

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Berdasarkan penelitian yang dilakukan (Yusran et al., 2021) dengan judul “Rancang Bangun Sistem Informasi Lowongan Kerja Berbasis Web” . Jadi untuk mendukung tujuan pemerintah mengenai ketenagakerjaan dan peran teknologi informasi, maka perlu dirancang dan dibangun media informasi untuk memberikan kemudahan bagi publik dalam memperoleh informasi pekerjaan. Salah satu media informasi berbasis internet yang dapat dimanfaatkan adalah situs web yang manfaatnya dapat diakses oleh pengguna untuk mendapatkan dan memenuhi kriteria lowongan kerja sesuai dengan kebutuhan perusahaan atau lembaga yang membutuhkan karyawan. Pengembangan sistem informasi ini menggunakan metode Waterfall, dengan bahasa pemodelan UML (Unified Modeling Language). Untuk pemrograman menggunakan bahasa pemrograman PHP dan MYSQL sebagai server basis data. Sistem informasi ini berbasis web, yang memiliki media penyimpanan terpusat di server database sehingga integritas data terjamin.

Berdasarkan penelitian dari (Suryanto et al., n.d.) (2019) dengan judul “Rancang Bangun Aplikasi Penyediaan Jasa Tenaga Kerja Berbasis Android” Metode penelitian yang digunakan meliputi studi pustaka dan perancangan

aplikasi ini menggunakan Android, Kodular, Block Coding dan metode RAD. Aplikasi jasa layanan Kerja ini memakai dua aplikasi. Aplikasi untuk customer dan mitra sendiri. Tujuan dari dibuatnya aplikasi jasa layanan kerja ini adalah menghasilkan aplikasi pencarian jasa layanan kerja yang dibutuhkan customer, Dan bagi mitra jasa layanan kerja sendiri.

Berdasarkan penelitian yang dilakukan (Susilawati, 2018) dengan judul “Rancang Bangun Sistem Informasi Lowongan Pekerjaan Berbasis Web” Adapun sistem informasi yang sudah ada hanya menampilkan informasi lowongan pekerjaan saja. Salah satu cara dalam memperbarui informasi lebih cepat adalah dengan merancang sebuah aplikasi sistem informasi pencarian lowongan pekerjaan. Pada sistem informasi lowongan pencarian kerja berbasis web yang dibangun ini akan ada informasi pencarian dari beberapa kriteria yang tersedia dan akan muncul peta untuk menampilkan lokasi. Hasilnya adalah sebuah sistem informasi pencarian lowongan kerja berbasis web yang dilengkapi dengan menu pencarian dan adanya peta.

Berdasarkan penelitian yang dilakukan oleh (Widiantoro, 2018) dengan judul “Rancang Bangun Sistem Informasi Lowongan Pekerjaan Berbasis Kerjasama”. Integrasi sistem dikembangkan menggunakan metode Software Development Life Cycle Waterfall mampu membentuk prototype yang dapat diimplementasikan diberbagai Lembaga Psikologi yang berhubungan dengan psikotest. Sistem dapat menyelesaikan permasalahan pencarian pekerjaan, pencarian pekerja, dan memperpendek waktu pencarian, sehingga pihak-pihak yang bekerjasama saling diuntungkan.

Berdasarkan penelitian yang dilakukan oleh (Eko et al., 2018) dengan judul “Rancang Bangun Aplikasi E-Jobs Berbasis Android” Untuk dapat membantu perusahaan dan pencari kerja diperlukan sebuah sistem yang dapat mencari dan mengundang pencari kerja bagi perusahaan dan mencari lowongan pekerjaan yang berada dekat dengan wilayahnya juga sesuai dengan keahlian bagi pencari kerja. Pembuatan aplikasi berbasis android dan menggunakan metode waterfall. Dengan aplikasi yang dibangun oleh penulis, diharapkan dapat membantu perusahaan menyediakan informasi pencari kerja yang berkualifikasi serta dapat mengundangnya dan membantu pelamar menemukan pekerjaan disekitar wilayahnya dan sesuai dengan keahlian mereka.

II.2. Landasan Teori

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa tinjauan pustaka pada penelitian ini yaitu:

II.2.1. Pengertian Aplikasi

Istilah aplikasi berasal dari bahasa Inggris *application* yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara umum, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta jasa pengguna aplikasi lain yang dapat digunakan oleh pengguna yang akan dituju. Menurut kamus komputer eksekutif, pengertian aplikasi merupakan pemecahan masalah yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan

maupun pemrosesan data yang diharapkan. Aplikasi biasanya berupa perangkat lunak yang berbentuk *software* yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan sebuah pekerjaan yang diinginkan. (Ayu Rizka Inayah, 2018).

II.2.2. Karyawan

Karyawan merupakan aset yang berharga bagi sebuah perusahaan dalam mencapai tujuannya fokus utama manajemen sumber daya manusia (SDM) adalah memberikan kontribusi atas suksesnya perusahaan agar produktifitas perusahaan berjalan lancar diperlukan tenaga kerja atau karyawan yang sesuai dengan prinsip “*the right man in the right place*” sejalan dengan itu maka langkah awal yang menjadi kunci utama yaitu proses rekrutmen dan seleksi untuk merekrut tenaga kerja sesuai dengan kebutuhanny proses seleksi merupakan serangkaian langkah kegiatan yang digunakan untuk memutuskan kandidat (calon karyawan) yang dapat di tempatkan secara tepat saat ini di mana persaingan untuk mendapatkan pekerjaan semakin kuat perusahaan sering kali mengalami kesulitan dalam menentukan kandidat yang tepat mengingat bahwa ada banyak kandidat yang tersedia tetapi sangat sedikit yang memiliki kualifikasi yang memadai apabila perusahaan sudah mempunyai gambaran tentang hasil analisis pekerjaan (Sonty Lena ; 2018).

II.2.3. *Rekrutment*

Pengertian *rekrutmen* adalah putusan sumber daya manusia (SDM) berupa banyak dibutuhkan, kapan dibutuhkan, serta pengetahuan, keterampilan, kemampuan khusus yang dimiliki. Perekrutan karyawan merupakan suatu proses atau tindakan yang dilakukan oleh organisasi untuk mendapatkan tambahan karyawan melalui beberapa tahapan mencakup identifikasi dan evaluasi sumber-sumber perekrutan tenaga kerja, menentukan kebutuhan tenaga kerja, proses seleksi, penempatan, dan orientasi tenaga kerja. Perekrutan karyawan bertujuan menyediakan karyawan yang cukup agar manajemen dapat memilih karyawan yang memenuhi kualifikasi yang mereka perlukan. (Rizky, 2019)

Pengertian *rekrutmen* adalah putusan sumber daya manusia (SDM) berupa banyak dibutuhkan, kapan dibutuhkan, serta pengetahuan, keterampilan, kemampuan khusus yang dimiliki. Perekrutan karyawan merupakan suatu proses atau tindakan yang dilakukan oleh organisasi untuk mendapatkan tambahan karyawan melalui beberapa tahapan mencakup identifikasi dan evaluasi sumber-sumber perekrutan tenaga kerja, menentukan kebutuhan tenaga kerja, proses seleksi, penempatan, dan orientasi tenaga kerja. Perekrutan karyawan bertujuan menyediakan karyawan yang cukup agar manajemen dapat memilih karyawan yang memenuhi kualifikasi yang mereka perlukan. (Eko et al., 2018)

II.2.4. *Pengertian Android*

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka

bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *TMobile*, dan *Nvidia*. Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode-kode *Android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler

Android merupakan sistem operasi *mobile*. *Android* tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data-data ponsel sekalipun, atau data sistem sendiri. *Android* merupakan sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Beberapa pengertian lain dari *Android*, yaitu :

1. Merupakan *platform* terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli *Google Inc.* dari *Android Inc.*
3. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time environment* yang disebut DVM (*Dalvik Virtual Machine*) yang

telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil. (Ceryna Dewi et al., 2018)

II.2.5. *Android Studio*

Android Studio adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi *Android* yang dibangun berdasarkan perangkat lunak IntelliJ IDEA. Selain editor kode dan alat pengembang yang didesain khusus untuk pengembangan *android*, *Android studio* merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi *android*. *Android studio* menyediakan banyak fitur untuk meningkatkan produktivitas dalam pengembangan aplikasi berbasis android seperti tersedianya sistem pembangunan berbasis *gradle* yang *fleksibel*, emulator yang memiliki banyak fitur, integrasi dengan GitHub, dukungan bawaan untuk *Google Cloud Platform* sehingga mempermudah pengembang aplikasi untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine*. (Kaban et al., 2019)

II.2.6. Perkembangan *Android*

Adapun perkembangan *Android* adalah sebagai berikut :

a. *Android* versi 1.1

Android memang diluncurkan pertama kali pada tahun 2007, namun sistem operasi ini mulai dirilis dan diterapkan ke berbagai gadget pada tanggal 9 Maret 2009 silam. *Android* versi 1.1 merupakan *Android* awal yang dimana versi ini baru memberikan sentuhan dibeberapa aplikasinya seperti sistem antar

muka bagi pengguna (*user interface*) yang lebih baik, serta beberapa aplikasi yang lain.

b. *Android* versi 1.5 (*Cupcake*)

Pada bulan Mei 2009 *Android* kembali mengalami perubahan versi. *Android* versi 1.1 kemudian disempurnakan dengan *Android* versi 1.5 atau yang dikenal sebagai *Android Cupcake*.

c. *Android* versi 1.6 (*Donut*)

Donut (versi 1.6) diluncurkan dalam tempo kurang dari 4 bulan semenjak peluncuran perdana *Android Cupcake*, yaitu pada bulan September 2009.

d. *Android* versi 2.0/2.1 (*Eclair*)

Masih ditahun yang sama, *Android* kembali merilis *operating* sistem versi terbarunya, yaitu *Android* versi 2.0/2.1 *Eclair*. *Android Eclair* diluncurkan oleh *Google* 3 bulan setelah peluncuran.

e. *Android* versi 2.2 (*Froyo: Frozen Yoghurt*)

Butuh 5 bulan bagi *Google* untuk melakukan regenerasi dari *Android Eclair* versi sebelumnya ke versi *Froyo Frozen Yoghurt*. Pada tanggal 20 Mei 2010, *Android* versi 2.2 alias *Android Froyo* ini dirilis.

f. *Android* versi 2.3 (*Gingerbread*)

7 bulan kemudian *Android* kembali melakukan gebrakan dengan merilis kembali *Android* versi 2.3 atau yang dikenal sebagai *Android Gingerbread*.

g. *Android* versi 3.0/3.1 (*Honeycomb*)

Pada bulan Mei 2011 *Android* versi 3.0/3.1 atau *Android Honeycom* dirilis. *Android Honeycomb* merupakan sebuah sistem operasi *Android* yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis *Android*.

h. *Android* versi 4.0 (ICS: *Ice Cream Sandwich*)

Android ICS atau *Ice Cream Sandwich* juga dirilis pada tahun yang sama dengan *Honeycomb*, yaitu pada bulan Oktober 2011.

i. *Android* versi 4.1 (*Jelly Bean*)

Android Jelly Bean merupakan versi *Android* yang terbaru pada saat ini. Salah satu *gadget* yang menggunakan sistem operasi *Jelly Bean* adalah *Google Nexus 7* yang diprakarsai oleh ASUS, *vendor* asal Taiwan yang juga menjadi teman satu kampung halaman dengan Acer.

j. *Android* versi 4.4 (*Kit Kat*)

Kehadiran *android kitkat* merupakan peluncuran produk OS anyar yang diluncurkan pada 4 september 2013, sebelumnya banyak kabar beredar jikalau android akan meluncurkan OS baru yang bernama *Android Key Lime Pie* namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan OS *Android KitKat* yang sebagian besar orang sudah familiar dengan itu.

k. *Android* versi 5.0.2 (*Lollipop*)

Android Lollipop merupakan keberadaan OS *Android* yang memang saat ini sudah menjadi *trend* baru di industri *smartphone*, hal ini tak lepas dari keunikan dan kelebihan yang banyak di miliki dari OS tersebut. Kehadiran

android versi ini amat di nanti oleh sekian banyak orang karna diharapkan sistem operasi *Lollipop* ini bias lebih baik dibandingkan versi-versi sebelumnya.

1. *Android* versi 6.0 (*Marshmallow*)

Android 6.0 *Marshmallow* adalah versi dari sistem operasi *mobile Android*. Pertama kali diperkenalkan Mei 2015 di *Google I / O* di bawah kode nama *Android M*, secara resmi dirilis pada Oktober 2015. *Android Marshmallow* memperkenalkan model izin aplikasi didesain ulang sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. (Syani & Werstantia, 2018)

Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

1. Kelebihan *Android*

- a. *Switching* dan multitasking yang lebih baik *Android* sangat mendukung *multitasking* aplikasi, kini hal tersebut kembali ditingkatkan. Dalam *Honeycomb* pengguna dapat dengan mudah berpindah aplikasi hanya dengan menyentuh sebuah *icon* pada system bar.
- b. Kapasitas yang lebih baik untuk beragam *widget* Kapabilitas terhadap beragam *widget* dijanjikan bakal makin memanjakan para penggunanya. Contohnya *widget* untuk email *Gmail* yang dipamerkan *Google*, pengguna tidak perlu membuka aplikasi *Gmail* untuk melihat isi di dalamnya.

- c. Peningkatan kemampuan *copy-paste*. Beberapa seri *Android* terdahulu memang sudah bisa melakukan *copy-paste*, namun beberapa pengguna masalah pemilihan teks yang agak sulit. Kini hal tersebut coba diselesaikan, selain *copy-paste* Google juga menambah share it pada teks yang diseleksi.
- d. *Browser Chrome* Lebih Cepat. Ada satu fitur yang hilang dalam *browser Chrome* yang diletakkan pada *Android* terdahulu, kemampuan *Tab Chrome* yang ada di *Honeycomb* kini dapat melakukan hal tersebut. Selain itu pengguna juga bisa mensinkronisasi antara *browser* di ponsel dengan *Chrome* yang ada di komputer.
- e. Notifikasi yang mudah terdengar. Dengan layar yang lebih besar, otomatis membuat *google* lebih leluasa menempatkan notifikasi pada layar.
- f. Peningkatan *Drag and Drop* serta *Multitouch*. Ukuran layar yang lebih besar, menuntut *google* untuk meningkatkan kemampuan *multitouch* di dalam *Android*, tak terkecuali fitur *drag and drop*. Pada demo yang ditayangkan, pengguna bisa melakukan *drag and drop* untuk memindahkan *email* di dalam aplikasi *gmail* (Sulihati , 2019).

2. Kekurangan *Android*

- a. Koneksi Internet yang terus menerus. Kebanyakan ponsel *Android* memerlukan koneksi internet yang simultan atau terus menerus aktif, itu artinya anda harus siap berlangganan paket GPRS 20 Jurnal Sains dan Teknologi Utama, Volume XI, Nomor 1, April 2019 yang sesuai dengan kebutuhan dan batre yang boros karena GPRS yang terus menyala.

- b. Iklan. Aplikasi di Ponsel *Android* memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu ada Iklan yang terpampang (Sulihati , 2019).

II.2.8. Basis Data (*Database*)

Pangkalan data atau basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*Relational database management system* (DBMS)). Sistem basis data dipelajari dalam ilmu informasi. Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Model yang umum digunakan sekarang adalah model relasional, yang mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel (Neni Purwati dan Hendra Kurniawan, 2018).

II.2.9. Normalisasi

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data / *database*, teknik pengelompokkan atribut dari suatu relasi sehingga

membentuk struktur relasi yang baik tanpa redundansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data (Mukhlisulfatih Latief : 2019).

1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

2. Tahapan Normalisasi :

- a. Bentuk tidak normal : Menghilangkan perulangan grup.

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- Tidak ada perulangan record data dalam tabel.
- Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan *array*.
- Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

- b. Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.
Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

- c. Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan key sudah tergantung penuh terhadap *key*-nya.

- d. Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan *key* tidak tergantung transitif terhadap *key*-nya.

II.2.10.MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (*RDBMS*) yang didistribusikan secara gratis dibawah lisensi *GPL* (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; *SQL* (*Structured Query Language*). *SQL* adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basis data (*DBMS*) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah *SQL* yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya.

Sebagai peladen basis data, *MySQL* mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modul operasi non-transaksional, *MySQL* dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data kompetitor lainnya. Namun pada modul non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modul non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (*wordpress*), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modul basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja *MySQL* pada modul transaksional tidak secepat unjuk kerja pada modul non-transaksional.

MySQL pada awalnya diciptakan pada tahun 1979, oleh *Michael "Monty" Widenius*, seorang *programmer* komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan *UNIREG* yang menggunakan koneksi *low-level ISAM database engine* dengan *indexing*. Pada saat itu Monty bekerja pada perusahaan bernama *TcX* di Swedia. *TcX* pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan *UNIREG* sebagai sistem *database*. Namun sayangnya, *UNIREG* dianggap tidak cocok untuk *database* yang dinamis seperti web. (Herpendi, 2019)

II.2.11.XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai

server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat *men-download* langsung dari *web* resminya. (Randi V Palit, dkk, 2018).

II.2.12.PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah *web* dan bisa digunakan pada dokumen *HTML*. *PHP* dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen *HTML* yang dapat mengakses *database* menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi dimana aplikasi tersebut yang dibangun oleh *PHP* pada umumnya akan memberikan hasil pada *web browser*, tetapi proses secara keseluruhan dijalankan di *server*. (Saipul Anwar, dkk : 2019)

II.2.13. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah

sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

Unified Modeling Language (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Alfina & Harahap, 2019)

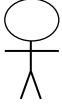
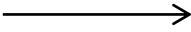
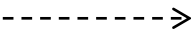
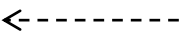
II.2.13.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut

.Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel

II.1 dibawah ini :

Tabel II.1. Simbol Use Case

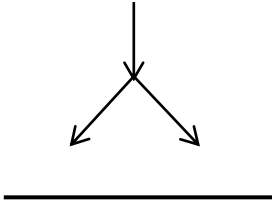
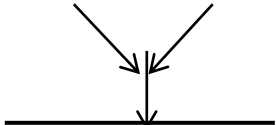
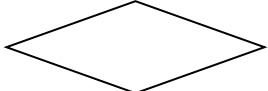
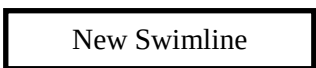
Gambar	Keterangan	Deskripsi
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Ade Hendini, 2019)

II.2.13.2. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol Activity Diagram

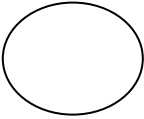
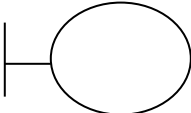
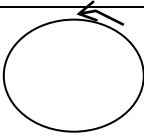
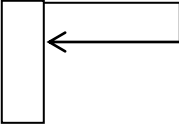
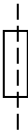
Gambar	Keterangan	Deskripsi
	<i>Start point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i>	Akhir aktifitas.
	<i>Activites</i>	Menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan).	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i>	Pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini, 2019 : 109)

II.2.13.3. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan	Deskripsi
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, 2019 : 110)

II.2.13.4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class* diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint*

yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2019 : 111).

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini, 2019 : 110)