

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian terdahulu tentang sistem penugasan kerja karyawan dapat dilihat pada tabel sebagai berikut:

1. (Farras et al., 2021) meneliti tentang Analisis Dan Perancangan Sistem Informasi Penilaian Kinerja Pegawai (Studi Kasus PT Multi Anugrah Abadi Jambi). PT Multi Anugrah Abadi belum memiliki teknologi sistem informasi penilaian kinerja pegawai yang tekomputerisasi untuk mempermudah penilaian kinerja dan pengolahan data pegawai. Dalam proses penilaian kinerja dilakukan menggunakan formulir penilaian kinerja berupa kertas yang wajib di isi tiap kepala departemen atas kinerja bawahannya, untuk mengetahui perkembangan penilaian atas kinerja pegawai, Departemen HRD harus merekapitulasi hasil penilaian tiap periode menggunakan office excel. Maka penulis bertujuan untuk menganalisa dan merancang sistem informasi penilaian kinerja pegawai dengan menerapkan sistem informasi penilaian kinerja berbasis web, diharapkan dapat memudahkan proses penilaian kinerja pegawai. Kerangka kerja penelitian ini terdiri dari studi literatur, pengumpulan data, analisis data, pengembangan sistem, dan pembuatan laporan. Metode pengembangan sistem penulis melakukan rancangan use case, activity diagram, class diagram, dan prototype sistem yang mengacu pada bidang ilmu rekayasa perangkat lunak (RPL). Sistem informasi penilaian kinerja pegawai ini dapat melakukan proses penilaian kinerja dan mengelolah data pegawai dengan baik agar mengefektifkan dan mengefisiensikan proses penilaian kinerja dan mempercepat waktu serta meminimalisasikan tingkat kesalahan saat memasukan data. Penulis juga mengharapkan sistem ini adanya pengembangan lagi

dari petugas yang dapat bertanggung jawab dalam mengoperasikan sistem untuk mencegah kecurangan.

2. (Kapoh & Gunawan, 2021) meneliti tentang Perancangan Sistem Informasi Penilaian Kinerja Karyawan Berbasis Web Pada Pt Dwi Wahana Ekualindo Jakarta. Hasil penelitian ini adalah aplikasi penilaian kinerja karyawan, memudahkan karyawan untuk mengetahui hasil penilaian kinerja tanpa harus menanyakan ke bagian HRD. Dengan adanya aplikasi penilaian kinerja karyawan, maka memudahkan bagian untuk mengelola data nilai kinerja karyawan pada periode tertentu. Dengan aplikasi ini pimpinan lebih mudah mengetahui laporan hasil penilaian kinerja karyawan.
3. (Maesaroh et al., 2021) meneliti tentang Sistem Informasi Penilaian Kinerja Karyawan Berbasis Web Pada PT. Hydro Raya Adhi Perkasa. PT. Hydroraya Adhi Perkasa Tangerang adalah perusahaan yang bergerak dibidang konstruksi yang mengerjakan beberapa project besar seperti pengolahan air limbah, minyak dan gas, listrik, air serta sektor industri lainnya secara umum. Sistem saat ini dirasa masih kurang efektif dalam melakukan penilaian kinerja terhadap karyawan. Untuk dapat melakukan penilaian kinerja dengan baik maka diperlukan sistem penilaian kinerja berbasis web untuk mempermudah dalam penilaian kinerja karyawan serta membantu pengolahan data menjadi informasi yang dihasilkan. Pada pembuatan sistem ini menggunakan metode analisa SWOT untuk mengidentifikasi kelemahan sistem menjadi rekomendasi dalam membuat perbaikan-perbaikan yang harus dibuat pada sistem yang harus dikembangkan.
4. (Yuliad & Rodianto, 2022) meneliti tentang Sistem Informasi Penilaian Kinerja Pegawai Berbasis Web (Studi Kasus: Kantor Kecamatan Empang). Kecamatan merupakan lembaga pemerintah perwakilan kabupaten yang aktivitas didukung oleh sumber daya manusia untuk menunjang kegiatan operasionalnya. Saat ini, penilaian

kinerja pegawai pada Kantor Kecamatan Empang masih menggunakan form penilaian berupa lembaran-lembaran kertas yang masih dirasa kesulitan oleh Kepala Sumber Daya Manusia (SDM) dan dikhawatirkan akan menimbulkan resiko kerusakan dan kehilangan data. Penulis membangun Sistem Informasi Penilaian Kinerja Pegawai Berbasis Web menggunakan metode waterfall dengan pemrograman PHP dan Database MySQL yang dapat mempermudah dalam melakukan penilaian kinerja pegawai. Hasil akhir dari penelitian ini adalah Sistem Informasi Penilaian Kinerja Pegawai Berbasis Web dapat mempermudah dalam melakukan penilaian kinerja pegawai lebih cepat dan otomatis.

5. (Ami Radianti et al., 2022) meneliti tentang Analisis Sistem Penilaian Kinerja Karyawan Dalam Meningkatkan Kualitas Sumber Daya Manusia Di Pt Jasa Raharja Cabang Sumatera Utara. Penelitian ini bertujuan untuk menganalisis sistem informasi penilaian kinerja karyawan dalam meningkatkan kualitas sumber daya manusia pada perusahaan jasa raharja. Penelitian ini merupakan penelitian deskriptif kualitatif merupakan metode penelitian yang berdasarkan pada filsafat postpositivisme digunakan untuk meneliti pada situasi objek yang alamiah dimana peneliti sebagai instrumen kunci teknik pengumpulan data dengan cara wawancara. Dalam pengumpulan data penelitian ini, peneliti membagi menjadi dua yaitu sumber data primer. Sumber data primer yang berasal dari hasil wawancara oleh Bapak Rival Irwanson Damanik Kasubag Sumbangan Eajib dan Humas PT Jasa Raharja Cabang Sumatera Utara dan sumber data sekunder berasal dari penelitian-penelitian sebelumnya dan web-web yang sangat berkaitan dengan topik permasalahan yang ingin peneliti teliti. Hasil penelitian ini adalah dalam Penerapan sistem penilaian kinerja pegawai pada PT Jasa Raharja (Persero) Cabang Sumatera Utara sesuai dengan teori yaitu digunakan untuk penetapan imbalan, program pendidikan dan pelatihan,

pembinaan karir. Pada PT Jasa Raharja (Persero) Cabang Sumatera Utara melakukan penilaian kinerja pegawai dengan menggunakan beberapa formulir yang telah ditetapkan sebagai acuan untuk menilai pegawai dalam melaksanakan pekerjaannya seperti formulir A sampai dengan formulir F

2.2. Pembangunan

Pembangunan adalah membuat dan menciptakan suatu aplikasi atau suatu sistem yang sebelumnya belum ada pada suatu perusahaan atau instansi tersebut (Muliadi et al., 2020).

2.3. Pengertian Sistem

Sistem adalah suatu jaringan kerja dari prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan sasaran tertentu (Pasha et al., 2020).

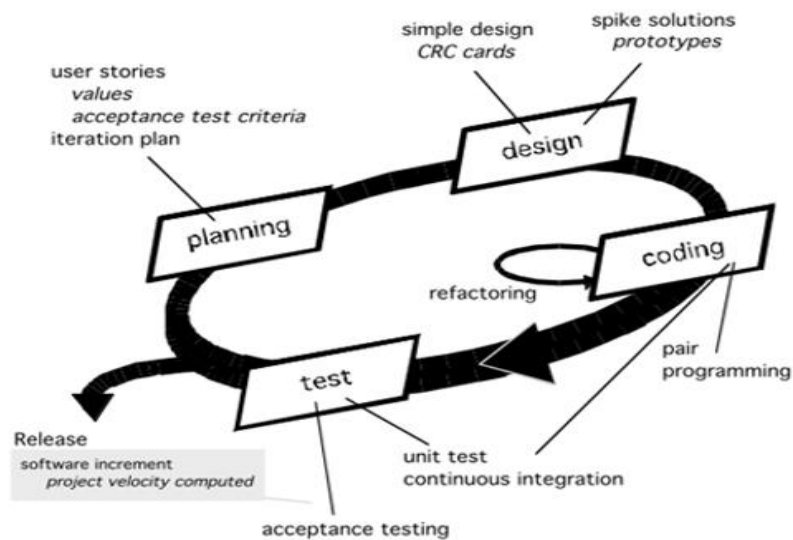
Sistem dapat didefinisikan sebagai kumpulan dari berbagai elemen-elemen saling berinteraksi untuk mencapai tujuan atau sasaran tertentu yang sama (Reza & Putra, 2021)

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

2.4. Pengembangan Sistem *Extreme Programming*

Extreme Programming (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan-perubahan *requirement* yang sangat cepat (Pressman, 2012).

Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah testing. Unsur-unsur lain dari *Extreme Programming* meliputi *paired programming* pada tahapan *coding*, unit testing pada semua kode, penghindaran pemrograman fitur kecuali benar-benar diperlukan, struktur manajemen yang datar, kode yang sederhana dan jelas, dan seringnya terjadi komunikasi antara programmer dan pelanggan ketika terjadi perubahan kebutuhan pelanggan seiring berlalunya waktu berlalu (Asrurin & Jupriyadi, 2021).



Gambar 2. 1 Ilustrasi Proses *Extreme Prgamming*
 Sumber: (Pressman, 2012)

1. *Planning* : Kegiatan *Planning* biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan-kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan *fungsi*. Pengumpulan *user stories* ini dilakukan guna memungkinkan tim developer memahami konteks bisnis aplikasi yang dikembangkan dan leluasa mengembangkan aplikasi guna memenuhi kebutuhan dan fungsi utama.
 - a. *User stories* menjadi alat komunikasi antara customer dan tim developer dapat mempertimbangkan perilsan pada tahap perencanaan.

- b. *User stories values* yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama.
 - c. *Acceptance test criteria iteration plan* melakukan perhitungan kecepatan project selama development, customer dapat menambah story, merubah value, membagi story atau menghapusnya.
2. *Design* : Design pada XP mengikuti prinsip KIS (Keep It Simple). Design yang dibuat lebih sederhana karena lebih banyak disukai dibandingkan design yang complex. Sedangkan untuk design yang rumit XP menyarankan menggunakan Spike Solution, dimana pembuatan design langsung ke tujuan.
 - a. *Simple design CRC Cards* untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment*
 - b. *Spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.
3. *Coding* : Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke pengkodean.
 - a. *Pair programming* melakukan kerja sama untuk membuat code dari satu *story*.
 - b. *Refactoring* adalah proses restrukturisasi kode program computer yang ada tanpa mengubah perilaku eksternalnya.
4. *Testing* : Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali.
 - a. *Unit test continuous integration* yaitu tahapan pengujian code yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh customer dan focus pada keseluruhan dan fungsional sistem.

- b. *Acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari *software realease*.

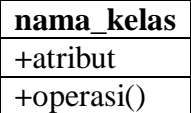



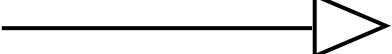
2.5. Unified Modeling Language (UML)


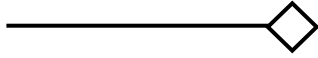
UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa A.S dan M. Shalahuddin, 2018)

2.5.1. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. simbol-simbol yang ada pada diagram kelas pada tabel class diagram 2.1:

Tabel 2. 1 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

Kebergantungan/ <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Sumber: (Rosa A.S dan M. Shalahuddin, 2018)


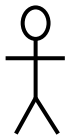

2.5.2. Use Case Diagram


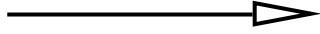

Menurut Rosa A.S dan M. Shalahuddin (2018)

“Use case diagram atau diagram use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. simbol-simbol yang ada pada diagram use case dapat dilihat pada tabel 2.2:

Tabel 2. 2 Simbol diagram use case


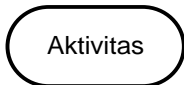


Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor
Ekstensi/ <i>extend</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan

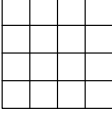



<<extend>> 	dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>Include/uses</i> <<include>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

2.5.3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3:

Tabel 2. 3 Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

<p>Tabel</p> 	<p>Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis</p>
<p>Dokumen</p> 	<p>Menunjukkan dokumen sumber atau laporan</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p>Swimlane</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

2.6. Pengertian SQL

MySQL merupakan *Database* yang menghubungkan script php menggunakan perintah *query* dan *escaps character* yang sama dengan php. MySQL mempunyai tampilan *client* yang mempermudah anda dalam mengakses *database* dengan kata sandi untuk mengizinkan proses yang bisa anda lakukan (Cahyono & Jayanti, 2022).

2.7.PHP (*Personal Home Page*)

PHP (*Personal Home Page*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada HTML. PHP merupakan singkatan dari “PHP: *Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen HTML sekaligus bekerja di sisi *server* (*server-side HTML-embedded scripting*). Artinya sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada halaman HTML biasa, sehingga script-nya tak tampak di sisi *client* (Puteri et al., 2022)

2.8.XAMPP

Menurut Jonny (2020) XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL.

Dalam XAMPP banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman web XAMPP berfungsi sebagai server offline yang berdiri sendiri (seringkali disebut sebagai localhost). XAMPP terdiri dari beberapa program yaitu Apache HTTP Server, MYSQL Database, PHP, dan Pearl XAMPP juga dilengkapi fitur manajemen database PHPMyAdmin seperti pada server hosting sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis database secara mudah.

Berikut ini Penjelasan dari bagian-bagian XAMPP:

1. X , Kenapa disebut dengan system operasi? karena XAMPP bisa dijalankan di 4 OS besar yang sering digunakan oleh pengguna komputer saat ini. Dan 4 OS tersebut tidak lain dan tidak bukan adalah Windows, Linux, Mac OS dan Solaris.
2. A (*Apache*) merupakan aplikasi web server. Apache ini bersifat opensource yang berarti gratis dan bisa diedit oleh penggunanya. Tugas utama Apache adalah menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. jika diperlukan juga berdasarkan kode PHP yang dituliskan, maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam MySQL) untuk mendukung halaman web yang dihasilkan.
3. M (MySQL), merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam database.

4. P (PHP), bahasa pemrograman web. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat server-side scripting. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah MySQL. namun PHP juga mendukung sistem manajemen *database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL*, dan sebagainya.
5. P (Perl), bahasa pemrograman, pertama kali dikembangkan oleh Larry Wall di mesin Unix. Perl pertama kali dirilis pada tanggal 18 Desember 1987 ditandai dengan keluarnya Perl 1. Dua diantara karakteristik utama perl adalah penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum. Perl sangat populer di gunakan dalam program-program CGI (*Common Gateway Interface*) dan protokol internet lainnya.

2.9.Sublime Text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP,

Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum didukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user.

2.10. Pengujian ISO 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang software engineering (Iqbal, 2016). Product quality ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

1. Kualitas dalam model penggunaan
2. Model kualitas produk
3. Data model kualitas

Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

- 1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk Merupakan tingkatan dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi spesifik tertentu dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna.
- 2) *Reliability* Merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik dalam hal ini perangkat lunak dapat beroperasi dan siap ketika dibutuhkan untuk digunakan dan juga dapat bertahan pada tingkat kemampuan tertentu terhadap kegagalan, kesalahan serta perangkat lunak kembali pada tingkat tertentu dalam mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak.

- 3) *Performance efficiency* Merupakan tingkatan dimana perangkat lunak dapat memberikan kinerja terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu dalam hal ini *performance efficiency* dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi dan perangkat lunak dapat menggunakan sejumlah sumber daya ketika melakukan aksi dari sebuah fungsi.
- 4) *Usability* Perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik pengguna ketika digunakan dalam hal ini perangkat lunak mudah dipelajari oleh pengguna, perangkat lunak dapat digunakan dan dioperasikan oleh pengguna.
- 5) *Security* Merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman atau keganjalan dalam hal ini perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna dan merupakan dari kelengkapan, ketepatan dari sejumlah *asset* yang telah dijaga sehingga aksi atau tindakan yang dilakukan telah terbukti dan hal tersebut tidak dapat ditolak.
- 6) *Compability* Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.
- 7) *Maintainability* Merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan, kebutuhan dan fungsionalitas yang spesifik. Selain itu perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak untuk mengidentifikasi bagian yang dapat dimodifikasi.
- 8) *Transferability*. Merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan satu ke lingkungan yang lain dalam hal ini perangkat lunak

dapat beradaptasi dengan cepat pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.

2.11. Skala Pengukuran

Skala pengukuran yang digunakan adalah skala Likert, skala yang didasarkan pada penjumlahan sikap responden dalam merespon pernyataan berkaitan indikator-indikator suatu konsep atau variable yang sedang diukur (Sugiyono, 2017). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). Skala Likert dapat dilihat pada Tabel 2.4.

Tabel 2. 4 Skala Likert

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat dibawah ini:

$$\text{Persentase} = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\%$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
≥ 90%-100%	Sangat Baik
≥ 70% - <90%	Baik
≥ 50% - <80%	Cukup Baik

$\geq 60\%$ - $<70\%$ 0% - 60%	Kurang Baik Tidak Baik
-----------------------------------	---------------------------