BAB II
TINJAUAN PUSTAKA

II.1 Profil Jurusan

Teknik Informatika yang merupakan salah satu disiplin ilmu pada bidang Teknologi Informasi, dikembangkan untuk memberikan pengetahuan tentang pemanfaatan Teknologi Informasi tersebut guna memenuhi kebutuhan-kebutuhan manusia yang semakin beragam dan kompleks. Jurusan Teknik Informatika Universitas Komputer Indonesia akan diarahkan untuk menghasilkan seorang analis berlandaskan pada pengetahuan informatika teoretik dengan memaksimalkan pemanfaatan teknologi *on-line* yang diharapkan mempunyai kemampuan untuk memecahkan berbagai masalah dalam dunia nyata. Lulusan Teknik Informatika diharapkan tidak hanya sekedar mampu menggunakan program-program aplikasi komputer (*user*) atau membuat program aplikasi komputer (*programmer*), tetapi menjembatani kebutuhan-kebutuhan *user* dan programmer yang kemudian diterjemahkan dalam sebuah dokumen spesifikasi formal.

II.1.1 Identitas Jurusan

Nama Perguruan Tinggi	: Universitas Komputer Indonesia
Alamat	: Perguruan Tinggi Jl. Dipati Ukur No.112-114-116 Bandung 40132.
Program Studi (PS)	: Teknik Informatika
Fakultas	: Teknik dan Ilmu Komputer (FTIK)
Alamat	: Kampus V Lt 3 Jl. Dipati Ukur 114 Bandung 40132 Telp. (022) 253 3825 Fax. (022) 253 3754

Website : http://if.unikom.ac.id E-mail : if@unikom.ac.id

SK Pendirian Program Studi (PS): SK Mendiknas No. 126/D/O/2000 Tanggal 8 Agustus 2000

Peringkat Akreditasi : B (Baik) SK. Badan Akreditasi PT No. 019/BAN-PT/Ak-X/S1/XII/200 Tanggal 8 Desember 2006

Ketua Jurusan : Mira Kania Sabariah, S.T., M.T.

Sekretaris Jurusan : Tati Harihayati M, M.T.

II.1.2 Sejarah Jurusan

Pada tahun 1994, Ir. Eddy Suryanto Soegoto, M.Sc mendirikan **Lembaga Pendidikan Komputer Indonesia Jerman (LPKIG)** untuk program pendidikan 1 tahun, menempati kampus Jl. Dipati Ukur No. 102. Di tahun ketiga, nama Lembaga Pendidikan Komputer Indonesia Jerman (LPKIG) diubah menjadi ***Indonesian Germany Institute*** **(IGI)**. Pada tanggal 24 Desember 1998 dibentuk **Yayasan Science dan Teknologi** yang dilanjutkan dengan Pengajuan ke Kopertis IV Jabar untuk pendirian STIE IGI dan STMIK IGI. Pada bulan Agustus 1999 keluar ***SK Mendiknas No. 143/D/O/1999*** atas **STIMK IGI** dengan **5 Program Studi** yaitu Teknik Informatika S1, Manajemen Informatika D3 & D1, Teknik Komputer D3 & D1, Teknik Informatika D3 & D1, Komputerisasi Akuntansi D3 & D1**.** Mengantisipasi pesatnya perkembangan Teknologi Informasi, IPTEK, Era Globaliasi dan Milenium ke-3 serta untuk memberikan yang terbaik bagi masa depan mahasiswa/i nya, Yayasan Science dan Teknologi kemudian mengajukan usulan ke DIKTI melalui Kopertis Wilayah IV Jabar untuk melakukan merger atas kedua Sekolah Tinggi diatas untuk menjadi **Universitas Komputer Indonesia**.

Pada hari selasa, tanggal 8 Agustus 2000 keluar **SK Mendiknas No. 126/D/O/2000** atas nama Akreditasi Universitas Komputer Indonesia, disingkat UNIKOM. Akreditasi Mendiknas atas UNIKOM disertai menetapkan kembali status Terdaftar kepada Program Studi Teknik Informatika jenjang pendidikan program S1 (Sarjana).

Visi

Visi program studi adalah menjadi program studi Teknik Informatika unggulan yang dapat menjawab tantangan perkembangan Teknologi Informasi dan Komputer yang berkembang sangat pesat, mampu menyediakan tenaga ahli kompeten, memiliki integritas kepribadian tinggi, dan tanggap terhadap perkembangan ilmu pengetahuan, teknologi, serta tuntutan masyarakat, bangsa, dan negara.

Misi

Misi program studi adalah menyelenggarakan program pendidikan tinggi, pelatihan dan penelitian yang terbaik di bidang teknik informatika berbasis kepada teknologi informasi dan komputer melalui pengembangan kurikulum akademik yang dinamis, fasilitas pendidikan dengan kualitas terbaik, dan staff akademik, teknisi dan administrasi yang memiliki pendidikan berkualitas.

Tujuan

Tujuan program studi adalah menghasilkan Sarjana yang memiliki pengetahuan dan keahlian dalam bidang teknik informatika yang profesional dalam berwawasan teknologi informasi dan komputer, mampu mengikuti perkembangan ilmu dan teknologi, serta beriman dan bertakwa kepada Tuhan Yang Maha Kuasa.

Program pendidikan untuk mendapatkan gelar sarjana S1 Program Studi Teknik Informatika Fakultas Teknik UNIKOM terdiri dari 147 sks yang dapat ditempuh dalam waktu 8 semester. Diharapkan nantinya mahasiswa akan memperoleh ilmu dan keterampilan yang memadai untuk terjun ke lapangan sesuai dengan bidang keahliannya. Berikut adalah contoh kurikulum yang digunakan di jurusan Teknik Informatika UNIKOM :

Tabel II.1. Daftar Matakuliah Kurikulum 2007

KURIKULUM 2007			
SMT	**KDMK**	**MATA KULIAH**	**SKS**
I	IF31102	PANCASILA	2
	IF31105	BAHASA INGGRIS	2
	IF31201	KALKULUS I	3
	IF31202	FISIKA DASAR I	3
	IF31203L	PRAKTIKUM FISIKA DASAR I	1
	IF31204	ALGORITMA DAN PEMROGRAMAN I	2
	IF31205	PENGANTAR ILMU KOMPUTER	2
	IF31206L	SOFTWARE TERAPAN I	2
	IF31207L	APLIKASI IT I	2
		TOTAL	**19**
II	IF32101	AGAMA DAN ETIKA	2
	IF32208	ALGORITMA DAN PEMROGRAMAN II	2
	IF32209	KALKULUS II	3
	IF32210	FISIKA DASAR II	2
	IF32211L	PRAKTIKUM FISIKA DASAR II	1
	IF32212	STATISTIKA DAN PROBABILITAS	2
	IF32213L	SOFTWARE TERAPAN II	2
	IF32214L	APLIKASI IT II	2
	IF32301	KONSEP TEKNOLOGI	2
		TOTAL	**18**
III	IF33215	KALKULUS LANJUT	3
	IF33216	LOGIKA MATEMATIKA	3
	IF33217	ORGANISASI KOMPUTER	3
	IF33218	ALJABAR LINEAR DAN MATRIKS	3
	IF33219	SISTEM BERKAS	3
	IF33302	PEMROGRAMAN I	2
	IF33303L	PRAKTIKUM PEMROGRAMAN I	2
		TOTAL	**19**
IV	IF34220	MATEMATIKA DISKRIT	3
	IF34221	METODA NUMERIK	3
	IF34222	STRUKTUR DATA	3
	IF34304	PEMROGRAMAN II	2
	IF34305L	PRAKTIKUM PEMROGRAMAN II	2
	IF34308	KOMUNIKASI DATA	3

	IF34332	BASIS DATA	3
		TOTAL	**19**
V	IF35223	TEORI BAHASA DAN AUTOMATA	3
	IF35309	PEMODELAN DAN SIMULASI	3
	IF35311	JARINGAN KOMPUTER	2
	IF35313L	APLIKASI TEKNOLOGI ONLINE (E-COMMERCE)	2
	IF35314	ARSITEKTUR KOMPUTER	3
	IF35317	SISTEM INFORMASI	3
	IF35333	SISTEM BASIS DATA	2
	IF35334L	PRAKTIKUM SISTEM BASIS DATA	2
		TOTAL	**20**
VI	IF36103	KEWARGANEGARAAN	2
	IF36312L	PRAKTIKUM JARINGAN KOMPUTER	2
	IF36315	RISET OPERASIONAL	3
	IF36316	PEMROGRAMAN BERORIENTASI OBJEK	3
	IF36318	REKAYASA PERANGKAT LUNAK	3
	IF36319	SISTEM OPERASI	3
	IF36340	ENTERPRENEURSHIP	3
		TOTAL	**19**
VII	IF37320	SISTEM TERDISTRIBUSI	3
	IF37321	KECERDASAN BUATAN	3
	IF37401	METODOLOGI PENELITIAN	2
	IF37402	INTERAKSI MANUSIA DENGAN KOMPUTER	3
	IF37501K	KERJA PRAKTEK	2
	IF373XXP	PILIHAN 1 (Diisi Lengkap)	3
	IF373XXP	PILIHAN 2 (Diisi Lengkap)	3
		TOTAL	**19**
VIII	IF38104	BAHASA INDONESIA	2
	IF38346L	PRAKTIKUM PERANGKAT KERAS	3
	IF383XXP	PILIHAN 1 (Diisi Lengkap)	3
	IF383XXP	PILIHAN 2 (Diisi Lengkap)	3
	IF38502S	SEMINAR DAN SKRIPSI	6
		TOTAL	**17**

II.2 Landasan Teori

Subbab ini berisi uraian tentang dasar-dasar teori yang melandasi dan sesuai dengan tujuan penelitian ini. Teori, konsep dasar dan metode yang berkaitan yang menjadi dasar tujuan akan dijelaskan dalam bab ini.

II.2.1 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan(SPK) atau yang lebih dikenal dengan istilah *Decision Support Systems (DSS)* adalah : [Turban, 2005]

a. Sistem berbasis komputer yang interaktif, yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tak terstruktur.
b. SPK mendayagunakan *resource* individu-individu secara intelek dengan kemampuan komputer untuk meningkatkan kualitas keputusan. Jadi ini merupakan sistem pendukung yang berbasis komputer untuk manajemen pengambilan keputusan yang berhubungan dengan masalah-masalah yang semi terstruktur.
c. Istilah SPK kadang digunakan untuk menggambarkan sembarang sistem yang terkomputerisasi
d. SPK digunakan untuk definisi yang lebih sempit dan digunakan istilah *Management Support Systems*(MSS) sebagai payung untuk menggambarkan berbagai tipe sistem pendukung.

Proses pengambilan keputusan terdiri dari 3 fase proses yaitu :

a. *Intelligence*, yaitu pencarian kondisi-kondisi yang dapat menghasilkan keputusan. Proses yang terjadi pada fase ini antara lain : menemukan masalah, klasifikasi masalah, penguraian masalah dan kepemilikan masalah.
b. *Design*, menemukan, mengembangkan dan menganalisis materi-materi yang mungkin untuk dikerjakan. Termasuk juga disini pemahaman masalah dan pengecekan solusi yang layak. Juga model dari masalahnya dirancang, dites dan divalidasi. Tugas-tugas yang ada pada tahap ini merupakan kombinasi dari seni dan pengetahuan, yaitu:
    1. Komponen-komponen model
    2. Struktur model
    3. Seleksi prinsip-prinsip pemilihan (kriteria evaluasi)
    4. Pengembangan (penyedian) alternatif
    5. Prediksi hasil

6. Pengukuran hasil
7. Skenario

c. *Choice,* yaitu pemilihan dari materi-materi yang tersedia, mana yang akan dikerjakan

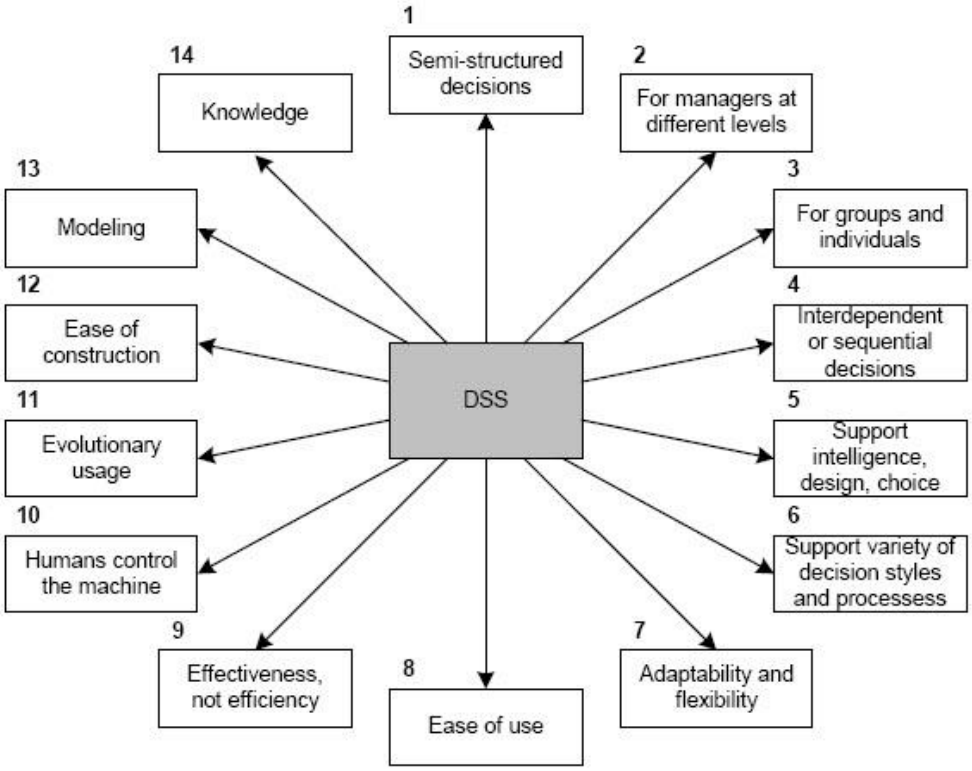
Proses-proses yang terjadi pada kerangka kerja SPK dibedakan menjadi 3 jenis yaitu :

a. **Terstruktur**, mengacu pada permasalahan rutin dan berulang untuk solusi standar yang ada.

b. **Tak terstruktur**, adalah "*fuzzy*', permasalahan kompleks dimana tak ada solusi serta merta. Masalah yang tak terstruktur adalah tak adanya 3 fase proses yang terstruktur.

c. **Semi terstruktur**, terdapat beberapa keputusan terstruktur, tetapi tak semuanya dari fase-fase yang ada.

Adapun karakteristik dan kemampuan yang dimiliki oleh SPK dapat dilihat pada gambar dibawah ini :

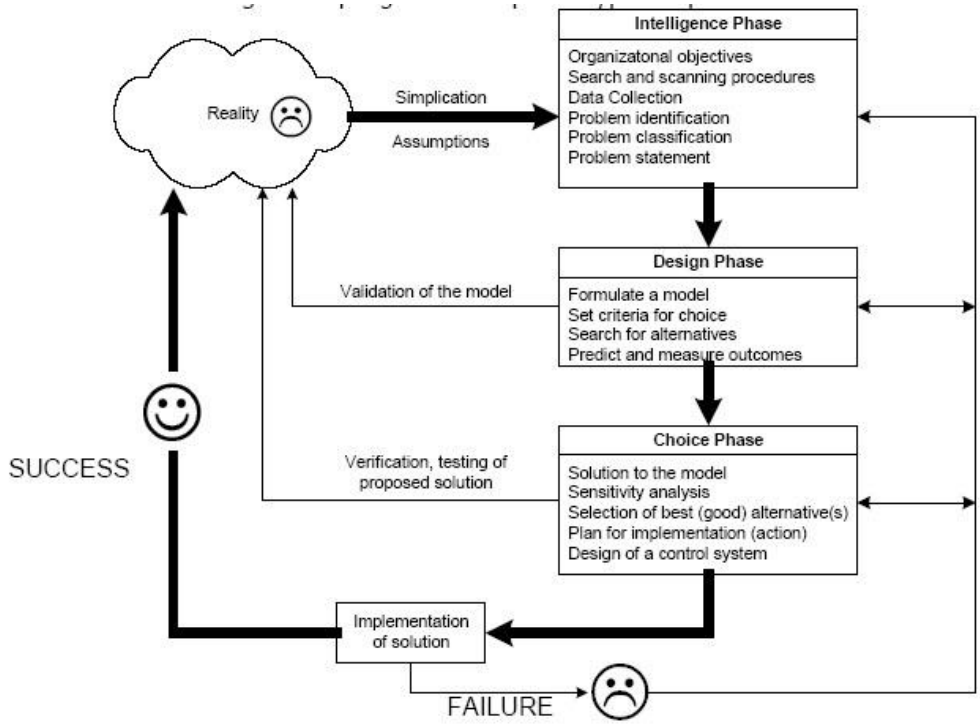


Gambar II.1 Karateristik dan kemampuan SPK [Turban, 2005]

1. SPK menyediakan dukungan bagi pengambil keputusan utamanya pada situasi semi terstruktur dan tak terstruktur dengan memadukan pertimbangan manusia dan informasi yang terkomputerisasi.
2. Dukungan disediakan untuk perbagai level manajerial yang berbeda, mulai dari pimpinan puncak sampai manajer lapangan.
3. Dukungan disediakan bagi individu dan juga bagi group Berbagai masalah organisasi yang melibatkan pengambilan keputusan dari orang dalam group.
4. SPK menyediakan dukungan ke berbagai keputusan yang berurutan atau saling berkaitan.
5. SPK mendukung perbagai fase proses pengambilan keputusan : *intelligence, design, dan choice.*
6. SPK mendukung berbagai proses pengambilan keputusan dan style yang berbeda-beda; ada kesesuaian diantara SPK dan atribut pengambil keputusan individu.
7. SPK selalu bisa beradaptasi sepanjang masa. Pengambil keputusan harus reaktif, mampu mengatasi perubahan kondisi secepatnya dan beradaptasi untuk membuat SPK selalu bisa menangani perubahan ini. SPK adalah fleksibel, sehingga user dapat menambahkan, menghapus, mengkombinasikan, mengubah atau mengatur kembali elemen-elemen dasar (menyediakan respon cepat pada situasi yang tak diharapkan). Kemampuan ini memberikan analisis yang tepat waktu dan cepat setiap saat.
8. SPK mudah digunakan. User harus merasa nyaman dengan sistem ini. *User friendly*, fleksibelitas, dukungan grafis terbaik dan antarmuka bahasa yang sesuai dengan bahasa manusia dapat meningkatkan efektivitas SPK. Kemudahan penggunaan ini diimplikasikan pada mode yang interaktif.
9. SPK mencoba untuk meningkatkan efektivitas dari pengambilan keputusan (akurasi, jangka waktu, kualitas), lebih daripada efisiensi yang bisa diperoleh (biaya membuat keputusan, termasuk biaya penggunaan komputer).

10. Pengambil keputusan memiliki kontrol menyeluruh terhadap semua langkah proses pengambilan keputusan dalam menyelesaikan masalah. SPK secara khusus ditujukan untuk mendukung dan menggantikan pengambil keputusan. Pengambil keputusan dapat menindaklanjuti rekomendasi komputer sembarang waktu dalam proses dengan tambahan pendapat pribadi atau pun tidak.
11. SPK mengarah pada pembelajaran, yaitu mengarah pada kebutuhan baru dan penyempurnaan sistem, yang mengarah pada pembelajaran tambahan dan begitu selanjutnya dalam proses pengembangan dan peningkatan SPK secara berkelanjutan.
12. User/pengguna harus mampu menyusun sendiri sistem yang sederhana. Sistem yang lebih besar dapat dibangun dalam organisasi user tadi dengan melibatkan sedikit saja bantuan dari spesialis dibidang *Information Systems* (IS).
13. SPK biasanya mendayagunakan berbagai model (standar atau sesuai keinginan user) dalam menganalisis berbagai keputusan. Kemampuan pemodelan ini menjadikan percobaan yang dilakukan dapat dilakukan pada berbagai konfigurasi yang berbeda.
14. SPK dalam tingkat lanjut dilengkapi dengan komponen *knowledge* yang bisa memberikan solusi yang efisien dan efektif dari berbagai masalah yang pelik.

    Adapun proses pengambilan keputusan dapat dilihat dari gambar dibawah ini :



Gambar II.2 Proses pengambilan keputusan [Turban, 2005]

Komponen-komponen pembangun SPK antara lain : [Turban, 2005]

1. *Data Management*

   Termasuk database, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh software yang disebut *Database Management Systems* (DBMS)

2. *Model Management*

   Melibatkan model finansil, statistical, management *science* atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan anlitis dan manajemen software yang diperlukan.

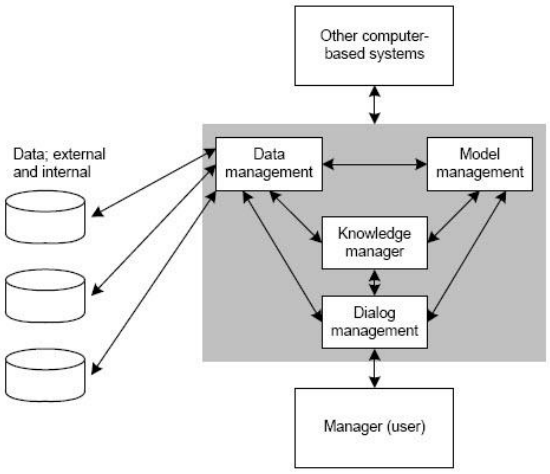
3. *Communication*

   User dapat berkomunikasi dan memberikan perintah pada SPK melalui subsistem ini. Ini berarti menyediakan antarmuka.

4. *Knowleadge Management*

   Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

Dibawah ini adalah model konseptual, yaitu sebagai berikut :

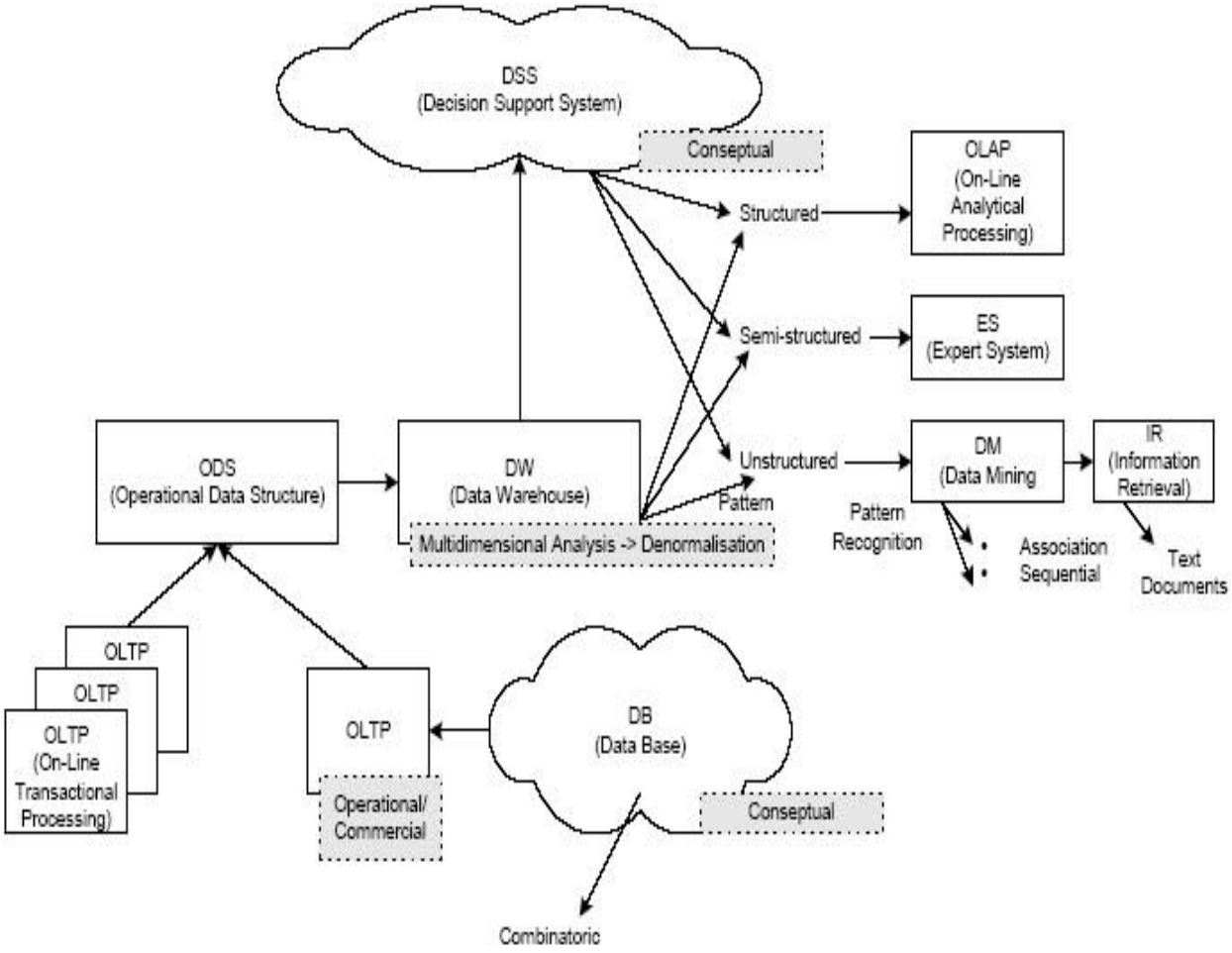


Gambar II.3 Model konseptual [Turban, 2005]

Kemampuan yang dimiliki DBMS dalam SPK adalah sebagai berikut : [Turban, 2005]

a. Mendapatkan/mengekstrak data agar bisa masuk ke dalam database SPK.
b. Secara cepat meng-update (menambah, menghapus, mengedit, mengubah) record data dan file.
c. Menghubungkan data dari berbagai *source*.
d. Secara cepat menampilkan data dari database dalam queiries dan *report*.
e. Menyediakan keamanan data menyeluruh (proteksi dan akses yang tidak berhak, kemampuan recover dan lain-lain).
f. Menangani data personal dan tidak resmi sehingga user dapat mencoba dengan berbagai solusi alternatif berdasarkan pertimbangan mereka sendiri.
g. Memberikan tampilan data yang lebih kompleks dan proses manipulasinya berdasarkan queries yang diberikan.
h. Melacak kegunaan data.

Berikut ini disajikan diagram hubungan dari bidang-bidang KDE :



Gambar II.4 Diagram hubungan bidang-bidang KDE [Turban, 2005]

II.2.2 Data Mining

II.2.2.1 Pengertian Data Mining

Data Mining merupakan salah satu cabang ilmu komputer yang relatif baru yang memiliki keterkaitan dengan *machine learning*, kecerdasan buatan (*artificial intelligence*), statistik, dan *database*. *Data Mining* mengacu kepada ekstraksi atau penggalian pengetahuan dari suatu data dalam jumlah besar. Ada banyak pengertian dari *data mining* itu sendiri, diantaranya seperti penggalian pengetahuan dari database, ekstraksi pengetahuan (*knowledge extraction*), analisis data atau pola (*pattern analysis*), penggalian data dan lain sebagainya.

Definisi *Data Mining* menurut **Budi Santosa** dalam bukunya yang berjudul ***Data Mining (Teknik Pemanfaatan Data untuk Keperluan Bisnis)*** adalah sebagai berikut:

> "Kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar."[Santosa, 2007]

Sedangkan menurut **Pang-Ning Tan** dalam bukunya yang berjudul ***Introduction to Data Mining***, definisi *Data Mining* adalah sebagai berikut:

> *"Process of automatically discovering useful information in large data repositories."*[Tan, 2006]

II.2.2.2 Proses Data Mining

Dalam aplikasinya, *data mining* sebenarnya merupakan bagian dari proses *Knowledge Discovery in Databases* atau KDD, bukan sebagai teknologi yang utuh dan berdiri sendiri. Data mining merupakan suatu bagian langkah yang penting dalam proses KDD terutama berkaitan dengan ekstraksi dan penghitungan pola-pola dari data yang ditelaah.

Proses – proses dari KKD sendiri, menurut **Iko Pramudiono** dalam artikelnya berjudul **Pengantar *Data Mining***, terdiri dari:

1. **Pembersihan data (*Data cleaning*)**

   Menghilangkan *noise* dan data inkonsisten.

2. **Integrasi data (*Data integration*)**

   Menggabungkan data dari berbagai sumber data yang berbeda.

3. **Pemilihan data (*Data selection*)**

   Mengambil data yang relevan dengan tugas analisis dari database.

4. **Transformasi data (*Data transformation*)**

   Mentransformasi atau menggabungkan data ke dalam bentuk yang sesuai untuk penggalian lewat operasi *summary* atau *aggregation.*

5. **Penambangan data (*Data mining*)**

   Proses esensial untuk mengekstrak pola dari data dengan metode cerdas.

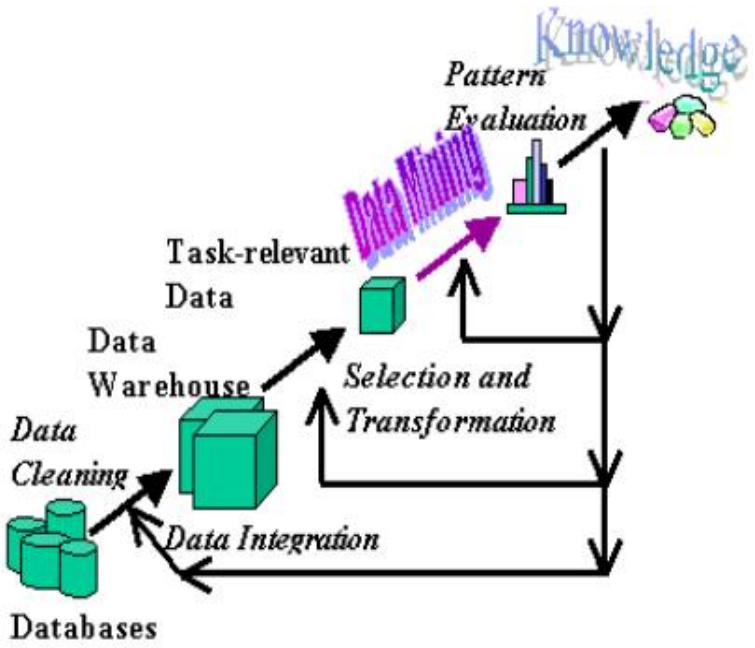
6. **Evaluasi pola (*Pattern evaluation*)**

   Mengidentifikasikan pola yang menarik dan merepresentasikan pengetahuan berdasarkan *interestingness measures.*

7. **Presentasi pengetahuan (*Knowledge presentation*)**

   Penyajian pengetahuan yang digali kepada pengguna dengan menggunakan visualisasi dan teknik representasi pengetahuan.

Tahap-tahap tersebut bersifat interaktif dimana pengguna atau *user* terlibat langsung atau dengan perantaraan basis pengetahuan (*knowledge base*) yang terintegrasi didalam sistem. Pola-pola yang menarik disajikan kepada pengguna dan disimpan sebagai pengetahuan baru di dalam basis pengetahuan. Dari tahapan tersebut dapat diketahui bahwa *data mining* hanya merupakan satu bagian langkah dari keseluruhan proses KDD.



Gambar II.5 *Data Mining* sebagai bagian dari proses KDD [HAN06]

Umumnya sistem *data mining* terdiri dari komponen-komponen berikut:

1. *Data Warehouse*

   Terdiri dari satu atau beberapa *database*, *data warehouse*, atau data dalam bentuk lain. Pembersihan data dan integrasi data dilakukan terhadap data tersebut.

2. *Data Warehouse Server*

   Bertanggung jawab terhadap pencarian data yang relevan sesuai dengan yang diinginkan pengguna atau *user*.

3. Basis Pengetahuan (*Knowledge base*)

   Merupakan basis pengetahuan yang digunakan sebagai panduan dalam pencarian pola.

4. *Data mining engine*

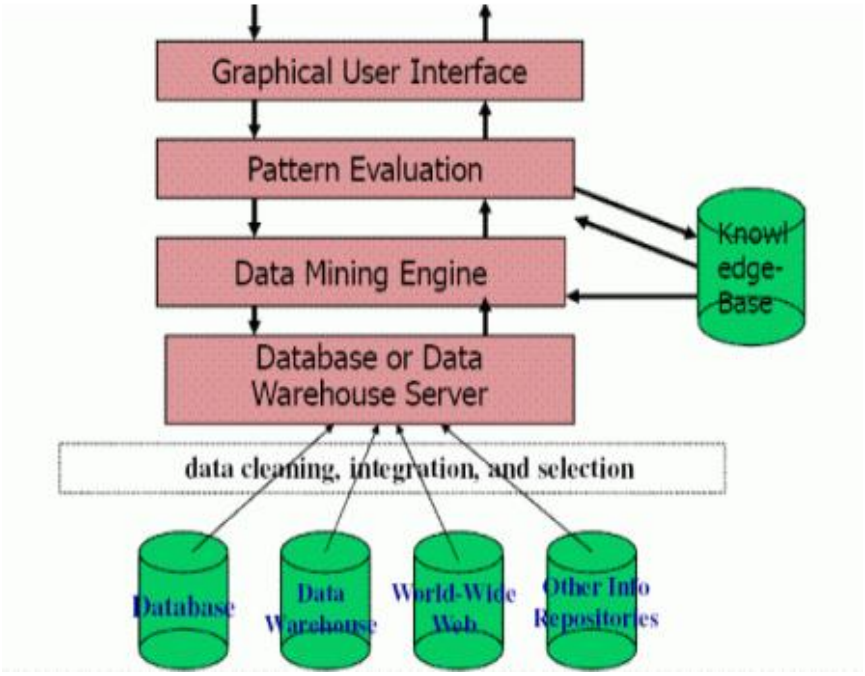
   Merupakan bagian penting dari sistem dan idealnya terdiri dari kumpulan modul-modul fungsi yang digunakan dalam proses karakterisasi (*characterization*), klasifikasi (*classification*) dan analisis kluster (*cluster analysis*).

5. Evaluasi Pola (*Pattern evaluation*)

   Komponen ini pada umumnya berinteraksi dengan modul-modul data mining.

6. Antarmuka (*Graphical User Interface*)

   Merupakan modul komunikasi antara pengguna atau user dengan sistem yang memungkinkan pengguna berinteraksi terhadap sistem untuk menentukan proses data mining itu sendiri.



Gambar II.6 Sistem Data Mining [HAN06]

II.2.2.3 Data Pada Data Mining

Pada prinsipnya data mining dapat diaplikasikan pada berbagai jenis data, yaitu pada *database relational*, *data warehouse*, database transaksional, advance *database system*, *flat-files* dan *world wide web*. Tetapi teknik mining pada setiap jenis data berbeda. [HAN06]

1. ***Database Relational***

   Suatu database relasional terdiri dari kumpulan tabel-tabel, dimana masing-masing tabel mempunyai nama yang unik. Setiap tabel terdiri dari kumpulan atribut (kolom atau *field*) dan biasanya menyimpan banyak *record*. Setiap *record* pada *database relational* mewakili suatu objek yang diidentifikasi oleh sebuah kunci yang unik.

   Database relasional dapat diakses melalui *query* yang ditulis dalam bahasa *query* seperti SQL, atau melalui antarmuka grafis yang tersedia. Database relasional merupakan database yang sangat populer dan kaya akan informasi yang tersimpan didalamnya oleh karena itu data mining banyak diterapkan pada *database relational*.

2. *Data Warehouse*

Suatu *data warehouse* mengkonsolidasikan data dari berbagai sumber data dengan aturan penamaan, ukuran atribut fisik, serta semantik yang konsisten. *Data warehouse* dibangun dari suatu proses pembersihan data, transformasi data dan integrasi data, pengambilan data dan penyegaran data yang dilakukan secara periodik.

Data warehouse biasanya dimodelkan dengan suatu struktur database multidimensional yang disebut *data cube*. Dimensi pada *data-cube* dapat bertingkat untuk memudahkan dalam mendapatkan rangkuman informasi dari tingkatan dimensi yang lebih luas atau umum sampai kepada tingkatan informasi yang lebih sempit atau khusus dengan operasi *roll-up* atau sebaliknya dengan operasi *drill-down*. Akan tetapi untuk mendapatkan informasi yang tidak diketahui secara eksplisit diperlukan satu tahap lagi yaitu aplikasi teknik *data mining.*

3. *Database Transactional*

Suatu *database transactional* terdiri dari sebuah *file* dimana tiap-tiap *record* mewakili suatu transaksi. Suatu transaksi umumnya mengandung suatu nomor identitas transaksi yang bersifat unik, dan suatu daftar *item-item* yang membentuk transaksi tersebut. Suatu *database transactional* dimungkinkan untuk mempunyai tabel-tabel tambahan yang saling terkait yang berisikan informasi-informasi yang saling berhubungan.

4. *Advance Database System*

Aplikasi *database* terbaru mampu untuk menangani data spesial (data peta), data desain (desain dari sebuah gedung), *hypertext* dan data multimedia (termasuk didalamnya data teks, gambar, video, dan audio) dan data pada *world wide web*. Aplikasi-aplikasi tersebut membutuhkan struktur data yang efisien dan metode-metode yang skalabilitas untuk menangani struktur objek yang komplek.

II.2.2.4 Teknik Data Mining dengan *Association Rule Mining*

Dari definisi *data mining* yang luas, terdapat banyak jenis teknik analisa yang dapat digolongkan dalam *data mining*. Teknik *data mining* yang akan digunakan dalam penelitian ini adalah *Association Rule Mining*.

Association rule mining adalah teknik *mining* untuk menemukan aturan asosiatif antara suatu kombinasi *item*. Pada mulanya digunakan untuk *Market Basket Analysis* dalam menemukan barang-barang yang dibeli oleh pelanggan secara berhubungan. Contoh dari aturan asosiatif dari analisa pembelian di suatu pasar swalayan adalah bisa diketahui berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut, pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu. Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, *support* yaitu persentase kombinasi item tersebut dalam database dan *confidence* yaitu kuatnya hubungan antar item dalam aturan asosiatif.

Adapun model data yang digunakan dalam *Association rule mining* adalah sebagai berikut :

a. $I = \{i_1, i_2, \ldots, i_m\}$: kumpulan item.

b. Sebuah transaksi t : t merupakan bagian dari item, $t \subseteq I$.

c. Transaksi Database T: kumpulan transaksi $T = \{t_1, t_2, \ldots, t_n\}$.

Secara umum, Association Rule Mining dapat dibagi menjadi dua tahap:

1. Pencarian *Frequent Itemset*

   Pada proses ini dilakukan pencarian *Frequent Itemset. Frequent Itemset* yang diperoleh harus memenuhi minimum support (*Itemset, Support, dan Confidence*).

2. *Rule Generation*

Frequent Itemset yang telah dihasilkan dari proses sebelumnya digunakan untuk membentuk *Association Rule*. *Association Rule* yang dihasilkan akan memenuhi minimum *support* dan *minimum confidence*.

Sedangkan perhitungan yang ada dalam *Association rule mining* yaitu : sebuah transaksi *t* yang berisi *X*, sekumpulan item (itemset) yang berada dalam *I*, jika $X \subseteq t$. Association rule adalah sebuah implikasi dari bentuk:

$$X \rightarrow Y, \text{ dimana } X, Y \subset I, \text{ dan } X \cap Y = \varnothing$$

- Itemset adalah sekumpulan item. Contohnya X = {milk, bread, cereal} ini sebuah itemset.
- *k*-itemset adalah sebuah itemset dengan *k* item. Contohnya, {milk, bread, cereal} itemset dengan 3 item.

Menurut Larose (2005) *support* dari suatu *association rule* adalah proporsi dari transaksi dalam *database* yang mengandung *X* dan *Y*, yaitu :

$$support = \frac{\text{jumlah transaksi yg mengandung } X \text{ dan } Y}{\text{jumlah total transaksi}}$$

Sedangkan *Confidence* dari *association rule* adalah ukuran ketepatan suatu *rule*, yaitu persentase transaksi dalam *database* yang mengandung *X* dan juga mengandung *Y*.

$$confidence = \frac{\text{jumlah transaksi yg mengandung } X \text{ dan } Y}{\text{jumlah transaksi yg mengandung } X}$$

Pada tahap *Frequent Itemset Candidate Generation* terdapat beberapa kendala yang harus dihadapi untuk memperoleh *Frequent Itemset.* Seperti banyaknya jumlah kandidat yang memenuhi *minimum support*, dan proses perhitungan *minimum support* dari *Frequent Itemset* yang harus melakukan scan database berulang-ulang. Pendekatan Apriori sangat membantu dalam mengurangi jumlah kandidat *Frequent Itemset*. Algoritma Apriori dikenal dengan *paradigma generate and test*, yaitu pembuatan kandidat kombinasi item

yang mungkin berdasar aturan tertentu lalu diuji apakah kombinasi item tersebut memenuhi syarat *support minimum*. Kombinasi item yang memenuhi syarat tersebut. disebut *frequent itemset*, yang nantinya dipakai untuk membuat aturan-aturan yang memenuhi syarat *confidence minimum*. Ciri dari algoritma Apriori adalah jika suatu *itemset* termasuk dalam *large itemset*, maka semua himpunan bagian (*subset*) dari *itemset* tersebut juga termasuk *large itemset* [Agrawal, 1994]. Tetapi dengan menggunakan *FP-growth*, kita dapat melakukan *Frequent Itemset Mining* tanpa melakukan *candidate generation*.

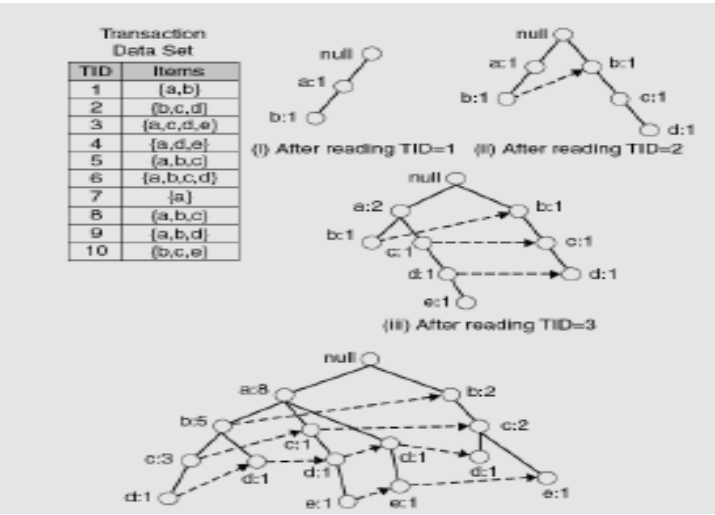
II.2.2.5 Pendekatan Algoritma *FP-Growth*

Dalam mendapatkan *frequent itemset*, algoritma *FP-Growth* melakukannya tanpa *candidate itemset generation*. Ada dua langkah yang dilakukan dalam pendekatan algoritma *FP-Growth* yaitu: [TAN, 2006]

Langkah 1: membangkitkan struktur data *tree* yang dikenal dengan *FP-tree* dimana scan *database* hanya dilakukan dua kali saja.

Langkah 2: mengekstrak *frequent itemsets* secara langsung dari *FP-tree*

Adapun struktur data yang ada pada *FP-tree* adalah sebagai berikut :



Gambar II.7 Struktur data pada *FP-tree* [TAN, 2006]

Penjelasan secara umum dari pembuatan struktur data pada *FP-tree* diatas adalah sebagai berikut :

1. Node sesuai dengan item dari data transaksi dan memiliki *counter*
2. *FP-Growth* membaca transaksi ke-1 dan memetakan kedalam *tree*
3. Dalam kasus ini, *counter* akan selalu bertambah
4. Pointer dibuat antara node yang berisi item sama (menyebabkan *single linked lists*)
5. Semakin banyak jalur yang dilalui maka kompresinya akan semakin tinggi
6. *Frequent itemset* diekstrak dari *FP-Tree*

Pada pendekatan *FP-Growth* saat mendapatkan *frequent itemset* akan melalui dua tahapan yaitu :

1. Tahap 1: *FP-Tree Construction*

Pada langkah ini akan dilakukan pemeriksaan *database* hanya dua kali saja yaitu:

Pertama :

a. *Scan* data dan temukan *support* untuk setiap item.
b. Item-item yang tidak *frequent (infrequent*) sudah tidak ada dalam *FP-tree*
c. Urutkan *frequent items* dalam *decreasing order* berdasarkan *support* yang dibuat

Kedua : membangkitkan *FP-tree*

a. Pembacaan transaksi ke-1 : {a, b}

   Buat 2 node a dan b, dan jalur menjadi null → a → b. Set count untuk a dan b menjadi 1.

b. Pembacaan transaksi ke-2 : {b, c, d}

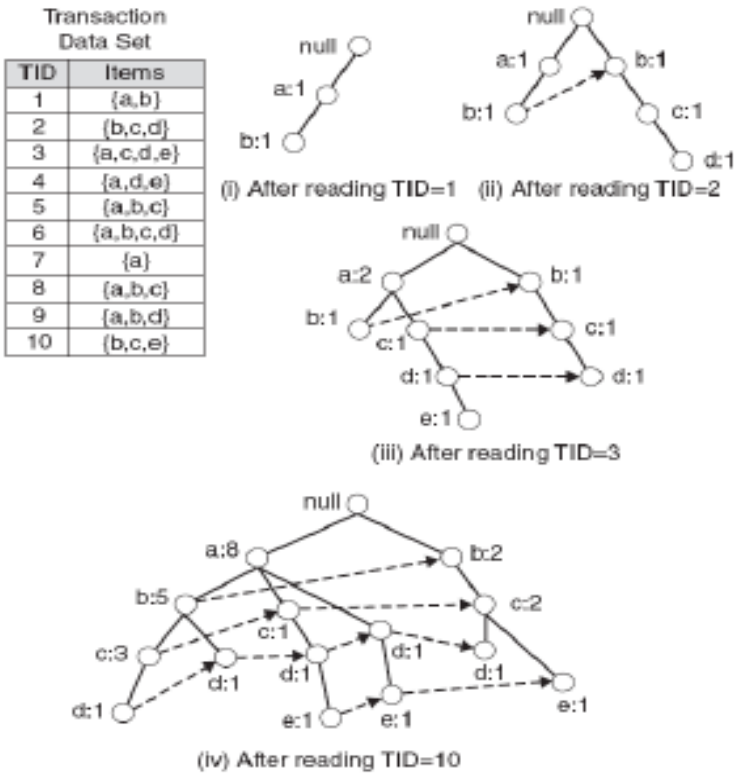
   Buat 3 node b, c and d, dan jalur menjadi null → b → c → d. Set count menjadi 1.

   Catatan : seluruh transaksi 1 dan 2 melibatkan item b sehingga tambahkan link antar item b

c. Pembacaan transaksi ke-3 : {a, c, d, e}

Akan menjadi *common prefix item* dengan transaksi ke-1 sehingga jalur untuk traksaksi 1 dan 3 akan menjadi *overlap* dan *count* untuk setiap node akan ditambahkan dengan 1. Tambahkan link antara c dan d

d. Lakukan hingga seluruh transaksi dipetakan dalam setiap jalur di *FP-tree*.



Gambar II.8 FP-tree yang telah dibangkitkan [TAN, 2006]

2. Tahap 2: *Frequent Itemset Generation*

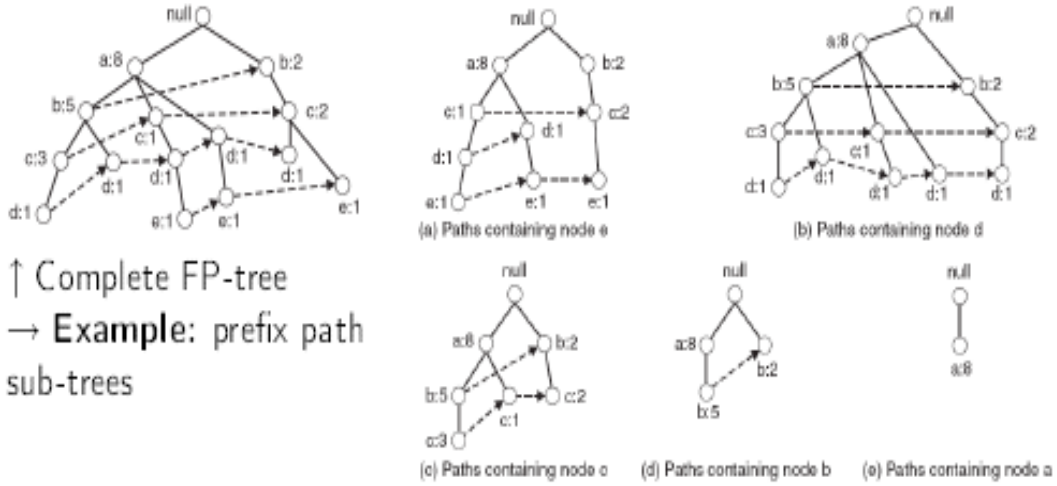
Pada tahapan ini ada beberapa hal yang dilakukan untuk menghasilkan *frequent itemset* yaitu:

a. *FP-Growth* mengekstrak *frequent itemsets* dari *FP-tree*.

b. Algoritma *Bottom-up* dari setiap *leaves* menuju *root*

c. Pertama, ekstrak *prefix path sub-trees* dalam setiap *item(set*)

d. Setiap *prefix path sub-tree* diproses secara rekursif untuk mengekstrak *frequent itemset*. Kemudian akan dilakukan penggabungan untuk mendapatkan *frequent itemset.*



Gambar II.9 Prefix path sub-trees dalam setiap item[TAN, 2006]

Algoritma *FP-growth* menemukan *frequent itemset* yang berakhiran *suffix* tertentu dengan menggunakan metode *divide and conquer* untuk memecah problem menjadi subproblem yang lebih kecil. Setelah mengetahui bahwa *item* e adalah *item* yang frequent, maka subproblem selanjutnya adalah menemukan *frequent itemset* dengan akhiran de, ce, be, dan ae. Dengan menggabungkan seluruh solusi dari subproblem yang ada, maka himpunan semua *frequent itemset* yang berakhiran *item* e akan didapatkan. Setelah memeriksa *frequent itemset* untuk beberapa akhiran (*suffix*), maka didapat hasil yang dirangkum dalam tabel berikut:

Tabel II.2. Hasil *Frequent Itemsets*

Suffix	*Frequent Itemset*
E	{e},{d,e},{a,d,e},{c,e},{a,e}
D	{d},{c,d},{b,c,d},{a,c,d},{b,d},{a,b,d},{a,d}
C	{c},{b,c},{a,b,c},{a,c}
B	{b},{a,b}
A	{a}

Dengan metode *divide and conquer* ini yaitu suatu metode yang digunakan untuk memecah problem menjadi subproblem yang lebih kecil, maka pada setiap langkah rekursif, algoritma FP-growth akan membangun sebuah *conditional FP-tree* baru yang telah

diperbaharui nilai *support count*, dan membuang lintasan yang mengandung item-item yang tidak *frequent* lagi.

II.2.2.6 Konsep Dasar Database

a. Pengertian Database

Basis Data (*database*) terdiri atas dua kata, yaitu Basis dan Data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili objek seperti manusia, barang, hewan, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar atau kombinasinya.

Database sendiri dapat didefinisikan dalam sejumlah sudut pandang sebagai berikut:

1. Himpunan kelompok data (arsip) yang saling berhubungan dan diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersamasama sedemikian rupa dan tanpa pengulangan yang tidak perlu untuk memenuhi berbagai kebutuhan.
3. Kumpulan file atau tabel atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.
4. Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data atau arsip dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data atau arsip tersebut.

b. Model-model Database

Database dipakai untuk aplikasi sederhana sampai aplikasi rumit yang melibatkan beberapa *user*. Ada tiga model database yang dibagi berdasarkan kompleksitasnya :

1. Database yang berdiri sendiri (*Stand-alone*)

Merupakan *database* yang sangat sederhana karena disimpan di sistem file lokal dan mesin *database* mengakses pada mesin yang sama. Desainer *database* tidak perlu

khawatir akan terjadinya konkurensi karena *database* hanya dipakai oleh satu aplikasi.

2. **Database terbagi** (*File Share*)

   Database ini hampir sama dengan *database stand-alone* tetapi dapat diakses oleh banyak user. Misalnya sebuah *database* karyawan dipakai oleh beberapa bagian. Jika satu bagian mengubah data, maka bagian lain juga dapat merasakan perubahan data tersebut. *Database* ini akan mengalami masalah jika aplikasi memerlukan banyak perhitungan dan pada saat pengaksesan secara bersamaan ke dalam *database*.

3. ***Database Client/Server***

   Merupakan *database* level tertinggi. *Database* ini memerlukan sebuah mesin khusus atau *server* yang dipakai untuk melayani sekelompok *user*.