

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Beberapa penelitian yang berkaitan dengan peneltiian dalam menentukan kelayakan berdasarkan jurnal penelitian dapat dilihat apda Tabel 2.1:

Tabel 2. 1 Tinjauan Pusataka

No	Nama (Tahun)	Judul	Hasil
1	Nasution (2021)	Implementasi Algoritma Alpha Skip Search Pada Perancangan Aplikasi Kitab Al-Hikam Berbasis Android.	Hasil penelitian ini adalah aplikasi Al-Hikam berbasis android dengan menerapkan algoritma alpha skip search dinilai sangat efektif untuk mempermudah dalam mempelajari dan memahami isi kitab Al-Hikam karena dapat dioperasikan dimana dan kapanpun secara online maupun offline.
2	Yulianti, Lestanti and Kholila (2021)	Perancangan Web Portal Pembelajaran Hadits Arba'in Nawawi	Hasil penelitian ini adalah aplikasi pembelajaran hadist Arba'in Nawawi dengan menggunakan algoritma Knuth-Morris-Pratt. Web portal ini dibuat untuk membantu para santri mempelajari hadist yang tidak dapat dilalukan pembelajaran secara langsung dalam masa pandemic Covid-19.
3	Novianti, Hasibuan and Suginam (2018)	Implementasi Algoritma Zhu Takaoka Pada Aplikasi Kitab Jurrumiyah Berbasis Android.	Hasil penelitian ini adalah aplikasi Kitab Al-Ajurrumiyah berbasis Android menggunakan Software Eclipse Juno. Pada penelitian ini penulis menerapkan algoritma zhu takaoka guna

No	Nama (Tahun)	Judul	Hasil
			mempemudah proses pencarian terjemahan isi ayat kitab Al-Ajurrumiyah pada Aplikasi Kitab Al-Ajurrumiyah berbasis android yang dirancang.
4	Santoso, Sundawa and Azhari (2016)	Implementasi Algoritma Brute Force Sebagai Mesin Pencari (Search Engine) Berbasis Web Pada Database	Hasil <i>Search Engine</i> ini akan menampilkan sebuah hasil pencarian berdasarkan sebuah kata atau beberapa kata yang dicari dan menentukan urutan-urutan teratas berdasarkan kata yang dicari terbanyak
5	Wahyuningsih and Nurdiana (2016)	Aplikasi Pencarian dan Pembelajaran Hadist Pada Kitab Riyadhus Shalihin Berbasis Mobile	Aplikasi ini dibuat untuk memberikan kemudahan kepada pengguna karena dilengkapi dengan pencarian hadits dan mendengarkan hadits.

1. Nasution (2021) mengeliti tentang Implementasi Algoritma Alpha Skip Search Pada Perancangan Aplikasi Kitab Al-Hikam Berbasis Android. Kitab Al-Hikam merupakan buah karya Syekh Ibnu Atha'illah, mursyid ketiga dari Thariqah Syadziliyah. Kitab Al-Hikam diperuntukkan bagi para pejalan (salik), yang di dalamnya berisi panduan lanjut bagi setiap pejalan untuk menempuh perjalanan spiritual. Kitab Al-Hikam membahas tentang tauhid dan akhlak yang mengarah kepada Tasawuf Islam. Kitab Al-Hikam berisi dengan berbagai terminologi suluk yang ketat, yang merujuk pada berbagai istilah dalam Al- Qur'an dan memiliki jumlah yang sangat banyak. Masalah yang sering terjadi pada penelitian ini yaitu kesulitan dalam menemukan hasil pencarian isi kitab Kitab Al-Hikam yang merujuk pada berbagai istilah dalam

Al-Qur'an dan memiliki jumlah yang sangat banyak bagi umat islam. sehingga membuat banyak umat muslim yang sulit untuk mendapatkan hasil pencarian bagian tertentu pada isi kitab dalam waktu yang cepat dan tepat untuk dipahami Algoritma Alpha Skip Search merupakan salah satu algoritma pencocokan string matching yang merupakan pengembangan dari algoritma skip search. Aplikasi Al-Hikam berbasis android dengan menerapkan algoritma alpha skip search dinilai sangat efektif untuk mempermudah dalam mempelajari dan memahami isi kitab Al-Hikam karena dapat dioperasikan dimana dan kapanpun secara online maupun offline. Mengatasi permasalahan tersebut penulis merancang aplikasi Kitab Al-Hikam berbasis android menggunakan editor eclipse juno dan database SQLite. Untuk meminimalisir waktu yang dibutuhkan dalam pencarian isi kitab Al-Hikam yang dirancang maka penulis menerapkan algoritma Alpha Skip Search guna mempermudah umat muslim dalam mencari isi kitab Al-Hikam untuk dipelajari dan dipahami.

2. Yulianti, Lestanti and Kholila (2021) meneliti tentang Perancangan Web Portal Pembelajaran Hadits Arba'in Nawawi Sebagai Media Penunjang Pembelajaran Santri. Pembelajaran yang kurang baik dan efektif merupakan salah satu masalah yang dihadapi pada masa pandemi ini, terutama pembelajaran para santri, karena dikhawatirkan salah pemahaman dalam mempelajarinya. Berdasarkan hal ini maka perlu dibuat sebuah web portal pembelajaran Hadits Arba'in Nawawi yang dapat mempermudah santri dalam mempelajari hadits pada masa Covid-19. Penelitian ini merancang aplikasi pembelajaran hadist berbasis web yang dapat membantu para santri dalam

mempelajari hadis meskipun tidak melakukan pembelajaran secara langsung. Pada aplikasi ini dilengkapi dengan fitur-fitur seperti audio hadist, pencarian hadist, dan juga latihan soal. Pada fitur pencarian hadist menerapkan algoritma Knuth-Morris-Pratt yang dapat menemukan kata-kata dalam hadist dengan mudah. Hasil dari penelitian ini ialah aplikasi pembelajaran hadist Arba'in Nawawi berbasis web yang dapat menjadi penunjang pembelajaran santri pada masa Covid-19 ini. Uji kelayakan aplikasi pada penelitian ini menggunakan pengujian Blackbox dan menghasilkan presentase kesesuaian sebesar 100% dan hasil pengujian terhadap pengguna dan pakar materi mendapatkan hasil presentase setuju sebesar 45% dan sangat setuju sebesar 55% yang artinya dengan adanya aplikasi pembelajaran ini dapat membantu menunjang pembelajaran santri.

3. Novianti, Hasibuan and Suginam (2018) meneliti tentang Implementasi Algoritma Zhu Takaoka Pada Aplikasi Kitab Jurrumiyah Berbasis Android. Kitab Al-Ajurrumiyah merupakan sebuah kitab yang berisikan tentang tata bahasa Arab dari mulai dari abad ke- 7 hingga abad ke-13 M yang disusun oleh ahli bahasa dari Maroko yang bernama Abu Abdillah Sidi Muhammad bin Daud Ash-Shanhaji alias Ibnu Ajurrum dengan tujuan untuk mempermudah santri yang ada di pesantren maupun masyarakat umum mempelajari kaidah-kaidah nahwu. Kitab Al-Ajurrumiyah memiliki isi yang sangat banyak dan menggunakan tipe penulisan bahasa arab sehingga untuk mempelajari isi kitab Al-Ajurrumiyah bukan hal yang mudah bagi santri maupun masyarakat umum terutama yang masih memanfaatkan kitab Al-Ajurrumiyah berbentuk buku cetak. Masalah yang sering terjadi dengan

pemanfaatan kitab Al-Ajurrumiyah berbentuk buku cetak yaitu pada saat melakukan proses pencarian isi kitab Al-Ajurrumiyah dikarenakan harus melakukan pencarian isi kitab secara manual dengan membuka satu persatu setiap lembaran kitab Al-Ajurrumiyah sehingga membutuhkan waktu yang banyak untuk menemukan isi kitab yang akan dipelajari dan membutuhkan tempat penyimpanan yang besar serta sulit untuk dibawa jika berpergian sehingga diperlukan pengembangan sistem dengan merancang aplikasi kitab Al-Ajurrumiyah berbasis android yang memiliki fitur pencarian yang baik guna meminimalisir waktu dibutuhkan untuk menemukan hasil pencarian isi kitab Al-Ajurrumiyah yang akan dipelajari. Berdasarkan uraian masalah tersebut penulis merancang aplikasi Kitab Al-Ajurrumiyah berbasis Android menggunakan Software Eclipse Juno. Pada penelitian ini penulis menerapkan algoritma zhu takaoka guna mempermudah proses pencarian terjemahan isi ayat kitab Al-Ajurrumiyah pada Aplikasi Kitab Al-Ajurrumiyah berbasis android yang dirancang. Aplikasi Kitab Al-Ajurrumiyah berbasis android yang dirancang pada penelitian dapat dioperasikan pada smartphone yang didukung oleh sistem operasi berbasis android sehingga penggunaan aplikasi Kitab Al-Ajurrumiyah berbasis android ini sangatlah mudah karena tidak memiliki ukuran perangkat yang terlalu besar sehingga dapat digunakan dimana dan kapanpun.

4. Santoso, Sundawa and Azhari (2016) meneliti tentang Implementasi Algoritma *Brute Force* Sebagai Mesin Pencari (*Search Engine*) Berbasis Web Pada Database. Tidak hanya itu, informasi yang diperlukan haruslah cepat, mudah, baik dan akurat. Dari berbagai macam teknologi untuk memperoleh

informasi, salah satunya adalah search engine. Peran serta teknologi search engine dalam teknologi informasi semakin lama semakin penting dan dibutuhkan oleh berbagai kalangan. Terutama untuk memperoleh informasi yang akurat dan cepat. Basis pengetahuan dalam search engine ini dapat diperoleh dari berbagai sumber diantaranya penelitian dan buku yang berhubungan dengan search engine tersebut. Search Engine ini akan menampilkan sebuah hasil pencarian berdasarkan sebuah kata atau beberapa kata yang dicari dan menentukan urutan-urutan teratas berdasarkan kata yang dicari terbanyak. Sebagai contoh, sering kita atau terkadang kita ingin mengetahui informasi yang dimiliki pada suatu obat dan ingin mengetahui kandungan zat-zat yang terdapat disuatu obat tertentu, karena mungkin informasi tersebut tidak terlalu up-to-date dan membutuhkan waktu yang agak lama. Maka lewat search engine kita dapat memperoleh informasi dan kandungan zat-zat yang terdapat didalam obat tersebut dengan up-to-date dan membantu kita dalam menghemat waktu. Dengan dikembangkannya aplikasi search engine ini, diharapkan dapat membantu dalam berbagai hal.

5. Wahyuningsih and Nurdiana (2016) meneliti tentang Aplikasi Pencarian dan Pembelajaran Hadist Pada Kitab Riyadhus Shalihin Berbasis Mobile. Pemasalahan yang ada pada saat bagi masyarakat luas kurang memahami isi dan makna dari hadits, alasannya karena dalam mencari informasi dan isi dari makna hadits kebanyakan masyarakat umumnya lebih banyak mengakses dengan media kertas. Sistem pencarian dari media kertas lebih banyak memakan waktu cukup lama dan untuk suatu permasalahan harus membuka secara perlembar. Dengan didukungnya oleh para pengembang sistem operasi

pada mobile khususnya android. Masalah bagi masyarakat umum tidak perlu repot membuka buku sebagai panduan pengetahuan pembelajaran tentang hadits. Cukup dengan membuka telepon genggam kita bisa meningkatkan pengetahuan dan pembelajaran isi dari makna hadits. Aplikasi ini bertujuan untuk mempermudah setiap orang yang ingin belajar isi dari makna hadits, yang tidak perlu membawa buku hanya cukup dengan menginstal aplikasi di handphone yang sistem operasinya berbasis Android.

2.2. Hadis

Secara etimologi, hadis adalah kata benda (isim) dari kata al-Taahdis yang berarti pembicaraan. Sedangkan pengertian hadis secara luas sebagaimana yang diberikan oleh sebagian ulama seperti Ath Thiby berpendapat bahwa hadits itu tidak hanya meliputi sabda Nabi, perbuatan dan taqir beliau (hadis marfu'), juga meliputi sabda, perbuatan dan taqir para sahabat (hadis mauquf), serta dari tabi'in (hadis maqthu') (Rofiah 2018).

Menurut Terjemahan Riyadus Shalihin. Secara stuktur hadits terdiri atas dua komponen yaitu sanad atau isnad (rantai rowi) dan matan (redaksi).

1. Sanad atau Isnad
2. Sanad ialah rantai Rawi atau perawi (periwayat) hadits. Sanad terdiri atas seluruh rawi mulai dari orang yang mencatat hadist tersebut dalam kitab hadist sehingga mencapai Rasulullah. Sanad, memberikan gambaran keaslian suatu riwayat.
3. Klasifikasi Hadits

Hadit dapat diklasifikasikan berdasarkan beberapa kriteria yakni bermulanya akhir sanad, keutuhan rantai sanad, jumlah rawi (perriwayat) serta tingkat keaslian hadits (dapat diterima tidaknya hadits bersangkutan).

2.3. Tafsir

Kata tafsir dalam bahasa Indonesia berasal dari bahasa Arab yaitu tafsir. Kata tafsir sendiri berasal dari akar kata fassara. Ada beberapa pendapat ahli bahasa dan ulama² tafsir tentang makna tafsir secara etimologi dan terminologi. Kata fassara juga berarti nadlaraan-Thayibullaal-Mai (penglihatan atau penelitian seorang dokter terhadap air) makna yang sama juga digunakan untuk kata al-Tafsirah. Ada pendapat yang mengatakan bahwa al-Tafsirah berarti: (buang air orang sakit yang digunakan para dokter untuk mendiagnosa penyakit seseorang)

Tafsir secara bahasa adalah sesuatu yang menjelaskan, menerangkan. Cara menerangkannya bisa dengan berbagai versi. Karena lafazh taf'îl menunjukkan makna kafsir atau menunjukkan makna banyak. Di antara adalah yang memiliki arti menyatakan (al-Ibanah), menjelaskan (al-Idharu), dan membuka (al-Kasyfu) (Al-Qaththan, 1973) (Hasanudin and Zulaiha 2022).

2.4. Kaidah

Kaidah adalah patokan atau ukuran sebagai pedoman bagi manusia dalam bertindak. Kaidah juga dapat dikatakan sebagai aturan yang mengatur perilaku manusia dan perilaku sebagai kehidupan bermasyarakat. Secara umum kaidah dibedakan atas dua hal yaitu kaidah etika atau kaidah hukum. Kaidah etika merupakan kaidah yang meliputi norma susila, norma agama dan norma kesopanan. Pada dasarnya kaidah etika datang dari diri dalam manusia itu sendiri contohnya menghormati orangnya yang lebih tua, berbuat baik pada orang tua,

saling menghargai, atau malu jika berbuat salah. Namun tidak jarang kaidah etika merupakan kaid dari diri manusia misalnya dari ajaran agama contohnya tidak boleh berperilaku jahat pada orang lain. Kaidah hukum merupakan kaidah yang memiliki sanksi tegas. Kaidah hukum ialah kaidah yang mengatur hubungan atau intraksi antar pribadi, baik secara langsung atau tidak langsung oleh karena itu kaidah hukum ditujukan untuk kedamaian, ketenteraman, dan ketertiban hidup bersama. Kaidah hukum biasanya ada paksaan yang berwujud ancaman bagi para pelanggarnya (Rofiah 2018).

2.5. *Vector Space Model*

Vector Space Model (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan similarity term dengan cara pembobotan term. Dokumen dipandang sebagai sebuah vektor yang memiliki magnitude (jarak) dan direction (arah). Pada *Vector Space Model*, sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang vektor. Relevansi sebuah dokumen ke sebuah query didasarkan pada similaritas diantara vektor dokumen dan *vektor query* *ector space* model digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu query (Leman and Andesa 2015).

Konsep dasar dari *Vector Space Model* adalah menghitung jarak antar dokumen kemudian mengurutkan berdasarkan tingkat kedekatannya. Cara kerja *Vector Space Model* dimulai dengan *case folding*, *cleaning data*, *indexing*, *filtering*, *stemming*, dan *tokenisasi* yaitu tahap pemotongan string input berdasarkan tiap kata yang meyusunnya dan memecah dokumen ke dalam tabel frekuensi kata. Seluruh kata dalam dokumen dibentuk menjadi satu yang disebut sebagai term. Tiap dokumen ditampilkan sebagai vektor yang akan dibandingkan

dengan term yang telah dibentuk. Similarity Analysis untuk mengukur kemiripan dokumen dilakukan dengan menghitung cosinus jarak antara dokumen tersebut.

Proses perhitungan *Vector Space Model* melalui tahapan perhitungan term *frequency* (tf) menggunakan persamaan (1)

$$tf = tf_{ij}$$

Dengan tf adalah term *frequency*, dan tf_{ij} adalah banyaknya kemunculan term t_i dalam dokumen d_j , Term *frequency* (tf) dihitung dengan menghitung banyaknya kemunculan term t_i dalam dokumen d_j

Perhitungan *Inverse Document Frequency* (idf), menggunakan persamaan (2)

$$idf_i = \log N/df_i$$

Dengan idf_i adalah inverse document frequency, N adalah jumlah dokumen yang terambil oleh sistem, dan df_i adalah banyaknya dokumen dalam koleksi dimana term t_i muncul di dalamnya, maka Perhitungan idf_i digunakan untuk mengetahui banyaknya term yang dicari (df_i) yang muncul dalam dokumen lain yang ada pada database.

Perhitungan term *frequency Inverse Document Frequency* (tfidf), menggunakan persamaan (3)

$$W_{ij} = tf_{ij} \cdot \log N/df_i$$

Dengan W_{ij} adalah bobot dokumen, N adalah Jumlah dokumen yang terambil oleh sistem, tf_{ij} adalah banyaknya kemunculan term t_i pada dokumen d_j , dan df_i adalah banyaknya dokumen dalam koleksi dimana term t_i muncul di dalamnya. Bobot dokumen (W_{ij}) dihitung untuk didapatkannya suatu bobot hasil perkalian

atau kombinasi antara term frequency (tf_{ij}) dan Inverse Document Frequency (idf).

Perhitungan Jarak persamaan (4)

$$|q| = \sqrt{\sum_{i=1}^t (W_{IA})^2}$$

Dengan |q| adalah Jarak query, dan W_{iq} adalah bobot query dokumen ke-i, maka Jarak query (|q|) dihitung untuk didapatkan jarak query dari bobot query dokumen (W_{iq}) yang terambil oleh sistem. Jarak query bisa dihitung dengan persamaan akar jumlah kuadrat dari query.

Perhitungan Jarak Dokumen, menggunakan persamaan (5)

$$|d_j| = \sqrt{\sum_{i=1}^t (W_{ij})^2}$$

Dengan |d_j| adalah jarak dokumen, dan W_{ij} adalah bobot dokumen ke-i, maka Jarak dokumen (|d_j|) dihitung untuk didapatkan jarak dokumen dari bobot dokumen dokumen (W_{ij}) yang terambil oleh sistem. Jarak dokumen bisa dihitung dengan persamaan akar jumlah kuadrat dari dokumen. Menghitung index terms dari dokumen dan query (q,d_j). menggunakan persamaan (6)

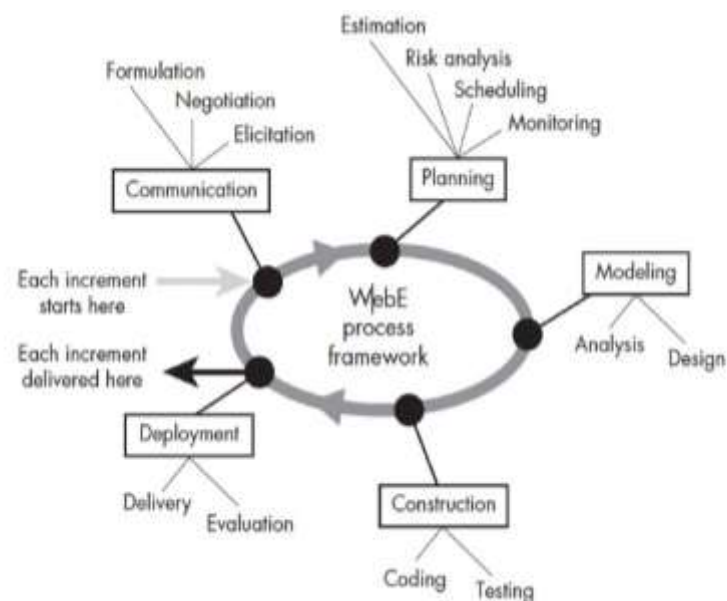
$$q, d_j = \sum_{i=1}^t (W_{iA}) (W_{ij})$$

2.6. Metode Pengembangan Sistem

Tahapan Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *web engineering* (rekayasa *web*) yaitu suatu model rekayasa perangkat lunak yang digunakan untuk perkembangan aplikasi-aplikasi berbasis *web*,

sedangkan menurut Pressman (2012) *web engineering* adalah suatu proses yang digunakan untuk membuat *web* yang berkualitas tinggi.

Web engineering tidak sama persis dengan RPL (Rekayasa Perangkat Lunak), tetapi rekayasa *web* memiliki konsep dan prinsip mendasar dari RPL. Proses rekayasa *web* lebih ditekankan pada aktivitas teknis dan manajemen yang hampir sama (Pressman 2015). Tahapan rekayasa *web* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tahapan Metode Rekayasa Web

Tahapan-tahapan dalam rekayasa *web* antara lain:

1. *Customer communication*

Komunikasi yang baik dengan *user* merupakan sarana efektif dalam membuat atau menerjemahkan apa saja yang *user* inginkan (*requirements*).

- a. *Formulation*: Melakukan perumusan masalah terhadap sistem yang akan dibangun.
- b. *Negotiation*: Melakukan perjanjian untuk melakukan pengembangan

sistem sesuai kebutuhan.

- c. *Elicitation*: Melakukan observasi terhadap sistem yang akan dibangun.

2. *Planning*

Yaitu tahap penggabungan *requirement* (kebutuhan) dan informasi dari *user* bisa berupa wawancara dan perencanaan teknis. Perencanaan teknis dilakukan dengan mengidentifikasi perangkat lunak maupun perangkat keras apa saja yang dibutuhkan.

- a. *Estimation*: Melakukan perkiraan biaya sistem yang akan dibangun.
- b. *Risk Analysis*: Melakukan analisis resiko yang akan dihadapi oleh pengembang sistem.
- c. *Scheduling*: Melakukan penjadwalan sistem yang akan diterapkan.
- d. *Monitoring*: Melakukan pemantauan terhadap sistem yang akan dikembangkan.

3. *Modeling* (Pemodelan)

- a. *Analysis modeling*, merupakan tahap berikutnya dari *planning* dan komunikasi dengan *user*. Yaitu merumuskan kebutuhan (*requirements*) dari *user* serta permasalahan apa yang akan diselesaikan, mengidentifikasi interaksi antara *user* dengan sistem berdasarkan hak akses pengguna, juga analisis fungsional yaitu mengidentifikasi operasi-operasi apa saja yang akan dijalankan pada sistem.
- b. *Design modeling*, merupakan desain antarmuka (*Interface*), merancang tampilan halaman dengan kombinasi warna, teks, dan gambar yang

sesuai dengan isi dan tujuan aplikasi *web*, Desain navigasi sesuai dengan alur kerja sistem, Desain tersebut dirancang berdasarkan kebutuhan informasi yang telah diidentifikasi pada tahap analisis.

4. *Construction* (kontruksi)

a. Implementasi (*coding*)

Implementasi dilakukan dengan mengaplikasikan halaman *web* dalam bentuk HTML berdasarkan hasil perancangan isi pada aktivitas pada *non technical member* sedangkan implementasi isi dan fungsi logika dibuat dalam bentuk PHP.

b. Pengujian (*testing*)

Dilakukan untuk mengetahui kemungkinan terjadinya kesalahan seperti kesalahan pada skrip atau *form*, navigasi ataupun tampilan, maupun bagian lainnya.

5. *Delivery & Feedback*

Serah terima dan respon dilakukan dengan cara menyebarkan kuisisioner kepada *user* berupa respon untuk mendapatkan penilaian dari setiap kriteria sebagai hasil evaluasi bagi pengembang. Kriteria yang dilakukan biasanya *usability, functionality*, serta *reliability*.

a. *Delivery*: Melakukan pelayanan sesuai sistem yang telah dibangun.

b. *Evaluation*: Melakukan penilaian apakah sistem yang dibangun sesuai dengan kebutuhan sistem.

2.7. Bahasa Pemodelan Pengembangan Sistem (UML)


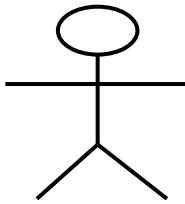
Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa A.S dan M. Shalahuddin 2018).





2.7.1. Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat (Rosa A.S dan M. Shalahuddin 2018)

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat, menjelaskan simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.2 di bawah ini:

Tabel 2. 2 Simbol Diagram *Use Case*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>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 <i>use case</i></p>
<p>Aktor/<i>actor</i></p> 	<p>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</p>

Simbol	Deskripsi
	nama <i>actor</i>
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>
Ekstensi/ <i>extend</i> << <i>extend</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan 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> << <i>include</i> >> 	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

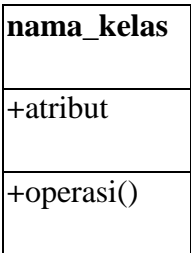



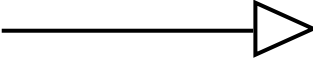


Sumber : (Rosa A.S dan M. Shalahuddin 2018)

2.7.2. 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 (Rosa A.S dan M.

Shalahuddin 2018), menjelaskan simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.3.

Tabel 2. 3 Simbol *Class Diagram*





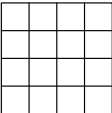


Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi 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>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

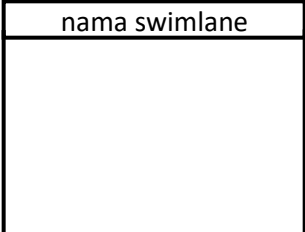
Sumber : (Rosa A.S dan M. Shalahuddin 2018)

2.7.3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Rosa A.S dan M. Shalahuddin 2018), menjelaskan Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2. 4 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
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Simbol	Deskripsi
<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

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

2.8. Alat Implementasi

2.8.1. Xampp

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data *MySQL* dikomputer local. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan internet (Wicaksono 2015).

Menurut MADCOMS (2016) Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain.

Xampp berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan web memerlukan PHP, Apache, MySQL dan PhpMyAdmin (MADCOM, 2016).

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.

2.8.2. 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 terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user (Sadeli, 2015). Hasil pembuatan web yang dibangun menggunakan bahasa PHP dan aplikasi *sublime text* dengan tampilan text editor sebagai berikut :



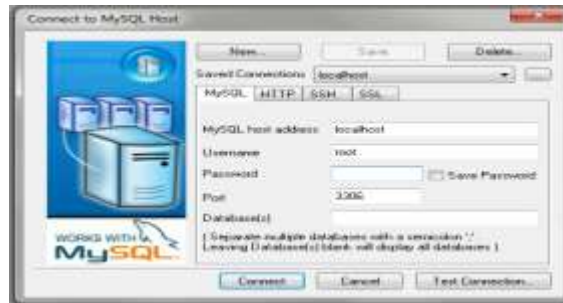
Gambar 2. 2 *Text Edior Sublime Text*

2.8.3. MySQL

Menurut MADCOM (2016) MySQL adalah sistem manajemen Database SQL yang bersifat Open Source dan paling populer saat ini. Sistem Database MySQL mendukung beberapa fitur seperti multithreaded, multiuser dan SQL Database managemen system (DBMS).

SQLyog merupakan *tool* untuk manajemen *database* MySQL yang memiliki tampilan grafis. Untuk dapat menggunakan SQLyog, *database server* MySQL harus dalam keadaan aktif. Jika *server* yang akan dihubungkan tidak aktif, maka *server* harus dijalankan terlebih dahulu. Ketika SQLyog dipanggil,

maka akan muncul jendela untuk koneksi ke MySQL server seperti gambar berikut:



Gambar 2. 3 login koneksi SQLyog

Berikut informasi yang harus diisi untuk melakukan koneksi *database server* MySQL:

1. *MySQL Host Address* diisi dengan nama *host* atau *IP address* dari *MySQL server* yang akan dihubungkan. Nilai *defaultnya* adalah *localhost*.
2. *Username* diisi dengan nama pemakai untuk melakukan koneksi ke *MySQL server*. Nama pemakai yang dimasukkan adalah nama pemakai *MySQL*, bukan nama pemakai *FTP* atau *Web Server*. *User defaultnya* adalah *root*.
3. *Password* diisi dengan *password* untuk nama pemakai *MySQL*.
4. *Port* diisi dengan nomor port *TCP/IP* untuk melakukan koneksi ke *MySQL server*. Nomor *port defaultnya* adalah 3306.

Setelah koneksi ke *database server* *MySQL* berhasil, maka akan muncul jendela utama *SQLyog* seperti gambar berikut :



Gambar 2. 4 Jendela Utama SQLyog

Jendela utama SQLyog dibagi menjadi 3 *panel*, yaitu *object browser*, *SQL window*, dan *result window*:

1. *Object browser* menampilkan secara detail semua *database*, tabel, kolom, dan *indeks* yang terdapat pada *MySQL server*.
2. *SQL Window* merupakan jendela tempat mengetikkan perintah *SQL* yang ingin dieksekusi.
3. *Result window* menampilkan hasil *query* yang dieksekusi. Jendela ini juga menampilkan informasi tentang *database*, tabel, kolom, dan *indeks* yang dipilih pada *object browser*.

2.9. ISO 25010

Model ISO-25010 merupakan bagian dari *Software product Quality Requirements and Evaluation (SQuaRE)*, yang merupakan pengembangan dari model kualitas perangkat lunak sebelumnya yaitu ISO-9126. Dalam model ISO-25010 ini digunakan untuk melihat kualitas suatu perangkat lunak yang digunakan oleh perusahaan, instansi ataupun organisasi. Metode ISO 25010 ini dapat digunakan untuk mengevaluasi kualitas sistem perangkat lunak secara spesifik berdasarkan dua dimensi umum, yaitu dimensi *product quality*, dimana prosesnya mengacu pada karakteristik intrinsik dari sebuah produk perangkat lunak, memiliki beberapa elemen antara lain meliputi *functional suitability*, *reliability*, *operability*, *performance efficiency*, *security*, *compatibility*, *maintainability* dan *transferability*. *Quality in use* dan *product quality*. Sedangkan pada *dimensi quality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user* antara lain *Usability in use*, *Flexibility in use*, dan *Safety* Adapun untuk mengetahui gambaran kualitas *system* aplikasi *M-Library* Gajah Mada

penulis melakukan analisis berdasarkan model ISO-25010 yang terdiri dari dua dimensi umum, yaitu dimensi *product quality* dan dimensi *quality in use* (Abran et al., 2008) dapat dilihat pada Gambar berikut :



Gambar 2. 5 Model kualitas produk ISO/IEC 25010
Sumber: (Kurniawan, Arifianto, and Muharom 2018)

Adapun untuk mengetahui gambaran kualitas model ISO-25010 yang terdiri dari dua dimensi umum, yaitu dimensi *product quality* dan *dimensi quality in use*. Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya:

- 1) Fungsionalitas merupakan kemampuan perangkat lunak berada pada level di mana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi tertentu, dalam hal ini perangkat lunak dapat memenuhi kelayakan suatu fungsi untuk melakukan pekerjaan tertentu bagi pengguna dan dapat memberikan hasil yang tepat serta akurasi untuk kebutuhan pengguna tingkat. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu.
 - a. Kelengkapan fungsional, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna tertentu.

- b. Ketepatan fungsional, sejauh mana produk atau sistem memberikan hasil yang benar sesuai kebutuhan.
- c. Kesesuaian fungsional, sejauh mana fungsi yang diberikan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.

Fungsionalitas mengukur berhasil atau gaga sehingga yang digunakan yaitu skala *Guttman*. Skala *Guttman* digunakan pada penelitian untuk mendapatkan jawaban yang tegas terhadap permasalahan yang dinyatakan. Dalam pengujian fungsional perangkat lunak dapat dikatakan baik jika hasil perhitungan item fungsi mendekati 1.

$$\text{Rumus : } X = \frac{I}{P}$$

Keterangan :

P : Jumlah fungsi yang dirancang

I : Jumlah fungsi yang berhasil diimplementasikan.

- 2) Kompatibilitas, sejauh mana produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen atau melakukan fungsi lain yang diperlukan secara bersamaan ketika berbagi lingkungan perangkat keras dan perangkat lunak yang sama. Ciri ini terbagi menjadi 2 ciri yaitu:
 - a. Koeksistensi, sejauh mana produk atau sistem dapat melakukan fungsi yang diperlukan secara efisien sambil berbagi sumber daya dengan produk atau sistem lain tanpa merusak produk atau sistem.
 - b. Interoperabilitas, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.

- 3) Kegunaan, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan secara efektif, efisiensi, dan kepuasan tertentu dalam konteks penggunaan. Ciri ini terbagi menjadi beberapa ciri yaitu:
- a. Ketepatan pengenalan, sejauh mana pengguna dapat mengetahui apakah sistem atau produk memenuhi kebutuhan mereka.
 - b. Learnability, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan belajar menggunakan sistem atau produk tersebut secara efisien, efektif, bebas dari risiko dan kepuasan dalam konteks tertentu.
 - c. Operabilitas, sejauh mana produk atau sistem mudah dioperasikan dan dikendalikan.
 - d. Perlindungan kesalahan pengguna, sejauh mana produk atau sistem melindungi pengguna dari membuat kesalahan.
 - e. Estetika antarmuka pengguna, sejauh mana antarmuka pengguna suatu produk atau sistem memungkinkan interaksi yang ramah pengguna dan memuaskan.
 - f. Aksesibilitas, sejauh mana suatu produk atau sistem dapat digunakan oleh semua kelompok untuk mencapai tujuan tertentu sesuai dengan konteks penggunaannya.
- 4) Reliabilitas Merupakan tingkat dimana perangkat lunak dapat bertahan pada tingkat tertentu ketika digunakan oleh pengguna dalam kondisi tertentu, dalam hal ini perangkat lunak dapat beroperasi dan siap pada saat dibutuhkan untuk digunakan dan juga dapat menahan tingkat

kemampuan tertentu terhadap kegagalan, kesalahan dan pengembalian perangkat lunak sampai tingkat tertentu dalam memulihkan pengembalian data karena kegagalan atau kesalahan perangkat lunak.

Ciri-ciri tersebut terbagi menjadi beberapa sub ciri, yaitu:

- a. Kematangan, sejauh mana suatu produk atau sistem dapat memenuhi kebutuhan dengan andal dalam keadaan normal.
 - b. Ketersediaan, sejauh mana produk atau sistem siap untuk dioperasikan dan dapat diakses pada saat perlu digunakan.
 - c. Toleransi kesalahan, sejauh mana produk atau sistem tetap berjalan sebagaimana mestinya meskipun ada kesalahan pada perangkat keras atau perangkat lunaknya.
 - d. Dapat dipulihkan, sejauh mana produk atau sistem dapat memulihkan data yang terpengaruh secara langsung dan mengatur ulang kondisi sistem sesuai keinginan ketika terjadi gangguan.
- 5) Keamanan, sejauh mana suatu produk atau system melindungi informasi dan data sehingga seseorang atau system lain dapat mengakses data sesuai dengan jenis dan tingkat otorisasi yang dimilikinya. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu:
- a. Kerahasiaan, sejauh mana produk atau perangkat lunak memastikan data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
 - b. Integritas, sejauh mana produk atau perangkat lunak dapat mencegah akses tidak sah untuk mengubah data.

- c. Non repudiation, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan atas peristiwa atau tindakan tersebut.
 - d. Akuntabilitas, sejauh mana tindakan suatu entitas dapat dilacak secara unik ke entitas tersebut.
 - e. Keaslian, sejauh mana identitas subjek atau sumber dapat dibuktikan menjadi yang diklaim.
- 6) Portabilitas, sejauh mana efektivitas dan efisiensi suatu sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan dalam lingkungan yang berbeda. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu:
- a. Adaptabilitas, sejauh mana produk atau sistem dapat secara efektif dan efisien diadaptasi ke perangkat lunak, perangkat keras, dan lingkungan yang berbeda.
 - b. Peningkatan, sejauh mana produk atau sistem berhasil diinstal atau dihapus di lingkungan tertentu.
 - c. Dapat diganti yaitu sejauh mana suatu produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama di lingkungan yang sama.
- 7) Kinerja terhadap sejumlah sumber daya yang digunakan dalam kondisi tertentu, dalam hal ini efisiensi kinerja dapat memberikan reaksi dan waktu yang diperlukan ketika melakukan suatu tindakan dari suatu fungsi dan perangkat lunak dapat menggunakan jumlah sumber daya saat

melakukan tindakan dari suatu fungsi. Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Ciri ini terbagi menjadi beberapa sub ciri yaitu :

- a. Perilaku waktu, sejauh mana respon dan waktu pemrosesan produk atau sistem dapat memenuhi persyaratan saat menjalankan suatu fungsi.
 - b. Pemanfaatan sumber daya, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem dapat memenuhi persyaratan saat menjalankan fungsinya.
 - c. Kapasitas, sejauh mana batas maksimum produk atau parameter sistem dapat memenuhi persyaratan.
- 8) Pemeliharaan adalah tingkat dimana perangkat lunak dapat dimodifikasi. Dalam hal ini, modifikasi adalah perbaikan, perubahan atau penyesuaian pada perangkat lunak untuk dapat mengubah lingkungan, persyaratan dan fungsionalitas tertentu. Selain itu, perangkat lunak dapat dianalisis untuk mengetahui penyebab kegagalan perangkat lunak dalam mengidentifikasi bagian-bagian yang dapat dimodifikasi. Sejauh mana efektivitas dan efisiensi suatu produk atau sistem dapat dicapai diobati. Ciri ini terbagi menjadi beberapa sub ciri yaitu.
- a. Modularitas, sejauh mana sistem terdiri dari komponen yang terpisah sehingga perubahan atau modifikasi salah satu komponen tersebut berdampak kecil pada komponen lainnya.
 - b. Dapat digunakan kembali, sejauh mana aset dapat digunakan lebih banyak oleh satu sistem atau digunakan untuk membangun aset lain.

- c. Analisis, tingkat efektivitas dan efisiensi untuk menilai dampak perubahan pada satu atau lebih bagian dari suatu produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.
- d. Dapat dimodifikasi sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa mengurangi kualitas produk yang ada.
- e. Testabilitas, tingkat keefektifan dan efisiensi untuk membentuk kriteria pengujian suatu produk, sistem atau komponen dan pengujian dapat dilakukan untuk mengetahui apakah kriteria tersebut telah terpenuhi.

2.7.1. 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 (Jogiyanto, 2008). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). *Skala Likert* dapat dilihat pada Tabel 2.5 :

Tabel 2. 5 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 2.6.

Tabel 2. 6 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
$90 \leq x$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

Keterangan:

x = persentase hasil pengujian.