

# Penentuan Pola Sekuensial pada Data Transaksi Perpustakaan IPB Menggunakan Algoritma *Graph Search Techniques*

Imas S. Sitanggang<sup>1</sup>, Firman Ardiansyah<sup>1</sup> dan Hamzah Agung<sup>2</sup>

<sup>1</sup>Staf Pengajar Departemen Ilmu Komputer, FMIPA IPB

<sup>2</sup>Mahasiswa Departemen Ilmu Komputer, FMIPA IPB

## Abstrak

Data transaksi di Perpustakaan IPB terutama peminjaman buku dicatat setiap hari sehingga menghasilkan kumpulan data transaksi peminjaman buku dalam ukuran besar. Untuk mengetahui pola perilaku peminjaman buku oleh seluruh anggota perpustakaan khususnya mahasiswa Strata 1 (S1) tahun masuk 2003 digunakan salah satu metode dalam data mining yaitu *sequential pattern mining* dengan algoritma *Graph Search Techniques (GST)* sebagai pembentuk *large sequence*. Algoritma GST dapat menemukan urutan *large k-sequence* ( $k \geq 3$ ) tanpa mengetahui langsung *large (k-1) sequence*. Pola sekuensial yang merupakan *large sequence* maksimal dicari dari seluruh *large sequence* yang terbentuk. Pola sekuensial yang diperoleh kemudian direpresentasikan agar mudah dipahami dan diinterpretasikan. Informasi yang diperoleh diharapkan dapat berguna bagi pengelola perpustakaan dalam peningkatan mutu layanan di Perpustakaan IPB.

Berdasarkan pola sekuensial yang diperoleh dalam penelitian ini maka dapat disimpulkan bahwa *minimum support* tertinggi hingga masih terbentuk *large sequence* berada pada nilai 30% dan *time constraint* 6 bulan dengan transaksi peminjaman terbanyak dilakukan oleh mahasiswa yang berasal dari Departemen Pemuliaan Tanaman dan Teknologi Benih (sebanyak 209 transaksi), sedangkan *minimum support* yang menghasilkan *sequence* dengan jumlah item terbanyak berada pada nilai 4% dan *time constraint* 2 bulan. Pada beberapa penggunaan *minimum support* ternyata *large sequence* tidak terbentuk yang diakibatkan oleh tidak ada *2-sequence* pada tabel  $L_2$  dan terjadi *cycle* pada IRG (*Item Relation Graph*). Variasi jumlah *large sequence* yang terbentuk tidak dipengaruhi oleh nilai *time constraint*. Sebagian besar transaksi yang dilakukan mahasiswa S1 tahun masuk 2003 masih dilakukan dalam skala kecil dengan sedikit jumlah item (kelas buku) yang dipinjam pada selang waktu antar transaksi peminjaman yang lama. Semakin tinggi penggunaan *minimum support*, maka jumlah item yang membentuk pola sekuensial cenderung semakin sedikit, selain itu waktu yang dibutuhkan untuk membentuk *large sequence* semakin cepat.

**Kata kunci:** *data mining, sequential pattern mining, algoritma GST, large sequence*

## PENDAHULUAN

### Latar Belakang

Data transaksi di Perpustakaan IPB terutama peminjaman buku dicatat setiap hari sehingga menghasilkan kumpulan data transaksi peminjaman buku dalam ukuran besar. Akan tetapi seringkali data tersebut hanya disimpan tanpa diolah lebih lanjut sehingga kumpulan data yang tersimpan tidak mempunyai nilai guna. Padahal jika diolah atau dianalisis lebih lanjut akan menghasilkan informasi atau pengetahuan yang penting dan berharga sebagai penunjang dalam pengambilan keputusan.

Pengolahan dan analisis yang mendalam terhadap kumpulan data transaksi peminjaman

buku menjadi suatu pengetahuan yang berharga dapat dilakukan dengan menerapkan *data mining*. *Data mining* merupakan proses ekstraksi informasi atau pola yang penting dalam basis data berukuran besar (Han & Kamber 2001). Penerapan *data mining* dalam data transaksi peminjaman buku di Perpustakaan IPB diharapkan akan mampu menggali informasi mengenai pola perilaku mahasiswa dalam meminjam buku yang dapat digunakan sebagai penunjang dalam mengambil keputusan yang berkaitan dengan peningkatan mutu layanan di Perpustakaan IPB.

Beberapa penelitian *data mining* dengan menggunakan metode *sequential pattern mining* telah menghasilkan berbagai macam algoritma,

di antaranya AprioriAll, AprioriSome, *Generalized Sequential Pattern* (GSP), dan *Graph Search Techniques* (GST). Algoritma yang digunakan pada penelitian ini adalah algoritma GST. Menurut Lin & Huang (2003), algoritma GST dapat menemukan urutan *large k-sequence* ( $k \geq 3$ ) tanpa mengetahui secara langsung *large (k - 1)-sequence*.

### Tujuan Penelitian

Tujuan dari penelitian ini adalah menerapkan metode *sequential pattern mining* terhadap data transaksi peminjaman buku di Perpustakaan IPB menggunakan algoritma GST untuk mencari *sequence* yang maksimal dari semua *frequent sequence* atau *large sequence*, yaitu *sequence* yang mempunyai *support count* lebih dari atau sama dengan *minimum support*.

### Ruang Lingkup Penelitian

Penelitian ini dibatasi pada penerapan proses *data mining* menggunakan metode *sequential pattern mining* pada data transaksi peminjaman buku di Perpustakaan IPB untuk mahasiswa Strata 1 (S1) tahun masuk 2003 mulai tanggal 1 September 2005 sampai dengan tanggal 29 Maret 2006. Pembentukan *large sequence* dilakukan menggunakan algoritma GST (*Graph Search Techniques*). Penelitian ini akan menghasilkan aplikasi *data mining* menggunakan metode *sequential pattern mining*.

### Manfaat Penelitian

Aplikasi yang dihasilkan dari penelitian ini dapat digunakan untuk mengetahui pola perilaku mahasiswa Strata 1 (S1) tahun masuk 2003 ketika meminjam buku di Perpustakaan IPB, diharapkan hasil yang diperoleh dapat berguna dalam peningkatan mutu layanan di Perpustakaan IPB.

## TINJAUAN PUSTAKA

### Basis Data

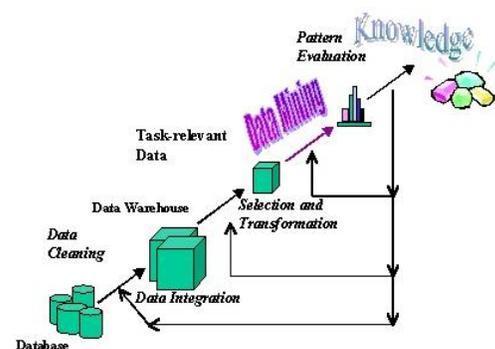
Basis data merupakan sekumpulan data/entitas (beserta deskripsinya) yang secara logika berelasi, dibuat untuk memenuhi kebutuhan informasi suatu organisasi serta dapat digunakan bersama-sama. Sistem Manajemen Basis Data (*Database Management System/DBMS*) adalah sistem perangkat lunak yang memungkinkan pengguna mendefinisikan, menciptakan, dan mengelola suatu basis data, serta menyediakan akses kontrol terhadap basis data tersebut (Connolly & Begg 2002).

### Data Mining

*Data mining* merupakan proses ekstraksi informasi atau pola yang penting dalam basis data berukuran besar (Han & Kamber 2001). *Data mining* merupakan penggunaan algoritma dalam proses *Knowledge Discovery in Database* (KDD) untuk menemukan pola yang bermanfaat (Goharian & Grossman 2003). *Data mining* juga didefinisikan sebagai suatu proses yang menggunakan berbagai perangkat analisis data untuk menemukan pola dan relasi data agar dapat digunakan untuk membuat prediksi dengan tepat (Two Crows Corporation 1999).

### Knowledge Discovery in Database (KDD)

*Knowledge Discovery in Database* (KDD) adalah proses menemukan informasi yang berguna serta pola-pola yang ada dalam data (Goharian & Grossman 2003). KDD merupakan sebuah proses yang terdiri dari serangkaian proses iteratif yang terurut, dan *data mining* merupakan salah satu langkah dalam proses KDD (Han & Kamber 2001). Tahapan-tahapan proses KDD secara berurut dapat dilihat pada Gambar 1.



Gambar 1 Tahapan dalam KDD (Han & Kamber 2001).

### Pola Sekuensial

Misal  $(i_1 i_2 i_3 \dots i_m)$  adalah *itemset* yang merupakan himpunan *item* yang tidak kosong dan  $i_k$  adalah *item*. *Sequence*  $s$  dinotasikan  $\langle s_1 s_2 s_3 \dots s_n \rangle$  di mana  $s$  adalah daftar urutan dari *itemset* dan  $s_j$  adalah *itemset* yang *itemnya* dibeli oleh *customer* pada waktu transaksi yang sama. *Item*  $i_k$  hanya dapat muncul satu kali pada  $s_j$ , akan tetapi dapat muncul lebih dari satu kali pada  $s_i$  dan  $s_j$  dengan waktu transaksi yang berbeda (Lin & Huang 2003).

*Sequence*  $s$  dikatakan maksimal jika  $s$  tidak termuat pada *sequence* lainnya. *Sequence*  $a = \langle a_1 a_2 a_3 \dots a_n \rangle$  dikatakan termuat pada *sequence* yang lain  $b = \langle b_1 b_2 b_3 \dots b_m \rangle$  jika terdapat integer  $i_1 < i_2 < \dots < i_n$ ,  $1 \leq i_k \leq m$ , sehingga  $a_1 \subseteq$

$b_{i1}, a_2 \subseteq b_{i2}, \dots, a_n \subseteq b_{in}$  (Agrawal & Srikant 1995). Dapat dikatakan juga bahwa *sequence a* merupakan *subsequence* dari *sequence b* (Lin & Lee 1998). Menurut Lin & Huang (2003), *sequence* dengan *k* item disebut *k-sequence*, dan mungkin mempunyai tiga tipe, yaitu terurut (*ordered*), tidak terurut (*non-ordered*), dan campuran (*hybrid*).

Semua transaksi dari *customer* yang sama dapat dikelompokkan bersama, kemudian diurut berdasarkan waktu transaksi secara menaik (*increasing*) menjadi *customer sequence*. *Support count* untuk *sequence s* didefinisikan sebagai total dari seluruh *customer sequence* yang memuat *s*. Suatu *sequence* dapat dikatakan sebagai *frequent sequence* atau *large sequence* jika memiliki *support count* lebih dari atau sama dengan *minimum support* yang telah ditentukan pengguna (Lin & Lee 1998).

Diberikan basis data transaksi *D* dan ambang batas *minimum support*  $\epsilon$ , dapat didefinisikan masalah dari *sequential pattern mining* adalah mencari *large sequence* yang maksimal di antara semua *sequence* yang mempunyai *support count* lebih besar atau sama dengan  $\epsilon$ . Setiap *large sequence* yang ditemukan merepresentasikan suatu pola sekuensial. Sebagai tambahan, *time constraint* juga akan dipertimbangkan ketika mencari pola sekuensial, dan hal ini akan membuat pola sekuensial yang ditemukan lebih efektif dan berguna (Lin & Huang 2003).

### Algoritma GST (*Graph Search Techniques*)

Algoritma GST (*Graph Search Techniques*) pertama kali diperkenalkan oleh Lin dan Huang (2003). Algoritma GST dapat menemukan *large k-sequence* ( $k \geq 3$ ) tanpa harus mengetahui *large (k-1)-sequence*. Dalam melakukan proses *mining* untuk mendapatkan pola sekuensial, algoritma GST mempunyai empat tahap yaitu (Lin & Huang 2003):

1. Menelusuri basis data untuk membuat *large 1-sequence* ( $L_1$ )

Untuk setiap *1-sequence* dalam tabel  $C_1$  akan dihitung *support count*-nya. Jika nilai *support count*-nya lebih dari atau sama dengan *minimum support* maka akan disisipkan ke dalam tabel  $L_1$ , selainnya akan dibuang. Struktur dari  $C_1$  dan  $L_1$  dapat dilihat pada Gambar 2, di mana *CID* adalah *customer-id*, *Time* adalah waktu transaksi, dan *Item* adalah *item* yang dibeli.

Item	CID	Time
------	-----	------

Gambar 2 Struktur  $C_1$  dan  $L_1$ .

2. Menggabungkan  $L_1$  dengan dirinya sendiri untuk membuat *large 2-sequence* ( $L_2$ )

Langkah selanjutnya adalah menggabungkan  $L_1$  dengan dirinya sendiri berdasarkan *customer-id* dan waktu transaksi untuk membuat  $L_2$ . Penggabungan ini dapat diklasifikasikan menjadi dua jenis. Jenis pertama dinotasikan  $\langle(AB)\rangle$  yang berarti bahwa *item A* dan *item B* dibeli oleh *CID* yang sama pada waktu yang sama pula. Jenis kedua dinotasikan  $\langle(A)B\rangle$  yang berarti bahwa *item A* dan *item B* dibeli oleh *CID* yang sama, tetapi *item A* terjadi sebelum *item B*.

*Support count* dari masing-masing *2-sequence* dihitung, kemudian membuang *2-sequence* yang mempunyai *support count* kurang dari *minimum support*. Struktur dari  $L_2$  dapat dilihat pada Gambar 3, di mana *Sequence* menunjukkan *2-sequence* yang memenuhi *minimum support*, *CID* adalah *customer-id*, *STime* adalah waktu dari *item* yang pertama kali muncul pada *Sequence*, dan *ETime* adalah waktu dari *item* yang terakhir kali muncul pada *Sequence*.

Sequence	CID	STime	ETime
----------	-----	-------	-------

Gambar 3 Struktur  $L_2$ .

Pada tahap ini, batasan waktu juga akan diperiksa dengan memangkas *sequence* yang mempunyai interval waktu yang lama. Prosedur pembuatan  $L_2$  dalam bentuk kalimat SQL (*Structured Query Language*) adalah sebagai berikut (Lin & Huang 2003):

```
(1) INSERT INTO L2
(2) SELECT (p.Item, q.Item), p.CID,
p.Time, q.Time /* <(AB)> */
(3) FROM L1 p, L1 q
(4) WHERE p.CID = q.CID
AND p.Time = q.Time
AND p.Item < q.Item /* sort
the items at the same time */
(5) UNION
(6) SELECT ((p.Item), (q.Item)),
p.CID, p.Time, q.Time /*
<(A)(B)> */
(7) FROM L1 p, L1 q
(8) WHERE p.CID = q.CID
AND p.Time < q.Time
AND (min-interval ≤ q.Time -
p.Time ≤ max-interval) /* time
constraints check */
```

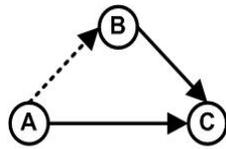
Selanjutnya, membuang semua *large 2-sequence* yang mempunyai *support count* kurang dari *minimum support*. Prosedurnya dalam kalimat SQL adalah sebagai berikut (Lin & Huang 2003):

```
(1) DELETE *
```

- (2) FROM  $L_2$
- (3) WHERE *Sequence*  
NOT IN (SELECT *Sequence*  
FROM  $L_2$   
GROUP BY *Sequence*  
HAVING COUNT(\*) > the minimum  
support)

### 3. Mengkonstruksi IRG (*Item Relation Graph*)

IRG (*Item Relation Graph*) merupakan suatu graf yang menggambarkan hubungan relasi antar *item*. *Node* pada IRG adalah *item* yang muncul pada *2-sequence* di dalam tabel  $L_2$ . Jenis *edge* pada IRG merepresentasikan hubungan (*relationship*) di antara *item*. Terdapat 2 jenis *edge* pada IRG. Jenis pertama, jika *2-sequence* adalah  $\langle(A)B\rangle$ , maka akan digambarkan seperti  $A \rightarrow B$ . Jenis kedua, jika *2-sequence* adalah  $\langle(AB)\rangle$ , maka akan digambarkan seperti  $A \dashrightarrow B$ . *Edge* akan diberi label (*CID*, *STime*, *ETime*). IRG untuk *sequence*  $\langle(AB)\rangle$ ,  $\langle(A)C\rangle$ , dan  $\langle(B)C\rangle$  dapat dilihat pada Gambar 4.



Gambar 4 IRG untuk *sequence*  $\langle(AB)\rangle$ ,  $\langle(A)C\rangle$ , dan  $\langle(B)C\rangle$ .

### 4. Menelusuri IRG untuk mencari *large sequence*

Untuk setiap *vertex (node)* dalam IRG, akan dibentuk semua *large sequence* di mana *rootnya* dimulai dari *vertex* tersebut. Jadi, setiap *path* adalah *large n-sequence* jika *n vertex* telah sukses dikunjungi.

## METODE PENELITIAN

Data yang akan di-*mining* pada penelitian ini adalah data transaksi peminjaman (sirkulasi) buku di Perpustakaan IPB. Data tersebut diperoleh dalam format CDS/ISIS (circ.mst) yang terdiri atas 57.839 *record* dan 14 *field*. Sebelum di-*mining*, data harus melewati tahap *praproses (preprocessing)* terlebih dahulu yang meliputi integrasi data, transformasi data, seleksi data, dan pembersihan data. Hal ini dilakukan agar data benar-benar lengkap, valid, dan sesuai dengan *input* yang dibutuhkan oleh algoritma.

### Proses Dasar Sistem

Proses dasar sistem mengacu pada proses dalam *Knowledge Discovery in Database (KDD)* yang meliputi:

#### 1. Integrasi data

Merupakan tahap di mana koleksi data yang berasal dari berbagai sumber digabungkan. Pada penelitian ini, integrasi data tidak akan dilakukan karena data hanya berasal dari satu sumber, yaitu Perpustakaan IPB.

#### 2. Transformasi data

Merupakan tahap untuk mengubah koleksi data ke dalam format yang sesuai agar dapat di-*mining*. Koleksi data tersebut dikonversi dari format CDS/ISIS (circ.mst) menjadi format Paradox (C1.db).

#### 3. Seleksi data

Merupakan tahap untuk memutuskan data mana yang akan digunakan pada tahap *data mining*. Selain itu, tahap seleksi data juga akan memilih *field* atau atribut data yang dibutuhkan sebagai *input* untuk algoritma yang digunakan pada tahap *data mining*.

#### 4. Pembersihan data

Merupakan tahap untuk membersihkan data yang mengandung *noise*, data yang mempunyai nilai kosong, dan data yang tidak konsisten untuk dibuang dari koleksi data. Data tersebut dibuang dan tidak diikuti pada tahap selanjutnya. Dari tahap ini akan dihasilkan koleksi data yang lengkap dan valid.

#### 5. *Data mining*

Merupakan tahap yang paling utama karena koleksi data yang telah diolah pada tahap sebelumnya akan di-*mining* menggunakan metode *sequential pattern mining* dengan menerapkan algoritma GST (*Graph Search Techniques*). Dalam tahap ini, prosedur yang telah ditetapkan pada algoritma GST diaplikasikan untuk mendapatkan semua kemungkinan *large sequence*.

#### 6. Evaluasi pola

Merupakan tahap untuk mendapatkan pola sekuensial, yaitu mencari *large sequence* yang maksimal dari semua *large sequence* yang diperoleh setelah proses *mining* selesai.

#### 7. Presentasi pengetahuan

Pola sekuensial yang telah ditemukan kemudian direpresentasikan kepada pengguna agar mudah dipahami dan diinterpretasikan. Teknik yang digunakan untuk merepresentasikan pola sekuensial adalah dalam bentuk aturan logika.

### Spesifikasi Implementasi Sistem

Implementasi sistem menggunakan komputer personal yang memiliki perangkat keras dan perangkat lunak sebagai berikut:

Perangkat keras:

1. *Processor* Intel Pentium Mobile 740 (1,7 GHz) Centrino.
2. Memori 768 MB DDR II.
3. *Harddisk* 60 GB.
4. Monitor 12" XGA dengan resolusi 1024 × 768 *pixel*.
5. *Mouse* dan *keyboard*.

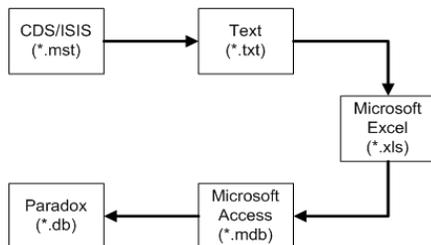
Perangkat lunak:

1. Microsoft Windows XP Professional sebagai sistem operasi.
2. Microsoft Excel 2003 sebagai lembar kerja (*worksheet*) dalam pengolahan data.
3. Microsoft Access 2003 sebagai *data converter* ke format Paradox.
4. Borland C++Builder 6 sebagai editor pemrograman.
5. CDS/ISIS for Windows sebagai *data converter* ke format *text*.
6. Paradox 7 sebagai sistem manajemen basis data (DBMS).

## HASIL DAN PEMBAHASAN

### Transformasi Data

Untuk dapat di-*mining*, koleksi data yang diperoleh dalam format CDS/ISIS (*circ.mst*) dikonversi ke dalam format Paradox (*C1.db*). Urutan konversi data dari format CDS/ISIS menjadi format Paradox dapat dilihat pada Gambar 5. Alasan penggunaan Paradox sebagai DBMS adalah kesederhanaan dan kemudahan konfigurasi ketika diintegrasikan dengan Borland C++Builder bila dibandingkan dengan penggunaan Microsoft Access sebagai DBMS. Tahap seleksi data dan pembersihan data juga dilakukan bersama pada tahap transformasi data. Hasil yang diperoleh dari tahap ini adalah sebuah basis data dalam format Paradox yang terdiri dari satu tabel, yaitu tabel *C1*.



Gambar 5 Urutan konversi data.

### Seleksi Data

Koleksi data transaksi peminjaman buku yang tersedia kemudian diseleksi untuk mendapatkan data yang akan di-*mining*.

Prosedur yang dilakukan pada tahap seleksi data adalah:

1. Memilih tiga *field* dari 14 *field* yang tersedia.

Prosedur ini diimplementasikan ketika data akan dikonversi dari format CDS/ISIS ke format *text*. *Input* data yang dibutuhkan oleh algoritma *Graph Search Techniques* (GST) hanya tiga *field*, yaitu *field* nomor identitas peminjam, *field* kelas buku yang dipinjam, dan *field* tanggal peminjaman. *Field* kelas buku memiliki tiga sub-*field*, yaitu kelas buku K, kelas buku L, dan kelas buku M. Ketiga sub-*field* ini menunjukkan tingkatan kelas buku berdasarkan klasifikasi UDC (*Universal Decimal Codes*).

2. Memilih data transaksi yang terjadi pada periode 1 September 2005 sampai dengan 29 Maret 2006.

Data transaksi yang sudah dalam format *text* kemudian dikonversi menjadi format Microsoft Excel. Hasil konversi ini menghasilkan data transaksi peminjaman buku dengan lima *field*, yaitu *field* nomor identitas peminjam, *field* kelas buku K, *field* kelas buku L, *field* kelas buku M, dan *field* tanggal peminjaman.

Dari 57.839 *record*, diambil sebanyak 54.904 *record* yang berada pada periode transaksi 1 September 2005 sampai dengan 29 Maret 2006. Pemilihan ini didasarkan melalui pengamatan bahwa *record* data transaksi yang terjadi di luar periode tersebut banyak yang hilang ketika proses *backup* yang dilakukan oleh petugas Perpustakaan IPB.

3. Memilih data transaksi yang dilakukan oleh mahasiswa Strata 1 (S1) tahun masuk 2003.

Pada tahap ini, dipilih nomor identitas peminjam yang berupa xx4x03xxx. Angka 4 pada *digit* ketiga menunjukkan S1, sedangkan angka 03 pada *digit* kelima dan keenam menunjukkan tahun masuk ke IPB, yaitu 2003. Dari 54.904 *record*, diambil sebanyak 11.472 *record* yang merupakan *record* transaksi peminjaman buku yang dilakukan oleh mahasiswa S1 tahun masuk 2003.

4. Memilih kelas buku M untuk dijadikan *item*.

Kelas buku M merupakan tingkatan kelas buku yang ketiga berdasarkan klasifikasi UDC. Pemilihan ini dilakukan karena kelas buku M memiliki jenis yang lebih banyak daripada kelas buku K dan kelas buku L, sehingga kombinasi dari *item* yang dipinjam menjadi lebih beragam. Kemudian mengganti nilai *field* pada kelas buku M agar bertipe numerik, misalnya nilai mKLS=639 diganti menjadi 639.

Hasil yang diperoleh adalah data transaksi dengan tiga *field*, yaitu *field* kelas buku M, *field* nomor identitas peminjam, dan *field* tanggal peminjaman. Kemudian nama *field* tersebut diganti, kelas buku M diganti menjadi *Item*, nomor identitas peminjam diganti menjadi *CID*, dan tanggal peminjaman diganti menjadi *Time*.

### Pembersihan Data

Data transaksi yang telah diseleksi kemudian dibersihkan untuk membuang *record* yang mempunyai nilai *invalid*. Dari pengamatan diperoleh:

1. 488 *record* yang mempunyai nilai *invalid* dan nilai NULL pada *field* *Item*.
2. 344 *record* yang mempunyai nilai *invalid* pada *field* *CID*.

Selanjutnya seluruh *record* tersebut (sebanyak 832 *record*) dibuang agar tidak ikut pada tahap selanjutnya, sehingga jumlah *record* yang tersisa sebanyak 10.640 *record*.

Untuk menghindari adanya *record* yang sama, maka dilakukan *query* terhadap data transaksi peminjaman buku untuk mengambil satu *record* saja terhadap beberapa *record* yang sama. Dari 10.640 *record*, tersisa sebanyak 7.799 *record* yang berbeda dengan komposisi 179 jenis *Item*, 1.037 jenis *CID*, dan 127 jenis *Time*.

Data yang tersisa sebanyak 7.799 *record* ini merupakan data yang sudah valid dan sesuai dengan input yang dibutuhkan algoritma GST. Data transaksi tersebut kemudian dikonversi dari format Microsoft Excel menjadi format Microsoft Access, untuk selanjutnya dikonversi menjadi format Paradox.

### Data Mining

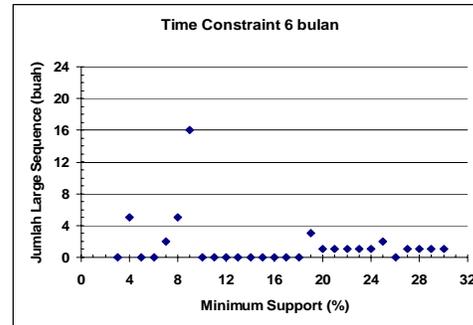
Pada tahap ini, metode *sequential pattern mining* diterapkan untuk membentuk semua kemungkinan *large sequence* dengan menggunakan algoritma GST. Ketika tahap keempat dari algoritma GST dilakukan, pemilihan *root* dimulai dari *node* (*vertex*) yang mempunyai nilai *item* terkecil dan *large n-sequence* yang terbentuk merupakan hasil penelusuran hingga ditemui *node* terakhir yang tidak mempunyai *arc* (*edge* berarah) yang keluar dari *node* tersebut.

Percobaan dilakukan pada data transaksi peminjaman buku yang telah melewati tahap praproses (7.799 *record*) dengan *minimum support* dari 3% sampai 30%, sehingga jumlah iterasi yang terjadi sebanyak 28 kali. *Time constraint* yang digunakan adalah 6 bulan, 4 bulan, 2 bulan, dan 20 hari. Pada penelitian ini,

diasumsikan bahwa jumlah hari dalam 1 bulan sama dengan 30 hari.

#### a. Pembentukan *large sequence* dengan *time constraint* 6 bulan

*Time constraint* 6 bulan menunjukkan bahwa selang waktu maksimal antar transaksi peminjaman buku yang dilakukan oleh seorang mahasiswa adalah 6 bulan (180 hari). Grafik yang menunjukkan jumlah *large sequence* yang terbentuk dapat dilihat pada Gambar 6.



Gambar 6 Grafik jumlah *large sequence* yang terbentuk dengan *time constraint* 6 bulan.

Dari grafik pada Gambar 6, jumlah maksimal *large sequence* yang terbentuk adalah 16 buah pada posisi *minimum support* 9%. Sedangkan nilai *minimum support* tertinggi hingga masih terdapat *large sequence* yang terbentuk adalah 30% dengan jumlah *large sequence* sebanyak 1 buah.

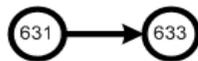
Pada beberapa nilai *minimum support* yang digunakan pada saat percobaan, ternyata *large sequence* tidak terbentuk. Hal ini disebabkan oleh tidak ada *2-sequence* pada tabel  $L_2$  atau terjadi *cycle* pada IRG (*Item Relation Graph*). Tidak terbentuknya *2-sequence* pada tabel  $L_2$  disebabkan oleh semua *2-sequence* hasil penggabungan dari kombinasi *1-sequence* mempunyai *support count* kurang dari *minimum support* atau selang waktu antar transaksi peminjaman melebihi *time constraint* yang ditetapkan.

*Cycle* pada IRG terjadi karena hanya sedikit kombinasi *item* (jenis buku) yang dipinjam oleh mahasiswa, sehingga jenis buku yang telah dipinjam pada transaksi sebelumnya seringkali dipinjam kembali. Hal ini wajar terjadi karena peminjaman buku di perpustakaan mempunyai batas waktu, sehingga jika waktu peminjaman sudah habis, mahasiswa dimungkinkan akan memperpanjang status peminjaman atau meminjam kembali pada waktu dan transaksi yang berbeda.

Tabel 1 Daftar nilai *minimum support* di mana tidak terbentuk *large sequence* pada *time constraint* 6 bulan

No	Minimum Support	Keterangan
1	3%	terjadi cycle
2	5%	terjadi cycle
3	6%	terjadi cycle
4	10%	terjadi cycle
5	11%	terjadi cycle
6	12%	terjadi cycle
7	13%	terjadi cycle
8	14%	terjadi cycle
9	15%	terjadi cycle
10	16%	terjadi cycle
11	17%	terjadi cycle
12	18%	terjadi cycle
13	26%	terjadi cycle

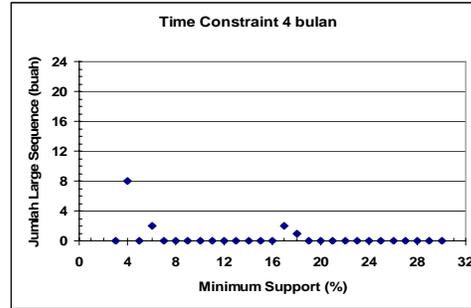
Berdasarkan Tabel 1, *large sequence* tidak terbentuk pada 13 nilai *minimum support* yang digunakan pada saat percobaan. Hal ini disebabkan oleh *cycle* yang terjadi pada IRG. Dengan menggunakan *time constraint* 6 bulan, ternyata maksimal *item* yang dapat dibentuk pada sebuah *sequence* adalah 6 buah (6-*sequence*). Hal ini menunjukkan bahwa terdapat paling banyak enam jenis buku yang pernah dipinjam mahasiswa secara sekuensial dalam selang waktu antar transaksi peminjaman paling lama 6 bulan (180 hari). Contoh 6-*sequence* yang terbentuk adalah  $\langle(632)(631)(519)(658)(665)(639)\rangle$ , sedangkan contoh IRG yang membentuk *large sequence*  $\langle(631)(633)\rangle$  pada nilai *minimum support* 30% dapat dilihat pada Gambar 7.



Gambar 7 IRG yang membentuk *large sequence*  $\langle(631)(633)\rangle$  pada nilai *minimum support* 30%.

**b. Pembentukan *large sequence* dengan *time constraint* 4 bulan**

Grafik yang menunjukkan jumlah *large sequence* yang terbentuk dapat dilihat pada Gambar 8. Dari grafik pada Gambar 8, jumlah maksimal *large sequence* yang terbentuk adalah 8 buah pada posisi *minimum support* 4%. Sedangkan nilai *minimum support* tertinggi hingga masih terdapat *large sequence* yang terbentuk adalah 18% dengan jumlah *large sequence* sebanyak 1 buah.



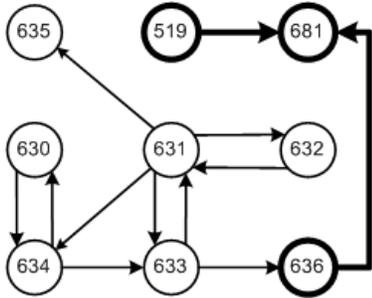
Gambar 8 Grafik jumlah *large sequence* yang terbentuk dengan *time constraint* 4 bulan.

Tabel 2 Daftar nilai *minimum support* di mana tidak terbentuk *large sequence* pada *time constraint* 4 bulan

No	Minimum Support	Keterangan
1	3%	terjadi cycle
2	5%	terjadi cycle
3	7%	terjadi cycle
4	8%	terjadi cycle
5	9%	terjadi cycle
6	10%	terjadi cycle
7	11%	terjadi cycle
8	12%	terjadi cycle
9	13%	terjadi cycle
10	14%	terjadi cycle
11	15%	terjadi cycle
12	16%	terjadi cycle
13	19%	terjadi cycle
14	20%	terjadi cycle
15	21%	terjadi cycle
16	22%	terjadi cycle
17	23%	terjadi cycle
18	24%	terjadi cycle
19	25%	terjadi cycle
20	26%	tabel $L_2$ kosong
21	27%	tabel $L_2$ kosong
22	28%	tabel $L_2$ kosong
23	29%	tabel $L_2$ kosong
24	30%	tabel $L_2$ kosong

Berdasarkan Tabel 2, tabel  $L_2$  kosong terjadi pada lima nilai *minimum support*, sedangkan *cycle* terjadi pada 19 nilai *minimum support*. Dengan menggunakan *time constraint* 4 bulan, ternyata maksimal *item* yang dapat dibentuk pada sebuah *sequence* adalah 5 (5-*sequence*). Contoh 5-*sequence* yang terbentuk adalah  $\langle(574)(551)(331)(301)(316)\rangle$ , sedangkan

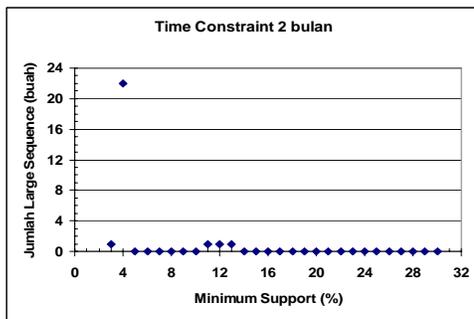
contoh IRG yang membentuk *large sequence*  $\langle(519)(681)\rangle$  dan  $\langle(636)(681)\rangle$  pada nilai *minimum support* 17% dapat dilihat pada Gambar 9.



Gambar 9 IRG yang membentuk *large sequence*  $\langle(519)(681)\rangle$  dan  $\langle(636)(681)\rangle$  pada nilai *minimum support* 17%.

**c. Pembentukan *large sequence* dengan *time constraint* 2 bulan**

Grafik yang menunjukkan jumlah *large sequence* yang terbentuk dapat dilihat pada Gambar 10. Dari grafik pada Gambar 10, jumlah maksimal *large sequence* yang terbentuk adalah 22 buah pada posisi *minimum support* 4%. Sedangkan nilai *minimum support* tertinggi hingga masih terdapat *large sequence* yang terbentuk adalah 13% dengan jumlah *large sequence* sebanyak 1 buah.

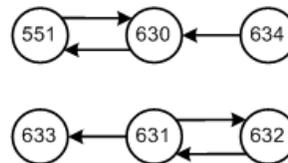


Gambar 10 Grafik jumlah *large sequence* yang terbentuk dengan *time constraint* 2 bulan.

Berdasarkan Tabel 3, tabel  $L_2$  kosong terjadi pada empat nilai *minimum support*, sedangkan *cycle* terjadi pada 19 nilai *minimum support*. Dengan menggunakan *time constraint* 2 bulan, ternyata maksimal *item* yang dapat dibentuk pada sebuah *sequence* adalah 8 (*8-sequence*). Contoh *8-sequence* yang terbentuk adalah  $\langle(665)(681)(519)(633)(630)(331)(301)(316)\rangle$ , sedangkan contoh IRG yang mengakibatkan tidak terbentuknya *large sequence* pada nilai *minimum support* 16% dapat dilihat pada Gambar 11.

Tabel 3 Daftar nilai *minimum support* di mana tidak terbentuk *large sequence* pada *time constraint* 2 bulan

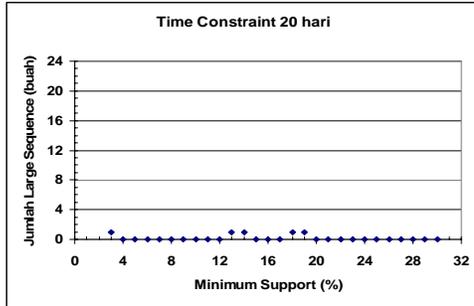
No	Minimum Support	Keterangan
1	3%	terjadi cycle
2	4%	terjadi cycle
3	5%	terjadi cycle
4	6%	terjadi cycle
5	7%	terjadi cycle
6	8%	terjadi cycle
7	14%	terjadi cycle
8	15%	terjadi cycle
9	16%	terjadi cycle
10	17%	terjadi cycle
11	18%	terjadi cycle
12	19%	terjadi cycle
13	20%	terjadi cycle
14	21%	terjadi cycle
15	22%	terjadi cycle
16	23%	terjadi cycle
17	24%	terjadi cycle
18	25%	terjadi cycle
19	26%	terjadi cycle
20	27%	tabel $L_2$ kosong
21	28%	tabel $L_2$ kosong
22	29%	tabel $L_2$ kosong
23	30%	tabel $L_2$ kosong



Gambar 11 IRG di mana tidak terbentuk *large sequence* pada nilai *minimum support* 16%.

**d. Pembentukan *large sequence* dengan *time constraint* 20 hari**

Grafik yang menunjukkan jumlah *large sequence* yang terbentuk dapat dilihat pada Gambar 12. Dari grafik pada Gambar 12, jumlah maksimal *large sequence* yang terbentuk adalah 1 buah pada posisi *minimum support* 3%, 13%, 14%, 18%, dan 19%. Sedangkan nilai *minimum support* tertinggi hingga masih terdapat *large sequence* yang terbentuk adalah 19% dengan jumlah *large sequence* sebanyak 1 buah.



Gambar 12 Grafik jumlah *large sequence* yang terbentuk dengan *time constraint* 20 hari.

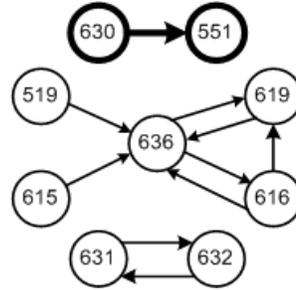
Tabel 4 Daftar nilai *minimum support* di mana tidak terbentuk *large sequence* pada *time constraint* 20 hari

No	Minimum Support	Keterangan
1	4%	terjadi cycle
2	5%	terjadi cycle
3	6%	terjadi cycle
4	7%	terjadi cycle
5	8%	terjadi cycle
6	9%	terjadi cycle
7	10%	terjadi cycle
8	11%	terjadi cycle
9	12%	terjadi cycle
10	15%	terjadi cycle
11	16%	terjadi cycle
12	17%	terjadi cycle
13	20%	tabel $L_2$ kosong
14	21%	tabel $L_2$ kosong
15	22%	tabel $L_2$ kosong
16	23%	tabel $L_2$ kosong
17	24%	tabel $L_2$ kosong
18	25%	tabel $L_2$ kosong
19	26%	tabel $L_2$ kosong
20	27%	tabel $L_2$ kosong
21	28%	tabel $L_2$ kosong
22	29%	tabel $L_2$ kosong
23	30%	tabel $L_2$ kosong

Tabel 5 Ringkasan hasil percobaan

Time Constraint	Large Sequence		Minimum Support		Maksimal Item pada Sequence
	Jumlah Maksimal	Minimum Support	Nilai Tertinggi	Jumlah Large Sequence	
6 bulan	16 buah	9%	30%	1 buah	6-sequence
4 bulan	8 buah	4%	18%	1 buah	5-sequence
2 bulan	22 buah	4%	13%	1 buah	8-sequence
20 hari	1 buah	3%, 13%, 14%, 18%, 19%	19%	1 buah	2-sequence

Berdasarkan Tabel 4, tabel  $L_2$  kosong terjadi pada sebelas nilai *minimum support*, sedangkan *cycle* terjadi pada 12 nilai *minimum support*. Dengan menggunakan *time constraint* 20 hari, ternyata maksimal *item* yang dapat dibentuk pada sebuah *sequence* adalah 2 (2-*sequence*). Contoh 2-*sequence* yang terbentuk adalah  $\langle(519)(658)\rangle$ , sedangkan contoh IRG yang membentuk *large sequence*  $\langle(630)(551)\rangle$  pada nilai *minimum support* 14% dapat dilihat pada Gambar 13.

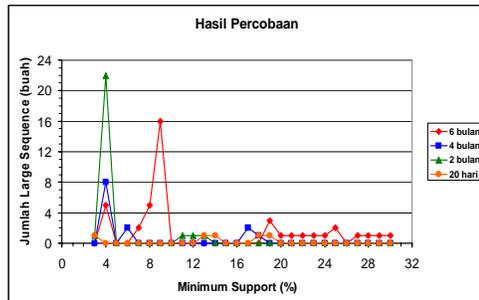


Gambar 13 IRG yang membentuk *large sequence*  $\langle(630)(551)\rangle$  pada nilai *minimum support* 14%.

Berdasarkan ringkasan hasil percobaan yang terdapat pada Tabel 5, diperoleh informasi sebagai berikut:

- Jumlah maksimal *large sequence* yang terbentuk adalah 22 buah yang terjadi pada *minimum support* 4% dan *time constraint* 2 bulan. Nilai *minimum support* 3%, 4%, 9%, 13%, 14%, 18% dan 19% memiliki jumlah *large sequence* yang maksimal pada masing-masing *time constraint* yang ditetapkan. Hal ini menunjukkan bahwa pada *minimum support* tersebut, jumlah *cycle* yang terjadi pada IRG lebih sedikit dibandingkan dengan penggunaan nilai *minimum support* yang lain pada *time constraint* yang sama.
- Nilai *minimum support* tertinggi hingga masih terbentuk *large sequence* adalah 30% pada *time constraint* 6 bulan dengan jumlah *large sequence* yang terbentuk sebanyak 1 buah. Hal ini menunjukkan bahwa untuk penggunaan *minimum support* di atas 30% tidak akan dihasilkan *large sequence* untuk sembarang *time constraint* yang ditetapkan.

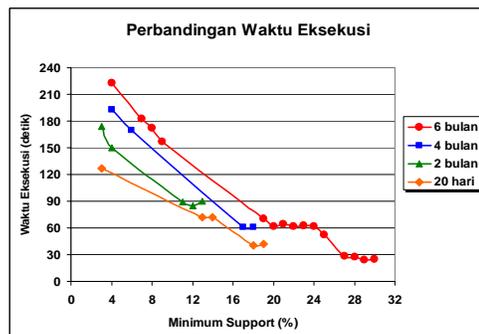
- Jumlah maksimal *item* yang terbentuk pada sebuah *sequence* adalah 8 buah (*8-sequence*). Hal ini menunjukkan bahwa jumlah peminjaman *item* (jenis buku) yang dilakukan oleh mahasiswa adalah paling banyak 8 *item*.



Gambar 14 Grafik jumlah *large sequence* yang terbentuk untuk seluruh *time constraint*.

Dari grafik pada Gambar 14 yang menunjukkan perbandingan jumlah *large sequence* yang terbentuk untuk berbagai *minimum support* pada seluruh *time constraint*, diperoleh informasi bahwa ternyata nilai *time constraint* tidak mempengaruhi jumlah *large sequence* yang terbentuk. Variasi jumlah *large sequence* yang terbentuk lebih dipengaruhi oleh variasi penggunaan nilai *minimum support*.

Waktu eksekusi yang diperlukan untuk membentuk *large sequence* sangat dipengaruhi oleh *minimum support* dan *time constraint* yang digunakan. Perbandingan waktu eksekusi yang diperlukan untuk membentuk *large sequence* dari seluruh percobaan yang telah dilakukan dapat dilihat pada Gambar 15.



Gambar 15 Grafik perbandingan waktu yang dibutuhkan untuk membentuk *large sequence*.

Menurut grafik pada Gambar 15, semakin tinggi nilai *minimum support* yang digunakan, maka semakin cepat waktu yang diperlukan untuk membentuk *large sequence*. Hal ini

dikarenakan semakin sedikit *sequence* yang mendukung terbentuknya *large sequence* sehingga proses pembentukan *large sequence* semakin cepat. Jumlah *sequence* yang sedikit diakibatkan oleh banyaknya *sequence* yang *support count*-nya tidak memenuhi nilai *minimum support* yang digunakan.

Akan tetapi, semakin lama *time constraint* yang digunakan, maka semakin lama waktu yang diperlukan untuk membentuk *large sequence* pada penggunaan *minimum support* yang sama. Hal ini dikarenakan semakin banyak *sequence* yang mendukung terbentuknya *large sequence* sehingga proses pembentukan *large sequence* semakin lama. Jumlah *sequence* yang banyak diakibatkan oleh banyaknya *sequence* yang selisih waktu antar transaksinya memenuhi *time constraint* yang digunakan.

### Evaluasi Pola

Seluruh *large sequence* yang dihasilkan pada tahap *data mining* kemudian dievaluasi untuk mendapatkan pola sekuensial. Evaluasi dilakukan dengan mencari *large sequence* yang maksimal dari seluruh *large sequence* yang ada. Suatu *sequence* dikatakan maksimal jika *sequence* tersebut tidak termuat pada *sequence* lainnya.

Dari seluruh pola sekuensial yang terbentuk, diambil pola sekuensial yang mempunyai *minimum support* tertinggi dari setiap jenis polanya. Hal ini berarti bahwa dari beberapa pola sekuensial yang sama, diambil satu pola sekuensial saja yang terjadi pada nilai *minimum support* tertinggi.

Setiap pola sekuensial yang terjadi pada *minimum support* yang tinggi merupakan pola yang menarik. Daftar pola sekuensial yang menarik dari seluruh percobaan yang telah dilakukan terhadap data transaksi peminjaman buku di Perpustakaan IPB dapat dilihat pada Tabel 6. Dari daftar pola sekuensial yang terdapat pada Tabel 6, diperoleh informasi sebagai berikut:

- Semakin tinggi nilai *minimum support*, maka jumlah *item* yang membentuk pola sekuensial cenderung semakin sedikit.
- Nilai *minimum support* yang menghasilkan *sequence* dengan *item* terbanyak adalah 4% dengan jumlah *item* yang terbentuk adalah 8 (*8-sequence*).
- Transaksi peminjaman buku yang dilakukan oleh mahasiswa S1 tahun masuk 2003 sebagian besar terjadi dalam skala kecil dengan sedikit jumlah *item* yang dipinjam.

Tabel 6 Daftar pola sekuensial yang menarik

Pola Sekuensial	Minimum Support Tertinggi
<(631)(633)>	30%
<(519)(658)>	25%
<(633)(636)>	25%
<(519)(681)>	19%
<(636)(681)>	19%
<(632)(631)>	18%
<(630)(551)>	14%
<(634)(633)>	13%
<(632)(631)(519)(658)(338)>	9%
<(632)(631)(519)(658)(665)(639)>	9%
<(633)(631)(519)(658)(338)>	9%
<(633)(631)(519)(658)(665)(639)>	9%
<(635)(631)(519)(658)(338)>	9%
<(635)(631)(519)(658)(665)(639)>	9%
<(664)(613)>	9%
<(681)(519)(658)(338)>	9%
<(681)(519)(658)(665)(639)>	9%
<(658)(338)>	8%
<(658)(665)(639)>	8%
<(664)(613)>	8%
<(330)(338)>	8%
<(665)(639)>	7%
<(615)(613)>	6%
<(633)(519)(334)>	4%
<(665)(519)(338)>	4%
<(681)(519)(334)>	4%
<(556)(551)(331)(301)(316)>	4%
<(574)(551)(331)(301)(316)>	4%
<(633)(630)(331)(301)(316)>	4%
<(634)(630)(331)(301)(316)>	4%
<(556)(551)(331)(338)(330)>	4%
<(574)(551)(331)(338)(330)>	4%
<(634)(630)(331)(338)(330)>	4%
<(665)(681)(519)(633)(630)(331)(301)(316)>	4%
<(665)(681)(519)(633)(630)(331)(338)(330)>	4%
<(338)(330)>	3%

### Presentasi Pengetahuan

Presentasi pengetahuan dari pola sekuensial yang diperoleh sangat diperlukan agar pola yang ada mudah untuk dipahami dan diinterpretasikan. Hasil presentasi pengetahuan yang dihasilkan adalah dalam bentuk aturan logika, diharapkan dapat diperoleh pengetahuan yang berharga dari koleksi data yang telah *dimining*. Pada penelitian ini, presentasi pengetahuan hanya dilakukan pada pola sekuensial yang terjadi dengan nilai *minimum*

*support* 30% dan pada pola sekuensial yang memiliki jumlah *item* terbanyak (8-*sequence*).

Pola sekuensial yang terjadi pada *minimum support* 30% adalah <(631)(633)> dengan *time constraint* 6 bulan. Pola sekuensial tersebut mempunyai arti bahwa sedikitnya sekitar 312 transaksi dilakukan oleh mahasiswa S1 tahun masuk 2003 untuk meminjam buku pada kelas 631 (teknik pertanian) terlebih dahulu kemudian diikuti meminjam buku pada kelas 633 (tanaman ladang dan perkebunan) pada waktu transaksi yang berbeda dalam selang waktu maksimal 6 bulan (180 hari). Distribusi departemen dari mahasiswa yang meminjam buku dengan pola sekuensial <(631)(633)> dapat dilihat pada Tabel 7.

Tabel 7 Distribusi departemen dari mahasiswa yang meminjam buku dengan pola sekuensial <(631)(633)>

No	Kode CID	Nama Departemen	Jumlah Transaksi
1	A141	Manajemen Agribisnis	1 buah
2	A143	Ekonomi Pertanian dan Sumberdaya	6 buah
3	A241	Ilmu Tanah	2 buah
4	A341	Agronomi	26 buah
5	A343	Hortikultura	12 buah
6	A344	Pemuliaan Tanaman dan Teknologi Benih	209 buah
7	A441	Hama dan Penyakit Tumbuhan	53 buah
8	A541	Gizi Masyarakat dan Sumberdaya Keluarga	1 buah
9	F341	Teknologi Industri Pertanian	1 buah
10	G241	Meteorologi	14 buah
11	G541	Matematika	2 buah

Berdasarkan distribusi departemen dari mahasiswa yang meminjam buku dengan pola sekuensial <(631)(633)> pada Tabel 7, diperoleh informasi sebagai berikut:

- Mahasiswa yang meminjam buku dengan pola sekuensial <(631)(633)> berasal dari sebelas departemen yang mayoritas berada pada Fakultas Pertanian.
- Jumlah transaksi terbanyak dari peminjaman buku yang mendukung pola sekuensial <(631)(633)> berada pada Departemen

Pemuliaan Tanaman dan Teknologi Benih sebanyak 209 buah.

- Nama departemen yang tidak tercantum pada Tabel 7 menunjukkan bahwa mahasiswa yang berasal dari departemen tersebut tidak pernah meminjam buku yang mendukung pola sekuensial  $\langle(631)(633)\rangle$ .

Contoh pola sekuensial yang memiliki jumlah *item* terbanyak (8-*sequence*) yang terjadi pada nilai *minimum support* 4% dan *time constraint* 2 bulan adalah  $\langle(665)(681)(519)(633)(630)(331)(301)(316)\rangle$ . Pola sekuensial tersebut mempunyai arti bahwa sedikitnya sekitar 42 transaksi dilakukan oleh mahasiswa S1 tahun masuk 2003 untuk meminjam buku pada kelas dengan urutan transaksi peminjaman sebagai berikut:

- 665 : teknologi minyak, lemak, lilin, dan gas industri.
- 681 : alat pengukur dan alat lainnya yang teliti.
- 519 : probabilitas dan matematika terapan.
- 633 : tanaman ladang dan perkebunan.
- 630 : pertanian.
- 331 : ilmu ekonomi perburuhan.
- 301 : sosiologi dan antropologi.
- 316 : statistik umum Afrika.

Pada setiap transaksi di atas hanya satu buku yang dipinjam dengan waktu kejadian transaksi yang berbeda dalam selang waktu maksimal 2 bulan (60 hari).

## KESIMPULAN DAN SARAN

### Kesimpulan

Dari percobaan yang dilakukan terhadap data transaksi peminjaman buku di Perpustakaan IPB untuk mahasiswa S1 tahun masuk 2003, diperoleh kesimpulan sebagai berikut:

1. *Minimum support* tertinggi hingga masih terbentuk *large sequence* terjadi pada nilai 30% dan *time constraint* 6 bulan dengan transaksi peminjaman terbanyak dilakukan oleh mahasiswa yang berasal dari Departemen Pemuliaan Tanaman dan Teknologi Benih (sebanyak 209 transaksi).
2. *Minimum support* yang menghasilkan *sequence* dengan jumlah *item* terbanyak (8-*sequence*) terjadi pada nilai 4% dan *time constraint* 2 bulan.
3. Penyebab tidak terbentuknya *large sequence* pada penggunaan beberapa nilai *minimum support* dan nilai *time constraint* adalah

tabel  $L_2$  kosong atau terjadi *cycle* pada IRG (*Item Relation Graph*).

4. Nilai *time constraint* tidak mempengaruhi jumlah *large sequence* yang terbentuk.
5. Transaksi peminjaman buku yang dilakukan oleh mahasiswa S1 tahun masuk 2003 sebagian besar terjadi dalam skala kecil dengan sedikit jumlah *item* yang dipinjam dalam selang waktu antar transaksi peminjaman yang lama.
6. Semakin tinggi nilai *minimum support*, maka jumlah *item* yang membentuk pola sekuensial cenderung semakin sedikit.
7. Semakin tinggi nilai *minimum support*, maka semakin cepat waktu yang diperlukan untuk membentuk *large sequence*.

### Saran

Aplikasi *data mining* yang dihasilkan pada penelitian ini masih perlu dikembangkan lebih lanjut, dalam hal:

1. Penambahan modul yang dapat menangani *cycle* yang terjadi pada IRG.
2. Penambahan modul yang dapat menentukan pola sekuensial secara langsung dari seluruh *large sequence* yang terbentuk.

## DAFTAR PUSTAKA

- Agrawal R, Srikant R. 1995. *Mining Sequential Patterns*. Proceedings of the 11<sup>th</sup> International Conference on Data Engineering pp 3-14.
- Connolly TM, Begg CE. 2002. *Database System: A Practical Approach To Design, Implementation, and Management*. England: Addison Wesley.
- Goharian, Grossman. 2003. *Introduction to Data Mining*. <http://ir.iit.edu/~nazli/cs422/CS422-Slides/DM-Introduction.pdf> [2 Oktober 2006]
- Han J, Kamber M. 2001. *Data Mining Concepts & Techniques*. USA: Simon Fraser University Academic Press.
- Lin SY, Huang YF. 2003. *Mining Sequential Patterns Using Graph Search Techniques*. Tesis. National Yunlin University of Science and Technology, Taiwan.
- Lin MY, Lee SY. 1998. *Incremental Update on Sequential Pattern in Large Databases*. Proceedings of the Tools for Artificial Intelligence Conference (TAI'98) pp 24-31.

[TCC] Two Crows Corporation. 1999.  
*Introduction to Data Mining and Knowledge  
Discovery: Third Edition.*  
<http://www.twocrows.com/intro-dm.pdf> [1  
Oktober 2006]