

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini berkaitan dengan pembuatan Sistem Informasi dengan berbagai macam platform yang sudah beberapa kali dilakukan. [1] Susanti, Nanik dan Arifin, Moh., "*Sistem Informasi Manajemen Laboratorium (SIMLAB) (Studi Kasus Laboratorium Progdi Sistem Informasi UMK)*", Majalah Ilmu INFORMASI, vol.3, 2013. Mereka mengembangkan sistem yang dapat memasukkan data yang meliputi asisten dosen, dan inventarisasi aset dan bahan, data jadwal praktikum laporan kegiatan dan perawatan laboratorium. Dengan adanya sistem ini akan mempermudah informasi yang cepat dan akan membantu kepala laboratorium maupun pimpinan lainnya dalam menerima laporan data asisten, inventaris dan jadwal praktikum serta dapat menyajikan informasi yang cepat dan akurat kepada pengguna lainnya. Sistem informasi yang dibangun menggunakan bahasa pemrograman PHP dan MySQL Server sebagai databasnya.

Selain itu, penelitian lain yang berkaitan dengan pembuatan Sistem Informasi inventaris, [2] Sholikhin, Akhmad dan Riasti, Berliana Kusuma. "*Pembangunan Sistem Informasi Inventarisasi Sekolah Pada Dinas Pendidikan Kabupaten Rembang Berbasis Web*", Indonesia Jurnal on Networking and Security (INJS), vol.2, 2013. Pada penelitiannya dikatakan bahwa Inventarisasi sekolah dengan sistem komputerisasi yang dikembangkan dapat mengatasi masalah yang berdampak meningkatkan efisiensi dalam pengelolaan inventaris sekolah yang ada di kabupaten rembang. Meski demikian terdapat saran dari peneliti untuk menambahkan fitur pencarian yang membantu pengguna dengan cepat mencari informasi yang dibutuhkan. Dalam pembuatan sistem informasi inventarisasi sekolah pada dinas pendidikan kabupaten rembang menggunakan bahasa pemrograman PHP dan MySQL sebagai databasnya.

Selanjutnya penelitian lain dari [3] Syamsudin, Shendy Aditya, “*Sistem Informasi Manajemen Inventaris Labolatorium Pada STIMIK Amikom Yogyakarta*”, 2015. Dalam penelitiannya untuk membangun sistem yang dapat membantu UPT laboratorium sebagai pengelola dalam proses manajemen data inventaris, mulai dari proses pengadaan barang sampai dengan proses perawatan dan perbaikan inventaris serta penyajian laporan dengan mudah dan cepat. Beberapa fitur yang di buat dalam Sistem Informasi manajemen inventaris ini meliputi manajemen user, manajemen pengadaan inventaris manajemen perawatan dan perbaikan, manajemen pengelompokan, pengecekan, pencarian, rekapitulasi dan pengarsipan data. Dalam pembuatan Sistem Informasi manajemen inventaris labolatorium pada STIMIK Amikom menggunakan *framework Codeigniter* salah satu dari *framework* bahasa pemrograman PHP dan MySQL sebagai databasenya.

Dari beberapa penelitian yang telah dilakukan, disimpulkan bahwa pemanfaatan Sistem Informasi berbasis *web* pada proses manajemen suatu organisasi sangat baik untuk diterapkan. Serta pemrograman berbasis *web* dipilih karena memiliki kelebihan dimana aplikasi atau hasil dari pemrograman berbasis *web* dapat digunakan di platform apapun (Windows, Linux, iOS, Android, dll). Dengan hal tersebut pada Sistem Informasi Manajemen Aset Akbid Ummi Khasanah yang akan penulis teliti, *scoop* yang akan diteliti lebih memfokuskan pada manajemen pendataan dan pengelompokan aset itu sendiri, sistem dikembangkan sesuai kebutuhan dari Akbid Ummi Khasanah. Objek penelitian dengan menggunakan *Codeigniter* sebagai *framework PHP* dan *Bootstrap* sebagai front-end *framework* serta didukung teknologi AJAX untuk mengutamakan performa dan *User Experience*.

2.2 Konsep Dasar Sistem

2.2.1 Penjelasan Umum

Definisi dari kata sistem secara umum adalah sekumpulan objek-objek yang saling berelasi dan saling berinteraksi serta hubungan antara objek bisa di lihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan. Namun menurut

[4] Murdick dan Ross (1993) mendefinisikan sistem sebagai perangkat elemen yang digabungkan antara satu dengan lainnya untuk mencapai suatu tujuan tertentu.

2.2.2 Karakteristik Sistem

Untuk memahami atau mengembangkan satu sistem maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut karakteristik sistem yang membedakan satu sistem dengan sistem yang lain:

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 yang berada di luar sistem seperti lingkungan objek, kendala, dan input terhadap suatu sistem.
3. Masukan (*input*). Sumber daya seperti data, bahan dan informasi dari lingkungan yang akan dimanipulasi oleh sistem.
4. Keluaran (*output*). Sumber daya atau produk seperti laporan, informasi, dokumen, atau barang yang disediakan untuk lingkungan sistem oleh kegiatan dalam sistem itu sendiri.

2.2.3 Pengertian Sistem Informasi

Sistem Informasi (SI) merupakan gabungan dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, sistem informasi ini sering dipakai untuk merujuk pada interaksi antar orang, proses algoritmik, data, dan teknologi. Dalam penjelasan ini, istilah yang dipakai untuk merujuk tidak hanya untuk penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk orang berinteraksi dengan teknologi dalam mendukung proses bisnis.

Sistem Informasi dapat dibedakan menjadi dua yaitu Sistem Informasi manual dan Sistem Informasi berbasis komputer (Computer Based Information System). CBIS atau biasa disebut Sistem Informasi (SI) adalah jenis sistem informasi yang menggunakan perangkat komputer. Menurut [5] John F. Nash (1995). Sistem Informasi adalah merupakan kombinasi dari manusia, fasilitas atau alat teknologi,

media, prosedur dan pengendalian yang ditujukan untuk mengatur jaringan komunikasi yang penting, proses transaksi tertentu dan rutin, membantu manajemen dan pemakai *intern* dan *ekstern* dan menyediakan dasar untuk pengambilan keputusan yang tepat.

2.2.4 Sistem Informasi Manajemen

Sistem Informasi Manajemen (SIM) adalah sebuah sistem informasi pada level manajemen yang berfungsi untuk membantu perencanaan, pengendalian, dan pengambilan keputusan dengan menyediakan resume rutin dan laporan-laporan tertentu. SIM mengambil data mentah dari lingkungan dan mengubahnya menjadi kumpulan data yang lebih berarti dan bermanfaat.

2.2.5 Sistem Basis Data

Basis data adalah kumpulan data yang saling berelasi. [6] Kusri (2006), Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau symbol).

Basis data dapat didefinisikan dalam berbagai sudut pandang sebagai berikut:

1. Himpunan kelompok data yang saling berhubungan yang di organisasikan sedemikian rupa hingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling terhubung yang di simpan secara bersama sedemikian rupa tanpa pengulangan yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file/table/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Sedangkan sistem basis data adalah sistem computer yang digunakan untuk menyimpan dan mengolah data.

Sistem basis data memiliki empat komponen penting, yaitu:

1. *Data*, merupakan informasi yang disimpan dalam suatu struktur tertentu yang terintegrasi.
2. *Hardware*, merupakan perangkat keras berupa computer dengan media penyimpanan sekunder yang digunakan untuk menyimpan data karena pada umumnya *database* memiliki ukuran besar.
3. *Software*, merupakan perangkat lunak yang digunakan untuk melakukan pengelolaan data. Perangkat lunak ini sering disebut sebagai *Database Managemen System (DBMS)*.
4. *User*, merupakan orang yang menggunakan data yang tersimpan dan dikelola. *User* dapat berupa seorang yang mengelola *database* tersebut, yang disebut dengan *database administrator (dba)*, bias juga *end user* yang mengambil hasil dari pengelolaan *database* melalui bahasa *query*. *User* juga dapat seorang *programmer* yang membangun aplikasi yang terhubung ke *database*.

2.3 Pengertian Aset

Aset atau aktiva dalam akuntansi memiliki pengertian sebagai sumber daya atau kekayaan yang dimiliki oleh suatu entitas. Aset tersebut diperoleh dari peristiwa di masa lalu dan diharapkan akan memberikan manfaat dimasa yang akan datang.

Ada beberapa cara untuk memperoleh Aset, yaitu bisa diperoleh dengan cara diproduksi atau dibangun sendiri, bisa didapat dengan dibeli, juga dengan pertukaran aset maupun sumbangan dari pihak lain. Aset ada yang sifatnya berwujud dan tidak berwujud. [7] Gobel, Fariz (2016).

2.4 Konsep Dasar Web

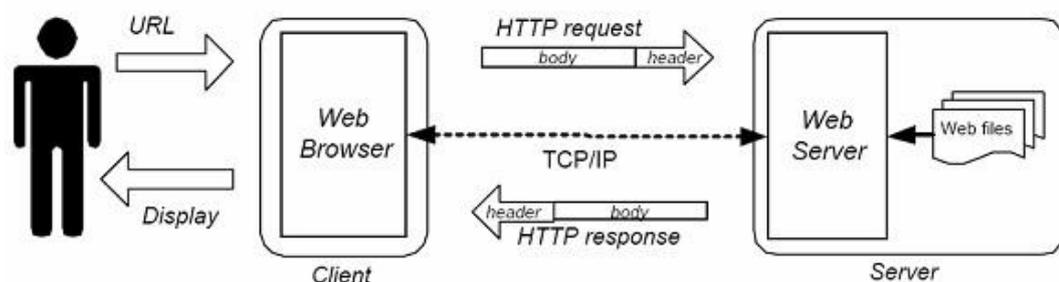
2.4.1 Web atau Word Wide Web

WWW pertama kali ditemukan oleh Tim Berners-Lee pada tahun 1991, WWW (*Word Wide Web*) atau yang disebut *Web* adalah suatu layanan di dalam

jarigan internet yang berupa ruang informasi [8] Raharjo, Budi (2011). *Web* terdiri dari jutaan situs *web* (*web site*) dan setiap *website* terdiri banyak halaman *web* (*web page*). Halaman-halaman *web* ini tersebar di seluruh dunia melalui komputer-komputer server yang telah terhubung dengan jaringan internet.

2.4.2 Web Server

Web server adalah *software* yang menjadi tulang belakang dari teknologi *world wide web* yang berfungsi untuk melayani permintaan halaman *web*. *Web server* menunggu permintaan dari *client* yang menggunakan *web browser* seperti Google Chrome, Netscap Navigator, Internet Explorer, Mozilla Firefox, dan program *web browser* lainnya. Jika permintaan dari pengguna yang menggunakan *web browser*, maka *web server* akan memproses permintaan tersebut kemudian memberikan hasil proses berupa data yang di inginkan kembali ke *web browser*. Cara kerja web server bisa di lihat pada gambar 2.1.



Gambar 2.1 Cara Kerja Web Server

2.4.2 Web Browser

Web browser atau yang disebut juga dengan penjelajah *web*, adalah sebuah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *web server*.

2.5 Konsep Dasar Pemrograman

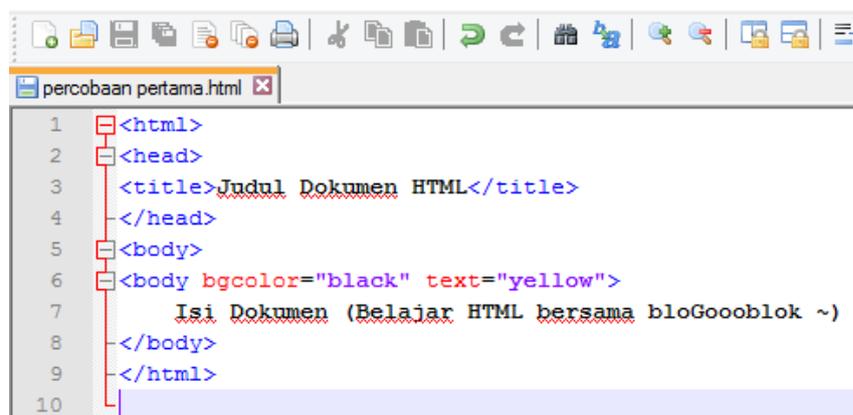
2.5.1 Pemrograman Web

Pemrograman *web* atau dalam bahasa inggrisnya “*web programming*” terdiri dari dua kata yaitu pemrograman dan *web*. Pemrograman adalah kumpulan instruksi atau perintah tertulis yang dibuat oleh manusia dengan bahasa pemrograman tertentu yang secara logis memerintahkan komputer agar melakukan langkah atau proses tertentu dalam menyelesaikan suatu masalah. Sedangkan kata *web*, dapat di artikan sebagai halaman *web* atau media informasi yang dapat diakses dengan perangkat lunak *web browser* melalui jaringan internet.

2.5.2 HyperText Markup Language (HTML)

HTML atau *HyperText Markup Language* adalah suatu bahasa pemrograman yang di gunakan untuk membuat sebuah halaman *web*. Kode-kode dari HTML ini menerjemahkan bagaimana tampilan sebuah halaman *web* akan diproses dan di tampilkan. Dalam sebuah halaman *web* terdiri dari banyak elemen HTML seperti elemen tabel, daftar, paragraph, dan lain sebagainya, untuk menandakan elemen-elemen tersebut dalam sebuah file HTML, maka digunakan sebuah tag.

Struktur dasar dari kode HTML adalah adanya tag elemen-elemen pendiri dari sebuah halama web seperti tag BODY dan HEAD seperti gambar 2.2.



```
1 <html>
2 <head>
3 <title>Judul Dokumen HTML</title>
4 </head>
5 <body>
6 <body bgcolor="black" text="yellow">
7     Isi Dokumen (Belajar HTML bersama bloGoooblok ~)
8 </body>
9 </html>
10
```

Gambar 2.2 Struktur Dasar HTML

2.5.3 PHP

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis [9] Sunyoto, Andi (2007). *Sintaks* dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirim ke browser dalam format HTML. Dengan demikian kode program yang di tulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan halan web lebih terjamin.

2.5.4 JavaScript

JavaScript adalah sebuah bahasa dalam pembuatan program yang dapat dikenal dan dieksekusi oleh *web browser* dengan tujuan untuk menjadikan suatu halaman *web* menjadi lebih interaktif [10] Arifin, M. Rudyanto (2011). JavaScript merupakan bahasa pemrograman yang tidak membutuhkan lisensi untuk dapat menggunakannya. Jika *web browser* yang digunakan mendukung *JavaScript* maka data langsung membuat aplikasi berbasis *web* dengan menggunakan *JavaScript*.

2.5.5 AJAX (Asynchronous JavaScript and XMLHttpRequest)

AJAX atau yang disebut *Asynchronous JavaScript and XMLHttpRequest* adalah suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web yang interaktif menyamai aplikasi *desktop* [9] Sunyoto, Andi (2007). Tujuannya adalah untuk melakukan pertukaran data dengan server di belakang layar, sehingga halaman web tidak harus dibaca ulang secara keseluruhan setiap kali seorang pengguna melakukan perubahan. Hal ini akan meningkatkan interaktivitas, kecepatan, dan *usability*.

2.5.6 JQuery

JQuery adalah sebuah *library JavaScript* yang sangat ringkas dan sederhana untuk manipulasi komponen DOM di dalam dokumen HTML [11] Komputer, Wahana (2012). Contoh kemampuan jQuery adalah menangani event, animasi, efek dan memproses interaksi ajax. JQuery dirancang sedemikian rupa agar para

pengembang web dapat membuat program *JavaScript* menjadi lebih mudah dan ringkas. JQuery dapat berjalan lancar disemua *web browser*.

2.5.7 MySQL

MySQL adalah sebuah perangkat lunak *Relational Database Management System* (RDMS) yang bersifat gratis dan *open source*. MySQL sangat cepat dan mudah di konfigurasi, mudah di pelajari dan tersedia *source-code*-nya. Itulah beberapa kelebihan utama dari MySQL dibanding RDMS komersial yang ada. MySQL menggunakan SQL (*Structured Query Language*) yang merupakan standar global dalam manajemen data dalam *database*.

2.5.8 CSS (Casing Style Sheet)

CSS atau *Casing Style Sheet* adalah suatu bahasa yang bekerjasama dengan DOM HTML untuk mendefinisikan cara bagaimana suatu isi halaman *web* dipresentasikan [8] Raharjo, Budi (2011). CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna mouse over, spasi antar paragraph, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS akan memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

2.5.9 Framework PHP

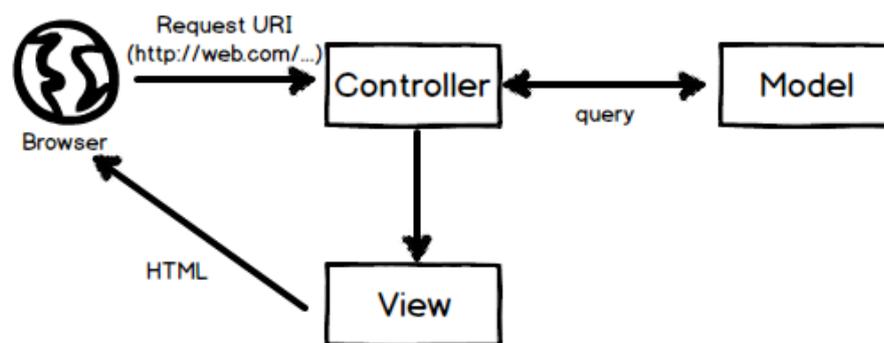
2.5.9.1 Penjeasan Umum

Framework dapat diartikan sebagai sebuah kerangka kerja dalam suatu proses. Jika dikaitkan dengan pemrograman PHP maka dapat diartikan sebagai suatu kerangka kerja yang telah terpola dan tersusun yang memudahkan pengembangan *web* dalam pembuatan *web* yang menggunakan bahasa pemrograman PHP. Dengan kata lain *Framework* PHP membantu mempromosikan pengembangan aplikasi cepat atau yang disebut *Rapid Application Developer* (RAD) dan akan menghemat waktu pemrograman dalam membangun aplikasi. Kerangka kerja juga dapat membantu programmer untuk membangun aplikasi yang lebih stabil dengan memastikan interaksi database dan coding pada tempat yang

terpisah. Hal ini memungkinkan anda untuk menghabiskan banyak waktu menciptakan aplikasi *web* yang sebenarnya, daripada menghabiskan waktu menulis kode berulang. Gagasan umum di belakang kerangka kerja PHP adalah *Model View Controller* (MVC).

2.5.9.2 MVC (Model View Controller)

Model View Controller merupakan suatu konsep rancangan yang sangat populer dalam pembangunan aplikasi perangkat lunak khususnya aplikasi *web*, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Arsitektur MVC dapat dilihat pada gambar 2.3.



Gambar 2.3 Arsitektu Model View Controller

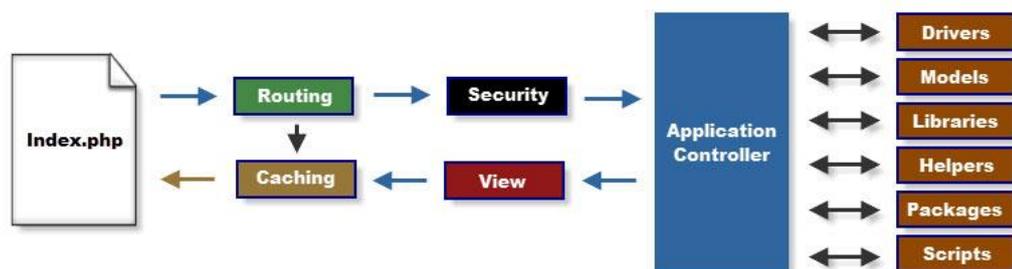
View, merupakan bagian yang menangani logika presentasi atau desain tampilan. Pada suatu aplikasi *web* bagian ini biasanya berupa berkas template HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan mempresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.

Model, biasanya berhubungan langsung dengan bagian database untuk manipulasi data, menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*. Semua skrip dari operasi database seperti skrip SQL akan dituliskan di bagian ini.

Controller, merupakan jembatan antara bagian *model* dan *view*, controller berfungsi untuk menerima request data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

2.5.9.3 Code Igniter

Code Igniter adalah sebuah aplikasi perangkat lunak open source yang berupa *framework* PHP dengan konsep MVC (*Model View Controller*). *Code Igniter* memudahkan pengembang perangkat lunak untuk membuat aplikasi *web* dengan cepat, mudah dan terstruktur dibandingkan dengan membuatnya dari awal, kelebihan penggunaan *Code Igniter* dibandingkan *framework* PHP lainnya adalah karena *Code Igniter* memiliki konfigurasi yang sangat minim, performa yang cepat, banyaknya komunitas pengembang perangkat lunak yang menggunakan serta dokumentasi penggunaan program yang sangat lengkap dibanding *framework* PHP lainnya. Cara Kerja *Framework Code Igniter* dapat dilihat pada gambar 2.4.



Gambar 2.4 Cara Kerja Framework Code Igniter

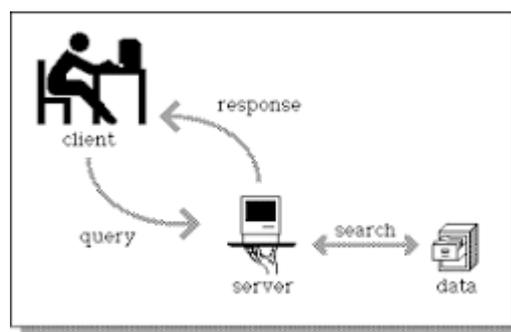
2.5.9.4 Framework CSS Bootstrap

Bootstrap merupakan sebuah *framework* CSS dari twitter, yang menyediakan kumpulan komponen-komponen antar muka dasar pada *web* yang telah di rancang sedemikian rupa untuk digunakan bersama-sama. Selain komponen anarmuka, *Bootstrap* juga menyediakan sarana untuk membangun *layout* halaman dengan mudah dan rapi, serta modifikasi pada tampilan dasar HTML untuk membuat seluruh halaman *web* yang dikembangkan senada dengan komponen-komponen lainnya.

2.6 Teori Perancangan

2.6.1 Arsitektur Client-Server

Client-server adalah suatu bentuk arsitektur pembuatan sistem yang memisahkan *client* dan *server*, dimana *client* adalah perangkat yang meminta dan menerima data, dan *server* adalah perangkat yang bertindak sebagai penyedia dan pengelola aplikasi, data, dan keamanannya tersebut. *Server* biasanya terhubung dengan *client* melalui sebuah kartu jaringan. Dalam teknologi informasi, *client-server* merujuk pada cara mendistribusikan aplikasi ke pihak *client* dan pihak *server*. Dalam model *client-server*, aplikasi dibagi menjadi dua bagian yang terpisah yakni komponen *client* dan komponen *server*. Pemodelan dari Arsitektur Client-Server dapat dilihat pada gambar 2.5.



Gambar 2.5 Model Umum dari Arsitektur Client-Server

2.6.2 UML (Unified Modeling Language)

UML (*Unified Modeling Language*) merupakan bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artefak dari proses analisa dan desain yang berorientasi pada objek [12] Hermawan, Julius (2004). UML memiliki beberapa artefak yang digunakan untuk memvisualisasikan dalam sebuah program. Srtifak dalam UML menginformasikan tentang berbagai bentuk permedelan yang digunakan atau yang dihasilkan dalam proses pembagunana program. Terdapat beberapa artefak dalam UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Squence Diagram*.

2.6.2.1 Use-Case Diagram

Use Case Diagram merupakan diagram yang menggambarkan interaksi antar *usecase* dan *Actor*. Dimana *Actor* dapat berupa orang peralatan atau sistem lain yang berinteraksi dengan sistem yang dibangun. *Use case* menggambarkan fungsionalitas sistem dari pandangan pemakai.

2.6.2.2 Activity Diagram

Merupakan diagram yang menggambarkan fungsionalitas bagian-bagian sistem. *Activity* diagram menunjukkan aliran kerja bisnis (*business work flow*) atau aliran kejadian (*flow of events*) dalam *use case*.

2.6.2.3 Class Diagram

Merupakan diagram yang menunjukan interaksi antar kelas dalam sistem yang menggambarkan bagian mereka saling berkolaborasi untuk mencapai suatu tujuan.

2.6.3 ERD (Entity Relationship Diagram)

Perancangan basis data dengan menggunakan model entity relationship adalah dengan menggunakan *Entity Relationship Diagram* (ERD). Terdapat tiga notasi dasar yang bekerja pada E-R yaitu entity sets, relationship, dan attributes. ERD berbeda dengan DFD (Data flow Diagram) yang merupakan suatu model jaringan fungsi atau proses alur dari data yang akan dijalankan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan hubungan data. Tiga notasi yang digunakan dalam ERD, yaitu:

1. Entitas

Entitas merupakan objek yang mewakili suatu yang nyata dan dapat dibedakan dari suatu yang lain. Simbol dari entitas ini biasanya digambarkan dengan persegi panjang.

2. Relasi

Relasi merupakan sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dalam ERD digambarkan dengan simbol belah ketupat.

Relasi yang terjadi antara dua himpunan entitas (misalnya A dan B) dalam suatu basis data juga dijabarkan menjadi 3 jenis yaitu:

1. Satu ke satu (*One to one*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan B.

2. Satu ke banyak (*One to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3. Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan B.

3. Atribut

Setiap entitas pasti mempunyai elemen yang disebut dengan atribut, atribut berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

2.6.4 Uji Coba Program

2.6.4.1 Black box Testing

Black box testing Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode, pada penelitian ini dipilih menggunakan *black box testing* karena dianggap lebih tepat dibandingkan *white box testing*. Perangkat lunak memerlukan seperangkat tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini *black box testing* lebih tepat dan sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Black box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineer untuk memperoleh input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.