



**APLIKASI *MOBILE* Pencarian *ATM* Terdekat di Kota Binjai
Berbasis Android dengan Penerapan *ALGORITMA*
*DIJKSTRA***

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA : PUTRI RAMADANI PASARIBU
NPM : 1514370866
PROGRAM STUDI : SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

PUTRI RAMADANI PASARIBU

APLIKASI MOBILE PENCARIAN ATM TERDEKAT DI KOTA BINJAI BERBASIS ANDROID DENGAN PENERAPAN ALGORITMA DIJKSTRA

2015

Salah satu masalah yang paling banyak ditemukan saat ini ialah sulitnya mengetahui *ATM* BRI terdekat dari lokasi kita. Sering kali kita bertanya dimana lokasi *ATM* BRI terdekat dan tidak jarang kita harus keluar rumah terlebih dahulu untuk mengetahui lokasi *ATM* BRI mana yang terdekat dengan rumah kita tanpa mengetahui terlebih dahulu lokasi pasti terdekat dari *ATM* BRI yang akan kita tuju. Pada penelitian ini, penulis mengangkat tentang sulitnya mencari mesin ATM terdekat dan memberikan solusi untuk memecahkan masalah tersebut dengan cara membuat suatu aplikasi yang dapat digunakan untuk menemukan lokasi *ATM* BRI terdekat dari lokasi pengguna. Dengan menggunakan aplikasi ini, nantinya pengguna dapat mengetahui *ATM* BRI mana yang terdekat dari lokasinya beserta nama *ATM* Bank BRI dan jarak tempuhnya. Penulis memanfaatkan algoritma dijkstra untuk membuat sistem aplikasi pencarian ATM terdekat ini. Algoritma *dijkstra* merupakan algoritma yang dipakai untuk memecahkan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*maps*). Dengan memanfaatkan *Google Maps API*, penulis akan membuat suatu sistem aplikasi yang dapat digunakan untuk mencari ATM terdekat dari lokasi pengguna. Dengan memanfaatkan *Google Maps API* juga, pengguna akan menampilkan detail dari nama bank ATM beserta jarak yang akan ditempuh pengguna untuk menuju ATM tersebut.

Kata Kunci: *ATM, Algoritma Dijkstra, dan Google Maps API.*

DAFTAR ISI

HALAMAN JUDUL

LEMBAR PERSETUJUAAN

LEMBAR PENGESAHAN

ABSTRAK

KATA PENGANTAR.....i

DAFTAR ISI iii

DAFTAR TABEL ix

DAFTAR GAMBAR..... x

DAFTAR LAMPIRAN xi

DAFTAR ISTILAH..... xii

TAHAPAN PENELITIAN xiii

BAB I PENDAHULUAN..... 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 3

1.3 Batasan Masalah 3

1.4 Tujuan Penulisan	4
1.5 Manfaat Penelitian	4
BAB II LANDASAN TEORI.....	5
2.1 Pengertian Sistem.....	5
2.2 Karakteristik Sistem.....	5
a. Komponen Sistem.....	5
b. Batasan Sistem.....	6
c. Lingkungan Luar Sistem.....	6
d. Penghubung Sistem.....	6
e. Masukan sistem	6
f. Keluaran Sistem.....	6
g. Pengolahan Sistem	6
h. Tujuan Sistem.....	7
2.2.1 Elemen Sistem.....	7
a. Tujuan.....	7
b. Batasan.....	7
c. Kontrol	7
d. Input.....	7
e. Proses.....	7
f. Output.....	7
2.3 Konsep Dasar Sistem.....	8

2.3.1 Aplikasi Mobile.....	9
2.3.2 Sistem Operasi Mobile.....	10
2.3.3 Jenis Aplikasi Mobile.....	11
a. Native Apps.....	12
b. Web-Based Apps.....	13
c. Hybrid Apps.....	14
2.3.4 ATM.....	15
2.3.5 Algoritma Dijkstra.....	16
2.3.7 Google Maps API.....	17
2.3.8 Android.....	18
2.3.8 Bahasa Pemrograman.....	19
2.4 Proses Pembuatan Suatu Program.....	20
a. Kompilasi.....	21
b. Interpretasi.....	22
2.4.1 Java.....	23
2.4.2 Android Studio.....	24
2.4.3 Use Case Diagram.....	25
2.4.4 Activity Diagram.....	29
2.4.5 Flowchart.....	37
BAB III ANALISA PERANCANGAN SISTEM.....	38
3.1. Tahapan Penelitian.....	38
3.2. Metode Pengumpulan Data.....	39
a. Observation.....	39

b. <i>Library search</i>	40
3.3 Analisis Sistem Sedang Berjalan	41
3.3.1 Perhitungan Metode Dijkstra.....	42
3.4 Rancangan Penelitian.....	43
3.4.1 Rancangan Sistem Secara Global.....	43
a. Use Case Diagram.....	44
b. Class Diagram.....	45
3.4.2 Rancangan Sistem Secara Detail.....	39
a. Activity Diagram.....	40
3.4.2.2. Activity Diagram Login	41
3.4.2.3 Activity Diagram Mencari Lokasi ATM.....	42
3.4.2.4 Activity Diagram Data ATM.....	43
3.4.2.5 Sequence Diagram	44
3.4.2.6 Sequence Diagram Login.....	45
3.4.2.7 Sequence Diagram Data ATM	46
3.4.2.8 Sequence Diagram Mencari Lokasi ATM	47
3.4.4 Desain Sistem Secara Detail.....	48
a. Desain Form Halaman Depan	49
b. Desain Form Halaman Mulai.....	50
c. Desain Form Halaman Lokasi ATM.....	56

d.Desain Form Halaman Tentang.....	57
BAB IV Hasil Dan Pembahasan.....	58
4.1. Implementasi Sistem.....	58
4.2 Spesifikasi Sistem.....	59
a.Analisa Hardware.....	59
b. Analisa Software.....	59
4.2.1 Hasil Rancangan Sistem.....	59
a. Tampilan Menu Login.....	60
b. Tampilan Menu Utama.....	61
c.Tampilan Menu Pencarian Lokasi ATM.....	62
d.Tampilan Tentang Aplikasi.....	63
e.Pengujian Black Box.....	64
BAB V PENUTUP.....	65
1. Kesimpulan.....	65
2. Saran.....	65
DAFTAR PUSTAKA	
BIOGRAFI PENULISAN	
LAMPIRAN - LAMPIRAN	

KATA PENGANTAR

Syukur Alhamdulillah penulis ucapkan kepada ALLAH SWT, yang telah memberikan kesehatan kepada penulis sehingga dapat menyelesaikan skripsi ini dengan judul yakni **“APLIKASI MOBILE PENCARIAN *ATM* TERDEKAT DI KOTA BINJAI BERBASIS ANDROID DENGAN PENERAPAN ALGORITMA DIJKSTRA”**.

Penulisan skripsi ini disusun untuk memenuhi salah satu syarat ujian akhir dan memperoleh gelar sarjana komputer pada Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi Medan.

Pada kesempatan ini, penulis menyampaikan rasa terima kasih dan penghargaan yang sebesar-besarnya kepada :

1. Teristimewa kepada Orangtua dan keluarga besar Saya, yang telah banyak memberikan semangat serta motivasi tetap optimis, ikhtiar dan bantuan baik moral maupun material selama Saya mengikuti Pendidikan kuliah hingga selesainya Skripsi saya.
2. Rektor Universitas Pembangunan Pancabudi Medan, Bapak DR. H. Muhammad Isa Indrawan, SE, MM.
3. Rektor I, Bapak Ir. Bhakti Alamsyah, M.T, Ph.D.
4. Dekan Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi Medan, Ibu Sri Shindi Indira, ST, M.Sc,
5. Ketua Program Studi Ilmu Komputer Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi Medan, Bapak Eko Hariyanto, S.Kom.M.Kom.,
6. Dosen Pembimbing I, Bapak Rian Farta Wijaya S.Kom.M.Kom.,
7. Dosen Pembimbing II, Ibu Supina Batubara S.Kom.M.Kom.,

8. Bapak/Ibu dosen beserta seluruh staf Universitas Pembangunan Panca Budi Medan yang telah mendidik dan membimbing penulis selama mengikuti perkuliahan.
9. Terimakasih juga untuk Verawaty Br Simamora, Siti Nurbayani, Rizka Ramadhani, Arnidya, Rian Dwi, Hamdika, Ranti Widia, Sugan, Asmaul Husna, Royhan, Prida, Anisa yang memberikan semangat serta motivasi tetap optimis, ikhtiar sehingga saya bisa menyelesaikan Skripsi ini dengan tekun dan semaksimal mungkin.
10. Kepada seluruh rekan-rekan di program Studi Sistem Komputer Universitas Pembangunan Panca Budi Medan yang telah memberikan dukungan moril kepada penulis.

Akhir kata, penulis berharap semoga penyusunan skripsi ini dapat bermanfaat bagi pembaca sekaligus menjadi gambaran jika suatu saat nantinya program pembelajaran ini dapat dimanfaatkan.

Medan, 5 November 2019

Penulis,

Putri Ramadani Pasaribu

1514370866

BAB I

PENDAHULUAN

1.1 Latar belakang

ATM (Anjungan Tunai Mandiri) merupakan tempat untuk melakukan transaksi nontunai seperti melakukan transfer uang, penarikan uang, cek rekening tabungan dan membayar tagihan. Saat ini, *ATM* merupakan tempat yang tidak pernah sunyi dari pengunjung dikarenakan manfaatnya yang tinggi dan ketersediaannya selama 24 jam.

Salah satu masalah yang paling banyak ditemukan saat ini ialah sulitnya mengetahui *ATM* BRI terdekat dari lokasi kita. Sering kali kita bertanya dimana lokasi *ATM* BRI terdekat dan tidak jarang kita harus keluar rumah terlebih dahulu untuk mengetahui lokasi *ATM* BRI mana yang terdekat dengan rumah kita tanpa mengetahui terlebih dahulu lokasi pasti terdekat dari *ATM* BRI yang akan kita tuju. Belum lagi ketersediaan *ATM* bank BRI yang ingin kita tuju membuat pencarian mesin *ATM* menjadi lebih rumit dan sulit diketahui lokasinya.

Pada penelitian ini, penulis mengangkat tentang sulitnya mencari mesin *ATM* terdekat dan memberikan solusi untuk memecahkan masalah tersebut dengan cara membuat suatu aplikasi yang dapat digunakan untuk menemukan lokasi *ATM* BRI terdekat dari lokasi pengguna. Dengan menggunakan aplikasi ini, nantinya pengguna dapat mengetahui *ATM* mana yang terdekat dari lokasinya beserta nama *ATM* Bank dan jarak tempuhnya.

Penulis memanfaatkan algoritma dijkstra untuk membuat sistem aplikasi pencarian ATM terdekat ini. Algoritma *dijkstra* merupakan algoritma yang dipakai untuk memecahkan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*maps*). Algoritma *dijkstra* bekerja dengan membuat jalur ke satu simpul optimal pada setiap langkah. Jadi pada langkah ke N, setidaknya ada N node yang sudah kita tahu jalur terpendek.

Google Maps merupakan satu teknologi pemetaan yang menggunakan algoritma *dijkstra* dalam penerapan pencarian lokasinya. Dengan algoritma dijkstra, *Google Maps* dapat menentukan rute mana yang paling terdekat yang dapat ditempuh pengguna untuk sampai pada tujuan yang ia inginkan.

Dengan memanfaatkan *Google Maps API*, penulis akan membuat suatu sistem aplikasi yang dapat digunakan untuk mencari ATM terdekat dari lokasi pengguna. Dengan memanfaatkan *Google Maps API* juga, pengguna akan menampilkan detail dari nama bank ATM beserta jarak yang akan ditempuh pengguna untuk menuju ATM tersebut.

Atas dasar inilah penulis memilih judul **“APLIKASI *MOBILE* PENCARIAN ATM TERDEKAT DI KOTA BINJAI BERBASIS ANDROID DENGAN PENERAPAN ALGORITMA *DIJKSTRA*”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka rumusan masalah dalam penelitian ini adalah :

- a. Bagaimana menentukan rute terpendek pencarian *ATM* BRI terdekat di Kota Binjai berbasis android?
- b. Bagaimana menerapkan algoritma Dijkstra pada aplikasi yang dirancang?
- c. Bagaimana menyediakan informasi lokasi *ATM* terdekat di Kota Binjai?

1.3 Batasan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka batasan masalah dalam penulisan ini adalah :

- a. Program pencarian *ATM* terdekat ini menggunakan *Google Maps API* untuk membuat sistem pencariannya.
- b. Program ini dibuat menggunakan software *Android Studio* dengan menggunakan bahasa *Java* sebagai bahasa pemrosesnya dan *XML* sebagai bahasa layoutnya.
- c. Pencarian *ATM* terdekat ini hanya akan diimplementasikan pada daerah kota Binjai saja.
- d. Objek yang diteliti adalah *ATM* terdekat yang terdapat di Kota Binjai?
- e. Rute/jalan yang ditampilkan pada aplikasi hanya dapat di akses/dilalui oleh kendaraan roda 4 saja.
- f. Program aplikasi pencarian *Atm* hanya bisa digunakan bagi pengguna *ATM Card* nasabah *BRI* saja.

1.4 Tujuan Penelitian

Berdasarkan menjawab rumusan masalah yang telah dijelaskan di atas, berikut merupakan tujuan penelitian dari penulisan skripsi ini yaitu:

- a. Untuk menerapkan algoritma *dijkstra* dalam perancangan aplikasi mobile pencarian ATM terdekat di Kota Binjai.
- b. Untuk membuat suatu program yang dirancang dan mempermudah masyarakat dalam pencarian lokasi ATM terdekat agar lebih efektif dan efisien .
- c. Penggunaan Google Maps API sebagai teknologi perantara yang akan disematkan pada aplikasi untuk menunjukkan lokasi ATM dan jarak yang akan ditempuh oleh pengguna.
- d. Memberikan informasi lengkap ke pengguna mengenai nama bank, jarak tempuh dan estimasi waktu yang akan ditempuh pengguna menuju lokasi ATM.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini diantaranya yaitu :

- a. Untuk mencari rute terdekat khususnya lokasi ATM terdekat.
- b. Sebagai media informasi bagi pengguna kendaraan untuk memudahkan mendapat rekomendasi rute ATM terdekat menuju lokasi tujuan.
- c. Sebagai aplikasi alternatif dan bahan pembelajaran dalam memahami pencarian rute ATM terdekat di Kota Binjai.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi, dan energi untuk mencapai suatu tujuan.

Menurut Eka Iswandy (2015) sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam unsur mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem-subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehinggalah sistem dapat bekerja secara efektif dan efisien.

2.2.1 Karakteristik Sistem

Karakteristik sistem adalah sistem yang mempunyai komponen-komponen, batas sistem, lingkungan sistem, penghubung, masukan, keluaran, pengolah dan sasaran. Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu:

a. Komponen Sistem

Sistem tidak mengenal betapa kecilnya, sistem selalu mengandung komponen-komponen. Dan adapun sistem yang berkapasitas tinggi yang di sebut dengan supra sistem.

b. Batasan Sistem

Batasan sistem memberikan sebuah batasi antara suatu sistem dengan menunjukkan tempat dari sistem tersebut.

c. Lingkungan Luar Sistem

Lingkungan luar sistem adalah apapun yang bersifat menguntungkan dan dapat bersifat merugikan sistem, merupakan suatu lingkungan luar sistem walaupun sistem di luar batas suatu sistem yang mempengaruhi operasi sistem.

d. Penghubung Sistem

Merupakan suatu jembatan antara penghubung antara satu dengan yang lainnya yang bersifat mengantar aliran informasi yang bertujuan untuk menyampaikan kepada yang lain, yang bertujuan untuk menyelesaikan pengiriman data tersebut.

e. Masukan Sistem

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*)

f. Keluaran Sistem

Merupakan hasil dari energi yang diolah dan diklarifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran sistem dapat berupa masukan untuk subsistem yang lain atau kepada supra sistem.

g. Pengolahan Sistem

Suatu sistem produksi akan mengelola suatu barang mentah untuk di olah menjadi barang yang sudah jadi.

h. Tujuan Sistem

Sasaran dari sistem menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem.

2.2.2. Elemen Sistem

Eka Iswandy (2015), Berdasarkan penjelasan di atas sudah tertera berbagai macam sistem tersebut, sistem terdiri dari elemen-elemen yang saling berkaitan. Elemen-elemen tersebut terdiri dari :

a. Tujuan

Tujuan merupakan tujuan dari sistem yang dapat berupa tujuan usaha, suatu goals dalam pencapaian tertentu.

b. Batasan

Batasan merupakan batasan-batasan yang berupa peraturan-peraturan, biaya biaya, dan peralatan.

c. Kontrol

Kontrol merupakan memonitoring dari pelaksanaan pencapaian suatu tujuan sistem yang terdiri dari pemasukan data (*input*), pengeluaran data (*output*), dan pengoperasian proses (*process*).

d. *Input*

Input merupakan bagian dari sistem yang bertugas untuk menerima data masukan yaitu masalah, dan jenis-jenis masukan.

e. Proses

Proses merupakan bagian yang memproses masukan data menjadi informasi sesuai dengan keinginan penerima.

f. *Output*

Output merupakan hasil keluaran atau goals akhir dari perancangan sistem.

2.2.3 Konsep Dasar Sistem

Sistem merupakan pendekatan prosedur komponen, dengan pendekatan prosedur sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur untuk membentuk suatu kesatuan dan tujuan tertentu.

Menurut Romney dan Steinbart (2015:3) “sistem adalah suatu rangkaian yang terdiri dari dua atau lebih komponen yang saling berhubungan dan saling berinteraksi satu dengan yang lain untuk mencapai tujuan dimana sistem biasanya terbagi dalam sub sistem yang lebih kecil yang mendukung sistem yang lebih besar”.

Menurut Gelinas dan Dull (2015:11), “Sistem merupakan seperangkat elemen yang saling bergantung yang bersama-sama mencapai tujuan tertentu. Dimana sistem harus memiliki organisasi, hubungan timbal balik, integrasi dan tujuan pokok”.

Sedangkan menurut Mulyadi (2016:4) “Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan”

2.2.4 Aplikasi

Perangkat lunak aplikasi (*software application*) adalah suatu subkelas perangkat lunak computer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja dan .

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan [1]. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi dua yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org* yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi.

Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran

yang akan dituju. Menurut kamus computer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”.

2.2.5 Aplikasi Mobile

Penggunaan perangkat *mobile* terus meningkat. Perangkat *mobile* seperti *smartphone* dan *tablet* telah menjadi sesuatu yang biasa di masyarakat. Hal yang menarik dalam penggunaan perangkat *mobile* itu sendiri adalah terdapatnya aplikasi *mobile*. Menurut Garhan(2015:3) aplikasi *mobile* sendiri adalah software yang yang dapat dijalankan pada perangkat genggam seperti *tablet*, *e-reader*, dan *smartphone* perangkat lunak aplikasi adalah pengolah kata, lembar kerja.

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi dua yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang

disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org* yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi.

2.2.6 Sistem Operasi *Mobile*

Berdasarkan pendapat Beal (2015:2) sistem operasi pada perangkat *mobile* memiliki peran penting agar aplikasi-aplikasi yang berada dalam perangkat yang dapat saling berkomunikasi atau bertukar data yang bekerja pada processor kecil yang ada pada *gadget*. Sebuah perangkat *mobile* yang tidak memiliki sistem operasi akan menyebabkan aplikasi-aplikasinya tidak berjalan dengan baik-baik.. Berikut ini adalah beberapa sistem operasi yang sudah umum, antara lain:

a. *Windows Mobile*

Windows Mobile adalah sistem operasi *mobile* yang beroperasi pada platform perangkat *Windows Mobile*.

b. *IOS*

IOS adalah sistem operasi keluaran dari Apple yang digunakan dalam semua lini gadgetnya yaitu *Iphone, Ipad, & Ipod*.

c. *Android*

Android adalah salah satu dari sistem operasi *open source* yang terkenal di kalangan masyarakat. Desain dari sistem operasi *Android* bisa

disesuaikan dengan *operator* dan produsen perangkat *mobile* yang ingin menggunakannya.

d. *Blackberry*

Blackberry merupakan sistem operasi besutan *RIM* yang populer pada tahun 2008 dengan aplikasi *chat blackberry messenger*.

2.2.7 Jenis Aplikasi *Mobile*

Orang-orang awam sering kali mempunyai pemahaman yang salah antar kategori dan jenis aplikasi *mobile*. Mereka menganggap itu adalah dua hal yang sama, padahal kategori dan jenis aplikasi *mobile* mempunyai pengertian yang berbeda. Yang dimaksud dengan kategori aplikasi *mobile* menurut menurut Budi (2015:12) adalah sebagai berikut:

a. *Native Apps*

Merupakan aplikasi yang dikembangkan dan digunakan pada sistem operasi tertentu. Kelebihan dari jenis aplikasi ini adalah kecepatan, kinerja dan antarmuka dari aplikasi paling optimal di antara semua jenis aplikasi. Selain itu, jenis aplikasi ini tidak membutuhkan koneksi internet. Namun, kelemahan dari jenis aplikasi ini adalah biaya pengembangan jenis aplikasi ini mahal karena hanya bisa berfungsi pada satu sistem operasi serta proses pengembangan memakan waktu yang lama.

b. *Web-Based Apps*

Aplikasi ini menggunakan *web browser* sebagai *client* yang dikembangkan dengan bahasa pemrograman yang dapat diolah dan dibaca oleh

web browser seperti *HTML*, *PHP*, *JavaScript* dan lainnya. Kelebihan dari jenis aplikasi ini adalah biaya pengembangan yang murah, dapat diakses oleh semua perangkat yang memiliki fitur *web browser* dan penggunaan *database* yang kecil karena semua data tersimpan di *server*. Kelemahannya adalah jenis aplikasi ini membutuhkan koneksi internet agar aplikasi dapat digunakan. Kelemahannya yang lain adalah apabila koneksi internet yang lemah atau tidak baik, akan menyebabkan mengurangi kinerja aplikasi.

c. Hybrid Apps

Hybrid Apps adalah jenis aplikasi yang relatif sama dengan *Native Apps*, hanya saja data-data di aplikasi tersimpan di *web*, bukan di perangkat. Kelebihannya adalah jenis aplikasi dapat diunduh melalui *app store*, selain lebih mudah untuk diterapkan pada sistem operasi yang berbeda disanding.

2.2.7 ATM (*Automated Teller Machine*)

ATM (*Automated teller machine*-anjungan tunai mandiri) adalah suatu produk Bank yang diberikan kepada nasabah simpanan pihak ketiga yaitu pemilik rekening tabungan atau rekening rekening lainnya yang ditentukan oleh bank atas nama pribadi bukan atas nama lembaga.

ATM (*Automated Teller Machine*) adalah mesin penghitung otomatis yang merupakan suatu produk Bank yang bertujuan mempermudah proses transaksi keuangan. (Stephen Liesty: Praktisi Perbankan)

ATM (*Automated Teller Machine*-Anjungan Tunai Mandiri) adalah semacam kartu yang diberikan kepada nasabah yang diterbitkan oleh kantor pusat Bank

tertentu melalui *Card Center* kantor pusat atau kantor cabang pada suatu lembaga keuangan Bank tersebut. ATM atau nama lainnya (*Technologi Asynchronous Transfer Mode*) saat ini memasuki operasional pelayanan secara penuh dan merupakan suatu teknologi yang menjadi dasar pembuatan jaringan-jaringan baru.

ATM (*Aotomated Teller Machine*) merupakan salah satu jasa Bank yang berfungsi sebagai perantara melakukan transaksi yang dilakukan oleh nasabah dengan petugas perbankan (*Teller*) yang bertujuan mencapai kemudahan dalam bertransaksi.

ATM (*Automated Teller Machine*) di Indonesia juga kadang singkatan dari (Anjungan Tunai Mandiri) adalah sebuah alat elektronik yang mengizinkan nasabah Bank untuk mengambil uang dan mengecek rekening tabungan mereka tanpa perlu dilayani oleh seorang "Teller" manusia. Banyak ATM juga mengizinkan penyimpanan uang atau cek, transfer uang atau bahkan membeli perangkat. (Wikipedia Indonesia: 2018)

ATM (*Automated Teller Machine*) sering disebut sebagai produk elektronik Bank yang berupa mesin ATM yang memiliki berbagai kegunaan dalam transaksi keuangan, pemerintah menetapkan undang-undang perbankan yang mengatur tentang ATM beserta kegunaannya dalam Undsang-Undang Perbankan Bab 1 pasal 1, dalam ketentuan umum Perbankan.

Anjungan tunai mandiri (ATM) adalah kegiatan kas yang dilakukan secara elektronik untuk memudahkan nasabah antara lain dalam menarik atau menyetor secara tunai atau melakukan pembayaran melalui pemindahbukuan dan

memperoleh informasi mengenai saldo, mutasi rekening nasabah.(Peraturan Bank Indonesia tentang Bank Umum) .

2.2.8 Algoritma Dijkstra

Algoritma *Dijkstra* dinamai sesuai dengan nama penemunya yaitu Edsger Dijkstra. Algoritma *Dijkstra* menggunakan prinsip greedy, dimana pada setiap langkah dipilih sisi dengan bobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma *Dijkstra* adalah suatu algoritma untuk menemukan jarak terpendek pada tiap lokasi (*vertices*) pada suatu *graph*. Algoritma *Dijkstra* ditemukan oleh seorang ilmuwan komputer berkebangsaan Belanda bernama Edsger Wybe Dijkstra. Dimana metode Algoritma *Dijkstra* ini berfungsi untuk mencari jarak atau rute terpendek dari satu lokasi ke lokasi tujuan. Algoritma *Dijkstra* merupakan metode yang paling efisien untuk menentukan lintasan jalur terpendek antara titik satu dengan titik lainnya. Algoritma *Dijkstra* dapat memecahkan masalah pencarian jalan atau rute terpendek dari satu titik dalam graf (awal) ke semua titik tujuan dalam waktu yang sama, sehingga permasalahan ini disebut permasalahan sumber tunggal pencarian jalan terpendek. Metode *Dijkstra* menurut pakar komputer dan matematika dalam penanganannya adalah sebagai berikut

- a. Metode Algoritma *Dijkstra* adalah suatu metode yang namanya sesuai dengan nama penemunya yaitu Edsger Dijkstra, dimana metode ini digunakan untuk memecahkan masalah pencarian jalur terpendek antara

tiap lokasi yang dihubungkan oleh jalur, dimana jarak antara jalur bukan nilai negatif.

- b. Metode *Dijkstra* adalah suatu algoritma untuk menemukan jarak terpendek pada tiap vertice (lokasi) pada suatu graph (Diana Okta Pugas, al at, 2015: 4).

Algoritma *Dijkstra* bekerja dengan membuat jalur ke satu simpul optimal pada setiap langkah. Jadi pada langkah ke n , setidaknya ada N_{node} yang sudah kita tahu jalur terpendek. Langkah-langkah algoritma *Dijkstra* dapat dilakukan dengan langkah-langkah berikut:

- a. Tentukan titik mana yang akan menjadi *node* awal, lalu beri bobot jarak pada *node* pertama ke *node* terdekat satu per satu, *Dijkstra* akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap.
- b. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada *node* awal dan nilai tak hingga terhadap *node* lain (belum terisi).
- c. Set semua *node* yang belum dilalui dan set *node* awal sebagai “*Node* keberangkatan”
- d. Dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum dilalui dan hitung jaraknya dari titik keberangkatan. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru
- e. Saat kita selesai mempertimbangkan setiap jarak terhadap *node* tetangga, tandai *node* yang telah dilalui sebagai “*Node* dilewati”. *Node* yang

dilewati tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.

- f. Set “*Node* belum dilewati” dengan jarak terkecil (dari *node* keberangkatan) sebagai “*Node* Keberangkatan” selanjutnya dan ulangi langkah e.

2.2.9 Google Maps API

Google Maps API adalah layanan pemetaan berbasis web dan teknologi yang telah disediakan oleh *Google*, bersifat *free* (tidak untuk penggunaan komersial), yang memiliki kemampuan terhadap banyak layanan pemetaan berbasis web (Sofyan;2015:7). *Google* menyediakan *API* dalam 2 bahasa pemrograman, yaitu *JavaScript* dan *ActionScript*. Dengan adanya *Google Map API*, memungkinkan untuk menambahkan peta berbasis *Google Map* secara *custom* (sesuai keinginan) pada suatu *website* dan aplikasi *android*.

Google map menyediakan elemen-elemen yang dapat digunakan untuk mengkustomisasi peta yang ingin ditampilkan. Elemen-elemen tersebut antara lain adalah *marker*, *info window*, dan *icon*. *Marker* untuk menandai objek dalam peta. Pada penelitian ini *marker* digunakan untuk menandai posisi calon penumpang dan sopir angkot berdasarkan informasi *longitude* dan *latitude* yang didapatkan dari *GPS* pada *mobile phone*. *Info window* merupakan informasi yang dapat ditambahkan pada *marker*. Informasi tersebut akan muncul ketika *marker* diklik. *Icon* merupakan lambang sederhana yang digunakan sebagai *marker*.

2.2.10 Android

Pengertian *Android* menurut (Safaat, 2015) *Android* adalah sebuah sistem operasi *linux* untuk *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Sistem operasi yang mendasari *Android* dilisensikan dibawah *GNU, General Public Lisence Versi 2(GPL2v2)*, yang sering dikenal dengan istilah “*copyleft*” lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh dibawah terms. *Android* didistribusikan dibawah *lisensi Apache Sostware (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua dan seterusnya.

Antar muka pengguna *Android* didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. *Android* adalah sistem operasi dengan sumber terbuka, dan *Googler* merilis kodenya dibawah *Lisensi Apache*. Kode dengan sumber terbuka dan lisensi perizinan pada *Android* memungkinkan perangkat lunak untuk di modifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, *operator nirkabel*, dan pengembang aplikasi. Selain itu, *Android* memiliki sejumlah besar komunitas pengembang aplikasi (*apps*) yang memperluas *fungsionalitas* perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman *Java*. Pada bulan oktober 2012, ada sekitar 700.000 aplikasi yang tersedia untuk *Android*, dan sekitas 25 juta aplikasi telah di unduh dari

Google Play, toko aplikasi utama *Android*. Sebuah *survey* pada bulan April-Mei 2013 menemukan bahwa *Android* adalah *platform* paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi seluler.

Faktor-faktor diatas telah memberikan kontribusi terhadap perkembangan *Android*, menjadikan sebagai sistem operasi telepon pintar yang paling banyak digunakan didunia, mengalahkan *Symbian* pada tahun 2010. *Android* juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikostumisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Akibatnya, meskipun pada awalnya sistem operasi ini dirancang khusus untuk telepon pintar dan tablet, *Android* jugak dikembangkan menjadi aplikasi tambahan di televisi, konsol permainan, kamera digital, dan perangkat elektronik lainnya. Sifat *Android* yang terbuka telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menamnah fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan *Android* pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Pada November 2013, *Android* menguasai pasar telepon pintar global, yang dipimpin oleh produk-produk Samsung, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat *Android* berbeda dengan beragam versi. Keberhasilan sistem operasi ini jugak menjasikan sebagai target litigasi paten “perang telepon pintar” antar perusahaan-perusahaan teknologi. Hingga bulan Mei 2013 total 900 juta perangkat *Android* telah diaktifkan

diseluruh dunia, dan 48 miliar aplikasi telah di pasang dari *Google Play*. Pada tanggal 3 September 2013, 1 miliar perangkat *Android* telah di aktifkan.

2.2.11 Sejarah Android

Android adalah sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari

34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

2.2.12 Penemu Android

Andy Rubin lahir pada tanggal 22 Juni 1946 di New Bedford, Amerika Serikat. Dia dikenal sebagai Penemu dari Android OS yang merupakan sistem operasi yang banyak digunakan di smartphone saat ini. Sejak kecil, Rubin sudah terbiasa melihat banyak gadget baru. Ini karena ayahnya, seorang psikolog yang

banting setir ke bisnis direct marketing, menyimpan produk elektronik yang akan dijualnya di kamar Rubin. Ia memiliki minat besar pada segala hal yang berbau robot. Di Carl Zeiss A.G., tempat pertama kali ia bekerja setelah lulus kuliah, Rubin ditempatkan di sebuah divisi robotika, tepatnya pada komunikasi digital antara jaringan dengan perangkat pengukuran dan manufaktur. Setelah dari Carl Zeiss, ia sempat bekerja di bidang robot di sebuah perusahaan di Swiss.

Karier Rubin di bidang robotika nampaknya semakin cerah, namun hidupnya berubah gara-gara liburan di Cayman Island pada tahun 1989. Saat sedang mengunjungi kepulauan tropis di Jamaika itu, Rubin tak sengaja bertemu dengan seorang bernama Bill Caswell. Pria ini sedang tidur di tepi pantai, terusir dari sebuah cottage setelah bertengkar dengan pacarnya. Andy menawarkan pria itu tempat tinggal dan sebagai balas budi, Casswell menawarkannya pekerjaan. Kebetulan yang menakjubkannya adalah pria itu bekerja di Apple. Di Apple, Rubin mengalami masa-masa yang menyenangkan. Pada saat itu, Apple masih dalam kondisi baik berkat komputer Macintosh. Budaya Apple pun menular pada diri Rubin. Di sana ia sempat melakukan kejahilan, seperti memprogram ulang sistem telepon sehingga ia bisa berpura-pura sebagai sang CEO, John Sculley. Lelucon seperti itu mungkin akan disukai Steve Jobs, pria yang gemar membuat lelucon lewat telepon, namun ketika itu adalah periode Apple tanpa Jobs.

Dari bagian manufaktur, Rubin pindah ke bagian riset di Apple. Kemudian, pada tahun 1990, Apple melakukan spin off untuk membentuk sebuah perusahaan bernama General Magic dan Rubin ikut di dalamnya. General Magic berfokus pada pengembangan perangkat genggam dan komunikasi. Para engineer yang gila

kerja, termasuk Rubin tentunya, berhasil mengembangkan sebuah peranti lunak bernama Magic Cap. Sayangnya, Magic Cap tidak mendapat sambutan dari perusahaan handset dan telekomunikasi. Beberapa yang menerapkan Magic Cap hanya melakukannya sebentar. General Magic pun akhirnya hancur.

Beberapa pengembang di General Magic, bersama beberapa veteran Apple, kemudian mendirikan Artemis Research. Perusahaan ini mengembangkan sesuatu bernama webTV, sebuah upaya awal untuk menggabungkan Internet dengan televisi. Rubin bergabung dengan Artemis untuk ikut mengembangkan webTV tersebut. Saat Microsoft membeli Artemis, di 1997, Rubin pun ikut bergabung dengan perusahaan raksasa itu. Episode gila khas Rubin kembali terjadi di Microsoft. Rubin membangun sebuah robot yang dilengkapi kamera untuk mengerjai rekan-rekannya. Gilanya, robot itu terhubung ke Internet dan pada satu insiden sempat dibobol oleh pihak di luar Microsoft. Pada tahun 1999, Rubin keluar dari webTV (dan artinya, ia tak lagi menjadi karyawan Microsoft). Ia kemudian menyewa sebuah toko di Palo Alto, California, dan menyebut toko itu sebagai laboratorium. Beberapa pengembang di General Magic, bersama beberapa veteran Apple, kemudian mendirikan Artemis Research. Perusahaan ini mengembangkan sesuatu bernama webTV, sebuah upaya awal untuk menggabungkan Internet dengan televisi. Rubin bergabung dengan Artemis untuk ikut mengembangkan webTV tersebut. Saat Microsoft membeli Artemis, di 1997, Rubin pun ikut bergabung dengan perusahaan raksasa itu. Episode gila khas Rubin kembali terjadi di Microsoft. Rubin membangun sebuah robot yang dilengkapi kamera untuk mengerjai rekan-rekannya. Gilanya, robot itu terhubung ke Internet dan pada satu insiden sempat dibobol oleh pihak di luar Microsoft. Pada tahun 1999, Rubin keluar dari webTV (dan

artinya, ia tak lagi menjadi karyawan Microsoft). Ia kemudian menyewa sebuah toko di Palo Alto, California, dan menyebut toko itu sebagai laboratorium.

Munculnya Sistem Operasi Android buatan Andy Rubin Pada awal tahun 2002, Rubin sempat memberikan sebuah kuliah di Stanford mengenai pengembangan Sidekick. Karena, meski penjualan Sidekick di pasaran tak meledak, perangkat itu dinilai cukup baik dari sisi engineering. Sebuah kebetulan bahwa Larry Page dan Sergey Brin, pendiri Google, ikut hadir dalam kuliah tersebut. Selepas kuliah, Page menemui Rubin untuk melihat Sidekick dari dekat. Rupanya, Page melihat, perangkat itu menggunakan search engine Google. "Keren," ujar Page. Ini adalah sebuah titik tolak bagi Page untuk sebuah ide yang dalam beberapa tahun kemudian akan terwujud, sebuah ponsel Google. Kurang lebih dua tahun setelah itu, Rubin telah meninggalkan Danger dan mencoba melakukan hal-hal baru. Termasuk di antaranya mencoba memasuki bisnis kamera digital sebelum akhirnya ia mendirikan Android.

2.2.13 Jenis Android

1. Android Apple Pie atau Alpha v 1.0

Jenis Android pertama ialah Apple Pie atau dikenal pula dengan Android Alpha. Untuk jenis Android ini terdapat dua versi yaitu yang dirilis oleh Google dan juga OHA. Jenis *Android Alpha* ini pertama kali diperkenalkan pada 23 September 2008. Fitur yang dimiliki pada Android ini ialah play store, web browser, kamera, sinkronisasi antara Gmail, kontak dan juga Google Agenda. Bukan hanya itu saja, akan tetapi pada versi *Android Alpha* ini pun telah dilengkapi dengan Maps serta streaming YouTube.

2. Android Banana Bread atau Beta v 1.1

Jenis *Android* ini dirilis pada bulan Februari 2009. Fitur yang dimiliki oleh *Android Beta* ini pun tidak jauh berbeda dengan *Android Alpha*.

Namun *Android* ini pun telah digunakan pada sebuah ponsel. *HTC* menjadi salah satu ponsel pertama yang menggunakan sistem operasi *Android Beta* ini.

3. Android Cup Cake v 1.5

Android Cup Cake ini merupakan jenis *Android* yang mulai terkenal. Hal ini dikarenakan versi dari *Android* ini mulai menunjukkan sistem operasi yang canggih.

4. Android Donut v 1.6

Setelah sebelumnya Google menamai produk sistem operasi dengan nama makanan, rupanya hal ini pun berlanjut. Pada perilisan di bulan September 2009, versi terbaru *Android* pun muncul kembali, yaitu dengan nama *Android Donut*. Jenis *Android* ini pun mulai bisa memperbaiki masalah atau bug. Selain itu, pada jenis *Android Donut* ini memiliki fitur tambahan berupa fitur navigasi Turn by Turn.

5. Android Eclair v 2.0 - 2.1

Android versi 2.0 ini pun dirilis pada Oktober 2009 dengan nama *Eclair*. Versi *Android Eclair* ini pun awal dari diterapkannya aplikasi Bluetooth 2.1 pada *Android*.

Tak berselang lama, versi 2.0 pun di update ke versi 2.1 yang tentu saja memiliki fitur tambahan lain. Fitur tambahan pada versi ini ialah Multi Touch, Live Wallpaper serta flash kamera.

6. Android Frozen Yogurt atau Froyo v 2.2

Jenis Android dari Froyo v 2.2 ini mulai dirilis pada bulan Mei 2010. Tentu saja saat update Android versi terbaru, ada pula fitur tambahan agar Android ini semakin diminati. Pada Android Froyo ini pada Google Chrome di Android terdapat tambahan Script Chrome. Hal ini dilakukan untuk menambah kecepatan kinerja pada Google Chrome.

7. Android Gingerbread v 2.3

Masih ditahun yang sama dengan perilsan Android Froyo, Android Gingerbread pun dirilis pada bulan Desember 2010. Pada Android versi 2.3 ini ada pula fitur NFC, internet calling dan juga download manager yang diterapkan pada sistem operasi Android Gingerbread.

8. Android Honeycomb v 3.0 - 3.2

Selanjutnya ialah sistem Android Honeycomb. Android Honeycomb ini resmi dirilis pada Februari 2011. Selain itu, jenis Android ini pun banyak digunakan pada tablet yang saat itu banyak diminati.

9. Android Ice Cream Sandwich v 4.0

Jenis Android Ice Cream Sandwich ini bisa dikatakan sebagai awal reformasi dari sistem operasi Android. Pasalnya sistem operasi versi 4.0 ini bisa digunakan pada semua jenis platform dan juga smartphone. Selain itu, fitur dari sistem operasi ini pun semakin bertambah dan juga menjadi lebih multitasking.

2.2.14 Bahasa Pemrograman

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer.

Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaksis dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang *programmer* dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari:

- a. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110.
- b. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah.Ingggris *Assembly*), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode *mnemonic*), contohnya *MOV, SUB, CMP, JMP, JGE, JL, LOOP*, dsb.
- c. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya *{, }, ?, <<, >>, &&, ||*, dsb.
- d. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia, contohnya *begin, end, if, for*,

while, and, or, dsb. Komputer dapat mengerti bahasa manusia itu diperlukan program *compiler* atau *interpreter*.

Sebagian besar bahasa pemrograman digolongkan sebagai Bahasa Tingkat Tinggi, hanya bahasa *C* yang digolongkan sebagai Bahasa Tingkat Menengah dan *Assembly* yang merupakan Bahasa Tingkat Rendah.

Bahasa Pemrograman (*programming language*) adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaksis dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang *programmer* dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program atau aplikasi. Contohnya adalah program yang digunakan oleh kasir di mal-mal atau swalayan, penggunaan lampu lalu lintas di jalan raya.

Bahasa Pemrograman yang kita kenal ada banyak sekali di belahan dunia, tentang ilmu komputer dan teknologi dewasa ini. Perkembangannya mengikuti tingginya inovasi yang dilakukan dalam dunia teknologi. Contoh bahasa pemrograman yang kita kenal antara lain adalah untuk membuat aplikasi *game*, antivirus, *web*, dan teknologi lainnya.

Bahasa pemrograman komputer yang kita kenal antara lain adalah *Java, Visual Basic, C++, C, Cobol, PHP, .NET*, dan ratusan bahasa lainnya. Namun tentu saja

kebutuhan bahasa ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya.

Secara umum bahasa pemrograman terbagi menjadi 4 kelompok, yaitu :

- a. *Object Oriented Language (Java, PHP, Javascript, C++)*
- b. *High Level Language (seperti Pascal dan Basic)*
- c. *Middle Level Language (seperti bahasa C)*
- d. *Low Level Language (seperti bahasa Assembly)*

Sedangkan menurut tingkatannya, bahasa pemrograman memiliki 3 jenis tingkatan yaitu :

- a. Bahasa tingkat tinggi :Bahasa pemrograman masuk tingkat ini karena bahasa tersebut mendekati bahasa manusia. Contohnya bahasa *Basic, Visual Basic, Pascal, Java, PHP*.
- b. Bahasa tingkat menengah :Disebut tingkat menengah karena bisa masuk ke dalam bahasa tingkat tinggi maupun rendah. Contohnya bahasa *C*.
- c. Bahasa tingkat rendah :Bahasa pemrograman masuk tingkat ini karena bahasanya masih jauh dari bahasa manusia. Contohnya bahasa *Assembly*.

2.2.15 Proses Pembuatan Suatu Program

Proses pembuatan program yaitu kita menulis kode sumber pada teks editor misalnya notepad kemudian mengubahnya menjadi bahasa mesin yang bisa

dieksekusi oleh *CPU*. Proses pengubahan kode sumber (*source code*) menjadi bahasa mesin (*machine language*) ini terdiri dari dua macam yaitu kompilasi dan interpretasi

a. Kompilasi

Dalam proses kompilasi semua kode sumber dibaca terlebih dahulu dan jika tidak ada kesalahan dalam menulis program maka akan dibentuk kode mesinnya sehingga program bisa dijalankan. Program yang melakukan tugas ini disebut *Compiler*. Program hasil kompilasi akan berbentuk *executable*. Program bisa langsung dijalankan tanpa harus memiliki *Compiler* di komputer yang menjalankan program tersebut. Bahasa yang menggunakan teknik kompilasi misalnya bahasa *C*, *C++*, *Pascal*, *Assembly* dan masih banyak lagi.

b. Interpretasi

Bahasa yang menggunakan teknik interpretasi akan membaca kode sumber perbaris dan dieksekusi perbaris. Jika ditemukan kesalahan dalam penulisan program maka di baris kesalahan itulah program akan dihentikan. Program yang melakukan tugas ini disebut *Interpreter*. Pada teknik interpretasi tidak akan dihasilkan program *standalone*, artinya untuk menjalankan program kita harus mempunyai kode sumbernya sekaligus *interpreter* program tersebut. Bahasa yang menggunakan teknik interpretasi misalnya bahasa *Perl*, *Python*, *Ruby*, *Java* dan masih banyak lagi.

2.2.16 Java

Java adalah suatu bahasa pemrograman yang bersifat *object-oriented*, memiliki fitur yang lengkap dan dapat digunakan untuk membuat aplikasi pada berbagai perangkat lunak di semua *platform* seperti *server*, *desktop*, dan *mobile*, (Liang 2014:12). Untuk memodifikasi data, dapat dibuat dari data yang sudah dimiliki sebelumnya, tidak perlu memulai segala sesuatu dari awal setiap kali melakukan sesuatu pekerjaan yang baru. Ada tiga edisi Java, yaitu :

- a. *Java Standard Edition (Java SE)*
- b. *Java Enterprise Edition (Java EE)*
- c. *Java Micro Edition (Java ME)*

2.3 Sederhana

Java memiliki fitur-fitur singkat yang membuatnya mudah untuk dipelajari dan digunakan.

2.4 Aman

Java menyediakan *firewall* antara aplikasi jaringan dengan komputer.

2.5 Portable

Java dapat dijalankan di berbagai *platform* tanpa perlu di-*compile* ulang.

2.6 Object-oriented

Karena *Java* adalah bahasa pemrograman *object-oriented*, maka hal ini memudahkan dalam mendesain, membuat, mengembangkan dan mengalokasikan kesalahan dalam program secara cepat, mudah dan terorganisir.

2.7 Multithreaded

Memungkinkan untuk menjalankan beberapa aktivitas di dalam Java.

2.2.17 Sejarah perkembangan Java

Bahasa pemrograman Java terlahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, dan James Gosling, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road* di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program *Java Oak* pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (*Star Seven*).

Setelah era *Star Seven* selesai, sebuah anak perusahaan Tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 *Hamilton Avenue, Palo Alto*.

Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinspirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995.

Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace. Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java).

2.2.18 Android Studio

Android Studio adalah lingkungan pengembangan *Android* berdasarkan *IntelliJIDEA*. Mirip dengan *Eclipse* dengan *ADT Plugin*, *Android Studio* menyediakan alat pengembang *Android* terintegrasi untuk pengembangan dan

debugging. *Android Lab* merupakan platform editor yang berbasis *Gradle*, dengan *Androidlab* proses *refactoring* dan perbaikan berlangsung dengan cepat, hal tersebut dapat dilakukan karena ada *tool* bernama Lint untuk menangkap kinerja, kegunaan, kompatibilitas versi *Android* dan masalah lainnya. *Android Lab* juga merupakan sebuah layout editor yang kaya akan fitur yang memungkinkan untuk *drag-and-drop* komponen *UI*.

2.2.19 UML (Unified Modelling Language)

Menurut Satriawaty Mallu (2015:38) *Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

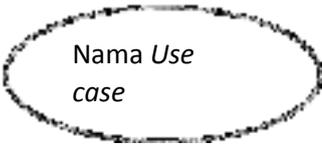
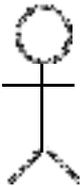
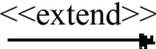
2.2.20 Use Case Diagram

Menurut Ade Hendini (2016:108) *Use 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.

Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel 2.1 Use Case Diagram

No	Simbol	Deskripsi
1	<p><i>Use case</i></p>  <p>Nama Use case</p>	Gambaran unit yang saling berkaitan antara aktor dengan sistem yang berjalan
2	<p>Aktor</p>  <p>Nama aktor</p>	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan dibuat.
3	<p>Asosiasi / <i>Association</i></p> 	Komunikasi antara aktor dan <i>use case</i>
4	<p>Ekstensi / <i>Extend</i></p>  <p><<extend>></p>	Kelakuanyang hanya berjalan dibawah kondisi tertentu. Seperti jika akun sesuai, atau jika <i>session</i> sesuai
5	<p>Generalisasi</p> 	Elemen yang menjadi spesialisasi elemen lain

6	<i>Include</i> <<include>> 	Kelakuan yang harus terpenuhi agar suatu <i>event</i> dapat terjadi
---	--	---

Sumber : Ade Hendini (2016:108)

2.2.21 Activity Diagram

Menurut Ade Hendini (2016:108) *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu :

Tabel 2.2 Simbol-Simbol Activity Diagram

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada aktivitas pilihan lebih dari satu.
4	Penggabungan / Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Tahap akhir dari proses sistem.

Sumber : Ade Hendini (2016:108)

2.2.22 Flowchart

Menurut Eka Iswandy (2015:73) *Flowchart* merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-

simbol yang disusun secara sistematis. Simbol-simbol yang digunakan untuk menggambarkan *flowchart* dapat dilihat pada tabel 3

Tabel 2.3 Flowchart

No	Simbol	Maksud	Simbol	Maksud
1		Terminal (<i>START</i> , <i>END</i>)		Titik sambungan pada halaman yang sama
2		<i>Input / Output</i>		Konektor pada halaman lain
3		Proses		<i>Call</i> (Memanggil subprogram)
4		<i>Decision</i> (<i>YES</i> , <i>NO</i>)		Dokumen
5		<i>Display</i>		<i>Stored Data</i>
6		Alur proses		<i>Preparation</i>

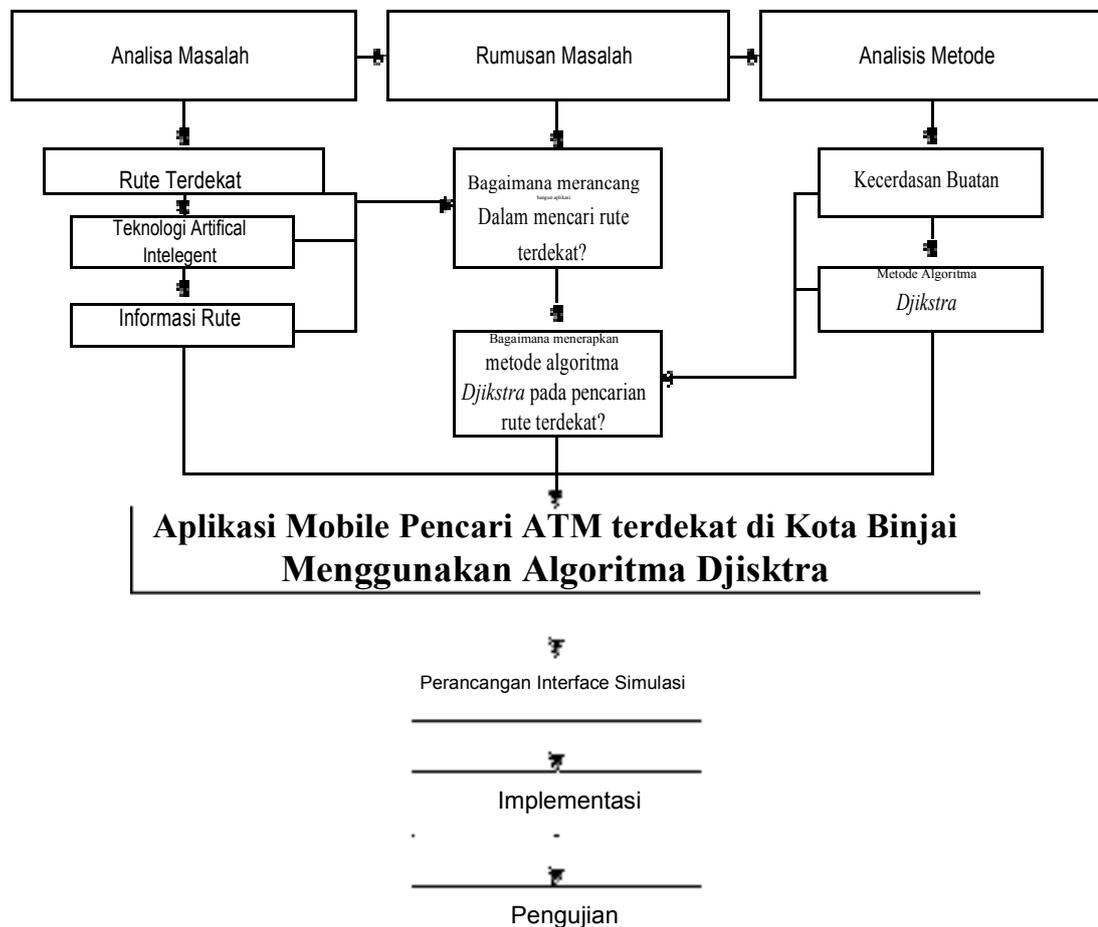
Sumber : Eka Iswandy (2015:73)

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Adapun tahapan penelitian dilakukan oleh penulis dengan judul Aplikasi mobile pencarian ATM terdekat di Kota Binjai Menggunakan Algoritma Dijkstra adalah sebagai berikut:



Gambar 3.1 Tahapan Penelitian

Keterangan Analisis Masalah:

1. Analisis Masalah berfungsi untuk menjelaskan masalah yang terjadi, yaitu masalah pencarian rute terdekat.
2. Teknologi AI adalah salah satu solusi penereapan algoritma dalam pencarian rute terdekat.
3. Informasi Rute berfungsi menghasilkan solusi dari teknologi AI.

Keterangan Rumusan Masalah:

1. Rumusan Masalah berfungsi untuk menjelaskan masalah yang terjadi, yaitu
 - a. bagaimana merancang aplikasi dalam pencarian rute terdekat?
 - b. Bagaimana menerapkan metode djisktra pada pencarian rute terpendek?
1. Kecerdasan buatan adalah metode untuk menciptakan aplikasi dalam pencarian rute terpendek.
2. Metode algoritma adalah sebuah algoritma yang menghitung jarak dari point a ke point selanjutnya agar mendapatkan rute terpendek dengan cepat.
4. Berdasarkan keterangan dari analisis masalah, rumusan masalah dan analisis metode, maka dapat disimpulkan sebuah skripsi yaitu dengan judul Aplikasi Mobile Pencari ATM terdekat di Kota Binjai Menggunakan Algoritma Djisktra.
5. Dalam pembuatan laporan skripsi hasil akhir dari skripsi tersebut adalah aplikasi simulasi yang menjadikan sebuah pembuktian untuk dijadikan solusi.

6. Setelah aplikasi sudah dibuat, maka selanjutnya proses yang dilakukan adalah implementasi.

3.2. Metode Pengumpulan Data

Pengumpulan data adalah pencarian terhadap sesuatu karena ada perhatian dan keinginan terhadap hasil suatu aktivitas. Metode pengumpulan data dalam penulisan ini dibagi menjadi 2, yaitu:

1. Pengamatan (*Observation*)

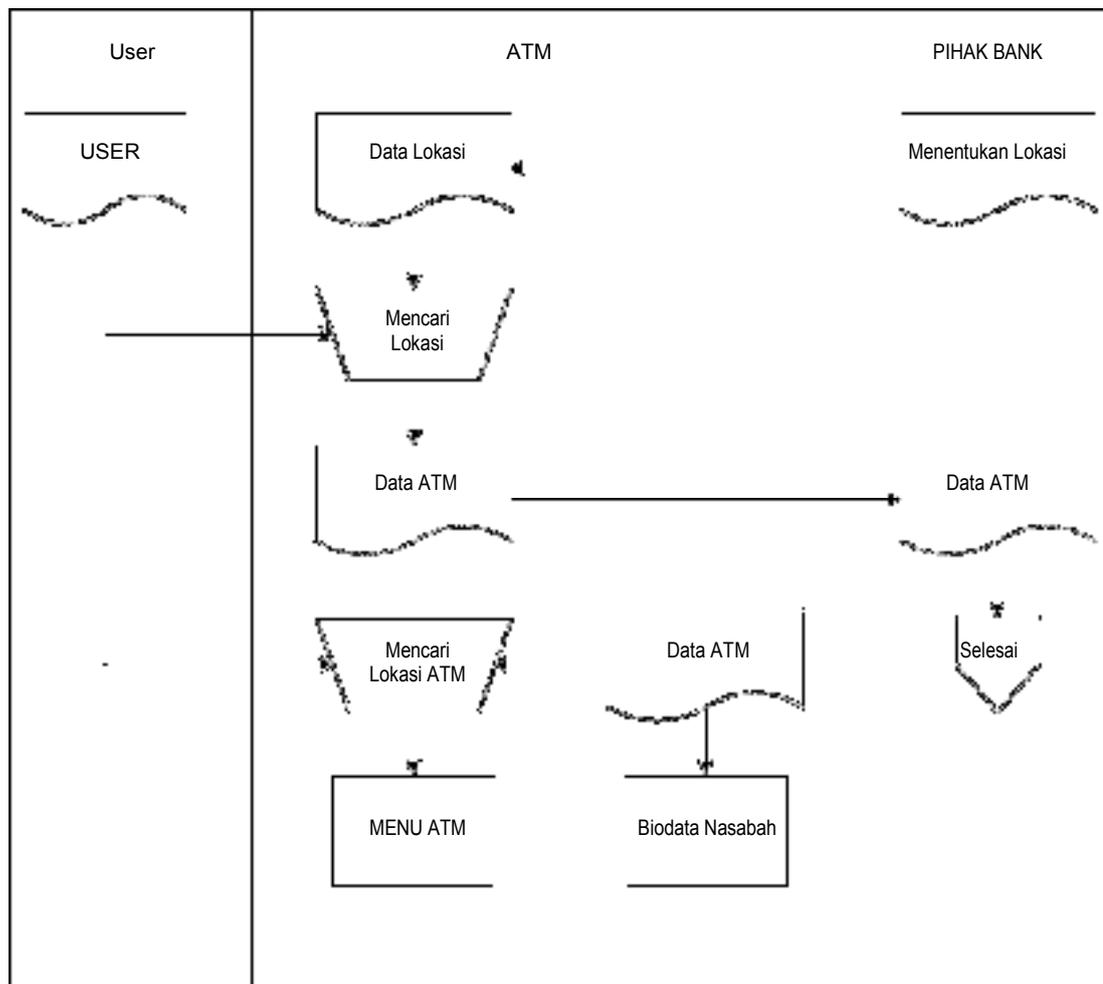
Penulis melakukan pengamatan langsung pada setiap lokasi ATM yang ada dikota Binjai.

2. Penelitian Kepustakaan (*Library Research*)

Merupakan cara untuk mencari referensi dengan mengumpulkan bahan-bahan pustaka yang dilakukan di perpustakaan kampus, maupun perpustakaan umum, juga melakukan pencarian lewat internet, dengan mengunjungi situs-situs seperti google Book online yang dapat membantu pembahasan materi.

3.3. Analisis Sistem Sedang Berjalan

Berikut ini adalah sistem FOD dari pencarian rute terpendek yang dilakukan untuk mencari lokasi ATM terdekat. Informasi-informasi mengenai tata letak atau kawasan ATM di BRI Binjai harus lebih informatif dan memudahkan dalam pencariannya. Oleh karena itu dibuat Sistem Informasi Geografis berbasis android mengenai daerah ATM di BRI Binjai.



Gambar 3.2 FOD Pencarian ATM Terdekat

Sistem yang sedang berjalan saat ini adalah, proses pencarian tempat wisata masih dengan cara sebagai berikut:

1. Penduduk datang ke lokasi lalu ATM yang pada umumnya terletak di pom bensin Pertamina.
2. Penduduk kebingungan dimana lokasi ATM berada didaerah Binjai.

3.3.1 Perhitungan Metode Dijkstra

Algoritma Dijkstra bekerja dengan membuat jalur ke satu simpul optimal pada setiap langkah. Jadi pada langkah ke n , setidaknya ada n node yang sudah kita tahu jalur terpendek. Langkah-langkah algoritma Dijkstra dapat dilakukan dengan langkah-langkah berikut:

1. Tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap.
2. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi).
3. Set semua node yang belum dilalui dan set node awal sebagai “Node keberangkatan”
4. Dari node keberangkatan, pertimbangkan node tetangga yang belum dilalui dan hitung jaraknya dari titik keberangkatan. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru
5. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah dilalui sebagai “Node dilewati”. Node yang dilewati tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.

6. Set “Node belum dilewati” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan ulangi langkah e.

Sebagai contoh hitunglah Jarak terdekat dari A ke H berikut ini.

Jumlah ATM	11
Start	Rumah
Stop	ATM

No	dari ↔ ke	jarak
A	Rumah ATM BRI (Supermarket, Jl. Medan Paya Geli)	1210 Meter
B	Rumah ATM BRI (Jl Medan Binjai, Krio Deli Serdang)	1141 Meter
C	Rumah ATM BRI (Bank BRI Jl. Medan Binjai Krio Deli Serdang)	4899 Meter 17 Meter
D	Rumah ATM BRI (Jl Pinang Baris, Medan Sunggal)	5149 Meter
E	Rumah ATM BRI (Jl. Setia No. 233 Tj Gusta)	4899 Meter
F	Rumah ATM BRI (Balai Pendidikan 8 Latihan, Jl, Tani Bonar Simatunpang)	4966 Meter
G	Rumah ATM BRI (Jl. Sei Mencirim, Sunggal Kanan)	C

H	Rumah	ATM BRI (Paya Geli, Deli Serdang Regency)	3029 Meter
I	Rumah	ATM BRI (Jl. Sei Mencirim No. 42A, Sunggal)	3233 Meter
J	Rumah	ATM BRI (RSU Lattersia, Jl. Soekarno-Hatta)	5011 Meter
K	Rumah	ATM BRI (Jl. Soekarno – Hatta No. 455, Tunggurono)	5007 Meter

Perhitungan dalam jarak terdekat:

Maka dapat disimpulkan:

Jarak Terpendek dari Rumah ke ATM	ATM BRI (Jl Medan Binjai, Krio Deli Serdang)
Rute Terpendek di tempuh Rumah - ATM	112711 Meter Meter

Graph ATM BRI (Paya Geli, Deli Serdang Regency) dengan matrik berhubungan langsung sebagai berikut ::

ATM BRI (Supermarket, Jl. Medan Paya Geli)	ATM BRI (Jl Medan Binjai, Krio Deli Serdang)	ATM BRI (Bank BRI Jl. Medan Binjai Krio Deli Serdang)	ATM BRI (Jl. Pinang Baris, Medan Sunggal)	ATM BRI (Jl. Setia No. 233 Tj Gusta)	ATM BRI (Jl. Sei Mencirim, Sunggal Kanan)	ATM BRI (Paya Geli, Deli Serdang Regency)
A0	4899 Meter Meter	112711 Meter Meter	∞	∞	∞	∞

B	4899 Meter 170	1210 Meter	1210 Meter	4899 Meter	∞	∞
C	112711 Meter	1210 Meter	∞	2711 Meter	∞	112711 Meter ∞
D	∞	1210 Meter	2711 Meter	0	4899 Meter17 Meter	1210 Meter ∞
E	∞	4899 Meter	∞	4899 Meter17 Meter	0	30 2711 Meter
F	∞	∞	112711 Meter Meter	1210 Meter	30	0 1210 Meter
G	∞	∞	∞	∞	2711 Meter	1210 Meter 0

PENYELESAIANNYA ::

Titik Vi	A	B	C	D	E	F	G
x (Vi)	∞	∞	∞	∞	∞	∞	0
T	A	B	C	D	E	F	G
Titik Vi	A	B	C	D	E	F	G
x (Vi)	∞	∞	∞	∞	2711 Meter	1210 Meter	0
T	A	B	C	D	E	F	-
Titik	A	B	C	D	E	F	G

Vi								
x (Vi)	∞	4899 Meter	17 Meter	∞	100	2711 Meter	1210 Meter	0
T	A	B	C	D	-	F	-	-
Titik Vi	A	B	C	D	E	F	G	
x (Vi)	∞	4899 Meter	17 Meter	14899 Meter	100	2711 Meter	1210 Meter	0
T	A	B	C	D	-	-	-	-
Titik Vi	A	B	C	D	E	F	G	
x (Vi)	14899 Meter	4899 Meter	17 Meter	110	100	2711 Meter	1210 Meter	0
T	A	-	C	D	-	-	-	-
Titik Vi	A	B	C	D	E	F	G	
x (Vi)	14899 Meter	4899 Meter	17 Meter	110	100	2711 Meter	1210 Meter	0
T	A	-	C	-	-	-	-	-
Titik Vi	A	B	C	D	E	F	G	

Vi							
x (Vi)	14899 Meter	4899 Meter	17 Meter	110	100	2711 Meter	1210 Meter
T	A	-	-	-	-	-	-
Titik	A	B	C	D	E	F	G
Vi							
x (Vi)	14899 Meter	4899 Meter	17 Meter	110	100	2711 Meter	1210 Meter
T	-	-	-	-	-	-	-

3.4 Rancangan Penelitian

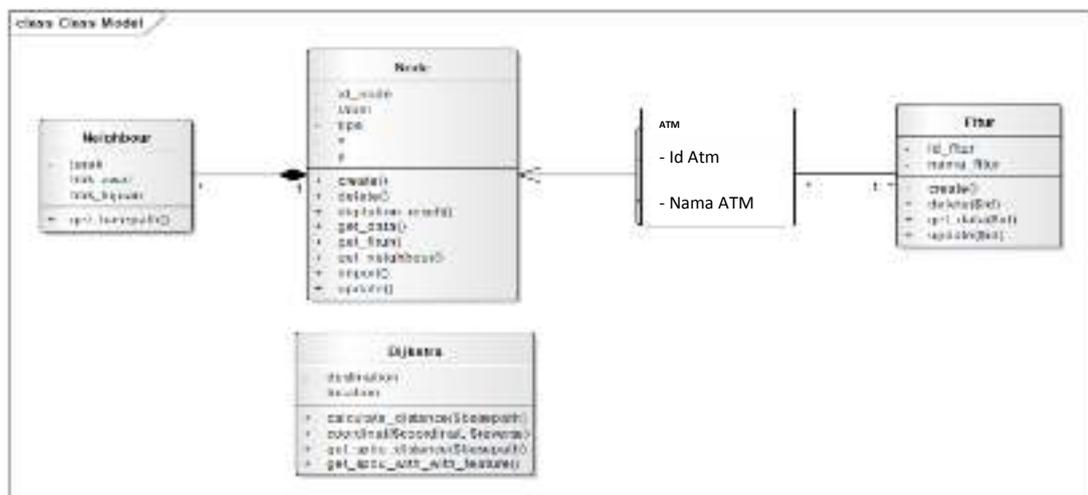
3.4.1 Rancangan Sistem Secara Global

1. Use Case Diagram Admin



Gambar 3.4 Use Case Sistem Aplikasi Mobile Pencarian *ATM* terdekat di Kota Binjai Berbasis Android dengan Penerapan *Algoritma Dijkstra*

2. Class Diagram

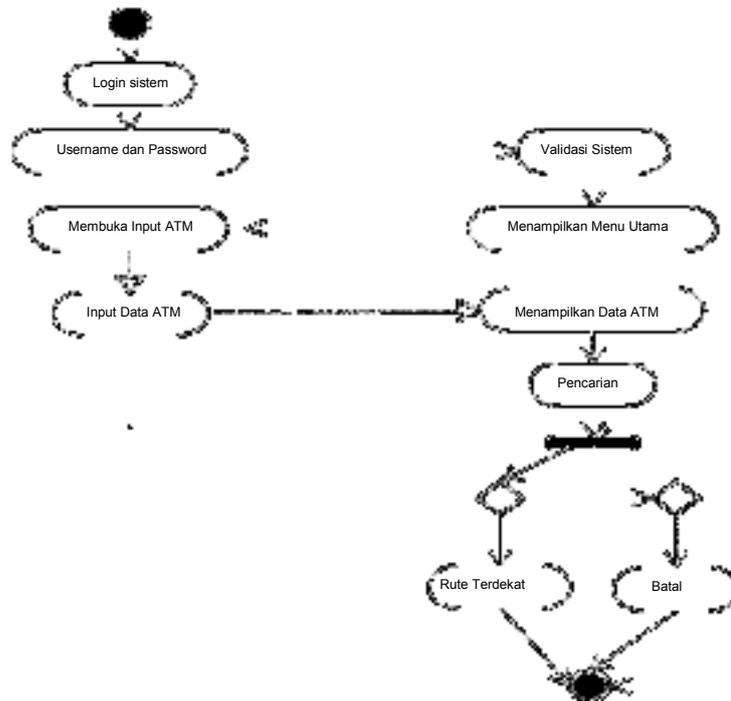


Gambar 3.6 Class Diagram

3.4.2 Rancangan Sistem Secara Detail

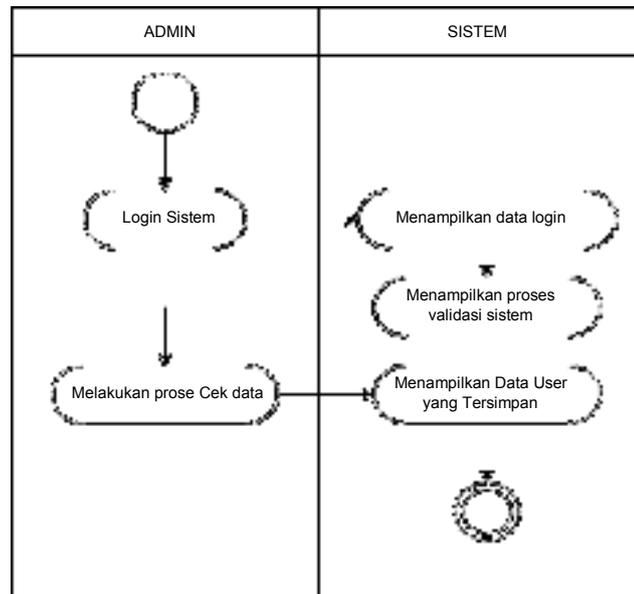
Berikut adalah rancangan sistem secara detail dari sistem pencarian rute terdekat:

3.4.2.1. Activity Diagram



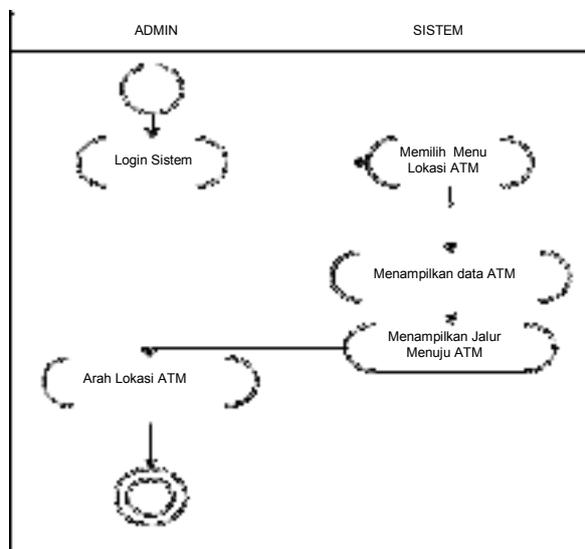
Gambar 3.7 Activity Sistem

3.4.2.2. Activity Diagram Login



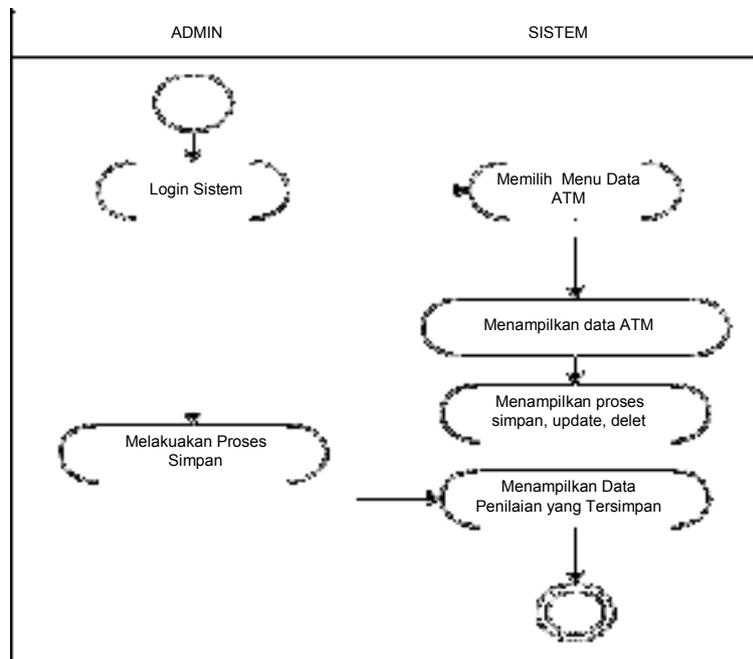
Gambar 3.8 Activity Sistem Login

3.4.2.3. Activity Diagram Mencari Lokasi ATM



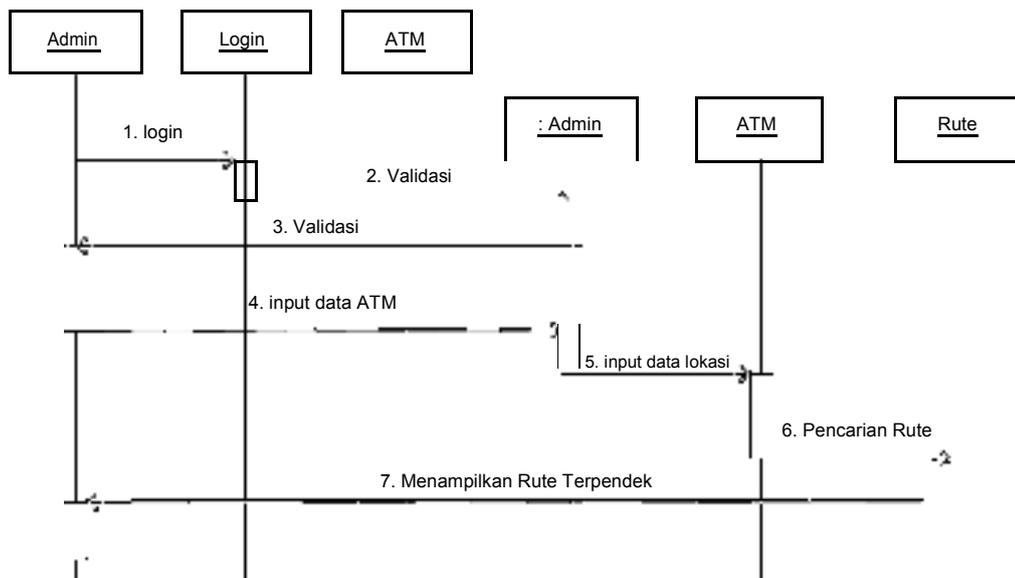
Gambar 3.9 Activity Sistem Mencari Lokasi ATM

3.4.2.4. Activity Diagram Data ATM



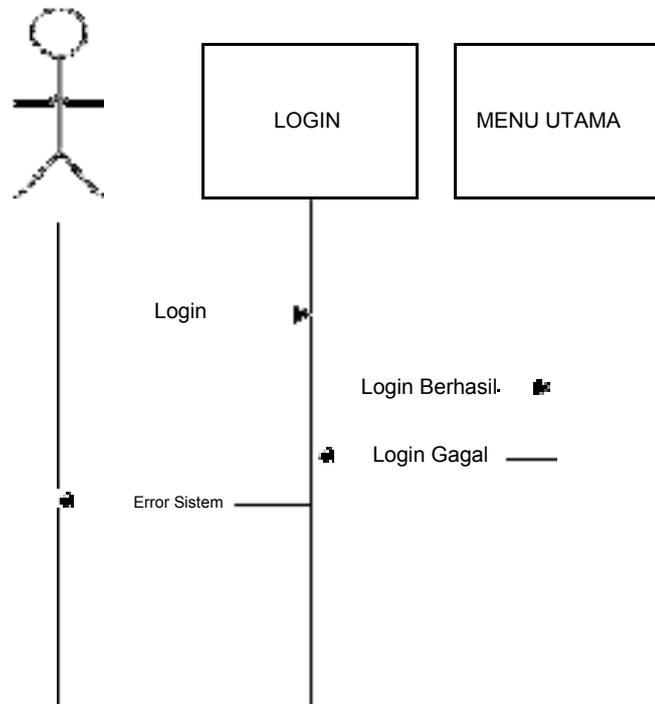
Gambar 3.10 Activity Sistem Data ATM

3.4.2.5. Sequence Diagram



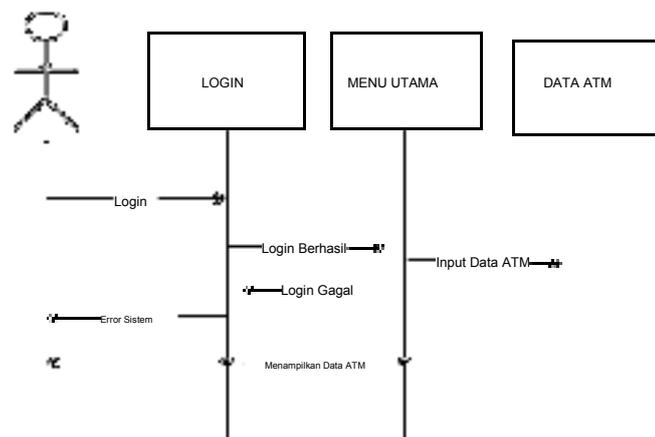
Gambar 3.11 Sequence Diagram

3.4.2.6. Sequence Diagram Login



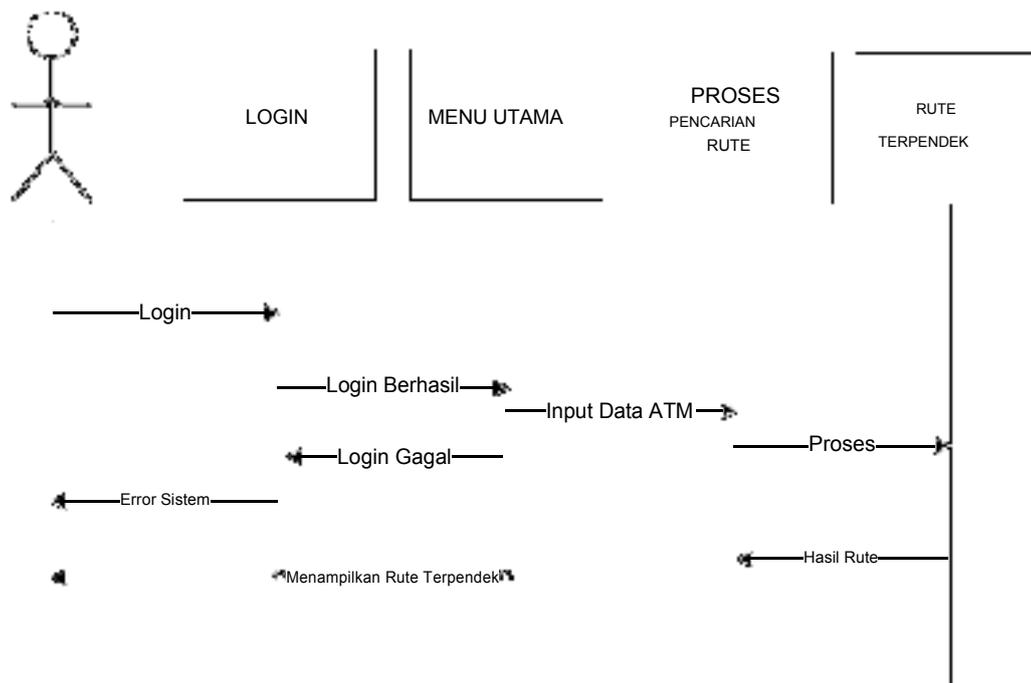
Gambar 3.12 Sequence Diagram Login

3.4.2.7. Sequence Diagram Data ATM



Gambar 3.13 Sequence Diagram Data ATM

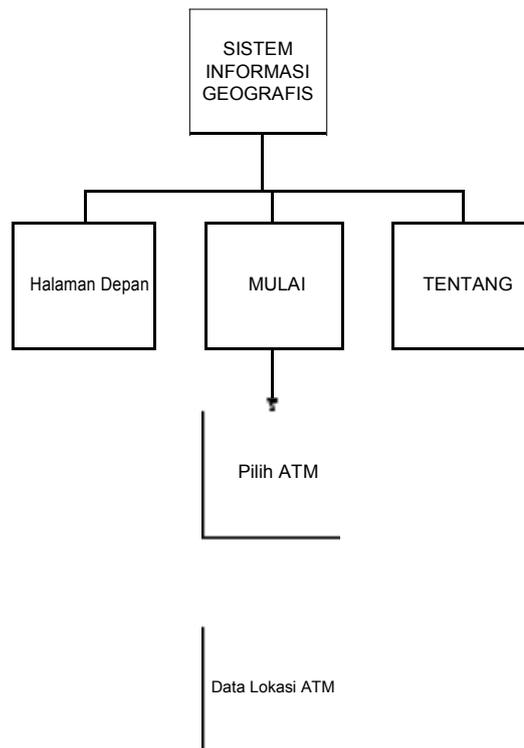
3.4.2.8. Sequence Diagram Mencari Lokasi ATM



Gambar 3.14 Sequence Diagram Mencari Rute Terpendek

3.4.4 Desain Sistem Secara Detail

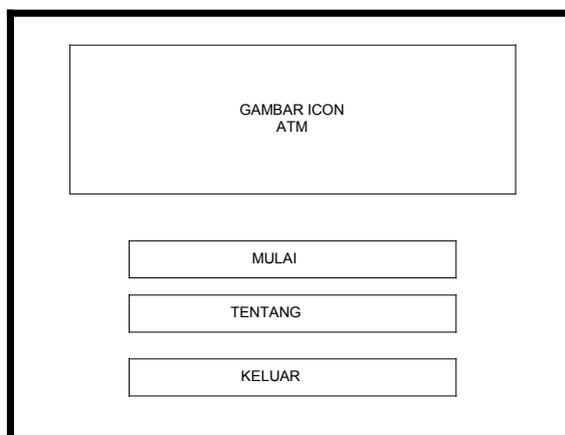
Sebelum dilakukannya proses perancangan tampilan pada Sistem Informasi Geografis Lokasi ATM di Kota Binjai, maka terlebih dahulu dilakukan flowmap pada perancangan sistem, adapun flowmap ada sistem informasi geografis Lokasi ATM di Kota Binjai adalah sebagai berikut:



Gambar 3.15 Rancangan FlowMap GIS Lokasi ATM di Kota Binjai

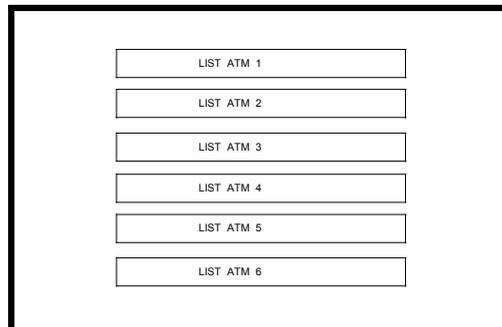
Perancangan merupakan bagian yang paling penting dalam merancang sistem. Adapun bentuk sistem informasi geografis Lokasi ATM di Kota Binjai adalah sebagai berikut:

1. Desain Form Halaman Depan



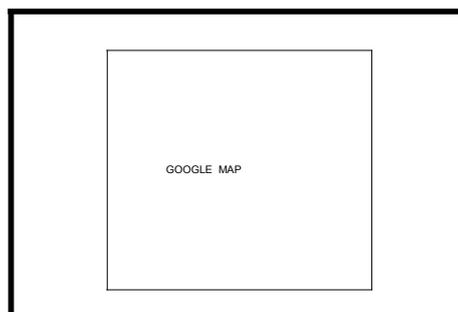
Gambar 3.16 Rancangan Halaman Depan

2. Desain Form Halaman Mulai



Gambar 3.17 Rancangan Halaman Mulai

3. Desain Form Halaman Lokasi ATM



Gambar 3.18 Rancangan Tampilan Halaman Lokasi ATM

4. Desain Form Halaman Tentang



Gambar 3.19 Rancangan Tampilan Halaman Tentang

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Menu yang terdapat didalam aplikasi ada berupa seorang admin dan *user* (pengguna). Sebelum mengaplikasikan aplikasi admin harus membuka Login, Menu yang dapat diaplikasikan oleh admin adalah *Menu Utama*, Mulai, Tentang Aplikasi, Keluar. Sedangkan *user* dapat mengaplikasikan jika si *user* memiliki *username* dan *password* untuk membuka Login yang diberikan oleh seorang admin.

4.1.1 Spesifikasi Sistem

Analisis kebutuhan sistem merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen – komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

1. Analisis Perangkat Keras (Hardware)

Perangkat keras minimum yang digunakan untuk membangun Sistem Informasi Penjualan ini adalah

- a. Processor Core I.5 Berkecepatan 3.0 Ghz
- b. RAM 4 Gb

- c. Hardisk minimal 100 Tb untuk menyimpan data
- d. LAN Card
- e. Keyboard dan Mouse
- f. Monitor 20 inch.

2. Analisis Perangkat Lunak (Software)

Untuk mendukung dalam penyimpanan informasi, dibutuhkan suatu fasilitas yang memadai. Yaitu berupa perangkat lunak (software) yang dirancang untuk memudahkan dalam pembangunan dan menjalankan sisten nantinya.

Adapun perangkat lunak yang digunakan adalah sebagai berikut :

- a. Microsoft Windows 10 , Windows 10 sebagai sistem operasi
- b. Android Studio, Sebagai Perancangan Program Aplikasi.

4.1.2 Hasil Rancangan Sistem

1. Tampilan Menu *Login*

Rancangan Tampilan *Login* adalah tampilan awal sebelum masuk ke aplikasi. Halaman ini berfungsi untuk memberikan hak akses bagi seorang user sebelum menggunakan aplikasi. Adapun fungsi dari tombol yang ada pada menu *Login* yaitu Tombol *Login* berfungsi untuk memverifikasi data valid untuk melanjutkan ke menu selanjutnya. Berikut tampilan Menu *Login* dapat dilihat pada gambar dibawah ini :



Gambar 4.1. Tampilan Menu *Login*

Tampilan login ini berisikan proses dimana kita memasukkan username dan password sehingga kita dapat masuk kedalam sistem dan menggunakan sistem pencarian atm terdekat dikota binjai.

2. Tampilan Menu Utama

Rancangan Tampilan *Menu Utama* adalah tampilan yang pertama kali ditampilkan setelah berhasil login menggunakan *username* dan *password* yang valid. Halaman ini memiliki beberapa fungsi untuk menghubungkan ke tampilan lainnya. Berikut tampilan Menu *Menu Utama* dapat dilihat pada gambar dibawah ini :

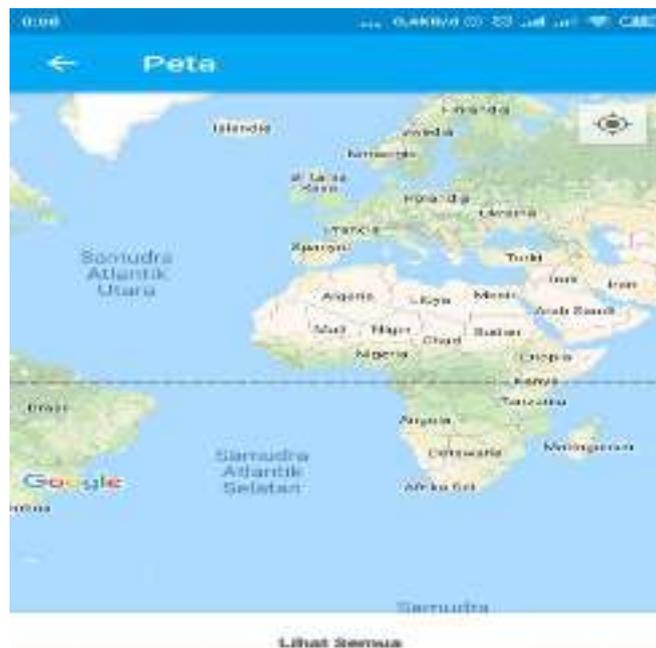


Gambar 4.2. Tampilan Menu Utama

Fungsi dari button mulai adalah untuk menjalankan program pencarian atm terdetak. Fungsi dari button tentang adalah untuk memberikan informasi tentang sistem. Fungsi dari button keluar adalah untuk keluar aplikasi setelah selesai digunakan.

3. Tampilan Menu Pencarian Lokasi ATM

Rancangan Tampilan Pencarian Lokasi ATM adalah tampilan yang berfungsi untuk menginputkan data atm pada android yang nantinya akan menampilkan data rute terpendek. Berikut tampilan Menu pencarian lokasi ATm pada gambar dibawah ini :

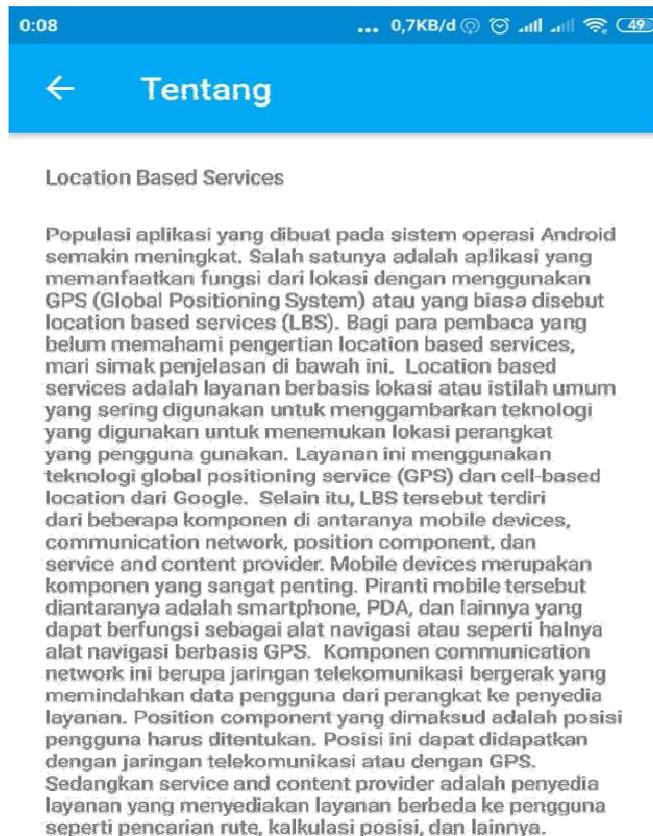


Gambar 4.3. Tampilan Menu Data ATM

Fungsi dari data atm ini, adalah untuk menampilkan atm terdekat dengan menggunakan google map yang ada pada android.

4. Tampilan Tentang Aplikasi

Rancangan Tampilan Menu tentang aplikasi yaitu menampilkan data tentang sistem yang digunakan dalam pembuatan aplikasi android untuk rute terpendek ATM di Kota Binjai.



Gambar 4.4. Tampilan Tentang Aplikasi

4.2 Pengujian Black Box

Untuk dapat menggunakan aplikasi ini dengan baik, dibutuhkan seperangkat komputer dengan spesifikasi minimal. Black Box pengujian adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan.

Tabel 4.1. Tabel Pengujian Black Box

No	Rancangan Proses	Hasil Yang Diharapkan	Hasil	Keterangan
1	Halaman Utama Interaktif dan Mudah Digunaka	Halaman Index (Awal)	Sesuai	-
2	Proses Pencarian Lokasi ATM	Halaman Mulai	Sesuai	-
3	Proses Menampilkan Rute Terpendek	Halaman Rute Terpendek	Sesuai	-
4	Rute	Menampilkan Rute Terpendek	Sesuai	-

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil aplikasi rancangan pencarian rute terpendek menggunakan andriod, maka didapat beberapa kesimpulan seperti berikut :

1. Dengan adanya aplikasi pencarian rute terpendek ATM menjadi lebih cepat dan teratur.
2. Algoritma yang digunakan dalam proses pencarian ruter terpendek ini sangat membantu dan mudah untuk dimengerti oleh penulis.
3. Informasi jarak disetiap titik menjadi lebih jelas dan dapat memanage waktu dalam proses pencarian ATM.

5.2 Saran

Berikut adalah saran dari penulis agar aplikasi rancangan pencarian rute terpendek menggunakan algoritma dapat bermanfaat dan dikembangkan menjadi lebih baik lagi :

1. Di masa yang akan datang, sebaiknya aplikasi ini dikembangkan dengan menerapkan sebuah metode lainnya serta diterapkan untuk rute pencarian ATM terdekat.
2. Aplikasi ini akan lebih baik jika dirancang dengan aplikasi android.

DAFTAR PUSTAKA

- Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). *Jurnal Media Informatika Budidarma*, 2(2).
- Fachri, B. (2018). Perancangan Sistem Informasi Iklan Produk Halal Mui Berbasis Mobile Web Menggunakan Multimedia Interaktif. *Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika)*, 3, 98-102.
- Fachri, B. (2018, September). APLIKASI PERBAIKAN CITRA EFEK NOISE SALT & PAPPER MENGGUNAKAN METODE CONTRAHARMONIC MEAN FILTER. In *Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 87-92)*.
- Fitria Apri Triansyah, *Implementasi Algoritma Dijkstra Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan*, Universitas Sriwijaya 2015.
- Dhandy Darmawan, *Aplikasi Pencarian Rute dan Lokasi ATM Terdekat Menggunakan Formula Haversine Dilengkapi dengan Call Center Bank Berbasis Android*, Universitas Dian Nuswantoro, 2015.
- Baskoro Adi Pri, Susanto Ajib, *Aplikasi Berbasis Android Pencarian ATM Mandiri Terdekat Menggunakan Algoritma Dijkstra*, Universitas Dian Nuswantoro, 2015.
- Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. *Int. J. Recent Trends Eng. Res*, 3(8), 58-64.
- Khairul, K., Ilhami Arsyah, U., Wijaya, R. F., & Utomo, R. B. (2018, September). IMPLEMENTASI AUGMENTED REALITY SEBAGAI MEDIA PROMOSI PENJUALAN RUMAH. In *Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 429-434)*.
- Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. *Jurnal Teknik dan Informatika*, 5(2), 13-19.
- Orien Rindy Erika, Didik Kurniawan, Febi Eka Febriansyah, *Aplikasi Pencarian Letak ATM Berbasis Android dengan GIS di Kota Bandar Lampung*, FMIPA Unila, 2016.
- Muhammad Khoiruddin Harahap, Nurul Khairina, *Pencarian Jalur Terpendek dengan Algoritma Dijkstra*, Politeknik Ganesha Medan, 2017.

- Muhammad Syamsuddin Yusuf, Hanifah Muslimah Az-Zahra, Diah Harnoni Apriyanti, *Implementasi Algoritma Dijkstra Dalam Menemukan Jarak Terdekat Dari Lokasi Pengguna Ke Tanaman Yang di Tuju Berbasis Android*, Universitas Brawijaya, 2017.
- Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. *Int. J. Secur. Its Appl*, 10(8), 173-180.
- Putra, Randi Rian, and Cendra Wadisman. "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means." *INTECOMS: Journal of Information Technology and Computer Science* 1.1 (2018): 72-77.
- Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., ... & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. In *Journal of Physics: Conference Series* (Vol. 1028, No. 1, p. 012052). IOP Publishing.
- Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. *Int. J. Sci. Res. Sci. Technol*, 3(6), 470-473.
- Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.
- Siahaan, MD Lesmana, Melva Sari Panjaitan, and Andysah Putera Utama Siahaan. "MikroTik bandwidth management to gain the users prosperity prevalent." *Int. J. Eng. Trends Technol* 42.5 (2016): 218-222.
- Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.
- Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 100-109.
- Wijaya, Rian Farta, et al. "Aplikasi Petani Pintar Dalam Monitoring Dan Pembelajaran Budidaya Padi Berbasis Android." *Rang Teknik Journal* 2.1 (2019).