

**PERSONALISASI E-LIBRARY MENGGUNAKAN
KAIDAH ASOSIASI ALGORITMA ECLAT**

**SKRIPSI
KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI**

Untuk memenuhi sebagian persyaratan
untuk mencapai gelar Sarjana Komputer



Disusun Oleh:
ADITA RIZKI P
0910962001

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
MALANG
2013**

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.

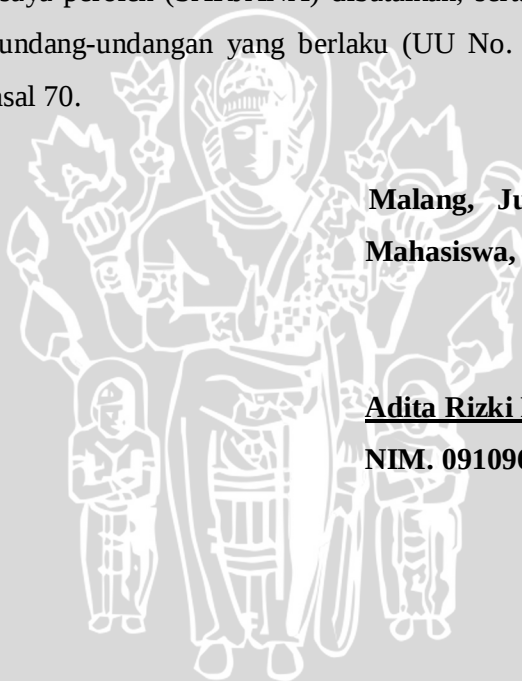
Apabila ternyata di dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat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

Malang, Juli 2013

Mahasiswa,

Adita Rizki P

NIM. 0910962001



ABSTRAK

Teknologi informasi dan komunikasi atau *Information and Communication Technology* (ICT) telah menjadi bagian yang tidak terpisahkan dari kehidupan global. Oleh karena itu, perpustakaan pada perguruan tinggi berusaha mengintegrasikan ICT agar dapat menyalurkan informasi dengan lebih baik dan mengembangkan minat baca dari mahasiswanya. Salah satunya dengan *E-Library*. *E-Library* akan lebih efektif, diminati, dan dapat mengembangkan minat baca apabila setiap pengguna mendapatkan rekomendasi sesuai dengan minat pengguna. Salah satu solusi yang tepat untuk mengatasi hal ini adalah dengan menggunakan personalisasi *e-Library*. Cara kerja personalisasi *e-Library* adalah sistem akan mengamati perilaku dari pengguna dan dicatat ke dalam database. Pengamatan dan pencatatan sistem ini meliputi buku apa saja yang dipinjam atau diunduh oleh pengguna. Dari data pengamatan tersebut kemudian diterapkan *Text Mining* untuk mendapatkan kata kunci (*keyword*) dari setiap buku yang dipinjam atau diunduh untuk kemudian dianalisis agar dapat memberikan rekomendasi dengan *keyword* yang sesuai dengan minat pengguna. Analisis pada sistem ini menggunakan Algoritma Eclat. Algoritma Eclat adalah salah satu metode kaidah asosiasi yang menggunakan perpotongan (*intersection*) antar transaksi *ID List* untuk mencari *frequent itemset*nya untuk kemudian dibentuk *rule* agar dapat dijadikan rekomendasi. Untuk evaluasi *rule* yang dihasilkan digunakan *Lift Ratio* agar rekomendasi yang dihasilkan lebih akurat. Pada penelitian ini rata-rata nilai *Lift Ratio* dari *rule* yang dihasilkan adalah 3,43 sedangkan titik optimum dari minimum support, minimum confidence dan jumlah transaksi dari penelitian ini adalah 10%, 60% dan 150 transaksi peminjaman.

ABSTRACT

Information and communication technology or Information and Communication Technology (ICT) has become an integral part of global life. Therefore, the library at the college seeks to integrate ICT in order to distribute information better and develop reading interest of students. One of them with the E-Library. E-Library will be more effective, attractive, and can develop an interest in reading when each user gets a recommendation in accordance with the user's interest. One right solution to overcome this is to use personalized e-Library. Personalized e-Library is observe the behavior of the system user and saved into a database. Observation and recording system includes any book borrowed or downloaded by the user. The observational data is then applied to obtain keyword with Text Mining of any books borrowed or downloaded for later analysis in order to provide recommendations to the keywords that match the user interest. The analysis in this system using the Eclat algorithm. Eclat algorithm is method of association rules using the intersection between the transaction ID list to search for frequent itemset then established rule that can be recommended. For the evaluation of the resulting rule this study used lift ratio so that produced rule more accurate for better recommendations. In this study, the average value of the ratio rule lift generated was 3.43 while the optimum point of minimum support, minimum confidence and the number of transactions of this study was 10%, 60% and 150 transactions.

KATA PENGANTAR

Alhamdulillah *rabbi alamin*. Puji syukur penulis panjatkan kehadiran Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, Skripsi yang berjudul **“PERSONALISASI E-LIBRARY MENGGUNAKAN KAIDAH ASOSIASI ALGORITMA ECLAT”** ini dapat berjalan dengan baik. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer (PTIIK), Universitas Brawijaya Malang.

Shalawat serta salam tetap tercurahkan kepada Baginda Rasulullah Muhammad *Shalallahu ‘alaihi wasallam*, makhluk paling mulia yang senantiasa memberikan cahaya petunjuk, seorang uswatun hasanah yang telah membawa agama Allah yaitu agama Islam menjadi agama yang *Rahmatan Lil ‘Alamin*.

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

1. Lailil Muflikhah, S.Kom., M.Sc, selaku dosen pembimbing utama yang telah meluangkan waktu untuk memberikan pengarahan dan masukan bagi penulis.
2. Dian Eka Ratnawati, S.Si., M.Kom, selaku dosen pembimbing kedua yang telah banyak memberikan bimbingan serta bantuan bagi penulis.
3. Drs. Marji, M.Si. dan Issa Arwani, S.Kom., M.Sc. selaku ketua dan sekretaris Program Studi Teknik Informatika / Ilmu Komputer.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Teknik Informasi / Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya.
5. Unit konseling Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya yang telah memberikan dukungan motivasi dan berbagai masukan terkait pelaksanaan penyusunan skripsi ini.

6. Segenap staff dan karyawan di Program Teknologi Informasi dan Ilmu Komputer (PTIHK) Universitas Brawijaya yang telah banyak membantu penulis dalam pelaksanaan penyusunan skripsi ini.
7. Kedua orang tua dan keluarga tercinta, terima kasih atas semua doa, kasih sayang dan perhatian yang tulus serta dukungan yang telah diberikan.
8. Rekan-rekan di Program Studi Teknik Informasi / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer (PTIHK) Universitas Brawijaya yang telah banyak memberikan bantuannya demi kelancaran pelaksanaan penyusunan skripsi ini.
9. Keluarga Chun, Ilkomers Muke Gile, Sari dan Tuzzu yang selalu mensupport dan membantu kelancaran penyusunan skripsi ini.
10. Dan semua pihak yang telah terlibat baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu persatu terima kasih atas semua bantuan yang telah diberikan.

Semoga skripsi ini bermanfaat bagi pembaca sekalian. Akhirnya, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca.

Malang, 26 Juli 2013

Penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang

“Perpustakaan merupakan sebuah institusi yang menyediakan koleksi bahan pustaka tertulis, tercetak dan terekam sebagai pusat informasi yang diatur menurut sistem aturan dan didayagunakan untuk keperluan pendidikan, penelitian serta rekreasi intelektual bagi masyarakat” [SUP-86]. Dari definisi perpustakaan oleh Supriyanto dan Muhsin tersebut maka perpustakaan merupakan hal yang mutlak bagi sebuah perguruan tinggi yang diharapkan sebagai pusat kegiatan pengembangan minat baca dan penyedia informasi bagi mahasiswanya.

Pada era informasi abad ini, teknologi informasi dan komunikasi atau ICT (Information and Communication Technology) telah menjadi bagian yang tidak terpisahkan dari kehidupan global. Oleh karena itu, perpustakaan pada perguruan tinggi berusaha mengintegrasikan ICT agar dapat menyalurkan informasi dengan lebih baik dan mengembangkan minat baca dari mahasiswanya. Salah satunya dengan *E-Library*.

“*E-Library* adalah sebuah sistem yang memiliki berbagai layanan dan obyek informasi yang mendukung akses obyek informasi tersebut melalui perangkat digital” [SIS-08]. Layanan ini diharapkan dapat mempermudah pencarian informasi di dalam koleksi obyek informasi seperti dokumen, gambar dan database dalam format digital dengan cepat, tepat, dan akurat. *E-Library* akan lebih efektif, diminati, dan dapat mengembangkan minat baca apabila setiap pengguna mendapatkan rekomendasi sesuai dengan minat pengguna, karena setiap pengguna mempunyai minat yang berbeda satu sama lain, dengan begitu maka *e-Library* akan tetap dikunjungi oleh penggunanya. Salah satu solusi yang tepat untuk mengatasi hal ini adalah dengan menggunakan personalisasi *e-Library*.

Personalisasi *e-Library* merupakan bagian dari personalisasi web. “Personalisasi web adalah suatu proses mengumpulkan dan menyimpan informasi tentang lokasi pengunjung, meneliti informasi, dan berdasarkan pada analisa, mengirimkan informasi yang tepat kepada masing-masing pengunjung di waktu yang tepat, personalisasi merupakan suatu upaya untuk memberikan layanan

dalam bentuk aplikasi dan informasi yang disesuaikan dengan minat, peran, dan kebutuhan pengunjung web” [CHI-01].

Cara kerja personalisasi *e- Library* adalah sistem akan mengamati perilaku dari pengguna dan dicatat ke dalam database. Pengamatan dan pencatatan sistem ini meliputi buku apa saja yang dipinjam atau diunduh oleh pengguna. Dari data pengamatan tersebut kemudian diterapkan *Text Mining* untuk mendapatkan *keyword* (kata kunci) dari setiap buku yang dipinjam atau diunduh untuk kemudian dianalisis agar dapat memberikan rekomendasi dengan *keyword* (kata kunci) yang sesuai dengan minat pengguna. Analisis data pada sistem ini menggunakan teknik *data mining*.

“*Data Mining* merupakan sebuah proses untuk menemukan hubungan, pola dan trend baru yang bermakna dengan menyaring data yang sangat besar, yang tersimpan dalam penyimpanan, menggunakan teknik pengenalan pola seperti teknik statistik dan matematika” [PON-06]. Hubungan yang dicari dalam *data mining* dapat berupa hubungan antara dua atau lebih dalam satu dimensi, misalnya dalam dimensi produk, dapat dilihat keterkaitan pembelian suatu produk dengan produk yang lain. Beberapa teknik dalam data mining yang sering digunakan antara lain Kaidah Asosiasi (*association rule*) [WEI-01].

Penggalian kaidah asosiasi mempunyai peranan penting dalam proses pengambilan keputusan. Tahapan besar dari proses *Data Mining* adalah mengidentifikasi *frequent itemset* dan membentuk kaidah asosiasi dari *itemset* tersebut. Kaidah asosiasi digunakan untuk menggambarkan hubungan antar *item* pada tabel data transaksional ataupun data relasional. Semakin berkembangnya teknologi komputer di dunia industri, semakin pesat pula perkembangan ukuran data yang dihasilkan. Dan pada data yang besar (*VLDB, Very Large Database*) tersebut, proses pencarian *frequent itemset* sangatlah sulit. Kaidah asosiasi umumnya dilakukan dengan memakai metode algoritma *Apriori*. Algoritma *Apriori* adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian *frequent itemset* dengan menggunakan teknik *association rule* [MOE-02]. Algoritma *Apriori* ini menggunakan pengetahuan mengenai *frequent itemset* yang telah diketahui sebelumnya, untuk memproses informasi selanjutnya. Namun algoritma ini memiliki kekurangan yaitu terlalu banyak melakukan *scanning*

database setiap kali iterasi, sehingga waktu yang diperlukan bertambah dengan makin banyak iterasi. Dengan adanya kekurangan pada algoritma *Apriori* tersebut, maka muncullah banyak penelitian algoritma-algoritma untuk mengatasi kekurangan algoritma *Apriori*. Pada penelitian perbandingan antara Algoritma *Apriori* dan Algoritma *Eclat* [BOR-03] ditemukan bahwa Algoritma *Eclat* dapat mengatasi kelemahan dari Algoritma *Apriori* dan bekerja sangat efisien.

Algoritma *Eclat* adalah salah satu metode dalam kaidah asosiasi. Kelebihan dari algoritma *Eclat* adalah menggunakan urutan *Lexicographic* (pengurutan kata/huruf sesuai kamus) dalam mengurutkan *itemset*-nya sehingga *itemset* yang sama hanya akan diolah satu kali. Algoritma ini men-*generate* kandidatnya dengan pencarian *depth-first* sehingga tidak perlu menghitung *support* semua *itemset* karena algoritma ini hanya menyimpan *frequent itemset* yang ditemukan. Algoritma ini juga direpresentasikan secara *vertikal* dalam *database* (untuk setiap *item*, dibuat *list/array* transaksi) sehingga mengurangi pemakaian memori yang menyebabkan algoritma ini bekerja lebih cepat [BOR-03].

Saat ini penulis ingin melakukan penelitian bagaimana menghasilkan personalisasi *e-Library* dengan menggunakan kaidah asosiasi algoritma *eclat*. Penelitian ini bermaksud menerapkan personalisasi *e-Library* dengan wujud rekomendasi berupa buku, jurnal, dan artikel yang sesuai dengan minat pengguna yang sudah pernah dilakukan oleh penulis sebelumnya [ARI-10]. Penulis sebelumnya menggunakan data *e-Learning* sedangkan pada studi kasus kali ini penulis menggunakan data *e-Library*, teknik yang digunakan penulis sebelumnya adalah *data mining* algoritma *Apriori*, sedangkan pada studi kasus kali ini penulis akan menggunakan teknik *data mining* algoritma *Eclat*. Berdasarkan latar belakang diatas maka penulis akan melakukan penelitian dengan judul “**Personalisasi E-Library menggunakan Kaidah Asosiasi Algoritma Eclat**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka dapat dirumuskan permasalahan yang akan dijadikan objek penelitian ini, yaitu:

1. Bagaimana menerapkan kaidah asosiasi algoritma *ecolat* pada sistem rekomendasi untuk personalisasi *e-Library*.
2. Bagaimana pengaruh perubahan nilai minimum *support* terhadap kevalidan *rule-rule* yang dihasilkan dari personalisasi *e-Library* dengan kaidah asosiasi algoritma *Eclat* dengan menggunakan *Lift Ratio*.
3. Bagaimana pengaruh perubahan nilai minimum *confidence* terhadap kevalidan *rule-rule* yang dihasilkan dari personalisasi *e-Library* dengan kaidah asosiasi algoritma *Eclat* dengan menggunakan *Lift Ratio*.
4. Bagaimana pengaruh perubahan jumlah transaksi terhadap kevalidan *rule-rule* yang dihasilkan dari personalisasi *e-Library* dengan kaidah asosiasi algoritma *Eclat* dengan menggunakan *Lift Ratio*.

1.3 Batasan Masalah

Penelitian yang dilaksanakan dalam penulisan skripsi ini memiliki beberapa batasan sebagai berikut :

1. Data pengamatan yang digunakan adalah data transaksi buku yang dipinjam atau diunduh oleh mahasiswa.
2. Data yang diolah menjadi rule adalah *keyword* (kata kunci) dari judul buku yang dipinjam atau diunduh pada data pengamatan.
3. Data pengamatan yang digunakan bersifat statis, atau tidak mengalami perubahan dan dengan jumlah yang tetap selama proses berlangsung.
4. *Stemming* pada *Pre-processing Text Mining* menggunakan Algoritma Nazief dan Adriani (Agusta, 2009) tidak dijabarkan dengan rinci karena skripsi ini hanya difokuskan pada penggunaan Kaidah Asosiasi.
5. Tidak ada perbandingan metode.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian dan penulisan skripsi ini yaitu :

1. Menerapkan kaidah asosiasi algoritma *ecolat* untuk menghasilkan sistem rekomendasi untuk personalisasi *e-Library*.
2. Mengetahui kevalidan *rule-rule* yang dihasilkan dari personalisasi *e-Library* dengan kaidah asosiasi algoritma *Eclat*.

1.5 Manfaat Penelitian

Manfaat yang ingin dicapai dari penulisan skripsi ini adalah sebagai berikut:

1. Agar *e-Library* tetap diminati, dapat mengembangkan minat baca pengguna dan menjadi sarana utama yang selalu dikunjungi oleh pengguna selama mengenyam pendidikan.
2. Agar *e-Library* dapat meningkatkan kualitas pendidikan dan menciptakan atmosfir akademik.

1.6 Sistematika Penulisan

Pembuatan tugas akhir ini dilakukan dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan tugas akhir.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini membahas tentang dasar teori yang terkait dengan topik penulisan skripsi yang diangkat yang menjadi acuan dasar dalam pembuatan sistem pengklasifikasian teks ini.

BAB III METODOLOGI DAN PERANCANGAN

Pada bab ini dijelaskan metode-metode yang digunakan dalam menyelesaikan masalah, perancangan sistem, perancangan basis data, serta perancangan antar muka sistem.

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan implementasi aplikasi, implementasi basis data.

BAB V PENGUJIAN DAN ANALISIS

Pada bab ini dijelaskan uji coba sistem beserta analisa hasil.

BAB V PENUTUP

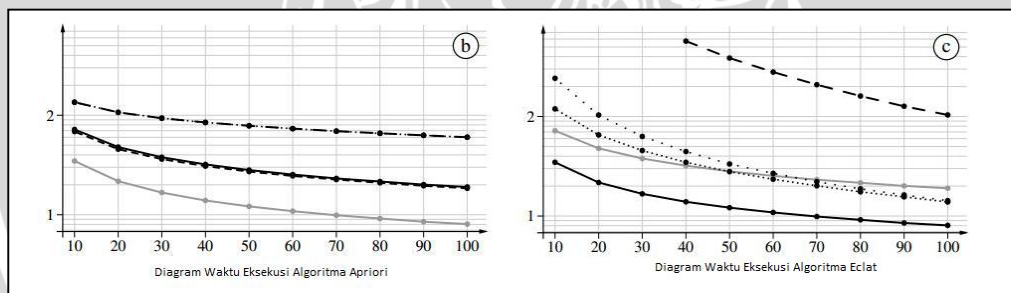
Bab ini berisi kesimpulan dari pembahasan dan saran yang diharapkan dapat bermanfaat untuk pengembangan tugas akhir ini selanjutnya.

BAB II

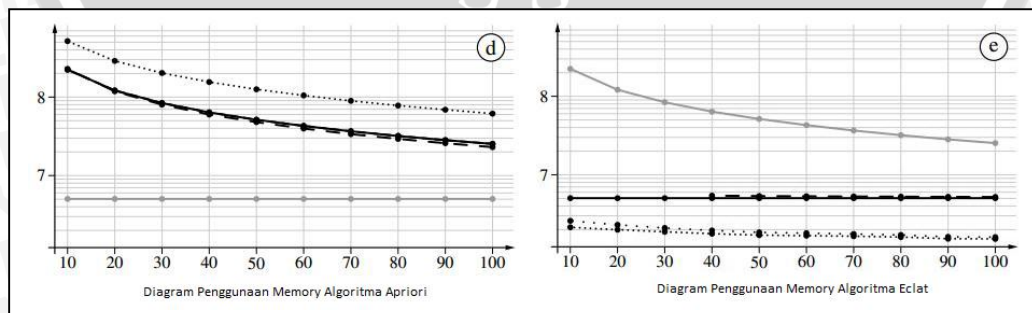
KAJIAN PUSTAKA DAN DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian terdahulu yang dijadikan kajian pustaka pada skripsi adalah jurnal karangan Christian Borgelt dengan judul “*Efficient Implementations of Apriori and Eclat*” yang diterbitkan pada tahun 2003. Dalam jurnal tersebut peneliti mengimplementasikan algoritma *apriori* dan algoritma *eclat* dengan beberapa optimasi untuk mendapatkan performansi yang maksimum dalam waktu eksekusi dan penggunaan *memory*. Penelitian tersebut mempunyai kesimpulan bahwa performansi algoritma *eclat* dapat mengalahkan algoritma *apriori* dalam waktu eksekusi dan penggunaan *memory*. Hasil percobaan dengan data census (dataset dari US census bureau 1994) dapat dilihat pada gambar diagram 2.1 untuk waktu eksekusi sedangkan untuk penggunaan memory dapat dilihat pada gambar diagram 2.2



Gambar 2.1 Diagram Hasil Percobaan waktu eksekusi
Algoritma Eclat dan Apriori



Gambar 2.2 Diagram Hasil Percobaan penggunaan memory
Algoritma Eclat dan Apriori

Jurnal lain yang dijadikan kajian pustaka adalah jurnal yang dibuat oleh Nugroho Wandu, Rully A. Hendrawan, dan Ahmad Mukhlason pada tahun 2012 dengan judul “Pengembangan Sistem Rekomendasi Penelusuran Buku dengan Penggalan Association Rule Menggunakan Algoritma Apriori (Studi Kasus Badan Perpustakaan dan Kearsipan Provinsi Jawa Timur)”. Dalam jurnal tersebut penulis melakukan analisa terhadap histori dari transaksi peminjaman buku yang ada. Pemilihan data histori sebagai bahan analisa dikarenakan dari data ini bisa digali pola-pola asosiasi antar buku yang dipinjam pada transaksi-transaksi yang ada. Metode yang digunakan dalam identifikasi pola yang dimaksud adalah association rule dengan algoritma apriori. Metode dan algoritma ini menghasilkan transaksi-transaksi peminjaman buku dengan strong association (keterkaitan yang kuat) antar buku dalam transaksi yang digunakan sebagai rekomendasi peminjaman buku yang membantu pengguna mendapatkan rekomendasi buku lain ketika pengguna melihat rincian dari buku yang dipilih atau hendak dipinjam.

Jurnal e-Library yang digunakan dalam skripsi ini adalah jurnal yang dibuat oleh Umami Rodliyah yang diterbitkan pada tahun 2012 dengan judul “PERPUSTAKAAN DIGITAL, DAN PROSPEKNYA MENUJU RESOURCE SHARING”. Jurnal tersebut membahas tentang penerapan Teknologi Informasi pada perpustakaan untuk mendukung pelaksanaan pendidikan dan pengajaran, penelitian serta pengabdian dan pelayanan pada masyarakat. Salah satu solusinya adalah membangun Perpustakaan ideal yang mampu memenuhi keinginan pengguna yang menyediakan informasi yang memadai atau menyediakan akses kepada berbagai sumber informasi, dapat diakses kapan saja, dimana saja dan dipandu oleh pustakawan yang profesional yaitu dengan membangun Perpustakaan Digital.

2.2 DASAR TEORI

2.2.1 E-Library

E-Library (perpustakaan digital) adalah sebuah sistem yang memiliki berbagai layanan dan obyek informasi yang mendukung akses obyek informasi tersebut melalui perangkat digital [SIS-08]. Layanan ini diharapkan dapat mempermudah pencarian informasi di dalam koleksi obyek informasi seperti

dokumen, gambar dan database dalam format digital dengan cepat, tepat, dan akurat. *E-Library* itu tidak berdiri sendiri, melainkan terkait dengan sumber-sumber lain dan pelayanan informasinya terbuka bagi pengguna di seluruh dunia. Koleksi *E-Library* tidaklah terbatas pada dokumen elektronik pengganti bentuk cetak saja, ruang lingkup koleksinya malah sampai pada artefak digital yang tidak bisa digantikan dalam bentuk tercetak. Koleksi menekankan pada isi informasi, jenisnya dari dokumen tradisional sampai hasil penelusuran. *E-Library* ini melayani mesin, manajer informasi, dan pemakai informasi. Semuanya ini demi mendukung manajemen koleksi, menyimpan, pelayanan bantuan penelusuran informasi.

2.2.2 Personalisasi Web

Salah satu aspek pengembangan dalam aplikasi web yang banyak diteliti adalah personalisasi web (website personalization). Personalisasi web merupakan sebuah respon kepada pengunjung web berupa konten yang disajikan dalam sebuah situs web mampu memberikan informasi yang sesuai dengan selera dari pengunjung tersebut.

Personalisasi web adalah suatu proses mengumpulkan dan menyimpan informasi tentang lokasi pengunjung, meneliti informasi, dan berdasarkan pada analisa, mengirimkan informasi yang tepat kepada masing-masing pengunjung di waktu yang tepat, personalisasi merupakan suatu upaya untuk memberikan layanan dalam bentuk aplikasi dan informasi yang disesuaikan dengan minat, peran, dan kebutuhan pengunjung web [CHI-01].

2.2.3 Data Mining

Data mining adalah sebuah proses untuk menemukan hubungan, pola dan trend baru yang bermakna dengan menyaring data yang sangat besar, yang tersimpan dalam penyimpanan, menggunakan teknik pengenalan pola seperti teknik statistik dan matematika [PON-06]. Data mining adalah suatu proses menggunakan berbagai perangkat analisis data untuk menemukan pola atau relasi dalam suatu data yang mungkin berguna dalam membuat prediksi yang valid [EDE-99]. Dari kedua pengertian di atas, data mining secara umum dapat

didefinisikan sebagai serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui dari suatu kumpulan data.

Istilah data mining dan *knowledge discovery in databases* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, akan tetapi berkaitan satu sama lain, dimana salah satu tahapan dalam keseluruhan proses KDD adalah data mining.

Proses KDD secara garis besar dapat dijelaskan sebagai berikut :

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing/ Cleaning*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak.

3. *Transformation*

Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data mining*

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation/ Evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan.

Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

2.2.4 Text Mining

Text mining merupakan salah satu aplikasi dari data mining. Text mining juga sering disebut sebagai Text Data Mining (TDM) dan Knowledge Discovery in Textual Database (KDT). Text mining adalah proses menambang data berupa teks dengan sumber data biasanya dari dokumen dan tujuannya adalah mencari kata-kata yang mewakili dalam dokumen sehingga dapat dilakukan analisa keterhubungan dalam dokumen [HER-10]. Pada intinya proses kerja text mining sama dengan proses kerja data mining pada umumnya hanya saja data yang dimining merupakan text database.

Proses kerja dari text mining terdapat beberapa tahapan, yaitu:

1. *Text Preprocessing*

Pada text mining, data teks akan diproses menjadi data numerik agar dapat dilakukan proses lebih lanjut. Sehingga dalam text mining ada istilah text preprocessing, yaitu proses pendahulu yang diterapkan terhadap data teks yang bertujuan untuk menghasilkan data numerik. Pada tahap text preprocessing ini ada beberapa tahapan yang dilakukan di antara lain:

- a. *Text cleanup*, merupakan proses pembersihan teks. Dalam tahapan ini data teks akan diubah menjadi bentuk normal. Dimana, pengubahan teks tersebut yaitu pengubahan bentuk alfabet normal (huruf kecil), pembersihan karakter dan simbol-simbol dalam teks tersebut.
- b. Parsing/tokenizing, adalah sebuah proses yang dilakukan untuk menjadikan sebuah kalimat teks menjadi lebih bermakna atau berada dengan cara memecah kalimat tersebut menjadi kata-kata atau frase-frase.
- c. Part of speech tagging, menandai kata-kata dalam sebuah teks sesuai dengan bagian-bagian yang disampaikan. Berdasarkan aturan yaitu tergantung pada aturan tata bahasa. Berdasarkan statistik yaitu tergantung pada probabilitas urutan kata yang berbeda.
- d. Word sense disambiguation, menentukan kata yang memiliki arti yang berbeda dalam kalimat tertentu (ambigu).

2. Text Transformation

Pada tahap ini dilakukan penyaringan teks dari hasil text preprocessing. Adapun proses transformasi ini dilakukan dengan stopwords removal atau filtering dan juga dengan mengubah kata tunggal menjadi kata dasar (stemming).

a. Stopwords removal/filtering

Kebanyakan bahasa resmi di berbagai Negara memiliki kata fungsi dan kata sambung seperti artikel dan preposisi yang hampir selalu muncul pada dokumen-dokumen teks. Biasanya kata-kata ini memiliki arti yang lebih di dalam memenuhi kebutuhan seorang searcher di dalam mencari informasi. Kata-kata tersebut (misalnya a, an, the on pada bahasa Inggris) disebut sebagai stopwords. Di dalam bahasa Indonesia stopwords dapat disebut sebagai kata tidak penting, misalnya “di”, “oleh”, “pada”, “sebuah”, “karena”. Sebelum proses stopwords removal dilakukan, terlebih dulu dibuat daftar stopwords (stoplist). Preposisi, kata hubung dan partikel biasanya merupakan kandidat stoplist.

Stopwords removal merupakan proses penghilangan kata tidak penting pada dokumen melalui pengecekan kata-kata hasil text preprocessing, apakah termasuk di dalam daftar kata tidak penting (stoplist) atau tidak. Jika termasuk di dalam stoplist maka kata-kata tersebut akan dihapus sehingga kata-kata yang tersisa di dalam dokumen dianggap sebagai kata-kata penting atau keywords.

b. Stemming

Stemming adalah proses pemetaan dan penguraian berbagai bentuk (variants) dari suatu kata menjadi bentuk kata dasarnya (stem). Proses ini juga disebut sebagai conflation. Proses stemming secara luas sudah digunakan di dalam Information retrieval (pencarian informasi) untuk meningkatkan kualitas informasi yang didapatkan. Kualitas informasi yang dimaksud misalnya untuk mendapatkan hubungan antara varian kata yang satu dengan yang lainnya.

Selain itu stemming juga dapat digunakan untuk mengurangi ukuran dari suatu ukuran index file. Misalnya dalam suatu dokumen terdapat variantkata “memberikan”, “diberikan”, “memberi” dan “diberi” hanya

memiliki akar kata (stem) yaitu “beri”. Ukuran file daftar index yang semula berjumlah lima record akan di-reduce sehingga menjadi satu record saja.

3. *Pattern Discovery*

Pattern discovery atau penemuan pola merupakan tahap terpendin dari proses text mining. Tahap ini berusaha menemukan pola atau pengetahuan dari keseluruhan teks.

2.2.5 Stemming Bahasa Indonesia dengan Algoritma Nazief dan Andriani

Algoritma stemming untuk bahasa yang satu berbeda dengan algoritma stemming untuk bahasa lainnya. Sebagai contoh bahasa Inggris memiliki morfologi yang berbeda dengan bahasa Indonesia sehingga algoritma stemming untuk kedua bahasa tersebut juga berbeda. Proses stemming pada teks berbahasa Indonesia lebih rumit/kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan root word (kata dasar) dari sebuah kata [AGU-09].

Pada umumnya kata dasar pada bahasa Indonesia terdiri dari kombinasi:

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut (Agusta, 2009) :

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah root word. Maka algoritma berhenti.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus Derivation Suffixes (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
 - a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
 - b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.

4. Hapus Derivation Prefix. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
 - a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
 - b. For $i = 1$ to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan
 - c. langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan Recoding.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word. Proses selesai.

2.2.6 Kaidah Asosiasi (*Association Rule*)

Association rule mining adalah suatu prosedur untuk menemukan aturan asosiatif antara suatu kombinasi item dalam suatu dataset yang ditentukan [WEI-01]. Contoh dari aturan asosiatif dari analisa pembelian di suatu pasar swalayan adalah bisa diketahui berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu.

Dalam menentukan suatu *association rule*, terdapat suatu *interestingness measure* (ukuran kepercayaan) yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu *support* dan *confidence*. Kedua ukuran ini nantinya berguna dalam menentukan *interesting association rules*, yaitu untuk dibandingkan dengan batasan (*threshold*) yang ditentukan oleh user. Batasan tersebut umumnya terdiri dari *minimum support* dan *minimum confidence* [LAR-05].

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap [LAR-05]:

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*, yang dapat dirumuskan [LAR-05] sebagai berikut:

$$\text{Support} = P(A \cap B) = \frac{\text{jumlah transaksi yang memuat item A dan item B}}{\text{total jumlah transaksi}} \quad (2.1)$$

dimana,

Support :suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item/itemset* dari keseluruhan transaksi yang menentukan apakah suatu *item/itemset* layak untuk dicari *confidence*-nya (misal, dari seluruh transaksi yang ada, seberapa besar tingkat dominasi yang menunjukkan bahwa *item A* dan *item B* dibeli bersamaan) [LAR-05].

2. Pembentukan aturan asosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif $A \rightarrow B$ dengan menggunakan rumus [LAR-05] berikut :

$$\text{Confidence} = P(B|A) = \frac{\text{Jumlah transaksi yang mengandung item A dan item B}}{\text{Jumlah transaksi yang mengandung item A}} \quad (2.2)$$

dimana,

Confidence:Suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional* (misal, seberapa sering *item B* dibeli jika orang membeli *item A*) [LAR-05].

2.2.7 Algoritma Eclat

Sesuai dengan namanya, *tid-list(transaction ID list)* dari *itemset* adalah *list* dari ID yang diurutkan dari semua transaksi yang mengandung *itemset* tersebut. *Frequent k-itemsets* diatur ke dalam *equivalence class* (kelas-kelas serupa berdasarkan kriteria tertentu yang terbentuk dari mempartisi suatu himpunan) yang terpisah dengan $(k-1)$ -*prefix*, sehingga kandidat $(k+1)$ -*itemsets* bisa di-*generate* dengan menggabungkan pasangan *frequent k-itemsets* dari kelas yang sama. *Support* dari kandidat *itemset* dapat dihitung hanya dengan memotong *tid-list* dari dua komponen *subsets*. Letak dari tugas paralel adalah dengan membagi tugas *mining* untuk kelas yang berbeda dari *itemsets* dari proses yang tersedia. Menetapkan *equivalence classes* dari semua *frequent 2-itemsets* ke dalam proses dan *tid-list* yang terkait didistribusikan. Setiap proses kemudian me-*mining*

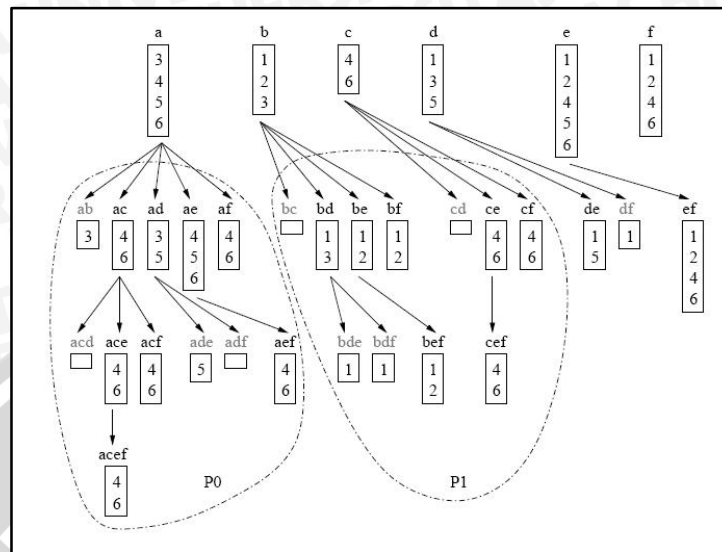
frequent itemsets yang di-generate dari *equivalence classes* secara independen, dengan *scanning* dan memotong “lokal” *tid-lists* [CHR-06].

Secara garis besar Algoritma Eclat mempunyai tiga fase [ZAK-97], yaitu:

1. Fase Inisialisasi: konstruksi perhitungan global untuk *frequent 2-itemsets*.
2. Fase Transformasi: mempartisi *frequent 2-itemsets* dan menjadwalkan partisi atas *processor*. Fase ini juga melakukan vertikal transformasi pada database.
3. Fase *Asynchronous*: konstruksi *frequent k-itemsets*.

Langkah-langkah Algoritma Paralel Eclat [CHR-06]:

1. Membagi data transaksi secara merata ke dalam partisi horizontal antar semua proses.
2. Setiap proses *scanning* partisi “lokal” data transaksinya untuk mengumpulkan perhitungan semua *1-itemsets* dan *2-itemsets*.
3. Semua proses mempertukarkan dan menjumlahkan perhitungan “lokal” untuk mendapatkan perhitungan global dari semua *1-itemsets* dan *2-itemsets*, dan mencari nilai *frequent* yang ada diantaranya.
4. Partisi *frequent 2-itemsets* ke dalam *equivalence class* dengan prefix.
5. Menetapkan *equivalence class* ke dalam proses.
6. Setiap proses mengubah partisi “lokal” data transaksinya menjadi vertikal *tid-lists* untuk semua *frequent 2-itemsets*.
7. Setiap proses menukar “lokal” *tid-lists*-nya dengan proses yang lain untuk mendapatkan semuanya secara global untuk *equivalence class* yang ditetapkan.
8. Untuk setiap *equivalence class* pada setiap proses, secara rekursif *mine* semua *frequent itemsets* dengan menggabungkan pasangan dari *itemsets* dari *equivalence class* yang sama dan memotong *tid-lists* yang berhubungan.



Gambar 2.3 Mining Frequent Itemsets menggunakan Algoritma Paralel Eclat

Presentasi Algoritma Eclat [ZAK-97]:

The Eclat Algorithm

Initialisation Phase:

- Scan local data transaction partition
- Compute local counts for all 2-itemsets
- Construct global L_2 count

Transformation Phase:

- Partition L_2 into equivalence class
- Schedule L_2 over the set of processor P
- Transform local data transaction into vertical form
- Transmit relevant tid-lists to other processor

Asynchronous Phase:

- For each equivalence class E_2 in L_2
- ComputeFrequent(E_2)

Final Reduction Phase:

- Aggregate Results and Output Associations

Gambar 2.6 Algoritma Eclat

Penjelasan Algoritma Eclat:

Pada gambar 2.3 menggambarkan bagaimana algoritma eclat me-mining frequent itemsets dari satu kelas ke kelas berikutnya menggunakan persimpangan tid-lists. Frequent 2-itemsets disusun ke dalam lima equivalence classes yang akan ditetapkan ke dalam dua proses (P0 dan P1). P0 akan bertanggung jawab untuk mining satu equivalence class, {ac,ad,ae,af}, sedangkan proses P1 akan bertanggung jawab untuk mining dua equivalence class, {bd,be,bf} dan {ce,cf}.

Kelas paling kanan, $\{de\}$ dan $\{ef\}$, tidak perlu *mining* lebih lanjut. Kedua proses tersebut kemudian diproses secara paralel, tanpa ketergantungan data antar proses. Misalnya, untuk *mining itemset* “ace”, proses P0 hanya perlu menggabungkan dua *itemsets*, “ac” dan “ae”, dan memotong *tid-lists*-nya, “46” dan “456”, untuk mendapatkan hasil *tid-list* “46”. Pada waktu yang sama proses P1 secara independen melakukan *mining itemsets* “bef” dari “be” dan “bf” dan asosiasi *tid-lists*-nya yang tersedia.

Algoritma Eclat menggunakan skema partisi *equivalence class* untuk data transaksi. Dasar dari *equivalence class* adalah *common prefix* yang diasumsikan *itemsets* sudah terurut secara *Lexicographic* (pengurutan kata/huruf sesuai kamus dalam mengurutkan *itemset*-nya sehingga *itemset* yang sama hanya akan diolah satu kali) (misal: AB, AC, AD termasuk dalam *equivalence* S_A class karena merupakan *common prefix* A).

Kelebihan dari Algoritma Eclat adalah algoritma ini hanya melakukan *scanning* data transaksi sebanyak tiga kali dan secara signifikan mengurangi biaya I/O disk. Ketergantungan antar proses dipecah sejak awal sehingga tidak ada komunikasi atau sinkronisasi yang diperlukan dalam fase *asynchronous* utama. Biaya komunikasi terbesar berasal dari pertukaran “lokal” *tid-lists* di semua proses ketika global *tid-lists* di set, biaya ini dapat dihilangkan pada iterasi selanjutnya [BOR-03].

2.2.8 Evaluasi

Untuk menghitung kevalidan rule-rule yang dihasilkan oleh Algoritma Eclat digunakan *Lift Rasio*.

Lift rasio merupakan salah satu cara yang lebih baik untuk melihat kuat tidaknya aturan asosiasi adalah membandingkan dengan nilai *benchmark*, dimana kita asumsikan kejadian item dari *concequent* (mewakili bagian “maka”) dalam suatu transaksi adalah *independent* dengan kejadian dari *antecedent* (mewakili bagian “jika”) dari suatu aturan asosiasi. Atau dengan kata lain, bila item-item dalam *antecedent* dan *concequent* saling *independent*, maka *support* adalah:

$$\frac{P(\text{Antecedent}) \times P(\text{Consequent})}{P(\text{Antecedent})} = P(\text{Consequent}) \quad (2.3)$$

Nilai estimasi dari *confidence benchmark* dihitung dari suatu aturan yang dihitung dengan:

$$\text{Confidence Benchmark} = \frac{\text{Jumlah transaksi dengan item dalam consequent}}{\text{jumlah transaksi dalam database}} \quad (2.4)$$

Kita bandingkan *confidence* terhadap *confidence benchmark* dengan melihat rasionya, yang dinamakan lift rasio. Jadi *lift rasio* adalah perbandingan antara *confidence* untuk suatu aturan dibagi dengan *confidence*, dimana diasumsikan *consequent* dan *antecedent* saling *independent*.

$$\text{Lift Rasio} = \frac{\text{Confidence}}{\text{Confidence Benchmark}} \quad (2.5)$$

Nilai *lift rasio* lebih besar dari 1 menunjukkan adanya manfaat dari aturan tersebut. Lebih tinggi lift rasio, lebih besar kekuatan asosiasi [SAN-07].



BAB III

METODOLOGI DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai metodologi dan perancangan untuk penelitian judul Sistem rekomendasi untuk personalisasi *e-Library* menggunakan kaidah asosiasi Algoritma Eclat. Proses penelitian ini akan diterapkan pada komputer dengan aplikasi berbasis web.

Penelitian dilakukan dengan langkah – langkah sebagai berikut :

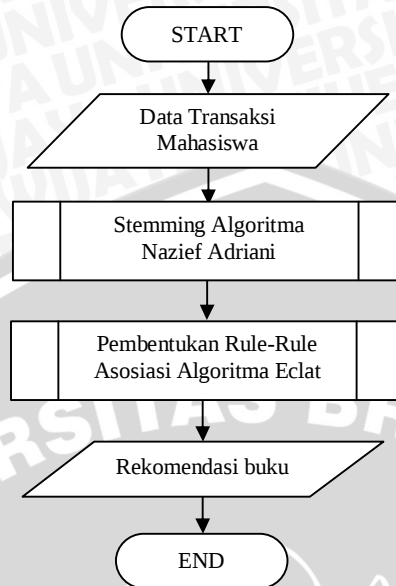
1. Melakukan studi literatur kaidah asosiasi dan Algoritma Eclat.
2. Melakukan analisa dan perancangan (perhitungan manual) untuk membangun perangkat lunak dari studi literature yang telah dilakukan sebelumnya terhadap obyek yang diteliti.
3. Mengimplementasikan analisa dan perancangan ke dalam perangkat lunak.
4. Melakukan uji coba terhadap perangkat lunak yang telah dibuat dan membandingkan data hasil dari perangkat lunak dengan data hasil dari perhitungan manual.
5. Melakukan evaluasi terhadap perangkat lunak yang telah dibuat.

3.1 Analisis Data

Kumpulan data yang digunakan dalam penelitian ini adalah data peminjaman buku *digital library* (Digilib). Kumpulan data ini didapatkan dari Politeknik Negeri Semarang.

3.2 Perancangan Sistem

Sistem yang akan dibuat merupakan sistem personalisasi *e-Library* dengan menggunakan kaidah asosiasi Algoritma Eclat. Gambar 3.1 menunjukkan alur deskripsi umum sistem personalisasi *e-Library*.



Gambar 3.1 Alur deskripsi umum sistem personalisasi e-Library

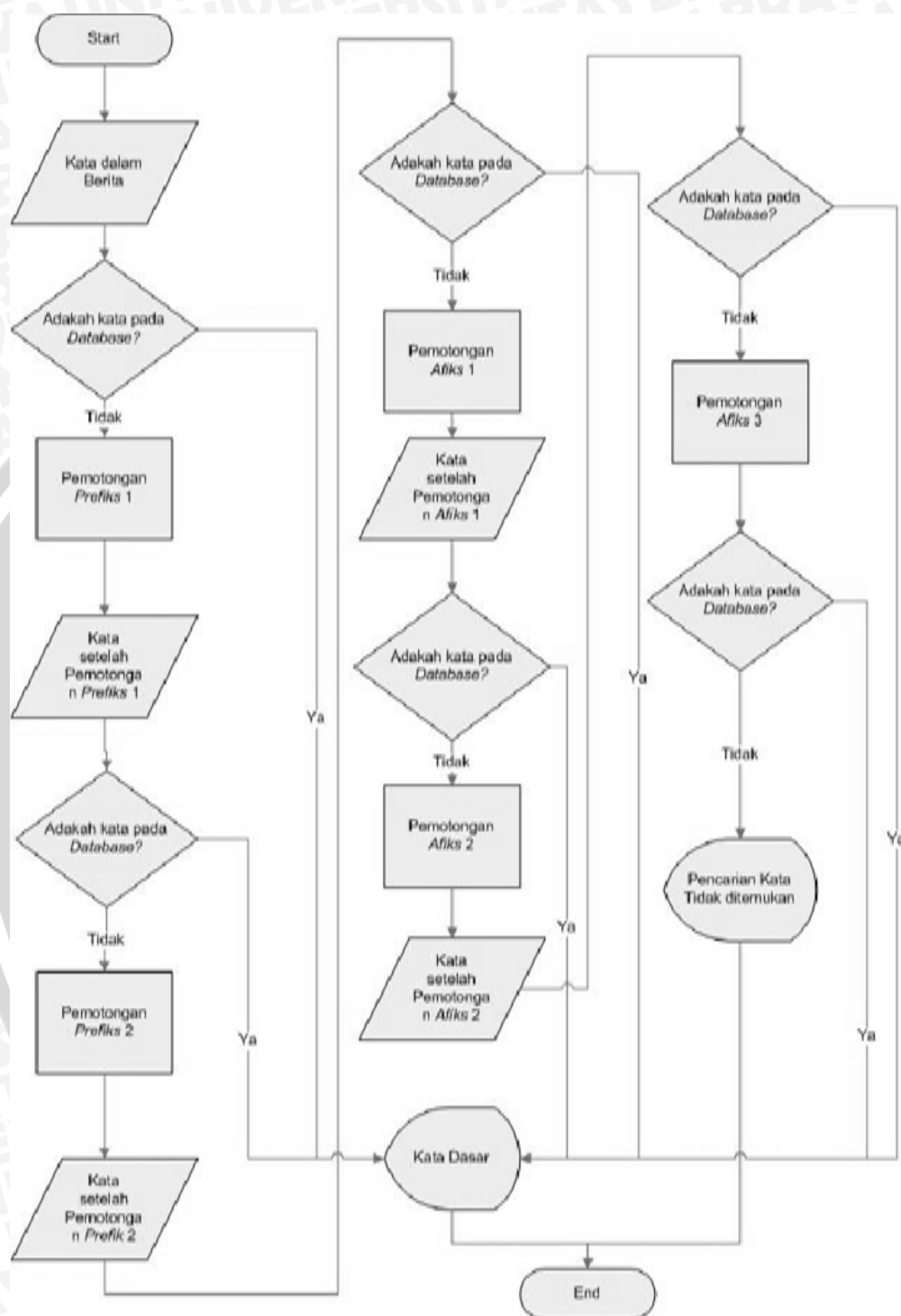
Penjelasan dari gambar 3.1, sebagai berikut:

1. Input berupa data transaksi buku yang dipinjam atau yg diunduh oleh mahasiswa.
2. *Stemming* judul buku dari data transaksi dengan Algoritma Nazief Adriani untuk mendapatkan *keyword*.
3. Pembentukan *rule-rule* asosiasi dengan metode Algoritma Eclat dari *keyword-keyword* yang didapatkan dari proses *stemming*.
4. Pemberian rekomendasi buku kepada user.

3.3 Perancangan Proses

3.3.1 Stemming dengan Algoritma Nazief Adriani

Alur dari Stemming dengan Algoritma Nazief Adriani akan ditunjukkan pada Gambar 3.2.



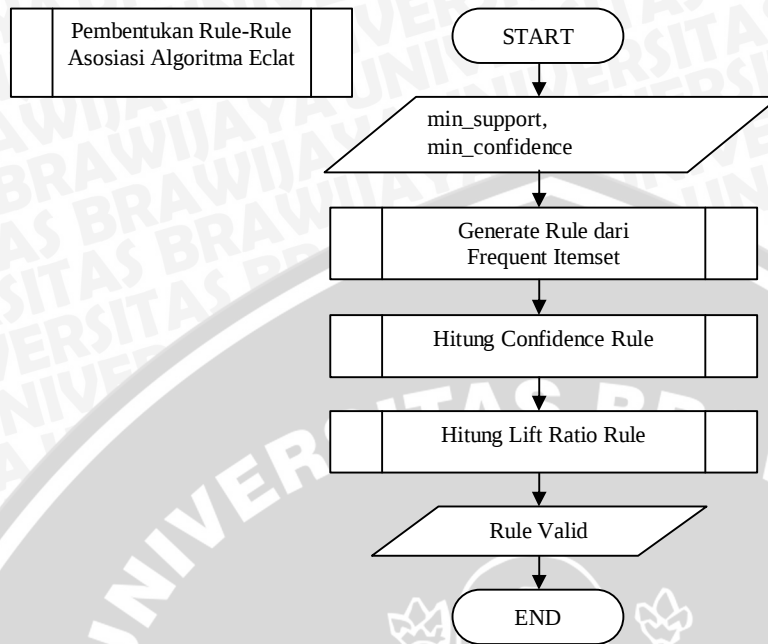
Gambar 3.2 Flowchart Stemming Algoritma Nazief Adriani [AGU-09]

Keterangan Gambar 3.2:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah root word. Maka algoritma berhenti.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus Derivation Suffixes (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
 - a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
 - b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus Derivation Prefix. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
 - a. Periksa tabel kombinasi awalan-akhirian yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
 - b. For $i = 1$ to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan Recoding.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word. Proses selesai.

3.3.2 Pembentukan *rule-rule* kaidah asosiasi dengan Algoritma Eclat

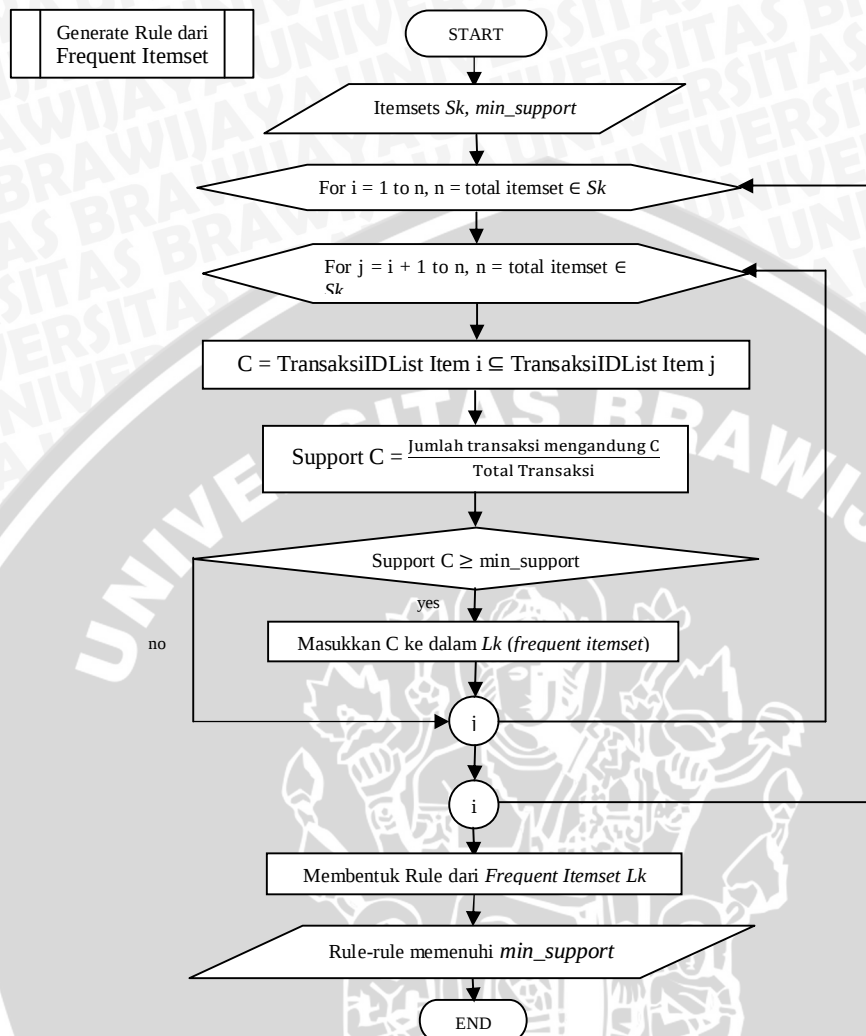
Alur dari pembentukan rule-rule akan ditunjukkan pada Gambar 3.3.



Gambar 3.3 Alur pembentukan rule-rule asosiasi valid

3.3.3 Generate Rule dari Frequent Itemset

Untuk menghasilkan *rule-rule* yang memenuhi *min_support*, generate kandidat *frequent k-itemset* untuk setiap *equivalence classes* (S_k) pada Algoritma Eclat ditunjukkan pada Gambar 3.4.



Gambar 3.4 Alur Generate Rule dari Frequent Itemset

Keterangan Gambar 3.4:

1. *Equivalence classes* (S_k) pada Algoritma Eclat didapatkan dari mengelompokkan item-item pada *frequent 1-itemset* (item-item yang memenuhi $min_support$).
2. Untuk mencari *frequent itemset* selanjutnya maka setiap *item* pada *equivalence classes* (S_k) dicari perpotongan transaksi ID list (TID) dengan *item* lainnya yang ada dalam S_k , yang hasilnya adalah C.
3. C dihitung nilai *support*-nya dan apabila nilai *support* lebih besar sama dengan $min_support$ maka C akan dimasukkan ke dalam *frequent itemset* (L_k).

4. Dari *frequent itemset* (L_k) dibentuk *rule-rule*-nya. *Rule-rule* yang dihasilkan merupakan *rule-rule* yang memenuhi *min_support*

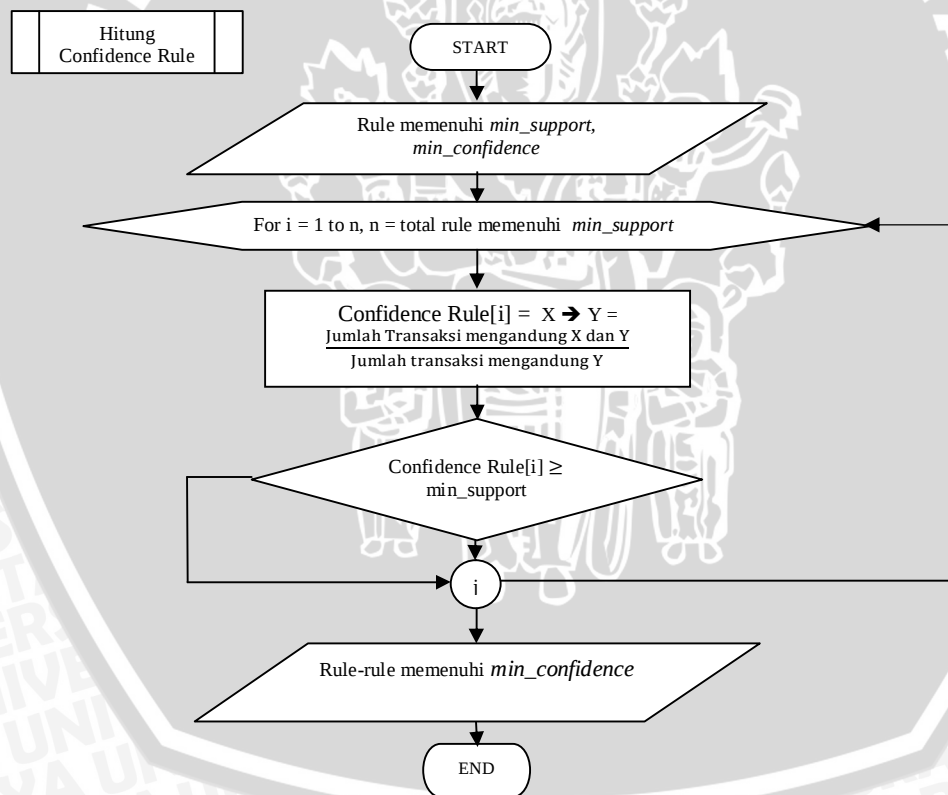
3.3.4 Hitung Confidence Rule

Pada tahap ini *rule-rule* yang memenuhi *min_support* dihitung *confidence*-nya untuk mengetahui *rule-rule* yang memenuhi *min_confidence*.

Langkah-langkah pada perhitungan *Confidence Rule*, adalah sebagai berikut:

1. Untuk setiap *rule-rule* yang memenuhi *min_support* yang dihasilkan dari proses sebelumnya, dihitung nilai *confidence*-nya.
2. Apabila nilai *confidence*-nya lebih besar sama dengan *min_confidence* maka rule tersebut adalah rule yang memenuhi *min_confidence*.

Alur hitung *confidence rule* dapat dilihat pada gambar 3.5.



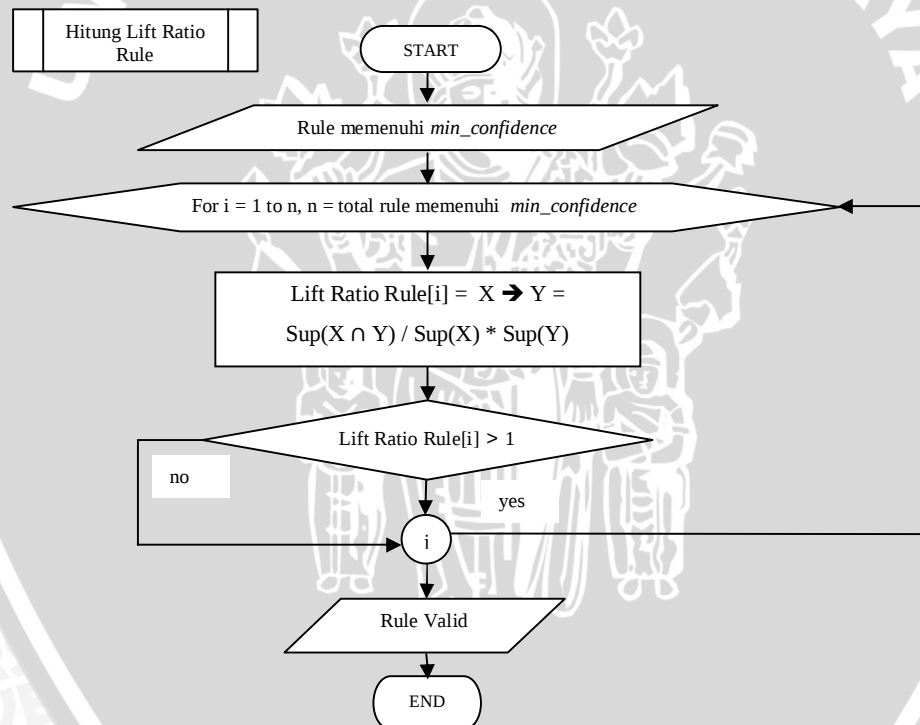
Gambar 3.5 Alur Hitung Confidence

3.3.5 Hitung Lift Ratio

Pada tahap ini *rule-rule* yang dihasilkan oleh algoritma eclat (memenuhi *min_support* dan *min_confidence*) dihitung menggunakan *Lift Ratio* untuk mengukur tingkat kevalidannya. Alur perhitungan *Lift Ratio* ditunjukkan pada gambar 3.6.

Langkah-langkah pada perhitungan *Lift Ratio Rule*, adalah sebagai berikut:

1. Untuk setiap *rule-rule* yang memenuhi *min_confidence* yang dihasilkan dari proses sebelumnya, dihitung nilai *Lift Ratio*-nya.
2. Apabila nilai *Lift Ratio*-nya lebih besar dari 1 maka rule tersebut adalah rule yang valid, yang berarti bahwa dalam transaksi tersebut, bahan ajar X dan Y benar-benar dipilih secara bersamaan.



Gambar 3.6 Alur Hitung Lift Ratio

3.4 Perancangan Basis Data

1. Tabel Buku

Tabel Buku berisi data buku .

Tabel 3.1 Tabel Buku

Field	Type
idBuku	varchar(15)
judulBuku	varchar(255)
thnTerbit	int(4)
pengarang	varchar(255)
idPenerbit	varchar(15)
kategori	varchar(25)
keyword	varchar(255)
jumlah	int(4)
jml_skrng	int(4)

2. Tabel Detail Buku

Tabel Detail Buku berisi data detail tiap buku.

Tabel 3.2 Detail Buku

Field	Type
idDetilBuku	varchar(15)
idBuku	varchar(15)
kode_buku	varchar(25)
Status	enum('hilang', 'rusak', 'booking', 'keluar', 'ada')
tglInput	date
user	varchar(30)

3. Tabel trx_pinjam

Tabel trx_pinjam digunakan untuk menyimpan data transaksi peminjaman buku tiap mahasiswa.

Tabel 3.3 Tabel trx_pinjam

Field	Type
idTrxPinjam	varchar(25)
idMahasiswa	varchar(25)
idDetilBuku	varchar(15)
tglPesan	date
tglPinjam	date
tglKembali	date
tglBatal	date
oleh	varchar(45)
isDelete	tinyint(1)

4. Tabel trx_arule

Tabel trx_arule digunakan untuk menyimpan data pemrosesan association rule yang dilakukan.

Tabel 3.4 Tabel trx_arule

Field	Type
idTrxARule	varchar(25)
idTrxPorter	varchar(25)
minSup	int(5)
minConf	int(5)
tglProses	date

5. Tabel trx_detil_arule

Tabel trx_detil_arule digunakan untuk menyimpan rule-rule yang dihasilkan.

Tabel 3.5 Tabel trx_detil_rule

Field	Type
idDetArule	varchar(25)
idTrxARule	varchar(25)
rule	varchar(50)
confidence	float(5,1)
statRule	enum('Valid', 'No Valid')
lift	float(5,1)

6. Tabel freq_itemset

Tabel freq_itemset digunakan untuk menyimpan data frequent itemset yang dihasilkan.

Tabel 3.6 Tabel freq_itemset

Field	Type
idFreqIt	varchar(25)
idTrxARule	varchar(25)
itemset	varchar(50)
sup	float(5,1)
statSup	enum('Memenuhi', 'Tidak Memenuhi')
jenisfreq	varchar(25)
class	varchar(25)

3.5 Contoh Perhitungan Manual

Untuk perhitungan manual akan digunakan tiga puluh transaksi peminjaman buku dari satu mahasiswa yang diambil dari tanggal 01-01-2008 sampai 01-04-2008. Data transaksi peminjaman buku dapat dilihat pada tabel 3.7.

Tabel 3.7 Tabel trx_pinjam perhitungan manual

idTrxPinjam	judulBuku	keyword
TRX-001	Belajar CorelDraw dalam Sehari	ajar,coreldraw,hari
TRX-002	Big Basics Book untuk Microsoft Office untuk Windows 95	big,basics,book,microsoft,office,windows
TRX-003	Buku Latihan Pemrograman DataBase dengan Visual Basic 6.0	buku,latih,pemrograman,database,visual,basic
TRX-004	Konsep dan perancangan DataBase / Cetakan 2	konsep,ancang,database,cetak
TRX-005	Layanan Pelanggan Di Era Informasi	layan,langgan,era,informasi

TRX-006	Mastering Microsoft Office Professional untuk Windows 95	mastering,microsoft,office,professional,win dows
TRX-007	Membangun Aplikasi DataBase Dengan Visual FoxPro 8.0 dan Bahasa SQL	bangun,aplikasi,database,visual,foxpro,bah asa,sql
TRX-008	Menguasai Visual Basic untuk Windows	kuasa,visual,basic,windows
TRX-009	Microsoft Excel 97	microsoft,excel
TRX-010	Microsoft Office 4.3 Professional	microsoft,office,professional
TRX-011	Microsoft Visual Basic	microsoft,visual,basic
TRX-012	Microsoft Visual Basic 4.0 Developers Workshop	microsoft,visual,basic,developers,worksho p
TRX-013	Microsoft Word 97	microsoft,word
TRX-014	PageMaker 5.0 untuk Windows	pagemaker,windows
TRX-015	Pemrograman Clie / Server Dengan Microsoft Visual Basic	pemrograman,clie,server,microsoft,visual, basic
TRX-016	Pemrograman Basis data	program,basis,data
TRX-017	Pengantar Komunikasi data	antar,komunikasi,data
TRX-018	Pengembangan Aplikasi Visual FoxPro 5.0	kembang,aplikasi,visual,foxpro
TRX-019	Penuntun 10 Menit Microsoft Mail untuk Windows	tuntun,menit,microsoft,mail,windows
TRX-020	Petunjuk Praktis Pembuatan Program FoxBase+	tunjuk,praktis,buat,program,foxbase
TRX-021	Pintar Belajar computer WordStar Release 7.0	pintar,ajar,computer,wordstar,release
TRX-022	Pintar Belajar computer WordStar Release 6.0	pintar,ajar,computer,wordstar,release
TRX-023	Praktikum Microsoft Windows XP	praktikum,microsoft,windows,xp
TRX-024	Rahasia Sukses Menjual Produk Lewat Wordpress e-Commerce	rahasia,sukses,jual,produk,lewat,wordpres s,e commerce
TRX-025	Referensi Cepat Word Prefect	referensi,cepat,word,prefect
TRX-026	Struktur data Menggunakan TurboPascal 6.0	struktur,data,guna,turbopascal
TRX-027	Teknik Pemrograman Aplikasi Print dari Keyboard	teknik,program,aplikasi,print,keyboard
TRX-028	Visual Basic Pemrograman Grafis dan Multimedia	visual,basic,program,grafis,multimedia
TRX-029	Visual FoxPro 3.0	visual,foxpro
TRX-030	WordStar Spell Star Mail Merge	wordstar,spell,star,mail,merge

Algoritma Eclat merepresentasikan data secara vertical dalam database (untuk setiap *item* dalam kasus ini merupakan *keyword*, dibuat *list/array*

transaksi). Dengan nilai TID (Transaksi ID) diambil dari idTrxPinjam, misal TID 1 = TRX-001, TID 2 = TRX-002, dan seterusnya. Sehingga dari tabel 3.7 didapatkan bentuk vertikal yang dapat dilihat pada tabel 3.8.

Tabel 3.8 Tabel Contoh Data Transaksi dalam vertikal

Keyword	TID
ajar	1,21,22
ancang	4
antar	17
aplikasi	7,18,27
bahasa	7
bangun	7
basic	3,8,11,12,15,28
basics	2
basis	16
big	2
book	2
buat	20
buku	3
cepat	25
cetak	4
clien	15
computer	21,22
coreldraw	1
data	16,17,26
database	3,4,7
developers	12
e commerce	24
era	5
excel	9
foxbase	20
foxpro	7,18,29
grafis	28
guna	26
hari	1
informasi	5
jual	24
kembang	18
keyboard	27
komunikasi	17
konsep	4

kuasa	8
langgan	5
latih	3
layan	5
lewat	24
mail	19,30
mastering	6
menit	19
merge	30
microsoft	2,6,9,10,11,12,13,15,19,23
multimedia	28
office	2,6,10
pagemaker	14
pintar	21,22
praktikum	23
praktis	20
prefect	25
print	27
produk	24
professional	6
proffessional	10
program	3,15,16,20,27,28
rahasia	24
referensi	25
release	21,22
server	15
spell	30
sql	7
star	30
struktur	26
sukses	24
teknik	27
tunjuk	20
tuntun	19
turbopascal	26
visual	3,7,8,11,12,15,18,28,29
windows	2,6,8,14,19,23
word	13,25
wordpress	24
wordstar	21,22,30
workshop	12
xp	23

3.5.1 Pembentukan Frequent 1-itemset

Untuk pembentukan *Frequent 1-Itemset* dihitung nilai *support* dari semua *item*. *Frequent 1-Itemset* adalah *item* (keyword buku) yang memenuhi *min_support*, pada contoh perhitungan manual ini *min_support* yang ditentukan adalah 10% .

$$\text{Support} = P(A \cap B) = \frac{\text{jumlah transaksi yang memuat item A dan item B}}{\text{total jumlah transaksi}}$$

Untuk perhitungan support keyword ajar =

$$\text{Support} = P(\text{ajar}) = \frac{\text{jumlah transaksi yang memuat keyword "ajar"}}{\text{total jumlah transaksi}}$$

$$\text{Support} = P(\text{ajar}) = \frac{3}{30} = 0,1$$

Karena support berupa % maka $\text{Support} = P(1) = 0,1 \times 100\% = 10\%$.

Sesuai dengan contoh perhitungan *support*, hasil perhitungan *support* dari tiap *keyword* buku dapat dilihat pada tabel 3.9.

Tabel 3.9 Hasil Perhitungan Support setiap Item(Keyword)

Keyword	TID	SUPPORT
ajar	1,21,22	10,00
ancang	4	3,33
antar	17	3,33
aplikasi	7,18,27	10,00
bahasa	7	3,33
bangun	7	3,33
basic	3,8,11,12,15,28	20,00
basics	2	3,33
basis	16	3,33
big	2	3,33
book	2	3,33
buat	20	3,33
buku	3	3,33
cepat	25	3,33
cetak	4	3,33
clien	15	3,33
computer	21,22	6,67
coreldraw	1	3,33
data	16,17,26	10,00

database	3,4,7	10,00
developers	12	3,33
e commerce	24	3,33
era	5	3,33
excel	9	3,33
foxbase	20	3,33
foxpro	7,18,29	10,00
grafis	28	3,33
guna	26	3,33
hari	1	3,33
informasi	5	3,33
jual	24	3,33
kembang	18	3,33
keyboard	27	3,33
komunikasi	17	3,33
konsep	4	3,33
kuasa	8	3,33
langgan	5	3,33
latih	3	3,33
layan	5	3,33
lewat	24	3,33
mail	19,30	6,67
mastering	6	3,33
menit	19	3,33
merge	30	3,33
microsoft	2,6,9,10,11,12,13,15,19,23	33,33
multimedia	28	3,33
office	2,6,10	10,00
pagemaker	14	3,33
pintar	21,22	6,67
praktikum	23	3,33
praktis	20	3,33
prefect	25	3,33
print	27	3,33
produk	24	3,33
professional	6	3,33
proffessional	10	3,33
program	3,15,16,20,27,28	20,00
rahasia	24	3,33
referensi	25	3,33
release	21,22	6,67
server	15	3,33

spell	30	3,33
sql	7	3,33
star	30	3,33
struktur	26	3,33
sukses	24	3,33
teknik	27	3,33
tunjuk	20	3,33
tuntun	19	3,33
turbopascal	26	3,33
visual	3,7,8,11,12,15,18,28,29	30,00
windows	2,6,8,14,19,23	20,00
word	13,25	6,67
wordpress	24	3,33
wordstar	21,22,30	10,00
workshop	12	3,33
xp	23	3,33

Dari tabel 3.9 dapat diketahui *frequent 1-itemset* yang memenuhi *min_support*. Dapat dilihat pada tabel 3.10.

Tabel 3.10 Frequent 1-itemset

Keyword	TID	SUPPORT
ajar	1,21,22	10,00
aplikasi	7,18,27	10,00
basic	3,8,11,12,15,28	20,00
data	16,17,26	10,00
database	3,4,7	10,00
foxpro	7,18,29	10,00
microsoft	2,6,9,10,11,12,13,15,19,23	33,33
office	2,6,10	10,00
program	3,15,16,20,27,28	20,00
visual	3,7,8,11,12,15,18,28,29	30,00
windows	2,6,8,14,19,23	20,00
wordstar	21,22,30	10,00

3.5.2 Pembentukan Frequent k-Itemset

Dari *frequent 1-itemset* sesuai dengan algoritma Eclat maka ditemukan S_k (*equivalence classes*) sebanyak dua belas item. Dari tiap S_k dicari *frequent itemset* yang memenuhi *min_support* untuk kemudian dibuat *rule-rule* yang memenuhi *min_support*.

Untuk perhitungan *support Itemset* basic,microsoft pada Kelas Basic =

Support = P(basic $\cap$ microsoft)

= $\frac{\text{jumlah transaksi yang memuat keyword basic dan keyword microsoft}}{\text{total jumlah transaksi}}$

Support = P(basic $\cap$ microsoft) = $\frac{3}{30} = 0,1$

Karena *support* berupa % maka Support = P(1 $\cap$ 3) = $0,1 \times 100\% = 10\%$.

Sesuai dengan contoh perhitungan *Support*, hasil perhitungan *support 2-Itemset* dari untuk tiap kelas dapat dilihat pada tabel 3.11.

Tabel 3.11 Hasil Perhitungan Support 2-Itemset tiap Kelas

Class	No	Itemset	Perhitungan	Support (%)
ajar	1	ajar,aplikasi	$0/30 \times 100$	0,00
	2	ajar,basic	$0/30 \times 100$	0,00
	3	ajar,data	$0/30 \times 100$	0,00
	4	ajar,database	$0/30 \times 100$	0,00
	5	ajar,foxpro	$0/30 \times 100$	0,00
	6	ajar,microsoft	$0/30 \times 100$	0,00
	7	ajar,office	$0/30 \times 100$	0,00
	8	ajar,program	$0/30 \times 100$	0,00
	9	ajar,visual	$0/30 \times 100$	0,00
	10	ajar,windows	$0/30 \times 100$	0,00
	11	ajar,wordstar	$2/30 \times 100$	6,67
aplikasi	1	aplikasi,basic	$0/30 \times 100$	0,00
	2	aplikasi,data	$0/30 \times 100$	0,00
	3	aplikasi,database	$1/30 \times 100$	3,33
	4	aplikasi,foxpro	$2/30 \times 100$	6,67
	5	aplikasi,microsoft	$0/30 \times 100$	0,00
	6	aplikasi,office	$0/30 \times 100$	0,00
	7	aplikasi,program	$1/30 \times 100$	3,33
	8	aplikasi,visual	$2/30 \times 100$	6,67
	9	aplikasi,windows	$0/30 \times 100$	0,00
	10	aplikasi,wordstar	$0/30 \times 100$	0,00
basic	1	basic,data	$0/30 \times 100$	0,00
	2	basic,database	$1/30 \times 100$	3,33
	3	basic,foxpro	$0/30 \times 100$	0,00
	4	basic,microsoft	$3/30 \times 100$	10,00
	5	basic,office	$0/30 \times 100$	0,00
	6	basic,program	$3/30 \times 100$	10,00

data	7	basic,visual	6/30*100	20,00
	8	basic,windows	1/30*100	3,33
	9	basic,wordstar	0/30*100	0,00
	1	data,database	0/30*100	0,00
	2	data,foxpro	0/30*100	0,00
	3	data,microsoft	0/30*100	0,00
	4	data,office	0/30*100	0,00
	5	data,program	1/30*100	3,33
	6	data,visual	0/30*100	0,00
database	7	data,windows	0/30*100	0,00
	8	data,wordstar	0/30*100	0,00
	1	database,foxpro	1/30*100	3,33
	2	database,microsoft	0/30*100	0,00
	3	database,office	0/30*100	0,00
	4	database,program	1/30*100	3,33
	5	database,visual	2/30*100	6,67
foxpro	6	database,windows	0/30*100	0,00
	7	database,wordstar	0/30*100	0,00
	1	foxpro,microsoft	0/30*100	0,00
	2	foxpro,office	0/30*100	0,00
	3	foxpro,program	0/30*100	0,00
	4	foxpro,visual	3/30*100	10,00
microsoft	5	foxpro,windows	0/30*100	0,00
	6	foxpro,wordstar	0/30*100	0,00
	1	microsoft,office	3/30*100	10,00
	2	microsoft,program	1/30*100	3,33
	3	microsoft,visual	3/30*100	10,00
office	4	microsoft,windows	4/30*100	13,33
	5	microsoft,wordstar	0/30*100	0,00
	1	office,program	0/30*100	0,00
	2	office,visual	0/30*100	0,00
program	3	office,windows	2/30*100	6,67
	4	office,wordstar	0/30*100	0,00
	1	program,visual	3/30*100	10,00
visual	2	program,windows	0/30*100	0,00
	3	program,wordstar	0/30*100	0,00
windows	1	visual,windows	1/30*100	3,33
	2	visual,wordstar	0/30*100	0,00
	1	windows,wordstar	0/30*100	0,00

Dari tabel 3.11 didapatkan *Frequent 2-itemset* yang dapat dilihat pada tabel 3.12

Table 3.12 Tabel *Frequent 2-Itemset*

No	Itemset	Perhitungan	Support (%)
1	basic,microsoft	3/30*100	10,00
2	basic,program	3/30*100	10,00
3	basic,visual	6/30*100	20,00
4	foxpro,visual	3/30*100	10,00
5	microsoft,office	3/30*100	10,00
6	microsoft,visual	3/30*100	10,00
7	microsoft,windows	4/30*100	13,33
8	program,visual	3/30*100	10,00

Dari *frequent 2-itemset* ditemukan dua kelas sesuai dengan Algoritma Eclat untuk mencari *frequent 3-itemset*. Dua kelas tersebut dapat dilihat pada tabel 3.13.

Untuk perhitungan support *Itemset* basic,microsoft,visual pada Kelas Basic =

Support = $P(\text{basic} \cap \text{microsoft} \cap \text{visual})$

= $\frac{\text{jumlah transaksi yang memuat keyword basic dan keyword microsoft dan keyword visual}}{\text{total jumlah transaksi}}$

$$\text{Support} = P(\text{basic} \cap \text{microsoft} \cap \text{visual}) = \frac{3}{30} = 0,1$$

Karena support berupa % maka

$$\text{Support} = P(\text{basic} \cap \text{microsoft} \cap \text{visual}) = 0,1 \times 100\% = 10\%.$$

Sesuai dengan contoh perhitungan *support*, hasil perhitungan *support* dari tiap 3-*Itemset* untuk tiap kelas untuk mencari *frequent 3-itemset* dapat dilihat pada tabel 3.13.

Tabel 3.13 Tabel Hasil Perhitungan Support 3-*Itemset* tiap Kelas

Class	No	Itemset	Perhitungan	Support (%)
basic	1	basic,microsoft,program	2/30*100	6,67
	2	basic,microsoft,visual	3/30*100	10,00
	3	basic,program,visual	3/30*100	10,00
microsoft	1	microsoft,office,visual	0/30*100	0,00
	2	microsoft,office,windows	2/30*100	6,67

Dari perhitungan manual yang telah dilakukan didapatkan *frequent itemset* yang memenuhi *min_support* yang dapat dilihat pada tabel 3.14

Table 3.14 Frequent Itemset

No	Itemset
1	basic,microsoft
2	basic,program
3	basic,visual
4	foxpro,visual
5	microsoft,office
6	microsoft,visual
7	microsoft,windows
8	program,visual
9	basic,microsoft,visual
10	basic,program,visual

Dari *frequent Itemset* pada tabel 3.14 maka terbentuk Rule-Rule pada tabel 3.15.

Tabel 3.15 Rule-Rule yang dihasilkan dari frequent itemset

No	Rule
1	basic => microsoft
2	microsoft => basic
3	basic => program
4	program => basic
5	basic => visual
6	visual => basic
7	foxpro => visual
8	visual => foxpro
9	microsoft => office
10	office => microsoft
11	microsoft => visual
12	visual => microsoft
13	microsoft => windows
14	windows => microsoft
15	program => visual
16	visual => program
17	basic,microsoft => visual
18	basic,visual => microsoft
19	microsoft,visual => basic
20	microsoft,basic => visual
21	visual,basic => microsoft
22	visual,microsoft => basic
23	basic,program => visual
24	basic,visual => program
25	program,basic => visual

26	program,visual => basic
27	visual,basic => program
28	visual,program => basic

3.5.3 Perhitungan Confidence Rule

Pada tahap ini *rule-rule* yang memenuhi *min_support* dihitung nilai *confidencenya* sesuai dengan rumus 2.2.

Untuk perhitungan *Confidence Rule* basic=>visual =

$$\text{Confidence} = P(B|A)$$

$$= \frac{\text{Jumlah transaksi yang mengandung item A dan item B}}{\text{Jumlah transaksi yang mengandung item A}}$$

$$\text{Confidence} = P(\text{basic}|\text{visual})$$

$$= \frac{\text{Jumlah transaksi yang mengandung keyword basic dan keyword visual}}{\text{Jumlah transaksi yang mengandung keyword visual}}$$

$$\text{Confidence} = P(\text{basic}|\text{visual}) = \frac{6}{6} = 1$$

Karena confidence berupa % maka,

$$\text{Confidence} = P(\text{basic}|\text{visual}) = 1 \times 100\% = 100\%.$$

Sesuai dengan contoh perhitungan *confidence*, hasil perhitungan *confidence* untuk setiap rule dapat dilihat pada tabel 3.16.

Tabel 3.16 Hasil Perhitungan Confidence Rule

No	Rule	Perhitungan	Confidence
1	basic => microsoft	3/6*100%	50,00
2	microsoft => basic	3/10*100%	30,00
3	basic => program	3/6*100%	50,00
4	program => basic	3/6*100%	50,00
5	basic => visual	6/6*100%	100,00
6	visual => basic	6/9*100%	66,67
7	foxpro => visual	3/3*100%	100,00
8	visual => foxpro	3/9*100%	33,33
9	microsoft => office	3/7*100%	42,86
10	office => microsoft	3/3*100%	100,00
11	microsoft => visual	3/10*100%	30,00
12	visual => microsoft	3/9*100%	33,33
13	microsoft => windows	4/7*100%	57,14
14	windows => microsoft	4/6*100%	66,67
15	program => visual	3/6*100%	50,00

16	visual => program	$3/9 \times 100\%$	33,33
17	basic,microsoft => visual	$3/3 \times 100\%$	100,00
18	basic,visual => microsoft	$3/6 \times 100\%$	50,00
19	microsoft,visual => basic	$3/3 \times 100\%$	100,00
20	microsoft,basic => visual	$3/3 \times 100\%$	100,00
21	visual,basic => microsoft	$3/6 \times 100\%$	50,00
22	visual,microsoft => basic	$3/3 \times 100\%$	100,00
23	basic,program => visual	$3/3 \times 100\%$	100,00
24	basic,visual => program	$3/6 \times 100\%$	50,00
25	program,basic => visual	$3/3 \times 100\%$	100,00
26	program,visual => basic	$3/3 \times 100\%$	100,00
27	visual,basic => program	$3/6 \times 100\%$	50,00
28	visual,program => basic	$3/3 \times 100\%$	100,00

Dari tabel 3.16 didapatkan rule-rule yang memenuhi *min_confidence* yang pada perhitungan manual ini ditentukan nilai *confidence*-nya 60%. Rule-rule yang memenuhi *min_confidence* ditunjukkan pada tabel 3.17.

Tabel 3.17 Rule-rule memenuhi *min_confidence*

No	Rule	Perhitungan	Confidence
1	basic => visual	$6/6 \times 100\%$	100,00
2	visual => basic	$6/9 \times 100\%$	66,67
3	foxpro => visual	$3/3 \times 100\%$	100,00
4	office => microsoft	$3/3 \times 100\%$	100,00
5	windows => microsoft	$4/6 \times 100\%$	66,67
6	basic,microsoft => visual	$3/3 \times 100\%$	100,00
7	microsoft,basic => visual	$3/3 \times 100\%$	100,00
8	microsoft,visual => basic	$3/3 \times 100\%$	100,00
9	visual,microsoft => basic	$3/3 \times 100\%$	100,00
10	basic,program => visual	$3/3 \times 100\%$	100,00
11	program,basic => visual	$3/3 \times 100\%$	100,00
12	program,visual => basic	$3/3 \times 100\%$	100,00
13	visual,program => basic	$3/3 \times 100\%$	100,00

3.5.4 Perhitungan Lift Ratio

Pada tahap ini rule-rule yang memenuhi *min_confidence* dihitung Lift Ratio-nya sesuai dengan rumus 2.3 untuk menemukan rule-rule yang valid.

Untuk perhitungan Lift Rasio rule basic=>visual =

Confidence Benchmark

$$= \frac{\text{Jumlah transaksi dengan item dalam consequent B}}{\text{jumlah transaksi dalam database}} \times 100\%$$

$$\text{Confidence Benchmark} = \frac{9}{30} \times 100\% = 30\%$$

$$\text{Lift Rasio} = \frac{\text{Confidence (basic} \Rightarrow \text{visual)}}{\text{Confidence Benchmark}}$$

$$\text{Lift Rasio} = \frac{100\%}{30\%} = 3,33$$

Sesuai dengan contoh perhitungan *Lift Rasio*, hasil perhitungan *Lift Rasio* untuk setiap rule yang memenuhi *min_confidence* dapat dilihat pada tabel 3.18.

Tabel 3.18 Perhitungan *Lift Ratio Rule*

No	Rule	Perhitungan Confidence Benchmark	Confidence Benchmark (%)	Confidence(%)	Perhitungan Lift	Lift
1	basic => visual	9/30*100	30,00	100,00	100/30	3,33
2	visual => basic	6/30*100	20,00	66,67	66,67/20	3,33
3	foxpro => visual	9/30*100	30,00	100,00	100/30	3,33
4	office => microsoft	10/30*100	33,33	100,00	100/33,33	3,00
5	windows => microsoft	10/30*100	33,33	66,67	66,67/33,33	2,00
6	basic,microsoft => visual	9/30*100	30,00	100,00	100/30	3,33
7	microsoft,basic => visual	6/30*100	20,00	100,00	100/20	5,00
8	microsoft,visual => basic	9/30*100	30,00	100,00	100/30	3,33
9	visual,microsoft => basic	6/30*100	20,00	100,00	100/20	5,00
10	basic,program => visual	9/30*100	30,00	100,00	100/30	3,33
11	program,basic => visual	9/30*100	30,00	100,00	100/30	3,33
12	program,visual => basic	6/30*100	20,00	100,00	100/20	5,00
13	visual,program => basic	6/30*100	20,00	100,00	100/20	5,00

Dari perhitungan pada tabel 3.18 ditemukan bahwa semua *rule-rule* yang dihasilkan mendapat nilai *lift* lebih dari satu sehingga *rule-rule* tersebut valid dan dapat dijadikan rekomendasi. Dari *rule-rule* yang dihasilkan dapat diketahui bahwa mahasiswa tersebut mendapatkan rekomendasi buku dengan keyword microsoft,basic dan visual.

3.6 Perancangan Uji Coba

Setelah seluruh sistem selesai dibuat, langkah selanjutnya adalah melakukan pengujian dan evaluasi terhadap sistem tersebut. Uji coba sistem digunakan untuk melakukan evaluasi terhadap kevalidan *rule-rule* yang dihasilkan oleh sistem. Tujuannya untuk mengetahui tingkat kevalidan sistem yang dibuat. Tingkat akurasi diukur menggunakan *Lift Ratio*.

Uji coba akan dilakukan dengan menghitung *Lift Ratio rule-rule* yang memenuhi berbagai nilai *min_support* dan *min_confidence* dari jumlah transaksi yang berbeda. Uji coba ditunjukkan pada Tabel 3.19.

Tabel 3.19 Uji Coba

Jumlah Data Transaksi	Min_Support	Min_Confidence	Rule	Lift Ratio
30	10%	60%		
		70%		
		80%		
	20%	60%		
		70%		
		80%		
	30%	60%		
		70%		
		80%		
50	10%	60%		
		70%		
		80%		
	20%	60%		
		70%		
		80%		
	30%	60%		
		70%		
		80%		
75	10%	60%		
		70%		
		80%		
	20%	60%		
		70%		
		80%		
	30%	60%		
		70%		
		80%		

100	10%	70%		
		80%		
		60%		
	20%	70%		
		80%		
		60%		
	30%	70%		
		80%		
		60%		

3.7 Perancangan Interface

User interface adalah bagian yang paling tampak dari sebuah inti dari isi program komputer yang memungkinkan terjadinya interaksi antara pengguna dengan program komputer. Rancangan *user interface* perlu dibuat untuk mendapatkan *user interface* terbaik menurut penggunaanya.

Gambar 3.7 Rancangan *Input* data Porter Stemmer

Gambar 3.8 Rancangan *Input* data pembentukan *Association Rule*

STEMMING		
No	Judul Buku	Keyword
1		
2		
3		
4		
5		

Gambar 3.9 Rancangan Hasil Stemming

Association Rule				
FREQUENT ITEMSET				
No	Jenis	Itemset	Support Itemset	Status
1	Freq1Itemset			
2	...			
3	...			
4	...			
5	FreqnItemset			

Gambar 3.10 Rancangan Hasil Frequent Itemset

Rule yang Dihasilkan				
No	Rule	Confidence	Nilai Lift	Status Rule
1				
2				
3				
4				
5				

Gambar 3.11 Rancangan Hasil Association Rule

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi

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

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan pada penelitian ini adalah sebuah PC (*Personal Computer*) dengan spesifikasi sebagai berikut :

1. *Processor* Intel® Core i5 M330 2.13 GHz
2. Memori 2 GB
3. *Harddisk* dengan kapasitas 320 GB
4. Monitor 14.0"
5. *Keyboard*
6. *Mouse*

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut :

1. Sistem Operasi *Microsoft Windows Vista Home Basic Edition* 32 bit.
2. Xampp 1.7.7
3. PHP 5.3.8
4. MySQL 5.5.16
5. Google Chrome

4.2 Implementasi

Sesuai dengan pembahasan pada BAB III, proses pada sistem ini terbagi menjadi tiga bagian utama yaitu pembentukan frequent itemset, pembentukan rule, dan perhitungan lift dari rule yang dihasilkan.

4.2.1 Frequent Itemset

Tujuan dari proses ini adalah untuk menemukan frequent itemset (*keyword*) dari transaksi peminjaman buku mahasiswa.

4.2.1.1 Fungsi Frequent 1-Itemset

Fungsi ini bertujuan untuk menemukan *frequent 1-itemset* (*keyword*) yang memenuhi minimum support yang hasilnya diinputkan ke dalam database. Fungsi *freq1itemset()* ditunjukkan pada Source Code 4.1.



Frequent 1-Itemset

```

/* Mencari Frequent 1-Itemset */
$arr_trx_porter = $arr_data_porter["arr_data"];
$keyword_awal = array();
$keyword_pembanding = array();
$arr_trk = array();
foreach($arr_trx_porter as $data_porter)
{
    $key = $data_porter["keyword"];
    $trk = $data_porter["idTrxPinjam"];
    $key_awal = explode(" ", $key);
    foreach($key_awal as $value_key)
    {
        //masukkan keyword ke array transaksi
        $arr_trk[$trk][] = $value_key;
        //masukkan keyword ke array untuk dijadikan pembanding
        $keyword_pembanding[] = $value_key;
        //masukkan keyword ke array (tdk boleh double)
        if(in_array($value_key, $keyword_awal)){
            continue;
        }else{
            $keyword_awal[] = $value_key;
        }
    }
}
//urutkan data keyword sesuai abjad
asort($keyword_awal);
// hitung support setiap keyword
foreach($keyword_awal as $val_keys){
    //mencari total transaksi utk tiap keyword
    $totkeyword = count(array_keys($keyword_pembanding, $val_keys));
    $totTrx = $data_porter["jmlTrx"];
    $sup_key = $totkeyword/$totTrx * 100;
    $min_sup = $arr_dataInput["minSup"];
    //jika >= min support maka memenuhi + input ke tbl_freq_itemset
    $this->str_tableName = "freq_itemset";
    $this->primKeyFormatFreqItem = "FreqIt-";
    if($sup_key >= $min_sup)
    {
        $arr_dataInputFreq1Itemset["idFreqIt"] = $this->create_newPrimKey("idFreqIt",
        $this->primKeyFormatFreqItem, 2);
        $arr_dataInputFreq1Itemset["idTrxARule"] = $arr_dataInput["idTrxARule"];
        $arr_dataInputFreq1Itemset["itemset"] = $val_keys;
        $arr_dataInputFreq1Itemset["sup"] = $sup_key;
        $arr_dataInputFreq1Itemset["statSup"] = "Memenuhi";
        $arr_dataInputFreq1Itemset["jenisFreq"] = "Freq1Itemset";
        $arr_dataInputFreq1Itemset["class"] = "";
    }else{
        $arr_dataInputFreq1Itemset["idFreqIt"] = $this->create_newPrimKey("idFreqIt",
        $this->primKeyFormatFreqItem, 2);
        $arr_dataInputFreq1Itemset["idTrxARule"] = $arr_dataInput["idTrxARule"];
        $arr_dataInputFreq1Itemset["itemset"] = $val_keys;
        $arr_dataInputFreq1Itemset["sup"] = $sup_key;
        $arr_dataInputFreq1Itemset["statSup"] = "Tidak Memenuhi";
        $arr_dataInputFreq1Itemset["jenisFreq"] = "Freq1Itemset";
        $arr_dataInputFreq1Itemset["class"] = "";
    }
    //printData($arr_dataInputFreq1Itemset);
    $this->insertData($arr_dataInputFreq1Itemset);
}
/* end Mencari Frequent 1-Itemset */

```

Source Code 4.1. Fungsi *Frequent 1-Itemset*

4.2.1.2 Fungsi Frequent 2-Itemset

Fungsi ini bertujuan untuk menemukan *frequent 2-itemset (keyword)* dari frequent 1-itemset yang telah ditemukan sebelumnya yang memenuhi minimum support yang hasilnya diinputkan ke dalam database. Fungsi *Freq 2-itemset* ditunjukkan pada Source Code 4.2.

Frequent 2-Itemset

```
$i = 2;
// mencari kombinasi freq 2-itemset + menghitung support
$comb = $this->mdl_trans_arule->getCombinations($freq1,$i);
foreach($comb as $c){
    $keywordcomb = implode(",",$c);
    $class = $c[0];
    $arr_trx = array();
    foreach($c as $datacomb)
    {
        $arr_trx[] = $this->mdl_trans_arule->getTrx($key_trx,$datacomb);
    }
    // cari intersect trx
    $tottrx = count(call_user_func_array('array_intersect', $arr_trx));

    //hitung support
    $supKey = $tottrx/$tot * 100;
    //input ke db freq_itemset
    if($supKey >= $minSUP)
    {
        $arr_dataInputFreq2Itemset["idFreqIt"] = $this->mdl_freq_itemset->getPrimKey();
        $arr_dataInputFreq2Itemset["idTrxARule"] = $idTrxRule;
        $arr_dataInputFreq2Itemset["itemset"] = $keywordcomb;
        $arr_dataInputFreq2Itemset["sup"] = $supKey;
        $arr_dataInputFreq2Itemset["statSup"] = "Memenuhi";
        $arr_dataInputFreq2Itemset["jenisFreq"] = "Freq2Itemset";
        $arr_dataInputFreq2Itemset["class"] = $class;
    }else{
        $arr_dataInputFreq2Itemset["idFreqIt"] = $this->mdl_freq_itemset->getPrimKey();
        $arr_dataInputFreq2Itemset["idTrxARule"] = $idTrxRule;
        $arr_dataInputFreq2Itemset["itemset"] = $keywordcomb;
        $arr_dataInputFreq2Itemset["sup"] = $supKey;
        $arr_dataInputFreq2Itemset["statSup"] = "Tidak Memenuhi";
        $arr_dataInputFreq2Itemset["jenisFreq"] = "Freq2Itemset";
        $arr_dataInputFreq2Itemset["class"] = "";
    }
    $this->mdl_freq_itemset->insertData($arr_dataInputFreq2Itemset);
}
```

Source Code 4.2. Fungsi *Frequent 2-Itemset*

Pada fungsi *Frequent 2-Itemset* terdapat fungsi *getCombinations()* fungsi ini berfungsi untuk membentuk kombinasi dari array *Frequent 1-Itemset* yang ditemukan. Fungsi *getCombinations()* dapat dilihat pada Source Code 4.3.

GetCombinations

```

function getCombinations($base,$n){
    $baselen = count($base);
    if($baselen == 0){
        return;
    }
    if($n == 1){
        $return = array();
        foreach($base as $b){
            $return[] = array($b);
        }
        return $return;
    }else{
        $oneLevelLower = $this->getCombinations($base,$n-1);
        $newCombs = array();

        foreach($oneLevelLower as $oll){
            $lastEl = $oll[$n-2];
            $found = false;
            foreach($base as $key => $b){
                if($b == $lastEl){
                    $found = true;
                    continue;
                }
                //last element found
                if($found == true){
                    //add to combinations with last element
                    if($key < $baselen){
                        $tmp = $oll;
                        $newCombination = array_slice($tmp,0);
                        $newCombination[]=$b;
                        $newCombs[] = array_slice($newCombination,0);
                    }
                }
            }
        }
    }
    return $newCombs;
}

```

Source Code 4.3. Fungsi *getCombinations*

Pada fungsi *Frequent 2-Itemset* terdapat fungsi *getTrx()* yang berfungsi untuk mencari transaksi tiap keyword pada kombinasi yang ditemukan dari fungsi *getCombinations()* yang ada didalam array *\$key_trx*, untuk kemudian dicari *intersectnya* (data transaksi yang sama yang mengandung *keyword* dalam kombinasi). Fungsi *getTrx()* dapat dilihat pada Source Code 4.4.

GetTrx

```
function getTrx($arr,$val){
    $arr_trx = array();
    foreach ($arr as $key => $row)
    {
        foreach($row as $cell)
        {
            if ($cell == $val)
                $arr_trx[] = $key;
        }
    }
    return $arr_trx;
}
```

Source Code 4.4. Fungsi *getTrx*

4.2.1.3 Fungsi Frequent n-Itemset

Fungsi ini bertujuan untuk menemukan *frequent n-itemset* (*keyword*) dengan $n > 2$ dari frequent $n-1$ -itemset yang telah ditemukan sebelumnya yang memenuhi minimum support yang hasilnya diinputkan ke dalam database. Fungsi *Freq n-itemset()* ditunjukkan pada Source Code 4.5.

Frequent n-Itemset

```

// looping untuk mendapat freq n-itemset
for ($n = 3; ;$n++)
{
    $sarr_freqn = $this->mdl_trans_arule->checkfreq($n,$idTrxRule);
    if($sarr_freqn != false){
        $scombn = $this->mdl_trans_arule->getCombinations($sarr_freqn,$n);
        foreach($scombn as $scn){
            $sarr_class = array();
            $keywordcombn = implode(",",$scn);
            //mencari class
            for($i = 0; $i <= $n-2; $i++){
                if(in_array($scn[$i],$sarr_class)){
                    continue;
                }else{
                    $sarr_class[] = $scn[$i];
                }
            }
            $classn = implode(",",$sarr_class);
            $sarr_trxn = array();
            foreach($scn as $datacombn)
            {
                $sarr_trxn[] = $this->mdl_trans_arule->getTrx($key_trx,$datacombn);
            }
            // cari intersect trx
            $stottrxn = count(call_user_func_array('array_intersect', $sarr_trxn));
            //hitung support
            $supKeyn = $stottrxn/$tot * 100;
            //input ke db freq_itemset
            if($supKeyn >= $minSUP)
            {
                $sarr_dataInputFreqnItemset["idFreqIt"] = $this->mdl_freq_itemset->getPrimKey();
                $sarr_dataInputFreqnItemset["idTrxARule"] = $idTrxRule;
                $sarr_dataInputFreqnItemset["itemset"] = $keywordcombn;
                $sarr_dataInputFreqnItemset["sup"] = $supKeyn;
                $sarr_dataInputFreqnItemset["statSup"] = "Memenuhi";
                $sarr_dataInputFreqnItemset["jenisFreq"] = "Freq".$n."Itemset";
                $sarr_dataInputFreqnItemset["class"] = $classn;
            }else{
                $sarr_dataInputFreqnItemset["idFreqIt"] = $this->mdl_freq_itemset->getPrimKey();
                $sarr_dataInputFreqnItemset["idTrxARule"] = $idTrxRule;
                $sarr_dataInputFreqnItemset["itemset"] = $keywordcombn;
                $sarr_dataInputFreqnItemset["sup"] = $supKeyn;
                $sarr_dataInputFreqnItemset["statSup"] = "Tidak Memenuhi";
                $sarr_dataInputFreqnItemset["jenisFreq"] = "Freq".$n."Itemset";
                $sarr_dataInputFreqnItemset["class"] = "";
            }
            $this->mdl_freq_itemset->insertData($sarr_dataInputFreqnItemset);
        }
    }
    else{
        break;
    }
}
}

```

Source Code 4.5. Fungsi *Frequent n-Itemset*

Pada fungsi *Frequent n-Itemset* terdapat fungsi *checkfreq()* yang berfungsi untuk memeriksa apakah ada *itemset* dari *frequent n-1-itemset* yang memenuhi minimum support untuk diolah lebih lanjut oleh fungsi *Frequent n-Itemset*. Fungsi *checkfreq()* ditunjukkan pada Source Code 4.6.

CheckFreq

```
function checkfreq($n,$idTrxRule){
    $freq = $n-1;
    $jnsFreq = "Freq".$freq."Itemset";
    // cari itemset dimana class > 1
    $queryclass = " SELECT Count(jenisfreq) AS totclass,class
                    FROM freq_itemset
                    WHERE `statSup` = 'Memenuhi'
                    AND `jenisfreq` = '".$jnsFreq."'
                    AND `idTrxARule` = '".$idTrxRule."'
                    group by class
                    ";
    $arr_DataClass= $this->freeQuery($queryclass);
    if(!empty($arr_DataClass["arr_data"]))
    {
        $arr_items = array();
        foreach ($arr_DataClass["arr_data"] as $dataClass )
        {
            if($dataClass["totclass"] > 1){
                $class = $dataClass["class"];
                // cari itemset dari class > 1
                $queryitem = " SELECT itemset FROM freq_itemset
                            WHERE `class` = '".$class."'
                            AND `jenisfreq` = '".$jnsFreq."'
                            AND `idTrxARule` = '".$idTrxRule."'
                            AND `statSup` = 'Memenuhi'
                            ";
                $arr_DataItem= $this->freeQuery($queryitem);
                foreach($arr_DataItem["arr_data"] as $dataItem){
                    $arrItemset = explode(",",$dataItem["itemset"]);
                    foreach($arrItemset as $item){
                        if(in_array($item,$arr_items)){
                            continue;
                        }else{
                            $arr_items[] = $item;
                        }
                    }
                }
            }
        }
        return $arr_items;
    }else{
        return false;
    }
}
```

Source Code 4.6. Fungsi *checkfreq*

4.2.2 Pembentukan Rule

Tujuan dari proses ini adalah untuk mendapatkan rule yang memenuhi minimum confidence dari *frequent itemset* yang memenuhi minimum support.

4.2.2.1 Proses Pembentukan Rule

Proses ini bertujuan untuk membentuk rule dengan cara melakukan permutasi pada *frequent itemset* yang memenuhi support yang dihasilkan. Fungsi pembentukan rule ditunjukkan pada Source Code 4.7.

Pembentukan Rule
<pre> \$arr_freq_itemset = \$this->mdl_freq_itemset->getFreqItemset(\$idTrxRule); if(\$arr_freq_itemset != false){ \$arr_rule = array(); if(\$arr_freq_itemset != false){ foreach(\$arr_freq_itemset as \$dataFreq){ // permutasi \$arr_rule[] = \$this->mdl_trans_arule->permutations(\$dataFreq); } } } </pre>

Source Code 4.7. Fungsi Pembentukan Rule

Pada fungsi pembentukan rule terdapat fungsi *getFreqItemset()* yang berfungsi mengambil data frequent itemset yang memenuhi minimum support. Fungsi *getFreqItemset()* ditunjukkan pada Source Code 4.8.

Get FreqItemset
<pre> function getFreqItemset(\$id){ \$qFreqItems = " Select itemset FROM freq_itemset WHERE statSup = 'Memenuhi' AND idTrxARule = '". \$id. "' AND jenisFreq <> 'Freq1Itemset' "; \$arr_DataFreqItem= \$this->freeQuery(\$qFreqItems); if(!empty(\$arr_DataFreqItem)){ \$arr_freqItems = array(); foreach(\$arr_DataFreqItem["arr_data"] as \$dataFreqItems){ \$arrFreq = explode(",", \$dataFreqItems["itemset"]); \$arrFr = array(); foreach(\$arrFreq as \$datafr){ \$arrFr[] = \$datafr; } arr_freqItems[] = \$arrFr; } return \$arr_freqItems; }else{ return false; } } </pre>

Source Code 4.8. Fungsi *getFreqItemset*

Pada fungsi pembentukan rule juga terdapat fungsi *permutations()* yang berfungsi membentuk permutasi dari setiap itemset untuk dijadikan rule. Fungsi *permutations()* ditunjukkan pada Source Code 4.9.

Permutasi
<pre>function permutations(\$array){ \$list = array(); \$array_count = count(\$array); \$number_of_permutations = 1; if (\$array_count > 1) { for (\$i = 1; \$i <= \$array_count; \$i++) { \$number_of_permutations *= \$i; } } for (\$i=0; count(\$list) < \$number_of_permutations; \$i++) { shuffle(\$array); \$tmp = implode(',', \$array); if (!isset(\$list[\$tmp])) { \$list[\$tmp] = 1; } } ksort(\$list); \$list = array_keys(\$list); return \$list; }</pre>

Source Code 4.9. Fungsi *permutations*

4.2.2.2 Perhitungan Confidence

Proses ini bertujuan untuk menghitung nilai confidence dari rule-rule yang telah dihasilkan untuk dimasukkan ke dalam database tabel *trx_detil_arule*. Fungsi perhitungan confidence ditunjukkan pada Source Code 4.10.

Perhitungan Confidence

```

foreach($arr_rule as $dataRule){
    foreach($dataRule as $valRule){
        //memecah rule
        $rule1 = explode(",", $valRule);
        $totn = count($rule1);
        //cari trx antecedent(A) dan consequent(B)
        $arr_trxR = array();
        foreach($rule1 as $r){
            $arr_trxR[] = $this->mdl_trans_arule->getTrx($key_trx, $r);
        }
        $tottrxR = count(call_user_func_array('array_intersect', $arr_trxR));
        //cari trx dengan item dalam antecedent (A)
        $arr_trxRA = array();
        for($j=0; $j<$totn-1;$j++){
            $arr_trxRA[] = $this->mdl_trans_arule->getTrx($key_trx, $rule1[$j]);
        }
        if($totn > 2){
            $tottrxRA = count(call_user_func_array('array_intersect', $arr_trxRA));
        }else{
            $tottrxRA = count($arr_trxRA[0]);
        }
        $confidence = $tottrxR/$tottrxRA *100;
        //input ke db rule
        if($confidence >= $minConf)
        {
            $arr_dataInputRule["idDetArule"] = $this->mdl_trx_detil_arule->getPrimKey();
            $arr_dataInputRule["idTrxARule"] = $idTrxRule;
            $arr_dataInputRule["rule"] = $valRule;
            $arr_dataInputRule["confidence"] = $confidence;
            $arr_dataInputRule["statRule"] = "Valid";
        }else{
            $arr_dataInputRule["idDetArule"] = $this->mdl_trx_detil_arule->getPrimKey();
            $arr_dataInputRule["idTrxARule"] = $idTrxRule;
            $arr_dataInputRule["rule"] = $valRule;
            $arr_dataInputRule["confidence"] = $confidence;
            $arr_dataInputRule["statRule"] = "No Valid";
        }
        $this->mdl_trx_detil_arule->insertData($arr_dataInputRule);
    }
}

```

Source Code 4.10. Fungsi Perhitungan Confidence

4.2.3 Perhitungan Lift Ratio

Proses ini bertujuan untuk menghitung nilai Lift Ratio dari rule-rule yang memenuhi minimum confidence yang telah dihasilkan oleh proses pembentukan rule. Fungsi perhitungan lift ratio ditunjukkan pada Source Code 4.11.

Perhitungan Lift Ratio

```

$sarr_ruleValid = $this->mdl_trx_detil_arule->getValidRule($idTrxRule);
foreach($sarr_ruleValid as $dataValRule){
    //hitung confidence benchmark
    $confRule = $dataValRule["confidence"];
    $idDetil = $dataValRule["idDetArule"];
    $sarr_val_rule = explode(",", $dataValRule["rule"]);
    $totRule = count($sarr_val_rule);
    $B = $totRule-1;
    $trxB = count($this->mdl_trans_arule->getTrx($key_trx, $sarr_val_rule[$B]));
    $confBenchmark = $trxB/$tot*100;
    //hitung lift
    $lift = $confRule/$confBenchmark;
    //update ke db detil arule
    $sarr_dataUpdate["lift"] = $lift;
    $this->mdl_trx_detil_arule->updateData($sarr_dataUpdate, array("idDetArule" =>
$idDetil));
}

```

Source Code 4.11. Fungsi Perhitungan Lift Ratio

Pada fungsi perhitungan lift ratio terdapat fungsi *getValidRule()* yang berfungsi untuk mendapatkan rule-rule yang memenuhi minimum confidence. Fungsi *getValidRule()* ditunjukkan pada Source Code 4.12.

GetValidRule

```

function getValidRule($id){
    $qValidRule= " Select idDetArule,rule,confidence
                  FROM trx_detil_arule
                  WHERE
                  statRule = 'Valid' AND
                  idTrxARule = '". $id."'
                  ";
    $sarr_ValidRule= $this->freeQuery($qValidRule);
    if(!empty($sarr_ValidRule)){
        return $sarr_ValidRule["arr_data"];
    }else{
        return false;
    }
}

```

Source Code 4.12. Fungsi *getValidRule*

4.3 Implementasi Antarmuka

Pada sistem ini terdapat dua menu utama untuk pembentukan rule yaitu *Porter Stemming* dan *Association Rule* dapat dilihat pada gambar 4.1.

Untuk melakukan pembentukan rule, yang pertama dilakukan *user* adalah melakukan pre-processing data yaitu mengambil keyword-keyword dari judul buku dari transaksi peminjaman buku mahasiswa. Proses ini ditunjukkan pada menu *Porter Stemming* yang dapat dilihat pada gambar 4.2. Pada proses ini *user* memilih mahasiswa mana yang transaksi peminjaman bukunya akan digunakan, kemudian memasukkan jumlah transaksi yang diinginkan, dan memasukkan tanggal transaksi yang akan digunakan. Setelah data-data yang diinginkan telah dimasukkan klik tombol transaksi. Maka hasil porter stemming akan ditampilkan seperti yang dapat dilihat pada gambar 4.3.



Gambar 4.1 Menu Utama Pembentukan Rule

Gambar 4.2 Menu Porter Stemming

Porter Stemmer

30 Data

No	Judul Buku	Keyword
1	Belajar CorelDraw dalam Sehari	ajar,coreldraw,dalam,hari
2	Big Basics Book untuk Microsoft Office untuk Windows 95	big,basics,book,microsoft,office,windows
3	Buku Latihan Pemrograman DataBase dengan Visual Basic 6.0	buku,latih,pemrograman,database,visual,basic
4	Konsep & perancangan DataBase / Cetakan 2	konsep,ancang,database,cetak
5	Layanan Pelanggan Di Era Informasi	layan,langgan,era,informasi
6	Mastering Microsoft Office Professional untuk Windows 95	mastering,microsoft,office,professional,windows
7	Membangun Aplikasi DataBase Dengan Visual FoxPro 8.0 dan Bahasa SQL	bangun,aplikasi,database,visual,foxpro,bahasa,sql
8	Menguasai Visual Basic untuk Windows	kuasa,visual,basic,windows
9	Microsoft Excel 97	microsoft,excel
10	Microsoft Office 4.3 Proffessional	microsoft,office,proffessional
11	Microsoft Visual Basic	microsoft,visual,basic
12	Microsoft Visual Basic 4.0 Developers Workshop	microsoft,visual,basic,developers,workshop
13	Microsoft Word 97	microsoft,word
14	PageMaker 5.0 untuk Windows	pagemaker, windows
15	Pemrograman Clien / Server Dengan Microsoft Visual Basic	pemrograman,clien,server,microsoft,visual,basic
16	Pemrograman Basis data	program,basis,data
17	Pengantar Komunikasi data	antar,komunikasi,data
18	Pengembangan Aplikasi Visual FoxPro 5.0	kembang,aplikasi,visual,foxpro
19	Penuntun 10 Menit Microsoft Mail untuk Windows	tuntun,menit,microsoft,mail,windows
20	Petunjuk Praktis Pembuatan Program FoxBase+	tunjuk,praktis,buat,program,foxbase
21	Pintar Belajar computer WordStar Release 7.0	pintar,ajar,computer,wordstar,release
22	Pintar Belajar computer WordStar Release 7.0	pintar,ajar,computer,wordstar,release
23	Praktikum Microsoft Windows XP	praktikum,microsoft, windows, xp
24	Rahasia Sukses Menjual Produk Lewat Wordpress e-Commerce	rahasia,sukses,jual,produk,lewat,wordpress,e-commerce
25	Referensi Cepat Word Prefect	referensi,cepat,word,prefect
26	Struktur data Menggunakan TurboPascal 6.0	struktur,data,guna,turbopascal
27	Teknik Pemrograman Aplikasi Print dari Keyboard	teknik,program,aplikasi,print,keyboard
28	Visual Basic Pemrograman Grafis dan Multimedia	visual,basic,program,grafis,multimedia
29	Visual FoxPro 3.0	visual,foxpro
30	WordStar Spell Star Mail Merge	wordstar,spell,star,mail,merge

Gambar 4.3 Hasil Porter Stemming

Setelah selesai melakukan *porter stemming*, untuk pembentukan rule maka *user* masuk ke menu *Association Rule* yang dapat dilihat pada gambar 4.4. Pada menu ini *user* memasukkan data minimum support dan minimum confidence dan data *Porter Stemming* mana yang dipilih untuk diproses. Setelah data-data diisikan maka *user* klik tombol transaksi. Maka hasil yang ditampilkan adalah Frequent Itemset baik yang memenuhi minimum support maupun tidak yang contoh hasilnya dapat dilihat pada gambar 4.5, Rule-rule yang memenuhi minimum confidence maupun tidak yang dapat dilihat pada gambar 4.6, dan rule-rule yang valid yang memenuhi minimum confidence dan nilai lift rasionya lebih besar dari satu yang dapat dilihat pada gambar 4.7.

Association Rule

Input Data Association Rule

Data Porter Stemmer : 0612651072 - ARDIAN M.H - 30TRANSAKSI *

Minimum Support : 10 %

Minimum Confidence : 60 %

Transaksi

Gambar 4.4 Menu Association Rule

Association Rule

FREQUENT ITEMSET

No	Jenis	Itemset	Support Itemset	Status	class
1	Freq1Itemset	ajar	10.0	Memenuhi	
2	Freq1Itemset	ancang	3.3	Tidak Memenuhi	
3	Freq1Itemset	antar	3.3	Tidak Memenuhi	
4	Freq1Itemset	aplikasi	10.0	Memenuhi	
5	Freq1Itemset	bahasa	3.3	Tidak Memenuhi	
81	Freq2Itemset	ajar,data	0.0	Tidak Memenuhi	ajar
82	Freq2Itemset	ajar,database	0.0	Tidak Memenuhi	ajar
83	Freq2Itemset	ajar,foxpro	0.0	Tidak Memenuhi	ajar
84	Freq2Itemset	ajar,microsoft	0.0	Tidak Memenuhi	ajar
85	Freq2Itemset	ajar,office	0.0	Tidak Memenuhi	ajar
86	Freq2Itemset	ajar,program	0.0	Tidak Memenuhi	ajar
87	Freq2Itemset	ajar,visual	0.0	Tidak Memenuhi	ajar
88	Freq2Itemset	ajar,windows	0.0	Tidak Memenuhi	ajar
89	Freq2Itemset	ajar,wordstar	6.7	Tidak Memenuhi	ajar
90	Freq2Itemset	aplikasi,basic	0.0	Tidak Memenuhi	aplikasi

Gambar 4.5 Contoh Hasil *Freq Itemset*

Rule yang Dihasilkan				
No	Rule	Confidence	Nilai Lift	Status Rule
1	basic,microsoft	50.0		No Valid
2	microsoft,basic	30.0		No Valid
3	basic,visual	100.0	3.3	Valid
4	visual,basic	66.7	3.3	Valid
5	foxpro,visual	100.0	3.3	Valid
6	visual,foxpro	33.3		No Valid
7	microsoft,office	30.0		No Valid
8	office,microsoft	100.0	3.0	Valid
9	microsoft,visual	30.0		No Valid
10	visual,microsoft	33.3		No Valid
11	microsoft,windows	40.0		No Valid
12	windows,microsoft	66.7	2.0	Valid
13	basic,microsoft,visual	100.0	3.3	Valid
14	basic,visual,microsoft	50.0		No Valid
15	microsoft,basic,visual	100.0	3.3	Valid
16	microsoft,visual,basic	100.0	5.0	Valid
17	visual,basic,microsoft	50.0		No Valid
18	visual,microsoft,basic	100.0	5.0	Valid

Gambar 4.6 Rule-rule yang dihasilkan

Rule VALID				
No	Rule	Confidence	Nilai Lift	Status Rule
1	basic,visual	100.0	3.3	Valid
2	visual,basic	66.7	3.3	Valid
3	foxpro,visual	100.0	3.3	Valid
4	office,microsoft	100.0	3.0	Valid
5	windows,microsoft	66.7	2.0	Valid
6	basic,microsoft,visual	100.0	3.3	Valid
7	microsoft,basic,visual	100.0	3.3	Valid
8	microsoft,visual,basic	100.0	5.0	Valid
9	visual,microsoft,basic	100.0	5.0	Valid

Gambar 4.7 Rule-rule valid yang dihasilkan

BAB V

PENGUJIAN DAN ANALISIS

Pada sub bab ini akan dibahas mengenai implementasi dari metode pengujian yang telah dilakukan oleh sistem dan hasil dari pengujian tersebut.

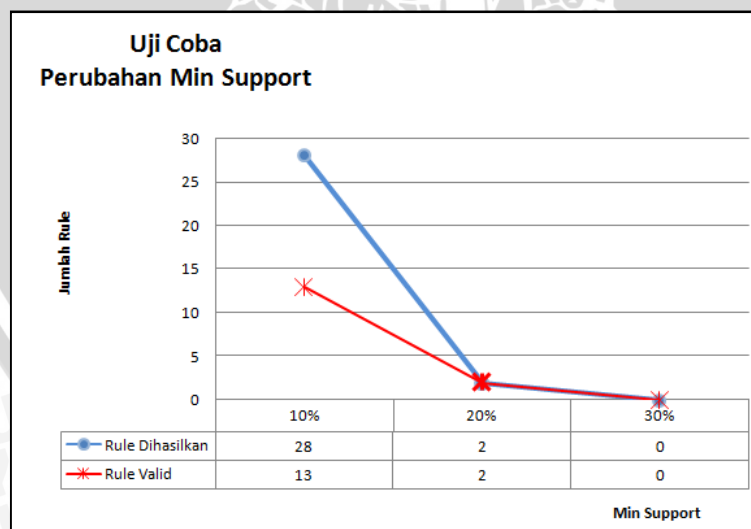
5.1 Skenario Pengujian

Pengujian dilakukan dengan melakukan uji coba proses pembentukan rule dengan perubahan nilai minimum support dan perubahan nilai minimum confidence pada jumlah transaksi yang berbeda. Pengujian ini untuk dilakukan untuk melihat pengaruh minimum support terhadap kevalidan *rule-rule* yang dihasilkan dan juga untuk melihat pengaruh minimum confidence terhadap kevalidan *rule-rule* yang dihasilkan.

5.2 Hasil Pengujian

5.2.1 Hasil Uji Coba Perubahan Nilai Minimum Support

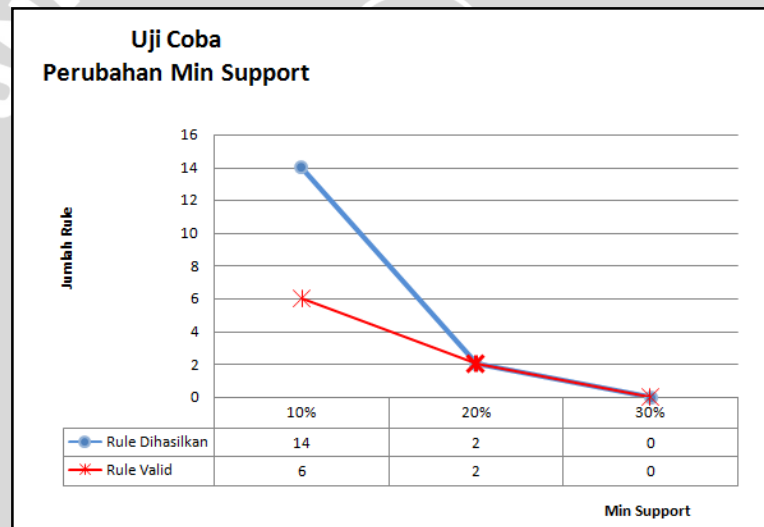
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 30 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.1.



Gambar 5.1. Grafik Uji Coba Jumlah Transaksi 30 nilai Min Confidence 60%

Pada gambar 5.1 pada nilai minimum support 10% dari 28 rule yang dihasilkan hanya 13 rule yang valid, karena hanya 13 rule tersebut yang memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% pada grafik 5.1 dari 2 rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum support 30% pada grafik 5.1 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

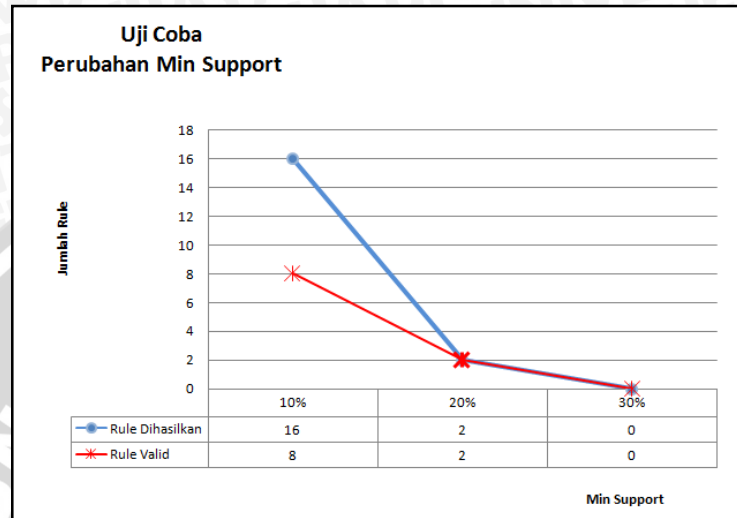
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 50 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.2.



Gambar 5.2. Grafik Uji Coba Jumlah Transaksi 50 nilai Min Confidence 60%

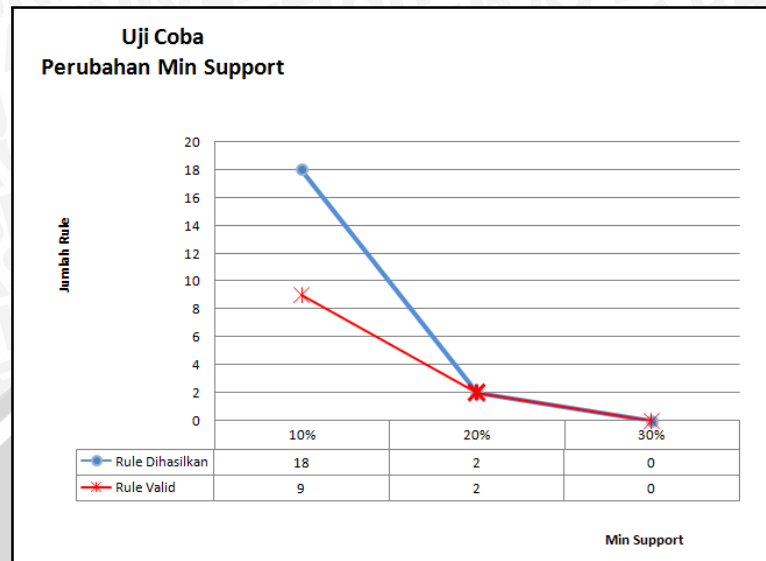
Pada gambar 5.2 pada nilai minimum support 10% dari 14 rule yang dihasilkan hanya 6 rule yang valid, karena hanya 6 rule tersebut yang memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% pada grafik 5.2 dari 2 rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum support 30% pada grafik 5.2 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 75 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.3.



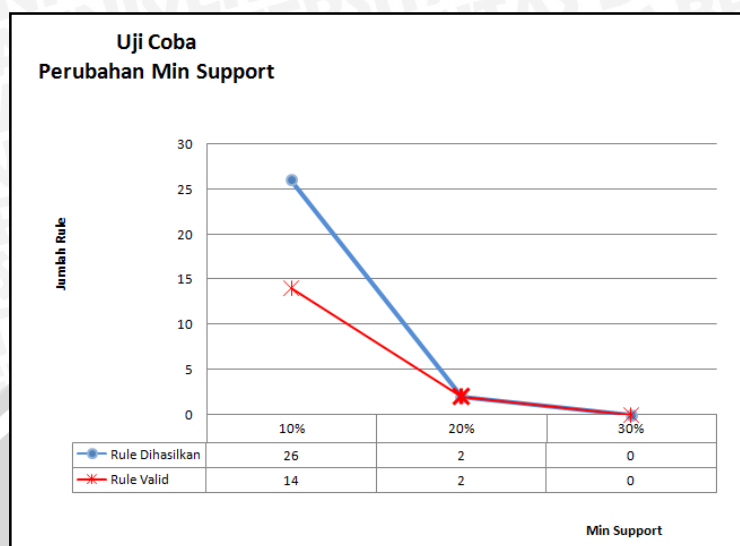
Gambar 5.3. Grafik Uji Coba Jumlah Transaksi 75 nilai Min Confidence 60% Pada gambar 5.3 pada nilai minimum support 10% dari 16 rule yang dihasilkan hanya 8 rule yang valid, karena hanya 6 rule tersebut yang memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% pada grafik 5.3 dari 2 rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum support 30% pada grafik 5.3 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 100 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.4.



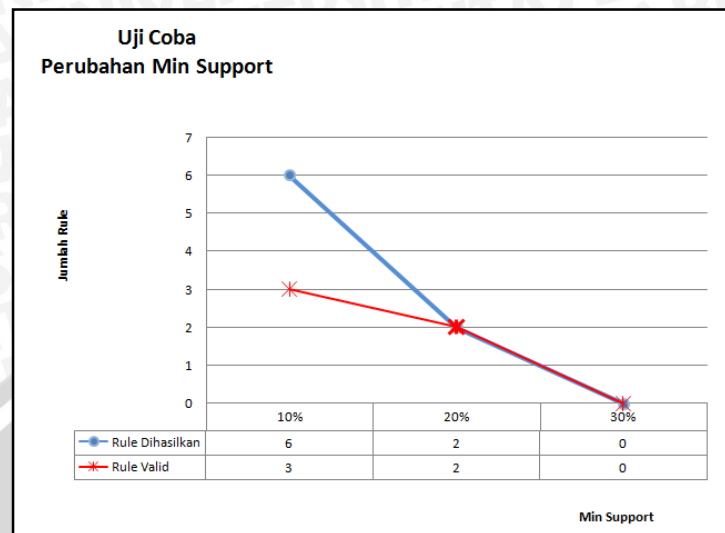
Gambar 5.4. Grafik Uji Coba Jumlah Transaksi 100 nilai Min Confidence 60% Pada gambar 5.4 pada nilai minimum support 10% dari 18 rule yang dihasilkan hanya 9 rule yang valid, karena hanya 9 rule tersebut yang memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% pada grafik 5.4 dari 2 rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum support 30% pada grafik 5.4 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 150 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.5.



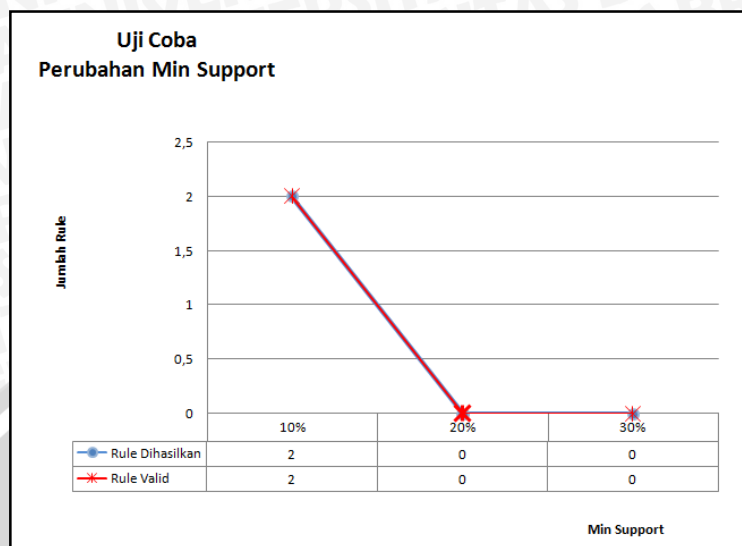
Gambar 5.5. Grafik Uji Coba Jumlah Transaksi 150 nilai Min Confidence 60% Pada gambar 5.5 pada nilai minimum support 10% dari 26 rule yang dihasilkan hanya 14 rule yang valid, karena hanya 14 rule tersebut yang memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% pada grafik 5.5 dari 2 rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum support 30% pada grafik 5.5 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 200 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.6.



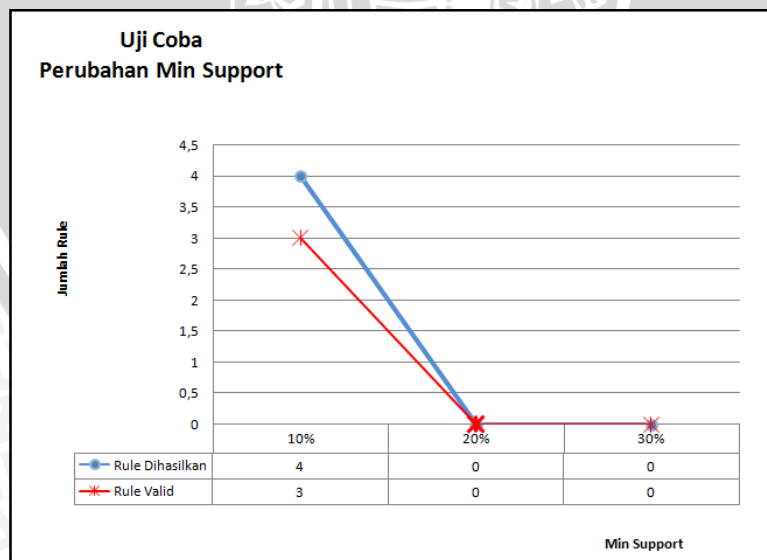
Gambar 5.6. Grafik Uji Coba Jumlah Transaksi 200 nilai Min Confidence 60% Pada gambar 5.6 pada nilai minimum support 10% dari 6 rule yang dihasilkan hanya 3 rule yang valid, karena hanya 3 rule tersebut yang memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% pada grafik 5.6 dari 2 rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum support 30% pada grafik 5.6 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 300 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.7.



Gambar 5.7. Grafik Uji Coba Jumlah Transaksi 300 nilai Min Confidence 60%
 Pada gambar 5.7 pada nilai minimum support 10% dari 2 rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% dan 30% pada grafik 5.7 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

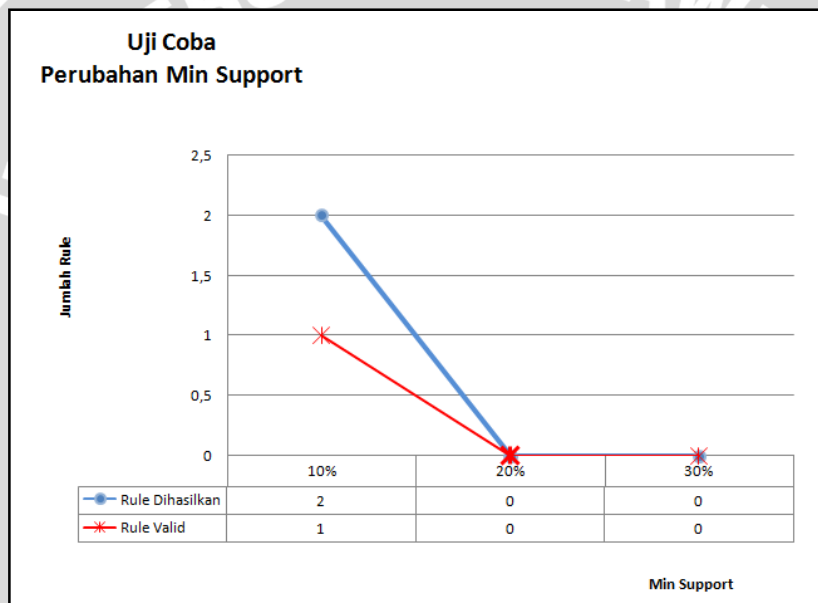
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 400 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.8.



Gambar 5.8. Grafik Uji Coba Jumlah Transaksi 400 nilai Min Confidence 60%

Pada gambar 5.8 pada nilai minimum support 10% dari 4 rule yang dihasilkan hanya 3 rule yang valid, karena hanya 3 rule tersebut yang memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% dan 30% pada grafik 5.8 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum support dengan jumlah transaksi 500 dan nilai minimum confidence 60% dapat dilihat pada gambar 5.9.

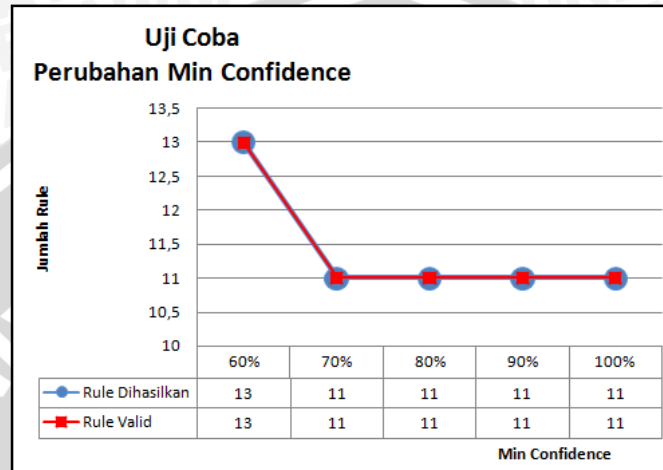


Gambar 5.9. Grafik Uji Coba Jumlah Transaksi 500 nilai Min Confidence 60%

Pada gambar 5.9 pada nilai minimum support 10% dari 2 rule yang dihasilkan hanya 1 rule yang valid, karena hanya 1 rule tersebut yang memenuhi minimum support dan minimum confidence. Pada nilai minimum support 20% dan 30% pada grafik 5.9 tidak ada rule yang dihasilkan karena frequent itemset yang ditemukan tidak ada yang memenuhi minimum support sehingga tidak ada rule yang bisa dihasilkan.

5.2.2 Hasil Uji Coba Perubahan Nilai Minimum Confidence

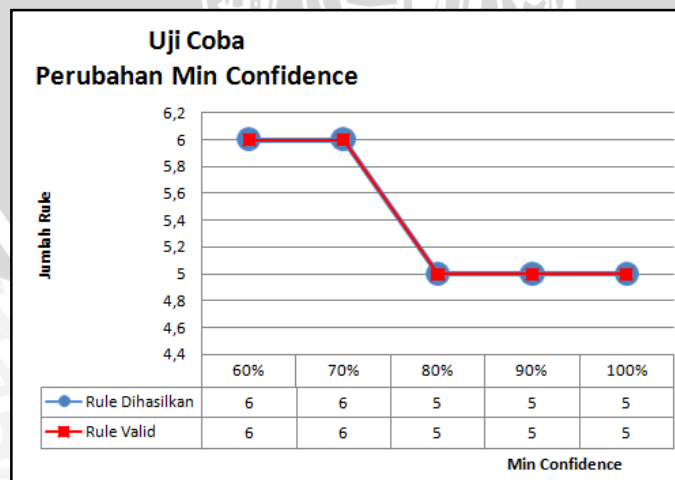
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 30 dan nilai minimum support 10% dapat dilihat pada gambar 5.10.



Gambar 5.10. Grafik Uji Coba Jumlah Transaksi 30 nilai Min Support 10%

Pada gambar 5.10 pada nilai minimum confidence 60%, 70%, 80%, 90% dan 100% dari rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support, minimum confidence dan mempunyai nilai lift ratio diatas 1.

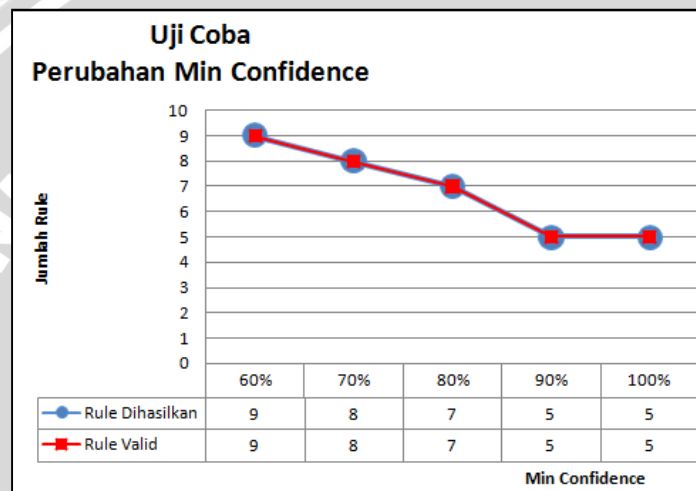
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 50 dan nilai minimum support 10% dapat dilihat pada gambar 5.11.



Gambar 5.11. Grafik Uji Coba Jumlah Transaksi 50 nilai Min Support 10%

Pada gambar 5.11 pada nilai minimum confidence 60%, 70%, 80%, 90% dan 100% dari rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support, minimum confidence dan mempunyai nilai lift ratio diatas 1.

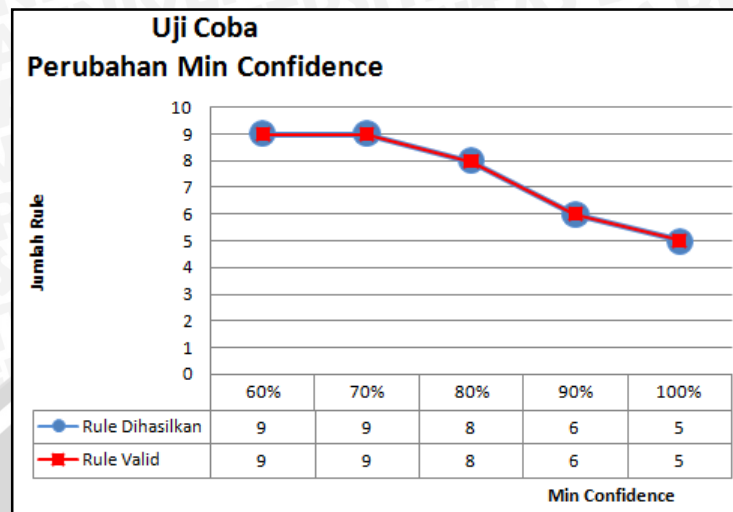
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 75 dan nilai minimum support 10% dapat dilihat pada gambar 5.12.



Gambar 5.12. Grafik Uji Coba Jumlah Transaksi 75 nilai Min Support 10%

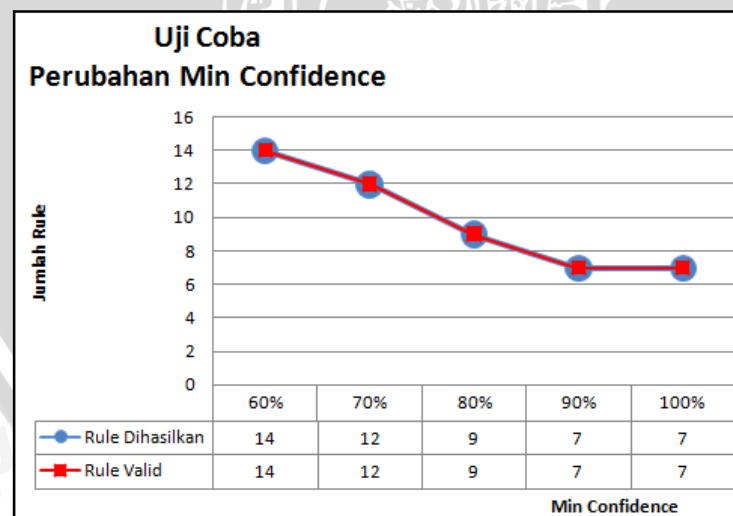
Pada gambar 5.12 pada nilai minimum confidence 60%, 70%, 80%, 90% dan 100% dari rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support, minimum confidence dan mempunyai nilai lift ratio diatas 1.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 100 dan nilai minimum support 10% dapat dilihat pada gambar 5.13.



Gambar 5.13. Grafik Uji Coba Jumlah Transaksi 100 nilai Min Support 10%
 Pada gambar 5.13 pada nilai minimum confidence 60%, 70%, 80%, 90% dan 100% dari rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support, minimum confidence dan mempunyai nilai lift ratio diatas 1.

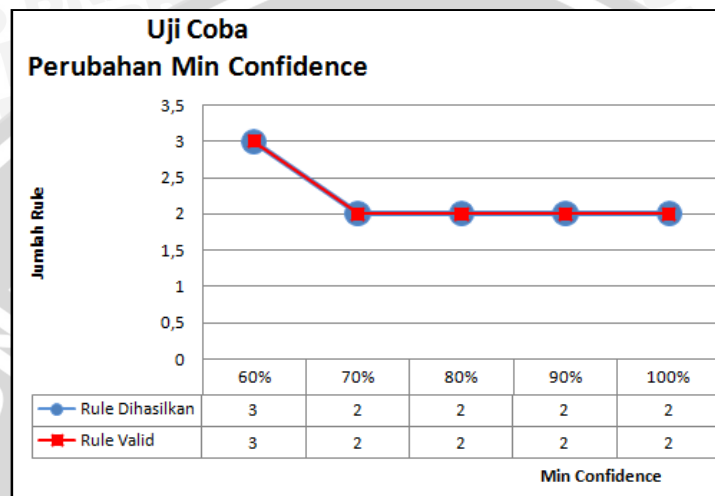
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 150 dan nilai minimum support 10% dapat dilihat pada gambar 5.14.



Gambar 5.14. Grafik Uji Coba Jumlah Transaksi 150 nilai Min Support 10%
 Pada gambar 5.14 pada nilai minimum confidence 60%, 70%, 80%, 90% dan 100% dari rule yang dihasilkan semuanya rule yang valid karena semua rule

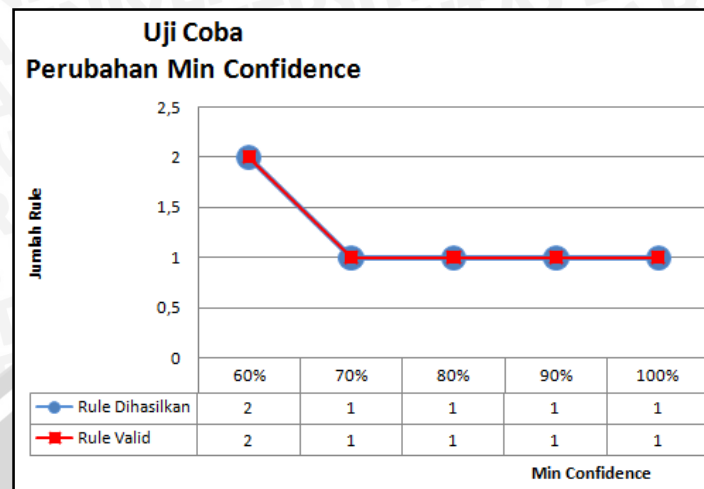
memenuhi minimum support, minimum confidence dan mempunyai nilai lift ratio diatas 1.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 200 dan nilai minimum support 10% dapat dilihat pada gambar 5.15.



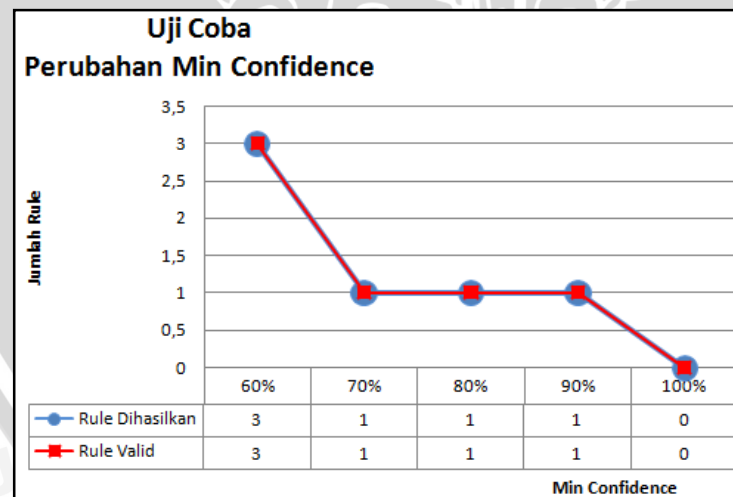
Gambar 5.15. Grafik Uji Coba Jumlah Transaksi 200 nilai Min Support 10%
Pada gambar 5.15 pada nilai minimum confidence 60%, 70%, 80%, 90% dan 100% dari rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support, minimum confidence dan mempunyai nilai lift ratio diatas 1.

Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 300 dan nilai minimum support 10% dapat dilihat pada gambar 5.16.



Gambar 5.16. Grafik Uji Coba Jumlah Transaksi 300 nilai Min Support 10%
 Pada gambar 5.16 pada nilai minimum confidence 60%, 70%, 80%, 90% dan 100% dari rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support, minimum confidence dan mempunyai nilai lift ratio diatas 1.

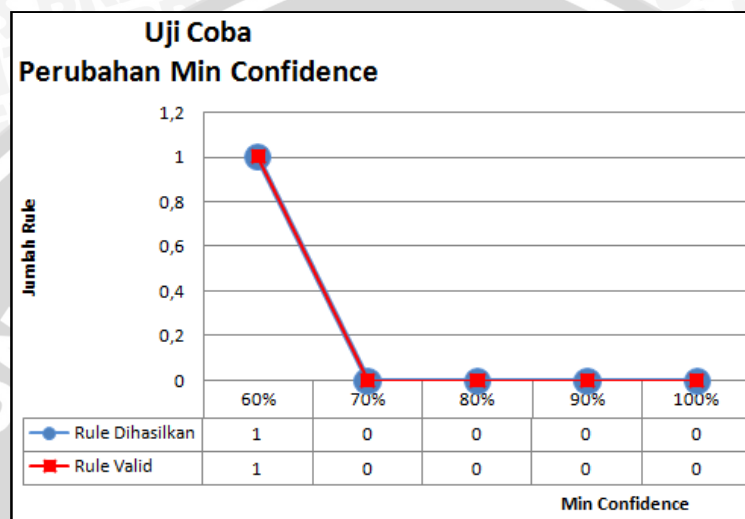
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 400 dan nilai minimum support 10% dapat dilihat pada gambar 5.17.



Gambar 5.17. Grafik Uji Coba Jumlah Transaksi 400 nilai Min Support 10%
 Pada gambar 5.17 pada nilai minimum confidence 60%, 70%, 80%, dan 90% dari rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum confidence

100% tidak ada rule yang dihasilkan dan tidak ada rule yang valid, karena semua rule yang dihasilkan tidak memenuhi nilai minimum confidence 100%.

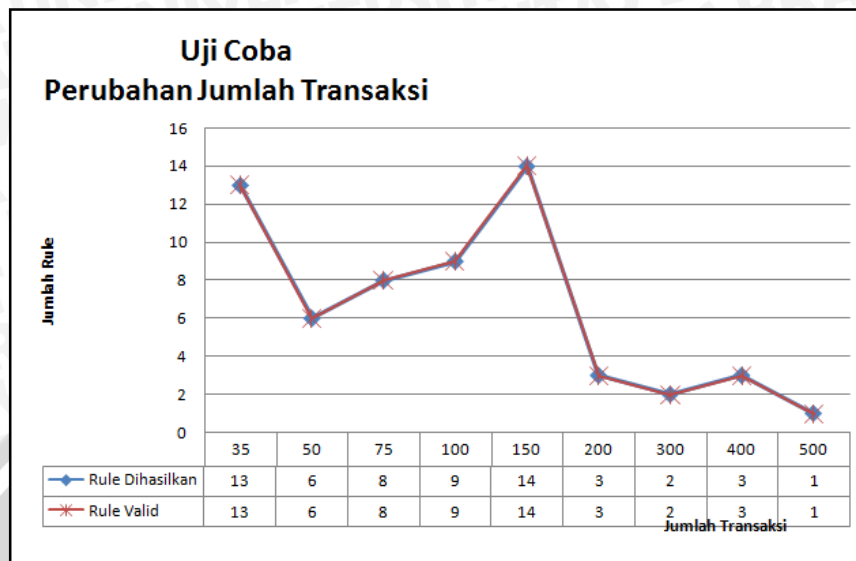
Hasil dari uji coba proses pembentukan rule dengan perubahan nilai minimum confidence dengan jumlah transaksi 500 dan nilai minimum support 10% dapat dilihat pada gambar 5.18.



Gambar 5.18. Grafik Uji Coba Jumlah Transaksi 500 nilai Min Support 10%
 Pada gambar 5.18 pada nilai minimum confidence 60% dari rule yang dihasilkan semuanya rule yang valid karena semua rule memenuhi minimum support dan minimum confidence. Pada nilai minimum support 70%, 80%, 90% dan 100% pada gambar 5.18 tidak ada rule yang dihasilkan tidak ada rule yang dihasilkan dan tidak ada rule yang valid, karena semua rule yang dihasilkan tidak memenuhi nilai minimum confidence.

5.2.3 Hasil Uji Coba Perubahan Jumlah Transaksi

Hasil dari uji coba proses pembentukan rule dengan perubahan jumlah transaksi dengan nilai minimum support 10% dan minimum confidence 60% dapat dilihat pada gambar 5.19



Gambar 5.19. Grafik Uji Coba Perubahan Jumlah Transaksi

Pada gambar 5.19 pada jumlah transaksi 30, 50, 75, 100, 150, 200, 300, 400, dan 500 dari semua rule yang dihasilkan semuanya rule yang valid karena memenuhi minimum support dan minimum confidence.

5.3 Analisis Hasil Pengujian

5.3.1 Analisis Hasil Uji Coba Pembentukan Rule dengan Perubahan Nilai Minimum Support

Dari grafik 5.1 sampai dengan grafik 5.9 dapat diketahui bahwa semakin kecil nilai minimum supportnya maka semakin banyak rule yang dihasilkan yang menunjukkan bahwa performansi sistem berjalan dengan baik. Sedangkan untuk pengaruh perubahan nilai minimum support terhadap kevalidan rule-rule yang dihasilkan dapat dilihat pada grafik 5.1 sampai dengan grafik 5.9 bahwa jumlah rule yang valid lebih kecil daripada rule yang dihasilkan, sehingga dapat disimpulkan bahwa nilai minimum support tidak besar pengaruhnya terhadap kevalidan suatu rule. Dapat disimpulkan juga bahwa kevalidan suatu rule tidak hanya bergantung pada nilai minimum support.

5.3.2 Analisis Hasil Uji Coba Pembentukan Rule dengan Perubahan Nilai Minimum Confidence

Dari grafik 5.10 sampai dengan grafik 5.18 dapat diketahui bahwa semakin kecil nilai minimum confidencenya maka semakin banyak rule yang dihasilkan yang menunjukkan bahwa performansi sistem berjalan dengan baik. Sedangkan untuk pengaruh perubahan nilai minimum confidence terhadap kevalidan rule-rule yang dihasilkan dapat dilihat pada grafik 5.10 sampai dengan grafik 5.18 bahwa jumlah rule yang valid sama dengan jumlah rule yang dihasilkan, sehingga dapat disimpulkan bahwa nilai minimum confidence sangat berpengaruh terhadap kevalidan suatu rule. Dapat disimpulkan juga bahwa setiap rule yang memenuhi nilai minimum confidence besar kemungkinan menjadi rule yang valid.

5.3.3 Analisis Hasil Uji Coba Pembentukan Rule dengan Perubahan Jumlah Transaksi

Dari grafik 5.19 mulai jumlah transaksi 30 sampai dengan jumlah transaksi 500 jumlah rule yang dihasilkan dan jumlah rule yang valid sangat bervariasi, menunjukkan bahwa jumlah transaksi tidak berpengaruh terhadap pembentukan rule dan kevalidan rule, karena rule dibentuk dari itemset yaitu himpunan dari item-item yang muncul bersama-sama yang memenuhi minimum support, sehingga walaupun jumlah transaksi besar dan item-itemnya bervariasi belum tentu item-item tersebut muncul secara bersamaan, begitu juga ketika jumlah transaksi kecil belum tentu item-itemnya muncul secara bersamaan.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil uji coba yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut.

1. Algoritma *eclat* dapat diterapkan pada personalisasi *E-Library* untuk perekomendasi buku untuk tiap mahasiswa. Dengan melakukan *pre-processing* terhadap judul buku yang didapat dari transaksi peminjaman untuk mendapatkan *keyword*-nya. Dari *keyword* yang didapatkan kemudian diterapkan algoritma *eclat* untuk mendapatkan *rule-rule* untuk rekomendasi. Langkah dari algoritma *eclat* yaitu merepresentasikan data secara vertical dalam database (untuk setiap *keyword*, dibuat *list/array* transaksi) untuk kemudian mencari frequent itemset yang memenuhi nilai minimum support, membentuk *rule-rule* dari frequent itemset yang ditemukan, mengambil *rule-rule* yang memenuhi nilai minimum confidence, dari *rule-rule* yang memenuhi nilai confidence dihitung nilai lift rasionya, *rule* yang memiliki nilai lift rasio lebih dari 1 maka *rule* tersebut valid dan bisa dijadikan rekomendasi.
2. Semakin kecil nilai minimum support, semakin banyak *rule* yang dihasilkan tetapi tidak semua *rule* yang dihasilkan valid, yang menunjukkan bahwa nilai minimum support berpengaruh pada pembentukan *rule* tetapi tidak banyak berpengaruh terhadap kevalidan *rule*. Titik optimum dicapai pada nilai minimum support = 10% pada semua jumlah transaksi uji coba.
3. Semakin kecil nilai minimum confidence, semakin banyak *rule* yang dihasilkan dan besar kemungkinan *rule* tersebut valid, yang menunjukkan bahwa nilai minimum confidence berpengaruh pada kevalidan *rule*. Titik optimum dicapai pada nilai minimum confidence = 60% pada semua jumlah transaksi uji coba.

4. Jumlah transaksi tidak berpengaruh dengan kevalidan rule, karena besar jumlah transaksi tidak menjamin item-item di dalamnya muncul bersamaan. Titik optimum dicapai pada jumlah transaksi = 150.

6.2 Saran

Dalam penelitian ini terdapat beberapa hal yang masih belum dieksplorasi secara mendalam dan dapat dijadikan bahan dalam penelitian lebih lanjut, antara lain :

1. Pada proses *pre-processing* seharusnya bisa dipisahkan penggunaan bahasa inggris dan bahasa indonesia sehingga data *keyword* yang didapatkan lebih akurat. Dikarenakan buku dengan bahasa indonesia terkadang mengandung istilah-istilah dari bahasa inggris.
2. Mengembangkan sistem untuk dapat memberikan rekomendasi tidak terbatas hanya buku, tetapi juga media yang lain.

DAFTAR PUSTAKA

- [AGU-09] Agusta, Ledy. 2009. Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia, Konferensi Nasional Sistem dan Informatika 2009; Bali.
- [ARI-10] Ariwibowo, Agus Sasmito. 2010. E-Elearning Cerdas Dengan Personalisasi Menggunakan Teknik Data Mining Dan Decision Support System, Seminar Nasional Informatika 2010, Yogyakarta.
- [BOR-03] Borgelt, C. 2003. Efficient Implementations of Apriori and Eclat, Workshop of Frequent Item Set Mining Implementations, Melbourne, FL, USA.
- [CHI-01] Chiu, W. 2001. "Website personalization", <http://www.ibm.com/developerworks/websphere/library/techarticles/hipods/personalize.html>.
- [CHR-06] Christen, Peter., Kennedy, Paul J., Li, Jiuyong., Simoff, Simeon J. and Williams, Graham J. 2006. Data Mining and Analytics 2006. Fifth Australasian Data Mining Conference (AusDM2006), Sydney, Australia
- [EDE-99] Edelstein, Herbert A. 1999. Introduction to Data Mining and Knowledge Discovery, Third Edition. Two Crows Corporation. USA.
- [HER-10] Herdianto, Aditya. 2010. Pencarian Ayat-Ayat Al-Qur'an Berdasarkan Konten Menggunakan Text Mining Berbasis Desktop.
- [LAR-05] Larose, Daniel T. 2005. Discovering Knowledge in Data: An Introduction to Data Mining, John Wiley & Sons, Inc. Canada.
- [MOE-02] Moertini S. Veronika. 2002. Data Mining Sebagai Solusi Bisnis.
- [PON-06] Ponniah, Paulraj. 2006. Data Warehousing Fundamentals. Wiley India Pvt. Limited. India.
- [SAN-07] Santoso, Budi. 2007. Data Mining: Teknik Pemanfaatan Data Untuk Keperluan Bisnis. Graha Ilmu. Yogyakarta.
- [SIS-08] Sismanto. 2008. Manajemen Perpustakaan Digital, Bandung.

- [SUP-86] Supriyadi. 1986. Pengantar Pengelolaan Perpustakaan Sekolah. Malang: Proyek Peningkatan/Pengembangan Peguruan Tinggi IKIP Malang.
- [WEI-01] weiyang lin, sergio a.alvarez, carolina ruiz. 2001. Efficient Adaptive-support association rule mining for recommender systems. Kluwer Academic Publishers, Netherlands
- [ZAK-97] Zaki, Mohammed J. 1997. Fast Mining of Sequential Patterns in Very Large Databases. The University of Rochester Computer Science Department Rochester. NewYork14627. USA



DAFTAR ISI

LEMBAR PERSETUJUAN

LEMBAR PENGESAHAN

PERNYATAAN ORISINALITAS SKRIPSI

ABSTRAK

ABSTRACT

KATA PENGANTAR

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SOURCE CODE

BAB I PENDAHULUAN 6

1.1 Latar Belakang 6

1.2 Rumusan Masalah 8

1.3 Batasan Masalah 9

1.4 Tujuan Penelitian 9

1.5 Manfaat Penelitian 10

1.6 Sistematika Penulisan 10

BAB II KAJIAN PUSTAKA DAN DASAR TEORI 6

2.1 Kajian Pustaka 6

2.2 Dasar Teori 7

2.2.1 *E-Library* 7

2.2.2 Personalisasi Web 8

2.2.3 Data Mining 8

2.2.4 Text Mining 10

2.2.5 Stemming Bahasa Indonesia Algoritma Nazief dan Andriani 12

2.2.6 Kaidah Asosiasi (*Association Rule*) 13

2.2.7 Algoritma Eclat 14

2.2.8 Evaluasi 17

BAB III METODOLOGI DAN PERANCANGAN 24

3.1 Analisis Data 24

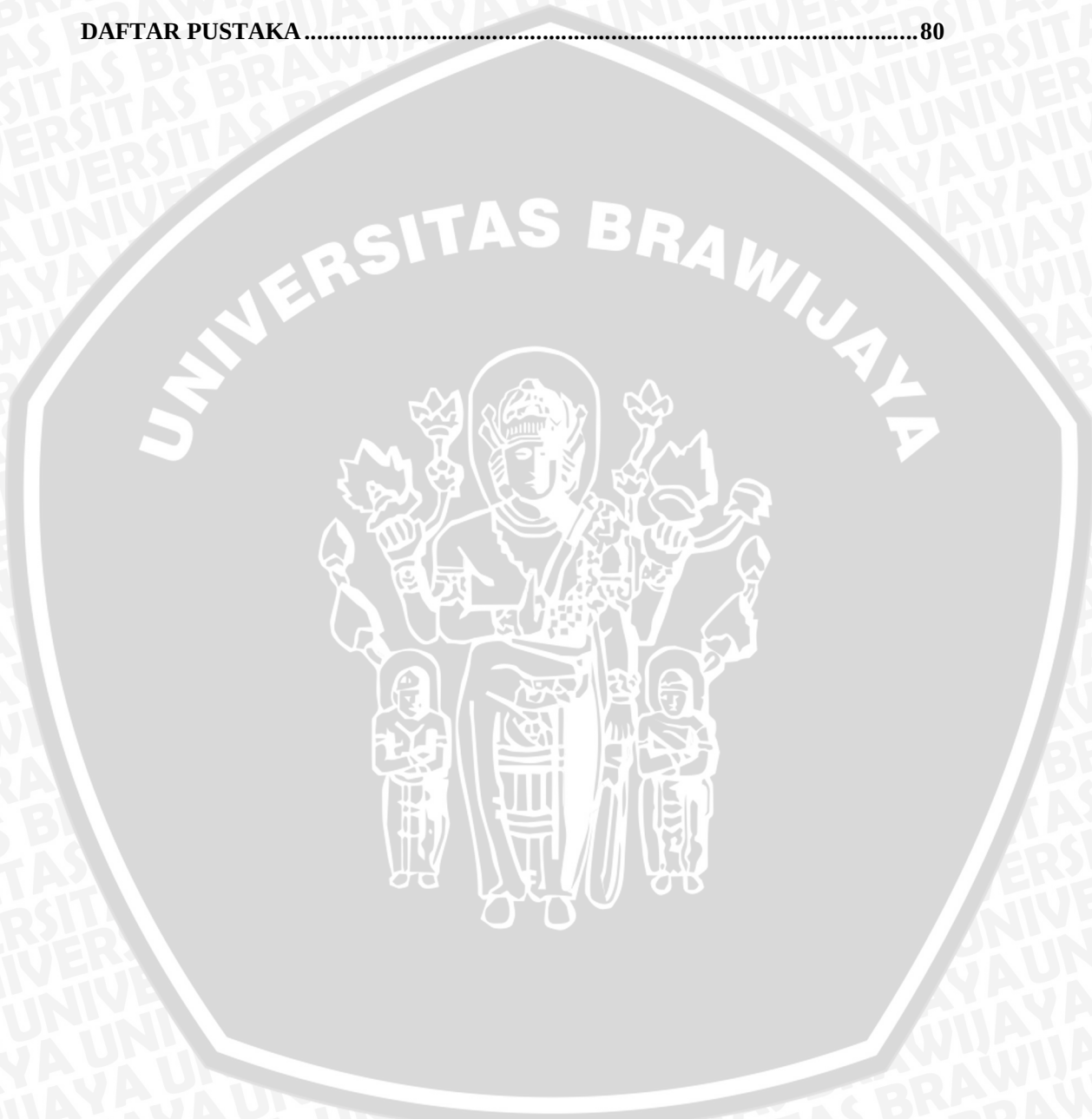
3.2 Perancangan Sistem 24

3.3 Perancangan Proses 25

3.3.1 Stemming dengan Algoritma Nazief Adriani 25

3.3.2	Pembentukan <i>rule-rule</i> kaidah asosiasi dengan Algoritma Eclat.....	27
3.3.3	Generate Rule dari Frequent Itemset.....	28
3.3.4	Hitung Confidence Rule.....	30
3.3.5	Hitung Lift Ratio.....	31
3.4	Perancangan BasisData.....	32
3.5	Contoh Perhitungan Manual	34
3.5.1	Pembentukan Frequent 1-itemset.....	38
3.5.2	Pembentukan Frequent k-Itemset.....	40
3.5.3	Perhitungan Confidence Rule	45
3.5.4	Perhitungan Lift Ratio.....	46
3.6	Perancangan Uji Coba	48
3.7	Perancangan interface.....	48
BAB IV IMPLEMENTASI DAN PEMBAHASAN.....		46
4.1	Lingkungan Implementasi	46
4.1.1	Lingkungan Perangkat Keras	46
4.1.2	Lingkungan Perangkat Lunak	46
4.2	Implementasi.....	47
4.2.1	Frequent Itemset	47
4.2.1.1	Fungsi Frequent 1-Itemset	47
4.2.1.2	Fungsi Frequent 2-Itemset	49
4.2.1.3	Fungsi Frequent n-Itemset	51
4.2.2	Pembentukan Rule	53
4.2.2.1	Proses Pembentukan Rule.....	54
4.2.2.2	Perhitungan Confidence.....	55
4.2.3	Perhitungan Lift Rasio	56
4.3	Implementasi AntarMuka	58
BAB V PENGUJIAN DAN ANALISIS.....		62
5.1	Skenario Pengujian.....	62
5.2	Hasil Pengujian	62
5.2.1	Hasil Uji Coba Perubahan Nilai Minimum Support	62
5.2.2	Hasil Uji Coba Perubahan Nilai Minimum Confidence.....	70
5.2.3	Hasil Uji Coba Perubahan Jumlah Transaksi	75
5.3	Analisis Pengujian.....	76
5.3.1	Analisis Hasil Uji Coba Perubahan Nilai Minimum Support.....	76
5.3.2	Analisis Hasil Uji Coba Perubahan Nilai Minimum Confidence	77
5.3.3	Analisis Hasil Uji Coba Perubahan Jumlah Transaksi.....	77

BAB VI KESIMPULAN DAN SARAN.....	78
6.1 Kesimpulan	78
6.2 Saran	79
DAFTAR PUSTAKA.....	80



DAFTAR GAMBAR

Gambar 2.1 Diagram Percobaan waktu eksekusi Algoritma Eclat dan Apriori.....	6
Gambar 2.2 Diagram Percobaan penggunaan memory Algoritma Eclat dan Apriori.....	6
Gambar 2.3 Mining Frequent Itemsets menggunakan Algoritma Paralel Eclat	111
Gambar 3.1 Alur deskripsi umum sistem personalisasi e-Library	20
Gambar 3.2 Flowchart Stemming Algoritma Nazief Adriani.....	21
Gambar 3.3 Alur pembentukan rule-rule asosiasi valid	23
Gambar 3.4 Alur Generate Rule dari Frequent Itemset.....	24
Gambar 3.5 Alur Hitung Confidence	25
Gambar 3.6 Alur Hitung Lift Ratio.....	26
Gambar 3.7 Rancangan Input data Porter Stemmer	44
Gambar 3.8 Rancangan Input data pembentukan Association Rule	44
Gambar 3.9 Rancangan Hasil Stemming.....	45
Gambar 3.10 Rancangan Hasil Frequent Itemset.....	45
Gambar 3.11 Rancangan Hasil Association Rule	45
Gambar 4.1 Menu Utama Pembentukan Rule	58
Gambar 4.2 Menu Porter Stemming	58
Gambar 4.3 Hasil Porter Stemming	59
Gambar 4.4 Menu Association Rule	60
Gambar 4.5 Contoh Hasil Freq Itemset.....	60
Gambar 4.6 Rule-rule yang dihasilkan.....	61
Gambar 4.7 Rule-rule valid yang dihasilkan	61
Gambar 5.1. Grafik Uji Coba Jumlah Transaksi 30 nilai Min Confidence 60%	62
Gambar 5.2. Grafik Uji Coba Jumlah Transaksi 50 nilai Min Confidence 60%	63
Gambar 5.3. Grafik Uji Coba Jumlah Transaksi 75 nilai Min Confidence 60%	64
Gambar 5.4. Grafik Uji Coba Jumlah Transaksi 100 nilai Min Confidence 60%	65
Gambar 5.5. Grafik Uji Coba Jumlah Transaksi 150 nilai Min Confidence 60%	66
Gambar 5.6. Grafik Uji Coba Jumlah Transaksi 200 nilai Min Confidence 60%	67
Gambar 5.7. Grafik Uji Coba Jumlah Transaksi 300 nilai Min Confidence 60%	68
Gambar 5.8. Grafik Uji Coba Jumlah Transaksi 400 nilai Min Confidence 60%	68
Gambar 5.9. Grafik Uji Coba Jumlah Transaksi 500 nilai Min Confidence 60%	69
Gambar 5.10. Grafik Uji Coba Jumlah Transaksi 30 nilai Min Support 60%.....	70
Gambar 5.11. Grafik Uji Coba Jumlah Transaksi 50 nilai Min Support 60%.....	70

Gambar 5.12. Grafik Uji Coba Jumlah Transaksi 75 nilai Min Support 60%.....	71
Gambar 5.13. Grafik Uji Coba Jumlah Transaksi 100 nilai Min Support 60%.....	72
Gambar 5.14. Grafik Uji Coba Jumlah Transaksi 150 nilai Min Support 60%.....	72
Gambar 5.15. Grafik Uji Coba Jumlah Transaksi 200 nilai Min Support 60%.....	73
Gambar 5.16. Grafik Uji Coba Jumlah Transaksi 300 nilai Min Support 60%.....	74
Gambar 5.17. Grafik Uji Coba Jumlah Transaksi 400 nilai Min Support 60%.....	74
Gambar 5.18. Grafik Uji Coba Jumlah Transaksi 500 nilai Min Support 60%.....	75
Gambar 5.19. Grafik Uji Coba Perubahan Jumlah Transaksi.....	76

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

Tabel 3.1 Tabel Buku.....	27
Tabel 3.2 Detail Buku	27
Tabel 3.3 Tabel trx_pinjam	28
Tabel 3.4 Tabel trx_arule	28
Tabel 3.5 Tabel trx_detil_rule	29
Tabel 3.6 Tabel freq_itemset	29
Tabel 3.7 Tabel trx_pinjam perhitungan manual	29
Tabel 3.8 Tabel Contoh Data Transaksi dalam vertikal	31
Tabel 3.9 Hasil Perhitungan Support setiap Item(Keyword).....	33
Tabel 3.10 Frequent 1-itemset	35
Tabel 3.11 Hasil Perhitungan Support 2-Itemset tiap Kelas.....	36
Table 3.12 Tabel Frequent 2-Itemset	38
Tabel 3.13 Tabel Hasil Perhitungan Support 3-Itemset tiap Kelas	38
Table 3.14 Frequent Itemset	39
Tabel 3.15 Rule-Rule yang dihasilkan dari frequent itemset.....	39
Tabel 3.16 Hasil Perhitungan Confidence Rule.....	40
Tabel 3.17 Rule-rule memenuhi min_confidence	41
Tabel 3.18 Perhitungan Lift Ratio Rule	42
Tabel 3.19 Uji Coba	43

DAFTAR SOURCE CODE

Source Code 4.1. Fungsi Frequent 1-Itemset.....	48
Source Code 4.2. Fungsi Frequent 2-Itemset.....	49
Source Code 4.3. Fungsi getCombinations.....	50
Source Code 4.4. Fungsi getTrx	51
Source Code 4.5. Fungsi Frequent n-Itemset.....	52
Source Code 4.6. Fungsi checkfreq.....	53
Source Code 4.7. Fungsi Pembentukan Rule.....	54
Source Code 4.8. Fungsi getFreqItemset.....	54
Source Code 4.9. Fungsi permutations.....	55
Source Code 4.10. Fungsi Perhitungan Confidence.....	56
Source Code 4.11. Fungsi Perhitungan Lift Ratio	57
Source Code 4.12. Fungsi getValidRule	57

