

BAB II

TINJAUAN PUSTAKA

2.1. Aplikasi

Menurut Maryono (2018) mendefinisikan aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Perangkat lunak aplikasi dibedakan menjadi beberapa macam berdasarkan kegunaannya.

2.2. *Logbook*

Menurut Febriani and Wahyuni (2017) mendefinisikan Buku Catatan Kegiatan Penelitian (*Logbook*) merupakan buku hasil rekaman atau catatan yang dilakukan oleh peneliti pada setiap kegiatan penelitiannya

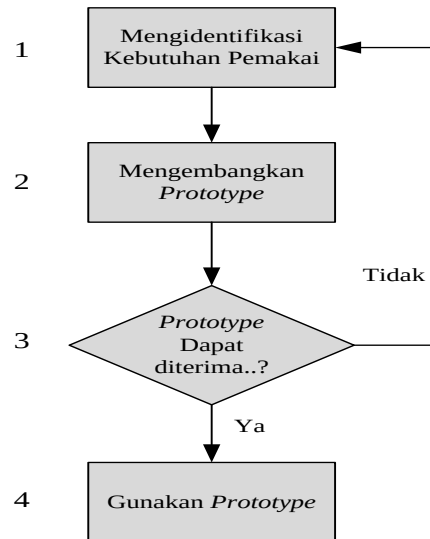
Menurut Febriani dan Wahyuni (2017) mendefinisikan *Logbook* adalah kegiatan mencatat, mengumpulkan data harian disebuah unit layanan dan laporan kegiatan rutin yang dilakukan setiap hari dari kegiatan telah terlaksana sampai dengan kegiatan yang akan dilaksanakan.

Berdasarkan beberapa pengertian diatas maka dapat disimpulkan bahwa *Logbook* adalah kegiatan mencatat atau merekam data harian disebuah unit layanan dan laporan kegiatan rutin yang dilakukan setiap hari dari kegiatan telah terlaksana sampai dengan kegiatan yang akan dilaksanakan.

2.3. Metode *Prototype*

Menurut McLoad (2017) mendefinisikan *Prototype* adalah suatu versi sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna yang memberikan gambaran bagaimana sistem akan berfungsi dalam bentuk yang

telah selesai akan berfungsi bila telah disusun dalam bentuk yang lengkap. Proses dalam memproduksi suatu *prototype* disebut *Prototyping*.



Gambar 2. 1 Metode Prototyping Evolutionary

Sumber: (Mcload,2017)

Ada empat tahapan dalam pengembangan sistem model *Evolutionary Prototype*, yaitu sebagai berikut:

1. Identifikasi kebutuhan pemakai

Pengembang mengidentifikasi terhadap pemakai untuk memperoleh suatu gagasan mengenai apa yang dibutuhkan dari sistem yang akan digunakan.

2. Mengembangkan *prototype*

Pengembang menggunakan satu atau lebih perkakas *prototyping* untuk mengembangkan satu *prototype*. Contoh perkakas *prototyping* adalah *integrated application generator* (perangkat pembuat aplikasi terintegrasi), yaitu sistem perangkat lunak *prewritten* yang mampu memproduksi semua fasilitas-fasilitas yang diharapkan ada dalam sistem baru serta dengan UML, melakukan rancangan membuat perancangan sementara yang berfokus pada

penyajian kepada pemakai (misalnya dengan membuat *input* dan format *output*).

3. Menentukan apakah *prototype* bisa diterima atau tidak

Tahap ini dilakukan oleh pemakai sistem apakah *prototyping* yang sudah dikembangkan bisa diterima atau tidak. Jika sudah sesuai maka langkah empat akan diambil, jika tidak *Prototyping* direvisi dengan mengulangi langkah satu, dua dan tiga dengan pemahaman yang lebih baik mengenai kebutuhan pemakai.

4. Gunakan *prototype*

Tahap ini dilakukan oleh pemakai sistem untuk menggunakan sistem yang telah dibangun.

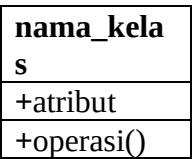
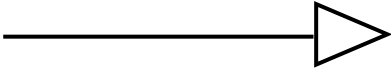
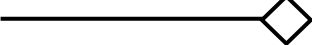
Pengertian UML (*Unified Modeling Language*)

Menurut Setyawan, Rafiq and Rosandy (2018) yang mengutip buku Rosa dan Shalahudin (2018) menyatakan 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.

1. *Class Diagram*

Menurut Rosa dan Shalahudin (2018) menyatakan Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas. Kelas memiliki apa yang disebut atribut dan metode atau operasi. simbol-simbol yang ada pada diagram kelas pada Tabel 2.2:

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>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 and Shalahudin, 2018)

2. *Use Case Diagram*

Menurut Rosa dan Shalahudin (2018) menyatakan *Use case diagram* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan.

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 Usecase Diagram

Simbol	Deskripsi
<i>Use Case</i> 	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>
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 aktor
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 aktor
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

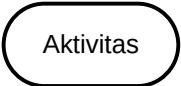
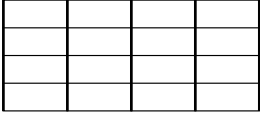
	fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>Include/uses</i> << <i>include</i> >>➔	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

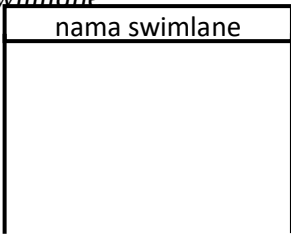
Sumber: (Rosa dan Shalahudin, 2018)

3. Activity Diagram

Menurut menyatakan bahwa *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 berikut:

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
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.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.4. Pengertian MySQL

Menurut MADCOM (2019) mendefinisikan bahwa MySQL adalah sistem manajemen Database SQL yang bersifat Open Source dan paling populer saat ini. Menurut Purwati, Halimah and Rahardi (2019) Sistem Database MySQL mendukung beberapa fitur seperti multithreaded, multiuser dan SQL Database managemen system (DBMS).

2.5. Xampp

Menurut Halimah *et al* (2018) dan Ali *et al* (2019) mendefinisikan bahwa Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain.

2.6. Aplikasi Mobile

Menurut Yuhefizar (2019) *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi.

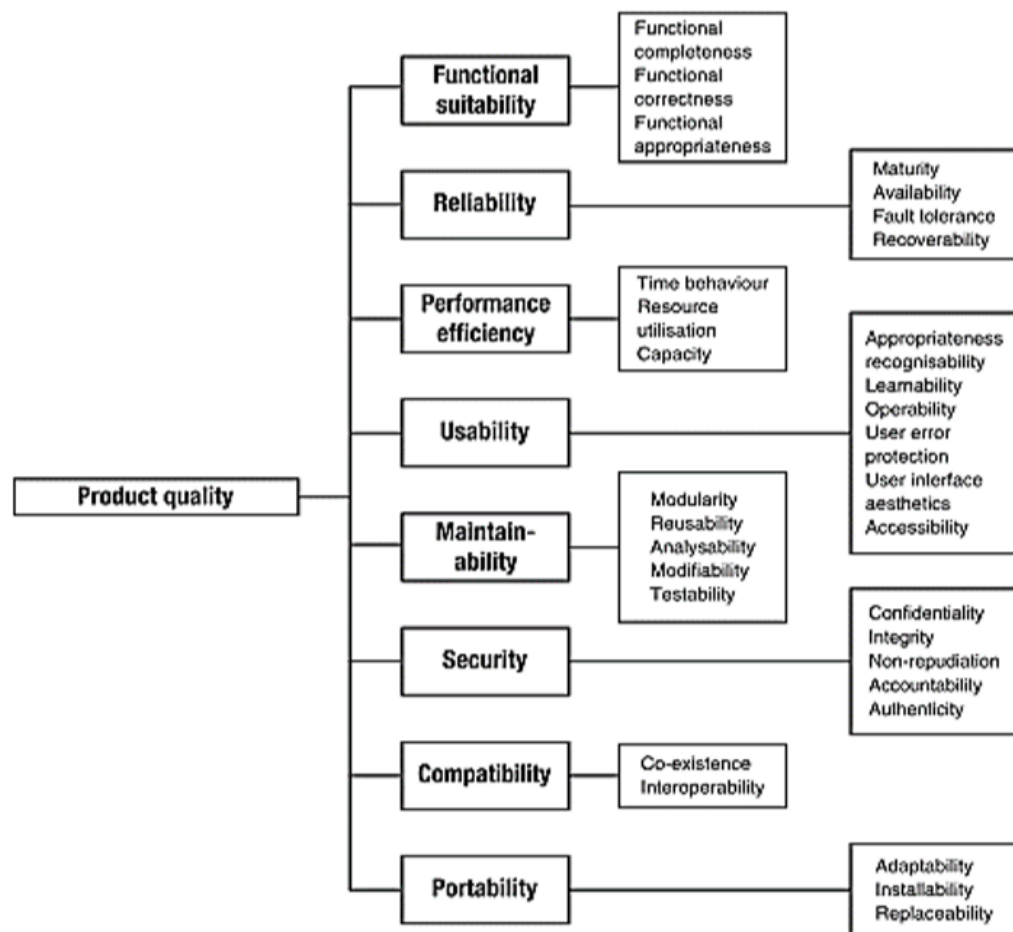
Sistem aplikasi mobile merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadipemutusan atau terputusnya komunikasi (Putri, Kurniawan and Sari, 2019). Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA (Yuhefizar, 2019). Adapun karakteristik perangkat *mobile* yaitu:

1. Ukuran yang kecil: Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas: Perangkat *mobile* juga memiliki memory yang kecil, yaitu *primary* (RAM) dan *secondary* (disk).
3. Daya proses yang terbatas: Sistem *mobile* tidaklah setangguh rekan mereka yaitu desktop.
4. Mengkonsumsi daya yang rendah: Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop
5. Kuat dan dapat diandalkan: Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

6. Konektivitas yang terbatas: Perangkat *mobile* memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung.
7. Masa hidup yang pendek : Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.7. Pengujian ISO 25010

Menurut Iqbal (2020) menyatakan bahwa 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 *dimensiquality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user* antara lain *Usability in use*, *Flexibility in use*, dan *Safety*.



Gambar 2. 2 Model kualitas produk ISO/IEC 25010

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. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu.

- a. *Functional completeness*, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna secara spesifik.
- b. *Functional correctness*, sejauh mana produk atau sistem menyediakan hasil yang benar sesuai kebutuhan.
- c. *Functional appropriateness*, sejauh mana fungsi yang disediakan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.

2) *Compatibility*

Sejauh mana sebuah produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen dan/atau menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat keras dan environment perangkat lunak yang sama. Karakteristik ini dibagi menjadi 2 karakteristik yaitu.

- a. *Co-existence*, sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien sementara berbagi sumber daya dengan produk atau sistem yang lain tanpa merugikan produk atau sistem tersebut.
- b. *Interoperability*, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.

3) *Usability*

Sejauh mana sebuah produk atau sistem dapat digunakan oleh user tertentu untuk mencapai tujuan dengan efektif, efisiensi, dan kepuasan

tertentu dalam konteks penggunaan. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu.

- a. *Appropriateness recognizability*, sejauh mana pengguna dapat mengetahui apakah sistem atau produk sesuai kebutuhan mereka.
 - b. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu yang belajar menggunakan sistem atau produk dengan efisien, efektif, kebebasan dari resiko dan kepuasan dalam konteks tertentu.
 - c. *Operability*, sejauh mana produk atau sistem mudah dioperasikan dan dikontrol.
 - d. *User error protection*, sejauh mana produk atau sistem melindungi pengguna terhadap membuat kesalahan.
 - e. *User interface aesthetics*, sejauh mana antarmuka pengguna dari produk atau sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.
 - f. *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai tujuan tertentu sesuai konteks penggunaan.
- 4) *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. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Maturity*, sejauh mana produk atau sistem mampu memenuhi kebutuhan secara handal di bawah keadaan normal.
- b. *Availability*, sejauh mana produk atau sistem siap beroperasi dan dapat diakses saat perlu digunakan.
- c. *Fault tolerance*, sejauh mana produk atau sistem tetap berjalan sebagaimana yang dimaksud meskipun terjadi kesalahan pada perangkat keras atau perangkat lunak.
- d. *Recoverability*, sejauh mana produk atau sistem mampu dapat memulihkan data yang terkena dampak secara langsung dan menata ulang kondisi sistem seperti yang diinginkan ketika terjadi gangguan.

5) *Security*

Sejauh mana sebuah produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan level otorisasi yang dimiliki. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu.

- a. *Confidentiality*, sejauh mana produk atau perangkat lunak memastikan data hanya bisa diakses oleh mereka yang berwenang untuk memiliki akses.

- b. *Integrity*, sejauh mana produk atau perangkat lunak mampu mencegah akses yang tidak sah untuk memodifikasi data.
- c. *Non-repudiation*, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan terhadap peristiwa atau tindakan tersebut.
- d. *Accountability*, sejauh mana tindakan dari suatu entitas dapat ditelusuri secara unik untuk entitas.
- e. *Authenticity*, sejauh mana identitas subjek atau sumber daya dapat terbukti menjadi salah satu yang diklaim.

6) *Portability*

Sejauh mana keefektifan dan efisiensi sebuah sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan pada lingkungan yang berbeda. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu.

- a. *Adaptability*, sejauh mana produk atau sistem dapat secara efektif dan efisien disesuaikan pada perangkat lunak, perangkat keras dan lingkungan yang berbeda.
- b. *Installability*, sejauh mana produk atau sistem dapat berhasil dipasang atau dihapus dalam lingkungan tertentu.
- c. *Replaceability*, sejauh mana produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama pada lingkungan yang sama.

7) *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. Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Time behaviour*, sejauh mana respon dan pengolahan waktu produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- b. *Resource utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- c. *Capacity*, sejauh mana batas maksimum parameter produk atau sistem dapat memenuhi persyaratan.

8) *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. Sejauh mana keefektifan dan efisiensi dari sebuah produk atau sistem dapat

dirawat. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Modularity*, sejauh mana sistem terdiri dari komponen terpisah sehingga perubahan atau modifikasi pada salah satu komponen tersebut memiliki dampak yang kecil terhadap komponen yang lain.
- b. *Reusability*, sejauh mana aset dapat digunakan lebih oleh satu sistem atau digunakan untuk membangun aset lain.
- c. *Analyzability*, tingkat efektivitas dan efisiensi untuk mengkaji dampak perubahan pada satu atau lebih bagian-bagian produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.
- d. *Modifiability*, sejauh mana produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menurunkan kualitas produk yang ada.
- e. *Testability*, tingkat efektivitas dan efisiensi untuk membentuk kriteria uji dari produk, sistem atau komponen dan uji dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.

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 (Sanusi, 2020). 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

Sumber : (Sanusi, 2020)

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
$90 \leq x$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

Sumber : (Sanusi, 2020)

Keterangan: x = persentase hasil pengujian.

2.8. Penelitian Terkait

Berikut ini merupakan beberapa Penelitian yang berhubungan dengan laporan skripsi ini yaitu:

Tabel 2. 6 Penelitian Terkait

No	Nama (Tahun)	Judul	Metode	Hasil	Kekurangan
1	Panggabea n, Budiman dan Pohny (2018)	Monitoring Kegiatan Kapal Pada Kantor Kesyahbandar an Dan Otoritas Pelabuhan Di Samarinda	SDLC	Hasil sistem ini merupakan sistem untuk memonitoring kapal yang digunakan untuk mengetahui informasi kegiatan kapal dengan cepat dan mudah. Dari hasil pengolahan tersebut menghasilkan sebuah laporan dalam bentuk cetak maupun file.	- masih dilakukan secara manual - lama dalam proses perekapan
2	Febriani dan Wahyuni (2018)	Perancangan Sistem <i>E-Document</i> Administrasi <i>Logbook</i> Penelitian Pada Unit Layanan Di Bandar Lampung	metode <i>waterfall</i>	sistem ini dapat memudahkan proses pembuatan sistem karena tahapan nya yang dilakukan secara bertahap dan menyesuaikan kebutuhan sistem. diharapkan dapat mempercepat pembuatan laporan <i>LogBook</i> serta meningkatkan kinerja dalam unit layanan tersebut tanpa batasan waktu serta pemanfaatan teknologi	- Sering terjadi kesalahan dalam mengelolaa n administراس i <i>logbook</i> - Lama dalam proses pencarian data

3	Faishal <i>et al</i> (2019)	Pengembangan Sistem Informasi Perjalanan Kapal dalam Satu Voyage pada PT. Lima Sekawan Marine Grup berbasis Website	RAD	Sistem ini dapat membantu dalam proses pemesanan kapal atau voyage charter. Sistem informasi ini berbasis web, dalam pengimplementasiannya menggunakan pola MVC (<i>Model View Controller</i>). Pada penelitian ini dilakukan pengujian yang terbagi menjadi pengujian unit, pengujian validasi, dan pengujian <i>compatibility</i>	- Belum dilakukan secara terkomputerisasi - Sering terjadi kekeliruan jadwal kapal
4	Andisa <i>et al</i> (2019)	Pengaruh Penerapan Sistem Absensi Online Dan Elogbook Terhadap Disiplin Kerja Pegawai Di Badan Kepegawaian Daerah Provinsi Kalimantan Timur	SDLC	Hasil penelitian menunjukkan bahwa secara parsial penerapan Sistem Absensi Online dan E-Logbook sama-sama memiliki pengaruh yang sedang, positif dan signifikan terhadap disiplin kerja pegawai di Badan Kepegawaian Daerah Provinsi Kalimantan Timur	- Sistem masih dilakukan secara manual - Sering terjadi kesamaan dalam melakukan pengelolaan data jadwal
5	Purnama and Wahju (2019)	Estimasi Kegiatan Alih Muat Pada Kapal Rawai Tuna Berdasarkan Data Vms Dan Komposisi Hasil Tangkapan	Metode deskriptif	Hasil penelitian ini menunjukkan bahwa pergerakan kapal yang hanya menampung hasil tangkapan dapat teridentifikasi dengan jelas melalui VMS.	- Sistem masih dilakukan secara manual - Tidak ada laporan secara lengkap

Berdasarkan penelitian terdahulu maka peneliti mengambil judul aplikasi *logbook* untuk kegiatan manajemen perusahaan yaitu sistem yang dibangun memiliki 3 hak akses yaitu admin, karyawan dan pimpinan. Admin dapat melakukan *login* sistem mengelola data jenis kegiatan, data kapal, mengelola data jadwal, melihat hasil kegiatan, dan mencetak laporan. Pimpinan dapat melakukan *login* dan mencetak laporan, sedangkan Karyawan dapat melakukan *login*, edit profile, input kegiatan. Sistem ini dibangun berbasis *mobile* dan diimplementasikan menggunakan bahasa pemrograman java dan MySQL sebagai *database*. Penelitian menggunakan metode pengembangan *prototype* dan akan diuji menggunakan metode pengujian sistem yaitu *ISO 25010*.