BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam penelitian ini akan digunakan sepuluh tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan studi yang diambil yaitu:

Tabel 2.1 Studi Literatur

1	Judul	Pemodelan Aplikasi Dunia Islam Mengaji Berbasis Android
	Penulis	Eva Rahmawati, Ehon Abdulmanan
	Tanggal/Tahun	2019
	Tujuan Penelitian	Memberi pemahaman tentang penggunaan teknologi mobile untuk belajar maupun mendapatkan mendapatkan informasi
	Permasalahan	Kurangnya pemahaman dalam pengguna teknologi mobile
	Subjek Penelitian	-
	Metode Penelitian	Metode Waterfall
	Hasil Penelitian	Memudahkan pengguna yang ingin belajar baik lewat video maupun teks serta mengetahui jadwal waktu sholat
2	Judul	Nilai-Nilai Dakwah Dalam Aplikasi Bisnis PayTren
	Penulis	T. Emy Kurniawan
	Tanggal/Tahun	2018
	Tujuan Penelitian	Pengetahuan tentang nilai-nilai dakwah dalam aplikasi PayTren
	Permasalahan	Bagaimana hubungan aplikasi Paytren dengan nilai-nilai dakwah serta tanggapan mitra Paytren terhadap Bisnis PayTren
	Subjek Penelitian	Pengguna Paytren
	Metode Penelitian	Metode Observasi dan Kuesioner
	Hasil Penelitian	Mengetahui nilai-nilai dakwah dalam aplikasi PayTren dan upaya menciptakan ekonomi islam yang merata

Tabel 2.1 Studi Literatur (lanjutan)

3	Judul	Pengembangan Sistem Informasi Pelayanan Wakaf Uang Berbasis Android
	Penulis	Abdillah Ahsan
	Tanggal/Tahun	2018
	Tujuan Penelitian	Meningkatkan value (nilai) dalam pendaftaran wakaf
	Permasalahan	Penurunan jumlah muwakif pendaftar wakaf
	Subjek Penelitian	Wakaf Center
	Metode Penelitian	Observasi dan Interview
	Hasil Penelitian	Dapat dikembangkan kembali dari sisi keamanan, serta terintegrasi dengan Bank
4	Judul	Perancangan Aplikasi Perhitungan Zakat Mal, Menentukan Waktu Sholat dan Arah Kiblat menggunakan GPS Berbasis Andorid
	Penulis	Yasmin Dara, Denny Kurniadi, Khairi Budayawan
	Tanggal/Tahun	2014
	Tujuan Penelitian	Memanfaatkan penggunaan teknologi informasi *mobile* untuk perhitungan zakat mal, penentuan waktu sholat serta arah kiblat
	Permasalahan	Membangun Aplikasi Android menggunakan java sebagai Bahasa pemograman dan Eclipse sebagai IDE serta penggunaan GPS sebagai penentu lokasi
	Subjek Penelitian	-
	Metode Penelitian	Metode Kualitatif dan Deskriptif.
	Hasil Penelitian	Aplikasi ini dapat membantu pengguna dalam perhitungan Zakat Mal, pengingat untuk melaksanakan sholat dan menentukan arah kiblat
5	Judul	Aplikasi pencarian ustadz untuk wilayah kota makassar menggunakan algoritma floyd warshall dan haversine formula berbasis android
	Penulis	Rifaldy Ramadhan Latief, Andani Achmad, Supriadi Sahibu
	Tanggal/Tahun	2019
	Tujuan Penelitian	Memudahkankan masyarakat mencari penceramah dan memudahkan penceramah mencari lokasi ceramah
	Permasalahan	Aplikasi hanya dikembangkan untuk wilayah Bandar Lampung
	Subjek Penelitian	-

	Metode Penelitian	Metode Kualitatif
	Hasil Penelitian	Aplikasi mobile dimana masyarakat dapat mencari penceramah dan penceramah dapat menentukan lokasi ceramah dengan lebih mudah

Berdasarkan tinjauan pustaka diatas maka perbedaan antara penelitian terdahulu dengan penelitian yang dilakukan adalah:

1. Pengembangan sistem yang akan digunakan adalah UML (*Unit Modelling Language*)
2. Analisis kebutuhan sistem
3. Mengunakan metode prototype
4. Pengujian sistem mengunakan *ISO* 25010 adapun aspek yang akan diuji yaitu aspek *functionality usability* dan aspek *Usability*
5. Sistem yang akan dibuat menggunakan *Android Studio*

2.2 Pengertian Rancang Bangun

Rancang bangun merupakan salah satu bagian penting dalam proses produksi. Ini dirancang untuk memberikan gambaran yang lengkap dan jelas untuk memprogram dan pakar teknis terkait.

Rancang bangun harus berguna dan mudah dipahami agar mudah digunakan. Menurut (Destiani et al., 2019), perancang adalah rangkaian proses yang menerjemahkan hasil analisis dan sistem ke dalam bahasa pemrograman untuk menggambarkan secara detail bagaimana komponen sistem diimplementasikan.

Menurut Pressman yang dikutip oleh Taufan dalam Jurnal Elektronik Teknik Informatika Vol. 11 No. 1 (2017), konsep membangun atau membangun

sistem adalah kegiatan membuat sistem baru atau mengganti atau memperbaiki seluruh sistem yang ada.

Dengan demikian Rancang Bangun adalah adalah menggambar, merencanakan dan membuat sketsa atau analisis ke dalam paket perangkat lunak dan kemudian membuat sistem atau memperbaiki sistem tersebut.

2.3 Dewan Dakwah

Dewan Dakwah adalah organisasi keagamaan yang ikut serta dalam dakwah lillah dengan melaksanakan al amru bi al ma'ruf wa an nahyu 'anil munkar. Panitia dakwah memiliki sifat sosial dan kemanusiaan, yang tercermin dalam kegiatan membangun umat, masyarakat, bangsa dan negara, terutama dalam tuntutan 'aqidah, penerapan hukum syariah, promosi persatuan ummat, dan dukungan untuk keutuhan negara kesatuan Republik Indonesia dan membangun persatuan umat di dunia.

2.4 Aplikasi Mobile

2.4.1 Pengertian Aplikasi Mobile

Menurut (Salamun, 2017), aplikasi mobile berasal dari kata application dan mobile. Aplikasi berarti menerapkan, menggunakan. Dari segi aplikasi, adalah program siap pakai yang dirancang untuk menjalankan suatu fungsi bagi pengguna atau aplikasi lain dan dapat digunakan oleh sasaran yang dimaksud, sedangkan perpindahan dapat diartikan sebagai berpindah dari satu tempat ke tempat lain.

Kata *mobile* berarti bergerak atau bergerak, jadi menurut (Purnama, 2010) adalah aplikasi yang berjalan pada perangkat mobile. Melalui aplikasi mobile,

Anda dapat dengan mudah melakukan berbagai aktivitas seperti hiburan, jualan, belajar, kerja kantoran, browsing, dan bermain game.

Penggunaan aplikasi *mobile* untuk untuk hiburan merupakan yang paling populer di kalangan pengguna ponsel karena dengan memanfaatkan fungsi game, music player dan video player, kita dapat lebih mudah menikmati hiburan kapanpun dan dimanapun.

Perngakat *mobile* tersedia dalam berbagai ukuran, desain, dan tata letak, tetapi memiliki karakteristik yang berbeda dari sistem desktop. Perangkat seluler memiliki memori yang sangat sedikit.

2.4.2 Karakteristik Perangkat Mobile

Perangkat *mobile* tersedia dalam berbagai ukuran, desain, dan tata letak, tetapi memiliki karakteristik yang berbeda dari sistem desktop, termasuk:

1. Ukuran yang kecil

   Perangkat *mobile* berukuran yang kecil. Konsumen menginginkan perangkat terkecil untuk kenyamanan dan mobilitas.

2. Memory yang terbatas

   Perangkat *mobile* juga memiliki *memory* kecil, yaitu *primary* (RAM) dan *secondary* (*disk*). Keterbatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk banyak jenis perangkat ini. Karena jumlah memori yang terbatas, perawatan khusus harus dilakukan untuk mempertahankan penggunaan sumber daya yang mahal ini.

3. Kekuatan pemrosesan terbatas

Sistem *mobile* tidak sekuat sistem desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status sumber daya ini. Seperti hard drive dan RAM, Anda dapat menemukannya dalam ukuran kecil.

4. Data konsumsi rendah

Perangkat *mobile* mengkonsumsi lebih sedikit daya daripada komputer desktop. Perangkat ini harus menghemat daya karena beroperasi dalam keadaan di mana catu daya dibatasi oleh baterai.

5. Kuat dan andal

Karena perangkat *mobile* seluler selalu bersama Anda, perangkat tersebut harus cukup tangguh untuk menahan benturan, gerakan, dan tetesan air sesekali.

6. Konektivitas terbatas

Perangkat mobile memiliki bandwidth rendah dan beberapa bahkan tidak memiliki koneksi. Kebanyakan dari mereka menggunakan koneksi nirkabel.

7. Umur pendek

Perangkat konsumen ini menyala dalam hitungan detik, dan sebagian besar selalu menyala. Ambil ponsel misalnya, mereka boot dalam hitungan detik dan kebanyakan orang bahkan tidak mematikan ponsel mereka di malam hari. Jika Anda menekan tombol daya PDA, PDA akan menyala.

2.5 Android

Menurut (Abd. Basith, 2022) “Android adalah sistem operasi untuk perangkat bergerak berbasis Linux, termasuk sistem operasi, middleware, dan aplikasi. Android menyediakan platform terbuka bagi pengembang untuk membuat aplikasi. Android sesuai kebutuhan Dikembangkan dari satu versi ke yang lain, beberapa versi Android yang dirilis sejauh ini termasuk Android 1.1 dirilis pada 2008 dan Android 1.5 (Cupcake) dirilis pada tahun yang sama, dan kemudian Android 1.6 (Donut), versi Android 2.1 dinamai For Eclair, Android 2.2 (Froyo) dirilis tahun berikutnya, Android 2.3 (Gingerbread) dirilis, kemudian Android 3.0/3.1 (Honeycomb) yang dirancang khusus untuk tablet. Kemudian meluncurkan Android 4.0 (Ice Cream Sandwich) sebelum merilis Android 4.1 (Jelly Beans) pada tahun 2012 dan Android 4.4 (Kitkat) pada tahun 2013. Ini diikuti oleh Android 5.1 (Lollipop), Android 6.0 (Marshmallow), dan terbaru, Android 7.0 (Nougat). Banyak versi Android yang dirilis merupakan pembaruan dan penambahan fitur sebagai perbaikan dari versi sebelumnya. Dari Android versi 1.1 hingga Android versi 1.6 (Donut), hadir dengan pembaruan untuk kamera, galeri, bluetooth, kontak telepon, resolusi layar, dan jaringan VPN. Kemudian pada Android versi 2.2 (Eclair) menjadi Android versi 2.3 (Gingerbread) dengan optimalisasi hardware, dukungan flash pada kamera, dukungan HTML5, optimalisasi kecepatan, memori dan dukungan USB tethering atau Wi-Fi hotspot, penambahan power management dan peningkatan performa serta penambahan sensor. Berbeda dengan versi Android sebelumnya, Android 4.0 (Ice Cream Sandwich) dan Android 4.1 (Jelly Beans) memiliki keunggulan memiliki platform yang dapat berjalan di tablet dan ponsel, konsumsi baterai lebih hemat, dan

peningkatan performa yang lebih signifikan. Pada saat yang sama, versi Android 4.4 (Kitkat) lebih bersahabat dengan perangkat dengan spesifikasi sementara. Salah satu perubahan besar dalam rilis Android 5.1 (Lollipop) adalah dukungan untuk arsitektur 64-bit, memungkinkan lebih dari 3 GB RAM digunakan pada perangkat perangkat keras. Secara visual, Android 6.0 (Marshmallow) hampir identik dengan Android 5.1 (Lollipop), kecuali fitur keamanan sistem operasi telah lebih ditingkatkan dalam versi ini. Versi Android terbaru adalah Android 7.0 (Nougat), pembaruan yang menambahkan emoji dan Google Assistant, dan sistem operasi dapat berbicara dua bahasa sekaligus.

2.5.1 Android Software Development Kit (Android SDK)

Menurut (Basith & Kom, 2022), mengemukakan bahwa, "Android SDK adalah alat yang diperlukan untuk mengembangkan aplikasi berbasis Android dengan menggunakan bahasa pemrograman Java". Pada titik ini Android SDK telah menjadi alat dan antarmuka pemrograman aplikasi (API) untuk mengembangkan aplikasi berbasis Android.

2.5.2 Android Studio

Menurut (- STMIK Nusa Mandiri Jakarta et al., 2018) mengemukakan bahwa, "Android Studio adalah IDE yang tersedia untuk pengembangan aplikasi Android, yang dikembangkan oleh Google. Android Studio merupakan evolusi dari Eclipse IDE, berdasarkan Java IDE yang populer, IntelliJ IDEA. Android Studio adalah pengembangan dari Eclipse IDE, berdasarkan Java IDE yang populer, IntelliJ IDEA. Android Studio berencana untuk menggantikan Eclipse sebagai IDE resmi untuk pengembangan aplikasi Android. Android Studio memiliki banyak fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang

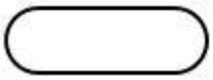
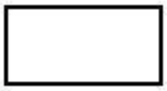
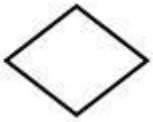
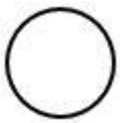
menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment-nya. Fitur lainnya adalah sebagai berikut:

1. Gunakan system *Gradle-based build* system yang *fleksibel*.
2. Bisa mem-*build multiple APK*.
3. *Template* mendukung untuk *Google Services* dan berbagai macam tipe perangkat.
4. *Layout editor* yang lebih baik.
5. *Built-in support* untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
6. Mampu Import library dari *Maven repository*.

2.6 Flowchart

Menurut (Rudi Haryadi, 2021) Flowchart adalah penggambaran secara grafik berdasarkan langkah-langkah dan urutan prosedur suatu program. Flowchart dapat menyelesaikan suatu permasalahan yang terdapat pada aliran algoritma dalam penelitian. Flowchart berikut ini merupakan aliran algoritma yang telah dirancang dalam pengembangan sistem penelitian ini. Berisi hasil penelitian dan dapat dilengkapi dengan tabel, grafik, atau gambar. Bagian pembahasan yang memamparkan dari hasil pengolahan data, interpretasi hasil penelitian yang diperoleh, dan mengaitkan dengan sumber rujukan yang relevan. Berikut adalah simbol – simbol yang ada pada diagram kelas di tabel 2.2 :

Tabel 2.2 Simbol dan Deskripsi dari Flowchart

SIMBOL	NAMA	FUNGSI
	Terminator	Permululaan / akhir program
	Garis alir (flow line)	Arah aliaran program
	Preparation	Proses ini sialisasi (pemberian harga awal)
	Proses	Proses Perhitungan (pengolahan data)
	Input / Output Data	Proses input/output data, parameter, informasi
	Predefined Proces (sub program)	Permulaan sub program/proses menjalankan sub program
	Decision	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	On Page Connector	Penghubung bagian" flowchart yang berada pada satu halaman
	Off Page Connector	Penghubung bagian" flowchart yang berada pada halaman berbeda

Sumber : (M. Shalahuddin, 2016)

2.7 Pengertian UML (Unified Modeling Language)

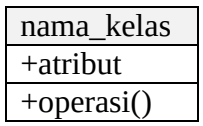
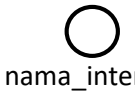
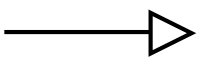
Menutut Rosa A.S. dan M. Shalahudin (2018) UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan dalam dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggembangkan arsitektur dalam pemograman berorientasi objek.

Adapun pengembangan diagram-diagram antara lain :

1. Class Diagram

   Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Berikut adalah simbol – simbol yang ada pada diagram kelas di tabel 2.2 :

Tabel 2.2 Simbol dan Deskripsi dari Diagram Kelas

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

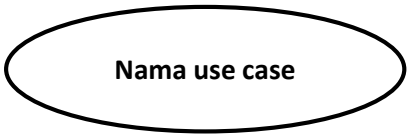
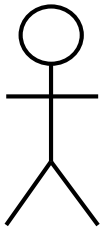
Kebergantungan / *dependency*	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi / *aggregation* 	Relasi antarkeals dengan makna semua-bagian (*whole-part*)

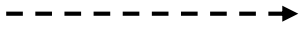
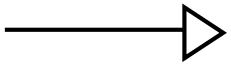
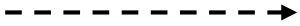
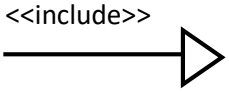
Sumber : (M. Shalahuddin, 2016)

2. Use Case Diagram

Use case atau diagram *use case* adalah pemodelan untuk kelakukan (*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 digunakan untuk membuat *use case* diagram dapat dilihat pada tabel 2.3 sebagai berikut :

Tabel 2.3 Simbol dan Deskripsi dari *Use Case Diagram*

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit – unit yang bertukar pesan antar unit atau aktor; biasanya digunakan mengunakan kata kerja di awal di awal fase nama *use case*
Aktor / *actor* 	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 merupan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / *association*	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau

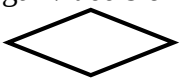
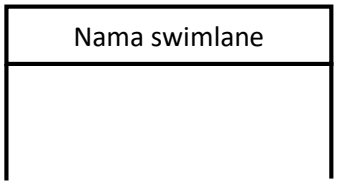
	use case memiliki interaksi dengan aktor
Ekstensi / *extend* <<extend>> 	Relasi *use case* tambahan ke sebuah use case dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambhakn misal
Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah *use case* diaman fungsi yang satu adalah fungsi yang leih umum dari lainnya
Menggunakan / *include / uses* <<uses>>  <<include>> 	Relasi *use case* tambahan ke sebuah *use case* di mana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini

Sumber : (M. Shalahuddin, 2016)

3. Activity Diagram

Diagram aktivitas atau acitity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Simbol – simbol yang digunakan untuk membuat diagram aktivitas dapat dilihat pada tabel 2.4 sebagai berikut :

Tabel 2.4 Simbol dan Deskripsi dari Diagram Aktivitas

Simbol	Deskripsi
Status awal 	Status awal aktibitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktiviatas biasanya diawali dengan kata kerja
Percabangan / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / *join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  Atau 	Memisahkan organiasai bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

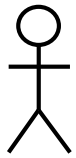
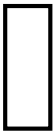
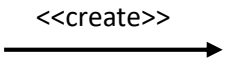
Sumber : (M. Shalahuddin, 2016)

4. Sequence diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirmkan dan

diterima antar objek. Simbol – simbol yang digunakan untuk membuat diagram sekuen dapat dilihat pada tabel 2.5 sebagai berikut :

Tabel 2.5 Simbol dan Deskripsi dari Diagram Sekuen

Simbol	Deskripsi
Aktor  **nama_aktor** Atau nama aktor Tanpa waktu aktir	Orang, proses atau sistem lain yang berinteraksi dengan sitem inormasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; diasanya dinyatakn menggunakan kata benda di awal frase nama aktor
Garis hidup / *lifeline* 	Menyatakn kehidupan suatu objek
Objek nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyaktan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
Pesan tipe create  	Menyatakn suatu objek membuata objek yang lain, arah panah mengarah pada objek yang dibuat

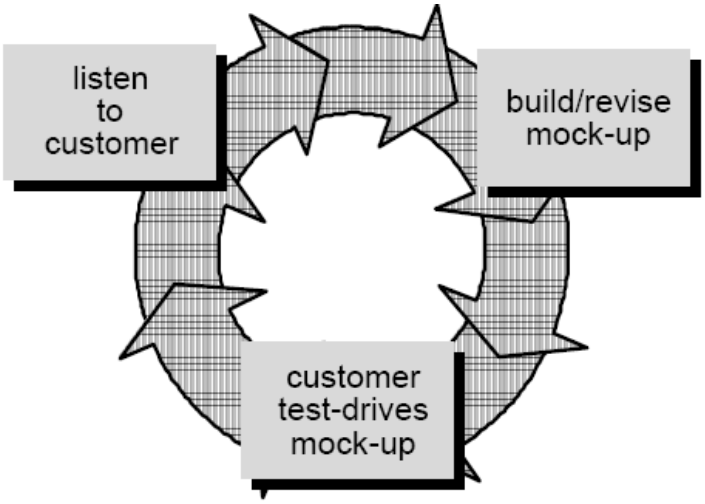
Pesan tipe call 1 : nama_metode()	Menyatakan suatu objel memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri
Pesan tipe send 1 : masukan	Menyatakan suatu objek mengirim suatu data / masukan / operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe return 1 : keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalain
Pesan tipe destroy <<destroy>>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy

Sumber : (M. Shalahuddin, 2016)

2.8 Metode Pengembangan Sistem

2.8.1 Metode Prototype

Menurut (Pressman, 2010), model prototype memiliki kemampuan dalam hal efisiensi algoritma, adaptasi terhadap sistem operasi ataupun bentuk-bentuk yang harus dilakukan oleh interaksi manusia dengan mesin menjadi pilihan. Dengan metode ini maka terjadi proses iteraktif dalam pengembangan sistem yang dibuat, dimana kebutuhan user diubah ke dalam sistem yang berkerja (*working system*) yang secara kontinue diperbaiki melalui kerjasama pengguna dengan analis.



Gambar 2.1 m*odel prototype* (Pressman, 2010)

Pada implementasinya, secara garis besar pengembangan dan perancangan sistem dilakukan menggunakan tahapan siklus pengembangan model prototype (Pressman, 2010) yaitu:

1. **Mendengarkan Pengguna**

   Tahap ini merupakan tahap pertama dalam merancang sistem. Pada tahap ini, seluruh kebutuhan-kebutuhan pengguna dirangkum menjadi informasi yang berguna untuk membangun sebuah aplikasi yang sesuai dengan tujuan dibuatnya aplikasi tersebut.

2. **Membangun Memperbaiki Prototype**

   Dalam tahap ini dilakukan perancangan dan pengkodean untuk sistem yang diusulkan yang mana tahapnya meliputi: perancangan proses-proses yang akan terjadi didalam sistem, perancangan diagram UML yang akan digunakan, perancangan antar muka keluaran dan dilakukan tahap pengkodean terhadap rancangan-rancangan yang telah didefinisikan, kelengkapan perangkat lunak dan perangkat keras.

3. Pengujian Prototype

Pada tahap ini dilakukan pengujian terhadap sistem yang telah disusun dan melakukan pengenalan terhadap sistem yang telah diujikan serta mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan.

2.8.2 ISO/IEC 25010

Standar ISO /IEC 25010 merupakan standar internasional yang diterbitkan oleh ISO/IEC untuk evaluasi kualitas sistem dan perangkat lunak yang merupakan perkembangan dari ISO 9126. Kualitas suatu sistem adalah sejauh mana sistem tersebut dapat memenuhi kebutuhan yang ditetapkan dari berbagai ukuran kualitas dan memberikan nilai. Model kualitas ISO 25010 mempunyai delapan ukuran kualitas yang ditetapkan oleh ISO/IEC 25010 yang dapat dilihat pada Gambar 2.2.

Gambar 2.2 Karakteristik Kualitas ISO/IEC 25010 (25010, 2023)

1. *Functionality Suitability*

Functionality Suitability merupakan suatu kumpulan atribut yang memuat adanya satu kumpulan fungsi dan spesifikasi dari properties-nya (25010, 2023). *Functionality* mencakup kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam

kondisi tertentu. Subkarakteristik *functionality suitability* meliputi *appropriateness, correctness, dan completeness*. Berikut ini penjelasan untuk masing-masing sub-karakteristik *functionality suitability* dapat di lihat pada tabel 2.6.

Tabel 2.6 Penjelasan Sub-Karakteristik *Functional Suitability*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Functional Suitability	*Appropriateness*	Sejauh mana fungsi memfasilitasi pencapaian tugas dan tujuan tertentu.
	Correctness	Sejauh mana produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang dibutuhkan.
	Completeness	Sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan.

Sumber : (25010, 2023)

2. *Reliability*

Reliability didefinisikan sejauh mana perangkat lunak dapat mempertahankan tingkat kinerja dalam kondisi tertentu. Subkarakteristik *Reliability* meliputi *availability, fault tolerance, recoverability, reliability compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *Reliability* yang dapat dilihat pada tabel 2.7.

Tabel 2.7 Penjelasan Sub-Karakteristik *Reliability*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Reliability	*Availability*	Sejauh mana komponen perangkat lunak beroperasi dan tersedia saat diperlukan untuk digunakan.
	Fault tolerance	Sejauh mana produk perangkat lunak dapat mempertahankan tingkat kinerja tertentu dalam kasus kesalahan perangkat lunak atau pelanggaran antarmuka yang ditentukan.
	Recoverability	Sejauh mana dalam hal gangguan atau kegagalan, produk atau sistem dapat memulihkan data yang terpengaruh secara langsung dan membangun kembali keadaan sistem yang diinginkan.
	Maturity	Sejauh mana sistem, produk atau perangkat lunak mematuhi keandalan dalam pengoperasian normal.

Sumber : (25010, 2023)

3. Performance efficiency

Performance efficiency merupakan sejauh mana perangkat lunak memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. Subkarakteristik *Performance efficiency* meliputi *time behaviour, resource utilization, capacity.* Berikut ini penjelasan untuk masing-masing subkarakteristik *Performance efficiency* yang dapat dilihat pada tabel 2.6.

Tabel 2.6 Penjelasan Subkarakteristik *Performance efficiency*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Performance efficiency	*Time behaviour*	Sejauh mana perangkat lunak memberikan respons yang tepat, waktu pemrosesan dan laju keluaran ketika menjalankan fungsinya.
	Resource utilization	Sejauh mana perangkat lunak menggunakan jumlah dan jenis sumber daya yang tepat ketika perangkat lunak menjalankan fungsinya.
	Capacity	Sejauh mana batas maksimum parameter produk atau sistem memenuhi persyaratan.

Sumber : (25010, 2023)

4. Usability

Usability merupakan sejauh mana produk perangkat lunak dapat dipahami, dipelajari, digunakan dan menarik bagi pengguna bila digunakan dalam kondisi tertentu. Subkarakteristik *Usability* meliputi *appropriateness recognisability, learnability, operability, user eror protection, user interface aesthetics, accessibility*. Berikut ini penjelasan untuk masing-masing subkarakteristik *Usability* yang dapat dilihat pada tabel 2.7.

Tabel 2.7 Penjelasan Sub-Karakteristik *Usability*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Usability	*Appropriateness Recognisability*	Sejauh mana perangkat lunak memeungkinkan pengguna untuk mengenali apakah perangkat lunak sesuai dengan kebutuhan pengguna.
	Learnability	Sejauh mana perangkat lunak pengguna untuk mempelajari aplikasinya.
	Operability	Sejauh mana perangkat lunak atau sistem memudahkan pengguna mengoperasikan dan mengendalikan.
	User error protection	Sejauh mana sistem melindungi pengguna dari membuat kesalahan.
	User interface aesthetics	Sejauh mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.

	Accessibility	Sejauh mana suatu produk atau sistem dapat digunakan oleh orang-orang dengan berbagai karakteristik dan kemampuan untuk mencapai tujuan tertentu dalam konteks penggunaan tertentu.

Sumber : (25010, 2023)

5. Security

Security merupakan perlindungan item sistem dari akses yang tidak disengaja atau berbahaya, penggunaan, modifikasi, perusakan dan pengungkapan. Subkarakteristik *Security* meliputi *confidentiality, integrity, non-repudiation, accountability, authenticity, security compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *security* yang dapat dilihat pada tabel 2.8.

Tabel 2.8 Penjelasan Sub-Karakteristik *Security*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Security	*Confidentiality*	Sejauh mana perangkat lunak memberikan perlindungan dari pengungkapan data atau informasi yang tidak sah, baik disengaja atau disengaja.
	Integrity	Sejauh mana ketepatan dan kelengkapan aset dijaga.

	Non-repudiation	Sejauh mana tindakan atau peristiwa dapat dibuktikan telah terjadi, sehingga peristiwa atau tindakan tidak dapat ditolak.
	Accountability	Sejauh mana tindakan suatu entitas dapat dilacak secara unik kepada entitas.
	Authenticity	Sejauh mana identitas suatu subjek atau sumber daya dapat dibuktikan sebagai yang diklaim.

Sumber : (25010, 2023)

6. Compatibility

Compatibility merupakan kemampuan dua atau lebih komponen perangkat lunak untuk bertukar informasi dan untuk melakukan fungsi yang diperlukan saat berbagi perangkat keras atau perangkat lunak yang sama. Subkarakteristik *compatibility* meliputi *replaceability, co-existence, interUsability, compatibility compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *compatibility* yang dapat dilihat pada tabel 2.9.

Tabel 2.9 Penjelasan Sub-Karakteristik *Compatibility*

Karakteristik	Subkarakteristik	Penjelasan
Compatibility	*Co-existence*	Sejauh mana perangkat lunak dapat bekerja sama dengan perangkat lunak independen lainnya dalam lingkungan umum berbagi sumber daya umum tanpa ada dampak yang merugikan.
	InterUsability	Sejauh mana perangkat lunak dapat dioperasikan secara kooperatif dengan satu atau lebih perangkat lunak lainnya.

Sumber : (25010, 2023)

7. Maintainability

Maintainability merupakan sejauh mana perangkat lunak dapat dimodifikasi. Modifikasi dapat mencakup koreksi, peningkatan atau adaptasi perangkat lunak terhadap perubahan lingkungan, dan persyaratan serta spesifikasi fungsional. Subkarakteristik *Maintainability* meliputi *modularity*, *reusability, analyzability, changeability, modification stability, testability, maintainability compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *maintainability* yang dapat dilihat pada tabel 2.10.

Tabel 2.10 Penjelasan Sub-Karakteristik *Maintainability*

Karakteristik	Subkarakteristik	Penjelasan
Maintainability	*Modularity*	Sejauh mana suatu sistem atau program komputer terdiri dari komponen-

		komponen terpisah sedemikian rupa sehingga perubahan pada satu komponen memiliki dampak minimal pada komponen lainnya.
	Reusability	Sejauh mana aset dapat digunakan lebih dari satu sistem perangkat lunak, atau dalam membangun aset lainnya.
	Analyzability	Tingkat dimana perangkat lunak dapat didiagnosis untuk kekurangan atau penyebab kegagalan dalam perangkat
	Modifiability	Sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menimbulkan cacat atau menurunkan kualitas produk yang ada.
	Testability	Tingkat efektivitas dan efisiensi dengan kriteria pengujian yang dapat ditetapkan untuk sistem, produk atau komponen dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.

Sumber : (25010, 2023)

8. Portability

Merupakan sejauh mana perangkat lunak dapat ditransfer dari satu lingkungan ke lingkungan lain. Subkarakteristik *portability* meliputi *adaptability, installability, replaceability*. Berikut ini penjelasan untuk masing-masing subkarakteristik *transferability* yang dapat dilihat pada tabel 2.11.

Tabel 2.11 Penjelasan Sub-Karakteristik *Transferability*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Transferability	*Adaptability*	Sejauh mana suatu produk atau sistem dapat secara efektif dan efisien diadaptasi untuk perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lainnya yang berbeda atau berkembang.
	Installability	Tingkat efektivitas dan efisiensi dimana produk atau sistem dapat berhasil dipasang dan/atau dicopot di lingkungan tertentu.
	Replaceability	mana suatu produk dapat menggantikan produk perangkat lunak lain yang ditentukan untuk tujuan yang sama di lingkungan yang sama.

Sumber : (25010, 2023)

Pada penelitian ini pengujian berfokus pada dua karakteritik yaitu *Functional Suitability* dan *Usability*.

2.8.3 Skala Pengukuran

Skala pengukuran merupakan kesepakatan yang digunakan sebagai acuan untuk menentukan panjang pendeknya interval yang ada dalam alat ukur, sehingga alat ukur tersebut bila digunakan dalam pengukuran akan menghasilkan data kuantitatif (Sugiyono, 2013)

2.8.4 Skala Likert

Skala likert menurut (Sugiyono, 2013) merupakan skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial skala likert dapat digunakan untuk mengukur sikap seseorang dengan menyatakan setuju atau tidak setuju terhadap subjek, objek atau kejadian tertentu. Dengan menggunakan skala likert, variabel dijabarkan menurut urutan variabel, sub variabel, indikator, dan deskriptor. Deskriptor kemudian dijadikan titik tolak untuk membuat butir instrumen berupa pernyataan atau pertanyaan yang perlu dijawab oleh responden. Item-item dalam skala likert menyediakan respon dengan kategori yang berjenjang, dan biasanya memiliki jenjang lima, yaitu: sangat setuju, setuju, ragu-ragu, tidak setuju, dan sangat tidak setuju. Setiap kategori tersebut diberi nilai atau skor. Pernyataan pada skala likert terdiri dari pernyataan positif dan pernyataan negatif. Contoh lima jenjang dalam skala likert dapat dilihat pada tabel berikut.

Tabel 2.8 Jenjang Dalam Skala Likert

Pernyataan positif	Nilai	Pernyataan Negatif	Nilai
Sangat setuju	5	Sangat setuju	1
Setuju	4	Setuju	2
Ragu-ragu (Netral)	3	Ragu-ragu (Netral)	3
Tidak Setuju	2	Tidak Setuju	4
Sangat tidak setuju	1	Sangat tidak setuju	5

Sumber : (Sugiyono, 2013)

Rumus perhitungan skala Likert adalah sebagai berikut (Sugiyono, 2013):

$$\%\ Skor\ Aktual = \frac{Skor\ Aktual}{Skor\ Ideal} \times 100\% \qquad \text{.....(4.68)}$$

Keterangan :

1. Skor aktual adalah hasil jawaban seluruh responden atas kuisoer yang telah diajukan.
2. Skor ideal adalah nilai tertinggi atau semua responden diasumsukan memilih jawaban dengan skor tertinggi.

Kemudian hasil perhitungan yang didapatkan dari angket, selanjutnya dibandingkan dengan rentang kriteria interpretasi skor untuk menyatakan hasil yang didapatkan dengan rentang sebagai berikut.

Tabel 2.9 Rentang Kriteria Interpretasi

No	Rentang Kriteria	Kriteria
1	0% - 20%	Sangat Tidak Baik
2	21% - 40%	Tidak Baik
3	41% - 60%	Kurang Baik
4	61% - 80%	Baik
5	81% - 100%	Sangat Baik

Sumber : (Sugiyono, 2013)