

PENENTUAN RUTE TERPENDEK PENDISTRIBUSIAN NASKAH UJIAN NASIONAL MENGGUNAKAN ALGORITMA DIJKSTRA (DINAS PENDIDIKAN DAN PENGAJARAN KOTA BINJAI)

Siswan Syahputra

Program Studi Sistem Informasi STMIK KAPUTAMA,
Jln. Veteran No. 4A-9A Binjai, Sumut, Indonesia
e-mail : siswansyahputra90@gmail.com

Abstrak

Penentuan rute terpendek menuju lokasi tujuan merupakan salah satu masalah yang sering dihadapi pengguna jalur darat. Hal ini juga terjadi saat proses pendistribusian Naskah Ujian Nasional yang dilakukan oleh Dinas Pendidikan dan Pengajaran Kota Binjai. Pada umumnya pemilihan rute saat pendistribusian Naskah Ujian Nasional menuju sekolah tujuan dilakukan dengan cara konvensional atau berdasarkan kesepakatan oleh petugas pendistribusi naskah tersebut, proses ini tidak dilakukan berdasarkan data yang akurat. Masalah rute terpendek dapat diselesaikan dengan sistem informasi geografis berbasis web menggunakan algoritma Dijkstra. Algoritma Dijkstra adalah algoritma pencarian rute terpendek berdasarkan graf untuk memecahkan masalah rute terpendek tunggal. Hal ini diterapkan hanya pada bobot graf positif. Algoritma Dijkstra digunakan untuk memecahkan jalur terpendek dengan biaya minimum.

Kata Kunci: Algoritma Dijkstra, Rute Terpendek, Graf Berbobot.

1. PENDAHULUAN

1.1 Latar Belakang

Penentuan rute terpendek menuju lokasi tujuan merupakan salah satu masalah yang sering dihadapi pengguna jalur darat. Hal ini juga terjadi saat proses pendistribusian Naskah Ujian Nasional yang dilakukan oleh Dinas Pendidikan dan Pengajaran Kota Binjai.

Kota Binjai merupakan kota yang memiliki banyak jalan – jalan alternatif yang menghubungkan satu lokasi dengan lokasi lainnya. Dengan banyaknya jalan alternatif, hal ini dapat memberikan sebuah keuntungan efisiensi waktu dalam pendistribusian Naskah Ujian Nasional, jika tepat dalam memilih rute yang lebih pendek atau menjadi masalah keterlambatan waktu, jika memilih rute yang lebih jauh. Pada umumnya pemilihan rute saat pendistribusian Naskah Ujian Nasional menuju sekolah tujuan dilakukan dengan cara konvensional atau berdasarkan kesepakatan oleh petugas pendistribusi naskah tersebut, proses ini tidak dilakukan berdasarkan data yang akurat.

Berdasarkan penelitian yang pernah dilakukan oleh Antonio Gusmao, dkk (2013),

masalah rute terpendek dapat diselesaikan dengan sistem informasi geografis berbasis web menggunakan algoritma Dijkstra. Ada beberapa algoritma yang dapat digunakan dalam menyelesaikan masalah rute terpendek, diantaranya algoritma Ant Colony dan Dijkstra. Algoritma Dijkstra lebih instensif dalam komputasi untuk pencarian jalur optimum dalam suatu jaringan seperti internet dan waktu rata – rata eksekusi algoritma Dijkstra lebih kecil dibanding algoritma Ant Colony. Maka jalur dalam permasalahan ini algoritma lebih tepat digunakan dengan menggunakan jaringan internet.

Algoritma Dijkstra ditemukan oleh seorang ilmuwan komputer bernama Edsger Dijkstra pada tahun 1956 dan dipublikasikan pada tahun 1959. Algoritma Dijkstra adalah algoritma pencarian rute terpendek berdasarkan graf untuk memecahkan masalah rute terpendek tunggal. Hal ini diterapkan hanya pada bobot graf positif. Algoritma Dijkstra digunakan untuk memecahkan jalur terpendek dengan biaya minimum.

Cara kerja algoritma Dijkstra memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang

menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter tempat asal dan tempat tujuan. Hasil akhir algoritma ini adalah jarak terpendek dari tempat asal ke tempat tujuan beserta rutenya (Bambang, 2014).

1.2 Rumusan Masalah

Berdasarkan uraian diatas dapat dirumuskan permasalahan pokok sebagai berikut:

1. Bagaimana menentukan jarak terpendek dalam pendistribusian Naskah Ujian Nasional.
2. Bagaimana penerapan algoritma Dijkstra untuk menentukan jarak terpendek dalam pendistribusian Naskah Ujian Nasional.

1.3 Tujuan Penelitian

Adapun tujuan yang diharapkan dalam penelitian ini adalah sebagai berikut :

1. Untuk memahami bagaimana algoritma Dijkstra menentukan rute terpendek.
2. Untuk menghasilkan model rute terpendek dalam pendistribusian Naskah Ujian Nasional menggunakan algoritma Dijkstra.
3. Membuat prototype perangkat lunak rute terpendek menuju sekolah – sekolah SMA/SMK di Kota Binjai.

2. LANDASAN TEORI

2.1 Lintasan / Rute Terpendek (*Shortest Path*)

Pemasalahan lintasan / rute terpendek yaitu menemukan lintasan terpendek antara dua atau beberapa simpul lebih yang saling berhubungan. Menurut Hayati dan Yohanes, 2014, Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. Graf yang digunakan dalam pencarian lintasan terpendek adalah graf berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan lain sebagainya. Asumsi yang digunakan di sini adalah bahwa semua bobot bernilai positif. Lintasan terpendek adalah jalur yang dilalui dari suatu *node* ke

node lain dengan besar atau nilai pada sisi yang jumlah akhirnya dari *node* awal ke *node* akhir paling kecil. Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat lain. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf yang berbobot yaitu graf setiap sisinya diberikan suatu nilai atau bobot.

Berikut contoh penerapan pencarian lintasan/rute terpendek :

1. Implementasi algoritma dijkstra dalam aplikasi untuk menentukan lintasan terpendek jalan darat antar kota (Fitri dan Triansyah, 2013). Dalam penelitian tersebut disimpulkan perlu sebuah algoritma untuk dapat menyelesaikan persoalan rute terpendek yaitu dengan algoritma dijkstra.
2. Implementasi algoritma dijkstra pada kartu FPGA (*Field-Programmable Gate Array*) untuk perhitungan telkom (Benaicha dan Taibi, 2013). Jaringan merupakan satu set perangkat komputer yang digunakan untuk memberikan arus informasi, untuk rute informasi yang benar jaringan menggunakan proses routing dengan menggabungkan fleksibilitas dan kecepatan sehingga diperlukan teknologi baru yaitu FPGA.
3. Penemuan rute terpendek pada aplikasi berbasis peta (Wira, 2010). Pada penelitian ini menghasilkan rute terpendek untuk berbagai keperluan masyarakat yang dihadapkan dengan permasalahan transportasi seperti kemacetan pada jalan raya.

Ada beberapa macam persoalan lintasan terpendek, antara lain:

1. Lintasan terpendek antara dua buah simpul (*all pairs shortest path*)
2. Lintasan terpendek antara semua pasangan simpul (*all pairs shortest path*)
3. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain (*single-source shortest path*).
4. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermedia shortest path*).

2.2 Algoritma Rute Terpendek (*Shortest Path Algorithm*)

Algoritma yang dapat digunakan untuk mencari rute terpendek telah banyak diteliti. Beberapa algoritma yang dapat digunakan untuk penyelesaian penentuan rute terpendek (Sanan, dkk, 2013), yaitu :

1. Algoritma Dijkstra
Dijkstra adalah algoritma berdasarkan Greedy dan memecahkan satu sumber masalah jalan terpendek
2. Algoritma Bellman-Ford
Bellman-Ford adalah algoritma berbasis Pemrograman Dinamis.
3. Algoritma A* Search
Algoritma A* Search adalah algoritma pencarian graf/pohon yang menemukan jalur dari node awal yang diberikan ke node tujuan tertentu.
4. Algoritma Floyd-Warshall
Algoritma ini memecahkan semua pasangan jalur terpendek dalam grafik tepi arah.
5. Algoritma Johnson
Algoritma ini memecahkan semua pasangan jalur terpendek, di jarang tertimbang, grafik berarah.

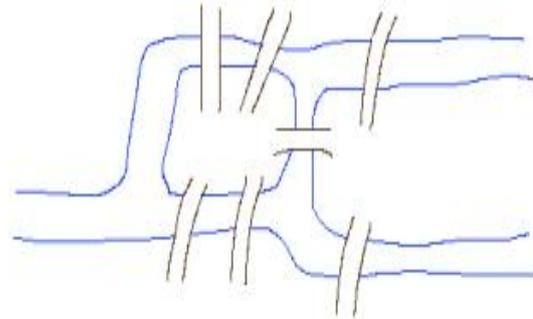
2.3 Teori Graf (*Graph*)

Teori graf merupakan hal yang penting dalam berbagai bidang aplikasi perhitungan, peneliti dapat menggunakan konsep dari teori graf untuk penelitiannya (Shirinivas, 2010). Seperti penelitian ilmu komputer yang dilakukan menggunakan data pertambangan, segmentasi citra, clustering, penangkapan gambar, jaringan dan lain-lain. Struktur data dapat dirancang dalam bentuk pohon (*tree*), begitu juga dengan pemodelan topologi jaringan dapat dilakukan dengan menggunakan konsep graf.

2.4 Definisi Graf

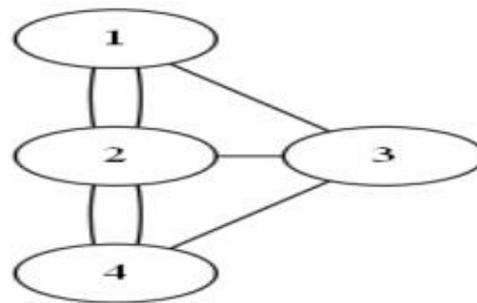
Menurut Setyawan (2014), Graf terdiri atas himpunan simpul V dan himpunan sisi E dengan setiap sisi memiliki ujung-ujung yang berupa simpul. Graf dapat digunakan untuk mempresentasikan berbagai macam sistem nyata, dengan simpul menyatakan unsur dalam sistem tersebut dan unsur-unsur yang saling berhubungan digambarkan dengan adanya sisi yang menghubungkan unsur-unsur itu.

Dalam sejarahnya, graf digunakan oleh Euler untuk memecahkan masalah jembatan Konigsberg. Masalah jembatan Konigsberg merupakan teka-teki yakni dari salah satu tempat tertentu apakah kita dapat berjalan dengan melalui ke tujuh jembatan itu masing-masing tepat satu kali.



Gambar 2.1 Jembatan Konigsberg

Banyak orang sudah mencoba melakukannya dengan berbagai cara namun tidak ada yang berhasil. Hal ini menarik Leonard Euler untuk memecahkan masalah tersebut dengan menggunakan konsep yang sekarang dikenal sebagai teori graf. Jika setiap tempat diwakili oleh simpul dan setiap jembatan diwakili oleh sisi, maka masalah tersebut dapat digambarkan sebagai graf dengan empat simpul dan tujuh sisi.



Gambar 2.2 Representasi Masalah Jembatan Konigsberg Sebagai Graf

Dalam representasi graf dari masalah jembatan Konigsberg ternyata keempat simpul tersebut semuanya memiliki derajat ganjil. Dengan latar belakang matematika yang dimilikinya, Euler membuktikan bahwa kita tidak mungkin pergi dari satu simpul tertentu dengan melalui ke tujuh sisi dari graf itu masing-masing tepat satu kali dan kembali ke simpul awal. Euler membuktikan bahwa ini dilakukan jika sebanyak-banyaknya ada dua simpul dengan derajat ganjil, yakni simpul awal dan simpul

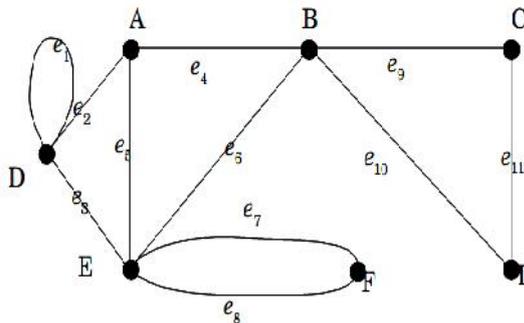
akhir. Dalam perkembangan teori graf, konsep ini dikenal dengan trail Euler (*Eulerian trail*).

2.5 Matrix Graf

Sebuah graf dapat disajikan dalam bentuk matrix (Samuel, 2008), yaitu :

1. Matrix titik (*Adjacent Matrix*)
2. Matrix rusuk (*Edge Matrix*)
3. Matrix titik – rusuk (*Incidence Matrix*)

Berikut graf yang dinyatakan dalam bentuk matrix titik, rusuk, dan titik – rusuk



Gambar 2.3 Graf dalam Bentuk Matrix

Matrix titik dari graf diatas adalah matrix 7x7, karena graf diatas mempunyai 7 buah titik.

	A	B	C	D	E	F	G
A	0	1	0	1	1	0	0
B	1	0	1	0	1	0	1
C	0	1	0	0	0	0	1
D	1	0	0	2	1	0	0
E	1	1	0	1	0	2	0
F	0	0	0	0	2	0	0
G	0	1	1	0	0	0	0

Gambar 2.4 Elemen – elemen Matrix

Cara mengisi elemen-elemen matrix:

1. Baris 1 kolom 1, dari A ke A = 0
2. Baris 1 kolom 2, dari A ke B = 1
Titik A dan B terhubung oleh sebuah rusuk
3. Baris 4 kolom 4, dari D ke D = 2
Titik D mempunyai loop
4. Baris 5 kolom 6, dari E ke F = 2
Titik E dan F terhubung oleh 2 buah rusuk e_7 dan e_8
5. Baris 7 kolom 1, dari G ke A = 0
Titik G dan A tidak terhubung oleh sebuah rusuk

Matrix rusuk dari graf di atas adalah matrix 11x11, karena graf di atas mempunyai 11 rusuk.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}
e_1	0	1	1	0	0	0	0	0	0	0	0
e_2	1	0	1	1	1	0	0	0	0	0	0
e_3	1	1	0	0	1	1	1	1	0	0	0
e_4	0	1	0	0	1	1	0	0	1	1	0
e_5	0	1	1	1	0	1	1	1	0	0	0
e_6	0	0	1	1	1	0	1	1	1	1	0
e_7	0	0	1	0	1	1	0	1	0	0	0
e_8	0	0	1	0	1	1	1	0	0	0	0
e_9	0	0	0	1	0	1	0	0	0	1	1
e_{10}	0	0	0	1	0	1	0	0	1	0	1
e_{11}	0	0	0	0	0	0	0	0	1	1	0

Gambar 2.5 Matrix Rusuk 11x11

Cara mengisi elemen-elemen matrix:

Bila sebuah rusuk bertemu dengan rusuk yang lain disebuah titik maka elemen amtriknya = 1, bila tidak bertemu di satu titik maka elemen matriknya = 0.

Matrix titik rusuk dari graph diatas adalah matrix 7x11, karena graph tersebut memiliki 7 titik dan 11 rusuk.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}
A	0	1	0	1	1	0	0	0	0	0	0
B	0	0	0	1	0	1	0	0	1	1	0
C	0	0	0	0	0	0	0	0	1	1	0
D	1	1	1	0	0	0	0	0	0	0	0
E	0	0	1	0	1	1	1	1	0	0	0
F	0	0	0	0	0	0	1	1	0	0	0
G	0	0	0	0	0	0	0	0	0	1	1

Gambar 2.6 Matrix Rusuk 7x11

Cara mengisi elemen-elemen matrix

Bila sebuah rusuk bertemu dengan sebuah titik maka nilai elemen matrix = 1, bila tidak bertemu maka nilai elemen matrix = 0.

2.6 Algoritma Dijkstra

Menurut Liu dan Chen (2010), ide dasar algoritma Dijkstra adalah untuk mengeksplorasi jalan terpendek dari titik sumber (s) ke luar secara bertahap. Dalam proses menetapkan nomor untuk setiap titik (*point*) yang menyatakan berat jalur terpendek dari s ke titik *point* (label P) atau batas atas berat jalur terpendek dari s ke titik *point* (label T). Dalam setiap langkah, memodifikasi label T dan mengubah jumlah titik dengan label T

untuk menunjukkan dengan label P, sehingga jumlah titik dengan label P dalam grafik G bertambah satu, maka bisa didapatkan jalur terpendek dari s ke setiap titik hanya dengan n-1 langkah (n adalah jumlah simpul dalam grafik G).

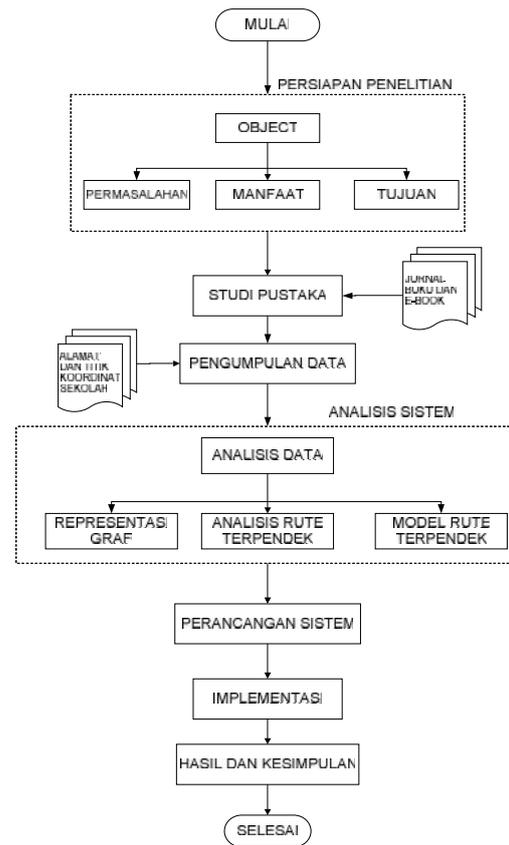
Berikut beberapa contoh penerapan algoritma Dijkstra yang telah dibahas oleh peneliti sebelumnya :

1. Algoritma dijkstra dapat diterapkan dalam pembuatan sistem bantuan bencana alam *Real Time* (Nandiroh, dkk, 2014). Sistem yang dibangun pada penelitian tersebut dapat diakses menggunakan telepon seluler dengan sistem navigasi dan mampu menunjukkan rute terpendek serta dapat menunjukkan rute alternatif jika terjadi kemacetan atau terjadi kecelakaan di salah satu jalur atau lokasi bencana sehingga tercipta sistem layanan yang *real time*.
2. Algoritma Dijkstra juga dapat di implementasikan pada aplikasi untuk menentukan lintasan terpendek jalan darat antar kota (Fitria dan Triansyah, 2013). Pada penelitian ini di hasilkan rute terpendek secara optimal dan cepat dalam menentukan rute terpendek menggunakan algoritma dijkstra
3. Pendistribusian barang farmasi menggunakan algoritma dijkstra juga pernah menjadi konsen penelitian (Sulindawaty, dkk, 2015). Pada penelitian tersebut algoritma Dijkstra dibandingkan dengan algoritma Prim. Hasil penelitian yang didapatkan adalah algoritma Dijkstra dan Prim mampu menemukan jalur terpendek ke tempat tujuan. Namun dari hasil analisa, algoritma Dijkstra lebih memiliki komposisi jalur yang lebih dekat dibandingkan dengan algoritma Prim.

3. METODOLOGI PENELITIAN

3.1 Kerangka Kerja Penelitian

Kerangka kerja ini merupakan langkah – langkah yang akan dilakukan dalam penyelesaian maslaah yang akan dibahas. Adapaun kerangka kerja penelitian ini adalah sebagai berikut :



Gambar 3.1 Kerangka Kerja Penelitian

- a. **Persiapan Penelitian**
Pada tahap ini ditentukan objek penelitian, permasalahan yang ada, tujuan dan manfaat dari penelitian.
- b. **Studi Pustaka**
Dalam penelitian ini dilakukan pengumpulan dan mempelajari literatur – literatur yang berhubungan dengan permasalahan, teori – teori, dan algoritma dijkstra baik dari buku, jurnal, *e-book* dan lain sebagainya.
- c. **Pengumpulan Data**
Data yang dikumpulkan berupa nama, alamat, lokasi, dan titik koordinat (latitude, longitude) sekolah SMA/SMK di kota Binjai. Data koordinat, data jalan dan persimpangan diperoleh dari aplikasi *One Touch Location* dan Google Map.
- d. **Analisis Sistem**
Pada tahap ini data yang telah dikumpulkan kemudian dianalisa dan di representasikan dalam bentuk graf. Dari hasil representasi graf selanjutnya dianalisa dengan algoritma Dijkstra untuk mendapatkan rute terpendek kemudian di

- gambarkan dalam pemodelan rute terpendek.
- e. Perancangan Sistem
Agar aplikasi *prototype* yang dibangun sesuai dengan hasil analisa maka pada tahap ini dilakukan perancangan sistem.
 - f. Implementasi Sistem
Tahap ini merupakan tahap pengujian terhadap sistem menggunakan aplikasi berbasis web sekaligus mengevaluasi kekurangan – kekurangan yang ada pada sistem.
 - g. Hasil dan Kesimpulan
Pada tahap ini di hasilkan rute terpendek dari Dinas Pendidikan dan Pengajaran kota Binjai menuju sekolah-sekolah SMA/SMK di kota Binjai, kemudian disimpulkan dengan algoritma Dijkstra apakah mampu menentukan rute terpendek dengan akurat dalam pendistribusian Naska Ujian Nasional.

4. ANALISA DAN PERANCANGAN

Untuk menyelesaikan masalah penentuan rute terpendek dalam distribusi naskah Ujian Nasional oleh Dinas Pendidikan dan Pengajaran Binjai ke sekolah – sekolah SMA/SMK di seluruh Kota Binjai, akan dilakukan analisa data lokasi yang akan diselesaikan menggunakan algoritma Dijkstra. Adapun hal yang pertama yang akan dilakukan adalah menganalisa data yang diperoleh dan di representasikan dalam bentuk graf berbobot dari data yang ada.

Sebuah graf berbobot dengan n buah simpul dinyatakan dengan matriks ketetanggaan $M = [m_{ij}]$, dalam hal ini,
 m_{ij} = bobot sisi (i, j) (pada graf tak – berarah $m_{ij} = m_{ji}$)
 $m_{ij} = 0$
 $m_{ij} = \infty$, jika tidak ada sisi dari simpul i ke simpul j

Selain matriks M , juga menggunakan tabel $S = [s_i]$ yang dalam hal ini,
 $s_i = 1$, jika simpul i termasuk ke dalam rute terpendek
 $s_i = 0$, jika simpul i tidak termasuk ke dalam rute terpendek

Tabel $D = [d]$ yang dalam hal ini,
 d_i = panjang rute dari simpul awal a ke simpul i .

$m_{ij} = \infty$, jika tidak ada sisi dari simpul i ke simpul j

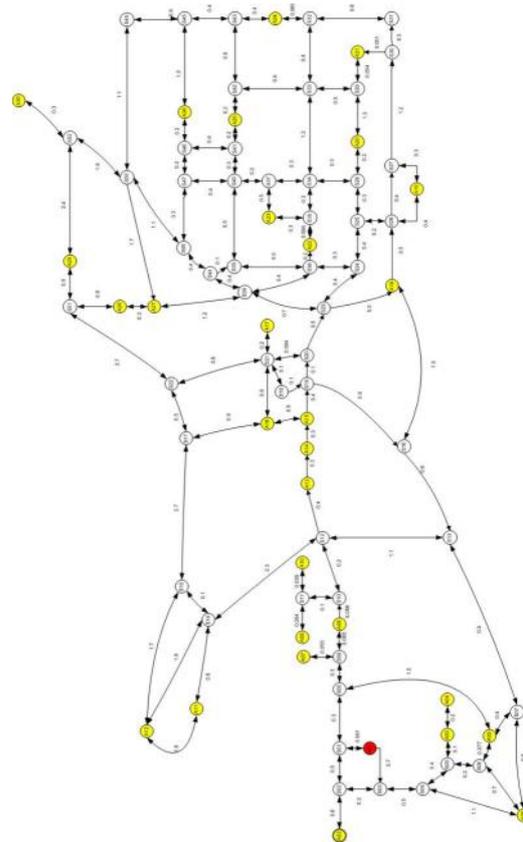
Larik $S = [s_i]$ yang dalam hal ini,

$s_i = 1$, jika simpul i termasuk ke dalam rute terpendek

$s_i = 0$, jika simpul i tidak termasuk ke dalam rute terpendek

Larik $D = [d]$ yang dalam hal ini,

d_i = panjang rute dari simpul awal a ke simpul i .

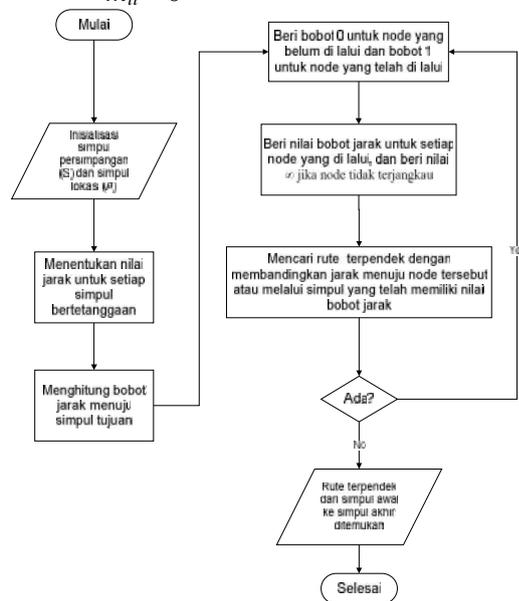


Gambar 4.1 Pemodelan Graf Lokasi Sekolah SMA/SMK Di Kota Binjai

Dari pemodelan graf lokasi sekolah SMA/SMK di kota Binjai pada gambar 4.7 simpul persimpangan diwakili dengan kode berawalan huruf “S” (S01, S02, S03,) sedangkan simpul lokasi awal diwakili dengan kode berawalan huruf “A” (A01) dan untuk simpul tujuan kelanjutan dari kode simpul awal (A02, A03, A04,), semua simpul tersebut dihubungkan lintasan – lintasan atau busur yang memiliki bobot jarak dengan satuan kilometer (km) pada setiap busur nya . Selanjutnya dilakukan perhitungan untuk mencari rute terpendek dari simpul awal menuju simpul tujuan. Berikut ini *sample* perhitungan dari beberapa simpul tujuan :

Sebuah graf berbobot dengan n buah simpul dinyatakan dengan matriks ketetanggaan $M = [m_{ij}]$, dalam hal ini,

m_{ij} = bobot sisi (i, j) (pada graf tak – berarah $m_{ij} = m_{ji}$)
 $m_{ii} = 0$



Gambar 4.2 Flowchart Algoritma Dijkstra

4.1 Analisa Kebutuhan Sistem

Proses analisis kebutuhan ini diawali dengan penjabaran umum aplikasi rute terpendek, identifikasi lokasi sekolah, penjabaran tentang kebutuhan dan kemudian dimodelkan kedalam digram use case. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan – kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.2 Gambaran Umum Aplikasi Rute Terpendek

Aplikasi rute terpendek ini merupakan aplikasi *prototype* yang dirancang untuk mempermudah pengguna untuk mendapatkan lokasi sekolah dalam proses pendistribusian naskah ujian. Ketika mengakses aplikasi ini, pengguna hanya memerlukan sedikit waktu, maka pengguna sudah mendapatkan lokasi sekolah – sekolah yang akan dituju

4.3 Daftar Kebutuhan Sistem

Daftar kebutuhan terdiri dari kebutuhan fungsional dan non-fungsional. Pada tabel 4.1

kebutuhan fungsional ditunjukkan dengan penomoran F, sedangkan kebutuhan non-fungsional ditunjukkan dengan penomoran NF.

Tabel 4.1 Daftar Kebutuhan Fungsional

Identifikasi	Kebutuhan	Uses Case
F01	Aplikasi harus mampu menampilkan lokasi - lokasi sekolah	Melihat lokasi – lokasi sekolah
F02	Aplikasi harus mampu menampilkan rute dari lokasi awal menuju lokasi tujuan yang telah di pilih	Melihat rute terpendek dari lokasi awal menuju lokasi tujuan
F03	Aplikasi harus mampu mencari lokasi tujuan awal hingga lokasi tujuan akhir berdasarkan rute terpendek	Melihat rute terpendek dari lokasi tujuan awal hingga lokasi tujuan akhir
F04	Aplikasi harus mampu menambahkan data lokasi sekolah yang baru	Penambahan data lokasi sekolah
F05	Aplikasi harus mampu menambahkan data rute perbandingan	Penambahan data rute perbandingan

Tabel 4.2 Daftar Kebutuhan Non-Fungsional

Parameter	Kode	Deskripsi Kebutuhan
Usability	NF01	Aplikasi harus dengan mudah digunakan oleh pengguna (Ketika pengguna menjalankan aplikasi maka aplikasi akan mampu menampilkan lokasi – lokasi sekolah pada peta dalam

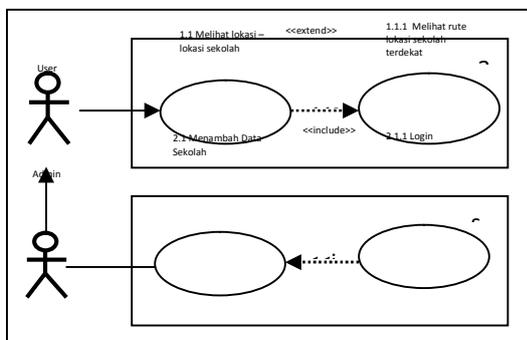
		bentuk marker)
Compatibility	NF02	Aplikasi harus dapat digunakan pada komputer dengan sistem operasi windows

4.4 Perancangan Sistem

Perancangan perangkat lunak dilakukan setelah semua kebutuhan perangkat lunak didapatkan melalui tahap analisis kebutuhan. Perancangan perangkat lunak berdasarkan *object oriented analysis* dan *object oriented design* yaitu menggunakan pemodelan UML (*Unified Modeling Language*). Perancangan di mulai dari perancangan alur atau aktifitas yang dilakukan pengguna secara prosedural yang dimodelkan dalam *activity diagram*. Interaksi antar objek yang telah diidentifikasi, di modelkan dalam *sequence diagram*. Selanjutnya, dilakukan perancangan sistem aplikasi rute terpendek dengan mengidentifikasi class dan layout yang dibutuhkan, serta kemudian dimodelkan dalam class diagram. Kemudian tahap perancangan dilanjutkan dengan perancangan antarmuka pengguna

4.5 Diagram Use Case

Pemodelan use case sistem diperoleh dari kebutuhan fungsional yang digunakan untuk menggambarkan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Gambar 4.3 menunjukkan use case sistem.



Gambar 4.3 Use Case Sistem

Gambar 4.3 merupakan use case sistem yang terdiri dari 2 aktor yaitu user dan admin, dimana admin dapat memanipulasi data lokasi sekolah dan mengakses aplikasi, sedangkan user hanya dapat melakukan akses aplikasi dan tidak dapat memanipulasi data lokasi sekolah. Setiap use case yang ada pada gambar 4.3 dapat dijelaskan oleh skenario use case.

4.6 Skenario Use Case

Tabel 4.3 menjelaskan kegiatan yang dilakukan pada saat melihat lokasi sekolah pada gambar use case sistem. Pertama pengguna membuka alamat aplikasi pada browser kemudian sistem akan menampilkan peta, marker lokasi sekolah – sekolah SMA/SMK.

Tabel 4.3 Skenario Use Case Melihat Lokasi Sekolah

Nomor Use Case	F01
Nama Use Case	Melihat lokasi – lokasi sekolah
Prasyarat Konteks	Komputer terkoneksi internet, aplikasi di buka pada browser.
Tujuan Konteks	Mempermudah pengguna untuk menentukan rute lokasi sekolah terdekat
Prakondisi	Aplikasi telah terpasang dan pengguna membukan aplikasi
Kondisi Akhir Sukses	Aplikasi menampilkan lokasi sekolah dan menampilkan rute lokasi sekolah terdekat
Kondisi Akhir Failed	-
Aktor	Pengguna
Alur Utama	Aktifitas
	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Sistem menampilkan lokasi – lokasi sekolah. 3. Sistem menampilkan

	rute lokasi sekolah terdekat
--	------------------------------

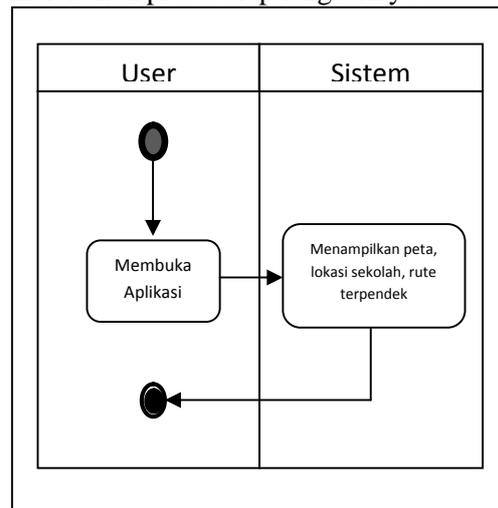
Tabel 4.3 menjelaskan kegiatan yang dilakukan pada saat admin login ke sistem pada gambar IV.3 Use Case Sistem. Pertama admin login ke sistem kemudian sistem akan menampilkan peta, marker lokasi sekolah – sekolah SMA/SMK, tampilan hampir sama dengan halaman pengguna. Namun ada penambahan kolom yaitu kolom Tambah Lokasi Sekolah dan Input Jarak Sekolah.

Tabel 4.4 Skenario Use Case Penambahan Data Lokasi Sekolah

Nomor Use Case	F02
Nama Use Case	Penambahan data lokasi sekolah
Prasyarat Konteks	Komputer terkoneksi internet, aplikasi di buka pada browser.
Tujuan Konteks	Memberi akses kepada admin untuk melakukan penambahan data
Prakondisi	Aplikasi telah terpasang dan admin membuka aplikasi
Kondisi Akhir Sukses	Aplikasi menampilkan lokasi sekolah dan memberi akses kepada admin untuk dapat melakukan penambahan data
Kondisi Akhir Failed	-
Aktor	Admin
Alur Utama	Aktifitas
	<ol style="list-style-type: none"> 1. Admin membuka aplikasi 2. Sistem menampilkan lokasi – lokasi sekolah. 3. Sistem memberi akses kepada admin untuk melakukan penambahan data

4.7 Diagram Activity

Diagram activity adalah diagram untuk memodelkan aktivitas antara pengguna dan sistem yang berjalan berdasarkan pada skenario use case. Skenario use case yang digambarkan pada tabel 4.4 dapat di modelkan kedalam diagram activity, dapat terlihat aktor yang melakukan proses tiap langkahnya.



Gambar 4.4 Aktivitas Diagram Melihat Lokasi Sekolah

Gambar 4.4 merupakan aktivitas diagram melihat lokasi sekolah dan rute terpendek. Aktivitas diagram dibuat untuk menjelaskan interaksi antara user dengan sistem. Pertama user membuka aplikasi pada browser di komputer dengan memanggil alamat server atau aplikasi. Kemudian aplikasi menampilkan peta, lokasi sekolah dan rute terpendek.

5. HASIL DAN PEMBAHASAN

Pada implementasi pengujian ini untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Pengujian validasi menggunakan metode pengujian *Black-Box*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan ditekankan untuk menemukan rute terpendek menuju sekolah - sekolah SMA/SMK di Kota Binjai. Pada penelitian ini dilakukan pengujian validasi terhadap aplikasi pencarian rute terpendek.

1. Kasus Uji Pengguna

Pada tahap ini dilakukan pengujian terhadap aplikasi untuk mengetahui aplikasi memenuhi kebutuhan fungsional atau tidak dengan menampilkan halaman utama pengguna, menampilkan lokasi – lokasi sekolah SMA/SMK dalam bentuk marker pada peta dan daftar list, serta aplikasi dapat menampilkan hasil pencarian rute terpendek. Berikut merupakan beberapa pengujian yang dilakukan :

- a. Kasus uji pembukaan aplikasi dan kasus uji melihat lokasi – lokasi sekolah, hasil pengujian di tampilkan pada tabel V.1

Tabel 5.1 Kasus Uji Membuka Aplikasi

Nama Kasus Uji	Kasus uji pembukaan aplikasi pencarian rute terpendek dan melihat lokasi – lokasi sekolah SMA/SMK
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi memenuhi kebutuhan fungsional untuk membuka aplikasi kemudian menampilkan lokasi – lokasi sekolah SMA/SMK
Prosedur Uji	Pengguna membuka browser kemudian memasukkan alamat aplikasi pada <i>address bar</i> .
Hasil yang diharapkan	Aplikasi dapat menampilkan halaman utama dan lokasi – lokasi sekolah SMA/SMK pada peta dalam bentuk marker.

- b. Kasus uji pencarian rute terpendek dengan menentukan lokasi awal dan lokasi tujuan

Tabel 5.2 Kasus Uji Pencarian Rute Terpendek

Nama Kasus Uji	Pencarian rute terpendek dengan menentukan lokasi awal dan lokasi tujuan
Objek Uji	Kebutuhan Fungsional
Tujuan Uji	Pengujian dilakukan untuk memastikan aplikasi mampu menampilkan rute terpendek

	dengan menentukan lokasi awal dan lokasi tujuan.
Prosedur Pengujian	Pengguna memilih lokasi awal dan lokasi tujuan pada daftar pilihan, kemudian pengguna menekan tombol “Cari Rute”
Hasil yang diharapkan	Aplikasi dapat mencari rute terpendek dari lokasi awal ke lokasi tujuan dan menampilkan jalan yang dilalui pada kolom panel rute

- c. Kasus uji pencarian lokasi tujuan awal hingga akhir berdasarkan rute terpendek

Tabel 5.3 Kasus Uji Pencarian Lokasi Tujuan Awal Hingga Lokasi Tujuan Akhir

Nama Kasus Uji	Pencarian lokasi tujuan awal hingga akhir berdasarkan rute terpendek
Objek Uji	Kebutuhan Fungsional
Tujuan Uji	Pengujian dilakukan untuk memastikan aplikasi mampu mencari lokasi tujuan awal hingga lokasi tujuan akhir berdasarkan rute terpendek, kemudian menampilkan hasil pencarian.
Prosedur Pengujian	Pengguna menekan tombol “Cari Rute Berikutnya” pada kolom cari semua rute, kemudian pengguna menekan tombol “Simpan Rute” untuk menyimpan hasil pencarian.
Hasil yang diharapkan	Aplikasi mampu mencari lokasi tujuan awal hingga lokasi tujuan akhir berdasarkan rute terpendek, kemudian menyimpan dan menampilkan kembali hasil yang di dapat.

2. Kasus Uji Admin

Pada tahap ini dilakukan pengujian terhadap aplikasi untuk mengetahui aplikasi memenuhi kebutuhan fungsional atau tidak dengan menampilkan halaman admin, menambah data lokasi – lokasi sekolah serta menambah data rute perbandingan.

Berikut merupakan beberapa pengujian yang dilakukan :

- a. Kasus uji penambahan data lokasi – lokasi sekolah, hasil pengujian di tampilkan pada tabel 5.4

Tabel 5.4 Kasus Uji Penambahan Lokasi Sekolah

Nama Kasus Uji	Penambahan lokasi sekolah
Objek Uji	Kebutuhan Fungsional
Tujuan Uji	Pengujian dilakukan untuk memastikan aplikasi mampu menambah lokasi sekolah baru.
Prosedur Pengujian	Admin memasukkan data sekolah baru pada kolom “Tambah Lokasi Sekolah” kemudian admin menekan tombol “Simpan”
Hasil yang diharapkan	Aplikasi mampu menambah dan menyimpan data lokasi sekolah baru.

- b. Kasus uji penambahan data rute perbandingan

Tabel 5.5 Kasus Uji Penambahan Rute Perbandingan

Nama Kasus Uji	Penambahan data rute perbandingan
Objek Uji	Kebutuhan Fungsional
Tujuan Uji	Pengujian dilakukan untuk memastikan aplikasi mampu menambah data rute perbandingan yang baru
Prosedur Pengujian	Admin memasukkan data rute perbandingan yang baru pada kolom “Input Jarak Lokasi” kemudian admin menekan tombol “Simpan”.
Hasil yang diharapkan	Aplikasi mampu menambah dan menyimpan data rute perbandingan yang baru.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil analisa, implementasi dan pengujian yang dilakukan, maka di ambil kesimpulan sebagai berikut :

1. Permasalahan pencarian rute terpendek pendistribusian naskah ujian nasional ke seluruh SMA/SMK di Kota Binjai dapat diselesaikan dengan algoritma Dijkstra.
2. Algoritma Dijkstra mampu menghasilkan model rute terpendek dalam pendistribusian naskah ujian nasional ke seluruh SMA/SMK di Kota Binjai.
3. Prototype perangkat lunak pencarian rute terpendek dapat menyimpan dan menampilkan hasil pencarian rute terpendek dari lokasi tujuan awal hingga lokasi tujuan akhir.
4. Dengan menggunakan algoritma Dijkstra dalam proses pencarian rute terpendek dari lokasi tujuan awal hingga lokasi tujuan akhir, jika ada beberapa lokasi yang berdekatan dengan lokasi asal ada kemungkinan lokasi tersebut akan berada di akhir urutan rute terpendek

6.2 Saran

Berikut ini beberapa saran yang dapat dipergunakan untuk pengembangan penelitian dalam menentukan rute terpendek

1. Memperluas cakupan wilayah penelitian seperti wilayah Sumatera Utara, sehingga dapat membantu Dinas Pendidikan dan Pengajaran kantor pusat dalam pendistribusian naskah ujian nasional.
2. Dilakukan pengembangan terhadap *prototype* perangkat lunak pencarian rute terpendek.

DAFTAR PUSTAKA

- [1]Antonio, G., Sholeh, H.P dan Sunaryo, 2013, Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencitraan Jalur Terpendek Dengan Algoritma Dijkstra, *Jurnal EECIS* Vol 7, No. 2
- [2]Benaicha, R. Dan Taibi, M., 2013, Dijkstra Algorithm Implementation On FPGA Card For Telkom Calculations, *International Journal of Engineering Sciences &*

- Emerging Technologies*, Volume 4, Issue 2, pp: 110-116 ©IJESET.
- [3]Fitria dan Apri.T.,2013,Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan, *Jurnal Sistem Informasi (JSI)*, Vol. 5, No.2.
- [4]Liu, X.Y dan Chen, Y.L.,2010, Application of Dijkstra Algorithm in Logistics Distribution Lines, *Proceeding of the Third International Symposium on Computer Science and Computational Technology(ISCST '10)* Jiaozuo, P.R China, 14-15 Agustus.
- [5]Putu Wira Buana, 2010, Penemuan Rute Terpendek Pada Aplikasi Berbasis Web, *Lontar Komputer*, Vol.1, No. 1
- [6]Siti, N., Haryanto dan Hafidh,M.,2014, Implementasi Algoritma Dijkstra Sebagai Solusi Efektif Pembuatan Sistem Bantuan Bencana Real Time, *Jurnal Ilmiah Teknik Industri*,Vol. 12, No. 2.
- [7]Sulindawaty, Hendryan, W., dan Trinanda, S., 2015, Pendistribusian Barang Farmasi Menggunakan Algoritma Dijkstra (Studi Kasus: PT. Air Mas Chemical), *Jurnal Santikom*, Vol. 14, No. 1.
- [8]Uning, L., dan Marwoto, 2012, Aplikasi Sistem Informasi Geografis Pemetaan Digital Loop Carrier, *Jurnal Teknologi Technoscientia*, Vol. 5, No.1.
- [9]Yudi, S., 2014, Visualisasi Graf Dan Algoritma – Algoritma Dalam Teori Graf Menggunakan Beberapa Paket Software, *Prosiding Seminar Nasional Aplikasi Sains & Teknologi(SNAST)*,Yogyakarta, 15 November.