

BAB II

TINJAUAN PUSTAKA

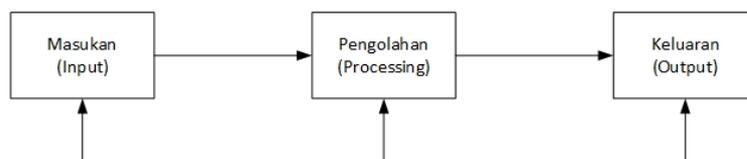
2.1. Teori Umum

Pada bab ini, penulis akan menjelaskan tentang teori umum yang dipakai dalam penulisan skripsi ini. Sesuai dengan judul skripsi “Rancang Bangun Aplikasi *E-Payroll* Pada CV Era Metal *Fabrication*”, berikut akan diuraikan konsep kajian pustaka untuk menjelaskan pengertian variable tersebut agar memudahkan pelaksanaan penelitian untuk tahap selanjutnya.

2.1.1. Sistem

Sistem adalah sekelompok unsur-unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Herliana & Rasyid 2016:42).

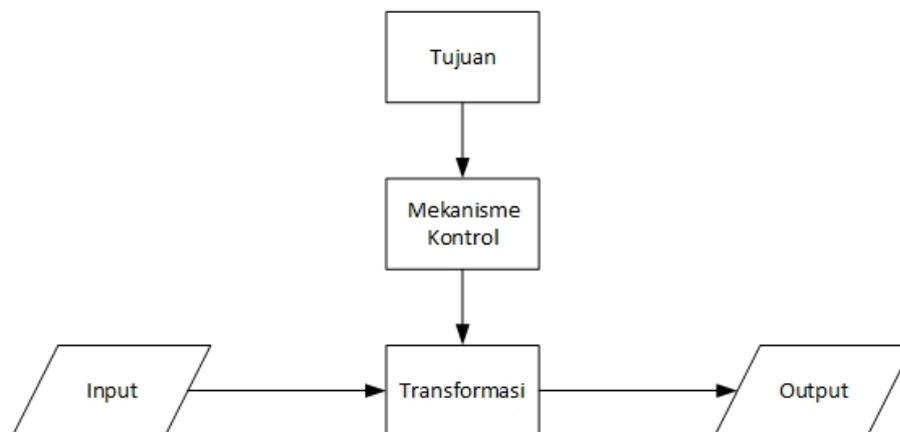
Sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*). Ciri pokok sistem ada empat, yaitu sistem itu beroperasi dalam suatu lingkungan, terdiri atas unsur-unsur, ditandai dengan saling berhubungan dan mempunyai satu fungsi atau tujuan utama.



Sumber: (Herliana & Rasyid 2016:42)

Gambar 2.1. Model Sistem

Gambar di atas menunjukkan bahwa sistem atau pendekatan sistem minimal harus mempunyai empat komponen, yakni masukan, pengolahan, keluaran dan balikan atau *control*. Sumber daya mengalir dari elemen *output* dan untuk menjamin prosesnya berjalan dengan baik maka dihubungkan dengan mekanisme *control*. Untuk lebih jelasnya elemen sistem tersebut dapat digambarkan dengan model sebagai berikut:



Sumber: (Herliana & Rasyid 2016:42)

Gambar 2.2. Model Hubungan Elemen-Elemen Sistem

Secara ringkas menjelaskan bahwa sistem adalah:

1. Komponen-komponen yang saling berhubung satu sama lain.
2. Suatu keseluruhan tanpa memisahkan komponen pembentuknya.
3. Bersama-sama dalam mencapai tujuan.
4. Memiliki *input* dan *output* yang dibutuhkan oleh sistem lainnya.
5. Terdapat proses yang mengubah *input* menjadi *output*.
6. Menunjukkan adanya entropi.
7. Memiliki aturan.
8. Memiliki subsistem yang lebih kecil.

9. Memiliki deferensiasi antar subsistem.
10. Memiliki tujuan yang sama meskipun mulainya berbeda.

Karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya:

1. Batasan (*boundary*), penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*), segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala dan *input* terhadap suatu sistem.
3. Masukan (*input*), sumber Daya (daya, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*), sumber daya atau produk (informasi, laporan, dokumen, tampilan layar computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*), Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan *input* menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*interface*), tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*), area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

2.1.2. Informasi

Informasi adalah kumpulan data yang relevan dan mempunyai arti yang menggambarkan suatu kejadian-kejadian atau kegiatan-kegiatan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan. Teori informasi lebih tepat disebut teori matematika komunikasi, yang mana konsep usia informasi menunjukkan hubungan interval informasi, jenis data dan penundaan pengolahan dalam menentukan usia informasi (Herliana & Rasyid, 2016:42).

Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai yang nyata atau dapat dirasakan manfaatnya dalam keputusan-keputusan yang akan datang. Yang lebih penting adalah mereka perlu mengerti informasi seperti informasi apa yang mereka butuhkan, untuk apa informasi tersebut digunakan serta bagaimana mutu dan kualitas informasi yang dapat membantu mereka mengidentifikasi dan memecahkan masalah, untuk memutuskan tindakan yang akan diambil saat ini maupun yang akan datang (Ali & Wangdra, 2017:10).

Tidak semua informasi berkualitas. Oleh karena itu, sudah seharusnya dilakukan penyaringan terhadap informasi yang beredar atau yang dapat ditangkap.

Kualitas informasi di tentukan oleh beberapa faktor yaitu:

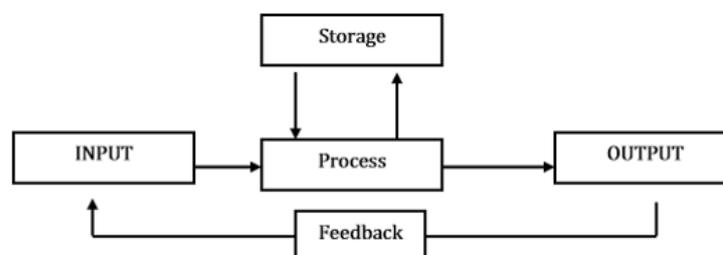
1. Keakuratan dan teruji kebenarannya.
2. Kesempurnaan informasi.

3. Tepat waktu.
4. Relevansi.
5. Mudah dan murah.

2.1.3. Sistem Informasi

Sistem informasi adalah sistem yang menyediakan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerima (Herliana & Rasyid, 2016:43).

Sistem informasi adalah suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan dan menyajikan sinergi organisasi pada proses (Fatta, 2017:9). Dengan demikian, sistem informasi berdasarkan konsep (*input, processing, output - IPO*) dapat dilihat pada gambar berikut:



Sumber: (Fatta, 2017:9).

Gambar 2.3. Konsep Sistem Informasi

Sistem informasi berbasis computer (CBIS) dalam suatu organisasi terdiri dari komponen-komponen berikut:

1. Perangkat keras, yaitu perangkat keras komponen untuk melengkapi kegiatan memasukan data, memproses data, dan keluaran data.
2. Perangkat lunak, yaitu program dan instruksi yang diberikan ke komputer.
3. *Database*, yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.
4. Telekomunikasi, yaitu komunikasi yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.
5. Manusia, yaitu personel dari sistem informasi, meliputi manajer, analis, programmer dan operator serta bertanggung jawab terhadap perawatan sistem.

Sistem informasi yang terdiri dari komponen-komponen di atas disebut dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*) dan blok Kendala (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sarannya.

1. Blok masukan, *input* mewakili data yang masuk ke dalam sistem informasi. *Input* di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan yang dapat berupa dokumen-dokumen dasar.
2. Blok model, blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data *input* dan data yang tersimpan

di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran, produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Blok teknologi, teknologi merupakan kotak alat (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.
5. Blok *database*, *database* merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
6. Blok kendali, pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

Sistem informasi berbasis *computer* (CBIS) biasanya dibedakan menjadi beberapa tipe aplikasi yaitu:

1. *Transaction Processing Systems (TPS)*.
2. *Management Information Systems (MIS)*.
3. *Decision Support Systems (DSS)*.
4. *Expert System And Artificial Intelligence (ES & AI)*.

2.1.4. Aliran Sistem Informasi (ASI)

Aliran sistem informasi sangat berguna untuk mengetahui permasalahan yang ada pada suatu sistem. Dari sini dapat diketahui apakah sistem informasi tersebut masih layak dipakai atau tidak, masih manual atau komputerisasi. Jika sistem informasinya tidak layak lagi maka perlu adanya perubahan dalam pengolahan datanya sehingga menghasilkan informasi yang cepat dan akurat serta keputusan yang lebih baik (Santoso, 2015: 26-27).

Tabel 2.1. Simbol Aliran Sistem Informasi

No	Simbol	Arti / Tujuan
1		Proses Komputerisasi
2		Proses Manual
3		Dokumen
4		Penyimpanan
5		Hard Disk
6		Penghubung
7		Arus Data

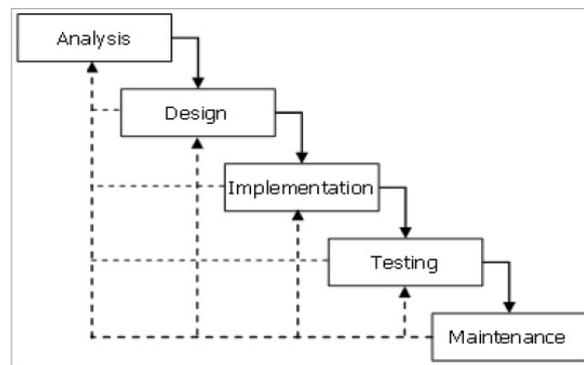
Sumber: (Santoso, 2015: 26-27)

2.1.5. SDLC (*System Development Life Cycle*)

SDLC adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (Rosa & Shalahuddin, 2017:24).

Upaya peningkatan kemampuan sistem dapat dilakukan oleh tim atau pihak manajemen manapun dalam perusahaan. Namun apabila sumber daya internal tidak

memungkinkan, perusahaan dapat menunjuk akuntan *public* untuk menangani pengembangan sistem tersebut. Tim tersebut dapat menyusun sistem baru memperbaiki ataupun memperluas sistem lama. Hasil pekerjaan ini akan diimplementasikan ke dalam perusahaan dan akan berlaku untuk beberapa tahun mendatang. Dan apabila terjadi perubahan lagi maka daur yang sama akan terulang.



Sumber: (Rosa & Shalahuddin, 2017:24)

Gambar 2.4. Daur SDLC

Ada beberapa model SDLC yang dapat digunakan dalam penerapan tahapan prosesnya, semuanya memiliki kelebihan dan kelemahan pada setiap model SDLC. Menurut (Rosa & Shalahuddin, 2017:27-39) model model SDLC adalah sebagai berikut:

1. Model *Waterfall*

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, disain, pengkodean, pengujian dan tahap pemeliharaan (*maintenance*)

2. Model *Prototype*

Model *prototype* dimulai dari pengumpulan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu buatlah program *prototype* agar

pelanggan lebih terbayang dengan apa sebenarnya diinginkan. *Program prototype* biasanya merupakan *program prototype* yang belum jadi.

2.1.6. UML (*Unified Modeling Language*)

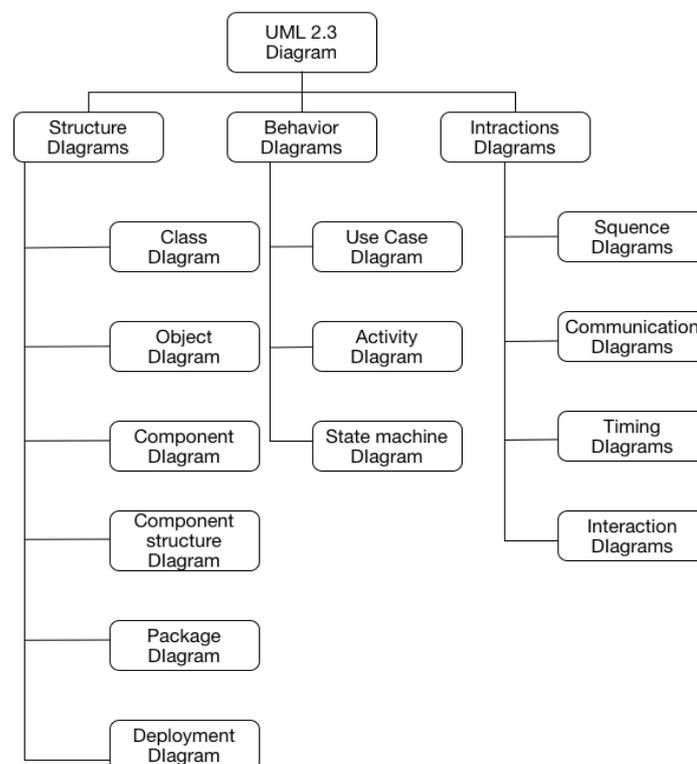
UML (*Unified Modelling Language*) adalah sebuah alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain. (Novita, 2015). Dalam UML terdiri dari 13 macam diagram yang terbagi dalam tiga kategori yaitu: (Rosa & Shalahuddin, 2017:121)

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

Menurut (Rosa & Shalahuddin, 2017) UML merupakan sintesis dari tiga metode analisis dan perancangan berbasis objek serta ditambah keunggulan metode-metode berorientasi objek lainnya (*Fision, Shlaer-Mellon, Coad-Yurdon*)

yang juga disintesakan dalam UML menawarkan pendekatan yang cukup baik serta digunakan cukup luas digunakan di industri perangkat lunak.

Diagram Unified Modeling Language (UML) adalah kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Diagram UML 2.3* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.5 dibawah ini:



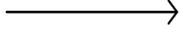
Sumber: (Rosa & Shalahuddin, 2017)

Gambar 2.5 Diagram UML

2.1.1.1. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut ini adalah simbol-simbol yang ada pada *class diagram*: (Rosa & Shalahuddin, 2017:122)

Tabel 2.2 Simbol *Class Diagram*

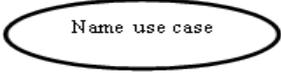
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
antarmuka / <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
asosiasi / <i>association</i> 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
asosiasi berarah / <i>directed association</i> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <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> 	semua bagian (<i>whole-part</i>)

Sumber: (Rosa & Shalahuddin, 2017: 123-124)

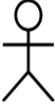
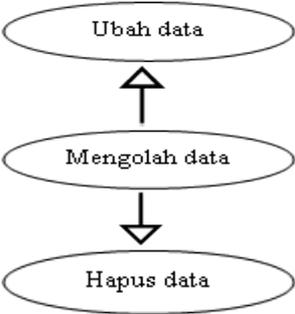
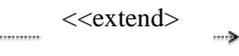
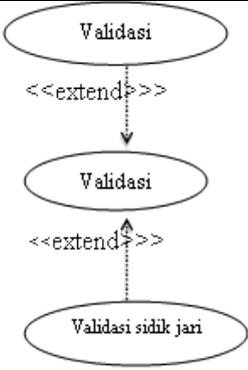
2.1.1.2. Use Case Diagram

Use Case Diagram merupakan pemodelan kelakuan (*behavior*) sistem informasi yang akan dibuat. Dan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada *use case diagram*: (Rosa & Shalahuddin, 2017:130)

Tabel 2.3 Simbol *Use Case*

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>

Tabel 2.3 Lanjutan

Simbol	Deskripsi
<p>Aktor/ <i>actor</i></p>  <p>nama aktor _____</p>	<p>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 <i>actor</i> belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum)</p>
<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <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, missal</p>
	 <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>

Tabel 2.3 Lanjutan

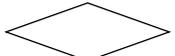
Simbol	Deskripsi
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p>.....</p> <p> <<include>></p> <p> <<uses>></p> 	<p>Relasi <i>use case</i> tambah ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i>. Ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut: • <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. Misal pada kasus: <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan</p>

Sumber: (Rosa & Shalahuddin, 2017: 131-133)

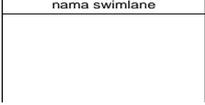
2.1.1.3. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Berikut adalah simbol-simbol yang ada pada *activity diagram*: (Rosa & Shalahuddin, 2017:134)

Tabel 2.4 Simbol Activity Diagram

Simbol	Deskripsi
<p>Status awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal</p>
<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja</p>
<p>Percabangan / <i>decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu</p>

Tabel 2.4 Lanjutan

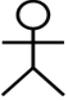
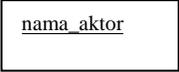
Simbol	Deskripsi
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: (Rosa & Shalahuddin, 2017: 134-135)

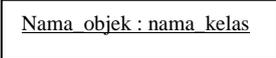
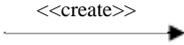
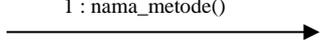
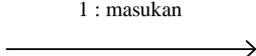
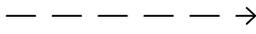
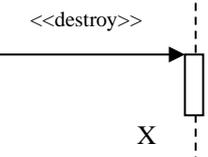
2.1.1.4. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada *sequence diagram*: (Rosa & Shalahuddin, 2017:137)

Tabel 2.5 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor  Atau 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor

Tabel 2.5 Lanjutan

Simbol	Deskripsi
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe <i>call</i> 	Menyatakan 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 mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
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 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, 2017: 138-139)

2.2. Teori Khusus

Pada bab ini, penulis akan menjelaskan tentang teori khusus yang dipakai dalam penulisan skripsi ini. Sesuai dengan judul skripsi “Rancang Bangun Aplikasi *E-Payroll* Pada CV Era Metal *Fabrication*”, berikut akan diuraikan konsep kajian pustaka untuk menjelaskna pengertian variabel tersebut agar memudahkan pelaksanaan penelitina untuk tahap selanjutnya.

2.2.1. Gaji

Gaji adalah sejumlah pembayaran kepada pegawai yang diberi tugas administratif dan manajemen yang biasanya ditetapkan secara bulanan (Jayanti & Siska Irian, 2014:37).

Gaji adalah pemberian pembayaran finansial kepada karyawan sebagai balas jasa untuk pekerjaan yang dilaksanakan dan sebagai motivasi pelaksanaan kegiatan di waktu yang akan datang (Mulyana & Mohamad Ridwan, 2017:128).

2.2.2. Sistem Penggajian

Pada umumnya penggajian karyawan atau pegawai dapat digolongkan menjadi tiga sistem (Rinawati & Lidya Sitanggang, 2014:6) , yaitu:

1. Sistem Skala Tunggal

Sistem skalah tunggal adalah suatu sistem penggajian yang memberikan gaji yang sama kepada pegawai yang berpangkat sama, dengan tidak memperhatikan sifat pekerjaan yang dilakukan dan beratnya tanggung jawab yang dipikul dalam melaksanakan pekerjaan itu.

2. Sistem Skala Ganda

Sistem skala ganda adalah sistem penggajian yang menentukan besarnya gaji bukan saja didasarkan pada pangkat tetapi didasarkan pula pada sifat pekerjaan yang dilakukan, prestasi kerja yang dicapai dan beratnya tanggung jawab yang dipikul dalam melaksanakan pekerjaan itu.

3. Sistem Skala Gabungan

Sistem penggajian yang ditetapkan di Indonesia ialah gabungan antara sistem skala tunggal dan skala ganda. Sistem penggajian ganda yaitu gaji diberikan berdasarkan sifat pekerjaan, prestasi yang dicapai, berat dan ringan tanggungjawab pekerjaan yang harus dipikul.

2.2.3. Database

Database adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses pengambilan keputusan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil (Jayanti & Siska Irian, 2014:37).

Database adalah kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras computer dan digunakan perangkat lunak untuk memanipulasinya (Riestiana & Sukadi, 2014:33).

Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali. Bahasa basis data (*database language*) adalah suatu cara untuk berinteraksi atau berkomunikasi antara pemakai dengan basis data yang diatur dalam bahasa khusus yang ditetapkan oleh perusahaan. *Database language* dipilah menjadi 3 yaitu;

1. *Data Definition Language (DDL)*,
2. *Data Manipulation Language (DML)*, dan
3. *Data Control Language (DCL)*.

2.2.4. SQL (*Structured Query Language*)

SQL adalah bahasa komputer Standart ANSI (*American National Standart Institute*) untuk mengakses dan memanipulasi sistem basis data. Pernyataan *SQL* digunakan untuk mengambil dan meng-update data dalam basis data. *SQL* bekerja dengna pemrograman basis data seperti *MS Access*, *DB2*, *Informix*, *MS SQLServer*, *Oracle* dan *Sybase* (Riestiana & Sukadi, 2014:34).

2.2.5. MySQL

MySQL adalah salah satu jenis *database server* yang sangat populer, hal ini disebabkan karena *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *datasenya* (Riestiana & Sukadi, 2014:33). *MySQL* bersifat *open source*, *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), bentuk *executable*-nya atau kode yang dapat dijalankan secara

langsung oleh sistem operasi. Kelebihan dari *MySQL* adalah menggunakan bahasa *query* standar yang dimiliki *SQL (Structure Query Language)*.

MySQL merupakan urutan kedua yang paling banyak digunakan sistem manajemen *database* relasional di dunia. Dengan kecepatan superior, kehandalan, dan kemudahan penggunaan, *MYSQL* telah menjadi pilihan untuk *Web* dan *Web 2.0*. Pengguna dapat menggunakan alat baris perintah yang disertakan atau menggunakan *MYSQL front end*, perangkat lunak desktop dan aplikasi *Web* yang membuat dan mengelola *database MYSQL*, backup data, memeriksa status, dan bekerja dengan catatan data (Mattoo & R. Kumar, 2016:20).

Menurut (Purbadian, 2015) tinjauan luas mengenai *MySQL (MyStructure Query Language)*, antara lain:

1. *MySQL* adalah suatu *Relational Database Management system (RDBMS)*
2. Perangkat lunak *MySQL* didistribusikan secara *open source*
3. *Database MySQL Server* sangat cepat, dapat dipercaya dan mudah digunakan
4. *Database Server MySQL* bekerja dalam lingkungan *client/server*
5. Dukungan terhadap perangkat lunak *MySQL* tersebar luas dan mudah ditemukan
6. *MySQL* mendukung penggunaan *database* dengan berbagai kriteria pengaksesan. Maksudanya adalah untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia
7. *MySQL* tersedia pada beberapa platform seperti Windows, Linux, Unix, dan lain-lain.

8. *MySQL* dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
9. *MySQL* mampu menangani basis data dalam skala besar.

2.2.6. HTML (*Hypertext Markup Language*)

HTML merupakan bahasa pemrograman yang digunakan untuk menampilkan dokumen pada *browser* dalam sebuah *Web*. *HTML* bertujuan untuk mendefinisikan struktur dokumen *Web* dan tata letak tampilan. *HTML* menggunakan beragam tag dan atribut. Sebuah dokumen *HTML* ditandai dengan tag awal `<HTML>` dan diakhiri dengan tag `</HTML>` (Jayanti & Siska Irian, 2014:37).

HTML adalah sebuah bahasa pemrograman yang berbentuk skrip-skrip yang berguna untuk membuat sebuah halaman *Web*. *HTML* dapat dibacakan oleh berbagai platform seperti: *Windows*, *Linux*, *Macintosh*. Kata “*Markup Language*” pada *HTML* menunjukkan fasilitas yang berupa tanda tertentu dalam skrip *HTML* dimana kita bisa mengatur judul, garis, *table*, gambar dan lain-lain dengan perintah yang telah ditentukan pada elemen *HTML*. *HTML* sendiri dikeluarkan oleh *W3C* (*World Wide Web Consortin*), setiap terjadi perkembangan level *HTML* harus dievaluasi ketat dan disetujui oleh *W3C* (Hasanah, 2013:41).

Hyper Text Markup Language (HTML) adalah bahasa markup standar yang digunakan untuk membuat halaman *Web*. *HTML* ini terutama dikembangkan oleh fisikawan Tim Berners-Lee di CERN pada tahun 1980 (Roy & Dibyendu Roy, 2016:62).

HTML (*Hypertext Markup Language*) adalah suatu bahasa yang digunakan untuk menulis halaman *web*. HTML dirancang untuk digunakan tanpa tergantung pada suatu *platform tertentu* atau *platform independent* (Sutarman, 2007; Rahmad *et al*, 2013). HTML (*Hypertext Markup Language*) adalah sebuah bahasa pemrograman berbentuk skrip-skrip yang berguna untuk membuat sebuah halaman *web*. HTML dapat dibaca oleh berbagai *platform* seperti Windows, Linux, Macintosh. Kata "*Markup Language*" pada HTML menunjukkan fasilitas yang berupa tanda tertentu dalam skrip HTML dimana kita bisa mengatur judul, garis, tabel, gambar dan lain-lain dengan perintah yang telah ditentukan pada elemen HTML. HTML sendiri dikeluarkan oleh W3C (*World Wide Web Consortin*), setiap terjadi perkembangan level HTML harus dievaluasi ketat dan disetujui oleh W3C (Kadir, 2002; Hariadi, 2013).

HTML yang mempunyai kepanjangan dari *Hyper Text Markup Language* yang merupakan pondasi awal akan terciptanya suatu aplikasi *web*, karena HTML memiliki fungsi untuk membangun kerangka ataupun format *web* yang digunakan untuk menampilkan suatu informasi kedalam halaman *web* atau *browser* (Saputra, 2012).

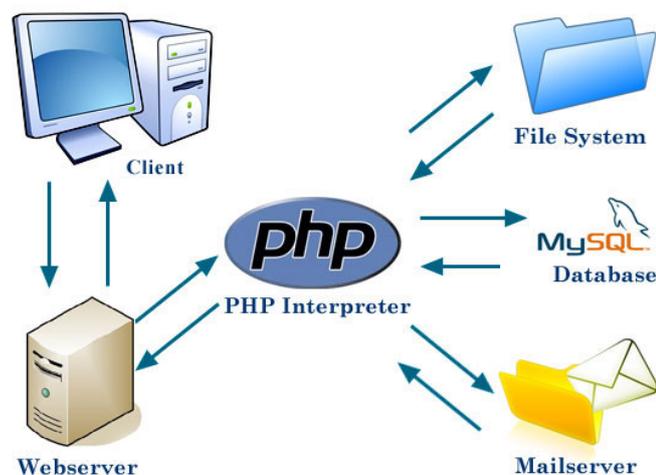
2.2.7. PHP (*Hypertext Preprocessor*)

PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang akan diberikan akan sepenuhnya akan dijalankan di *server* disertakan pada dokumen HTML. Pembuatan *web* ini

merupakan kombinasi antara PHP sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman *web* (Sunarfrihantono, 2002; Hendrianto, 2014). PHP merupakan bahasa pemrograman berbentuk *script* yang ditempatkan dalam *server* yang pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Hasil dari pengolahan akan dikirimkan ke klien, tempat pemakai menggunakan *browser*. Secara khusus PHP dirancang untuk membentuk *web* dinamis. Artinya ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya bisa menampilkan isi basis data ke halaman *web* (Lee, 2007; Renatha *et al.*, 2016).

PHP merupakan singkatan dari PHP (*hypertext preprocessor*). Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya yang dikirim ke klien, tempat pemakai menggunakan *browser*. Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi *tool* yang disebut "*personal home page*". Paket inilah yang menjadi cikal bakal PHP (Kadir, 2002; Rahmad & Purnama, 2013). Selain dapat digunakan untuk berbagai sistem operasi dan bersifat *open source*.

Prinsip kerja PHP mirip dengan kode HTML, hanya saja ketika berkas PHP yang diminta didapatkan oleh *web server*, isinya segera dikirim ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya berupa kode HTML ke *web server* dan selanjutnya *web server* menyampaikan ke *client*.



Sumber: (Renatha *et al.*, 2016)

Gambar 2.6 Skema Kerja PHP

2.2.8. CSS (*Cascading Style-Sheet*)

CSS adalah singkatan dari *Cascading Style-Sheet*, yaitu sebuah pengembangan atas kode *HTML* yang sudah ada sebelumnya. Dengan CSS, bisa menentukan sebuah struktur dasar halaman *Web* secara lebih mudah dan cepat, serta irit *size* (Hasanah, 2013:42).

2.2.9. Javascript

Javascript merupakan modifikasi dari bahasa *c++* dengan pola penulisan yang lebih sederhana. Interpreter bahasa ini sudah disediakan ASP ataupun *internet explorer*. Kelebihan *javascript* adalah berinteraksi dengan *HTML*, ini membolehkan pembuat *Web* untuk memasukkan *Web* mereka dengan kandungan-kandungan yang dinamik, menukar warna *background*, menukar *banner*, efek mouse, menu interaktif dan sebagainya (Hasanah, 2013:42).

2.2.10. JQuery

JQuery adalah *javascript library* yang dirancang untuk meringkas kode-kode *javascript*, sehingga dapat menyederhanakan penulisan skrip program, sesuai dengan slogan “*write less, do more*” (B, 2014:2). Kelebihan JQuery antara lain:

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

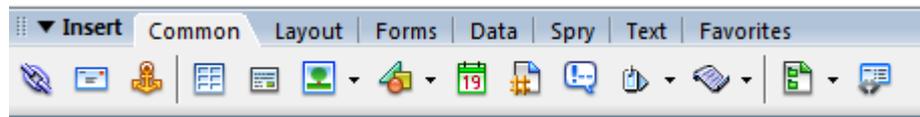
2.2.11. Adobe Dreamweaver CS 6

Adobe Dreamweaver CS6 merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah *website*, baik secara grafis maupun dengan menuliskan kode sumber secara langsung. (Team, 2017; Rahmad & Purnama, 2013).

Menurut (Prabantini, 2013) *Dreamweaver* adalah sebuah *HTML* editor profesional untuk mendesain *web* secara visual dan mengelola situs atau halaman *web*. *Dreamweaver CS6* memiliki kemampuan untuk mendesain *web*, menyunting kode, serta pembuatan aplikasi *web* dengan menggunakan berbagai bahasa pemrograman *web*, antara lain: *JPS, PHP, ASP* dan *ColdFusion*.

Menurut (Prabantini, 2013) komponen-komponen yang disediakan oleh ruang kerja *Dreamweaver CS6* antara lain adalah *InsertBar, Document Toolbar, Jendela Dokumen, Panel Group, Tag Selector, Property Inspector, dan Site Panel*. Penjelasan dari komponen-komponen ruang kerja *Dreamweaver CS6* adalah sebagai berikut:

1. *Insert Bar*, berisi tombol-tombol untuk menyisipkan berbagai macam objek seperti: *image*, tabel, dan layer ke dalam dokumen.



Sumber: (Prabantini, 2013)

Gambar 2.7 Tampilan *Insert Bar*

2. *Document Toolbar*, berisi tombol-tombol dan menu *pop-up* yang menyediakan tampilan berbeda dari jendela dokumen.



Sumber: (Prabantini, 2013)

Gambar 2.8 Tampilan *Document Toolbar*

3. Jendela Dokumen, berfungsi untuk menampilkan dokumen yang sedang kita kerjakan.



Sumber: (Prabantini, 2013)

Gambar 2.9 Tampilan Jendela Dokumen

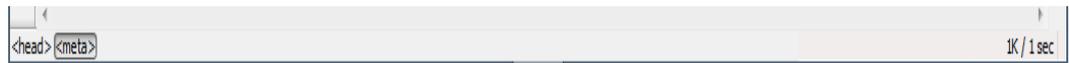
4. *Panel Group* adalah kumpulan panel yang saling berkaitan satu sama lain yang dikelompokkan di bawah satu judul.



Sumber: (Prabantini, 2013)

Gambar 2.10 Tampilan *Panel Group*

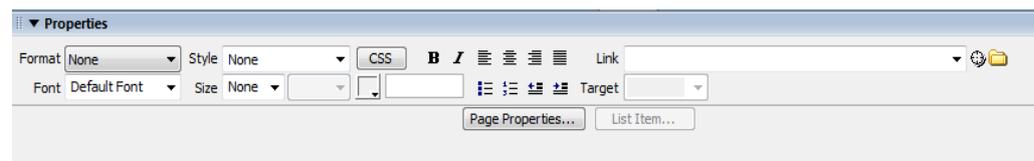
5. *Tag Selector* merupakan pada bagian bawah jendela dokumen yang berfungsi menampilkan hierarki *tag design view* yang sedang aktif.



Sumber: (Prabantini, 2013)

Gambar 2.11 Tampilan *Tag Selector*

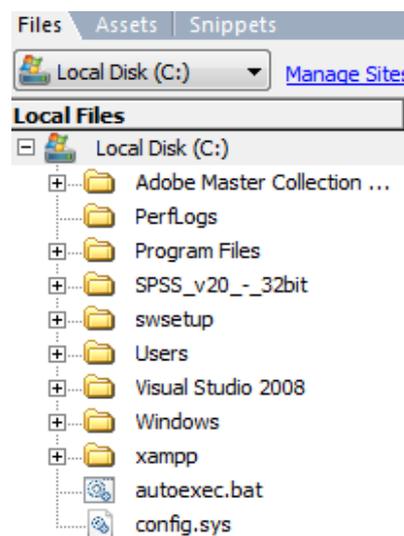
6. *Property Inspector*, digunakan untuk melihat dan mengubah berbagai properti objek atau teks.



Sumber: (Prabantini, 2013)

Gambar 2.12 Tampilan *Property Inspector*

7. *Site Panel*, digunakan untuk mengatur *file-file* dan *folder-folder* yang membentuk situs *web*.



Sumber: (Prabantini, 2013)

Gambar 2.13 Tampilan *Site Panel*

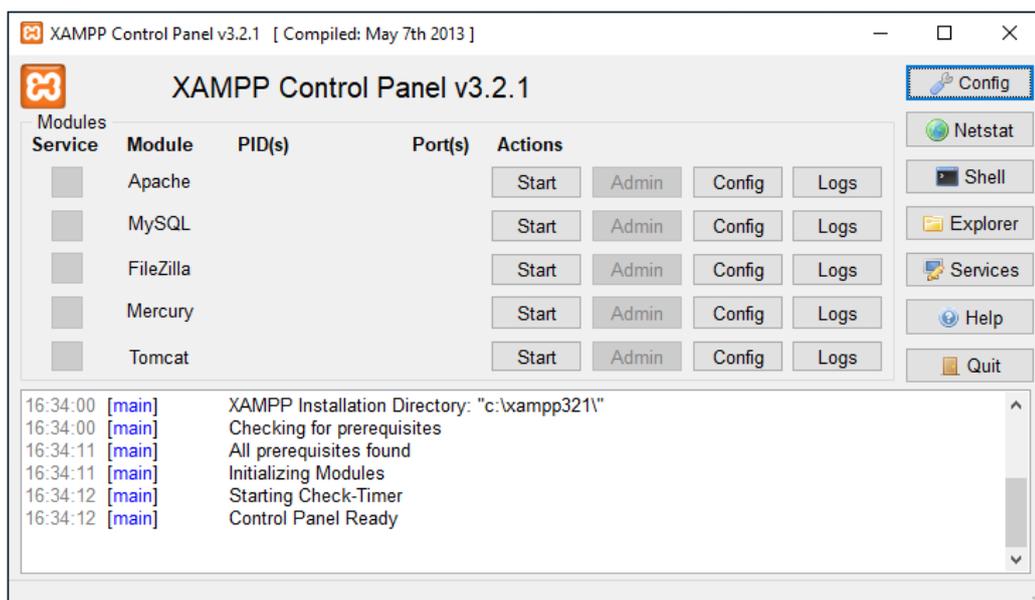
2.2.11. XAMPP

Menurut (Prabantini, 2013) XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan *project non-profit* yang di kembangkan oleh *Apache friends* yang didirikan Kai Oswald Seidler

dan Kay Vogelgesang pada tahun 2002, *project* mereka ini bertujuan mempromosikan penggunaan *Apache web server*.

Menurut (Prabantini, 2013) XAMPP merupakan salah satu paket instalasi *Apache*, *PHP* dan *MySQL instant* yang dapat kita gunakan untuk membantu proses instalasi ketiga produk tersebut, Adapun langkah-langkah pembuatan *database MySQL* dengan XAMPP sebagai berikut:

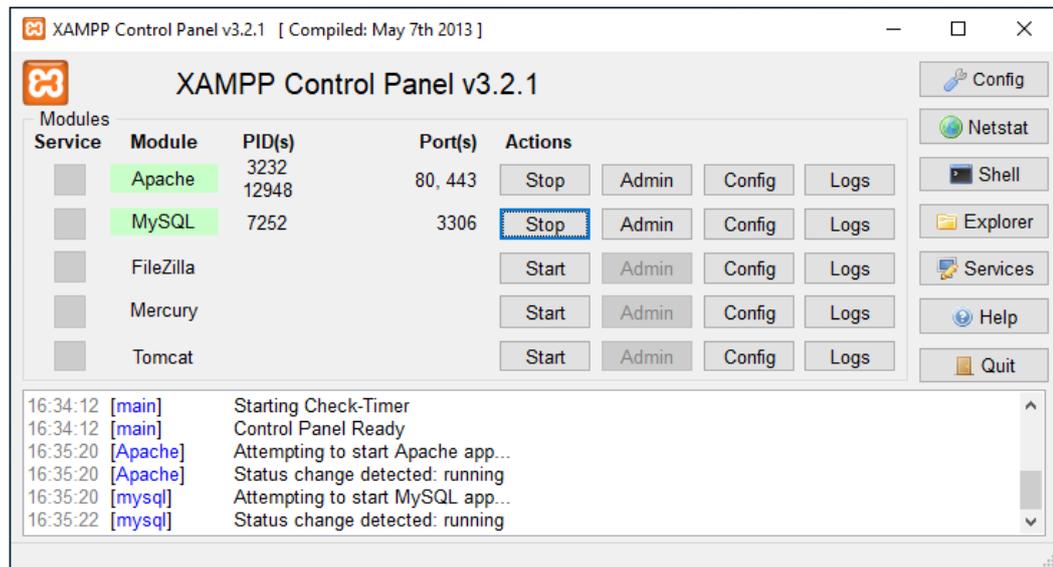
- 1 Buka XAMPP *control panel*, sehingga pada *Layer* akan muncul seperti pada gambar 2.14 dibawah ini.



Sumber: (Prabantini, 2013)

Gambar 2.14 XAMPP *Control Panel*

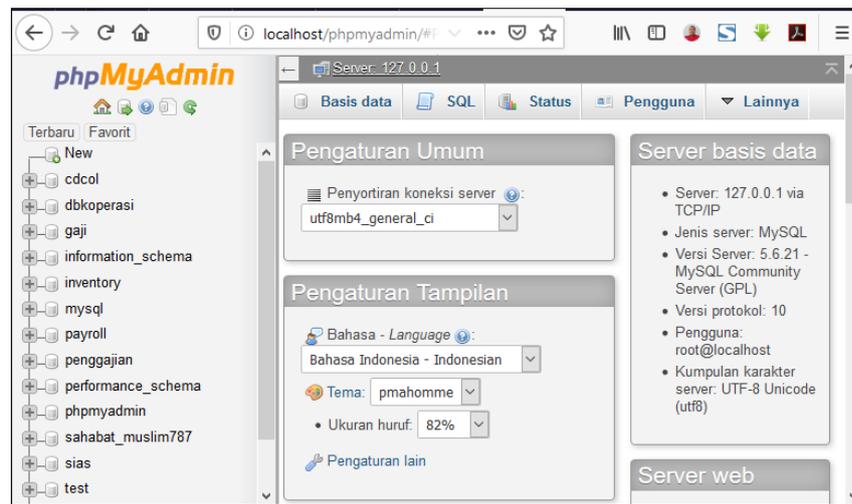
- 2 Kemudian klik tombol *start* yang ada pada *control panel* milik *Apache* dan *MySQL*, sehingga pada *Layer* akan muncul tulisan *Running* seperti pada gambar 2.15.



Sumber: (Prabantini, 2013)

Gambar 2.15 Apache dan MySQL Running

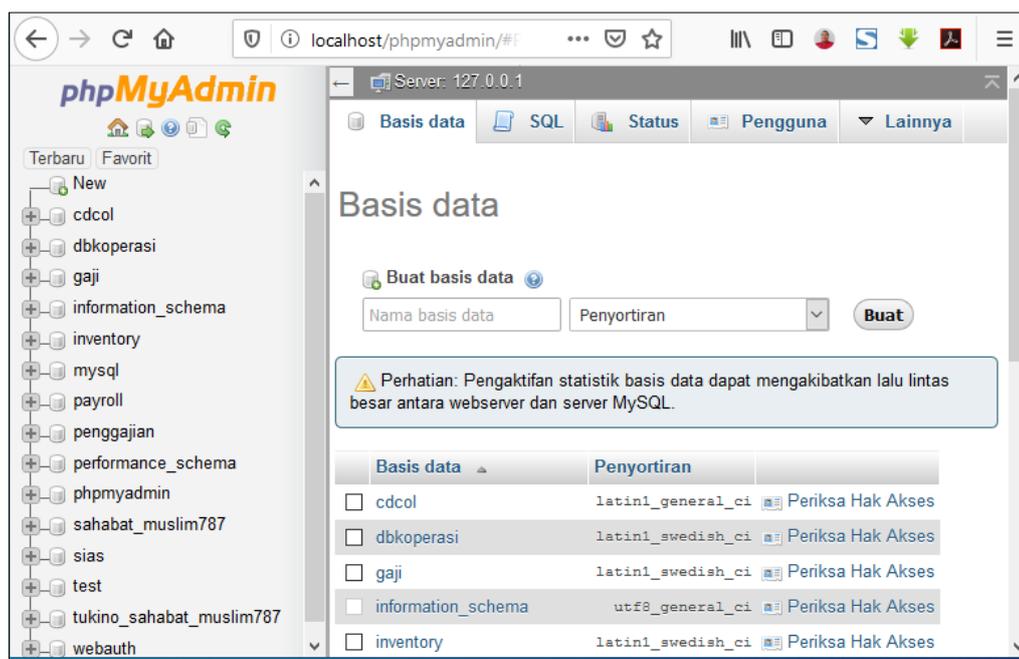
- 3 Setelah svc milik nya Apache dan MySQL diaktifkan langkah selanjutnya adalah dengan membuka browser internet, kemudian ketikkan localhost/phpmyadmin, sehingga pada Layer akan muncul seperti pada gambar 2.16.



Sumber: (Prabantini, 2013)

Gambar 2.16 Localhost/Phpmyadmin

- 4 Langkah selanjutnya adalah pembuatan *database*, isikan nama *database* pada kolom *create new database* lalu klik tombol *create*, sehingga pada *Layer* akan muncul seperti pada gambar 2.17.



Sumber: (Prabantini, 2013)

Gambar 2.17 Create New Database

2.2.12. Black Box Testing

Black box testing merupakan *testing* yang berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program (Mustaqbal, Firdaus, & Rahmadi, 2015:34). *Black box testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur *data* dan akses basis *data*.

4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.3. Penelitian Terdahulu

Berikut ini akan dikemukakan hasil penelitian terdahulu yang digunakan sebagai dasar di dalam penelitian ini. Adapun penelitian terdahulu yang penulis jabarkan adalah penelitian yang berhubungan dengan penelitian ini.

1. Dalam penelitian Fahmi Ferdiansyh (2012) yang berjudul pengembangan sistem informasi penggajian karyawan pada CV. DIVA menggunakan *PHP*. Penelitian ini rekayasa perangkat lunaknya menggunakan metode pendekatan *waterfall* (air terjun). Model ini mengumpulkan sebuah pendekatan kepada perkembangan software sistematis dan sekuensial. Mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode dan pengujian. Hasil penelitian ini dapat disimpulkan bahwa data-data yang berkaitan dengan penggajian karyawan dapat tersusun rapi dan sebuah *database*, dalam pengelolaan dan pencarian data, mempercepat dan mempermudah dalam memberikan informasi secara terbuka mengenai detail penggajian berupa slip gaji dan memberikan laporan penggajian baik kepada karyawan maupun perusahaan.
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.