

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

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

Berdasarkan penelitian yang dilakukan Ardelia Astriany Rizky (2019) Penelitian ini bertujuan untuk menganalisis Sistem Informasi perekrutan Karyawan baru di PT Ria Indah Mandiri yang saat ini masih manual dan belum terkomputerisasi, dimana sistem perekrutan karyawan yang berjalan saat ini menyebabkan kurang efektifnya dalam proses penseleksian dan terjadinya penumpukan arsip lamaran kerja. Untuk mengatasi permasalahan tersebut dibangunlah Sistem Informasi Rekrutmen Karyawan Berbasis *Web*. Sistem Informasi yang dibangun ini menggunakan pemograman *PHP* dan *MySQL database server*.

Berdasarkan penelitian dari Fajari Sidik (2018) bahwa Perkembangan teknologi dari masa ke masa semakin maju bisa kita lihat diberbagai aspek kehidupan misalnya untuk berkomunikasi dengan orang di dunia luar cukup menggunakan email atau media *online* lainnya atau untuk melakukan pendaftaran sekolah cukup terkoneksi ke internet dan masuk ke halaman *website* yang dituju sehingga untuk semua aktivitas yang kita lakukan menjadi lebih mudah dan efisien. *Website* ini bisa mempermudah para calon siswa dalam mendaftar sekolah karena cukup terhubung ke internet saja, dan pendaftaran bisa dilakukan dimana

saja dan kapan saja. Sekolah pun dalam melakukan pencatatan laporan keuangan maupun laporan data penerimaan peserta didik baru lebih mudah karena bisa melakukan pengecekan data pendaftar melalui *website* secara langsung. Sedangkan sistem yang akan dirancang oleh penulis adalah penelitian yang berbasis android yang mempermudah siswa dalam melakukan pendaftaran dan sistem yang menggunakan Bahasa pemrograman android studio dan mysql sebagai database. Tujuan dari penelitian yang dilakukan penulis adalah untuk mempermudah siswa dan SMA Budi Murni I Medan dalam melakukan proses pendaftaran dan pembayaran biaya pendaftaran siswa baru.

Berdasarkan penelitian yang dilakukan oleh Achmad Syafi ZAIN (2018) bahwa Saat ini belum banyak sekolah di Indonesia mengembangkan sistem informasi Penerimaan Siswa Baru (PSB) pada sistem informasi berbasis web. Dengan manfaat dan kemudahan teknologi yang sudah ada, sudah seharusnya pengembangan sistem informasi penerimaan siswa baru (PSB) ini dikembangkan oleh tiap-tiap sekolah. Sistem PSB di SMA 1 Annuqayah masih menggunakan sistem manual atau melalui brosur. Dari permasalahan tersebut maka tujuan dari penelitian ini adalah untuk membantu pihak sekolah dalam proses penerimaan siswa baru berbasis web agar proses PSB berjalan efektif. Sedangkan sistem yang akan dirancang oleh penulis dengan menggunakan aplikasi *android studio* dan aplikasinya bisa di download dengan menggunakan fitur aplikasi *play store* sehingga mempermudah siswa melakukan pendaftaran di mana pun.

Berdasarkan penelitian yang dilakukan Sonty Lena (2018) dengan judul Tujuan yang ingin dicapai dalam penelitian ini adalah untuk mengetahui

bagaimana proses rekrutmen dan seleksi karyawan pada PT Bestprofit futures bandung untuk mengetahui hambatan-hambatan yang terjadi pada proses rekrutmen dan seleksi karyawan baru untuk mengatasi hambatan dalam proses rekrutmen karyawan baru metode pembahasan yang digunakan dalam penelitian ini adalah dengan menggunakan metode diskriptif kuantitatif penelitian itu untuk mengetahui proses rekrutmen dan seleksi para pelamar karyawan baru di PT. Bestprofit futures bandung yang melalui data- data yang di dapat pada perusahaan dengan cara wawancara dan observasi pada perusahaan. Adapun saran yang diberikan perusahaan berdasarkan kesimpulan yang didapat yaitu perlu dilakukan perekrutan dan seleksi yang lebih cermat dan mendasar agar supaya tidak terjadi karyawan baru yang tidak sesuai antara bidang jurusan dan penempatan kerja.

. Berdasarkan penelitian yang dilakukan oleh Putri Hapsari (2017) Penelitian ini bertujuan untuk menghasilkan perangkat lunak Sistem Informasi Penerimaan Karyawan Baru Berbasis Web dengan PHP dan SQL yang mampu mengelola pelaksanaan Penerimaan Karyawan Baru yaitu pendaftaran, tes pengetahuan umum, tes psikotest dan pengumuman. Pada penelitian ini juga ditentukan tingkat kelayakan perangkat lunak yang telah dibuat. Jenis penelitian ini adalah *Research and Development*. Metode pengembangan perangkat lunak menggunakan *Waterfall Model*. Pengujian perangkat lunak dilakukan melalui pengujian *white-box* dan *black-box*. Teknik pengumpulan data pada penelitian menggunakan wawancara. Hasil penelitian menunjukkan bahwa Sistem Informasi Penerimaan Karyawan Baru Berbasis *Web* dengan PHP dan SQL ini mampu mengelola pelaksanaan Penerimaan Karyawan Baru

## **II.2. Landasan Teori**

### **II.2.1. Aplikasi**

Pengertian aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer. Sedangkan pengertian penjualan adalah suatu proses seseorang atau organisasi untuk meyakinkan *customer* membeli produk yang ditawarkan. Aplikasi *mobile* dapat diartikan sebagai sebuah produk dari sistem komputasi *mobile*, yaitu sistem komputasi yang dapat dengan mudah dipindahkan secara fisik dan yang komputasi kemampuan dapat digunakan saat mereka sedang dipindahkan. Contohnya adalah *personal digital assistant* (PDA), *smartphone* dan ponsel. (Listianto et al : 2017)

### **II.2.2. Penerimaan Siswa Baru**

Penerimaan siswa baru merupakan gerbang awal yang harus dilalui peserta didik dan sekolah didalam penyaringan objek-objek pendidikan. Peristiwa penting bagi suatu sekolah, karena peristiwa ini merupakan titik awal yang menentukan kelancaran tugas suatu sekolah. Kesalahan dalam penerimaan siswa baru dapat menentukan sukses tidaknya usaha pendidikan disekolah yang bersangkutan. Penerimaan siswa baru dilakukan bukanlah suatu hal yang ringan. Sekolah harus menyiapkan strategi-strategi yang tepat dalam menjalankannya, supaya dapat menarik siswasiswa yang berkualitas yang mana input sekolah juga bisa lebih baik sehingga proses belajar bisa maksimal dan kualitas sekolah meningkat. (Sarwindah : 2018)

### II.2.3. *Android*

*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.2.4. *Android Studio*

*Android Studio* adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi *Android*, berdasarkan IntelliJ IDEA. Konsep yang dimiliki *Android Inc* ternyata menggugah minat *Google* untuk memilikinya. Pada bulan Agustus 2005, Akhirnya *Android Inc* diakuisisi oleh *Google Inc.* seluruh sahamnya dibeli oleh *Google*. Banyak yang memperkirakan nilai pembelian *Android Inc* Oleh *Google* adalah sebesar USD 50 juta. saat itu banyak yang berspekulasi bahwa akuisisi ini

adalah langkah awal yang dilakukan *Google* untuk masuk ke pasar mobile phone. Maka dari situlah banyak pengembang sistem maupun software berlomba-lomba untuk membuat atau merancang sistem *Android* menggunakan software – software yang support dengan *Android*, dan sebagai contoh disini kita pengenalan *Android Studio*. Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, *Android Studio* menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi *Android*, misalnya:

1. Sistem versi berbasis *Gradle* yang *fleksibel*
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat *Android*
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah - masalah lain
8. Dukungan C++ dan *NDK*
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*

Laman ini berisi pengantar dasar fitur-fitur *Android Studio*. Untuk memperoleh rangkuman perubahan terbaru, lihat Catatan Rilis *Android Studio* (Ceryna Dewi et al : 2018)

#### II.2.5. Basis Data (*Database*)

Basis data atau *database* merupakan koleksi dari data-data yang terorganisir dengan rapi sehingga data dapat dengan mudah disimpan dan dimanipulasi. Kita dapat menjumpai pemanfaatan *database* dalam kehidupan sehari-hari, seperti penggunaan mesin ATM, sistem akademi universitas/sekolah, sistem informasi penjualan. Salah satu tujuan dari *database* adalah memberikan pengguna suatu pandangan abstrak dari data, yaitu sistem menyembunyikan rincian bagaimana data disimpan dan dipelihara. Sistem *database* harus dibuat semudah mungkin untuk dimengerti karena kebanyakan pengguna sistem *database* adalah orang-orang yang kurang terlatih di bidang teknologi. (Ceryna Dewi et al : 2018)

#### II.2.6. MySQL

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat database yang bersifat *Open Source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *Windows* maupun *linux*. MySQL juga merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi user. MySQL juga sering dikenal dengan nama sistem manajemen *database*

relasional. Suatu *database* relasional menyimpan data dalam tabel yang terpisah. Tabel - tabel tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa tabel dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *PHP MyAdmin* dengan aplikasi *XAMPP*. (Anwar et al : 2016)

#### II.2.7. 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. (Riyadli et al : 2020)

#### II.2.8. PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa

PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* yang akan menterjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”. (Firman et al : 2016)

#### **II.2.9. 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)

UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat dan pengembangan sistem. (Urva et al : 2018)

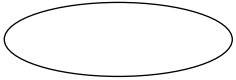
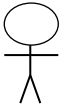
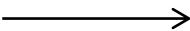
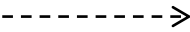
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah Empat alat bantu sebagai berikut: *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

### **1. Use Case Diagram**

*Use Case Diagram* merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi

antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan Use Case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Symbol-simbol yang digunakan dalam Use Case Diagram Yaitu

**Tabel II.4. Simbol Use Case**

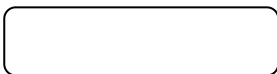
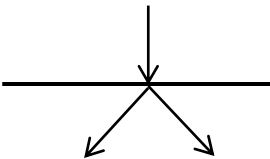
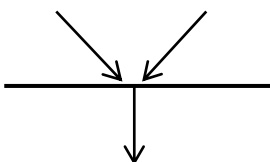
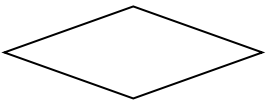
| Gambar                                                                              | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <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 adalah <i>abstraction</i> dari orang atau 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. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> . |
|  | Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.                                                                                                                                                                                                                                     |
|  | Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan 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 : Gellysa Urva : 2018)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol- simbol yang digunakan dalam *activity diagram* yaitu:

**Tabel II.5. Simbol *Activity Diagram***

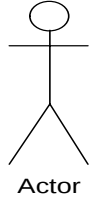
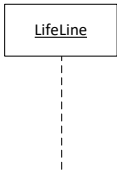
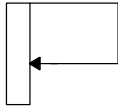
| Gambar                                                                              | Keterangan                                                                                                                                                 |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <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 paralel menjadi satu. |
|  | <i>Join</i> (penggabungan) atau rake, 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 diagram</i> untuk menunjukkan siapa melakukan apa.                                                                 |

(Sumber : Gellysa Urva : 2018)

### 3. Sequence Diagram

*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antara objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

**Tabel II.6. Sequence Diagram**

| Gambar                                                                              | Keterangan                                                                                                                                                                           |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | Orang, Proses atau sistem lain yang berinteraksi dengan sistem informasi.                                                                                                            |
|   | Merepresentasikan entitas tunggal dalam sequence diagram, digambarkan dengan kotak. Entitas ini memiliki nama, <i>Stereotype</i> atau berupa <i>instance</i> .                       |
|  | <i>Message</i> , simbol mengirim pesan antar <i>class</i> .                                                                                                                          |
|  | <i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.                                                                                                |
|  | <i>Activation</i> ,<br><i>Activation</i> , <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 : Gellysa Urva : 2018)

### 4. Class Diagram (Diagram Kelas)

Merupakan hubungan antara kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan

tanggung jawab entitas yang merupakan 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 (*Operations/ Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

**Tabel II.7. Multiplicity Class Diagram**

| <b>Multiplicity</b> | <b>Penjelasan</b>                                               |
|---------------------|-----------------------------------------------------------------|
| 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 : Gellysa Urva : 2018)**