

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Visualisasi**

Pengertian visualisasi menurut McCormick et al., (1987), yaitu:

- a. Metode penggunaan komputer untuk mentransformasi simbol menjadi geometrik.
- b. Memungkinkan peneliti mengamati simulasi dan komputasi
- c. Memberikan cara untuk melihat yang tidak terlihat
- d. Memperkaya proses penemuan ilmiah dan mengembangkan pemahaman yang lebih dalam dan tak diduga.
- e. Dalam berbagai bidang telah merevolusikan cara ilmuwan meneliti sains.

Pepatah mengatakan “Sebuah gambar bernilai seribu kata”. Untuk beberapa tugas, presentasi visual-seperti peta atau foto-secara dramatis lebih mudah digunakan atau dipahami daripada deskripsi tekstual atau laporan yang diucapkan.

Jadi, visualisasi informasi adalah suatu metode penggunaan komputer untuk menemukan metode terbaik dalam menampilkan data untuk mengingat informasi dengan cara penerimaan alami manusia serta memberikan cara untuk melihat data yang sulit dilihat dengan pemikiran sehingga peneliti bisa mengamati simulasi dan komputasi, juga memperkaya proses penemuan ilmiah dan mengembangkan pemahaman yang lebih dalam dan tak diduga, salah satu contohnya adalah dengan menampilkan data/informasi dalam bentuk gambar. Contoh struktur tree dan grafik.

## 2.2. Media Penyajian Data

Media penyajian data bergantung pada jenis informasi dan pesan yang ingin disampaikan, serta kebutuhan dan selera pengguna. Perancang dapat memilih bentuk teks, grafik, atau mengkombinasikan keduanya. Pemilihan media penyajian yang kurang tepat akan mengurangi efektifitas penyampaian informasi, dan menimbulkan persepsi yang salah mengenai informasi yang disajikan. Ada dua prinsip dalam memilih media penyajian informasi menurut Hariyanti, (2008:15) yaitu:

- a. Media dapat memberikan persepsi yang benar tentang informasi yang disajikan pada *dashboard*.
- b. Media dapat menyajikan informasi sesuai dengan tujuan, tanpa, memakan banyak tempat pada layar *dashboard*.

Setiap jenis media memiliki kekuatan dan kelemahan tersendiri dalam menyajikan suatu jenis informasi. Media yang paling banyak digunakan dalam menyajikan informasi pada *dashboard* adalah tabel dan grafik. Tabel dapat menyajikan data dengan lebih detail, menyajikan angka-angka dengan format yang lebih presisi. Tabel dapat menampilkan data kuantitatif maupun kualitatif. Namun demikian, informasi yang disajikan dalam tabel tidak dapat dipahami secara cepat dan sekilas.

### 2.2.1 Grafik

Ada beberapa tipe grafik atau diagram yang dapat digunakan untuk menampilkan gambaran informasi supaya lebih jelas, antara lain:

1. Diagram garis

Diagram garis digunakan untuk menunjukkan perubahan nilai dari sederetan data relatif terhadap waktu, karena diagram garis biasanya digunakan untuk menunjukkan suatu kecenderungan atau tren.

2. Diagram Batang

Diagram batang digunakan untuk menyajikan nilai relatif terhadap data yang lain. Misal, eksekutif ingin melihat grafik pendaftar per-tahun dan per-gelombang.

Diagram Roti (*Pie*)

Diagram pie biasanya digunakan untuk menggambarkan besarnya presentase data. Misalkan menggambarkan besarnya presentase alasan mahasiswa keluar. (Santoso, 1994)

Grafik dapat digunakan untuk menunjukkan keterhubungan antar data, seperti perbandingan nominal, *time-series*, deviasi, korelasi, dan sebagainya. Ada berbagai macam bentuk grafik yang dapat dipilih untuk menggambarkan setiap jenis keterhubungan data, seperti yang terdapat pada tabel 2.1. Namun demikian, grafik kurang bisa menampilkan angka dengan format yang presisi.

Tabel 2.1 Keterhubungan Data dan Jenis Grafik yang sesuai

Keterhubungan Data	Jenis Grafik yang sesuai
Perbandingan nominal	<ol style="list-style-type: none"><li>1. Grafik <i>bar</i> (horisontal atau vertikal)</li><li>2. Grafik titik (jika 0 tidak termasuk dalam skala nilai)</li></ol>
<i>Time-Series</i>	<ol style="list-style-type: none"><li>1. Grafik garis (untuk melihat tren seluruh data)</li><li>2. Grafik <i>bar</i> (untuk melihat perbandingan antar nilai individu)</li><li>3. Grafik titik yang dihubungkan dengan garis (untuk melihat nilai individu sekaligus tren data secara</li></ol>

Tabel 2.1 Keterhubungan Data dan Jenis Grafik yang sesuai (Lanjutan)

Keterhubungan Data	Jenis Grafik yang sesuai
	4. keseluruhan)
Ranking	1. Grafik <i>bar</i> (horisontal atau vertikal) 2. Grafik titik (jika 0 tidak termasuk dalam skala nilai)
Bagian-dari-keseluruhan	1. Grafik <i>bar</i> (horisontal atau vertikal) 2. Grafik <i>stack bar</i> 3. <i>Pie chart</i>
Deviasi	1. Grafik garis 2. Grafik titik yang dihubungkan dengan garis
Distribusi frekuensi	1. Grafik <i>bar</i> vertikal / histogram (untuk menunjukkan nilai individu) 2. Grafik garis / poligon frekuensi (untuk menunjukkan tren data secara keseluruhan)
Korelasi	1. Grafik titik dan garis ( <i>scatter-plot</i> )

(Sumber: Hariyanti, 2008:16)

### 2.2.2 Pendekatan Pembangunan *Dashboard*

Secara umum, ada dua pendekatan yang digunakan dalam pembangunan perangkat lunak, khususnya *dashboard*, yaitu *data centric* dan *user centric*. Kedua pendekatan memiliki perbedaan mendasar. Pendekatan *data-centric* menekankan pada aktifitas pembuatan model dan struktur data, yang digunakan sebagai dasar dalam membangun kode program dan desain antar muka. Sedangkan pendekatan *user-centric* menekankan pada pembangunan antar muka melalui perancangan *prototype*, dengan fokus pada kebutuhan dan selera pengguna (Hariyanti, 2008:19). Beberapa perbedaan antara pendekatan *data-centric* dengan *user-centric*, dapat dilihat pada tabel 2.2.

Tabel 2.2 Perbedaan Pendekatan *Data-Centric* dan *User-Centric*

<i>Data-Centric</i>	<i>User-Centric</i>
Menggunakan pendekatan <i>bottom-up</i>	Menggunakan pendekatan <i>top-down</i>
Fokus pada hal-hal teknis, yaitu data	Fokus pada kebutuhan dan selera pengguna
Dimulai dengan identifikasi data yang digunakan dalam proses bisnis	Dimulai dengan identifikasi informasi mengenai kebutuhan pengguna, seperti informasi apa yang perlu disajikan, kepada siapa informasi diberikan, dan seberapa detail informasi tersebut perlu untuk disampaikan
Menekankan pada aktifitas pembuatan model dan struktur data	Menekankan pada aktifitas pengembangan antar muka pengguna, melalui perancangan <i>prototype</i> .
Desain antar muka dan kontrol navigasi dilakukan dengan mengikuti struktur data yang dibuat	Model data mengikuti desain <i>prototype</i> yang telah dibuat. Pembuatan struktur data dilakukan melalui teknik <i>filtering</i> dan <i>summarizing</i> .
Menghasilkan struktur tabel yang efisien	Menghasilkan struktur tabel yang tidak efisien
Bertujuan menyediakan akses secara cepat ke data tertentu saja, yang diperlukan pengguna.	Bertujuan membuat sistem yang menyajikan data dan informasi, yang mudah dipahami oleh pengguna.

(Sumber : Hariyanti, 2008:20)

*Dashboard* merupakan alat yang mengandalkan antar muka visual dalam menyajikan informasi di dalamnya. Antar muka yang sesuai dengan kebutuhan dan selera pengguna merupakan kunci keberhasilan *dashboard*. Informasi yang disajikan pada *dashboard* harus dapat dipahami secara cepat dan dipersepsi dengan benar oleh penggunanya. Pendekatan *user-centric* yang memfokuskan pada perancangan antar muka sesuai dengan kebutuhan dan selera pengguna, merupakan pendekatan yang paling tepat untuk pembangunan *dashboard* dibandingkan dengan pendekatan *data-centric*.

### 2.3. *Dashboard*

*Dashboard* adalah sebuah tampilan *visual* dari informasi terpenting yang dibutuhkan untuk mencapai satu atau lebih tujuan, digabungkan dan diatur pada sebuah layar, menjadi informasi yang dibutuhkan dan dapat dilihat secara sekilas. Tampilan *visual* disini mengandung pengertian bahwa penyajian informasi harus dirancang sebaik mungkin, sehingga mata manusia dapat menangkap informasi secara cepat dan otak manusia dapat memahami maknanya secara benar. *Dashboard* itu sebuah tampilan pada satu monitor komputer penuh, yang berisi informasi yang bersifat kritis, agar kita dapat melihatnya dengan segera, sehingga dengan melihat *dashboard* itu saja, kita dapat mengetahui hal-hal yang perlu diketahui. Biasanya kombinasi dari teks dan grafik, tetapi lebih ditekankan pada grafik (Few, 2006).

#### 2.3.1 Tujuan Penggunaan *Dashboard*

Tujuan dalam penggunaan *dashboard* menurut Eckerson (Hariyanti, 2008:7), yaitu:

a. Mengkomunikasikan Strategi

Mengkomunikasikan strategi dan tujuan yang dibuat oleh eksekutif, kepada semua pihak yang berkepentingan, sesuai dengan peran dan levelnya dalam organisasi.

b. Memonitor dan menyesuaikan Pelaksanaan Strategi

Memonitor pelaksanaan dari rencana dan strategi yang telah dibuat. Memungkinkan eksekutif untuk mengidentifikasi permasalahan kritis dan membuat strategi untuk mengatasinya.

- c. Menyampaikan Wawasan dan Informasi ke semua pihak  
Menyajikan informasi menggunakan grafik, simbol, bagan dan warna yang memudahkan pengguna dalam memahami dan mempersepsi informasi secara benar.

### 2.3.2 Karakteristik *Dashboard*

Beberapa karakteristik *dashboard* menurut Malik (Hariyanti, 2008), yaitu:

- a. *Synergetic*  
Ergonomis dan memiliki tampilan *visual* yang mudah dipahami oleh pengguna. *Dashboard* mensinergikan informasi dari berbagai aspek yang berbeda dalam satu layar.
- b. *Monitor*  
Menampilkan KPI yang diperlukan dalam pembuatan keputusan dalam domain tertentu, sesuai dengan tujuan pembangunan *dashboard* tersebut.
- c. *Accurate*  
Informasi yang disajikan harus akurat, dengan tujuan untuk mendapatkan kepercayaan dari penggunanya.
- d. *Responsive*  
Merespon *threshold* yang telah didefinisikan, dengan memberikan *alert* (seperti bunyi alarm, *blinker*, *email*) untuk mendapatkan perhatian pengguna terhadap hal-hal yang kritis.
- e. *Timely*  
Menampilkan informasi terkini yang diperlukan untuk pengambilan keputusan.

f. *Interactive*

Pengguna dapat melakukan *drill down* dan mendapatkan informasi lebih detail, analisis sebab akibat dan sebagainya.

g. *More Data History*

Melihat tren sejarah KPI contohnya perbandingan jumlah mahasiswa baru periode saat ini dengan beberapa tahun yang lalu, untuk mengetahui apakah kondisi sekarang lebih baik atau tidak.

h. *Personallized*

Penyajian informasi spesifik untuk setiap jenis pengguna sesuai domain tanggung jawab, hak akses, dan batasan akses data.

i. *Analytical*

Fasilitas untuk melakukan analisis, seperti analisis sebab akibat.

j. *Collaborative*

Fasilitas pertukaran catatan (laporan) antar pengguna mengenai hasil pengamatan *dashboardnya* masing-masing yaitu sarana komunikasi dalam melakukan fungsi manajemen dan kontrol.

k. *Trackability*

Memungkinkan setiap pengguna untuk mengkustomisasi nilai yang akan dilacak.

### 2.3.3 Ciri-ciri *Dashboard* yang Baik

*Dashboard* yang didesain baik, akan menampilkan informasi yang:

- a. Luar biasa terorganisir dengan baik
- b. Meringkas, terutama dalam bentuk ringkasan dan bentuk pengecualian
- c. Spesifik dan disesuaikan untuk user dan tujuan *dashboard*.



- d. Ditampilkan secara ringkas, kadang dalam media kecil yang mengkomunikasikan data dan pesan tersebut dengan jelas dan langsung pada intinya.

#### **2.3.4 Jenis *Dashboard***

Menurut Eckerson dan Few dalam (Hariyanti, 2008) *Dashboard* bisa dikelompokkan sesuai dengan level manajemen yaitu:

##### **a. *Strategic Dashboard***

- 1) Mendukung manajemen level strategis.
- 2) Informasi untuk membuat keputusan bisnis, memprediksi peluang, dan memberikan arahan pencapaian tujuan strategis.
- 3) Fokus pada pengukuran kinerja *high-level* dan pencapaian tujuan strategis organisasi.
- 4) Mengadopsi konsep *Balance Score Card*.
- 5) Informasi yang disajikan tidak terlalu detail.
- 6) Konten informasi tidak terlalu banyak dan disajikan secara ringkas.
- 7) Informasi disajikan dengan mekanisme yang sederhana. melalui tampilan yang “*unidirectional*”.
- 8) Tidak didesain untuk berinteraksi, dalam melakukan analisis yang lebih detail.
- 9) Tidak memerlukan data *realtime*

##### **b. *Tactical Dashboard***

- 1) Mendukung manajemen level taktikal
- 2) Memberikan informasi yang diperlukan oleh analis untuk mengetahui penyebab suatu kejadian.

- 3) Fokus pada proses analisis untuk menemukan penyebab dari suatu kondisi atau kejadian tertentu.
- 4) Dengan fungsi *drill-down* dan navigasi yang baik.
- 5) Memiliki konten informasi yang lebih banyak (analisis perbandingan, pola/tren, evaluasi kinerja)
- 6) Menggunakan media penyajian yang “cerdas”, yang memungkinkan pengguna melakukan analisis terhadap data yang kompleks
- 7) Tidak memerlukan data *realtime*

**c. Operational Dashboard**

- 1) Mendukung manajemen level operasional
- 2) Memberikan informasi mengenai aktifitas yang sedang terjadi, beserta perubahannya secara *realtime* untuk memberikan kewaspadaan terhadap hal-hal yang perlu direspon secara cepat.
- 3) Fokus pada monitoring aktifitas dan kejadian yang berubah secara konstan.
- 4) Informasi disajikan spesifik, tingkat kedetailan yang cukup dalam
- 5) Media penyajian yang sederhana
- 6) *Alert* disajikan dengan cara yang mudah dipahami, dan mampu menarik perhatian pengguna.
- 7) Bersifat dinamis, sehingga memerlukan data *real-time*.
- 8) Didesain untuk berinteraksi dengan data, untuk mendapatkan informasi yang lebih detail, maupun informasi pada level yang lebih atas (*higher-level data*)

### 2.3.5 Kesalahan Umum Pembuatan Dashboard

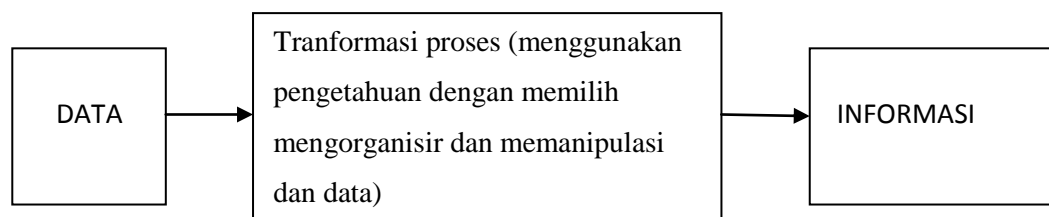
Beberapa hal di bawah ini merupakan 13 kesalahan umum pada pembuatan *dashboard* (Few,2006:48):

- a. Melebihi batas pada satu layar monitoring komputer. Hal ini mengacu pada tampilan *dashboard*.
- b. Menyediakan data yang tidak memadai : misal *dashboard* tentang penerimaan mahasiswa baru, seharusnya *dashboard* yang ada tidak hanya berisi jumlah mahasiswa baru pada tahun itu saja, melainkan berisi informasi jumlah mahasiswa baru tahun lalu.
- c. Menampilkan detil atau presisi yang berlebihan: *dashboard* hampir selalu memerlukan informasi tingkat tinggi untuk mampu mendukung penggunanya untuk peninjauan cepat. Jadi dengan detil yang berlebihan, hanya akan memperlambat penangkapan si pengguna tanpa menambah keuntungan pengguna. Contoh: \$3.8 akan lebih baik dibanding \$3.848.352
- d. Memilih ukuran kurang tepat: misalnya, bila seorang pengguna *dashboard* hanya memerlukan persentase tingkat penjualan, maka sebaiknya hanya disajikan dalam bentuk persentase (-9% akan lebih baik dibanding -\$8.066)
- e. Memilih media tampilan yang tidak tepat: maksudnya adalah salah memilih media (*bar*, *pie*, *circle*, atau radar)
- f. Menyajikan variasi berbeda yang sia-sia: misalnya, menyajikan chart penjualan pada beberapa daerah dengan menggunakan *pie*, radar, dan *bar* pada *dashboard* yang sama.
- g. Menggunakan media tampil yang desainnya payah.

- h. Menampilkan kuantitas data secara tidak akurat: contoh sebuah grafik batang yang dimulai angka \$500.000 bukan \$0.
- i. Mengatur tampilan data dengan payah. *Dashboard* pada dasarnya menampilkan informasi yang banyak dengan tampilan seminimalis mungkin. Jadi, bila data yang ada tidak diatur sedemikian rupa, akan semakin membingungkan penggunanya.
- j. Menyoroti data penting secara tidak efektif atau tidak sama sekali. *Dashboard* yang baik adalah menonjolkan data yang lebih penting dibanding yang lain. Sehingga pengguna langsung melihatnya.
- k. Mengacaukan tampilan dengan dekorasi yang tak perlu. Sebaiknya tampilan *dashboard* tidak terlalu “wah” tampilannya, hal ini akan menyebabkan mata penggunanya mudah lelah di kemudian hari.
- l. Salah atau berlebihan menggunakan warna. Sebaiknya menggunakan warna yang tepat. Dan tidak serampangan dalam menggunakan warna.

## 2.4 Data dan Informasi

Data adalah sebuah kebenaran, atau kenyataan, contoh nama pegawai, order penjualan, nomor penjualan. Informasi adalah sekumpulan kebenaran atau kenyataan yang terorganisir sedemikian rupa yang menyebabkan mereka memiliki nilai tambah daripada kumpulan kebenaran itu sendiri (Stair & George, 2006).



Gambar 2.1 Perubahan Data menjadi Informasi

Untuk menjadi bernilai bagi manajer dan pembuat keputusan menurut Stair & George, (2006) informasi seharusnya memiliki karakteristik seperti di bawah ini :

a. Akurat

Informasi yang akurat adalah informasi yang bebas dari *error*. Dalam beberapa kasus, informasi yang tidak akurat dihasilkan karena data yang digunakan pada pemrosesan tidak akurat.

b. Lengkap

Informasi yang lengkap berisi semua kebenaran (data) yang lengkap. Contoh, informasi mahasiswa keluar tidak akan lengkap tanpa informasi alasan mahasiswa keluar.

c. Ekonomis

Informasi seharusnya ekonomis dalam pembuatannya. Para pembuat keputusan selalu akan membandingkan nilai guna informasi dan biaya yang dikeluarkan untuk membuatnya.

d. Fleksibel

Informasi yang fleksibel dapat digunakan untuk berbagai tujuan.

e. Handal

Informasi yang handal dapat diandalkan. Dalam banyak kasus, kehandalan sebuah informasi bergantung pada metode pengumpulan data tersebut. Dalam contoh lain, kehandalan ini bergantung pada sumber dari informasi tersebut.

f. Relevan (Berhubungan)

Informasi yang relevan penting bagi pembuat keputusan. Istilahnya, informasi bahwa harga kayu turun, tidak relevan bagi pabrik chip computer.

g. Sempel

Informasi seharusnya juga simpel, tidak terlalu rumit. Informasi yang mutakhir dan detil mungkin tidak dibutuhkan. Kenyataannya, informasi yang berlebih dapat menyebabkan *overload* informasi, dimana para pembuat keputusan mempunyai informasi berlebih dan tidak bisa menentukan mana yang penting.

h. Tepat Waktu

Informasi yang tepat waktu adalah informasi yang ada pada saat yang dibutuhkan.

i. Dapat dibuktikan

Informasi seharusnya dapat dibuktikan. Ini berarti anda dapat memeriksa untuk memastikan bahwa informasi tersebut benar, mungkin dengan memeriksa sumber lain untuk informasi yang sama.

j. Dapat Diakses

Informasi seharusnya mudah diakses oleh pengguna untuk mendapatkan bentuk informasi yang tepat dan disaat yang tepat untuk mendapatkan yang mereka butuhkan.

k. Aman

Informasi seharusnya aman dari jamahan pengguna yang tidak berhak mengakses.

## **2.5 Key Performance Indicator**

*Key Performance Indicator* adalah indikator yang merepresentasikan kinerja dari proses yang dilaksanakan (Hariyanti, 2008:11). *Key Performance Indicator* merupakan sekumpulan ukuran mengenai aspek kinerja yang paling kritis, yang menentukan kesuksesan organisasi pada masa sekarang dan masa yang akan datang. *Key Performance Indicator* digunakan memprediksi peluang kesuksesan atau kegagalan dari proses-proses yang dilaksanakan organisasi, sehingga *Key Performance Indicator* dapat digunakan sebagai alat untuk meningkatkan kinerja organisasi secara dramatis. Contoh dari penjelasan di atas adalah penentuan parameter nilai dalam trend penerimaan mahasiswa baru membuat user dapat dengan mudah mengetahui kondisi penerimaan mahasiswa baru apakah sedang bagus atau tidak.

## **2.6 Akreditasi Sekolah**

Akreditasi sekolah/madrasah adalah proses penilaian secara komprehensif terhadap kelayakan satuan atau program pendidikan, yang hasilnya diwujudkan dalam bentuk sertifikat pengakuan dan peringkat kelayakan yang dikeluarkan oleh suatu lembaga yang mandiri dan profesional.

Penggunaan instrumen akreditasi yang komprehensif dikembangkan berdasarkan standar yang mengacu pada SNP. Hal ini didasarkan pada Peraturan Pemerintah Nomor 19 Tahun 2005 yang memuat kriteria minimal tentang komponen pendidikan. Seperti dinyatakan pada pasal 1 ayat (1) bahwa SNP adalah kriteria minimal tentang sistem pendidikan di seluruh wilayah hukum Negara Kesatuan Republik Indonesia. Oleh karena itu, SNP harus dijadikan acuan

guna memetakan secara utuh profil kualitas sekolah/madrasah. Di dalam pasal 2 ayat (1), lingkup SNP meliputi:

- a. Standar isi;
- b. Standar proses;
- c. Standar kompetensi lulusan;
- d. Standar pendidik dan tenaga kependidikan;
- e. Standar sarana dan prasarana;
- f. Standar pengelolaan;
- g. Standar pembiayaan; dan
- h. Standar penilaian pendidikan.

SNP diharapkan menjadi pendorong dan dapat menciptakan suasana kondusif bagi pertumbuhan pendidikan dan memberikan arahan untuk evaluasi diri sekolah/ madrasah yang berkelanjutan, serta menyediakan perangsang untuk terus berusaha mencapai mutu yang diharapkan. (Mendiknas, 2008)

## **2.7 Teknik Penskoran Akreditasi SMA/MA**

### **2.7.1 Bobot Komponen dan Butir Instrumen Akreditasi SMA/MA**

Instrumen Akreditasi SMA/MA disusun berdasarkan delapan komponen yang mengacu pada Standar Nasional Pendidikan. Instrumen Akreditasi ini terdiri dari 165 butir pernyataan tertutup yang terdiri dari lima opsi jawaban. Bobot Komponen dan Bobot Butir Instrumen Akreditasi SMA/MA diperlihatkan pada Tabel 1 berikut ini:



Tabel 2.2. Bobot Komponen dan Bobot Butir Instrumen Akreditasi SMA/MA

No	Komponen Akreditasi	Nomor Butir	Jumlah Butir	Bobot Komponen	Bobot Butir
1	Standar Isi	1 – 15	15	15	1,00
2	Standar Proses	16 – 25	10	10	1,00
3	Standar Kompetensi Lulusan	26 – 50	25	10	0,40
4	Standar Pendidik dan Tendik	51 – 70	20	15	0,75
5	Standar Sarana dan Prasarana	71 – 100	30	15	0,50
6	Standar Pengelolaan	101 – 120	20	10	0,50
7	Standar Pembiayaan	121 – 145	25	15	0,60
8	Standar Penilaian	146 – 165	20	10	0,50

Keterangan: Bobot Butir = Bobot Komponen : Jumlah Butir

### 2.7.2 Penentuan Skor Butir dan Skor Tertimbang Maksimum

Seluruh butir pernyataan Instrumen Akreditasi SMA/MA merupakan pernyataan tertutup yang terdiri dari lima opsi jawaban yaitu A, B, C, D, atau E. Setiap pernyataan yang dijawab A memperoleh skor = 4, B memperoleh skor = 3, C memperoleh skor = 2, D memperoleh skor = 1, dan E memperoleh skor = 0. Jika perolehan skor maksimum setiap butir sama dengan 4, maka Skor Tertimbang Maksimum dapat dihitung dengan rumus:

- a. 
$$\text{Skor Tertimbang Maksimum} = \text{Jumlah Butir} \times \text{Skor Butir Maksimum} \times \text{Bobot Butir}$$

Skor Tertimbang Maksimum pada setiap komponen akreditasi seperti nampak pada Tabel 2 Kolom 5 berikut ini. Dari tabel tersebut terlihat bahwa Jumlah Skor Tertimbang Maksimum sama dengan 400.

Tabel 2.3 Skor Tertimbang Maksimum Akreditasi SMA/MA

No	Komponen Akreditasi	Jumlah Butir	Skor Butir Max	Bobot Butir	Skor Tertimbang Max
1	Standar Isi	15	4	1,00	60
2	Standar Proses	10	4	1,00	40

No	Komponen Akreditasi	Jumlah Butir	Skor Butir Max	Bobot Butir	Skor Tertimbang Max
3	Standar Kompetensi Lulusan	25	4	0,40	40
4	Standar Pendidik dan Tendik	20	4	0,75	60
5	Standar Sarana dan Prasarana	30	4	0,50	60
6	Standar Pengelolaan	20	4	0,50	40
7	Standar Pembiayaan	25	4	0,60	60
8	Standar Penilaian	20	4	0,50	40
Jumlah Skor Tertimbang Maksimum					400

Keterangan:

Skor Tertimbang Maksimum=Jumlah Butir x Skor Butir Maksimum x Bobot

Butir

### 2.7.3 Penentuan Nilai Akhir Akreditasi

Langkah-langkah penentuan Nilai Akhir Akreditasi adalah sebagai berikut.

1. Menghitung Jumlah Skor Butir yang dijawab A, B, C, D, atau E pada komponen 1 (Standar Isi). Sebagai contoh, jawaban butir pernyataan instrumen pada standar isi seperti Tabel 3 berikut.

Tabel 2.4. Jawaban Butir Pernyataan padaKomponen Standar Isi

No. Butir	Jawaban	Skor Butir
1	A	4
2	B	3
3	B	3
4	A	4
5	B	3
6	D	1
7	C	2
8	B	3
9	A	4
10	E	0
11	A	4
12	C	2
13	B	3
14	A	4
15	A	4
Jumlah Skor butir		44

1. Mengulang langkah C.1 di atas untuk komponen 2 (Standar Proses) sampai dengan komponen 8 (Standar Penilaian).
2. Menghitung Skor Tertimbang untuk setiap komponen dengan rumus: Skor Tertimbang setiap Komponen = Jumlah Skor Butir x Bobot Butir. Sebagai contoh: Skor Tertimbang untuk Standar Isi = 44 x 1,00 = 44,00.
3. Menjumlahkan Skor Tertimbang untuk setiap komponen mulai dari komponen 1 sampai komponen 8. Sebagai contoh: Jumlah Skor Tertimbang dari seluruh komponen akreditasi sama dengan 336,15
4. Menentukan Nilai Akhir Akreditasi dalam skala ratusan (0—100) dengan rumus:

$$\begin{aligned} \text{Nilai Akhir Akreditasi} &= \frac{\text{Jumlah Skor Tertimbang}}{\text{Jumlah Skor Tertimbang Maksimum}} \times 100 \\ &= \frac{\text{Jumlah Skor Tertimbang}}{400} \times 100 \end{aligned}$$

Tabel 2.5. Perhitungan Akhir Nilai Akreditasi SMA/MA

No	Komponen Akreditasi	Jumlah Skor Butir	Bobot Butir	Skor Tertimbang Max
1	Standar Isi	44	1,00	44,00
2	Standar Proses	30	1,00	30,00
3	Standar Kompetensi Lulusan	87	0,40	34,80
4	Standar Pendidik dan Tendik	71	0,75	53,25
5	Standar Sarana dan Prasarana	106	0,50	53,00
6	Standar Pengelolaan	67	0,50	33,50
7	Standar Pembiayaan	81	0,60	48,60
8	Standar Penilaian	78	0,50	39,00
9	Jumlah Skor Tertimbang			336,15
10	Nilai Akhir = $\frac{\text{Jumlah Skor Tertimbang}}{\text{Jumlah Skor Tertimbang maksimum}} \times 100$ $= \frac{336.15}{400} \times 100 = 84,04$			84,04

#### **2.7.4 Peringkat Akreditasi**

1. Peringkat akreditasi A (Amat baik), jika memperoleh Nilai Akhir (NA) lebih besar dari 85 sampai dengan 100 ( $85 < NA < 100$ ), dengan ketentuan kriteria status terakreditasi terpenuhi.
2. Peringkat akreditasi B (Baik), jika memperoleh Nilai Akhir lebih besar dari 70 sampai dengan 85 ( $70 < NA < 85$ ), dengan ketentuan kriteria status terakreditasi terpenuhi.
3. Peringkat akreditasi C (Cukup Baik), jika memperoleh Nilai Akhir lebih besar dari atau sama dengan 56 sampai dengan 70 ( $56 < NA < 70$ )

### **2.8 Teori Pendukung**

#### **2.8.1 DFD (*Data Flow Diagram*)**

Diagram alur data (*Data Flow Diagram*) atau DFD, adalah suatu gambaran grafis dari suatu sistem yang menggunakan sejumlah bentuk-bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang berkaitan. Simbol-simbol tersebut menggambarkan unsur lingkungan yang berhubungan dengan sistem, proses, arus data, dan penyimpanan data (McLeod & Schell, 2007). DFD merupakan suatu alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*Structured Analysis and System*).

DFD dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. Dengan kata lain, DFD adalah pembuatan model yang memberikan penekanan pada fungsi sistem. Simbol-simbol yang digunakan di DFD mewakili maksud tertentu, yaitu:

1. *External Entity* (kesatuan Luar) atau *Boundary* (batas sistem).

Setiap sistem pasti memiliki batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainya yang berada di lingkungan luarnya yang memberikan input atau menerima *output* dari sistem.

2. *Data flow*( arus data).

3. Arus data di DFD diberi simbol panah. Arus data ini mengalir diantara proses, simpanan, dan kesatuan luar.

4. *Process* (proses).

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

5. *Data store* (simpanan data).

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa suatu *file* atau *database* di komputer, suatu arsip atau catatan manual dan lain sebagainya. Terdapat 4 empat simbol dasar untuk perwakilan tertentu dalam membuat data flow diagram (DFD), yaitu simbol entitas, aliran data, proses dan data *storage*.

### 2.8.2 ERD (*Entity Relationship Diagram*)

ERD (*Entity Relational Diagram*) digunakan untuk mendokumentasikan data organisasi dengan mengidentifikasi jenis entitas dan hubungannya antara entitas. ERD merupakan peralatan pembuatan data yang paling fleksibel, dapat di

adaptasi untuk berbagai pendekatan dalam pengembangan sistem (McLeod & Schell, 2007).

### 2.8.3 Basis Data

Pangkalan data atau basis data (bahasa Inggris: *database*), atau sering pula dieja basisdata, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Menurut “(Ramakrishnan dan Gehrke, 2003) basis data sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan”. Dari definisi di atas dapat disimpulkan bahwa basis data adalah kumpulan informasi yang disusun berdasarkan cara tertentu dan merupakan suatu kesatuan yang utuh. Dengan sistem tersebut data yang terhimpun dalam suatu *database* dapat menghasilkan informasi yang berguna.

### 2.8.4 Testing dan Implementasi

Masalah testing program muncul secara simultan bersamaan dengan pengalaman pertama dalam menulis program. Di awal debutnya, testing merupakan aktifitas yang tidak hanya bertujuan untuk menemukan *error* tapi juga bertujuan untuk mengoreksi dan menghilangkannya. Sehingga pembahasan masalah testing saat itu lebih banyak ke arah “*debugging*”, serta kesulitan dalam mengoreksi dan menghilangkan *error*.

Namun sudut pandang ini telah bergeser di tahun 1957, dimana testing program telah dibedakan secara jelas dengan debugging. Sejak konferensi pertama tentang testing *software*, yang diadakan pada bulan Juni 1972 di University of North Carolina, mulai banyak konferensi dan workshop yang bertemakan tentang kualitas, reliabilitas dan rekayasa *software*, dimana secara bertahap telah memasukan disiplin testing sebagai elemen yang terorganisasi dalam teknologi *software*. Selama beberapa tahun terakhir, telah banyak buku-buku testing diterbitkan, bahkan telah banyak pula literatur manajemen pemrograman dan proyek yang memasukan masalah testing dalam beberapa bab di dalamnya.

Testing secara terus-menerus berkembang dalam memenuhi kebutuhan di tiap sektor industri untuk keberadaan metode-metode praktis dalam memastikan kualitas. Walaupun demikian disiplin testing masih jauh dari kematangan, bahkan perjanjian akan definisi dari testing itu sendiri masih belum dapat memuaskan semua pihak.

Menurut Romeo (2003,3) Testing *software* adalah proses mengoperasikan *software* dalam suatu kondisi yang di kendalikan untuk

1. verifikasi apakah telah berlaku sebagaimana telah ditetapkan (menurut spesifikasi),.
2. mendeteksi *error*

Untuk menentukan apakah sesuatu hal terjadi bilamana tidak tidak seharusnya terjadi.

3. validasi

apakah spesifikasi yang telah ditetapkan sudah memenuhi keinginan atau kebutuhan dari pengguna yang sebenarnya.

Testing seharusnya berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut tidak terjadi dimana seharusnya mereka ada. Dari beberapa definisi di atas, dapat kita lihat akan adanya banyak perbedaan pandangan dari praktisi terhadap definisi testing. Namun secara garis besar didapatkan bahwa testing harus dilihat sebagai suatu aktifitas yang menyeluruh dan terus-menerus sepanjang proses pengembangan. Testing merupakan aktifitas pengumpulan informasi yang dibutuhkan untuk melakukan evaluasi efektifitas kerja. Jadi tiap aktifitas yang digunakan dengan obyektifitas untuk menolong kita dalam mengevaluasi atau mengukur suatu atribut *software* dapat disebut sebagai suatu aktifitas testing. Termasuk di dalamnya *review*, *walk-through*, inspeksi, dan penilaian serta analisa yang ada selama proses pengembangan. Dimana tujuan akhirnya adalah untuk mendapatkan informasi yang dapat diulang secara konsisten (*reliable*) tentang hal yang mungkin sekitar *software* dengan cara termudah dan paling efektif, antara lain:

- a. Apakah software telah siap digunakan?
- b. Apa saja risikonya?
- c. Apa saja kemampuannya?
- d. Apa saja keterbatasannya?
- e. Apa saja masalahnya?
- f. Apakah telah berlaku seperti yang diharapkan?



### **2.8.5 FusionChart Free**

*FusionCharts* adalah komponen pemetaan dalam basis *flash* yang dapat digunakan untuk merender data dalam bentuk animasi grafik. Dengan basis *Adobe Flash*, *FusionCharts* dapat digunakan dengan berbagai macam bahasa *scripting web* seperti *HTML*, *.NET*, *ASP*, *JSP*, *PHP*, *ColdFusion* dan lain sebagainya, untuk menghasilkan chart yang interaktif dan powerfull. Menggunakan *XML* sebagai data *interface-nya*, *FusionCharts* mampu menciptakan charts yang *compact*, interaktif, dan *visually-arresting* (Arifianto, 2012).

### **2.8.6 UIBox**

*UIBox* adalah sebuah komponen *library* antar muka berbasis *HTML*, *CSS* dan *JavaScript*. Komponen *uiBox* merupakan wadah tata letak yang secara otomatis mengatur objek yang horizontal atau vertical. Dengan menggunakan *layout* otomatis, UI objek dapat direposisi dan diubah ukurannya secara otomatis dalam menanggapi perubahan dalam gaya teks dan isi. Ini memberikan fleksibilitas yang besar ketika mengubah pengujian gaya teks dan isi, tanpa harus kembali menghitung posisi dan ukuran yang diinginkan secara manual (<http://www.uibox.in>, 2014).

### **2.8.7 Apache**

Untuk menjalankan aplikasi *web* membutuhkan *web-server*. *Apache* adalah *web-server* yang mendukung bahasa *PHP* sehingga dapat dipakai untuk implemetasi aplikasi berbasis *PHP*. *Web-server* akan menterjemahkan bahasa *PHP* yang dipakai pada aplikasi *score online* untuk ditampilkan secara visual pada *browser* (Welling, 2001).

### 2.8.8 MySQL

MySQL merupakan software *RDBMS* (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Saat ini, MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengelolaan data, mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah, maupun besar. *Licensi MySQL* terbagi menjadi dua. Anda dapat menggunakan MySQL sebagai produk *open source* dibawah *GNU (General Public License)* gratis atau membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis (Raharjo, 2011).

### 2.8.9 PHP

PHP adalah sebuah bahasa *scripting* yang terpasang pada *HTML (Hypertext Markup Language)*. Tujuan utama bahasa ini adalah memungkinkan perancang *web* menulis halaman *web* dinamik dengan cepat. PHP dikatakan sebagai sebuah *serve-side embedded script language*, artinya perintah – perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi di sertakan pada halaman *html* biasa. Aplikasi–aplikasi yang di bangun dengan menggunakan *php* umumnya akan memberikan hasil pada *web browser*, tapi proses secara keseluruhan akan dijalankan di *server*. Pada prinsipnya *server* akan berkerja apabila ada permintaan dari *client* yang menggunakan kode–kode PHP.

Kode PHP di simpan sebagai *Plain Text* dalam format *ASCII* sehingga kode dapat di tulis di hampir semua *text editor* seperti *notepad*, *windows wordpad*.

Salah satu faktor mengapa *PHP* sangat populer sebagai bahasa *script* dalam pembuatan aplikasi berbasis *web* maupun situs *website* dinamis adalah karena bahasa ini mendukung demikian banyak sistem basis data, mulai dari *mySQL*, *MS-SQL*, *MS. Access*, *PostgreSQL* dan bahkan *Oracle*. (Irvan, 2006).

#### **2.8.10 Skala Likert**

Menurut Likert dalam buku Azwar S (2011, p. 139), sikap dapat diukur dengan metode rating yang dijumlahkan (*Method of Summated Ratings*). Metode ini merupakan metode penskalaan pernyataan sikap yang menggunakan distribusi respons sebagai dasar penentuan nilai skalanya. Nilai skala setiap pernyataan tidak ditentukan oleh derajat *favourable* nya masing-masing akan tetapi ditentukan oleh distribusi respons setuju dan tidak setuju dari sekelompok responden yang bertindak sebagai kelompok uji coba (*pilot study*). Prosedur penskalaan dengan metode rating yang dijumlahkan didasari oleh 2 asumsi (Azwar S, 2011, p 139), yaitu:

- a. Setiap pernyataan sikap yang telah ditulis dapat disepakati sebagai pernyataan yang *favorable* atau pernyataan yang tidak *avourable*.
- b. Jawaban yang diberikan oleh individu yang mempunyai sikap positif harus diberi bobot atau nilai yang lebih tinggi daripada jawaban yang diberikan oleh responden yang mempunyai pernyataan negatif.

Suatu cara untuk memberikan interpretasi terhadap skor individual dalam skala rating yang dijumlahkan adalah dengan membandingkan skor tersebut dengan harga rata-rata atau *mean* skor kelompok di mana responden itu termasuk (Azwar S, 2011, p.155). Salah satu skor standar yang biasanya digunakan dalam skala model Likert adalah skor-T, yaitu:

$$T = 50 + 10 \left[ \frac{X - \bar{X}}{s} \right]$$

Keterangan:

X = Skor responden pada skala sikap yang hendak diubah menjadi skor T

$\bar{X}$  = *Mean* skor kelompok

S = Deviasi standar skor kelompok

Perlu pula diingat bahwa perhitungan  $\bar{X}$  dan s tidak dilakukan pada distribusi skor total keseluruhan responden, yaitu skor sikap para responden untuk keseluruhan pernyataan (Azwar S, 2011,p.156).Skor sikap yaitu skor X perlu diubah ke dalam skor T agar dapatdiinterpretasikan. Skor T tidak tergantung pada banyaknya pernyataan, akan tetapi tergantung pada *mean* dan deviasi standar pada skor kelompok. Jika skor T yang didapat lebih besar dari nilai mean maka mempunyai sikap cenderung lebih *favourable* atau positif. Sebaliknya jika skor T yang didapat lebih kecil dari nilai mean maka mempunyai sikap cenderung tidak *favourable* atau negatif (Azwar S, 2011, p. 157).

### **2.8.11 System Development Life Cycle (SDLC)**

Menurut Jogiyanto (2002), siklus hidup sistem (*system life cycle*), atau yang disingkat SLC adalah “proses evolusi yang diikuti dalam menetapkan sistem dan sub sistem informasi berbasis komputer”. SLC terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem, karena tugas-tugas tersebut mengikuti sebuah pola yang teratur dan dilakukan secara *top-down*, SLC sering disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pengembangan dan penggunaan sistem.

Sedangkan *System Development Life Cycle* atau yang disingkat SDLC (Jogiyanto, 2002) adalah “metoda tradisional yang digunakan untuk membangun, memelihara dan mengganti suatu sistem informasi”. SDLC ini terdiri dari tujuh fase, diantaranya adalah :

*a. Project Identification and Selection*

Fase dimana kebutuhan sistem informasi secara keseluruhan diidentifikasi dan dianalisa guna mendapatkan gambaran yang utuh sehingga dapat dilakukan proses pengembangan sistem secara maksimal.

*b. Project Initiation and Planning*

Fase dimana suatu proyek sistem informasi yang potensial dilakukan dan direncanakan terinci dikembangkan untuk pengembangan sistem.

*c. Analisis*

Suatu fase dimana sistem yang sedang berjalan dipelajari dan alternatif sistem baru diusulkan.

*d. Logical Design*

Suatu fase dimana semua kegiatan fungsional dari sistem yang diusulkan untuk dikembangkan dan digambarkan secara independen.

*e. Physical Design*

Suatu fase dimana rancangan logis dari sebelumnya diubah dalam bentuk teknis yang terinci dimana pemrograman dan bentuk sistem dapat dibuat.

*f. Implementation*

Suatu fase dimana sistem informasi diuji dan digunakan untuk mendukung suatu organisasi.

#### *g. Maintenance*

Sistem informasi secara sistematis diperbaiki dan dikembangkan. siklus hidup pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan-tahapan utama dan langkah-langkah dalam proses pengembangannya. Sebagai awal dari pelaksanaan pengembangan sistem adalah proses kebijaksanaan dan perencanaan sistem. Kebijaksanaan sistem merupakan landasan dan dukungan dari manajemen puncak untuk membuat perencanaan. Sedangkan perencanaan sistem merupakan pedoman untuk melakukan pengembangan sistem.

#### **2.8.12 Metode Black Box**

Menurut Pressman (2005), pengujian black box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* diaplikasikan selama tahap akhir pengujian. Karena pengujian *black box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi.

Pengujian *black box* berusaha menemukan kesalahan dalam fungsi-fungsi yang tidak benar atau hilang, kesalahan interface, kesalahan dalam struktur data atau akses database eksternal, kesalahan kinerja (*performa*), inisialisasi dan kesalahan terminasi.