



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Android

Menurut Karman dan Mulyono (2020:1920), “Android adalah sistem operasi perangkat lunak berbasis linux yang mencakup sistem operasi, middleware dan aplikasi”.

Menurut Yusmiarti (2020:3), “Android adalah sebuah sistem operasi untuk *mobile* yang mencakup sistem operasi dan aplikasi”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa Android adalah sistem operasi *mobile* yang mencakup sistem operasi, *middleware* dan aplikasi.

2.1.2 Metode Location Based Service

Menurut Rosyd sikumbang dan Habibi (2020:60), “*Location Based Service* adalah layanan informasi yang dapat diakses melalui mobile device dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut”.

Menurut Susanty dan Astari (2019:54), “*Location based service* adalah layanan informasi yang di akses menggunakan piranti mobile melalui jaringan internet dan seluler serta memanfaatkan kemampuan penunjuk lokasi pada piranti *mobile*”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Location Based Service* adalah sistem layanan informasi berupa informasi yang dapat diakses menggunakan telepon seluler untuk menemukan lokasi perangkat secara tepat.

2.2 Teori Judul

2.2.1 Pengertian Implementasi

Menurut Dewi (2019:24) ”Implementasi berkenaan dengan berbagai kegiatan yang diarahkan pada realisasi program”.



Menurut Annur dan Suhono (2019:70), “Implementasi adalah bukan sekedar aktivitas, tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh- sungguh berdasarkan acuan norma tertentu untuk mencapai tujuan kegiatan”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa Implementasi adalah tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang.

2.2.2 Pengertian Push Notification

Menurut Siddik dan Nasution (2018:2), “Push Notification adalah sebuah layanan yang banyak digunakan untuk keperluan pemberitahuan melalui pesan pendek yang ada di smartphone”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Push Notification* adalah pesan pendek yang muncul secara otomatis dilayar *smartphone*.

2.2.3 Pengertian Penempatan

Menurut Suparman (2020:5), “Penempatan merupakan proses pemberian tugas dan pekerjaan kepada tenaga kerja yang lulus seleksi untuk dilaksanakan sesuai ruang lingkup yang telah ditetapkan, serta mampu mempertanggung jawabkan segala resiko dan kemungkinan-kemungkinan yang terjadi atas tugas dan pekerjaan, wewenang, serta tanggung jawab”.

Menurut Kamus Besar Bahasa Indonesia Daring penempatan merupakan proses, cara, pembuatan menepati atau menempatkan.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa Penempatan adalah proses pemberian tugas kepada tenaga kerja yang telah ditetapkan sesuai dengan tempat/ruang lingkup yang telah ditetapkan.



2.2.4 Implementasi Push Notification Pada Penempatan PAM (Pengamanan) kantor Satpol PP Provinsi Sumatera Selatan Menggunakan Metode Location Based Service (LBS) Berbasis Android

Implementasi Push Notification Pada Penempatan PAM (Pengamanan) kantor Satpol PP Provinsi Sumatera Selatan Menggunakan Metode *Location Based Service* (LBS) Berbasis Android adalah sebuah aplikasi yang akan dibuat untuk memberikan kemudahan untuk para pegawai dalam mendapatkan tugas melalui pesan pendek yang muncul secara otomatis di layar *smartphone* para pegawai sehingga para pegawai terbantu dan dapat efisien dalam menjalankan tugas.

2.3 Teori khusus

2.3.1 Pengertian UML

Menurut Dwi dan Taufiq (2020:29), "*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".

Menurut Waluyo dan Nurul Fatich (2017:188), "UML (Unified Modelling Language) adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerja bersama Grady Booch, James Rumbaugh dan Ivar Jacobson".

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah suatu model yang dapat memberikan bahasa pemodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.

2.3.2 Pengertian Use Case Diagram

Menurut Mukhtar (2019:83), "*Use case Diagram* atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat".

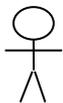
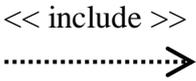
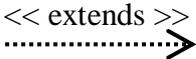


Menurut Wiharko dan Setiawan (2018:38), “*Use Case Diagram* merupakan deskripsi fungsi – fungsi di dalam sebuah sistem dari perspektif pengguna”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Use Case Diagram* adalah Teknik untuk merekam persyaratan fungsional sebuah sistem.

Adapun simbol-simbol dari *Use Case Diagram*.

Tabel 2.1 Simbol-simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		Actor	Seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun
2.		Use case	Menggambarkan bagaimana seseorang menggunakan sistem
3.		Relasi asosiasi	Relasi yang dipakai untuk menunjukkan hubungan antara aktor dan use case
4.		Relasi include	Memungkinkan satu use case menggunakan fungsionalitas yang disediakan oleh use case lainnya
5.		Relasi extend	Memungkinkan suatu use case secara optional menggunakan fungsionalitas yang disediakan oleh use case lainnya

(Sumber : Rusmawan, 2019:72-73)



2.3.3 Pengertian Activity Diagram

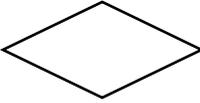
Menurut Mukhtar (2019:84), “Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Menurut Syahputra (2019:38), “ Activity diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sisten atau menu yang ada pada perangkat lunak”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Activity Diagram* merupakan gambaran rangkaian aktivitas yang terjadi di dalam suatu sistem yang akan di rancang.

Adapun simbol-simbol yang sering digunakan dalam *Activity Diagram* dapat dilihat pada tabel berikut ini:

Tabel 2.2 Simbol-simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		Start state	Titik Awal atau permulaan
2.		End state	Titik akhir atau akhir dari aktivitas
3.		Activity	<i>Activity</i> atau aktivitas yang dilakukan oleh actor
4.		Decision	Pilihan untuk mengambil keputusan
5.		Interaction	Alur

(Sumber : Rusmawan, 2019:80)



2.3.4 Pengertian Class Diagram

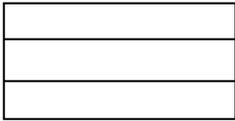
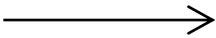
Menurut Wiharko dan Heru Setiawan (2018:38), “*Class Diagram* adalah pandangan aplikasi yang bersifat statis. *Class Diagram* tidak hanya menggambarkan visualisasi, tetapi juga menggambarkan dan mendokumentasikan aspek yang berbeda dalam sistem, tetapi juga untuk konstruksi eksekusi kode dalam software aplikasi. *Class Diagram* digunakan untuk mengelompokan hal-hal inti dari setiap proses yang ingin dilakukan. Semua proses dimasukkan ke dalam tiap-tiap class dan saling dihubungkan pada class-class lainnya yang saling berhubungan”.

Menurut Syahputra (2019:42), “Class Diagram membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Class Diagram* merupakan visual dari struktur sistem program pada jenis-jenis yang dibentuk.

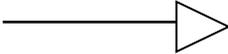
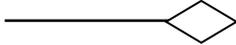
Berikut adalah simbol-simbol yang digunakan pada *Class Diagram*

Tabel 2.3 Simbol-simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi/ <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah/ directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>



Lanjutan Tabel 2.3 Simbol-simbol *Class Diagram*

No.	Simbol	Deskripsi
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Kebergantungan/ dependency 	Relasi antar kelas dengan makna kebergantungan antarmuka
7.	Agregasi/aggregation 	Relasi antar kelas dengan makna semua bagian (whole-part)

(Sumber : Mukhtar, 2019:85)

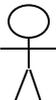
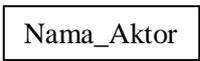
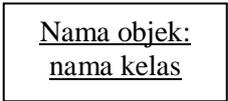
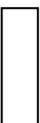
2.3.5 Pengertian Sequence Diagram

Menurut Bayumanis dan Fitriannah (2017:6), “Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario”.

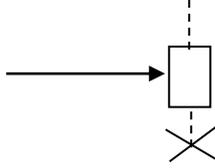
Menurut Mukhtar (2019:84), “*Diagram sequence* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa Sequence Diagram merupakan perilaku antar objek yang ada di dalam sistem dan disusun pada rangkaian interksi antar objek termasuk penggunaan display dan pesan yang dikirim/diterima objek.

Tabel 2.4 Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	Aktor  Atau  Tanpa waktu aktif	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu berupa orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor
2.	Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek
3.	Objek 	Menyatakan objek yang berinteraksi pesan
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya
5.	Pesan tipe <i>create</i>  <<create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7.	Pesan tipe <i>send</i> 1 : masukan  	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe <i>return</i> 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 kembalian

Lanjutan Tabel 2.4 Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
9.	Pesan tipe <i>destroy</i> <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

(Sumber : Mukhtar, 2019:84-85)

2.4 Teori Program

2.4.1 Pengertian Java

Menurut Wahyu Andrian (2019:269), “Java adalah bahasa pemrograman yang berkembang yang berorientas objek. Teknologi *object oriented* memandang software sebagai sebuah interaksi antarbagian dalam sebuah sistem dan menggambarkan bagian tersebut ke dalam satu objek yang memiliki sifat/*property*/data dan kemampuan untuk melakukan suatu tugas tertentu”.

Menurut Wikipedia “Java adalah bahasa pemograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *sun microsystem* saat ini merupakan bagian dari *oracle* dan dirilis tahun 1995”.

(<https://id.wikipedia.org/wiki/Java>)

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa Java merupakan bahasa pemograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam.

2.4.2 Pengertian Android Studio

Menurut Juansya (2015:2-3), “Android Studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada *event Google I/O Conference* untuk tahun 2013.



Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android”.

Menurut Nasution dan Kamil (2019:56), “Android Studio merupakan pengembangan terpadu *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa Android Studio adalah sebuah IDE untuk pengembangan aplikasi android yang bersifat open source (gratis).

2.4.3 Pengertian XML

Menurut Wahyudi (2019:11), “XML adalah singkatan dari *eXtensible Markup Language*. Bahasa markup adalah sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan tag. XML terletak pada inti *Web Service* yang digunakan untuk mendeskripsikan data”.

Menurut Wikipedia, “XML (*Extentsible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan Bahasa standar untuk melacak internet”.

(<https://id.m.wikipedia.org/wiki/XML>)

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa XML merupakan bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup, keperluan pertukaran data antar system yang beraneka ragam.