

BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

Adapun kajian teori-teori umum yang digunakan dalam pengembangan sistem ini adalah sebagai berikut:

2.1.1. Pengertian Sistem Informasi

“Sistem informasi adalah suatu kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi”. (Ali & Wangdra, 2010:13)

“Sistem informasi adalah sekumpulan komponen yang saling berhubungan, mengumpulkan, memproses, menyimpan, mendistribusikan informasi untuk menunjang pengambilan keputusan dan pengawasan dalam suatu organisasi”. (Laudon & Laudon, 2007:15)

(Laudon & Laudon, 2007:16) melaporkan terdapat tiga aktivitas di dalam sistem informasi yang dibutuhkan di dalam organisasi untuk membuat keputusan, mengendalikan operasi, menganalisis permasalahan, dan menciptakan produk baru. Ketiga aktivitas tersebut adalah:

1. *Input*

Input adalah kegiatan merekam atau mengumpulkan data mentah dari dalam maupun luar organisasi.

2. *Pemrosesan*

Pemrosesan adalah mengubah data input mentah menjadi bentuk yang berarti.

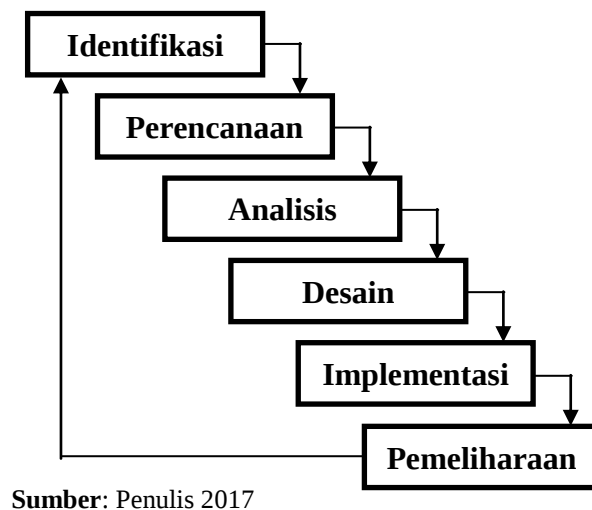
3. *Output*

Output adalah mengirmkan informasi yang telah diproses tersebut ke orang-orang yang akan menggunakan atau kepada aktivitas yang akan menggunakan informasi tersebut.

Dari pendapat di atas, dapat disimpulkan bahwa sistem informasi adalah sekumpulan komponen yang saling berhubungan dan sudah diolah menjadi informasi untuk pengambilan keputusan.

2.1.2. SDLC (*System Development Life Cycle*)

“SDLC (*System Development Life Cycle*) merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan desain”. (Fatta, 2007:25)



Gambar 1.1. Tahapan SDLC

(Fatta, 2007:26) melaporkan terdapat 6 tahapan dalam SDLC, yaitu:

1. Identifikasi

Merupakan langkah pertama dalam SDLC keseluruhan informasi yang dibutuhkan oleh sistem: identifikasi, analisis, prioritas, dan susunan ulang. Dalam hal ini ada beberapa hal yang harus dilakukan:

- a. Mengidentifikasi proyek-proyek yang potensial
- b. Melakukan kalsifikasi dan merangking proyek
- c. Memilih proyek untuk dikembangkan

2. Perencanaan (*Planning*)

Dalam tahap ini, proyek SI yang potensial dijelaskan dan argumentasi untuk melanjutkan proyek dikemukakan. Pada tahapan ini ditentukan secara *detail* rencana kerja yang harus dikerjakan,

durasi yang perlukan masing-masing tahap, Sumber Daya Manusia, perangkat lunak, perangkat keras maupun *financial* diestimasi.

3. Analisis

Tahapan analisis merupakan tahapan di mana sistem yang sedang berjalan dipelajari dan sistem pengganti diusulkan.

Adapun aktivitas utama dalam analisis ini terdiri dari:

- a. Pengumpulan informasi
- b. Mendefinisikan sistem *requirement*
- c. Memprioritaskan kebutuhan
- d. Mengulas kebutuhan dengan pihak manajemen

4. Desain

Tahapan desain adalah tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang riil. Tahapan desain sistem dapat dibagi menjadi 2 tahap, yaitu desain logis (*logical design*) dan tahapan desain fisik (*physical design*).

5. Implementasi

Pada tahapan ini, terdapat beberapa hal yang perlu dilakukan, yaitu:

- a. *Testing*, yaitu menguji hasil kode program yang telah dihasilkan dari tahapan desain fisik.
- b. Instalasi, proses ini dilakukan setelah program lulus dicoba, maka perangkat lunak dan keras akan diinstall pada perusahaan klien dan secara resmi mulai digunakan untuk menggantikan sistem lama.

6. Pemeliharaan

Dalam tahapan ini, sistem secara sistematis diperbaiki dan ditingkatkan. Hasil dari tahapan ini adalah versi baru dari perangkat lunak yang telah dibuat.

2.2. Teori Khusus

Adapun kajian teori-teori khusus yang digunakan dalam pengembangan sistem ini adalah sebagai berikut:

2.2.1. Pengertian Gaji

Mulyadi (2011) dalam (Mulyapradana & Hatta, 2016:6) melaporkan gaji merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan yang mempunyai jenjang jabatan tertentu seperti manager, sementara upah merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan pelaksana (buruh).

Hadi Purwono (2003) dalam (Mulyapradana & Hatta, 2016:6) melaporkan gaji biasanya dikatakan upah yang dibayarkan kepada pimpinan, pengawas, dan tata usaha, karyawan kantoran atau manajer lainnya. Upah adalah pembayaran kepada karyawan atau pekerja yang dibayarkan menurut lamanya jam kerja dan diberikan kepada mereka yang biasanya tidak mempunyai jaminan untuk dipekerjakan secara terus-menerus.

“Gaji merupakan bayaran tetap yang diterima seseorang dari keanggotaannya dalam sebuah perusahaan, sedangkan upah merupakan imbalan finansial langsung yang dibayarkan kepada karyawan berdasarkan jam kerja dan jumlah barang yang dihasilkan.” (Mulyapradana & Hatta, 2016:27)

2.2.2. Komponen Gaji

(Mulyapradana & Hatta, 2016:27) melaporkan komponen-komponen dari gaji meliputi hal-hal, sebagai berikut:

1. Gaji Pokok

Gaji Pokok adalah imbalan dasar yang diberikan kepada karyawan menurut tingkat atau jenis pekerjaannya, yang besarnya ditetapkan berdasarkan kesepakatan pada saat karyawan akan diterima bekerja.

2. Tunjangan

Tunjangan merupakan bentuk imbalan kepada karyawan untuk berbagai alasan dan sebagai bentuk penghargaan dari perusahaan kepada karyawan atas kontribusinya.

1. Tunjangan Tetap

Tunjangan tetap yaitu suatu pembayaran yang teratur berkaitan dengan pekerjaan yang diberikan secara tetap untuk pekerja dan keluarganya.

2. Tunjangan Tidak Tetap

Tunjangan tidak tetap yaitu suatu pembayaran yang secara langsung atau tidak langsung berkaitan dengan pekerja, yang diberikan secara tidak tetap untuk pekerja dan keluarga.

Pada umumnya, komponen-komponen gaji atau daftar gaji (*payroll*) ditetapkan berdasarkan kebijakan masing-masing perusahaan. Namun dalam penerapannya, setiap perusahaan memiliki kebijakan dan tujuan berbeda-beda dalam menentukan komponen gaji. berikut adalah komponen-komponen tunjangan tidak tetap:

- a. Gaji pokok atau *basic salary*
- b. Tunjangan jabatan
- c. Tunjangan kinerja atau bonus kinerja
- d. Uang makan atau *meal allowance*
- e. Uang lembur atau *overtime*
- f. Tunjangan pajak atau *tax allowance*
- g. Tunjangan lain-lain atau *others allowance*
- h. Tunjangan pengobatan atau *medical allowance*
- i. Tunjangan asuransi atau *insurance incentive*
- j. Tunjangan premi jaminan sosial (BPJS)
- k. Tunjangan transportasi atau *transportation allowance*
- l. Potongan premi jaminan sosial (BPJS)
- m. Potongan transportasi atau *transport deduction*

- n. Potongan absensi atau *absence deduction*
- o. Potongan pajak atau *tax cuts*
- p. Potongan lain-lain atau *others deduction*

2.2.3. Pajak Penghasilan (PPh 21)

(Mulyapradana & Hatta, 2016:97) melaporkan peserta wajib pajak terbagi menjadi 6 kategori, antara lain pegawai, bukan pegawai yang memperoleh penghasilan melalui pemberian jasa, penerima pensiun dan pensangon, anggota dewan komisaris atau dewan pengawas, mantan pegawai, dan peserta yang memperoleh penghasilan dalam kegiatan itu. Peserta wajib pajak yang dimaksud adalah para peserta wajib pajak yang memiliki nomor pokok wajib pajak (NPWP). Berikut ini merupakan penghasilan dan persentase tarif pajak PPh 21 berdasarkan peraturan ditjen pajak nomor PER-32/PJ/2015:

1. Wajib pajak dengan penghasilan tahunan sampai dengan Rp 50.000.000 adalah 5%.
2. Wajib pajak dengan penghasilan tahunan di atas Rp 50.000.000 sampai dengan Rp 250.000.000 adalah 15%
3. Wajib pajak dengan penghasilan tahunan di atas Rp 250.000.000 sampai dengan Rp 500.000.000 adalah 25%
4. Wajib pajak dengan penghasilan tahunan di atas Rp 500.000.000 adalah 30%

5. Untuk wajib pajak yang tidak memiliki NPWP, dikenai tarif 20% lebih tinggi dari mereka yang memiliki NPWP.

2.2.4. Iuran kesehatan

(Mulyapradana & Hatta, 2016:99) melaporkan besaran iuran yang harus dibayarkan ke BPJS Kesehatan adalah, sebagai Berikut:

1. Bagi peserta penerima bantuan iuran (PBI), jaminan kesehatan iuran dibayar oleh pemerintah.
2. Bagi peserta pekerja penerima upah yang bekerja pada lembaga pemerintahan, yang terdiri atas PNS, anggota TNI, anggota Polri, pejabat Negara, dan pegawai pemerintah non-pegawai negeri pembayaran iuran sebesar 5% dari gaji per bulan dengan ketentuan, yaitu 3% dibayar oleh pemberi kerja dan 2% dibayar oleh peserta.
3. Bagi peserta pekerja penerima upah yang bekerja di BUMN, BUMD, dan swasta pembayaran iuran sebesar 5% dari gaji per bulan dengan ketentuan, yaitu 4% dibayar oleh pemberi kerja dan 1% dibayar oleh peserta.
4. Untuk keluarga tambahan pekerja penerima upah yang terdiri atasanak ke-4 dan seterusnya, ayah, ibu, dan mertua, pembayaran iuran sebesar 1% dari gaji per orang per bulan, dibayar oleh pekerja penerima upah.

5. Bagi kerabat lain dari pekerja penerima upah (seperti saudara kandung atau ipar dan asisten rumah tangga), peserta bukan penerima upah, serta peserta bukan pekerja adalah, sebagai berikut:
 - a. Sebesar Rp 25.500 per orang per bulan dengan manfaat pelayanan di ruang perawatan kelas III.
 - b. Sebesar Rp 51.000 per orang per bulan dengan manfaat pelayanan di ruang perawatan kelas II.
 - c. Sebesar Rp 80.000 per orang per bulan dengan manfaat pelayanan di ruang perawatan kelas I.
6. Bagi veteran, perintis kemerdekaan, serta janda, duda, atau anak yatim piatu dari beteran atau perintis kemerdekaan, pembayaran iuran ditetapkan sebesar 5% dari 45% gaji pokok PNS golongan ruang III/a dengan masa kerja 14 tahun per bulan, dibayar oleh pemerintah.
7. Pembayar iuran paling lambat tanggal 10 (sepuluh) setiap bulan.

2.2.5. Iuran Sosial

(Mulyapradana & Hatta, 2016:102) melaporkan iuran sosial merupakan salah satu program pemerintahan yang wajib diikuti oleh semua perusahaan. BPJS Ketenagakerjaan (Badan Penyelenggara Jaminan Sosial Ketenagakerjaan) yang sebelumnya bernama Jamsostek, merupakan program yang memberikan perlindungan kepada karyawan dalam mengatasi risiko sosial ekonomi tertentu dan merupakan hak dasar karyawan yang wajib diterima dalam

rangka mewujudkan rasa adil, sejahtera, makmur, dan pemerataan. Adapun program-program dalam BPJS Ketenagakerjaan, sebagai berikut:

1. Program Jaminan Kecelakaan Kerja (JKK)

Memberikan perlindungan atas risiko-risiko kecelakaan yang terjadi dalam hubungan kerja, termasuk kecelakaan yang terjadi dalam perjalanan dari rumah menuju tempat kerja atau sebaliknya dan penyakit yang disebabkan oleh lingkungan kerja dan mengacu pada tabel sebagai berikut:

Tabel 2.1. Persentase Tingkat Risiko Lingkungan Kerja

No	Tingkat Risiko Lingkungan Kerja	Besaran Persentase
1.	Tingkat risiko sangat rendah	0,24% dari upah sebulan
2.	Tingkat resiko rendah	0,54% dari upah sebulan
3.	Tingkat resiko sedang	0,89% dari upah sebulan
4.	Tingkat resiko tinggi	1,27% dari upah sebulan
5.	Tingkat resiko sangat tinggi	1,74% dari upah sebulan

Sumber: Mulyapradana & Hatta, 2016:104 Persentase Tingkat Risiko Lingkungan Kerja

2. Program Jaminan Kematian (JKM)

Memberikan manfaat uang tunai kepada ahli waris, ketika peserta meninggal dunia bukan akibat kecelakaan kerja. Adapun iuran JKM adalah sebagai berikut:

- a. Bagi peserta penerima upah sebesar 0,30% dari gaji sebulan
- b. Bagi peserta bukan penerima upah sebesar Rp 6.800 setiap bulan.

3. Program Jaminan Hari Tua (JHT)

Memberikan manfaat uang tunai yang besarnya merupakan nilai akumulasi iuran ditambah hasil pengembagannya yang dibayarkan secara sekaligus jika peserta telah mencapai usia 56 tahun, meninggal dunia, atau cacat total tetap. Besaran iuran dihitung sebesar 5,7%, yang terdiri atas 3,7% iuran pemberi kerja dan 2% iuran pekerja.

4. Program Jaminan Pensiun (JP)

Memberikan sejumlah uang setiap bulan kepada peserta yang memasuki usia pensiun, mengalami cacat total tetap, atau ahli waris bagi peserta yang meninggal dunia. Besaran iuran program ini dihitung sebesar 3%, yang terdiri dari atas 2% iuran pemberi kerja dan 1% iuran pekerja.

2.2.6. *Web*

(Fathurrahman, 2014:2) melaporkan *web* merupakan kumpulan yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terikat di mana masing-masing dihubungkan dengan jaringan halaman (*hyperlink*).

2.2.7. *Browser*

“*Browser* merupakan *software* yang diinstall di mesin *client*, berfungsi untuk menerjemahkan *tag* HTML menjadi halaman *web*. *Browser* yang sering digunakan adalah Internet Explorer, Mozilla Firefox, Opera, Netscape, dan lainnya.” (Anhar, 2010:6)

2.2.8. HTML (*Hyper Text Markup Language*)

“HTML (*Hyper Text Markup Language*) merupakan bahasa pemrograman *web* yang memiliki sintak atau aturan tertentu dalam menuliskan *script* atau kode-kode, sehingga *browser* dapat menampilkan informasi dengan membaca kode-kode HTML. HTML adalah sekumpulan symbol-simbol atau *tag-tag* yang dituliskan dalam sebuah file yang digunakan untuk menampilkan halaman pada *web browser*.” (Anhar, 2010:40)

“HTML (*Hyper Text Markup Language*) adalah sebuah protokol yang digunakan untuk membuat format suatu dokumen *web* yang mampu dibaca dalam *browser* dari berbagai *platform* komputer. Sifat bahasa HTML ini adalah *client script*, di mana dokumen tersebut dapat dibuka dalam komputer *stand alone* yang tidak membutuhkan *server* untuk dapat menampilkannya di dalam *browser*. Dokumen HTML merupakan file yang pada umumnya beresktensi . htm atau . html, di mana bahasa HTML tersebut tersusun atas *tag* yang berformat *<isi tag>*.” (Sugiri & Kurniawan, 2007:1)

2.2.9. XAMPP

“Bahwa XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP berperan sebagai *server web* pada komputer lokal. XAMPP juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi *website* tanpa harus *online* atau terakses dengan internet.” (Wicaksono, 2008:7)

2.2.10. PHP

“PHP singkatan dari PHP: *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server (server side HTML embedded scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan.” (Anhar, 2010:3)

2.2.11. MySQL

(Enterprise, 2014:2) melaporkan MySQL adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan. MySQL dikembangkan oleh MySQL AB Swedia.

Berikut ini hal-hal yang menyebabkan MySQL menjadi begitu Populer:

1. Berlisensi *open-source*, sehingga Anda dapat menggunakannya secara gratis.
2. Merupakan program yang *powerful* dan menyediakan fitur yang lengkap.
3. Menggunakan bentuk standar bahasa data SQL.
4. Dapat bekerja dengan banyak sistem operasi dan dengan bahasa-bahasa pemrograman seperti PHP, PERL, C, C++, JAVA, dan lain-lain.
5. Bekerja dengan cepat dan baik, bahkan dengan data set yang banyak.
6. Sangat mudah digunakan dengan PHP untuk pengembangan aplikasi *web*.
7. Mendukung banyak *database*, sampai 50 juta baris atau lebih dalam suatu tabel.
8. Dapat dikostumisasi sesuai dengan keinginan Anda.

2.2.12. CSS (*Cascading Style Sheets*)

“CSS (*Cascading Style Sheets*) adalah bahasa pemrograman *web* untuk mengendalikan komponen pada sebuah *web* sehingga lebih terstruktur. CSS lebih cenderung pada *style*, seperti ukuran gambar, warna teks dan tabel, ukuran dan warna *border*, warna *hyperlink* dan *mouse over*, spasi antar

paragraf, antar teks, margin kiri, kanan, atas, bawah, dan parameter lain.”
(Tohirudin, 2011:3)

“CSS (*Cascading Style Sheets*) adalah sebuah cara untuk memisahkan isi dengan *layout* dalam halaman-halaman *web* yang dibuat. CSS memperkenalkan template yang berupa style untuk membuat dan mempermudah penulisan dari halaman-halaman yang dirancang. CSS merupakan style yang banyak digunakan karena berbagai kemudahan dan kelengkapan atribut yang dimilikinya.” (Sugiri & Kurniawan, 2007:1)

2.2.13. JavaScript

(Yatini I., 2014:2) melaporkan JavaScript adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client*. JavaScript digunakan dalam pembuatan *website* agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi *browser*. JavaScript dapat merespon perintah *user* dengan cepat dan menjadikan halaman web menjadi *responsif*.

JavaScript memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi. Struktur penulisan JavaScript adalah sebagai berikut.

```
<script language = "javascript">

<!--Penulisan kode javascript-->

</script>
```

2.2.14. jQuery

(Yatini I., 2014:2) melaporkan jQuery adalah JavaScript *library* yang dirancang untuk meringkas kode-kode JavaScript, sehingga dapat menyederhanakan penulisan skrip program, sesuai dengan slogan “*write less, do more*”. jQuery pertama kali dirilis oleh John Resig pada tahun 2006, pada perkembangannya jQuery tidak hanya sebagai *framework* JavaScript, namun memiliki kelebihan antara lain.

1. Kemudahan mengakses dan memanipulasi elemen-elemen HTML.
2. Manipulasi CSS.
3. Penanganan event HTML.
4. Efek-efek JavaScript dan animasi.
5. Memodifikasi elemen HTML DOM.

Sintak dasar jQuery `$(selector).action()`, tanda `$` untuk mendefinisikan jQuery, jQuery *selector* digunakan untuk mendapatkan elemen HTML, *action* adalah tindakan yang dilakukan jQuery pada elemen (). Contoh penggunaan jQuery untuk menyembunyikan elemen dengan id “test” sebagai berikut.

```
$("#test").hide()
```

Semua metode jQuery berada di dalam fungsi *document.ready()* yaitu perintah inisialisasi yang menunjukkan dokumen telah siap ditampilkan dan sekaligus menjalankan perintah yang terdapat didalam fungsi.

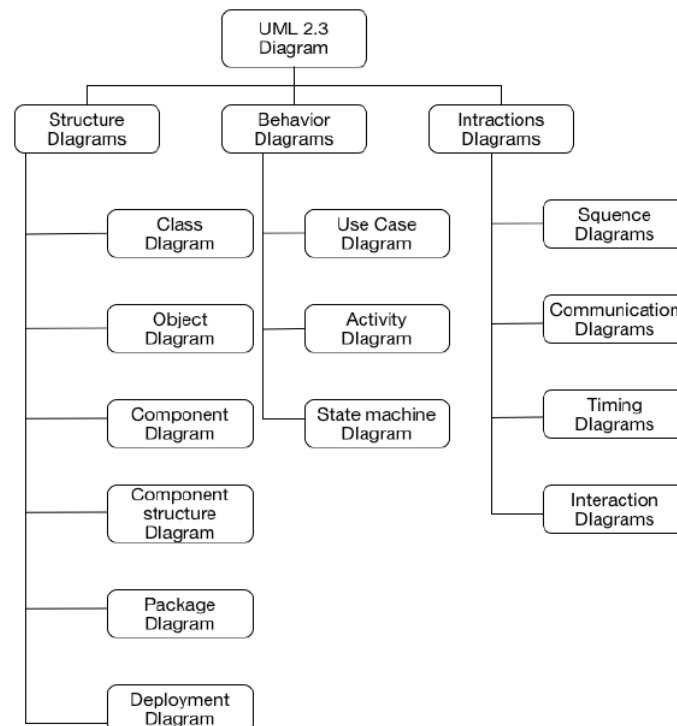
2.2.15. Diagram UML

“UML (*Unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.” (Rosa dan Shalahuddin, 2011: 118)

“UML (*Unified Modelling Language*) adalah bahasa visual untuk menggambarkan pembangunan perangkat lunak yang menggunakan teknik berorientasi objek.” (Nugraha, Wirawan, & Arthana, 2016:5)

Berdasarkan beberapa pendapat yang dikemukakan di atas, maka dapat disimpulkan bahwa *Unified Modelling Language* merupakan bahasa pemrograman yang digunakan untuk visualisasi, merancang, dan mendokumentasikan sebuah piranti lunak yang berorientasi objek.

Diagram UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah ini:



Sumber: Rosa dan Shalahuddin, 2011: 121

Gambar 1.2. Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut (Rosa dan Shalahuddin, 2011:121):

1. *Structure diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkain perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antara subsistem pada suatu sistem.

Ada 4 diagram yang digunakan oleh penulis dalam penggunaan UML dalam pemrograman berorientasi objek yaitu *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* untuk membantu dalam analisa dan desain, berikut penjelasnnya:

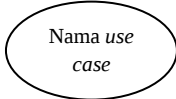
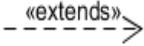
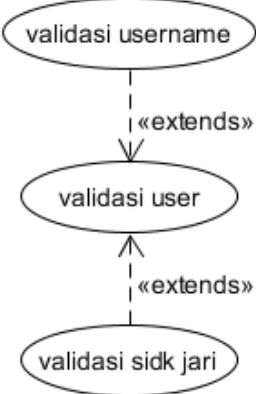
1. *Use Case Diagram*

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. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

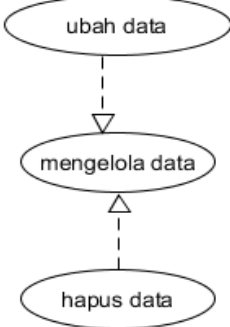
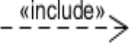
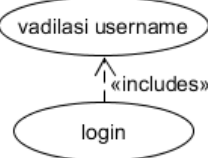
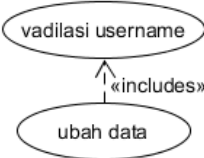
- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol dari *use case diagram*:

Tabel 2.2. Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Use Case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal di awal frase nama <i>use case</i>
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun <i>symbol</i> dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
	Asosiasi/ <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
	Ekstensi/ <i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan yaitu , mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan sama dengan <i>use case</i> yang ditambahkan  Arah panah mengarah pada <i>use case</i> yang ditambahkan

Tabel Lanjutan 2.2. Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:  Arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum)
	Menggunakan/ <i>Include/uses</i>	Relasi <i>use case</i> tambah ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>: • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> yang ditambahkan dijalankan  • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan  Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan

Sumber: Rosa & Shalahuddin, 2011: 131

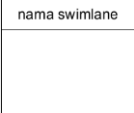
2. *Activity Diagram*

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem/*user interface* di mana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan khusus ujinya.

Berikut adalah simbol-simbol dari *activity diagram*:

Tabel 2.3. Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan/ <i>decision</i>	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
	Penggabungan/ <i>join</i>	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Rosa & Shalahuddin, 2011: 134

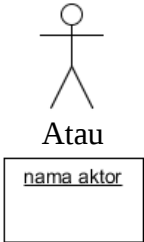
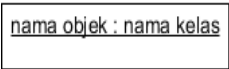
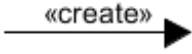
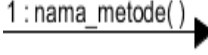
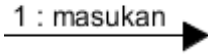
3. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya diagram sekuen yang harus digambar 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 sekuen sehingga semakin

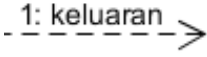
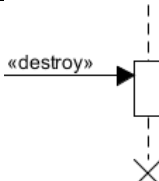
banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol dari *sequence diagram*:

Tabel 2.4. Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
 Atau Tanpa waktu aktif	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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 di awal frase nama aktor
	Garis hidup/ <i>lifeline</i>	Menyatakan bahwa kehidupan suatu objek
	Objek	Menyatakan bahwa objek yang berinteraksi pesan
	Waktu Aktif	Menyatakan bahwa objek dalam keadaan aktif dan berinteraksi pesan
	Pesan tipe <i>create</i>	Menyatakan bahwa suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
	Pesan tipe <i>call</i>	Menyatakan bahwa suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
	Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim

Tabel Lanjutan 2.4. Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
	Pesan tipe <i>destroy</i>	Menyatakan bahwa suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

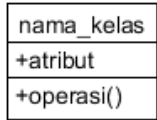
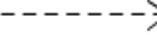
Sumber: Rosa & Shalahuddin, 2011: 138

4. Class Diagram

Class diagram menggambarkan struktur sistem dari pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam suatu sebuah kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.5. Simbol *Class Diagram*

Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem
	Antaramuka/ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
	Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Asosiasi berarah/ <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai <i>multiplicity</i>
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
	Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
	Agregasi/ <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole part</i>)

Sumber: Rosa & Shalahuddin, 2011: 123

2.3. Penelitian Terdahulu

Berikut ini dikemukakan hasil penelitian terdahulu yang digunakan sebagai dasar dalam penelitian ini, yaitu:

1. Pada penelitian yang dilakukan oleh **Mumpuni dan Maknunah (2014)** dalam penelitiannya yang berjudul **Aplikasi Sistem Informasi Gaji Berbasis Web di STMIK Pradnya Paramita dengan Pendekatan Data Warehouse**. Dalam penelitian ini objek penelitiannya adalah dosen STMIK Pradnya Paramita, dengan menggunakan pendekatan data warehouse. Hasil yang diperoleh adalah dapat mempermudah kinerja dalam melakukan pemantauan dan pengambilan keputusan yang tepat bagi pihak lembaga dan juga membantu administrator menggunakan manajemen kerja secara efektif, efisien dan tepat waktu dalam memproses penggajian serta pembuatan laporan pertanggungjawaban kepada pemimpin.
2. Pada penelitian yang dilakukan oleh **Nugroho dan Wulandary (2016)** dalam penelitian yang berjudul **Pengelolaan Presensi dan Gaji Asisten Lab Berbasis Web di Fasilkom Universitas Mercu Buana**. Dalam penelitian ini objek penelitiannya adalah asisten lab Fasilkom Universitas Mercu Buana, Hasil yang diperoleh adalah data kehadiran lebih dapat dipertanggungjawabkan karena tersimpan langsung ke dalam aplikasi dari data kehadiran yang di *approve* oleh dosen, pergantian asisten lab terpantau dengan menggunakan fitur menu pengganti asisten yang dipilih langsung oleh laboran, perhitungan gaji

asisten lab secara otomatis berdasarkan data kehadiran asisten lab yang masuk pada aplikasi, gaji yang diperoleh sesuai dengan hak sehingga tidak merugikan asisten lab ataupun Universitas Mercu Buana, dan bagian keuangan dapat melihat laporan melalui aplikasi yang dapat mengurangi penggunaan kertas.

3. Pada penelitian yang dilakukan oleh **Ariesna (2014)** dalam jurnalnya yang berjudul **Pembangunan Sistem Absensi dan Honor Guru SMA Negeri 2 Kotabumi Menggunakan Web Framework Codeigniter**. Dalam penelitian ini objek penelitiannya adalah guru SMA Negeri 2, dengan menggunakan rekayasa perangkat lunak metode *waterfall*. Hasil yang diperoleh adalah sistem pengolahan data absensi guru dan honor guru dapat memudahkan dalam hal pengimputan data dan pengolahan laporan yang diperlukan, mempermudah penyimpanan dan pencarian data yang diperlukan baik secara individu maupun data keseluruhan berdasarkan tanggal, bulan, dan tahun yang diperlukan, *framework codeigniter* dan *library* pendukungnya lebih memudahkan dalam pembangunan aplikasi karena terbagi dalam model, *view*, dan *controller*.
4. Pada penelitian yang dilakukan oleh **Jannah dan Afirin (2015)** dalam jurnalnya yang berjudul **Sistem Informasi Absensi Haul Berbasis Web di Pondok Pesantren Muhyiddin Surabaya**. Dalam penelitian ini objek penelitiannya adalah peserta Haul pondok pesantren Muhyiddin, Hasil yang diperoleh adalah memungkinkan *user*

menggunakan sistem dalam waktu yang bersamaan, *database* pun dapat diakses bersamaan secara *online*, dan proses absen menjadi lebih cepat.