BAB II

TINJAUAN PUSTAKA

2.1. Model

Dalam menyelesaikan suatu permasalahan yang akan dipecahkan, para ahli manajemen biasanya selalu menggunakan model. Biasanya, model mewakili sejumlah objek atau aktivitas yang menimbulkan permasalahan dan disebut sebagai entitas (*entity*).

2.1.1 Pengertian Model Secara Umum

Model merupakan kerangka konseptual yang digunakan sebagai pedoman dalam melakukan suatu aktivitas tertentu. Model dapat pula diartikan sebagai barang tiruan, metafora, atau kiasan yang dirumuskan secara eksplisit dan mengandung sejumlah unsur yang saling ketergantungan. Setiap model menjelaskan fenomena dalam bentuk yang tidak sebagaimana biasanya dirasakan, dan setiap model diperlukan untuk menjelaskan sesuatu yang lebih atau berbeda dari data.

Ackoff (Togar,1994) menyatakan bahwa model dirancang sebagai penggambaran operasi dari suatu sistem nyata secara ideal guna menjelaskan atau menunnjukan hubungan-hubungan penting yang terkait.

Didalam model ada istilah simulasi, validasi, *error* (kesalahan). Kegiatan mencoba-coba berbagai *alternative*, untuk melihat perubahan dan hasil yang terbentuk disebut simulasi.

Misalnya kita tidak menyukai kursi sofa yang terletak dibawah jendela ruang tamu, maka kita bisa mencobanya untuk mengubah dan meletakkannya di bagian lain di ruang tamu tersebut. Namun apa jadinya jika kita mencoba-coba hal tersebut dilakukan pada bangunan yang sudah paten (sesungguhnya/bukan model), misalnya sebuah bangunan rumah. Betapa repot dan mahalnya kegiatan mencoba-coba itu.

Kegiatan membandingkan model dengan yang asli dikenal dengan validasi. Besarnya perbedaan perbandingan disebut denga *error* (kesalahan). Kegiatan validasi bertujuan agar *error* (kesalahan) dapat seminimal mungkin.

Seorang pemodel akan selalu mempertaruhkan kemampuannya dengan model yang memiliki kualitas model, setidak-tidaknya pada bebeapa langkah perancang model. Model kualitatif yaitu deskripsi abstrak dari pengetahuan yang dideskripsikan oleh ilmuan dengan semua komplektifitasnya. Model kkualitatif menggambarkan hubungan causal (sebab akibat) antar variable-variablenya. Hubungan antara *variable* ini menggunakan data kualitatif yang berupa data, turunan atau konstan.

Model kualitatif didasarkan pada ikhtisar data, dalam persoalan ini didasarkan dalam percobaan dan uraian. Umumnya model kualitatif didasarkan pada cerita, seiring dalam bentuk teks atau kalimat. Dalam dunia bisnis, model kualitatif dibangun dari hasil yang difokuskan pada pengelompokan data. Sebagai contoh suatu pendeskripsian persoalan manajemen dipaparkan dengan menggunakan data kualitatif. Situasi persoalan ini akan dimaknai dengan uraian kata-kata oleh pemodel. Hal itu disajikan dalam bentuk cerita, tekstual, dan

diagramatik. Pemodel harus memulai membangun dan menstruktur model dari situasi bisnis atau persoalan. Setiap pemodel umumnya harus bekerja dengan model kualitatif.

Jadi model kualitatif menggambarkan uraian, pemodel struktur bisnis dan menndokumentasikan model user secara luas dan kompleks. Hal itu berdasarkan pada pengalaman kualitatif yang menyoroti pemanfaatan objek dan hubungannya dengan tujuan komunikasi dan cara mendokumentasikan model *user*.

Model kualitatif dibangun dari batasan data yang luas mencangkup *image* dan film, suara dan kalimat, dan cerita dan dialog, sedangkan yang banyak digunakan oleh pemodel adalah teks yang bersifat elementer dengan menggunakan teknik-teknik pemeragaan antara lain teori, persandian dan teknik jaringan kerja yang menggunakan ilmu sosial.

2.1.2 Proses Perancang Model

Tahap pengembangan suatu model (Togar,1994) adalah:

1. Definisi masalah, dalam tahap ini masalah yang sulit didefinisikan dan diurai menjadi unsur-unsur pembentuk masalah. Didefinisikan juga sistem dan faktor eksternal (di luar sistem). Dicari komponen masalah yang paling penting dan signifikan dalam pemecahan masalah. Dicari pula komponen masalah yang bisa dijadikan titik acuan awal pemecahan masalah.
2. Strukturisasi model konseptual, pada tahap ini diuraikan hubungan antara komponen penyusun masalah, sistem dan tujuan studi.

3. Formulasi model, yaitu proses merumuskan perilaku model, dan hubungan antara variable.Interaksi antar *variable* yang kompleks sering disederhanakan dengan menggunakan asumsi yang tepat.
4. Kalibrasi model yaitu menyesuaikan parameter-parameter dalam model sesuai dengan kondisi nyata di lapangan.
5. Validasi model yaitu tahap penguji model dengan mengubah-ubah nialai *variable* model.
6. Uji sensitifitas yaitu tahap penguji perilaku model dengan mengubah-ubah nilai *variable* model.
7. Analisis dan solusi model. Model akan menghasilkan *alternative* solusi sesuai dengan skenario yang kita buat. Hasil model yang dirasakan kurang tepat, perlu dijalankan ulang (biasanya mengggunakan komputer), sampai tercapai yang memuaskan. Proses ini dikenal dengan simulasi model.
8. Implementasi model. Agar model dapat diterapkan dengan baik, maka pihak perancang model dan pengguna model (misalnya para pengambbil keputusan) perlu bekerja sama sejak awal. Perancang model akan membuat model sedinamis dan semudah mungkin operasionalnya (*user friendly*), dan pengguna model akan memberi masukan-masukan sesuai dengan kebutuhan pengguna.

2.2 Pengertian Sistem Pendukung Keputusan

Pemecahan masalah merupakan aktivitas utama yang sering digunakan sebagai acuan keberhasilan atau kegagalan karier manajer. Biasanya orang selalu

berpendapat bahwa suatu masalah merupakan sesuatu yang buruk karena biasanya hanya menaruh perhatian yang relatif kecil pada cara meraih peluang (*opportunity seizing*).

Raymond McLeod, Jr dan George P. Schell dalam bukunya Sistem Informasi Manajemen (2007, hal 19) :

> "*Kita memisahkan antara meraih peluang dengan memecahkan masalah – dengan mendefinisikan* ***masalah*** *sebagai kondisi atau kejadian yang membahayakan atau berpotensi membahayakan perusahaan secara negatif. Namun masalah juga dapat menguntungkan atau berpotensi memberikan keuntungan secara positif. Hasil dari penyelesaian masalah adalah* ***solusi*** *atau jawaban permasalahan."*

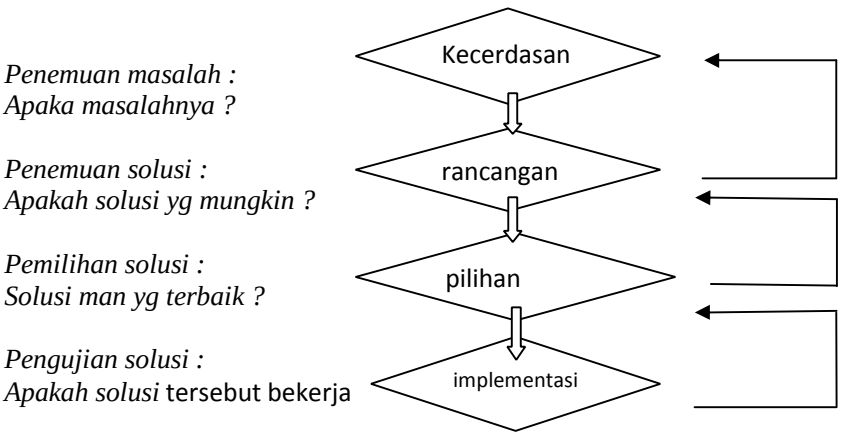
Selama proses pemecahan masalah, pimpinan (manajer) terlibat dalam pembuatan keputusan, dimana ia bertugas menyeleksi beberapa alternatif dari tindakan-tidakan yang mungkin diambil. Lebih lanjut Raymond McLeod, Jr dan George P. Schell dalam bukunya Sistem Informasi Manajemen (2007, hal 19) mengatakan :

> *"Sebuah Keputusan adalah pilihan yang diambil dari berbagai pilihan alternatif tindakan yang ada."*

Fungsi utama Sistem Pendukung Keputusan (DSS) adalahmelakukan analisis mengenai masalah-masalah terstruktur,mendukung masalah semi terstruktur dan tidak terstruktur. Kemampuan analisis DSS didasarkan pada teori atau model yang kuat, digabungkan dengan suatu antarmuka pengguna yang baik yang membuat sistem tersebut mudah digunakan.

Mengambil keputusan adalah proses yang melibatkan banyak langkah meliputi kecerdasan, rancangan, pilihan, dan implementasi. Menurut Herbert A.Simon (1960) dalam buku Kenneth C. Laudon dan Jane P. Laudon (*Management Imformation System, 2007, 150-151*) menyatakan ada 4 tahapan berbeda dalam mengambil keputusan: kecerdasan, rancangan, pilihan, dan implementasi. Keempat tahapan ini bersesuaian dengan empat langkah pemecahan masalah :

1. **Kecerdasan** (*intelegence*) terdiri ats menemukan, mengidentifikasi, dan memahami masalah yang terjadi pada organisasi – mengapa masalah itu terjadi, dimana, dan akibat apa yang dialami perusahaan.
2. **Rancangan** (*design*) melibatkan identifikasi dan pencarian berbagai solusi masalah yang dialami perusahaan.
3. **Pilihan** (*choice*) adalah tentang memilih alternatif solusi yang ada.
4. **Implementasi** (*implementation*) adalah tentang membuat alternatif yang dipilih dapat bekerja, dan tetap mengawasi seberapa baik kerja solusi tersebut.



Gambar 2.1 Tahapan Pengambilan Keputusan

Sistem Pendukung Keputusan merupakan system berbasis komputer interaktif, yang membantu pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah tidak terstruktur (Scott Morton, 1971 dalam Turban et al, 2005). Sedangkan menurut Daihani, Sistem Pendukung Keputusan adalah sistem yang berbasis komputer yang ditujukan untuk membantupengambil keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai masalah yangsemi terstruktur dan tidak terstruktur (Daihani,2001).

Sistem pendukung keputusan ini memadukan sumberdaya intelektual dari individu dengan kavabilitas computer untuk meningkatkan kualitas keputusan. Dengan demikian Sistem Pendukung Keputusan ini sering diartikan sebagai sitem pendukung berbasis computer yang digunakan oleh para pengambil keputusan (manager) yang menangani masalah-masalah yang tidak terstruktur.

Dari beberapa pengertian diatas dapat ditentukan karakteristik dari Sistem Pendudukan Keputusan, diantaranya :

1. Mendukung proses pengambilan keputusan yang menitikberatkan pada *management by perception*.
2. Terjadinya interface manusia atau mesin dimana manusia (*user*) tetap memegang control proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur dan tidak terstruktur.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.

5. Memiliki subsistem-subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan item.
6. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan informasi seluruh tingkatan manjemen.

2.2.1. Konsep Dasar Pendukung Keputusan

Sistem Pendukung Keputusan disampaikan pertama kalli oleh Michael S. Scott Morton pada awal tahun 1970-an dengan istilah *Management Decision System*. Sistem Pendukung Keputusan didefinisikan Morton sebagai "*Sistem Berbasis Interaktif*, yang membantu para pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah yang tidak terstruktur".

Sistem Pendukung Keputusan dibangun untu mendukung solusi atas suatu kejadian (masalah) atau untuk mengevaluasi suatu peluang dan disebut sebagai Aplikasi Sistem Pendukung Keputusan. Aplikasi Sistem Pendukung Keputusan digunakan dalam pengambilan keputusan menggunakan *Computer Based Information System*(CBIS) yang fleksibel, interaktif, dan dapat diadaptasi, serta dapat dikembangkan untuk mendukung solusi atas manajemen spesifik yang tidak terstruktur.

Aplikasi Sistem Pendukung Keputusan berbasis data, memberikan antar muka pengguna yang sanngat mudah dan dapat menggabungkan pemikiran-pemikira para pengambil keputusan. Sistem Pendukung Keputusan lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan analitis

dalam situasi yang kurang terstruktur dan kriteria yang kurang jelas. Sistem Pendukung Keputusan tidak dimaksudkan untuk mengotomatisasi pengambilan keputusan akan tetapi merupakan perangkat interakti yang member kemungkinan pengambil keputusan dalam melakukan berbagai analisis menggonakan model-model yang tersedia.

Tujuan Sistem Pendukung Keputusan mengacu pada 3 prinsip dasar sebagai berikut (Oetomo, 2002):

1. Struktur Masalah: untuk masalah yang terstruktur, penyelesaian dapat dilakukan dengan menggunakan rumus-rumus yang sesuai, sedangkan untuk masalah yang tidak terstruktur tidak dpst dikomputerisasi. Sementara iti Sistem Pendukung Keputusan dikembangkan khususnya untuk menyelesaikan masalah yang semi-terstruktur.
2. Dukungan Keputusan : Sistem Pendukung keputusan tidak dimsksudkan untuk menggantikan manajer, karena computer berada di bagian terstruktur, sementara manajer berada di bagian tak terstruktur untuk memberikan penilaian dan melakukan analisis. Manajer dan computer bekerja sama sebagai sebuah tim pemecah masalah semi terstruktur.
3. Efektifitas Keputusan : tujuan utama dari Sistem Pendukung Keputusan bukanlah mempersingkat waktu pengambilan keputusan, tetapi agar keputusan yang dihasilkan dapat lebih baik.

2.2.2. Komponen Sistem Pendukung Keputusan

Sistem Pendukung Keputusan terdiri dari 3 komponen utama atau sub sistem(Suryadi dan Ramdani, 1998), yaitu :

1. Sub Sistem Pengelolaan Data (*Database*)

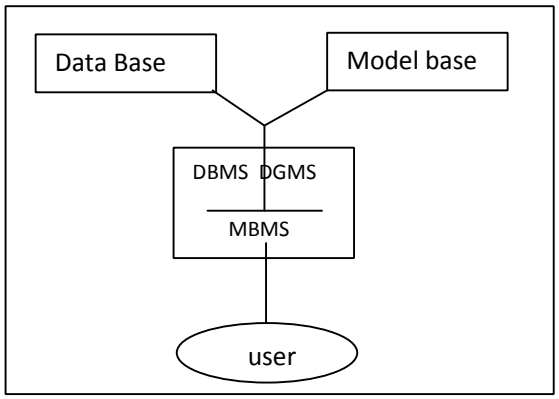
   Sub Sistem pengelolaan data merupakan komponen Sistem Pendukung Keputusanyang berguna sebagai penyedia data bagi system. Data tersebut disimpan dan diorganisasikan dalam sebuah basis data yang diorganisasikan oleh suatu system yang disebut dengan system manajemen basis data (*Database Managemant System*).

2. Sub Sistem Pengolahan Model (*Model Base*)

   Keunikan dari Sistem Pendukung Keputusan adalah kemampuannya dalam mengintegrasikan data dengan model-model keputusan. Model adalah suatu tiruan dari alam nyata.Kendala yang sering dihadapi dalam merancang suatu model, biasanya model yang dirancang tidak mampu mencerminkan seluruh variable alam nyata, sehingga keputusan yang diambil tidak sesuai dengan kebutuhan oleh karena itu, dalam menyimpan berbagai model harus diperhatikan dan harus dijaga flesibilitasnya. Hal lain yang perlu diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian keterangan dan penjelasan yang komprehensif mengenai model yang dibuat..

3. Sistem Pengolahan Dialog (*user interface*)

   Keunikan lainnya dari Sistem Pendukung Keputusan adalah adanya fasilitas yang mampu mengintegrasikan system yang terpasang dengan pengguna secara interaktif, yang dikenal dengan subistem dialog. Melalui system dialog, system dipermentasikan sehingga pengguna dapat berkomunikasi dengan system yang dirancang.



Gambar 2.2 Komponen Utama Sistem Pendukung Keputusan

2.3. Teorema Bayes

Bayes merupakan suatu teknik prediksi yang berbasis peluang (probabilistik) sangat sederhana yang berdasrkan pada penerapan aturan teori Bayes dengan asumsi ketidakbergantungan (independensi) yang kuat (naif). Sehingga dalam Teorema Bayes (Naif Bayes), model yang digunakan adalah "model fitur independen".

Dalam terori Bayes (khususnya Naif Bayes), yang dimaksud independensi yang kuat pada fitur adalah bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama. Contoh pada kasus klasifikasi hewan dengan fitur penutup kulit, melahirkan, berat, dan menyusui. Dalam hal ini memperlihatkan ada ketergantungan yaitu pada fitur menyusuinya, karena hewan yang menyusui biasanya melahirkan.

Inti dari Teorema Bayes ialah suatu penelitian yang cermat tentang tindakan apa atau alternative tindakan apa yang tersedia, sesudah itu dilanjut dengan mempertimbangkan resiko (untung/rugi) untuk tiap keadaan yang bakal terjadi dimasa depan. Antara Teorema Bayes dengan teori probabilitas terdapat

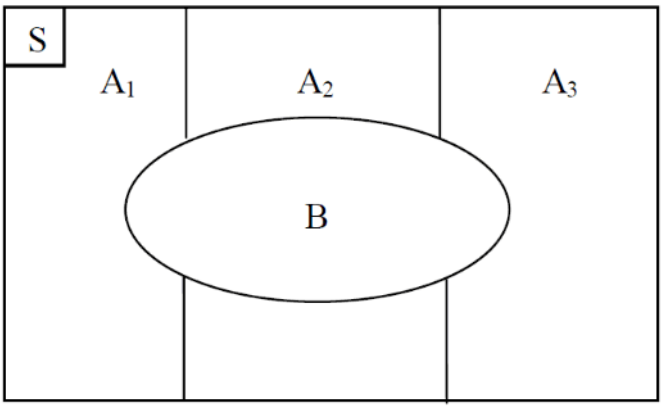
hubungan yang sangat erat, karena untuk membuktikan Teorema Bayes tidak terlepas dari penggunaan teori probabilitas, dengan kata lain teori probabilitas adalah konsep dasar bagi Teorema Bayes. [Bayesian Probability Journal, 2010]

Pada Teorema Bayes, terdapat beberapa bentuk probabilitas, yaitu sebagai berikut :

1. Probabilitas awal (*probabilitas prior*), yaitu probabilitas berdasarkan informasi yang tersedia.
2. Probabilitas bersyarat, yaitu probabilitas dimana terjadinya suatu peristiwa didahului oleh terjadinya peristiwa lain.
3. Probabilitas marginal, yaitu gabungan dari beberapa probabilitas.
4. Probabilitas posterior, yaitu probabilitas yang diperbaiki dengan adanya informasi tambahan.

Adapun Bunyi Teorema Bayes :

Misalkan A1 dan A2 adalah kelompok peristiwa yang *mutually exclusive* (dua peristiwa yang tidak dapat terjadi bersamaan) dan *exhaustive* (lengkap) merupakan kombinasi dari 2 peristiwa keseluruhannya. Dan B adalah peristiwa yang memotong (intersection) peristiwa A, seperti ditunjukkan dalam gambar 2.3. berikut:



Gambar 2.3 Diagram Venn dengan i = 1, 2

Dari gambar di atas, bagian B yang berada dalam A_1 mewakili daerah A_1 dan B, sedangkan bagian B yang berada di dalam A_2 mewakili daerah A_2 dan B. Probabilitas peristiwa A_1 dengan peristiwa B tertentu adalah:

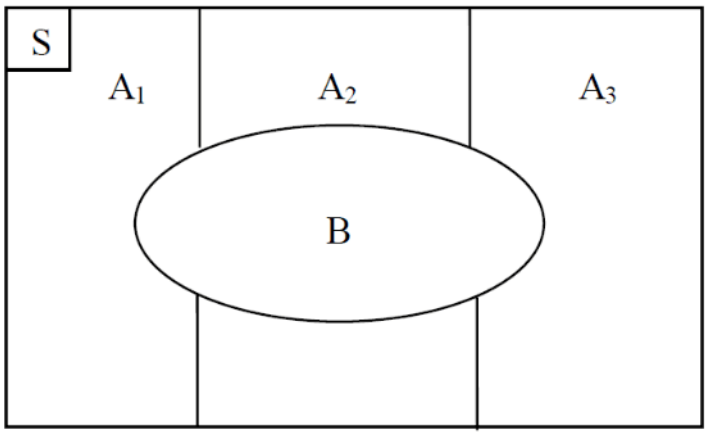
$$P(A_i \mid B) = \frac{P(A_i \cap B)}{P(B)}$$

Dengan cara yang sama probabilitas A2 dengan peristiwa B adalah :

$$P(A_2 \mid B) = \frac{P(A_2 \cap B)}{P(B)}$$

Selanjutnya untuk i = 3

A_1, A_2, A_3 adalah kelompok peristiwa yang *mutually exclusive* dan *exhaustive* seperti yang ditunjukkan oleh gambar 2.4 berikut :



Gambar 2.4 Diagram Venn dengan i = 1, 2, 3

Maka probabilitas peristiwa A_1 dengan peristiwa B adalah :

$$P(A_i \mid B) = \frac{P(A_i \cap B)}{P(B)}$$

Dengan cara yang sama probabilitas A_2 dengan peristiwa B adalah :

$$P(A_2 \mid B) = \frac{P(A_2 \cap B)}{P(B)}$$

Dengan cara yang sama probabilitas A_3 dengan peristiwa B adalah :

$$P(A_3 \mid B) = \frac{P(A_3 \cap B)}{P(B)}$$

Dan begitu juga selanjutnya bila i = 4,5,6, …

Teorema ini menerangkan hubungan antara probabilitas terjadinya peristiwa A dengan syarat peristiwa B telah terjadi dan probabilitas terjadinya peristiwa B dengan syarat peristiwa A telah terjadi. Teorema ini didasarkan pada prinsip bahwa tambahan informasi dapat memperbaiki probabilitas.

Berikut ini rumus umum teori Bayes :

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Dimana P(X|Y) menyatakan peluang suatu kejadian Y bila diketahui kejadian X terjadi. P(X) merupakan marginal probability dari kejadian X yang didapatkan dengan cara menjumlahkan X yang dipengaruhi oleh Y dengan kondisi fase.

$$P(X) = P(X \cap Y) + P(X \cap Y)$$

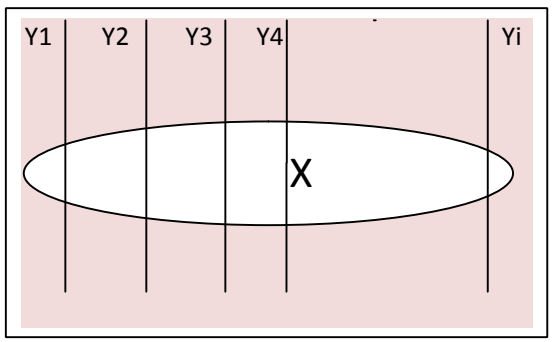
Maka untuk kondisi probabilitas P(Yi|X) untuk i=1…n didapatkan dengan peluang kejadian X :

$$P(Yi|X) = \frac{P(X \cap Yi)}{P(X)}$$

Menggunakan aturan perkalian untuk menemukan probabilitas gabungan dalam pembilang dan hukum probabilitas total dalam penyebut, didapatkan rumus sebagai berikut :

$$P(Yi|X) = \frac{P(X|Yi).P(Yi)}{\sum_{i=1}^{n} P(X|Yi).P(Yi)}$$

Jika digambarkan dalam bentuk diagram venn, teorema Bayes tersebut akan nampak seperti gambar 2.5.



Gambar 2.5. Diagram Venn untuk Bayesian Theorem

2.3.3Naive Bayes Classifier

Bayes merupakan teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema Bayes (atau aturan Bayes) dengan Asumsi independensi (ketidaktergantungan) yang kuat (naif). Dengan kata lain, dalam naive Bayes, model yang digunakan adalah "model fitur independen"

Dalam Bayes (terutama Naive Bayes), maksud independensi yang kuat pada fitur adalah bahwa sebuah fitur pada sebuah data *tidak ada kaitannya* dengan *adanya* atau *tidak adanya fitur yang lain* dalam data yang sama. Contoh: kasus klasifikasi hewan dengan fitur: penutup kulit, melahirkan, berat, dan menyusui. Dalam dunia nyata, hewan yang berkembang biak dengan cara melahirkan dipastikan hewan tersebut menyusui juga, disini ada ketergantungan pada fitur menyusui karena hewan yang menyusui biasanya melahirkan, atau hewan yang bertelur biasanya tidak menyusui. Dalam Bayes, hal tersebut tidak dipandang, sehingga masing-masing fitur seolah tidak ada hubungan apa-apa.

Prediksi Bayes didasarkan pada teorema Bayes dengan formula umum sebagai berikut :

$$P(H \mid E) = \frac{P(E \mid H) \times P(H)}{P(E)}$$

Penjelasan dari formula tersebut adalah sebagai berikut :

- P(H|E): Probabilitas akhir bersyarat (*Conditional Probability*) suatu hipotesis H terjadi jika diberikan evidence/bukti E terjadi
- P(E|H): Probabilitas sebuah evidence E terjadi akan mempengaruhi hipotesis H

- P(H): Probabilitas awal (priori) hipotesis H terjadi tanpa memandang evidence apapun
- P(E): Probabilitas awal (priori) evidence E terjadi tanpa memandang hipotesi/evidence yang lain

Ide dasar aturan Bayes adalah bahwa hasil dari hipotesis atau peristiwa (H) dapat diperkirakan berdasarkan pada beberapa evidence (E) yang diamati. Ada beberapa hal penting dari aturan Bayes tersebut, yaitu :

Sebuah probabilitas awal/priori H atau P(H), adalah probabilitas dari suatu hipotesis sebelum bukti diamati.

Sebuah probabilitas posterior H atau P(H|E), adalah probabilitas dari suatu hipotesis setelah bukti-bukti yang diamati ada.

Contoh, dalam suatu peramalan cuaca untuk memperkirakan terjadinya hujan, misal ada faktor yang mempengaruhi terjadinya hujan yaitu mendung. Jika diterapkan dalam Naïve Bayes maka probabilitas terjadinya hujan jika bukti mendung sudah diamati, dinyatakan dengan :

$$P(Hujan \mid Mendung) = \frac{P(Mendung \mid Hujan) \times P(Hujan)}{P(Mendung)}$$

P(Hujan|Mendung) adalah nilai probabilitas hipotesis hujan terjadi jika bukti mendung sudah diamati. P(Mendung|Hujan) adalah probabilitas bahwa mendung yang diamati akan mempengaruhi terjadinya hujan. P(Hujan) adalah probabilitas awal hujan tanpa memandang bukti apapun. P(Mendung) adalah probabilitas terjadinya mendung.

Contoh tersebut dapat dikembangkan dengan menambahkan beberapa observasi yang lain sebagai bukti. Semakin banyak bukti yang dilibatkan, semakin baik hasil prediksi yang diberikan. Namun, tentu saja bukti tersebut harus benar-benar berkaitan dan memberi pengaruh pada hipotesis. Dengan kata lain, penambahan bukti yang diamati tidak sembarangan. Misalnya,gempa bumi. Bukti gempa bumi tentu saja tidak berkaitan dengan hujan sehingga penambahan gempa bumi. Bukti gempa bumi tentu saja tidak berkaitan dengan hujan sehingga penambahan bukti gempa bumi dalam prediksi cuaca akan memberikan hasil yang salah. Walaupun ada bukti lain yang mempengaruhi cuaca seperti suhu udara, tetap saja ada nilai probabilitas P(Suhu) yang harus dinilai secara independen dalam teorema Bayes, yang sulit dilakukan karena suhu udara juga dipengaruhi oleh faktor lain seperti cuaca kemarin, mendung, polusi dan sebagainya. Jadi, penilaian probabilitas tersebut tidak memandang faktor yang lain. Inilah sebabnya disebut Naive Bayes (Bayes Naif).

Teorema Bayes juga bisa menangani beberapa evidence, misalnya ada E_1, E_2, dan E_3, maka probabilitas posterior untuk hipotesis hujan dapat dihitung dengan cara berikut :

$$P(H \mid E_1, E_2, E_3) = \frac{P(E_1, E_2, E_3 \mid H) \times P(H)}{P(E_1, E_2, E_3)}$$

Karena asumsi yang digunakan untuk bukti adalah independen, bentuk diatas dapat diubah menjadi:

$$P(H \mid E_1, E_2, E_3) = \frac{P(E_1 \mid H) \times P(E_2 \mid H) \times P(E_3 \mid H) \times P(H)}{P(E_1, E_2, E_3)}$$

Untuk contoh diatas, jika ditambahkan bukti suhu udara dan angin, bentuknya berubah menjadi

$$P(Hujan \mid Mendung, Suhu, Angin) = \frac{P(Mendung \mid Hujan) \times P(Suhu \mid Hujan) \times P(Angin \mid Hujan) \times P(Hujan)}{P(Mendung, Suhu, Angin)}$$

2.3.4 Bayesian Network

Bayesian Network adalah sebuah "*directed acyclic graph*" (DAG) dengan sebuah tabel probabilitas untuk setiap simpul dan merupakan representasi atribut proporsional dalam suatu domain dan garis yang menunjukan hubungan ketergantungan diantara variable-atribut. Dalam kaitannya dengan jaringan, simpul merupakan representasi dari kumpulan ancaman, kategori ancaman, dan asset-aset didalam jaringan.

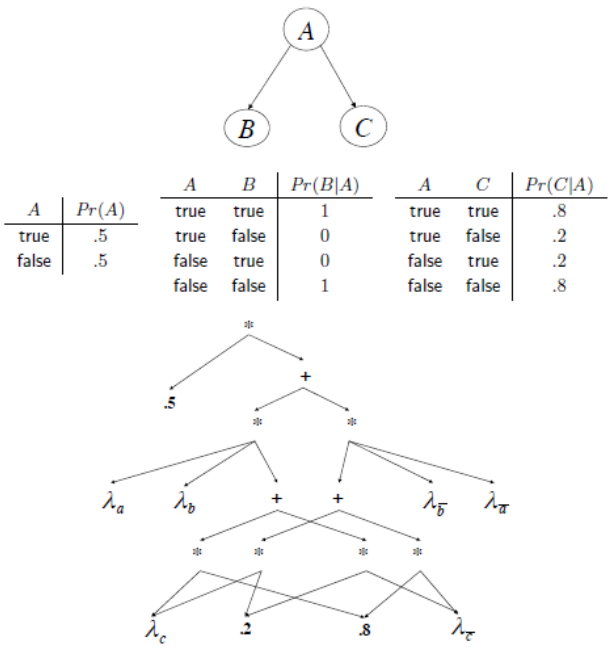
Ada 3 langkah utama dalam penyusunan sebuah Bayesian Network (Darwiche A, 2009), yaitu :

1. Menentukan atribut yang relevan dan nilai probabilitas dari masing-masing-masing atribut.
2. Menyusun jaringan grafik asiklik terarah atau *directed acyclic graph* (DAG).
3. Menentukan tabel peluang kondisi atau *Conditional Probability Table* (CPT)

Beberapa istilah yang digunakan dalam Bayesian Network :

- Notasi Graf yang menyatakan peluang bersyarat dalam suatu domain.
- Simpul menyatakan sebuah atribut acak.

- *Arc* (*directed edge*) menyatakan hubungan kausal langsung (*direct influence*).
- Arah sebuah *Arc* yaitu dari atribut sebab ke atribut akibat (*causal reasoning*) atau arah sebaliknya (*diagnostic reasoning*)
- Simpul *sibling* menyatakan atribut yang *conditionally independent* karena *parent*-nya.
- Distribusi peluang untuk setiap simpul terhadap *parent*-nya : P(Xi|*parent*(Xi)).
- Tidak ada siklus berulang di dalam *Bayesian Network*.

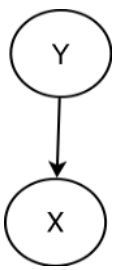


Gambar 2.6. Contoh notasi grafik dari Bayesian Network

2.3.5 Topologi Jaringan Bayesian Network

Ada tiga tipe koneksi yang terkait dengan struktur jaringan yang terdapat pada *Bayesian Network*, yakni koneksi *serial*, koneksi *divergen* dan koneksi *Convergen*. Identifikasi topologi jaringan berguna untuk menentukan ketergantungan informasi diantara simpul-simpul yang terdapat pada *Bayesian network*.

1. *Koneksi Serial*.Koneksi serial ini ditunjukkan oleh gambar 2.5, nilai dari simpul Y akan mempengaruhi probabilitas dari simpul X. Jika diasumsikan bahwa Y yaitu ancaman terhadap sutu jaringan X, maka gambae tersebut menjelaskan peluang kemungkinan terjadinya ekploitsi terhadap asset X yang dipengaruhi oleh peluang terjadinya ancaman Y.



Gambarb 2.7 KoneksiSerial dalam Bayesian Network

Maka dapat diperoleh perhitungan untuk koneksi serial pada gambar 2.7sebagai berikut:

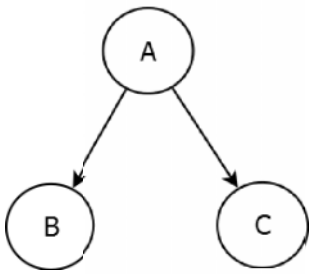
$P(X) = \sum_{Y \in T,F} P(X|Y)$

$= P(X|Y=T) + P(X|Y=F)$

$= P(X|Y=T) \cdot P(Y=T) + P(A|Y=F) \cdot P(Y=F)$

P(X) merupakan nilai *marginal probability* dari kemungkinan kejadian X yang dipengaruhi oleh penjumlahan dari peluang kejadian Y dengan kondisi *true*dan *false*.

2. *Koneksi divergen.* Nilai peluang dalam tipe koneksi divergen dapat mengalir pada semua jalur dari atribut A yang merupakan parent menuju B dan C yang merupakan child. Ketika nilai A diketahui maka jalur-jalur berada pada simpul child A akan terputus , sehingga A dan B akan saling tidak bergantung satu sama lain (*independent*).



Gambar 2.8. Koneksi divergent dalam Bayesian Network

Dari gambar 2.8 diatas dapat dilustrasikan bahwa sebagai peluang terjadinya suatu ancaman A yang dapat mempengaruhi peluang tereksploitasi dua atau lebih asset didalam jaringan, sehingga untuk menentukanpeluang suatu asset tereksploitasi dapat di gunakan rumusan sebagai berikut :

$P(B) = \sum_{A \in T,F} P(B|A)$

$= P(B|A=T) + P(B|A=F)$

$= P(B|A=T) \cdot P(A=T) + P(B|A=F) \cdot P(A=F)$

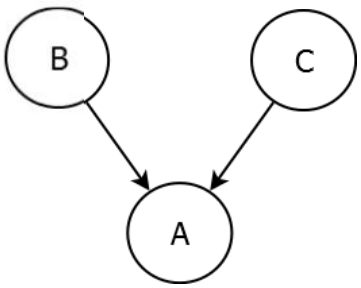
$P(C) = \sum_{A \in T,F} P(C|A)$

$= P(C|A=T) + P(C|A=F)$

$= P(B|A=T) \cdot P(A=T) + P(C|A=F) \cdot P(A=F)$

P(B) dan P(C) merupakan nilai *marginal probability* dari kemungkinan kejadian B yang dipengaruhi oleh penjumlahan dari peluang kejadian A dengan kondisi *true* dan *false*.

3. *Koneksi Convergent.* Gambar 2.9 menunjukkan bahwa jika tidak ada informasi yang diketahui pada simpul A, maka informasi diantara simpul-simpul parent dari A, yaitu B dan C akan *independent*, sehingga nilai peluang di masing-masing simpul *parent*-nya tidak akan mempengaruhi nilai peluang simpul-simpul lainnya. Tetapi ketika nilai P diketahui maka simpul-simpul *parent* A akan saling bergantung (*dependent*) sehingga nilai untuk simpul-simpul *parent* akan mempengaruhi nilai simpul lainnya.



Gambar 2.9 Koneksi Convergent dalam Bayesian Network

Jika kita perhatikan pada gambar 2.9 memiliki keterkaitan terhadap peluang tejadinya eksploitasi terhadap sauatu asset, yang dipengaruhi oleh peluang gabungan dari ancaman B dan C, dimana nilai peluang untuk masing-masing ancaman bersifat independent. Untuk menentukan nilai peluang asset tersebut tereksploitasi P(A) berdasarkan rumusan Bayes sebagai berikut :

$P(A) = \sum_{B,C \in T,F} P(A|B,C)$

= P(A|B= T, C= T) + P(A|B=F, C = T) + P(A|B=T, B=F) + P(A|B=F, C=F)

= P(A|B=T, C=T) . P(A=T). P(A|B=T) . P(B|T) + P(B|A=F) . P(A=T). P(A|B=F, C=T) . P(B=F) . P(C=T) + P(A|B=T . C=F) . P(B=T) . P(C=F) + P(A|B=F, C=F) . P(B=F) . P(C=F)

P(A) merupakan nilai *marginal probability* dari kemungkinan kejadian A yang dipengaruhi oleh penjumlahan dari peluang kejadian B dan C dengan kondisi *true* dan *false*.

2.3.6 Pembuatan *Conditioning Probability* (CPT)

Salah satu keunggulan dalam pembuatan model dengan memakai Bayesian Network adalah potensi ukuran dari *Conditioning Probability* (CPT) , sebagai contoh jika sebuah atribut E memiliki parent dari C1……Cn dan atribut tersebut hanya memiliki 2 nilai kondisi, maka diperlukan 2” parameter independent untuk menentukan CPT dari attribute. Untuk jumlah parent yang merupakan daftar ancaman yang mugkin terjadi terhadap suatu asset jaringan sebanyak 5 buah, maka diperlukan CPT dengan parameter independent untuk masing-masing kondisi sebanya $2^5 = 32$.

Number of Parent : n	Parameter Count 2”
2	4
5	8
6	64
10	1,024
20	1,048,576
30	1,073,741,82

Solusi yang dapat digunakan dalam menyederhanakan masalah tersebut adalah dengan menggunakan model Noisy-OR (Darwiche A. 2009). Untuk memahami model ini, cara terbaik adalah mentafsirkan parent C1…Cn sebagai penyebab dari atribut E sebagai efek bersama dari penyebab tersebut. Dengan kata lain setiap C_i merupakan penyebab terhadap efek Yang diterima oleh E. selain itu adanya atribut L *leak (bocoran*) dimaksudkan untuk mewakili penyebab lain dari E yang tidak dimodelkan secara *explicit.* Oleh karena itu ketika tidak ada kejadian Ci, yang berakibat terhadap E, maka akibat terhadap E dari kejadian, akan tetap terjadi yang terbentuk oleh atribut L.

2.3.7 Kelebihan dan Kekurangan Klasifikasi Naive Bayes

Kelebihan Naive Bayes (Domingos, 2007):

- Mudah diimplementasikan
- Hasilnya robbust untuk data yang memuat noisy dan untuk data yang tidak berkaitan
- Dapat menangani *missing value*
- Hasilnya cukup baik untuk sebagian besar kasus
- Bila asumsi saling bebas dipenuhi, maka tingkat akurasinya sangat tinggi dibandingkan metode lainnya

Kelebihan Naive Bayes (Domingos, 2007):

- Adanya asumsi saling bebas antar atributnya terkadang akan menurunkan tingkat akurasi
- Biasanya dalam kehidupan nyata selalu ada hubungan antar atribut sehingga asumsi saling bebas menjadi tidak dipenuhi

2.3.8 Algoritma-algoritma dalam Bayesian Network

Berikut ini adalah algoritma-algoritma dalam *Bayesian network* yang digunakan dalam pembangunan struktur *Bayesian network* .

Algoritma K2

Algortima K2 termasuk algoritma yang dikembangkan dengan menggunakan metode *search* and *scoring*. Algoritma ini dikenal juga sebagai metode Bayesian yaitu suatu metode yang dikembangkan oleh Cooper dan Herskovits dengan bertujuan untuk menemukan suatu *directed acyclic graph* (DAG) G' yang diduga akan memaksimumkan $score_B$ (d,G). $Score_B$ (d,G) yang maksimum diperoleh dengan menemukan nilai PAi untuk setiap i secara lokal yang akan memaksimumkan $score_B$ (d,Xi,PA). Satu-satunya operasi dalam algoritma ini adalah menambahkan sebuah parent ke suatu node. Pendekatan algoritma pencarian ini bersifat heuristic yaitu solusi yang didapatkan tidak dijamin sebagai solusi yang paling optimal tetapi secara logika adalah yang mendekati solusi optimal.

Algoritma K2 adalah:

Diasumsikan node–node adalah terurut yaitu bila Xi mendahului Xj maka tidak boleh ada edge dari Xj ke Xi. Misal Pred(Xi) adalah himpunan node yang mendahului Xi dalam urutan. Pertama ditetapkan himpunan parent PAi adalah kosong dan scoreB (d,Xi,PA) dihitung. Selanjutnya kita mengunjungi node dalam barisan berdasarkan urutannya. Ketika sampai pada node Xi, kemudian ditentukan node dalam Pred(Xi) yang paling besar dalam meningkatkan scoreB (d,Xi,PA). Kemudian secara "greedy", node tersebut akan ditambahkan pada PAi. Hal ini akan dilakukan terus menerus sampai tidak ada node yang meningkatkan scoreB (d,Xi,PA).

2.4. Belajar

Belajar adalah suatu proses perubahan tingkah laku individu melalui interaksi dengan lingkungan. (Hamalik Pemar : 2001) Menurut pengertian ini belajar merupakan suatu proses yakni suatu kegiatan dan bukan suatu hasil atau tujuan.

Yang menjadi hasil dari belajar bukan penguasan hasil latihan melainkan perubahan tingkah laku. Karena belajar merupakan suatu perubahan tingkah laku, maka diperlukan pembelajaran yang bermutu yang langsung menyenangkan dan mencerdaskan siswa.

Belajar adalah kegiatan yang berproses dan merupakan unsur yang sangat fundamental dalam setiap penyelenggaraan jenis dan jenjang pendidikan. Ini berarti bahwa berhasil atau gagalnya pencapaian tujuan pendidikan itu amat tergantung pada proses belajar yang dialami siswa, baik ketika ia berada di sekolah maupun di lingkungan rumah atau keluarga.5 Belajar atau learning, adalah perubahan yang secara relative berlangsung lama pada perilaku yang diperoleh dari pengalaman pengalaman. Belajar merupakan salahsatu bentuk perilaku yang amat penting bagi kelangsungan hidup manusia. Belajar mem-bantu manusia menyesuaikan diri (adaptasi) dengan lingkungan.

Menurut Gagne (1984), belajar didefinisikan sebagai proses dimana suatu organisme berubah perilakunya akibat suatu pengalaman.7 Menurut James O.Wittaker, “learning may be defined as the process by which behavior organites or is altered through training or experience”. Belajar dapat didefini-sikan sebagai proses dimana tingkahlaku ditimbulkan atau diubah melalui latihan atau pengalaman.

Menurut Howard L. Kingsley, "*learning is the process by which behavior (in the broader sense) I s organited or changed through practice or training*". Belajar adalah proses dimana tingkahlaku ditimbulkan atau diubah melalui praktek atau latihan.

2.4.1. Minat Belajar

2.4.1.1. Pengertian Minat Belajar

Minat berperan sangat penting dalam kehidupan peserta didik dan mempunyai dampak yang besar terhadap sikap dan perilaku. Siswa yang berminat terhadap kegiatan belajar akan berusaha lebih keras dibandingkan siswa yang kurang berminat.

Menurut Hilgard (1977 :19) memberi rumusan pengertian tentang minat sebagai berikut: "Interest is persisting tendency to pay attention to and enjoy some activity or content" yang berarti minat adalah kecenderungan yang tetap untuk memperhatikan dan mengenang beberapa kegiatan. Kegiatan yang diminati seseorang, diperhatikan terus-menerus yang disertai dengan rasa senang dan diperoleh suatu kepuasan.

Menurut Slameto (2003 : 57) minat adalah kecenderungan yang tetap untuk memperhatikan dan mengenang beberapa kegiatan. Kegiatan yang diminati siswa, diperhatikan terus-menerus yang disertai rasa senang dan diperoleh rasa kepuasan. Lebih lanjut dijelaskan minat adalah suatu rasa suka dan ketertarikan pada suatu hal atau aktivitas, tanpa ada yang menyuruh.

Minat adalah kecenderungan dalam diri individu untuk tertarik pada sesuatu objek atau menyenangi sesuatu objek (Sumadi Suryabrata, 1988 :109). Minat adalah sesuatu pemusatan perhatian yang tidak disengaja yang terlahir dengan penuh kemauannya dan yang tergantung dari bakat dan lingkungan.

Dari beberapa pengertian di atas dapat disimpulkan bahwa minat adalah kecenderungan tertarik pada sesuatu yang relatif tetap untuk lebih memperhatikan dan mengingat secara terus-menerus yang diikuti rasa senang untuk memperoleh suatu kepuasan dalam mencapai tujuan pembelajaran.

Dalam belajar diperlukan suatu pemusatan perhatian agar apa yang dipelajari dapat dipahami. Sehingga siswa dapat melakukan sesuatu yang sebelumnya tidak dapat dilakukan. Terjadilah suatu perubahan kelakuan. Perubahan kelakuan ini meliputi seluruh pribadi siswa; baik kognitip, psikomotor maupun afektif. Untuk meningkatkan minat, maka proses pembelajaran dapat dilakukan dalam bentuk kegiatan siswa bekerja dan mengalami apa yang ada di lingkungan secara berkelompok.

Sedangkan yang penulis maksudkan dengan minat belajar di sini adalah suatu kemampuan umum yang dimiliki siswa untuk mencapai hasil belajar yang optimal yang dapat ditunjukkan dengan kegiatan belajar.

2.4.1.2 Ciri-ciri Siswa Berminat dalam Belajar

Menurut Slameto (2003 :58) siswa yang berminat dalam belajar mempunyai ciri-ciri sebagai berikut:

1) Mempunyai kecenderungan yang tetap untuk untuk memperhatikan dan mengenang sesuatu yang dipelajari secara terus menerus.
2) Ada rasa suka dan senang pada sesuatu yang diminati.
3) Memperoleh suatu kebanggaan dan kepuasan pada sesuatu yang diminati. Ada rasa keterikatan pada sesuatu aktivitas-aktivitas yang diminati.
4) Lebih menyukai suatu hal yang menjadi minatnya daripada yang lainnya.
5) Dimanifestasikan melalui partisipasi pada aktivitas dan kegiatan.

Minat sangat besar pengaruhnya terhadap hasil belajar, karena apabila bahan pelajaran yang dipelajari tidak sesuai dengan minat, siswa tidak akan belajar dengan baik sebab tidak menarik baginya. Siswa akan malas belajar dan tidak akan mendapatkan kepuasan dari pelajaran itu. Bahan pelajaran yang menarik minat siswa, lebih mudah dipelajari sehingga dapat mingkatkan prestasi belajar.

Minat terhadap sesuatu hal tidak merupakan yang hakiki untuk dapat mempelajari hal tersebut, asumsi umum menyatakan bahwa minat akan membantu seseorang mempelajarinya. Membangkitkan minat terhadap sesuatu pada dasarnya adalah membantu siswa melihat bagaimana hubungan antara materi yang diharapkan untuk dipelajari dengan diri sendiri sebagai individu.

Menurut Slameto (2003 :180) proses ini berarti menunjukkan pada siswa bagaimana penetahuan atau kecakapan tertentu mempengaruhi dirinya, melayani tujuan-tujuannya, dan memuaskan kebutuhan-kebutuhannya. Bila siswa

menyadari bahwa belajar merupakan suatu alat untuk mencapai tujuan yang dianggap penting, dan bila siswa melihat bahwa hasil dari pengalaman belajar akan membawa kemajuan pada dirinya, ia akan lebih berminat untuk mempelajarinya.

Minat pada dasarnya merupakan penerimaan akan suatu hubungan antara diri sendiri dengan sesuatu di luar diri, semakin kuat atau dekat hubungan tersebut, semakin besar minatnya.

Jika terdapat siswa yang kurang berminat dalam belajar dapat diusahakan agar mempunyai minat yang lebih besar dengan cara menjelaskan hal-hal yang menarik dan berguna bagi kehidupannya serta berhubungan dengan cita-cita yang berkaitan dengan materi yang dipelajari. Minat dapat diekspresikan melalui suatu pernyataan yang menunjukkan bahwa siswa lebih menyukai suatu hal dari pada hal lainnya, dapat pula dimanifestasikan melalui partisipasi dalam suatu aktivitas. Siswa yang memiliki minat terhadap subyek tertentu cenderung untuk memberikan perhatian yang lebih besar terhadap subyek tersebut. Minat tidak dibawa sejak lahir, melainkan diperoleh kemudian. Minat terhadap pelajaran mempengaruhi belajar selanjutnya serta mempengaruhi minat-minat baru. Menurut ilmuwan pendidikan cara yang paling efektif untuk membangkitkan minat belajar pada siswa adalah dengan menggunakan minat-minat siswa yang telah ada dan membentuk minat-minat baru pada diri siswa. Hal ini dapat dicapai dengan jalan memberikan informasi pada siswa mengenai hubungan antara suatu bahan pengajaran yang akan diberikan dengan bahan pengajaran yang lalu, menguraikan kegunaan bagi siswa dimasa yang akan datang. Minat dapat

dibangkitkan dengan cara menghubungkan materi pelajaran dengan suatu berita sensasional yang sudah diketahui kebanyakan siswa.

Indikator-indikator minat belajar siswa terdiri dari: adanya perhatian, adanya ketertarikan, dan rasa senang. Indikator adanya perhatian dijabarkan menjadi tiga bagian yaitu: perhatian terhadap bahan pelajaran, memahami materi pelajaran dan menyelesaikan soal-soal pelajaran. Ketertarikan dibedakan menjadi ketertarikan terhadap bahan pelajaran dan untuk menyelesaikan soal-soal pelajaran. Rasa senang meliputi rasa senang mengetahui bahan belajar, memehami bahan belajar, dan kemampuan menyelesaikan soal-soal.

2.5. Proses Peminatan

Proses peminatan di SMA Negeri 14 Bandung mutlak harus dilakukan guna memberikan solusi bagi siswa/siswi kelas X melalui (1) Perolehan Nilai UN SMP, (2) hasil Prestasi Akademis yakni Nilai Raport SMP semester 1,2,3,4, dan 5, (3) hasil penelusuran minat dan bakat (psikotes) yang direkomendasi oleh pakar yang kompeten, (4) hasil Placemen Test yang diselenggarakan pihak Sekolah, (5) berdasarkan minat siswanya sendiri, serta (6) dukungan pihak orang tuanya, sesuai dengan Permendikbud no 69 tahun 2013.

2.6. Struktur Kurikulum

Pengembangan kurikulum SMA Negeri 14 Bandung tahun pelajaran 2013-2014 mencakup hal-hal sebagai berikut:

1. Kerangka dasar dan struktur kurikulum yang merupakan pedoman dalam pengembangan kurikulum SMA Negeri 14 Bandung;
2. Beban belajar bagi peserta didik pada SMA Negeri 14 Bandung yang didasarkan pada hasil analisis konteks, analisis keunggulan lokal serta potensi *dan minat peserta didik;*
3. Kurikulum SMA Negeri 14 Bandung dikembangkan berdasarkan hasil revisi kurikulum tahun 2012-2013, pemanfaatan hasil *analisis kondisi riil sekolah, terutama tenaga pendidik dan sarana-prasaran, serta analisis terhadap kurikulum 2013.*
4. Kalender pendidikan SMA Negeri 14 Bandung disusun berdasarkan hasil perhitungan minggu efektif untuk tahun pelajaran 2013-2014.

Tahun Pelajaran 2013-2014 SMA pelaksana Kurikulum 2013 memiliki kewajiban untuk menyusun KTSP dan melaksanakannya dengan mencakup dua kurikulum sekaligus, yaitu kurikulum 2006 dan Kurikulum 2013.

Sebagai salah satu SMA pelaksana Kurikulum 2013, maka SMA Negeri 14 Bandung memilki struktur kurikulum untuk kedua kurikulum tersebut yang diberlakukan bagi kelas X (kurikulum 2013) dan kelas XI dan kelas XII (kurikulum 2006).

Untuk itu, kelas XI dan kelas XII masih mengikuti Peraturan Pemerintah No. 19 Tahun 2005 tentang Standar Nasional Pendidikan pasal 6 ayat (1) yang menyatakan bahwa kurikulum untuk jenis pendidikan umum terdiri atas :

a. Kelompok mata pelajaran agama dan akhlak mulia
b. Kelompok mata pelajaran kewarganegaraan dan kepribadian.
c. Kelompok mata pelajaran ilmu pengetahuan dn teknologi
d. Kelompok mata pelajaran estetika.
e. Kelompok mata pelajaran jasmani, olehraga dan kesehatan.

No	Kelompok Mata Pelajaran	Cakupan
1	Agama dan Akhlak Mulia	Kelompok mata pelajaran agama dan akhlak mulia dimaksudkan untuk membentuk peserta didik menjadi manusia yang beriman dan bertaqwa kepada Tuhan yang Maha Esa serta berakhlak mulia. Akhlak mulia mencakup etika, budi pekerti atau moral sebagai perwujudan dari pendidikan agama.
2	Kewarganegaraan dan Kepribadian	Kelompok mata pelajaran kewarganegaraan dan kepribadian dimaksudkan untuk peningkatan kesadaran dan wawasan peserta didik akan status hak dan kewajibannya dalam kehidupan bermasyarakat, berbangsa dan bernegara serta peningkatan kualitas dirinya sebagai manusia. Kesadaran dan wawasan termasuk wawasan kebangsaan, jiwa patriotisme bela Negara, penghargaan terhadap hak-hak asasi manusia, kemajemukan bangsa, pelestarian lingkungan hidup, kesetaraan jender, demokrasi, tanggungjawab social, ketaatan pada hokum, ketaatan membayar pajak,dan sikap serta prilaku anti korupsi, kolusi, dan nepotisme.
3	Ilmu Pengetahuan dan Teknologi	Kelompok mata pelajaran ilmu pengetahuan dan teknologi dimaksudkan untuk memperoleh kompetensi lanjut ilmu pengetahuan dan teknologi serta membudayakan berfikir ilmiah secara kritis, kreatif dan mandiri.
4	Estetika	Kelompok mata pelajaran estetika dimaksudkan untuk meningkatkan sensitifitas, kemampuan mengekspresikan dan kemampuan mengapresiasi keindahan dan harmoni. Kemampuan mengapresiasi dan mengekspresikan keindahan serta harmoni mencakup apresiasi dan ekspresi baik dalam kehidupan individual sehinga mampu menikmati dan mensyukuri hidup, maupun dalam kehidupan kemasyarakatan sehingga mampu menciptakan kebersamaan yang harmonis.

No	Kelompok Mata Pelajaran	Cakupan
5	Jasmani, Olah raga dan kesehatan	Kelompok mata pelajaran jasmani, Olah raga dan kesehatan pada SMA dimaksudkan untuk meningkatkan potensi fisik serta membudayakan sikap sportif, disiplin, kerjasama, dan hidup sehat. Budaya hidup sehat termasuk kesadaran, sikap, dan prilaku hidup sehat yang bersifat individual ataupun kolektif kemasyarakatan seperti keterbebasan dari prilaku seksual bebas, kecanduan narkoba, HIV atau AIDS, demam berdarah, muntaber, dan penyakit lain yang potensial mewabah.

Untuk kelas X mengikuti pola dan ketentuan Kurikulum 2013, yaitu adanya kelompok mata Pelajaran Wajib A dan Wajib B, Kelompok Peminatan, Lintas Minat, dan Pendalaman Minat yang kesemuanya mengusung ke pencapaian Standar Kompetensi Lulusan sebagai Berikut:

No.	Domain	Kompetensi
1.	Sikap	Memiliki perilaku yang mencerminkan sikap orang beriman, berakhlak mulia, berilmu, percaya diri, dan bertanggung jawab dalam berinteraksi secara efektif dengan lingkungan sosial dan alam serta dalam menempatkan diri sebagai cerminan bangsa dalam pergaulan dunia.
2.	Pengetahuan	Memiliki pengetahuan faktual, konseptual, prosedural, dan metakognitif dalam ilmu pengetahuan, teknologi, seni, dan budaya dengan wawasan kemanusiaan, kebangsaan, kenegaraan, dan peradaban terkait penyebab serta dampak fenomena dan kejadian.
3.	Keterampilan	Memiliki kemampuan pikir dan tindak yang efektif dan kreatif dalam ranah abstrak dan konkret sebagai pengembangan dari yang dipelajari di sekolah secara mandiri.

Kompetensi Lulusan dapat dicapai melalui Kompetensi Inti sebagai berikut;

No.	Domain	Kompetensi Inti
1.	Sikap	Sesuai dengan karakteristik sikap, maka salah satu alternatif yang dipilih adalah proses afeksi mulai dari menerima, menjalankan,menghargai,menghayati,hingga mengamalkan. Seluruh aktivitas pembelajaran erorientasi pada tahapan kompetensi yang mendorong siswa untuk melakuan aktivitas tersebut.
2.	Pengetahuan	Pengetahuandimilikimelaluiaktivitasmengetahui, memahami, menerapkan,menganalisis, mengevaluasi, hingga mencipta. Karakteritik aktivititas belajar dalam domain pengetahuan ini memiliki perbedaan dan kesamaan dengan aktivitas belajar dalam domain keterampilan. Untuk memperkuat pendekatan saintifik, tematik terpadu, dan tematik sangat disarankan untuk menerapkan belajar berbasis penyingkapan/penelitian (discovery/inquiry learning). Untuk mendorong peserta didik menghasilkan karya kreatif dan kontekstual, baik individual maupun kelompok, disarankan menggunakan pendekatan pembelajaran yang menghasilkan karya berbasis pemecahan masalah (project based learning).
3.	Ketampilan	Keterampilandiperolehmelaluikegiatan mengamati, menanya, mencoba, menalar, menyaji, dan mencipta. Seluruh isi materi (topik dan subtopik) mata pelajaran yang diturunkan dari keterampilan harus mendorong siswa untuk melakukan proses pengamatan hingga penciptaan. Untuk mewujudkan keterampilan tersebut perlu melakukan pembelajaran yang menerapkan modus belajar berbasis penyingkapan/penelitian (discovery/ inquirylearning)dan pembelajaran yang menghasilkan karya berbasis pemecahan masalah (project based learning).

2.7 Implementasi

Penelitian ini, dapat diterapkan ke dalam sebuah sistem informasi peminatan dengan basis desktop (*stand alone*) atau web based (jaringan internet). Implementasi dapat dilakukan oleh pihak sekolah dengan bantuan bahasa

pemrograman PHP, database MySQL, serta HTML, Javascript dan CSS sebagai *client side*-nya.

Suatu sistem pada komputer dapat berjalan dengan baik, jika ditunjang oleh tiga komponen yang saling melengkapi, yaitu perangkat keras (*hardware*), perangkat lunak (*software*), dan pengguna (*brainware*).

2.7.1 Perangkat Keras (*Hardware*)

Sistem informasi pendukung keputusan untuk peminatan dapat dijalankan pada komputer server, PC, laptop. Aplikasi memerlukan sebuah server jika akan dijalankan melalui internet, atau melalui intranet dengan jumlah pengguna yang cukup banyak. Berikut spesifikasi kebutuhan perangkat keras:

Spesifikasi minimal server:

1. Prosessor 2 GHz.
2. RAM 2 GB.
3. Hardisk 160 GB.
4. Bandwidth 100 Kbps

Spesifikasi minimal pemasangan pada PC/Laptop:

1. Prosessor 1 GHz.
2. RAM 1 GB.
3. Hardisk 80 GB.
4. VGA Card 128 MB
5. Resolusi monitor 1200 x 800 piksel (optimal).
6. Mouse dan Keyboard standar.
7. Sound Card

2.7.2 Perangkat Lunak (*Software*)

Berikut spesifikasi perangkat lunak yang digunakan untuk membuat dan membuka sistem informasi pendukung keputusan untuk peminatan:

1. *Operating Sistem*: Windows XP, Windows Vista, Windows 7, Linux (untuk klien); Windows Server 2008, Linux Server (untuk server).
2. *Pengolah Citra*: Adobe Photshop CS2.
3. *Pengolah Audio*: Cool Edit, Total Video Converter.
4. *Server Script:* PHP dengan Framework CodeIgniter
5. *OCR:* ABBYY FineReader 9.0
6. *Tools* UML : Enterprise Architect ver. 8.0
7. *Client Script:* CSS, Javascript, jQuery, HTML5
8. *Browser:* Firefox 8.0, Chrome 10.0, Opera 10.0
9. *Script Editor:* Notepad++, Adobe Dreamweaver

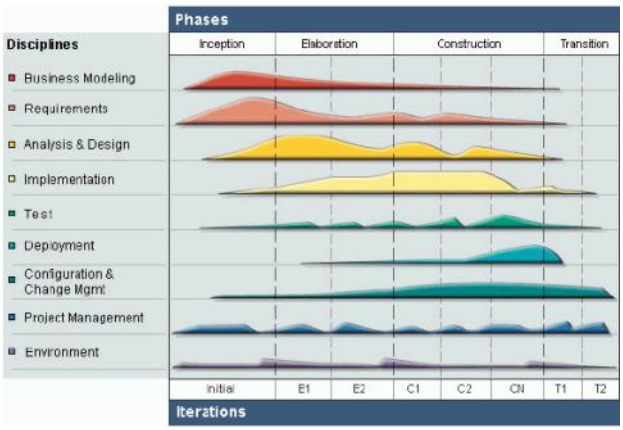
2.7.4 Pemodelan Perangkat Lunak

Agar penelitian ini dapat diterapkan ke dalam suatu sistem aplikasi, selain dibangun pemodelan yang dirancang menggunakan model Bayes, perlu dibuat juga model dalam pengembangannya dalam bentuk aplikasi.

2.7.4Metode Pengembangan RUP (*Rational Unified Process*)

RUP adalah metodologi pengembangan perangkat lunak, yang diformulasikan oleh *Rational Software Corporation* (sekarang menjadi salah satu divisi IBM), yang menggunakan UML (*Unified Modeling Language*) sebagai bahasa pemodelan selama periode pengembangan dan *iterative incremental* sebagai model siklus pengembangan perangkat lunak. Model ini membagi suatu

sistem aplikasi menjadi beberapa komponen sistem dan memungkinkan para developer aplikasi untuk menerapkan metoda *iterative* (analisis, desain, implementasi dan pengujian) pada tiap komponen.



Gambar 2.10. Aristektur RUP

Melalui gambar 1.2 Arsitektur RUP, dapat dilihat bahwa RUP memiliki:

1. **Dimensi pertama** digambarkan secara horizontal. Dimensi ini mewakiliaspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari *phase* selanjutnya. Setiap *phase* dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.
2. **Dimensi kedua** digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business*

Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration dan *Change Manegement, Project Management, Environtment.*

Pada penggunaan kedua standard tersebut diatas yang berorientasi obyek (*object orinted*) memiliki manfaat yaitu:

a. *Improve productivity*

*Standard*ini dapat memanfaatkan kembali komponen-komponen yang telah tersedia/dibuat sehingga dapat meningkatkan produktifitas.

b. *Deliver high quality system*

Kualitas sistem informasi dapat ditingkatkan sebagai sistem yang dibuat pada komponen-komponen yang telah teruji (*well-tested* dan *well-proven*) sehingga dapat mempercepat *delivery* sistem informasi yang dibuat dengan kualitas yang tinggi.

c. *Lower maintenance cost*

*Standard*ini dapat membantu untuk menyakinkan dampak perubahan yang terlokalisasi dan masalah dapat dengan mudah terdeteksi sehingga hasilnya biaya pemeliharaan dapat dioptimalkan atau lebih rendah dengan pengembangan informasi tanpa *standard* yang jelas.

d. *Facilitate reuse*

Standard ini memiliki kemampuan yang mengembangkan komponen-komponen yang dapat digunakan kembali untuk pengembangan aplikasi yang lainnya.

e. *Manage complexity*

*Standard*ini mudah untuk mengatur dan memonitor semua proses dari semua tahapan yang ada sehingga suatu pengembangan sistem informasi yang amat kompleks dapat dilakukan dengan aman dan sesuai dengan harapan semua manajer proyek IT/IS yakni*deliver good quality software within cost and schedule time and the users accepted.*

RUP membagi tahapan pengembangan perangkat lunaknya ke dalam 4 fase yaitu sebagai berikut:

1. *Inception*

Merupakan tahap untuk mengidentifikasi sistem yang akan dikembangkan. Aktivitas yang dilakukan pada tahap ini antara lain mencakup analisis sistem eksisting, perumusan sistem target, penentuan arsitektur global target, identifikasi kebutuhan, perumusan persyaratan (fungsional, performansi, keamanan, GUI, dan lain-lain), perumusan kebutuhan pengujian (level unit, integrasi, sistem, performansi, fungsionalitas, keamanan, dan lain-lain), pemodelan diagram UML (*use case diagram* dan *activitydiagram*) serta pembuatan dokumentasi.

2. *Elaboration*

Merupakan tahap untuk melakukan desain secara lengkap berdasarkan hasil analisis di tahap *inception.* Aktivitas yang dilakukan pada tahap ini antara lain mencakup pembuatan desain arsitektur subsistem (*architecture pattern*), desain komponen sistem, desain format data (protokol komunikasi), desain database, desain antarmuka/tampilan, desain peta aliran tampilan, penentuan *design pattern* yang digunakan, pemodelan diagram UML (*sequ*

3. *Construction*

Merupakan tahap untuk mengimplementasikan hasil desain dan melakukan pengujian hasil implementasi. Pada tahap awal *construction*, ada baiknya dilakukan pemeriksaan ulang hasil analisis dan desain, terutama desain pada domain perilaku (*sequencediagram*) dan domain struktural (*classdiagram*, *component*, *deployment*). Apabila desain yang dibuat telah sesuai dengan analisis sistem, maka implementasi dengan bahasa pemrogramanan tertentu dapat dilakukan. Aktivitas yang dilakukan pada tahap ini antara lain mencakup pengujian hasil analisis dan desain (misal menggunakan *Class Responsibility Collaborator* untuk kasus pemrograman berorientasi obyek), pendataan kebutuhan implementasi lengkap (berpedoman pada identifikasi kebutuhan di tahap analisis), penentuan *coding pattern* yang digunakan, pembuatan program, pengujian, optimasi program, pendataan berbagai kemungkinan pengembangan/perbaikan lebih lanjut, dan pembuatan dokumentasi.

4. *Transition*

Merupakan tahap untuk menyerahkan sistem aplikasi ke konsumen (*roll-out*), yang umumnya mencakup pelaksanaan pelatihan kepada pengguna dan testing beta aplikasi terhadap ekspetasi pengguna.

2.7.6UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan

sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software* (http://www.omg.org).

Secara resmi bahasa UML dimulai pada bulan oktober 1994, ketika Rumbaugh bergabung dengan Booch untuk membuat sebuah *project* pendekatan metoda yang *uniform*/seragam dari masing-masing metoda mereka. Saat itu baru dikembangkan *draft* metoda UML versi 0.8 dan diselesaikan serta di rilis pada bulan oktober 1995. Bersamaan dengan saat itu, Jacobson bergabung dan UML tersebut diperkaya ruang lingkupnya dengan metoda OOSE sehingga muncul *release version* 0.9 pada bulan Juni 1996. Hingga saat ini sejak Juni 1998 UML *version* 1.3 telah diperkaya dan direspon oleh OMG (*Object Management Group*), *Anderson Consulting*, *Ericsson*, *Platinum Technology*, *ObjectTime Limited*, dll serta di pelihara oleh OMG yang dipimpin oleh Cris Kobryn.

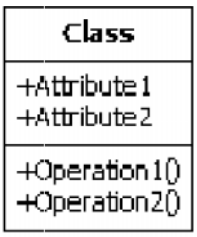
UML adalah standar dunia yang dibuat oleh *Object Management Group* (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object-oriented* dan *software component*.

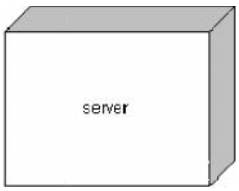
Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML seperti *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*). Terdapat 3 (tiga) macam yang ada dalam *building block* yaitu katagori benda/*things*, hubungan, dan diagram. Benda/*things* adalah abstraksi yang pertama dalam sebuah model, hubungan sebagai alat komunikasi dari benda-benda, dan diagram sebagai kumpulan / group dari benda-benda/*things*.

2.7.7Benda/*Things*

Benda/*Things* adalahhal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemen-elemen lainnya dari sebuah konsep dan atau fisik. Bentuk dari beberapa benda/*thing*dapat diamati pada tabel 2.1 Berikut ini:

Tabel 2.1 *Benda/Things*

Nama	Simbol	Keterangan
Kelas		Sekelompok dari objek yang mempunyai atribute, operasi, hubungan yang semantik.Sebuah kelas mengimplementasikan 1 atau lebih *interfaces*.Sebuah kelas dapat digambarkan sebagai sebuah persegi panjang, yang mempunyai sebuah nama, atribute, dan metoda pengoperasiannya.
Interface	 *Ispelling*	Sebuah antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen. *'Interface'*/antar-muka mendefinisikan sebuah set/kelompok dari spesifikasi pengoperasian, umumnya digambarkan dengan sebuah lingkaran yang disertai dengan namanya. Sebuah antar-muka berdiri sendiri dan umumnya merupakan pelengkap dari kelas atau komponen
Collaboration		Interaksi dan sebuah kumpulan / kelompok dari kelas-kelas/elemen-elemen yang bekerja secara bersama-sama. C*ollaborations* mempunyai struktura dan dimensi. Pemberian sebuah kelas memungkinkan berpartisipasi didalam beberapa *collaborations* dan digambarkan dengan sebuah '*elips*' dengan garis terpotong-potong.
Use Case		Rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau

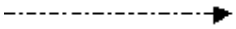
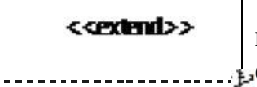
		diawasi oleh sebuah aktor. '*use case*' digunakan untuk membentuk tingkah-laku benda/ *things* dalam sebuah model serta di realisasikan oleh sebuah collaboration. Umumnya '*use case*' digambarkan dengan sebuah '*elips*' dengan garis yang solid, biasanya mengandung nama.
Node		Merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem, contohnya adalaha sebuah komputer, umumnya mempunyai sedikitnya *memory* dan *processor*. Sekelompok komponen mungkin terletak pada sebuah *node* dan juga mungkin akan berpindah dari node satu ke node lainnya. Umumnya node ini digambarkan seperti kubus serta hanya mengandung namanya.

2.7.8Hubungan/*Relationship*

Ada 6 macam hubungan didalam penggunaan UML, yaitu; *dependency, association, generalization, realization, Extend dan Include*, dapat diamatipada tabel 2.2berikut ini:

Tabel 2.2 *Hubungan/Relationship*

Nama	Simbol	Keterangan
Dependency		Hubungan semantik antara dua benda/*things* yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula. Umumnya sebuah *dependency* digambarkan sebuah panah dengan garis terputus-putus.
Association	0..1_____* employee employee	Hubungan antar benda struktural yang terhubung diantara objek. Kesatuan obyek yang terhubung merupakan hubungan khusus, yang menggambarkan

		sebuah hubungan struktural diantara seluruh atau sebagian. Umumnya *assosiation* digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status.
Generalization		Menggambarkan hubungan khusus dalam obyek anak/*child* yang menggantikan obyek *parent*/induk . Dalam hal ini, objek anak memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada objek induk.
Realization		Merupakan hubungan semantik antara pengelompokkan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara *interface* dan kelas atau *elements*, serta antara *use cases* dan *collaborations*.
Extend		Y *extend* X berarti *use case* X menggunakan *use case* Y secara opsional (tergantung keputusan saat runtime atau implementasi system)
Include		X *include* Y berarti *use case* X menggunakan *use case* Y sepenuhnya. *Include* digunakan untuk menghindari redundansi *flow of event* (seperti fungsi).

2.7.8.1Diagram

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 8 diagram, yaitu: *Diagram Use Case, Diagram Class, Diagram Package, Diagram Sequence, Diagram Collaboration, Diagram StateChart, Diagram Activity, dan Diagram Deployment.*

Semakin kompleks bentukan sistem yang akan dibuat, maka semakin sulit komunikasi antara orang-orang yang saling terkait dalam pembuatan dan pengembangan *software* yang akan dibuat. Pada masa lalu, UML mempunyai peranan sebagai *software blueprint* (gambaran) *language* untuk analisis sistem, *designer*, dan *programmer*. Sedangkan pada saat ini, merupakan bagian dari *software trade* (bisnis).

Semua diagram diatas tidak perlu di buat semuanya, hanya yang akan dibutuhkan oleh sistem yang akan kita buat. Beberapa diagram yang akan di buat dalam pembuatan perangkat lunak ini diantaranya:

a. use case diagram

b. class diagram

c. activity diagram

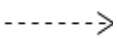
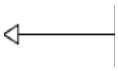
d. sequence diagram

e. deployment diagram

2.7.8.2 *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Berikut ini merupakan simbol-simbol *Use Case Diagram* dalam tabel 2.3:

Tabel 2.3 *Simbol-simbol dalam Use Case Diagram*

Simbol	**Keterangan**
Actor 	Menggambarkan pihak yang berhubungan dengan sistem baik itu merupakan pengguna atau sistem lainnya yang berada dari sistem yang sedang dibahas.
Use Case 	Menggambarkan suatu Kegiatan (Aktifitas) yang terjadi pada sistem.
Use Case Realization 	Menggambarkan suatu Kegiatan (Aktifitas) yang terjadi pada sistem dan hanya dapat dilihat di dalam sistem.
Relasi/Association	Menghubungkan keterkaitan antara Use Case dengan Actors atau dengan Use Case lainnya.
Dependency 	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
Generalization 	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
Include 	Menspesifikasikan bahwa *use case* sumber secara *eksplisit.*
Extend 	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.

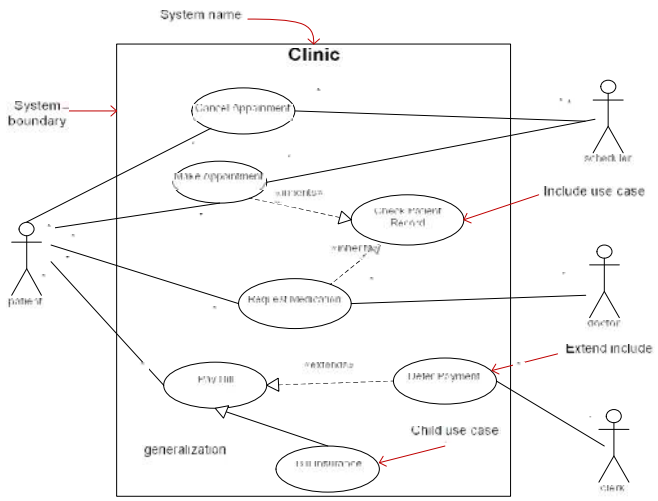
Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extenduse case* lain dengan *behaviour*-nya sendiri.Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Contoh *use case diagram*dapat dilihat pada gambar 2.2 berikut ini:



Gambar 2.11 *Contoh use case diagram*

2.7.8.3*Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

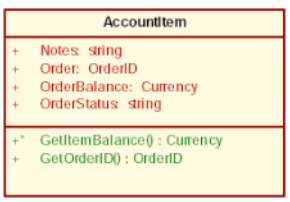
Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.*Class* memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
c. *Public*, dapat dipanggil oleh siapa saja

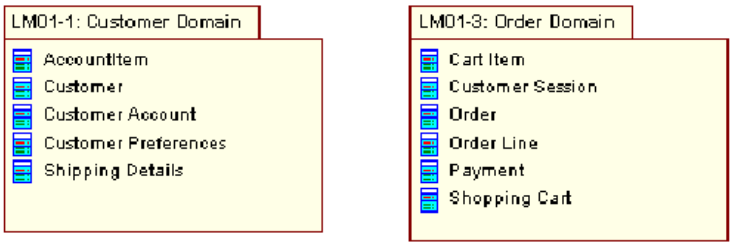
Contoh *class* pada *class diagram* dapat dilihat pada gambar 2.3 berikut:



Gambar 2.12*Contoh class pada class diagram*

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*. Contoh *package*pada *class diagram* dapat dilihat pada gambar 2.4 berikut:



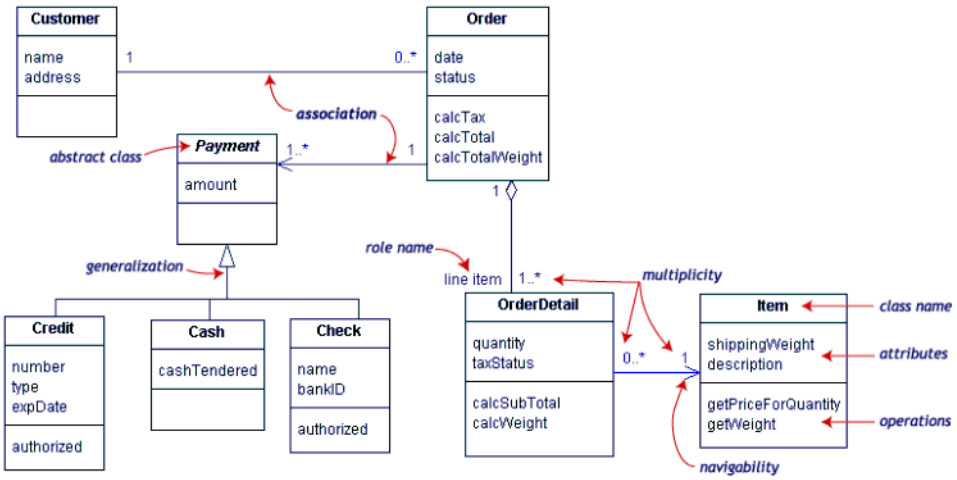
Gambar 2.13 Contoh *package* pada *class diagram*

Hubungan antar *class*pada *class diagram:*

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. *Agregasi*, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

Contoh *class diagram*dapat dilihat pada gambar 2.5 berikut ini:



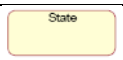
Gambar 2.14 *Contoh class diagram*

2.7.8.5 *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Beberapa simboldalam *activity diagram*digambarkan dalam tabel 2.4 berikut:

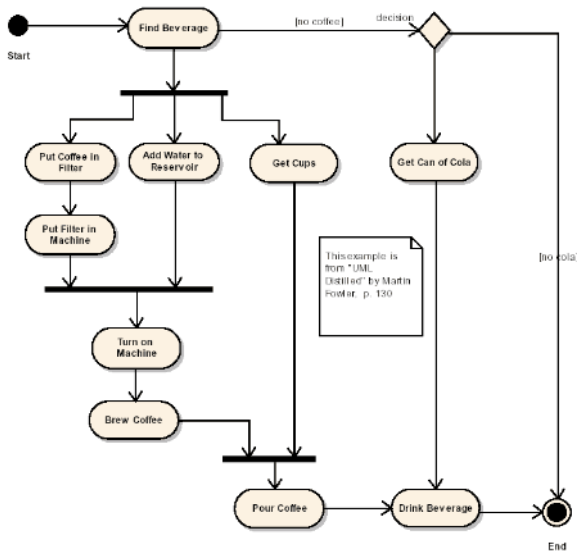
***Tabel 2.4** simbol dalam activity diagram*

Simbol	Nama	Keterangan
	Start State	Menyatakan awal dari aktifitas.
	Stop State	Menyatakan akhir dari aktivitas
	States	Menggambarkan aktivitas yang terjadi
	State Condition	Kondisi proses yang sedang berlaku
	State Transitions	Menggambarkan perubahan dari States satu ke States lainnya
	Fork Node/Percabangan	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
	Join Node/penggabungan	Dari beberapa aliran pada tahap tertentu yang berubah menjadi satu aliran

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal*Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab

untuk aktivitas tertentu.Contoh *activity diagram* tanpa *swimlane*dapat dilihat pada gambar 2.6:



Gambar 2.15 *Contoh activity diagram*

2.7.8.6 Sequence Diagram

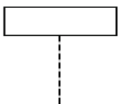
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequencediagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya.

Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

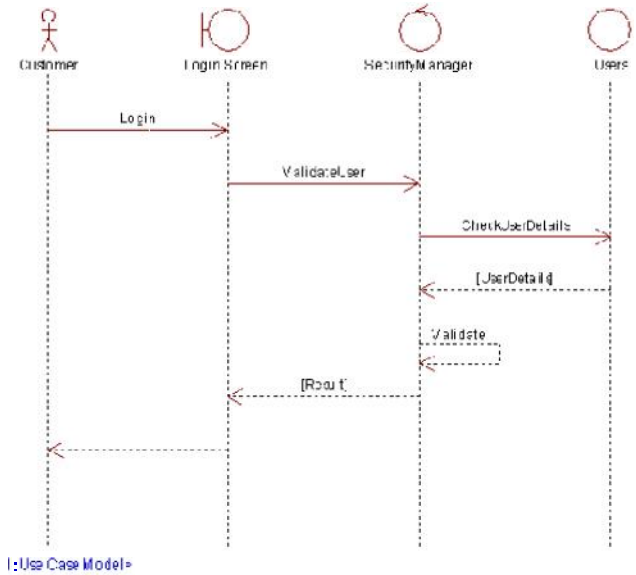
Berikut adalah beberapa simbol dalam *sequence diagram* yang digunakan dalam laporan tugas akhir ini pada tabel 2.5:

Tabel 2.5 *Simbol-simbol Sequnce Diagram*

Simbol	**Keterangan**
LifeLine 	Objek *entity*, antarmuka yang saling berinteraksi.
Message 	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
Message 	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

Contoh *sequence diagram* dapat dilihat pada gambar 2.7 berikut ini:



Gambar 2.16 *Contoh sequence diagram*

2.7.8.7 *Deployment Diagram*

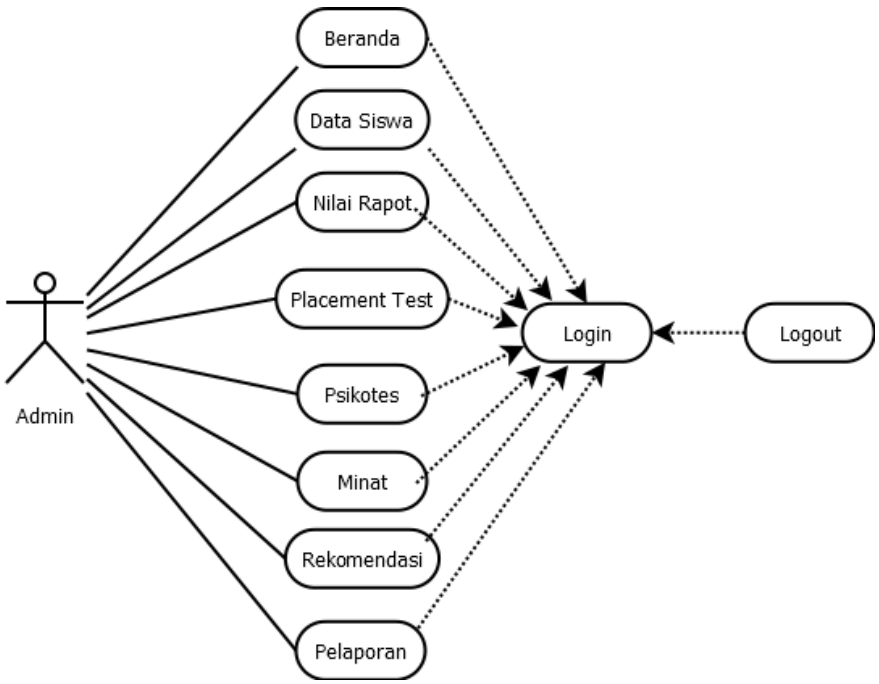
Deployment diagram(diagram pendistribusian) menggambarkan detail bagaimana komponen disebarkan dalam infrastruktur sistem, di mana komponen akan terletak dalam piranti perangkat keras dan bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, serta hal-hal lain yang bersifat fisikal.Sebuah node adalah server, *workstation*, atau piranti perangjat keras lain yang digunakan untuk menyebarkan komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

Deployment diagram digunakan untuk melayani pemodelan hardware yang digunakan dalam implementasi sistem dan asosiasinya antara komponen-

komponen tersebut. Elemen yang digunakan dalam deployment diagram adalah nodes (ditunjukkan sebagai sebuah cube), komponen (ditunjukkan sebagai sebuah kotak bujursangkar) dan juga asosiasi.

2.7.9Pembuatan *Use Case* Diagram

Use Case diagram merupakan gambaran tentang hubungan antara aktor dengan sistem, serta menguraikan kebutuhan fungsional dari sebuah sistem. Gambar 4.14 merupakan gambar perancangan *use case* pada sistem informasi peminatan menggunakan metode Bayes.



Gambar 2.17 Use case sistem informasi peminatan menggunakan metode Bayes

Aplikasi yang telah dibangun dapat dikelola oleh seorang admin dengan fitur-fitur sebagai berikut:

1. *Beranda*. Merupakan halaman awal aplikasi, berisi rangkuman dari keseluruhan sistem, seperti statistik, aktivitas admin, dan rekapitulasi global.
2. *Data siswa*. Merupakan halaman untuk mengelola data siswa. Pada halaman ini, admin dapat melakukan *input, update*, dan *delete*data siswa.
3. *Nilai rapot*. Merupakan halaman untuk mengelola data rapot. Pada halaman ini, admin dapat melakukan *input, update*, dan *delete*data rapot.
4. *Placement Test*. Merupakan halaman untuk mengelola data nilai *placement test*. Pada halaman ini, admin dapat melakukan *input, update,* dan *delete*data *placement test*.
5. *Psikotes*. Merupakan halaman untuk mengelola data nilai psikotes. Pada halaman ini, admin dapat melakukan *input, update*, dan *delete*data *psikotes*.
6. *Minat*. Merupakan halaman untuk menyimpan minat siswa terhadap salah satu jurusan (IPA/IPS).
7. *Rekomendasi*. Merupakan halaman untuk menyimpan dukungan orang tua dan rekomendasi dari guru terhadap salah satu jurusan (IPA/IPS).
8. *Pelaporan*. Berisi laporan hasil pengolahan data, baik data yang nyata (sesuai dengan keadaan sebenarnya) maupun data probabilitas menggunakan metode Bayes. Di sini dapat ditampilkan perbandingan antara teori Bayes dengan data sebenarnya (lihat tabel 4.2).

2.7.10Pembuatan Use Case Skenario

Untuk setiap *use case* harus dibuatkan sebuah skenario di mana skenario tersebut harus menyebutkan prekondisi (kondisi aktor dan sistem sebelum melakukan aksi) dan postkondisi (kondisi aktor dan sistem setelah sistem melakukan aksi). Contoh berikut adalah use case untuk login.

Tabel 2.3 Skenario usecase login

No	**:**	U1
Nama		*Login*
Tujuan		Validasi *login*
Deskripsi		
Aktor		Admin
Skenario Utama		
Kondisi Awal		Admin memiliki *username* dan *password*
Aksi Aktor		**Reaksi Sistem**
1. Memasuki halaman *login* Admin		
2. Mengisi*username* dan *password*		
3. Mengklik tombol *Login*.		4. Validasi data *login* yang diisi oleh Admin 5. Sistem menampilkan halaman admin, jika *login* berhasil 6. Sistem menampilkan peringatan "*Username atau Password Salah*." Jika username atau password salah, dan kembali ke aksi no 2.
		7. Tampilkan halaman beranda Admin,
8. Masuk ke halaman utama Admin		
Kondisi Akhir		Admin *login*