

BAB II

LANDASAN TEORI

2.1 Perusahaan Manufaktur

Perusahaan industri (manufaktur) adalah suatu perusahaan yang aktivitas utamanya adalah membeli bahan baku (*raw material*) kemudian diproses lebih lanjut untuk menjadi barang jadi (*finished goods*) dan kemudian dijual. Bagi perusahaan manufaktur, persediaan mempunyai peranan penting karena persediaan merupakan sumber utama pendapatan dan merealisasikan laba perusahaan [1].

Proses pengendalian manufaktur yaitu pengendalian persediaan bahan baku. Ketika perusahaan menanggung persediaan bahan baku yang berlebihan, mereka mungkin perlu meminjam tambahan dana untuk mendanai persediaan tersebut. Hal ini akan menyebabkan naiknya biaya penyimpanan (*carrying cost*), atau biaya yang dikeluarkan perusahaan dalam memelihara (menyimpan) persediaan [2].

2.2 Akuntansi

Akuntansi disebut sebagai bahasa bisnis karena merupakan suatu alat untuk menyampaikan informasi keuangan kepada pihak-pihak yang memerlukannya. Semakin baik kita mengerti bahasa tersebut, maka semakin baik pula keputusan kita, dan semakin baik kita di dalam mengelola keuangan [3]. Untuk menyampaikan informasi-informasi tersebut, maka digunakanlah laporan akuntansi atau yang dikenal sebagai laporan keuangan. Laporan keuangan suatu perusahaan biasanya terdiri atas empat jenis laporan, yaitu Laporan Posisi Keuangan, Laporan Laba Rugi Komprehensif, Laporan Perubahan Ekuitas, Laporan Arus Kas dan Catatan Atas Laporan Keuangan.

2.2.1 Pengertian Akuntansi

Tujuan akuntansi secara keseluruhan adalah memberikan informasi yang dapat digunakan dalam pengambilan keputusan. Menurut APB Statement No. 4 (Tahun 1970) yang berjudul “*Basic Concepts and Accounting Principles Underlying Financial Statements of Business Enterprises*” akuntansi adalah “sebuah aktivitas jasa, dimana fungsinya adalah memberikan informasi kuantitatif, terutama informasi mengenai keuangan dan entitas ekonomi, yang dimaksudkan akan

menjadi berguna dalam pengambilan keputusan ekonomi (dalam membuat pilihan di antara berbagai alternatif yang ada)".

Menurut *A Statement of Basic Accounting Theory (ASOBAT)* yang diterbitkan oleh *American Accounting Association (AAA)* pada tahun 1966, akuntansi didefinisikan sebagai "proses mengidentifikasi, mengukur dan menyampaikan informasi ekonomi bagi para penggunanya dalam mempertimbangkan berbagai alternatif yang ada dan membuat kesimpulan".

Sedangkan menurut *American Institute of Certified Public Accountants (AICPA)*, akuntansi adalah "seni pencatatan, pengklasifikasian dan pengikhtisaran transaksi dan peristiwa keuangan dengan cara tertentu dan dalam ukuran moneter, termasuk penafsiran atas hasil-hasilnya" [3].

2.2.2 Siklus Akuntansi

Proses atau siklus akuntansi merupakan urutan prosedur yang secara normal digunakan untuk meyakinkan bahwa pengaruh transaksi telah dicatat secara benar. Secara garis besar, siklus akuntansi dibagi menjadi dua tahap, yaitu tahap pencatatan (*recording phase*) dan tahap pengikhtisaran (*summarizing phase*) yang dapat digambarkan seperti di bawah ini [1].

Dalam setiap praktik bisnis, tahap dasar pencatatan adalah sebagai berikut.

1. Analisis setiap transaksi (*transaction identification*) dalam arti pengaruhnya pada akun aktiva, kewajiban, ekuitas, pendapatan dan beban.
2. Masukkan informasi transaksi dalam jurnal (*journalizing*).
3. Pindahkan informasi jurnal ke akun buku besar yang berhubungan (*posting*).
4. Tahap pengikhtisaran mencakup langkah-langkah sebagai berikut.
5. Neraca saldo (*trial balance*); neraca saldo yang belum disesuaikan harus disajikan pada akhir periode atas suatu periode tertentu setelah berbagai ayat jurnal dicatat dalam jurnal dan di-posting ke buku besar (*ledger*), neraca saldo ini berfungsi untuk menguji apakah penjurnalan dan posting ke akun telah dilakukan dengan benar.
6. Penyesuaian (*adjustment*); untuk membuat akun menjadi masa kini (*up-to-date*) untuk penyiapan laporan keuangan, maka diperlukan jurnal penyesuaian.
7. Neraca saldo setelah penyesuaian (*adjusted trial balance*); setelah ayat jurnal penyesuaian ini di-posting ke buku besar, maka akan tampak neraca saldo setelah ayat penyesuaian.

8. Penyusunan laporan keuangan (*financial statement*); data untuk penyusunannya dapat diambil secara langsung dari saldo-saldo akun buku besar yang telah disesuaikan, tetapi bisa juga dengan menggunakan lembar kerja (*worksheet*).
9. Jurnal penutup (*closing entry*); prosedur yang secara umum diikuti untuk mengurangi saldo akun minimal menjadi nol dalam upaya menyiapkan akun untuk periode transaksi di masa mendatang dikenal dengan istilah proses penutupan.
10. Neraca saldo setelah jurnal penutup di-posting (*post closing trial balance*); neraca saldo ketiga telah dapat dibuat setelah jurnal penutup di-posting, menunjukkan bahwa kesamaan debit dan kredit telah di-posting ke akun ikhtisar laba-rugi, neraca saldo setelah jurnal penutup di-posting hanya terdiri atas akun aktiva, kewajiban, kepemilikan (akun riil saja).
11. Jurnal pembalik (*reversing entry*); setelah laporan keuangan telah disajikan dan buku telah ditutup, lazimnya berguna untuk membalik beberapa jurnal penyesuaian sebelum melakukan pencatatan regular pada periode mendatang. Jurnal pembalik dibuat pada permulaan periode akuntansi pada masa mendatang dan secara pasti merupakan kebalikan atas jurnal penyesuaian yang dibuat pada periode sebelumnya.
12. Penggunaan lembar kerja (*worksheet*) untuk menyiapkan laporan keuangan, suatu lembar kerja adalah suatu lembar kertas berkolom yang digunakan untuk menyesuaikan saldo akun dan menyiapkan laporan keuangan.

2.2.3 Basis Akuntansi

Basis akuntansi merupakan himpunan dari standar akuntansi yang menetapkan kapan dampak keuangan dari transaksi harus diakui untuk tuuan pelaporan keuangan. Basis akuntansi ini berhubungan dengan kapan pengukuran dilakukan. Basis akuntansi pada umumnya ada dua, yaitu basis kas dan basis akrual [3].

1. Basis Kas (*Cash Bases*)

Basis kas berarti mengakui dan mencatat transaksi keuangan pada saat kas diterima atau dibayarkan. Basis kas mendasarkan konsepnya pada dua pilar yaitu, pertama berupa pengakuan pendapatan dan kedua berupa pengakuan beban [4].

2. Basis Akrual (*Accrual Bases*)

Basis akrual berarti mengakui dan mencatat transaksi atau kejadian keuangan pada saat terjadi atau saat perolehan. Basis akrual mengakui pengaruh transaksi dan peristiwa lainnya

pada saat transaksi, yaitu ketika transaksi dan peristiwa itu terjadi tanpa memperhatikan saat kas atau setara kas diterima atau dibayar [4].

2.3 Kode Rekening

Istilah akun sering juga disebut dengan istilah perkiraan atau rekening. Biasanya, setiap perusahaan telah menetapkan akun-akun apa saja yang digunakan dalam kegiatan akuntansinya. Daftar akun biasanya disusun atas dasar subklasifikasi dan kelompok sesuai dengan cara penyajian laporan keuangan. Biasanya, akun-akun diklasifikasi menjadi kelompok akun neraca (akun rill) dan kelompok akun laba rugi (akun nominal/akun sementara).

Sebutan akun rill berasal dari fakta bahwa jumlah saldo akun rill pada suatu waktu memang merepresentasikan jumlah rupiah yang benar-benar ada dan mengidentikkan akun tertentu ke suatu waktu tertentu juga. Sebutan akun nominal atau sementara berasal dari fakta bahwa jumlah total akun tersebut pada akhir periode akan ditutup (dipindahkan ke akun modal), sehingga akun nominal tersebut pada awal periode berikutnya tidak mempunyai saldo [3].

Tabel II-1 Kode Rekening [3]

Kelompok Rekening	Kode Rekening	Saldo Normal
Rekening Aktiva	100-199	Debit
Rekening Kewajiban	200-299	Kredit
Rekening Modal	300-399	Kredit
Pendapatan	400-499	Kredit
Beban	500-599	Debit

2.4 Jurnal

Jurnal merupakan catatan yang pertama kali dilakukan. Jurnal adalah suatu catatan yang digunakan untuk mencatat transaksi berdasarkan dokumen perusahaan secara kronologis (menurut tanggal terjadinya transaksi). Tiap transaksi yang telah direkam dalam dokumen akan disalin ke dalam jurnal.

Untuk setiap ayat jurnal, total debit harus sama total kredit. Dengan demikian, tidak ada ayat jurnal yang hanya memuat debit atau kredit saja. Beberapa istilah berkaitan dengan jurnal tersebut adalah sebagai berikut [5].

1. Kolom Tanggal berisi tahun, bulan, dan tanggal transaksi.
2. Kolom Keterangan berisi nama akun dan keterangan transaksi (keterangan dibuat singkat dan jelas). Akun di debit dicatat/dijurnal terlebih dahulu baru dilanjutkan dengan akun di kredit, dan untuk mencatat akun di kredit agak masuk ke dalam dari akun debit.
3. Kolom Ref berisi kode akun, biasanya diisi pada saat memindahkan data dari jurnal ke buku besar (*posting*).
4. Kolom Debit dan Kredit berisi jumlah rupiah yang ada di masing-masing ruas (debit dan kredit).
5. Jika dibutuhkan, juga dibuat kolom tentang “No. Bukti” sebelum kolom “Akun/Keterangan”.

2.5 Buku Besar

Buku besar (*ledger*) adalah suatu catatan yang berisi kumpulan akun-akun yang merupakan suatu kesatuan tersendiri dan saling berhubungan. Pencatatan transaksi dalam buku besar didasarkan pada jenis akun dan dicatat secara kronologis. Ini berlainan dengan pencatatan pada buku jurnal, dimana walaupun sudah dilakukan secara kronologis tapi akunnya masih terpisah-pisah. Dalam buku besar, akun-akun sudah diklasifikasikan atau dikelompokkan sehingga lebih jelas diketahui perubahan setiap akun secara kronologis [6].

Dalam akuntansi terdapat beberapa bentuk buku besar yaitu buku besar bentuk “T”, buku besar bentuk dua kolom, buku besar bentuk empat kolom, dan buku besar bentuk enam kolom. Semua bentuk buku besar tersebut dapat diuraikan sebagai berikut.

1. Buku Besar Bentuk T

Buku besar bentuk T adalah buku besar yang paling sederhana, yaitu berbentuk huruf T serta mempunyai dua ruas kiri dan ruas kanan. Ruas kiri berisi jumlah rupiah debit dan ruas kanan berisi jumlah rupiah kredit [1].

2. Buku Besar Bentuk Dua Kolom

Buku besar dua kolom mempunyai dua bagian yaitu debit dan kredit. Setiap bagian itu mempunyai beberapa kolom tanggal, keterangan, ref, dan jumlah [1].

3. Buku Besar Bentuk Empat Kolom

Buku besar bentuk empat kolom (sering disebut dengan buku besar saldo) tidak hanya terdiri dari empat kolom, karena yang dimaksud dengan empat kolom tersebut adalah kolom-kolom tanggal, debit, kredit, dan saldo. Dalam buku besar ini juga terdapat kolom-kolom keterangan,

ref, dan sering juga dibuat satu kolo lagi yakni “D/K” yang menunjukkan saldo debit atau kredit akun setiap kali terjadi perubahan [1].

2.6 Manajemen Persediaan Bahan Baku

Manajemen persediaan adalah kemampuan suatu perusahaan dalam mengatur dan mengelola setiap kebutuhan barang baik barang baik barang mentah, barang setengah jadi, dan barang jadi agar selalu tersedia baik dalam kondisi pasar yang stabil dan berfluktuasi [4].

Salah satu fungsi utama manajemen persediaan adalah mampu menyediakan produk dalam jumlah yang selalu mencukupi, dan itu semua adalah sebagian dari berbagai tugas lainnya yang harus diemban oleh manajer gudang. Manajer gudang memiliki tanggungjawab seperti itu tidak hanya pada saat musim panen namun juga semua itu harus selalu tersedia pada musim-musim paceklik.

Ada beberapa keuntungan memiliki persediaan yang cukup, yaitu.

1. Adanya kesempatan untuk menjual barang,
2. Biaya pemesanan dapat dikurangi, dan
3. Menjamin kelancaran proses produksi.

2.6.1 Economic Order Quantity (EOQ)

Economic Order Quantity (EOQ) merupakan jumlah pesanan yang memiliki biaya pemesanan dan biaya penyimpanan minimum. Dengan demikian, untuk menentukan jumlah pesanan yang ekonomis perlu dilihat pertambahan biaya pesanan dan biaya penyimpanan serta besarnya persediaan rata-rata. Pembelian (pemesanan) berdasarkan EOQ ini dapat dilaksanakan apabila memakai asumsi-asumsi sebagai berikut [4].

1. Harga pembelian bahan per unit bernilai konstan.
2. Bahan tersebut mudah diperoleh.
3. Jumlah produksi yang menggunakan bahan tersebut stabil, artinya kebutuhan akan bahan mentah tersebut relatif stabil sepanjang tahun.

Secara umum model perhitungan (rumus) EOQ adalah sebagai berikut.

Menentukan jumlah setiap kali pesan

$$Q_o = \sqrt{\frac{2}{H}}$$

Menentukan frekuensi pemesanan

$$n_o = \frac{D}{Q_o}$$

Menentukan total biaya optimum

$$TO(Q) = \frac{D}{Q}H + n_oO$$

Keterangan:

Q_o = nilai EOQ (jumlah unit yang dipesan)

D = demand (permintaan) selama 1 periode waktu

O = biaya per pesanan

T = periode waktu

H = biaya penyimpanan per periode

n_o = banyaknya pesanan selama satu periode

$TO(Q)$ = total biaya yang optimum

2.6.2 Safety Stock

Safety Stock atau persediaan penyelamat adalah persediaan tambahan yang dilakukan untuk melindungi atau mengantisipasi terjadinya kekurangan bahan (*stock out*). *Stock out* mungkin terjadi karena penggunaan bahan baku yang lebih besar dari perkiraan semula atau keterlambatan dalam penerimaan bahan baku yang dipesan [4].

Faktor-faktor yang menentukan jumlah *Safety Stock* adalah penggunaan bahan baku rata-rata, factor waktu atau lead time [4].

1. Penggunaan bahan baku rata-rata

Rata-rata penggunaan bahan baku pada masa sebelumnya perlu diperhatikan karena setelah melakukan pemesanan ulang, permintaan pelanggan sebelum barang yang dipesan datang harus dapat dipenuhi dengan menggunakan persediaan yang ada. Kebutuhan atau permintaan pelanggan biasanya turun naik dan tidak dapat diramalkan dengan pasti. Oleh karena itu, walaupun kita telah meramalkan atau menaksir penggunaan persediaan untuk kebutuhan atau permintaan pelanggan, akan selalu ada resiko yang tidak dapat dihindarkan, yaitu kehabisan persediaan sebelum bahan pesanan datang. Sehingga untuk itu, diperlukan *Safety Stock* untuk menjaga kelancaran kegiatan perusahaan.

2. Faktor waktu atau *Lead Time*

Lead Time adalah selisih atau jeda waktu antara saat dilakukan pemesanan sampai dengan kedatangan barang pesan tersebut di gudang persediaan. Jeda waktu tersebut tidaklah sama antara satu pesanan dengan pesanan lain. Oleh karena itu, dibutuhkan *Safety Stock* untuk mengantisipasi masalah kemacetan produksi yang ditimbulkan keterlambatan bahan pesanan.

Untuk menaksir besarnya *Safety Stock*, dapat dipakai metode sebagai berikut.

$$\text{Safety Stock} = (\text{pemakaian maksimum} - \text{pemakaian rata-rata}) \times \text{Lead Time}$$

2.6.3 Reorder Point (ROP)

Reorder Point atau titik pemesanan kembali adalah waktu minimal untuk melakukan pemesanan ulang sehingga bahan pesanan dapat diterima tepat waktu. Dalam penentuan *Reorder Point* harus memperhatikan faktor-faktor sebagai berikut [4].

1. Penggunaan material selama masa tenggang waktu sebelum barang pesanan datang.
2. Jumlah *Safety Stock*.

Dari faktor di atas, maka *Reorder Point* dapat dicari dengan rumus berikut.

$$\text{Reorder Point} = (\text{Lead Time} \times \text{pemakaian rata-rata}) + \text{Safety Stock}$$

2.7 Bill Of Material (BOM)

Kebutuhan dasar produksi disediakan dengan daftar material (*bill of materials*) dan daftar operasi master. Spesifikasi rinci material untuk produk dicatat pada daftar material. Daftar material berisi daftar semua bahan yang diperlukan dan deskripsinya dalam pesanan subperakitan. Daftar

operasi mirip dengan daftar material, yang berisi operasi tenaga kerja rinci, urutan pengerjaan dan kebutuhan mesin yang berkaitan dengan pengerjaan tersebut. Daftar material dan daftar operasi digunakan secara luas dalam fungsi pengendalian produksi [5].

Ada beberapa manfaat dari adanya *Bill of Material (BOM)*, antara lain.

1. Sebagai alat pengendali produksi yang menspesifikasikan bahan-bahan kandungan yang penting dari suatu produk (bahan-bahan mentah dan komponen), pesanan yang harus digabungkan dan seberapa banyak yang dibutuhkan untuk membuat satu batch
2. Untuk peramalan (*forecasting*) barang yang keluar masuk dan inventori maupun transaksi produksi dan bisa menghasilkan pesanan-pesanan produksi dari pesanan pelanggan
3. Menghitung berapa banyak yang dapat diproduksi berdasarkan segala keterbatasan sumber daya yang ada pada saat kita ini. Apabila sumber daya yang ada tidak mencukupi, sistem dapat menghitung lagi berapa sumber daya yang diperlukan sekaligus membantu dalam proses pengadaan barang. Ketika hendak mendistribusikan hasil produksi, sistem juga dapat menentukan cara pembuatan dan pengangkutan yang optimal kepada tujuan yang ditentukan pelanggan. Dalam proses ini segala aspek yang berhubungan dengan keuangan akan tercatat dalam sistem tersebut termasuk menghitung berapa biaya produksi.
4. Menjamin bahwa jumlah bahan yang tepat telah dikirim ke tempat yang tepat pada waktu yang tepat.

2.8 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pengembangan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis.

Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

2.8.1 Analisis Berorientasi Objek

Analisis berorientasi objek atau *Object Oriented Analysis (OOA)* adalah tahapan untuk menganalisis spesifikasi atau kebutuhan akan system yang akan dibangun dengan konsep

berorientasi objek, apakah benar kebutuhan yang ada dapat diimplementasikan menjadi sebuah sistem berorientasi objek.

2.8.2 Desain Berorientasi Objek

Desain berorientasi objek atau *Object Oriented Design (OOD)* adalah tahapan perantara untuk memetakan spesifikasi atau kebutuhan system yang akan dibangun dengan konsep berorientasi objek ke desain pemodelan agar lebih mudah diimplementasikan dengan pemrograman berorientasi objek.

Pemodelan berorientasi objek biasanya dituangkan dalam dokumentasi perangkat lunak dengan menggunakan perangkat pemodelan berorientasi objek, di antaranya adalah UML (*Unified Modeling Language*).

2.8.3 CASE Tools

CASE (*Computer-Aided Software Engineering*) tools atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak adalah aplikasi atau perangkat lunak yang membantu pembuatan sebuah sistem perangkat lunak. Istilah *CASE tools* muncul karena kebutuhan para pelaku rekayasa perangkat lunak akan perangkat yang memudahkan pekerjaan mereka.

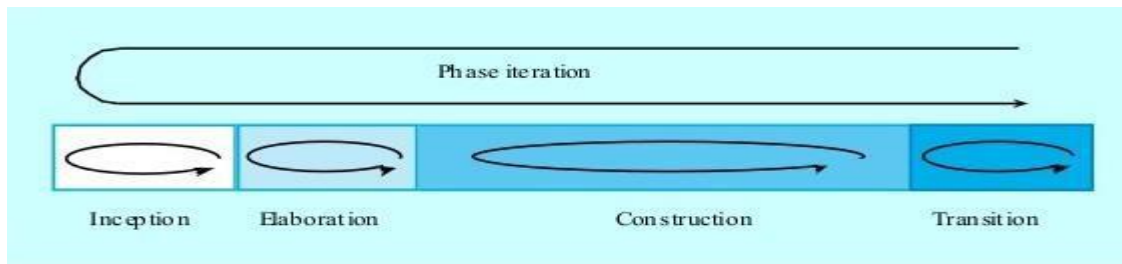
Perangkat lunak yang termasuk *CASE tools* dapat merupakan perangkat lunak dalam tahap analisis, desain, dokumentasi, maupun implementasi ke dalam kode program dan pengujian program. *CASE tools* termasuk editor desain perangkat lunak, *editor* kode program, *compiler*, *debugger*, perangkat pembangunan sistem perangkat lunak, dan lain sebagainya.

2.8.4 Rational Unified Process

Rational Unified Process (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), focus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*).

RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak.

RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara *iterative*. Berikut ini gambar alur hidup RUP.



Gambar II-1 Alur Hidup RUP

[7] Berikut ini penjelasan untuk setiap fase pada RUP.

a. Inception (Permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut adalah tahap yang dibutuhkan pada tahap ini:

- 1) Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya).
- 2) Membangun kasus bisnis yang dibutuhkan.

Jika pada akhir tahap ini target yang diinginkan tidak tercapai maka dapat dibatalkan atau diulang kembali setelah dirancang ulang agar kriteria yang diinginkan dapat tercapai.

b. Elaboration (Perluasan/Perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

Jika pada akhir tahap ini target yang diinginkan tidak dicapai maka dapat dibatalkan atau diulang kembali. Batas/tonggak arsitektur digunakan untuk mendeteksi apakah sebuah kebutuhan akan sistem dapat diimplementasikan atau tidak melalui pembuatan arsitektur.

c. Construction (Konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat

lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

d. Transition (Transisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user.

Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap. Keempat fase pada RUP dijalankan secara berurutan dan iteratif dimana setiap iterasi dapat digunakan untuk memperbaiki iterasi berikutnya.


2.9 Flowchart





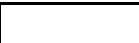
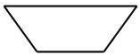

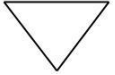

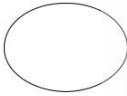

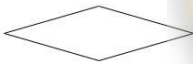
Flowchart atau bagan alir merupakan teknik analitis yang digunakan untuk menjelaskan aspek-aspek Sistem informasi secara jelas, tepat, dan logis. *Flowchart* atau bagan alir menggunakan serangkaian simbol standar untuk menguraikan prosedur pengolahan transaksi yang digunakan oleh sebuah perusahaan, sekaligus menguraikan aliran data dalam sebuah Sistem [8].

Ada beberapa pedoman yang harus diperhatikan dalam membuat *flowchart* , seperti.

1. *Flowchart* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Gunakan symbol-simbol flowchart yang standar.

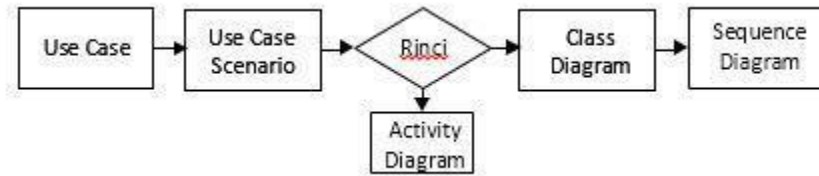
Tabel II-2 Simbol Flowchart [7]

SIMBOL	NAMA	PENJELASAN
	Dokumen	Sebuah dokumen atau laporan, dokumen dapat dibuat dengan tangan atau dicetak oleh komputer.

	Dokumen rangkap	Digambarkan dengan menumpuk simbol dokumen dan pencetakan nomor dokumen di bagian belakang dokumen.
	<i>Input/Output</i>	Digunakan untuk menggambarkan berbagai media <i>Input</i> dan <i>output</i> dalam sebuah bagan alir program.
	Pemasukan data <i>Online</i>	Entri data oleh alat <i>online</i> seperti terminal CRT atau komputer pribadi.
	Tampilan	Informasi ditampilkan oleh alat <i>output online</i> seperti terminal CRT atau memonitor komputer PC.
	Pemrosesan komputer	Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer, biasanya menghasilkan perubahan terhadap data atau informasi.
	Kegiatan manual	Sebuah kegiatan pemrosesan yang dilaksanakan secara manual.
	Penyimpanan <i>Online</i>	Data disimpan sementara dalam <i>File online</i> dalam sebuah media <i>direct access</i> .
	Arsip	Arsip dokumen disimpan dan diambil secara manual.
	Arus dokumen atau pemrosesan	Arah arus dokumen atau pemrosesan, arus normal adalah ke kanan atau ke bawah.
	Penghubung dalam sebuah halaman	Menghubungkan bagan alir pada halaman yang sama. Simbol ini digunakan untuk menghindari terlalu banyak anak panah yang saling melintang dan membingungkan.
	Terminal	Digunakan untuk memulai, mengakhiri, atau titik henti dalam sebuah proses atau program, juga digunakan untuk menunjukkan pihak eksternal.
	Keputusan	Sebuah tahap pembuatan keputusan, digunakan dalam bagan alir program komputer untuk menunjukkan cabang bagi alternatif cara.

2.10 Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman yang berorientasi objek, saat ini UML akan mulai menjadi standar masa depan bagi industri pengembangan sistem/perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan oleh banyak perusahaan rekayasa seperti IBM, Microsoft, dan sebagainya [9].













Gambar II-2 Alur Unified Modeling Language [9]

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Berikut adalah symbol-simbol yang ada pada *use case diagram* [9].

Setelah membuat *use case diagram* yang secara umum belum terperinci oleh karena itu disarankan untuk membuat *use case scenario* yang bertujuan memperinci perilaku sistem yang ada pada *use case diagram*, sementara jika dengan menggunakan *use case scenario* masih belum rinci maka diperlukan *activity diagram* untuk menjelaskan lebih dalam proses perilaku sistem secara general ataupun detail.

Tabel II-3 Simbol Use Case Diagram [7]

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Setelah membuat *use case diagram*, diagram yang selanjutnya yaitu *activity diagram* yang berisi representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Pada pemodelan UML, diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen satu sistem.


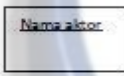



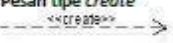
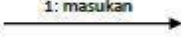
Diagram selanjutnya yaitu *class diagram*, diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka [10]. *Class diagram* juga menunjukkan *property* dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunakan istilah fitur sebagai istilah umum yang meliputi *property* dan operasi sebuah *class*. Berikut adalah symbol-simbol yang ada pada *class diagram*.

Diagram selanjutnya yaitu diagram sequence, diagram ini menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan

diterima antar objek [7]. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sequence yang harus digambarkan adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sequence sehingga makin banyak *use case* yang didefinisikan maka diagram sequence yang harus dibuat juga makin banyak. Berikut adalah symbol-simbol yang ada pada *diagram sequence*.

Tabel II-4 Simbol Sequence Diagram [7]

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>Nama aktor</p> <p>tanpa waktu aktif</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 aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
<p>Garis hidup/ <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p>  <p>Nama objek : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.</p>
<p>Pesan tipe <i>create</i></p>  <p><<create>></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah.</p>
<p>Pesan tipe <i>call</i></p>  <p>1: masukan</p>	<p>Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi atau metode, karena ini memanggil operasi atau metode maka operasi atau metode yang</p>

2.11 Aplikasi Berbasis Web

Aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program computer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu [11]. Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 yaitu :

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Website, adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain. Web adalah sebuah sistem dengan informasi yang disajikan dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah *server web* internet yang disajikan dalam bentuk *hypertext*.

Aplikasi berbasis web merupakan sebuah aplikasi yang diakses menggunakan *web browser* melalui jaringan internet atau intranet. Aplikasi web juga merupakan suatu perangkat lunak komputer yang dikodekan dalam bahasa pemrograman yang mendukung perangkat lunak berbasis web seperti Php dan bahasa pemrograman lainnya dan menggunakan beberapa *database* yang sudah ada untuk mendukung akses Php seperti MySQL [2].

2.11.1 Hypertext Processor (PHP)

Bahasa pemrograman PHP adalah bahasa pemrograman yang bekerja dalam sebuah *web server*. Script-script PHP yang dibuat harus tersimpan dalam sebuah *server* dan dieksekusi atau diproses dalam *server* tersebut. Penggunaan pemrograman PHP memungkinkan sebuah website menjadi lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung website/komputer client akan diolah dan disimpan dalam database *web server* dan bias ditampilkan kembali apabila diakses [2].

2.11.2 My Structured Query Language (MySQL)

My Structured Query Language (MySQL) atau yang biasa dibaca “mai-se-kuel” adalah sebuah program pembuat dan pengelola database atau yang sering disebut dengan *Data Base Management Sistem (DBMS)*, sifat dari DBMS ini adalah *open source*. MySQL merupakan

program pengakses database yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi *multiuser* (banyak pengguna). Kelebihan lain dari MySQL adalah menggunakan bahasa *query* (permintaan) standar *Structured Query Language (SQL)*. SQL adalah bahasa permintaan yang terstruktur, SQL telah distandarkan untuk semua program pengakses database seperti Oracle, PostgreSQL, SQL Server, dan lain-lain. Sebagai sebuah program penghasil database, MySQL tidak mungkin berjalan tanpa adanya sebuah aplikasi pengguna (*interface*) yang berguna sebagai program aplikasi pengakses database yang dihasilkan [12].

2.12 Black Box Testing

Black box testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses testing di bagian luar. Jenis testing ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan [10].