

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian dilaksanakan oleh (Dalis, 2017) berjudul “**Rancang Bangun Sistem Informasi Lembaga Penelitian Dan Pengabdian Masyarakat Berbasis Website**”. Penelitian sistem LPPM ini dibangun dengan tujuan mengelola kumpulan data penelitian, pengabdian masyarakat yang akan dan telah dilaksanakan oleh dosen di lingkungan Akademik Bina Sarana Informatika(BSI) dalam kegiatan yang berhubungan dengan penelitian, pemograman berorientasi objek. Penelitian yang dengan metode *Software Development Life Cycle (SDLC)* model *waterfall* dalam merancang antarmuka sistem informasi Lembaga Penelitian dan Pengabdian Masyarakat berbasis web. Dirancang menggunakan database MySQL sebagai media penyimpanan data dan penggunaan bahasa pemrograman PHP. Hasil akhir dalam melaksanakan penelitian ini berupa sistem yang membantu dosen beserta peneliti di lingkungan BSI dalam mendapatkan informasi yang efektif, efisien dan akurat.

Penelitian yang dilaksanakan (Lestari, Anjarwani, & Agitha, 2018) berjudul “**Rancang Bangun Sistem Informasi Surat Perintah Perjalanan Dinas pada Sekretariat Daerah Provinsi Berbasis Web**”. Tujuan pelaksanaan penelitian ini menghasilkan sistem informasi yang bisa mempermudah admin dalam proses membuat Surat Perintah Tugas(SPT) dan Surat Perintah Perjalanan Dinas(SPPD) pegawai yang akan melaksanakan perjalanan dinas, serta pembuatan laporan SPT dan SPPD.

Melaksanakan penelitian dengan metode *Software Development Life Cycle*(SDLC), bahasa pemrograman PHP *framework codeigniter* dan penyimpanan data menggunakan database MySQL. Dengan melaksanakan penelitian ini, membantu admin dalam menata data SPT dan mengelola data serta kelola laporan bulanan dan tahunan.

Penelitian yang dilakukan oleh (Sitohang, 2018) berjudul “**Sistem Informasi Pengagendaan Surat Berbasis Web Pada Pengadilan Tinggi Medan**”. Tujuan melaksanakan penelitian ini agar dapat mempermudah para pengelola surat dalam penggunaan sistem informasi yang disesuaikan dengan kebutuhan. Sistem ini menggunakan media penyimpanan database MySQL dan bahasa pemrograman PHP dalam proses pembuatan sistem. Penelitian yang dihasilkan dapat mempermudah para pegawai mengelola surat masuk dan surat keluar sehingga memininkan waktu, agar efesiensi kerja mengalami peningkatan.

Penelitian dilaksanakan oleh (Cahyo Nugroho, 2019) berjudul penelitian “**Rancang Bangun Sistem Informasi Manajemen Surat Tugas Berbasis Web Menggunakan Waterfall Model**”. Tujuan penelitian adalah membantu proses pembuatan surat tugas menjadi lebih mudah, cepat dan terdokumentasi dengan baik.

Mempermudah dosen dalam mengakses surat tugasnya masing-masing serta meringankan tugas Admin. Penelitian dengan metode *Software Development Life Cycle*(SDLC), bahasa pemrograman PHP serta basis data MySQL. Penelitian yang menghasilkan Sistem informasi yang mengatasi permasalahan kergantungan dengan ketelitian dan kerapian admin prodi dalam melakukan manajemen berkas. Kumpulan Berkas yang sudah dalam bentuk digital sehingga memudahkan dalam hal penyimpanan maupun distribusi.

Penelitian berjudul "**Rancang Bangun Sistem Administrasi Kerja Praktek dan Tugas Akhir Berbasis Web Menggunakan Framework CodeIgniter**" oleh

(Subari, Tadeus, Winarno, & Yuwono, 2018). Tujuan melaksanakan penelitian ini adalah sistem administrasi yang menyiapkan sarana penunjang dengan menggunakan komputer dalam kegiatan kerja praktek dan tugas akhir baik sebelum, selama dan setelah dengan mudah, cepat dan akurat sesuai dengan peraturan yang berlaku. Sistem informasi ini di bangun dengan PHP *framework codeigniter*. Hasil akhir penelitian ini berbentuk sistem informasi yang dapat menyediakan sarana penunjang kegiatan kerja praktek ataupun tugas akhir.

Tabel 2.1 Hasil Penelitian

Peneliti	Judul Penelitian	Tahun	Hasil Penelitian
(Dalis, 2017)	Rancang Bangun Sistem Informasi Lembaga Penelitian Dan Pengabdian Masyarakat Berbasis Website	2017	Sistem yang membantu dosen beserta peneliti di lingkungan BSI dalam mendapatkan informasi yang efektif, efisien dan akurat.
(Lestari et al., 2018)	Rancang Bangun Sistem Informasi Surat Perintah Perjalanan Dinas pada Sekretariat Daerah Provinsi Berbasis Web	2018	Sistem yang membantu admin dalam mengelola data SPT dan mengelola data serta kelola laporan bulanan dan tahunan
(Sitohang, 2018)	Sistem Informasi Pengagendaan Surat Berbasis Web Pada Pengadilan Tinggi Medan	2018	Sistem yang mempermudah para pegawai dalam pengelolaan surat masuk dan surat keluar sehingga memininkan waktu agar efesiensi kerja mengalami peningkatan

(Cahyo Nugroho, 2019)	Rancang Bangun Sistem Informasi Manajemen Surat Tugas Berbasis Web Menggunakan Waterfall Model	2019	Sistem informasi yang mengatasi permasalahan kergantungan dengan ketelitian dan kerapian admin prodi dalam melakukan manajemen berkas. Kumpulan Berkas yang sudah dalam bentuk digital sehingga memudahkan dalam hal penyimpanan maupun distribusi
(Subari et al., 2018)	Rancang Bangun Sistem Administrasi Kerja Praktek dan Tugas Akhir Berbasis Web Menggunakan Framework CodeIgniter	2018	Sistem informasi yang dapat menyediakan sarana penunjang kegiatan kerja praktek ataupun tugas akhir

Dari penelitian yang dilaksanakan diatas, maka penulis akan merancang sebuah sistem Pengajuan Surat Tugas berbasis Website seperti yang telah dilakukan oleh (Dalis, 2017). Penggunaan bahasa Pemograman PHP dan MySQL seperti yang telah digunakan oleh (Sitohang, 2018). Proses membangun sistem ini dengan *framework CodeIgniter* seperti yang telah digunakan oleh (Subari et al., 2018). Melihat kondisi dan memahami proses yang berjalan pada saat melakukan observasi, maka dengan demikian akan melakukan wawancara seperti konsep yang diterapkan oleh (Lestari et al., 2018). Menggunakan metode SDLC(*Software Development Life Cycle*) model *waterfall* dalam membangun sistem dengan menerapkan cara (Cahyo Nugroho, 2019) dalam proses perancangan sistem informasi.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Merupakan sistem yang didirikan dari macam-macam bagian yang sama-sama terhubung sehingga menjadi suatu informasi bermanfaat bagi yang memerlukan (Hermawan & Hidayat, 2016). Sistem informasi dibagi menjadi beberapa jenis :

1. Sistem Informasi Manajemen

Adalah informasi yang menyediakan kumpulan informasi untuk keperluan sebuah organisasi dalam bentuk sistem berbasis komputer (Khairina, Maharani, & Hatta, 2018).

2. Sistem Informasi Eksekutif

Sistem yang digunakan untuk meringankan pemecahan sebuah masalah yang tidak teratur oleh manajer tingkat atas (Fatmasari, 2017).

3. Sistem Informasi Akuntansi

merupakan informasi keuangan yang didapatkan dari kumpulan data transaksi bisnis yang berguna untuk pemakainya (Purwa Abhimantra & Suryanawa, 2016).

4. Sistem Informasi Keuangan

Adalah data transaksi keuangan sebuah perusahaan yang diolah sistem komputer secara terintegrasi dalam mengatasi permasalahan penerapan data yang tidak efisien, kebenaran informasi yang ketidakpastian, dan pada saat terjadinya pengulangan data (Santi & Yulianton, 2016).

5. Sistem Informasi Manufaktur

Adalah solusi yang tepat dalam sudut pandang perusahaan jangka panjang. Karena lebih menfokuskan proses produksi, mulai dari barang baku yang diinput sampai barang jadi dalam pertimbangan semua kemungkinan prosedur yang terjadi pada saat proses pembuatan (Sucipto, 2017).

6. Sistem Informasi Sumber Daya Manusia

Adalah sumber daya yang memiliki peran penting untuk memperoleh suatu keberhasilan (Samsuni, 2017).

7. Sistem Informasi Pemasaran

Adalah alur informasi yang tepat dalam proses mengambil keputusan berdasarkan interaksi rumit antara mesin, manusia, dan prosedur-prosedur sehingga sumber dalam ataupun luar suatu perusahaan bisa dipertanggungjawabkan (Rahmawati & Mulyono, 2016).

2.2.2 *System Development Life Cycle*

Suatu metode proses pengubahan sistem informasi dengan menggunakan metodologi orang dan model untuk pengembangan sistem sebelumnya (Larasati & Masripah, 2017). Model SDLC yang populer tidak lain adalah Waterfall, model ini banyak diterapkan dalam sistem yang dirancang. Dalam menggunakan peredaran SDLC, proses pembangunan sistem terbagi menjadi bertahap, ada empat tahapan dalam pengembangan sistem secara garis besar sebagai berikut :

1. Perencanaan

Tahapan rancangan baik secara teknik maupun secara teknologi yang disesuaikan dengan kebutuhan serta membuat penjadwalan proyek (Sofyan, Puspitorini, & Yulianto, 2016).

2. Analisis

Analisis adalah mengumpulkan kebutuhan yang dilakukan dengan tekun dan berkelanjutan hingga memahami apa yang dibutuhkan pengguna (Larasati & Masripah, 2017).

3. Desain

Merupakan tahapan dimana fokus kepada *interface* sistem yang akan dibangun termasuk susunan data, representasi tampilan dan rancang bangun perangkat lunak (Larasati & Masripah, 2017).

4. Implementasi

Tahap ini adalah proses pengujian yang dilakukan pada sistem yang sudah diselesaikan untuk menentukan baik secara logik dan fungsional semua sudah teruji dengan baik. Dilakukan untuk menekan kesalahan yang terjadi serta memastikan sistem diselesaikan sudah cocok dengan yang diinginkan oleh *user*. (Larasati & Masripah, 2017).

Model *Waterfall* ialah prosedur pengembangan sistem yang menggunakan cara pengurutan bagaikan air terjun yang terus mengalir ke bawah. Tahapan yang mesti di lalui terdiri dari analisis, pemodelan, pembuatan kode program, pengujian, dan pemeliharaan (Hartati & Wijaya, 2016). Definisi tahapannya sebagai berikut:

1. Analisis

Proses menganalisis kebutuhan sistem dengan mengumpulkan data, melakukan penelitian serta wawancara dengan narasumber. Hasil akhir dari tahapan ini merupakan dokumen yang berisi kebutuhan dan permintaan *user* yang akan dijadikan sebagai referensi dalam mengubahnya kedalam bahasa pemrograman.

2. Pemodelan

Tahap prosedur mengimplementasikan kondisi kebutuhan kedalam sistem yang dirancang. Prakiraan terlebih dahulu sebelum memasuki tahap coding, yaitu melaksanakan proses menyusun data, design perangkat lunak, demonstrasi *interface* serta detail prosedural.

3. Pembuatan kode program

Tahapan dimana menggunakan komputer mengerjakan sistem dengan maksimal dalam mengubah desain menjadi bahasa yang di pahami komputer.

4. Pengujian

Tahapan dimana sistem sudah selesai dan akan dipakai oleh pengguna.

5. Pemeliharaan

Tahap pemeliharaan merupakan tahap jika adanya terjadi perubahan pada sistem, perubahan bisa karena ada terjadi kesalahan di sistem yang harus menyesuaikan lingkungan baru ataupun membutuhkan pengembangan fungsional yang di inginkan oleh pengguna.

2.2.3 Website

Website adalah halaman yang dikumpulkan dalam memperlihatkan informasi data teks, gambar, suara, animasi, video serta campuran dari segalanya. Baik secara statis ataupun dinamis dalam membentuk satu hubungan bangunan yang saling terhubung, yang tiap-tiap dihubungkan dengan halaman (Destiningrum & Adrian, 2017). *Website* adalah gabungan halaman web yang sudah diterbitkan di jaringan internet yang mempunyai domain/*URL*(*Uniform Resource Locator*) yang bisa di akses oleh semua pemakai internet dengan mengetikkan alamat *URL*. Dimungkinkan karena adanya teknologi *World Wide Web*(*WWW*). Halaman *website* adalah dokumen yang tertulis dalam format *Hyper Text Markup Language*(*HTML*) yang dapat di akses melalui *HTTP*, *HTTP* ialah suatu jenis protokol yang mengirimkan beraneka macam informasi didapatkan dari server *website* yang dimunculkan untuk pemakai melalui *web browser* (Nofyat, Ibrahim, & Ambarita, 2018).

2.2.4 PHP

(*Hypertext Preprocessor*) *PHP* merupakan bahasa pemrograman web berbasis *server* yang dapat memarsing kode *PHP* dari kode ekstensi *PHP*, maka dari itu menghasilkan *website* yang dinamis pada tampilan *client* (Sitohang, 2018). *PHP* merupakan *server-side programming language* yang digunakan dalam pengembangan suatu *website* dimana bisa disisipkan pada file *HTML* (Cahyo Nugroho, 2019). *PHP* open source dan banyak digunakan dalam pengembangan sebuah *website*, bisa diintergrasi dengan database *MySQL* memungkinkan membangun aplikasi yang bisa olah dan memanipulasi data. *PHP* adalah bahasa pemrograman yang bersifat *server side scripting*, dimana bahasa pemrograman ini bekerja pada sisi

server. *Server-side scripting* yang menyatu dengan HTML dalam membuat halaman web yang dinamis, dikarenakan PHP *server-side* maka perintah-perintah akan dieksekusi sisi *server* dan dikirimkan hasilnya ke *browser* menggunakan format HTML(Wibawa & F., 2017).

2.2.5 *Codeigniter*

Adalah *framework web open source* yang pada umumnya diperlukan dalam membangun aplikasi php yang dinamis. Tujuan utama dari pengembangan *Codeigniter* agar dapat memudahkan *developer* dalam mengembangkan aplikasi tidak harus menetik semua coding dari awal dan tentunya proses yang lebih cepat. *Codeigniter* juga terdapat macam-macam *library* yang bisa digunakan oleh *developer* untuk mempermudah pengembangan. *Codeigniter* menggunakan rancangan *Model-View-Controller(MVC) development pattern*. Pada *Codeigniter*, *Controller* yang berinteraksi dengan *browser*, semua *request* yang dikirimkan dari *browser* akan di terima oleh *Controller*. *Controller* akan meminta data ke bagian *Model*, sedangkan dalam menampilkan ke pengguna di tangani oleh bagian *View*. Jadi *Controller* adalah inti dari aplikasi, tampilan aplikasi ditangani oleh *View* dan data berada di *Model*. (Subari et al., 2018).

2.2.6 *Model View Controller (MVC)*

MVC merupakan rancangan dalam mengenkapsulasi data bersama dengan pemrosesan(*model*), memisahkan dari proses manipulasi(*controller*) dan tampilan(*view*) untuk tampilkan pada sebuah *user interface*. MVC menggunakan pendekatan *Layering* yang paling umum. *Layering* adalah logika dimana memecah kode kita kedalam fungsi pada kelas yang berbeda. Pendekatan ini mudah dikenali dan

yang paling banyak diterima. Keuntungan utama dari pendekatan ini ialah (*reusability*) kode sehingga coding tidak perlu dibuat ulang lagi dan bisa dipakai berulang kali. MVC dibagi membentuk tiga bagian yaitu terdiri dari *model*, *view* dan *controller*. Susunan MVC memiliki keuntungan dalam menyisahkan antara *model* dan *view* membolehkan beberapa *view* menggunakan *model* yang sama. Alhasil, bagian model seubah aplikasi lebih gampang untuk dijalankan, diuji, dan dipelihara karena semua akses ke *model* beroperasi melalui bagian ini (Subari et al., 2018).

Model yang mengoperasikan informasi serta menginformasikan pengamat ketika ada terjadinya perubahan informasi. Dalam *model* terdapat data dan fungsi yang berkaitan pada pemrosesan suatu data, *model* dapat menyatakan selain dari data dan fungsi yang beroperasi didalamnya. Contohnya *programmer* bisa menentukan model yang menjadi penghubung antara *back-end* dengan *front-end*.

View yang mengelola pemetaan tampilan pada sebuah perangkat. *View* pada umumnya memiliki 1-1 dengan sebuah permukaan layar dan tahu bagaimana cara pembuatannya. *View* datanya dari model dan menampilkan datanya ke *Interface*, dan jika ada perubahan pada *model* maka *view* akan otomatis menampilkan kembali segmen yang berubah dan menampilkan perubahan tersebut.

Controller berfungsi dalam mengambil inputan dari user dan mengarahkan *model* dan *view* dalam melaksanakan aksi bersumber pada inputan pengguna. Maka dari itu *controller* bertanggung jawab dalam proses aktivitas *user* terhadap respon aplikasi. Contohnya jika *user* klik tombol pada menu, *controller* yang menentukan bagaimana aplikasi semestinya berjalan.

2.2.7 HTML

Hypertext Markup Language(HTML) adalah dokumen yang disediakan dalam *browser web surfer*, pada umumnya dokumen ini memiliki informasi, tampilan sebuah sistem didalam internet(Prabowo, 2017). HTML ialah bahasa yang dipergunakan dalam menggambarkan struktur halaman web. HTML berperan dalam memublikasikan dokumen online, *tags* adalah perintah dasar HTML. *Tags* diakui dengan kurung siku(<>). *Tags* yang digunakan dalam sebuah dokumen harus dibuat berpasangan. Mencakup *tag* pembuka dan *tag* penutup, *tag* penutup ada penambahan tanda garis miring(/) diawal nama *tag* (Pahlevi, Mulyani, & Khoir, 2018).

2.2.8 CSS

Cascading Style-Sheets(CSS) adalah rangkain instruksi yang menentukan bagaimana *interface* sebuah web. Mulai dari perancangan desain *text* dengan menentukan huruf, warna(*colors*), ukuran(*margins*), latar belakang(*background*), ukuran font(*font sizes*). CSS dapat diartikan menempatkan *styles* yang berbeda pada sebuah lapisan(*layers*). CSS umumnya digunakan dalam format halaman web yang tertulis HTML(Martiana & Irfan, 2016).

2.2.9 JavaScript

JavaScript merupakan *Client Side Programming Language*, *Client Side Programming Language* maksudnya jenis bahasa pemrograman yang prosedurnya dijalankan oleh *client*. Aplikasi *client*, maksudnya adalah *web browser* semacam Google Chrome, Mozilla Firefox. *JavaScript* dikembangkan untuk pertama kalinya pada pertengahan dekade 90'an. Walaupun mempunyai nama yang nyaris sama, *JavaScript* tidak sama dengan bahasa pemrograman *Java*. Penulisan *JavaScript* bisa

disisipkan kedalam dokumen HTML atau dokumen tersendiri. *JavaScript* menerapkan fitur yang dirancang dalam mengendalikan bagaimana sebuah halaman web berinteraksi dengan pengguna (Pahlevi et al., 2018).

2.2.10 *Bootstrap*

Merupakan *framework front-end* yang digunakan dalam mempermudah dan mempercepat pengembangan sebuah website yang memiliki tampilan yang bagus. *Bootstrap* mempersiapkan *HTML*, *CSS*, dan *Javascript* yang tersedia untuk digunakan dan bisa dikembangkan sesuai kebutuhan dengan mudah. Sejak tahun 2012 *Bootstrap* dilengkapi fitur *responsive*, *bootstrap* banyak diminati dan semakin banyak website yang menggunakan *bootstrap*. Dengan fitur *responsive* yang disediakan oleh *bootstrap* ini, sebuah website disesuaikan dalam berbagai ukuran layar seperti *smartphone*, *tablet* dengan tampilan yang rapi mengikuti ukuran layar. Dengan penggunaan *bootstrap* mempermudah proses desain web (Ariansyah, Fajriyah, & Prasetyo, 2017). *Bootstrap* ialah salah satu jenis *framework* campuran dari *CSS* dan *Javascript* yang diusulkan menjadi pilihan dalam konsistensi tampilan dalam membangun sebuah website. *Bootstrap* adalah aplikasi *open-source* dan mendukung platform seperti *HTML5* dan *CSS3*. Dalam *framework* ini sendiri mempunyai fungsi meringankan developer dalam membangun *interface* website yang dibangun, template yang disediakan untuk *font*, tombol, navigasi, ikon, form (Martin & Tanaamah, 2018).

2.2.11 *Database*

Adalah media penyimpanan data yang digunakan sebagai fasilitas pergantian kumpulan data bisnis tanpa harus pengeluaran biaya yang besar dengan menjalankan proses pergantian data (Hanafi, Sukarsa, & Agung Cahyawan Wiranatha, 2017).

Relational Database Management Systems (RDBMS) merupakan suatu perangkat lunak dalam menangani penciptaan, pemeliharaan, dan pengendalian akses sebuah data. Dengan RDBMS, mengelola data menjadi mudah. Ada beberapa aplikasi RDBMS saat ini, salah satunya adalah MySQL, aplikasi ini *open source* maka bisa dikembangkan dengan mudah (Wibawa & F., 2017).

MySQL adalah sebuah program aplikasi dalam membuat *database* yang biasanya digunakan pada *web server*. Banyak yang menggunakan MySQL karena mampu menerima data berskala cukup besar dan proses *loading* yang cepat dalam proses pengambilan data. Kelebihan yang dimiliki MySQL sebagai berikut:

1. *Open Source*, maka bisa digunakan secara bebas.
2. *Multi user*, bisa digunakan oleh lebih dari satu *user* di waktu yang sama. Dengan demikian, *database server* bisa di akses *client* secara bersamaan.
3. *Security*, MySQL mempunyai berbagai susunan security, yaitu *level subnet mask*, nama *host*, dan izin akses user menggunakan *password* yang terekripsi.
4. *Connectivity*, bisa melakukan koneksi dengan *client* menggunakan TCP/IP(NT) atau *Unix socket*(Unix).

Database tersusun dibagi menjadi 2 jenis penggolongan, terdiri dari:

1. *Database hierarki*

Merupakan susunan tingkat abstraksi yang membentuk layaknya struktur sebuah pohon yang memiliki aturan-aturan khusus berdasarkan peringkat.

Terdiri dari Sistem *Database*, *File*, *Record*, *Field*, *Byte*, dan *Bit*. *Database* merupakan kumpulan data yang terkait satu dengan yang lainnya. *File*

merupakan kumpulan *record* yang telah terkumpul. *Record* merupakan atribut yang terbentuk dari kumpulan *field*. *Field* adalah atribut yang berasal dari sebuah *record*. *Byte* merupakan atribut dari *column* berupa huruf yang membentuk sebuah nilai dari *field*. *Bit* merupakan segmen terkecil dari gabungan data secara totalitas, yang berupa *American Standar Code Form Information Interchange(ASCII)*(Swara & Pebiadi, 2016).

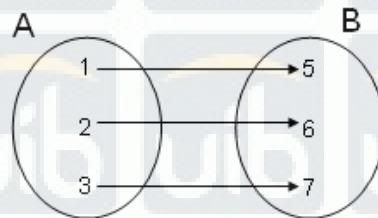
2. *Database* relasional

Adalah ikatan logik basis data dengan cara memvisualisasi dalam bentuk tabel dua dimensi, berupa sejumlah baris dan kolom dalam menampilkan atribut. Database ini menggunakan tabel dua dimensi yang terdiri baris dan kolom untuk menampilkan suatu berkas data. Dalam database relasional mencakup *Table*, *Row*, *Column*, *Field*, *Primary key*, *Foreign key*. *Table* merupakan pengarsipan yang tersusun dari satu maupun lebih baris dan kolom. *Row* adalah kumpulan nilai yang tergabung dari *Column* dalam sebuah tabel. *Column* merupakan gambaran jenis data apa pada tabel, menginput nama kolom serta tipe data tertentu. *Field* ialah suatu baris dan kolom yang saling bertemu. *Primary key* merupakan kolom unik dalam membedakan antar baris dan baris lainnya. *Foreign key* merupakan kolom-kolom yang berdasar *Primary key* didalam tabel yang sama atau pun tabel yang lain.(Farozi, 2017).

Jenis-jenis relasi dalam suatu ERD terdiri dari:

1. *One to One (1:1)*

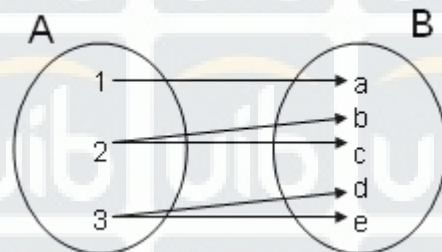
Entitas yang dimana satu anggota entitas mempunyai relasi dengan satu anggota entitas, lebih jelasnya lihat pada Gambar 2.1:



Gambar 2.1 One to One, Sumber: Farozi (2017)

2. *One to Many (1:N)*

Artinya satu anggota entitas mempunyai relasi dengan banyak anggota entitas. Untuk lebih jelasnya dapat di lihat pada Gambar 2.2 :

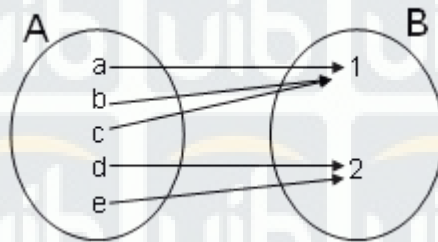


Gambar 2.2 One to Many, Sumber: Farozi (2017)

3. *Many to Many (M:N)*

Artinya lebih dari satu anggota entitas yang mempunyai relasi dengan anggota entitas lainnya. Entitas dibagi menjadi one to many ditengahnya adalah penghubung yang berupa entitas baru. Untuk jelasnya perhatikan Gambar 2.3 :

Gambar 2.3 :






Gambar 2.3 Many to Many, Sumber: Farozi (2017)

2.2.12 Unified Modeling Language (UML)

Adalah bahasa standar suatu perusahaan dalam membayangkan, merancang dan mendokumentasi sistem perangkat lunak yang diusulkan untuk merencanakan model sistem informasi (Kurniawan & Syahputra, 2018). Jenis-jenis UML yang digunakan :

1. Use Case Diagram

Merupakan diagram yang digunakan untuk menggambarkan aktor, *use case*, dan hubungan sebuah aksi yang sudah di susun serta aktor sudah diberikan nilai yang terukur (Ropianto, 2016)

Simbol	Nama	Keterangan
	Actor	Actor adalah orang yang berinteraksi dalam sebuah sistem. Aktor memiliki tanggung jawab atas data yang diinput dan mengharapkan sebuah output yang dihasilkan.
	Use Case	Use Case adalah gambaran fungsional dalam sebuah sistem, sehingga aktor mengerti fungsi sistem yang dibangun.
	Association	Association adalah sebuah garis yang digunakan untuk menghubungkan Actor dengan use Case.

Gambar 2.4 Use Case Diagram, Sumber: Ropianto (2016)

2. *Entity Relational Diagram*

ERD(*Entity Relationship Diagram*) merupakan diagram yang beroperasi dalam melakukan pemodelan pada susunan dan relasi data. Penggunaan ERD dapat menurunkan tingkat kesulitan dalam proses penyusunan database yang baik(Irsajidin, 2018).

Penggolongan ERD terdiri dari beberapa komponen menurut (Rahmayu, 2016) yaitu :

1. Entitas merupakan objek-objek yang dapat didefinisikan. Dalam ERD, entitas pada umumnya berbentuk persegi panjang. Contohnya seperti lapangan basket, ruangan, kantor.
2. *Relationship* (Relasi) merupakan hubungan antara satu entitas ataupun lebih yang terhubung. Relasi menggunakan kata kerja dasar dalam penamaan sehingga mempermudah pembacaan, dan digambarkan dengan bentuk belah ketupat.
3. *Relationship Degree* (Derajat Relasi) merupakan entitas-entitas yang ikut serta dalam suatu relasi.
 - a *Binary Relationship*, suatu model hubungan antara kumpulan instansi suatu tipe entitas dua *entity* yang bersumber dari *entity* yang sama. Hubungan ini umumnya digunakan dalam proses penyusunan suatu model data.

b *Unary Relationship* merupakan model relasi antara *entity* yang berasal dari *entity set* yang sama, relasi dapat disebut sebagai *recursive relationship*.

c Atribut(*Attribute*) merupakan yang memberikan penjelasan detail sebuah entitas. Atribut terdiri dari dua jenis yaitu:

1. *Identifier (Key)*, sebuah entitas unik pada umumnya disebut sebagai *Primary key*.
2. *Descriptor (nonkey attribute)*, karakteristik entitas yang tidak unik.

d *Ternary Relationship* merupakan bentuk relasi antara kumpulan instansi dari tiga tipe entitas secara searah, yang keterlibatan terdiri dari satu entitas atau lebih.

e Kardinalitas(*Cardinality*) merupakan relasi kejadian- kejadian yang diakui kedalam dua entitas. Kardinalitas mendefinisikan sejumlah kemungkinan maksimum dan minimum dalam satu entitas yang bisa terhubung dengan entitas tunggal yang lainnya.

Beberapa bentuk Kardinalitas yakni:

1. One to One (1:1)

merupakan relasi satu kejadian suatu entitas yang terhubung pada satu kejadian dalam satu entitas yang lain.

2. One to Many(1:N)/Many to One(N:1)



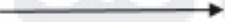

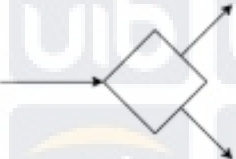

Relasi antara satu ke banyak(1:N) sama artinya dengan relasi antara banyak ke satu(N:1) terkait dari arah hubungan ini dilihat.






3. Many to Many(M:N)

Relasi banyak ke banyak(M:N) yang artinya setiap entitas pada himpunan entitas AB dapat terhubung dengan banyak entitas pada himpunan entitas AA, sebaliknya setiap entitas pada himpunan AA mempunyai hubungan banyak entitas dengan himpunan entitas AB.

3. *Activity Diagram*

ialah pemodelan alur kerja dan urutan kegiatan pada suatu proses (Sofyan et al., 2016). *Activity diagram* merupakan diagram yang berfungsi dalam menjelaskan suatu proses bisnis urutan kegiatan dalam *business modeling* agar menyederhanakan urutan kegiatan suatu bisnis (Rahmawati & Mulyono, 2016).

Simbol	Nama	Keterangan
	Initial State	<i>Initial State</i> merupakan awalan sebuah <i>activity diagram</i> dimulai
	Action or Activity State	<i>Action or Activity State</i> adalah merepresentasikan tindakan pada sebuah objek atau oleh objek. <i>Activity State</i> digambarkan menggunakan persegi panjang dengan sudut bulat.
	Action Flow or Control flows	<i>Action flow or Control flows</i> digunakan untuk menunjukkan satu transisi aktivitas ke transisi yang lainnya
	Decision node and Branching	<i>Decision node and Branching</i> digunakan ketika kondisi harus membuat <i>decision</i> . <i>Decision</i> biasanya lebih dari dua atau lebih output
	Guards	<i>Guards</i> digunakan untuk mengacukan sebuah pernyataan yang ditulis ke <i>decision node</i> menggunakan tanda panah
	Fork	<i>Fork</i> digunakan untuk mendukung lebih dari satu aktivitas secara bersamaan

	Join	<i>Join</i> digunakan untuk mendukung kegiatan dimana secara bersamaan menjadi satu. Untuk gambar disamping ini memiliki dua input dan memiliki satu output
	Merge or Merge Event	<i>Merge</i> digunakan ketika dimana skenario aktivitas yang seharusnya tidak dijalankan bersamaan harus digabungkan. <i>Merge</i> bisa menggabungkan dua atau lebih aktivitas menjadi satu jika aktivitas selanjutnya bisa terlepas dari aktivitas yang dipilih
	Swimlanes	<i>Swimlanes</i> digunakan untuk mengelompokkan aktivitas terkait dalam satu kolom atau satu baris, bisa dikelompokkan secara vertikal maupun horizontal. <i>Swimlanes</i> tidak wajib dibuat tetapi ini adalah praktik yang disarankan.
	Time Event	<i>Time event</i> digunakan pada saat skenario dimana suatu event membutuhkan waktu untuk diselesaikan
	Final State or End State	<i>Final state of End State</i> ialah tahapan akhir/selesaiannya dari sebuah <i>activity diagram</i>

Gambar 2.5 Activity Diagram, Sumber: Rahmawati & Mulyono (2016)

2.3 Aplikasi Yang Digunakan Dalam Pengembangan Sistem

2.3.1 Visual Studio Code

Ialah aplikasi *text editor* seperti *Sublime* atau *Atom*, *visual studio code* yang mendukung beraneka ragam jenis bahasa pemrograman. Contohnya bahasa

pemrograman *JavaScript*, PHP, C++, C#. Aplikasi *text editor* ini otomatis mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberikan perbedaan warna sesuai dengan tiap-tiap peran dari susunan kode tersebut. *Visual Studio Code* sudah bisa terhubung ke GitHub, jadi bisa memanggil atau menyimpan semua kode aplikasi yang sedang dikerjakan langsung dari aplikasi ini (Dzatussiri., 2017).

2.3.2 XAMPP

Merupakan perangkat lunak bebas yang mendukung banyak sistem informasi. Fungsi xampp menjadi *server localhost* yang berdiri sendiri, termasuk dari program Apache HTTP Server, database MySQL dan penerjemah bahasa yang dituliskan dengan bahasa pemrograman PHP dan Perl. Nama XAMPP kependekan dari X, X adalah sistem operasi Apache, MySQL, PHP dan Perl. Program ini menyediakan GNU *General Public License* dan bebas, adalah *web server* yang gampang dipakai serta bisa membantu tampilan website dengan dinamis (Martiana & Irfan, 2016).