BAB II

LANDASAN TEORI

2.1 *Information Retrieval dan Information Filtering*

Perkembangan internet yang sangat pesat secara tidak langsung memberikan kemampuan kepada pengguna untuk memilih di antara banyak dari informasi yang ada di internet. Informasi ini bisa berkaitan dengan profesi mereka, kejadian yang ada di dunia, atau bahkan informasi yang berkaitan dengan gaya hidup. Informasi yang dibutuhkan oleh pengguna internet terus mengalami peningkatan dan ini bisa datangnya dari berbagai sumber yang berbeda-beda. Misalnya melalui halaman web, email, artikel, berita, jurnal, situs belanja, dan situs multimedia. Perkembangan ini memicu terjadinya *information explosion* di dunia maya yang menyulitkan pengguna dalam mencari suatu informasi dengan cepat dan relevan. Masalah ini menjadi alasan dikembangkannya beberapa teknik untuk *information retrieval* dan *information filtering* (Mortensen, Magnus. 2007).

2.1.1 *Information Retrieval*

Information Retrieval (IR) sering sekali dikaitkan dengan pencarian data, merupakan teknologi yang mencakup *crawling, processing* dan *indexing* konten, serta *querying* untuk konten. *Crawling* adalah tindakan mengakses *web server* dan *file system* untuk mendapatkan informasi. Dengan mengikuti *link* yang ada, pengguna dapat melintasi hirarki konten *web* berdasarkan sebuah URL *(Uniform Resource Locator)*. Pada tahapan *processing* memungkinkan pengguna untuk melakukan penambahan, menghapus, dan mengubah informasi menjadi sebuah dokumen, seperti menambah informasi meta baru untuk *laguange processing* atau menggali informasi tentang bahasa yang tertulis pada dokumen tersebut. *Indexing* sendiri merupakan proses untuk memeriksa konten yang telah diperiksa pada tahap *processing* dan membuat sebuah struktur data yang dicari, disebut dengan *index*, dimana *index* berisi

tentang referensi konten. *Querying* adalah tahapan untuk melakukan *request* informasi. IR memperbolehkan pengguna untuk menulis *query* dalam bentuk *keyword* yang menggambarkan informasi yang dibutuhkan oleh pengguna. Disini pengguna dapat berinteraksi dengan *query interface*, lalu sebuah *query-processor* akan menggunakan *index* untuk mencari referensi informasi berdasarkan *keyword* dan kemudian menampilkan referensi yang ada. Tujuannya adalah untuk menganalisa dan mengidentifikasi tujuan dari penggunaan query dan memberikan hasil yang paling relevan.

Penyaringan informasi pada sistem IR dilakukan dengan memberikan akses bagi pengguna untuk menentukan informasi apa yang dibutuhkan dengan menulis secara manual *keyword* yang menggambarkan informasi yang ingin dicari. IR cukup sukses untuk mendukung pengguna yang tahu bagaimana menjelaskan dengan tepat apa yang mereka cari dengan cara yang kompatibel dengan mendeskripsi konten yang diciptakan selama *indexing* (Mortensen, Magnus. 2007).

2.1.2 *Information Filtering*

Information Filtering (IF) berfokus pada penyaringan informasi berdasarkan profil dari *user*. Profil *user* dapat dibentuk dengan membiarkan pengguna menentukan dan menggabungkan minat secara eksplisit, atau dengan membiarkan sistem implisit memonitor prilaku pengguna. Keuntungan IF adalah kemampuannya beradaptasi dengan minat jangka panjang dari pengguna. Sistem ini dibangun atas gagasan untuk memiliki sistem yang mampu memberikan panduan kepada pengguna untuk memilih sesuatu informasi yang bersifat personal atau sesuai dengan selera pengguna. Sistem yang merealisasikan ide ini disebut dengan *recommender system* Mortensen, Magnus. 2007).

2.2 *Recommender System* (Sistem Rekomendasi)

Recommender System merupakan sebuah aplikasi *information filtering* untuk mencari dan memberikan rekomendasi *item* berupa informasi, produk, atau layanan kepada *user* berdasarkan prediksi yang bersifat personal (Sarwar, Badrul. 2001). Pengembangan *recommender system* oleh berbagai *online vendor* merupakan langkah

untuk menarik lebih banyak perhatian pengguna dan meningkatkan kepuasan pengguna terhadap hasil pencarian informasi secara *online*. Pada *e-commerce* misalnya, dimana sistem rekomendasi digunakan secara luas untuk menyarankan produk kepada pelanggan dan untuk menyediakan pelanggan dengan informasi sehingga membantu pelanggan memutuskan pilihan produk yang akan dibeli (Mortensen, Magnus. 2007). Hal ini menjadi sangat penting bagi keberhasilan industri di bidang teknologi informasi dan *e-commerce* saat ini yang secara bertahap mendatangkan keuntungan dari segi popularitas di berbagai aplikasi, misalnya proyek Netflix, Google news, dan Amazon.

Sistem rekomendasi dibangun dengan tujuan membantu *user* untuk memilih *item-item* yang disukainya dari sekian banyak *item* yang tersedia. Teknik pencari *item* yang akan direkomendasikan dapat dilakukan berdasarkan kemiripan, bisa berupa kemiripan suatu *item* dengan *item* lainnya, berdasarkan konten atau kemiripan selera suatu *user* dengan *user* lain berdasarkan *rating* yang diberikan pada *item*.

2.2.1 Teknik Pengumpulan Data *Recommender System*

Dalam pengumpulan data yang akan digunakan dalam sistem rekomendasi dapat dilakukan dengan dua cara (Sanjung, Ariyani. 2011) yaitu secara eksplisit dan implisit :

1. Secara eksplisit dapat dilakukan misalnya dengan meminta *user* untuk me*rating* sebuah *item* dari *range* tertentu, memberikan ranking berdasarkan dari *item* yang disukai dan yang tidak disukai oleh *user*, ataupun dengan meminta *user* untuk melakukan *list* terhadap *item-item* yang disukai.
2. Implisit dilakukan dengan mengamati pola kecenderungan *user* (*behavior user*) melalui observasi *social network* atau dengan mengamati *item-item* yang sering dilihat atau didengar oleh *user*. Dengan kata lain, jika secara eksplisit *user* memberikan penilaian preferensi terhadap *item* secara langsung, sedangkan implisit tidak secara langsung.

2.2.2 Metode – Metode *Recommender System*

Berdasarkan pendekatan yang digunakan untuk menghasilkan rekomendasi, *recommender system* dapat diklasifikasikan dalam tiga kelompok utama (Leimstoll, U., Stormer, H. 2007) adalah sebagai berikut :

1. *Content Based Recommender System.*

   Metode *content based* membuat rekomendasi dengan menganalisa deskripsi setiap *item* untuk mengidentifikasi *item* mana yang mempunyai hubungan ketertarikan khusus dari seorang *user*. Deskripsi ketertarikan *user* diperoleh dari profil *user* yang didasarkan atas penilaian menarik atau tidaknya suatu *item* yang ada pada *recommender system* ini (Pazzani, Michael J. & Billsus, Daniel. 2007). Pada data *item* film misalnya, deskripsi konten bisa berupa *genre,* aktor yang berpartisipasi, sutradara, dan sebagainya.

2. *Collaborative Filtering Recommender System*.

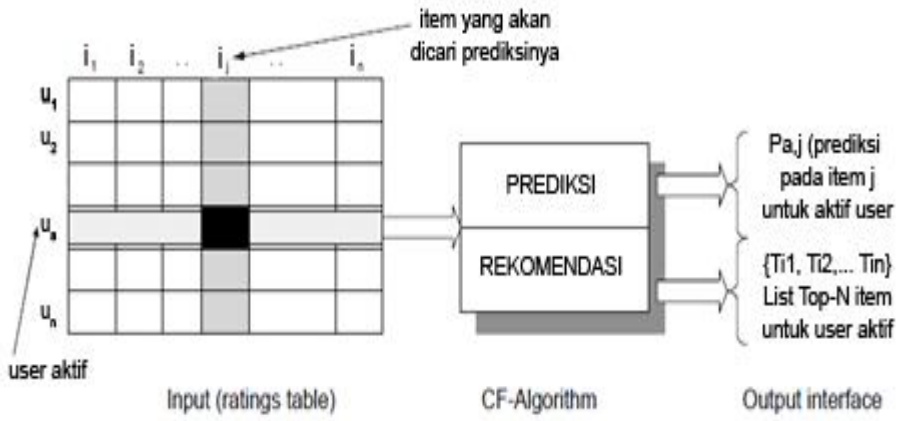
   Pendekatan *collaborative filtering* merekomendasikan *item* kepada *user* berdasarkan korelasi antara *rating user* yang berbeda terhadap *item* atau transaksi *user* di dalam sistem. Sementara *user* me*rating item*, sistem akan menggunakan data *rating user* tertentu untuk melakukan perhitungan prediksi dan memberikan rekomendasi *item* yang belum pernah di*rating* oleh *user* tersebut.

3. *Hybrid Recommender System*.

   Secara umum pendekatan *hybrid recommendations* adalah dengan menggabungkan lebih dari satu metode yang ada pada *recommender system*, kombinasi yang ada pada teknik ini misalnya dengan menggabungkan metode *content based* dengan *collaborative filtering* (Leimstoll, U., Stormer, H. 2007).

2.3 Metode *Collaborative Filtering*

Collaborative filtering (CF) adalah teknologi *recommender system* yang paling sukses dan populer saat ini, serta penggunaan CF sangat sukses untuk berbagai recommender system yang ada di internet. Teknik ini menggunakan teknik statistik untuk menemukan satu set *user* yang dikenal sebagai *neighbors*, dimana setiap *user* memiliki kesamaan minat dan pendapat dengan target *user* (yaitu, mereka memiliki beberapa *rating item* yang sama atau kecenderungan *user* menyukai *item* yang sama). Setelah lingkungan *neighbors* terbentuk, sistem ini akan menggunakan beberapa algoritma untuk menghasilkan rekomendasi.



Gambar 2.1. Proses *Collaborative Filtering* (Sarwar, Badrul. 2001).

Dalam skenario CF terdapat daftar pengguna m *user* U = {u1, u2,…, um} dan daftar *item* I = {p1, p2,…, pn}. Setiap ui *user* mengekspresikan pendapatnya tentang daftar *item* miliknya. Kumpulan set dari pendapat itu disebut dengan *rating* dari *user* ui dan dilambangkan dengan Iui. Setelah sistem ini menentukan ketetanggaan terdekat, maka sistem akan merepresentasikan *item* yang mungkin disukai *user* dalam dua bentuk, yaitu:

1. Prediksi, merupakan nilai numerik dimana Pa,j adalah nilai prediksi *rating item* j yang mungkin disukai oleh *active user* (Ua). Nilai prediksi ini digunakan dengan skala yang sama dengan nilai yang disediakan (misalnya, dari skala 1 sampai 5).

2. Rekomendasi adalah daftar N *item* yang mungkin akan disukai oleh *user* Ua. Daftar yang direkomendasikan biasanya terdiri dari *item* yang belum pernah dibeli atau di*rating* oleh *active user*. Output dari algoritma CF ini juga dikenal sebagai *Top-N Recommendation.*

Gambar 2.1 menunjukkan diagram skema dari proses *collaborative filtering*. Algoritma CF merepresentasikan seluruh m x n *user-item* sebagai matriks *rating* dimana setiap entri merupakan nilai *rating* dari *user* untuk setiap *item*. *Active user* (Ua) pada skema ini merupakan *user* yang akan dicari *item* yang mungkin disukainya dengan menggunakan algoritma CF (Sarwar, Badrul. 2001).

2.3.1 Pendekatan Berbasis *Collaborative Filtering*

Terdapat dua bentuk pendekatan berbasis *collaborative filtering*, yaitu *user-based* CF dan *item-based* CF.

1. *User-based* CF mengasumsikan bahwa cara yang baik untuk menemukan *item* yang menarik bagi *user* tertentu adalah dengan mencari *user* lain yang memiliki minat yang sama. Jadi, pada awalnya *user-based* CF mencoba untuk menemukan *user neighbor* berdasarkan pada *user similarity* dan kemudian setiap nilai *rating* dari *user neighbor* yang telah ditentukan sebelumnya.
2. Algoritma *Item-based* CF memiliki skema yang hampir sama dengan *user-based* CF, jika sebelumnya *user-based* CF yang dicari adalah korelasi antara *user*, maka pada *item-based* CF korelasi yang dicari adalah antar *item* yang disukai oleh *user* kemudian *item* yang berkorelasi tersebut direkomendasikan terhadap sejumlah *user* lainnya.

2.3.2 Konsep Dasar *Collaborative Filtering*

Collaborative filtering melakukan penyaringan data berdasarkan kemiripan karakteristik *user* sehingga mampu memberikan informasi yang baru kepada *user* karena sistem memberikan informasi berdasarkan pola satu kelompok *user* yang hampir sama. Perbedaan minat pada beberapa anggota kelompok menjadikan sumber informasi baru yang mungkin bermanfaat bagi anggota kelompok lainnya.

Secara umum proses pemberian rekomendasi terdiri atas tiga langkah, yaitu: penemuan *similar user*, pembuatan ketetanggaan (*neighbour*), dan penghitungan prediksi berdasarkan tetangga yang dipilih. *Collaborative filtering* menghasilkan prediksi atau rekomendasi bagi pengguna atau pelanggan yang dituju terhadap satu *item* atau lebih. *Item* dapat terdiri atas apa saja yang dapat disediakan manusia seperti misalnya buku, film, seni, artikel, atau tujuan wisata. *Rating* dalam *collaborative filtering* dapat berbentuk dari model-model sebagai berikut:

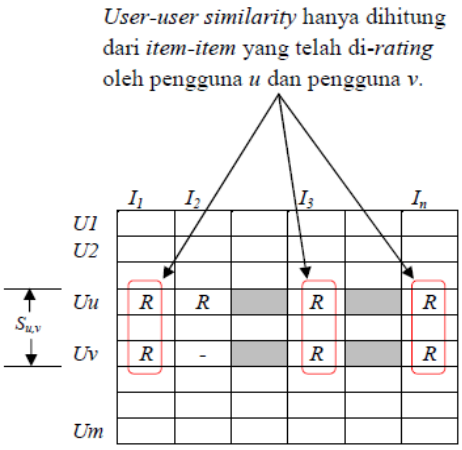
a) Model *rating* skalar yang terdiri atas *rating* numerik seperti 1 (satu) sampai 5 (Lima).
b) Model *rating* biner dengan memilih antara setuju atau tidak setuju, atau dapat pula baik atau buruk.

2.4 Metode *User-Based Collacborative Filtering*

User-based Collaborative Filtering menggunakan teknik statistika untuk menemukan sekumpulan pengguna, dikenal sebagai tetangga (neighbour), yang memiliki sejarah setuju dengan pengguna yang menjadi sasaran. Setelah sekumpulan tetangga terbentuk, sistem menggunakan algoritma yang berbeda untuk menggabungkan kesukaan *neighbours* untuk menghasilkan prediksi atau rekomendasi N-teratas untuk user aktif. (Sarwar, Badrul. 2001).

Pada pendekatan *user based collaborative filtering* sistem memberikan rekomendasi kepada *user item-item* yang disukai atau *dirating* oleh *user-user* lain yang memiliki banyak kemiripan dengannya. Misalnya, *user* a menyukai atau *merating item* 1, 2 dan 3, kemudian *user* b menyukai *item* 1, 2 dan 4 maka sistem akan merekomedasikan *item* 3 kepada *user* b dan *item* 4 kepada *user* a.

Kelebihan dari pendekantan *user based collaborative filtering* adalah dapat menghasilkan rekomendasi yang berkualitas baik. Berikut merupakan skema *user based collaborative filtering*.



Gambar 2.2 Pola *user based collaborative filtering* (Sarwar, Badrul. 2001).

2.4.1 Algoritma *K-Means Clustering*

Algoritma yang digunakan pada proses pembentukan *user cluster* adalah algoritma *K-means.* Jumlah k merupakan *input* yang digunakan algoritma untuk menentukan seberapa banyak jumlah *cluster* yang ingin dibentuk. Adapun langkah-langkah pada *K-means clustering* adalah sebagai berikut (Xue, Gui-Rong. 2005) :

1. Tentukan beberapa k *user* yang dijadikan sebagai *centroid*.
2. Setiap *user* yang tidak menjadi *centroid* dibandingkan ke *centroid* terdekat berdasarkan nilai *similarity*.
3. Kalkulasi ulang nilai *centroid* berdasarkan rata-rata nilai kumpulan *user* di setiap *cluster* yang terbentuk.
4. Lakukan proses pembentukan ulang *cluster* dengan nilai *centroid* yang baru hingga nilai *centroid* stabil atau mendekati nilai *centroid* sebelumnya.

Nilai *similarity* antara *centroid user* dan *user* lainnya didefinisikan mengunakan *Pearson correlation-coefficient*, adalah sebagai berikut (Persamaan 2.1):

$$Sim_{u,u'} = \frac{\sum_{t \in T(u) \wedge T(u')} (R_u(t) - \overline{R_u}) \,.\, (R_{u'}(t) - \overline{R_{u'}})}{\sqrt{\sum_{t \in T(u) \wedge T(u')} (R_u(t) - \overline{R_u})^2} \sqrt{\sum_{t \in T(u) \wedge T(u')} (R_{u'}(t) - \overline{R_{u'}})^2}} \qquad (2.1)$$

$Sim_{u,u'}$ merupakan nilai *similarity* antara *user* centroid dan *user* lainnya.

$R_{u'}(t)$ merupakan nilai *rating centroid* untuk *item t*.

$\overline{R_{u'}}(t)$ merupakan rata-rata rating *user centroid*.

$R_u(t)$ merupakan nilai rating *user u* untuk *item t*.

$\overline{R_u}(t)$ merupakan rata-rata rating *user u*.

$t \in T(u)\text{^}T(u')$ adalah irisan *item-item* yang dirating *user centroid* dan *user u*.

2.4.2 Data *Smoothing*

Sparsity adalah masalah fundamental yang ada pada *collaborative filtering*, dengan menerapkan data *smoothing* pada sistem CF maka masalah ini dapat diminimalisir. *Smoothing* dilakukan dengan mengisi *missing value* pada *dataset* dengan *rating* bayangan. Berdasarkan hasil *clustering*, teknik *smoothing* data untuk *rating* bayangan didefinisikan sebagai penilaian khusus berikut (Xue, Gui-Rong. 2005) :

$$R_u(t) = \begin{cases} R_u(t), & jika\ user\ u\ merating\ item\ t \\ \hat{R}_u(t), & lainnya \end{cases} \quad (2.2)$$

$\hat{R}_u(t)$ merupakan nilai *smoothing* untuk *user-user* terhadap *item t*, dimana $\hat{R}_u(t)$ didefinisikan seperti berikut ini:

$$\hat{R}_u(t) = \overline{R_u} + \Delta R_{C_u}(t) \quad (2.3)$$

$\Delta R_{C_u}(t)$ merupakan rata-rata deviasi untuk semua *user* di dalam *cluster* C_u untuk *item t*, yang mana didefinisikan sebagai:

$$\Delta R_{C_u}(t) = \sum_{u' \in C_u(t)} \frac{(R_{u'}(t) - \overline{R_{u'}})}{|C_u(t)|} \quad (2.4)$$

$C_u(t) \in C_u$ adalah kumpulan *user* yang berada di *cluster* C_u yang telah merating *item t*.

$|C_u(t)|$ merupakan jumlah *user* di *cluster* C_u yang telah merating *item t*.

2.4.3 *Neighbor Pre-Selection*, Menentukan *Cluster* untuk *Active User*

Tahapan terpenting dari algoritma *collaborative filtering* adalah mencari tetangga terdekat dari *active user* di dalam suatu *cluster*. Nilai *similarity* berperan dalam proses mencari *user-user* yang memiliki kesamaan dengan *active user* dan menentukan tetangga terdekat dari *active user* tersebut. Pada metode CF proses semacam ini akan dilakukan dengan mencari tetangga terdekat ke seluruh *database*. Hal ini berpengaruh pada proses komputasi ketika *item* dan *user* baru ditambahkan ke dalam *database*. Dengan penerapan konsep *cluster*, komputasi sistem dapat menjadi lebih baik. Kelompok *user* di dalam *cluster* direpresentasikan oleh *centroid cluster*. *Centroid* sendiri direpresentasikan sebagai hasil rata-rata *rating* dari seluruh *user* yang ada di *cluster*. Dalam menentukan beberapa *cluster* yang memiliki nilai kesamaan dengan *active user* dilakukan dengan mengkalkulasi nilai *similarity* antara *active user* dengan kumpulan *user-user* di suatu *cluster* menggunakan persamaan berikut (Xue, Gui-Rong. 2005) :

$$Sim_{u_a,C} = \frac{\sum_{t \in T(u_a)\wedge T(C)} \Delta R_C(t)(R_{u_a}(t) - \overline{R_{u_a}})}{\sqrt{\sum_{t \in T(u_a)\wedge T(C)} (\Delta R_C(t))^2} \sqrt{\sum_{t \in T(u_a)\wedge T(C)} (R_{u_a}(t) - \overline{R_{u_a}})^2}} \quad (2.5)$$

$Sim_{u_a,C}$ merupakan nilai *similarity* antara *active user* dengan *user-user* di suatu *cluster* atau nilai *centroid.*

$\Delta R_C(t)$ merupakan selisih antara *rating centroid* dengan rata-rata *rating centroid* suatu *cluster.*

$R_{u_a}(t)$ merupakan *rating active user* untuk *item t.*

$\overline{R_{u_a}}$ merupakan rata-rata *rating item active user.*

$t \in T(u_a)\wedge T(C)$ merupakan irisan *item-item* yang dirating oleh *active user* dan kumpulan *user-user* di *cluster.*

Setelah melakukan kalkulasi terhadap nilai *similarity* antara setiap grup dan *active user*, maka dapat ditentukan beberapa kumpulan *cluster* yang memiliki kemiripan besar dengan *active user*.

2.4.4 *Neighbor Selection* (Pencarian Tetangga Terdekat)

Jika cluster *active user* telah ditentukan, maka dilanjutkan dengan tahap pencarian tetangga terdekat. Hal ini berdampak pada proses perhitungan untuk mencari tetangga terdekat bagi *active user* sehingga terdapat nilai untuk variabel w_{ut} yang merupakan *confidential weight* untuk *user* u terhadap *item* t, didefinisikan sebagai berikut (Xue, Gui-Rong. 2005) :

$$w_{ut} = \begin{cases} 1 - \lambda, jika\ user\ u\ merating\ item\ t \\ \lambda,\ lainnya \end{cases} \quad (2.6)$$

λ merupakan parameter yang digunakan untuk penyesuaian antara *original rating* dan *group rating*. Nilai λ berkisar antara 0 hingga 1. Dalam pencarian tetangga terdekat digunakan persamaan berikut ini untuk menentukan nilai *similarity* :

$$Sim_{u_a,u} = \frac{\sum_{t \in T(u_a)} w_{ut}.(R_u(t) - \overline{R_u}).(R_{u_a}(t) - \overline{R_{u_a}})}{\sqrt{\sum_{t \in T(u_a)} w_{ut}{}^2.(R_u(t) - \overline{R_u})^2} \sqrt{\sum_{t \in T(u_a)} (R_{u_a}(t) - \overline{R_{u_a}})^2}} \quad (2.7)$$

$Sim_{u_a,u}$ merupakan nilai *similarity* antara *active user* dengan *user-user* dalam suatu *cluster*.

2.4.5 Perhitungan Nilai Prediksi

Jumlah tetangga terdekat terhadap *active user* serta *confidential weight* yang menggabungkan *rating-rating* mempengaruhi proses perhitungan nilai prediksi. Perhitungan nilai prediksi tersebut menggunakan persamaan sebagai berikut (Xue, Gui-Rong. 2005) :

$$R_{u_a}(t) = \overline{R_{u_a}} + \frac{\sum_{i=1}^{K} w_{ut}.sim_{u_a,u}.(R_u(t) - \overline{R_u})}{\sum_{i=1}^{K} w_{ut}.sim_{u_a,u}} \quad (2.8)$$

K merupakan jumlah *user* yang menjadi tetangga terdekat *active user.*

2.5 *Mean Absolute Error* (MAE)

Mean Absolute Error merupakan persamaan yang termasuk jenis *statistical accuracy metrics*, dimana MAE merupakan *metrics* yang paling sering digunakan untuk pengukuran akurasi dari *recommender system*. Sesuai dengan namanya, MAE menghitung nilai rata-rata selisih antara nilai prediksi dengan nilai sebenarnya (Xue, Gui-Rong. 2005). Nilai MAE sendiri, berkisar antara 0 hingga 1. Semakin kecil nilai MAE menandakan semakin akurat prediksi nilai *rating* dari sebuah *recommender system* (Xue, Gui-Rong. 2005).

$$MAE = \frac{\sum_{u \in T} |R_u(t_j) - \widetilde{R_u}(t_j)|}{|T|} \qquad (2.9)$$

$R_u(t_j)$ merupakan nilai *rating item* t_j yang diberikan oleh *user u*.

$\widetilde{R_u}(t_j)$ merupakan nilai *rating* prediksi *user u* terhadap *item* t_j.

$|T|$ adalah jumlah data yang dikalkulasikan.

2.6 *Blackbox*

Blackbox testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitamnya. Sama seperti pengujian *blackbox*, mengevaluasi hanya dari tampilan luar fungsionalitasnya (*Interfacenya* saja). Tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui input dan output).

- Menurut Myers (1979) :

  Proses menjalankan program dengan maksud menemukan kesalahan.

- Menurut IEEE (1990) :

  1. Pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada output yang dihasilkan yang merespon input yang dipilih dan kondisi eksekusi.

2. Pengujian yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu.

2.7 Sistem

Suatu sistem pada dasarnya merupakan suatu susunan yang teratur dari kegiatan yang berhubungan satu sama lain dan prosedur-prosedur yang berkaitan yang melaksanakan dan mempermudah pelaksanaan kegiatan utama dari suatu organisasi.

2.7.1 Pengertian Sistem

Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul, bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Jogiyanto. HM. 2005). Definisi sistem dapat dilihat dari dua kelompok pendekatan yaitu yang menekankan pada prosedur dan yang menekankan pada komponen atau elemennya.

1. Berdasarkan Penekanan Prosedur

   Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama untuk melakukan sesuatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

2. Berdasarkan Penekanan Komponen

   Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

2.7.2 Elemen Sistem

Adapun dalam suatu sistem, dikenal berbagai elemen-elemen yang menyusun suatu sistem, yaitu sebagai berikut :

a) Tujuan

Dalam suatu sistem pasti memiliki suatu tujuan, mungkin hanya satu atau mungkin juga banyak. Dalam suatu tujuan inilah yang menjadi pemotivasi yang mengarahkan akan suatu sistem. Tanpa adanya suatu tujuan, maka suatu sistem menjadi tak terarah dan tak terkendali. Hal ini dapat menyebabkan tujuan yang tidak akan tercapai dengan baik.

b) Masukan

Masukan input sistem adalah segala sesuatu yang masuk ke dalam suatu sistem yang kemudian menjadi bahan yang nantinya menjadi bahan untuk diproses. Masukan sistem ini dapat berupa hal-hal yang berwujud (kenyataan) ataupun yang tidak tampak (khayalan). Adapun beberapa contoh, seperti masukan sistem yang berjalan yang berupa bahan mentah yang berwujud. Sedangkan masukan sistem yang tidak tampak atau tidak berwujud adalah sebuah informasi, dikarnakan sebuah informasi tidak berwujud atau tidak tampak dengan kasat mata.

c) Proses

Proses adalah suatu bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai dalam mencapai suatu tujuan dapat berupa informasi dan produk, akan suatu proses bisa juga berupa hal-hal yang tidak berguna atau yang tidak diinginkan, misalnya seperti pembuangan limbah yang nantinya juga dapat menimbulkan suatu masalah dalam pencapaian suatu tujuan

d) Keluaran

Keluaran atau output disini adalah lanjutan tahapan dari tahapan proses. Dalam output yang dihasilkan dari tahapan-tahapan sebelumnya ialah dapat berupa informasi yang diinginkan, laporan dan lain-lain.

e) Batas atau *boundary*

Batasan atau boundary adalah pemisah antara suatu sistem dengan daerah di luar sistem atau yang bisa disebut juga lingkungan. Dalam batas suatu sistem akan menentukan konfigurasi, ruang lingkup dan kemampuan suatu sistem. Contohnya adalah suatu perusahaan yang memiliki peraturan akan tetapi perusahaan atau organisasi tersebut memilik keterbatasan kemampuan karyawan atau pegawainya. Tentu saja batas suatu sistem dapat dikurangi atau diperbaharui sehingga sistem tersebut akan mengubah perilaku sistem itu sendiri.

f) Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian disini dapat diwujudkan dengan menggunakan umpan balik, yang mencuplik keluaran. Umpan balik ini digunakan dengan tujuan untuk mengendalikan suatu masukan maupun proses. Tujuannya utamanya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

g) Lingkungan

Lingkungan merupakan segala sesuatu yang berada diluar sistem. Lingkungan ini bisa berpengaruh terhadap suatu operasi sistem dalam arti bisa merugikan atau menguntungkan suatu sistem itu sendiri. Lingkungan yang merugikan tentu saja harus dihindari dan dikendalikan supaya tidak mengganggu akan kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, dikarena akan memacu terhadap kelangsungan hidup sistem tersebut.

2.7.3 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), pengolahan (*process*) keluaran (*output*) dan sasaran (*objektif*) atau tujuan (*goals*) .

a) Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sub-sub sistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

b) Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem dengan sistem lainnya atau dengan lingkungan lainnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukan ruang lingkup (*scope*) dari sistem tersebut.

c) Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem dengan demikian harus dijaga dan dipelihara, sedangkan lingkungan luar yang merugiakan harus ditahan dan dikendalikan kalau tidak maka akan menggangu kelangsungan hidup dari sistem.

d) Penghubung Sistem

Penghubung merupakan media penghubung antara sub sistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) pada sistem lainnya dengan penghubung satu subsistem dapat berintegrasi dengan subsistem lainnya membentuk satu kesatuan.

e) Masukan Sistem

Masukan sistem adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan (signal input). maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluarannya.

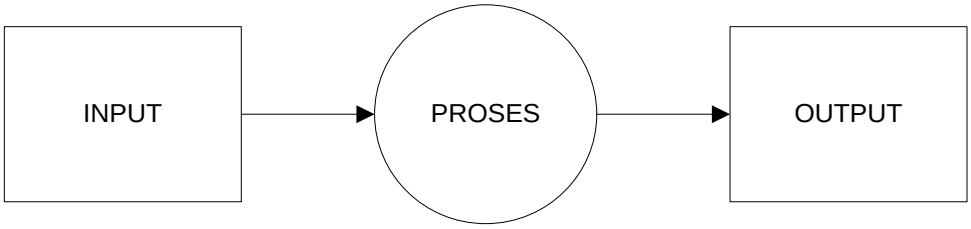
f) Keluaran Sistem

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

g) Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

Bentuk umun dari suatu sistem terdiri atas masukan (*input*), proses dan keluaran (*output*) seperti terlihat pada gambar 2.1 dalam bentuk umum sistem ini bisa melakukan satu atau lebih masukan yang akan diproses dan menghasilkan keluaran sesuai dengan rencana yang telah ditentukan sebelumnya .



Gambar 2.3 Bentuk Umum Sistem (Jogiyanto. HM. 2005).

2.8 Artificial Intelligince

Dewasa ini Artificial Intelligince (AI) banyak menarik minat dan perhatian dari masyarakat luas. Menurut (Kusumadewi, Sri. 2003), AI adalah salah satu bagian ilmu komputer yang membuat agar komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Pada awal diciptakannya, komputer hanya dipakai sebagai alat hitung saja. Namun, seiring perkembangan jaman maka peran komputer semakin mendominasi kehidupan umat manusia.

Manusia bisa menjadi pandai dalam menyelesaikan suatu persoalan karena manusia mempunyai pengehtahuan dan disertai pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengehtahuan yang dimiliki seseorang, maka orang tersebuat diharapkan dapat lebih mampu dalam menyelesaikan suatu persoalan. Agar komputer pun dapat bertindah seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengehtahuan dan mempunyai kemampuan untuk berpikir. Pada AI, komputer dibekali beberapa metode yang diharapkan agar komputer dapat menjadi benda yang pintar.

2.8.1 Bidang – Bidang Kecerdasan Buatan

a. Sistem Pakar (*Expert System*)

   Sistem pakar adalah program komputer yang didesain untuk berlaku sebagai seorang ahli dalam suatu bidang khusus. Namun sekarang sistem

pakar biasanya hanya digunakan untuk membantu para ahli dalam memecahkan suatu masalah.

b. *Natural Language Procesing* (NLP)

Natural Language Processing adalah komputer yang dimaksudkan untuk mengenal makna dari bentuk kalimat berbeda-beda. Selain mampu mengerti bahasa sehari-hari, NLP juga mencakup kemampuan membentuk kalimat dalam bahasa sehari-hari.

c. Pengenalan Pola

Recognition adalah program komputer yang ditujukan untuk mengenali suatu objek. Contohnya dalam *Speech Recognation*, komputer dapat mengenali suara, dan sekaligus bias membedakan berbagai macam bentuk sinyal. Contoh lain dalam *Character Recognation*, komputer dapat mengenali karakter, sekaligus dapat membedakan berbagai macam bentuk karakter.

d. *Computer Vision*

Ilmu pengetahuan dan teknologi komputer yang berkaitan dengan pengolahan citra komputer seakan dapat melihat benda, terkait dengan teori untuk membangun suatu sistem yang terdiri dari gambar. Contoh yang sering digunakan adalah pendeteksian plat nomor kendaraan.

e. *Robotic*

Mesin yang diprogram untuk melakukan tugas-tugas mekanik, berintelegensi dan dapat member respon terhadap perubahan lingkungan. Contohnya adalah Asimo dan Aibo.

f. *Automatic Programming*

Komputer yang dapat membuat program sendiri dan menyesuaikan dengan spesifikasi yang dinginkan oleh pemrogram.

g. *Planning and Decission Support*

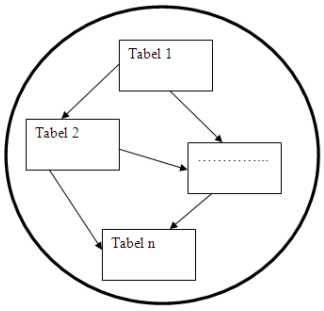
Komputer yang memiliki kemampuan khusus untuk membantu manajer dalam membuat perancanaan dan pengambilan keputusan.

h. *Soft Computing*

Soft Computing adalah koleksi dari beberapa metodologi yang bertujuan untuk mengeksploitasi adanya toleransi terhadap ketidaktepatan, ketidakpastian dan kebenaran parsial untuk dapat diselesaikan dengan mudah, kuat, dan biaya penyelesaiannya murah. Definisi ini pertama kali diungkapkan oleh Prof. Lofti Zadeh pada tahun 1992. *Soft Computing* merupakan inovasi baru dalam membangun sistem cerdas. Sistem cerdas ini merupakan sistem yang memiliki keahlian seperti manusia pada domain tertentu, mampu beradaptasi dan belajar agar dapat bekerja lebih baik jika terjadi perubahan lingkungan.

2.9 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamannya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi.



Gambar 2.4 Ilustrasi Basis Data (Shalahuddin, Muhammad & Rosa Ariani S. 2011)

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi :

1. memasukkan, menyimpan , dan mengambil data
2. membuat laporan berdasarkan data yang telah disimpan

2.9.1 Bahasa Basis Data

DBMS merupakan perantara bagi pemakai dengan basis data dalam disk. Cara berkomunikasi/berinteraksi antara pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa tersebut dapat disebut sebagai bahasa basis data yang terdiri atas sejumlah perintah yang diformulasikan oleh user dan diproses oleh DBMS untuk melakukan suatu aksi atau pekerjaan tertentu.

Ada 3 bahasa yang digunakan dalam basis data yaitu :

1. DDL (*Data Definition Language*)
   Merupakan bahasa definisi data yang digunakan untuk membuat dan mengelola objek database seperti database, tabel dan view.
2. DML (*Data Manipulation Language*)
   Merupakan bahasa manipulasi data yang digunakan untuk memanipulasi data pada objek database seperti tabel.
3. DCL (*Data Control Language*)
   Merupakan bahasa yang digunakan untuk mengendalikan pengaksesan data.

2.9.2 SQL (*Structured Query Language*)

SQL(*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

SQL mulai berkembang pada tahun 1970. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standarts Institute*) dan pada tahun 1987 oleh ISO (*International Organization for Standardization*) dan disebut sebagai SQL-86.

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut, sehingga ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal query yang dimasukkan sama. Berikut ini merupakan contoh pengaksesan data pada DBMS sengan SQL yang secara umum terdiri dari 4 hal sebagai berikut:

1. Memasukkan data (*insert*)

```
INSERT INTO Tabel_Mahasiswa
(nim, nama, tanggal_lahir)
VALUES
('10108530','Teguh Budianto','17-08-1990')
```

Gambar 2.5 Query Insert

Query di atas digunakan untuk memasukkan data mahasiswa dengan nim 10108530, nama Teguh Budianto, dan tanggal lahir 17 agustus 1990 ke tabel tabel_mahasiswa.

2. Mengubah data (update)

```
Update Tabel_mahasiswa
SET
       tanggal_lahir = '1990-03-21'
WHERE
       nim = '10108530'
```

Gambar 2.6 Query Insert

Query pada gambar 2.4 digunakan untuk mengubah data lahir mahasiswa dengan nim = 10108530 menjadi 21 Maret 1990 dalam tabel_mahasiswa.

3. Menghapus data (*delete*)

```
DELETE FROM Tabel_Mahasiswa
WHERE
nim = '10108530'
```

Gambar 2.7 *Query Delete*

Query pada gambar 2.5 digunakan untuk menghapus data mahasiswa dengan NIM = 10108530 dari tabel_mahasiswa

4. Menampilkan data (*select*)

```
SELECT nim, nama
FROM Tabel_mahasiswa
WHERE nim='10108530'
```

Gambar 2.8 *Query Select*

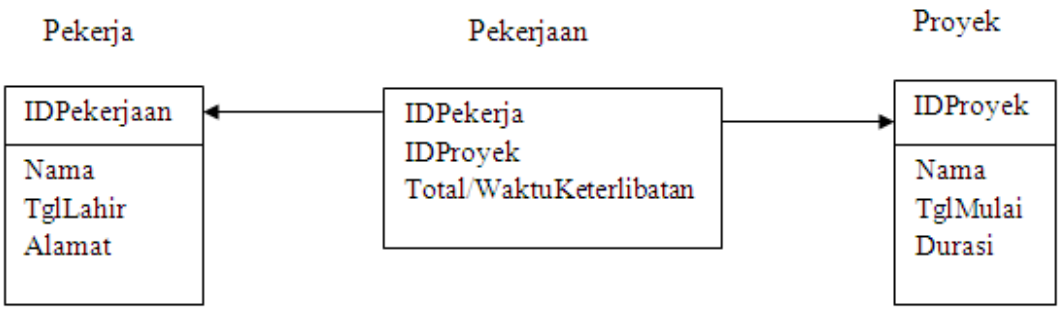
Query pada gambar 2.6 digunakan untuk menampilkan data mahasiswa yang tersimpan dalam tabel_mahasiswa dengan nim = 10108530.

2.9.3 ERD (*Entity Relationship Diagram*)

Pemodelan awal basis data yang paling banyak digunakan adalah *Entity Relationship Diagram (ERD)*. *ERD* dikembangkan berdasarkan teori himpunan dalam bidang matematika. *ERD* digunakan untuk pemodelan basis data relasional, sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan *ERD*.

2.9.4 Diagram Skema Relasi

Skema basis data yang menyatakan kebergantungan kunci utama dan *foreign key* dapat digambarkan secara grafis dengan diagram skema relasi. Berikut adalah diagram relasi untuk relasi. Pekerja, relasi pekerjaan dan relasi proyek.



Gambar 2.9 Diagram Skema Relasi (Erlangga. 2011)

Masing masing relasi digambarkan sebagai kotak (persegi) dengan atribut – atribut didaftarkan didalamnya dan nama relasi muncul diatasnya. Jika tedapat atribut – atribut kunci utama direlasi, satu garis horizontal membelah persegi dimana atribut – atribut kunci utama didaftarkan diatas garis kebergantungan *foreign key* muncul sebagai anak panah dari atribut – atribut *foreign key* relasi yang mengacu ke kunci utama dari relasi yang diacu.

Kita tidak boleh menyamakan skema relasi dengan diagram ER. Perbedaan utama ialah diagram ER tidak menunjukkan secara eksplisit atribut – atribut *foreign key* sementara diagram skema relasi telah menunjukkan atribut – atribut *foreign key* secara eksplisit.

Kebanyakan sistem basis data menyediakan perancangan dengan antarmuka pemakai secara grafis untuk menciptakan diagram skema relasi. Sekali lagi diagram yang langsung menyatakan *foreign key* dan kebergantungan terhadap kunci utama suatu relasi tidak disebut diagram ER melainkan skema relasi. Diagram skema relasi terdapat di *Microsoft Access*, *PowerDesigner* level fisik, serta dimasukkan dalam paket *MS SQL Server*.

2.10 UML (*Unified Modelling Language*)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan *Object Management Group,* sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.

Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan

sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.10.1 Diagram UML

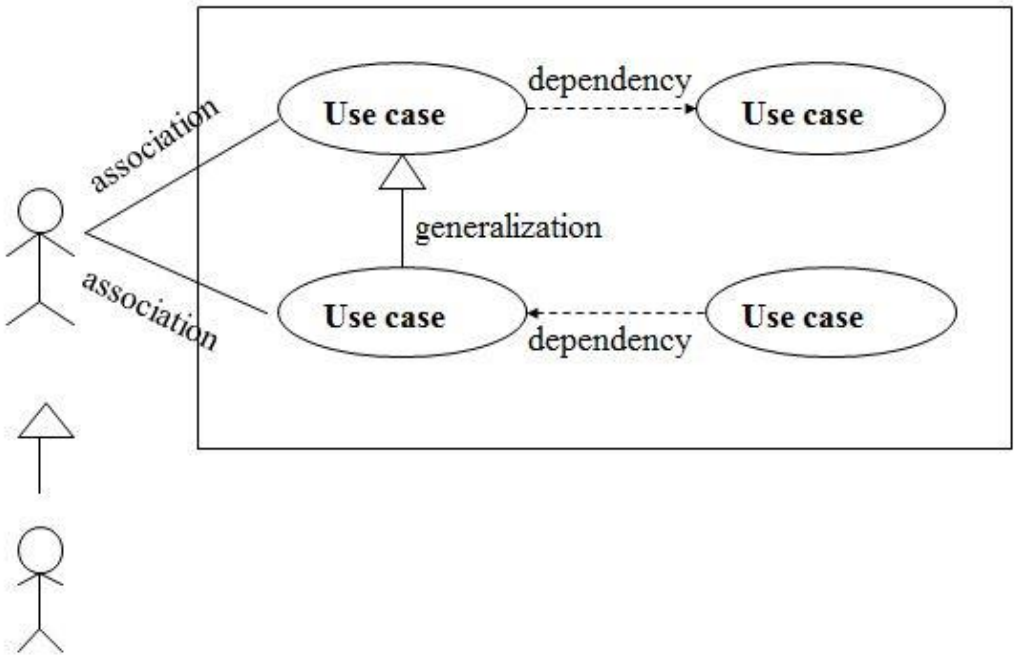
UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar objek.
4. *Collaboration Diagram* untuk memodelkan interaksi antar objek.
5. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan objek di dalam sistem.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur objek.
9. *Component Diagram* untuk memodelkan komponen object.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case diagram, sequence diagram, collaboration diagram, dan class diagram*.

1. *Use Case Diagram*

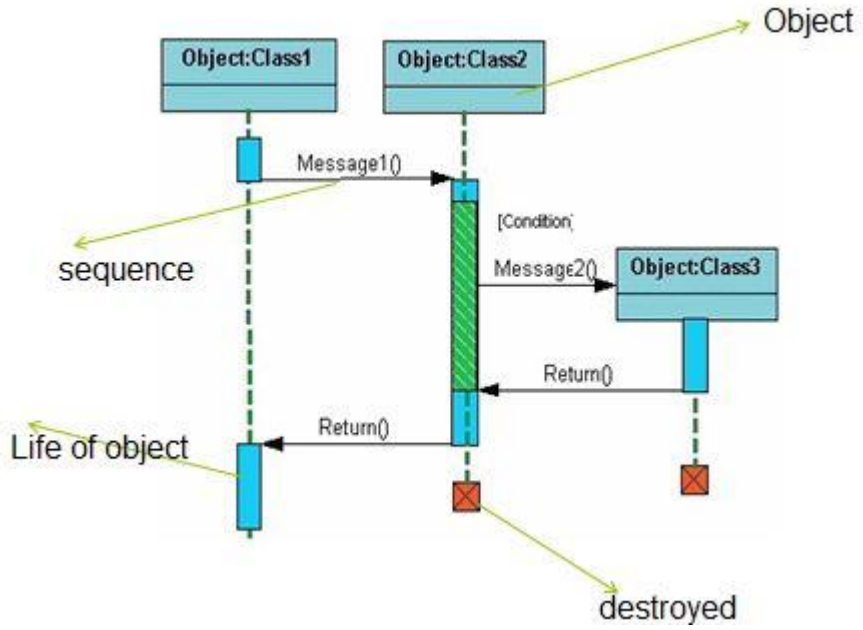
Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk use case dan actor. Aktor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh aktor. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan didalamnnya. Aktor yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.



Gambar 2.10 Contoh *Use Case Diagram* (Erlangga. 2011).

2. *Sequence Diagram*

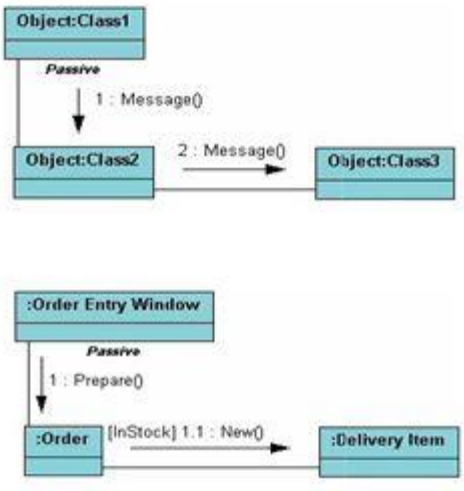
Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.



Gambar 2.11 Contoh *Sequence Diagram* (Erlangga. 2011).

3. *Collaboration Diagram*

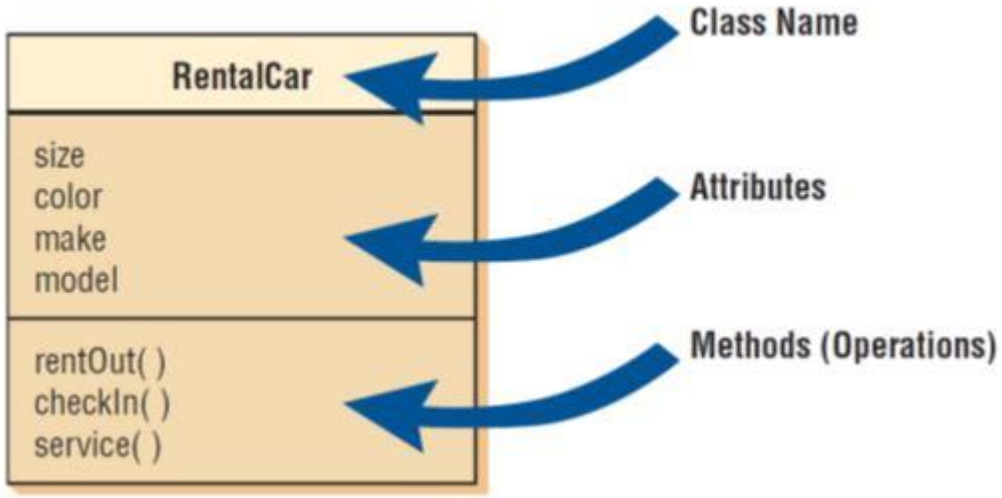
Collaboration diagram dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence diagram* yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration diagram* lebih focus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.



Gambar 2.12 Contoh *Collaboration Diagram* (Erlangga. 2011).

4. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.



Gambar 2.13 Contoh *Class Diagram* (Erlangga. 2011).

2.11 PHP (Personal Home Page)

PHP merupakan bahasa pemograman *server side* yang didesain untuk aplikasi web. PHP merupakan bahasa pemograman yang dapat disisipkan dalam html, pada awalanya php diperuntukan untuk pembuatan website pribadi yang bersifat dinamis.
Sedangkan menurut Sutarman dalam bukunya yang berjudul Membangun Aplikasi Web dengan PHP & MySQL edisi 2 mengatakan bahwa :
PHP adalah salah satu bahasa server side yang di desain khusus untuk aplikasi web (Septian, Gungun. 2011).

2.12 *Framework*

Framework sebagaimana arti dalam bahasa indonesianya yaitu kerangka kerja dapat diartikan sebagai kumpulan dari *library (class)* yang dapat diturunkan, atau dapat langsung dipakai fungsinya oleh modul-modul atau fungsi yang akan kita kembangkan. (Septian, Gungun. 2011).

2.12.1 Pengertian PHP *Framework*

PHP *Framework* adalah sebuah lingkungan pengembangan aplikasi berbasis php yang berisi sejumlah *class* yang telah dibuat agar digunakan kembali untuk membuat aplikasi. Berbagai jenis *class* ini tergantung tujuan dari PHP *framework* tersebut, dan php *framework* yang satu dengan *framework* yang lain kemungkinan berbeda. Sebagian besar php *framework* yang tersedia saat ini berbasis MVC (*model view controller*). Ini adalah model php *framework* yang modern. (Septian, Gungun. 2011).

2.12.2 *Framework CodeIgniter*

CodeIgniter adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis. Dengan menggunakan PHP *CodeIgniter* akan memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuat dari awal. (Septian, Gungun. 2011).

2.12.3 MVC (*Model View Controller*)

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. Ini meminimalkan *script* dari halaman-halaman web sejak *script* presentasi (HTML, CSS, *JavaScript*, dsb) dipisahkan dari PHP *scripting*, istilah umum yang familiar adalah menghindari terjadinya *spaghetti code*. (Septian, Gungun. 2011).

A. *Model*

   Model merepresentasikan struktur data yang dibangun. Umumnya kelas model berisi fungsi-fungsi yang membantu *developer* untuk mengelola, memasukkan, dan mengupdate informasi dalam database.

B. *View*

View adalah informasi yang disajikan untuk *user*, berupa tampilan atau *user interface*. *View* umumnya adalah tampilan sebuah halaman web itu sendiri, tetapi dalam *CodeIgniter*, *view* dapat juga menjadi bagian-bagian atau penggalan-penggalan halaman seperti *header* atau *footer*. *View* dapat juga sebagai halaman RSS, atau tipe-tipe halaman lainnya.

C. Controller

Controller bertugas sebagai penghubung antara *Model, View,* dan beberapa *resource* lainnya yang dibutuhkan untuk memproses HTTP *request* untuk *meng-generate* sebuah halaman web.

2.13 Pengertian *jQuery*

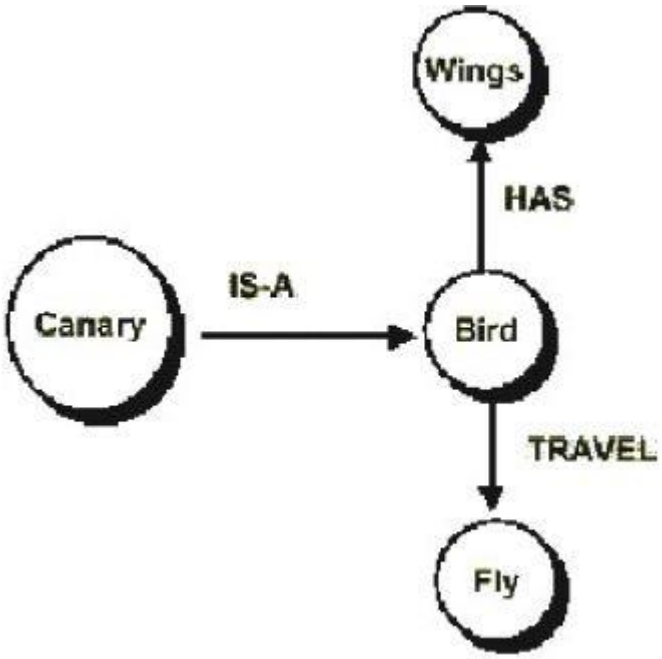
jQuery merupakan *library Java Script* yang banyak digunakan saat ini. *jQuery* di buat oleh John Resig pada tahun 2006. Banyak *website* yang memanfaatkan *library* ini untuk menyederhanakan fungsi-fungsi yang ada pada *JavaScript* atau *Ajax*. Sesuai dengan slogannya *jQuery* sendiri "*Write less, do more*", menulis sedikit namun dapat mengerjakan banyak hal, sehingga Anda dapat menghemat *coding program* Anda.

2.14 Pengertian HTML *(Hypertext markup Language)*

HTML adalah Format Dokumen bahasa yang digunakan untuk merancang halaman web. HTML adalah sistem untuk membuat naik atau menandai, sebuah dokumen sehingga dapat dipublikasikan di web. HTML mendefinisikan apa yang umumnya ditularkan antara *node* dalam *network*. HTML telah dikembangkan dengan maksud agar berbagai macam jenis perangkat harus dapat memanfaatkan informasi di *web* PC dengan tampilan grafis dari kedalaman resolusi dan warna yang bervariasi.

2.15 Jaringan Semantik

Jaringan semantik merupakan pengetahuan secara grafis yang menunjukkan hubungan antar berbagai objek. Suatu jaringan semantik sederhana bisa dilihat pada gambar berikut :



Gambar 2.13 Contoh jaringan semantik (J, Durkin. 1994)

Jaringan semantik disusun dari node dan arc (Lines). Node merupakan representasi dari objek, objek properti atau properti value. (digambarkan dengan lingkaran).

Arc merupakan representasi dari hubungan antar node. (digambarkan dengan garis). Baik node maupun arc memiliki label yang secara jelas menjelaskan representasi dari objek dan hubungan antar node.

Hampir semua objek, atribut, pemikiran atau apapun dapat dirumuskan dan dihubungkan antara satu dengan lainnya oleh garis-garis (arc). Arc umumnya menggunakan istilah seperti : “IS-A”, “HAS-A”, dll. Pada gambar contoh tadi: Link “IS-A” diantara node “Canary” dan “Bird” menjelaskan hubungan “*spesific-to-general*” diantara keduanya.