

## 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 Santi dalam penelitiannya yang berjudul Penerapan Algoritma *Best First Search* (BFS) Dalam Pencarian Lokasi Apotek K-24 Berbasis Android Di Kota Makasar. Penelitian ini menggunakan metode pengumpulan data yang bersifat observasi yaitu dengan mengunjungi semua cabang apotek K-24 yang ada di kota Makasar. Keberadaan lokasi apotek masih sulit ditemukan dengan menggunakan *smartphone* Android yang dilengkapi dengan *Global Positioning System* (GPS). Akan tetapi pengguna terkadang masih kebingungan untuk memilih jalur mana yang harus dilalui agar cepat sampai dilokasi tujuan. Melihat permasalahan tersebut peneliti memberikan solusi merancang aplikasi pencarian apotek K-24 menggunakan algoritma *Best First Search* (BFS) berbasis Android di kota Makasar yang diharapkan dapat membantu masyarakat dalam mencari apotek K-24 dengan jalur terpendek, sehingga dapat lebih mengefisienkan waktu, tenaga, dan menghemat biaya perjalanan. Perancangan aplikasi ini menggunakan metode *waterfall* yang terdiri dari beberapa tahapan yaitu, studi literatur, analisa kebutuhan aplikasi, desain *interface* aplikasi, *coding*, dan pengujian aplikasi. Hasil dari penelitian ini menghasilkan sebuah Aplikasi Pencarian Lokasi Apotek K-24 Berbasis Android yang menerapkan algoritma *Best First Search* (BFS) yang dilengkapi dengan GPS agar pengguna dapat lebih mudah dalam menemukan lokasi apotek K-24 terdekat dari posisinya. [4]

Menurut Thoha Nurhardiyana dan A. Syamsul Hidayatullah dalam penelitiannya yang berjudul Sistem Informasi Geografis Pemetaan Lokasi ATM BNI Terdekat di Kota Serang Berbasis Android. Masalah pada penelitian ini adalah

masyarakat tidak tahu lokasi ATM berdasarkan posisi mereka saat ini, di daerah sekitarnya maupun di daerah yang baru di datangi, serta tidak ada keinginan untuk bertanya pada warga sekitar dimana lokasi ATM BNI. Aplikasi berbasis android mampu membantu masyarakat dalam mencari lokasi ATM BNI terdekat di wilayah kota Serang. Aplikasi sistem informasi ini memanfaatkan Sistem Informasi Geografis (SIG) yang diharapkan dapat membantu masyarakat/nasabah Bank BNI dalam mencari lokasi ATM BNI terdekat di kota Serang dengan informasi lebih detail dan mengetahui ATM yang sedang rusak. Perancangan aplikasi ini digambarkan dengan menggunakan model UML (*Unified Modelling Language*) yang terdiri dari *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram* dan dibuat menggunakan Android Studio. Hasil dari penelitian ini menghasilkan Sistem Informasi Geografis Pemetaan Lokasi ATM BNI Berbasis Android yang dapat digunakan oleh semua nasabah Bank BNI dimanapun dan kapanpun, sehingga dapat memudahkan pengguna untuk mengakses ATM BNI terdekat di wilayah kota Serang. [5]

Menurut Desaga Asnanda Poetra, Sujito, dan Ali Syaifulloh dalam penelitiannya yang berjudul Penerapan Algoritma *Best First Search* Untuk Pemilihan Angkutan Kota Malang. Jumlah penduduk kota Malang pada akhir tahun 2015 sebanyak 881.794 jiwa, menjadikan kota Malang menjadi daerah dengan populasi terbanyak ke-2 se-Jawa Timur dan populasi terbanyak ke-10 se-Indonesia. Transportasi angkutan umum merupakan fasilitas pendukung yang melayani jasa mobilisasi masyarakat perkotaan dalam melakukan aktifitas sehari-hari. Ada 25 trayek angkutan umum di kota Malang dan masing-masing trayek memiliki jumlah yang berbeda, dengan banyaknya angkutan kota dengan berbagai macam trayek membuat masyarakat Kota Malang maupun Luar Kota Malang tidak mengetahui informasi tentang angkutan umum, seperti informasi tentang tujuan, harga, dan jam kerja angkutan umum Kota Malang dan tidak adanya sarana yang memudahkan masyarakat untuk mendapatkan informasi mengenai angkutan umum tersebut. Sehingga peneliti mengusulkan aplikasi informasi rute angkutan umum sesuai dengan tujuan yang diinginkan di Kota Malang berbasis

android. Metode yang digunakan pada aplikasi ini adalah *Best First Search*, dimana angkutan dengan tujuan yang sama akan dibandingkan untuk menemukan angkutan dengan rute terpendek berdasarkan lokasi awal dan lokasi tujuan. Hasil dari penelitian ini adalah aplikasi pemilihan angkutan umum berbasis android dengan menerapkan algoritma *Best First Search* yang dapat membantu masyarakat untuk mendapatkan informasi rute, tarif, dan jam operasional angkutan umum Kota Malang dengan mudah. [6]

Tabel 2.1. Perbandingan Referensi

Judul	Penulis	Persamaan	Perbedaan
Penerapan Algoritma <i>Best First Search</i> (BFS) Dalam Pencarian Lokasi Apotek K-24 Berbasis Android Di Kota Makasar	Santi	Penerapan algoritma <i>Best First Search</i> untuk pencarian lokasi Apotek	Perancangan sistem menggunakan metode <i>waterfall</i>
Sistem Informasi Geografis Pemetaan Lokasi ATM BNI Terdekat di Kota Serang Berbasis Android	Thoha Nurhardiyana, A. Syamsul Hidayatullah	Pencarian lokasi ATM terdekat	Menggunakan perancangan pemodelan <i>Unified Modeling Language</i> (UML) yang terdiri dari <i>Use Case Diagram</i> , <i>Class Diagram</i> , <i>Activity Diagram</i> dan <i>Sequence Diagram</i>
Penerapan Algoritma <i>Best First Search</i> Untuk Pemilihan Angkutan Kota Malang	Desaga Asnanda Poetra, Sujito, Ali Syaifulloh	Penentuan rute terpendek pemilihan angkutan umum dengan menerapkan algoritma <i>Best First Search</i>	Menggunakan pemodelan dengan <i>use case diagram</i>

Penelitian-penelitian diatas digunakan sebagai acuan untuk penelitian yang nantinya dipakai dalam pererapan *Best First Search* pada aplikasi pencarian gas

LPG 3 Kg berbasis android di wilayah Jepara. Menggunakan pemodelan *Unified Modelling Language* (UML) yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*. Aplikasi ini menggunakan *Rapid Application Development* sebagai pengembang sistem, yang terdiri dari Rencana kebutuhan (*Requirement Planning*), Proses Desain Sistem (*Design System*), dan Implementasi (*Implementation*). Aplikasi yang akan dibuat nantinya akan menampilkan 10 (sepuluh) sampel lokasi pangkalan dan penjual gas LPG 3 Kg yang ada di wilayah Jepara dengan memperhitungkan jarak yang paling dekat dari tempat kita berada sekarang ke tempat yang akan dituju. Setiap pangkalan dan penjual gas akan ditampilkan dan apabila di klik akan menampilkan informasi lokasi, detail pangkalan atau penjual gas, dan informasi stok gas LPG 3 kg yang tersedia. Selibuhnya, pangkalan atau penjual gas LPG 3 kg bisa mendaftarkan toko atau warungnya melalui aplikasi yang ada.

## **2.2 Tinjauan Pustaka**

### **2.2.1 Android Studio**

Android Studio adalah Lingkungan Pengembang Terpadu - *Integrated Development Environment* (IDE) untuk pengembang aplikasi Android berdasarkan IntelliJ IDEA. Android studio ini diumumkan pada tanggal 16 Mei 2013 pada konferensi Google I/O oleh Produk Manajer Google, Ellie Power. Android Studio dimulai dengan versi 0.1 pada bulan Mei 2013, kemudian dibuat versi 0.8 pada bulan Juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai pada versi 1.0. Berbasiskan JetBrains' IntelliJ IDE, Android Studio di desain khusus untuk *Android Development*. [7] Android Studio memiliki fitur untuk meningkatkan produktivitas pada saat membuat aplikasi android, sebagai berikut :

1. Sistem versi berbasis Gradle yang fleksibel
2. Emulator cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan perangkat android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru

5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama
6. Dukungan C++ dan NDK
7. Dukungan bawaan untuk Google Cloud Platform, untuk mempermudah pengintegrasian Google Google Messaging dan App Engine [8]

### 2.2.2 Android

Android merupakan sistem operasi perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android merupakan platform terbuka (*Open Source*) bagi pengembang aplikasi. Awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian pada tahun 2005 dibeli oleh *Google Inc*. Dimulai sejak Oktober 2003 ketika 4 orang pakar IT, Andi Rubin, Rich Miner, Nick Sears dan Chris White mendirikan *Android.Inc*, di California US. Visi Android untuk mewujudkan *mobile device* yang lebih peka dan mengerti pemiliknya. Kemudian untuk mengembangkan Android dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. [9]

Tabel 2.2. Daftar Versi *Android*. [10]

Versi	Code Name	Tanggal Rilis	API Level	DVM/ART	Fitur Baru
1.5	Cupcake	27 April 2009	3	-	-
1.6	Donut	15 September 2009	4	-	<ul style="list-style-type: none"> <li>• Quick Search Box</li> <li>• Screen Size Diversity</li> <li>• Android Market</li> </ul>

2.0 - 2.1	Eclair	26 Oktober 2009	5	-	<ul style="list-style-type: none"> <li>• Google Maps Navigation</li> <li>• Home Screen Customization</li> <li>• Speech-to-Text</li> </ul>
2.2 – 2.2.3	Froyo	20 Mei 2010	8	DVM	<ul style="list-style-type: none"> <li>• Voice Action</li> <li>• Portable Hotspot</li> <li>• Dalvik JIT</li> </ul>
2.3 – 2.3.7	Gingerbread	9 Februari 2011	9 – 10	DVM	<ul style="list-style-type: none"> <li>• Gaming APIs</li> <li>• NFC</li> <li>• Battery Management</li> </ul>
3.0 – 3.2.6	HoneyComb	22 Februari 2011	11 -13	DVM	<ul style="list-style-type: none"> <li>• Tablet-Friendly Design</li> <li>• System Bar</li> <li>• Quick Settings</li> </ul>
4.0 – 4.6	Ice Cream Sandwich	19 Oktober 2011	15	DVM	<ul style="list-style-type: none"> <li>• Custom Home Screen</li> <li>• Data Usage Control</li> <li>• Android Beam</li> </ul>
4.1 – 4.3.1	Jelly Bean	9 Juli 2012	18	DVM	<ul style="list-style-type: none"> <li>• Google Now</li> <li>• Actionable Notifications</li> <li>• Account Switching</li> </ul>
4.4 – 4.4.4	Kitkat	31 Oktober 2013	19 -20	DVM (and ART 1.6.0)	<ul style="list-style-type: none"> <li>• Voice : Ok Google</li> <li>• Immersive Design</li> <li>• Smart Dialer</li> </ul>



5.1 – 5.1.1	Lollipop	12 Novemb er 2014	21 -22	ART	<ul style="list-style-type: none"> <li>• Material Design</li> <li>• Multiscreen</li> <li>• Notifications</li> </ul>
6.0 – 6.0.1	Marshmallow	5 Oktober 2015	23	ART	<ul style="list-style-type: none"> <li>• Now On Tap</li> <li>• Permissions</li> <li>• Battery (Doze &amp; App Standby)</li> </ul>
7.1 – 7.1.2	Nougat	22 Agustus 2016	24 – 25	ART	<ul style="list-style-type: none"> <li>• Multi window</li> <li>• GIF Keyboard</li> </ul>
8.0 – 8.1	Oreo	25 Oktober 2017	26 – 27	ART	<ul style="list-style-type: none"> <li>• Picture-in-Picture</li> </ul>
9	Pie	6 Agustus 2018	28	ART	<ul style="list-style-type: none"> <li>• Adaptive Battery</li> <li>• Adaptive Brightness</li> </ul>
10	Ten	Mei 2019	29	ART	<ul style="list-style-type: none"> <li>• Privacy Feature</li> <li>• New API for Gesture Navigation</li> <li>• Dark Mode</li> </ul>

### 2.2.3 Kotlin

Kotlin merupakan bahasa pemrograman dengan tipe statis (*statically typed*) yang berjalan pada JVM (*Java Virtual Machine*) atau menggunakan *Compiler LLVM* yang dapat dikompilasikan ke dalam bentuk *Source Code* JavaScript. Kotlin didesain untuk dapat bekerja sama dengan kode bahasa Java dan bergantung pada kode bahasa Java dari *Java Library Class* yang ada, seperti berbagai *framework* Java yang ada. Pengembang Kotlin berasal dari tim

programer JetBrains yang berada di Rusia. Tim pengembang memutuskan menamakan Kotlin dengan mengambil nama dari sebuah pulau yang ada di Rusia. Setelah Google mengumumkan Kotlin menjadi bahasa kelas satu untuk Android, maka bersama Java dan C++, Kotlin menjadi bahasa resmi untuk pengembangan aplikasi-aplikasi Android. [11]

#### **2.2.4 Firebase**

Pada bulan April 2012, James Tamplin dan Andrew Lee meluncurkan *Firebase*. Pada tahap awal, *firebase* menyediakan API yang membantu mengintegrasikan modul *chat* online ke situs web. Sekarang *firebase* adalah salah satu platform BaaS (*Backend as a Service*) yang dominan yang terus meningkatkan kinerja dan fungsionalitas baru. *Firebase* sangat cocok digunakan untuk pengembangan aplikasi karena kecepatannya yang *real time* dan responsif. *Firebase* didukung untuk banyak platform seperti Android, iOS, JavaScript, Java, Objective-C, Swift dan Node.js. Penyimpanan data pada *firebase* menggunakan JSON supaya lebih mudah dalam penggunaannya dalam membaca data dan tampilan website *firebase* yang *user-friendly*. [12]

#### **2.2.5 Javascript Object Notation (JSON)**

*JavaScript Object Notation* (JSON) adalah salah satu jenis format data yang tidak bergantung bahasa pemrograman tertentu, yang diperkenalkan pertama kali oleh Douglas Crockford. JSON merupakan pertukaran data yang ringan, mudah dibaca dan ditulis manusia dan mudah diterjemahkan oleh komputer. Kumpulan data JSON diawali dengan tanda { dan diakhiri dengan tanda }. JSON dapat digunakan dalam bahasa pemrograman C, C++, C#, Java, JavaScript, Perl, Python dll. Dengan JSON kita dapat melakukan penerimaan dan pengiriman data dengan mudah dan fleksibel. [13]

#### **2.2.6 Google Maps**

Google Maps adalah layanan aplikasi peta online gratis yang disediakan oleh Google. Google Maps dapat digunakan untuk pencarian lokasi di suatu daerah, yang memberikan peta jalan, kondisi lalu lintas, transit publik, satelit, medan, serta perencanaan rute perjalanan dengan jalan kaki, mobil, sepeda, atau angkutan umum. [14]



Adapun fungsi Google Maps adalah sebagai berikut :

1. Mempercepat pencarian lokasi dalam waktu singkat.
2. Membantu mencari jalan yang tercepat untuk menuju lokasi yang akan dituju.
3. Mempermudah sistem penyimpanan data.
4. Mengetahui tempat baru yang mungkin belum diketahui sebelumnya.
5. Adanya pembaruan data yang lebih cepat daripada menggunakan peta konvensional.
6. Bias diakses dimana saja dan kapan saja

### 2.2.7 Best First Search (BFS)

*Best First Search* merupakan metode yang membangkitkan simpul dari simpul sebelumnya. Simpul yang dibangkitkan adalah simpul yang memiliki biaya terkecil diantara semua *leaf nodes* (simpul-simpul pada level terdalam) yang pernah dibangkitkan. *Best First Search* merupakan kombinasi dari kelebihan metode *Depth First Search* dan *Breath First Search*. Pada setiap langkah proses *Best First Search*, dipilih simpul-simpul dengan menerapkan fungsi *heuristik* yang memadai pada setiap simpul yang dipilih dengan menggunakan aturan-aturan tertentu untuk menghasilkan penggantinya. [15]

Simpul dengan nilai heuristik terbaik akan dibuka atau dikerjakan lebih dahulu. Nilai heuristik dikatakan terbaik artinya apabila nilai tersebut mendekati nilai sebenarnya. Diasumsikan bahwa dalam permasalahan pencarian rute terpendek, nilai yang dianggap baik apabila nilai tersebut memberikan hasil yang lebih kecil dari nilai yang lainnya karena konteks yang dibahas adalah jarak. Setelah simpul dengan nilai terbaik diperoleh, jika *goal state* belum ditemukan maka akan dilakukan pemeriksaan pada simpul berikutnya dengan nilai heuristik terbaik pada kedalaman yang sama. Simpul tersebut kemudian dibuka dan diperiksa apakah terdapat *goal state* pada cabang-cabangnya. Bila *goal state* belum ditemukan, akan dilakukan proses yang sama pada simpul berikutnya. [16] Fungsi heuristik yang digunakan merupakan perkiraan (estimasi) *cost* dari *initial state* ke *goal state*, yang dinyatakan dengan :

$$f'(n) = g(n) + h'(n)$$

Dimana  $f'(n)$  = perkiraan *cost* dari *initial* ke *goal*

$g(n)$  = *cost* dari *initial state* ke *current state* (*biaya* yang sudah dikeluarkan dari keadaan awal sampai keadaan  $n$ )

$h'(n)$  = estimasi *cost* dari *current state* ke *goal state* (*perkiraan biaya* untuk sampai pada suatu tujuan mulai dari  $n$ ).

Algoritma pada best first search yaitu :

1. Buat sebuah stack kemudian inialisasikan node akar sebagai node pertama.
2. Bila node pertama  $\neq$  GOAL, node diganti dan dihapus dengan anak-anaknya.
3. Kemudian, keseluruhan node pada stack diurutkan secara ascending berdasarkan fungsi heuristik yang digunakan.
4. Bila node pertama  $\neq$  GOAL, ulangi langkah 2. Bila node pertama = GOAL, cari solusi dengan mengunjungi jalur dari GOAL ke node akar.

### 2.2.8 Location Based Service (LBS)

*Location Based Service* (LBS) merupakan layanan informasi berbasis lokasi geografis yang didasarkan pada posisi pengguna dan memberikan layanan informasi yang tersedia berdasarkan lokasi pengguna pada saat itu. Dengan teknologi *Location Based Service* (LBS) dapat memudahkan pengguna untuk menemukan tempat-tempat yang diinginkan. [17]

*Location Based Service* (LBS) memiliki lima komponen yaitu :

1. Data dan *content provider*, informasi yang disesuaikan dengan permintaan pengguna di lokasi tertentu pada saat itu.
2. *Mobile device*, perangkat yang dilengkapi dengan koneksi internet dan *Global Positioning System* (GPS) yang digunakan pengguna untuk informasi yang dibutuhkan di lokasi tertentu pada saat itu.
3. *Network communication*, media penghubung antara perangkat komunikasi pengguna dengan server.

4. *Positioning system*, memungkinkan posisi pengguna diketahui secara geografis.
5. *LBS provider*, untuk melayani permintaan data pengguna yang disesuaikan dengan lokasi pengguna pada saat itu

### 2.2.9 Unified Modeling Language

*Unified Modeling Language* (UML) merupakan bahasa pemodelan standar. UML memiliki sintak dan semantik. UML memiliki aturan-aturan yang harus diikuti, dimana elemen pada model-model yang dibuat berhubungan antara satu dengan yang lainnya harus mengikuti standar yang ada. [18] UML diaplikasikan dan memiliki tujuan sebagai berikut :

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis
3. Menjabarkan sistem secara rinci untuk analisis dan mencari yang dibutuhkan oleh sistem
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.


### 2.2.10 Use Case Diagram

Membuat *use case* merupakan hal yang sangat penting karena akan mendapatkan banyak informasi yang berkaitan dengan aturan-aturan yang akan dipahami. Dalam hal ini, setiap objek yang berinteraksi dengan sistem merupakan aktor, sedangkan *use case* merupakan deskripsi lengkap tentang bagaimana sistem berperilaku pada aktornya. Dengan demikian *use case* merupakan deskripsi lengkap tentang interaksi antara aktor dengan sistem yang akan kita kembangkan. [19] Berikut merupakan simbol-simbol yang biasa digunakan pada *Use Case Diagram* [20]:

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




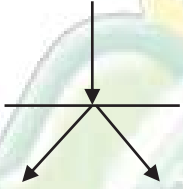

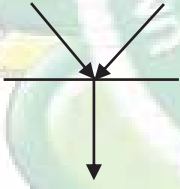

Gambar 2.1 merupakan *use case* sistem penjualan teh hitam. Digambarkan dengan dua aktor yang ditunjukkan pada diagram diatas yaitu *user* dan *admin*. *User* pada sistem dapat melakukan registrasi sebagai pelanggan, kemudian dapat login untuk melakukan pemesanan di keranjang belanja, melakukan transaksi, dan mengisi buku tamu. *Admin* pada sistem dapat login untuk melakukan pengecekan dan penginputan data barang (Teh Hitam), data pelanggan, data pemesanan, dan merekap laporan data distribusi teh hitam, laporan pelanggan, laporan pemesanan, dan laporan penjualan.

### 2.2.11 Activity Diagram

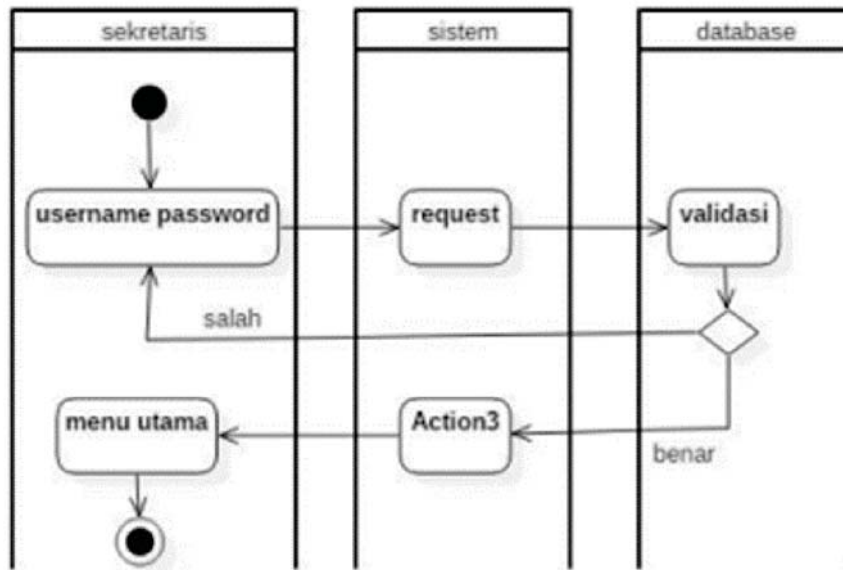
*Activity* diagram merupakan sebuah diagram alur kerja yang menjelaskan masing-masing kegiatan pada sistem, orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut. Digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu sistem operasi sehingga dapat digunakan untuk aktivitas lainnya. *Activity* diagram seperti *flowchart* karena memodelkan alur kerja dari suatu aktivitas ke aktivitas lainnya. [21] Berikut merupakan simbol-simbol yang biasa digunakan pada *Activity Diagram* [20]:

Tabel 2.4. Simbol yang Digunakan pada *Activity Diagram*

(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*

Gambar 2.2 menjelaskan pada proses login terdapat penginputan *username* dan *password*. Jika dalam validasi sesuai dengan data yang ada dalam database, maka akan masuk ke menu utama. Sebaliknya jika validasi tidak sesuai maka akan ada konfirmasi dan kembali ke menu login.




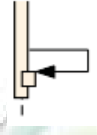


### 2.2.12 *Sequence Diagram*

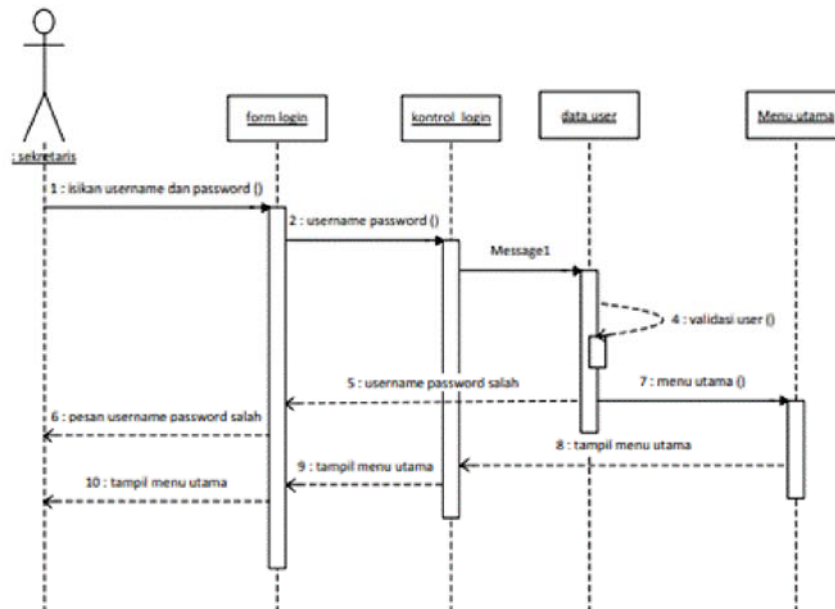
*Sequence* diagram adalah diagram yang digunakan untuk mendefinisikan *input* dan *output* serta urutan interaksi antara pengguna dan sistem untuk sebuah *use case*. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu yang sesuai dengan *use case* diagram. [22] Berikut merupakan simbol-simbol yang biasa digunakan pada *Sequence Diagram* [20]:

Tabel 2.5. Simbol yang Digunakan pada *Sequence Diagram*

(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> .

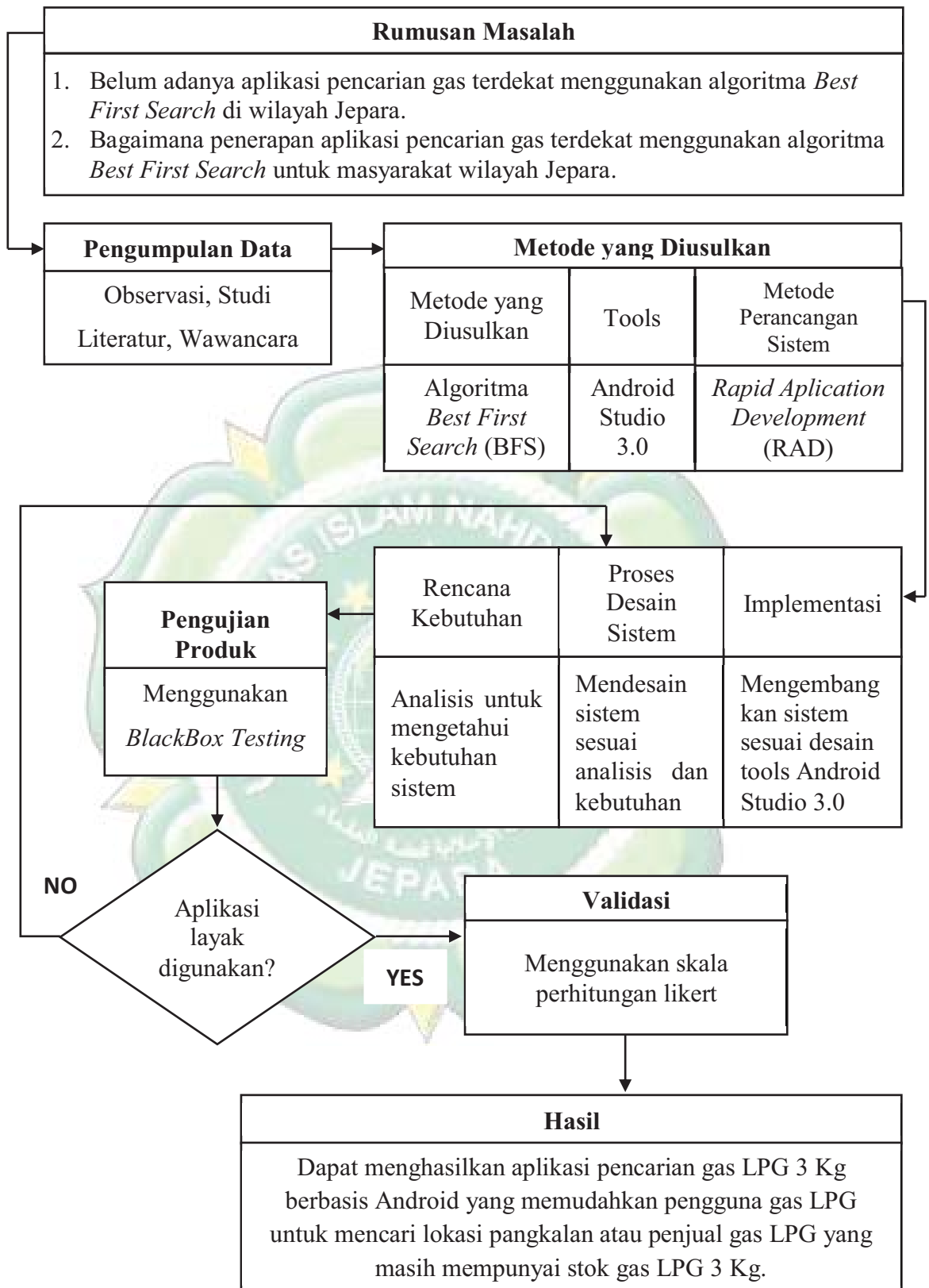
Gambar 2.3. Contoh *Sequence Diagram*

Gambar 2.3 menjelaskan sekretaris disini sebagai aktor. Form login, kontrol login, data *user*, dan menu utama sebagai objek. Ketika sekretaris akan melakukan login maka sekretaris harus menginput *username* dan *password*, kemudian akan diproses pada kontrol login. Jika pada proses penginputan *username* dan *password* tidak sesuai dengan data *user* maka sistem akan memberi pesan untuk kembali ke menu login, sebaliknya jika penginputan sesuai dengan data *user* maka akan masuk ke menu utama kemudian dapat melihat tampilan menu utama.

### 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