

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Tinjauan Studi berisikan jurnal yang berhubungan dengan penelitian untuk dijadikan sebagai referensi sekaligus media bertukar informasi dengan fakta yang ada :

Menurut Daniel Prastiwa dan Wiji Setyaningsih dalam penelitiannya yang berjudul Aplikasi Reservasi Persewaan Lapangan Futsal Pada Champion Futsal Menggunakan GPS dengan Metode Navigasi *Waypoint* Berbasis Android. Penelitian ini dilatar belakangi pada proses pelayanan yang masih manual, petugas masih menggunakan buku untuk manajemen jadwal lapangan. Proses reservasi juga merepotkan petugas karena sering memesan secara mendadak. Oleh karena itu diperlukan suatu aplikasi yang dapat memudahkan pelanggan dan petugas pada saat reservasi. Aplikasi ini menggunakan *Global Positioning System* (GPS) dengan metode Navigasi *Waypoint*. Metode Navigasi *Waypoint* akan mendeteksi koordinat lokasi posisi pelanggan dengan melakukan kalkulasi jarak terhadap koordinat lokasi cabang yang telah tersimpan. Aplikasi ini menggunakan pemodelan *Unified Modelling Language* (UML) yang terdiri dari *Usecase* Diagram, *Class* Diagram, *Sequence* Diagram, dan *Activity* Diagram. Pembuatan user interface aplikasi ini menggunakan pemrograman Android dengan *tools Android Developer Tools*. Hasil dari penelitian ini menghasilkan sebuah aplikasi reservasi lapangan futsal menggunakan GPS dengan metode Navigasi *Waypoint* berbasis android yang dapat memudahkan pelanggan dalam pemesanan lapangan futsal dan dapat menentukan jarak cabang Champions Futsal yang terdekat dari posisi *device android*. [1]

Menurut Aji Agustian, Samirah Rahayu, dan Lani Nurlani dalam penelitiannya yang berjudul Aplikasi E-Futsal dengan Metode Mobile-GIS dan GPS Berbasis Android. Futsal kini menjadi salah satu cabang olahraga yang banyak diminati oleh berbagai kalangan usia. Dengan banyaknya peminat futsal, banyak juga usaha penyewaan lapangan futsal. Akan tetapi dalam penelitian ini,

masalah yang terjadi adalah kurangnya sistem informasi tempat penyewaan dan pemesanan lapangan futsal. Pelayanan juga masih dilakukan secara manual, khususnya dalam melakukan manajemen jadwal lapangan dan proses pemesanan cukup merepotkan karena dilakukan secara langsung dengan datang ke lokasi. Oleh karena itu dibutuhkan sistem informasi untuk memudahkan masyarakat yang ingin memesan lapangan futsal dan pengelola penyewaan lapangan futsal dalam mengelola jadwal pemesanan. Pada aplikasi ini memanfaatkan teknologi *Global Positioning System* (GPS) kemudian diintegrasikan ke dalam perangkat *mobile* berbasis android dengan metode *Geographic Information System* (GIS). Desain sistem pada aplikasi ini diimplementasikan menggunakan pemodelan *Unified Modelling Language* (UML) yang terdiri dari *use case* diagram, *class* diagram, *sequence* diagram, *activity* diagram, dan *deployment* diagram. Hasil dari penelitian ini adalah sebuah aplikasi E-Futsal berbasis android dengan metode GIS dan GPS yang dapat memudahkan pengguna dalam mencari lokasi lapangan futsal dan mempermudah memesan lapangan futsal. [2]

Menurut Robby Rachmatullah dan Bintang Pradana Kusuma dalam penelitiannya yang berjudul Aplikasi Smart Map Apotek Surakarta Berbasis Android. Penelitian ini dilatar belakangi masalah sulitnya mencari letak dan detail apotek yang ada di Surakarta. Jadwal buka atau tutup apotek juga menjadi kendala, banyak membuat pasien kecewa karena tidak ada pemberitahuan lebih dulu ketika akan libur dan tutup. Maka dari itu perlu adanya informasi detail dari apotek yang berisikan informasi-informasi akurat tentang apotek dan jalur rute menuju lokasi apotek tersebut. Aplikasi ini menggunakan metode pengembangan sistem model *waterfall*, yang terdiri dari analisis sistem, desain sistem, *coding*, pengujian dengan *blackbox testing*, yang terakhir implemetasi dan pemeliharaan sistem. Pemodelan aplikasi ini menggunakan model *Unified Modelling Language* (UML) yang terdiri dari *use case* diagram, *class* diagram, *activity* diagram, dan *sequence* diagram. Hasil dari penelitian ini adalah sebuah aplikasi Smart Map Apotek Surakarta berbasis android yang dapat memberi informasi apotek secara detail, navigasi transportasi, dan rute perjalanan. Dengan Smart Map dapat mempermudah pencarian lokasi apotek di Surakarta secara detail. [3]

2.2 Tinjauan Pustaka

2.2.1 Android Studio

Android Studio adalah sebuah lingkungan pengembangan terpadu (IDE) untuk mengembangkan pada platform Android. Hal itu disampaikan pada tanggal 16 Mei 2013 di Google I/O konferensi dengan Product Manager Google, Katherine Chou. Android Studio tersedia secara bebas di bawah Lisensi Apache 2.0. Android Studio berada di awal tahap preview akses mulai dari versi 0.1 Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada bulan Juni 2014. Yang pertama membangun stabil dirilis pada bulan Desember 2014, mulai dari versi 1.0. Berdasarkan software IDEA JetBrains ' IntelliJ, Android Studio dirancang khusus untuk pengembangan Android. Ini tersedia untuk di download pada Windows, Mac OS X dan Linux, dan diganti Eclipse Pengembangan Android Tools (ADT) sebagai IDE utama Google untuk pengembangan aplikasi Android asli. [4]

2.2.2 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi. Awalnya Google Inc. Membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. [5]

Tabel 2.1. Daftar Versi *Android*. [6]

(Sumber : Buku Menjadi Android Developer Expert, Ahmad Imaddudin dan Sidiq Permana, 2017)

| Versi | Code Name | Tanggal Rilis | API Level | DVM/ART | Fitur Baru |
|-------|-----------|---------------|-----------|---------|--|
| 1.5 | Cupcake | 27 April 2009 | 3 | - | - |
| 1.6 | Donut | 15 Septemb | 4 | - | <ul style="list-style-type: none"> Quick Search Box |

| Versi | Code Name | Tanggal Rilis | API Level | DVM/ART | Fitur Baru |
|-------------|--------------------|------------------|-----------|---------|---|
| | | er 2009 | | | <ul style="list-style-type: none"> • Screen Size Diversity • Android Market |
| 2.0 - 2.1 | Eclair | 26 Oktober 2009 | 5 | - | <ul style="list-style-type: none"> • Google Maps Navigation • Home Screen Customization • Speech-to-Text |
| 2.2 – 2.2.3 | Froyo | 20 Mei 2010 | 8 | DVM | <ul style="list-style-type: none"> • Voice Action • Portable Hotspot • Dalvik JIT |
| 2.3 – 2.3.7 | Gingerbread | 9 Februari 2011 | 9 – 10 | DVM | <ul style="list-style-type: none"> • Gaming APIs • NFC • Battery Management |
| 3.0 – 3.2.6 | HoneyComb | 22 Februari 2011 | 11 -13 | DVM | <ul style="list-style-type: none"> • Tablet-Friendly Design • System Bar • Quick Settings |
| 4.0 – 4.6 | Ice Cream Sandwich | 19 Oktober 2011 | 15 | DVM | <ul style="list-style-type: none"> • Custom Home Screen • Data Usage Control • Android Beam |
| 4.1 – 4.3.1 | Jelly Bean | 9 Juli 2012 | 18 | DVM | <ul style="list-style-type: none"> • Google Now • Actionable Notifications • Account Switching |

| Versi | Code Name | Tanggal Rilis | API Level | DVM/ART | Fitur Baru |
|-------------|-------------|------------------|-----------|---------------------|--|
| 4.4 – 4.4.4 | Kitkat | 31 Oktober 2013 | 19 -20 | DVM (and ART 1.6.0) | <ul style="list-style-type: none"> • Voice : Ok Google • Immersive Design • Smart Dialer |
| 5.1 – 5.1.1 | Lollipop | 12 November 2014 | 21 -22 | ART | <ul style="list-style-type: none"> • Material Design • Multiscreen • Notifications |
| 6.0 – 6.0.1 | Marshmallow | 5 Oktober 2015 | 23 | ART | <ul style="list-style-type: none"> • Now On Tap • Permissions • Battery (Doze & App Standby) |
| 7.1 – 7.1.2 | Nougat | 22 Agustus 2016 | 24 – 25 | ART | <ul style="list-style-type: none"> • Multi window • GIF Keyboard |
| 8.0 – 8.1 | Oreo | 25 Oktober 2017 | 26 – 27 | ART | <ul style="list-style-type: none"> • Picture-in-Picture |
| 9 | Pie | 6 Agustus 2018 | 28 | ART | <ul style="list-style-type: none"> • Adaptive Battery • Adaptive Brightness |
| 10 | Ten | Mei 2019 | 29 | ART | <ul style="list-style-type: none"> • Privacy Feature • New API for Gesture Navigation • Dark Mode |

2.2.3 Dart

Dart merupakan bahasa pemrograman yang diproduksi oleh Google, dirancang oleh Lars Bak dan Kasper Lund. Diperkenalkan pertama kali pada 10 Oktober 2011 pada versi 1.0 yang masih digunakan oleh tim programmer internal Google dan baru dirilis ke publik pada November 2013. Dapat digunakan untuk membuat aplikasi server, web, maupun android atau ios (mobile). [7]

2.2.4 Firebase

Firebase adalah Backend as a Service (BaaS) yang saat ini dimiliki oleh Google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah pengembangan aplikasi *mobile*. Dua fitur menarik dari Firebase adalah *Firebase Remote Config* dan *Firebase Real Time Database*. Selain itu juga terdapat fitur pendukung untuk aplikasi yang memerlukan *push notification* yaitu *Firebase Notification Console*. Firebase Database merupakan penyimpanan basis data nonSQL yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data itu antara lain *String*, *Long*, dan *Boolean*. Data pada Firebase Database disimpan sebagai objek JSON *tree*. Tidak seperti basis data SQL, tidak ada tabel dan baris pada basis data non-SQL. Ketika ada penambahan data, data tersebut akan menjadi node pada struktur JSON. *Node* merupakan simpul yang berisi data dan bisa memiliki cabang-cabang berupa node lainnya yang berisi data pula. Proses pengisian suatu data ke Firebase Database dikenal dengan istilah *push*. [8]

Tabel 2.2. Daftar Produk Layanan *Firebase* [9]

(Sumber : Buku Membangun Ojek Online Menggunakan *Firebase*, Nando Septian Husni dan Muhamad Alam Syah, 2019)

| No. | Jenis Layanan | Produk Layanan |
|-----|-----------------------|---|
| 1 | Pengembangan Aplikasi | <ul style="list-style-type: none"> • <i>Firebase Cloud Firestore</i> • <i>Realtime Database</i> • <i>Storage</i> • <i>ML Kit</i> • <i>Cloud Function</i> |

| No. | Jenis Layanan | Produk Layanan |
|-----|-------------------------------|---|
| | | <ul style="list-style-type: none"> • <i>Authentication</i> • <i>Hosting</i> |
| 2 | Peningkatan Kualitas Aplikasi | <ul style="list-style-type: none"> • <i>Crash Reporting</i> • <i>Performance Monitor</i> • <i>Test Lab</i> |
| 3 | Pengembangan Bisnis | <ul style="list-style-type: none"> • <i>Analytics</i> • <i>Predictions</i> • <i>A/B Testing</i> • <i>Cloud Messaging</i> • <i>Remote Config</i> • <i>Dynamic Links</i> • <i>App Indexing</i> • <i>Invites</i> |

2.2.5 Navigasi Waypoint

Navigasi adalah penentuan kedudukan (*position*) dan arah perjalanan baik di medan sebenarnya atau di peta, dan oleh sebab itulah pengetahuan tentang pedoman arah (*compass*) dan peta serta teknik penggunaannya haruslah dimiliki dan dipahami. *Waypoint* adalah titik acuan/kumpulan koordinat yang digunakan untuk keperluan navigasi untuk mengidentifikasi sebuah titik di peta. Koordinat-koordinat itu biasanya menyertakan *longitude*, *latitude*, dan kadang *altitude* untuk keperluan navigasi di udara. *Waypoint* digunakan di berbagai navigasi yang tidak memiliki jalur yang tampak seperti navigasi di udara dan navigasi di laut.

Navigasi *Waypoint* adalah suatu metode untuk mengatur gerak dari suatu posisi ke posisi lain yang dituju, dengan mengasumsikan setiap posisi dalam proses pergerakannya menjadi suatu titik dalam sistem koordinat tertentu (*latitude* dan *longitude* bila berdasarkan sistem koordinat Bumi). [10]

2.2.6 JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan

dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. [11]

2.2.7 Flutter

Flutter merupakan Software Development Kit (SDK) dari Google yang digunakan untuk membuat aplikasi mobile untuk android dan IOS hanya dengan satu kode yaitu menggunakan bahasa pemrograman Dart. Flutter ditujukan untuk mempermudah dan mempercepat proses pengembangan aplikasi berbasis mobile. Flutter dirilis pada bulan Mei 2017. [7]

2.2.8 Google Map

Google Maps adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh Google dapat ditemukan di <http://maps.google.com>. Google Maps menyediakan gambar resolusi tinggi satelit untuk daerah perkotaan sebagian besar di Amerika Serikat (termasuk Hawaii, Alaska, PuertoRico, dan US Virgin Islands), Kanada, dan Inggris, serta sebagai bagian dari Australia dan banyak negara lainnya. Seperti banyak aplikasi web lainnya Google, Google Maps menggunakan JavaScript secara ekstensif. Beberapa tujuan dari penggunaan Google Maps adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya. Hampir semua hal yang berhubungan dengan peta dapat memanfaatkan Google Maps. [12]

2.2.9 Location Base Service (LBS)

Location Based Service (LBS) merupakan layanan yang mengidentifikasi lokasi atau objek, seperti keberadaan seseorang, lokasi bank, rumah sakit atau sekolah terdekat. Dengan teknologi LBS (*Location Based Service*) yang dapat digunakan untuk mengetahui posisi berdasarkan titik geografis dari lokasi pengguna dan lokasi lain yang diinginkan. Selain itu LBS ini dapat diaplikasikan untuk SIG (Sistem Informasi Geografis). Perbedaan LBS dan

SIG adalah LBS hanya akan mendapatkan titik geografis dari lokasi saja, sedangkan dengan SIG didapatkan informasi keruangan secara geografis dengan bantuan LBS sebagai layanan untuk mendapatkan titik geografisnya. Untuk mengaplikasikan SIG dengan LBS paling tidak harus mengetahui titik koordinat lokasi yang diinginkan dan database atau array untuk menyimpan titik koordinat dan informasi lain yang berguna mengenai lokasi tersebut. [13]

2.2.10 Unified Modeling Language

Unified Modeling Language merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML digunakan sebagai salah satu alat untuk melakukan perancangan atau memodelkan sistem. UML sering digunakan karena penggunaannya yang tidak terpengaruh pada perangkat lunak, perangkat keras, sistem operasi, jaringan, basis data dan bahasa pemrograman yang digunakan. UML terdiri dari tiga kategori yaitu [14]:





1. Struktur Diagram, terdiri dari *Class* diagram, *Object* diagram, *Component* diagram, *Deployment* diagram, *Composite structure* diagram, *Package* diagram.
2. *Behavior* Diagram, terdiri dari *Use case* diagram, *Activity* diagram, *State Machine* diagram (*State chart* diagram in version 1.x).
3. *Interaction* diagram, terdiri dari *Communication* diagram, *Interaction Overview* diagram, *Sequence* diagram, *Timing* diagram.

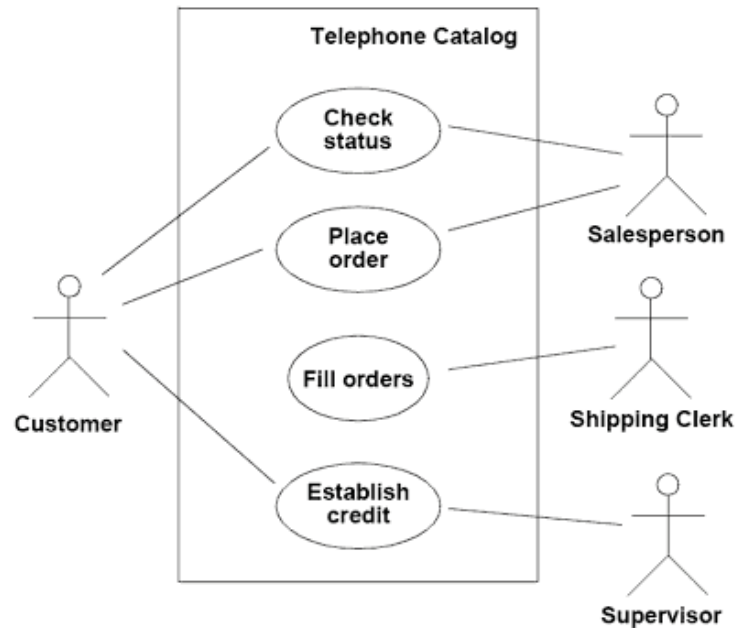
2.2.11 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. [15]

Tabel 2.3. Simbol yang Digunakan pada *Use Case Diagram*

(Sumber : Jurnal Pemodelan Uml Sistem Informasi Monitoring Penjualan dan Stok Barang, Ade Hendini, 2016)

| No. | Simbol | Nama | Keterangan |
|-----|---|-----------------------|--|
| 1 |  | <i>Use Case</i> | Deskripsi dari urutan aksi yang ditampilkan sistem. |
| 2 |  | <i>Actor</i> | Pengguna sistem yang berhubungan dengan sistem lain atau waktu. |
| 3 |  | <i>Include</i> | Memungkinkan satu <i>use case</i> menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lain secara bersamaan dengan <i>use case</i> utama. |
| 4 |  | <i>Generalization</i> | Hubungan pada objek anak berbagi perilaku dan struktur data dari objek induk. |
| 5 |  | <i>Extend</i> | Memungkinkan satu <i>use case</i> secara optimal menggunakan fungsilitas dari <i>use case</i> yang lain setelah <i>use case</i> utama dijalankan. |
| 6 |  | <i>Collaboration</i> | Interaksi elemen lain yang berkerja sama untuk menyediakan perilaku yang besar dari jumlah yang ada. |
| 7 |  | <i>Association</i> | Menghubungkan antara objek satu ke objek yang lainnya. |



Gambar 2.1. Contoh *Use Case Diagram*

(Sumber : Buku Metode Analisis dan Perancangan Sistem, Prof. Dr. Sri Mulyani, Ak., CA., 2016)




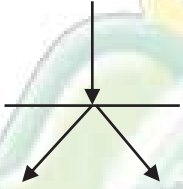

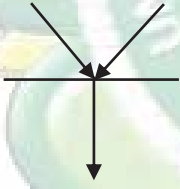

Contoh *Use Case Diagram* tersebut merupakan deskripsi fungsi dari sistem katalog telepon. Digambarkan dalam bentuk orang atau disebut *actor* mewakili dari *customer*, *salesperson*, *shipping clerk*, dan *supervisor*. Setiap *actor* dapat berinteraksi dengan fungsi atau fitur dari sistem tersebut dan dapat berinteraksi dengan lebih dari satu fungsi atau fitur yang terdapat di sistem. Contoh pada *actor* customer dapat mengakses *check status*, *place order*, dan *establish credit*.

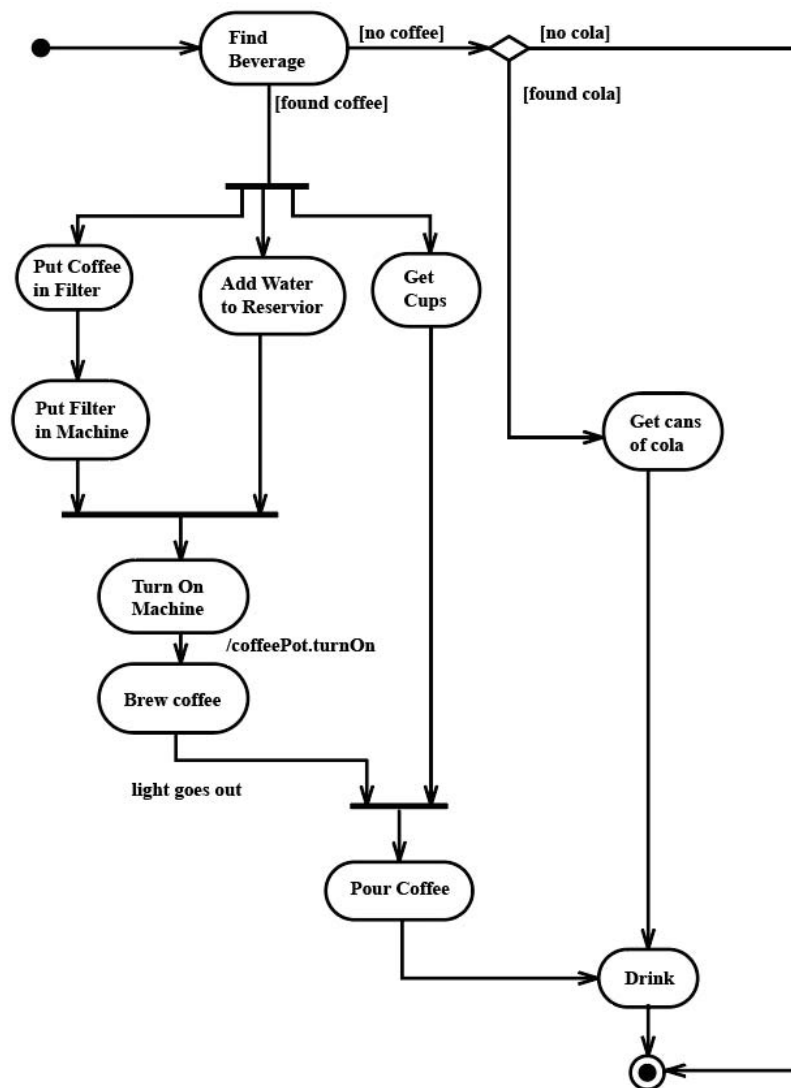
2.2.12 Activity Diagram

Activity diagram merupakan sebuah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity* diagram digunakan untuk menganalisis *behaviour* dengan *use case* yang lebih kompleks dan menunjukkan interaksi-interaksi diantara mereka satu sama lain. *Activity* diagram biasanya digunakan untuk menggambarkan aktivitas bisnis yang lebih kompleks, dimana digambarkan hubungan antara satu *use case* dengan *use case* yang lainnya. [16]

Tabel 2.4. Simbol yang Digunakan pada *Activity Diagram* [15]

(Sumber : Jurnal Pemodelan Uml Sistem Informasi Monitoring Penjualan dan Stok Barang, Ade Hendini, 2016)

| No. | Simbol | Nama | Keterangan |
|-----|---|-----------------------|--|
| 1 |  | <i>Start Point</i> | <i>Start Point</i> merupakan awal dari aktivitas. |
| 2 |  | <i>End Point</i> | <i>End Point</i> merupakan akhir dari aktivitas. |
| 3 |  | <i>Activities</i> | Menggambarkan suatu proses atau kegiatan bisnis. |
| 4 |  | <i>Fork</i> | <i>Fork</i> (percabangan) simbol yang digunakan untuk menggambarkan kegiatan yang dilakukan secara paralel atau penggabungan dua atau lebih kegiatan menjadi satu. |
| 5 |  | <i>Decision Point</i> | Digunakan untuk menentukan pilihan <i>true</i> atau <i>false</i> . |
| 6 |  | <i>Join</i> | <i>Join</i> (penggabungan) simbol yang digunakan untuk menunjukkan adanya dekomposisi. |
| 7 |  | <i>Swimlane</i> | Digunakan untuk membagi <i>activity diagram</i> dan menentukan perilaku dalam aktifitas. |



Gambar 2.2. Contoh *Activity Diagram* [15]

(Sumber : Buku Metode Analisis dan Perancangan Sistem, Prof. Dr. Sri Mulyani, Ak., CA., 2016)

Contoh gambar *Activity Diagram* tersebut menjelaskan tentang alur aktifitas dalam memilih minuman. *Activity Diagram* hanya menyebutkan aturan rangkaian dasar yang harus diikuti. Hal ini penting untuk permodelan bisnis yang memiliki alur proses yang muncul secara paralel. [15]

2.2.13 Sequence Diagram

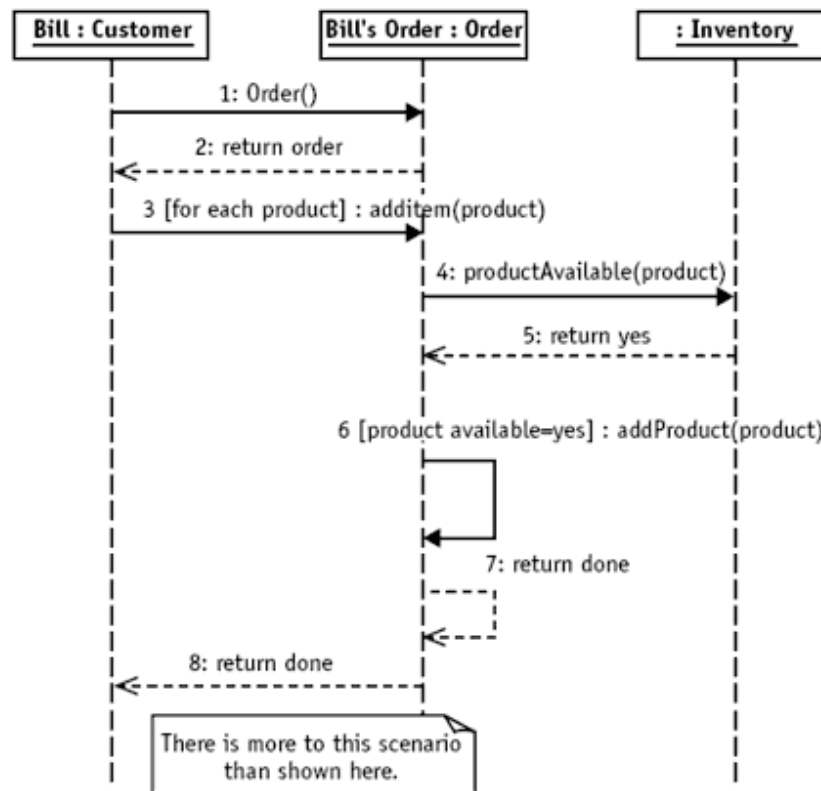
Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Urutan waktu yang dimaksud adalah urutan kejadian yang

dilakukan oleh seorang aktor yang menjalankan sistem. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. [17]

Tabel 2.5. Simbol yang Digunakan pada *Sequence Diagram* [15]

(Sumber : Jurnal Pemodelan Uml Sistem Informasi Monitoring Penjualan dan Stok Barang, Ade Hendini, 2016)

| No. | Simbol | Nama | Keterangan |
|-----|---|-----------------------|--|
| 1 |  | <i>Entity Class</i> | Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data. |
| 2 |  | <i>Boundary Class</i> | Berisi kumpulan kelas yang menjadi <i>interface</i> antara satu atau lebih aktor dengan sistem. |
| 3 |  | <i>Control Class</i> | Berisikan logika aplikasi yang tidak memiliki tanggung jawab terhadap entitas. |
| 4 |  | <i>Message</i> | Simbol pengirim pesan antar kelas |
| 5 |  | <i>Recursive</i> | Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri. |
| 6 |  | <i>Activation</i> | Mewakili sebagian eksekusi operasi dari objek, panjang kotak berbanding lurus dengan durasi aktivitas sebuah operasi. |
| 7 |  | <i>Lifeline</i> | Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> . |



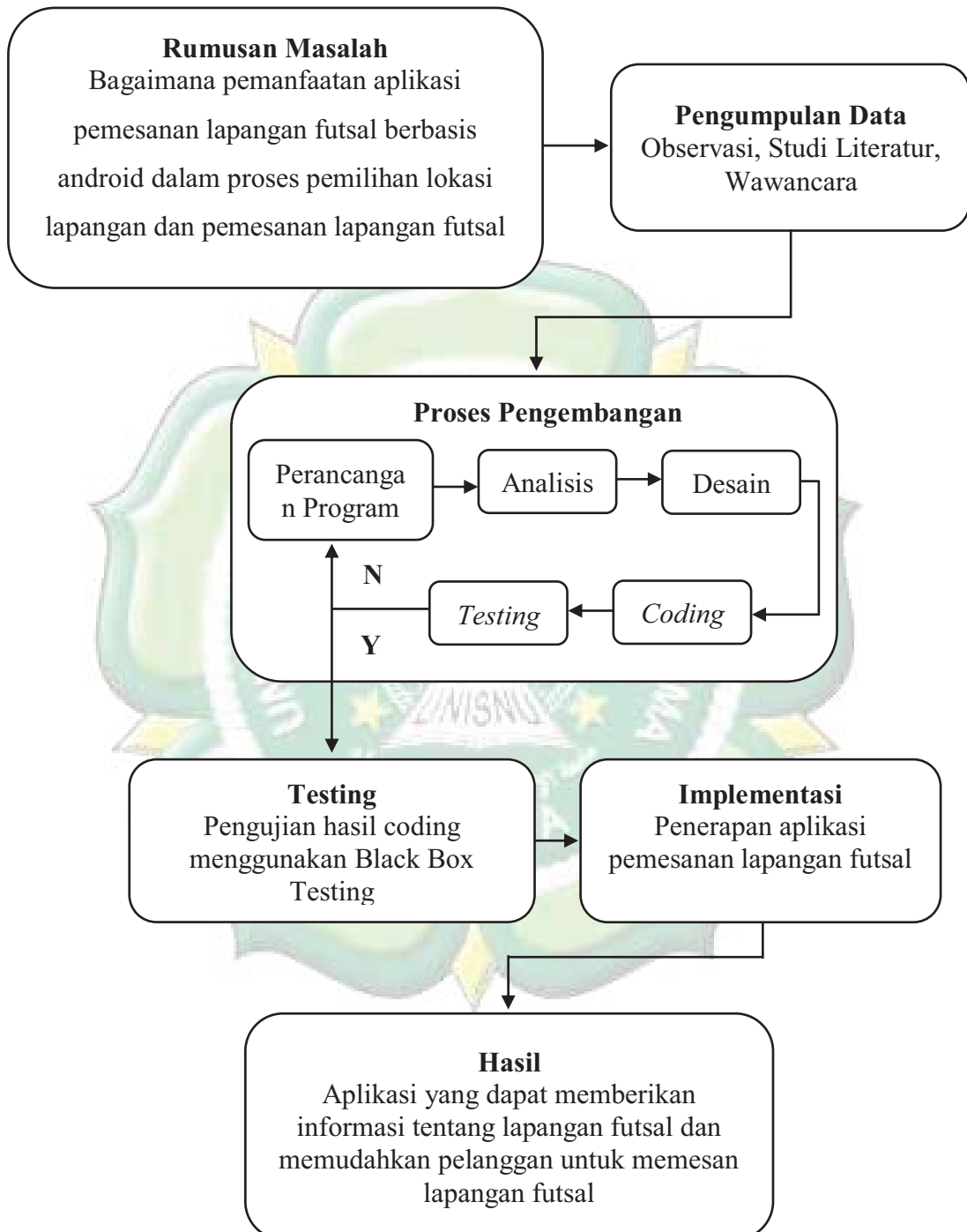
Gambar 2.3. Contoh *Sequence Diagram* [15]

(Sumber : Buku Metode Analisis dan Perancangan Sistem, Prof. Dr. Sri Mulyani, Ak., CA., 2016)

Contoh *Sequence Diagram* tersebut menjelaskan tentang skenario proses pemesanan suatu produk. Terdapat tiga *class* yaitu *Customer*, *Order*, dan *Inventory* pada contoh *Sequence Diagram*. *Customer* melakukan permintaan fungsi order yang diteruskan melalui ke *class Order*. Ketika *class Order* merespon seketika melakukan pengembalian respon order ke *class Customer*, kemudian *class Customer* meminta produk yang akan diorder ke *class Order*. *Class Order* melakukan pengecekan ketersediaan produk ke *class Inventory*. *Class Inventory* mengkonfirmasi ketersediaan produk ke *class Order* dan selanjutnya *class Order* melakukan penambahan produk yang dipesan ke *class Customer*.

2.3 Kerangka Pemikiran

Penulis membuat kerangka pemikiran secara bertahap. Kerangka pemikiran merupakan penjelasan terhadap konsep guna memberi pandangan terhadap penelitian yang akan dilakukan dari awal sampai selesai.



Gambar 2.4. Kerangka Pemikiran