

no. 641/11/04

20.452/11/04



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

**PERANCANGAN DAN PEMBUATAN
PERANGKAT LUNAK DATA MINING
UNTUK PENCARIAN POLA ASOSIASI LOKAL
DALAM MARKET BASKET DATA
DENGAN MENGGUNAKAN ALGORITMA CLASD**

TUGAS AKHIR



RSIF
005-1
mai
p-1
2003

Oleh :

MOHAMMAD MAIMUN

NRP. 5197100003

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA**

2003

PERPUSTAKAAN ITS	
Tgl. Terima	15-8-2003
Terima Dari	H
No. Agenda Pyp.	218727

**PERANCANGAN DAN PEMBUATAN
PERANGKAT LUNAK DATA MINING
UNTUK PENCARIAN POLA ASOSIASI LOKAL
DALAM MARKET BASKET DATA
DENGAN MENGGUNAKAN ALGORITMA CLASD**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Sepuluh Nopember
Surabaya**

**Mengetahui / Menyetujui
Dosen Pembimbing**



**Dr. Ir. Arif Djunaidy M.Sc
NIP. 131 633 403**

SURABAYA

Juni, 2003

ABSTRAK

Sudah banyak algoritma data mining untuk pencarian pola asosiasi pada basis data yang sangat besar yang sudah diimplementasikan dalam beberapa Tugas Akhir sebelumnya. Tetapi, pada umumnya pencarian pola asosiasi pada algoritma-algoritma tersebut dilakukan secara global pada keseluruhan basis data sehingga tidak memungkinkan untuk menemukan pola-pola yang bersifat lokal. Di lain pihak, penggunaan algoritma pencarian pola asosiasi pada data yang sudah terlokalisasi memungkinkan untuk mendapatkan pola yang lebih spesifik yang akan lebih bermanfaat untuk pemasaran yang ditargetkan untuk tujuan-tujuan tertentu (*targetted marketing*).

Dalam Tugas Akhir ini dibuat sebuah perangkat lunak pengekstraksi pola asosiasi yang didasarkan pada algoritma CLASD (*CLustering for ASsociations Discovery*) yang merupakan salah satu teknik untuk pencarian pola asosiasi pada basis data yang sudah terklusterisasi. Secara garis besar algoritma CLASD terdiri dari tiga tahapan utama yaitu, pembentukan perwakilan klaster, pembentukan klaster-klaster berdasarkan perwakilan klaster yang terbentuk sebelumnya, dan pencarian 1-itemset dan 2-itemset utama untuk masing-masing klaster yang terbentuk. Selain itu, dalam tugas akhir ini ditambahkan algoritma untuk melakukan pencarian seluruh itemset utama dan pola asosiasi dari masing-masing klaster.

Perangkat lunak yang telah dibuat diuji coba untuk menggali pola asosiasi pada sejumlah data transaksi sintetis yang dibangkitkan dengan menggunakan perangkat lunak pembangkit data otomatis. Data transaksi yang digunakan adalah market basket data biasa bukan data yang bersifat lokal, karena belum adanya pembangkit data sintetis untuk data yang bersifat lokal. Hasil uji coba menunjukkan bahwa algoritma CLASD tidak dipengaruhi oleh proses pembacaan basis data, tetapi semakin besar jumlah masukan klaster, maka jumlah total itemset utama, pola asosiasi lokal dan waktu komputasi akan semakin besar pula. Waktu komputasi dipengaruhi juga oleh nilai persentase ambang batas rata-rata transaksi dan persentase ambang batas total transaksi, dimana waktu komputasi berbanding lurus dengan persentase ambang batas rata-rata transaksi, tetapi berbanding terbalik dengan persentase ambang batas total transaksi. Hasil uji coba perbandingan dengan perangkat lunak HYBRID menunjukkan bahwa perangkat lunak CLASD mampu melakukan proses komputasi yang lebih cepat tetapi membutuhkan memori yang lebih besar.

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah SWT atas segala nikmat, rahmat dan karunia-Nya. Semua tidak terlepas dari pertolongan dan kekuatan yang diberikan kepada penulis sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.

Laporan tugas akhir ini dengan judul **“Perancangan Dan Pembuatan Perangkat Lunak Data Mining Untuk Pencarian Pola Asosiasi Lokal Dalam Market Basket Data Dengan Menggunakan Algoritma CLASD”** disusun sebagai dokumentasi sekaligus merupakan penutup rangkaian pelaksanaan tugas akhir.

Tugas akhir ini disusun guna memenuhi sebagian persyaratan untuk memperoleh gelar sarjana pada jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan tugas akhir ini. Untuk itu kritik dan saran yang membangun akan diterima dengan terbuka.

Akhir kata, penulis berharap semoga karya tugas akhir ini dapat memberikan manfaat yang sebesar-besarnya bagi generasi sekarang dan yang akan datang.

Surabaya, Juni 2003

Penulis

UCAPAN TERIMA KASIH

Alhamdulillah, segala puji bagi Allah SWT yang Maha Pengasih lagi Maha Penyayang dan Shalawat serta salam bagi Rosulullah Muhammad SAW. Penulis dalam kesempatan ini juga mengucapkan terima kasih dan penghargaan yang sebesar-besarnya bagi semua pihak yang telah membantu dalam pelaksanaan Tugas Akhir, baik secara langsung maupun tidak langsung. Secara khusus penulis mengucapkan terima kasih kepada :

1. Kedua orang tua tercinta, Aba H.A. Fauzi S. dan Ummi Hj. Fauziah. Terima kasih yang tak terhingga atas kasih sayang, do'a, dukungan, didikan, dan segala bantuan yang telah diberikan. Hanya Allah SWT yang mampu membalasnya.
2. Saudara-saudara penulis, Alek Udi, Alek Aan, Rossi dan seluruh keluarga besar penulis, Kai H. Mannan (Alm), Mbuk Hj. Zainah, Anom Kardi, Mbak Zah, Mba H. Mursyid dan Mba Hj. Faizah atas dukungan dan do'anya.
3. Bapak Dr. Ir. Arif Djunaidy, M.Sc selaku dosen pembimbing dan dosen wali penulis. Terima kasih atas dorongan, bimbingan dan saran serta dukungan yang telah banyak diberikan.
4. Bapak Rully Soelaiman, S.Kom, M.Kom selaku dosen penguji sekaligus pembimbing, yang telah memberikan banyak bimbingan dalam perbaikan Tugas Akhir ini.
5. Segenap Dosen di Jurusan Teknik Informatika yang telah telah mendidik dan menularkan ilmunya selama masa kuliah.



6. Seluruh staf pengajaran dan tata usaha Jurusan Teknik Informatika, Pak Yudi, Pak Soleh, Mbak Erna, Mbak Eva, Mas Sugeng, Mas Hudan, Mas Hermono serta para staf lainnya.
7. Bapak Drs. EC. Ir. Riyanarto Sarno MSc. PhD, selaku Direktur Utama PT. Limaxindo Intersistem.
8. Mas Luthfi, Pak Endra, Mbak Asfiah, Mas Jalil, Mbak Sisin, Tony, Mas Seger, Pak Wibawa, Eko, Kakuz, Goank, Ketip, Banzer, Junaidi, Mas Arif, Mas Dwi, Kurniawan, Muhyi, Andre dan para LIMAXers lainnya. Terima kasih atas segala dukungan yang diberikan selama ini.
9. Teman-teman sesama pejuang TA, Erry, Kurniawan, Jan Waren, Deddy Cahyadi, Edi Wahyudi, Arif Kurniawan, Sukses selalu!
10. Dody Widi (trims bantuannya waktu seminar TA), Tony (thanks 4 Mining & Java tutorial), Aris, Kurnix, Zuber, Muhyi, Wahyudi, Andre Zola, Jan Waren, Agung, Agus, Luqman, Juned, Guntar, dan semua teman-teman C-0D yang tidak bisa penulis sebutkan satu-persatu. Terima kasih atas persahabatan dan bantuannya selama ini.
11. Teman-teman di Laguna Indah P3/40, Arif Ndut (CM teruuus), Anaz+Ummi (nempel teruuus), Krizna+Rita (selamat menempuh hidup baru), Dody+Sandy (proyek teruuus). Teman-teman penulis lainnya, para Kucing Garong Blok D, AMITS (Sidig, Pak Lurah Suyud, Mas Ronald, Mas Darlis, Mas Doni, Mas Zainal, dll), Fajari Litmanen, Tugiek, Candra, Harun, Memet, Aank, Horry. Terima kasih atas segala bantuan, dukungan dan canda tawa tiap harinya.

DAFTAR ISI

KATA PENGANTAR.....	iii
UCAPAN TERIMA KASIH.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	ix
DAFTAR TABEL.....	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	4
1.5 Metodologi Pelaksanaan.....	4
1.6 Sistematika Penulisan.....	6
BAB II PENGGUNAAN ALGORITMA CLASD	8
UNTUK PENGGALIAN POLA ASOSIASI.....	8
2.1 Konsep Data Mining.....	8
2.2 Konsep Klasterisasi Data.....	9
2.3 Algoritma CLASD.....	10
2.4 Terminologi Penting.....	11
2.5 Penggalian Pola Asosiasi.....	12
2.3.1 Pembentukan Perwakilan Klaster.....	13
2.3.1.1 Perhitungan <i>Afinitas</i> Pasangan Item.....	13
2.3.1.2 Inisialisasi Perwakilan Klaster.....	15
2.3.1.3 Penggabungan <i>Perwakilan Klaster</i> Terdekat.....	16
2.3.1.4 Pemasukan Transaksi ke Dalam <i>Klaster</i> Terdekat.....	18
2.3.1.5 Penghapusan <i>Perwakilan Klaster</i> yang Kurang Berbobot.....	19
2.3.1.6 Pengecekan Kondisi <i>Klaster</i>	19
2.3.2 Pembentukan <i>Klaster</i>	20
2.3.3 Pembangkitan <i>Frequent Itemset</i>	20
2.3.4 Pencarian Pola Asosiasi Lokal.....	24
BAB III PERANCANGAN DAN PEMBUATAN	26
PERANGKAT LUNAK	26
3.1 Perancangan Perangkat Lunak.....	26
3.1.1 Perancangan Data.....	26
3.1.1.1 Data Masukan.....	26
3.1.1.2 Data Proses.....	27
3.1.1.3 Data Luaran.....	29
3.1.2 Perancangan Proses.....	31
3.1.3 Perancangan Antar Muka.....	35
3.2 Pembuatan Perangkat Lunak.....	37
3.2.1 Pembuatan Data.....	37
3.2.1.1 Data Masukan.....	37
3.2.1.2 Data Proses.....	38
3.2.1.3 Data Luaran.....	39
3.2.2 Pembuatan Proses.....	40
3.2.3 Implementasi Antar Muka.....	43
BAB IV UJI COBA DAN EVALUASI	51
4.1 Lingkungan Uji Coba.....	51
4.2 Data Uji Coba.....	51
4.3 Pelaksanaan Uji Coba.....	52

4.3.1	Uji Coba Kebenaran.....	52
4.3.1.1	Uji Coba Kebenaran Pembentukan Klaster.....	52
4.3.1.2	Uji Coba Kebenaran Pencarian Frequent Itemset dan Pola Asosiasi.....	54
4.3.2	Uji Coba Kinerja.....	56
4.3.3	Uji Coba Perbandingan.....	62
4.3.4	Uji Coba Keandalan.....	64
BAB V	PENUTUP.....	65
5.1	Kesimpulan.....	65
5.2	Kemungkinan Pengembangan.....	66
DAFTAR PUSTAKA.....		67

DAFTAR GAMBAR

Gambar 2.1	Pembangkitan frequent itemset	24
Gambar 3.1	Format data hasil proses	29
Gambar 3.2	Format data klaster	30
Gambar 3.3	Format data <i>cluster support</i>	31
Gambar 3.4	Format data pola asosiasi lokal	31
Gambar 3.5	DAD level 0	32
Gambar 3.6	DAD level 1	33
Gambar 3.7	DAD level 2	34
Gambar 3.8	Struktur toolbar aplikasi	35
Gambar 3.9	Koneksi ke basis data dan pembacaan tabel transaksi	38
Gambar 3.10	Pembentukan data afinitas	38
Gambar 3.11	Pembentukan data transaksi	39
Gambar 3.12	Class Meta_CLASD	39
Gambar 3.13	Prosedur-prosedur proses pembentukan klaster	42
Gambar 3.14	Class Cbits_CLASD	42
Gambar 3.15	Form utama aplikasi CLASD	44
Gambar 3.16	Toolbar aplikasi CLASD	44
Gambar 3.17	Form konfigurasi - tabulasi Connection	45
Gambar 3.18	Form konfigurasi – tabulasi Tables	45
Gambar 3.19	Form konfigurasi – tabulasi Clusters	46
Gambar 3.20	Form utama yang menampilkan informasi proses mining	47
Gambar 3.21	Form pilihan klaster	47
Gambar 3.22	Form utama dengan informasi data transaksi per klaster	48
Gambar 3.23	Form utama dengan informasi data <i>cluster support</i> per klaster	49
Gambar 3.24	Form utama dengan informasi data <i>pola asosiasi lokal</i> per klaster	50
Gambar 3.25	Form About	50
Gambar 4.1	Perbandingan <i>jumlah masukan klaster</i> – waktu pembacaan basis data	57
Gambar 4.2	Perbandingan <i>jumlah masukan klaster</i> – jumlah total itemset utama	58
Gambar 4.3	Perbandingan <i>jumlah masukan klaster</i> – jumlah total pola asosiasi lokal	59
Gambar 4.4	Perbandingan <i>jumlah masukan klaster</i> – total waktu komputasi	60
Gambar 4.5	Perbandingan nilai <i>threshold</i> – waktu pencarian itemset utama	61
Gambar 4.6	Perbandingan nilai <i>threshold'</i> – waktu pencarian itemset utama	62
Gambar 4.7	Perbandingan waktu komputasi untuk data N1kD10k	63
Gambar 4.8	Perbandingan waktu komputasi untuk data N1kD50k	63

DAFTAR TABEL

Tabel 2.1	Terminologi Penting dalam algoritma CLASD	12
Tabel 2.2	Contoh data item dan kodenya.....	12
Tabel 2.3	Contoh data transaksi.....	13
Tabel 2.4	Aggregate support untuk 1-itemset	14
Tabel 2.5	Aggregate support untuk 2-itemset	14
Tabel 2.6	Nilai afinitas pasangan item.....	15
Tabel 2.7	Perwakilan kluster dengan bobot item.....	16
Tabel 2.8	Pasangan perwakilan kluster terdekat.....	17
Tabel 2.9	Hasil perwakilan kluster setelah proses penggabungan.....	18
Tabel 2.10	Hasil perwakilan kluster dan kluster setelah proses pemasukan transaksi .	18
Tabel 2.11	Hasil perwakilan kluster setelah proses penghapusan.....	19
Tabel 2.12	Hasil akhir kluster	20
Tabel 2.13	BitSet item-item untuk kluster 1	21
Tabel 2.14	Frequent 1-Itemset	21
Tabel 2.15	Contoh perhitungan support untuk 2-itemset.....	22
Tabel 2.16	Frequent 2-itemset	22
Tabel 2.17	Contoh perhitungan support untuk 3-itemset.....	23
Tabel 2.18	Frequent 3-itemset sampai k-itemset	23
Tabel 2.19	Hasil pola asosiasi lokal	25
Tabel 3.1	Format data afinitas.....	27
Tabel 3.2	Format data transaksi.....	28
Tabel 3.3	Format data metatransaksi.....	29
Tabel 3.4	Struktur tabel data transaksi.....	37
Tabel 4.1	Spesifikasi data uji coba kinerja	52
Tabel 4.2	Spesifikasi data uji coba perbandingan dan kehandalan.....	52
Tabel 4.3	Spesifikasi data uji coba kebenaran pembentukan kluster	53
Tabel 4.4	Hasil pembentukan kluster dengan algoritma CLASD.....	53
Tabel 4.5	Hasil perhitungan <i>similarity</i> antar transaksi	54
Tabel 4.6	Spesifikasi data uji coba pencarian frequent itemset dan pola asosiasi	55
Tabel 4.7	Hasil uji coba kebenaran algoritma HYBRID	56
Tabel 4.8	Hasil uji coba kebenaran algoritma CLASD.....	56
Tabel 4.9	Hasil uji coba perbandingan penggunaan memori untuk data N1kD10k.....	64



BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Market basket data merupakan kumpulan data transaksi pembelian yang dilakukan para pelanggan setiap harinya [APY-02], sehingga jumlahnya dari hari ke hari akan semakin besar yang biasanya akan mencapai ribuan sampai jutaan data transaksi. Masing-masing data transaksi itu sendiri terdiri dari data item-item barang atau produk yang dibeli pelanggan. Dari *market basket data* tersebut bisa dicari suatu pola pembelian yang dilakukan oleh pelanggan yang biasa disebut dengan pola asosiasi.

Banyak penelitian dan percobaan yang telah dilakukan oleh para peneliti dalam 10 tahun terakhir untuk menemukan pola asosiasi diantara item-item dalam sejumlah besar transaksi [AIS-93, ASR-94]. Semua metode dari penelitian tersebut berusaha untuk mencari pola asosiasi secara langsung dari data yang sangat besar. Teknik ini juga telah banyak dibahas dalam judul-judul tugas akhir sebelumnya [PRA-01, HHE-02, DTY-02, IPD-02, TAW-03].

Namun beberapa tahun belakangan ini ada teknik baru untuk mencari pola asosiasi yang mulai di perkenalkan oleh para peneliti [GRS-98, GRS-99]. Teknik ini melakukan pencarian pola asosiasi *lokal* yaitu pencarian pola asosiasi pada data yang telah dikelompokkan menjadi sejumlah data yang lebih kecil berdasarkan kesamaan yang dimilikinya. Teknik ini merupakan penggabungan dua teknik dalam data mining yaitu *klasterisasi data (clustering)* dan *pola asosiasi (association rule)*.

Tugas akhir ini difokuskan pada pembuatan segmen dari *market basket data* yang dapat memberi pandangan yang lebih baik dengan mendapatkan pola asosiasi dari suatu data dalam segmen yang kecil. Pemikiran ini didasarkan pada kenyataan dimana suatu asosiasi bisa didapatkan dari suatu segmen data dan tidak bisa didapatkan dari perhitungan secara global. Pembuatan segmen ini biasa disebut dengan teknik *klasterisasi data (clustering)*.

Teknik *klasterisasi data* secara umum bisa diklasifikasikan menjadi dua kategori yaitu *partisi (partitional)* dan *hirarki (hierarchical)* [HMK-00]. Pada teknik *partisi*, satu obyek data dipartisi menjadi *klaster-klaster (clusters)* sehingga obyek-obyek dalam sebuah klaster memiliki lebih banyak persamaan dibandingkan dengan obyek pada klaster lain. Sedangkan pada *hirarki agglomerasi (agglomerative hierarchical)*, teknik ini dimulai dengan penempatan setiap obyek pada klasternya masing-masing dan kemudian menggabungkan klaster-klaster kecil menjadi klaster yang lebih besar dengan cara dari bawah ke atas (bottom up) sampai suatu kriteria tertentu terpenuhi.

Dari penggabungan dua teknik klasterisasi data di atas, maka diperoleh suatu algoritma untuk pencarian pola asosiasi lokal yaitu algoritma **CLASD** (**CLustering for ASsociation Discovery**) [APY-02] yang akan dibahas lebih lanjut dalam tugas akhir ini.

1.2 Permasalahan

Dalam pembuatan tugas akhir ini akan dirumuskan beberapa permasalahan sebagai berikut:

- Perancangan struktur data yang baik untuk algoritma CLASD sehingga dapat mengoptimalkan kinerja perangkat lunak.

- Pembentukan *klaster-klaster* yang baik (klaster-klaster yang mempunyai transaksi–transaksi dengan kesamaan yang dekat dalam satu klaster dan mempunyai transaksi–transaksi dengan kesamaan yang jauh dari transaksi–transaksi di klaster lainnya).
- Perancangan dan pembuatan perangkat lunak yang tangguh.
- Pelaksanaan uji coba untuk perbandingan dengan algoritma-algoritma terbaik yang pernah dibuat pada tugas akhir-tugas akhir sebelumnya yang menggunakan pencarian pola asosiasi secara global.

1.3 Batasan Masalah

Adapun batasan-batasan dalam pembuatan tugas akhir ini antara lain:

- Data transaksi dalam basis data yang akan digunakan sebagai obyek mining merupakan data hasil pembangkitan perangkat lunak pembangkit data yang telah dibahas dalam tugas akhir Adhita Pratiwi [APR-02], karena belum ada pembangkit data untuk data lokal.
- Penyimpanan data transaksi hasil dari pembacaan basis data dilakukan secara vertikal.
- Dalam pembuatan perwakilan klaster digunakan pengambilan data transaksi dari basis data secara random.
- Uji coba perbandingan yang dilakukan dengan algoritma dari tugas akhir sebelumnya dilakukan dengan cara membandingkan waktu komputasi dan penggunaan memori untuk perhitungan pola asosiasi secara global (klaster=1).

1.4 Tujuan dan Manfaat

Tujuan dari pembuatan tugas akhir ini antara lain :

- Merancang dan membuat perangkat lunak data mining yang bisa mencari pola asosiasi dalam *market basket data* dengan menerapkan algoritma CLASD.
- Menguji coba dan mengaplikasikan perangkat lunak tersebut pada suatu basis data yang sangat besar dan membandingkan dengan algoritma-algoritma yang lain.

Sedangkan manfaat yang diharapkan dari tugas akhir ini antara lain:

- Mendapatkan informasi yang berupa pola asosiasi dari suatu *market basket data*.

1.5 Metodologi Pelaksanaan

Dalam penyusunan tugas akhir ini, dibagi dalam beberapa tahap yang masing-masing akan dijelaskan sebagai berikut:

- *Studi literatur*

Dalam tahap ini dilakukan studi dan pemahaman literatur yang berupa referensi jurnal-jurnal ataupun buku-buku panduan tentang data mining secara umum dan yang berkaitan dengan algoritma CLASD. Selain itu, studi dan pemahaman literatur juga dilakukan terhadap tugas akhir-tugas akhir yang telah ada dan yang berkaitan dengan data mining sebagai bahan perbandingan yang berguna dalam tahap penyusunan tugas akhir ini.

- *Perancangan perangkat lunak*

Dalam tahap ini dilakukan beberapa tahap perancangan perangkat lunak yaitu:

- *Perancangan data masukan* : data masukan yang akan digunakan merupakan hasil perangkat lunak pembangkit data yang telah dibahas dalam tugas akhir Adhita Pratiwi [APR-02], karena belum ada perangkat lunak pembangkit data untuk data lokal.
- *Perancangan proses* : perancangan proses perangkat lunak menggunakan Data Flow Diagram (DFD) level 0 sampai 2.
- *Perancangan antar muka* : perancangan antar muka menggunakan bentuk Single Design Interface (SDI).

- *Pembuatan perangkat lunak*

Dalam tahap ini dilakukan beberapa tahap pembuatan perangkat lunak, yaitu:

- *Data* : basis data yang digunakan untuk perangkat lunak ini adalah Oracle 8i versi 8.1.7.
- *Proses / algoritma* : algoritma dari perangkat lunak ini diterapkan dengan menggunakan bahasa pemrograman Oracle JDeveloper 3.2.
- *Antar muka* : desain antar muka juga menggunakan bahasa perograman yang sama dengan tahapan proses.

- *Uji coba dan evaluasi*

Dalam tahap ini perangkat lunak yang telah dibuat diuji dengan kondisi sebagai berikut :

- *Lingkungan uji coba* : perangkat lunak ini diuji coba pada suatu perangkat komputer dengan spesifikasi yang sama dengan tugas akhir-tugas akhir sebelumnya [DTY-02, TAW-03].
- *Data* : data yang digunakan untuk uji coba merupakan data hasil data generator yang telah dibahas dalam tugas akhir sebelumnya [APR-02].

Percobaan dilakukan beberapa kali dengan data yang berbeda spesifikasinya, baik jumlah data ataupun jumlah *item*.

- *Pelaksanaan dan evaluasi* : pelaksanaan dan evaluasi uji coba dilakukan dengan beberapa kriteria percobaan antara lain : percobaan kebenaran, percobaan hasil luaran, percobaan kehandalan, percobaan pembacaan basis data dan percobaan perbandingan.
- *Penyusunan buku Tugas Akhir*

Dalam tahap terakhir ini disusun sebuah buku sebagai laporan dari dokumentasi pelaksanaan tugas akhir.

1.6 Sistematika Penulisan

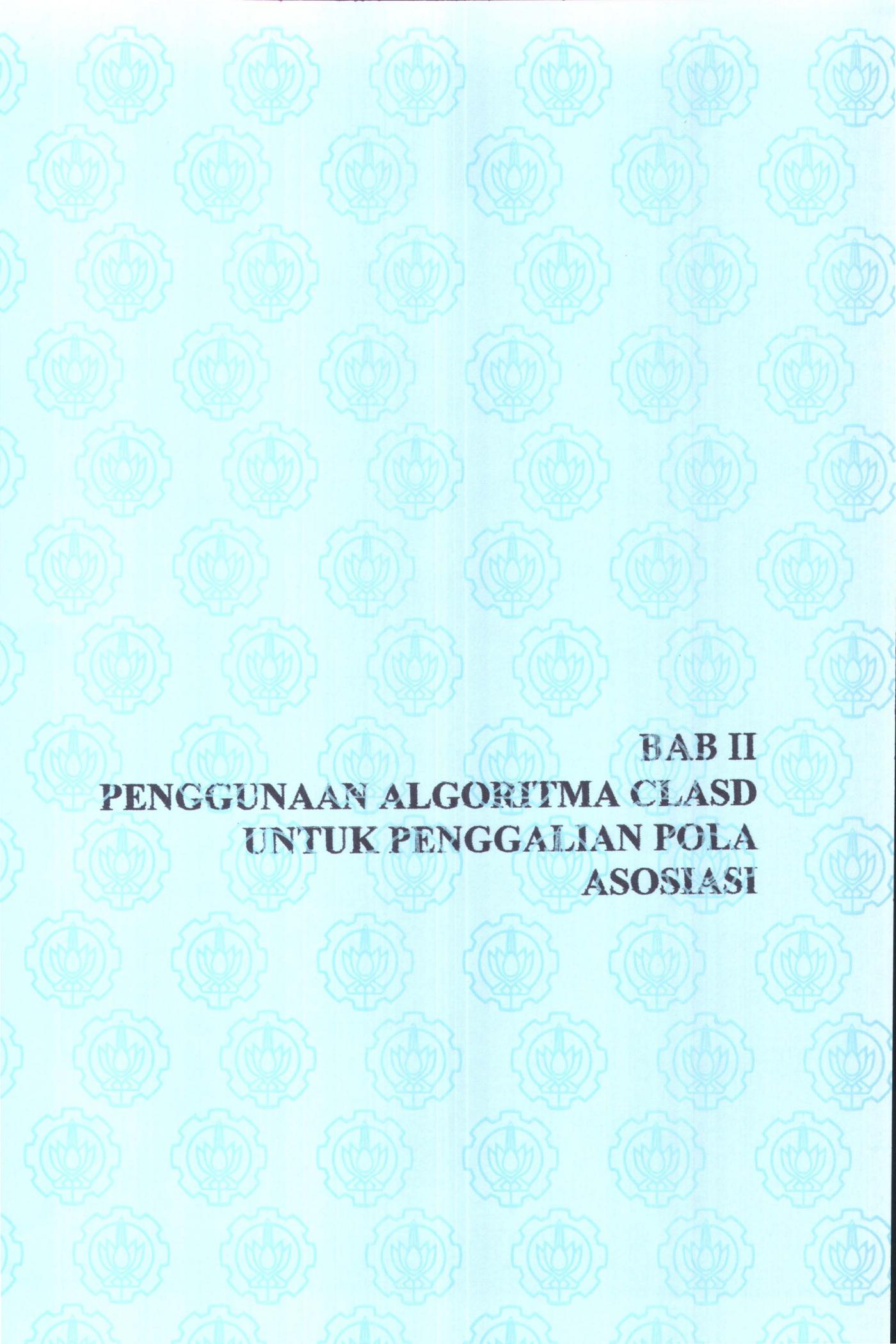
Dalam penyusunannya, laporan tugas akhir ini dibagi dalam enam bab yang membahas bagian-bagian sebagai berikut. Bab I pendahuluan, dalam bab ini dijelaskan beberapa hal pokok tentang tugas akhir yang dilakukan yaitu latar belakang, permasalahan dan batasannya, tujuan dan manfaat, metodologi serta sistematika penulisan buku tugas akhir.

Bab II penggunaan algoritma CLASD untuk penggalian pola asosiasi, bab ini berisi tahap-tahap penggunaan algoritma CLASD dalam proses pembentukan kluster dan penggalian pola asosiasi.

Bab III perancangan dan pembuatan perangkat lunak menguraikan perancangan data, perancangan proses dengan menggunakan Diagram Aliran Data (DAD) dan perancangan antar muka dari perangkat lunak yang akan dibuat. Serta berisi pembahasan mengenai implementasi struktur data, pseudocode proses dan tampilan antar muka dari perangkat lunak yang telah dibuat.

Bab IV uji coba perangkat lunak, bagian ini menjelaskan proses uji coba yang dilakukan terhadap perangkat lunak yang dibuat pada beberapa kriteria pengujian yang berbeda dengan menggunakan data yang telah dipersiapkan.

Bab V penutup, bab ini merupakan bagian terakhir dari buku tugas akhir ini yang berisi kesimpulan berdasarkan uji coba perangkat lunak dan kemungkinan pengembangan lebih lanjut yang dapat dilakukan.



BAB II
PENGGUNAAN ALGORITMA CLASD
UNTUK PENGGALIAN POLA
ASOSIASI

BAB II

PENGGUNAAN ALGORITMA CLASD

UNTUK PENGGALIAN POLA ASOSIASI

Banyaknya transaksi-transaksi yang terjadi tiap harinya membuat suatu data dari hari ke hari bertambah besar. Untuk itu diperlukan suatu teknik tertentu agar bisa mengolah basis data yang sangat besar tersebut menjadi suatu informasi yang berguna, teknik ini biasa disebut dengan *data mining*. Dalam bab ini dibahas mengenai konsep dasar dari *data mining*, klusterisasi data dan penggunaan algoritma CLASD dalam penggalian pola asosiasi.

2.1 Konsep Data Mining

Data mining adalah proses eksplorasi dan analisa data dalam basis data yang besar dari berbagai perspektif dan perhitungan menjadi suatu informasi atau pola yang bisa berguna bagi pengguna [HMK-00]. Dengan menggunakan teknik data mining maka pola dan aturan yang tersirat dalam suatu basis data yang sangat besar baik yang berupa pola asosiasi, pola sequensial dan pola lainnya dapat diketahui.

Data mining merupakan salah satu alat analisis untuk menganalisa data, yang memungkinkan pengguna untuk menganalisa data dari berbagai dimensi atau sudut, mengelompokkan data, meringkas identifikasi relasi data. Secara teknik, data mining adalah proses untuk pencarian hubungan dan pola diantara sejumlah field dalam suatu database relasional yang besar. Mengingat objek basis data yang sangat besar maka dalam teknik data mining sangat

memperhatikan efektifitas dan efisiensi algoritma yang digunakan dalam penggalian hubungan dan pola tersebut.

2.2 Konsep Klasterisasi Data

Klasterisasi data (Clustering) merupakan metode dalam data mining yang mengelompokkan objek data menjadi klaster-klaster (clusters) [HMK-00]. Sedangkan *klaster* itu sendiri merupakan kumpulan objek data yang memiliki kesamaan dengan objek data lainnya dalam satu klaster yang sama dan memiliki perbedaan dengan objek data pada lain klaster.

Salah satu tujuan penting dalam metode klasterisasi data adalah untuk mendapatkan klaster-klaster yang berkualitas tinggi. Adapun ciri-ciri dari klaster yang berkualitas tinggi tersebut antara lain :

- Memiliki kesamaan yang tinggi dalam satu klaster (*intra klaster*).
- Memiliki kesamaan yang rendah antara satu klaster dengan klaster yang lain (*antar klaster*).

Kualitas dari hasil *klasterisasi data* bergantung pada metode yang digunakan untuk kedua pengukuran kesamaan di atas dan implementasinya. Kualitas dari metode *klasterisasi data* diukur juga dari kemampuannya untuk mencari beberapa atau keseluruhan dari pola yang tersembunyi.

Salah satu contoh penggunaan klasterisasi data adalah dari data transaksi para pelanggan, mereka bisa dikelompokkan berdasarkan item barang yang mereka beli dari tiap transaksi. Misalnya salah satu klaster (kelompok) didominasi oleh pelanggan yang sudah menikah dengan membeli susu, popok, mainan, gula, makanan bayi dan lain-lain. Sedangkan klaster yang lain

didominasi oleh pelanggan berpenghasilan tinggi yang biasa membeli barang-barang import seperti keju, anggur perancis, coklat dan lain-lain.

Klaster-klaster tersebut bisa digunakan untuk membuat ciri-ciri pada kelompok pelanggan yang berbeda-beda. Sedangkan ciri-ciri ini bisa digunakan untuk membuat target pemasaran dan untuk promosi suatu produk tertentu kepada kelompok pelanggan tertentu pula. Ciri-ciri ini juga bisa digunakan untuk memprediksi pola pembelian pelanggan baru berdasarkan profil mereka.

2.3 Algoritma CLASD

Pendekatan algoritma **CLASD** (*CLustering for ASsociation Discovery*) menggunakan *perwakilan klaster-klaster* (*cluster representatives*) dengan tujuan untuk membentuk partisi-partisi dari basis data [APY-02]. Pencarian *perwakilan klaster* yang bagus merupakan salah satu titik sukses dari algoritma ini. Algoritma ini memerlukan inputan parameter k yang mengindikasikan jumlah klaster yang diinginkan oleh pengguna. Jumlah *perwakilan klaster* awal berjumlah *startsize* yang jumlahnya selalu lebih besar dari inputan k .

Harapan awalnya banyak dari *perwakilan klaster* nantinya masuk ke *klaster* yang sama atau tidak masuk ke *klaster* manapun. Untuk alasan tersebut maka jumlah awal *perwakilan klaster* lebih besar dari jumlah k *klaster* yang diinginkan. Secara iterasi, akan dihapus *perwakilan klaster* yang tidak masuk ke *klaster* manapun dan akan digabung *wakil-wakil klaster* ke *klaster* yang memiliki kesamaan yang paling dekat. Dalam tiap iterasi jumlah *perwakilan klaster* akan dikurangi dengan faktor α . Untuk itu sepasang *perwakilan klaster* yang paling dekat akan digabung dalam setiap iterasinya.

Penggunaan *teknik partitioning* adalah untuk mengikutsertakan informasi dari seluruh basis data dalam setiap iterasi dengan tujuan untuk memastikan bahwa hasil akhir tidak hanya terwakilkan dari sejumlah *startsize* perwakilan klaster. Akan tetapi, penggunaan metode *partitioning* secara full akan sangat berat dan akan membutuhkan waktu yang lama dalam setiap iterasinya. Karena itu maka diambil pilihan dengan penerapan pengambilan data transaksi secara random untuk dijadikan *perwakilan klaster*.

Adapun keuntungan penggunaan algoritma *CLASD* antara lain :

- Pencarian pola asosiasi dari data dalam segmen kecil akan mendapatkan pola asosiasi yang lebih detil daripada algoritma pencarian pola asosiasi secara menyeluruh.
- Tidak seperti kebanyakan aplikasi klasterisasi data lainnya yang perhitungan luarannya harus sesuai dengan inputan k yaitu jumlah klaster yang diinputkan oleh pengguna, pada *CLASD* jumlah klaster tidak harus sama dengan inputan k , asalkan kriteria lainnya terpenuhi, yaitu sejumlah k klaster terbesar memiliki jumlah transaksi yang lebih banyak dari *threshold'* persen dari total transaksi.
- Mampu memisahkan klaster-klaster yang bersifat *noise* dan *outlier* (tidak berguna dan hanya akan memperlambat proses).

2.4 Terminologi Penting

Dalam algoritma *CLASD* terdapat beberapa terminologi penting yang perlu diketahui sehingga dapat memudahkan pembaca dalam memahami algoritma tersebut. Terminologi penting tersebut terangkum dalam tabel 2.1 berikut ini :

Tabel 2.1 Terminologi Penting dalam algoritma CLASD

Istilah	Notasi	Keterangan
Basis Data (Database)	D	Basis data yang menyimpan data transaksi pembelian
Transaction ID	TID	ID Transaksi
Item ID	IID	ID Item
Transaksi	T	Kumpulan dari item.
Metatransaksi	m	Kumpulan item yang terhubung dengan bobot nilai pada masing-masing item.
Afinitas (Affinity)	$A(i,j)$	Perbandingan dari persentase transaksi yang mengandung kedua item (i, j) dan transaksi yang hanya mengandung salah satu dari kedua item (i, j)
Bobot Item	$Wt_m(i)$	Bobot item i dalam metatransaksi M
Itemset	I	Himpunan item dalam database
Aggregate Support	$Sup(X)$	Jumlah transaksi dalam seluruh database dimana item X berada sebagai subset.
Cluster Support	$Sup_C(X)$	Jumlah transaksi dalam suatu kluster dimana item X berada sebagai subset.
Jumlah Kluster	K	Nilai yang ditentukan oleh pengguna yang menunjuk pada jumlah kluster yang diinginkan
Perwakilan Kluster	M	Perwakilan kluster yang diambil dari transaksi-transaksi dalam database dan merupakan perwakilan awal dari kluster yang akan terbentuk
Kluster	C	Kluster yang merupakan kumpulan dari transaksi yang memiliki kesamaan.
Threshold	$Threshold$	Jumlah persentase dari rata-rata transaksi yang sudah masuk ke kluster.
Threshold'	$Threshold'$	Jumlah persentase dari total transaksi yang sudah masuk ke kluster.

2.5 Penggalan Pola Asosiasi

Untuk mempermudah penjelasan mengenai jalannya algoritma CLASD, maka akan disertakan contoh satu data transaksi seperti yang terlihat pada tabel 2.2 dan 2.3 di bawah ini :

Tabel 2.2 Contoh data item dan kodenya

Nama Item	Kode
Coklat	C
Gula	G
Kopi	K
Susu	S
The	T

Tabel 2.3 Contoh data transaksi

TID	ITEM
1	C K S T
2	C G S
3	C K S T
4	C G S T
5	C G K S T
6	C G K

Secara garis besar algoritma CLASD terbagi dalam 4 (empat) tahap, yaitu pembentukan *perwakilan klaster*, pembentukan *klaster*, pembangkitan *frequent itemset* dan pencarian pola asosiasi. Penjelasan algoritma CLASD akan dilakukan dengan menggunakan contoh data transaksi di atas dengan masukan awal jumlah klaster (k) = 2, *cluster minimum support* = 50%, *cluster confidence support* = 100%, *threshold* = 20% dan *threshold'* = 33%.

2.3.1 Pembentukan Perwakilan Klaster

Tahap pertama dari algoritma CLASD adalah pembentukan perwakilan klaster yang akan digunakan untuk memasukkan data transaksi ke dalam perwakilan klaster-perwakilan klaster sehingga menjadi klaster-klaster. Dalam pembentukan perwakilan klaster ada beberapa tahapan yang akan dijelaskan lebih lanjut dalam sub bab-sub bab berikut.

2.3.1.1 Perhitungan *Afinitas* Pasangan Item

Langkah pertama yang harus dilakukan dalam proses algoritma CLASD adalah perhitungan *Afinitas* yaitu nilai kedekatan dari sepasang item. *Afinitas* antara dua item i dan j , yang dilambangkan dengan $A(i,j)$, adalah perbandingan antara persentase transaksi yang mengandung dua item i dan j dengan persentase transaksi yang mengandung salah satu dari item i atau item j , dengan rumus sebagai berikut [APY-02] :

$$A(i, j) = \frac{sup(\{i, j\})}{sup(\{i\}) + sup(\{j\}) - sup(\{i, j\})} \dots \dots \dots (2.1)$$

Untuk melakukan perhitungan nilai *afinitas* di atas maka di perlukan perhitungan *aggregate support* untuk 1-itemset dan 2-itemset terlebih dahulu. *Aggregate support* untuk 1-itemset hasilnya seperti terlihat pada tabel 2.4 di bawah ini :

Tabel 2.4 *Aggregate support* untuk 1-itemset

Itemset	Aggregate Support
C	6
G	4
K	4
S	5
T	4

Perhitungan *aggregate support* untuk item C atau $Sup(C)$ adalah 6 karena jumlah kemunculan item C dalam data transaksi terdapat pada enam transaksi yaitu pada TID = 1, 2, 3, 4, 5 dan 6. Demikian juga proses perhitungan untuk item-item yang lain. Sedangkan *aggregate support* untuk 2-itemset dari contoh data transaksi di atas hasilnya terlihat pada tabel 2.5 di bawah ini :

Tabel 2.5 *Aggregate support* untuk 2-itemset

Itemset	Aggregate Support	Itemset	Aggregate Support
{C,G}	4	{G,S}	3
{C,K}	4	{G,T}	2
{C,S}	5	{K,S}	3
{C,T}	4	{K,T}	3
{G,K}	2	{S,T}	4

Cara perhitungan *aggregate support* untuk 2-itemset sama dengan 1-itemset yaitu dengan cara menghitung jumlah kemunculan suatu itemset dalam data transaksi. Contohnya *aggregate support* untuk itemset {C,G} atau $Sup(\{C,G\})$ adalah 4 karena munculnya item C dan item G secara bersamaan dalam data transaksi di atas ada pada empat transaksi yaitu pada TID = 2, 4, 5

dan 6. Dengan menggunakan rumus 2.1 di atas, maka perhitungan nilai *Afinitas* antara pasangan item hasilnya akan terlihat seperti pada tabel 2.6 di bawah ini :

Tabel 2.6 Nilai afinitas pasangan item

Itemset	Afinitas	Itemset	Afinitas
{C, G}	4/6	{G, S}	1/2
{C, K}	4/6	{G, T}	1/3
{C, S}	5/6	{K, S}	1/2
{C, T}	4/6	{K, T}	3/5
{G, K}	1/3	{S, T}	4/5

Setelah didapat nilai nilai afinitas atau nilai kedekatan/kesamaan antara dua item maka dapat dihitung kesamaan antara sepasang *Transaksi T1* = { i_1, \dots, i_s } dan transaksi *T2* = { j_1, \dots, j_t } yang ditentukan sebagai rata-rata *Afinitas* dari item-item mereka. Dengan rumus 2.2 berikut [APY-02]:

$$Sim(T1, T2) = \frac{\sum_{p=1}^s \sum_{q=1}^t A(i_p, j_q)}{s.t} \dots\dots\dots(2.2)$$

Sedangkan kesamaan antara sepasang *metatransaksi* ditentukan dengan cara yang sama, dengan penambahan perhitungan bobot item-item yang terhubung dengan masing-masing item pada *metatransaksi*. Jika $m1 = \{i_1, \dots, i_s\}$ dan $m2 = \{j_1, \dots, j_t\}$ adalah dua *metatransaksi*, maka kesamaan antara $m1$ dan $m2$ ditentukan dengan rumus 2.3 berikut [APY-02] :

$$Sim(m1, m2) = \frac{\sum_{p=1}^s \sum_{q=1}^t (wt_{m1}(i_p) \cdot wt_{m2}(j_q) \cdot A(i_p, j_q))}{\sum_{p=1}^s \sum_{q=1}^t (wt_{m1}(i_p) \cdot wt_{m2}(j_q))} \dots\dots\dots(2.3)$$

2.3.1.2 Inisialisasi Perwakilan Kluster

Jumlah dari *perwakilan kluster (cluster representative)* diinisialisasikan dengan *currentsize* dan nilai ini selalu lebih besar dari data masukan jumlah

klaster k dalam hal ini ditentukan bahwa nilai *currentsize* sama dengan 30 kali k . Perwakilan klaster ini diambil secara random dari basis data sejumlah *currentsize*. Dengan harapan bahwa *perwakilan klaster* akan masuk ke dalam klaster yang sama atau tidak masuk ke klaster manapun. Karena alasan ini, pembuatan *perwakilan klaster* dengan jumlah yang lebih besar dari target dan secara iterasi membuang yang tidak mempunyai kesamaan dengan klaster manapun dan menggabung *perwakilan klaster* dengan klaster yang memiliki kesamaan paling dekat.

Perwakilan klaster yang diambil dari basis data akan diubah bentuknya dari data transaksi menjadi data *metatransaksi*, yaitu dengan cara memberi bobot 1 pada item yang terwakili dan bobot 0 pada item yang tidak ada pada transaksi yang bersangkutan. Untuk implementasi pada contoh data transaksi di atas maka *perwakilan klaster* awal yang diambil secara acak dari data transaksi tersebut sebanyak 4 (empat) transaksi yang diubah menjadi *metatransaksi* dan hasilnya terlihat dalam tabel 2.7 berikut ini :

Tabel 2.7 Perwakilan klaster dengan bobot item

m	C	G	K	S	T
1	1	0	1	1	1
3	1	0	1	1	1
5	1	1	1	1	1
6	1	1	1	0	0

2.3.1.3 Penggabungan *Perwakilan Klaster* Terdekat

Pada proses ini, langkah pertama yang dilakukan adalah melakukan perhitungan *ComputeClosestPair* yang akan menghasilkan daftar dari beberapa pasang *perwakilan klaster* yang memiliki kesamaan terdekat. Adapun rumus yang dipakai untuk melakukan perhitungan *ComputeClosestPair* adalah sebagai berikut [GRS-99]:

$$\text{similarity}(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} \dots\dots\dots(2.4)$$

Dimana $|T_1 \cap T_2|$ merupakan jumlah item yang sama dalam transaksi T_1 dan T_2 . Sedangkan $|T_1 \cup T_2|$ merupakan jumlah total item-item yang berbeda dalam dua transaksi T_1 dan T_2 . Adapun daftar pasangan *perwakilan klaster* dari tabel 2.7 yang memiliki kesamaan terdekat adalah seperti terlihat pada tabel 2.8 berikut :

Tabel 2.8 Pasangan perwakilan klaster terdekat

Pasangan M	Nilai Kedekatan
(1,3)	1
(5,6)	3/5

Pasangan yang paling dekat dihapus dan akan digantikan dengan hasil penggabungan dua perwakilan klaster tersebut dengan cara menjumlahkan bobot dari masing–masing item. Setelah proses penggabungan perwakilan klaster, proses selanjutnya adalah penghapusan item-item yang mempunyai bobot kurang. proses ini dimulai dengan pengurutan item-item dalam *perwakilan klaster* didasarkan pada bobot nilai masing-masing item secara descending atau dari yang paling berat ke yang paling ringan. Setelah itu item-item yang bobotnya kurang dari bobot item terberat dibagi nilai β dihilangkan dari *perwakilan klaster* dimana nilai β yang digunakan sama dengan 10. Setelah itu, dicek lagi jika item yang mempunyai bobot lebih besar dari 0 jumlahnya lebih besar dari d maka diambil d item yang tertinggi, dimana d merupakan nilai masukan dari proses sebelumnya. Setelah proses penggabungan di atas maka akan di peroleh data *perwakilan klaster* terbaru, seperti terlihat dalam tabel 2.9 berikut :

Tabel 2.9 Hasil perwakilan klaster setelah proses penggabungan

m	C	G	K	S	T
1	2	0	2	2	2
2	2	2	2	1	1

2.3.1.4 Pemasukan Transaksi ke Dalam *Klaster* Terdekat

Proses ini bertujuan untuk memasukkan transaksi-transaksi yang diambil secara random dari basis data ke dalam *klaster* yang mempunyai *perwakilan klaster* dengan kesamaan paling dekat. Langkah pertama yang dilakukan adalah membentuk sejumlah *klaster* sesuai dengan *perwakilan klaster* yang ada. Langkah selanjutnya adalah transaksi-transaksi dari basis data dibandingkan dengan *perwakilan klaster-perwakilan klaster* yang ada untuk mendapatkan *perwakilan klaster* dengan kesamaan paling dekat. Setelah diperoleh *perwakilan klaster* yang terdekat maka transaksi tersebut dimasukkan ke *klaster* dengan indek yang sama dengan *perwakilan klaster*nya.

Setelah transaksi-transaksi yang ada sudah dimasukkan ke *klaster-klaster* yang terdekat, maka *perwakilan klaster-perwakilan klaster* yang ada diinisialisasi lagi dengan cara mengubah transaksi-transaksi pada *klaster-klaster* tersebut menjadi bentuk *metatransaksi* dan menjumlahkan bobot item-itemnya. Jika proses ini diterapkan terhadap contoh data transaksi dengan masukan 2 *perwakilan klaster* dan 2 transaksi yaitu TID 2 dan 4 maka akan menghasilkan 2 *perwakilan klaster* dan 1 *klaster*. Hasil tersebut bisa dilihat dalam tabel 2.10 berikut :

Tabel 2.10 Hasil perwakilan klaster dan klaster setelah proses pemasukan transaksi

M	C	G	K	S	T	C	TID
1	4	2	2	4	3	1	2 4
2	2	2	2	1	1	2	-

2.3.1.5 Penghapusan *Perwakilan Klaster* yang Kurang Berbobot

Proses ini bertujuan untuk membuang *perwakilan klaster* yang sedikit anggotanya yang nantinya hanya akan memperlambat proses. Langkah pertama yang dikerjakan adalah mencari *klaster* yang memiliki jumlah transaksi kurang dari nilai *threshold* (*ambang batas rata-rata transaksi*), dan membuang *perwakilan klaster* yang memiliki indek yang sama dengan *klaster* tersebut. Nilai *threshold* disini merupakan masukan pengguna yang nilai defaultnya 20% dari rata-rata jumlah transaksi yang telah dimasukkan ke dalam *klaster-klaster*. Hasil proses ini terhadap contoh data transaksi menghasilkan *perwakilan klaster* baru. Hasil tersebut bisa dilihat dalam tabel 2.11 berikut :

Tabel 2.11 Hasil perwakilan klaster setelah proses penghapusan

m	C	K	K	S	T
1	4	2	2	4	3

2.3.1.6 Pengecekan Kondisi *Klaster*

Proses ini bertujuan untuk mengecek salah satu kondisi *perwakilan klaster* dan *klaster* yang sudah terbentuk apakah sudah memenuhi kriteria atau tidak. Ada dua kriteria yang akan menghentikan proses pembentukan *perwakilan klaster*, yang pertama jika jumlah *perwakilan klaster* kurang atau sama dengan jumlah masukan *k* (*jumlah klaster*) dari pengguna. Kriteria kedua adalah jika sejumlah *k* *klaster* terbesar memiliki jumlah transaksi yang lebih besar dari *threshold'* (*ambang batas total transaksi*). Nilai *threshold'* ini adalah persentase dari total jumlah transaksi yang sudah ada di seluruh *klaster* dan merupakan masukan pengguna dengan nilai default sama dengan 33%. Untuk contoh data transaksi diatas, karena jumlah *perwakilan klaster* kurang dari data masukan *k* maka proses pembentukan *perwakilan klaster* dihentikan.

2.3.2 Pembentukan *Klaster*

Proses pembentukan *klaster* merupakan proses yang sama dengan proses pemasukan transaksi ke dalam *klaster* terdekat yang telah dijelaskan dalam bab 2.5.1.4 diatas. Perbedaannya hanya pada jumlah transaksi yang akan dimasukkan, proses sebelumnya hanya memasukkan sejumlah transaksi sedangkan pada proses ini transaksi yang akan dimasukkan adalah seluruh transaksi dalam basis data. Langkah pertama dari proses ini membandingkan transaksi-transaksi tersebut dengan *perwakilan klaster-perwakilan klaster* untuk mendapatkan *perwakilan klaster* dengan kesamaan paling dekat, kemudian transaksi-transaksi tersebut dimasukkan ke dalam *klaster-klaster* dengan indek yang sama.

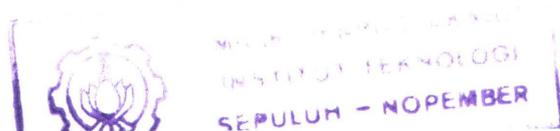
Untuk contoh transaksi di atas, jumlah *klaster* yang terbentuk adalah satu karena *perwakilan klaster* yang terbentuk hanya satu. Jadi *klaster* yang terbentuk pada proses akhir ini sama dengan data transaksi sebelum proses iterasi. Hasil *klaster* tersebut bisa dilihat pada tabel 2.12 berikut :

Tabel 2.12 Hasil akhir klaster

C	TID
1	1 2 3 4 5 6

2.3.3 Pembangkitan *Frequent Itemset*

Dari *klaster-klaster* yang terbentuk akan dilakukan pembangkitan *frequent itemset* untuk masing-masing *klaster* dan dilakukan proses perhitungan *cluster support* dari itemset-itemset yang memenuhi *cluster minimum support*. *Cluster support* suatu itemset X merupakan jumlah transaksi yang mengandung itemset X dalam suatu *klaster*.



Untuk mempermudah proses perhitungan *cluster support*, maka pada saat proses pembentukan *klaster* disimpan juga *BitSet* item-item untuk masing-masing *klaster* yang merepresentasikan ada tidaknya item pada suatu transaksi. Untuk contoh data transaksi di atas maka *BitSet* yang terbentuk akan terlihat seperti pada tabel 2.13 berikut :

Tabel 2.13 *BitSet* item-item untuk klaster 1

TID	C	G	K	S	T
1	1	0	1	1	1
2	1	1	0	1	0
3	1	0	1	1	1
4	1	1	0	1	1
5	1	1	1	1	1
6	1	1	1	0	0

Untuk perhitungan *cluster support* 1-itemset dapat dilakukan dengan menghitung jumlah Bit 1 pada masing-masing *BitSet*. Jadi hasil *cluster support* 1-itemset yang memenuhi *cluster minimum support* 50% atau minimal 3 transaksi akan terlihat seperti pada tabel 2.14 berikut :

Tabel 2.14 Frequent 1-Itemset

Itemset	Cluster Support
C	6
G	4
K	4
S	5
T	4

Untuk pembangkitan *frequent 2-itemset* dilakukan dengan proses iterasi dengan berdasarkan pada *1-itemset* yang telah terbentuk. Jadi untuk itemset-itemset 2-itemset didapat dari kombinasi itemset-itemset 1-itemset yang sudah terbentuk. Perhitungan *cluster support* dilakukan dengan mengANDkan *BitSet*-*BitSet* dari itemset-itemset pembentuknya kemudian dihitung jumlah Bit 1 dari *BitSet* hasil operasi AND jika jumlahnya lebih besar atau sama dengan nilai *cluster minimum support* maka itemset tersebut dimasukkan ke dalam *frequent*

itemset. Contohnya adalah perhitungan untuk itemset CG seperti pada tabel 2.15 berikut :

Tabel 2.15 Contoh perhitungan support untuk 2-itemset

C	G	CG
1	0	0
1	1	1
1	0	0
1	1	1
1	1	1
1	1	1

Seperti terlihat pada tabel di atas setelah proses AND itemset C dan itemset G maka untuk menghitung support itemset CG dapat dilakukan dengan menghitung jumlah bit 1 pada BitSet CG yaitu 4. Setelah dilakukan proses yang sama terhadap itemset yang lain maka *frequent 2-itemset* yang terbentuk dapat dilihat pada tabel 2.16 berikut :

Tabel 2.16 Frequent 2-itemset

Itemset	Cluster Support	Itemset	Cluster Support
CG	4	GS	3
CK	4	KS	3
CS	5	KT	3
CT	4	ST	4

Untuk pembangkitan frequent 3-itemset sampai k-itemset digunakan suatu teknik pengkombinasian dari itemset-itemset dibawahnya, jadi untuk 3-itemset bisa didapat dengan mengkombinasikan itemset-itemset dari 2-itemset dan demikian seterusnya. Untuk mempercepat proses komputasi maka itemset pembentuknya tidak dikombinasikan secara keseluruhan, tetapi hanya dikombinasikan dengan itemset yang memiliki sejumlah (n-1) item awal yang sama. Contohnya itemset CG hanya akan dikombinasikan dengan itemset yang memiliki 1 item awal sama dengan C, yaitu itemset CK, CS, CT. Sedangkan untuk menghitung support dari itemset yang terbentuk caranya sama dengan perhitungan 2-itemset di atas, yaitu dengan mengANDkan BitSet-BitSet

pembentuknya dan menghitung jumlah bit 1 pada BitSet yang terbentuk. Contohnya pada perhitungan itemset CGK yang merupakan hasil kombinasi itemset CG dan CK yang dapat dilihat pada tabel 2.17 berikut :

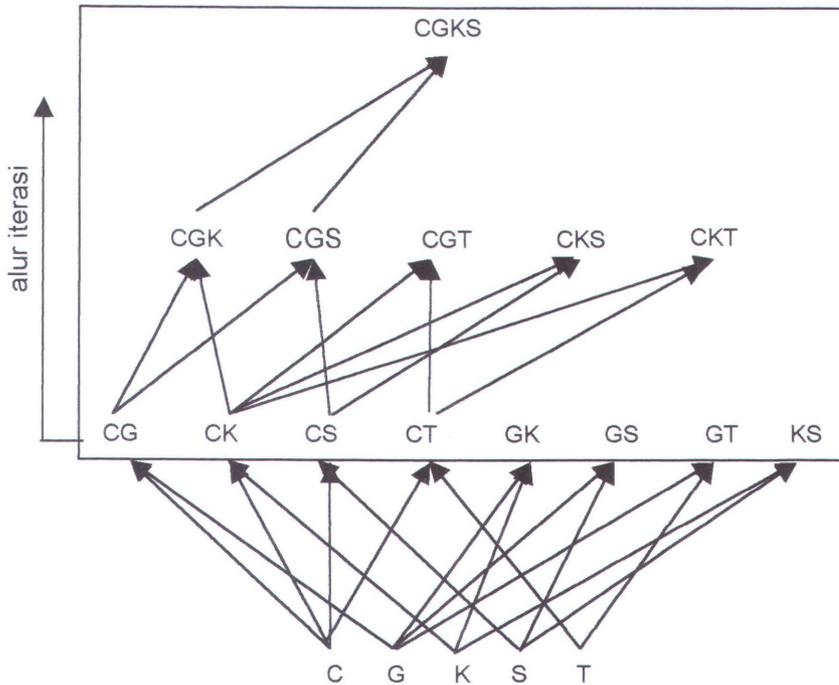
Tabel 2.17 Contoh perhitungan support untuk 3-itemset

CG	CK	CGK
0	1	0
1	0	0
0	1	0
1	0	0
1	1	1
1	1	1

Seperti terlihat pada tabel di atas setelah proses AND itemset CG dan itemset CK maka untuk menghitung support itemset CGK dapat dilakukan dengan menghitung jumlah bit 1 pada BitSet CGK yaitu 2. Proses pembangkitan frequent itemset ini dilakukan sampai tidak ada lagi itemset yang memenuhi *cluster minimum support*. Gambaran untuk pembangkitan frequent itemset secara keseluruhan dapat dilihat pada gambar 2.1. Setelah dilakukan proses yang sama terhadap itemset yang lain maka *frequent 3-itemset* sampai *k-itemset* yang terbentuk dapat dilihat pada tabel 2.18 berikut :

Tabel 2.18 Frequent 3-itemset sampai k-itemset

3-Itemset	Cluster Support	4-Itemset	Cluster Support
CGS	3	CKST	3
CKS	3		
CKT	3		
CST	4		
KST	3		



Gambar 2.1 Pembangkitan frequent itemset

2.3.4 Pencarian Pola Asosiasi Lokal

Secara matematis perhitungan pola asosiasi $A \rightarrow B$ merupakan perbandingan antara jumlah support itemset (A,B) dengan jumlah support item A dan dapat diformulakan sebagai berikut :

$$A \rightarrow B = \frac{Sup_c(\{A,B\})}{Sup_c(\{A\})} \dots\dots\dots(2.5)$$

Dimana A disebut sebagai *antecedent* atau sisi bagian kiri (left-hand side) dan B adalah *consequent* atau sisi bagian kanan (right-hand side). Dan dapat dinyatakan pula dengan kalimat “jika ada item A maka akan ada item B”. Sebagai contoh dalam asosiasi “jika orang membeli palu lalu mereka membeli paku”. “Membeli palu” sebagai antecedent dan “membeli paku” sebagai consequent. Sedangkan contoh untuk mencari pola asosiasi $G \rightarrow C$ pada contoh data transaksi di atas maka:

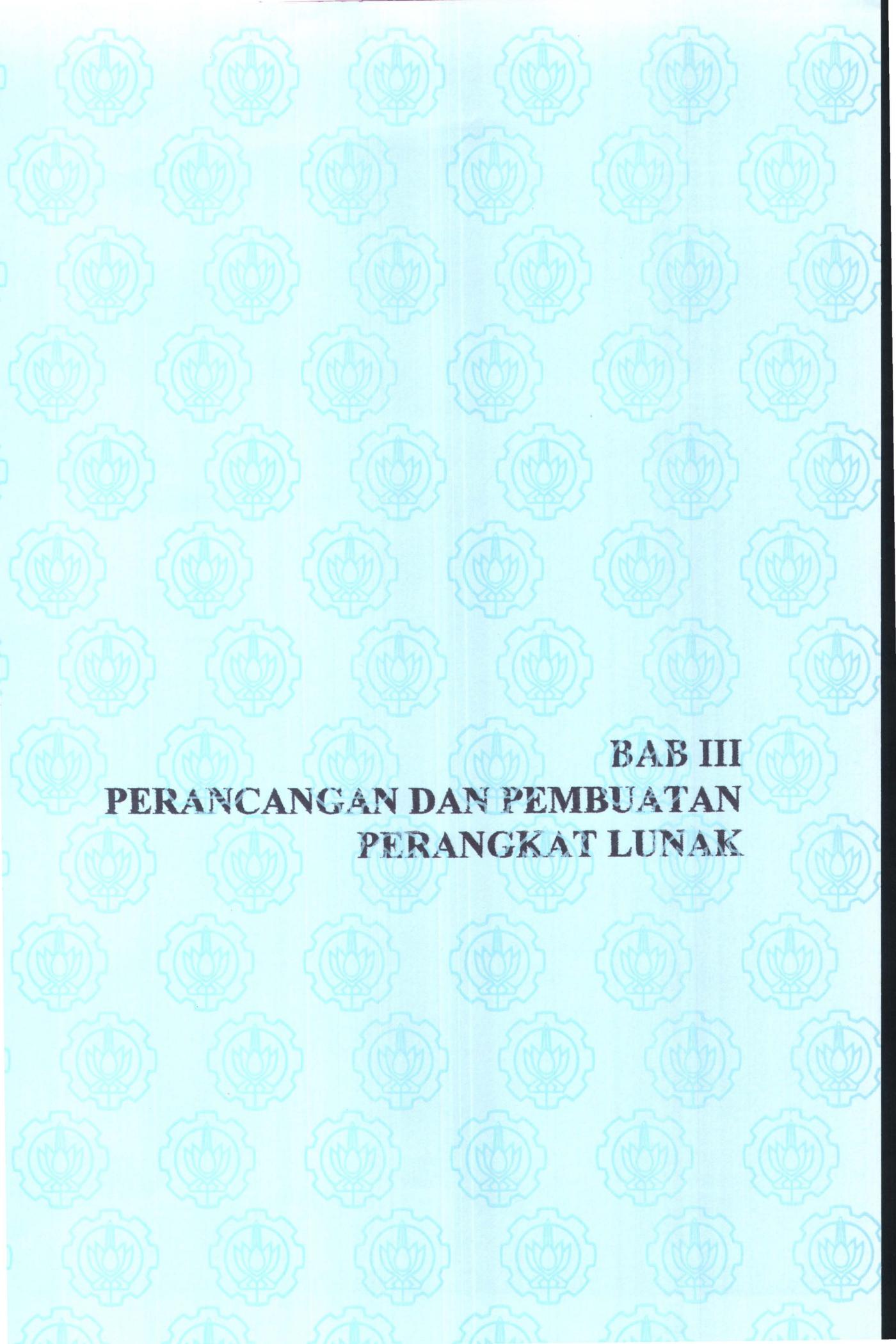
$$G \rightarrow C = \frac{Sup_c(\{C, G\})}{Sup_c(\{G\})}$$

$$= \frac{4}{4} \times 100\% = 100\%$$

$G \rightarrow C$ merupakan salah satu *pola asosiasi lokal* karena memenuhi *cluster confidence support* yang telah ditentukan yaitu 100%. Dengan menggunakan proses yang sama pada data *cluster support* diatas maka hasil seluruh pola asosiasi lokal yang diperoleh seperti terlihat dalam tabel 2.19 berikut:

Tabel 2.19 Hasil pola asosiasi lokal

Pola Asosiasi	Nilai	Pola Asosiasi	Nilai	Pola Asosiasi	Nilai
$G \rightarrow C$	4/4	$CT \rightarrow S$	4/4	$KS \rightarrow C$	3/3
$K \rightarrow C$	4/4	$ST \rightarrow C$	4/4	$KT \rightarrow CS$	3/3
$S \rightarrow C$	5/5	$KT \rightarrow S$	3/3	$CKT \rightarrow S$	3/3
$T \rightarrow C$	4/4	$KS \rightarrow T$	3/3	$KST \rightarrow C$	3/3
$T \rightarrow S$	4/4	$KT \rightarrow C$	3/3	$CKS \rightarrow T$	3/3
$T \rightarrow CS$	4/4	$GS \rightarrow C$	3/3	$KS \rightarrow TC$	3/3



BAB III
PERANCANGAN DAN PEMBUATAN
PERANGKAT LUNAK

BAB III

PERANCANGAN DAN PEMBUATAN

PERANGKAT LUNAK

Dalam bab ini diuraikan mengenai perancangan perangkat lunak yang meliputi perancangan data, perancangan proses dengan pendekatan fungsional yang digambarkan dalam Diagram Alir Data (DAD) dan perancangan antar muka perangkat lunak yang akan dibuat berikut pembuatannya dalam bahasa pemrograman yang digunakan.

3.1 Perancangan Perangkat Lunak

Dalam sub bab ini akan dibahas mengenai perancangan data, baik data masukan, data proses dan data keluaran, perancangan proses serta perancangan antar muka.

3.1.1 Perancangan Data

Dalam perancangan data terdiri dari 3 (tiga) tahapan yaitu data masukan, data proses dan data luaran. Berikut ini adalah pembahasan mengenai tiga tahapan tersebut.

3.1.1.1 Data Masukan

Data masukan yang digunakan dalam perangkat lunak ini adalah data transaksi sintetis yang dibangkitkan *pembangkit data* [APR-02] dan beberapa parameter yang diperlukan dalam proses *mining*. Tabel transaksi sintetis yang

akan digunakan memiliki *field* utama yaitu ID transaksi (TID) dengan tipe data *number* dan ID item (ITEM) dengan tipe data *varchar(10)*. Sedangkan parameter yang diperlukan adalah *jumlah kluster* yang diinginkan pengguna, *cluster minimum support* yang digunakan untuk mendapatkan *cluster support* sampai *k*-itemset dan *cluster confidence support* yang diperlukan untuk mendapatkan pola asosiasi lokal. *Cluster minimum support* dan *cluster confidence support* merupakan prosentase kemunculan yang dikehendaki dari jumlah transaksi yang ada dalam tiap kluster.

3.1.1.2 Data Proses

Pada saat proses mining terdapat beberapa jenis data yang dirancang dan disesuaikan untuk mengimplementasikan algoritma CLASD. Jenis data tersebut antara lain sebagai berikut:

- *Data Afinitas*

Data afinitas merupakan data perbandingan antara *aggregate support 2*-itemset dengan *aggregate support 1*-itemset para pembentuknya. Data ini akan digunakan untuk memperhitungkan kedekatan antara satu transaksi dengan transaksi lainnya untuk membentuk kluster yang baik. Data ini bertipe real positif dengan batasan 0 sampai 1. Sedangkan format data afinitas dapat dilihat pada tabel 3.1 berikut :

Tabel 3.1 Format data afinitas

Itemset	Afinitas
{C,G}	4/6
{C,K}	4/6
{C,S}	5/6
{C,T}	4/6

- *Data Transaksi*

Algoritma CLASD melakukan sekali pembacaan pada basis data dan memindah data tersebut ke dalam bentuk data transaksi yaitu data yang terdiri dari TID dan Item-item anggotanya. Data yang disimpan dalam memori ini akan digunakan untuk proses dalam data mining selanjutnya untuk membentuk *klaster-klaster*. Data ini bertipe array dari string dimana karakter-karakter dari string ini mewakili item-item pada masing-masing transaksi. Adapun format data transaksi dapat dilihat pada tabel 3.2 berikut :

Tabel 3.2 Format data transaksi

TID	ITEM
1	C K S T
2	C G S
3	C K S T
4	C G S T
5	C G K S T
6	C G K

- *Data Metatransaksi*

Data metatransaksi merupakan data transaksi yang mempunyai bobot untuk masing-masing item yang dimilikinya. Untuk transaksi yang memiliki item X, maka bobot untuk item X adalah 1, sedangkan untuk transaksi yang tidak memiliki item X bobot untuk item X adalah 0. Data metatransaksi bisa melakukan operasi penjumlahan antar data metatransaksi. Operasi penjumlah tersebut bisa dilakukan dengan cara menjumlahkan bobot masing-masing item yang dimiliki oleh data metatransaksi. Untuk algoritma CLASD data berbentuk metatransaksi ini dipakai sebagai data *perwakilan klaster*. Data metatransaksi ini bertipe array string dimana karakter-karakter yang berbeda pada string mewakili item-item dan jumlah karakter yang sama

merupakan bobot dari item tersebut untuk masing-masing metatransaksi.

Format data metatransaksi dapat dilihat pada tabel 3.3 berikut :

Tabel 3.3 Format data metatransaksi

M	C	G	K	S	T
1	1	0	1	1	1
2	1	1	0	1	0
3	1	0	1	1	1
4	1	1	0	1	1
5	1	1	1	1	1
6	1	1	1	0	0

3.1.1.3 Data Luaran

Data luaran yang dihasilkan dari perangkat lunak selama proses *mining* terdiri dari 4 (empat) data luaran yang disimpan dalam memori. Adapun empat data luaran tersebut antara lain:

- *Data Hasil Proses*, menyimpan informasi sub proses beserta waktu yang dibutuhkan pada saat proses mining berlangsung maupun waktu yang dibutuhkan untuk seluruh proses. Data ini juga berisi data tentang detil dari data masukan baik itu database, tabel, jumlah klaster, cluster minimum support dan cluster confidence support. Selain itu, data ini juga berisi data tentang jumlah klaster yang terbentuk dan jumlah cluster support yang terbentuk. Data ini berupa data yang ditampilkan di teks pada program seperti yang terlihat pada gambar 3.1 berikut :

Table name	= ITEM5
Record number	= 23
Transaction number	= 6
Item number	= 5
Number of Clusters	= 1
Cluster Minimum Support	= 50.0 %
Cluster Confidence Support	= 100 %
Threshold	= 20 %
Threshold'	= 33 %
TOTAL RUNNING TIME	= [.02]

Gambar 3.1 Format data hasil proses

- *Data Klaster*, menyimpan informasi berupa daftar transaksi-transaksi untuk masing-masing klaster yang terbentuk. Data klaster merupakan data yang berisi data transaksi-transaksi (TID-TID) yang menjadi anggota suatu klaster. Data ini hanya menyimpan data TID saja bukan data transaksi secara keseluruhan, hal ini dilakukan untuk tujuan agar menghemat memori yang dipakai. Dalam algoritma CLASD data klaster ini dipakai dalam pembentukan klaster-klaster. Data ini bertipe array bilangan bulat dimana bilangan ini mewakili transaksi-transaksi pada masing-masing klaster. Format dari data klaster dapat dilihat pada gambar 3.2 berikut :

```

-----
CLUSTER : 1
-----
trans 0 : (COKLAT, KOPI, SUSU, TEH)
trans 1 : (COKLAT, GULA, SUSU)
trans 2 : (COKLAT, KOPI, SUSU, TEH)
trans 3 : (COKLAT, GULA, SUSU, TEH)
trans 4 : (COKLAT, GULA, KOPI, SUSU, TEH)
trans 5 : (COKLAT, GULA, KOPI)

```

Gambar 3.2 Format data klaster

- *Data Cluster Support*, menyimpan informasi berupa daftar cluster support yang diperoleh dari proses mining untuk masing-masing klaster. Data cluster support merupakan data yang berisi data frequent-frequent dari 1-itemset sampai k-itemset untuk tiap klaster yang terbentuk. Dalam masing-masing frequent berisi data itemset beserta nilai support berdasarkan klaster (cluster support). Data cluster support ini dipakai pada proses akhir dalam algoritma CLASD yang akan dipakai untuk perhitungan pola asosiasi lokal. Data ini bertipe array dari Vector dimana masing-masing vector menyimpan data frequent 1-itemset sampai k-itemset untuk masing-masing klaster. Format dari data *Cluster Support* dapat dilihat pada gambar 3.3 berikut :

```

-----
Frequent 1-itemset
-----
(TEH)      support = 4
(SUSU)     support = 5
(KOPI)     support = 4
(GULA)     support = 4
(COKLAT)   support = 6

-----
Frequent 2-itemset
-----
(SUSU TEH ) support = 4
(KOPI TEH ) support = 3
(KOPI SUSU ) support = 3
(GULA SUSU ) support = 3
(COKLAT TEH ) support = 4
(COKLAT SUSU ) support = 5
(COKLAT KOPI ) support = 4
(COKLAT GULA ) support = 4

```

Gambar 3.3 Format data *cluster support*

- *Data Pola Asosiasi Lokal*, menyimpan data hasil pola asosiasi lokal untuk masing-masing klaster. Data ini berupa data hasil pembangkitan pola asosiasi dari data *cluster support* yang hasilnya akan ditampilkan di teks pada program. Format dari data *pola asosiasi lokal* dapat dilihat pada gambar 3.4 berikut :

```

-----
CLUSTER : 1
-----
TEH => SUSU           = 100%
TEH => COKLAT         = 100%
SUSU => COKLAT        = 100%
KOPI => COKLAT        = 100%
GULA => COKLAT        = 100%
KOPI SUSU => TEH      = 100%
KOPI TEH => SUSU      = 100%

```

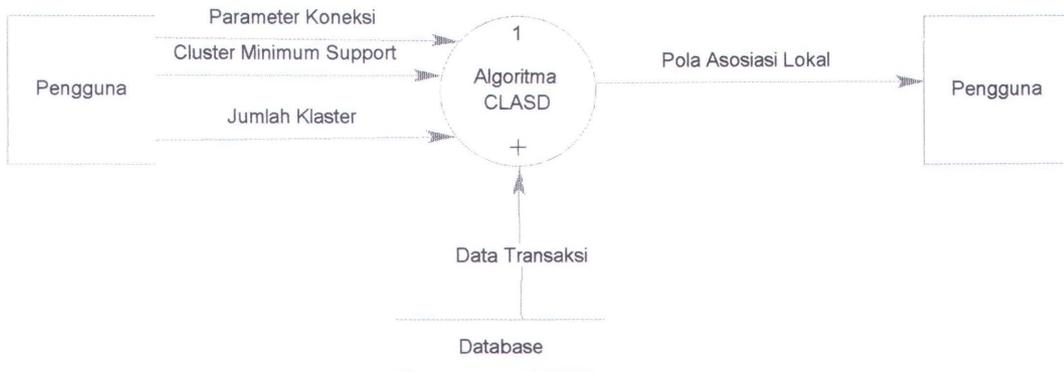
Gambar 3.4 Format data pola asosiasi lokal

3.1.2 Perancangan Proses

Perancangan proses perangkat lunak ini menggunakan pendekatan fungsional yang divisualisasikan dalam Diagram Alir Data (DAD). Pembuatan DAD tersebut menggunakan perangkat lunak Power Designer versi 6.0. Diagram

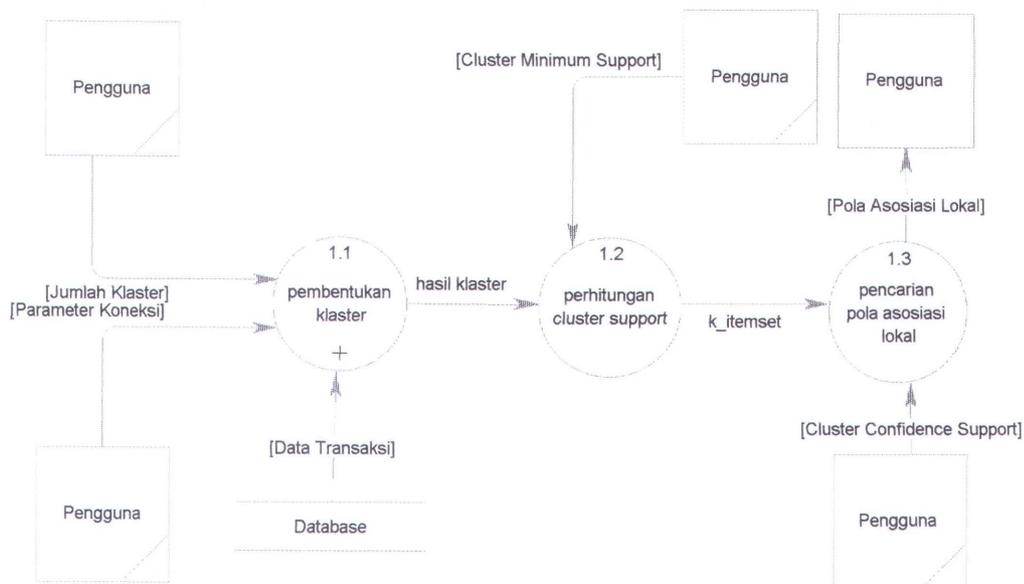
alir data ini akan menjelaskan proses data mining yang dilakukan dari level 0 hingga level 2.

Proses dimulai dari pengguna sebagai entitas eksternal yang melakukan permintaan untuk mendapatkan hasil berupa pola asosiasi lokal. Pengguna memberikan data masukan yang diperlukan untuk menjalankan proses *mining* yaitu parameter koneksi ke basis data termasuk tabel transaksi yang digunakan sebagai obyek mining, jumlah klaster, *cluster minimum support* dan *cluster confidence support*. Dalam proses ini data transaksi sesuai dengan konfigurasi dari pengguna dibaca dari basis data dan keluaran yang akan diperoleh adalah pola asosiasi yang diharapkan. Diagram untuk pencarian pola asosiasi lokal pada level 0 bisa dilihat pada gambar 3.5 berikut:



Gambar 3.5 DAD level 0

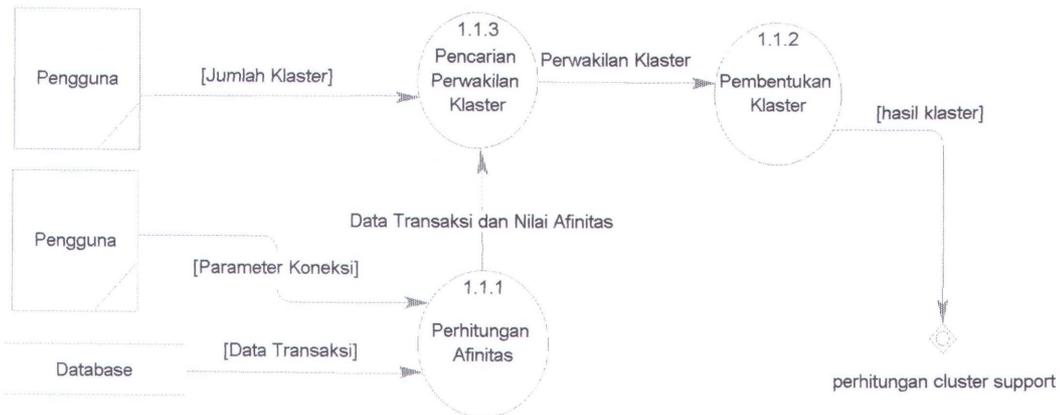
Proses algoritma CLASD dapat didekomposisi dalam DAD level 1 dan dapat dilihat pada gambar 3.6 berikut:



Gambar 3.6 DAD level 1

Proses penggalian pola asosiasi lokal terdiri dari 3 (tiga) sub proses yaitu proses pembentukan kluster, perhitungan *cluster support* dan proses pencarian pola asosiasi lokal. Proses pembentukan kluster akan membaca data transaksi dari basis data dan menghasilkan kluster-kluster yang terbentuk berdasarkan jumlah kluster yang ditentukan oleh pengguna. Proses akan didekomposisi lebih lanjut sampai DAD level 2. Dari kluster-kluster yang terbentuk dilakukan proses perhitungan *cluster support* untuk masing-masing kluster dan mendapatkan k-itemset yang memenuhi syarat *cluster minimum support* yang ditentukan oleh pengguna. Hasil dari proses ini berupa daftar seluruh k-itemset utama yang menjadi masukan untuk proses pencarian pola asosiasi dan juga akan disimpan dalam data *cluster support*. Dari hasil k-itemset utama tersebut dilakukan proses pencarian seluruh pola asosiasi lokal untuk masing-masing kluster yang memenuhi *cluster minimum confidence* yang telah ditentukan. Hasil seluruh pola asosiasi yang diperoleh akan ditampilkan kepada pengguna dan juga akan disimpan dalam data pola asosiasi lokal. Proses

pembentukan kluster dapat didekomposisi dalam DAD level 2 dan dapat dilihat pada gambar 3.7 berikut :

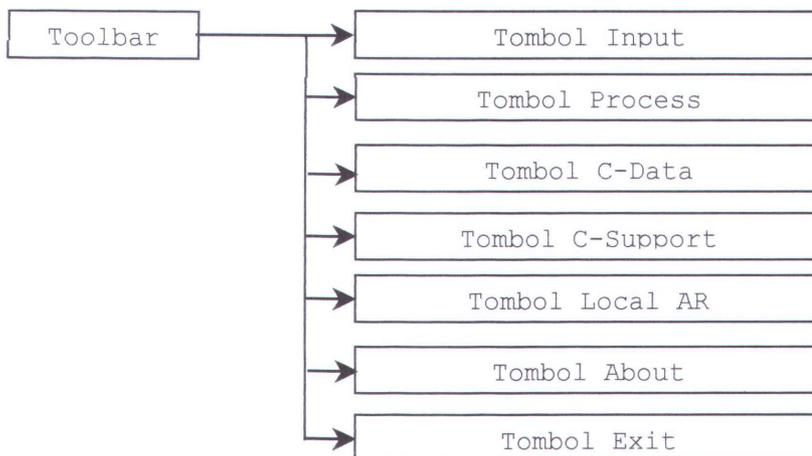


Gambar 3.7 DAD level 2

Ada tiga sub proses yang ada dalam proses pembentukan kluster yaitu proses perhitungan *afinitas*, proses pencarian *perwakilan kluster* dan proses pembentukan *kluster*. Proses perhitungan *afinitas* merupakan proses awal dari seluruh proses algoritma CLASD yang terdiri dari pengaksesan basis data, perhitungan *aggregate support* untuk 1-itemset dan 2-itemset dan dari *aggregate support* tersebut dihitung nilai *afinitas* dari pasangan-pasangan item. Proses perhitungan *afinitas* tadi akan dihasilkan data transaksi dan data nilai *afinitas*. Proses selanjutnya adalah pencarian perwakilan kluster yang berdasarkan pada jumlah kluster yang merupakan masukan pengguna dan data transaksi yang dihasilkan oleh proses sebelumnya. Jumlah perwakilan kluster jauh lebih besar dari jumlah kluster data masukan pengguna. Proses akhir dari pembentukan kluster ini adalah proses pembentukan kluster itu sendiri yaitu proses pembagian data transaksi berdasarkan perwakilan kluster yang terbentuk pada proses sebelumnya. Proses ini akan menghasilkan sejumlah data kluster yang maksimal sama dengan jumlah kluster masukan pengguna.

3.1.3 Perancangan Antar Muka

Perangkat lunak dibangun dalam bentuk *Single Document Interface* (SDI). Untuk melakukan konfigurasi, penggalian pola asosiasi maupun mendapatkan laporan dari proses *mining* pengguna dapat mengakses toolbar yang disediakan. Pada toolbar ada 7 (tujuh) tombol yang disediakan dengan fungsi masing-masing. Rancangan dari toolbar tersebut dapat dilihat pada gambar 3.8 berikut :



Gambar 3.8 Struktur toolbar aplikasi

Fungsi dari tiap tombol pada toolbar tersebut adalah sebagai berikut:

- *Tombol Input*

Tombol ini digunakan untuk mengatur koneksi ke basis data di Oracle, pemilihan tabel data transaksi sebagai obyek *mining*, penentuan parameter-parameter kluster yang terdiri dari jumlah kluster, *cluster minimum support* dan *cluster confidence support* yang diperlukan untuk mendapatkan kluster dari basis data dan untuk mendapatkan seluruh k-itemset utama dan pola asosiasi untuk masing-masing kluster yang terbentuk.

- *Tombol Process*

Tombol ini berfungsi untuk menjalankan proses *cluster mining* yang akan disertai laporan mengenai tiap sub proses dan akumulasi waktunya. Selain itu juga dilaporkan mengenai data basis data dan tabel serta parameter-parameter yang merupakan masukan pengguna.

- *Tombol C-Data*

Tombol ini digunakan untuk menampilkan data transaksi untuk tiap-tiap klaster hasil proses *mining*. Sebelum data ditampilkan pengguna bisa memilih data transaksi dari klaster yang mana yang akan ditampilkan atau dari seluruh klaster yang terbentuk.

- *Tombol C-Support*

Tombol ini digunakan untuk menampilkan data *cluster support* untuk tiap-tiap klaster. Sebelum data ditampilkan pengguna bisa memilih *cluster support* dari klaster yang mana yang akan ditampilkan atau dari seluruh klaster yang terbentuk.

- *Tombol Local AR*

Tombol ini digunakan untuk menampilkan data *pola asosiasi lokal* yang terbentuk berdasarkan *cluster support* untuk tiap-tiap klaster. Sebelum data ditampilkan pengguna bisa memilih *pola asosiasi lokal* dari klaster yang mana yang akan ditampilkan atau dari seluruh klaster yang terbentuk.

- *Tombol About*

Tombol ini berfungsi untuk menampilkan data identitas diri dari perancang aplikasi.

- *Tombol Exit*

Tombol ini digunakan untuk keluar dari aplikasi.

3.2 Pembuatan Perangkat Lunak

Pada sub bagian ini akan dijelaskan pembuatan dari perancangan data, proses, desain antar muka dan alur penggunaannya yang dibangun dengan menggunakan bahasa pemrograman Java.

3.2.1 Pembuatan Data

Pembuatan data terdiri dari tiga tahap yaitu data masukan, data proses, dan data luaran menggunakan bahasa pemrograman Java. Tahapan-tahapan tersebut akan dijelaskan pada sub bab-sub bab berikut:

3.2.1.1 Data Masukan

Data transaksi yang digunakan sebagai obyek *mining* disimpan dalam database di RDBMS Oracle. Nama *field* yang digunakan pada data transaksi adalah TID, menunjukkan ID transaksi dan ITEM, menunjukkan nama item. Pembuatan tabel transaksi di Oracle dan pengisian data transaksi dibangkitkan dengan menggunakan *pembangkit data* [APR-02]. Sedangkan struktur tabelnya dapat dilihat pada tabel 3.4 berikut ini:

Tabel 3.4 Struktur tabel data transaksi

Nama Field	Tipe Data
TID	Number
ITEM	Varchar(10)

Untuk koneksi ke basis data dan pembacaan tabel transaksi dilakukan dengan menggunakan JDBC Thin Driver. Adapun data masukan yang diperlukan adalah IP server, nomor port, nama SID, username, dan password. Proses koneksi ke basis data dan pembacaan tabel transaksi dapat dilihat pada gambar 3.9 berikut :

```

Connection conn;
ResultSet rs;
Statement stmt;

//koneksi ke basis data
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
String conn_str = "jdbc:oracle:thin:@IPServer:noPort:namaSID;
conn = DriverManager.getConnection(conn_str,username,password);

//pembacaan tabel transaksi
stmt = conn.createStatement();
rs = stmt.executeQuery("select * from namatabel order by item,tid");

```

Gambar 3.9 Koneksi ke basis data dan pembacaan tabel transaksi

3.2.1.2 Data Proses

Seperti yang telah dijelaskan dalam sub bab perancangan data di atas, data proses terdiri dari tiga bagian yaitu :

- *Data Afinitas*

Dengan menggunakan bahasa pemrograman Java, data afinitas diimplementasikan dalam bentuk variabel hAffinity yang merupakan Array class double sebanyak $((n-1)*n)/2$ dimana n adalah jumlah item. Masing-masing double menyimpan nilai *afinitas* dari masing-masing pasangan item. Pembentukan data tersebut dapat dilihat pada gambar 3.10 sebagai berikut:

```

private double[] hAffinity;

int if2 = ((intItem*(intItem-1))/2);
//intItem = jumlah item;
//if2 = jumlah seluruh kombinasi dari item
hAffinity = new double[if2];

```

Gambar 3.10 Pembentukan data afinitas

- *Data Transaksi*

Data *transaksi* yang merupakan hasil pembacaan basis data diimplementasikan dalam bentuk variabel hTrans yang merupakan Array class String sebanyak jumlah transaksi dimana masing-masing string terdiri dari

karakter-karakter yang mewakili item-item yang dimiliki tiap data transaksi.

Pembentukan data tersebut dapat dilihat pada gambar 3.11 berikut:

```
private String[] hTrans;

hTrans = new String[intTID];
//intTID = jumlah transaksi
```

Gambar 3.11 Pembentukan data transaksi

- *Data Metatransaksi.*

Data *metatransaksi* merupakan data transaksi yang memiliki bobot untuk masing-masing item yang dimilikinya. Implementasi dari data metatransaksi ini berupa class *Meta_CLASD*. Gambaran mengenai class *Meta_CLASD* bisa dilihat pada gambar 3.12 berikut :

```
public class Meta_CLASD {
    String smeta, strans
    //smeta=string untuk menyimpan data dalam bentuk metatransaksi
    //contohnya:CCGKSST
    //strans=string untuk menyimpan data dalam bentuk transaksi
    //contohnya:CGKST

    Meta_CLASD()
    //konstruktor
    void concat(String padd)
    //menjumlahkan dengan data metaransaksi padd
    void set(String padd)
    //mengeset data metatransaksi=padd
    String getMeta()
    //mendapatkan data dalam bentuk metatransaksi
    String getTrans()
    //mendapatkan data dalam bentuk transaksi
```

Gambar 3.12 Class Meta_CLASD

3.2.1.3 Data Luaran

Seperti dalam perancangan data di atas, data luaran terdiri dari empat bagian yaitu:

- *Data Hasil Proses*

Data hasil proses yang berupa akumulasi waktu proses, dan data masukan oleh pengguna disimpan dalam bentuk variabel dalam memory disesuaikan

dengan type data dari masing-masing data. Data ini langsung ditampilkan di teks pada program setelah proses mining selesai.

- *Data Klaster*

Implementasi data *klaster* diimplementasikan dengan menggunakan Array class *int* sebanyak jumlah klaster yang terbentuk dimana masing-masing *int* mewakili TID atau id transaksi yang merupakan anggota dari masing-masing klaster.

- *Data Cluster Support*

Data *Cluster Support* diimplementasikan dalam bentuk Array class *Vector* sebanyak jumlah klaster yang terbentuk. Dimana vector itu sendiri terdiri dari class *Hashtable* yang mewakili 1 sampai k-itemset dan berisi data nama itemset dan nilai *support*.

- *Data Pola Asosiasi Lokal*

Data pola asosiasi lokal diimplementasikan dalam bentuk display di teks saja dan tidak disimpan dalam memori.

3.2.2 Pembuatan Proses

Proses-proses pada algoritma CLASD diimplementasikan bentuk prosedur-prosedur untuk masing-masing proses, yaitu:

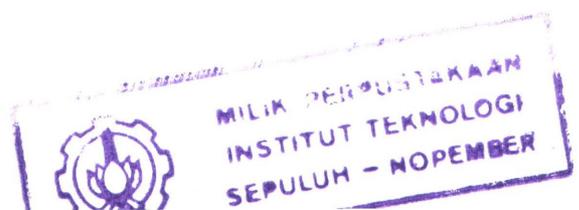
a. Proses Pembentukan Klaster

Proses awal dari pembentukan klaster ini adalah menghitung nilai Afinitas untuk pasangan item. Pada proses ini dimulai dengan perhitungan *aggregate support* 1-itemset dan *aggregate support* 2-itemset untuk seluruh kemungkinan itemset tanpa minimum support. Dari data 1-itemset dan 2-itemset dihitung nilai afinitas untuk pasangan-pasangan item.

Proses selanjutnya adalah membentuk perwakilan klaster dimana jumlahnya adalah tiga puluh kali jumlah masukan *klaster*. Pembentukan perwakilan klaster ini adalah dengan cara mengambil secara random dari data transaksi sejumlah tiga puluh kali jumlah *klaster* awal. Setelah itu proses berikutnya adalah operasi *gabung* yang merupakan proses untuk menggabung perwakilan-perwakilan klaster yang memiliki kedekatan dibandingkan dengan lainnya.

Proses *tugas* adalah proses selanjutnya dimana data transaksi sejumlah 1000 yang diambil secara random dibandingkan dengan perwakilan-perwakilan klaster untuk mendapatkan perwakilan klaster yang paling dekat kemudian data transaksi tersebut dimasukkan ke klaster dengan indek yang sama. Setelah proses *tugas* dilakukan proses *hapus* yang berfungsi untuk menghapus perwakilan klaster yang punya indek yang sama dengan klaster yang mempunyai data transaksi kurang dari 20% data transaksi rata-rata seluruh data yang telah ada di klaster-klaster.

Proses tersebut di atas dilakukan perulangan sampai suatu kondisi tertentu terpenuhi. Kondisi tersebut adalah dimana jumlah perwakilan klaster kurang dari jumlah klaster masukan, atau jumlah transaksi pada klaster yang terbesar lebih dari 33% jumlah total transaksi yang ada di seluruh klaster. Jika kondisi tersebut di atas terpenuhi maka proses akhir dari pembentukan klaster ini adalah proses *tugas* untuk seluruh transaksi dalam basisdata dibandingkan dengan perwakilan klaster yang tersisa. Prosedur-prosedur untuk proses pembentukan klaster bisa dilihat pada gambar 3.13 berikut:



```

public class Frame_CLASD {
    void getAffinity()
        //perhitungan afinitas
    void createClustRep()
        //pembentukan perwakilan klaster
    void Merge(int iCount)
        //proses penggabungan dua perwakilan klaster yang terdekat
    void Assign(int istart,int iend,double dthres)
        //penempatan transaksi ke klaster-klaster terdekat
    void Kill(double dthreshold, boolean doProject)
        //penghapusan perwakilan klaster yang kurang berguna
    boolean UpdateCriterion(double dthreshold)
        //cek kriteria iterasi
}

```

Gambar 3.13 Prosedur-prosedur proses pembentukan klaster

b. Proses Perhitungan Cluster Support

Pada tahap akhir dari proses pembentukan klaster dibentuk suatu data class BitSet yang menyimpan data untuk masing-masing item representasi ada tidaknya suatu klaster pada suatu transaksi tertentu. Data ini dibuat dalam bentuk class Cbits_CLASD seperti yang terlihat pada gambar 3.14 berikut:

```

class Cbits_CLASD extends Object {
    BitSet[] hBitz;
    Cbits_CLASD(int pJumItem)
        //konstruktor
    void inc(int pIndex, int pTrans)
        //isi data bitset
    BitSet get(int pIndex)
        //mendapatkan data bitset
}

```

Gambar 3.14 Class Cbits_CLASD

Proses perhitungan *cluster support* selanjutnya adalah perhitungan 1-itemset dengan cara menghitung representasi data transaksi pada tiap BitSet item yang lebih besar atau sama dengan nilai *cluster minimum suport*. Untuk perhitungan 2-itemset sampai k-itemset dilakukan dengan cara mengkombinasikan data itemset sebelumnya dan mengANDkan BitSet-BitSet itemset pembentuknya dan menghitung representasi data transaksi pada BitSet

hasil proses AND tersebut yang lebih besar atau sama dengan nilai *cluster minimum suport*. Proses ini dilakukan untuk masing-masing klaster.

c. Proses perhitungan Pola Asosiasi Lokal

Proses perhitungan pola asosiasi lokal dilakukan setelah proses perhitungan *cluster support*. Proses perhitungannya dilakukan dengan cara melakukan proses perulangan mulai dari 2-itemset sampai k-itemset. Dimana pada tiap perulangan dilakukan pencarian subset-subset dari itemset yang mempunyai nilai *cluster support*, kemudian dilakukan perhitungan dengan cara membagi nilai cluster support itemset dibagi dengan nilai cluster support subsetnya. Jika hasil pembagian tersebut lebih besar atau sama dengan masukan *cluster confidence support* maka ditampilkan.

3.2.3 Implementasi Antar Muka

Implementasi antar muka dari aplikasi CLASD terdiri dari form utama yang memiliki toolbar pada bagian atas untuk mengakses fungsi-fungsi dari aplikasi dan pada bagian tengah terdapat area teks untuk menampilkan laporan sesuai dengan fungsi dari masing-masing tombol pada toolbar. Adapun gambarannya dapat dilihat pada gambar 3.15 berikut:



Gambar 3.15 Form utama aplikasi CLASD

Berdasarkan perancangan antar muka pada sub bab sebelumnya, dilakukan implementasi sesuai dengan tombol-tombol pada toolbar dengan fungsi dari masing-masing tombol. Adapun implementasinya bisa dilihat pada gambar 3.16 berikut:



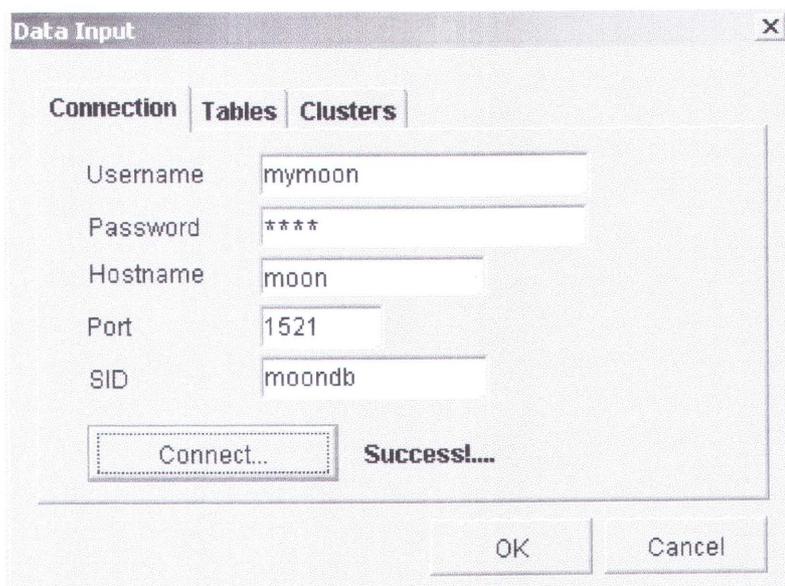
Gambar 3.16 Toolbar aplikasi CLASD

Sedangkan gambaran fungsi dari masing-masing tombol pada toolbar adalah sebagai berikut:

a. Tombol *Input*

Sesuai dengan fungsinya bila tombol ini dipilih maka akan muncul form yang terdiri dari 3 (tiga) tabulasi antara lain tabulasi *Connection* untuk pengisian

parameter koneksi ke basis data, tabulasi *Tables* untuk memilih tabel yang akan digunakan untuk proses *mining* dan tabulasi *Clusters* untuk pengisian parameter untuk klaster yaitu *number of cluster*, *cluster minimum support*, *cluster confidence support*, *threshold* dan *threshold'*. Gambar untuk ketiga tabulasi ini dapat dilihat masing-masing pada gambar 3.17, 3.18 dan 3.19.

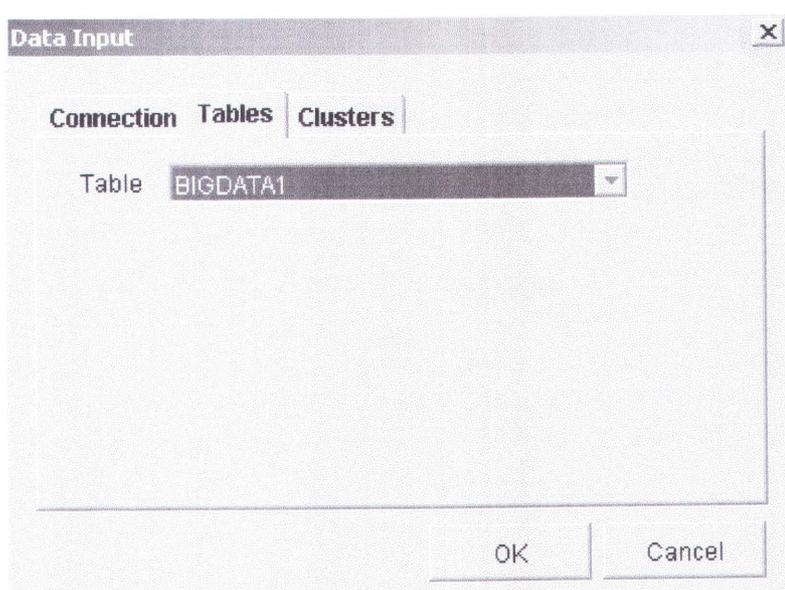


The screenshot shows a 'Data Input' dialog box with three tabs: 'Connection', 'Tables', and 'Clusters'. The 'Connection' tab is selected. It contains the following fields and values:

Username	mymoon
Password	****
Hostname	moon
Port	1521
SID	moondb

Below the fields is a 'Connect..' button and a 'Success!...' label. At the bottom are 'OK' and 'Cancel' buttons.

Gambar 3.17 Form konfigurasi - tabulasi Connection



The screenshot shows a 'Data Input' dialog box with three tabs: 'Connection', 'Tables', and 'Clusters'. The 'Tables' tab is selected. It contains a 'Table' dropdown menu with 'BIGDATA1' selected. At the bottom are 'OK' and 'Cancel' buttons.

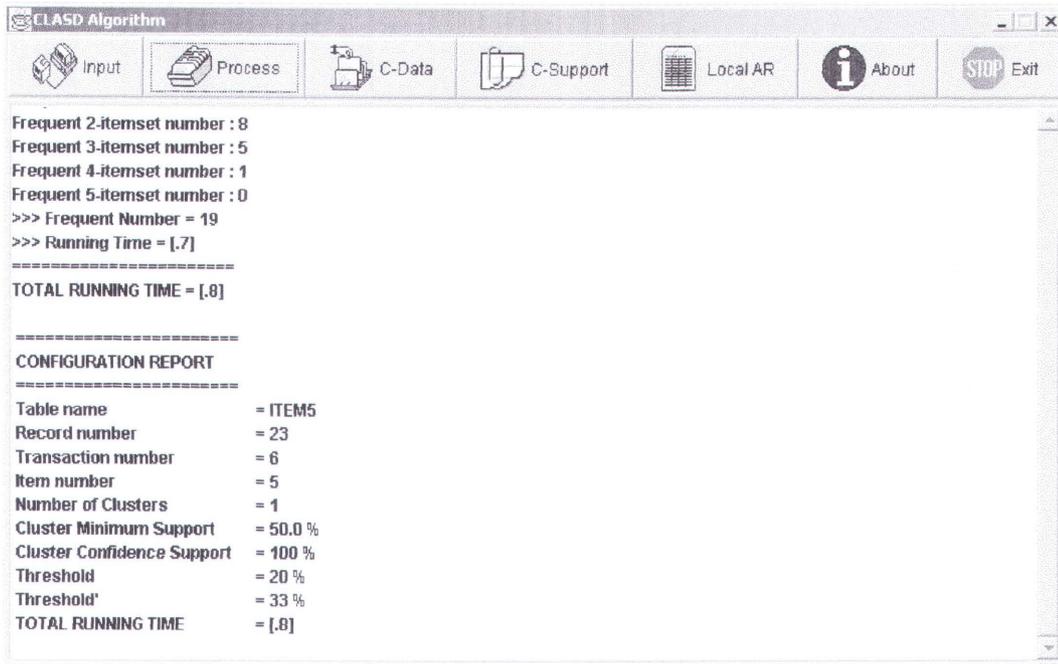
Gambar 3.18 Form konfigurasi – tabulasi Tables

Parameter	Value	Unit
Number of Cluster	10	
Cluster Minimum Support	10.0	%
Cluster Confidence Support	50	%
Threshold	20	%
Threshold'	33	%

Gambar 3.19 Form konfigurasi – tabulasi Clusters

b. Tombol *Process*

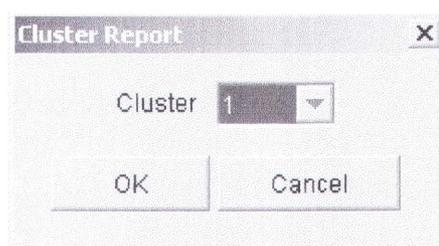
Tombol *Process* adalah untuk menjalankan proses algoritma CLASD sesuai dengan data masukan yang telah dimasukkan sebelumnya melalui tombol *Input*. Proses ini akan menghasilkan laporan berupa sub proses-sub proses yang dikerjakan beserta akumulasi waktunya masing-masing, data mengenai jumlah data transaksi dan jumlah *frequent* dari *cluster support* untuk masing-masing klaster dan konfigurasi data masukan. Gambarannya dapat dilihat pada gambar 3.20 berikut:



Gambar 3.20 Form utama yang menampilkan informasi proses mining

c. Tombol C-Data

Tombol ini berfungsi untuk menampilkan data transaksi untuk tiap kluster yang merupakan hasil dari proses yang telah dijalankan sebelumnya. Jika tombol ini dipilih maka akan muncul form yang berfungsi untuk memilih *kluster* mana yang akan dipilih atau seluruh *kluster* yang akan ditampilkan datanya. Gambar form tersebut dapat dilihat pada gambar 3.21 berikut:



Gambar 3.21 Form pilihan kluster

Setelah dipilih kluster mana yang akan ditampilkan maka akan dilaporkan data transaksi yang merupakan anggota dari kluster tersebut. Gambar 3.22

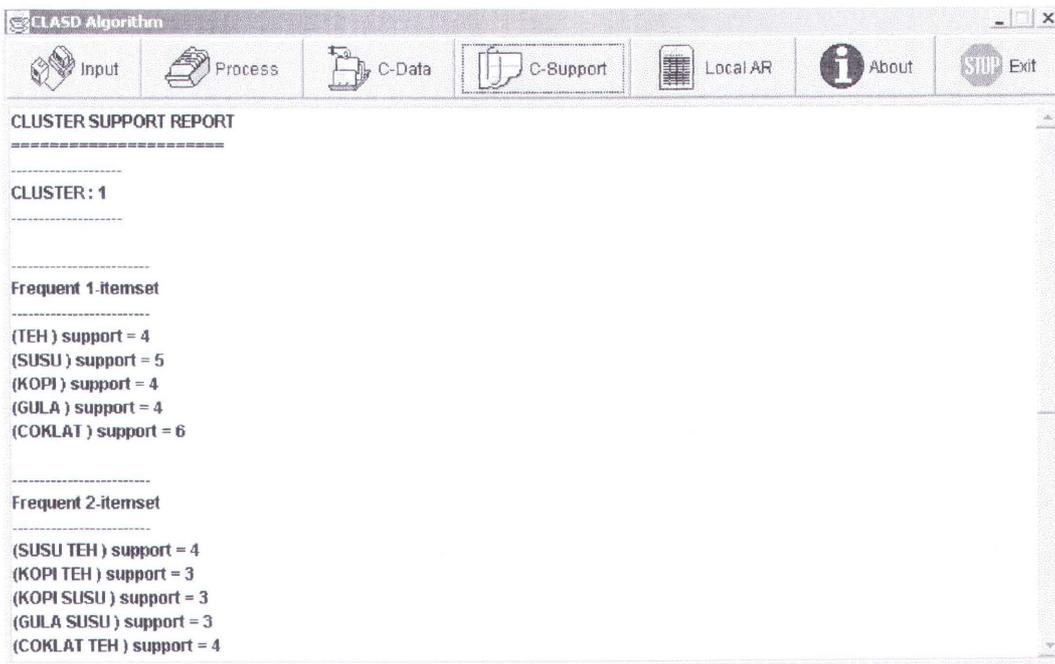
berikut ini adalah gambar form utama yang menampilkan laporan data transaksi berdasarkan klaster yang dipilih:



Gambar 3.22 Form utama dengan informasi data transaksi per klaster

d. Tombol *C-Support*

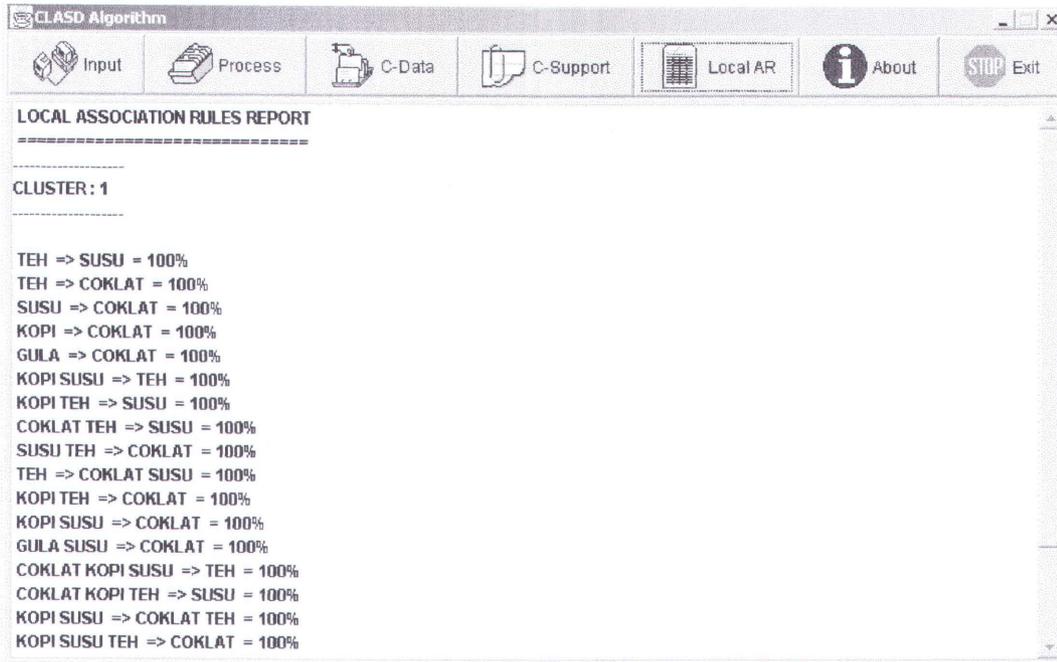
Tombol ini berfungsi untuk menampilkan data *cluster support* untuk tiap klaster yang memenuhi nilai *cluster minimum support*. Jika tombol ini dipilih maka akan muncul form yang berfungsi untuk memilih *klaster* mana yang akan dipilih atau seluruh *klaster* yang akan ditampilkan datanya seperti terlihat pada gambar 3.21 di atas. Setelah ditentukan klaster mana yang akan ditampilkan maka akan diinformasikan data *cluster support* untuk klaster tersebut. Gambar 3.23 berikut ini adalah gambar form utama yang menampilkan laporan data *cluster support* berdasarkan klaster yang dipilih:



Gambar 3.23 Form utama dengan informasi data *cluster support* per klaster

e. Tombol *Local AR*

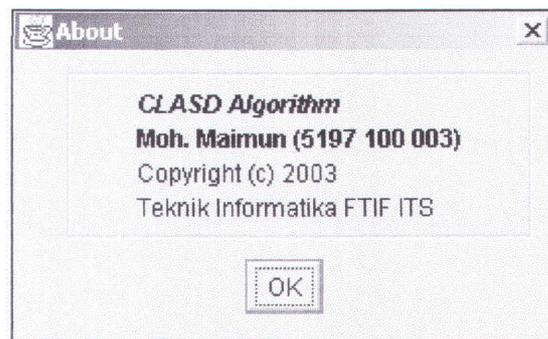
Tombol ini berfungsi untuk menampilkan data *pola asosiasi lokal* untuk tiap klaster yang memenuhi nilai *cluster confidence support*. Jika tombol ini dipilih maka akan muncul form yang berfungsi untuk memilih *klaster* mana yang akan dipilih yang akan ditampilkan datanya seperti terlihat pada gambar 3.21 di atas. Setelah ditentukan klaster mana yang akan ditampilkan maka akan diinformasikan data *pola asosiasi lokal* untuk klaster tersebut. Gambar 3.24 berikut ini adalah gambar form utama yang menampilkan laporan data *pola asosiasi lokal* berdasarkan klaster yang dipilih:



Gambar 3.24 Form utama dengan informasi data *pola asosiasi lokal* per kluster

f. Tombol *About*

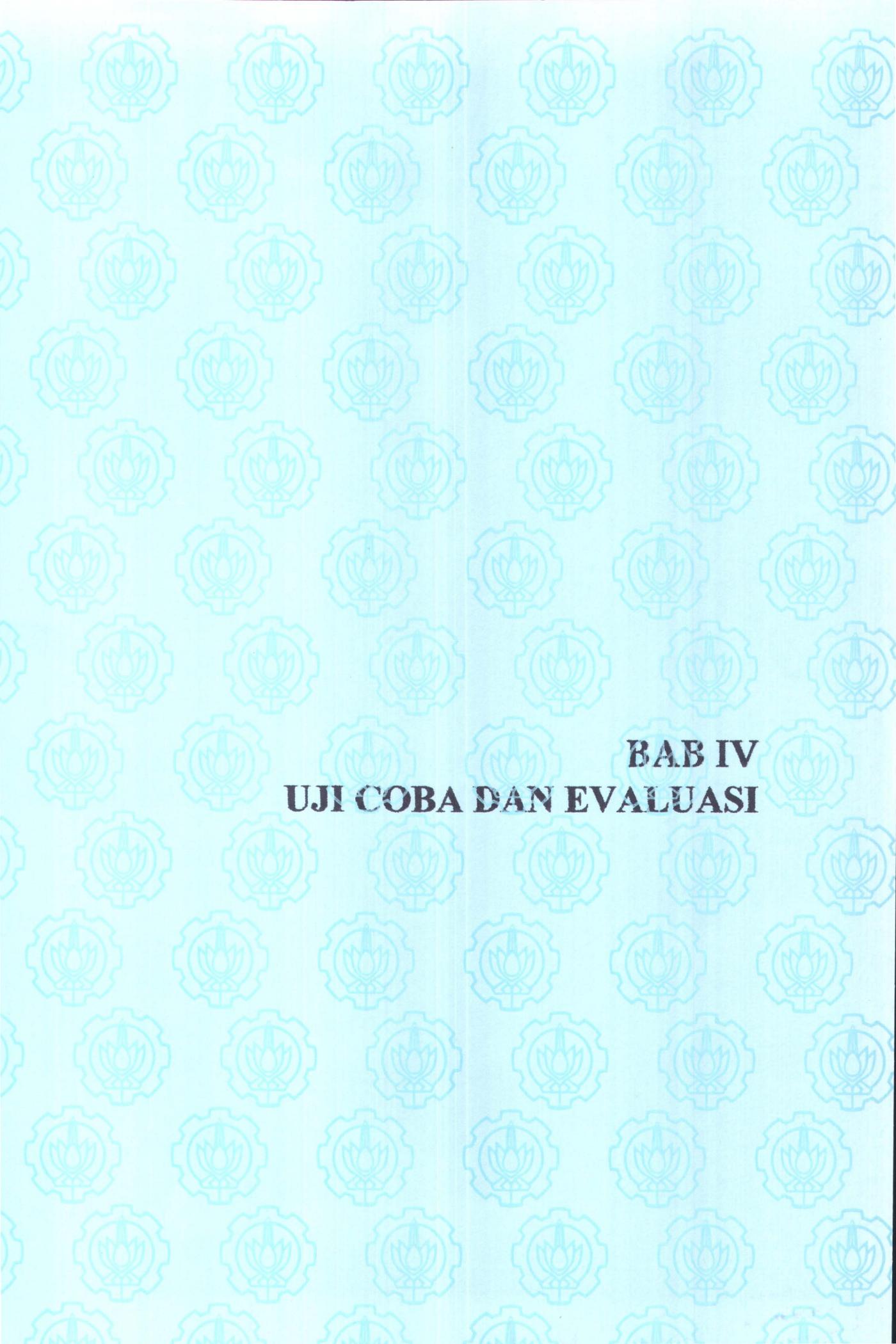
Tombol ini berfungsi untuk menampilkan data identitas pembuat aplikasi CLASD. Gambar 3.25 berikut ini adalah gambar form about:



Gambar 3.25 Form About

g. Tombol *Exit*

Tombol ini berfungsi untuk keluar dari aplikasi CLASD.



BAB IV
UJI COBA DAN EVALUASI

BAB IV

UJI COBA DAN EVALUASI

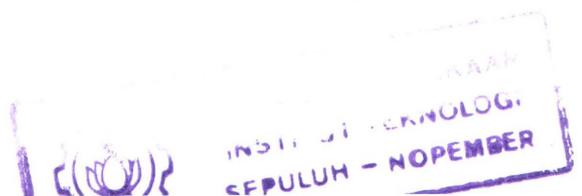
Dalam bab ini diuraikan mengenai hasil uji coba dari perangkat lunak yang telah dibuat. Pengujian ini dilakukan pada beberapa kriteria uji dengan menggunakan data transaksi sintetis yang dibangkitkan dari *pembangkit data* Adhita Pratiwi [APR-02].

4.1 Lingkungan Uji Coba

Pembuatan perangkat lunak ini dilakukan dengan menggunakan bahasa Java menggunakan *compiler* Oracle Jdeveloper versi 3.2 pada sistem operasi Windows 2000 Professional. Data transaksi disimpan dalam RDBMS Oracle8i Enterprise versi 8.1.7. Dengan koneksi antar server dan client database yang menggunakan satu komputer (*single tier*). Perangkat keras yang digunakan adalah komputer dengan prosesor Intel Pentium III 450 MHz dan memori sebesar 256 Mb.

4.2 Data Uji Coba

Data transaksi yang digunakan sebagai obyek pengujian perangkat lunak data mining ini merupakan data sintetis yang juga digunakan pada pengujian perangkat lunak data mining pada tugas akhir yang lain. Spesifikasi dari setiap data yang digunakan pada uji coba kinerja perangkat lunak dapat dilihat pada tabel 4.1. Sedangkan untuk spesifikasi dari setiap data yang



digunakan pada uji coba kehandalan dan perbandingan perangkat lunak dapat dilihat pada tabel 4.2.

Tabel 4.1 Spesifikasi data uji coba kinerja

	NAMA TABEL	JUMLAH RECORD	JUMLAH TRANSAKSI	JUMLAH ITEM
Data1	N100D1k	4869	1000	100
Data2	N198D2k	9676	2000	198
Data3	N1kD10k	99785	10000	1000

Tabel 4.2 Spesifikasi data uji coba perbandingan dan kehandalan

	NAMA TABEL	JUMLAH RECORD	JUMLAH TRANSAKSI	JUMLAH ITEM
Data4	N1kD10k	99785	10000	1000
Data5	N1kD50k	1619021	50000	1000
Data6	N1kD60k	1943087	60000	1000

4.3 Pelaksanaan Uji Coba

Dalam sub bab ini dijelaskan hasil uji coba perangkat lunak dengan menggunakan data sintetis yang telah disebutkan pada sub bab sebelumnya. Pengujian yang dilakukan meliputi pengujian kebenaran hasil pola asosiasi yang diperoleh, pengujian kinerja dan pengujian kehandalan.

4.3.1 Uji Coba Kebenaran

Uji coba kebenaran terdiri atas dua macam uji coba yaitu, uji coba kebenaran pembentukan klaster dan uji coba kebenaran pencarian frequent itemset dan pola asosiasi. Kedua uji coba tersebut akan dijelaskan lebih lanjut pada sub bab-sub bab berikut.

4.3.1.1 Uji Coba Kebenaran Pembentukan Klaster

Uji coba kebenaran pembentukan klaster dilakukan terhadap data sederhana dengan spesifikasi seperti yang terlihat pada tabel 4.3 berikut:

Tabel 4.3 Spesifikasi data uji coba kebenaran pembentukan klaster

KRITERIA	NILAI
Jumlah Record	43
Jumlah Transaksi	20
Jumlah Item	6

Parameter mining yang digunakan adalah jumlah klaster = 2, *cluster minimum support* = 10%, dan *cluster confidence support* = 50%. Dengan menggunakan parameter di atas, hasil yang didapat dari algoritma CLASD dapat dilihat pada tabel 4.4 berikut:

Tabel 4.4 Hasil pembentukan klaster dengan algoritma CLASD

Klaster 1		Klaster 2	
Trnsaksi	Item-item	Transaksi	Item-item
2	B	0	BDGS
3	D	1	GMS
6	D	4	BDKM
8	B	5	BDMS
11	D	7	GKM
13	D	9	BDGM
15	BS	10	GM
16	BD	12	BDGM
17	D	14	DK
18	B		
19	D		

Dari hasil di atas dapat dilihat transaksi-transaksi dalam satu klaster mempunyai kesamaan yang lebih dekat dibandingkan dengan transaksi-transaksi pada klaster yang lain. Dengan menggunakan rumus 2.4 yaitu rumus similiarity (kesamaan antar dua transaksi), maka dapat dihitung nilai kedekatan antar transaksi-transaksi di atas. Secara detil hasil perhitungan rumus tersebut dapat dilihat pada tabel 4.5 berikut:

Tabel 4.5 Hasil perhitungan *similarity* antar transaksi

	Klaster 1											Klaster 2										
	2	3	6	8	11	13	15	16	17	18	19	0	1	4	5	7	9	10	12	14		
Klaster 1	2		0	0	1	0	0	1/2	1/2	0	1	0	1/4	0	1/4	1/4	0	1/4	0	1/4	0	
	3	0		1	0	1	1	0	1/2	1	0	1	1/4	0	1/4	1/4	0	1/4	0	1/4	1/2	
	6	0	1		0	1	1	0	1/2	1	0	1	1/4	0	1/4	1/4	0	1/4	0	1/4	1/2	
	8	1	0	0		0	0	1/2	1/2	0	1	0	1/4	0	1/4	1/4	0	1/4	0	1/4	0	
	11	0	1	1	0		1	0	1/2	1	0	1	1/4	0	1/4	1/4	0	1/4	0	1/4	1/2	
	13	0	1	1	0	1		0	1/2	1	0	1	1/4	0	1/4	1/4	0	1/4	0	1/4	1/2	
	15	1/2	0	0	1/2	0	0		1/3	0	1/2	0	1/2	1/4	1/5	1/2	0	1/5	0	1/5	0	
	16	1/2	1/2	1/2	1/2	1/2	1/2	1/3		1/2	1/2	1/2	1/2	1/2	0	1/2	1/2	0	1/2	0	1/2	1/3
	17	0	1	1	0	1	1	0	1/2		0	1	1/4	0	1/4	1/4	0	1/4	0	1/4	1/2	
	18	1	0	0	1	0	0	1/2	1/2	1		0	1/4	0	1/4	1/4	0	1/4	0	1/4	0	
19	0	1	1	0	1	1	0	1/2	0	0		1/4	0	1/4	1/4	0	1/4	0	1/4	1/2		
Klaster 2	0	1/4	1/4	1/4	1/4	1/4	1/4	1/2	1/2	1/4	1/4	1/4		2/5	1/3	3/5	1/6	1/5	1/5	3/5	1/5	
	1	0	0	0	0	0	0	1/4	0	0	0	0	2/5		1/6	2/5	1/2	2/5	2/3	2/5	0	
	4	1/4	1/4	1/4	1/4	1/4	1/4	1/5	1/2	1/4	1/4	1/4	1/3	1/6		3/5	2/5	3/5	1/5	3/5	1/2	
	5	1/4	1/4	1/4	1/4	1/4	1/4	1/2	1/2	1/4	1/4	1/4	3/5	2/5	3/5		1/6	3/5	1/5	3/5	1/5	
	7	0	0	0	0	0	0	0	0	0	0	0	1/6	1/2	2/5	1/6		2/5	2/3	2/5	1/4	
	9	1/4	1/4	1/4	1/4	1/4	1/4	1/5	1/2	1/4	1/4	1/4	1/5	2/5	3/5	3/5	2/5		1/2	1	1/5	
	10	0	0	0	0	0	0	0	0	0	0	0	1/5	2/3	1/5	1/5	2/3	1/2		1/2	0	
	12	1/4	1/4	1/4	1/4	1/4	1/4	1/5	1/2	1/4	1/4	1/4	3/5	2/5	3/5	3/5	2/5	1	1/2		1/5	
	14	0	1/2	1/2	0	1/2	1/2	0	1/3	1/2	0	1/2	1/5	0	1/2	1/5	1/4	1/5	0	1/5		

Dari hasil perhitungan *similarity* di atas dapat dilihat bahwa nilai kesamaan (*similarity*) antar transaksi yang paling besar terdapat dalam klaster yang sama. Contohnya untuk transaksi 2 nilai kesamaan yang paling besar adalah dengan transaksi 8 senilai 1 yang juga terdapat dalam klaster 1. Dari tabel 4.5 di atas dapat disimpulkan bahwa hasil pembentukan klaster dari algoritma CLASD benar.

4.3.1.2 Uji Coba Kebenaran Pencarian Frequent Itemset dan Pola Asosiasi

Skenario uji coba yang dilakukan untuk pencarian frequent itemset dan pola asosiasi dengan menggunakan obyek tabel sederhana sesuai dengan contoh data transaksi yang digunakan dalam pembahasan algoritma CLASD

dalam buku ini. Adapun spesifikasi tabel tersebut dapat dilihat pada tabel 4.6 berikut:

Tabel 4.6 Spesifikasi data uji coba pencarian frequent itemset dan pola asosiasi

KRITERIA	NILAI
Jumlah Record	23
Jumlah Transaksi	6
Jumlah Item	5

Parameter mining yang digunakan adalah jumlah klaster = 1, *cluster minimum support* = 50%, dan *cluster confidence support* = 100%. Hasil yang diperoleh dari proses mining ini sesuai dengan yang tertulis dalam buku. Laporan yang diberikan oleh perangkat lunak adalah sebagai berikut:

```

Frequent 1-itemset number      = 5
Frequent 2-itemset number      = 8
Frequent 3-itemset number      = 5
Frequent 4-itemset number      = 1
Total frequent itemset number = 19
Association rules number       = 18

```

Uji coba kebenaran selanjutnya dilakukan dengan membandingkan hasil mining algoritma CLASD dan hasil yang diperoleh pada algoritma HYBRID [DTY-02] dengan menggunakan *data1*. Parameter yang digunakan untuk algoritma CLASD adalah jumlah klaster = 1, *cluster confidence support* = 50% dengan *cluster minimum support* mulai 1%-10%. Sedangkan parameter untuk algoritma Hybrid adalah *confidence support* = 50% dan *minimum support* mulai 1%-10%. Hasil numerik waktu proses mining baik waktu pencarian seluruh itemset utama maupun waktu pembentukan pola asosiasi yang disebutkan pada kedua tabel tersebut disajikan dalam format *detik,milidetik*. Tabel 4.7 dan 4.8 berikut menunjukkan hasil mining kedua algoritma tersebut. Dari hasil ini dapat diketahui bahwa jumlah itemset utama dan jumlah pola asosiasi yang dihasilkan oleh

kedua algoritma sama sehingga dapat disimpulkan bahwa hasil yang diperoleh dari algoritma CLASD adalah valid.

Tabel 4.7 Hasil uji coba kebenaran algoritma HYBRID

Kriteria	Minimum Support									
	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Jml itemset utama	931	392	221	146	112	85	68	59	52	39
Jml pola asosiasi	743	308	175	113	96	74	58	53	51	50

Tabel 4.8 Hasil uji coba kebenaran algoritma CLASD

Kriteria	Cluster Minimum Support									
	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Jml itemset utama	931	392	221	146	112	85	68	59	52	39
Jml pola asosiasi	743	308	175	113	96	74	58	53	51	50

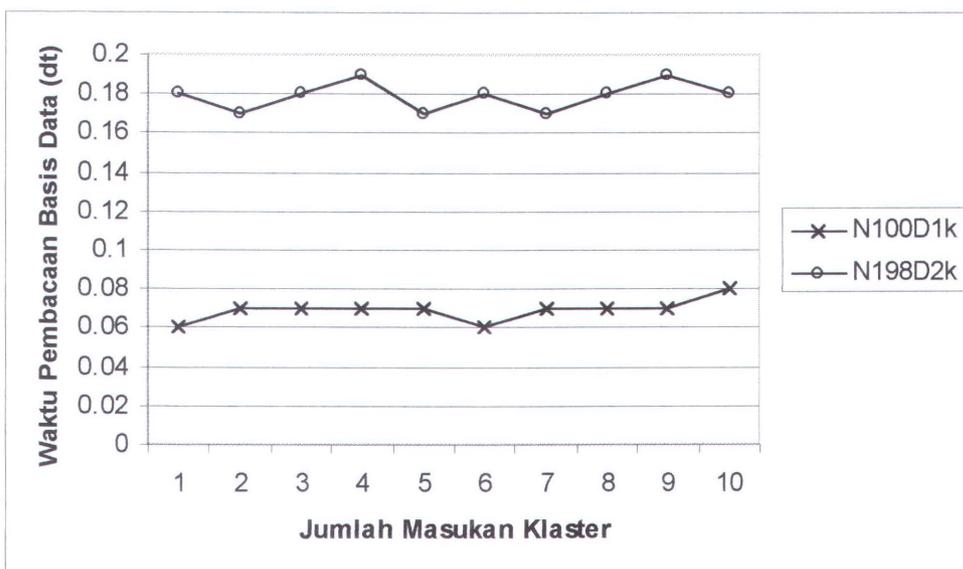
4.3.2 Uji Coba Kinerja

Dalam bagian ini diuraikan hasil uji coba kinerja dari perangkat lunak yang telah dibuat dengan melakukan perbandingan *jumlah masukan klaster* dengan waktu pembacaan basis data, total itemset utama yang ditemukan, jumlah pola asosiasi yang terbentuk, dan total waktu komputasi. Pada setiap perbandingan diberikan grafik hasil uji coba kinerja algoritma. Obyek tabel yang digunakan adalah N100D1k dan N198D2k dengan mengubah parameter *jumlah masukan klaster* dan parameter lainnya adalah *cluster minimum support* = 10%, *cluster confidence support* = 50 %, *threshold* = 20 %, dan *threshold'* = 33 %.

- **Perbandingan *jumlah masukan klaster* dengan waktu pembacaan basis data**

Waktu pembacaan basis data merupakan waktu untuk mendapatkan seluruh record dari data transaksi. Hasil uji coba perangkat lunak menunjukkan bahwa waktu pembacaan basis data cenderung tetap dan tidak dipengaruhi oleh *perubahan jumlah masukan klaster* sehingga waktu pembacaan untuk satu obyek tabel cenderung tetap walaupun dilakukan

perubahan parameter *jumlah masukan klaster*. Pembacaan basis data yang cenderung tetap disebabkan karena perangkat lunak CLASD melakukan pembacaan basis data hanya sekali dan proses berikutnya menggunakan penyimpanan dalam memory. Jadi meskipun dilakukan perubahan parameter *jumlah masukan klaster* pembacaan basis data tetap dilakukan sebanyak satu kali. Grafik pada gambar 4.1 menunjukkan perbandingan masukan *jumlah masukan klaster* dengan waktu pembacaan basis data.

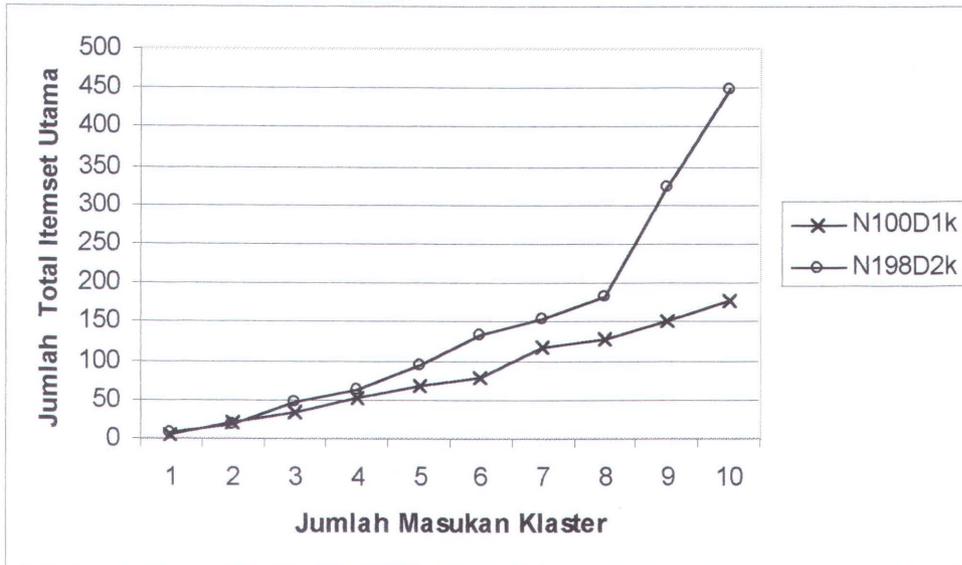


Gambar 4.1 Perbandingan *jumlah masukan klaster* – waktu pembacaan basis data

- **Perbandingan jumlah masukan klaster dengan jumlah total itemset utama yang diperoleh**

Hasil pengujian perangkat lunak ini menunjukkan bahwa semakin besar *jumlah masukan klaster* maka jumlah itemset utama yang diperoleh cenderung semakin besar. Hal ini disebabkan karena semakin besar *jumlah masukan klaster* maka jumlah *klaster* yang terbentuk akan semakin besar dan pencarian itemset yang dilakukan pada masing-masing *klaster* akan menghasilkan jumlah itemset yang semakin besar pula. Grafik perbandingan

jumlah masukan klaster dengan total itemset utama yang diperoleh dapat dilihat pada gambar 4.2 berikut:

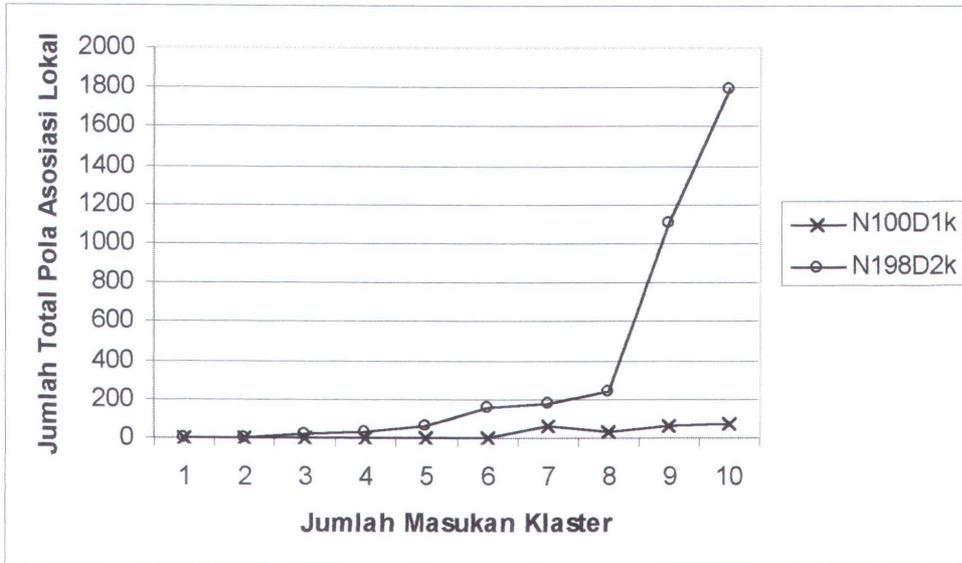


Gambar 4.2 Perbandingan *jumlah masukan klaster* – jumlah total itemset utama

- **Perbandingan jumlah masukan klaster dengan jumlah total pola asosiasi lokal yang terbentuk**

Hasil pengujian perangkat lunak ini menunjukkan bahwa semakin besar *jumlah masukan klaster* maka jumlah total pola asosiasi lokal yang terbentuk cenderung semakin besar. Hal ini disebabkan karena berkaitan dengan semakin besarnya jumlah total itemset utama yang terbentuk sehingga pola asosiasi lokal yang terbentuk juga semakin besar. Grafik perbandingan *jumlah masukan klaster* dengan jumlah total pola asosiasi lokal yang terbentuk dapat dilihat pada gambar 4.3:

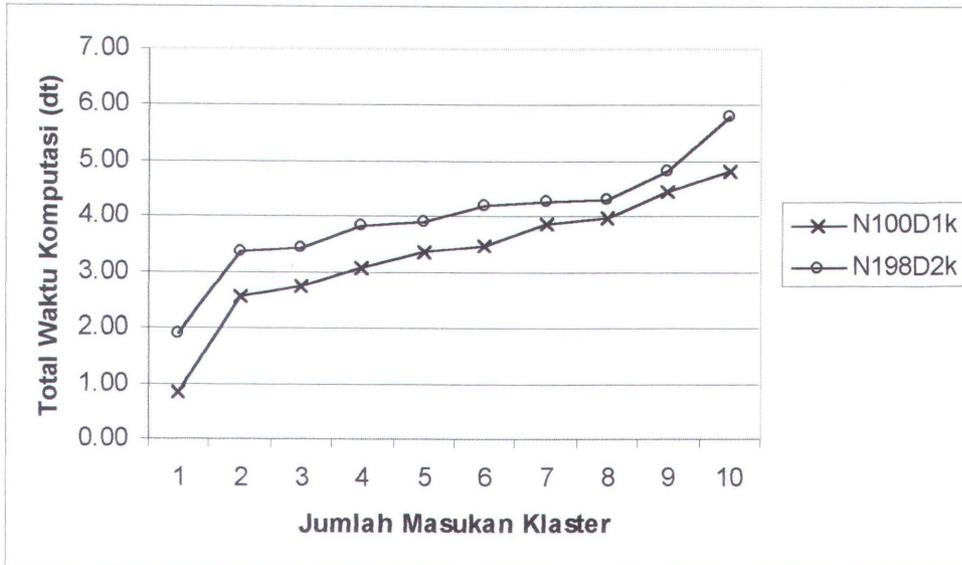




Gambar 4.3 Perbandingan *jumlah masukan klaster* – jumlah total pola asosiasi lokal

- **Perbandingan jumlah masukan klaster dengan total waktu komputasi**

Total waktu komputasi merupakan hasil penjumlahan waktu pencarian itemset utama dan waktu pembentukan pola asosiasi. Hasil pengujian perangkat lunak ini menunjukkan bahwa semakin besar *jumlah masukan klaster* maka total waktu komputasi yang diperlukan cenderung makin besar. Ini berkaitan dengan kecenderungan yang sama untuk hubungan *jumlah masukan klaster* dengan waktu pencarian itemset utama dan waktu pembentukan pola asosiasi. Grafik perbandingan *jumlah masukan klaster* dengan total waktu komputasi dapat dilihat pada gambar 4.4:



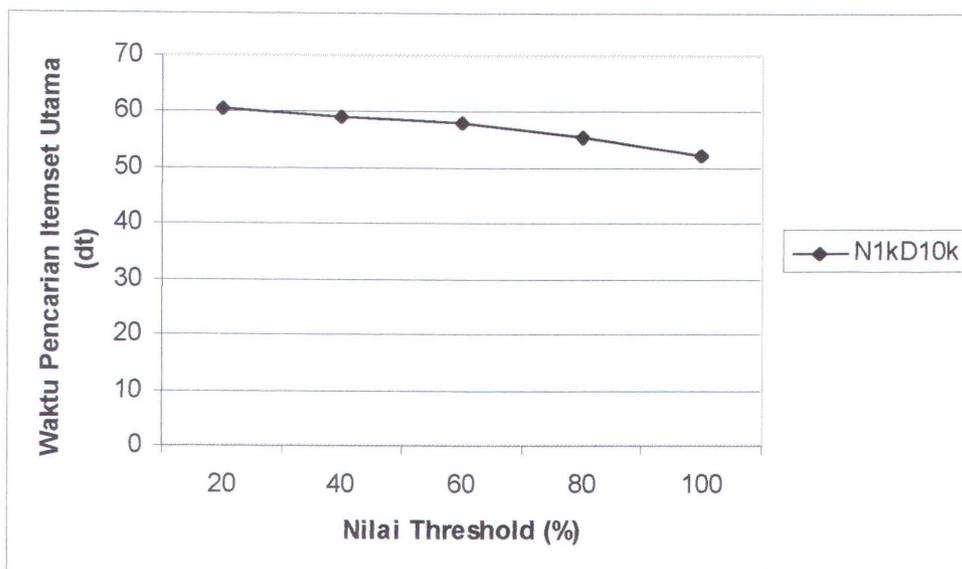
Gambar 4.4 Perbandingan *jumlah masukan klaster* – total waktu komputasi

- **Perbandingan nilai *threshold* dengan waktu pencarian itemset utama**

Pada uji coba ini data masukan tabel yang digunakan adalah N1kD10k dan parameter yang digunakan adalah *jumlah klaster* = 10, *cluster minimum support* = 10%, *cluster confidence support* = 50%, dan *threshold'* = 33%. Nilai *threshold* merupakan nilai persentase dari jumlah rata-rata transaksi yang ada di perwakilan klaster. Nilai ini akan digunakan untuk salah satu proses dalam pembentukan klaster yaitu pada proses penghapusan perwakilan klaster yang kurang berbobot, dimana seperti yang telah dijelaskan pada bab sebelumnya bahwa perwakilan klaster yang memiliki jumlah transaksi kurang dari nilai *threshold* akan dihapus.

Hasil pengujian perangkat lunak ini menunjukkan bahwa semakin besar nilai *threshold* maka waktu pencarian itemset utama cenderung semakin kecil. Hal ini dipengaruhi oleh kondisi jika nilai *threshold* semakin besar maka jumlah perwakilan klaster yang tersaring akan semakin besar. Grafik

perbandingan nilai *threshold* dengan waktu pencarian k-itemset utama dapat dilihat pada gambar 4.5 berikut:



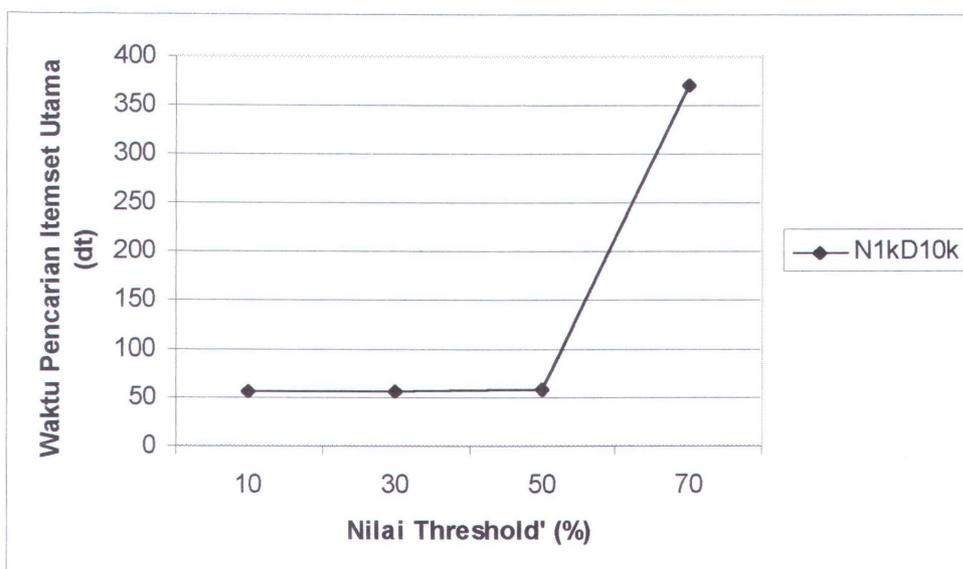
Gambar 4.5 Perbandingan nilai *threshold* – waktu pencarian itemset utama

- **Perbandingan nilai *threshold'* dengan waktu pencarian itemset utama**

Pada uji coba ini data tabel yang digunakan adalah N1kD10k dengan parameter yang digunakan adalah *jumlah klaster* = 10, *cluster minimum support* = 10%, *cluster confidence support* = 50%, dan *threshold* = 20%. Nilai *threshold'* merupakan nilai persentase dari jumlah total transaksi yang ada di perwakilan klaster. Nilai ini akan digunakan untuk salah satu proses dalam pembentukan klaster yaitu pada proses pengecekan kondisi klaster, dimana seperti yang telah dijelaskan pada bab sebelumnya bahwa jika jumlah transaksi dalam k klaster terbesar lebih besar dari nilai *threshold'*, maka proses iterasi akan berhenti.

Hasil pengujian perangkat lunak ini menunjukkan bahwa waktu pencarian itemset utama cenderung tetap, tetapi akan naik secara drastis pada suatu titik tertentu. Hal ini disebabkan karena pada suatu nilai *threshold'*

tertentu (dalam uji coba ini 70%) proses iterasi yang dilakukan dalam algoritma CLASD menjadi lebih banyak. Jadi semakin besar nilai *threshold'* maka semakin besar kemungkinan untuk memperbesar proses iterasi dan memperbesar waktu pencarian k-itemset utama. Grafik perbandingan *nilai threshold'* dengan waktu pencarian k-itemset utama dapat dilihat pada gambar 4.6 berikut:



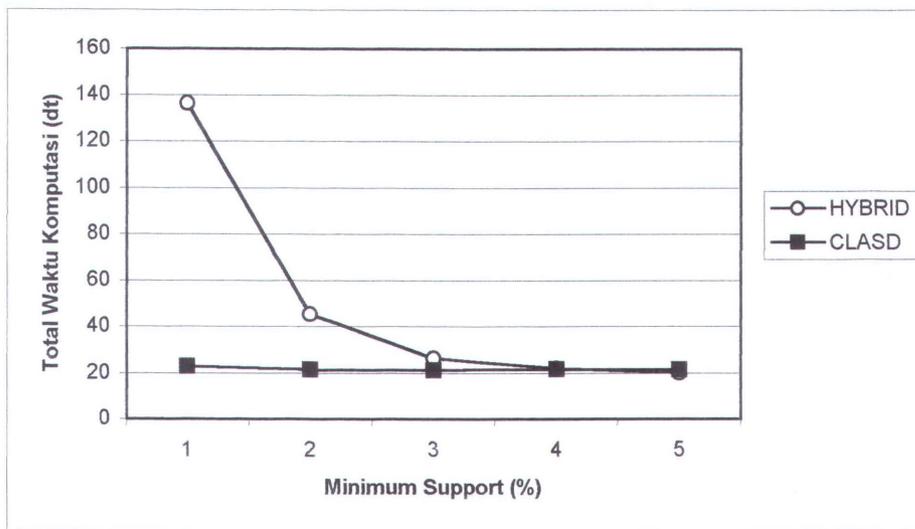
Gambar 4.6 Perbandingan *nilai threshold'* – waktu pencarian itemset utama

4.3.3 Uji Coba Perbandingan

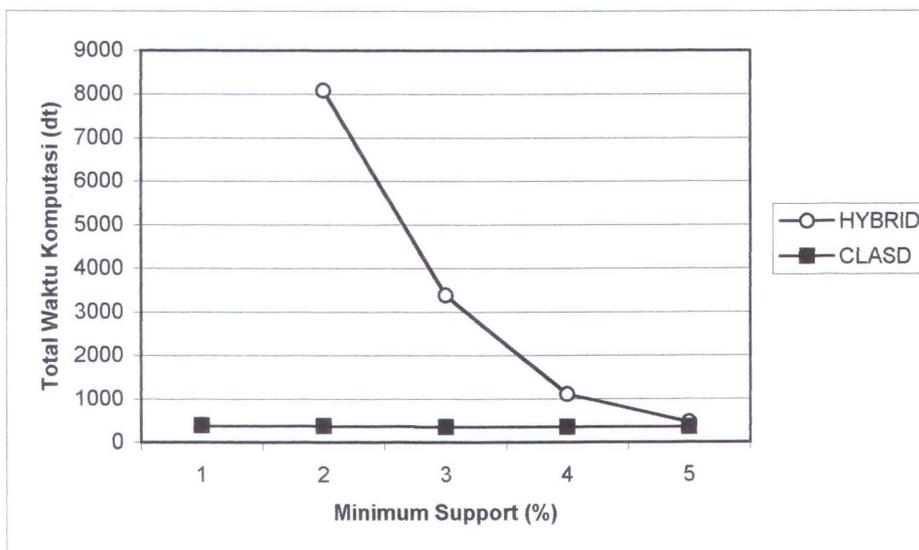
Uji coba perbandingan ini bertujuan untuk menguji kemampuan perangkat lunak algoritma CLASD dibandingkan dengan algoritma HYBRID [DTY-02]. Data tabel yang dipergunakan adalah data pada tabel 4.2. Uji coba yang dilakukan adalah perbandingan waktu komputasi dan penggunaan memori. Karena algoritma HYBRID [DTY-02] merupakan algoritma yang melakukan pencarian pola asosiasi secara global maka untuk menyamakan fungsi, masukan jumlah kluster untuk algoritma CLASD diset sama dengan 1.

- **Perbandingan waktu komputasi**

Hasil uji coba perbandingan antara algoritma CLASD dan algoritma HYBRID [DTY-02] untuk perbandingan waktu komputasi menunjukkan bahwa algoritma CLASD mampu melakukan proses komputasi lebih cepat dari algoritma HYBRID [DTY-02]. Hasil lengkap dari uji coba perbandingan waktu komputasi dapat dilihat pada gambar 4.7 dan 4.8 berikut:



Gambar 4.7 Perbandingan waktu komputasi untuk data N1kD10k



Gambar 4.8 Perbandingan waktu komputasi untuk data N1kD50k

- **Perbandingan penggunaan memori**

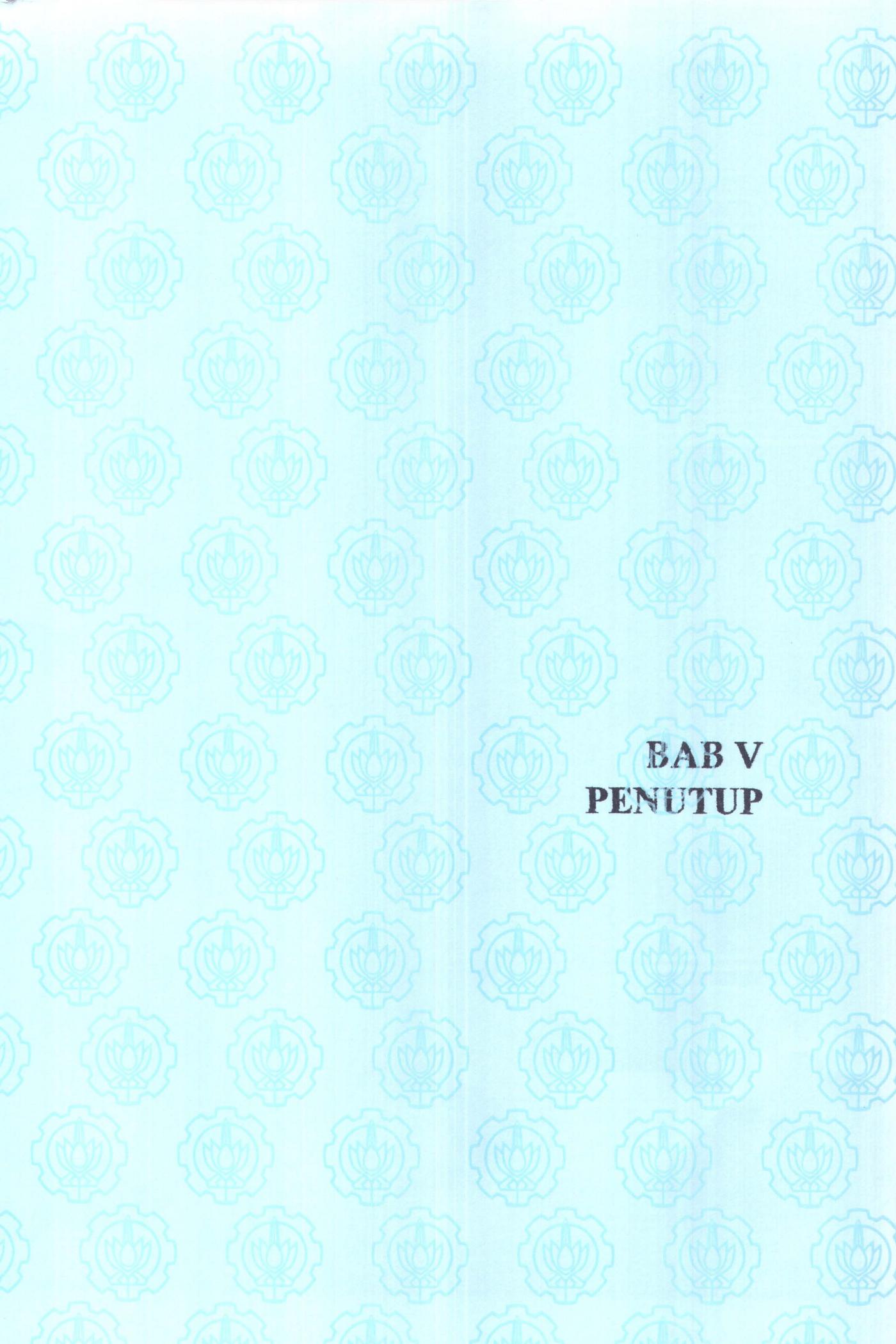
Dari hasil uji coba perbandingan antara algoritma CLASD dan algoritma HYBRID [DTY-02] untuk penggunaan memori menunjukkan bahwa algoritma CLASD membutuhkan memori yang jauh lebih besar daripada algoritma HYBRID [DTY-02]. Hasil dari uji coba perbandingan penggunaan memori dapat dilihat pada tabel 4.9 berikut:

Tabel 4.9 Hasil uji coba perbandingan penggunaan memori untuk data N1kD10k

	penggunaan memori
CLASD	18.064.384 Bytes
HYBRID	3.616.960 Bytes

4.3.4 Uji Coba Kehandalan

Uji coba kehandalan ini bertujuan untuk menguji kemampuan perangkat lunak algoritma CLASD dalam melakukan proses komputasi untuk data yang sangat besar (hampir 2 juta record). Uji coba ini menggunakan data N1kD60k dengan parameter *jumlah klaster* = 1, *cluster minimum support* = 1%, *cluster confidence support* = 50%, *threshold* = 20% dan *threshold'* = 33%. Hasil dari uji coba ini menunjukkan bahwa algoritma CLASD mampu melakukan proses komputasi untuk data yang sangat besar.



BAB V
PENUTUP

BAB V

PENUTUP

Bab ini berisi beberapa kesimpulan dari tugas akhir yang dibuat berdasarkan uji coba yang telah dilakukan. Selain itu disertakan pula kemungkinan yang dapat dilakukan untuk pengembangan lebih lanjut dari tugas akhir ini

5.1 Kesimpulan

Dari beberapa hasil uji coba perangkat lunak yang dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

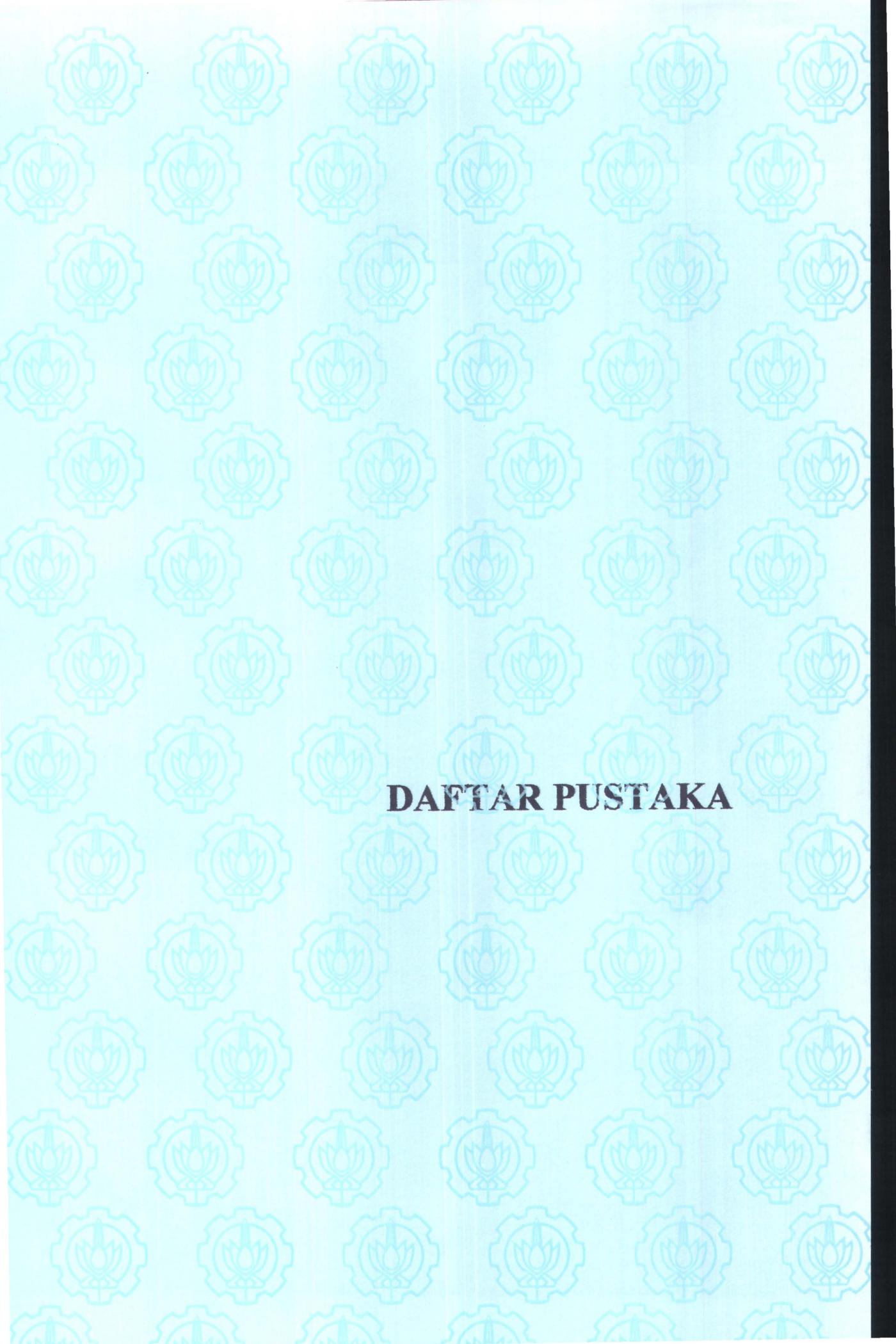
- a. Proses pembacaan basis data mempunyai kecenderungan yang tetap dan tidak dipengaruhi oleh besar kecilnya jumlah masukan klaster, karena pembacaan basis data hanya dilakukan satu kali.
- b. Semakin besar jumlah masukan klaster, maka semakin besar jumlah total itemset utama, pola asosiasi lokal yang diperoleh, dan waktu komputasi yang diperlukan.
- c. Semakin besar nilai *threshold* (*ambang batas rata-rata transaksi*), maka semakin kecil waktu yang diperlukan untuk mendapatkan seluruh itemset utama. Tetapi semakin besar nilai *threshold'* (*ambang batas total transaksi*), maka semakin besar waktu yang diperlukan untuk mendapatkan seluruh itemset utama.

- d. Hasil uji coba perbandingan dengan algoritma HYBRID [DTY-02] menunjukkan bahwa algoritma CLASD mampu melakukan proses komputasi yang lebih cepat daripada algoritma HYBRID [DTY-02], tetapi algoritma CLASD memerlukan memori yang lebih besar daripada algoritma HYBRID [DTY-02].

5.2 Kemungkinan Pengembangan

Perangkat lunak yang didasarkan pada algoritma CLASD memiliki kelemahan pada manajemen memori dan waktu komputasi yang lambat terutama untuk data yang sangat besar dan untuk jumlah masukan klaster yang besar. Pengembangan yang dapat dilakukan adalah dengan menggunakan komputasi paralel, yang dapat dilakukan dengan memecah proses *mining* menjadi beberapa sub proses dan diselesaikan secara terpisah. Bagian pada algoritma CLASD yang memungkinkan untuk dilakukan secara paralel adalah proses pembentukan itemset dan perhitungan cluster supportnya untuk masing-masing klaster yang terbentuk.





DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [AIS-93] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Very Large Databases", *Proceeding of the 1993 ACM SIGMOD Conference*, May, 1993.
- [APR-02] Adhita Pratiwi. "Perancangan dan Pembuatan Perangkat Lunak Pembangkit Data yang Mengikuti Pola Distribusi Poisson untuk Pencarian Kaidah Asosiasi dan Pola Sekuensial". *Tugas Akhir Sarjana Jurusan Teknik Informatika FTIF ITS Surabaya*, 2002.
- [APY-02] C.C. Aggarwal, Cecilia Procopiuc, and P.S. Yu, "Finding Localized Associations in Market Basket Data", *IEEE Transactions on Knowledge and Data Engineering*, vol. 14 no. 1, January/February 2002
- [ASR-94] R. Agrawal and R. Srikant. "Fast Algorithm for Mining Association Rules". *Proceeding of the International Conference on Very Large Data Bases*, 1994.
- [DTY-02] Daning Tyaspamadya. "Perancangan dan Pembuatan Perangkat Lunak Data Mining untuk Penggalan Kaidah Asosiasi menggunakan Metode Hybrid". *Tugas Akhir Sarjana Jurusan Teknik Informatika FTIF ITS Surabaya*, 2002.
- [GRS-98] S. Guha, R. Rastogi, and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Databases", *Proceeding ACM SIGMOD Conference.*, 1998
- [GRS-99] S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering for Categorical Attributes", *Proceeding ACM SIGMOD Conference*, 1999
- [HHE-02] Hera Hikmah Evianti. "Perancangan dan Pembuatan Perangkat Lunak Data Mining untuk Pencarian Kaidah Asosiasi dengan Metode Bottom-Up". *Tugas Akhir Sarjana Jurusan Teknik Informatika FTIF ITS Surabaya*, 2002.
- [HMK-00] Jiawei Han and Micheline Kamber, "Data Mining : Concepts and Techniques". *Intelligent Database Systems Research Lab School of Computing Science Simon Fraser University, Canada* (<http://www.cs.sfu.ca>). 2000.
- [IPD-02] I Putu Darmasusila. "Pengembangan Aplikasi Data Mining Untuk Pencarian Pola Asosiasi Dengan Metode Pattern Decomposition". *Tugas Akhir Sarjana Jurusan Teknik Informatika FTIF ITS Surabaya*. 2002.

- [JRD-98] A.Jain and R. Dubes, *Algorithms for Clustering Data*, New Jersey:Prentice Hall, 1998
- [PRA-00] Pramudyo Ananto. "Perancangan dan Pembuatan Perangkat Lunak Data Mining untuk Menemukan Aturan-Aturan Sesuai Dengan Permintaan Pengguna". *Tugas Akhir Sarjana Jurusan Teknik Informatika FTIF ITS Surabaya*, 2000.
- [TAW-03] Tony Aries Wardianto. "Perancangan dan Pembuatan Perangkat Lunak Data Mining untuk Penggalan Pola Asosiasi dengan Menggunakan Algoritma VIPER". *Tugas Akhir Sarjana Jurusan Teknik Informatika FTIF ITS Surabaya*, 2003.