

BAB II

LANDASAN TEORI

II.1. Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Teori sistem melahirkan konsep-konsep futuristik. Salah satu konsep yang terkenal adalah konsep sibernetika (*cybernetics*). Konsep kajian ilmiah ini terutama berkaitan dengan upaya menerapkan berbagai disiplin ilmu, yaitu ilmu perilaku, fisika, biologi dan teknik. Oleh karena itu, sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan oleh manusia sehingga studi tentang robotika, kecerdasan buatan (*artificial intelligence*), dan lain adalah masukan (*input*), pengolahan (*processing*), dan keluaran (*output*). (Lindawati dan Irma Salamah, 2012)

Konsep lain yang terkandung di dalam definisi tentang sistem adalah konsep sinergi. Konsep ini mengandaikan bahwa di dalam suatu sistem, *output* dari suatu organisasi diharapkan lebih besar dari pada *output* individual atau *output* masing-masing bagian.

Sebuah sistem terdiri atas bagian-bagian atau komponen yang terpadu untuk satu tujuan. Model dasar dari bentuk sistem ini adalah adanya masukan, pengolahan, dan keluaran. Namun demikian sistem ini dapat dikembangkan hingga

menyertakan media penyimpanan. Sistem dapat terbuka dan tertutup. Sistem informasi biasanya adalah sistem terbuka, yang berarti bahwa sistem tersebut dapat menerima berbagai masukan dari lingkungan sekitarnya. (Hastutie, 2012)

II.2. Informasi

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks pengambilan keputusan. (Lindawati dan Irma Salamah, 2012)

Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Informasi yang disampaikan kepada pemakai mungkin merupakan hasil dari data yang dimasukkan ke dalam pengolahan. Akan tetapi dalam kebanyakan pengambilan keputusan yang kompleks, informasi hanya dapat menambah kemungkinan kepastian atau mengurangi bermacam-macam pilihan.

II.3. Sistem Informasi

Sistem informasi bukan merupakan hal yang baru. Yang baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi yang memungkinkan manajer merencanakan serta mengendalikan operasi telah ada.

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat *manajerial* dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Lindawati dan Irma Salamah, 2012).

II.4. Biaya Produksi

Sugianto dan kawan-kawan dalam bukunya “Ekonomi mikro” mengatakan bahwa: “Biaya produksi adalah sejumlah uang yang dikeluarkan untuk mendapatkan sejumlah input yaitu secara akuntansi sama dengan jumlah uang keluar yang tercatat”.

Hansen dan Mowen dalam terjemahan Fitriasisari dan Kwary juga menyatakan bahwa: “Biaya produksi merupakan biaya yang berkaitan dengan pembuatan barang dan penyediaan jasa”. Biaya produksi adalah sejumlah biaya / uang yang dikeluarkan untuk dapat melakukan kegiatan produksi barang.

II.4.1. Unsur Biaya Produksi

Kebanyakan perusahaan manufaktur membagi biaya produksi kedalam tiga kategori besar yaitu biaya bahan langsung (*direct material*), tenaga kerja

langsung (*direct labour*), dan biaya overhead pabrik (*manufacturing goverhead*). (Nur Bahagia, 2006).

a. Biaya Bahan Baku

Bahan baku merupakan bahan yang membentuk bagian menyeluruh produk jadi, bahan baku yang diolah dalam perusahaan manufaktur dapat diperoleh dari pembelian lokal, impor atau dari pengolahan sendiri.

b. Biaya Tenaga Kerja Langsung

Biaya tenaga kerja langsung adalah salah satu unsur dari harga pokok produk karena itu biaya tenaga kerja diperlukan dalam penentuan harga pokok produk per unit.

c. Biaya Overhead Pabrik

Biaya *overhead* pabrik merupakan beban biaya pabrik tidak langsung selama periode yang akan datang.

II.4.2. Macam-macam Biaya Produksi

1. Biaya produksi jangka pendek

Biaya produksi jangka pendek diturunkan dari fungsi produksi jangka pendek. Dengan demikian biaya produksi jangka pendek juga dicirikan oleh adanya biaya tetap. (Yuke, 2009)

2. Biaya produksi jangka panjang

Biaya produksi jangka panjang biaya yang dapat disesuaikan untuk tingkat-tingkat produksi tertentu. (Yuke,2009)

II.4.3. Metode Harga Pokok Produksi

Menurut Endra Setyaningsih (2013) dua pendekatan tersebut yaitu *full costing* dan *variabel costing*, berikut penjelasannya :

1. *Full Costing*

Metode *Full Costing* yaitu metode yang menentukan harga pokok produksi yang memperhitungkan semua unsur biaya produksi ke dalam harga pokok produksi, yang terdiri dari biaya bahan baku, biaya tenaga kerja langsung, dan biaya *overhead* pabrik, baik yang berperilaku variabel maupun tetap. Dengan demikian harga pokok produksi menurut *full costing* terdiri dari unsur biaya produksi. Berikut ini merupakan contoh perhitungan dalam metode *Full Costing* :

Harga Pokok Produksi :

Biaya Bahan Baku	Rp. xxx.xxx
Biaya Tenaga Kerja Langsung	Rp. xxx.xxx
Biaya Overhead Pabrik Tetap	Rp. xxx.xxx
Biaya Overhead Pabrik Variabel	Rp. xxx.xxx
	----- +
Harga Pokok Produk	Rp. xxx.xxx

(Sumber : W. Wiratna Sujarweni, 2015).

2. *Variable Costing*

Metode *Variable Costing* yaitu metode yang menentukan harga pokok produksi yang hanya memperhitungkan unsur biaya produksi yang berperilaku variabel ke dalam harga pokok produksi, yang terdiri dari

biaya bahan baku, biaya tenaga kerja langsung, dan biaya overhead pabrik variabel. Dengan demikian harga pokok produksi menurut *Variabel Costing* terdiri dari unsur biaya produksi.

II.4.4. Harga Jual

Menurut Risa Ferisa (2016), pendekatan umum dalam penentuan harga jual adalah menambahkan angka perkiraan laba (*markup*) pada harga pokok. *Markup* adalah selisih antara harga jual dan harga pokok produk. *Markup* biasanya berupa persentase tertentu dari harga pokok produk. Pendekatan ini disebut dengan *cost-plus pricing* karena persentase markup yang telah ditentukan dimuka ditambahkan pada angka harga pokok untuk menentukan harga jual.

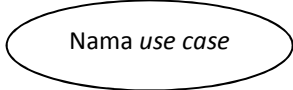
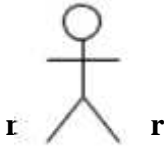


II.5. Unified Modeling Language (UML)

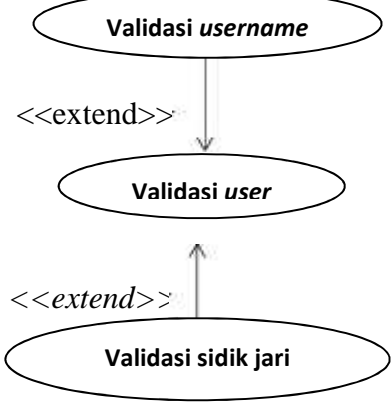

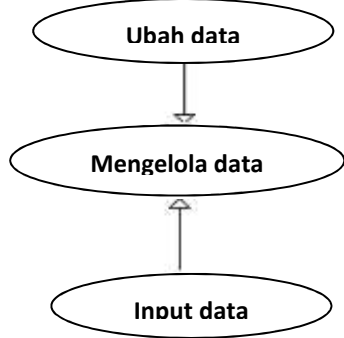
UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam berkomunikasi beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. (Yuni Sugiarti, 2013)

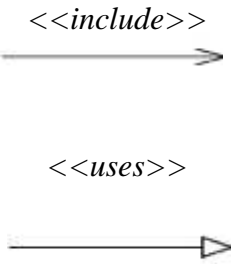
II.5.1. Use Case Diagram

Use case diagram atau diagram use case merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antar satu atau lebih *actor* dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Terdapat beberapa simbol dalam menggambarkan diagram *use case*, yaitu :

Tabel II.1. Simbol – Simbol *Use Case* Diagram

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal diawal frase nama <i>use case</i></p>
<p>Aktor / <i>actor</i></p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>aktor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal nama <i>aktor</i>.</p>
<p>Asosiasi/ <i>association</i></p> 	<p>Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p> <p><<<i>extend</i>>></p> 	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu: mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek; biasanya <i>use</i></p>

	<p><i>case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <pre> graph TD A([Validasi sidik jari]) -- "<<extend>>" --> B([Validasi user]) B -- "<<extend>>" --> C([Validasi username]) </pre> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <pre> graph TD A([Input data]) --> B([Mengelola data]) B --> C([Ubah data]) </pre> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>

<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p style="text-align: center;">  </p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <p><i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</p> <p><i>include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.</p>
--	---

(Sumber : Rosa A.S, M.Shalahuddin; 2013)

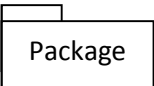
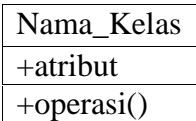

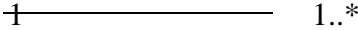
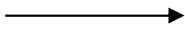
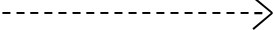
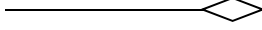
II.5.2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Atribut mendeskripsikan property dengan sebaris teks didalam kotak kelas tersebut.
3. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel II.2. Simbol-Simbol Diagram Kelas

Simbol	Deskripsi
Package 	Package merupakan bungkusan dari satu kelas atau lebih
Operasi 	Kelas pada struktur system
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>Directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan 	Relasi antar kelas dengan makna Kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna Semua bagian (<i>whole part</i>)

(Sumber : Yuni Sugiarti ; 2013)



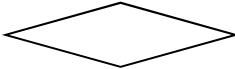

II.5.3. Activity Diagram


Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. *Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir.

Activity Diagram merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses- proses dan jalur-jalur aktivitas dari level atas secara umum (Yuni Sugiarti, 2013).

Berikut adalah tabel II.3. yang menggambarkan simbol-simbol yang digunakan pada diagram aktivitas :

Tabel II.3. Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir	Status akhir yang dilakukan sistem, sebuah diagram

	aktivitas memiliki sebuah status akhir.
Swimlane <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">nama swimlane</div>	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

(Sumber : Rosa A.S, M.Shalahuddin ; 2013)

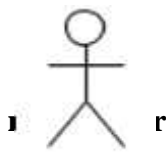
II.5.4. Sequence Diagram


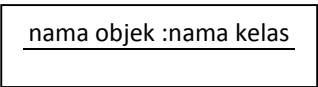


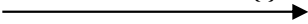
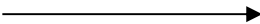


Diagram sekuen menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

Banyaknya diagram *sequence* yang digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalanya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (Yuni Sugiarti, 2013).

Berikut adalah tabel II.4. yang menerangkan simbol-simbol yang ada pada diagram sekuen:

Tabel II.4. Diagram Sequence

Simbol	Deskripsi
Aktor 	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal nama <i>actor</i> .

<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah tahapan yang dilakukan didalamnya.</p>
<p>Pesan tipe create</p> <p style="text-align: center;"><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> <p style="text-align: center;">1 : nama_metode{ }</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe send</p> <p style="text-align: center;">1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe return</p> <p style="text-align: center;">1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe destroy</p> <p style="text-align: center;"><<destroy>></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>.</p>

(Sumber : Rosa A.S, M.Shalahuddin; 2013)

II.6. Microsoft Visual Basic

Visual basic merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi *windows*. *Visual basic 2008* merupakan aplikasi pemrograman yang menggunakan teknologi. *NET Framework 3.5*. Teknologi. *NET Framework 3.5* merupakan komponen *windows* yang terintegrasi serta mendukung pembuatan, penggunaan aplikasi, dan halaman *web*. Teknologi *.Net Framework 3.5* mempunyai 2 komponen utama, yaitu *CLR (Common Language Runtime)* dan *Class Library*. *CLR* digunakan untuk menjalankan aplikasi yang berbasis *NET*, sedangkan *Library* adalah kelas pustaka atau perintah yang digunakan untuk membangun aplikasi (Wahana Komputer, 2010).

II.7. Microsoft SQL Server

Bahasa query merupakan bahasa khusus yang digunakan untuk melakukan manipulasi dan menanyakan pertanyaan (query) yang berhubungan dengan bahasa pemrograman, dimana bahasa query tidak memiliki kemampuan untuk menyelesaikan banyak masalah seperti bahasa pemrograman pada umumnya. Dalam pemrograman basis data, salah satu bahasa yang harus kita kuasai adalah SQL. SQL merupakan bahasa komputer standar yang digunakan untuk berkomunikasi dengan sistem manajemen basis data relasional (RDBMS) (Ema Utami dan Anggi Dwi Hartanto, 2012).