

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem Informasi**

Menurut Kusriani (2010:8), Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dan laporan-laporan yang diperlukan.

Defenisi umum sistem informasi adalah suatu sistem yang terdiri atas rangkaian sub sistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.

##### **II.1.1. Komponen Sistem Informasi**

Menurut Kusriani (2010:9), Dalam suatu sistem informasi terdapat komponen-komponen sebagai berikut :

- a. Perangkat Keras (*hardware*), mencakup sebagai piranti fisik seperti *computer* dan printer.
- b. Perangkat lunak (*Software*) atau program, sekumpulan intruksi yang memungkinkan.
- c. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dalam pembangkitan keluaran yang dikehendaki.
- d. Orang atau semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.

- e. Basis data (*Database*) yaitu sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.
- f. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*Resources*) dipakai secara bersama-sama atau diakses oleh sejumlah pemakai.

## **II.2. Sistem Informasi Akuntansi**

Menurut Kusri (2010:9), Sistem Informasi Akuntansi merupakan sistem informasi yang mengubah data transaksi bisnis yang menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari Sistem Informasi Akuntansi adalah :

- a. Mendukung operasi sehari-hari.
- b. Mendukung pengambilan keputusan manajemen.
- c. Memenuhi kewajiban yang berhubungan dengan pertanggungjawaban.

Komponen-komponen yang terdapat dalam Sistem Informasi Akuntansi adalah sebagai berikut :

- a. Orang-orang yang mengoperasikan sistem tersebut.
- b. Prosedur-prosedur, baik manual maupun terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
- c. Data tentang proses-proses bisnis.
- d. Software yang dipakai untuk memproses data organisasi.
- e. Infrastruktur teknologi informasi.

Didalam organisasi Sistem Informasi Akuntansi berfungsi untuk :

- a. Mengumpulkan dan menyimpan aktivitas yang dilaksanakan disuatu organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut dan para pelaku aktivitas tersebut.
- b. Mengubah data dan informasi yang berguna bagi manajemen.
- c. Menyediakan pengendalian yang memadai.

Sistem Informasi Akuntansi merupakan pendukung aktivitas organisasi.

Yang termasuk pendukung aktivitas organisasi adalah :

- a. Infrastruktur perusahaan, akuntansi, hukum dan administrasi umum
- b. Sumber daya manusia : perekrutan, pengontrolan, pelatihan dan kompensasi kepada pegawai.
- c. Teknologi : Peningkatan produk dan jasa (Penelitian)
- d. Pembelian

Sementara itu aktivitas utamanya adalah :

- a. *Inbount Logistics*, penerimaan, penyimpanan dan distribusi bahan-bahan masukan.
- b. Operasi : aktivitas untuk mengubah masukan menjadi barang atau jasa.
- c. *Outbount Logistics* : distribusi produk kepelanggan.
- d. Pemasaran dan Penjualan.
- e. Pelayanan : Dukungan purna jual *maintenance*.

### II.2.1. Siklus Sistem Informasi Akuntansi

Menurut Kusrini (2010:11), Sistem Informasi Akuntansi memiliki beberapa sistem bagian (*sub system*) yang berupa siklus akuntansi. Siklus akuntansi menunjukkan prosedur akuntansi, mulai dari sumber data sampai ke proses pencatatan/pengolahan akuntansinya. Berikut ini adalah pembagian dari siklus akuntansi.

a. Siklus pendapatan

Siklus pendapatan merupakan prosesur pendapatan yang dimulai dari bagian penjualan otorisasi kredit, pengambilan barang, penerimaan barang, penagihan sampai dengan penerimaan kas

b. Siklus pengeluaran kas

Siklus pengeluaran kas merupakan prosedur pengeluaran kas yang dimulai dari proses pembelian sampai proses pembayaran.

c. Sikluas konversi

Siklus konversi merupakan siklus produksi, dimulai dari bahan mentah sampai barang jadi.

d. Siklus Manajemen Sumber Daya Manusia (SDM)

Siklus Manajemen Sumber Daya Manusia (SDM) merupakan siklus yang melibatkan proses penggajian pada karyawan

e. Siklus buku besar dan laporan keuangan

Siklus ini berupa prosedur pencatatan dan perekaman ke jurnal dan buku besar dan pencetakan laporan keuangan yang datanya diambil dari buku besarnya.

Didalam sebuah Sistem Informasi Akuntansi, tidak semua siklus harus diimplementasikan. Yang wajib ada dalam system tersebut adalah siklus buku besar dan laporan keuangan. Transaksi-transaksi yang termasuk dalam siklus tetapi tidak diimplementasikan, misalnya penggajian, dapat dimasukkan dalam siklus buku besar.

### **II.3. Persediaan**

Menurut Dr. Waluyo (2008:66), Persediaan merupakan barang jadi yang telah diproduksi atau barang dalam penyelesaian, termasuk bahan serta perlengkapan yang akan digunakan dalam proses produksi. Dalam perusahaan dagang persediaan meliputi barang yang dibeli dan disimpan untuk dijual kembali, sedangkan dalam perusahaan jasa, persediaan termasuk biaya jasa seperti upah dan biaya personalia lainnya yang berhubungan langsung dengan pemberian jasa. Dengan demikian pengertian persediaan menurut PSAK(2007) digunakan untuk menyatakan asset yang :

- a. Tersedia untuk dijual dalam usaha normal
- b. Dalam proses produksi atau dalam perjalanan
- c. Dalam bentuk bahan atau perlengkapan untuk digunakan dalam proses produksi atau pemberian jasa.

Persediaan dapat pula dikaitkan dengan hak pemilikan barang sesuai syarat penyerahan pada saat transaksi yang meliputi :

a. Barang dalam perjalanan (*Intransit*)

Pemilikan barang ini sangat tergantung pada syarat penyerahannya, kemungkinan biaya pengangkutan ditanggung pembeli, maka barang tersebut menjadi milik pembelian demikian pula sebaliknya.

b. Barang titipan (barang komisi)

Barang komisi yang belum terjual jelas milik pihak yang menitipkan barang. Ditinjau dari pihak yang menitipkan, barang tersebut sering disebut barang konsinyasi.

#### **II.4. Pengendalian Persediaan**

Menurut Desi Efrianti (2014:100), Dalam sistem pengendalian persediaan pada perusahaan di Indonesia, istilah *Inventory Control* sering diartikan sebagai manajemen persediaan. Oleh karena itu pengendalian persediaan dapat diartikan juga sebagai manajemen persediaan. Adapun pengertian Pengendalian Persediaan itu sendiri adalah kegiatan untuk menentukan tingkat dan komposisi persediaan komponen rakitan (*part*), bahan baku dan barang hasil/produk, sehingga perusahaan dapat melindungi kelancaran produksi dan penjualan serta kebutuhan-kebutuhan pembelajaran perusahaan dengan efektif dan Efisien.

Suatu pengendalian persediaan yang dijalankan oleh perusahaan, tentu mempunyai tujuan-tujuan tertentu. Tujuan pengendalian persediaan secara rinci dapat dinyatakan sebagai berikut :

- a. Menjaga jangan sampai perusahaan kehabisan persediaan sehingga mengakibatkan terhentinya kegiatan produksi.

- b. Menjaga agar pembentukan persediaan oleh perusahaan tidak terlalu besar atau berlebihan, sehingga biaya-biaya yang timbul dari persediaan tidak terlalu besar.
- c. Menjaga agar pembelian kecil-kecilan dapat dihindari karena ini akan memperbesar biaya pemesanan.

Secara umum model persediaan dapat di kelompokkan menjadi dua model, yaitu :

- a. Model pengendalian deterministik

Model Deterministik, yakni model yang menganggap semua variabel telah diketahui dengan pasti. Model ini dapat dibagi menjadi dua karakteristik, yaitu:

1. Deterministik Statis. Di dalam model ini permintaan diketahui dengan pasti atau total permintaan unit pada setiap periode waktu adalah diketahui dan konstan serta laju permintaan adalah sama untuk setiap periode.
2. Deterministik Dinamis. Dalam model ini permintaan untuk setiap periode diketahui dan konstan, tetapi laju permintaan dapat bervariasi dari satu periode ke periode lainnya. Permintaan dalam model persediaan sederhana bersifat deterministik atau pasti, dengan pola yang berubah pada tiap periode dan penyelesaianpun menggunakan teknik-teknik yang berbeda pada tiap inventori. Ketika permintaan bervariasi diperlukan horizon perencanaan untuk menganalisis setiap kegiatan yang dilakukan untuk perkembangan yang lebih lanjut. Pengisian kembali persediaan yang telah habis harus mulai diterapkan dengan dinamis karena program statis tidak

dapat memberikan solusi pada masalah yang diterapkan. Model-model yang dapat digunakan untuk pengendalian persediaan deterministik antara lain: *Production Order Quantity* (POQ), *Quantity Discount*, *Economic Order Quantity* (EOQ), *Just In Time* (JIT) dan *Back Order Inventory*.

b. Model pengendalian probabilistik

Model pengendalian probabilistik digunakan apabila salah satu dari permintaan, *leadtime* atau keduanya tidak dapat diketahui dengan pasti. Suatu hal yang harus diperhatikan dalam model ini adalah adanya kemungkinan *stockout* yang timbul karena pemakaian persediaan bahan baku yang tidak diharapkan atau karena waktu penerimaan yang lebih lama dari *leadtime* yang diharapkan. Untuk menghindari *stockout* perlu diadakan suatu fungsi persediaan pengaman yaitu suatu persediaan tambahan untuk melindungi atau menjaga kemungkinan terjadinya *stockout*. Model ini dapat dibagi menjadi dua karakteristik, yaitu:

1. Probabilistik Statis. Dalam model ini variabel permintaan bersifat random dan distribusi probabilistik dipengaruhi oleh waktu setiap periode.
2. Probabilistik Dinamis. Model ini mirip dengan probabilistik statik dengan pengecualian bahwa distribusi probabilitas permintaan dapat bervariasi dari satu periode ke periode lainnya.

#### **II.4.1. Metode JIT (*Just In Time*)**

Menurut Desi Efrianti (2014:101), kualitas, menekan biaya, dan mencapai waktu penyerahan seefisien mungkin dengan menghapus seluruh jenis pemborosan yang terdapat dalam proses produksi sehingga perusahaan mampu

menyerahkan produknya (baik barang maupun jasa) sesuai kehendak konsumen tepat waktu. Untuk mencapai sasaran dari sistem ini, perusahaan memproduksi hanya sebanyak jumlah yang dibutuhkan atau diminta konsumen, dan pada saat dibutuhkan, sehingga dapat mengurangi biaya pemeliharaan maupun menekan kemungkinan kerusakan atau kerugian akibat menimbun barang.

Sistem ini dirintis oleh *Toyota Motor Corporation* dan dikenal juga dengan *Sistem Produksi Toyota*, yang kemudian dikenal juga dengan istilah Sistem Produksi Ramping (*Lean Production System*) dan sistem *kanban*. Tujuan utama dari JIT adalah menghilangkan pemborosan dan konsisten dalam meningkatkan produktivitas. Oleh karena itu penggunaan istilah JIT seringkali diartikan dengan “*zero inventories*”. JIT pada dasarnya berusaha menghilangkan semua biaya (pemborosan) yang tidak memberikan nilai tambah terhadap produk yang dihasilkan. Untuk mencapai tujuan JIT tersebut, diperlukan asumsi sebagai berikut:

1. Ukuran lot kecil,
2. Konsistensi kualitas tinggi,
3. Pekerja dapat diandalkan,
4. Persediaan menjadi minimum atau sebisa mungkin menjadi nol,
5. Mesin dapat diandalkan,
6. Rencana produksi stabil,
7. Kepastian jadwal operasi,

8. Keseragaman komitmen dan pandangan antara manajemen perusahaan dan karyawan, di mana memiliki komitmen yang tinggi terhadap penerapan JIT yang dilakukan di perusahaan.

JIT mempunyai empat aspek pokok sebagai berikut:

1. Semua aktivitas yang tidak bernilai tambah terhadap produk atau jasa harus dieliminasi. Aktivitas yang tidak bernilai tambah meningkatkan biaya yang tidak perlu misalnya persediaan sedapat mungkin nol.
2. Adanya komitmen untuk selalu meningkatkan mutu yang lebih tinggi, sehingga produk rusak dan cacat sedapat mungkin nol, tidak memerlukan waktu dan biaya untuk pengerjaan kembali produk cacat, dan kepuasan pembeli dapat meningkat.
3. Selalu diupayakan penyempurnaan yang berkesinambungan (*Continuous Improvement*) dalam meningkatkan efisiensi kegiatan.
4. Menekankan pada penyederhanaan aktivitas dan meningkatkan pemahaman terhadap aktivitas yang bernilai tambah.

Menurut Carien Valerie Sakkung (2011), Metode JIT dapat menghilangkan atau mengurangi aktivitas yang tidak bernilai tambah pada produk sehingga proses produksi dapat berjalan lebih efisien. Metode JIT berusaha mendorong biaya pemesanan dan biaya penyimpanan sampai nol atau mendekati nol sehingga total biayanya dapat diefisienkan, mengingat total biaya dapat dihitung dari total biaya pemesanan dan biaya penyimpanan. Biaya penyimpanan tentunya dapat menjadi sangat rendah karena JIT pada dasarnya mengurangi persediaan sampai

pada tingkat yang sangat rendah. Dengan kata lain metode ini mendorong untuk mencapai persediaan sampai pada tingkat nol.

Jika pada metode EOQ biaya pemesanan dilakukan berdasarkan perhitungan EOQ untuk mengetahui tingkat pemesanan yang paling ekonomis untuk mengefisiensikan biaya, pada JIT pemesanan akan dilakukan kembali setelah ada order dari pelanggan. Pada umumnya biaya pemesanan dengan konsep dengan konsep JIT lebih rendah dibandingkan dengan konsep EOQ, karena pemesanan dengan JIT dilakukan berdasarkan permintaan dari pelanggan (*demand pull*) sedangkan pada EOQ pemesanan dilakukan berdasarkan tindakan antisipasi terhadap permintaan di masa datang (*push inventory system*), yang mana permintaan datang dari pangsa pasar yang luas (tidak hanya dari pelanggan).

Jika pada perhitungan total biaya pada PT Leo Otomotif Abadi yang menggunakan cara perhitungan EOQ diperoleh total biaya sebesar Rp 657.267.100,-, tentunya saat menghitung total biaya dengan memakai konsep JIT akan diperoleh total biaya yang lebih rendah sehingga total biaya menjadi lebih efisien. Hal ini dikarenakan cara perhitungan EOQ memperhitungkan biaya-biaya yang terjadi pada persediaan bahan baku yang disimpan digudang sebagai biaya penyimpanan, sedangkan jika memakai prinsip JIT biaya penyimpanan sebisa mungkin menjadi nol karena JIT tidak memperhitungkan adanya persediaan, atau dengan kata lain persediaan dianggap nol atau mendekati nol.

Gambaran perhitungan efisiensi biaya dengan konsep JIT dapat dihitung sebagai berikut:

Apabila biaya pemesanan dengan menggunakan EOQ adalah sebesar Rp 200.000.000,- maka berdasarkan konsep JIT, dimana biaya pemesanan dilakukan berdasarkan permintaan dari pelanggan, diasumsikan biaya pemesanan lebih rendah dari Rp 200.000.000,-, misalnya sebesar Rp 150.000.000,-. Sedangkan biaya penyimpanan diasumsikan mendekati nol atau sama dengan nol maka diperoleh:

$$\begin{aligned}\text{Biaya Total (TC)} &= \text{biaya pemesanan} + \text{biaya penyimpanan} \\ &= \text{Rp } 150.000.000,- + \text{Rp } 0,- \\ &= \text{Rp } 150.000.000,-\end{aligned}$$

Dengan menerapkan JIT, perusahaan diharapkan mendapatkan kesempurnaan dengan berusaha melakukan perbaikan terus-menerus untuk mendapatkan yang terbaik, menghilangkan pemborosan dan ketidakpastian, sehingga diperoleh biaya yang lebih efisien dari sebelumnya dan adanya peningkatan produktivitas.

#### **II.4.2. Metode *Economic Order Quantity***

Menurut Andy Wijaya dkk (2014:15), EOQ adalah jumlah unit (kuantitas) barang yang dapat dibeli dengan biaya minimal. Tujuan metode persediaan ini adalah menentukan jumlah pesanan yang dapat meminimumkan biaya penyimpanan dan biaya pemesanan persediaan. Dengan menggunakan EOQ, maka persediaan yang ada di dalam gudang tidak terlalu banyak, tetapi juga tidak akan terlalu sedikit, sehingga aktivitas perusahaan tidak akan terganggu. Salah satu masalah dalam menentukan analisis EOQ adalah bahwa sulit bagi kita untuk dapat menentukan titik pemesanan kembali.

Perlu diingat bahwa titik pemesanan kembali diperlukan untuk mencegah terjadinya kehabisan/kekurangan stok selama waktu antara melakukan pemesanan dan penerimaan pesanan tersebut. Titik pemesanan kembali adalah suatu tingkat persediaan yang tetap ada dalam stok yang jumlahnya sama dengan permintaan selama masa waktu yang dibutuhkan untuk menerima pesanan (*lead time*). Ketika permintaan bersifat pasti, persediaan ini akan berkurang/dihabiskan pada tingkat yang diketahui, sehingga pesanan akan sampai tepat pada saat tingkat persediaan mencapai titik nol.

$$EOQ = Q^* = \sqrt{\frac{2CR}{H}}$$

Keterangan:

- Q\* = jumlah/nilai EOQ (unit).
- C = biaya pemesanan per pesanan.
- R = permintaan per periode (unit).
- H = biaya penyimpanan

## II.5. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013:4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.


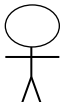


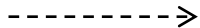
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar

bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

### 1. Use case Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

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

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	<i>Actor</i> atau aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanda panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan didalam <i>use case</i> lain (required) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi




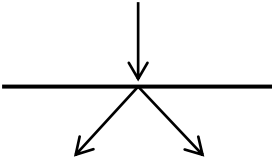
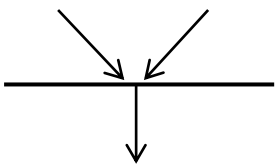
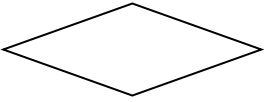

	program.
←-----	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau saran terpenuhi.

(Sumber : Windu Gata ; 2013 : 5)

## 2. Diagram aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

**Tabel II.2. Diagram Aktivitas**

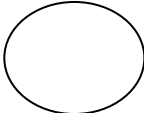
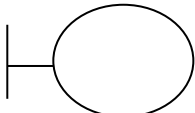
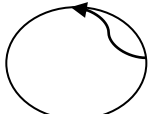

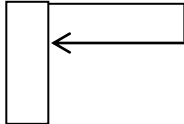


Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i> , akhir aktivitas.
	<i>Activities</i> , menggambarkan suatu proses atau kegiatan bisnis.
	<i>Fork</i> (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>Rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence* diagram, yaitu :

**Tabel II.3. Diagram Urutan**

Gambar	Keterangan
	<i>Entity class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan form <i>entry</i> dan form cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek kontrol objek mengkoordinir pesan antara <i>boundary</i> dan entitas.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata ; 2013 : 7)

#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

**Tabel II.4. Tabel Diagram Kelas**

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara . contoh : 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 9)

## II.6. Pengertian Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulan-kumpulan *file* yang saling berkaitan.

Menurut Wahana Komputer (2010:140), Database diartikan sebagai representasi fakta dunia nyata yang mewakili sebuah objek, misalnya manusia, hewan, barang, peristiwa, konsep dan lain sebagainya yang direkam dalam bentuk huruf, teks, symbol, angka, suara, gambar dan lainnya. Sedangkan basis data dapat diartikan sebagai tempat berkumpul, sarang atau gudang untuk menyimpan sesuatu. Dengan demikian basis data/database dapat diartikan sebagai tempat berkumpul, menyimpan data-data suatu benda atau kejadian yang saling berhubungan.

Menurut Kusri (2010:2), pengertian Basis Data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau symbol).

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.

3. Kumpulan *file/tabel/arsip* yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

### II.6.1. Tujuan Basis Data

Menurut Kusrini (2010:2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basisdata yang baik adalah sebagai berikut :

- a. Tidak adanya redudansi dan inkonsistensi data

Redudansis terjadi jika suatu informasi disimpan di beberapa tempat. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

- b. Kesulitan pengaksesan data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan query ataupun dari tool yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari software DBMNnya.

- c. *Multiple user*

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian server yang bisa diakses dari banyak *client*, sudah menyediakan akses kesemua pengguna dari komputer *client* ke sumber informasi yaitu basis data.

## II.6.2. Manfaat/Kelebihan Basis Data

Menurut Kusri (2010:5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

a. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan matematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

b. Kebersamaan pemakai (*sharability*)

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencacatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama.

c. Pemusatan kontrol data

Karena cukup satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

d. Efisiensi ruang penyimpanan

Dengan pemakaian bersama, tidak perlu menyediakan tempat penyimpanan diberbagai tempat tetapi cukup satu saja, sehingga ini dapat menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

e. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain, dapat menekan ketidakakuratan dalam pemasukan/penyimpanan data.

f. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat dibackup, memilah-milah data mana yang masih diperlukan yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi sebuah organisasi dari lain waktu ke waktu membutuhkan penyimpanan yang semakin besar.

g. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan password untuk membatasi orang yang diaksesnya.

h. Kemudahan dalam pembuatan program aplikasi baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS. Sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

i. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan tools yang disediakan oleh DBMS.

j. Kebebasan data

Perubahan dapat dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

k. *User View*

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

## II.7. Normalisasi

Menurut Samiaji Sarosa (2010:5), Normalisasi adalah teknik yang dirancang untuk merancang tabel basis data relasional untuk meminimalkan duplikasi data dan menghindarkan basis data tersebut anomali. Suatu basis data dikatakan tidak normal jika terjadi 3 (tiga) anomali berikut :

a. *Insertion Anomaly*

Anomali yang terjadi jika ada data yang tidak bisa disisipkan kedalam tabel.

b. *Update/Modification anomaly*

Anomali yang terjadi jika ada perubahan pada suatu item data maka harus mengubah lebih dari satu baris data.

Langkah-langkah normalisasi sampai pada bentuk 3NF adalah sebagai berikut :

a. *First Normal Form (1NF)*

Untuk menjadi 1NF suatu table harus memenuhi dua syarat. Syarat pertama tidak ada kelompok data atau *field* yang berulang. Syarat kedua harus ada *primary key (PK)* atau kunci unik, atau kunci yang membedakan satu bari

dengan baris yang lain dalam satu table. Pada dasarnya sebuah table selamat tidak ada kolom yang sama merupakan bentuk table dengan 1NF.

*b. Second Normal Form (2NF)*

Untuk menjadi 2NF suatu table harus berada dalam kondisi 1NF dan tidak memiliki *partial dependencies*. *Partial dependencies* adalah suatu kondisi jika atribut non kunci (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

*c. Third Normal Form (3NF)*

Untuk menjadi 3NF suatu table harus berada dalam kondisi 2NF dan tidak memiliki *transitive dependencies*. *Transitive dependencies* adalah suatu kondisi dengan adanya ketergantungan fungsional antara 2 atau lebih atribut non kunci (Non PK).

## **II.8. Visual Basic 2010**

Menurut Edi Winarno dkk (2010:1), *Visual Basic* adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh *programmer* didunia. Pemograman ini dipakai oleh jutaan *programmer* dan tercatat sebagai program yang paling disukai oleh mayoritas orang.

*Visual Studio 2010* pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual Studio 2010* selain disebut dengan bahasa pemrograman,

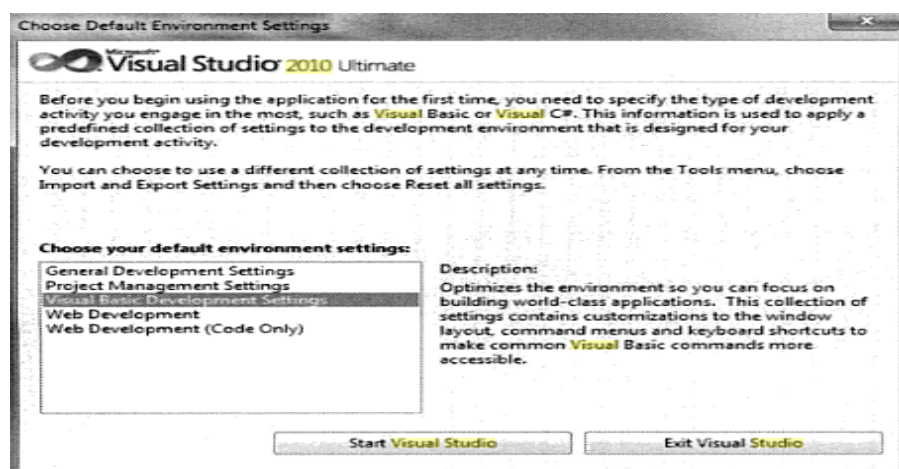
juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis *windows*.

Beberapa kemampuan atau manfaat dari *Visual Studio 2010* diantaranya seperti :

1. Untuk membuat program aplikasi berbasis *windows*.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol *ActiveX*, *file Help*, aplikasi Internet dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program berakhiran EXE yang bersifat *executable* atau dapat langsung dijalankan.

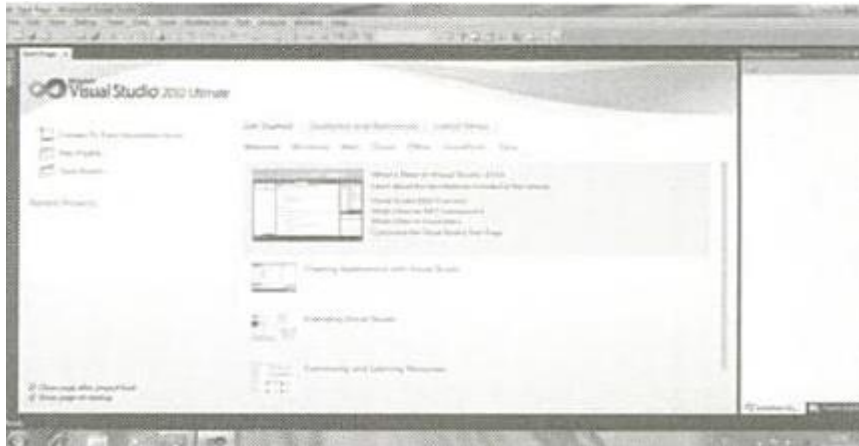
### II.8.1. Antar Muka *Visual Basic 2010*

Saat menjalankan *Visual Basic 2010* (Edi Winarno dkk, 2010:1) pertama kali muncul jendela *chose default environment settings*. Disini bisa memilih apakah ingin memilih antar muka di *Visual Studio*. Untuk *programmer* Visual Basic lebih baik memilih *Visual Basic Development Centre*.



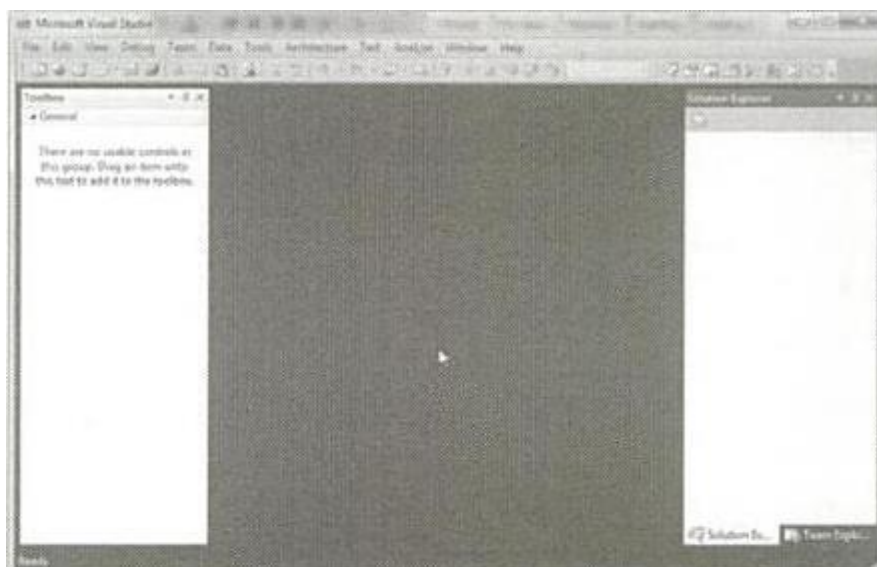
Gambar II.1. Form *Chose Default Environment Settings*

Dibagian awal visual basic, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup*.



**Gambar II.2. Start Page Visual Basic 2010**

Jika *start page* ditutup terlihat tampilan sebagai berikut :



**Gambar II.3. Tampilan IDE (Integrated Development Environment) setelah Start Page ditutup**



## **II.9. *SQL Server 2008 Express Edition***

Menurut Wahana Komputer (2010:2), *SQL Server 2008 Express Edition* sebuah terobosan baru dalam bidang database, *SQL Server* adalah sebuah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti *IBM* dan *Oracle*. *SQL Server 2008 Express Edition* dibuat pada saat kemajuan dalam bidang *hardware* semakin pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008 Express Edition* membawa terobosan dalam bidang pengolahan dan penyimpanan data.

### **II.9.1 Kebutuhan *Hardware***

Adapun *hardware* yang diperlukan untuk instalasi *SQL Server 2008 Express Edition* minimal adalah sebagai berikut :

- a. *Processor* minimal 1 GHz
- b. Memori minimal 512 MB
- c. Sistem Operasi *Windows*

Biar dapat diinstal pada sistem komputer dengan memori 512 MB, tetapi disarankan menggunakan memori 1 GB. Sedangkan untuk jaringannya diperlukan adalah :

- a. *Sharer Memory*.
- b. TCP/IP.
- c. *Named Pipes*.
- d. *Virtual Interface Adapter (VIA)*.