

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Edi Pranoto (2007) membuat aplikasi *sequential association data mining* dengan algoritma *generalized sequential pattern*. Tujuan dari pembuatan aplikasi ini adalah membangun suatu aplikasi untuk mengolah data transaksi menjadi suatu informasi sehingga dapat membantu pelaku bisnis dalam pengambilan keputusan, antara lain mengatur tata letak barang dagangan atau penyediaan paket terhadap pasangan barang dagangan yang paling jarang dibeli oleh *customer*. Metode yang dipergunakan dalam aplikasi ini adalah metoda *Sequential Association Data Mining*, yang menggunakan algoritma *Generalized Sequential Pattern*. Hasil dari aplikasi ini adalah suatu informasi yang berguna bagi pelaku bisnis, mengenai pasangan item apa saja yang paling banyak dibeli oleh *customer* dalam kurun waktu tertentu.

Dari penelitian dan pembangunan aplikasi yang telah dibuat, maka penulis akan mencoba membangun aplikasi *sequential association data mining* terhadap data sekuensial sirkulasi peminjaman buku di perpustakaan dengan algoritma *generalized sequential pattern*. Tujuan dari pembangunan aplikasi ini adalah Menggali informasi dari *database* perpustakaan Universitas Atma Jaya yang ada untuk mengetahui *pattern* asosiasi sekuensial antar item buku berdasarkan data *sequential* transaksi peminjaman buku. Metode yang dipergunakan dalam aplikasi ini adalah metoda *Sequential Association Data Mining*, yang menggunakan

algoritma *Generalized Sequential Pattern*. Hasil dari aplikasi ini adalah suatu informasi mengenai relasi antar *item* buku yang biasa dipinjam secara bersamaan atau berurutan oleh anggota perpustakaan, dimana informasi ini dapat membantu pemimpin perpustakaan dalam pengambilan keputusan, salah satunya keputusan pembelian buku baru. Aplikasi ini dibangun dengan *Microsoft visual c#.net 2005*, menggunakan *database SQL server 2000*.

2.2 Sistem Informasi

Perkembangan teknologi informasi yang sudah sangat maju sampai saat ini sangat berpengaruh terhadap perkembangan sistem informasi yang ada. Sistem informasi menjadi salah satu kunci sukses dari berbagai organisasi/perusahaan untuk melakukan berbagai aktivitasnya. Dampak positif dari perkembangan IT (*information technology*) adalah proses data dan informasi yang menjadi tulang punggung dunia bisnis dapat dilakukan dengan cepat, akurat dan tepat waktu. Sedangkan dampak negatifnya akan timbul kejahatan penyelewengan dari penggunaan perangkat keras tersebut. Terlepas dari semua itu, kemajuan teknologi yang dapat mendukung pengolahan informasi menjadi alat pemicu persaingan dunia bisnis dan ekonomi yang semakin kompetitif.

Pada masa-masa sekarang ini, untuk pengolahan informasi dilakukan dengan bantuan komputer. Alasan utama mengapa organisasi membutuhkan sistem informasi yang baik dengan bantuan komputer (*Computer Based Information System*) adalah menerima data dari berbagai

sumber dari dalam maupun dari luar organisasi (sebagai input), mengolah data untuk menghasilkan informasi, dan memberikan informasi bagi pihak yang berkepentingan.

Sistem Informasi merupakan hal yang sangat penting bagi suatu manajemen di dalam pengambilan keputusan. Untuk memahami arti dari sistem informasi, terlebih dahulu kita harus mengerti dua kata yang menyusunnya yaitu sistem dan informasi. Kata sistem didefinisikan sebagai kumpulan elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu sedangkan kata informasi itu sendiri didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Pengertian sistem informasi tidak bisa dilepaskan dari pengertian sistem dan informasi. Secara lugas sistem informasi didefinisikan sebagai kumpulan orang, prosedur, *hardware*, *software* yang saling berinteraksi untuk memberikan suatu pelayanan informasi bagi user. Informasi juga meliputi data atau sumber daya yang tersedia dalam suatu organisasi/perusahaan yang dapat mempengaruhi hasil kinerja bagian-bagian atau elemen-elemen yang ada dalam organisasi/perusahaan.

Sistem Informasi (SI) didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan, memproses, dan menyimpan serta mendistribusikan informasi (Oetomo, 2002). Dengan kata lain, SI merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan

mendukung pembuatan keputusan dan melakukan kontrol terhadap jalannya perusahaan.

John Burch dan Gary Grudnitski dalam bukunya yang berjudul "*Information System Theory and Practice*" mengemukakan bahwa sistem informasi sendiri terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building Block*), yaitu :

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media yang menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Proses/Model adalah sebetuk abstraksi atau representasi dari realita. Proses biasanya merupakan bentuk penyedarhanaan dari hal yang nyata/kenyataan. Proses atau model diklasifikasikan ke dalam berbagai cara, misalnya:

1) Model Prosedural, adalah seperangkat pernyataan deklaratif. Setiap pernyataan bersifat *action oriented* yang berisi aksi yang harus diambil dan sebuah objek yang dikenai aksi.

2) Model Logika, hal mendasar yang ada pada model logika adalah aturan/*rule*. Sebuah aturan mengandung kondisi yang harus diuji, dan bergantung kepada hasil kondisi. Sebuah aturan juga mengandung aksi yang harus diambil.

3) Model Matematika, adalah representasi kuantitatif realita.

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkah manajeme serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi merupakan "kotak alat" dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sitem secara keseluruhan. Teknologi terdiri dari 3 bagian utama yaitu teknisi (*brainware*), perangkat keras (*Hardware*) dan perangkat lunak (*software*).

5. Blok basis data (*database block*)

Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di dalam perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan didalam basis data untuk keperluan penyediaan informasi lebih lanjut.

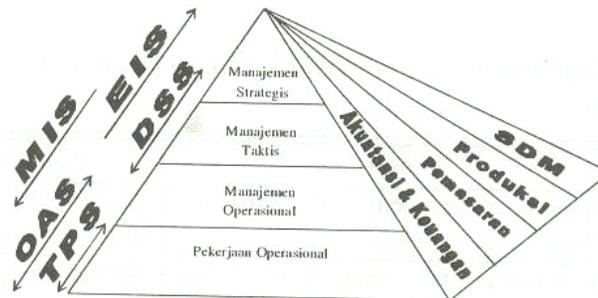
Data didalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyipannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak yang disebut dengan DBMS (*Database Management System*).

6. Blok kendali (control block)

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, air debu, kecurangan-kecurangan, kegagalan sistem, ketidak-efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan dapat langsung diatasi (John Burch; Grudnitski dalam Hartono, 2005).

Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarnya. Sistem informasi dapat juga didefinisikan sebagai suatu himpunan orang-orang, data, proses (*procedure*) yang berinteraksi untuk mendukung operasi, *management* dan informasi pembuat keputusan yang akan memberikan informasi bagi pengendali keputusan atau untuk mengendalikan organisasi.

2.2.1 Jenis-jenis Sistem Informasi



Gambar 2.1 bagian-bagian Sistem Informasi

(sumber : Kadir, 2003)

Sistem informasi dikembangkan untuk tujuan yang berbeda-beda, tergantung pada kebutuhan bisnis. Sistem informasi dapat dibagi menjadi beberapa bagian :

1. Transaction Processing Systems (TPS)

TPS adalah sistem informasi yang terkomputerisasi yang dikembangkan untuk memproses data dalam jumlah besar untuk transaksi bisnis rutin seperti daftar gaji dan inventarisasi. TPS berfungsi pada level organisasi yang memungkinkan organisasi bisa berinteraksi dengan lingkungan eksternal. Data yang dihasilkan oleh TPS dapat dilihat atau digunakan oleh manajer.

2. Office Automation Systems (OAS) dan Knowledge Work Systems (KWS)

OAS dan KWS bekerja pada level knowledge. OAS mendukung pekerja data, yang biasanya tidak menciptakan pengetahuan baru melainkan hanya menganalisis informasi sedemikian rupa untuk mentransformasikan data atau memanipulasikannya dengan cara-cara tertentu sebelum menyebarkannya secara keseluruhan dengan organisasi dan kadang-kadang diluar organisasi. Aspek-aspek OAS seperti word processing, spreadsheets, electronic scheduling, dan komunikasi melalui voice mail, email dan video conferencing.

KWS mendukung para pekerja profesional seperti ilmuwan, insinyur dan doktor dengan membantu menciptakan pengetahuan baru dan memungkinkan mereka mengkontribusikannya ke organisasi atau masyarakat.

3. Management Information Sistem (MIS)

MIS tidak menggantikan TPS , tetapi mendukung spektrum tugas-tugas organisasional yang lebih luas dari TPS, termasuk analisis keputusan dan pembuat keputusan. MIS menghasilkan informasi yang digunakan untuk membuat keputusan, dan juga dapat membantu menyatukan beberapa fungsi informasi bisnis yang sudah terkomputerisasi (basis data). Sering disebut sistem peringatan manajemen (management alerting system) - Haag, 2000; atau sistem pelaporan manajemen (management reporting system) - Zwass, 1998.

4. Decision Support Systems (DSS)

DSS hampir sama dengan MIS karena menggunakan basis data sebagai sumber data. DSS bermula dari MIS karena menekankan pada fungsi mendukung pembuat keputusan diseluruh tahap-tahapnya, meskipun keputusan aktual tetap wewenang eksklusif pembuat keputusan.

Dalam DDS terdapat tiga tujuan yang harus di capai yaitu :

- ✓ Membantu manajer dalam pembuatan keputusan untuk memecahkan masalah semiterstruktur.
- ✓ Mendukung keputusan manajer, dan bukannya mengubah atau mengganti keputusan tersebut.
- ✓ Meningkatkan efektivitas manajer dalam pembuatan keputusan, dan bukannya peningkatan efisiensi.

Tujuan ini berkaitan dengan tiga prinsip dasar dari konsep DSS, yaitu struktur masalah, dukungan keputusan, dan efektivitas keputusan.

5. Executive Information Systems (EIS)

Executive Information System (EIS) atau disebut juga sebagai *Executive Support System* (ESS) adalah sistem berbasis komputer yang interaktif, yang memungkinkan pihak eksekutif untuk mengakses data dan informasi, sehingga dapat dilakukan pengidentifikasian masalah, pengeksplorasian solusi, dan menjadi dasar dalam proses perencanaan yang sifatnya strategis.

EIS mengintegrasikan data yang berasal dari sumber data internal maupun eksternal, kemudian melakukan transformasi data ke dalam bentuk rangkuman laporan yang berguna. Laporan ini biasanya digunakan oleh manajer dan level eksekutif untuk mengakses secara cepat laporan yang berasal dari seluruh perusahaan dan departemen, sehingga dapat diperoleh pengetahuan yang berguna bagi pihak eksekutif. Laporan ini digunakan untuk menemukan alternatif solusi untuk menekan permasalahan manajerial dan membuat perencanaan keputusan untuk perusahaan.

2.3 Data Mining

Alasan utama mengapa *data mining* menarik perhatian banyak pihak yang bergerak di bidang teknologi informasi beberapa tahun belakangan ini adalah adanya ketersediaan data dalam jumlah luar biasa besarnya sekaligus juga adanya kebutuhan untuk mengolah data tersebut menjadi informasi yang lebih berguna. Informasi dan pengetahuan yang diperoleh dapat digunakan dan di aplikasikan dalam berbagai bidang,

mulai dari manajemen bisnis, kontrol produksi, dan sampai kepada analisa pasar.

Data mining dapat dikatakan sebagai hasil dari evolusi teknologi informasi mulai dari sistem *data collection, database creation, data management* (termasuk *storage* dan *retrieval* dan *database transaction processing*), dan *data analysis and understanding*.

Situasi dimana tersedianya data dalam jumlah banyak dan sekaligus juga kebutuhan akan alat analisa data yang memadai dideskripsikan juga sebagai *data rich but information poor situation* (data tersedia dalam jumlah yang banyak tapi tanpa informasi yang cukup).

Kecepatan berkembangnya pengumpulan dan penyimpanan data dalam jumlah yang luar biasa banyaknya mengakibatkan data tersebut menjadi sulit untuk dianalisa oleh seorang manusia tanpa bantuan alat bantu yang dapat meng-extract informasi penting yang terkubur di sejumlah besar data yang tersedia yang dapat menyebabkan seringkali keputusan-keputusan penting yang diambil pun dibuat tidak berdasarkan informasi yang didapatkan dari data yang ada, melainkan berdasarkan intuisi pengambil keputusan semata, hal ini sering terjadi dikarenakan para pengambil keputusan tidak memiliki *tools* yang memadai untuk melakukan pencarian informasi yang akurat dari data yang tersedia.

Program *data mining* melakukan analisa data dan dapat menemukan informasi penting mengenai pola data, dapat memberikan kontribusi besar-besaran pada strategi bisnis, *knowledge base*, dan penelitian serta riset medical.

2.4 Definisi Data Mining

Data mining mengandung arti proses menggali (*mining*) pengetahuan / *knowledge* dari sejumlah besar data, *data mining* seringkali disebut juga dengan *knowledge mining from databases*, *knowledge extraction*, *data/ pattern analysis*, *data archeology*, dan *data dredging*. Selain itu *data mining* juga sering digunakan oleh banyak orang sebagai sinonim dari *Knowledge Discovery in Databases* atau KDD. (Hand, 2001).

Teknik analisa *data mining* pada umumnya diorientasikan untuk dapat mengerjakan data yang ada dalam jumlah sebanyak mungkin, dengan tujuan *mining* terhadap data tersebut dapat menghasilkan keputusan dan kesimpulan yang terjamin keakuratannya.

Arsitektur utama dari sebuah sistem *data mining* pada umumnya mengandung unsur-unsur sebagai berikut:

1. *Database*, *datawarehouse*, atau media penyimpanan informasi : media dalam hal ini bisa jadi berupa *database*, *data warehouse*, *spreadsheets*, atau jenis-jenis penampungan informasi lainnya. *Data cleaning* dan *data integration* dapat dilakukan pada data tersebut.
2. *Database* atau *data warehouse server* : *database* atau *datawarehouse server* bertanggung jawab untuk menyediakan data yang relevan berdasarkan permintaan dari *user* pengguna *data mining*.
3. *Data mining engine*, yaitu bagian dari *software* yang menjalankan program berdasarkan algoritma yang ada.
4. *Pattern evaluation module*, yaitu bagian dari *software* yang berfungsi untuk menemukan *pattern*

atau pola-pola yang terdapat di dalam *database* yang diolah sehingga nantinya proses *datamining* dapat menemukan *knowledge* yang sesuai.

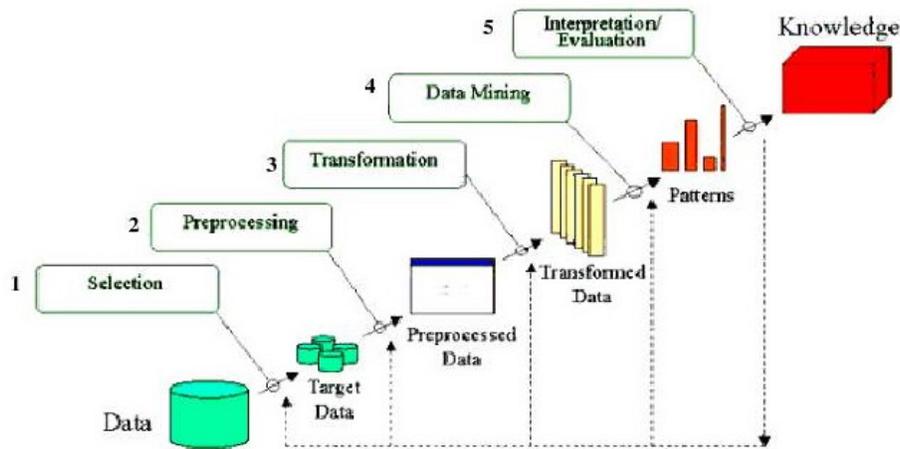
5. *Graphical user interface* : bagian ini merupakan sarana antara *user* dan sistem *data mining* untuk berkomunikasi, dimana *user* dapat berinteraksi dengan sistem melalui *data mining query*, untuk menyediakan informasi yang dapat membantu dalam pencarian *knowledge*. Lebih jauh lagi, bagian ini mengijinkan *user* untuk melakukan *browsing* pada *database* dan *data warehouse*, mengevaluasi *pattern* yang telah di hasilkan, dan menampilkan *pattern* tersebut dengan tampilan yang berbeda-beda.

Suatu sistem *data mining* yang baik, seharusnya dibangun dengan algoritma yang baik, terstruktur, cepat, dan juga dapat menangani data dalam jumlah besar, sehingga ketika menangani suatu *database* dengan ukuran besar maupun kecil, *running time-nya* pun akan berkembang secara proporsional. Dengan melakukan *data mining*, *knowledge* yang menarik, *high level information* dapat di extract dari *database* atau ditampilkan dari berbagai sudut pandang.

Data mining pada umumnya bisa dilakukan terhadap segala macam data yang tersimpan baik pada relational *database*, *dataware house*, transactional *databases*, dan tidak tertutup kemungkinan pada sebuah sistem *database* pada internet, seperti misalnya: *mining* informasi transaksi *online*.

2.5 Tahapan-Tahapan Data Mining

Tahapan-tahapan pada *data mining* dimulai dari pemrosesan *raw data* atau data mentah sampai pada penyaringan hingga ditemukannya *knowledge* (Han 2001), dijabarkan sebagai berikut :



Gambar 2.2. Tahap-tahapan dalam *data mining*
(sumber : Han 2001)

1. *Selection*, yaitu proses memilih dan memisahkan data berdasarkan beberapa kriteria, misalnya berdasarkan kota tempat tinggal konsumen.
2. *Preprocessing*, yaitu mempersiapkan data, dengan cara membersihkan data, informasi atau *field* yang tidak dibutuhkan, yang jika dibiarkan hanya akan memperlambat proses *query*, misalnya nama pelanggan jika kita sudah mengetahui kode pelanggannya. Selain itu juga, di tahap ini dilakukan penyeragaman format terhadap data yang tidak konsisten, misalnya pada suatu *field* dari suatu tabel, data jenis kelamin diinputkan dengan "L" atau "P", sedangkan pada tabel yang lain, data tersebut diinputkan sebagai "P" atau "W".

3. *Transformation*, data yang telah melalui proses *select* dan *pre-processing* tidak begitu saja langsung digunakan, tapi ditransformasikan terlebih dahulu ke bentuk yang lebih *navigable* dan *useable*, misalnya dengan menambahkan field-field tertentu yang bersifat demografi, seperti propinsi, kota, atau informasi apapun yang biasanya digunakan pada riset pemasaran.
4. *Data mining*, tahap ini dipusatkan untuk mendapatkan pola dari data (*extraction of data*).
5. *Interpretation and evaluation* dalam proses ini *pattern* atau pola-pola yang telah diidentifikasi oleh sistem kemudian diterjemahkan / diinterpretasikan ke dalam bentuk *knowledge* yang lebih mudah dimengerti oleh *user* untuk membantu pengambilan keputusan, misalnya menunjukkan item yang saling berasosiasi melalui grafik atau bentuk lain yang lebih mudah dimengerti.

Pada umumnya, *data mining* dapat digunakan untuk menganalisa dan menemukan empat tipe relasi berikut (Palace, 1996):

1. *Classes*, data yang tersedia dapat digunakan untuk menemukan hubungan dari beberapa hal yang ingin diketahui. Sebagai contoh, sebuah restoran dapat melakukan *mining* terhadap data penjualannya selama periode-periode tertentu, untuk menemukan kapan restoran tersebut menerima kunjungan paling banyak dan kunjungan paling sedikit, setelah menemukan restoran mungkin dapat menawarkan paket-paket istimewa, mungkin harga makanan dengan harga yang

lebih rendah pada jam-jam sepi untuk menarik pelanggan lebih banyak lagi pada jam-jam tersebut sehingga nantinya dapat meningkatkan penjualan dari restoran tersebut secara keseluruhan.

2. *Clusters*, data item dapat dikelompokkan / dipecah-pecah kedalam beberapa grup, berdasarkan syarat yang sudah ditentukan. Misalnya *data mining* dapat langsung mencari dan mengelompokkan konsumen di daerah yang mana saja yang mempunyai daya beli tinggi dan yang mana memiliki daya beli rendah.
3. *Associations*, data dapat di *mine* untuk menemukan item-item apa saja yang dibeli konsumen yang saling berasosiasi. Misalnya bila seorang konsumen membeli kacang ternyata konsumen tersebut juga membeli pisang atau minuman kaleng.
4. *Sequential patterns*, data dapat di *mine* untuk menemukan "*patterns and trends*" yaitu pola belanja konsumen dan juga trend yang terjadi secara berurutan. Sebagai contoh konsumen yang membeli sebuah computer bisa diperkirakan akan membeli juga *software -software game* maupun *installer*.

2.6 Sequential Pattern

Seperti yang telah disebutkan sebelumnya, bahwa *sequential pattern* mencari dan menemukan hubungan antar item yang ada pada suatu *dataset*. Program *data mining* dengan *sequential pattern* bertujuan menemukan informasi item-item yang saling berhubungan dalam bentuk *rule*, dengan demikian *sequential pattern* paling tepat diterapkan pada *mining sequential pattern* (Agrawal, 1995).

2.6.1 Konsep Dasar Sequential Pattern

Inputan datanya merupakan sekumpulan *sequence* yang disebut *data-sequences*. Setiap data *sequential* merupakan suatu daftar dari transaksi-transaksi, dimana setiap transaksi merupakan sekumpulan *item*. Umumnya setiap transaksi diasosiasikan dengan waktu transaksinya. Suatu *sequential-pattern* juga terdiri dari suatu daftar dari sekumpulan *item*. Masalahnya adalah untuk menemukan *sequential pattern* dengan minimum *support* yang ditentukan oleh *user*, dimana *support* dari sebuah *sequential pattern* merupakan persentase dari *data-sequences* yang mengandung suatu pola tertentu. Sebagai contoh: pada *database* dari suatu perpustakaan / peminjaman buku, masing-masing *data-sequences* mungkin berkoresponden kepada semua pilihan buku dari seorang *customer* dan masing-masing transaksi buku-buku yang dipilih oleh *customer* dalam satu *order*. Sebuah *sequential pattern* dapat menjadi "5% dari *customer* membeli "*Foundation*" kemudian "*Foundation and Empire*" dan kemudian "*Second Foundation*". Sebuah data *sequential* menghubungkan seorang *customer* yang membeli beberapa buku lainnya dimana termasuk diantara buku-buku yang terdiri dari suatu pola *sequential* tertentu; Data *sequential* mungkin juga memiliki buku-buku yang lain dalam transaksi yang sama sebagai salah satu buku-buku dalam pola tersebut. Bagian-bagian dari *sequential pattern* dapat merupakan sekumpulan dari *item* sebagai contohnya: "*Foundation*" dan "*Ringworld*" diikuti kemudian dengan "*Foundation and Empire*" dan "*Ringworld Engineers*" dan kemudian diikuti oleh "*Second Foundation*". Bagaimanapun juga, semua *item* dalam sebuah

bagian dari suatu pola *sequential* harus terdapat dalam sebuah transaksi untuk data *sequential* untuk mendukung pola tersebut. Untuk mudahnya dapat diasumsikan bahwa tidak ada *data-sequence* memiliki lebih dari satu transaksi dengan waktu transaksi yang sama, dan menggunakan waktu transaksi sebagai pengenal transaksi tersebut. Dan dalam hal ini kuantitas dari suatu *item* dalam sebuah transaksi tidak diperhitungkan.

2.6.2 Algoritma Generalized Sequential Pattern(GSP)

Struktur dasar dari algoritma GSP yaitu untuk menemukan pola *sequential*. Algoritma GSP ini, melakukan *multiple passes* melalui data yang ada (kumar, 2006). Fase pertama menentukan *support* dari masing-masing *item* yang mana merupakan nomor dari *data-sequences* yang termasuk *item-item* tersebut pada akhir dari fase pertama, algoritma ini akan mengetahui atau mendapatkan *item* mana yang akan menjadi *frequent*, yaitu yang memenuhi *minimum support*. Masing-masing *item* menghasilkan sebuah *frequent sequence* yang pertama yang terdiri dari *item* tersebut. Masing-masing *subsequence* pada setiap fase pada awalnya dimulai dengan sekumpulan calon kandidat: suatu *frequent-sequence* yang ditemukan atau dihasilkan pada fase sebelumnya. Sekumpulan calon kandidat tersebut digunakan untuk menghasilkan *frequent sequences* baru yang berpotensi, yang disebut *kandidat sequences*. Masing-masing *kandidat-sequence* memiliki lebih dari satu *item* daripada calon *sequence*; sehingga semua *kandidat sequences* dalam suatu fase akan memiliki *item* dengan nomor yang sama. *Support* dari *kandidat sequences* ini ditemukan selama proses melalui

data yang ada. Pada akhir dari fase tersebut, algoritma akan menghasilkan yang mana *candidate sequences* yang termasuk dalam *frequent*. Dimana *frequent candidate* tersebut menjadi calon kandidat untuk fase selanjutnya. Dimana algoritma tersebut berakhir ketika tidak ditemukan lagi *frequent sequences* pada akhir suatu fase, atau ketika tidak ada lagi *candidate sequences* yang dihasilkan. Terdapat 2 langkah utama dalam algoritma ini, yaitu *candidate generation* dan *support counting*.

2.6.2.1 Candidate Generation

Candidate-candidate tersebut dihasilkan melalui 2 langkah :

1. *Join Phase*. Dimana *candidate sequence* dihasilkan dengan melakukan proses *join* atau penggabungan antara L_{k-1} dengan dirinya sendiri. Sekumpulan *candidate* yang dihasilkan dalam proses *join* ini nantinya akan dinotasikan dalam C_k . Dimana aturan dari fase *join* ini adalah setiap *candidate* yang dihasilkan tidak boleh mengandung *candidate* yang kembar antara satu dengan yang lainnya.
2. *Prune Phase*. Fase ini melakukan penghapusan *candidate sequence* yang tidak memenuhi *minimum support* yang telah ditentukan. Berdasarkan definisi, semua *candidate* yang memiliki jumlah *support* yang lebih besar sama dengan dari *minimum support* yang telah ditentukan disebut *frequent*, yang artinya juga memenuhi syarat untuk masuk menjadi L_k . Dimana C_k dapat juga mengandung *candidate* dengan jumlah yang sangat besar yang berarti juga akan menyebabkan proses penghitungan

C_k selanjutnya akan berjalan sangat lambat. Untuk mengurangi jumlah *candidate* C_k maka semua *Itemset* yang tidak *frequent* tidak mungkin dapat menjadi subset dari *frequent k-itemset*. Oleh karena itu jika ada sebuah subset dari *candidate k-itemset* yang tidak termasuk dalam L_{k-1} maka *candidate* tidak mungkin *frequent* juga dan oleh karena itu dapat dihapus dari C_k .

2.6.2.2 Counting Candidates

Sementara melakukan proses melalui data yang ada, juga dilakukan penambahan jumlah *support* dari *candidate* yang termasuk dalam *data-sequences*. Misalnya, diberikan sekumpulan *candidate sequences* C dan *data-sequences* d , disini yang perlu ditemukan adalah semua *sequence* pada C yang termasuk di dalam d .

Berikut adalah algoritma GSP yang akan digunakan untuk menemukan *frequent itemset*. Dimana *output* yang akan dihasilkan oleh program adalah semua *frequent itemset* yang memenuhi *minimum support*. Adapun *input* yang harus disediakan oleh *user* antara lain untuk dapat mencapai *output* yang diinginkan adalah *database* dari suatu transaksi serta *minimum support* yang ditentukan langsung oleh *user*. Berikut akan diberikan penjelasan lebih lanjut melalui contoh pemakaian algoritma GSP untuk menemukan *sequential pattern*. Contoh berikut adalah merupakan *database* transaksi dari sebuah toko buku dimana setiap transaksi menunjukkan *item* yang dibeli oleh konsumen dalam suatu waktu tertentu.

Customer-Id	Transaction-Time	Items
1	10	C D
1	15	A B C
1	20	A B F
1	25	A C D F
2	15	A B F
2	20	E
3	10	A B F
4	10	D C H
4	20	B F
4	25	A C H

Tabel 2.1. Tabel contoh data

Dari data pada label 2.1., didapatkan 8 *item* (A s/d H), 4 *customer* dan 10 transaksi, dengan diberikan *minimum support* 50% atau 2 orang *customer* maka akan diperoleh *frequent sequences* seperti dibawah ini :

$$F_1 = \{(A)[4], (B)[4], (D)[2], (F)[4]\}$$

$$F_2 = \{(AB)[3], (AF)[3], (B \rightarrow A)[2], (BF)[4], (D \rightarrow A)[2], (D \rightarrow B)[2], (D \rightarrow F)[2], (F \rightarrow A)[2]\}$$

$$F_3 = \{(ABF)[3], (BF \rightarrow A)[2], (D \rightarrow BF)[2], (D \rightarrow B \rightarrow A)[2], (D \rightarrow F \rightarrow A)[2]\}$$

$$F_4 = \{(D \rightarrow BF \rightarrow A)[2]\}$$

Dari suatu data *sequential* di atas dapat digunakan untuk menghasilkan suatu *rule* yang menjelaskan relasi antara *item* yang berbeda, sebagai contohnya : transaksi (BF) terdapat pada 4 *customer* , sementara (ABF) terdapat pada 3 *customer*. Maka dapat dikatakan bahwa (BF) \rightarrow (BFA) memiliki *confidence* 75%. Contoh lainnya yaitu *rule* (D \rightarrow BF) \rightarrow (D \rightarrow BF \rightarrow A) memiliki 100% *confidence*.

2.6.3 Confidence

Setelah *rule frequent itemset* telah ditemukan atau dihasilkan, maka akan dilakukan proses untuk menghasilkan *rule* yang memenuhi baik *minimum confidence*, maupun *minimum support*. Dimana proses penghitungan *confidence* dilakukan sebagai berikut :

```
RuleGen( $\mathcal{F}$ , min_conf):  
  for all frequent sequences  $\beta \in \mathcal{F}$  do  
    for all subsequences  $\alpha \prec \beta$  do  
      conf =  $fr(\beta)/fr(\alpha)$ ;  
      if (conf  $\geq$  min_conf) then  
        output the rule  $\alpha \Rightarrow \beta$  and conf
```

Ket : Algoritma Rule Generation

dimana *support_count* dari *frequent sequence* (ABF) adalah semua *subsequences* dan *frequent sequence* (ABF) yaitu (A), (B), (F), (AB), (AF), dan (BF), dan *support_count* (A) adalah jumlah transaksi yang mengandung *itemset* A.

Karena *rule* berikut dihasilkan dari *frequent itemset*, maka setiap *item* pasti juga secara otomatis memenuhi *minimum support*. Sehingga menghasilkan *output rule* yang ditampilkan sebagai berikut :

1. Rule 1 : (A)→(ABF) ,tingkat *confidence* nya = $3/4$
= 75%.
2. Rule 2 : (B)→(ABF) ,tingkat *confidence* nya = $3/4$
= 75%.
3. Rule 3 : (F)→(ABF) ,tingkat *confidence* nya = $3/4$
= 75%.
4. Rule 4 : (AB)→(ABF) ,tingkat *confidence* nya = $3/3$
= 100%.

5. Rule 5 : (AF)→(ABF) ,tingkat *confidence* nya = $\frac{3}{3}$
= 100%.

6. Rule 6 : (BF)→(ABF) ,tingkat *confidence* nya = $\frac{3}{4}$
= 75%.

Sehingga dapat dijelaskan bahwa transaksi (BF) terdapat pada 4 *customer* , sementara (ABF) terdapat pada 3 *customer*. Maka dapat dikatakan bahwa (BF) → (BFA) memiliki *confidence* 75%. Contoh Lainnya yaitu rule (D→BF) → (D→BF→A) memiliki 100% *confidence*. Misalnya minimum *confidence* yang ditentukan *user* adalah 80%, maka hanya rule keempat dan kelima saja yang memenuhi syarat sebagai *output* bagi *user*.

2.7 Teknologi .NET

Microsoft .NET adalah sebuah platform untuk membangun, menjalankan dan meningkatkan generasi lanjut dari aplikasi terdistribusi (Jaenudin, 2006). Microsoft .NET memperluas klien, server dan service-service yang terdiri atas:

1. Sebuah model pemrograman yang memungkinkan developer membangun XML Web services dan aplikasi. Sekumpulan XML Web services seperti Microsoft .NET
2. My Services, yang membantu developer menghasilkan aplikasi yang simple dan terpadu. Sekumpulan server, termasuk Windows 2000, SQL Server, dan BizTalk Server, yang memadukan, menjalankan dan mengoperasikan serta menangani XML Web Services dan aplikasi.
3. Tool seperti Visual Studio.NET untuk membangun XML Web Service dan aplikasi window serta web.

4. Piranti lunak klien, seperti Windows XP dan Windows CE.

2.7.1 .NET Framework

Microsoft .NET adalah model pemrograman dari platform .NET untuk membangun, menyebarkan dan menjalankan XML Web Service dan aplikasi (Jaenudin, 2006) .NET Framework menyediakan lingkungan berbasis standar produksi yang tinggi untuk memadukan investasi yang ada dengan aplikasi dan service generasi yang akan datang. .NET Framework terdiri atas dua bagian utama, yaitu CLR dan gabungan kelas library termasuk ASP.NET untuk aplikasi web dan XML Web Services, Windows forms untuk aplikasi klien dan ADO.NET.

Visual Studio.NET dibangun menggunakan fondasi .NET Framework. .NET Framework menyediakan lingkungan yang cerdas, mudah dikembangkan untuk membangun, menyebarkan dan menjalankan XML Web Services yang terdistribusi serta aplikasi. Dalam istilah yang mudah, .NET memisahkan platform Sistem Operasi menjadi dua layer, yaitu sebuah layer pemrograman dan layer eksekusi (Jaenudin, 2006).