

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

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

Berdasarkan penelitian dari Setiawan, Deni Tri. (2017)., Peneliti mengambil judul Perancangan Sistem Informasi Penggajian Guru Menggunakan Pemerograman Delphi 7. Studi kasus yang dipilih yaitu di SMA N 2 Kota Agung. Hasil dari penelitian ini adalah pengembangan sistem dengan menggunakan bahasa pemrograman borland delphi 7. yang pada kesimpulannya agar proses penggajian guru lebih cepat sehingga mengurangi kesalahan dalam menginput data dan proses pembayarannya lebih efektif dan efisien. Karena proses penggajian guru pada SMA N 2 Kotaagung masih manual, sehingga masih sering terjadi kesalahan pengolahan data dan keterlambatan pembayaran gaji, hal ini memerlukan adanya upaya-upaya untuk mengembangkan rancangan sistem agar proses penggajian tidak selalu menimbulkan masalah.

Berdasarkan penelitian yang dilakukan oleh Irawan, Muhammad Dedi. (2017), peneliti mengambil judul Sistem Penggajian Karyawan Pada LKP Grace Education Center. Hasil dari penilitian ini adalah membangun sistem penggajian karyawan pada LKP Grace Education Center. Hasil dari penelitian ini adalah adanya aplikasi yang dapat memudahkan dalam pembuatan informasi yang dibutuhkan diantaranya Laporan Pegawai, Laporan Penggajian dan Slip Gaji

secara cepat dan mengurangi tingkat kesalahan pemahaman informasi Penggajian yang berlangsung di LKP Grace Education Center.

Berdasarkan penelitian yang dilakukan oleh Dani, AAH (2018), dengan judul Sistem Informasi Penggajian (Studi Kasus: kantor Ratona Televisi Palopo). Hasil dari penelitian ini adalah membangun sebuah aplikasi sistem informasi penggajian yang dibuat terdiri dari proses pengolahan data master, absensi, dan penggajian yang disertai dengan laporan-laporan yang berkaitan dengan penggajian. Sebagai kesimpulan bahwa sistem informasi penggajian yang dibuat telah terkomputerisasi secara online, efisien, bermanfaat dan mempermudah pengolahan data penggajian

Berdasarkan penelitian yang dilakukan Rosadi D Mardira (2020), dengan penelitiannya yang berjudul Sistem Informasi Gaji Karyawan Menggunakan Codeigniter. Peneliti mengambil objek di CV. Davina 23. peneliti menyimpulkan dengan adanya aplikasi sistem informasi penggajian yang dihasilkan dapat berfungsi untuk melakukan pemrosesan, pemrosesan transaksi, dan pembuatan laporan. Dengan sistem informasi yang terkomputerisasi, informasi diproses lebih cepat dan perhitungan dapat dilakukan dengan lebih tepat dan akurat.

Berdasarkan penelitian yang dilakukan oleh Kusumo, Bayu (2020), dengan judul Aplikasi Penggajian Karyawan Pada CV System. Tujuan dari penelitian tersebut adalah untuk meningkatkan internal control dan mencegah pihak-pihak yang tidak berwenang mengakses data. Sistem informasi penggajian yang dirancang menghasilkan data yang mengintegrasikan data dari berbagai

aktivitas yang berhubungan dengan penggajian seperti data absensi, lembur, pinjaman, uang makan dan transport.

Penelitian yang dilakukan oleh Gusmayeni , Avy Sukmawati dan Dika Cahya Ramadhan (2017) dengan judul Perancangan Sistem Aplikasi Penggajian Karyawan Pada PT. Adika Safetama Tunggal Berbasis Web. Sistem penggajian di PT. Adika Safetama Tunggal saat ini memiliki banyak kelemahan dan keterbatasan. Dari kelemahan kelemahan tersebut diantaranya adalah sistem absensi yang kurang efektif, cara perhitungan gaji yang kurang efisien, penyimpanan data gaji yang kurang terjamin, sulitnya akses data dan informasi mengenai karyawan dan ada kesulitan dalam pembuatan laporan mengenai penggajian karyawan. Absensi PT. Adika Safetama Tunggal masih menggunakan form absensi yang diprint setiap bulan dan direkap oleh admin. Sistem absensi diatas dinilai masih kurang efektif. tujuan peneliti untuk menganalisis sistem penggajian di lokasi, merancang sistem informasi penggajian dan memberikan usulan perencanaan sistem informasi penggajian sesuai kebutuhan.

II.2. Landasan Teori

II.2.1. Perancangan

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari berbagai elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Pengertian perancangan menurut para ahli diantaranya adalah sebagai berikut :

1. Menurut Varzello/John Reuter III perancangan adalah tahap setelah analisis dari siklus pengembang sistem : Pendefinisian dari kebutuhankebutuhan fungsional dan persiapan untuk rancang bangun implementasi : “ Mengembangkan bagaimana suatu sistem dibentuk”.
2. Menurut John Buch & Gary Grudnitski perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
3. Menurut George M. Scott perancangan adalah menentukan bagaimana sistem akan menyelesaikan apa yang mesti diselesaikan ; tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. (Kholik Hidayatulloh, et. al. ; 2020 : 20)

II.2.2. Aplikasi

Pengertian aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi user dengan menggunakan kemampuan komputer. Sedangkan pengertian penjualan adalah suatu proses seseorang atau organisasi untuk meyakinkan customer membeli produk yang ditawarkan. Aplikasi mobile dapat diartikan sebagai sebuah produk dari sistem komputasi mobile, yaitu sistem komputasi yang dapat dengan mudah dipindahkan secara fisik dan yang komputasi kemampuan dapat digunakan

saat mereka sedang dipindahkan. Contohnya adalah personal digital assistant (PDA), smartphone dan ponsel. (Listianto et al., 2017).

II.2.3. Codeigniter

Codeigniter merupakan *framework* untuk pemrograman php yang di dalamnya terdiri dari *library-library* untuk mempermudah dalam melakukan pemrograman PHP. *Codeigniter* merupakan toolkit untuk orang-orang yang ingin membuat atau membangun aplikasi web menggunakan bahasa pemrograman PHP. Adapun beberapa keunggulan yang ditawarkan oleh *Codeigniter* adalah sebagai berikut:

1. Codeigniter merupakan framework yang bersifat gratis atau open source,
2. Codeigniter memiliki ukuran file yang relatif kecil dibandingkan Framework php lain,
3. Aplikasi yang dibuat menggunakan codeigniter dapat berjalan dengan cepat,
4. Codeigniter menggunakan pola desain Model-View-Controller (MVC) yang memungkinkan pada satu file tidak akan berisi banyak code,
5. Codeigniter dapat diperluas sesuai dengan kebutuhan,
6. Codeigniter juga memiliki dokumentasi yang sangat baik,
7. Pack a Punch, codeigniter hadir dengan library yang akan membantu tugas-tugas di pengembangan web. (Ariyanti : 2018).

I.2.4. Penggajian

Gaji merupakan suatu imbalan berupa uang yang diterima oleh karyawan dari hasil jasa mereka dalam memberikan tenaga dan pikiran mereka untuk mencapai tujuan dari perusahaan. Setiap orang yang bekerja pasti mendapatkan berbeda-beda dalam menerima gaji tersebut. Gaji biasanya diberikan berdasarkan jabatan, pekerjaan yang dikerjakan, dan juga berdasarkan pengalaman kerja karyawan. Gaji sangat berhubungan dengan kinerja karyawan. Jika gaji yang diterima oleh karyawan dapat memenuhi kebutuhan dan sesuai dengan yang mereka kerjakan maka kinerja juga akan mengalami peningkatan. Sebaliknya jika perusahaan tidak memberikan gaji sesuai dengan apa yang telah mereka kerjakan maka akan mempengaruhi terhadap kualitas dan kuantitas kerja karyawan. (Hasni Nur : 2020)

II.2.5. MySQL

MySQL merupakan turunan dari konsep utama database yang sudah ada sebelumnya yaitu SQL (*Structured Query Language*). SQL merupakan sebuah konsep untuk pengoprasian database yaitu untuk pemilihan, seleksi, memasukkan data. Sehingga hal tersebut akan memungkinkan pengoprasian data dapat dilakukan dengan cara yang mudah. MySQL digunakan untuk menyimpan serta manajemen data atau biasa dikenal sebagai perangkat lunak sistem manajemen basis data SQL (*Database Management system*). MySQL juga dikenal sebagai DBMS yang *multithread* dan *multiuser* dengan jumlah instalasi mencapai 6 juta diseluruh dunia.

MySQL merupakan perangkat lunak gratis dibawah lisensi GPL (*General Public License*). Sehingga pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

Tujuan awal dikembangkan MySQL adalah untuk mengembangkan aplikasi berbasis webpada sisi client.

(Eko Prio Utomo ; 2016 : 192)

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan (*database management system*), database ini multith read, multiuser. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL). Kekuatan MySQL tidak ditopang oleh sebuah komunitas, seperti Apache, yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh pemilik masing-masing, tetapi MySQL didukung penuh oleh pemilik sebuah perusahaan profesional dan komersial, yakni MySQL AB dari Swedia.

MySQL adalah *RelationalDatabase Management System* (DBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya,

terutama dalam kecepatan. Berikut ini beberapa keistimewaan MySQL, antara lain :

1. *Portability*

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *windows*, *linuc*, *freeBSD*, *Mac Os X Server*, *Solaris*, *Amiga* dan masih banyak lagi.

2. *Multiuser*

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

3. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

4. *Scalability* dan *Limits*

MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya (M Agus ; 2015 : 181).

II.2.6. Normalisasi

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria

tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data).

Tujuan dari normalisasi adalah untuk :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh *base relations* (file)
2. Untuk mengurangi kompleksitas
3. Untuk mempermudah pemodifikasian data

Gambaran proses normalisasi adalah :

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat, kemudian
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

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

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.3. 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.4. 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.5. 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.6. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

Sumber : Mukhlisulfatih Latief : 2016

II.2.7. Basis Data (*Database*)

Database merupakan sekumpulan dari data yang terintegrasi. Sistem *Database* merupakan sistem komputerisasi yang bertujuan untuk menyimpan sejumlah data sehingga memudahkan pengguna untuk mendapatkan dan mengupdate informasi sesuai dengan kebutuhan Informasi yang disimpan dalam *Database* dapat berupa *text* maupun angka. Salah satu penggunaan *Database* dalam kehidupan sehari-hari adalah pada aplikasi *software* pembayaran oleh kasir supermarket. Ketika melakukan pembayaran dikasir, kasir tersebut melakukan pekerjaan dengan cara menempelkan *bar-code* di salah satu sensor *infrared*. Kemudian muncul harga dari barang tersebut dalam komputer. Sehingga kasir tersebut tidak perlu mengetik ulang barang yang dibeli oleh pembeli tersebut.

Aplikasi pembayaran oleh kasir merupakan salah satu contoh dari penggunaan *Database* yang terintegrasi oleh komputer. Dalam aplikasi komputer tersebut telah terekam data-data barang beserta harga, sehingga kasir dengan mudah mendapatkan data barang serta saat melakukan input barang, maka *Database* tersebut secara otomatis terupdate. Salah satu keuntungan dari *Database* adalah dapat mempersingkat waktu pekerjaan. Sebuah *Database* dapat menampung data yang kompleks yang dapat digunakan secara internal maupun

eksternal. *Software* yang digunakan oleh suatu organisasi untuk mengendalikan *Database* disebut dengan *Database Manegement System* atau dapat disingkat dengan DBMS. DBMS merupakan suatu kumpulan program yang memudahkan pengguna untuk membuat dan mengelola *Database*. Data yang ada pada DBMS dapat dimanipulasi atau dikelola dengan menggunakan *query*, dimana dengan *query* tersebut pengguna dapat mendapatkan data spesifik yang dibutuhkan, serta dapat mengupdate informasi dalam data-data tersebut. (Intan Mardiono ; 2019 : 103-104)

II.2.8. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat men-download langsung dari web resminya. (Riyadli et al., 2020).

II.2.9. PHP

PHP atau kependekan dari Hypertext Preprocessor adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk

pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP merupakan bahasa scripting server – side, dimana pemrosesan datanya dilakukan pada sisi server. Sederhananya, server yang akan menterjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada client yang melakukan permintaan. Adapun pengertian lain PHP adalah akronim dari Hypertext Preprocessor, yaitu suatu bahasa pemrograman berbasis kode – kode (script) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke web browser menjadi kode HTML”. (Firman et al., 2016).

II.2.10. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat blue print atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

Unified Modeling Language (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan use case dan actor.

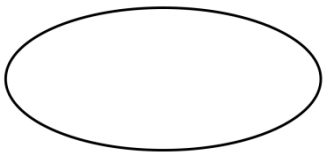
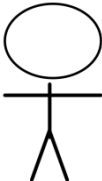
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan interaction diagrams.
- c. Menggambarkan representasi struktur static sebuah sistem dalam bentuk class diagrams.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan state transition diagrams.
- e. Menyatakan arsitektur implementasi fisik menggunakan component and development.
- f. Menyampaikan atau memperluas functionality dengan stereotypes. (Alfina & Harahap, 2019).

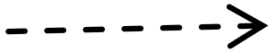
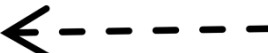
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah Empat alat bantu sebagai berikut: Use Case Diagram, Activity Diagram, Sequence Diagram dan Class Diagram.

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk melakukan (*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>.
	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 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 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.

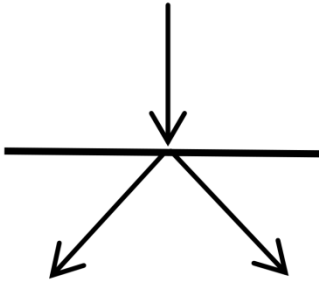
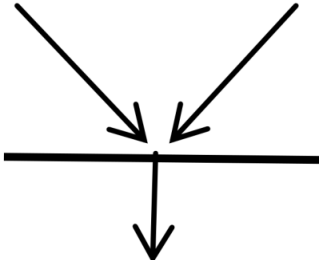
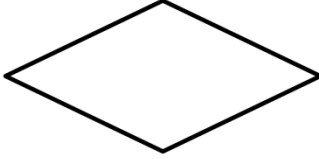
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015 : 94)

2. Activity Diagram (Diagram Aktivitas)

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. 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.

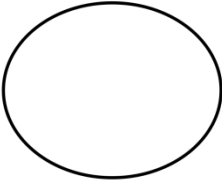
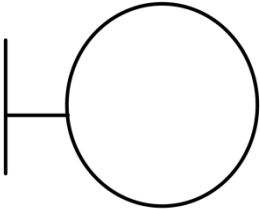
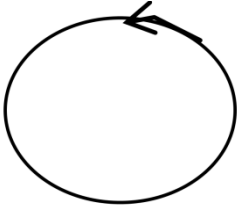
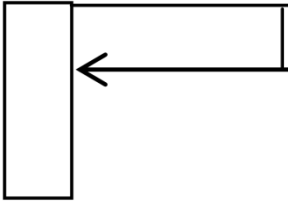
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.
	<i>Swimlane</i>, pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015 : 94)

3. *Sequence Diagram* (Diagram Urutan)

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. 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 interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>Form Entry</i> atau 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.
	<i>Message</i>, simbol mengirim pesan antar <i>class</i>.
	<i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<i>Activation</i>, activation 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 : Gellysa Urva dan Helmi Fauzi Siregar, 2015 : 95)

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 *cardinality*.

Tabel II.4. *Multiplicity Class Diagram*

<i>Multiplicity</i>	Penjelasan
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 : Gellysa Urva dan Helmi Fauzi Siregar, 2015 : 95)