

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Tabel II.1. Penelitian Terkait

No	Nama	Judul	Hasil Penelitian	Perbedaan Penelitian
1	Priyo Sutopo (2016)	Sistem Informasi Eksekutif Sebaran Penjualan Kendaraan Bermotor Roda 2 Di Kalimantan Timur Berbasis Web	Tujuan penelitian ini adalah untuk mengatasi kesulitan dalam melihat penyajian data penjualan kendaraan bermotor atau grafik penjualan kendaraan, data yang di berikan untuk eksekutif hanyalah laporan manual dan mungkin sangat sulit dipahami, sehingga tidak efektif dan efisien, sehingga dibutuhkan sebuah aplikasi untuk membantu eksekutif dalam melihat penyajian data penjualan kendaraan bermotor dan grafik penjualan kendaraan. Memudahkan dalam pengambilan keputusan yang strategis seperti	Pada penelitian sebelumnya diterapkan sistem informasi eksekutif dalam penjualan kendaraan bermotor guna mencapai profit yang lebih tinggi dibanding periode sebelumnya sedangkan penelitian yang dilakukan oleh penulis untuk mengatasi proses over produksi yang terjadi tiap bulan dan dapat mengatasi kerugian akan produksi permen.

			<p>pengurangan dan penambahan target penjualan dan pengadaan produk per wilayah secara cepat dan tepat, karena tersedianya data yang aktual dan dapat di gunakan setiap saat.</p>	
2	Fauzan Aziz(2015)	<p>Sistem Informasi Eksekutif Berbasis Web Pada Fakultas Pertanian Universitas Muhammadiyah Palembang</p>	<p>Penelitian yang dilakukan oleh Fauzan Aziz pada tahun 2015 adalah SIE ini dapat memberikan informasi ringkas dan mudah dipahami dekan Fakultas Pertanian UMP dengan tampilan dashboard berupa grafik. Mempermudah dekan dalam mengambil keputusan jangka panjang. Sehingga dapat mempengaruhi kinerja dari bagian kepegawaian tersebut. Sistem ini akan mendukung eksekutif dalam pengambilan keputusan serta merencanakan strategi selanjutnya karena informasi yang disajikan nantinya lebih cepat, mudah, ringkas, interaktif dan keluar dari pendekatan trad</p>	<p>Pada penelitian sebelumnya sistem informasi eksekutif diterapkan pada kinerja pegawai dalam proses belajar mengajar sedangkan penelitian yang dilakukan oleh penulis sistem informasi eksekutif diterapkan pada proses produksi permen.</p>

3	Fatmasari (2017)	Sistem Informasi Eksekutif Penjualan Pada Rku Komputer	Tujuan dari penelitian ini adalah menganalisis dan merancang sistem informasi eksekutif pada RKU Komputer sehingga mempercepat proses pengelolaan pelaporan dari bagian Administrasi. Hasil dari penelitian ini adalah sistem informasi eksekutif yang diharapkan dapat membantu dan mempermudah kerja pimpinan dalam menganalisis laporan-laporan yang disajikan secara tekstual dan grafik pada RKU Komputer secara cepat, tepat dan akurat. Dengan penggunaan sistem informasi eksekutif yang berbasis komputer, maka dapat meningkatkan keefektifan dan kemudahan pimpinan dalam pengambilan keputusan bagi kemajuan RKU Komputer dalam bidang penjualan.	Pada penelitian sebelumnya membahas tentang penjualan computer berdasarkan permintaan konsumen tiap hari, fluktuasi perkembangan penjualan computer diterapkan dengan menggunakan aplikasi sederhana sedangkan penelitian yang dilakukan oleh penulis menghitung jumlah produksi permen tiap periodenya dengan menggunakan aplikasi visual studio 2010 dan Sql server 2008 dan dapat membantu perusahaan dalam melakukan perhitungan produksi permen.
4	Intan Putri Komara (2017)	Sistem Informasi Eksekutif PT. Pln (Persero) Sumedang Rayon Tanjungsari	Penelitian yang dilakukan oleh Intan Putri Komara dapat memberikan informasi mengenai jumlah pelanggan yang mendaftar dan menggunakan	Penelitian intan putri komara membahas tentang jumlah tagihan listrik pelanggan tiap periode dan mengefisiensikan

			<p>berdasarkan kategori tertentu. Informasi jumlah pelanggan yang ada pada sistem informasi eksekutif PT. PLN (Persero) Sumedang Rayon Tanjungsari meliputi jumlah pelanggan berdasarkan daerah setiap tahun, jumlah pelanggan berdasarkan periode setiap tahun, jumlah pelanggan berdasarkan daya setiap tahun, jumlah pelanggan berdasarkan pembayaran setiap tahun dan jumlah pemasangan listrik setiap tahunnya, jumlah pelanggan berdasarkan daerah dengan kategori tertentu setiap bulan.</p>	<p>kegiatan operasional masyarakat. Sedangkan penelitian yang dirancang oleh penulis berdasarkan Produksi Permen Pada PT. Union Confectionary.</p>
--	--	--	---	--

II.2. Uraian Teoritis

II.2.1. Sistem

Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*) yaitu sistem komputer yang terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar, dan kemudian subsistem-subsistem tersebut akan berinteraksi

sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu. (Deppi Linda : 2016)

II.2.2. Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

1. Akurat (*Accurate*)

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*Timelines*)

Berarti informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai logika karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan (*Relevance*)

Berarti informasi tersebut bermanfaat bagi pemakainya (Deppi Linda, 2016 : 62-63).

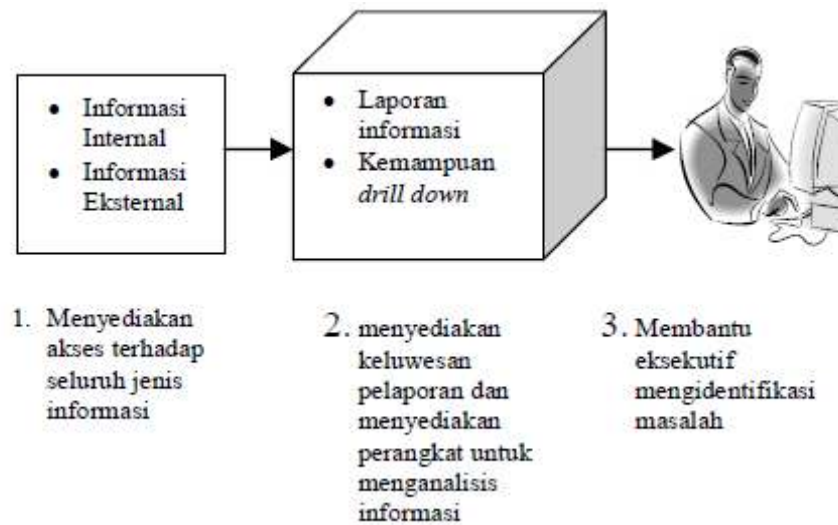
II.2.3. Sistem Informasi

Sistem informasi adalah sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Penggunaan sistem informasi telah banyak diterapkan diberbagai bidang

termasuk dalam bisnis. Salah satu tujuan penerapan sistem informasi dalam bidang bisnis agar dapat meningkatkan keuntungan bisnis dengan menggunakan kemampuan yang didapatkan dari sistem informasi. Ada beberapa kemampuan dari sistem informasi yang dapat mendukung dalam bidang bisnis. Kemampuan tersebut seperti pengurangan biaya, mempercepat pekerjaan, dapat meningkatkan kemudahan dalam pengambilan keputusan, dan peningkatan pelayanan terhadap pelanggan. (Alfian Nurlifa : 2017 : 19)

II.2.4. Sistem Informasi Eksekutif

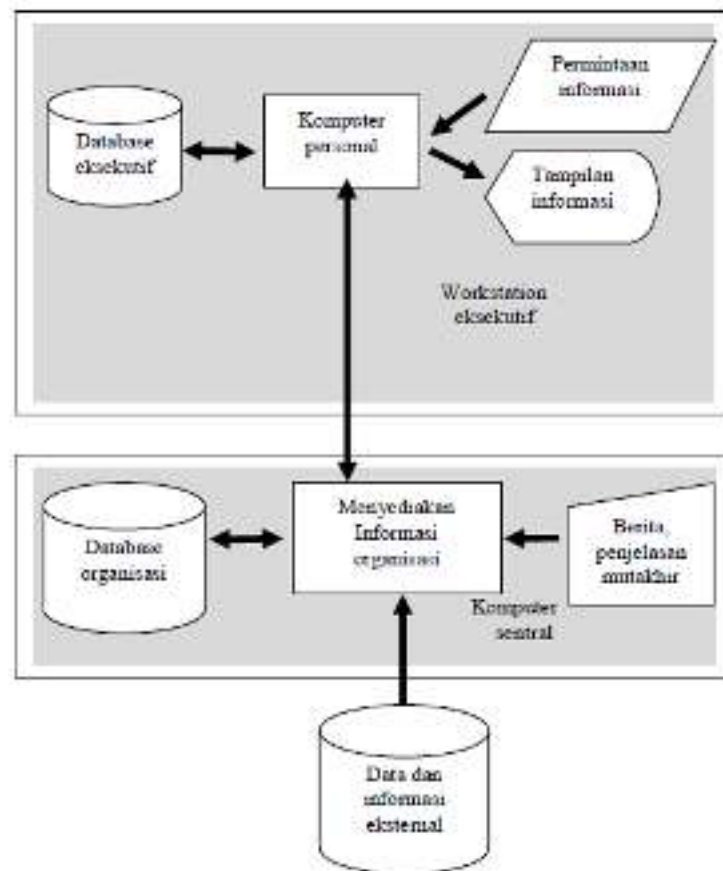
Sistem Informasi Eksekutif (EIS) merupakan perpaduan sistem – sistem terkomputerisasi yang menyediakan informasi internal maupun eksternal sehingga sangat berguna dan bermanfaat bagi para eksekutif dalam proses pengambilan keputusan. Mengirimkan, menganalisis dan menyajikan informasi pada station kerja para mengambil keputusan yang memberikan gambaran jelas kepadanya mengenai standar penting serta kejadian-kejadian, sebelum terlambat menanganinya. Kemampuan *drill down* yang tersedia pada sistem ini memungkinkan eksekutif dapat melihat lebih rinci suatu informasi. *Driil down* berarti eksekutif dapat memulai dari gambaran sekilas dan kemudian secara bertahap mengambil informasi yang lebih terinci. (Irfan capelli G : 2017)



Gambar II.1 Sistem Informasi Eksekutif

(Sumber : Irfan Capelli G : 2017)

Gambaran model dari Sistem Informasi Eksekutif dapat dilihat pada gambar II.2 menunjukkan bahwa sistem informasi eksekutif melakukan dialog dengan perangkat lunak sistem dengan memasukkan instruksi kedalam sistem melalui menu. Pemilihan menu dilakukan dengan *mouse*. Penggunaan *keyboard* dikurangi. Sistem Informasi Eksekutif memantau seberapa baik organisasi berjalan dalam mencapai tujuan sesuai dengan visi, misi, tujuan dan sasaran yang ada. (Irfan Capelli G : 2017)



Gambar II.2 Model Sistem Informasi Eksekutif

(Sumber : Irfan Capelli G : 2017)

II.2.5. Karakteristik dan Sifat SIE (Sistem Informasi Eksekutif)

Adapun Karakteristik dan sifat Sistem Informasi Eksekutif (EIS) adalah sebagai berikut:

1. Sesuai dengan pemakai eksekutif perorangan.
2. Menyediakan akses status secara *on-line* dan *drilldown*.
3. Sangat *user-friendly* dan hanya membutuhkan sedikit pelatihan atau bahkan tidak sama sekali dalam penggunaannya.
4. Digunakan oleh eksekutif secara langsung tanpa perantara atau operator.

5. Menampilkan informasi berupa grafik, tabel maupun tekstual. (Irfan Capelli G : 2017)

II.2.6. Database

Pangkalan data atau basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi. Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Model yang umum digunakan sekarang adalah model relasional, yang mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel (Neni Purwati, 2015 : 50).

II.2.7. Normalisasi

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal. Bentuk data yang tidak normal / data mentah biasa disebut juga *unnormalized form*. Masing – masing level normalisasi mempunyai aturan tersendiri. (Ema Utami ; 2012 : 73-76)

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data / database, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redundansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data. (Mukhlisulfatih Latief ; 2016 : 234 - 236)

1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

2. Tahapan Normalisasi :

- 1) Bentuk tidak normal : Menghilangkan perulangan grup.

Tabel II.1. Contoh bentuk tidak normal (Unnormal)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
			M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
			Akn201	Akuntansi	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

(Sumber : Mukhlisulfatih Latief : 2016)

- 2) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

Tabel II.2. Contoh Bentuk Normal Pertama (1NF)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
2683	Welli	MI	M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
5432	Bakti	AK	Akn201	Akuntansi	D310	Lia	B
5432	Bakti	AK	MKT300	Dasar Pemasaran	B212	Lola	A

(Sumber : Mukhlisulfatih Latief : 2016)

3) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan key sudah tergantung penuh terhadap key-nya.

Tabel II.3. Contoh Bentuk Normal Kedua (2NF)

Kode-MK	Nama-MK	Kode Dosen	Nama Dosen
M1350	Manajemen DB	B104	Ati
M1465	Analisis Perc. Sistem	B317	Dita
M1350	Manajemen DV	B104	Ati
Akn201	Akuntansi	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

(Sumber : Mukhlisulfatih Latief : 2016)

4) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap key-nya.

Tabel II.4. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

(Sumber : Mukhlisulfatih Latief : 2016)

II.2.8. Kamus Data

Kamus data umumnya berguna pada pengembangan model sistem dan dapat digunakan untuk menangani semua informasi dari semua tipe model sistem. Kamus data sederhanya adalah daftar alfabetis dari nama-nama termasuk pada berbagai model sistem. Seperti namanya, kamus harus mencakup deskripsi yang berhubungan dengan entitas bernama tersebut dan, jika nama itu merepresentasikan objek komposit, mungkin saja ada deskripsi mengenai komposisinya. Informasi lain seperti tanggal pembuatan, pembuatnya dan representasi entitas juga dapat dimasukkan, tergantung pada tipe model yang sedang dikembangkan. Keuntungan penggunaan kamus data adalah :

1. Kamus data merupakan mekanisme untuk manajemen nama. Banyak orang yang harus menciptakan nama untuk entitas dan relasi ketika mengembangkan model sistem yang besar. Nama-nama ini harus dipakai secara konsisten dan tidak boleh bentrok. Perangkat lunak kamus data dapat memeriksa keunikan nama dan memberitahu analisis persyaratan sekiranya terjadi duplikasi nama.
2. Kamus data berfungsi sebagai tempat penyimpanan informasi organisasional yang dapat menghubungkan analisis, desain, implementasi dan evolusi. Sementara sistem dikembangkan, informasi diambil untuk memberitahu perkembangan. Informasi baru ditambahkan pada sistem. Semua informasi mengenai entitas berada pada satu tempat (Ian Sommerville ; 2012 : 151).

Tabel II.5. Entri Kamus Data

Nama	Keterangan	Tipe	Tanggal
Has-labels	Relasi 1 N antara entitas bertipe node atau link dan entitas bertipe label	Relasi	5.10.1998
Label	Berisi informasi terstruktur atau tidak terstruktur mengenai node atau link. Label dipresentasikan oleh ikon (yang bisa berupa kotak transparan) dan teks yang berhubungan	Entitas	8.10.1998
Link	Relasi 1 : 1 antara entitas desain yang dipresentasikan sebagai node. Link diketikkan dan dapat diberi nama	Relasi	8.10.1998
Name (Label)	Setiap label memiliki nama yang mengidentifikasi tipe label. Nama harus unik dalam set tipe label yang dipakai pada desain	Atribut	8.10.1998
Name (Node)	Setiap node harus memiliki nama yang unik di dalam sebuah desain. Panjang nama bisa mencapai 64 karakter.	Atribut	15.10.1998

(Sumber : Ian Sommerville ; 2012 : 151)

II.2.9. Visual Studio 2010

Aplikasi yang digunakan untuk melakukan perancangan sistem informasi akuntansi ini menggunakan aplikasi *visual studio* 2008. VB.NET adalah salah satu bahasa pemrograman yang paling mudah dimengerti dan dipahami karena desain user yang tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada *platform .NET* yang diluncurkan tahun 2002 dan untuk menjembatani *programmer Visual Basic*. Bahasa VB.NET secara teknis mengadopsi sintak bahasa *Visual Basic*. Konsistensi API membuat bahasa VB.NET menjadi pilihan

dalam membuat kode program diatas *platform Windows*. Fitur baru bahasa *VB.NET* dibandingkan *Visual Basic* bahwa bahasa *VB.NET* mendukung *object-oriented* dan juga *dynamics programming*. Ini menambah daftar kemudahan untuk belajar bahasa *VB.NET* (Agus Kurniawan ; 2013 : 10).

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasi lainnya dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*. *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). *Visual Studio* kini telah menginjak versi *Visual Studio 12.0* atau dikenal dengan sebutan *Microsoft Visual Studio 2013* yang diluncurkan pada 17 Oktober 2013 yang ditujukan untuk *platform Microsoft .NET Framework 4.5.1*. (Ahmad Rasi Ruli; 2017 : 10)

II.2.10. SQL Server 2008

SQL Server 2008 merupakan *RDBMS (Relational Database Management System)* yang bersifat *powerful* dan telah terbukti kekuatannya

dalam mengolah data. *SQL Server 2008* juga memiliki banyak fitur yang bisa diandalkan untuk meningkatkan *performa database*. *SQL Server 2008* memiliki suatu GUI (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari-hari berkaitan dengan *database*, seperti menulis T-SQL, melakukan *backup* dan *restore database*, melakukan *security database* terhadap aplikasi, dan sebagainya. Pada GUI (*Graphic User Interface*) tersebut kita bisa melakukan *settingan* terhadap *SQL Server* untuk berkerja lebih optimal. *Settingan* juga bisa dilakukan menggunakan *script* untuk memudahkan *developer* mengubah *Setting Options* pada *SQL Server 2008*. (Ruslan; 2013 : 9)

II.2.11. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak sebuah sistem. UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem, karena tergolong bahasa visual yang lebih mudah dan lebih cepat dipahami dibandingkan dengan bahasa pemrograman. *Unified Modelling Language* (UML) biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.

3. Menggambarkan representasi struktur static sebuah sistem dalam bentuk *class diagram*.
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagrams* UML.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *fungsi* dengan *stereo types*.


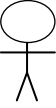


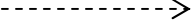
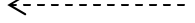
Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Saipul Anwar, et al., 2016 : 75-76).

II.2.11.1. Use Case Diagram

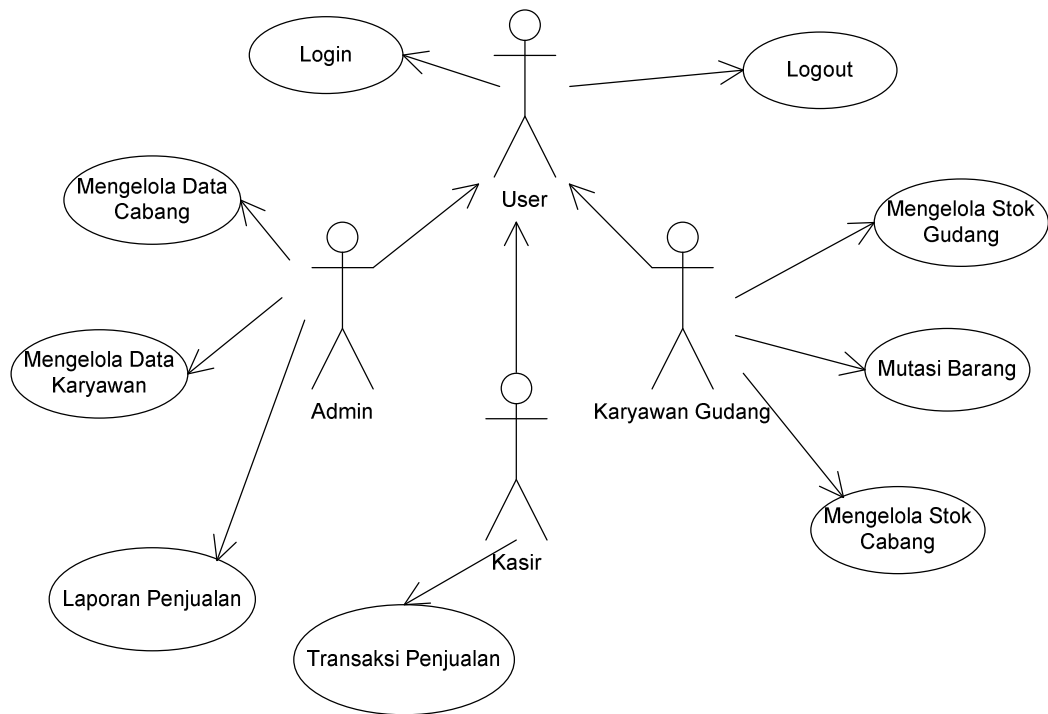
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2016 : 108).

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.6 dibawah ini :

Tabel II.6. 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> .
	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 control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) 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 syarat terpenuhi.

(Sumber : Sumber : Ade Hendini, 2016)



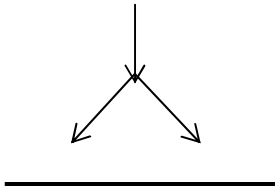
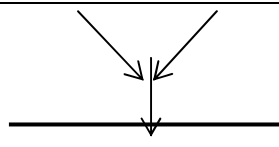
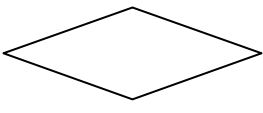
Gambar II.3. Use Case Diagram
(Sumber : Ade Hendini, 2016 : 112)

II.2.11.2. Activity Diagram

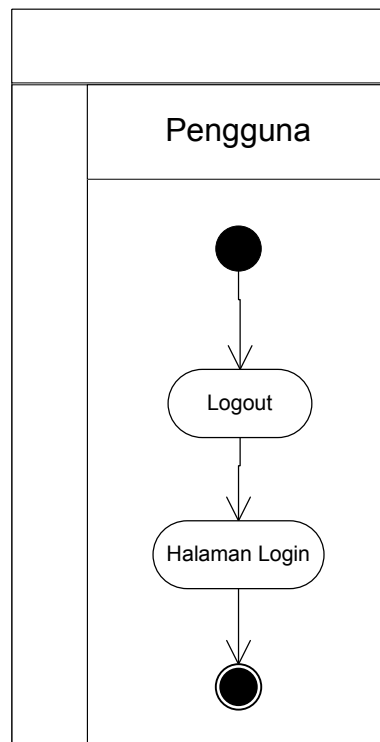
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.7 dibawah ini:

Tabel II.7. Simbol Activity Diagram

Gambar	Keterangan
●	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
●	<i>End point</i> , akhir aktifitas.
□	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.

	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
<div style="border: 2px solid black; padding: 2px; display: inline-block;">New Swimlane</div>	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

(Sumber : Ade Hendini, 2016 : 109)

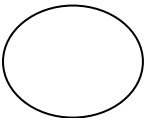
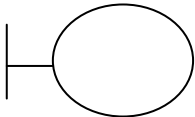
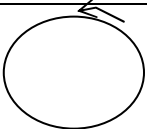

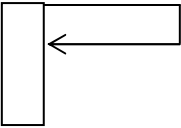




Gambar II.4. Activity Diagram
(Sumber : Ade Hendini, 2016 : 112)

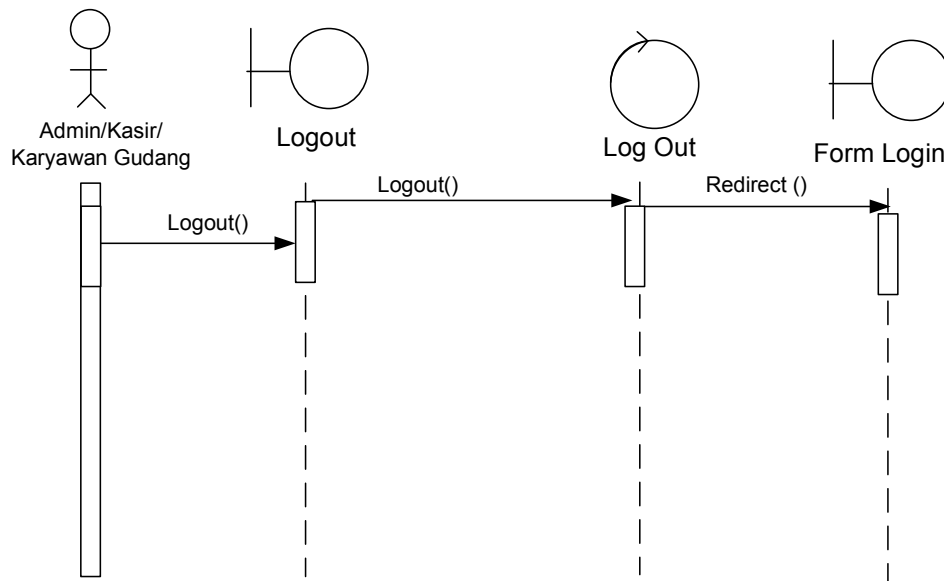
II.2.11.3. Sequence Diagram

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* dapat dilihat pada tabel II.8 dibawah ini :

Tabel II.8. Simbol *Sequence Diagram*

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 aktor dengan sistem, seperti tampilan formentry dan <i>form</i> 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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

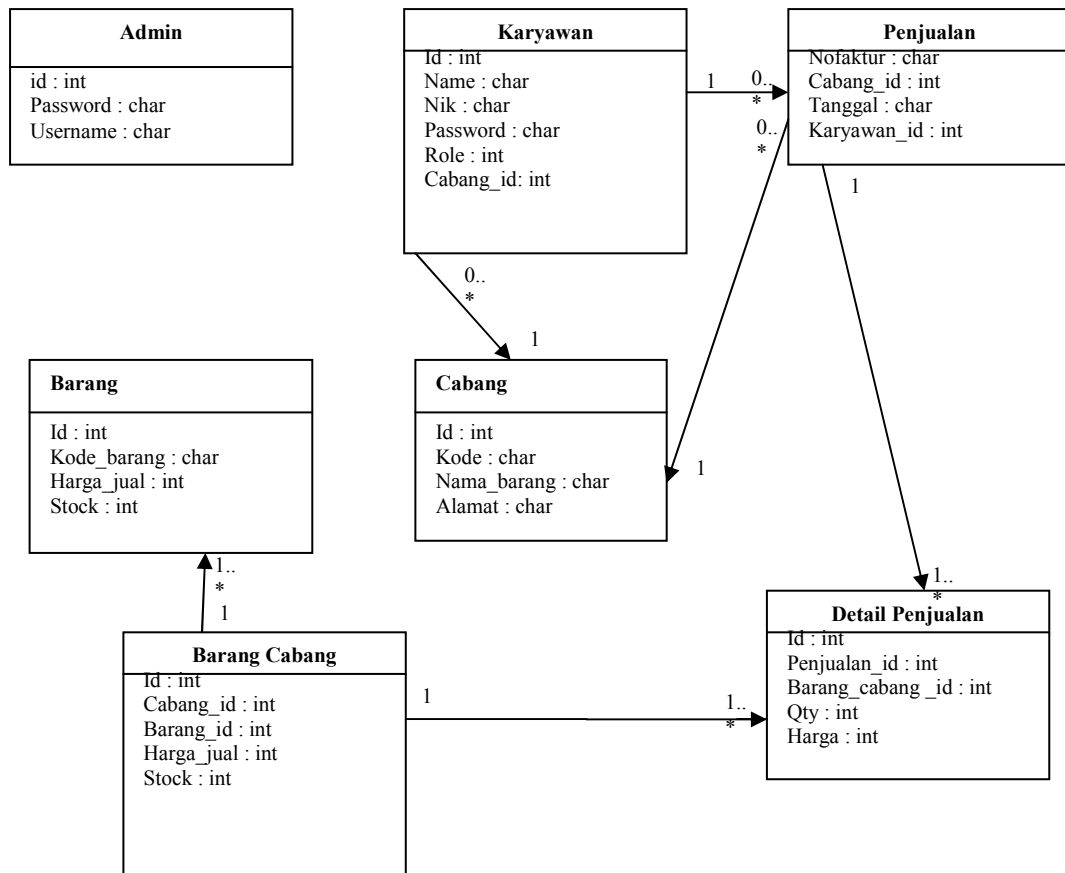
(Sumber : Ade Hendini, 2016 : 110)



Gambar II.5. Sequence Diagram
(Sumber : Ade Hendini, 2016 : 114)

II.2.11.4. Class Diagram

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 (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016 : 111).



Gambar II.6. Class Diagram
(Sumber : Ade Hendini, 2016 : 115)