



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1 Helpdesk

Pengertian *helpdesk* yaitu sumber dukungan teknis untuk perangkat keras atau perangkat lunak. *Helpdesk* dikelola oleh orang-orang yang dapat memecahkan masalah langsung atau meneruskan masalah ini ke orang lain. Sistem informasi *helpdesk* merupakan aplikasi yang digunakan untuk melayani *user* apabila mengalami kesulitan dalam menjalankan tugas. Kendala atau masalah yang terjadi biasanya disampaikan melalui *email* ke bagian departemen yang mengatasi hal tersebut, yaitu *helpdesk*.

*Helpdesk* mengatur permintaan *user* melalui software *helpdesk*, seperti *Local Bug Tracker* (LBT), sistem ini melakukan pelacakan terhadap permintaan *user*. Adanya sistem informasi *helpdesk* ini dapat membuat *IT service management* lebih baik, juga supaya tidak ada masalah yang tidak tercatat, hilang, atau terlupakan. Kebutuhan sistem informasi *helpdesk* ini sangat diperlukan sehingga dapat melayani *user* secara tepat.

#### 2.2 Sistem

Menurut Kendall dan Kendall (2008), sistem merupakan serangkaian subsistem yang saling terkait dan tergantung satu sama lain, bekerja bersama-sama untuk mencapai tujuan dan sasaran yang sudah ditetapkan sebelumnya. Suatu sistem mempunyai karakteristik seperti berikut ini :

### 1. Komponen Sistem (*Component*)

Komponen terdiri dari beberapa subsistem dimana setiap subsistem tersebut memiliki fungsi khusus dan akan mempengaruhi proses sistem secara keseluruhan.

### 2. Batas Sistem (*Boundary*)

Batas merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas sistem menunjukkan ruang lingkup dari sistem tersebut.

### 3. Penghubung Sistem (*Interface*)

Penghubung merupakan media penghubung antar subsistem. Penghubung ini memungkinkan sumber daya mengalir dari satu subsistem ke subsistem lainnya.

### 4. Lingkungan Sistem (*Environment*)

Lingkungan merupakan segala sesuatu di luar batas dari sistem yang mempengaruhi operasi sistem.

### 5. Masukan Sistem (*Input*)

Masukan merupakan energi yang dimasukkan ke dalam sistem, dapat berupa masukan perawatan dan masukan sinyal. Masukan perawatan yaitu energi yang dimasukkan agar sistem dapat beroperasi sedangkan masukan sinyal yaitu energi yang diproses untuk mendapatkan keluaran.

#### 6. Pengolahan Sistem (Process)

Suatu sistem dapat memiliki suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

#### 7. Keluaran Sistem (Output)

Keluaran merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain.

#### 8. Tujuan Sistem (Goal) atau Sasaran Sistem (Object)

Suatu sistem mempunyai tujuan (goal) atau sasaran (objective).

Suatu sistem dikatakan berhasil apabila mencapai tujuannya.

### 2.3 Decision Support System (DSS)

*Decision Support System* atau sistem pendukung pengambilan keputusan merupakan sistem berbasis komputer interaktif yang membantu pengambil keputusan dalam menggunakan data dan model untuk menyelesaikan masalah yang tidak terstruktur. DSS menggabungkan sumber daya intelektual seorang individu dengan kemampuan komputer yang bertujuan untuk meningkatkan kualitas pengambilan keputusan. Sebagai tambahan bagi para pengambil keputusan serta memperluas kapabilitas, sistem pendukung keputusan tidak untuk menggantikan pertimbangan manajemen dalam pengambilan keputusannya.

*Decision Support Sistem* memiliki tujuan sebagai berikut :

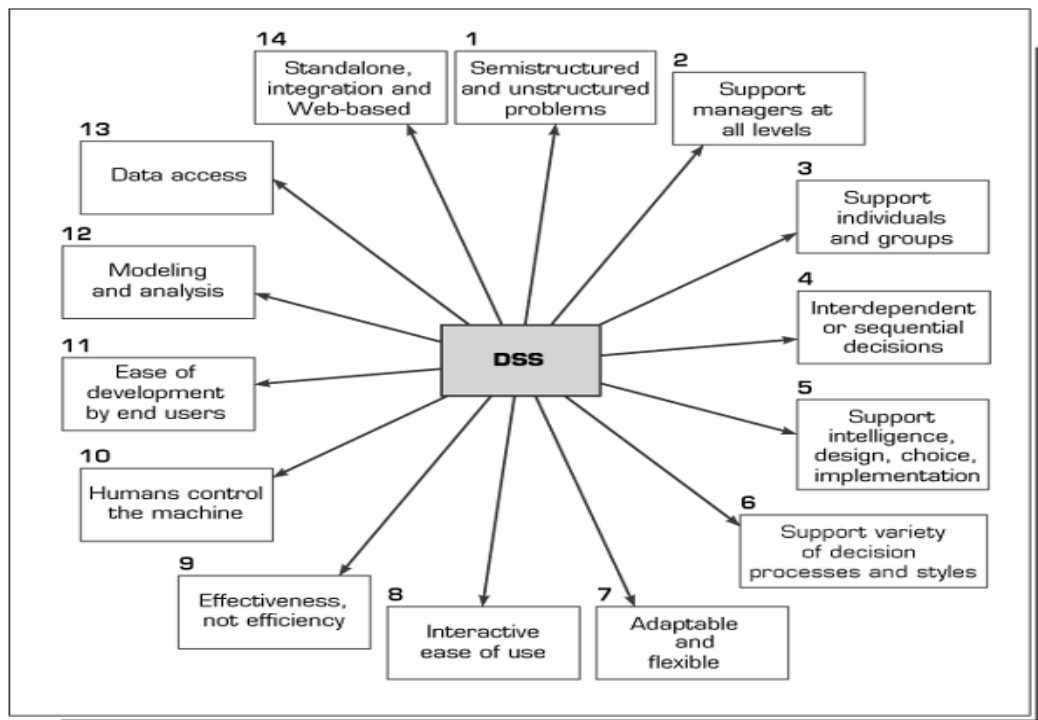
- a. membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur,
- b. memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer,
- c. meningkatkan efektivitas keputusan yang diambil manajer lebih daripada perbaikan efisiensinya,
- d. kecepatan komputasi. Pengambil keputusan dapat melakukan banyak komputasi secara cepat dengan biaya yang rendah dengan menggunakan komputer,
- e. peningkatan produktivitas,
- f. dukungan kualitas,
- g. berdaya saing,
- h. mengatasi keterbatasan kognitif dalam memproses dan penyimpanan.

Turban (2005) mengemukakan kunci karakteristik dan kapabilitas dari DSS sebagai berikut (Gambar 2.1)

- a. dukungan untuk pengambilan keputusan, terutama pada situasi semiterstruktur dan tak terstruktur,
- b. dukungan untuk semua level manajerial, mulai dari pimpinan puncak sampai manajer lapangan,
- c. dukungan untuk individu dan kelompok,
- d. dukungan untuk semua keputusan saling berkaitan dan atau sekuensial,

- e. dukungan di semua fase proses pengambilan keputusan: intelegensi, desain, pilihan, dan implementasi,
- f. dukungan pada berbagai proses dan gaya pengambilan keputusan,
- g. kemampuan sistem beradaptasi dengan cepat dimana pengambil keputusan dapat menghadapi masalah-masalah baru dan pada saat yang sama dapat menanganinya dengan cara mengadaptasikan sistem terhadap kondisi-kondisi perubahan yang terjadi,
- h. pengguna merasa seperti di rumah. *User-friendly*, kapabilitas grafis yang kuat, dan sebuah bahasa interaktif yang alami,
- i. peningkatan terhadap keefektifan pengambilan keputusan (akurasi, timelines, kualitas) dari pada efisiensi (biaya),
- j. pengambil keputusan mengontrol penuh semua langkah proses pengambilan keputusan dalam memecahkan masalah,
- k. pengguna akhir dapat mengembangkan dan memodifikasi sistem sederhana,
- l. menggunakan model-model dalam penganalisisan situasi pengambilan keputusan,
- m. disediakannya akses untuk berbagi sumber data, format, dan tipe, mulai dari sistem informasi geografi (GIS) sampai berorientasi objek,
- n. dapat dilakukan sebagai alat standalone yang digunakan oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan

di satu organisasi keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.



Gambar 2.1 Karakteristik dan kapabilitas DSS (sumber : Turban et al. (2005))

## 2.4 Analytic Hierarchy Process (AHP)

*Analytic Hierarchy Process* merupakan pendekatan pengambilan keputusan multi kriteria yang diperkenalkan oleh Thomas L. Saaty. AHP telah menarik minat banyak peneliti karena metode penghitungan matematika yang baik dan masukan data yang dibutuhkan mudah diperoleh. AHP merupakan alat bantu pendukung pengambilan keputusan yang dapat digunakan untuk menyelesaikan masalah yang kompleks dengan menggunakan struktur hirarki yang terdiri dari tujuan, kriteria, subkriteria, dan alternatif.

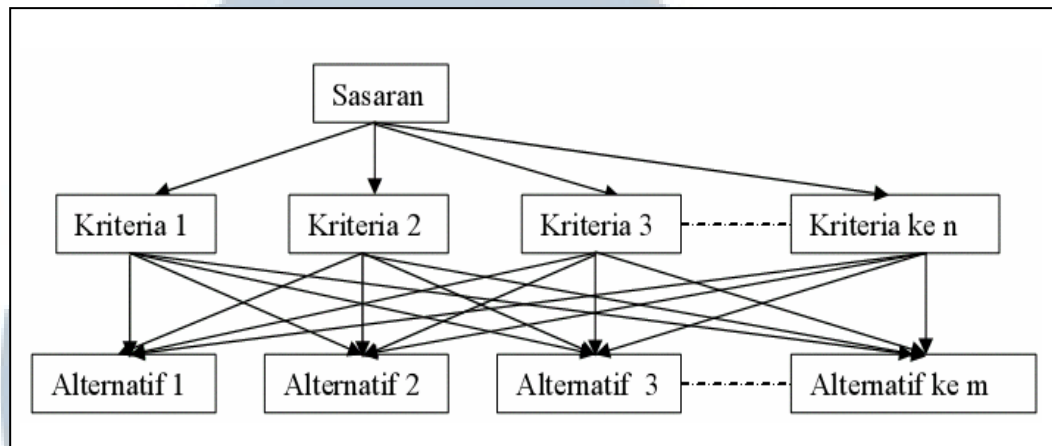
Menurut Badiru (1995), AHP merupakan suatu pendekatan praktis untuk memecahkan masalah keputusan kompleks yang meliputi perbandingan alternatif. Pengambilan keputusan menyajikan hubungan hirarki antara faktor, atribut, karakteristik atau alternatif.

Menurut Bourgeois (2005), AHP umumnya digunakan dengan tujuan untuk menyusun prioritas dari berbagai alternatif atau pilihan yang ada dan pilihan-pilihan tersebut bersifat kompleks. Dengan menggunakan AHP, prioritas yang dihasilkan akan bersifat konsisten dengan teori, logis, transparan, dan partisipatif. AHP akan sangat cocok digunakan untuk penyusunan prioritas kebijakan publik yang menuntut transparansi dan partisipasi.

Metode AHP mula-mula dikembangkan di Amerika pada tahun 1970 dalam hal perencanaan kekuatan militer untuk menghadapi berbagai kemungkinan (*contingency planning*). Kemudian dikembangkan di Afrika khususnya di Sudan dalam hal perencanaan transportasi. Peralatan utama dari model ini adalah sebuah hirarki fungsional dengan input utamanya persepsi manusia. Suatu masalah yang kompleks dan tidak terstruktur dipecah kedalam kelompok-kelompoknya dan kelompok-kelompok tersebut menjadi suatu bentuk hirarki.

Proses membentuk skor secara numerik untuk menyusun ranking setiap alternatif keputusan berbasis pada bagaimana sebaiknya alternatif itu dicocokkan dengan kriteria pembuat keputusan adalah dasar berpikirnya metode AHP. Adapun struktur hirarki AHP ditampilkan pada gambar 2.2 berikut





Gambar 2.2 Struktur Hirarki AHP (Sumber: Saaty, 1994)

Prinsip AHP yaitu

1. *Decomposition*

Setelah permasalahan didefinisikan, maka perlu dilakukan decomposition yaitu memecah permasalahan yang utuh menjadi unsur-unsurnya. Jika ingin mendapatkan hasil yang akurat, pemecahan juga dilakukan terhadap unsur-unsurnya sampai tidak mungkin dilakukan pemecahan lebih lanjut sehingga didapatkan beberapa tingkatan dari persoalan tadi.

2. *Comparative Judgement*

Tahap ini adalah membuat penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu dalam kaitannya dengan tingkat di atasnya. Penilaian ini merupakan sangat penting karena akan berpengaruh terhadap prioritas elemen. Hasil dari penilaian ini disajikan dalam bentuk matriks *pairwise comparison*.

### 3. *Synthesis of Priority*

Setelah membuat matriks *pairwise comparison* lalu dicari *eigen vector* untuk mendapatkan *local priority*. Untuk mendapatkan *global priority* harus dilakukan sintesa di antara *local priority* karena matriks *pairwise comparison* terdapat pada setiap tingkat. Prosedur melakukan sintesa berbeda menurut bentuk hirarki.

### 4. *Logical Consistency*

*Logical consistency* menyatakan ukuran tentang konsisten tidaknya suatu penilaian atau pembobotan perbandingan berpasangan. Pengujian ini diperlukan karena pada keadaan yang sebenarnya akan terjadi beberapa penyimpangan dari hubungan tersebut sehingga matriks tersebut tidak konsisten sempurna.

Matriks bobot yang diperoleh dari hasil perbandingan secara berpasangan tersebut harus mempunyai hubungan kardinal dan ordinal. Hubungan tersebut dapat ditunjukkan sebagai berikut (Suryadi & Ramdhani, 1998):

Hubungan kardinal :  $a_{ij} \cdot a_{jk} = a_{ik}$

Hubungan ordinal :  $A_i > A_j, A_j > A_k$  maka  $A_i > A_k$

Hubungan diatas dapat dilihat dari dua hal sebagai berikut :

Dengan melihat preferensi multiplikatif, misalnya bila anggur lebih enak empat kali dari mangga dan mangga lebih enak dua kali dari pisang maka anggur lebih enak delapan kali dari pisang.

Dengan melihat preferensi transitif, misalnya anggur lebih enak dari mangga dan mangga lebih enak dari pisang maka anggur lebih enak dari pisang.

Pada keadaan sebenarnya akan terjadi beberapa penyimpangan dari hubungan tersebut, sehingga matriks tersebut tidak konsisten sempurna. Hal ini terjadi karena ketidakkonsistenan dalam preferensi seseorang.

Thomas L. Saaty memakai metode matriks perbandingan dalam menentukan bobot kriteria dalam membuat keputusan yang terbaik, adapun bentuk matriksnya adalah sebagai berikut.

Tabel 2.1 Matriks Comparison

|       |          |          |     |          |
|-------|----------|----------|-----|----------|
| C     | $A_1$    | $A_2$    | ... | $A_n$    |
| $A_1$ | $a_{11}$ | $a_{12}$ | ... | $a_{1n}$ |
| $A_2$ | $a_{21}$ | $a_{22}$ | ... | $a_{2n}$ |
| ...   | ...      | ...      | ... | ...      |
| $A_n$ | $a_{n1}$ | $a_{n2}$ | ... | $a_{nn}$ |

Skala perbandingan perbandingan berpasangan dan maknanya yang diperkenalkan oleh Saaty bisa dilihat pada Tabel 2.2 berikut.

Tabel 2.2 Skala Penilaian AHP

| Intensitas dari kepentingan | Definisi  | Penjelasan  |
|-----------------------------|---|---|
| 1                           | Elemen yang satu sama pentingnya dibanding dengan elemen yang lain ( <i>equal importance</i> )              | Kedua elemen menyumbang sama besar pada sifat tersebut      |
| 3                           | Elemen yang satu sedikit lebih penting daripada elemen yang lain ( <i>moderate more importance</i> )        | Pengalaman menyatakan sedikit memihak pada satu elemen      |
| 5                           | Elemen yang satu jelas lebih penting daripada elemen yang lain ( <i>essential, strong more importance</i> ) | Pengalaman menunjukkan secara kuat memihak pada satu elemen |

| Intensitas dari kepentingan | Definisi   | Penjelasan  |
|-----------------------------|--|---|
| 7                           | Elemen yang satu sangat jelas lebih penting daripada elemen yang lain ( <i>demonstrated importance</i> )                           | Pengalaman menunjukkan secara kuat disukai dan didominasi oleh sebuah elemen tampak dalam praktek |
| 9                           | Elemen yang satu mutlak lebih penting daripada elemen yang lain ( <i>absolutely more importance</i> )                              | Pengalaman menunjukkan satu elemen sangat jelas lebih penting                                     |
| 2,4,6,8                     | Nilai tengah di antara dua nilai keputusan yang berdekatan ( <i>grey area</i> )  | Nilai ini diberikan bila diperlukan kompromi  |
| Berbalikan                  | Jika aktivitas i mempunyai nilai yang lebih tinggi dari aktivitas j maka j mempunyai nilai berbalikan ketika dibandingkan dengan i |   |
| rasio                       | Rasio yang didapat langsung dari pengukuran  |   |

Langkah-langkah pada metode AHP menurut Kardi Teknomo (2006)

adalah

1. Tentukan permasalahan yang ingin dicari solusinya
2. Tentukan jenis kriteria, subkriteria (bila ada), dan alternatif pilihan solusi
3. Susun dalam bentuk hirarki
4. Susun kriteria dalam bentuk matriks berpasangan dan hitung perbandingan bobot relatifnya sesuai dengan skala perbandingan Saaty
5. Dapatkan ranking prioritas kriteria melalui perhitungan eigenvektor
  - a. Konversikan nilai setiap elemen dalam matriks dalam bentuk decimal

- b. Jumlahkan setiap kolom dari matriks reproksikal
  - c. Lakukan pembagian pada setiap elemen matriks setiap kolom dengan jumlah per kolom
  - d. Cari nilai rata-ratanya per baris, akan didapat matriks kolom yang disebut dengan *priority vector* atau *normalized principal eigen-vector*, dimana setiap elemen barisnya mewakili kriteria dan nilai setiap elemen menunjukkan prioritas kriteria.
6. Lambda Maximum (*principal eigen-value*) didapat dengan cara menjumlahkan perkalian antara setiap elemen eigen vector dan jumlah kolom dari matriks reproksikal
  7. Lakukan langkah 4, 5, dan 6 untuk matriks subkriteria. Misalkan terdapat  $n$  subkriteria, maka jumlah matriks subkriteria adalah sebanyak  $n$  buah matriks.
  8. Lakukan langkah 4,5,6 untuk matriks alternatif pada setiap level subkriteria
  9. Hasil untuk bobot masing-masing subkriteria dikalikan dengan bobot induknya untuk mendapatkan bobot yang sebenarnya
  10. Menghitung *Consistency Index* ( *CI* )

$$CI = \frac{\lambda_{\max} - n}{n - 1} \dots \dots \dots (2.1)$$

11. Menghitung *Consistency Ratio* ( *CR* )

$$CR = \frac{CI}{RC} \dots \dots \dots (2.2)$$

12. Pada tahap ini, nilai alternatif dikalikan nilai bobot per subkriteria, dimana subkriteria telah memiliki bobot yang sebenarnya (telah melalui proses langkah kesembilan). Dari perhitungan tersebut didapat perbandingan nilai prioritas secara menyeluruh antara alternatif satu dengan yang lainnya. Selanjutnya, hasil akhir adalah bergantung pada keputusan yang diambil dari pihak pengambil keputusan.

Berdasarkan perhitungan Saaty dengan menggunakan 500 sampel dan matriks random dengan skala penilaian 1 sampai dengan 9, diperoleh nilai rata-rata *random index* (*RI*) atau *random consistency* (*RC*) untuk setiap ukuran matriks (*n*) tertentu seperti ditunjukkan pada tabel nilai *random index*. Tabel tersebut menjadi acuan nilai *RI* dalam tabel 2.3.

Tabel 2.3 RC (Sumber: Saaty, 1994)

| Ordo Matrik | RI   | Ordo Matrik | RI   | Ordo Matrik | RI   |
|-------------|------|-------------|------|-------------|------|
| 1           | 0    | 6           | 1,24 | 11          | 1,51 |
| 2           | 0    | 7           | 1,32 | 12          | 1,48 |
| 3           | 0,58 | 8           | 1,41 | 13          | 1,56 |
| 4           | 0,9  | 9           | 1,45 | 14          | 1,57 |
| 5           | 1,12 | 10          | 1,49 | 15          | 1,59 |

Apabila hasil  $CR < 0,1$  atau  $CR < 10\%$  maka nilai bobot perbandingan berpasangan pada matriks kriteria dapat dikatakan konsisten dan dapat diterima. Sedangkan apabila hasil  $CR \geq 0,1$  atau  $CR \geq 10\%$  maka nilai dari perbandingan berpasangan antar-kriteria dari matriks kriteria tersebut tidak konsisten. Untuk itu, pengambilan keputusan harus meninjau ulang

permasalahan dan merevisi pengisian nilai bobot matriks berpasangan pada kriteria, subkriteria, maupun alternatif.

## 2.5 Data Flow Diagram (DFD)

*Data Flow Diagram* merupakan perangkat-perangkat analisis dan perancangan yang terstruktur sehingga memungkinkan penganalisis sistem memahami sistem dan subsistem secara visual sebagai suatu rangkaian aliran data yang saling berkaitan.

Diagram konteks adalah tingkatan tertinggi dalam DFD yang hanya memuat satu proses dan menunjukkan sistem secara keseluruhan. Diagram konteks diberi nomor 0 (nol) dan tidak memuat penyimpanan data. Diagram 1 adalah pengembangan diagram konteks dan dapat mencakup sampai 9 proses. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut kiri atas diagram dan mengarah ke sudut sebelah kanan bawah. Penyimpanan data utama dari sistem dan semua entitas eksternal dimasukkan ke dalam diagram 1.

## 2.6 Entity Relationship Diagram (ERD)

Penyusunan basis data (*database*) selalu didahului dengan pekerjaan pemodelan data. ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD merupakan suatu alat utama pemodelan data dan membantu menggambarkan data ke dalam entitas dan hubungan antar entitas. Model data E-R (*Entity Relationship*) didasarkan

pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan relasi.

Elemen ERD terdiri dari entitas, relationship, dan atribut. Entitas adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak di mana data tersimpan atau di mana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat kelas, yaitu peran, kejadian, lokasi, sesuatu yang tidak nyata/ konsep. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Hubungan antara entitas akan menyangkut dua komponen yang menyatakan jalinan ikatan yang terjadi, yaitu derajat dan partisipasi hubungan. Atribut merupakan deskripsi kelompok data yang mempunyai karakteristik yang sama (data yang mendeskripsikan entity dan relationship), merupakan field yang akan disimpan.

## 2.7 Email

*Email* merupakan sarana mengirim surat melalui jaringan komputer seperti internet. Sejarah *email* dimulai pada tahun 1971 oleh Ray Tomlinson yang bekerja pada sistem operasi TENEX untuk BBN teknologi sebagai kontraktor ARPANET. Tomlinson membuat aplikasi *email* pertama ketika ia menyempurnakan program yang disebut CPYNET untuk SNDMSG. Pada tahun 1972, Roy Tomlinson berhasil menyempurnakan program *email* yang telah diciptakannya. Pada tahun yang sama tersebut, kebutuhan sebuah protokol dibutuhkan, aplikasi FTP (File Transfer Protocol) digunakan untuk mengirim pesan *email*. Kelemahan dari FTP yaitu membuat *email* yang



terpisah untuk setiap penerima sehingga butuh pemakaian memori yang besar sehingga pada tahun 1980-an diciptakanlah SMTP.

Ada dua *email server* yang digunakan saat menggunakan *email* yakni *incoming* dan *outgoing server*. SMTP (*Simple Mail Transfer Protocol*) pada port 25 merupakan *server* yang digunakan untuk mengirim pesan (*outgoing*) sedangkan POP3 (*Post Office Protocol v3*) merupakan *server* yang digunakan untuk menerima *email (incoming)* pada port 110.

Saat mengirim *email*, maka *email* tersebut akan ditangani oleh SMTP *server* dan akan dikirim ke SMTP *server* tujuan, baik langsung ataupun melalui beberapa SMTP *server* di jalurnya. *Email* akan dikirim apabila *server* tujuan terkoneksi. Namun, apabila tidak terkoneksi maka akan dimasukkan ke dalam queue dan diresend kembali dalam beberapa jangka waktu. Jika telah melewati batas waktu, maka akan diberikan pemberitahuan bahwa pesan tidak dapat dikirimkan kepada pengirim.

Saat menerima *email* yang telah dikirimkan oleh pengirim maka akan masuk ke POP3 *server*. Apabila hendak membaca *email* maka *email* pada *server* diunduh sehingga *email* hanya akan ada pada mesin yang mengunduh *email* tersebut.

## 2.8 Hypertext Markup Language (HTML)

HTML merupakan bahasa yang dipahami oleh browser guna menampilkan tampilan halaman web yang bisa dilihat di browser. HTML merupakan singkatan dari *Hypertext Markup Language*, artinya bahasa ini

adalah bahasa markup untuk memformat konten halaman web. Dokumen HTML punya beberapa elemen yang dikelilingi oleh *tag* teks yang dimulai dengan `<` dan berakhir dengan sebuah `>`.

## 2.9 Hypertext Preprocessor (PHP)

PHP merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script* language artinya sintaks-sintaks dan perintah yang diberikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa.

Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*. Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini, *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*. Hampir semua jenis webserver didukung oleh PHP.

Ketika menggunakan PHP sebagai *server-side embedded script language* maka *server* akan melakukan hal-hal sebagai berikut

- Membaca permintaan dari *client/browser*
- Mencari halaman/*page* di *server*
- Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/*page*

- Mengirim kembali halaman tersebut kepada *client* melalui internet atau intranet

## 2.10 My Structured Query Language (MYSQL)

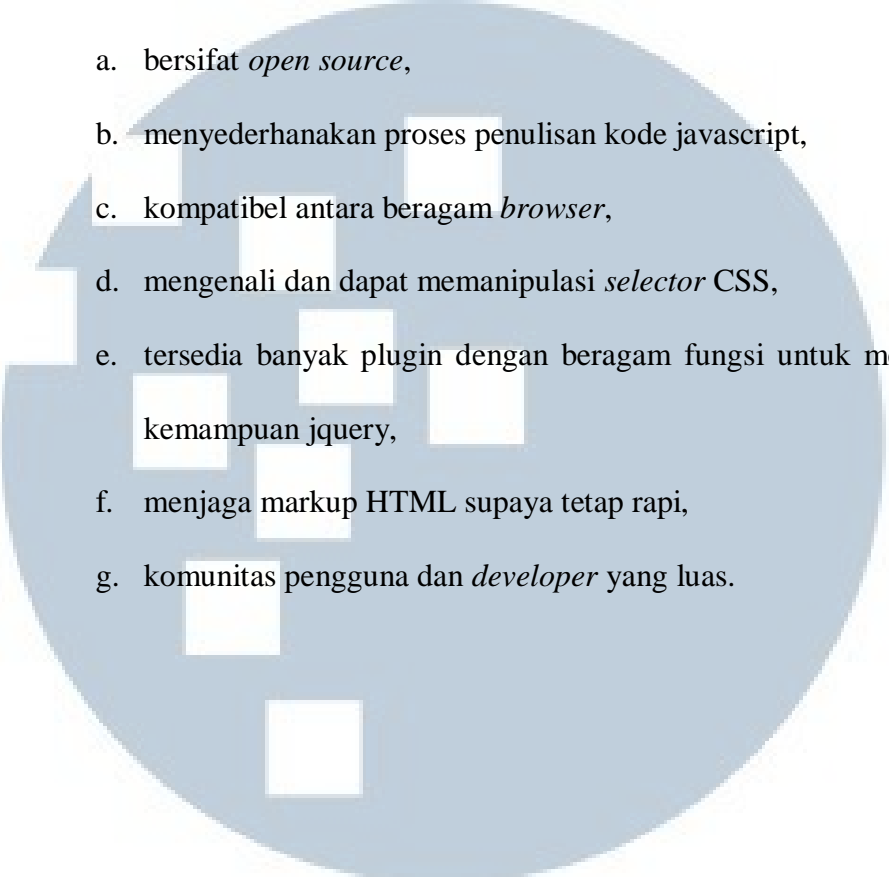
MySQL adalah sebuah *Relational Database Management System* (RDMS). MySQL menggunakan bahasa SQL untuk mengakses database. MySQL tersedia untuk beberapa platform, di antaranya adalah untuk versi windows dan versi linux. MySQL sangat populer untuk aplikasi *web* dan bertindak sebagai komponen basis data dari berbagai macam platform, seperti Linux, BSD, Mac, Windows-Apache-MySQL-PHP, Perl, Python.

Saat ini sudah tersedia MySQLi yang merupakan singkatan dari MySQL Improved yang direkomendasikan oleh pengembang PHP ketika berhubungan dengan MySQL *server* versi 4.1.3 atau lebih baru karena memiliki beberapa keuntungan. MySQLi mempunyai enam kelebihan seperti *interface* yang *object-oriented*, mendukung *prepared statement*, mendukung *multiple statement*, mendukung *transaction*, *debugging* yang lebih baik, dan mendukung *embedded server*.

## 2.11 JQuery

Jquery merupakan kumpulan library javascript untuk mempermudah proses pengolahan dokumen HTML, seperti mengubah konten, penanganan *event*, membuat animasi, menyiapkan *user interface*, dan lain sebagainya.

Kelebihan jquery yaitu

- 
- a. bersifat *open source*,
  - b. menyederhanakan proses penulisan kode javascript,
  - c. kompatibel antara beragam *browser*,
  - d. mengenali dan dapat memanipulasi *selector* CSS,
  - e. tersedia banyak plugin dengan beragam fungsi untuk memperluas kemampuan jquery,
  - f. menjaga markup HTML supaya tetap rapi,
  - g. komunitas pengguna dan *developer* yang luas.

# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA