

PENENTUAN JALUR TERPENDEK PADA PELAYANAN AGEN TRAVEL KHUSUS PENGANTARAN WILAYAH SEMARANG BERBASIS SIG DENGAN ALGORITMA BRANCH AND BOUND

Windi Rayina Rosa, Suhartono, Helmie Arif Wibawa
Program Studi Teknik Informatika, Universitas Diponegoro

ABSTRAK

Bagi perusahaan jasa transportasi, khususnya agen travel, permasalahan pemilihan jalur atau rute perjalanan sangat diperhatikan. Terutama rute yang lebih pendek pada umumnya akan menghasilkan biaya yang lebih sedikit dan waktu yang lebih singkat. Oleh karena itu diperlukan suatu cara untuk menentukan rute terpendek agar aspek optimalitas dari segi biaya dan waktu terpenuhi. Masalah penentuan jalur terpendek dapat diselesaikan dengan menggunakan algoritma Branch and Bound. Algoritma ini cukup baik dalam memberikan solusi optimal pada masalah pemilihan jalur terpendek. Dalam pemilihan jalur terpendek tersebut dikembangkan sebuah sistem informasi yang disebut Sistem Informasi Geografis Pencarian Jalur Terpendek (SIGPEJAP). Sistem ini dikembangkan dengan menggunakan metode Unified Process. Sistem yang dihasilkan dapat membantu agen travel dalam memilih rute terpendek yang sebaiknya dilewati oleh sopir.

Kata kunci: Sistem Informasi Geografis, Jalur/Rute, *Branch and Bound*, *Unified Process*.

I. PENDAHULUAN

Travel merupakan suatu pelayanan transportasi yang cukup sering digunakan karena pelayanannya yang nyaman dan terjangkau. Untuk mempermudah pelayanan bagi para penumpang dan agen travel, diperlukan suatu Sistem Informasi Geografis yang saat ini mulai banyak digunakan pada berbagai macam aplikasi yang berhubungan dengan data yang berbasis geografis.[3]

Sistem Informasi Geografis di sini pada dasarnya dibuat untuk membantu penentuan jalur terpendek yang dapat dilalui travel angkutan.

Penentuan jalur terpendek dicari menggunakan algoritma *Branch and Bound*. Berdasarkan riset pada tahun 1998 yang telah dilakukan oleh SEAS, *School of Engineering and Applied Science*, *Branch and Bound* merupakan algoritma yang *powerfull*, karena dapat diaplikasikan dalam berbagai masalah yang tidak dapat diselesaikan dengan menggunakan metode *greedy* dan pemrograman dinamis [2].

Berdasarkan latar belakang dapat dirumuskan masalah sebagai berikut bagaimana merancang dan membangun sistem informasi geografis yang dapat menentukan jalur terpendek yang menghubungkan semua titik wilayah tujuan travel dengan menggunakan algoritma *Branch and Bound*.

II. TINJAUAN PUSTAKA

2.1. Algoritma *Branch and Bound*

Algoritma *Branch and Bound* (B&B) merupakan suatu metode pencarian di dalam ruang solusi secara sistematis. Tiap simpul yang dibangkitkan akan diberikan nilai ongkos (*cost*). Untuk mempercepat pencarian solusi maka simpul berikut yang diekspansi tidak berdasarkan urutan pembangkitnya tetapi berdasarkan *cost* yang dimiliki oleh simpul-simpul hidup. *Cost* pada setiap simpul *i* menyatakan perkiraan ongkos termurah lintasan dari simpul *i* ke simpul solusi [1].

2.2. Skema Umum Algoritma *Branch And Bound*

Langkah-langkah pencarian solusi dengan menggunakan algoritma B&B [1]:

- 1) Simpul akar dimasukkan ke dalam antrian *Q*. Jika simpul akar adalah simpul solusi, maka solusi telah ditemukan dan pencarian berhenti.
- 2) Antrian *Q* diidentifikasi
 - a) Jika antrian *Q* kosong, maka solusi tidak ada dan pencarian berhenti
 - b) Jika antrian *Q* tidak kosong, maka dipilih dari antrian *Q* simpul *i* yang mempunyai nilai *c* paling kecil. Jika

terdapat beberapa simpul i yang memenuhi, maka dipilih satu secara sembarang.

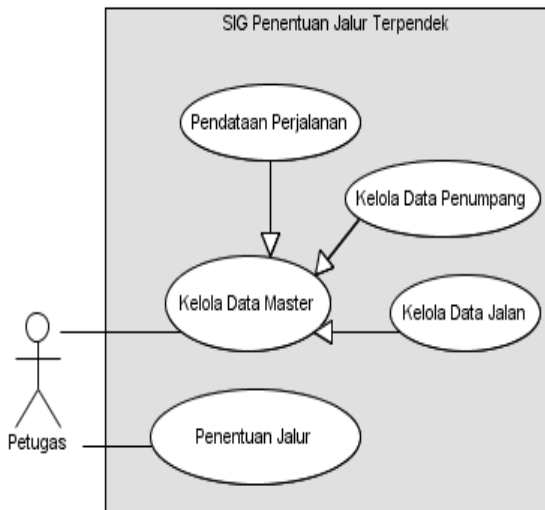
- 3) Simpul i diidentifikasi
 - a) Jika simpul i adalah simpul solusi, maka solusi telah ditemukan dan pencarian berhenti.
 - b) Jika simpul i bukan simpul solusi, maka semua anaknya dibangkitkan. Jika simpul tidak memiliki anak, maka kembali ke langkah 2.
- 4) Untuk setiap anak j dari simpul i , nilai c dihitung dan semua anak yang sudah dibangkitkan dimasukkan ke dalam antrian Q .
- 5) Kembali ke langkah 2.

III. PEMBAHASAN

Hasil dan Pembahasan sistem Informasi SIGPEJAP adalah sebagai berikut:

3.1. Sistem Use Case Diagram

Sistem ini hanya digunakan oleh satu macam *user*, yaitu petugas agen travel. Petugas ini bertanggung jawab untuk semua *use case*, yaitu Pendataan Perjalanan, Kelola Data Penumpang, Kelola Data Jalan, dan Penentuan Jalur, lihat gambar 1

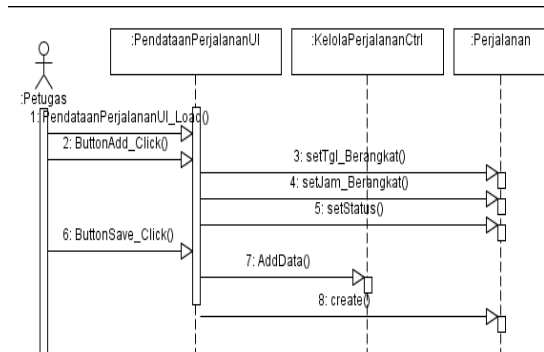


Gambar 1. Sistem Use Diagram

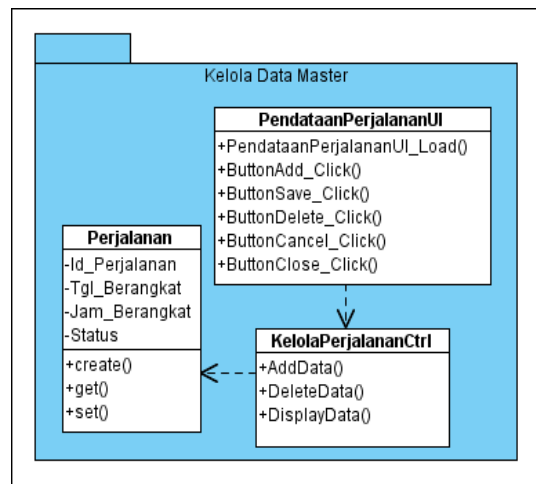
3.2. Perancangan Realisasi Use case

Realisasi *use case* merupakan kolaborasi objek perancangan dan *class* perancangan yang merealisasikan *use case*. Setiap *use case* direalisasikan dengan menggunakan *sequence diagram* dan *class diagram*. Berikut realisasi *use*

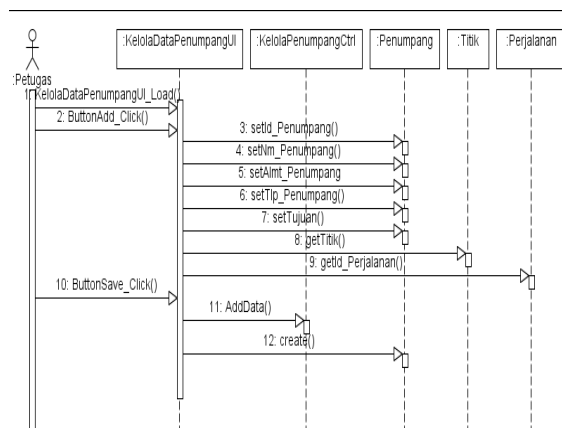
case tahap perancangan untuk *use case* Pendataan Perjalanan (gambar 2 dan 3) Kelola Data Penumpang (gambar 4 dan 5), dan Penentuan Jalur (gambar 6 dan 7).



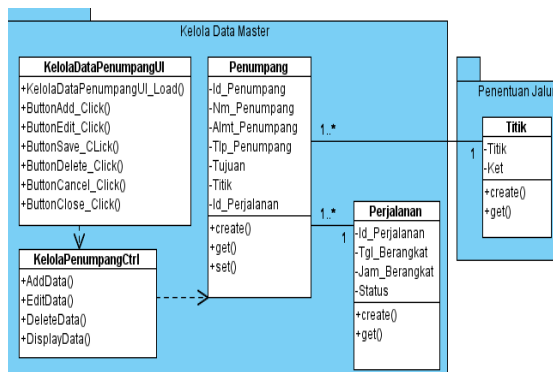
Gambar 2. Sequence Diagram Pendataan Perjalanan



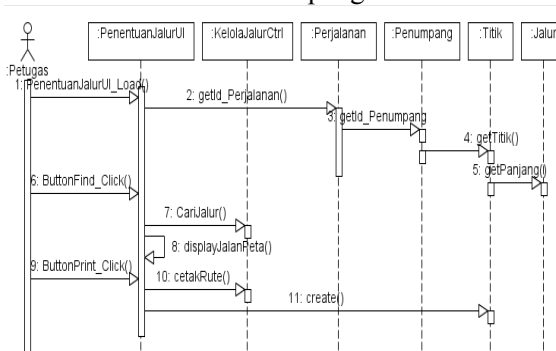
Gambar 3. Class Diagram Pendataan Perjalanan



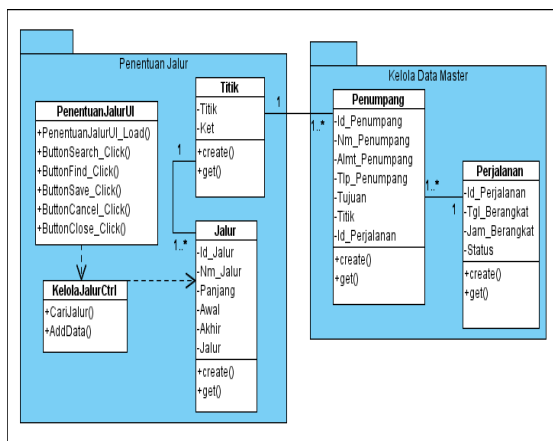
Gambar 4. Sequence diagram Kelola Data Penumpang



Gambar 5. Class Diagram Kelola Data Penumpang



Gambar 6. Sequence Diagram Penentuan Jalur



Gambar 7. Class Diagram Penentuan Jalur

3.3. Perancangan Algoritma

Perancangan algoritma digunakan untuk menjelaskan rincian perilaku dari class. Berikut ini adalah perancangan algoritma untuk class Penentuan Jalur dari operasi CariJalur().

Algoritma:

```
/* mencari jalur terpendek antara satu titik ke
semua titik tujuan */
/* CariJarak */
```

```
i,found = 0
while i<N & found=0
if (T[i].start = start)&&(T[i].destiny =
destiny)then
    jarak = T[i].jarak
    found = 1
end if
if found<=0 then
    jarak = -1
end if
end while
```

```
/* BuatMatriks */
for i<=0 to Nkota do
for j<=0 to Nkota do
if i=j
    M[i,j]←-1
else
M[i, j] = findJarak
end if
end for
end for
/* ReduksiMatriks */
/* reduksi baris */
for i<=0 to n do
j<=0; min←99999999; stop←0
while j<=m & stop=0
if T[i, j]←0 then
    stop = 1
else if T[i, j]<min & T[i, j]!=-1
    min = T[i, j]
end if
end while
if
min!=99999999 & stop!=1
red += min
for j<=0 to m do
if T[i, j]!=-1
    T[i, j] -= min
end if
end for
end if
end for
```

```
/* reduksi kolom */
for j<=0 to m do
i<=0; min←99999999; stop←0;
while i<=n & stop=0
if T[i, j]←0
    stop = 1
else if T[i, j]<min & T[i, j]!=-1
    min = T[i, j]
end if
end while
```

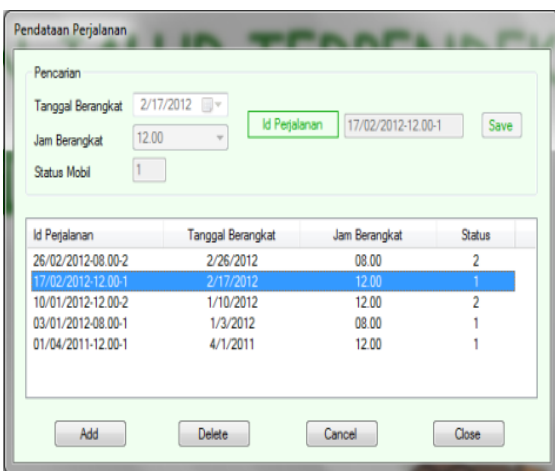
```

if min!=99999999 & stop!=1
red += min
for i<-0 to n do
if T[i, j]!=-1
T[i, j] -= min
end if
end for
end if
end for
/* Proses child */
copyMatriks
c = T[p, q]
for j<-0 to m do
M[p, j] = -1
end for
for i<-0 to n do
M[i, q] = -1
end for
M[q, 0] = -1
red = reduksiMatriks
r = root + c + red
    
```

3.4. Implementasi

Implementasi halaman Pendataan Perjalanan

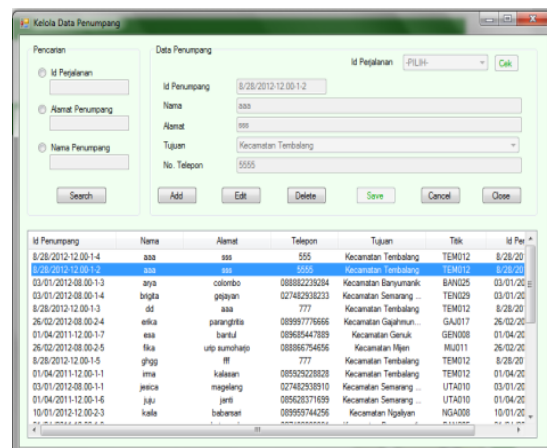
Pada implementasi ini, Petugas melakukan penambahan data perjalanan dengan mengisi tanggal keberangkatan, jam berangkat, dan status mobil yang akan digunakan untuk melakukan perjalanan tersebut. Kemudian data perjalanan akan muncul pada *listview* dan disimpan dalam *database* dengan menekan tombol 'Save'. Id Pada implementasi ini gambar 8



Gambar 8. Implementasi Halaman Pendataan Perjalanan

Implementasi Halaman Kelola Data Penumpang

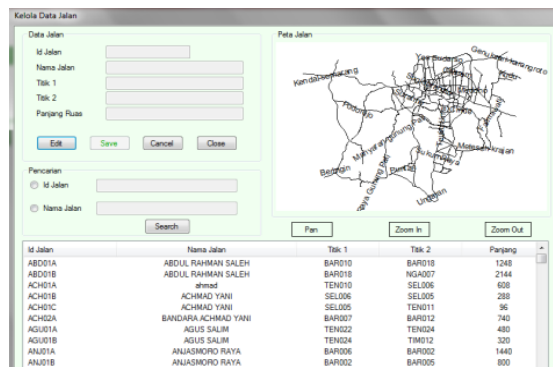
Implementasi halaman Kelola Data Penumpang dapat dilihat pada gambar 9. Petugas dapat melakukan penambahan data dengan menekan tombol 'Add' dan mengisi setiap *textbox* yang dibutuhkan untuk mendata penumpang. Petugas juga dapat melakukan pengubahan data dengan memilih data pada *listview* kemudian menekan tombol 'Edit' dan menekan tombol 'Save' untuk menyimpannya. Tombol 'Delete' digunakan untuk menghapus data penumpang apabila terdapat penumpang yang tidak jadi melakukan pemesanan perjalanan.



Gambar 9. Implementasi Halaman Kelola Data Penumpang

Implementasi Halaman Kelola Data Jalan

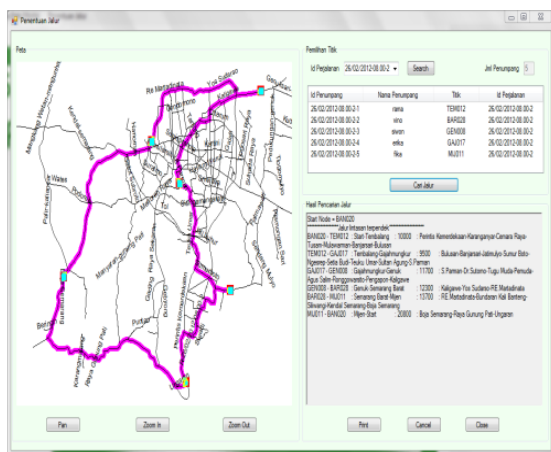
Pada implementasi ini petugas dapat melakukan pengubahan data dengan memilih data pada *listview* kemudian menekan tombol 'Edit' dan selanjutnya 'Save' lihat gambar 10.



Gambar 10. Implementasi Halaman kelola Jalan

Implementasi halaman Penentuan Jalur

Implementasi dapat dilihat pada gambar 11. Petugas melakukan pencarian data penumpang yang seperjalanan dengan menggunakan id perjalanan. Setelah data penumpang ditemukan, petugas menekan tombol ‘Cari Jalur’ untuk melakukan pencarian jalur terpendek yang dapat ditempuh. Pada implementasi ini digunakan algoritma Branch and Bound



Gambar 11. Implementasi Halaman Penentuan Jalur

IV. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil adalah Sistem Informasi Geografi berbasis GIS untuk Pencarian Jalur Terpendek (SIGPEJAP) telah berhasil dibangun, sistem dibangun berdasarkan algoritma Branch and Bound sehingga dapat

memberikan keuntungan pada agen travel dalam pengantaran penumpang untuk penentuan rute terpendek yang berdampak pada penghematan biaya transportasi. Dan memperpendek waktu perjalanan.

Saran untuk pengembangan sistem ini adalah titik pengantaran penumpang dibuat sesuai alamat tujuan penumpang, pencarian jarak antar titik tujuan dapat dicari menggunakan aplikasi yang lebih canggih lagi, seperti aplikasi berbasis *mobile*. Selain itu perhitungannya tidak hanya berdasarkan jarak saja, tetapi juga kepadatan lalu lintas, perbedaan jalan kecil dan jalan besar, serta memperhatikan aspek baik buruknya keadaan jalan.

V. DAFTAR PUSTAKA

- [1] Munir R., 2003, “*Algoritma Branch and Bound*”, (<http://kur2003.if.itb.ac.id/transbahan%20kuliah%20ke-11.pdf>) diakses pada tanggal 9 Juli 2009
- [2] Oemaryadi C. S., 2008, “*Aplikasi Algoritma B&B untuk Memperoleh Poin Maksimum pada Permainan “Diner Dash”*”, (<http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2007-2008/Makalah2008/MakalahIF2251-2008-040.pdf>) diakses pada tanggal 9 Juli 2009
- [3] Nuarsa I. W., 2004, “*Mengolah Data Spasial dengan MapInfo Professional*”, Penerbit Andi, Yogyakarta

