BAB II
TINJAUAN PUSTAKA

2.1 *Data Mining*

Menurut Turban dalam bukunya yang berjudul *"Decision Support Systems and Intelligent Systems*", data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam basis data. *Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar [3].

Pengertian data *mining* menurut Han, Jiawei (2006) *data mining* merupakan pemilihan atau "menambang" pengetahuan dari jumlah data yang banyak. menurut Berry (2004) data mining adalah kegiatan mengeksplorasi dan menganalisis data jumlah yang besar untuk menemukan *pattern* dan *rule* yang berarti. Kemudian menurut Proscott, Hoffer dan McFadden (2005) *data mining* adalah penemuan pengetahuan dengan menggunakan teknik-teknik yang tergabung dari statistik, tradisional, *artificial intelligence* dan grafik komputer, dan *data mining* merupakan sebuah analisa dari observasi data dalam jumlah besar untuk menemukan hubungan yang tidak diketahui sebelumnya dan metode baru untuk meringkas data agar mudah dipahami serta kegunaannya untuk pemilik data (David Hand, 2001).Tahapan data mining menurut Jiawei Han (2006)

a. *Data cleaning*, untuk membersihkan data dari *noise data* dan data yang tidak konsisten
b. *Data integration*, megkombinasikan/mengintegrasikan beberapa sumber data.
c. *Data selection*, mengambil data-data yang relevan dari *database* untuk dianalisis
d. *Data transformation*, mentransformasikan *data summary* ataupun operasi agregasi
e. *Data mining*, merupakan proses yang esensial dimana metode digunakan untuk mengekstrak pola data yang tersembunyi

f. *Pattern evaluation*, untuk mengidentifikasi pola sehingga mereperesentasikan pengetahuan berdasarkan nilai-nilai yang menarik
g. *Knowledge presentation*, dimana teknik representasi dan visualisai data digunakan untuk mempresentasikan pengetahuan yang diadapat kepada *user*

2.1.1 Metodologi Data Mining

1. *Classification* adalah tindakan untuk memberikan kelompok pada setiap keadaan. Setiap keadaan berisi sekelompok atribut, salah satunya adalah class attribute. Metode ini butuh untuk menemukan sebuah model yang dapat menjelaskan *class attribute* itu sebagai fungsi dari *input attribute*.
2. *Clustering* adalah metode data mining yang *Unsupervised*, karena tidak ada satu atributpun yang digunakan untuk memandu proses pembelajaran, jadi seluruh atribut input diperlakukan sama. Kebanyakan Algoritma *Clustering* membangun sebuah model melalui serangkaian pengulangan dan berhenti ketika model tersebut telah memusat atau berkumpul (batasan dari segmentasi ini telah stabil)
3. *Association* juga disebut sebagai *Market Basket Analysis*. Sebuah problem bisnis yang khas adalah menganalisa tabel transaksi penjualan dengan mengidentifikasi produk-produk yang seringkali dibeli bersamaan oleh customer.
4. Metode *Regression* mirip dengan metode *Classification*, yang membedakannya adalah metode regression tidak bisa mencari pola yang dijabarkan sebagai class (kelas). Metoda *regression* bertujuan untuk mecari pola dan menentukan sebuah nilai numerik.
5. *Forecasting* adalah teknik perkiraan dengan mengambil sederetan angka yang menunjukkan nilai yang berjalan seiring waktu dan kemudian teknik forecasting ini akan menghubungkan nilai masa depan dengan menggunakan bermacam-macam teknik *machine-learning* dan teknik statistik yang berhubungan dengan musim, *trend*, dan *noise* pada data.

6. *Sequenc*e Anlysis digunakan untuk mencari pola pada serangkaian kejadian yang disebut dengan *Sequence.*

2.2 *Association Rules*

Analisis asosiasi atau *association rule mining* adalah teknik data mining untuk menemukan aturan assosiatif antara suatu kombinasi item. Contoh aturan assosiatif dari analisa pembelian di suatu pasar swalayan adalah dapat diketahuinya 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.

Analisis asosiasi menjadi terkenal karena aplikasinya untuk menganalisa isi keranjang belanja di pasar swalayan. Analisis asosiasi juga sering disebut dengan istilah *market basket analysis*. Analisis asosiasi dikenal juga sebagai salah satu teknik *data mining* yang menjadi dasar dari berbagai teknik *data mining* lainnya. Khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frequensi tinggi *(frequent pattern mining)* menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien.

Penting tidaknya suatu aturan assosiatif dapat diketahui dengan dua *parameter*, *support* (nilai penunjang) yaitu persentase kombinasi item tersebut dalam *database* dan *confidence* (nilai kepastian) yaitu kuatnya hubungan antar item dalam aturan assosiatif. Metodologi dasar analisis asosiasi terbagi menjadi dua tahap yaitu :

a. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam *database.* Nilai support sebuah item diperoleh dengan rumus berikut:

$$\text{Support (A)} = \frac{\sum \text{Transaksi Mengandung A}}{\sum \text{Total Transaksi}} \times 100\%$$

Gambar 2.1 Rumus Mencari Nilai *Support Item*

Sedangkan nilai support dari 2 itemset diperoleh dari rumus berikut

$$\text{Support } (A \cap B) = \frac{\sum \text{Transaksi Mengandung A dan B}}{\sum \text{Total Transaksi}} \times 100\%$$

Gambar 2.2 Rumus *Support* dan *Confidence*

b. Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan assosiatif yang memenuhi syarat minimum untuk confidence dengan menghitung confidence aturan assosiatif A →B. Nilai confidence dari aturan A →B diperoleh dari rumus :

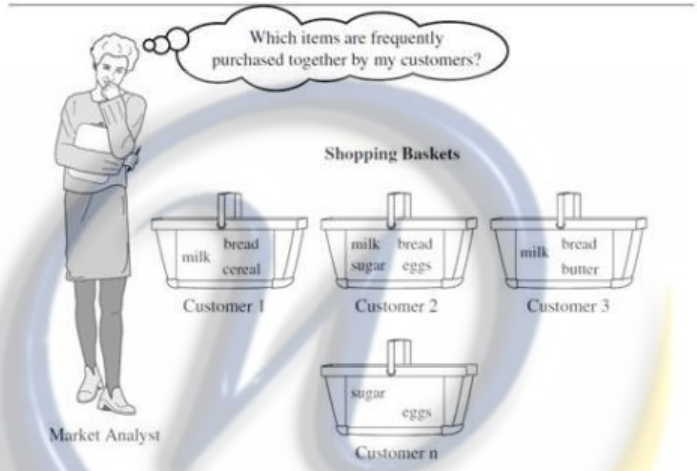
$$\text{Confidence} = P(B|A) = \frac{\sum \text{Transaksi mengandung A dan B}}{\sum \text{Transaksi mengandung A}} \times 100\%$$

Gambar 2.3 Rumus Menentukan Aturan Asosiatif

2.3 *Market Basket Analysis*

Menurut [5], fungsi *Association Rules* seringkali disebut dengan "*market basket analysis*", yang digunakan untuk menemukan relasi atau korelasi diantara himpunan item-item. *Market Basket Analysis* adalah Analisis dari kebiasaan membeli *customer* dengan mencari asosiasi dan korelasi antara item-item berbeda yang diletakkan *customer* dalam keranjang belanjaannya. Dari jumlah besar aturan yang mungkin

dikembangkan, perlu memiliki aturan-aturan yang cukup kuat tingkat ketergantungan antar item dalam *antecedent* dan *consequent.* Untuk mengukur kekuatan aturan asosiasi ini, digunakan ukuran *support* dan *confidence* seperti pada persamaan . *Support* adalah rasio antara jumlah transaksi yang memuat antecedent dan consequent dengan jumlah transaksi. *Confidence* adalah rasio antara jumlah transaksi yang meliputi semua item dalam antecedent dan consequent dengan jumlah transaksi yang meliputi semua item dalam *antecedent.*



Gambar 2.4 Konsep *Market Basket Analysis [*

2.4 Algoritma *Frequent Pattern-Growth*

FP-Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul *(frequent itemset)* dalam sebuah kumpulan data. Struktur data yang digunakan untuk mencari *frequent* itemset dengan algoritma *FP-Growth* adalah perluasan dari sebuah pohon *prefix*, yang biasa disebut *FP-Tree*. Cara kerja FP-Growth **:**

1. Tahap Pembangkitan *Conditional Pattern Base*

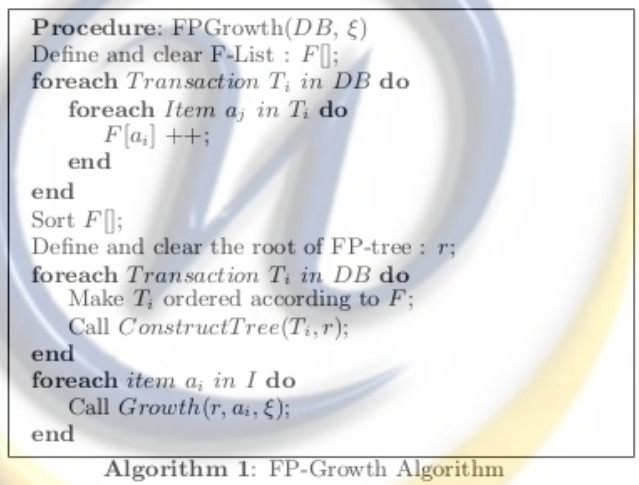
   Conditional Pattern Base merupakan subdatabase yang berisi *prefix path* (lintasan prefix) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui FP-tree yang telah dibangun sebelumnya.

2. Tahap Pembangkitan *Conditional FP-tree*

   Pada tahap ini, *support count* dari setiap item pada setiap *conditional pattern base* dijumlahkan, lalu setiap item yang memiliki jumlah support count lebih besar sama dengan *minimum support count* akan dibangkitkan dengan *conditional FP-tree.*

3. Tahap Pencarian *frequent* itemset.

   Apabila *Conditional FP-tree* merupakan lintasan tunggal *(single path),* maka didapatkan *frequent* itemset dengan melakukan kombinasi item untuk setiap *conditional FP-tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan FP-growth secara rekursif.



Procedure: FPGrowth(DB, ξ)
Define and clear F-List : $F[]$;
foreach *Transaction* T_i *in* DB **do**
  foreach *Item* a_j *in* T_i **do**
    $F[a_i]$ ++;
  end
end
Sort $F[]$;
Define and clear the root of FP-tree : r;
foreach *Transaction* T_i *in* DB **do**
  Make T_i ordered according to F;
  Call $ConstructTree(T_i, r)$;
end
foreach *item* a_i *in* I **do**
  Call $Growth(r, a_i, \xi)$;
end

Algorithm 1: FP-Growth Algorithm

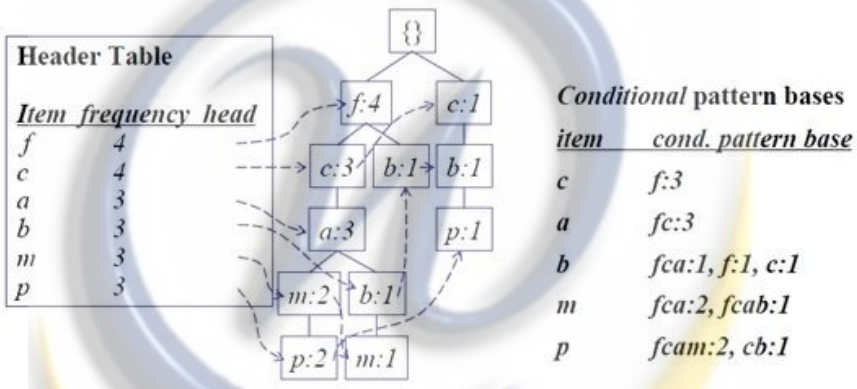
Gambar 2.5 *Pseudocode* Algoritma Fp-Growth

2.4.1 Pengertian FP-Tree

FP-Tree *(Frequent Pattern – Tree)* merupakan suatu algoritma yang dirancang untuk mengatasi kendala *bottleneck* pada proses penggalian data dengan algoritma Apriori (Zhao et al. 2003). Cara kerja algoritma ini adalah dengan memanfaatkan data dengan model struktur data pohon untuk menhindari pengulangan scanning database tanpa memerlukan *candidate generation,* kemudian dilanjutkan dengan proses algortima FP-Growth yang dapat langsung mengekstrak *frequent* itemset dari FP-Tree yang telah terbentuk dengan prinsip *divide* dan *conquer.*

FP-Tree Terdiri atas sebuah root dengan *label 'null',* sekumpulan subtree yang menjadi *child* dari *root* dan sebuah tabel *frequent header*. Setiap *node* dalam FP-Tree mengandung tiga informasi penting. yaitu :

1. *Label item*, yaitu yang menginformasikan jenis item yang direpresentasikan node tersebut,
2. *Support count* yaitu yang merepresentasikan jumlah lintasan transaksi yang melalui *node* tesebut,
3. *Pointer* adalah penghubung yang menghubungkan node-node dengan label item sama antar-lintasan, ditandai dengan garis panah putus-putus.



Gambar 2.6 Bagian FP-Tree

Untuk lebih jelasnya perhitungan algoritma fp-growth dapat dilihat pada contoh kasus yang dirujuk dari [4] di bawah ini :

1. Data yang digunakan adalah data transaksi sehingga dapat dipilih dataset untuk selanjutnya proses data mining.

Tabel 2.1 Tabel Dataset Transaksi Penjualan

Tanggal	No Transaksi	Nama Barang
10-Feb-13	21020130001	Gula
10-Feb-13	21020130001	Kopi
10-Feb-13	21020130001	tea
10-Feb-13	21020130002	Gula
10-Feb-13	21020130002	Kopi
10-Feb-13	21020130002	tea
10-Feb-13	21020130003	Gula
10-Feb-13	21020130003	Susu
10-Feb-13	21020130003	Roti
10-Feb-13	21020130004	Roti
10-Feb-13	21020130004	Gula
10-Feb-13	21020130004	Air
10-Feb-13	21020130005	Gula
10-Feb-13	21020130005	Susu
10-Feb-13	21020130005	Kopi

Data kemudian akan diintegrasikan masing-masing per nomor transaksi seperti pada tabel berikut :

Tabel 2.2 Data Transaksi Awal

No Transaksi	Barang
21020130001	Gula, Kopi, Tea
21020130002	Gula, Kopi, Tea
21020130003	Gula, Susu, Roti
21020130004	Roti, Gula, Air
21020130005	Gula, Susu, Kopi

Setelah mengintegrasikan masing-masing data tiap nomor transaksi kemudian adalah menentukan frekuensi setiap item dari transaksi secara keseluruhan.

Tabel 2.3 Frekuensi Item dari Seluruh Transaksi

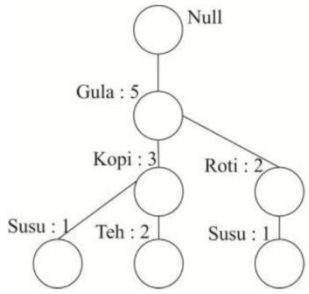
Nama Barang	Jumlah
Gula	5
Kopi	3
Tea	2
Susu	2
Roti	2
Air	1

Setelah mengetahui frekuensi dari masing-masing item langkah selanjutnya adalah menentukan minimum support, jika jumlah *item* kurang dari jumlah *minimum support* yang ditentukan maka item tersebut akan dihapus. Angka minimum support yang ditentukan misalnya Φ 2, karena *item* air jumlah frekuensinya kurang dari *min support* yang ditentukan, makan *item* air dihapuskan.

Tabel 2.4 Dataset Setelah Difilter

No Transaksi	Barang
21020130001	Gula, Kopi, Tea
21020130002	Gula, Kopi, Tea
21020130003	Gula, Susu, Roti
21020130004	Roti, Gula
21020130005	Gula, Susu, Kopi

Langkah selanjutnya adalah membuat FP-Tree dari data transaksi yang telah difilter



Gambar 2.7 FP-Tree dari 5 Transaksi

Setelah FP-Tree terbentuk langkah selanjutnya adalah pembangkitan *conditional pattern base*, *Conditional FP-Tree*, dan *Frequent Item Set* tahap ini adalah tahap pencarian frequent item set dengan melihat struktur FP-Tree yang telah digambarkan pada gambar 2.7.

Tabel 2.5 *Conditional Pattern Base*

Item	*Conditional Pattern Base*
Tea	{Gula, Kopi : 2}
Susu	{{Gula, Kopi : 1}, { Gula, Roti : 1}}
Roti	{Gula:2}
Kopi	{Gula:3}

Tabel 2.6 Conditional FP-Tree

Item	*Conditional FP-Tree*
Tea	{{Gula:2}, {Kopi:2}, {Gula, Kopi:2}}
Susu	{Gula:}
Roti	{Gula:2}
Kopi	{Gula:3}

Tabel 2.7 *Frequent Item Set*

Item	*Frequent Item Set*
Tea	{{Gula, Tea:2}, {Kopi, Tea:2}, {Gula, Kopi, Tea:2}}
Susu	{Gula, Susu:2}
Roti	{Gula, Roti:2}
Kopi	{Gula, Kopi:3}

Langkah selanjutnya adalah tahap pembentukan association rule dengan menentukan nilai support dan confidence pada setiap itemset dengan rumus perhitungan pada gambar 2.3

Tabel 2.8 Hasil *Association Rules*

Rule	*Support*	*Confidence*
Roti=> Gula	0.4	1
Susu=>Gula	0.4	1
Tea=>Gula	0.4	1
Tea=>Kopi	0.4	1
Tea=>Gula, Kopi	0.4	1
Kopi=>Gula	0.4	1

2.5 *Recommender System*

Dijelaskan pada penelitian [6] *Recommender system* merupakan bagan pengolah informasi yang dimaksudkan untuk mempresentasikan informasi yang mungkin diminati oleh pengguna pada masa kini. Pada saat ini *recommender system* sudah banyak dijumpai disetiap aplikasi seperti facebook, instagram, *twitter*, *recommender* pada aplikasi diatas berguna untuk merekomendasikan penggunanya untuk menambah teman yang mungkin kita kenali atau biasa dikenal dengan *people you may know*, biasanya aplikasi tersebut merekomendasikan *account* lain apa yang kita *follow*.

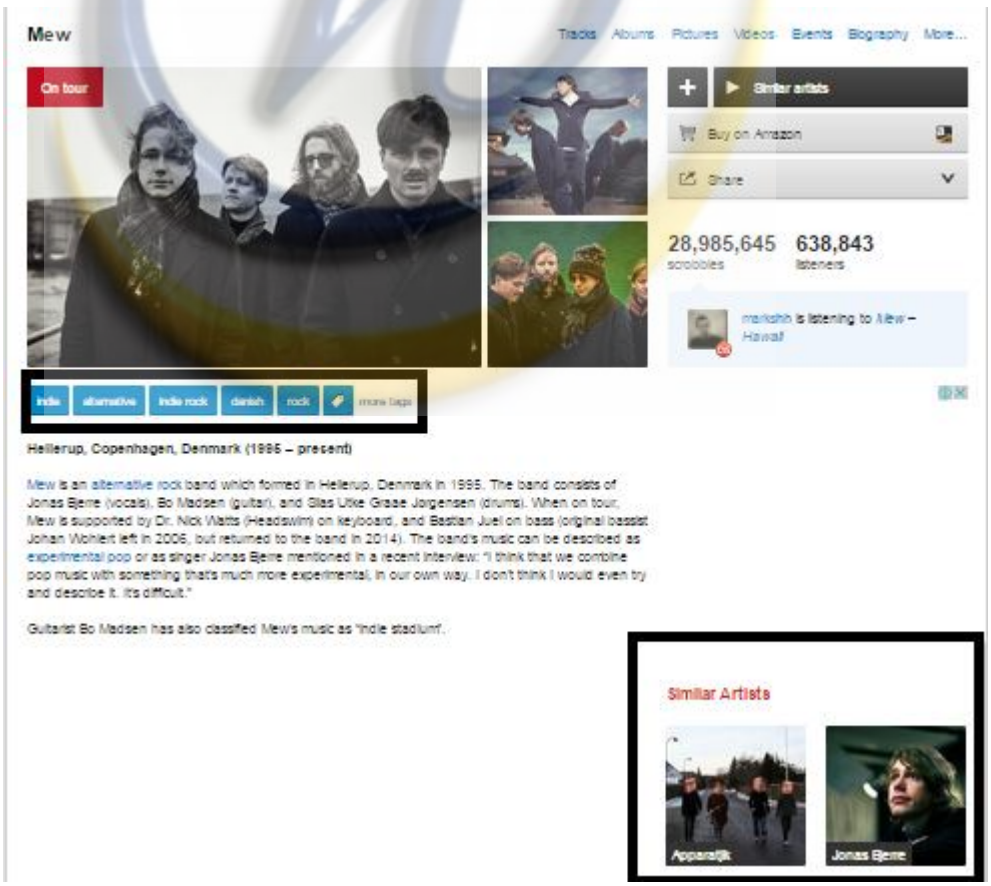
Recommender system dapat dikatakan sebagai perangkat lunak, atau teknik yang berfungsi untuk memberi saran kepada *user* tentang item apa yang sebaiknya digunakan atau dipilih". Saran ini berkaitan dengan bermacam-macam proses pengambilan keputusan seperti musik apa yang sebaiknya didengarkan, item apa yang sebaiknya dibeli atau buku apa yang sebaiknya dibaca [10]. Teknik rekomendasi sistem teridiri dari :

1. *Content Based Recommender*

   Content based recommender adalah konten yang memberikan rekomendasi untuk item berdasarkan pengguna saat barang permintaan / kebutuhan informasi itu sendiri dan juga profil pengguna, jika ada. Karakteristik pengguna dikumpulkan dari waktu ke waktu dan diprofilkan secara otomatis berdasarkan umpan balik sebelum pengguna dan pilihan. Oleh karena sistem tersebut tidak hanya mengambil informasi yang berkaitan dengan

item saat ini, juga mencoba untuk memastikan bahwa rekomendasi diambil sesuai dengan preferensi pengguna. Pendekatan berbasis konten untuk rekomendasi berakar dalam pencarian informasi (IR) masyarakat klasifikasi, dan mempekerjakan banyak teknik yang sama. Masalah recommender dapat dinyatakan sebagai memperpanjang masalah teks kategorisasi menggunakan classifier seperti Naïve Bayes. Pelatihan set terdiri dari item yang pengguna menemukan menarik. Barang-barang ini membentuk contoh pelatihan yang semua memiliki atribut. Atribut ini menetapkan kelas item berdasarkan baik rating pengguna atau bukti implisit.

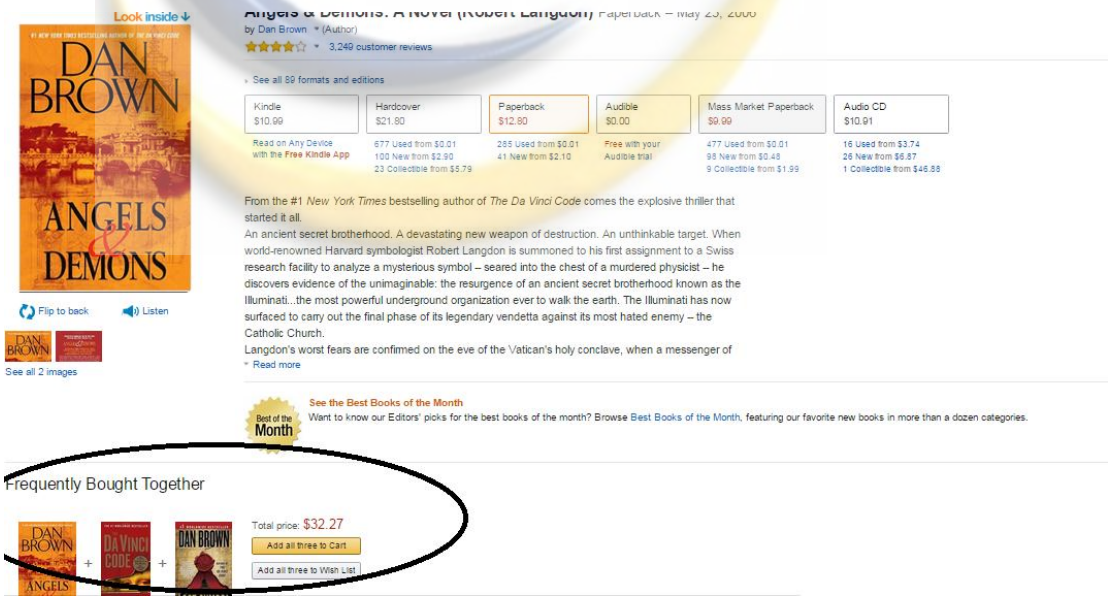
Teknik rekomendasi ini dapat kita jumpai pada beberapa situs ternama seperti salah satu contohnya adalah last.fm. Last.fm merupakan situs penyedia database music terbesar. Ketika pengguna mengetikan artist yang dimaksud, maka pada kolom bawah site akan direkomendasikan beberapa *similar artist*. Perekomendasian berdasarkan music tag.



Gambar 2.8 *Recommender System* pada web lastfm.com

2. *Collaborative Filtering*

Kolaborasi Filtering adalah metode berbasis lingkungan sosial rekomendasi digunakan untuk mengusulkan item yang berpikiran pengguna mendukung (dan pengguna aktif belum terlihat). Rekomendasi ini sesuai dengan kebutuhan pengguna berdasarkan informasi yang dikumpulkan dari waktu ke waktu dari orang-orang lain yang memiliki kepentingan yang cocok dengan pengguna saat ini. Pendekatan ini memberikan rekomendasi berdasarkan korelasi antara pengguna. Kolaborasi Penyaringan adalah poros sistem hari *recommender modern. Collaborative Filtering* efektif sejak selera orang biasanya tidak ortogonal. Skema Kolaborasi Penyaringan bertujuan untuk membuat saran kepada pengguna berdasarkan / nya kesukaan sebelumnya dan juga preferensi yang berpikiran pengguna yaitu pengguna jatuh ke setara kategori / kelompok / komunitas sebagai pengguna saat ini. Situs yang menerpakan sistem *collaborative filtering* salah satunya adalah amazon.com



Gambar 2.9 *Collaborative Filtering* pada Amazon.com

3. *Hybrid Systems*

   Kekurangan dari *Collaborative Filtering* dan pendekatan konten berdasarkan dapat diselesaikan dengan menggabungkan dua menjadi metode *hybrid.* Banyak pendekatan *hybrid* menggunakan dua algoritma rekomendasi dan menggabungkan hasil mereka dalam beberapa cara, seperti menggabungkan hasil dengan relevansi mereka, pencampuran *output* dari dua algoritma, beralih dari CB ke CF setelah fase dingin-*star*t berakhir, atau menggunakan *output* satu algoritma sebagai masukan untuk algoritma kedua.

   Sistem rekomendasi *Hybrid* membantu mengatasi masalah yang disebutkan dalam pendekatan di atas dan dapat menghasilkan output yang mengungguli sistem komponen tunggal dengan menggabungkan beberapa teknik ini. Metodologi hibridisasi paling umum adalah menggabungkan teknik yang berbeda dari berbagai jenis, misalnya, pencampuran pendekatan berbasis konten dan penyaringan masyarakat. Hal ini juga memungkinkan untuk mencampur teknik yang berbeda dari jenis yang sama, seperti naif Rekomendasi Berdasarkan *Content* berdasarkan Bayes ditambah berdasarkan kNN *Collaborative Filtering*

2.6 Pemodelan Sistem [8]

Terdapat tiga alasan kenapa perlunya pemodelan sistem

1. Dapat melakukan perhatian pada hal penting dalam sistem meskipun tidak terlibat terlalu jauh
2. Untuk mendiskusikan perubahan dan koreksi terhadap kebutuhan pemakai
3. Menguji pengertian penganalisa sistem terhadap kebutuhan pemakai.

Sedangkan perangkat untuk memodelkan sistem tersebut diantaranya:

1. *Context Diagram*
2. *Data Flow Diagram*

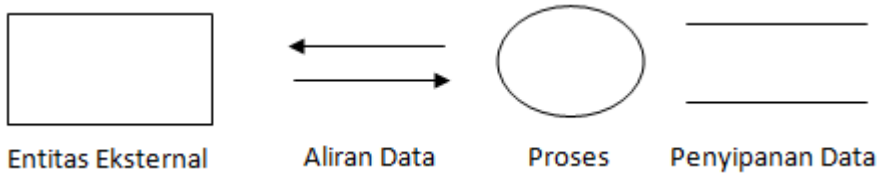
2.6.1 *Context Diagram*

Context diagram merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses yang menunjukkan sistem secara keseluruhan. Dalam context diagram digambarkan semua entitas eksternal berikut aliran-aliran datanya menuju dan dari sistem. Dalam diagram ini tidak memuat penyimpanan data dan ditampilkan dalam bentuk yang paling sederhana. Context diagram menggarisbawahi sejumlah karakteristik penting dari suatu sistem yaitu:

1. Kolompok pemakai, organisasi, atau sistem lain dimana sistem kita melakukan komunikasi disebut juga sebagai Terminator.
2. Aliran data dapat berupa data yang diterima sistem dari lingkungan luar, kemudian diproses dengan cara tertentu. Atau data yang dihasilkan oleh sistem untuk diberikan kepada lingkungan luar.
3. Penyimpanan data yang digunakan secara bersama antara sistem dengan terminator.
4. Batasan antara sistem dan lingkungan.

2.6.2 Data Flow Diagram

Data flow diagram (DFD) merupakan diagramyang digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir dan lingkungan fisik dimana data tersebut disimpan. Adapun komponen-komponen dalam DFD menurut Yourdan dan De Marco :



Gambar 2.10 Komponen DFD Menurut Yourdan dan De marco

1. Entitas Eksternal (External Entity)

   Entitas Eksternal (entity) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem.

2. Aliran data

   Aliran data mengalir diantara proses (process), simpanan data (data store) dan kesatuan luar (External entity). Aliran data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. Proses

4. Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu aliran data yang masuk ke dalam proses untuk dihasilkan aliran data yang akan keluar dari proses.

5. Penyimpan Data (Data Store)

   Penyimpan data (data store) merupakan penyimpan data yang dapat berupa:

   a. Suatu file atau basis data di sistem komputer.
   b. Suatu arsip atau catatan manual.
   c. Suatu tabel acuan manual.
   d. Suatu agenda atau buku.

2.7 *Extract, Transform, Loading Data* (ETL)

Menurut Rainardi (2008), ETL adalah suatu proses mengambil dan mengirim data dari sumber data ke *data warehouse*. Dalam proses pengambilan data, data harus bersih agar didapat kualitas data yang baik. Contohnya ada nomor telepon yang *invalid*, ada kode buku yang tidak eksis lagi, ada beberapa data yang *null*, dan lain sebagainya. Pendekatan tradisional pada proses ETL mengambil data dari data sumber, meletakkan pada *staging area*, dan kemudian mentransformasi dan meng-*load* ke *data warehouse*.

Proses ETL (Extract, Transform, Loading) terbagi menjadi 3, yaitu:

1. *Extract*

   Extract adalah proses penentuan *source* yang akan digunakan sebagai sumber data bagi *data warehouse*. Di sini kita bias menentukan data apa saja yang diperlukan, tabel apa saja yang dijadikan sumber. Langkah pertama pada proses ETL adalah mengekstrak data dari sumber-sumber data. Kebanyakan proyek *data warehouse* menggabungkan data dari sumber-sumber yang berbeda. Pada hakekatnya, proses ektraksi adalah proses penguraian, pembersihan dari data diekstrak untuk mendapatkan struktur atau pola data yang diharapkan.

2. *Transform*

   Setelah *source* ditentukan, maka data tersebut diubah agar sesuai dengan *standard* yang ada pada *data warehouse*. Tahapan transformasi menggunakan serangkaian aturan atau fungsi untuk mengekstrak data dari sumber dan selanjutnya dimasukkan dalam *data warehouse*. Dibawah ini hal-hal yang dilakukan dalam tahapan transformasi, yaitu:

   a. Hanya memilih kolom tertentu saja untuk dimasukkan ke dalam *data warehouse*.
   b. Menterjemahkan nilai-nilai yang berupa kode, misalnya sumber *database* menyimpan nilai 1 untuk laki-laki dan nilai 2 untuk perempuan, tetapi *data warehouse* yang telah ada menyimpan A untuk dewasa dan C untuk anak-anak, maka ini disebut juga dengan *automated data cleaning* (tidak ada pembersihan secara manual yang ditunjukkan selama proses ETL).
   c. Melakukan perhitungan nilai-nilai baru seperti total = h_jual * qty.

d. Mengkodekan nilai-nilai ke dalam bentuk bebas seperti memetakan yang umurnya “65” ke kategori “D”.
e. Membuat ringkasan dari sekumpulan baris data seperti total penjualan untuk setiap toko atau setiap bagian.
f. Menggabungkan data secara bersama-sama dari berbagai sumber.

3. *Load*

Load adalah proses memasukkan data-data yang sudah di transformasi ke dalam *data warehouse* untuk disimpan sebagai *summary* atau *archieve*. Fase *load* merupakan tahapan yang berfungsi untuk memasukkan data ke dalam target akhir, yang biasanya ke dalam suatu *data warehouse*. Jangka waktu proses ini tergantung pada kebutuhan organisasi. Beberapa *data warehouse* dapat setiap minggu menulis keseluruhan informasi yang ada secara kumulatif, data diubah, sementara *data warehouse* yang lain satau bagian lain dari *data warehouse* yang sama dapat menambahkan data baru dalam suatu bentuk *historical*, contohnya setiap jam. Waktu dan jangkauan untuk mengganti atau menambah data tergantung dari perancangan *data warehouse* pada waktu menganalisis keperluan informasi.