditemukan *frequent sequence* yaitu <(bd)cba>. Hal ini menggambarkan bahwa pada transaksi pertama membeli b dan d, maka pada transaksi kedua orang cenderung membeli c, transaksi ketiga cenderung membeli b, dan pada transaksi terakhir cenderung membeli a.

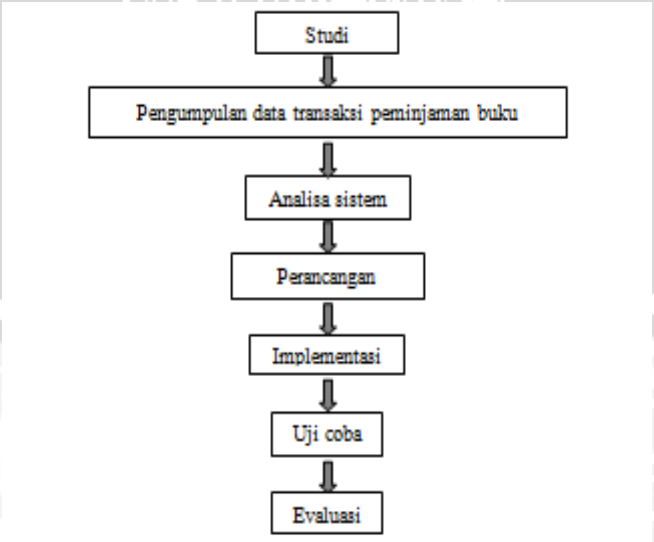
BAB III
METODE PENELITIAN DAN PERANCANGAN

Pada bab ini akan dijelaskan tentang metode dan tahap perancangan dalam analisis data transaksi peminjaman buku dengan metode *Association Rule* menggunakan algoritma *Generalized Sequential Patterns*.

Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Melakukan studi literatur mengenai *association rule* dan algoritma *Generalized Sequential Patterns.*
2. Mengumpulkan data-data yang diperlukan dalam penelitian ini yaitu data transaksi peminjaman buku di perpustakaan.
3. Menganalisa dan melakukan perancangan sistem untuk menggali *association rule* dari data transaksi peminjaman buku di perpustakaan.
4. Mengimplementasikan rancangan yang dilakukan pada tahap sebelumnya menjadi sebuah perangkat lunak untuk analisa pola *association rule* dari data transaksi peminjaman buku di perpustakaan.
5. Melakukan uji coba terhadap perangkat lunak menggunakan data transaksi peminjaman buku perpustakaan.
6. Mengevaluasi hasil analisa yang dilakukan oleh sistem.

Langkah – langkah penelitian ini dilihat pada Gambar 3.1.



Gambar 3.1 Langkah-langkah Penelitian

3.1. Studi Literatur

Penelitian ini dimulai dengan studi literatur yaitu mempelajari literatur atau sumber-sumber yang berkaitan. Teori-teori tentang *data mining*, *association rule*, dan algoritma GSP yang digunakan sebagai dasar penelitian. Studi literatur dilakukan dari sumber buku, jurnal maupun *browsing* internet.

3.2. Pengumpulan Data

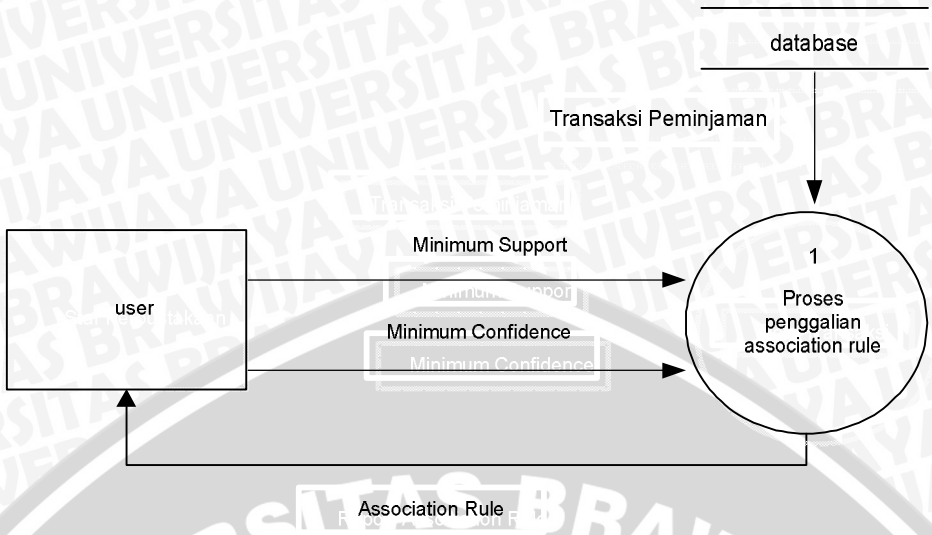
Data yang digunakan dalam penelitian ini adalah *database* perpustakaan Fakultas Hukum Universitas Brawijaya. Tabel yang digunakan dalam penelitian ini adalah Tabel Transaksi Peminjaman dan Tabel Klasifikasi Buku, sehingga tidak semua tabel yang ada pada *database* tersebut dipakai, disesuaikan dengan kebutuhan dalam penelitian ini.

3.3. Analisa Sistem danPerancangan Sistem

3.3.1 Deskripsi Sistem

Sistem yang dibangun adalah untuk menemukan asosiasi antar *item* dari data transaksi peminjaman buku pada perpustakaan. Dari asosiasi yang dihasilkan akan diketahui keterkaitan *item* yang terdapat dalam transaksi peminjaman buku. Pada sistem ini akan menghasilkan *frequent itemset*, yaitu *itemset* yang memenuhi *minimum support*. Dari *frequent itemset* yang telah dihasilkan akan digunakan untuk menemukan *association rule*. *Association rule* yang terbentuk adalah *frequent itemset* yang memenuhi *minimum confidence*.

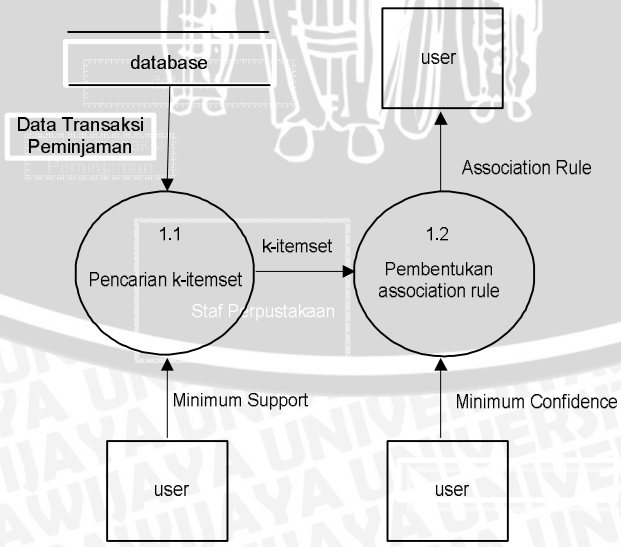
Secara garis besar sistem yang dibangun dapat dilihat pada *context diagram* Gambar 3.2.



Gambar 3.2 *Context Diagram*

Berdasarkan *context diagram* pada Gambar 3.2 dapat dijelaskan bahwa proses pengenalan pola transaksi membutuhkan tiga data masukan, yaitu data transaksi peminjaman buku, *minimum support* dan *minimum confidence*. Data *minimum support, minimum confidence* dan transaksi peminjaman dimasukkan oleh *user*. *Output* data dari proses tersebut berupa laporan hasil dalam bentuk *association rule*, yang diberikan kepada *user*.

Conteks diagram pada Gambar 3.2 dapat dijabarkan dalam *Data Flow Diagram* seperti pada Gambar 3.3.



Gambar 3.3 *Data Flow Diagram* (DFD) level 1

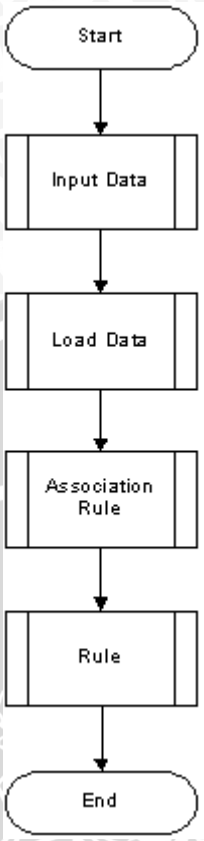
Pada Gambar 3.3 *Data Flow Diagram* level 1, proses pertama adalah pencarian k-*itemset*. Pada proses ini membutuhkan masukan data transaksi peminjaman dan *minimum support.* Setelah ditemukan k-*itemset* selanjutnya sistem akan memproses pembentukan *association rule.* Pada proses ini dibutuhkan masukan *minimum confidence* dari *user*. *Output* dari proses akhir ini adalah *association rule* yang diterima oleh *user*.

3.3.2 Perancangan Pembuatan Sistem

Untuk pembuatan sistem yang menerapkan metode *association rule dengan algoritma* GSP dibutuhkan perancangan sistem yang terdiri dari perancangan proses, perancangan struktur data sistem, dan perancangan *interface.*

3.3.2.1 Perancangan Proses

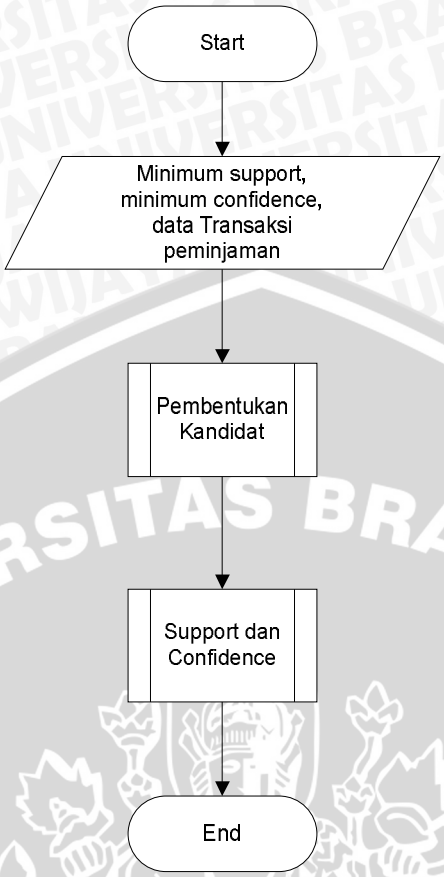
Untuk menentukan *data mining* pada penelitian ini digunakan metode *association rule* menggunakan algoritma *Generalized Sequential Pattens.* Sebelum sistem dibuat perlu dilakukan pembuatan perancangan proses. Secara umum perancangan proses dapat dilihat pada Gambar 3.4. Sistem membutuhkan data masukan berupa *minimun support*, *minimum confidence*, data transaksi peminjaman buku. Kemudian sistem akan melakukan proses *load* data, *association rule* dan pengujian.



Gambar 3.4 Gambaran Sistem Secara Umum

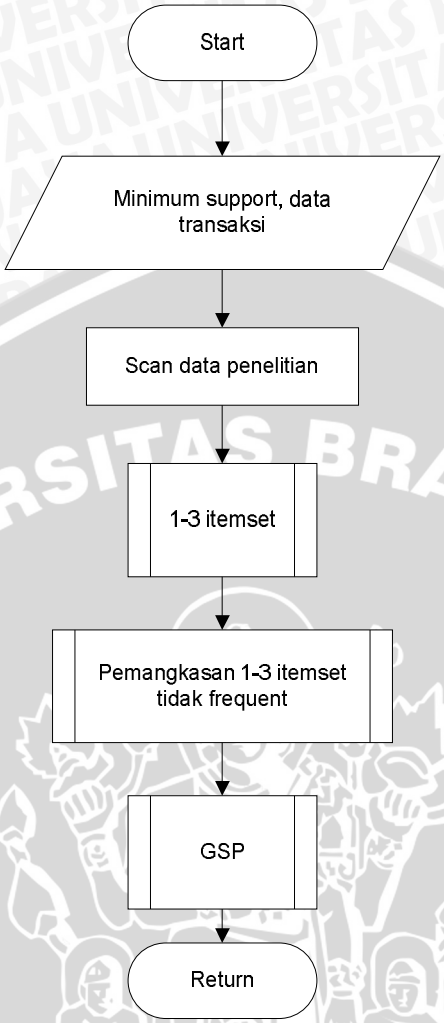
Proses *load* data adalah proses pengambilan data transaksi peminjaman dari *database* sesuai periode data yang dikehendaki.

Pada *association rule* yaitu proses menemukan aturan assosiatif antara suatu kombinasi *item* dalam suatu *data set* yang ditentukan. Setelah diperoleh data awal melalui penelusuran seluruh *record* di basis data transaksi dan menghitung *support* dari tiap *item*, yang merupakan kandidat *1-itemset*, C_1, selanjutnya dilakukan *join* untuk menghasilkan C_2 dan C_3. Kandidat yang terbentuk paling tinggi pada proses C_3, hal ini disebabkan bahwa peminjaman buku oleh pengguna hanya dibatasi paling banyak tiga buku. Pada proses *association rule* ini akan diketahui kategori buku yang berelasi yang paling banyak dipinjam.



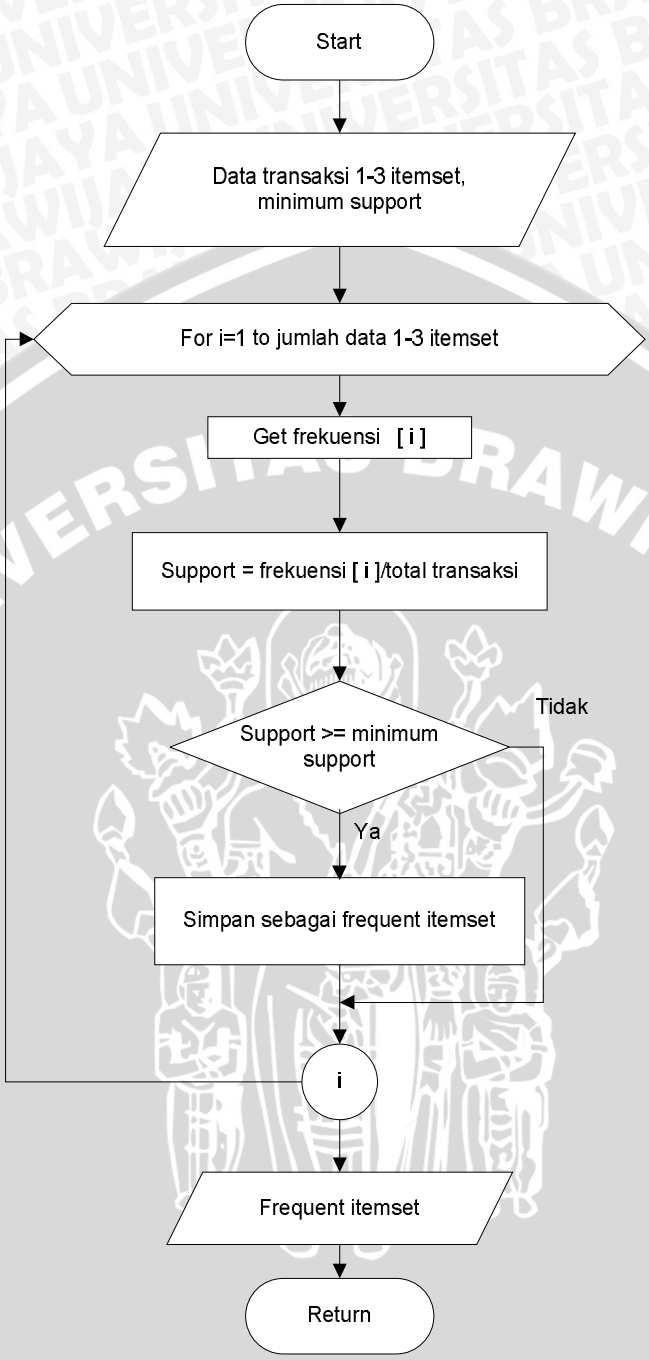
Gambar 3.5 *Flowchart* Proses Pembentukan Kandidat

Proses pembentukan kandidat merupakan yaitu proses penelusuran seluruh *record* di basis data transaksi dan menghitung *support* dari tiap *item*, yang merupakan kandidat *1-itemset*, C_1. Setelah itu L_1 dibangun dengan menyaring C_1 dengan *support count* yang lebih besar sama dengan *min_support* untuk dimasukkan ke dalam L_1. Selanjutnya dilakukan kombinasi 2-*itemset* untuk membentuk C_2. Setelah C_2 terbentuk, dilakukan penyaringan berdasarkan *support count* yang lebih besar sama dengan *min_support* untuk dimasukkan. Proses selanjutnya adalah pembentukan kandidat ke 3 dengan melakukan kombinasi 3-*itemset*. Jika ditemukan maka terbentuklah C_3. Setelah itu L_3 dibangun dengan menyaring C_3 dengan *support count* yang lebih besar sama dengan *min_support* untuk dimasukkan ke dalam L_3.



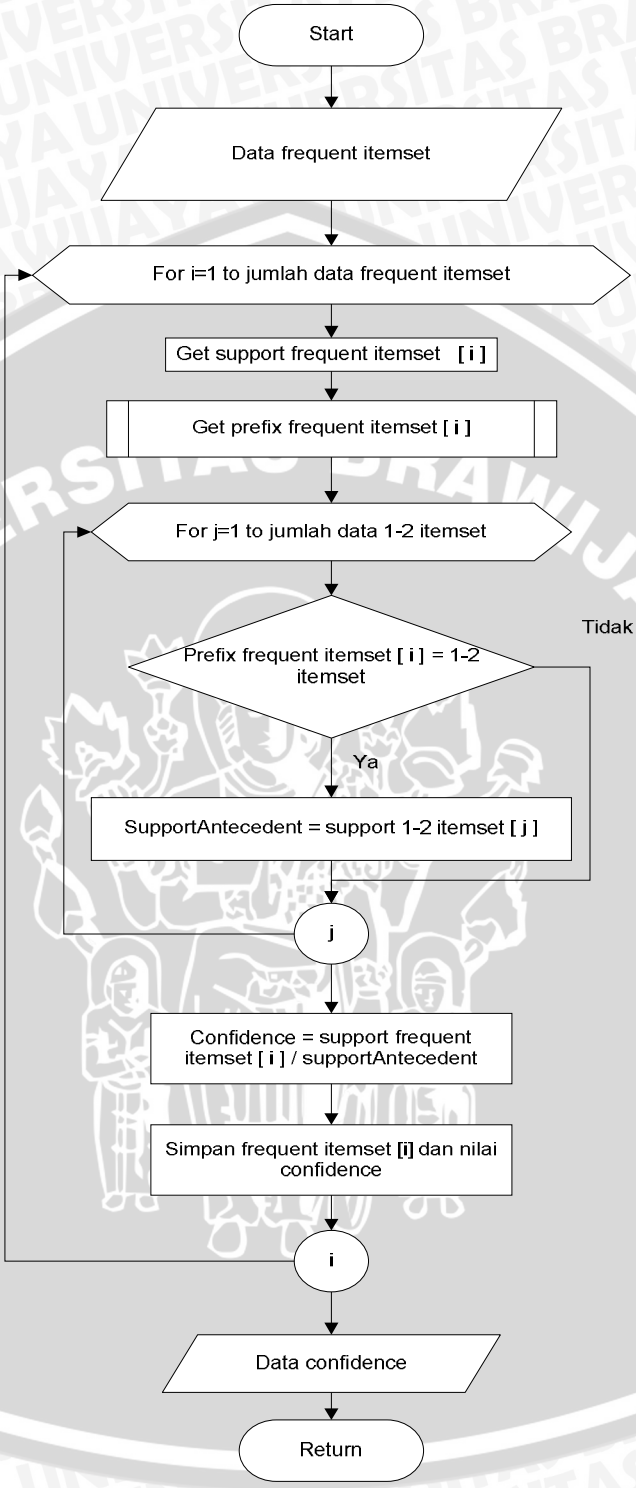
Gambar 3.6 *Flowchart* Proses GSP

Pada Gambar 3.6 dijelaskan bahwa setelah dilakukan proses penelusuran data transaksi akan ditemukan kandidat 1-*itemset,* 2-*itemset* dan 3-*itemset* dari setiap data transaksi. Proses ini membutuhkan data masukan berupa *minimum support*. Proses selanjutnya dilakukan pemangkasan 1-3 *itemset.*



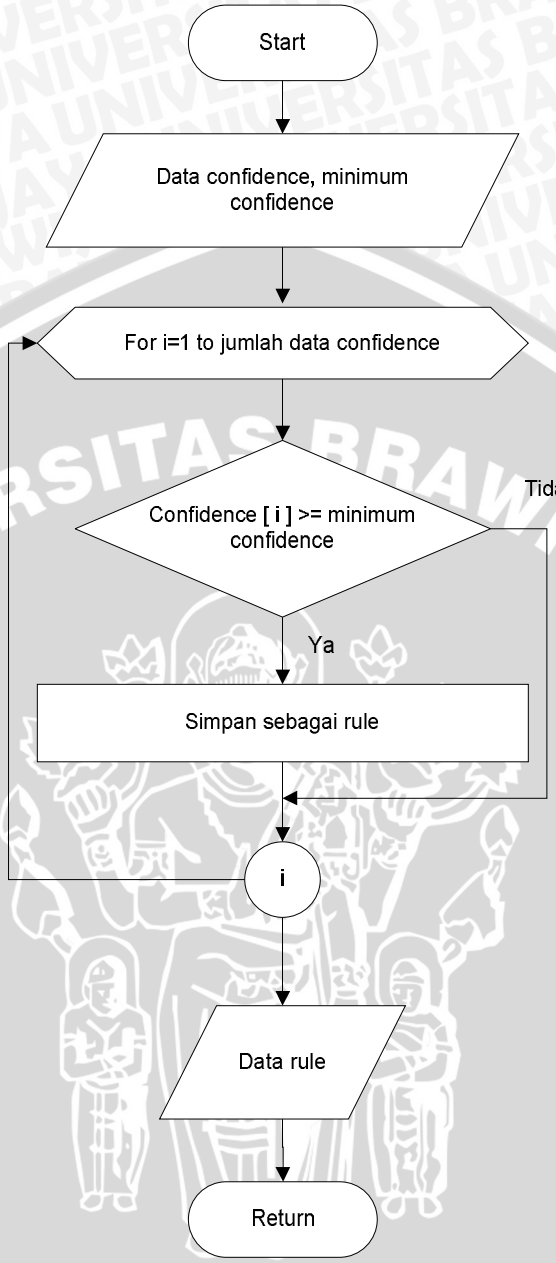
Gambar 3.7 *Flowchart* Proses Pemangkasan 1-3 *itemset*

Proses pemangkasan 1-3 *itemset* sebagaimana pada Gambar 3.7, digunakan untuk mendapatkan data 1-*itemset*, 2-*itemset* dan 3-*itemset* yang nilai *support*-nya memenuhi nilai minimum *support*. Selanjutnya akan disimpan sebagai data yang *frequent itemset*.



Gambar 3.8 Proses Perhitungan *Confidence*

Pada Gambar 3.8 dijelaskan proses perhitungan *confidence* yaitu untuk menghitung nilai *confidence* setiap *frequent itemset*.



Gambar 3.9 *Flowchart* Proses Seleksi *Rule*

Proses seleksi *rule* digunakan untuk menyeleksi *frequent itemset* yang dapat dijadikan *rule.* Pada proses ini dicari nilai *confidence* yang lebih besar atau sama dengan nilai minimum *confidence*. Hasil dari seleksi ini disimpan sebagai *rule*.

3.4. Perancangan Struktur Data Sistem

Data yang digunakan dalam sistem ini adalah data dari transaksi peminjaman buku. Selanjutnya akan disimpan dalam *database*. *Database* berupa kumpulan tabel yang berfungsi untuk tempat menyimpan dan mendapatkan data yang diperlukan. Tabel-tabel yang digunakan adalah sebagai berikut :

1. Tabel Transaksi

   Tabel transaksi adalah tabel yang digunakan untuk data penelitian yang meliputi data transaksi peminjaman buku. Data yang dicatat dapat dilihat pada Tabel 3.1

Tabel 3.1 Tabel Transkasi

Field	***Type***	***Size***	**Keterangan**
kode_buku	*Varchar*	255	Kode Buku
Id_pinjam	*Varchar*	255	No transaksi
Kode_klas	*Int*	4	Id Kategori
nim	*Varchar*	255	Id peminjam
Id_temp	*int*	4	No. Nota

2. Tabel Klasifikasi

   Tabel klasifikasi digunakan untuk menentukan kategori atau klasifikasi buku. Atribut yang ada dalam tabel klasifikasi dapat dilihat dalam Tabel 3.2.

Tabel 3.2 Tabel Klasifikasi

Field	***Type***	***Size***	**Keterangan**
Kode_klas	*Int*	4	Id Kategori
klasifikasi	*Varchar*	255	Jenis/kategori/klasifikasi buku

Gambar 3.10 Diagram Relasi Basis Data

Berdasarkan Diagram Relasi Basis Data pada Gambar 3.11, tabel yang berelasi adalah tabel transaksi dan tabel klasifikasi. Tiap transaksi dapat melakukan banyak peminjaman kategori buku. Tabel transaksi terhadap tabel klasifikasi memiliki kardinalitas relasi satu ke banyak.

3.5 Perancangan *Interface*

Perancangan *interface* terdiri dari beberapa bagian (Gambar 3.10) yaitu:

A : Periode Transaksi, digunakan untuk menentukan tanggal awal dan tanggal akhir yang akan diolah.

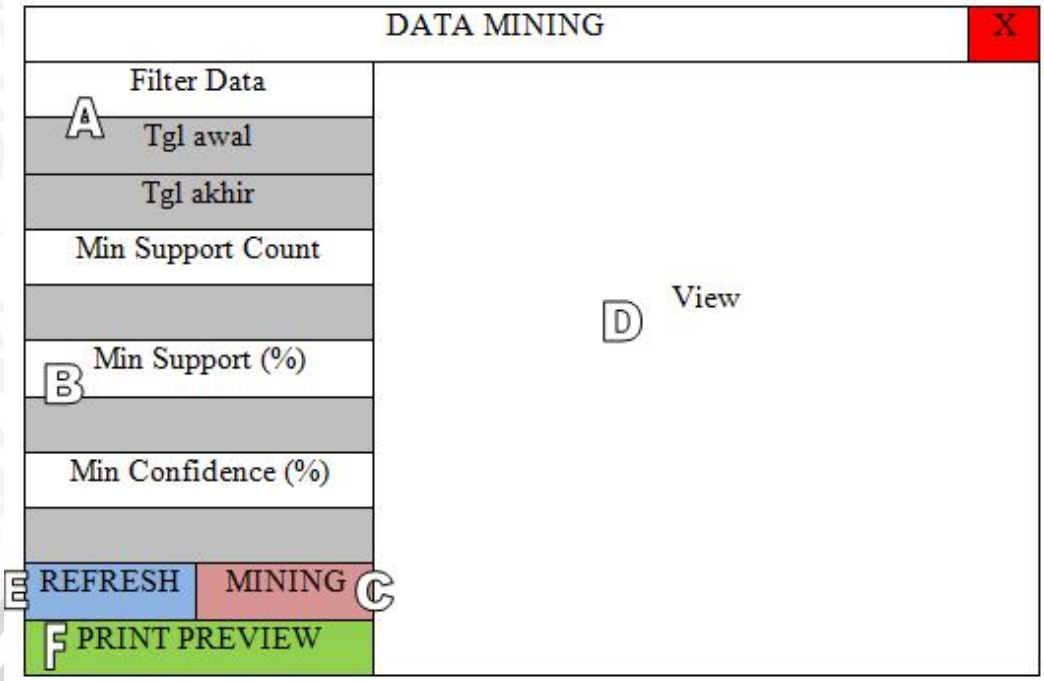
B : *Treshold*, digunakan untuk menentukan berapa *minimum support (%)* dan *minimum confidence (%)* yang diinginkan.

C: Proses Data Mining, digunakan untuk mengeksekusi program atau pemrosesan pada sistem.

D : *view result,* menampilkan hasil *association rule* dalam bentuk tabel.

E : *Refresh,* untuk mengembalikan ke posisi menu utama dan membersihkan layar.

F : *Print Preview,* untuk mencetak tampilan di layar



Gambar 3.11 Rancangan *Form Association Rule*

3.5 Penghitungan Manual

Sampel data untuk perhitungan manual diambil sebanyak 37 data *record* transaksi. *Sampel* data ini seperti terlihat pada Tabel 3.3.

Tabel 3.3 Sampel Data Transaksi

Nim	Tanggal	Kode Buku	Kode Gol Klasifikasi
0910110196	02/04/2012	18503753	344
0910113142	02/04/2012	11201180	346
0910113142	02/04/2012	10700095	346
0910113151	02/04/2012	10801761	346
0910113151	02/04/2012	18903432	343
105010113111007	02/04/2012	11201656	342
115010113111007	02/04/2012	11200098	340
115010113111007	02/04/2012	18601897	344
115010113111008	02/04/2012	11200177	340
115010113111008	02/04/2012	18413867	345
105010101121009	04/04/2012	11100441	345
105010101121009	04/04/2012	19701345	346
115010102111007	04/04/2012	18404528	347
115010102111007	04/04/2012	11100053	341

Nim	Tanggal	Kode Buku	Kode Gol Klasifikasi
0910110195	05/04/2012	11201208	343
0910110195	05/04/2012	10804288	346
115010107111009	05/04/2012	19800298	340
105010107111007	05/04/2012	10900368	340
105010107111007	05/04/2012	10900600	342
0910113142	06/04/2012	10600028	343
0910113142	06/04/2012	10700545	347
115010113111007	09/04/2012	19800298	340
115010113111007	09/04/2012	18500437	345
115010113111007	09/04/2012	18800650	346
0910113149	11/04/2012	10806085	340
0910113149	11/04/2012	18606978	340
115010109111007	11/04/2012	11100448	340
115010109111007	11/04/2012	10700742	340
0910113197	15/04/2012	19303568	346
0910113197	15/04/2012	11201286	340
105010113111004	19/04/2012	18804936	341
105010109111008	26/04/2012	17600795	346
105010109111008	26/04/2012	11100507	345
105010109111008	26/04/2012	10806136	346
115010107121009	26/04/2012	19600451	346
105010113111004	30/04/2012	18602360	341
105010113111004	30/04/2012	10700272	346

Tabel 3.4 Kode Klasifikasi

Kode Klasifikasi	Klasifikasi
340	Ilmu Hukum
341	Hukum Internasional
342	Hukum Konstitusional dan Administratif
343	Aneka Ragam Hukum Publik
344	Hukum Sosial
345	Hukum Pidana
346	Hukum Perdata
347	Hukum Acara Perdata dan Pengadilan
348	Undang-undang, Peraturan-peraturan, Perkara-perkara
349	Hukum Negara dan Bangsa Tertentu

Setelah dilakukan *preprocessing* terhadap golongan klasifikasi pada transaksi peminjaman, yaitu jika kode id sama dan kode klasifikasi yang sama dianggap satu kemunculan, dapat dilihat pada Tabel 3.5.

Tabel 3.5 Data Transaksi *Preprocessing* Klasifikasi *Item*

Nim	Tanggal	Kode Gol Klasifikasi
0910110196	02/04/2012	344
0910113142	02/04/2012	346
0910113151	02/04/2012	346
0910113151	02/04/2012	343
105010113111007	02/04/2012	342
115010113111007	02/04/2012	340
115010113111007	02/04/2012	344
115010113111008	02/04/2012	340
115010113111008	02/04/2012	345
105010101121009	04/04/2012	345
105010101121009	04/04/2012	346
115010102111007	04/04/2012	347
115010102111007	04/04/2012	341
0910110195	05/04/2012	343
0910110195	05/04/2012	346
115010107111009	05/04/2012	340
105010107111007	05/04/2012	340
105010107111007	05/04/2012	342
0910113142	06/04/2012	343
0910113142	06/04/2012	347
115010113111007	09/04/2012	340
115010113111007	09/04/2012	345
115010113111007	09/04/2012	346
0910113149	11/04/2012	340
115010109111007	11/04/2012	340
0910113197	15/04/2012	346
0910113197	15/04/2012	340
105010113111004	19/04/2012	341
105010109111008	26/04/2012	345
105010109111008	26/04/2012	346
115010107121009	26/04/2012	346
105010113111004	30/04/2012	341
105010113111004	30/04/2012	346

Selanjutnya dilakukan transformasi data hasil *preprocessing*, yaitu data diubah ke bentuk sesuai kebutuhan, seperti terlihat pada Tabel 3.6.

Tabel 3.6 Hasil Transformasi Data Transaksi

Nim	Tanggal	Kode Gol Klasifikasi
0910110196	02/04/2012	344
0910113142	02/04/2012	346
0910113151	02/04/2012	343,346
105010113111007	02/04/2012	342
115010113111007	02/04/2012	340,344
115010113111008	02/04/2012	340,345
105010101121009	04/04/2012	345,346
115010102111007	04/04/2012	341,347
0910110195	05/04/2012	343,346
115010107111009	05/04/2012	340
105010107111007	05/04/2012	340,342
0910113142	06/04/2012	343,347
115010113111007	09/04/2012	340,345,346
0910113149	11/04/2012	340
115010109111007	11/04/2012	340
0910113197	15/04/2012	340,346
105010113111004	19/04/2012	341
105010109111008	26/04/2012	345,346
115010107121009	26/04/2012	346
105010113111004	30/04/2012	341,346

Dari hasil transformasi data transaksi akan dilakukan penelusuran seluruh *record* dan menghitung *support* dari tiap *item*, yang merupakan kandidat *1-itemset*, C_1, bisa dilihat pada Tabel 3.7.

Tabel 3.7 Kandidat Pertama (C1)

Kode Gol Klasifikasi	Frekuensi
340	8
341	3
342	2
343	3
344	2
345	4
346	9
347	2
348	0
349	0

Untuk membangun L_1 dari hasil menyaring C_1, *user* harus memasukkan

min_support count. *Min_support count* ini ditentukan sendiri oleh *user* sesuai dengan keinginannya. Misalnya ditentukan *min_support count = 2*, selanjutnya dilakukan proses mining yaitu kandidat yang nilainya kurang dari 2 akan dihapus. Hasil dari proses ini adalah dibentuknya L_1, seperti terlihat pada Tabel 3.8.

Tabel 3.8 L1

Itemset	Frekuensi
340	8
341	3
342	2
343	3
344	2
345	4
346	9
347	2

Untuk membangun L_2, algoritma GSP menggunakan prosedur *join* untuk menghasilkan C_2.

Tabel 3.9 Kandidat Kedua (C2)

Itemset	Frekuensi
340,341	0
340,342	1
340,343	0
340,344	1
340,345	2
340,346	2
340,347	0
341,342	0
341,343	0
341,344	0
341,345	0
341,346	1
341,347	1
342,343	0
342,344	0
342,345	0
342,346	0
342,347	0
343,344	0
343,345	0

Itemset	Frekuensi
343,346	2
343,347	1
344,345	0
344,346	0
344,347	0
345,346	3
345,347	0
346,347	0

Dari C_2, 2-*itemset* yang memiliki *support* yang lebih besar sama dengan *min_support* =2 akan disimpan ke dalam L_2, sedangkan yang kurang dari *min_support* akan dihapus, bisa dilihat pada Tabel 3.10.

Tabel 3.10 L2

Itemset	Frekuensi
340,345	2
340,346	2
343,346	2
345,346	3

Untuk membangun L_3, maka masing-masing itemset akan di-*join*-kan untuk menghasilkan C_3. Seperti terlihat pada Tabel 3.11.

Tabel 3.11 Kandidat Ketiga (C3)

Itemset	Frekuensi
340,345,346	1

Jika ditentukan *min_support* =2, maka L_3 tidak bisa dibentuk karena tidak ditemukan kandidat dengan 3-*itemset* yang lebih besar atau sama dengan 2.

Dari hasil tersebut yang digunakan adalah kandidat dengan 2-*itemset* (L_2) seperti pada Tabel 3.10. Selanjutnya dilakukan perhitungan nilai *support* dan *confidence*.

Tabel 3.12 Hasil Perhitungan *support* dan *confidence*

Frequent Itemset	**Frekuensi**	***Support***		***Confidence***	
340,345	2	0,10	10%	0,25	25%
340,346	2	0,10	10%	0,25	25%
343,346	2	0,10	10%	0,67	67%
345,346	3	0,15	15%	0,75	75%

Support 340,342= *Frekuensi* (340,342)/jumlah transaksi = 2/20

Confidence 340,342= *Frekuensi* (340,342)/*Frekuensi* (340) = 2/8

Support 340,345= *Frekuensi* (340,345)/jumlah transaksi = 2/20

Confidence 340,345= *Frekuensi*(340,345)/*Frekuensi* (340) = 2/8

Support 340,346= *Frekuensi* (340,346)/jumlah transaksi = 2/20

Confidence 340,346= *Frekuensi*(340,346)/*Frekuensi* (340) = 2/8

Support 343,346= *Frekuensi* (343,346)/jumlah transaksi = 2/20

Confidence 343,346= *Frekuensi*(343,346)/*Frekuensi* (343) = 2/3

Support 345,346= *Frekuensi* (345,346)/jumlah transaksi = 3/20

Confidence 345,346= *Frekuensi*(345,346)/*Frekuensi* (345) = 3/4

Berdasarkan Tabel 3.12 dapat dilihat bahwa yang memiliki nilai minimum *support* dan nilai minimum *confidence* adalah *frequent itemset* <(345),(346)> yaitu nilai minimum *support* sebesar 15% dan nilai minimum *confidence* sebesar 75%. Dengan demikian yang akan dibangkitkan menjdi *rule* adalah <345>, sehingga *rule* yang terbentuk adalah jika meminjam buku Hukum Pidana maka meminjam juga buku Hukum Perdata, seperti terlihat pada Tabel 3.13.

Tabel 3.13 *Rule* yang Terbentuk

Rule	Keterangan
$345 \rightarrow 346$	Jika meminjam Hukum Pidana maka meminjam juga Hukum Perdata

3.6. Perancangan Uji Coba

Uji coba sistem merupakan penerapan algoritma GSP untuk data transaksi peminjaman buku untuk memperoleh hasil analisa yang dihasilkan sistem. Tujuan dari uji coba ini adalah untuk mengetahui pengaruh nilai *minimum support* dan nilai *minimum confidence* yang digunakan terhadap jumlah *rule* yang dihasilkan, dapat dilihat pada Tabel 3.14.

Parameter yang digunakan untuk uji pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan adalah data periode waktu. Periode waktu yang digunakan adalah 1 bulan, 3 bulan dan 5 bulan.

Nilai *minimum support* yang digunakan pada uji pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang diterapkan adalah 2% sampai 10% tiap parameter dengan nilai *minimum confidence* 10% sampai 50%.

Tabel 3.14 Tabel Uji Pengaruh Nilai *Minimum Support* dan Nilai *Minimum Confidence* terhadap Jumlah *Rule* yang Dihasilkan

Periode	*Minimum Support* (%)	*Minimum Confidence* (%)	Jumlah *Rule* yang dihasilkan
1 bulan			
3 bulan			
5 bulan			

BAB IV
IMPLEMENTASI

4.1 Lingkungan Implementasi

Lingkungan impelementasi yang digunakan dalam skripsi ini adalah lingkungan perangkat keras dan lingkungan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pembentukan sistem penentuan pola pada data transaksi buku ini adalah sebagai berikut :

1. Prosesor Intel(R) Celeron(R) CPU 1007U @1.50 GHz.
2. RAM 2048 MB
3. *Harddisk* dengan kapasitas 320 GB
4. Monitor
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pembangunan sistem penentuan pola pada data transaksi buku ini adalah sebagai berikut :

1. Sistem Operasi *Microsoft Windows 7 Home Premium*
2. *Microsoft Visual Basic 6.0* sebagai *software development* dalam mengembangkan aplikasi.
3. *MySQL Server* untuk pengaksesan *database.*

4.2 Implementasi Perangkat Lunak

Sebagaimana dijelaskan pada Bab III, aplikasi untuk *association rule* dengan menggunakan algoritma GSP ini terdiri dari tiga proses, yaitu proses *input* data yang merupakan proses *input* data transaksi peminjaman buku, *input minimum support* dan *minimum confidence* yang dilakukan oleh *user*. Proses kedua adalah *load* data yaitu pengambilan data transaksi yang ada pada *database*

transaksi. Proses ketiga adalah proses penemuan pola GSP. Hasil dari proses penemuan pola GSP adalah *association rule.*

Pada tampilan utama, *user* diminta untuk menentukan batasan tanggal yang akan diproses serta *minimum support* dan *minimum confidence* ke dalam sistem. Proses dimulai saat *user* menekan tombol proses mining.

4.2.1 Tahap *Load* Data

```
Sql RsTrans, "SELECT
DISTINCT(id_temp),transaksi.nim,transaksi.tanggal FROM
transaksi WHERE tanggal >= '" & _
Format(Me.DTPicker1(0).Value, "yyyy-MM-dd") & "' AND tanggal
<= '" & Format(Me.DTPicker1(1).Value, "yyyy-MM-dd") & "'"

no = 0
rt = ""
For i = 1 To RsTrans.RecordCount
Set Parent = Record.Childs.Add
no = no + 1
Set Item = Parent.AddItem(no)
Set Item = Parent.AddItem(RsTrans.Fields("nim").Value)
Set Item =
Parent.AddItem(Format(RsTrans.Fields("tanggal").Value, "dd-
MM-yyyy"))
Sql RsTrans1, "SELECT DISTINCT(kode_klas) as kode_klas FROM
transaksi WHERE id_temp=" & RsTrans.Fields("id_temp").Value &
""
rt = ""
For j = 1 To RsTrans1.RecordCount
Db.Execute "INSERT INTO temp (nim,tanggal,kode_klas,id_temp)
VALUES ('" & RsTrans.Fields("nim").Value & "','" & _
RsTrans.Fields("tanggal").Value & "','" &
RsTrans1.Fields("kode_klas").Value & "','" &
RsTrans.Fields("id_temp").Value & "')"
If j > 1 Then
rt = rt & "," & RsTrans1.Fields("kode_klas").Value

Else
rt = RsTrans1.Fields("kode_klas").Value
End If
RsTrans1.MoveNext
Next j
Set Item = Parent.AddItem(rt)

RsTrans.MoveNext
Next i
```

Source Code 4.1 *Load Data*

Pada *source code* 4.1 fungsi *query "Sql RsTrans, "SELECT DISTINCT(id_temp),transaksi.nim,transaksi.tanggal FROM transaksi""*

untuk mengambil *items* yang sama berdasarkan *id_temp* ditampilkan dalam 1 *items*, sedangan fungsi *query "Sql RsTrans1, "SELECT * FROM transaksi WHERE id_temp=" & RsTrans.Fields("id_temp").Value & ""* melakukan pencarian *kode_klas* didalam tabel transaksi peminjaman buku berdasarkan *field id_temp.*

4.2.2 Tahap Pembentukan *Association Rule*

Prosedur *Preprocessing* Klasifikasi Item

```
Sql RsTrans1, "SELECT DISTINCT(kode_klas) as kode_klas FROM
transaksi WHERE id_temp=" & RsTrans.Fields("id_temp").Value &
""
rt = ""
For j = 1 To RsTrans1.RecordCount
Db.Execute "INSERT INTO temp (nim,tanggal,kode_klas,id_temp)
VALUES ('" & RsTrans.Fields("nim").Value & "','" & _
RsTrans.Fields("tanggal").Value & "','" &
RsTrans1.Fields("kode_klas").Value & "','" &
RsTrans.Fields("id_temp").Value & "')"
```

Source Code 4.2 Seleksi Klasifikasi Item

Pada *source code* 4.2 *query* "Sql RsTrans1, "SELECT DISTINCT(kode_klas) as kode_klas FROM transaksi WHERE id_temp=" & RsTrans.Fields("id_temp").Value & """ berfungsi untuk menyeleksi transaksi pada kategori yang sama. Sehingga jika dalam transaksi pada pengguna yang sama meminjam buku dengan kategori yang sama, maka hanya dihitung satu dan disimpan pada tabel temp.

Prosedur Pembentukan Kandidat Pertama

```
Sql RsTrans1, "SELECT COUNT(kode_klas) AS jml FROM temp WHERE
kode_klas=" & RsTrans.Fields("kode").Value & ""
jm = 0
For j = 1 To RsTrans1.RecordCount
jm = RsTrans1.Fields("jml").Value
RsTrans1.MoveNext
Next j

If jm >= Me.Spin1.Value Then
no = no + 1
Set Parent = Record.Childs.Add
Set Item = Parent.AddItem(no)
Set Item = Parent.AddItem(RsTrans.Fields("kode").Value)
Set Item = Parent.AddItem(jm)
End If

Db.Execute "INSERT INTO c1 (kode_klas,frekuensi) VALUES (" &
RsTrans.Fields("kode").Value & "," & jm & ")"

RsTrans.MoveNext
Next i
```

Source Code 4.3 Pembentukan Kandidat Pertama

Pada *source code* 4.3 fungsi *query* "Sql RsTrans1, "SELECT COUNT(kode_klas) AS jml FROM temp WHERE kode_klas=" & RsTrans.Fields("kode").Value & "" berfungsi untuk menghitung *items* yang sama berdasarkan *field kode_klas* sedangkan *Db.Execute "INSERT INTO* C1 *(kode_klas,frekuensi) VALUES (" & RsTrans.Fields("kode").Value & "," & jm & ")"* berfungsi untuk menyimpan data didalam Tabel C1 *(kandidat pertama).*

Prosedur Pembentukan Kandidat Kedua

```
Sql RsTrans2, "SELECT * FROM temp WHERE id_temp='" &
RsTrans.Fields("temp1").Value & "'"
For k = 1 To RsTrans2.RecordCount

If j <> k And j < k Then

dt = RsTrans1.Fields("kode_klas").Value & "," &
RsTrans2.Fields("kode_klas").Value

Db.Execute "INSERT INTO c2 (kode_klas) VALUES ('" & dt & "')"
```

Source Code 4.4 Pembentukan Kandidat Kedua

Untuk menentukan kandidat kedua dilakukan kombinasi antar *item*. Seperti terlihat pada *source code* 4.4 fungsi *query* "Sql RsTrans2, "SELECT * FROM temp WHERE id_temp="" & RsTrans.Fields("temp1").Value & "" berfungsi untuk menghitung *items* yang sama berdasarkan *field kode_klas*, jika item sama atau terbalik maka tidak dikombinasikan. Sedangkan *Db.Execute "INSERT INTO C1 (kode_klas,frekuensi) VALUES (" & RsTrans.Fields("kode").Value & "," & jm & ")"* berfungsi untuk menyimpan data didalam Tabel C*2 (kandidat kedua).*

Prosedur Pembentukan Kandidat Ketiga

```
Sql RsTrans1, "SELECT * FROM temp WHERE id_temp=" &
RsTrans.Fields("temp1").Value & ""
For j = 1 To RsTrans1.RecordCount

Sql RsTrans2, "SELECT * FROM temp WHERE id_temp=" &
RsTrans.Fields("temp1").Value & ""
For k = 1 To RsTrans2.RecordCount

Sql RsTrans3, "SELECT * FROM temp WHERE id_temp=" &
RsTrans.Fields("temp1").Value & ""
For g = 1 To RsTrans3.RecordCount

If k <> g And k < g And j < k Then
'And k <> g Then
no = no + 1

dt = RsTrans1.Fields("kode_klas").Value & "," &
RsTrans2.Fields("kode_klas").Value & "," &
RsTrans3.Fields("kode_klas").Value

Db.Execute "INSERT INTO c3 (kode_klas) VALUES ('" & dt & "')"
```

Source Code 4.5 Pembentukan Kandidat Ketiga

Seorang pengguna/peminjam saat melakukan transaksi peminjaman dimungkinkan meminjam buku sebanyak tiga *item*. Sehingga dimungkinkan adanya pembentukan kandidat ketiga. Untuk menentukan kandidat ketiga dilakukan kombinasi antar *item*, seperti terlihat pada *source code* 4.5 pada code “dt = RsTrans1.Fields("kode_klas").Value & "," & RsTrans2.Fields("kode_klas").Value & "," & RsTrans3.Fields("kode_klas").Value. Jika ditemukan kandidat ketiga selanjutnya akan disimpan pada Tabel C3.

Prosedur Pembentukan Rule

Setelah *large itemset* terbentuk, langkah berikutnya adalah pembentukan *assosciation rule* yang memenuhi *minimum support* dan *minimum confidence* sesuai masukan *user*. Proses pembentukan *rule* seperti pada *source code* 4.6.

```
Sql RsTrans7, "SELECT * FROM rule"
no = 0

For y = 1 To RsTrans7.RecordCount

nl() = Split(RsTrans7.Fields("rule").Value, ",")
hj = ""
For u = 0 To UBound(nl)

Sql RsTrans8, "SELECT * FROM klasifikasi WHERE kode_klas='" &
nl(u) & "'"
For z = 1 To RsTrans8.RecordCount
If u > 0 Then
hj = hj & " -> " & RsTrans8.Fields("klasifikasi").Value & " (
Jika meminjam " & hj & _
" maka meminjam " & RsTrans8.Fields("klasifikasi").Value & "
)"
Else
hj = RsTrans8.Fields("klasifikasi").Value
End If
```

Source Code 4.6 *Code* Pembentukan *Rule*

4.3 Penerapan Aplikasi

Penerapan aplikasi ini dimulai dengan memasukkan data sesuai dengan keinginan *user*. Selanjutnya dilakukan penentuan tanggal data transaksi yang akan dianalisa yaitu dengan memasukkan batasan tanggal atau periode transaksi. Dengan adanya batasan tanggal transaksi, *user* bisa melakukan analisa data pada periode tanggal atau bulan tertentu secara fleksibel. Setelah memasukkan batasan tanggal transaksi, maka *user* harus memasukkan batasan nilai *minimum support* dan *minimum confidence*. Nilai *minimum support* yang telah dimasukkan akan berpengaruh pada banyak atau sedikit proses di dalam aplikasi. Selain itu juga

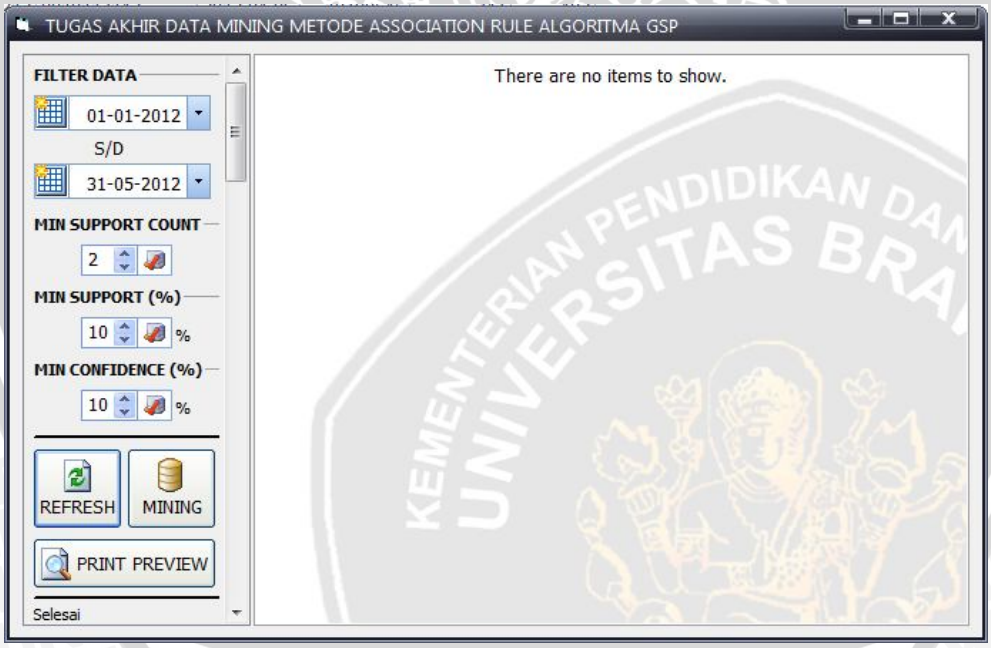
akan berpengaruh terhadap lama atau cepatnya proses aplikasi. Sedangkan jumlah aturan yang dihasilkan sangat dipengaruhi oleh nilai *minimum confidence* yang dimasukkan. Setelah proses *input* dilakukan, maka *user* dapat langsung memulai proses dengan menekan tombol proses *mining*.

4.3.1 Implementasi *Interface*

Implementasi *interface* terdiri dari form utama dan enam menu, yaitu menu proses mining, data awal, kandidat-1, kandidat-2, kandidat-3, *association rule*, *refresh* dan *print preview*. Berikut ini adalah beberapa tampilan (*interface*) aplikasi.

4.3.1.1 Form Utama

Pada form utama terdapat isian tanggal transaksi, minimum support dan minimum confidence yang harus diisikan oleh *user*.

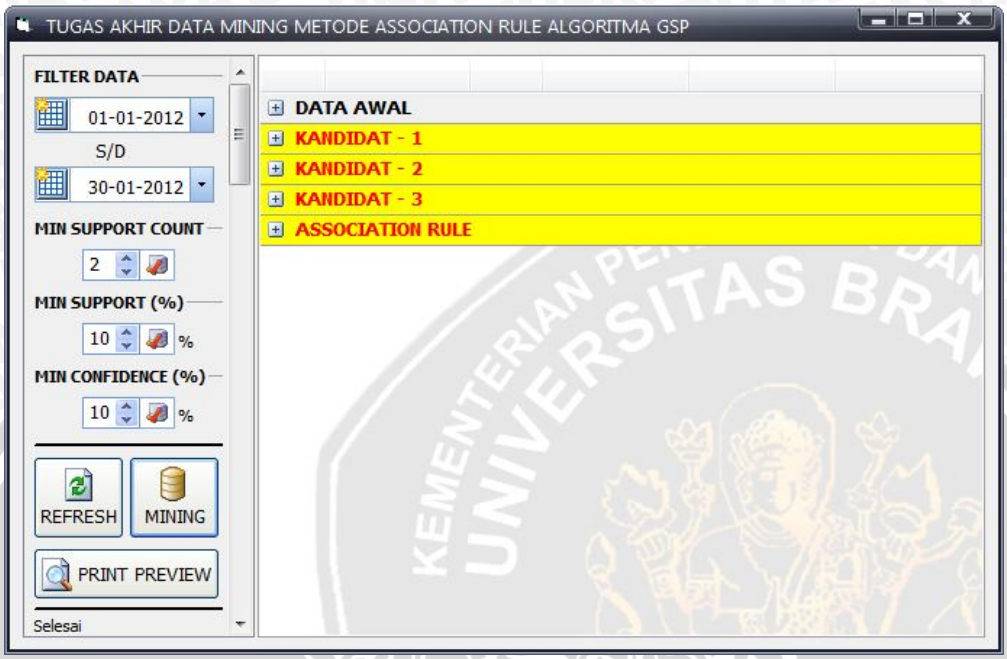


Gambar 4.1 Tampilan *interface Form* Utama

4.3.1.2 Menu Proses *Mining*

Menu proses *mining* adalah tombol untuk mengeksekusi program. Sebelum dilakukan proses *mining*, maka semua isian harus sudah terisi, terutama rentang tanggal transaksi karena tanggal transaksi dibuat *default* tanggal tertentu.

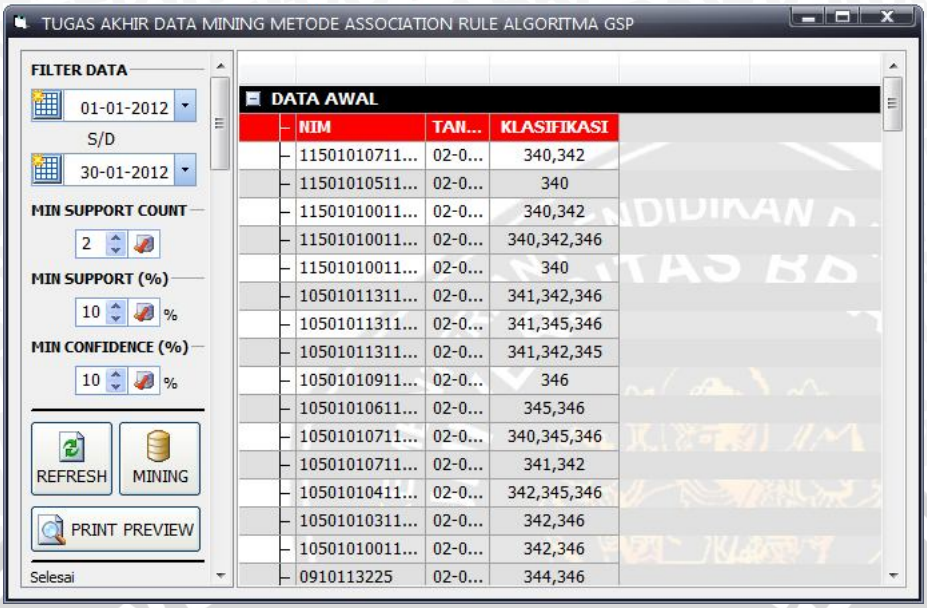
Setelah menekan tombol proses *mining*, hasilnya ditampilkan seperti pada Gambar 4.2.



Gambar 4.2 Tampilan *interface* Proses *Mining*

4.3.1.3 Menu Data Awal

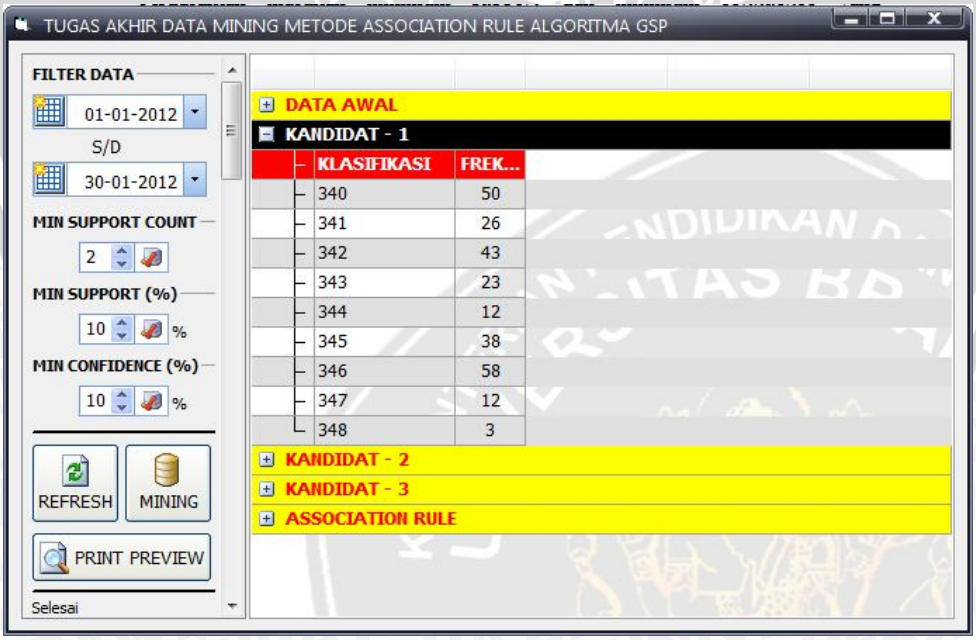
Menu data awal berisi tampilan *listview* (tabel) transformasi data transaksi menurut id peminjam dan tanggal transaksi. Jika dalam transaksi dan id peminjam terdapat kategori buku yang sama, maka hanya dihitung satu. Setelah menekan tombol data awal, hasilnya ditampilkan seperti pada Gambar 4.3.



Gambar 4.3 Tampilan *interface* Data Awal

4.3.1.4 Menu Kandidat-1

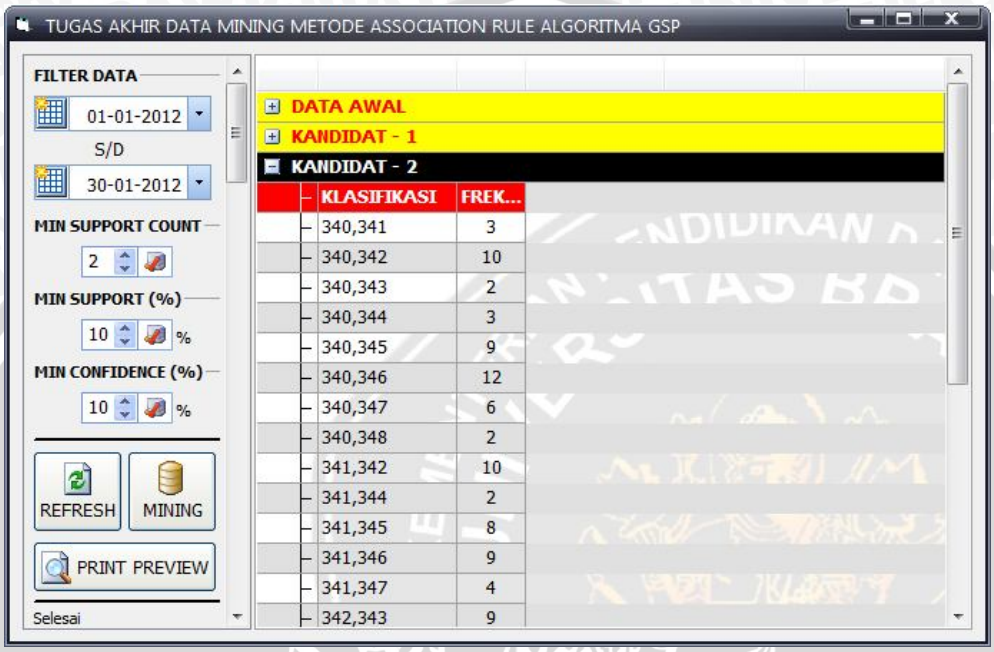
Menu kandidat-1 menampilkan data hasil pembentukan lenght 1 (L1) berdasarkan masukan minimum *support* dan minimum *confidence* yang dikehendaki *user*. Setelah menekan tombol kandidat-1, hasilnya ditampilkan seperti pada Gambar 4.4.



Gambar 4.4 Tampilan *interface* Kandidat-1

4.3.1.5 Menu Kandidat-2

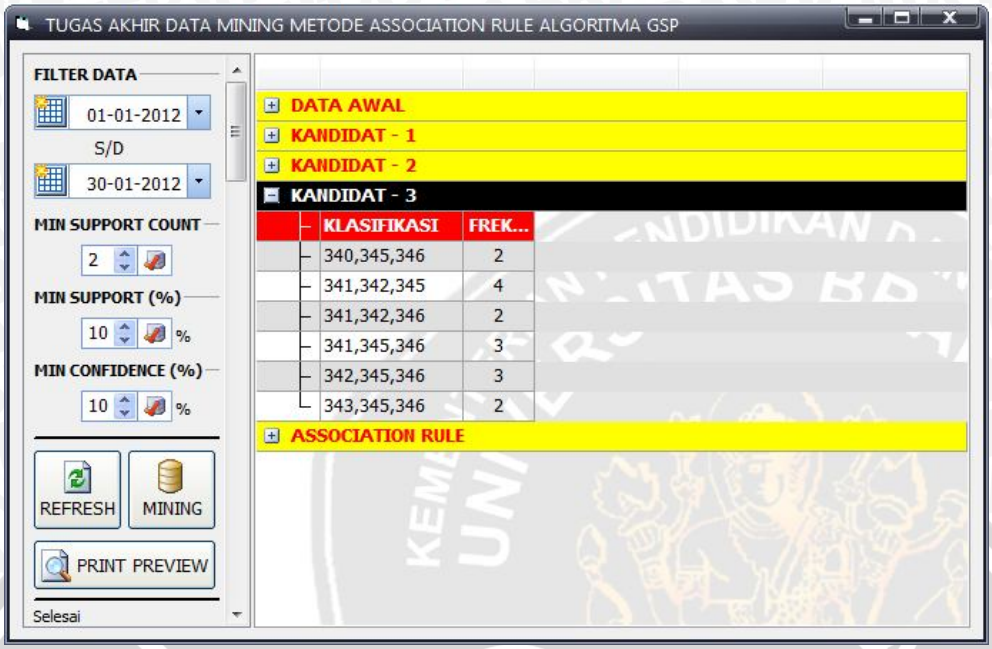
Menu kandidat-2 menampilkan data hasil pembentukan lenght 2 (L2), berdasarkan masukan minimum *support*, minimum *confidence* yang dikehendaki *user* dan kombinasi dari dua kategori. Setelah menekan tombol kandidat-2, hasilnya ditampilkan seperti pada Gambar 4.5.



Gambar 4.5 Tampilan *interface* Kandidat-2

4.3.1.6 Menu Kandidat-3

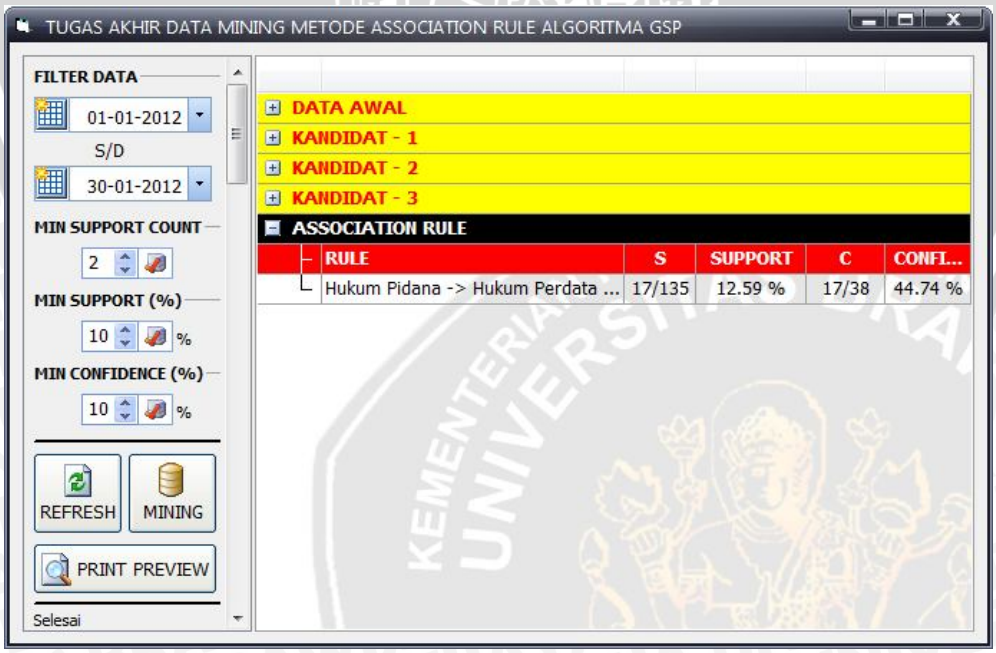
Menu kandidat-3 menampilkan data hasil pembentukan lenght 3 (L3), berdasarkan masukan minimum *support*, minimum *confidence* yang dikehendaki *user* dan kombinasi dari tiga kategori. Setelah menekan tombol kandidat-3, hasilnya ditampilkan seperti pada Gambar 4.6.



Gambar 4.6 Tampilan *interface* Kandidat-3

4.3.1.7 Menu *Association Rule*

Menu *association rule* menampilkan *rule* yang telah terbentuk berdasarkan masukan minimum *support*, minimum *confidence* yang dikehendaki *user*. Setelah menekan tombol *association rule*, hasilnya ditampilkan seperti pada Gambar 4.7.



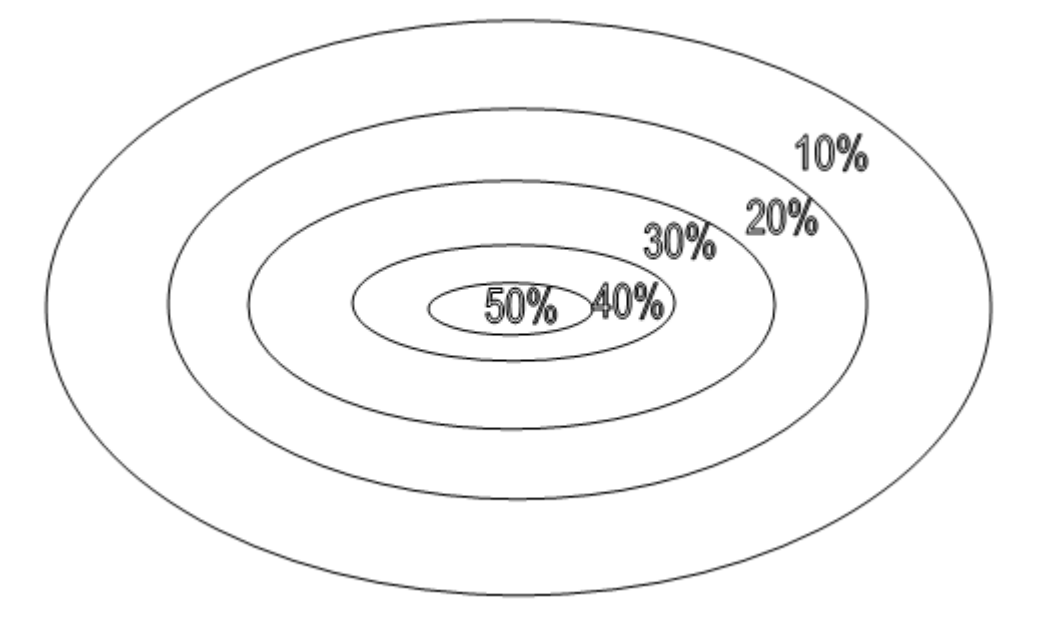
Gambar 4.7 Tampilan *interface Association Rule*

BAB VPENGUJIAN DAN ANALIS

5.1 Pengujian

Pada implementasi pengujian ini dilakukan tiga kali pengujian dengan parameter yang digunakan untuk uji pengaruh nilai *minimum support* dan nilai *minimum confidence* terhadap jumlah *rule* yang dihasilkan pada data periode 1 bulan, 3 bulan, dan 5 bulan.

Nilai *minimum support* yang diterapkan adalah 2% sampai 10% dengan nilai *minimum confidence* 10% sampai 50% tiap periode waktu. Hubungan jumlah *rule* terhadap nilai *minimum confidence* dapat dilihat pada Gambar 5.1.



Gambar 5.1 Hubungan Jumlah *Rule* yang dihasilkan berdasarkan nilai *minimum confidence* (%)

Dari Gambar 5.1 dapat diketahui bahwa jumlah *rule* yang terbentuk pada *minimum confidence* 10% adalah jumlah *rule* yang paling banyak karena meliputi jumlah *rule* yang terbentuk pada *minimum confidence* diatasnya yang lebih tinggi. Semakin tinggi *minimum confidence*, semakin sedikit *rule* yang terbentuk. Jumlah *rule* yang terbentuk pada *minimum confidence* 50% merupakan bagian dari *rule* yang terbentuk pada *minimum confidence* 40%, begitu seterusnya. Semakin kecil *minimum confidence*, semakin banyak *rule* yang terbentuk.

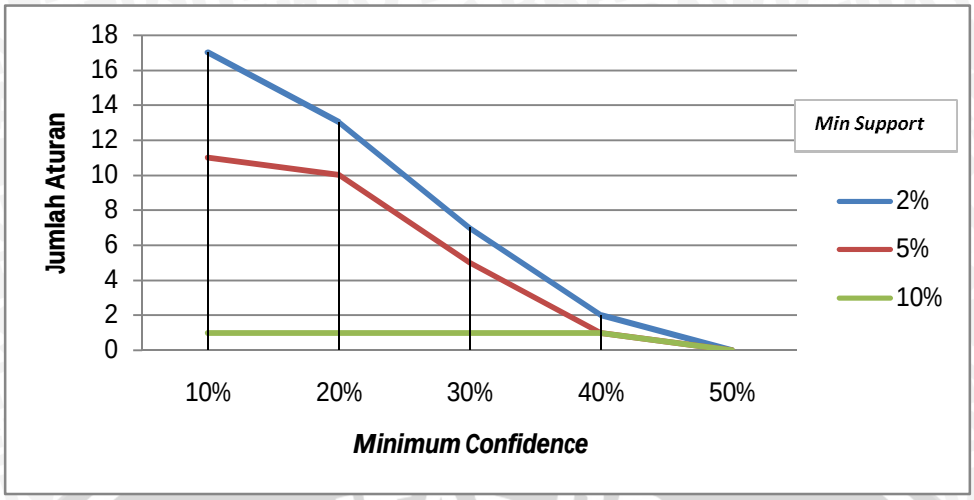
5.2 Pengujian dan Analisis untuk Periode 1 Bulan

Pengujian periode 1 bulan (Januari 2012) digunakan data 354 *record*, dengan masukan nilai *minimum support* 2% sampai 10% dan nilai *minimum confidence* 10% sampai 50% dapat dilihat pada Tabel 5.1.

Tabel 5.1 Tabel Uji Pengaruh Nilai *Minimum Support* dan Nilai *Minimum Confidence* berbeda-beda terhadap Jumlah *Rule* yang Dihasilkan Periode 1 Bulan

Periode	*Minimum Confidence*	*Minimum Support*		
		2%	5%	10%
1 bulan	10 %	17	11	1
	20 %	13	10	1
	30 %	7	5	1
	40 %	2	1	1
	50 %	0	0	0

Pada Tabel 5.1, dapat dilihat bahwa untuk pengujian 1 bulan dihasilkan aturan sebanyak 17, 11 dan 1 untuk *minimum confidence* 10% dan *minimum support* masing-masing 2%, 5% dan 10%. Sedangkan dengan *minimum confidence* 40% dihasilkan aturan sebanyak 2, 1, dan 1 untuk *minimum support* masing-masing 2%, 5% dan 10%. Khusus untuk masukan dengan nilai *minimum support* 10% hanya dihasilkan 1 aturan saja dan sama untuk nilai *minimum confidence* 10%-40%. Jika diberikan masukan dengan *minimum confidence* 50% ternyata tidak dihasilkan aturan. Untuk lebih jelas diperlihatkan pada Gambar Grafik 5.2.



Gambar 5.2 Grafik Hubungan Jumlah *Rule* yang dihasilkan dengan Nilai *Minimum Support* 2%-10%, dan *Minimum Confidence* 10%-50% Periode 1 Bulan

Dari Gambar Grafik 5.2 jika dianalisa secara keseluruhan dapat dilihat bahwa besarnya *minimum confidence* berbanding terbalik dengan jumlah aturan yang dihasilkan, semakin kecil nilai *minimum confidence* yang diberikan maka semakin banyak jumlah aturan yang dihasilkan. Begitu juga sebaliknya semakin besar nilai *minimum confidence* yang diberikan maka semakin sedikit jumlah aturan yang dihasilkan.

Association rule yang dihasilkan dari hasil uji periode 1 bulan hanya terbentuk satu *rule* dengan frekuensi kemunculan sebanyak 17 kali dengan nilai *support* 12,06% dan *confidence* 44,74% sebagai nilai tertinggi. *Rule* tersebut adalah dengan kode kategori <345>→<346> yaitu Hukum Pidana → Hukum Perdata, seperti terlihat pada Tabel 5.2 dan 5.3.

Tabel 5.2 *Rule* yang Terbentuk dari Hasil Uji Periode 1 Bulan

Frequent Itemset	**Frekuensi**	***Support***	***Confidence***
345,346	17	12,06%	44,74%

Tabel 5.3 *Rule* yang Terbentuk dari Hasil Uji Periode 1 Bulan

Rule	Keterangan
345 → 346	Jika meminjam Hukum Pidana maka meminjam juga Hukum Perdata

5.2 Pengujian dan Analisis untuk Periode 3 Bulan

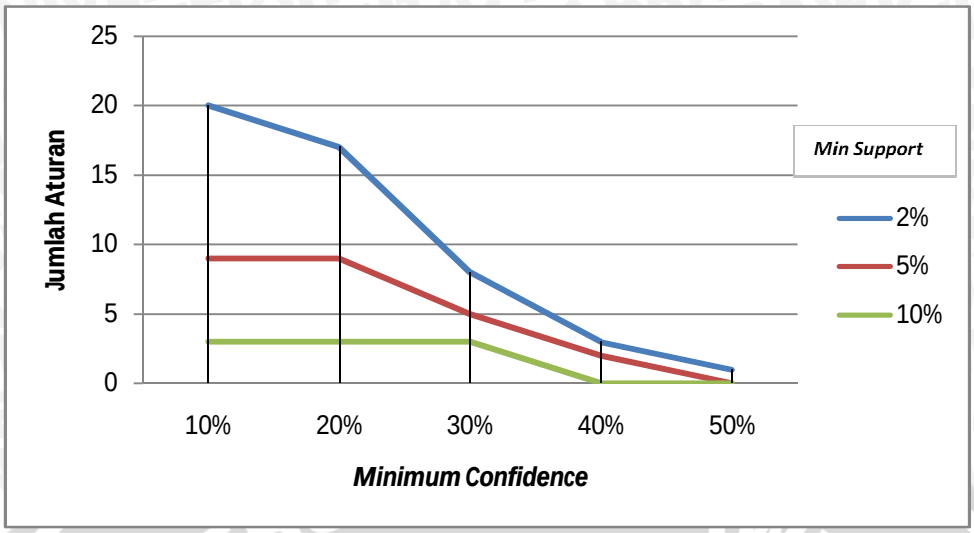
Pengujian periode 3 bulan (Januari-Maret 2012) digunakan data 960 *record*, dengan masukan nilai *minimum support* 2% sampai 10% dan nilai *minimum confidence* 10% sampai 50%. Hasil dari pengujian periode 3 bulan dapat dilihat pada Tabel 5.4.

Tabel 5.4 Tabel Uji Pengaruh Nilai *Minimum Support* dan Nilai *Minimum Confidence* berbeda-beda terhadap Jumlah *Rule* yang Dihasilkan Periode 3 Bulan

Periode	*Minimum Confidence*	*Minimum Support*		
		2%	5%	10%
3 bulan	10 %	20	9	3
	20 %	17	9	3
	30 %	8	5	3
	40 %	3	2	0
	50 %	1	0	0

Dari Tabel 5.4, dapat dilihat hasil pengujian untuk periode 3 bulan. Untuk masukan dengan nilai *minimum confidence* 10%-40% menghasilkan aturan sebanyak 20, 17, 8 dan 3 aturan pada nilai *minimum support* 2%. Namun pada pengujian ini telah dihasilkan 1 aturan pada nilai *minimum confidence* 50%. Pada pengujian 1 bulan aturan ini belum dihasilkan. Sedangkan untuk nilai *minimum support* 5% dan nilai *minimum confidence* 10%-40% dihasilkan aturan 9, 9, 5, dan 2 aturan. Hasil pengujian ini mirip dengan pengujian pada periode 1 bulan.

Pada pengujian dengan nilai *minimum support* 10% ada hal menarik yang dapat kita lihat yaitu tidak dihasilkan aturan pada nilai *minimum confidence* 40%-50%. Sedangkan pada nilai *minimum confidence* 10%-30% tetap dihasilkan aturan yang memiliki nilai sama yaitu 3 aturan. Untuk lebih jelas hasil dari pengujian periode 3 bulan dapat dilihat pada Gambar Grafik 5.3.



Gambar 5.3 Grafik Hubungan Jumlah *Rule* yang dihasilkan dengan Nilai Minimum Support 2%-10%, dan Minimum Confidence 10%-50% Periode 3 Bulan

Dari hasil pengujian periode 3 bulan dapat ditarik kesimpulan bahwa aturan yang dihasilkan masih memperlihatkan dengan semakin besar nilai *minimum confidence* dihasilkan aturan yang semakin kecil, dan dengan semakin kecil nilai *minimum support* dihasilkan aturan semakin banyak. Jika dibandingkan dengan pengujian periode 1 bulan masih memperlihatkan bahwa besarnya *minimum confidence* berbanding terbalik dengan jumlah aturan yang dihasilkan.

Sedangkan *rule* yang terbentuk dari hasil uji periode 3 bulan dapat dilihat pada table 5.5.

Tabel 5.5 *Rule* yang Terbentuk dari Hasil Uji Periode 3 Bulan

Frequent Itemset	**Frekuensi**	***Support***	***Confidence***
340,345	47	12,02%	31,76%
342,345	41	10,49%	34,17%
345,346	51	13,04%	34,93%.

Dari Tabel 5.5 dapat dilihat bahwa *rule* yang terbentuk sebanyak 3 *rule* dengan nilai *Minimum Support* dan *Minimum Confidence* di atas 10%. Namun dari ketiga *rule* yang terbentuk yang memiliki nilai tertinggi adalah dengan kode kategori <345>→<346> yaitu Hukum Pidana → Hukum Perdata dengan

frekuensi kemunculan terbanyak sebanyak 51 kali, nilai *support* sebesar 13,04% dan nilai *confidence* sebesar 34,93%.

Tabel 5.6 *Rule* yang Terbentuk dari Hasil Uji Periode 3 Bulan

Rule	Keterangan
$345 \rightarrow 346$	Jika meminjam Hukum Pidana maka meminjam juga Hukum Perdata

5.3 Pengujian dan Analisis untuk Periode 5 Bulan

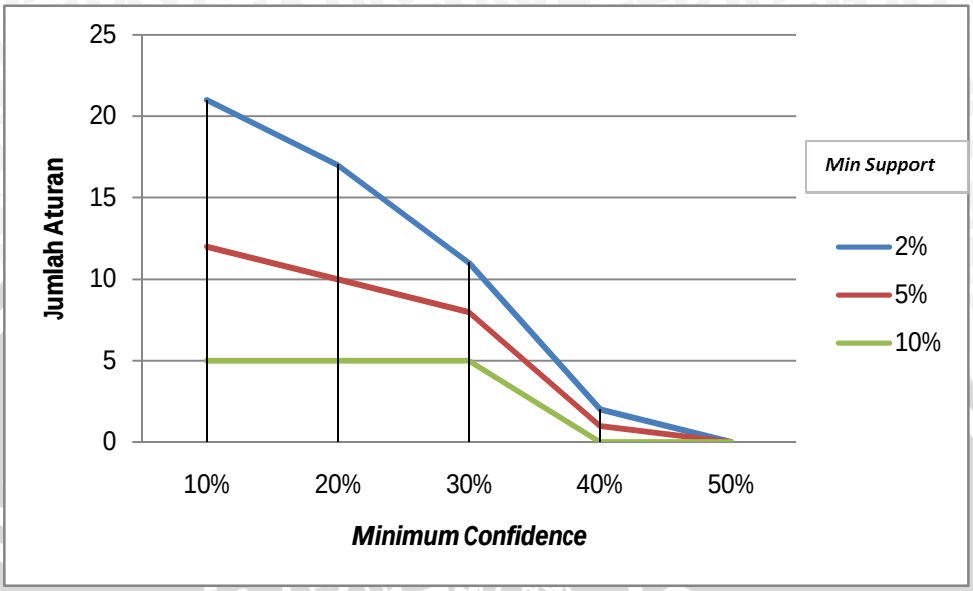
Pengujian periode 5 bulan (Januari-Mei 2012) adalah pengujian yang ketiga dengan menggunakan data 2006 *record*, dengan masukan nilai *minimum support* 2% sampai 10% dan nilai *minimum confidence* 10% sampai 50%. Hasil dari pengujian periode 5 bulan dapat dilihat pada Tabel 5.7

Tabel 5.7 Tabel Uji Pengaruh Nilai *Minimum Support* dan Nilai *Minimum Confidence* berbeda-beda terhadap Jumlah *Rule* yang Dihasilkan Periode 5 Bulan

Periode	*Minimum Confidence*	*Minimum Support*		
		2%	5%	10%
5 bulan	10 %	21	11	5
	20 %	17	10	5
	30 %	11	8	5
	40 %	2	1	0
	50 %	0	0	0

Dari Tabel 5.7 dapat dilihat bahwa dengan nilai *minimum support* 10% menghasilkan aturan sebanyak 5 aturan untuk *minimum confidence* 10%-30%. Kalau dibandingkan dengan hasil pengujian periode 1 bulan dan 3 bulan maka terlihat ada kenaikan terhadap jumlah aturan yang dihasilkan. Hal ini disebabkan data yang digunakan lebih banyak dari pada periode 1 bulan dan 3 bulan. Sedangkan aturan yang dihasilkan dari nilai *minimum support* masing-masing 2% dan 5% terhadap nilai *minimum confidence* 10%-40% masih tetap dihasilkan aturan beragam yang kecendungannya menurun sesuai besar nilai *minimum confidence*. Hal ini sama dengan hasil yang diperlihatkan pada pengujian periode 1 bulan dan 3 bulan. Untuk pengujian dengan nilai *minimum confidence* 50%

terhadap nilai *minimum support* 2%-10% tidak dihasilkan aturan sama sekali. Hal ini sama dengan pada pengujian peroiode 1 bulan. Untuk lebih jelas hasil pengujian periode 5 bulan bisa dilihat pada Gambar Grafik 5.4



Gambar 5.4 Grafik Hubungan Jumlah *Rule* yang dihasilkan dengan Nilai Minimum Support 2%-10%, dan Minimum Confidence 10%-50% Periode 5 Bulan

Association rule yang dihasilkan dari pengujian periode 5 bulan didapatkan sebanyak 5 aturan. Selengkapnya dapat dilihat pada Tabel 5.8.

Tabel 5.8 *Association Rule* yang Terbentuk dari Hasil Uji Periode 5 Bulan

Frequent Itemset	**Frekuensi**	***Support***	***Confidence***
340,345	107	13,14%	33,86%
340,346	95	11,67%	30,06%
342,345	96	11,79%	36,78%
342,346	85	10,44%	32,57%
345,346	112	13,76%	35,56%.

Dari Tabel 5.8 dapat dilihat bahwa *rule* yang terbentuk sebanyak 5 *rule* dengan nilai *Minimum Support* dan *Minimum Confidence* di atas 10%. Namun

dari kelima *rule* yang terbentuk yang memiliki nilai tertinggi adalah dengan kode kategori <345>→<346> yaitu Hukum Pidana → Hukum Perdata dengan frekuensi kemunculan terbanyak sebanyak 112 kali, nilai *support* sebesar 13,78% dan nilai *confidence* sebesar 35,44%. Walaupun pada *rule* <(342),(345)> memiliki *confidence* lebih tinggi yaitu 36,78% namun dari nilai *support* lebih kecil daripada *rule* 345,346.

Tabel 5.9 *Rule* yang Terbentuk dari Hasil Uji Periode 5 Bulan

Rule	Keterangan
345 → 346	Jika meminjam Hukum Pidana maka meminjam juga Hukum Perdata

Berdasarkan hasil pengujian yang telah dilakukan sebanyak tiga kali periode pengujian yaitu 1 bulan, 3 bulan dan 5 bulan dimana ada perbedaan jumlah data uji dapat disimpulkan bahwa aturan yang dihasilkan dalam metode *association rule* dipengaruhi oleh nilai *support* dan *confidence*. Nilai *support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item* atau *itemset* dari keseluruhan transaksi, pada penelitian ini *item* atau *itemset* yang dimaksud adalah kategori buku. Sedangkan *confidence* adalah suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional* atau prosentase tingkat kepercayaan *item* yang saling berasosiasi. Selain itu, variasi terbentuknya *association rule* juga dipengaruhi oleh jumlah maksimal buku yang dipinjam oleh seorang pelanggan. Pada penelitian ini maksimal buku yang dipinjam hanya tiga *item*, sehingga kandidat yang terbentuk paling tinggi pada kandidat-3.

Dari analisa data hasil pengujian sebagaimana ditunjukkan pada Tabel 5.1-5.9 dan Grafik 5.2-5.4, maka hasil pengujian berdasarkan *minimum support* 2%, 5%, dan 10% dengan *minimum confidence* yang berbeda-beda dapat disimpulkan bahwa pada *minimum support* yang sama dan jumlah sekuen *item* yang sama, nilai *minimum confidence* berbanding terbalik dengan jumlah aturan yang dihasilkan. Semakin kecil nilai *minimum confidence* yang diberikan maka semakin banyak jumlah aturan yang dihasilkan. Begitu juga sebaliknya semakin besar nilai

minimum confidence yang diberikan maka semakin sedikit jumlah aturan yang dihasilkan.

Dari ketiga hasil uji dapat disimpulkan bahwa dalam transaksi peminjaman buku memiliki hubungan asosiasi kategori buku yang sering dipinjam adalah kategori Hukum Pidana dan Hukum Perdata sehingga jika pelanggan meminjam kategori Hukum Pidana maka juga meminjam Hukum Perdata.

BAB VI
PENUTUP

6.1 Kesimpulan

Dari hasil uji dan analisis yang telah dilakukan maka dapat diambil kesimpulan antara lain :

1. Pola peminjaman buku berdasarkan kategori buku dari data transaksi peminjaman buku dapat diketahui dengan menerapkan metode *association rule* menggunakan algoritma GSP. Masing-masing transaksi peminjaman buku sesuai periode yang dikehendaki dihitung nilai *support*nya sehingga ditemukan kandidat 1-*itemset* (C1), dilanjutkan dengan penemuan large 1-*itemset* (L1). Iterasi dilakukan berulang kali hingga mencapai large 3-*itemset* (L3) dimana setiap iterasinya terdapat proses *join* dan *prune*.
2. Pola peminjaman buku berdasarkan kategori buku di perpustakaan Fakultas Hukum Universitas Brawijaya adalah kategori <345> Hukum Pidana dan <346> Hukum Perdata sehingga jika pelanggan meminjam buku kategori Hukum Pidana maka juga meminjam Hukum Perdata.

6.2 Saran

Dari penelitian yang telah dilakukan dapat direkomendasikan saran-saran sebagai berikut:

1. Pada pengembangan yang akan datang, dapat ditambahkan fasilitas untuk melakukan *mining* berdasarkan kategori judul buku tertentu.
2. Penerapan *association rule* tidak hanya pada transaksi tetapi juga pada peminjaman *user* tertentu pada rentang waktu tertentu.
3. Dalam pencarian pola *asossiasion rule* pada transaksi peminjaman buku bisa digunakan metode selain algoritma GSP. Sehingga dapat diketahui metode mana yang lebih baik dan efisien dari segi waktu.
4. Untuk pihak pengelola Perpustakaan Fakultas Hukum Universitas Brawijaya semoga aplikasi ini bisa dimanfaatkan dan dikembangkan.

Daftar Pustaka

Ahola, Jussi, 2001, *Mining Sequential Patterns*, Finland: VTT Information Technology.

Agrawal, R and Srikant, R., 1994., *Fast Algorithm for Mining Association Rules*, In Proceedings of the International Conference on Very Large Data Bases.

Budhi, Gregorius Satia, Handojo, Andreas dan Wirawan, Christine Oktavina, 2009, Algoritma *Generalized Sequential Pattern* untuk Menggali data Sekuensial Sirkulasi Buku Pada Perpustakaan UK Petra. Jurnal.

Fayyad, U., Piatetsky-Shapiro, G. dan Smyth, P., 1996, *From Data Mining to Knowledge Discovery in Databases*, AAAI and The MIT Pres.

Han, Jiawei, Micheline Kamber., 2001, *Data Mining: Concepts and Techniques.* Morgan Kaufmann, California.

Han, Jiawei, dan Kamber, Micheline, 2004, *Data Mining: Concepts and Techniques.* 2nd Edition.Morgan Kaufmann.

Ikhsan, Muhammad, dkk, 2007, Penerapan *Association Rule* dengan Algoritma Apriori pada Proses Pengelompokan Barang di Perusahaan Retail. Jurnal (online).

Kantardzic, Mehmed. 2003. *Data Mining: Concepts, Models,Methods, and Algorithms*. John Wiley & Sons. New Jersey.

Kusrini, dan Taufiq, Emha Luthfi, 2009, Algoritma Data *Mining*, Penerbit Andi, Yogyakarta.

Larose, Daniel T., 2005. *Discovering Knowledge in Data: AnIntroduction to Data Mining*. John Wiley & Sons, Inc. New Jersey.

Lasa, Hs., 1998, *Kamus Istilah Perpustakaan*, Gadjah Mada University Press, Yogyakarta.

Lin, Dao-I., 1998, *Fast Algorithms for Discovering the Maximum Frequent Set.* Dissertation, Department of Computer Science, New York University.

Santoso, Leo Willyanto, 2003, Pembuatan Perangkat Lunak Data *Mining* untuk Penggalian Kaidah Asosiasi Menggunakan Metode Apriori, Jurnal.

Srikant, Ramakrishnan, 1996, *Fast Algorithm for Mining Association Rules and Sequential Patterns,* A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy at the University of Wisconsin – Madison

Zaki, Mohammed J., 1997, *Fast Mining of Sequential Patterns in Very Large Databases.*The University of Rochester Computer Science Department Rochester, New York 14627. Technical report 668