

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam suatu perusahaan presensi menjadi sesuatu hal yang amat penting untuk bisa mencatat daftar hadir dari setiap karyawan yang nantinya hasil catatan kehadiran karyawan bisa dijadikan sebagai informasi yang akurat, cepat dan efisien oleh perusahaan dalam bentuk daftar presensi karyawan.

Pengembangan sistem presensi berbasis *web* dalam bidang usaha dan industri telah banyak berkembang. Penelitian ini berpegang pada beberapa penelitian sebelumnya yang berkaitan dengan sistem yang akan dibuat. Adapun beberapa penelitian terdahulu yang digunakan sebagai perbandingan penelitian ini

Dalam skripsi yang berjudul “Rancang Bangun Aplikasi Presensi Karyawan Berbasis Android dengan Metode *Extreme Programming* pada PT. Geschool Cerdas Mandiri” membuat sebuah aplikasi presensi karyawan di PT. Geschool Cerdas Mandiri untuk digitalisasi presensi karyawan. Aplikasi ini selain digunakan untuk presensi dapat juga digunakan untuk melakukan manajemen data jam kerja karyawan (Hutama, 2016).

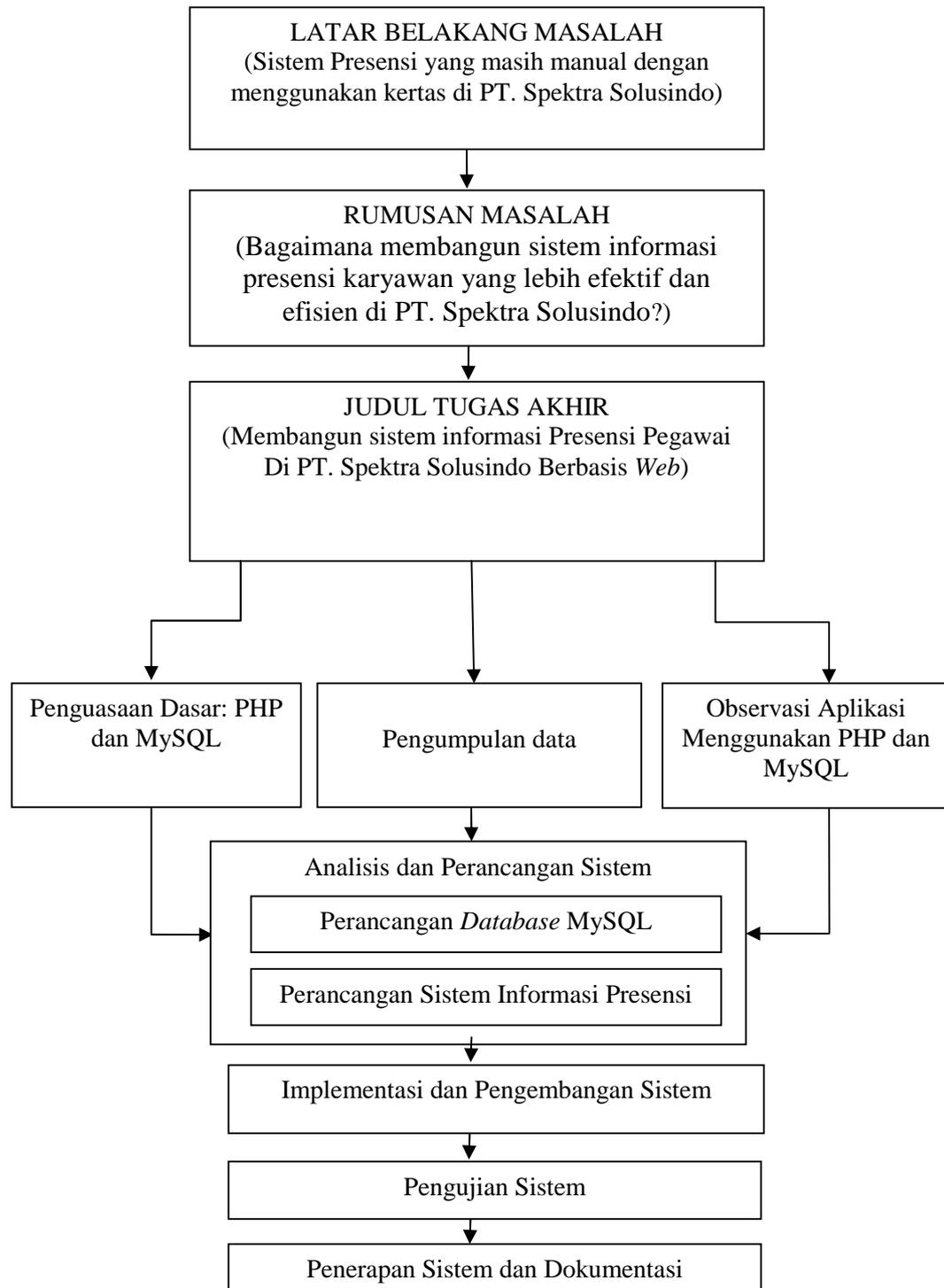
Dalam jurnalnya yang berjudul “Strategi Sistem Informasi Manajemen Presensi Karyawan (*Finger*) Terhadap Penyesuaian Meja Kerja Workshop (Stacking Division) Di Pt. Schneider Indonesia” membuat sebuah aplikasi presensi yang diterapkan di jurusan Sistem Informasi Universitas Sahid Surakarta. Aplikasi ini dapat melakukan monitoring presensi *fingerprint* untuk perkuliahan yang ada di jurusan Sistem Informasi (Erna Indriastiningsih dkk, 2013).

Dalam jurnal yang berjudul “Aplikasi Absensi Karyawan di Konsuil Bogor Menggunakan GPS” membuat aplikasi absensi dimana aplikasi akan bekerja melakukan absensi karyawan saat karyawan berada di area jangkauan aplikasi (Akbar dkk, 2012).

Berdasarkan ketiga penelitian di atas, maka penulis berinisiatif membuat aplikasi presensi karyawan berbasis *web*. target dari aplikasi ini yaitu karyawan di PT. Spektra Solusindo yang beroperasi di wilayah Bojonegoro, Jawa Timur.

2.2 Kerangka Pemikiran

Gambar kerangka pemikiran dapat dilihat pada Gambar 2.1



Gambar 2.1 Diagram Kerangka Pemikiran

Keterangan Diagram Kerangka Pemikiran Gambar 2.1 :

1) Latar Belakang Masalah

Identifikasi adanya masalah yang menjadi sebuah permasalahan pada PT. Spektra Solusindo di wilayah Bojonegoro. Permasalahan yang dibahas merupakan permasalahan presensi yang menjadi kendala bagi perusahaan dan dari pihak perusahaan belum memiliki solusi untuk menyelesaikan masalah tersebut.

2) Rumusan Masalah

Solusi yang baik untuk menyelesaikan sebuah permasalahan yang ada pada PT. Spektra Solusindo dapat berupa sebuah sistem informasi presensi.

3) Judul Tugas Akhir

Judul yang sekiranya tepat untuk menangani permasalahan yang ada dan diambil dari PT Spektra Solusindo.

4) Pengumpulan Data Tertulis dan Tak Tertulis

Semua data yang dibutuhkan dikumpulkan, baik melalui *interview* dengan karyawan, observasi di PT Spektra Solusindo.

5) Penguasaan Dasar (PHP, CSS, MySQL)

Aplikasi sederhana dibuat sebagai percobaan sebagai tujuan agar dapat lebih menguasai bahasa pemrograman PHP, CSS dan *database* MySQL sehingga hasilnya lebih maksimal.

6) Observasi Aplikasi

Aplikasi yang sudah ada dilakukan pengamatan, baik dari karya ilmiah, buku atau internet sehingga dapat dijadikan referensi untuk membangun sebuah aplikasi.

7) Analisis dan Perancangan Sistem

Sistem yang akan dibangun dianalisa dan dirancang seperti apa, bagaimana desainnya, dan apa saja isinya sehingga sistem ini dapat membantu memecahkan permasalahan yang ada pada PT. Spektra Solusindo.

Analisa dan Perancangan Sistem dibagi 2 bagian yaitu :

a) Perancangan *database* MySQL

Database dari data data yang telah didapatkan dibuat sesuai dengan kebutuhan sistem.

b) Perancangan aplikasi pengolahan data

Aplikasi dibuat dengan dasar database yang telah selesai dibuat sebelumnya.

8) Implementasi dan Pengembangan Sistem

Menerapkan dan mengembangkan sistem yang telah dibuat, mulai dari penulisan kode program, instalasi pada pengembang aplikasi dan pelatihan user.

9) Pengujian Sistem

Pengujian sistem ini dilakukan bertujuan untuk mengetahui jika ternyata masih ada kesalahan ataupun kekurangan pada sistem yang telah dibuat.

10) Penerapan Sistem dan Dokumentasi

Sistem telah siap digunakan setelah melewati tahap pengujian dan membuat dokumentasi dari keseluruhan kegiatan penyusunan tugas akhir.

2.3 Landasan Teori

2.3.1 Pengertian Sistem

Sutabri (2012:15), “sistem merupakan suatu bentuk integrasi antara satu komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam sistem tersebut.”

Sukanto dan Shalahuddin (2013:2), “sistem adalah kumpulan komponen yang saling berkaitan dan mempunyai satu tujuan yang ingin dicapai.

Dari kedua pengertian di atas, dapat disimpulkan sistem adalah kumpulan dari elemen-elemen atau bagian-bagian yang saling berinteraksi dan saling berhubungan membentuk satu kesatuan untuk mencapai satu tujuan.

2.3.2 Pengertian Informasi

Sutabri (2012:21), “informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah.”

Sutarman (2012:4), “informasi (*information*) adalah hasil kegiatan pengolahan data yang memberikan bentuk yang lebih berarti dari suatu kejadian.”

Dari kedua pengertian di atas, dapat disimpulkan informasi adalah data yang sudah diproses dan diolah menjadi lebih berarti dan memiliki nilai tambah.

2.3.3 Pengertian Sistem Informasi

Sutabri (2012:38), “sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.”

Ladjamudin (2013:13), “sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.”

Dari kedua pengertian di atas, dapat disimpulkan bahwa sistem informasi adalah suatu sistem yang dibuat dari berbagai komponen yang berhubungan untuk menghasilkan suatu informasi.

2.3.4 Karakteristik Sistem

Sutabri (2012:13) menjelaskan tentang karakteristik sistem sebagai berikut: Karakteristik Sistem adalah mode umum sebuah sistem terdiri dari dari input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Adapun karakteristik yang dimaksud adalah sebagai berikut:

a. **Komponen Sistem (*Components*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan.

b. **Batasan Sistem (*Boundary*)**

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya.

c. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

d. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*.

e. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

f. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasi menjadi keluaran yang berguna.

g. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik

2.3.5 Pengertian Presensi

Presensi atau daftar hadir adalah formulir berisi data identitas dan validasi dengan paraf atau tandatangan seseorang yang digunakan sebagai bukti kehadiran/keikutsertaan seseorang tersebut dalam suatu acara/kegiatan (Badunk, 2013). Terkadang di sebuah perusahaan seseorang atau karyawan melakukan presensi saat datang dan pulang dari kerja. Selain itu, setiap perusahaan atau instansi memiliki sistem presensi yang berbeda-beda.

Sistem presensi yang terkomputerisasi sudah banyak dikembangkan sebelumnya dengan perangkat lunak dan metode yang berbeda-beda. Sistem yang dikembangkan juga bervariasi, berbasis *website*, *fingerprint*, maupun *desktop*.

Penelitian yang dilakukan oleh (Muhammad, dkk. 2013) menjelaskan bahwa sistem presensi MONIKUL V.1 yang telah dikembangkan dapat melakukan monitoring absensi *fingerprint* untuk perkuliahan yang ada pada

Jurusan Sistem Informasi ITS. Sistem ini juga dapat melakukan pengolahan jadwal belajar mengajar terkait presensi *fingerprint*. Selain itu pemanfaatan teknologi ini dapat meringankan beban TU jurusan untuk rekapitulasi presensi saat akhir semester. Melalui uji coba performa menggunakan *apache benchmark* pada lingkungan uji coba, MONIKUL V.1 dapat menangani *request* lebih dari 1000 *user* berbeda secara bersamaan.

Sistem ini memungkinkan semua civitas yang tergabung di dalamnya dapat melihat dan merekap absensinya. Civitas yang dimaksud adalah *user* yang tidak lain adalah Dosen, Mahasiswa, dan Admin. *User* tersebut memiliki *previlage* berbeda tergantung jabatan *user* tersebut.

Pengertian presensi menurut (Ayu Pintasari, 2014) adalah suatu pendataan kehadiran, bagian dari pelaporan aktifitas suatu institusi, atau komponen institusi itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan. Presensi juga merupakan suatu cara untuk mengetahui sejauh mana tingkat disiplin kerja pegawai, apakah pegawai tersebut bisa mentaati peraturan yang diterapkan atau tidak. Pengelolaan presensi adalah pengelolaan sistem yang sudah terkomputerisasi yang mana perusahaan tidak mengalami masalah dalam memasukkan data, mengirim data, merevisi data maupun dalam melihat ulang data data yang sudah lama.

Adapun tujuan dari presensi tersebut adalah sebagai berikut:

- 1) Untuk melihat kehadiran pegawai
- 2) Untuk meningkatkan kedisiplinan pegawai
- 3) Untuk meningkatkan semangat kerja pegawai
- 4) Untuk mengetahui keadaan bawahan dihari kerja

2.3.6 Pengertian Pegawai

Pengertian pegawai menurut (Mardiasmo, 2011) adalah orang pribadi yang bekerja pada pemberi kerja baik sebagai pegawai tetap atau pegawai tidak tetap/tenaga kerja lepas berdasarkan perjanjian atau kesepakatan kerja baik secara

tertulis maupun tidak tertulis untuk melaksanakan suatu pekerjaan dalam jabatan atau kegiatan tertentu dengan memperoleh imbalan yang dibayarkan berdasarkan periode tertentu, penyelesaian pekerjaan atau ketentuan lain yang ditetapkan pemberi kerja, termasuk orang pribadi yang melakukan pekerjaan dalam jabatan negeri atau badan usaha milik negara.

2.3.7 Pengertian Perseroan Terbatas (PT)

Purba (2015:20) menjelaskan Perseroan Terbatas adalah badan hukum yang merupakan persekutuan modal, didirikan berdasarkan perjanjian, dan melakukan kegiatan usaha dengan modal dasar yang seluruhnya terbagi dalam saham. Sebagai badan hukum, sebuah PT dianggap layaknya orang- perorangan secara individu yang dapat melakukan perbuatan hukum sendiri, memiliki harta kekayaan sendiri, dan dapat menuntut serta dituntut di muka pengadilan. Untuk menjadikannya sebagai badan hukum PT, sebuah perusahaan harus mengikuti tata cara pembuatan, pendaftaran, dan pengumuman sebagaimana yang diatur dalam Undang-Undang Nomor 40 Tahun 2007 Tentang Perseroan Terbatas (UU PT).

2.3.8 Pengertian Website

Asropudin (2013:109), “*website* adalah sebuah kumpulan halaman (*webpages*) yang diawali dengan halaman muka (*homepage*) yang berisikan informasi, iklan, serta program interaksi.”

Nilasari (2014:2), “*website* atau juga sering disebut situs merupakan kumpulan halaman web yang dijalankan dari satu alamat *web domain*.”

Dari kedua pengertian di atas, dapat disimpulkan bahwa *website* adalah suatu halaman yang dapat menampilkan suatu informasi dan melakukan interaksi dengan penggunanya.

2.3.9 Pengertian Basis Data (Database)

Ichwan (2011:7), “Basis data (*database*) merupakan sebuah bangunan data yang terdiri dari kumpulan table yang saling berelasi satu dengan yang lainnya untuk memberikan suatu batasan tertentu.”

Kadir (2013:15), “*database* adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data.”

Dari kedua pengertian di atas, dapat disimpulkan bahwa basis data (*database*) adalah suatu kumpulan table yang saling berelasi dan saling berkaitan yang disusun secara sistematis untuk memudahkan penyimpanan dan pengaksesan data.

2.3.10 Sekilas Tentang PHP (*Hypertext Preprocessor*)

2.3.10.1 Pengertian PHP (*Hypertext Preprocessor*)

Kadir (2013:120), “PHP merupakan bahasa pemrograman yang ditujukan untuk membuat aplikasi *web*.”

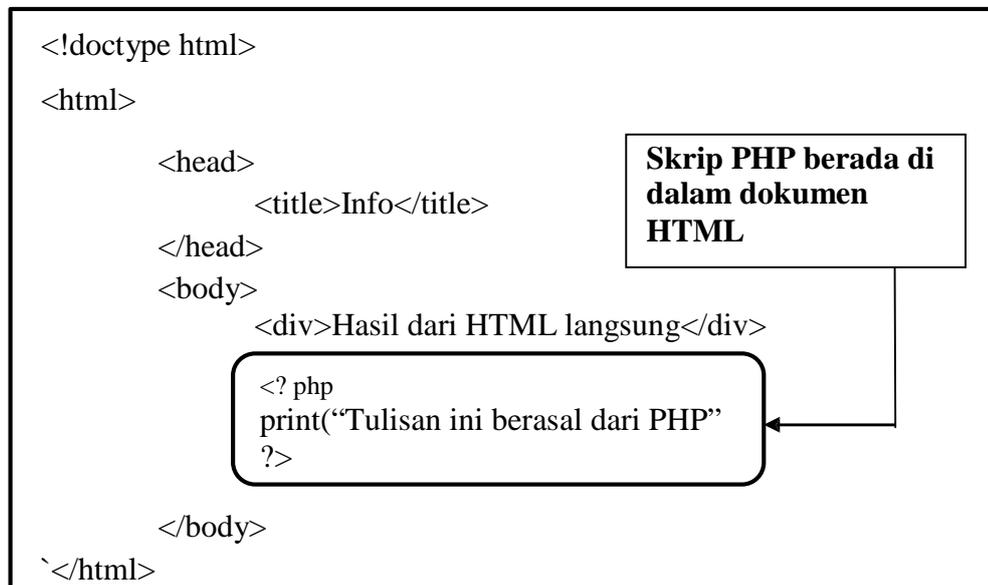
Nugroho (2013:153), “PHP adalah bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis web (*website, blog*, atau aplikasi *web*).”

Prasetio (2014:122), “ PHP (*Hypertext Preprocessor*) adalah bahasa *script* yang ditanam di sisi server.

Dari ketiga pengertian di atas, dapat disimpulkan bahwa PHP adalah bahasa pemrograman yang digunakan untuk membuat aplikasi web sehingga web menjadi lebih dinamis dan ditanam di sisi server.

2.3.10.2 Skrip PHP (*Hypertext Preprocessor*)

Kadir (2013:123) menjelaskan tentang posisi skrip PHP adalah melekat pada dokumen HTML. Dengan demikian, di dokumen HTML bisa disisipkan skrip PHP. Namun, konsekuensinya dokumen HTML harus disimpan dengan ekstensi berupa.php. Skrip PHP dapat dilihat pada Gambar 2.2.



Gambar 2.2 Skrip PHP (*Hypertext Preprocessor*)

2.3.10.3 Tipe Data PHP (*Hypertext Preprocessor*)

Kadir (2013:129) menjelaskan bahwa terdapat 5 tipe data PHP. Tipe data PHP dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tipe Data PHP (*Hypertext Preprocessor*)

Tipe Data Skalar	Keterangan	Contoh
Integer	Bilangan bulat	123
Float	Bilangan real berpresisi tunggal	56.8
Double	Bilangan real berpresisi ganda	56.89999777
String	Deretan karakter	"Sukses!"
Boolean	Menyatakan nilai <i>true</i> (benar) atau <i>false</i>	<i>True, false</i>

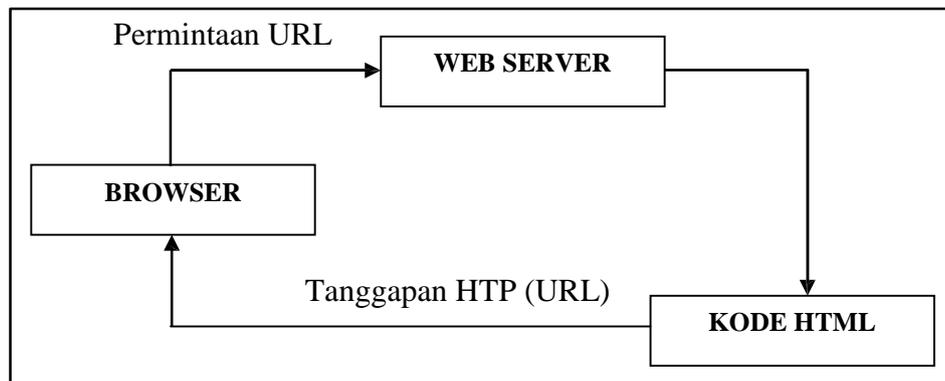
(Sumber:Kadir, 2013:129)

2.3.10.4 Konsep Kerja PHP (*Hypertext Preprocessor*)

Saputra (2013:4) menjelaskan bahwa konsep kerja PHP adalah sebagai berikut:

- a. *Server* membaca permintaan dari *client/browser*.
- b. Kemudian dilanjutkan untuk mencari halaman/page pada *server*.
- c. *Server* melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/*page*.
- d. Selanjutnya hasil modifikasi tersebut akan dikembalikan kepada *client/browser*.

Gambar konsep PHP dapat dilihat pada Gambar 2.3



(Sumber:Saputra, 2013:5)

Gambar 2.3 Konsep Kerja PHP (*Hypertext Preprocessor*)

2.3.11 Pengertian *WampServer*

WampServer adalah aplikasi terpaket yang berisi PHP, MySQL, dan Apache untuk menyimpan dan menerjemahkan database. menjadi sebuah halaman *website*”.

WampServer merupakan sebuah aplikasi yang dapat menjadikan komputer maupun laptop anda menjadi sebuah *server* atau bias dikatakan *server offline*”.

Berdasarkan kutipan diatas dapat disimpulkan bahwa wampserver adalah aplikasi yang dapat mencakup MySql, PHP, dan Apache yang dapat menjadikan komputer atau laptop menjadi sebuah *server*.

2.3.12 Pengertian MySQL

Ichwan (2011:23) menjelaskan, “MySQL adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*).”

Nugroho (2013:26), “MySQL adalah program berbasis DOS, perintah dasarnya adalah SQL (*Structured Query Language*) yang dapat diakses melalui jendela *DOS Prompt* atau *Command Prompt*.”

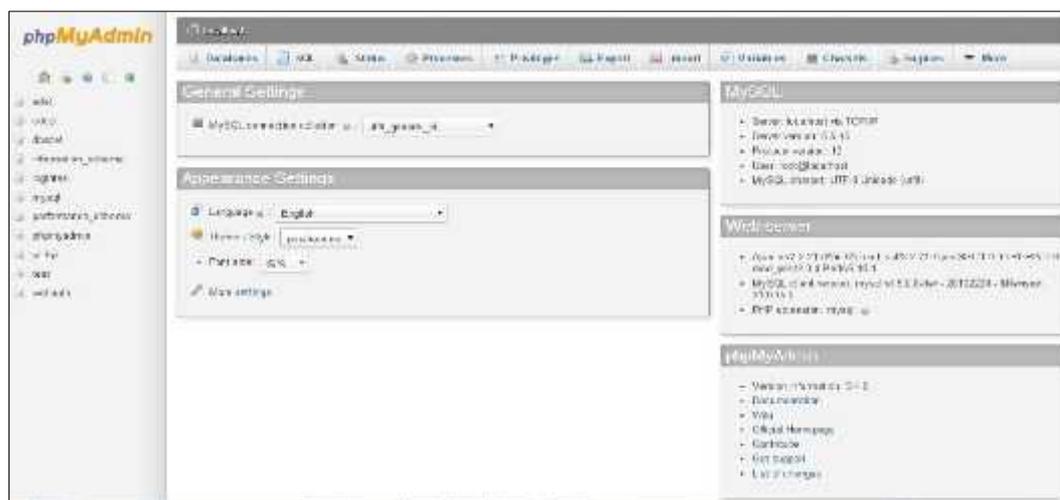
Dari kedua pengertian di atas dapat disimpulkan MySQL adalah suatu program yang digunakan untuk mengatur sebuah sistem basis data yang dibuat berdasarkan perintah SQL yang dijalankan melalui program *Command Prompt*.

2.3.13 Pengertian PHPMYAdmin

Nugroho (2012:6), “*PHPMYAdmin* adalah *MySQL Client* berbasis web, atau program yang dapat kita gunakan untuk mengakses *database MySQL* melalui *browser*.”

Hadi (2013:25), “*PHPMYAdmin* merupakan sebuah aplikasi *open source* yang berfungsi untuk memudahkan pengelolaan *database MySQL*.”

Dari kedua pengertian di atas dapat disimpulkan *PHPMYAdmin* adalah aplikasi yang digunakan untuk mengakses serta memudahkan dalam mengelola database. Tampilan *PHPMYAdmin* dapat dilihat pada Gambar 2.4



Gambar 2.4 PHPMYAdmin

2.3.14 *HyperText Markup Language (HTML)*

Prasetio (2014:93) menjelaskan, HTML merupakan kependekan dari *Hyper Text markup Language* yang berisikan tag-tag markup guna memberitahukan browser bagaimana harus menampilkan sebuah halaman. *Hypertext* berarti anda dapat membuat sebuah *link* pada sebuah halaman *web* yang mengarahkan pengunjung ke halaman *web* lain pada *internet*. Itu berarti informasi pada *web* bisa diakses dari berbagai arah. Universalitas maksudnya bahwa dikarenakan dokumen HTML disimpan sebagai *Text Only file*, secara virtual setiap komputer dapat membaca dan membuka halaman *web* tersebut.

2.3.15 *Cascading Style Sheets (CSS)*

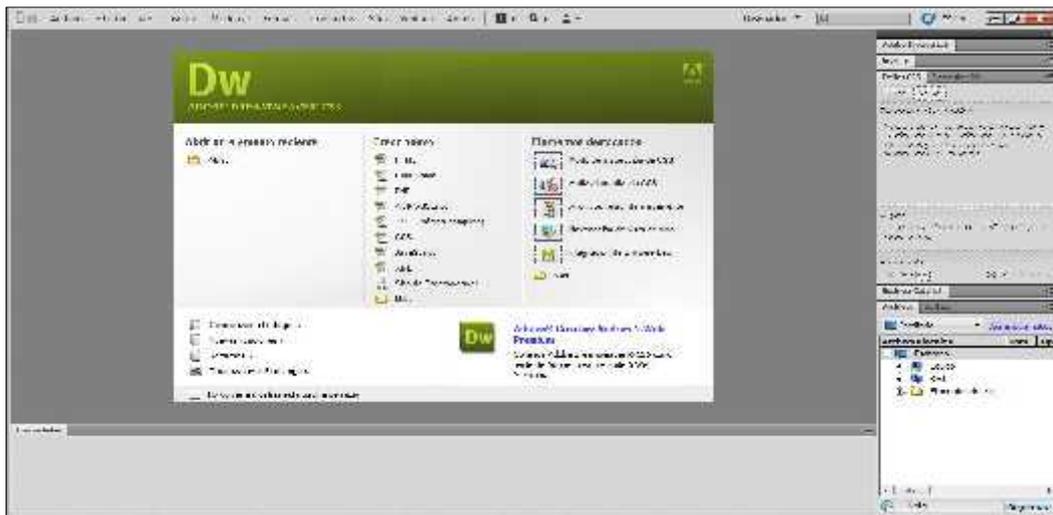
Menurut Richard, York (2011:3), *Beginning Cascading Style Sheet (CSS)* merupakan bahasa yang dirancang untuk menggambarkan penampilan dokumen yang ditulis dalam bahasa pemrograman seperti HTML, ide dari W3C untuk menghadapi kebutuhan manufaktur dari *browser* dengan memperkenalkan lebih dari sekedar HTML *tags*, yaitu sebuah *physical formatting tags* yang baru (seperti *font* atau *table tags*). CSS merupakan sebuah penyelesaian, menyediakan fitur *page layout* yang diinginkan oleh *web authors* dengan menambahkan elemen dari CSS ke HTML *tags* yang ada. CSS menguasai banyak *logical structure* dari sebuah *web page* ketika mengirimkan berbagai fitur *page layout* agar mudah dimengerti dan mempunyai efek yang lebih *powerful*.

2.3.16 *Bootstrap*

Menurut Spurlock (2013:1), *bootstrap* adalah sebuah *framework* tampilan depan *open source* yang dikembangkan oleh *twitter* yang dapat digunakan oleh *front-end developer* untuk membuat suatu *website* yang *responsive* untuk berbagai resolusi layar *device*. Adapun *framework* tampilan depan ini terdiri dari CSS dan *Javascript* yang sudah diatur sedemikian rupa agar sesuai standar *website* profesional yang telah banyak beredar. *Bootstrap* dapat di-*download* langsung di website resmi *bootstrap*, yaitu <http://getbootstrap.com>.

2.3.17 Sekilas Tentang *Adobe Dreamweaver CS5*

Menurut Sadeli (2011:2), “*Adobe Dreamweaver CS5* adalah suatu perangkat lunak *web editor* keluaran *Adobe System* yang digunakan untuk membangun dan mendesign suatu *website* dengan fitur-fitur yang menarik dan kemudahan dalam penggunaannya”. Tampilan awal *Adobe Dreamweaver* dapat dilihat pada Gambar 2.5.



Gambar 2.5 *Adobe Dreamweaver CS5*

2.3.18 *Unified Modelling Language (UML)*

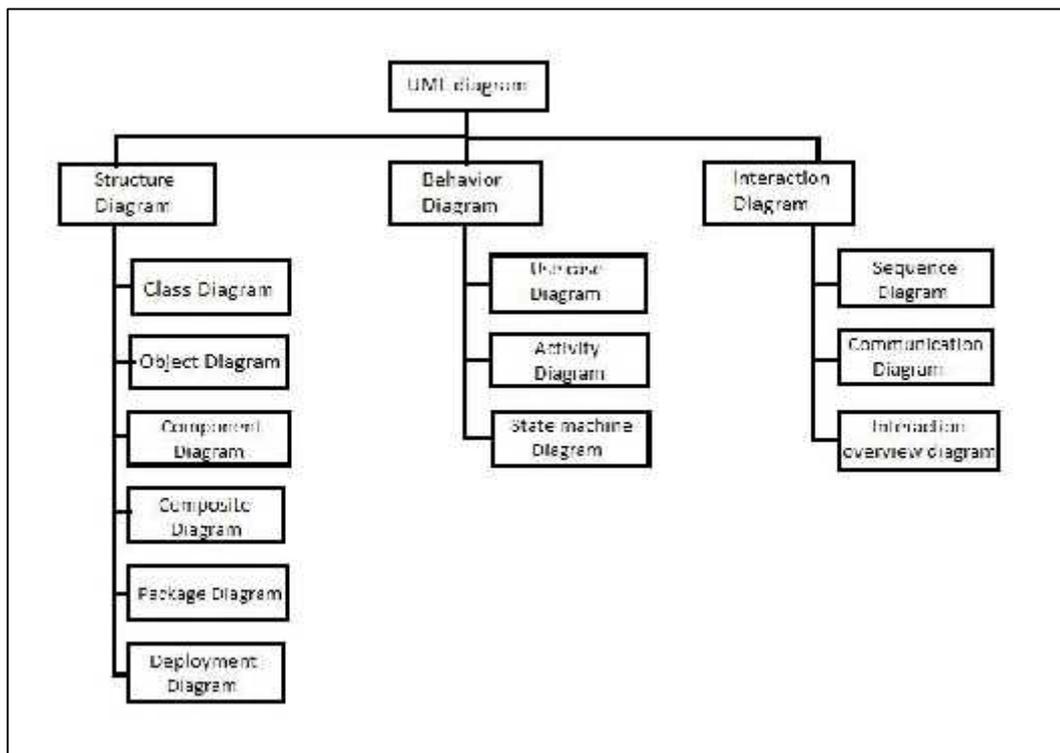
Menurut Sukamto dan Shalahuddin (2014:133), “*UML (Unified Modelling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”

Menurut Widodo dan Herlawati (2011:6), “*UML* singkatan dari *Unified Modeling Language* yang berarti bahasa permodelan standar.”

Asropuddin (2013:102), “*UML* singkatan dari “*Unified Modeling Language*” adalah bahasa pemrograman yang digunakan untuk perangkat lunak berorientasi objek.

2.3.18.1 Macam-macam Diagram UML

Menurut Sukamto dan Salahuddin (2014:140), UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada Gambar 2.6.



Gambar 2.6 Macam-macam Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. Structure Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. Behavior Diagram

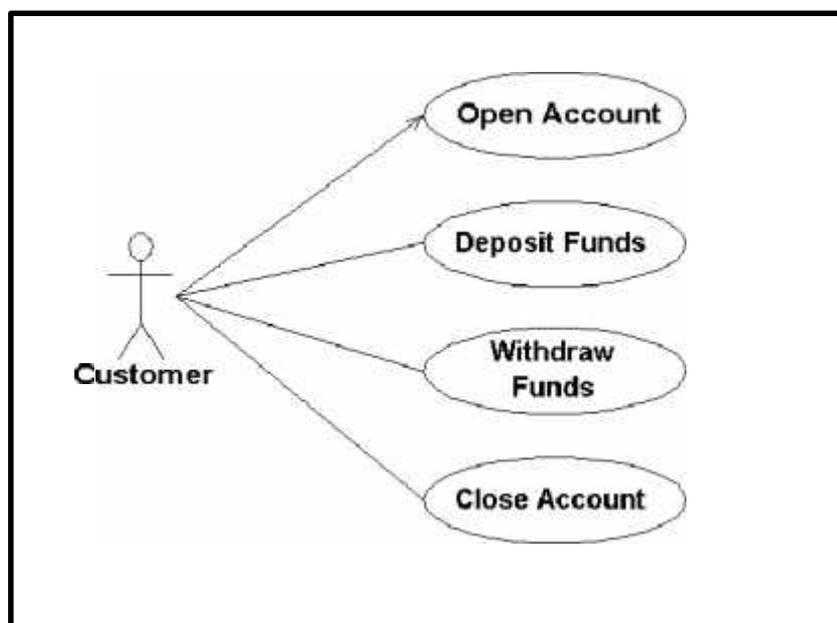
Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. Interaction Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem

2.3.18.2 Use Case Diagram

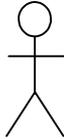
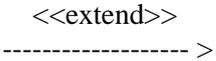
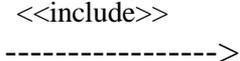
Menurut Sukamto dan Shalahuddin (2014:155), “*Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. *Use Case* juga merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan. Contoh *Use Case Diagram* dapat dilihat pada Gambar 2.7



Gambar 2.7 Contoh *Use Case Diagram*

Macam-macam simbol diagram *Use Case Diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-simbol Diagram *Use Case*

No.	Simbol	Deskripsi
1.	<p><i>Use Case</i></p>  <p>nama use case</p>	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> .
2.	<p>Aktor / Actor</p>  <p>nama aktor</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<p>Asosiasi / Association</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.
4.	<p>Ekstensi / Extend</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 itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5.	<p>Generalisasi /</p>  <p><i>Generalization</i></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.
6.	<p>Menggunakan / Include / Uses</p> 	Relasi <i>use case</i> tambahan 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. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

Sumber: Buku Karangan Rosa A.S dan M. Shalahuddin (2014:156)

2.3.18.3 Class Diagram

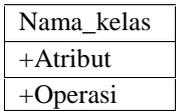
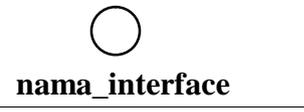
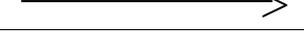
Widodo dan Herlawati (2011:3) menjelaskan, “*Class diagram* adalah penggambaran satu set objek yang memiliki atribut dan *behavior* yang sama.”

Menurut Sukamto dan Shalahuddin (2014:141), bahwa diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- a. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Macam-macam simbol diagram *Class Diagram* dapat dilihat pada Tabel 2.3

Tabel 2.3 Simbol-simbol Diagram Kelas

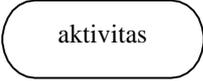
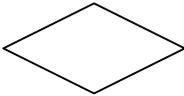
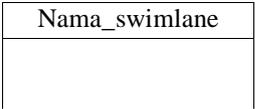
No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka / <i>Interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi / <i>Association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>Directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / <i>Dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
7.		Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber: Buku Karangan Rosa A.S dan M. Shalahuddin (2014:146)

2.3.18.4 Activity Diagram

Menurut Sukamto dan Shalahuddin (2014:161), Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) dari sebuah sistem. Macam-macam simbol diagram *Activity Diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-simbol Diagram Aktivitas

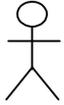
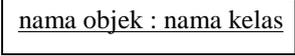
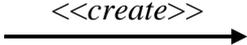
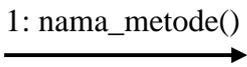
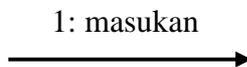
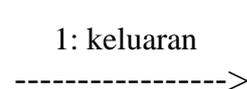
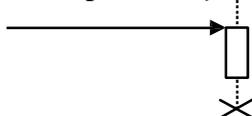
No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>Decision</i> 	Asosiasi percabangan dimana jika ada ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>Join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Buku Karangan Rosa A.S dan Shalahuddin (2014:162)

2.3.18.5 Sequence Diagram

Menurut Sukamto dan Shalahuddin (2014:165), *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Macam-macam simbol diagram *Sequence Diagram* dapat dilihat pada Tabel 2.5

Tabel 2.5 Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	Aktor  nama aktor atau  nama aktor tanpa waktu aktif	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.
2.	Garis Hidup 	Menyatakan kehidupan suatu objek.
3.	Objek  nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	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
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: Buku Karangan Rosa A.S dan Shalahuddin (2014:165)

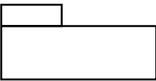
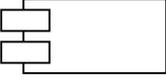
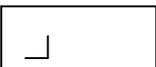
2.3.18.6 *Component Diagram*

Diagram komponen dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen focus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut.

- 1) *Source code* program perangkat lunak.
- 2) Komponen *executable* yang dilepas ke *user*.
- 3) Basis data secara fisik.
- 4) Sistem yang harus beradaptasi dengan sistem lain.
- 5) *Framework* sistem, *framework* pada perangkat lunak merupakan kerangka kerja yang dibuat untuk memudahkan pengembangan dan pemeliharaan aplikasi.

Macam-macam simbol diagram *Component Diagram* dapat dilihat pada Tabel 2.6

Tabel 2.6 Simbol-simbol *Component Diagram*

No.	Simbol	Deskripsi
1.	 <i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen.
2.	 <i>Component</i>	Komponen Sistem.
3.	 <i>Dependency</i>	Ketergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
4.	 <i>Nama_interface</i>	Sama dengan <i>interface</i> pada pemrograman berbasis objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.
5.	 <i>Link</i>	Relasi antar komponen.

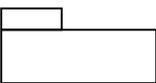
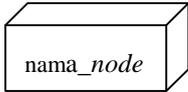
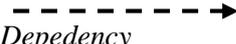
2.3.18.7 Deployment Diagram

Diagram *deployment* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut.

- 1) Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
- 2) Sistem terdistribusi murni dan rekayasa ulang aplikasi.

Macam-macam simbol diagram *Deployment Diagram* dapat dilihat pada Tabel 2.7

Tabel 2.7 Simbol-simbol *Deployment Diagram*

No.	Simbol	Deskripsi
1.	 <i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen.
2.	 <i>Node</i>	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah di definisikan.
3.	 <i>Dependency</i>	Ketergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.	 <i>Link</i>	Relasi anatar node.

2.3.19 Metode Pengujian

Perangkat lunak dapat diuji dengan dua cara, yaitu:

- 1) Pengujian dengan menggunakan data uji untuk menguji semua elemen program (data internal, loop, keputusan dan jalur). Data uji dibangkitkan dengan mengetahui struktur internal (kode sumber) dari perangkat lunak.
- 2) Pengujian dilakukan dengan mengeksekusi data uji dan mengecek apakah fungsional perangkat lunak bekerja dengan baik. Data uji dibangkitkan dari spesifikasi perangkat lunak.

2.3.19.1 Pengujian *BlackBox*

Shihab (2011), “*Black Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program”.

Shihab (2011), mengemukakan ciri-ciri *black box testing*, diantaranya sebagai berikut:

- 1) *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
- 2) *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih dari pada itu, ia merupakan pendekatan pelengkap dalam mencakup error dengan kelas yang berbeda dari metode *white box testing*.
- 3) *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

Pada *black box testing* terdapat jenis teknik *design test* yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan, diantaranya sebagai berikut:

- 1) *Equivalence Class Partitioning*
- 2) *Boundary Value Analysis*
- 3) *State Transitions Testing*
- 4) *Cause-Effect Graphing*

Kategori kesalahan/error yang akan diketahui melalui *black box testing*:

- 1) Fungsi yang hilang atau tak benar/salah
- 2) *Error* dari antar-muka/*interface*
- 3) *Error* dari struktur data atau akses eksternal database
- 4) *Error* dari kinerja atau tingkah laku/*perform*
- 5) *Error* dari inialisasi dan terminasi