

BAB II

LANDASAN TEORI

Pada bab ini akan dijelaskan landasan teori yang terkait dengan permasalahan yang dibahas dan juga menjelaskan sistem yang digunakan pada Tugas Akhir ini. Adapun landasan teori yang digunakan sebagai berikut:

2.1 Data, Informasi, dan Proses Pengolahan Data

Menurut Bocij dkk.(2008:6) Data adalah fakta atau hasil pengamatan yang dianggap memiliki sedikit atau tidak memiliki nilai sampai mereka telah diproses dan diubah menjadi informasi. Informasi adalah data yang diolah sesuai kebutuhan sehingga bisa dipahami oleh pengguna atau penerima informasi.

Proses pengolahan data harus dilakukan dengan baik dan sesuai dengan konteks agar mudah dipahami oleh pengguna. Pada gambar berikut menunjukkan bagaimana sebuah data diubah menjadi informasi. Beberapa cara yang dapat digunakan untuk melakukan proses data seperti berikut ini (Bocij dkk., 2008:8).

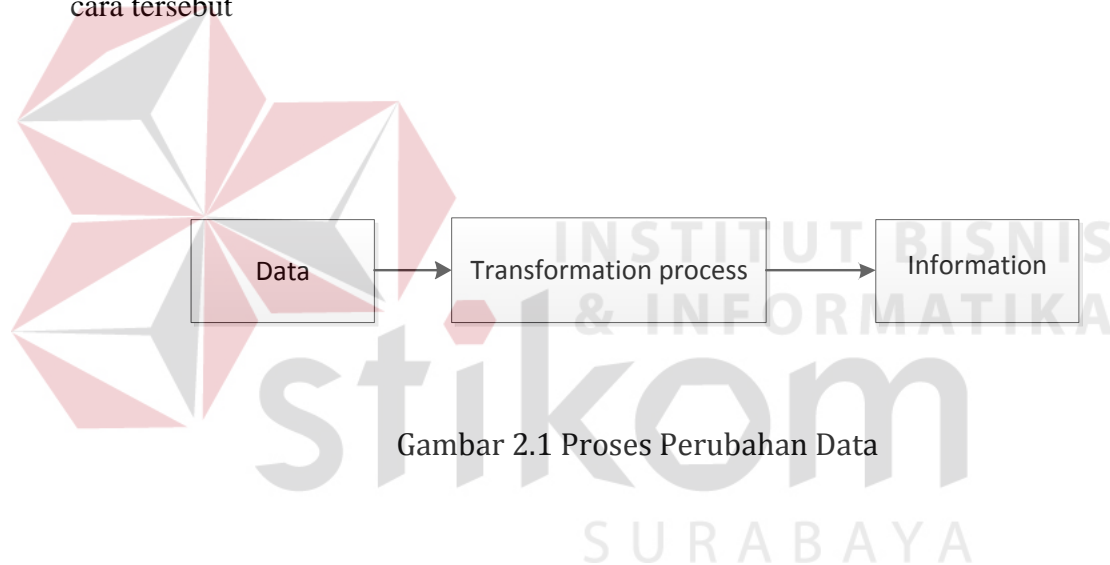
1. *Classification*, mengelompokkan data berdasarkan kategori tertentu.
2. *Rearranging/Sorting*, mengelola data dengan dalam satu kelompok dan diatur atau diurutkan berdasarkan urutan tertentu.
3. *Aggregating*, mengolah data dengan meringkasnya seperti menghitung rata-rata, total, atau subtotal.
4. *Performing calculations*, mengolah data dengan melakukan perhitungan terhadap data bisa dengan melakukan perkalian, penjumlahan, atau pengurangan. Sebagai contoh dalam penjualan barang untuk mendapatkan total

harga yang harus dibayar, kita melakukan perhitungan harga satuan dikalikan harga barang.

5. *Selection*, mengolah data dengan memilih atau memilah data berdasarkan kriteria tertentu. Sebagai contoh, sebuah contoh sebuah perusahaan dapat membuat daftar pelanggan potensial berdasarkan besarnya pendapatan diatas rata-rata pelanggan lainnya.

Lima cara tersebut bukannya merupakan cara mutlak bisa saja ada cara lain yang dapat digunakan untuk mengolah data, misalkan dengan kombinasi diantara lima

cara tersebut



Gambar 2.1 Proses Perubahan Data

Hasil pengolahan data menjadi informasi harus memenuhi karakteristik sebagai berikut (Tarigan dkk., 2010).

1. *Relevant*, informasi dikatakan relevan jika berguna untuk menjawab kebutuhan organisasi.
2. *Relieble*, informasi yang disajikan lengkap tanpa menghilangkan elemen penting dari bagian yang akan digunakan dalam mengambil keputusan.
3. *Timely*, disajikan tepat waktu atau sesuai dengan waktunya agar tidak kehilangan *value of moment*-nya.

2.2 Produktivitas Bahan Baku Tebu

Kualitas dan kuantitas produksi gula tergantung pada bahan baku tebu yang diproduksi, ada beberapa parameter yang digunakan untuk mengetahui produktivitas bahan baku tebu.

1. Luas Areal

Luas areal merupakan sasaran yang harus terpenuhi guna memproduksi tanaman tebu untuk memenuhi pasok bahan baku tebu selama musim giling.

Satuan yang digunakan hektar(ha).

2. Tebu Digiling

Tebu digiling merupakan jumlah tebu yang tergiling selama musim giling dalam satuan berat(ton). Pendapatan perusahaan mayoritas terdorong dari ketersediaan bahan baku semakin banyak bahan baku yang mendukung proses produksi semakin banyak pula produksi yang dihasilkan oleh perusahaan, semakin besar pula potensi pendapatan perusahaan dan begitu sebaliknya. Oleh karena itu jumlah bahan baku sangat berpengaruh terhadap pendapatan perusahaan sehingga hal ini menjadi titik kritis dan kunci dalam keberhasilan usaha di pabrik gula.

3. Hablur

Hablur adalah gula sukrosa yang dikristalkan. Pembentukan gula yang sebenarnya terjadi di lahan(on farm). Tugas pabrik gula berfungsi sebagai alat ekstraksi untuk mengeluarkan nira dari batang tebu dan mengolahnya menjadi gula kristal. Hablur yang dihasilkan mencerminkan rendemen tebu. Satuan yang digunakan berat(ton).

4. Rendemen

Rendemen tebu adalah kadar kandungan gula yang ada didalam batang tebu yang dinyatakan dalam persen. Bila dikatakan rendemen tebu 10% , artinya ialah bahwa dari 100 kg tebu yang digiling dipabrik gula akan diperoleh gula 10 kg. Untuk mendapatkan rendemen tinggi tanaman harus bermutu baik dan di tebang pada saat yang tepat.

5. Tebu/ha

Menunjukkan produksi tanaman tebu dalam satuan luas hektar. Baik buruknya produksi dipengaruhi beberapa faktor diantaranya: kesuburan tanah, pemeliharaan, potensi varietas dan iklim.

6. Hablur/ha

Merupakan parameter dari produktivitas lahan. Semakin banyak kuintal tebu yang diproduksi dalam hektar tanam dan rendemennya, maka kebun tersebut akan mendapatkan hablur yang tinggi.

2.3 Sistem Dashboard

2.3.1 Visualisasi

Menurut Frey (2008:4), sebuah visualisasi yang tepat adalah semacam narasi yang memberikan jawaban jelas atas pertanyaan tanpa rincian yang tidak berhubungan/asing. Dengan berfokus pada tujuan awal dari pertanyaan, Anda dapat menghilangkan rincian seperti itu karena pertanyaan itu memberikan acuan untuk apa yang diperlukan dan apa yang tidak diperlukan.

Menurut McCormick (1987:3), Visualisasi adalah metode komputasi. Mengubah simbol ke dalam geometris, memungkinkan peneliti untuk mengamati

simulasi dan perhitungan. Visualisasi menawarkan metode untuk melihat yang tak terlihat. Memperkaya proses penemuan ilmiah dan mendorong pengetahuan yang tak terduga. Dalam banyak bidang hal ini sudah merevolusi cara pandang ilmuwan terhadap ilmu pengetahuan.

Visualisasi mencakup baik pemahaman gambar dan perpaduan gambar. Artinya, visualisasi adalah alat untuk menafsirkan data gambar yang dimasukkan ke komputer, dan untuk menghasilkan gambar dari data multi-dimensi yang kompleks. Mempelajari mekanisme tersebut pada manusia dan komputer yang memungkinkan dengan tujuan untuk memahami, menggunakan, dan mengkomunikasikan informasi visual. Visualisasi menyatukan sebagian besar bidang independen dan konvergen, dari berikut ini:

1. *Computer Graphic*
2. *Image Processing*
3. *Computer Vision*
4. *Computer Aided Design (CAD)*
5. *Signal Processing*
6. *User Interface Studies*

2.3.2 Pengertian 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. *Dashboard* itu sebuah tampilan pada satu monitor komputer penuh yang berisi informasi yang bersifat kritis, agar kita dapat mengetahui hal-hal yang perlu

diketahui. Biasanya kombinasi teks dan grafik, tetapi lebih ditekankan pada grafik (Few, 2006:34).

2.3.3 Tujuan Penggunaan *Dashboard*

Tujuan penggunaan *dashboard* menurut Eckerson (2006a:5) yaitu:

1. Mengkomunikasikan Strategi

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

2. 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.

3. 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.

2.3.4 Jenis *Dashboard*

Dashboard bisa dikelompokkan sesuai dengan *level* manajemen yang didukungnya menurut Eckerson dan Few pada (Hariyanti 2008:10) yaitu:

1. *Strategic Dashboard*

- a. Mendukung manajemen level strategis.
- b. Informasi untuk membuat keputusan bisnis, memprediksi peluang, dan memberikan arahan pencapaian tujuan strategis.

- c. Fokus pada pengukuran kinerja *high-level* dan pencapaian tujuan strategis organisasi.
- d. Mengadopsi konsep *Balance Score Card*.
- e. Informasi yang disajikan tidak terlalu detail.
- f. Konten informasi tidak terlalu banyak dan disajikan secara ringkas.
- g. Informasi disajikan dengan mekanisme yang sederhana, melalui tampilan yang *unidirectional*.
- h. Tidak di desain untuk berinteraksi dalam melakukan analisis yang lebih detail.
- i. Tidak memerlukan data *real time*.

2. *Tactical Dashboard*

- a. Mendukung manajemen *tactical*.
- b. Memberikan informasi yang diperlukan oleh analisis untuk mengetahui penyebab suatu kejadian.
- c. Fokus pada analisis untuk menemukan penyebab dari suatu kondisi atau kejadian tertentu.
- d. Dengan fungsi *drill down* dan navigasi yang baik.
- e. Memiliki konten informasi yang lebih banyak (Analisis perbandingan, pola/tren, evaluasi kerja).
- f. Menggunakan media penyajian yang “cerdas” yang memungkinkan pengguna melakukan analisis terhadap data yang kompleks.
- g. Didesain untuk berinteraksi dengan data.
- h. Tidak memerlukan data *real time*.

3. *Operational Dashboard*

- a. Mendukung manajemen *level* operasional.
- b. Memberikan informasi tentang aktivitas yang sedang terjadi, beserta perubahannya secara *real time* untuk memberikan kewaspadaan terhadap hal-hal yang perlu direspon secara cepat.
- c. Fokus pada *monitoring* aktifitas dan kejadian yang berubah secara konstan.
- d. Informasi disajikan spesifik, tingkat kedetailan yang cukup dalam.
- e. Media penyajian yang sederhana.
- f. *Alert* disajikan dengan cara yang mudah dipahami dan mampu menarik perhatian pengguna.
- g. Bersifat dinamis, sehingga memerlukan data *real time*.
- h. Didesain untuk berinteraksi dengan data, untuk mendapatkan informasi yang lebih detail, maupun informasi pada level lebih atas (*Higher Level Data*).

2.3.5 Karakteristik *Dashboard*

Karakteristik *dashboard* menurut (Eckerson, 2006b:117) yaitu:

1. Model pemrosesan berdasarkan kejadian yaitu menangkap kejadian setiap saat dari beberapa sistem yang mencakup dan mempengaruhi proses bisnis.
2. Aturan bisnis yang kuat yaitu mengizinkan penggunaannya membuat peringatan, target, ambang untuk menilai kinerja individu.
3. *Dashboard* bisnis yang *user friendly* yaitu memperbarui nilai sebagai aliran kejadian melalui sistem dan menempatkan nilai tersebut dalam hubungan dengan menghubungkan ke pencapaian bisnis.

4. Sebuah sistem aliran kerja yang bergabung dan bekerjasama yang mengijinkan penggunaanya untuk memulai proses secara formal dan informal, yang dengan proses itu pengguna dapat berkolaborasi mendiskusikan hasilnya.

Beberapa karakteristik *dashboard* menurut Malik (Hariyanti, 2008:8)

yaitu:

1. Sinergi

Ergonomis dan memiliki tampilan visual yang mudah dipahami oleh pengguna. *Dashboard* mensinergikan informasi dari berbagai aspek yang berbeda dalam satu layar.

2. Monitor

Menampilkan KPI yang diperlukan dalam pembuatan keputusan dalam domain tertentu, sesuai dengan tujuan pembangunan *dashboard* tersebut.

3. Akurat

Informasi yang disajikan harus akurat, dengan tujuan untuk mendapatkan kepercayaan dari penggunaanya.

4. Responsif

Merespon *threshold* yang telah didefinisikan, dengan memberikan *alert* (seperti bunyi *alarm*, *blinker*, *email*) untuk mendapatkan perhatian pengguna terhadap hal-hal yang kritis.

5. *Timely*

Menampilkan informasi terkini yang diperlukan untuk pengambilan keputusan.

6. Interaktif

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

7. *More Data History*

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

8. *Personalized*

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

9. *Analitical*

Fasilitas untuk melakukan analisis seperti sebab akibat.

10. *Collaborative*

Fasilitas pertukaran catatan laporan antar pengguna mengenai hasil pengamatan *dashboard*-nya masing-masing yaitu sarana komunikasi dalam melakukan fungsi manajemen dan *control*.

11. *Trackability*

Memungkinkan setiap pengguna untuk mengkustomisasi nilai yang akan dilacak.

2.3.6 *Komponen Dashboard*

Dalam memahami perbedaan diantara ketiga jenis *dashboard* kinerja, perlu untuk mengetahui masing-masing komponen aplikasi yang digunakan. Meskipun tidak ada aturan keras dan cepat tentang penggunaan komponen, Gambar 2.2 yang memberikan beberapa pedoman umum (Eckerson, 2006b:106).

PERFORMANCE DASHBOARD COMPONENTS			
	Operational Dashboard	Tactical Dashboard	Strategic Dashboard
Monitoring	Dashboard	BI Portal	Scorecard
Analysis	Statistical models Decision engines	OLAP analysis Interactive reporting Advanced visualization Scenario modeling	Time-series analysis Standard reports
Management	Alerts Agents	Workflow Usage monitoring Auditing	Meetings Annotations Strategy maps

The monitoring, analysis, and management components that most commonly comprise the three types of performance dashboards.

Gambar 2.2 Komponen *Dashboard* Kinerja

1. Komponen *Dashboard* Operasional

Dashboard operasional menggunakan antarmuka *dashboard* untuk memantau proses operasional. *Dashboard* memberikan peringatan yang memberitahukan pengguna tentang kondisi pengecualian dalam proses yang sedang mereka pantau sehingga mereka dapat bertindak cepat untuk memperbaiki masalah atau memanfaatkan peluang.

2. Komponen *Dashboard* Taktis

Dashboard taktis sering menampilkan hasil dalam *business intelligence* (BI) portal yang berisi grafik dan tabel serta dokumen lainnya pengguna perlu untuk memantau proyek atau proses yang mereka kelola. Portal ini dibangun ke sebagian besar alat BI dan biasanya mengintegrasikan dengan portal komersial yang banyak digunakan perusahaan untuk menjalankan *intranet* perusahaan mereka.

3. Komponen *Dashboard* Strategis

Dashboard Strategis menggunakan antarmuka *scorecard* untuk melacak kinerja terhadap tujuan strategis. Meskipun mereka mirip dengan antarmuka *dashboard*, *scorecard* umumnya melacak kemajuan kelompok secara bulanan daripada secara tepat waktu. *Scorecard* umumnya menampilkan lebih metrik seluruh spektrum yang lebih luas dari organisasi daripada *dashboard*, terutama di *scorecard* perusahaan. Informasi kinerja dalam antarmuka *scorecard* biasanya lebih diringkas dari dalam antarmuka *dashboard*.

2.4 Kesalahan Umum Pembuatan *Dashboard*

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

1. Melebihi batas pada satu layar monitor komputer. Hal ini mengacu pada tampilan *dashboard*.
2. 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.
3. 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: \$3.8M akan lebih baik dibanding \$3.848.352,93.

4. 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)
5. Memilih media tampilan yang tidak tepat: maksudnya adalah salah memilih media (bar, pie, circle, atau radar) .
6. Menyajikan variasi berbeda yang sia-sia: misalnya, menyajikan chart penjualan pada beberapa daerah dengan menggunakan pie, radar, dan bar pada *dashboard* yang sama.
7. Menggunakan media tampil yang desainnya payah.
8. Menampilkan kuantitas data secara tidak akurat: contoh sebuah grafik batang yang dimulai angka \$500.000 bukan \$0.
9. 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.
10. 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.
11. 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.
12. Salah atau berlebihan menggunakan warna. Sebaiknya menggunakan warna yang tepat. Dan tidak serampangan dalam menggunakan warna.
13. Mendesain tampilan yang tidak atraktif. Misalnya tidak ada comboboxnya.

2.5 *Key Performance Indikator*

Key Performance Indicator (Hariyanti, 2008) adalah indikator yang merepresentasikan kinerja dari proses yang dilaksanakan. *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 KPI dapat digunakan sebagai alat untuk meningkatkan kinerja organisasi secara dramatis. Contoh dari penjelasan diatas 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 Grafik

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 (Hariyanti, 2008).

No	Keterhubungan Data	Jenis Grafik yang sesuai
1	Perbandingan nominal	a. Grafik bar (horisontal atau vertikal). b. Grafik titik (jika 0 tidak termasuk dalam skala nilai).
2	<i>Time-series</i>	a. Grafik garis (untuk melihat tren seluruh data). b. Perbandingan (antar nilai individu). c. Grafik titik yang dihubungkan dengan garis (untuk melihat nilai individu sekaligus tren data secara keseluruhan).
3	<i>Ranking</i>	a. Grafik bar (horisontal atau vertikal) b. Grafik titik (jika 0 tidak termasuk dalam skala nilai).
4	Bagian dari keseluruhan	a. Grafik bar (horisontal maupun vertikal). b. Grafik stack bar. c. Pie chart.
5	<i>Deviasi</i>	a. Grafik garis. b. Grafik titik yang dihubungkan dengan garis.
6	Distribusi frekuensi	a. Grafik bar vertikal/histogram (untuk menunjukkan nilai individu). b. Grafik garis/poligon frekuensi (untuk menunjuk tren data secara keseluruhan).
7	Korelasi	a. Grafik titik dan garis (<i>scatter-plot</i>)

2.7 Monitoring

Menurut Casley dan Kumar (1989:76), *monitoring* merupakan pengidentifikasian kesuksesan atau kegagalan secara nyata maupun potensial sedini mungkin dan sewaktu-waktu bisa menyelesaikan operasionalnya dengan tujuan meninjau kemajuan dan mengusulkan langkah untuk mewujudkan tujuan. *Monitoring* juga dapat diartikan sebagai penilaian yang terus menerus terhadap

aktifitas proyek sebagaimana telah direncanakan, waktu pelaksanaan sebagaimana telah dijadwalkan, dan kemajuan dalam mencapai tujuan.

2.8 Controlling

Menurut Williams (2009:11), *controlling* atau pengendalian adalah proses pemantauan kemajuan menuju pencapaian tujuan dan mengambil tindakan korektif ketika kemajuan tidak sedang dibuat. Dasar dari proses pengendalian mencakup penetapan standar untuk mencapai tujuan, membandingkan kinerja actual dengan standar tersebut, dan kemudian membuat perubahan untuk kembali menuju ke performa standar tersebut.

2.9 Analisis dan Perancangan Sistem

Menurut Kendall dan Kendall (2003:7), analisis dan perancangan sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi. Analisis sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi.

2.10 *Unified Modeling Language*

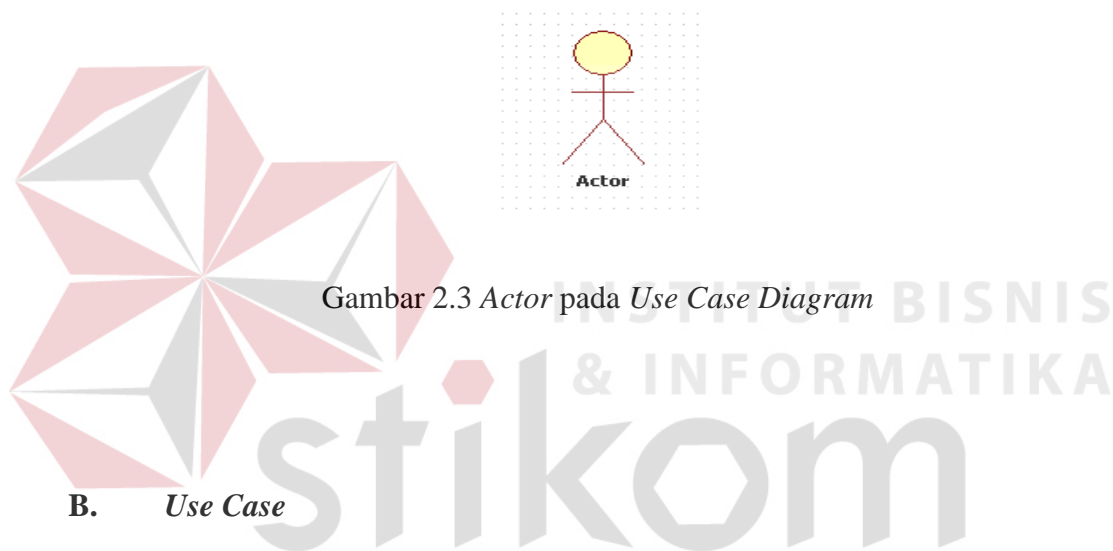
Menurut Nugroho (2005:16), pemodelan *visual* adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Notasi-notasi baku sangat penting demi suatu alasan komunikasi. Dengan notasi-notasi pemodelan yang bersifat baku komunikasi yang baik akan terjalin dengan mudah antar anggota tim pengembang sistem/perangkat lunak dan antara anggota tim pengembang dengan para pengguna. Untuk melakukan pemodelan sistem/perangkat lunak, dalam buku ini notasi-notasi *Unified Modeling Language* (UML) yang akan digambarkan secara elektronik (dengan bantuan komputer) lewat sarana perangkat lunak. Dengan pemodelan menggunakan UML ini, pengembang dapat melakukan:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
2. Penelaahan bagaimana objek-objek dalam sistem saling mengirim pesan (*message*) dan saling bekerjasama satu sama lain.
3. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.
4. Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.

A. Actor

Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan

informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man*. *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*.

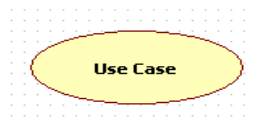


Gambar 2.3 Actor pada Use Case Diagram

B. Use Case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Catatan : *Use case diagram* adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian.








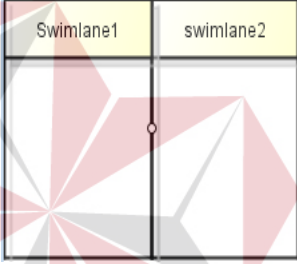
Gambar 2.4 *Use Case* pada *Use Case Diagram*

C. *Activity Diagram*

Activity diagram memiliki pengertian yaitu lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Memiliki struktur diagram yang mirip *flowchart* atau *data flow diagram* pada perancangan terstruktur. Memiliki pula manfaat yaitu apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Dan *activity* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram* (Felici, 2004).

Tabel 2.2 Simbol *Activity Diagram*

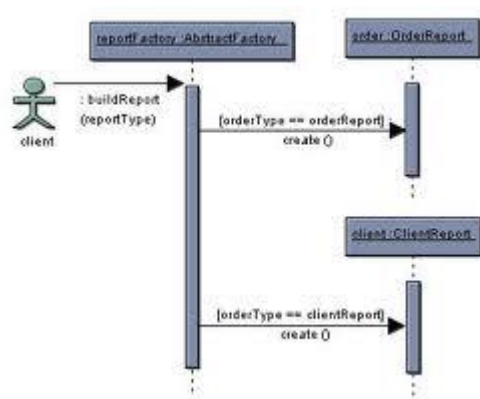
No.	Notasi	Penjelasan
1.		<i>Activity</i> menggambarkan sebuah pekerjaan atau tugas dalam <i>workflow</i> , pada UML <i>activity</i> digambarkan dengan simbol kotak
2.		Start state dengan tegas menunjukkan dimulainya suatu <i>workflow</i> pada sebuah <i>activity diagram</i> . Hanya terdapat satu <i>start state</i> dalam sebuah <i>workflow</i> serta pada UML, <i>start state</i> digambarkan dengan simbol lingkaran yang solid

3.		<p><i>End state</i> menggambarkan akhir atau terminal dari pada sebuah <i>activity diagram</i>. Bisa terdapat lebih dari satu <i>end state</i> pada sebuah <i>activity diagram</i>. Pada UML, <i>end state</i> digambarkan dengan simbol sebuah <i>bull's eye</i></p>
4.		<p><i>State transition</i> menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya. Pada UML, <i>state transition</i> digambarkan oleh sebuah <i>solid line</i> dengan panah</p>
5.		<p><i>Decision</i> adalah suatu titik atau <i>point</i> pada <i>activity diagram</i> yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi. Pada UML, <i>decision</i> digambarkan dengan sebuah simbol <i>diamond</i></p>
6.		<p>Obyek <i>swimlane</i> untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu atau biasa disebut <i>entitas</i> pada <i>work flow</i>.</p>

D. Sequence Diagram

Sequence diagram adalah interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas 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. Gambar 2.5 merupakan contoh dari *sequence diagram*.

Gambar 2.5 *Sequence Diagram*

Sequence diagram biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi *method* didalam sebuah *system*.

E. Class Diagram

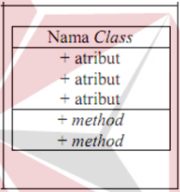


kumpulan objek-objek dengan dan yang mempunyai struktur umum, *behavior* umum, relasi umum, dan *semantic*/kata yang umum. *Class-class* ditentukan/ditemukan dengan cara memeriksa objek-objek dalam *sequence diagram* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. *Class* sebaiknya diberi nama menggunakan kata benda sesuai dengan domain/bagian/kelompoknya (Whitten L. Jeffery et al, 2004).



Elemen-elemen *class diagram* dalam pemodelan UML terdiri dari: *Class-class*, struktur *class*, sifat *class* (*class behavior*), perkumpulan/gabungan (*association*), pengumpulan/kesatuan (*agregation*), ketergantungan (*dependency*),

relasi-relasi turunannya, keberagaman dan indikator navigasi, dan *role name* (peranan/tugas nama).

Simbol-simbol pada *class diagram*:

Tabel 2.3 Simbol *Class Diagram*

No.	Simbol	Penjelasan
1.		<p><i>Class</i> adalah blok-blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i>. Bagian tengah mendefinisikan <i>property</i>/atribut <i>class</i>. Bagian akhir mendefinisikan <i>method</i> dari sebuah <i>class</i>.</p>
2.		<p><i>Association</i> merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i>. Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i>.(Contoh: <i>One-to-one</i>, <i>one-to-many</i>,<i>many-to-many</i>).</p>
3.		<p><i>composition</i> Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.</p>

4.		<p><i>Dependency</i> : Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i>. Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.</p>
5.		<p><i>Aggregation</i> : <i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi.</p>

2.11 Database

Menurut Marlinda (2004:1), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya. Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, kesulitan pengaksesan data, isolasi data untuk standarisasi, banyak pemakai(*multiple user*), masalah keamanan(*security*), masalah kesatuan(*integration*), dan masalah kebebasan data(*dataindependence*).

2.12 Hypertext Preprocessor

Menurut Firdaus (2007:2), PHP merupakan singkatan dari *Hypertext Preprocessor*, adalah sebuah bahasa *scripting* berbasis *server side scripting* yang

terpasang pada HTML dan berada di *server* dieksekusi di *server* dan digunakan untuk membuat halaman *web* yang dinamis. Sebagian besar sintaksnya mirip dengan bahasa C atau java, ditambah dengan beberapa fungsi PHP yang spesifik. Tujuan utama bahasa ini adalah untuk memungkinkan perancang *web* menulis halaman *web* dinamis dengan cepat.

Halaman *web* biasanya disusun dari kode-kode HTML yang disimpan dalam sebuah *file* berekstensi *.html*. *File* HTML ini dikirimkan oleh *server* (atau *file*) ke *browser*, kemudian *browser* menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan program PHP, program ini harus diterjemahkan oleh *web server* sehingga menghasilkan kode html yang dikirim ke *browser* agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode HTML sehingga dapat langsung ditampilkan bersama dengan kode-kode HTML tersebut. Program php dapat ditambahkan dengan mengait program tersebut di antara tanda `<? dan ?>`. Tanda-tanda tersebut biasanya digunakan untuk memisahkan kode php dari kode HTML. *File* HTML yang telah dibubuhi program php harus diganti ekstensi-nya menjadi *.php* atau *.php3*.

2.13 MySQL

MySQL adalah *database* yang menghubungkan *script* PHP menggunakan perintah *query* dan *escape character* yang sama dengan PHP. PHP memang mendukung banyak *database*, tetapi untuk membuat sebuah *web* yang dinamis selalu *Up to Date*, MySQL merupakan pilihan *database* tercepat saat ini (Firdaus, 2007:3).

MySQL (*My Structured Query Language*) atau yang bisa dibaca mai-sekuel adalah program pembuat dan pengelola *database*. Selain itu data Mysql juga merupakan program pengakses *database* yang bersifat jaringan, sehingga dapat digunakan untuk Aplikasi *Multi User* (banyak pengguna). Kelebihan dari MySQL adalah menggunakan bahasa *query* (permintaan) standar SQL (*Structured Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur.

2.14 Java Script

Menurut Hakim (2010:2), *java script* merupakan bahasa *scripting* yang dapat bekerja di sebagian besar *web browser*. *Java script* dapat disisipkan di dalam *web* menggunakan *tag script*. *Java script* dapat digunakan untuk banyak tujuan, misalnya untuk membuat efek *roolover* baik gambar maupun *text*, dan untuk membuat AJAX *Java script* adalah bahasa yang digunakan untuk AJAX. Kode *java script* juga dapat diletakkan di *file* tersendiri yang berekstensi *javascript* (.js). *Script* tersebut akan dieksekusi ketika dipanggil berdasarkan *trigger* pada *event* tertentu.

2.15 Highcharts

Highcharts adalah *library* pembuatan *chart* yang ditulis dalam *JavaScript* murni, menawarkan cara mudah untuk menambahkan grafik interaktif ke situs *web* atau aplikasi *web*. *Highcharts* saat ini mendukung *line*, *spline*, *area*, *area spline*, *column*, *bar*, *pie*, *scatter*, *angular gauges*, *area range*, *area spline range*, *column range*, *bubble*, *box plot*, *error bars*, *funnel*, *waterfall* dan *polar chart types*.