Jurnal Tera
Volume 1, Issue 2, September 2021
Page 227-240

E-ISSN : 2776-9666
P-ISSN : 2776-1789

RANCANG BANGUN ANTARMUKA INTERAKTIF BERBASIS GRAFIK UNTUK SIMULASI PENCARIAN RUTE TERPENDEK DENGAN ALGORITMA DJIKSTRA

Arum Kusuma Wardani[1]
[1] Jurusan Teknik Informatika, Fakultas Teknik dan Informatika, Universitas Dian Nusantara, Jakarta

Corresponding author
E-mail: arum.kusuma.wardani@undira.ac.id

Diterima : 15/03/2021
Direvisi : 25/04/2021
Dipublikasi : 19/05/2021

Abstrak: Algoritma dijkstra ditemukan oleh Edger Wybe Dijkstra merupakan salah satu algoritma untuk menentukan lintasan terpendek. Salah satu implementasi Algoritma Djikstra dengan bahasa C dirasa tidak efektif karena data titik dan garis harus dibuat dan disimpan terlebih dahulu dalam bentuk file teks sehingga menyulitkan pemakai. Maka diperlukan aplikasi dengan antarmuka interaktif berbasis grafik yang dapat mempermudah dalam menyediakan semua komponen penunjang yang dibutuhkan pada Algoritma Djikstra, meliputi pembuatan titik, garis, memasukan gambar/peta dan meyimpanan data. Metode yang digunakan dalam pembuatan aplikasi ini adalah metode the linier sequential Model yang terdiri dari 5 tahap yaitu pengumpulan data, analisis, desain, pengkodean, pengujian. Rancang Bangun Antarmuka Interaktif berbasis grafik untuk Pencarian Rute Terpendek dengan Algoritma Djikstra dapat menyediakan semua komponen penunjang yang dibutuhkan pada seperti pembuatan titik, garis, memasukan gambar/peta dan meyimpanan data lebih interaktif dan mudah serta memberikan hasil perhitungan yang baik, sehingga pemakai dapat mengetahui jarak terpendek yang bisa digunakan untuk mencapai satu tujuan.

Kata Kunci: Rancang Bangun Antarmuka Interaktif , Algoritma Djikstra

PENDAHULUAN

Algoritma dijkstra ditemukan oleh Edger Wybe Dijkstra merupakan salah satu algoritma untuk menentukan lintasan terpendek. Pada Algoritma Djiksta, kota disimbolkan dengan sebuah vertek dan jalan disimbolkan dengan sebuah edge (sisi) berarah dengan bobot yang menandakan jarak ,yang menghubungkan antar vertek. Penjelasan mengenai Algoritma

Djikstra banyak tersedia di internet, salah satunya diimplementasikan dengan program bahasa C. Pada bahasa C pengguna diharuskan untuk membuat file teks yang berisi data titik, titik mana saja yang dihubungkan dan besarnya jarak dari titik yang berhubungan. Hal tersebut dirasa tidak efektif dan menyulitkan pemakai.

Atas dasar tersebut maka diperlukan adanya suatu aplikasi dengan antarmuka interaktif berbasis grafik yang dapat mempermudah dalam menyediakan semua komponen penunjang yang dibutuhkan pada Algoritma Djikstra, meliputi pembuatan titik, garis, memasukan gambar/peta dan meyimpanan data. Berdasarkan kordinat titik dan hubungan antar titik tersebut, dapat ditentukan rute terpendeknya. Implementasi aplikasi ini menggunakan bahasa pemrograman visual basic 6.0. Salah satu implementasi Algoritma Djikstra pada bahasa C mengharuskan untuk membuat data kordinat titik dan garis dalam bentuk file teks terlebih dahulu. Hal tersebut dirasa tidak efektif bagi pemakai, sehingga diperlukan adanya suatu aplikasi yang memiliki antarmuka interaktif

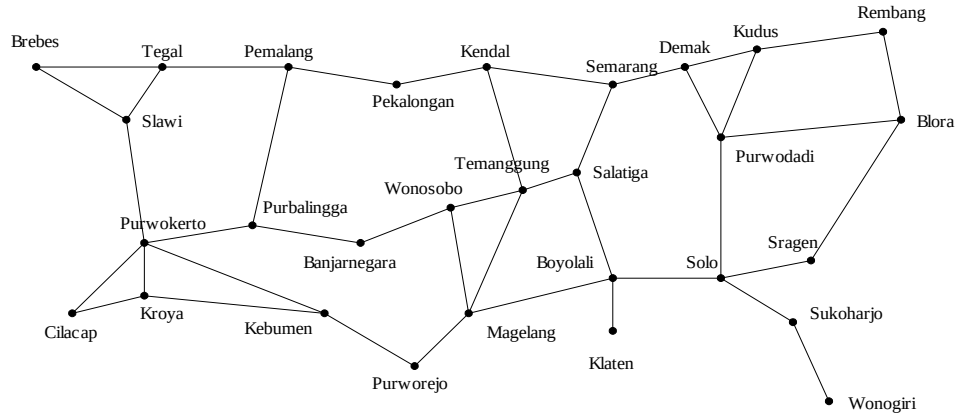
KAJIAN PUSTAKA

Salah satu permasalahan yang sering dijumpai dalam kehidupan sehari-hari adalah penentuan lintasan terpendek (shortest path). Permasalahan ini dapat dimodelkan dengan sebuah graf berbobot dengan nilai pada sisinya yang merepresentasikan jarak. Graf merupakan suatu cabang ilmu yang memiliki banyak terapan. Banyak sekali struktur yang bisa direpresentasikan dengan graf, dan banyak masalah yang bisa diselesaikan dengan bantuan graf. Misalnya graf digunakan untuk merepresentasikan jaringan jalan raya yang dimodelkan graf dengan kota sebagai simpul (vertex/node) dan jalan yang menghubungkan setiap kotanya sebagai sisi (edge) yang bobotnya (weight) adalah panjang dari jalan tersebut.

Teori Dasar Graf

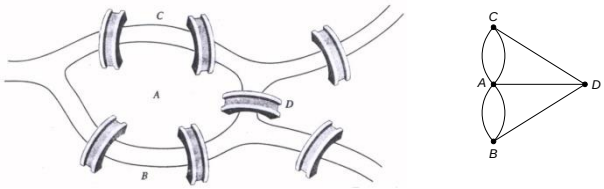
Sebuah Graf G didefinisikan sebagai pasangan himpunan (V,E), dimana V=himpunan verteks (V1,V2,V3 dst) dan E=himpunan edge (arc) yang menghubungkan verteks (e1,e2,e3 dst). Dapat ditulis dengan notasi G=(V,E)

- Sebuah graf yang menyatakan peta jaringan jalan raya yang menghubungkan sejumlah kota di Provinsi Jawa Tengah ditunjukan pada gambar 2.1 peta jawa tengah.



Gambar 1. Peta Jawa Tengah

- Sejarah Graf: masalah jembatan Königsberg (tahun 1736) ditunjukan pada gambar 2.2 masalah jembatan koningsberg

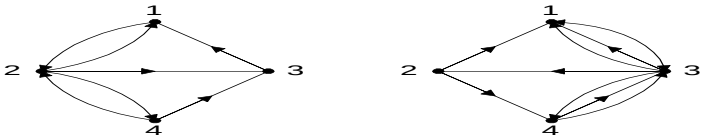


Gambar 2. Masalah Jembatan Konigsberg

Graf yang merepresentasikan jembatan Konigsberg:
Simpul (*vertex*) → menyatakan daratan
Sisi (*edge*) → menyatakan jembatan

Jenis-Jenis Graf

- Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis:
  1. Graf sederhana (*simple graph*).
     Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana. G_1 pada Gambar 2 adalah contoh graf sederhana
  2. Graf tak-sederhana (*unsimple-graph*).
     Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple graph*).
- Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:
  1. Graf tak-berarah (*undirected graph*)
     Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.
  2. Graf berarah (*directed graph* atau *digraph*)
     Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.
     Dua buah graf pada Gambar 3 adalah graf berarah.

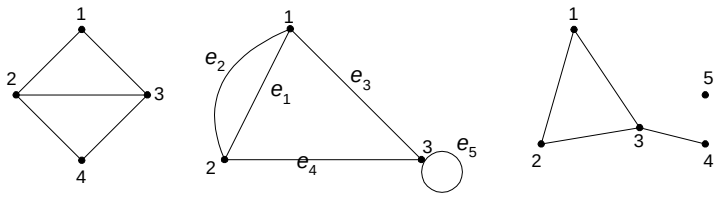


(a) G_4 (b) G

Gambar 3. (a) graf berarah, (b) graf-ganda berarah

Tabel 1 Jenis-Jenis Graf

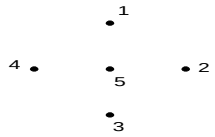
Jenis	**Sisi**	Sisi ganda dibolehkan?	Sisi gelang dibolehkan?
Graf sederhana	Tak-berarah	Tidak	Tidak
Graf ganda	Tak-berarah	Ya	Tidak
Graf semu	Tak-berarah	Ya	Ya
Graf berarah	Bearah	Tidak	Ya
Graf-ganda berarah	Bearah	Ya	Ya



G_1 G_2 G_3

Gambar 4. Graf Untuk Menjelaskan Terminologi Pada Graf

1) Ketetanggaan (*Adjacent*)
   Dua buah simpul dikatakan *bertetangga* bila keduanya terhubung langsung.
   Tinjau graf G_1 : simpul 1 bertetangga dengan simpul 2 dan 3, simpul 1 tidak bertetangga dengan simpul 4.
2) Bersisian (*Incidency*)
   Untuk sembarang sisi $e = (v_j, v_k)$ dikatakan e bersisian dengan simpul v_j , atau e bersisian dengan simpul v_k
   Tinjau graf G_1: sisi (2, 3) bersisian dengan simpul 2 dan simpul 3, sisi (2, 4) bersisian dengan simpul 2 dan simpul 4, tetapi sisi (1, 2) tidak bersisian dengan simpul 4.
3) Simpul Terpencil (*Isolated Vertex*)
   Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya. Tinjau graf G_1: simpul 5 adalah simpul terpencil.
4) Graf Kosong (*null graph* atau *empty graph*)
   Graf yang himpunan sisinya merupakan himpunan kosong (N_n).
5) Derajat (*Degree*)
   Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut.



6) Lintasan (*Path*)
   Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf *G1* ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk v_0, e_1, v_1, e_2, v_2,... , v_{n-1}, e_n, v_n sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ... , $e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf *G1*. Tinjau graf G_1: lintasan 1, 2, 4, 3 adalah lintasan dengan barisan sisi (1,2), (2,4), (4,3). Jika semua verteks berbeda hanya dilewati satu kali maka suatu lintasan dikatakan sederhana (simple path). Jika lintasan dimulai dan diakhiri dengan verteks yang sama Vo=Vn maka disebut verteks tertutup (close path) sebaliknya jika verteks awal dan akhir berbeda maka disebut verteks terbuka (open path). Panjang lintasan adalah jumlah sisi dalam lintasan tersebut. Lintasan 1, 2, 4, 3 pada G_1 memiliki panjang 3.
7) Siklus (*Cycle*) atau Sirkuit (*Circuit*)
   Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus. Tinjau graf G_1: 1, 2, 3, 1 adalah sebuah sirkuit. Panjang sirkuit adalah jumlah sisi dalam sirkuit tersebut. Sirkuit 1, 2, 3, 1 pada G_1 memiliki panjang 3.
8) Terhubung (*Connected*)
   Dua buah simpul v_1 dan simpul v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2. Graf terhubung (*connected graph*) jika untuk setiap pasang simpul v_i dan v_j dalam himpunan V terdapat lintasan dari v_i ke v_j. Jika tidak, maka G disebut graf tak-terhubung (*disconnected graph*).
9) Graf Berbobot (*Weighted Graph*)
   Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot).

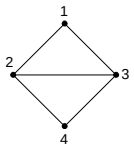
Representasi Graf pada Matriks

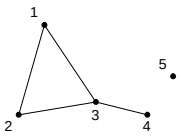
Matriks dapat digunakan untuk merepresentasikan graf. Adapun beberapa matriks adalah :

Matriks Ketetanggaan (*adjacency matrix*)

$A = [a_{ij}]$,

$$a_{ij} = \begin{cases} 1, \text{ jika simpul } i \text{ dan } j \text{ bertetangga} \\ 0, \text{ jika simpul } i \text{ dan } j \text{ tidak bertetangga} \end{cases}$$



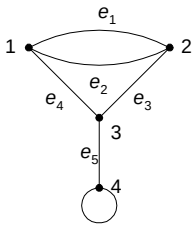


1 2 3 4 1 2 3 4 5

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

jika graf yang diberikan adalah graf berbobot maka elemen matriks yang terhubung diisi dengan bobot graf.

Matriks Bersisian (*incidency matrix*)



$A = [a_{ij}]$,

$$a_{ij} = \begin{cases} 1, & \text{jika simpul } i \text{ bersisian dengan sisi } j \\ 0, & \text{jika simpul } i \text{ tidak bersisian dengan sisi } j \end{cases}$$

Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut urutan prosedur dari suatu program.[Indra Yatini B,2010] Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yan lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Pedoman-Pedoman Dalam Membuat Flowchart

Ada beberapa petunjuk yang harus diperhatikan, seperti :

- Flowchart digambarkan dari atas ke bawah dan dari kiri ke kanan.
- Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
- Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
- Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan menghitung pajak penjualan.
- Setiap langkah dari aktivitas harus berada pada urutan yang benar.
- Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

- Gunakan simbol-simbol flowchart yang standar.

Jenis Flowchart

Flowchart terbagi atas lima jenis, yaitu :

- Flowchart Sistem (System Flowchart)
- Flowchart Paperwork / Flowchart Dokumen (Document Flowchart)
- Flowchart Skematik (Schematic Flowchart)
- Flowchart Program (Program Flowchart)
- Flowchart Proses (Process Flowchart)

Simbol-Simbol Flowchart

Flow Direction Symbols; dipakai untuk menggabungkan antara symbol yang satu dengan symbol lainnya

- Symbol Off-line Connector (Simbol untuk keluar/masuk prosedure
- atau proses dalam lembar/halaman yang lain)
- Symbol Connector (Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang sama)
- Symbol Comunication Link (Simbol transmisi untuk informasi dari satu lokasi ke lokasi lainnya)
- Input out put simbol menyatakan jenis peralatan yang digunakan untuk media input dan output

Lintasan Terpendek (Shortest Path)

Lintasan terpendek adalah lintasan dengan jarak minimum yang terhubung dari satu tempat ke tempat lain. Lintasan minimum dapat dicari dengan menggunakan graf berbobot. Graf berbobot adalah graf dengan sisi-sisinya memiliki nilai yang merepresentasikan jarak antar tempat. Dalam hal ini bobot bernilai positif. Lintasan terpendek dengan verteks awal A dan Vertek tujuan B didefinisikan sebagai lintasan terpendek dari A- B dengan bobot minimum. Misalkan lintasan diperlukan dalam menentukan jarak yang diawali dengan verteks A dan berakhir di verteks B. Lintasan dapat diperoleh dengan mendaftarkan semua lintasan yang mungkin dari A ke B, yang melalui setiap verteks tepat satu kali dan memilih yang terpendek. Lintasan terpendek dari vertek A ke verteks B dalam suatu jaringan adalah lintasan graf berarah sederhana dari A- B, dengan sifat dimana tidak ada lintasan lain yang memiliki jarak dengan nilai terendah. Untuk menentukan lintasan terpendek dalam suatu graf ,dapat menggunakan metode algoritma Djikstra.

Sejarah Algoritma Djikstra

Algoritma Djikstra ditemukan oleh Edsger W. Djikstra yang merupakan salah satu bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah penentuan lintasan terpendek. Algoritma Djikstra adalah sebuah rakus yang digunakan untuk memecahkan permasalahan jarak terpendek untuk sebuah graf berarah dengan bobot sisi yang bernilai tak negatif. [Http://Wikipedia/Indonesia.com]

Cara Kerja Algoritma Djikstra

Secara umum langkah-langkah dalam algoritma Djikstra adalah sebagai berikut:[Anis Cherid,2009]

1) Berikan nilai nol (0) untuk jarak yang disimpan pada titik awal dan berikan nilai infinity untuk jarak yang disimpan pada seluruh titik lainnya. Tentukan titik awal sebagai titik optimal saat ini (current node)

2) Periksa tetangga langsung dari titik optimal saat ini(current node) yang belum dikunjungi dan hitung jarak dari titik awal melalui titik optimal saat ini. Jika jarak dari titik awal melalui titik optimal lebih kecil dari nilai yang dicatat sebelumnya, maka ganti nilai jarak dengan yang baru dan catat nilai optimal yang menyebabkan berkurangnya nilai jarak.
3) Periksa semua titik selain titik optimal saat ini (current node) dan semua titik yang belum pernah dikunjungi dan pilih jaraknya yang paling kecil. Tentukan titik yang nilai jaraknya paling kecil ini sebagai titik optimal berikutnya. Titik optimal saat ini sebagai titik yang sudah dikunjungi.
4) Jika masih ada tetangga titik optimal saat ini (current node) yang belum dikunjungi, kembali ke langkah 2.

Dibawah ini adalah kode program yang mengimplementasikan algoritma djikstra dalam bahasa C (Scvalex dikutip dalam Anis Cherid,2009)

```
# include <stdio.h>
# include <conio.h>
# define graphsize 30
# define infinity graphsize * graphsize
# define max (a,b)((a>b)?(a):(b))
Int e;
Int n;
Long dist[graphsize] [graphsize];
Long d [graphsize];
Int prev [graphsize] [graphsize+1]
Void printd()
{
Int I
Printf(“Distance:\n”)
For(i=1 ti i<=n;++i)
Printf(%10d”,i);
Printf(“\n”);
For (i=1;i<=n;++i)
{
Printf(“%10id”,d[i]); }
Printf(“\n”);
Printf (“Press any key to continue….”)
Getch()
Printf (“\n\n\n”);
Void printpath (int dest, int depth)
{
Int I,j;
Printf(“-%d\n”,dest);
For (i=1;i<=prev[dest][0];++i)
{
For (j=0;j<=depth;++j)
Printf(“|”);
Printpath(prev[dest][i],depth+1);
}
}
Void dijkstra (int s)
{
```

```
Int I,k,mini;
Int visited[graphsize];
For (i=1;i<=n;++i)
{
d[i] = infinity;
prev[i][0]=0;
visited[i]=0;
}
d[s]=0
for(k=1;k<=n;++k)
{
Mini=-1;
For(i=1;i<=n;++i)
If(!visited[i]&&((mini==-1)||(d[i]<d[mini])))
Mini=I;
Visited[mini]=1;
For (i=1;i<=n,++i)
If (dist[mini][i])
{
If(d[mini]+dist[mini][I]<d[i])
d[i]=d[mini]+dist[mini][i];
prev[i][0]=1;
prev[i][1]=mini;
}
Else if (d[mini]+dist[mini][i]==d[i])
{
++prev[i][0];
Prev[i](prev[i][0])=mini;
}
}
}
}
Int main(int argc,char*argv[])
{
Int i,j,u,v,w;
File *fin=fopen("dist.txt","rt");
Fscanf(fin,"%d",&e);
For(i=0;i<e;++i)
For(j=0;j<e;++j)
dist[i][j]=0;
n=-1;
for(i=0;i<=e;++i)
{
Fscanf(fin,"%d%d%d",&u,&v,&w);
Dist[u][v]=w;
N=max(u,max(v,n));
}
Fclose(fin);
Dijikstra(1);
printD()
```

```
for(i=1;i<=n;++i)
{
Printf(“path to %d:\n”,i);
Printpath(2,0)
Printf(“Press any key to continues…”)
Getch()
Printf (“\n\n\n”);
```

METODE PENELITIAN

Metode yang digunakan untuk penelitian ini adalah metode waterfall atau sekarang disebut dengan the linier sequential model yang terdiri dari 5 tahap yaitu:

1) Pengumpulan Bahan
   Pengumpulan bahan-bahan yang berhubungan dengan implementasi dari aplikasi ini seperti materi Graf, Algoritma Djiktra, Grafik di VB
2) Analisis
   Tahap analisa terhadap beberapa langkah-langkah yang akan dilakukan sistem pada proses, seperti mengurai konsep algoritma Djiktra dalam menentukan lintasan terpendek
3) Desain
   Dari langkah-langkah yang sudah ditentukan dari hasil analisa, maka pada proses desain akan dilakukan pemodelan sistem.
4) Pengkodean
   Hasil desain akan ditulis ke dalam bahasa pemrograman untuk menghasilkan perangkat lunak yang diinginkan.
5) Pengujian
   Tahap ini akan digunakan untuk menguji aplikasi yang telah dibangun. Melalui tahap ini akan dapat diketahui apakah fungsi-fungsi pada aplikasi yang telah dibangun berjalan sesuai dengan yang diharapkan.

HASIL DAN PEMBAHASAN

Pengujian Data dalam Grid
Penambahan Data Titik

Skenario Pengujian

- Saat user membuat titik dengan menekan tombol mouse pada layar, baris pada grid kordinat bertambah dan data titik berupa no titik, kordinat X, kordinat Y ditampilkan pada grid tersebut. Selain itu baris dan kolom pada grid matriks bertambah dengan jumlah baris dan kolom sesuai dengan banyaknya titik yang dibuat. Header baris dan kolom adalah no titik dan isi pada grid matriks kosong.

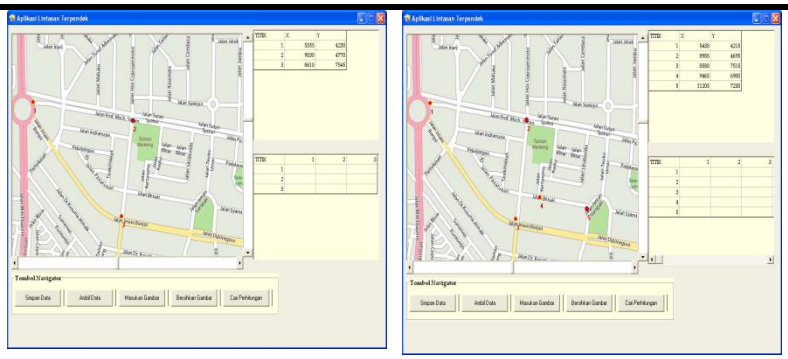
Hasil yang diharapkan

- Baris pada grid kordinat bertambah sesuai dengan jumlah titik dan ditampilkan data berupa no titik, kordinat X dan kordinat Y pada kolom yang ada. Baris dan kolom pada grid matriks bertambah sesuai dengan jumlah titik yang dibuat.

Hasil pengujian

Berhasil, baris pada grid kordinat bertambah sesuai dengan jumlah titik yang dibuat. Data titik berupa no titik, kordinat X dan kordinata Y tampil pada kolom yang ada.

- Baris dan kolom pada grid matriks bertambah sesuai dengan sama sesuai dengan jumlah titik yang dibuat. Header baris dan kolom adalah no titik.



Gambar 5. Penambahan Data Titik

Keterangan Gambar 5.15 Penambahan Data Titik:

- Semula data pada grid kordinat hanya titik 1,titik 2 dan titik 3 bertambah menjadi titik 4 dan titik 5. Data pada grid matriks semula hanya 3 titik menjadi 5 titik.

Penambahan Data Garis

Skenario Pengujian

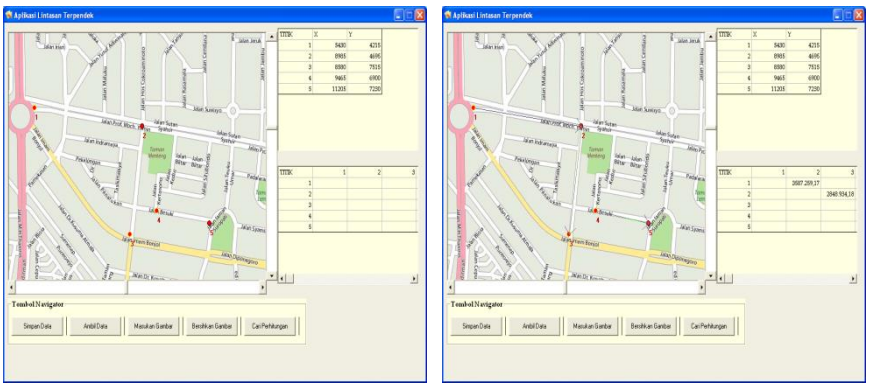
- User menekan tombol mouse pada titik 1 kemudian menarik ke titik tujuan lalu melepas tombol mouse sehingga terbentuk garis yang menghubungkan keduanya. Pada saat garis terbentuk, data jarak garis tersebut dan no garis ditampilkan pada grid matriks. Data tersebut ditampilkan pada baris yang sesuai dengan no titik 1 dan pada kolom yang sesuai dengan no titik tujuan.

Hasil yang diharapkan

- Terbentuk garis pada titik yang dihubungkan. Data berupa jarak garis dan no garis ditampilkan pada grid matriks di baris yang sesuai dengan no titik asal dan pada kolom yang sesuai dengan no titik tujuan.

Hasil pengujian

- Berhasil terbentuk garis dan data jarak beserta no garis ditampilkan pada grid matriks di baris dan kolom sesuai dengan no titik asal dan no titik tujuan.



Gambar 6. Penambahan Data Garis

Keterangan Gambar 6 Penambahan Data Garis

- Semula data pada grid matriks data kosong menjadi terdapat isi jarak dan no garis berdasarkan garis yang dibuat pada layar.

Pengujian Penyimpanan Data dalam File Teks

Perekaman Data ke dalam File Teks

Skenario Pengujian

- User menekan tombol navigator simpan data kemudian meyimpan data dengan format .txt

Hasil yang diharapkan

- Data tersimpan pada file teks dalam bentuk .txt. pada file tersebut tersimpan data dengan format:
  Titik Titik | Kordinat x dan kordinat y dari titik yang terhubung garis.
  Matriks | No titik yang terhubung garis beserta jaraknya.
  File | Alamat directori file dan nama file peta.

Hasil pengujian

- Berhasil tersimpan data dalam bentuk files teks .txt dengan format :
  Titik Titik | kordinat x dan kordinat y dari titik yang terhubung garis.
  Matriks | no titik yang terhubung garis beserta jaraknya.
  File | alamat directori file disimpan dan nama file peta.

Pengambilan Data dari File Teks

Skenario Pengujian

- User menekan tombol navigator ambil data dan memilih file yang akan ditampilkan.

Hasil yang diharapkan

- File yang berisi data berupa data titik meliputi kordinat x dan kordinat y serta jarak titik dan no garis akan ditampilkan pada grid masing-masing.
  Data titik berupa no titik , kordinat x dan kordinat y akan tampil di grid kordinat.Data jarak antara titik yang terhubung dan no garis akan tampil di grid matriks. Gambar peta lengkap dengan titik dan garis akan muncul dilayar.

Hasil pengujian

- Data titik berupa no titik, kordinat x dan y tampil pada grid kordinat.
- Data jarak dari tiap titik yang terhubung dan no garis tampil pada grid matriks.
- Gambar beserta titik dan garis tampil di layar.

Pengujian Algoritma Djistra

Pencarian Jarak dan Lintasan Terpendek

Berikut adalah contoh studi kasus untuk menjelaskan pencarian jarak terpendek.Pada layar ditampilkan bagian peta Jakarta Pusat. Tetapkan titik awal adalah Bunderan HI (Titik 1) dan titik tujuan adalah Taman Suropati (Titik 4). Untuk bisa sampai ke Taman Suropati ada beberapa pilihan jalan yang bisa dilalui. Jalan tersebut digambarkan dengan titik titik yang terhubung garis, yaitu :

a) Jalan Prof Moch Yamin (Titik 1 ke Titik 2)
b) Jalan Sutan Syahrir (Titik 2 ke Titik 3)
c) Jalan Teuku Umar (Titik 3 ke Titik 4)
d) Jalan Imam Bonjol (Titik 1 ke Titik 5, Titik 5 ke Titik 6, Titik 6 ke Titik 7)
e) Jalan Taman Suropati (Titik 7 ke Titik 4)

Informasi Data Titik

Data Kordinat X dan Kordinat Y:

Titik 1 = 5205 , 4305
Titik 2 = 8940 , 4680
Titik 3 = 11940 , 5190
Titik 4 = 11760 , 6795
Titik 5 = 6585 , 6120
Titik 6 = 8505 , 7665
Titik 7= 11160 , 7905

Menghitung Jarak 2 Titik :

Titik1 ke Titik 2 = $\sqrt{(X2-X1)^2 + (Y2-Y1)^2}$
= $\sqrt{(8940-5205)^2 + (4680-4305)^2}$
= $\sqrt{13950225 + 140625}$
= $\sqrt{14090850}$
= 3753,778

Dengan cara yang sama didapat :
Titik 1 ke Titik 5 = 2280.049
Titik 2 ke Titik 3 = 3043.041
Titik 3 ke Titik 4 = 1615.062
Titik 5 ke Titik 6 = 2464.432
Titik 6 ke Titik 7 = 2665.825
Titik 7 ke Titik 4 = 1261.784

Perhitungan Lintasan Terpendek

1. Titik 1 ke Titik 2 ke Titik 3 ke Titik 4
   3753,778+3043.041+1615.062= 8411.881
2. Titik 1 ke Titik 5 ke Titik 6 ke Titik 7 ke Titik 4
   2280.049+2464.432+2665.825+1261.784= 8671.09

Dari hasil diatas diketahui bahwa lintasan terpendek melalui 1-2-3-4, dengan jarak *8411.881*. Lintasan dari Bunderan HI ke Taman Suropati melalui *Jalan Prof Moch Yamin, Jalan Sutan Syahrir dan Jalan Teuku Umar*

Skenario Pengujian

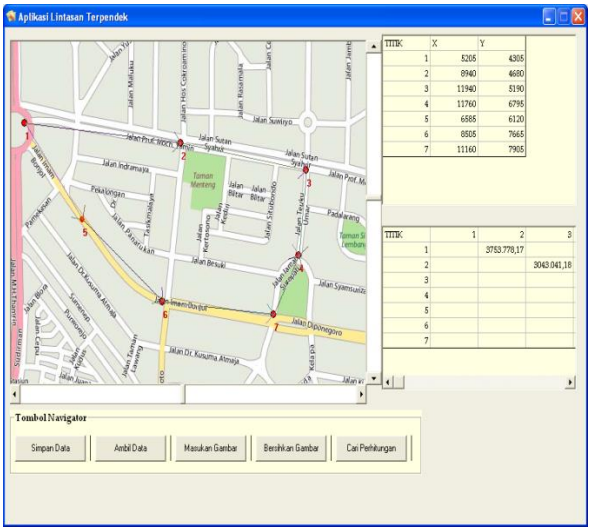
- User menentukan titik asal dan tujuan yang ingin dicapai. Pada contoh kasus diatas titik asal adalah Bunderan HI dan titik tujuan adalah Taman Suropati

Hasil yang diharapkan

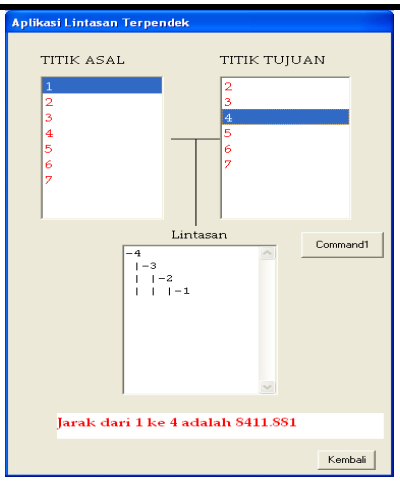
- Didapat jarak dan lintasan terpendek dari Bunderan HI ke Taman Suropati melalui beberapa titik (jalan) yang ada.

Hasil pengujian

- Diperoleh jarak dan lintasan terpendek dari Bunderan HI ke Taman Suropati yaitu melalui titik 1-2-3-4, dengan jarak 8411.881, melalui Jalan Prof Moch Yamin, Jalan Sutan Syahrir dan Jalan Teuku Umar



Gambar 7. Pencarian Jarak dan Lintasan Terpendek



Gambar 8. Perhitungan Algoritma Djikstra

Keterangan Gambar 5.17 Perhitungan Algoritma Djikstra

- User pilih titik 1 pada list titik asal dan pilih titik 4 pada titik tujuan maka akan ada informasi lintasan pada list lintasan dan total jarak.

Petunjuk Penggunaan Program

Dalam menggunakan aplikasi ini, beberapa petunjuk yang harus di lakukan adalah:

1) Tekan tombol mouse bagian kiri (klik tombol kiri mouse) pada layar untuk membuat titik.
2) Klik tombol kiri mouse pada tepat daerah titik, kemudian geser ke arah tepat titik lain lalu lepas tombol mouse adalah untuk membuat garis antara titik.
3) Tekan tombol shift + klik tombol kiri mouse pada tepat daerah titik adalah untuk menghapus titik. Apabila titik terhapus, semua garis yang terhubung pada titik tersebut akan terhapus.
4) Tombol Simpan Data berfungsi untuk menyimpan data titik, garis, jarak dan gambar pada layar ke dalam file teks.
5) Tombol Ambil Data berfungsi untuk menampilkan file teks yang berupa data titik, garis, jarak dan gambar ke layar dan grid yang sesuai.
6) Tombol Masukan Gambar berfungsi untuk mengambil gambar dari directori dan menampilkannya pada layar.
7) Tombol Bersihkan Layar berfungsi untuk membersihkan layar dari gambar dan sekaligus menghapus informasi berupa data titik, garis dan jarak pada grid yang ada.
8) Tombol Cari Perhitungan berfungsi untuk pindah ke form 2 yang berisi list titik asal dan list titik tujuan untuk mencari jarak dan lintasan terpendek.
9) Klik pada list titik asal untuk menentukan titik yang menjadi titik asal.
10) Klik pada list titik tujuan untuk menentukan titik yang menjadi titik tujuan. Titik yang ada pada list titik tujuan adalah titik-titik yang terhubung dengan titik pada list titik asal secara langsung maupun tidak. Pada saat list titik tujuan dipilih , pada list lintasan akan ditampilkan lintasan terpendek untuk mencapai titik tujuan dari titik asal dan pada label hasil ditampilkan jarak terpendeknya.

KESIMPULAN DAN SARAN

Kesimpulan

Dari hasil penelitian, analisis, perancangan sistem, pembuatan program sampai tahap penyelesaian program, maka penulis dapat mengambil kesimpulan bahwa rancang bangun antarmuka interaktif berbasis grafik untuk pencarian rute terpendek dengan algoritma djikstra dapat melakukan beberapa hal yaitu :

1) Membuat titik dan garis pada layar secara interaktif dengan menggunakan mouse.
2) Menghapus titik dan garis pada layar secara interaktif dengan menekan tombol shif dan menggunakan mouse.
3) Menampilkan gambar pada layar, data kordinat titik pada grid kordinat , data jarak antar titik dan no garis pada grid matriks
4) Menyimpan data titik dan garis berupa kordinat titik, jarak antar titik dan nama gambar dalam bentuk file teks.
5) Melakukan perhitungan algoritma djikstra untuk menampilkan lintasan dan jarak terpendek berdasarkan jarak antar titik..

Saran

Adapun saran-saran yang penulis kemukakan untuk pengembangan aplikasi ini selanjutnya adalah sebagai berikut :

1) Aplikasi bisa menampilkan gambar secara online melalui web browser
2) Aplikasi memberikan informasi apabila nama file teks yang ditampilkan tidak sesuai dengan nama file teks yang telah tersimpan.
3) Aplikasi bisa menampilkan jarak lebih detail sesuai dengan skala peta sebenarnya.
4) Aplikasi secara otomatis bisa membedakan jalan, sungai, rumah dan lain-lain pada peta yang dimasukan.

DAFTAR RUJUKAN

Indra Yatini B. 2010, Flowchart, Algoritma dan Pemrograman menggunakan Bahasa C++ Bulider , Graha Ilmu, Jakarta

Newcobe, Richard. 2008, Sorting Algorithms In VB, Diakses pada tanggal 27 Februari 2020 dari Http://Codeguru.com

Rocky Mountain Computer Consulting Inc. 2001, Draw Arrow, Diakses pada tanggal 27 Februari 2020 dari http://Vb-Helper.com

Bobby.2005, Removing Duplicate Entries From Listbox/Combobox, Diakses pada tanggal 15 Maret 2020 dari http://XtremeVisualBasicTalk.com