

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Pada penelitian ini penulis menggunakan beberapa tinjauan pustaka yang akan digunakan untuk membantu penelitian penulis, berikut ini merupakan beberapa tinjauan pustaka yang akan digunakan yaitu :

1. Penelitian yang digunakan oleh (Awani, Nugraha and Puspita, 2018). Perancangan *Performance Appraisal* Menggunakan Metode *Behaviorally Anchored Rating Scales (BARS)*, Penelitian ini mengulas tentang kurangnya perhatian terhadap kehadiran kerja disebabkan oleh tidak adanya *Performance Appraisal* yang baik.
2. Penelitian yang digunakan oleh (Kustiadi, 2018). Perancangan Penilaian Kinerja Dengan Metode *Behaviorally Anchored Rating Scales (BARS)* di PT. Zeno Alumi Indonesia, Survei penelitian tentang menawarkan dukungan dan lebih jauh lagi hasil terbaik untuk klien dan juga rekan usahanya.
3. Penelitian yang digunakan oleh (Purnamasari and Ardhiyani, 2021). Peningkatan Persepsi Karyawan Terhadap Objektivitas Penilaian Kinerja Menggunakan Metode *Behaviorally Anchored Rating Scales (BARS)* dan *Management By Objectives (MBO)*, Penelitian ini mengulas tentang agar dapat menyediakan pelayanan terbaik kepada *customer* dan mitra bisnisnya.
4. Penelitian yang digunakan oleh (Rouza and Yanto, 2020). Penerapan *Behaviorally Anchored Rating Scales (BARS)* Berbasis *Web* Dalam

Penilaian Kinerja Karyawan, Penelitian ini mengulas tentang pengukuran kinerja karyawan masih menggunakan sistem yang manual, belum adanya teknologi IT ataupun sistem informasi sebagai alat bantu yang digunakan untuk proses pengarsipan, pencarian data dan mempercepat proses pelaporan penilaian kinerja karyawan baik itu perbulan.

5. Penelitian yang digunakan oleh (Evita and Muizu, 2017) Evita, S. N. Penilaian Kinerja Karyawan Dengan Menggunakan Metode *Behaviorally Anchored Rating Scales (BARS)* dan *Management By Objectives (MBO)*, Penelitian ini mengulas tentang laporan kerja harian pekerja. Bagaimanapun, sedikit demi sedikit, kedua teknik ini masih dianggap tidak efektif.
6. Penelitian yang digunakan oleh (Triwibisono *et al.*, 2021). Perancangan penilaian kinerja pegawai menggunakan Metode *Behaviorally Anchored Rating Scales (BARS)* di PT. Widya Trans Cargo, penelitian ini mengulas tentang penilaian yang dilakukan pada divisi operasi hanya dengan mengamati setiap karyawannya seperti kehadiran, hasil pengiriman yang dilakukan, kedisiplinan, dan *attitude* karyawan, sehingga perusahaan tidak dapat mengetahui performansi dan pencapaian yang sudah didapatkan setiap karyawannya.
7. Penelitian yang digunakan oleh (Putra *et al.*, 2021). *Literature Review Analisis Kinerja SDM Menggunakan Metode Behaviorally Anchored Rating Scales (BARS)*, penelitian ini mengulas tentang hasil yang ditemukan menunjukkan bahwa penerapan metode *BARS* lebih banyak

digunakan pada organisasi swasta dibandingkan dengan organisasi negeri.

8. Penelitian yang digunakan oleh (Yanti and Bast, 2015). Sistem Informasi Manajemen Penggajian dan Penilaian Kinerja Pegawai Pada SMK Taman Siswa, penelitian ini mengulas tentang menghasilkan kerangka data keuangan dewan dan pelaksanaan ujian perwakilan SMK Taman Siswa Teluk Betung yang dapat memberikan akomodasi dalam pembiayaan pelaksanaan latihan dan evaluasi pelaksanaan pekerja.

Perbedaan Metode *Behaviorally Anchored Rating Scales (BARS)* dengan metode yang lain yaitu, metode penilaian yang memfokuskan kombinasi kritis dan diperingatkan, spesifikasi kinerja pegawai dilakukan skala spesifikasi tentang kinerja pegawai baik dan buruk.

### **2.1.1 *Behaviorally Anchor Rating Scale (BARS)***

Menurut Siti Nova, Wa Ode Zusnita Muizu dan Raden Tri Wahyu Atmojo (2017) metode penilaian kinerja dengan model *Behaviorally Anchored Rating Scales (BARS)* merupakan metode penilaian yang menggabungkan pendekatan perilaku kerja dengan sifat pribadi. Metode *BARS* terdiri dari *anchor* 5 sampai 10 skala perilaku vertikal untuk setiap indikator. Untuk setiap aspek, 5 hingga 10 *anchor*. *anchor* yang dimaksud, adalah cara berperilaku tertentu yang menunjukkan pelaksanaan untuk setiap aspek. *Anchor* tersusun dari nilai terbesar hingga terkecil. *anchor* dapat berupa *critical incident* yang diperoleh melalui analisa pekerja. Metode ini sebagian besar dikembangkan oleh kelompok yang terdiri dari ahli materi pelajaran Sumber Daya Manusia, Manajer, dan Karyawan. Kelompok ini dipercaya untuk mengenali atribut aspek presentasi dan membedakan 5 sampai

10 kesempatan yang dimaksudkan untuk setiap aspek. Kemudian, kejadian tertentu dieksplorasi dan dievaluasi oleh semua rekan. Acara-acara luar biasa yang dipilih kemudian ditempatkan pada skala dari yang paling tinggi ke yang paling rendah.



Gambar 2. 1 Prosedur Penelitian Penyusunan Metode BARS (Suradiraja, 2015).

### 2.1.2 Penilaian Kinerja Pegawai

Menurut Nelly and Adam Mahmudi (2015) Penilaian Kinerja Pekerja adalah sekumpulan kerangka kerja yang digunakan untuk memastikan produktivitas dan kelangsungan hidup dalam serangkaian kegiatan. Sedangkan Menurut Mulyadi dan Setyawan dalam Mahmudi (2015) Evaluasi pelaksanaan juga dapat diartikan sebagai jaminan yang terputus-putus dari kelangsungan fungsional asosiasi dan tenaga kerjanya berdasarkan pedoman dan langkah-langkah yang telah ditetapkan sebelumnya. Eksekusi ujian adalah strategi untuk memperkirakan presentasi orang-orang dalam suatu asosiasi terkait dengan tingkat komitmen individu atau eksekusi dalam menyelesaikan tanggung jawab yang mereka mampu.

Menurut Widodo (2015) bahwa kinerja karyawan dipengaruhi oleh berbagai faktor, antara lain:

1. Sasaran : adanya rumusan sasaran yang jelas tentang apa yang diharapkan oleh perusahaan untuk dicapai.
2. Standar : apa ukurannya bahwa seseorang telah berhasil mencapai sasaran yang diinginkan oleh perusahaan.
3. Umpan Balik : Informasi terhadap kegiatan yang berkaitan dengan upaya mencapai sasaran sesuai standar yang telah di tentukan.
4. Peluang : Beri kesempatan orang itu untuk melaksanakan tugasnya mencapai sasaran tersebut.
5. Sarana : Sediakan sarana yang diperlukan untuk mendukung pelaksanaan tugasnya.
6. Kompetensi : Beri pelatihan yang efektif, yaitu bukan sekedar belajar tentang sesuatu, tetapi belajar bagaimana melakukan sesuatu.
7. Motivasi : Harus bisa menjawab pertanyaan “mengapa saya harus melakukan pekerjaan ini?”

### **2.1.3 Skala Penilaian Kinerja**

Skala Penilaian Kinerja merupakan penilaian eksekusi untuk setiap aspek. *Anchor* diatur dari yang paling berharga hingga yang paling tidak berharga. *Anchor* dapat berupa episode dasar yang diperoleh melalui ujian kerja. Teknik ini sebagian besar dikembangkan oleh kelompok yang terdiri dari profesional terlatih Sumber Daya Manusia, Manajer, dan Karyawan. Kelompok ini dipercaya untuk membedakan atribut aspek-aspek pameran dan mengenali 5 hingga 10 kesempatan yang dimaksudkan untuk setiap aspek. Kemudian, pada saat itu, Periode tersebut

diperiksa dan disurvei oleh semua rekan. Acara-acara unik yang dipilih kemudian dimasukkan ke dalam skala dari yang paling penting hingga yang paling tidak penting (Kustiadi, 2018).

#### **2.1.4 Teori Likert**

Menurut (Nempung, Setiyaningsih and Syamsiah, 2015) Skala *Likert* adalah suatu skala *psikometrik* yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Nama skala ini diambil dari nama *Rensis Likert*, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala *Likert*, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan 5 pilihan skala dengan format berikut:

1. Sangat Bagus
2. Bagus
3. Cukup
4. Kurang
5. Sangat Kurang

#### **2.1.5 Pengertian Sistem**

Menurut (Agustin, 2018) Sistem adalah seperangkat komponen yang saling berhubungan dan saling bekerja sama untuk mencapai beberapa tujuan. Selain itu pengertian yang lain sistem terdiri dari unsur-unsur dan masukan (*input*), pengolahan (*processing*), serta keluaran (*output*). Dengan demikian, secara sederhana sistem dapat diartikan sebagai kumpulan atau himpunan dari unsur atau variabel-variabel yang terorganisasi, saling berinteraksi dan saling bergantung satu

sama lain. Sistem di desain untuk memperbaiki atau meningkatkan pemrosesan informasi. Setelah dirancang, sistem diperkenalkan dan diterapkan ke dalam organisasi penggunanya. Jika sistem yang diterapkan itu digunakan maka implementasi sistem dapat dikatakan berhasil. Sedangkan jika para penggunanya menolak sistem yang diterapkan, maka sistem itu dapat digolongkan gagal.

### **2.1.6 XAMPP**

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak kerangka kerja, adalah kumpulan dari beberapa proyek. Kapasitasnya sebagai server sendiri (*localhost*), yang terdiri dari program Apache HTTP Server, basis informasi *MySQL*, dan juru bahasa yang ditulis dalam dialek pemrograman *PHP* dan *Perl*. Program ini dapat diakses di bawah GNU (*General Public License*) dan gratis, merupakan server web sederhana yang dapat digunakan untuk melayani tampilan halaman website dinamis. (Sari, 2016).

### **2.1.7 Bahasa Pemrograman**

#### 1. *Hyper Text Markup Language (HTML)*

Menurut Sidik dan Husni (2017:10) “*HTML* kependekan dari *Hyperlink Text Markup Language*. Dokumen *HTML* adalah teks murni yang dapat dibuat dengan pengolah kata apa pun. Laporan-laporan ini dikenal sebagai halaman situs web. Laporan *HTML* adalah catatan yang diperkenalkan di browser internet peselancar. Catatan ini pada umumnya berisi data atau antarmuka aplikasi di web”.

Berdasarkan uraian diatas maka kesimpulannya *HTML* adalah bahasa yang digunakan untuk menulis halaman *web* sebagai pertukaran dokumen

*web* dan dokumen ini umumnya berisi informasi atau *interface* aplikasi di dalam *internet*.

## 2. *Cascading Style Sheet (CSS)*

*Cascading Style Sheet (CSS)* merupakan sebuah bahasa pemrograman web yang digunakan untuk mengontrol dan membangun bagian-bagian yang berbeda pada web sehingga tampilan web akan lebih memuaskan, teratur, dan seragam. Kapasitas *CSS* sebagai pengaturan desain, bingkai, teks, gambar, warna, tabel, spasi, dan sebagainya. (sanusi, 2018).

Berdasarkan pengertian diatas dapat disimpulkan adalah sebuah perangkat lunak (*software*) yang dikembangkan untuk gaya 16 pengaturan halaman *web* dan pengelola isi (*content*) yang ada di dalam *website* sehingga tampilan *web* akan lebih rapi, terstruktur dan seragam.

## 3. *PHP*

Menurut Hikmah, dkk (2015:1) ”*PHP* merupakan kependekan dari *Hypertext Preprocessor*. *PHP* tergolong sebagai perangkat lunak *open source* yang diatur dalam aturan *general purpose licences (GPL)*. Bahasa pemograman *PHP* sangat cocok dikembangkan dalam lingkungan *web*, karena *PHP* bisa diletakkan pada script *HTML* atau sebaliknya. *PHP* dikhususkan untuk pengembangan *web* dinamis”.

Dari pendapat di atas dapat disimpulkan bahwa *PHP* adalah penerjemahan baris kode yang bisa dibaca atau dimengerti oleh komputer karena *PHP* bisa diletakkan pada *script HTML* atau sebaliknya. *PHP* dikhususkan untuk pengembangan *web* dinamis.



### **2.1.8 Framework**

*Framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks. Istilah ini sering digunakan antara lain dalam bidang perangkat lunak untuk menggambarkan suatu desain sistem perangkat lunak yang dapat digunakan kembali gabungan antara kelas dan fungsi yang telah tersusun secara sistematis yang dapat membantu mempermudah *programmer* dalam menuliskan kode *program* secara terstruktur, sehingga dalam membuat suatu program aplikasi harus mengikuti aturan dari *framework* (Anggaraini, 2020).

### **2.1.9 CodeIgniter (CI)**

Menurut Purbadian (2016:18) “*Codeigniter* adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman *PHP* yang bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*”.

Menurut Supono dan Putratama (2016:109) “*Codeigniter* adalah aplikasi *open source* yang berupa *framework* dengan model basis *Model View Controller (MVC)* untuk membangun *website* dinamis dengan menggunakan *PHP*”.

Berdasarkan pengertian di atas dapat ditarik kesimpulan bahwa *Codeigniter* adalah aplikasi *open source* yang merupakan *framework PHP* dengan eksekusi tercepat dan menggunakan model basic *Model View Controller (MVC)* untuk membangun sebuah *website* yang dinamis dan bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*.

### 2.1.10 Database MySQL

Menurut Kurniawan (2010:16) *MySQL* merupakan software *database server* yang sangat umum digunakan dan sangat terkenal. *MySQL* termasuk jenis *Relational Database Manajement System (RDBMS)*. *MySQL* mendukung bahasa pemrograman *PHP*, bahasa permintaan yang tertata rapi, karena pada penggunaannya *SQL* memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama *ANSI*. *MySQL* merupakan *Relational Database Manajement System (RDBMS) server*. *RDBMS* adalah sebuah *program* yang memungkinkan pengguna informasi untuk membuat, membuat, dan menggunakan informasi dalam model sosial. Dengan cara ini, tabel di basis informasi memiliki koneksi antara satu tabel dengan tabel lainnya. Beberapa keunggulan dari *MySQL* yaitu :

1. Cepat, dapat diandalkan, dan mudah digunakan. *MySQL* tiga hingga beberapa kali lebih cepat daripada server basis informasi bisnis yang sekarang dapat diakses, mudah diatur dan tidak memerlukan spesialis untuk mengontrol pembentukan *MySQL*.
2. Didukung oleh berbagai bahasa. *Server Database MySQL* dapat memberikan pesan Kesalahan dalam bahasa yang berbeda seperti Belanda, Portugis, Spanyol, Inggris, Prancis, Jerman, dan Italia.
3. Dilengkapi untuk membuat tabel yang sangat besar. Ukuran terbesar dari setiap tabel yang dapat dibuat dengan *MySQL* adalah 4 GB hingga ukuran dokumen yang dapat ditangani oleh kerangka kerja yang digunakan.
4. Lebih murah, *MySQL* adalah *open source* dan tersebar tanpa biaya untuk tahapan *UNIX*, *OS/2* dan *Platform Windows*. Penggabungan intrinsik

*PHP* dengan *MySQL*. Koneksi antara *PHP* dan *MySQL*, yang keduanya merupakan *Open-Source Software*, sangat solid, sehingga koneksi lebih cepat jika dibandingkan dengan menggunakan *server* kumpulan data lainnya. Modul *MySQL* di *PHP* sudah dikonfigurasi sehingga tidak perlu pengaturan ekstra dalam dokumen *setup PHP* ini.

#### **2.1.11 *phpMyAdmin***

Menurut MADCOMS (2016: 186) “*PhpMyAdmin* adalah sebuah aplikasi *Open Source* yang berfungsi untuk memudahkan manajemen *MySQL*. Dengan menggunakan *PhpMyAdmin*, dapat membuat *database*, membuat tabel, meng-insert, menghapus dan meng-*update* data dengan *GUI* dan terasa lebih mudah, tanpa perlu mengetikkan perintah *SQL* secara manual”. Menurut Buana (2014:2), *phpMyAdmin* adalah salah satu aplikasi yang digunakan untuk memudahkan dalam melakukan pengelolaan database *MySQL*. *PhpMyAdmin* merupakan aplikasi web yang bersifat *open source*. Menurut Muhammad Sadeli (2014:10) *PhpMyadmin* adalah sebuah *software* yang berbentuk seperti halaman situs yang terdapat pada *web server*. Fungsi dari halaman ini adalah sebagai pengendali *database MySQL*. Karena dengan adanya halaman ini semua hal tersebut dapat dilakukan hanya dengan meng-klik menu fungsi yang ada pada halaman *PhpMyadmin*.

#### **2.1.12 *Sublime Text***

Menurut Supono (2016:14) “*Sublime Text* merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau meng-edit aplikasi. *Sublime Text* memiliki modul tambahan yang membuatnya lebih mudah untuk programmer perangkat lunak. Selain itu, *Sublime Text* tampaknya kaya untuk *Syntax editor*. Selain ringan, *Sublime Text* memiliki kecepatan dalam menyimpan dan membuka

file. Tidak mengherankan *Sublime Text* adalah yang paling banyak digunakan, terutama di kalangan programmer berbasis web”.

### **2.1.13 *Extreme Programming (XP)***

Pengembangan *Extreme Programming* sering juga dikenal dengan metode *XP*. Metode yang dicetuskan oleh Kent Beck, seorang pakar *software engineering*. *Extreme Programming* adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien, adaptif dan fleksibel. (Ahmad, 2016).

#### **1. *Planning***

Tahapan perencanaan dimana untuk memahami konsep, mengumpulkan kebutuhan sistem, menggambarkan *output* yang akan digunakan, fitur-fitur dan fungsionalitas yang akan dibangun.

#### **2. *Design***

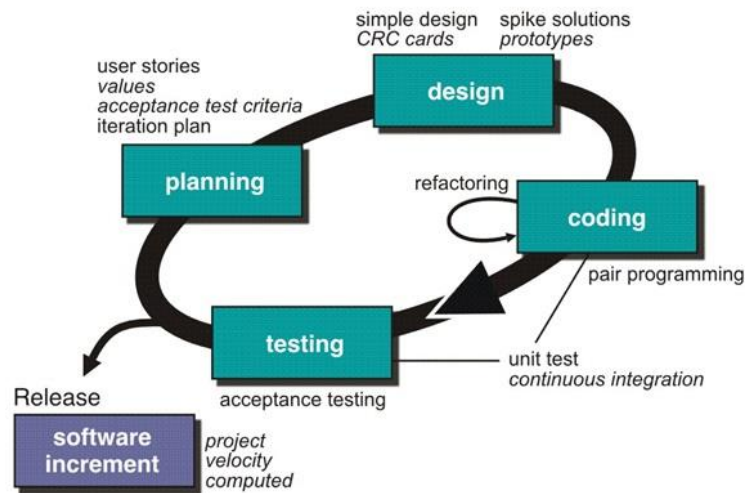
Merupakan tahapan data yang didapatkan dari tahapan *planning* sehingga membutuhkan *UML* yang mempunyai manfaat dalam pemodelan sistem yang akan dibuat.

#### **3. *Coding***

Tahapan pengkodean perangkat lunak dengan panduan alur sistem yang akan dirancang.

#### **4. *Testing***

Tahapan terakhir pada *Extreme Programming* yang akan digunakan dengan cara pengujian untuk mendapatkan *feedback* dari pengujian fungsional perangkat lunak.



Gambar 2. 2 Prosedur *Extreme Programming* (Ahmad, 2016).

#### 2.1.14 *Black Box Testing*

Pengujian sistem bertujuan untuk melihat apakah sistem yang telah dibuat sudah sesuai dengan tujuan awal pembuatan dan layak untuk dipergunakan. Pengujian pada sistem menggunakan metode *Black Box*, tujuannya untuk mengetahui bahwa bagian-bagian dalam sistem aplikasi telah benar menampilkan pesan-pesan kesalahan jika terjadi kesalahan dalam penginputan data (Sandy 2015). *Black Box Testing* sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *black box* ini menitik beratkan pada fungsi sistem (Rizki 2015).

#### 2.1.15 *Usability Testing*

Menurut (Supriyatna, 2019), Usability adalah atribut kualitas yang menjelaskan atau mengukur seberapa mudah penggunaan suatu antar muka (*interface*). Kata “*usability*” juga merujuk pada suatu metode untuk meningkatkan kemudahan pemakaian selama proses desain.




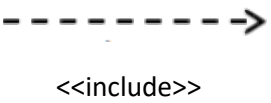
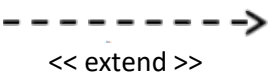
## **2.2 Unified Modeling Language (UML)**

Menurut (Pakaya, Tapate and Suleman, 2018) *Unified Modeling Language (UML)* merupakan kesatuan struktur dan caras bagi pemodelan desain program berorientasi objek (OOP) serta aplikasinya. *UML* adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. *UML* mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang *UML* sudah mulai banyak digunakan oleh para praktisi OOP. *UML* merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. *UML* adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. *UML* dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian *UML* dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan *UML* dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

### **2.2.1 Use Case Diagram**

*Use Case Diagram Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) 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 (Hanifah and Alit, 2016). Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:



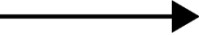
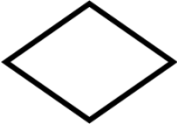
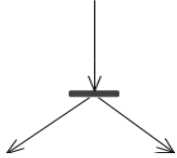
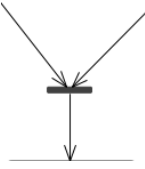

Tabel 2. 1 *Use Case*

| No | Simbol  | Keterangan   |
|----|---|--|
| 1  |    | <p><i>Actor</i> : <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi sikan aktif, 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 kontrol terhadap use case.</p> |
| 2  |  | <p><i>Use Case</i> : 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 data.</p>  |
| 3  |  | <p><i>Association</i> : Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara</p>  |
| 4  |  | <p><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</p>   |
| 5  |  | <p><i>Extend</i> : merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>   |

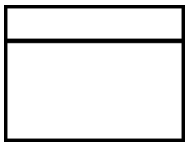
### 2.2.2 Activity Diagram

*Activity Diagram* merupakan sesuatu yang menggambarkan kemajuan kehiatan dalam program yang sedang dirancang, bagaimana interaksi aliran dimulai, pilihan yang mungkin terjadi, dan bagaimana kerangka kerja akan berakhir (Hanifah and Alit, 2016). simbol-simbol pada *Activity Diagram* dapat dilihat pada tabel 2.2.

Tabel 2. 2 *Activity Diagram*

|   | Simbol  | Keterangan  |
|---|---|---|
| 1 |    | <i>Start Point</i> : diletakkan pada pojok kiri atas dan merupakan awal aktivitas   |
| 2 |   | <i>Activity</i> : menggambar kan suatu proses/kegiatan bisnis   |
| 3 |  | <i>Join</i> : digunakan untuk menunjukkan adanya dekomposisi.   |
| 4 |  | <i>Decision</i> : menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .  |
| 6 |  | <i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu |
| 7 |  | <i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi  |
| 8 |  | <i>End Point</i> : akhir aktivitas  |



|   |   |   |
|---|---|---|
| 9 |  | <p><i>Swimlane</i> : pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.</p> |
|---|---|---|



### 2.2.3 *User Story*

*User Story* merupakan Catatan cerita didapat dari hasil wawancara dengan user tentang hal-hal apa saja yang harus dilakukan oleh sistem sesuai dengan keinginan *user* (Ningsih dkk, 2017). *User Story* akan digunakan sebagai semacam perspektif untuk membuat fitur-fitur yang harus ada dalam sistem nanti.

### 2.2.4 *Class Responsibility Collaborator (CRC) Card*

*Class Responsibility Collaborator Card* atau *CRC Card* berguna untuk menggambarkan *class* apa yang akan dibuat di samping kegunaan yang diperlukan. Berikut ini adalah beberapa rencana *CRC Card* yang telah dibuat mengingat efek samping dari pemeriksaan pada tahap penyusunan.

Tabel 2. 3 *CRC Card*

|   | <b>Simbol</b>   | <b>Keterangan</b>   |
|---|---|---|
| 1 |  | <p><i>CRC (Class Responsibility Collaborator)</i> : digunakan untuk mengidentifikasi dan mengorganisasikan <i>object-oriented classes</i>. <i>CRC Card</i> hanya men-<i>design</i> sistem yang dihasilkan sebagai bagian dari proses <i>XP</i>.</p> |
| 2 |  | <p>Digunakan untuk menghubungkan <i>CRC</i> yang saling berkolaborasi.</p>  |

