BAB II
TINJAUAN PUSTAKA

2.1 Sistem

Efendi dkk (2020) menyatakan bahwa “Sistem adalah bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud. Secara garis besar ada dua kelompok pendekatan sistem, yaitu Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya didefinisikan sebagai suatu jaringan kerja dari prosedur prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu aturan tertentu”.

2.2 Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhanpengelolaan harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Wicaksono dan Widodo,2020).

2.3 WEB Aplication

Menurut Setyawan dan Munari (2020), aplikasi berbasis *web* ini menggunakan protokol HTTP, aplikasi di sisi *server* berkomunikasi dengan client melalui *web server*. Aplikasi di sisi client umumnya berupa *web browser*. Jadi, aplikasi berbasis *web* (*client/server-side script*) berjalan diatas aplikasi berbasis internet.

2.4 Karakteristik WEB

Menurut Setyawan dan Munari (2020), karakteristik aplikasi berbasis *web* antara lain:

1. Kepadatan jaringan aplikasi *web* umumnya berada pada suatu jaringan komputer dan harus bisa melayani kebutuhan komunitas klien yang beragam.

2. Keserempakan sejumlah besar pengguna mungkin akan mengakses aplikasi secara serempak.
3. Jumlah pengguna yang tidak dapat diprediksi Jumlah pengguna mungkin beragam dari hari ke hari. Pengguna di hari libur mungkin berada di hari kerja.
4. Kinerja penggunaan aplikasi berbasis *web* akan pergi jika pengguna terlalu lama menunggu aplikasi *web.*
5. Ketersediaan menyediakan akses layanan 24 jam.
6. Digerakkan oleh data beberapa aplikasi *web* bergantung pada layanan data, contoh: *online shop.*
7. Peka terhadap isi kualitas isi dan keindahan tetap merupakan faktor penting kualitas *web.*
8. Evolusi yang berkesinambungan merupakan hal yang biasa untuk aplikasi *web* jika ada pembaharuan setiap saat.
9. Keamanan terhubung ke internet, menyebabkan aplikasi *web* rentan serangan dari pihak lain.
10. Estetika salah satu daya tarik aplikasi *web* adalah tampilan dan nuansanya.
11. Kesegeraan aplikasi *web* sering memerlukan kedisiplinan waktu untuk merilis produk ke pasar dalam waktu beberapa hari atau beberapa minggu.

2.5 Bahasa Pemrograman

2.5.1 PHP

Menurut Yudhanto & Prasetyo (2019) "PHP atau *Hypertext Preprocessor* adalah Bahasa pemograman script *server* side yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan *web*".

2.5.2 HTML

Menurut Saputra (2019), yaitu "HTML atau *Hyper Text* Markup *Language* merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat laman *website* yang dapat diakses atau ditampilkan menggunakan *web browser*."

2.5.3 CSS

Menurut Ummy Gusti Salamah, S.ST,.MIT (2021), yaitu "CSS (*Cascading Style Sheet*) adalah bahasa yang dapat digunakan untuk mendefinisikan bagaimana suatu bahasa markup ditampilkan pada suatu media dimana bahasa markup ini salah satunya adalah HTML."

2.5.4 Xampp

Menurut Aryanto dalam (Kesuma & Kholifah, 2019) "XAMPP merupakan sebuah aplikasi perangkat lunak pemrograman dan database yang didalamnya terdapat berbagai macam aplikasi pemrograman seperti; Appache HTTP *Server*, *database* MySQL, bahasa pemrograman PHP serta *Perl*."

2.5.5 Javascript

Menurut Siahaan & Rismon (2020), yaitu "*JavaScript* adalah sebuah bahasa script dinamis yang dapat dipakai untuk membangun interaktifitas pada halamanhalaman HTML statis. Ini dilakukan dengan menamakan blok-blok kode *JavaScript* di hamper semua tempat pada halaman *web*."

2.6 Database PHPMyAdmin

Menurut Agung Baitul, dkk. dalam (Erawati, 2019) bahwa "PHPMyAdmin merupakan aplikasi yang dapat digunakan untuk membuat database, pengguna (user), memodifikasi tabel, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan perintah (*command*) SQL."

2.7 UML(Unified Modeling Language)

Menurut Sukamto dan Shalahuddin (2019), "Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu,

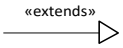
meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek". Beberapa diagram dalam UML yang akan digunakan dalam membantu pengembangan sistem yaitu :

Alat Bantu perancangan sistem yang dapat digunakan adalah :

2.7.1 Use Case Diagram

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *Use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami Sukamto dan Shalahuddin (2019). Adapun simbol-simbol *Use case* diagram dapat dilihat pada table dibawah ini.

Tabel 2.1 Tabel *Use case* Diagram

Keterangan	Simbol	Deskripsi
Use Case		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; dinyatakan dengan menggunakan kata kerja diawal-awal frase nama *Use case*
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
Asosiasi		Komunikasi antara aktor dan *Use case* memiliki interaksi dengan aktor
Ekstensi		Relasi *Use Case* tambahan ke sebuah *Use case*, dimana *Use case* yang ditambahkan dapat berdiri sendiri walau tanpa *Use case* tambahan itu; mirip

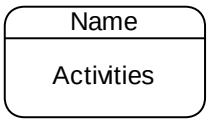
Keterangan	Simbol	Deskripsi
		dengan prinsip *inheritance* pada pemograman berorientasi objek.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *Use Case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ include/uses		Ada dua sudut pandang yang cukup besar mengenai include di *Use case*: a. Include berarti *Use case* yang ditambahkan akan selalu dipanggil saat *Use case* tambahan dijalankan b. Include berarti *Use case* yang tambahan akan selalu melakukan pengecekan apakah *Use case* yang ditambahkan telah dijalankan sebelum *Use case* tambahan dijalankan.

2.7.2 Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem Sukamto dan Shalahuddin (2019). Simbol-simbol yang terdapat pada activity diagram dapat dilihat pada table dibawah ini.

Tabel 2.2 Tabel Activity Diagram

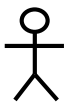
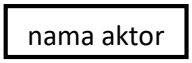
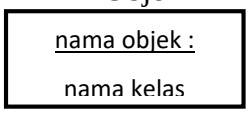
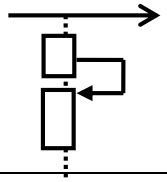
Keterangan	Simbol	Deskripsi
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

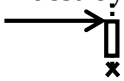
Keterangan	Simbol	Deskripsi
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.7.3 Sequence Diagram

Sequence diagram menggambarkan kelakukan objek pada *Use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *Use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat Sequence diagram juga dibutuhkan untuk melihat skenario yang ada pada *Use case* Sukamto dan Shalahuddin (2019). Adapun *symbol sequence diagram* dapat diliat pada table dibawah ini.

Tabel 2.3 Tabel *Sequence Diagram*

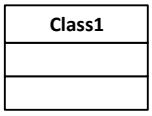
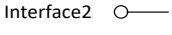
Simbol	**Deskripsi**
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari
nama actor Atau  Tanpawaktu aktif	Aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Garishidup/ lifeline	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create <<create>> 	Menyatakan suatu objek membuat objek lain, arah panah objek yang dibuat
Pesan tipe call 1: nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri 1 : nama_metode()
Pesan tipe send 1 : Masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi
Pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian

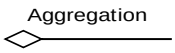
Simbol	Deskripsi
	ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, s ebaiknya jika ada create maka ada destroy

2.7.4 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. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas Sukamto dan Shalahuddin (2019).. Simbol-simbol yang ada pada *class diagram* dapat dilihat dibawah ini.

Tabel 2.4 Tabel Class Diagram

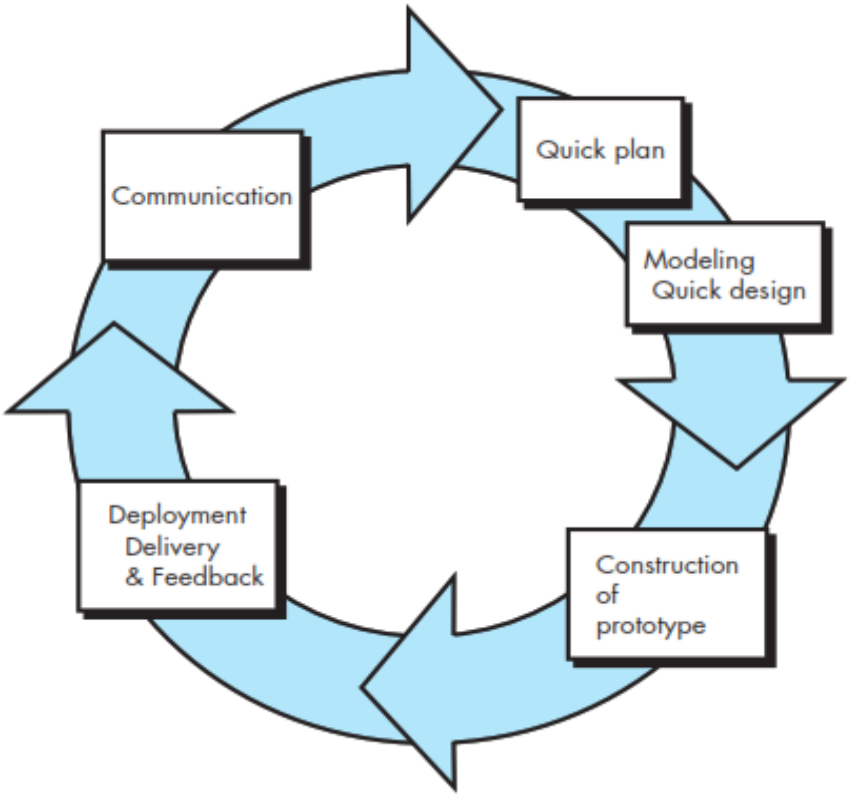
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
atarmuka/interface 	Sama dengan konsep interface dalam pemograman berorientasi objek.
Asosiasi	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan	Relasi antar kelas dengan makna kebergantungan antar kelas.

Simbol	Deskripsi
Agregasi 	Relasi antar kelas dengan makna semua bagian (whole-part).

2.8 Metode Pengembangan Sistem

Metodelogi pengembangan sistem pada penelitian ini menggunakan metode *Prototype*. Menurut Candra Novitasari (2020) Metode Prototype merupakan satu metode dalam pengembangan perangkat lunak, metode ini merupakan suatu paradigma baru dalam pembuatan / pengembangan perangkat lunak. dan juga salah satu metode Prototype merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai.

Metode *Prototype* ini dapat digambarkan sebagai berikut :

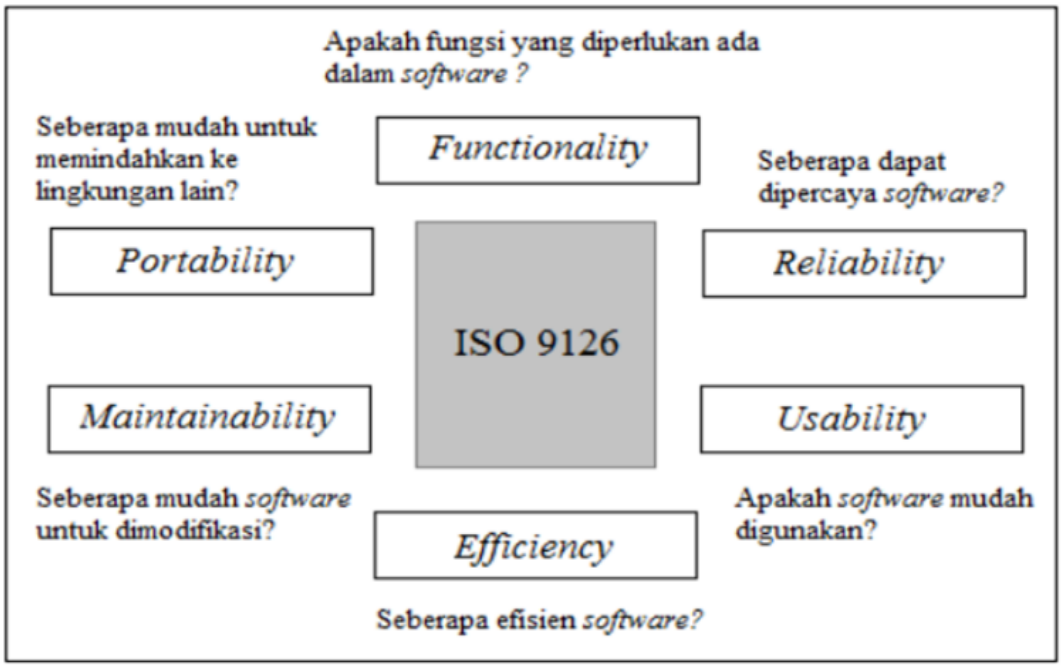


Gambar 2.1 Metode *Prototype*

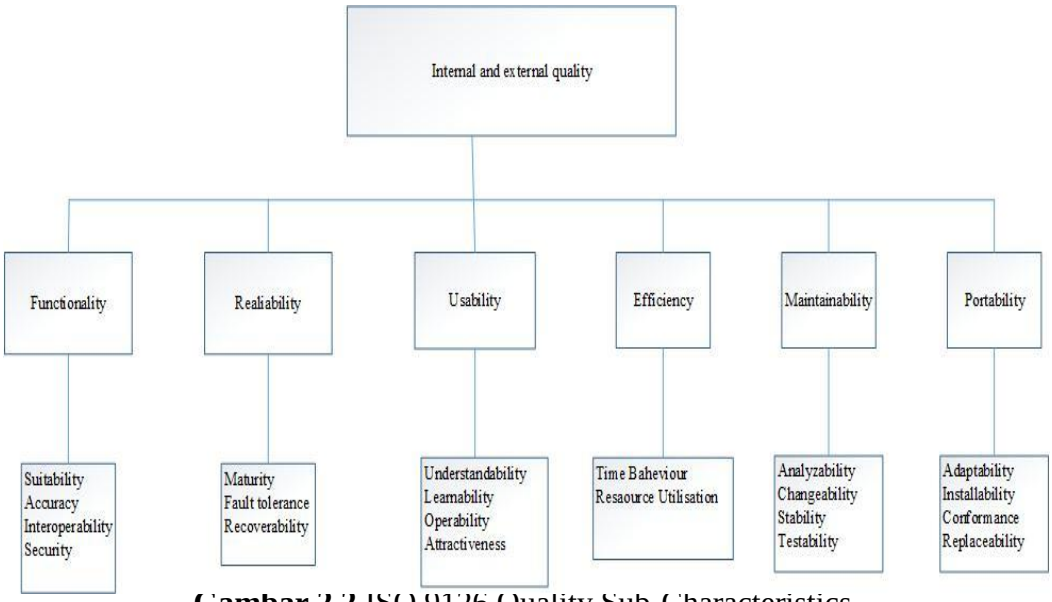
2.9 ISO 9126

Menurut (Al-Qutaish, 2019) model kualitas ISO 9126-1 yang dibuat oleh *International Organization for Standardization* (ISO) dan *International Electro technical Commission* (IEC) ini adalah model yang paling efisien karena pengembangannya berdasarkan konsensus internasional dan merupakan persetujuan dari semua negara anggota organisasi ISO.

Dikutip dalam buku (Tian, 2005) yang berjudul *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, ISO* 9126 menyediakan sebuah *framework* yang hirarki untuk menjelaskan kualitas yang terorganisir dalam karakteristik dan sub-karakteristik kualitas. Modelkualitas *ISO* 9126 mempunyai 6 karakteristik yang dapat dilihat pada Gambar 2.2.



Dari enam karakteristik tersebut, masing-masing karakteristik mempunyai sub-karakteristik. Masing-masing sub-karakteristik dapat dilihat pada Gambar 2.2.



Gambar 2.2 ISO 9126 Quality Sub-Characteristics

Berikut ini penjelasan dari Gambar 2.2 tentang karakteristik dan sub karakteristik dalam ISO 9126 :

a) Functionality

Functionality merupakan satu kumpulan atribut yang memuat adanya satu kumpulan fungsi dan spesifikasi dari properties-nya (Tian, 2005). *Functionality* mencakup kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu. Sub-karakteristik *functionality* meliputi *suitability, accuracy, interoperability*, dan *security*. Berikut ini penjelasan untuk masing-masing sub-karakteristik *functionality*.

Tabel 2.5 Penjelasan Sub-karakteristik *Functionality*

Karakteristik	Sub-Karakteristik	Penjelasan
Functionality	*Suitability*	Apakah software dapat melakukan tugas yang diperlukan?
	Accuracy	Apakah hasil sesuai dengan yang diharapkan?

Karakteristik	Sub-Karakteristik	Penjelasan
	Interoperability	Apakah sistem berinteraksi dengan sistem lain?
	Security	Apakah software menghalangi unauthorized access?

b) *Reliability*

Reliability merupakan satu kumpulan atribut yang meliputi kapabilitas *software* untuk memelihara tingkat performa dalam suatu kondisi pada waktu tertentu (Tian, 2005). *Reliability* didefinisikan kemampuan mempertahankan tingkat kinerja dalam suatu kondisi. Sub-karakteristik *reliability* meliputi *maturity, fault tolerance*, dan *recoverability*. Penjelasan untuk masing-masing subkarakteristik *reliability* dapat dilihat pada Tabel 2.6

Tabel 2.6 Penjelasan Sub-karakteristik *Reliability*

Karakteristik	Sub-Karakteristik	Penjelasan
Reliability	*Maturity*	Seberapa banyak kesalahan dalam software dapat dihapuskan dalam waktu tertentu?
	Fault tolerance	Apakah software mampu menangani kesalahan?
	Recoverability	Dapatkah software meneruskan kinerja dan mengembalikan data yang hilang setelah terjadi gangguan?

c) *Usability*

Usability adalah satu kumpulan atribut yang memuat usaha yang dibutuhkan untuk digunakan dan penilaian masing-masing individu atas penggunaannya yang dinyatakan secara langsung maupun tidak (Tian, 2005). Sub-karakteristik *Usability* meliputi *understandability, learnability, operability*,

dan *attractiveness.* Penjelasan untuk masing-masing sub-karakteristik *usability* dapat dilihat pada Tabel 2.7.

Tabel 2.7 Penjelasan Sub-karakteristik *Usability*

Karakteristik	Sub-Karakteristik	Penjelasan
Usability	*Understandability*	Apakah pengguna memahami bagaimana menggunakan sistem dengan mudah?
	Learnability	Dapatkah pengguna belajar menggunakan sistem dengan mudah?
	Operability	Dapatkah pengguna menggunakan sistem tanpa upaya yang lebih?
	Attractiveness	Apakah antarmuka terlihat baik?

d) *Efficiency*

Efficiency yaitu satu kumpulan atribut yang dikenakan pada hubungan antara tingkat kinerja perangkat lunak dan jumlah sumber daya yang digunakan, dalam kondisi tertentu (Tian, 2005). Sub-karakteristik *efficiency* meliputi *time behavior* dan *resource utilization*. Penjelasan untuk masing-masing subkarakteristik efficiency dapat dilihat pada Tabel 2.8.

Tabel 2.8 Penjelasan Sub-karakteristik *Efficiency*

Karakteristik	Sub-Karakteristik	Penjelasan
Efficiency	*Time Behaviour*	Seberapa cepat respon sistem?
	Resource Utilisation	Apakah sistem menggunakan sumber dengan efisien?

e) *Maintainability*

Maintainability adalah satu kumpulan atribut yang dikenakan pada usaha yang diperlukan untuk membuat modifikasi tertentu (Tian, 2005). Subkarakteristik *maintainability meliputi analyzability*, *changeability*, *stability*, dan *testability*.

Penjelasan untuk masing-masing sub-karakteristik *maintainability* dapat dilihat pada Tabel 2.9

Tabel 2.9 Penjelasan Sub-karakteristik *Maintainability*

Karakteristik	Sub-Karakteristik	Penjelasan
Maintainability	*Analyzability*	Dapatkah kesalahan didiagnosa dengan mudah?
	Changeability	Dapatkah software dimodifikasi dengan mudah?
	Stability	Dapatkah software tetap berfungsi jika ada perubahan?
	Testability	Dapatkah software dapat diuji dengan mudah?

f) *Portability*

Portability merupakan satu set atribut yang dikenakan pada kemampuan perangkat lunak yang akan ditransfer dari satu *device* ke *device* yang lainnya (Tian, 2005). Sub-karakteristik *Portability* meliputi *adaptability*, *installability*, *coexistence*, dan *replaceability.* Penjelasan untuk masing-masing subkarakteristik portability dapat dilihat pada Tabel 2.10.

Tabel 2.10 Penjelasan Sub-karakteristik *Portability*

Karakteristik	Sub-Karakteristik	Penjelasan
Portability	*Adaptability*	Dapatkah software dipindah ke lingkungan lain?
	Installability	Dapatkah software di-install dengan mudah?
	Conformance	Apakah software memenuhi standar portability?
	Replaceability	Dapatkah software dengan mudah menggantikan software lain?

2.10 Referensi Penelitian Terkait

Tabel 2.11 Tabel Referensi Penelitian Terkait

No	Nama Peneliti	Judul Penelitian	Tahun	Metode	Hasil	Sumber
1	Jiki Romadoni, Anjas Safendra	Rancang Bangun Sistem Informasi Himpunan Mahasiswa Teknik Informatika Politeknik Hasnur Berbasis Web	2022	*Waterfall*	Sistem pengelolaan data logistik dan inventaris yang mana sistem ini dirancang untuk membantu memberikan kemudahan untuk melakukan pengelolaan data dan pengarsipan data-data.	ISSN 2722-0524
2	Meliyana Winda Perdana, Dedi Haryanto, Aminullah Imal Alfresi, Syafi'ul	Sistem Informasi Himpunan Mahasiswa Berbasis web pada Prodi Teknologi Informasi	2022	UML	Sistem Informasi himpunan mahasiswa teknologi informasi (HMTI) Universitas Muhammad	*E-ISSN 2714-9706*

No	Nama Peneliti	Judul Penelitian	Tahun	Metode	Hasil	Sumber
	Hamidini, Aidil Fadli Tegriansyah	Universitas Muhammadiyah Palembang			iyah Palembang tempat untuk mempromosikan organisasi mereka, dan untuk meningkatkan kinerja organisasi	
3	Hanni Funica Granatuma, Arum Fatayan	Analisis Prestasi Peserta Didik dari Sistem Manajemen Berbasis Sekolah di Sekolah Dasar Islam	2022	kualitatif deskriptif	Hasil dari penelitian menjelaskan bahwa Prestasi peserta didik di sekolah dasar islam sangat baik dilihat dari penerapan manajemen berbasis sekolah. Dimulai dari	Jurnal Basicedu Vol 6 Nomor 3 Tahun 2022

No	Nama Peneliti	Judul Penelitian	Tahun	Metode	Hasil	Sumber
					kepemimina n yang baik, pengemban gan kurikulum, manajemen peserta didik berbasis sekolah, pendidik dan tenaga kependidika n, dukungan sarana pra sarana serta hubungan masyarakat/ orang tua dengan sekolah.	
4	Rani Puspita Dhaniawat y, Erna Susilawati	Pembangun an Sistem Informasi Pelaporan Program Kerja dan Pengelolaan Data	2022	*Prototype*	Sistem informasi yang sudah dibuat pada penelitian ini membantu dosen	*Jurnal Informatik a, Vol. 15, No. 2,*

No	Nama Peneliti	Judul Penelitian	Tahun	Metode	Hasil	Sumber
		Pengurus Himpunan Mahasiswa pada Program Studi Sistem Informasi			kemahasiswaan dalam mengelola data pengurus HIMA SI, mengintegrasikan jadwal proker Prodi SI dengan HIMA SI, selain itu dosen kemahasiswaan dapat melakukan laporan bulanan proker HIMA SI dengan menggunakan data yang akurat dan tepat waktu.	