BAB III

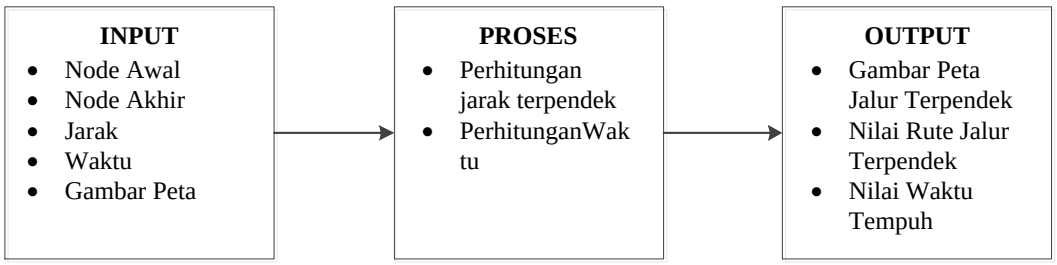
METODOLOGI PENELITIAN

3.1 Analisis Masalah

Analisis sistem bertujuan untuk melakukan identifikasi persoalan - persoalan yang muncul dalam pembuatan sistem, hal ini dilakukan agar pada saat proses perancangan program simulasi pencarian rute terpendek tidak terjadi kesalahan – kesalahan yang berarti sehingga sistem dapat berjalan dengan baik dan selesai tepat pada waktu yang telah ditentukan. Dalam analisis sistem ini, sistem yang akan di analisa meliputi, analisis kebutuhan sistem, spesifikasi aplikasi, dan lingkungan operasi.

3.1.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem membahas secara garis besar kebutuhan sistem. Sistem yang ada disini dapat memberikan suatu penyeleseian masalah dengan menghasilkan gambar peta yang telah dipersiapkan sebelumnya serta jarak terpendek dan waktu tempuh, seperti digambarkan pada gambar 3.1.



Gambar 3.1 Rancangan Sistem

Dalama rancangan sistem, secara garis besar langkah – langkah pembuatan sistem ini pertama – tama adalah memasukan node awal, node akhir, jarak, waktu

tempuh dan gambar peta yang sudah di analisis. Kemudian dari keseluruhan data tersebut akan dicari nilai rute terpendek dengan menggunakan Algoritma Greedy dan waktu tempuh. Dari keseluruhan hasil analisis akan dilakukan sistem simulasi untuk menghasilkkan gambar peta rute terpendek dari data gambar yang sudah ditentukan.

3.1.2 Spesifikasi Aplikasi

Berikut ini adalah spesifikasi aplikasi dari program simulasi pencarian rute terpendek :

1. Sistem optimasi perjalanan rute angkot Cicaheum – Ciroyom di Kota Bandung menentukan rute terpendek dengan menggunakan Algoritma Greedy akan memberikan data dan keluaran.
2. Memberikan informasi rute terpendek dari titik awal keberangkatan menuju titik tujuan beserta nilai perhitungan jarak dan waktu tempuh.
3. Memberikan informasi node awal, node akhir, jalur yang ditempuh, jarak tempuh, waktu tempuh dan data antar node yang merupakan jalur yang terpendek untuk dilewati.
4. Memberikan informasi gambar peta jarak yang ditempuh.
5. Memberikan informasi *help* untuk cara penggunaan program simulasi pencarian rute terpendek.

3.1.3 Lingkungan Operasi

Untuk membangun program simulasi rute angkot cicaheum ciroyom menuju jalan dipati ukur, dibutuhkan lingkungan operasi sebagai berikut:

a. Sistem Operasi *Windows* 7

Sistem operasi *Windows* 7 ini di pilih karena sudah banyak dikenal sehingga mudah dalam pengoperasiannya dan lebih familiar.

b. JavaScript

Bahasa pemograman ini digunakan untuk membuat program simulasi rute terpendek angkot Cicaheum Ciroyom sampai tujuan Dipatiukur, sehingga dari sinilah pengguna dapat menggunakan apliksi ini pada komputer.

3.2 Analisis Algoritma *Greedy*

3.2.1 Algoritma *Greedy*

Algoritma *Greedy* adalah algoritma yang memecahkan masalah langkah demi langkah dan merupakan salah satu metode dalam masalah optimasi. Algoritma *Greedy* membentuk solusi langkah per langkah sebagai berikut:

1. Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya.
2. Pendekatan yang digunakan didalam algoritma *Greedy* adalah membuat pilihan yang terlihat memberikan perolehan terbaik yaitu dengan membuat pilihan *optimum local* pada setiap langkah dan diharapkan akan mendapatkan solusi *optimum global.*

Algoritma Greedy didasarkan pada pemindahan edge (arc) per edge (arc) dan pada setiap langkah yang diambil tidak memikirkan konsekuensi ke depan, Greedy tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada serta sebagian masalah Greedy tidak selalu berhasil memberikan solusi yang benar-benar *optimum* tapi pasti memberikan solusi yang mendekati nilai *optimum.* Masalah optimasi dalam konteks Algoritma Greedy disusun oleh elemen-elemen sebagai berikut:

1. Himpunan kandidat

   Himpunan ini berisi elemen-elemen yang memiliki peluang untuk pembentuk solusi. Pada persoalan lintasan terpendek dalam *graf*, himpunan kandidat ini adalah himpunan simpul dari *graf* tersebut.

2. Himpunan solusi

   Himpunan ini berisi solusi dari permasalahan yang diselesaikan dan elemennya terdiri dari elemen dalam himpunan kandidat, namun tidak semuanya atau dengan kata lain himpunan solusi ini adalah bagian dari himpunan kandidat.

3. Fungsi seleksi (*selection function*)

   Fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

4. Fungsi kelayakan (*feasible*)

   Fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

5. Fungsi obyektif

   Fungsi yang memaksimumkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain). Tujuannya adalah memilih satu saja solusi terbaik dari masing-masing anggota himpunan solusi.

3.2.2 Cara Kerja Algoritma *Greedy*

Diberikan sebuah graph berbobot G (V,E). Tentukan lintasan dari simpul awal a, ke setiap simpul lainnya di G. Asumsi bahwa bobot semua edge(arc) bernilai positif. Algoritma *greedy* untuk mencari lintasan terpendek dapat dirumuskan sebagai berikut :

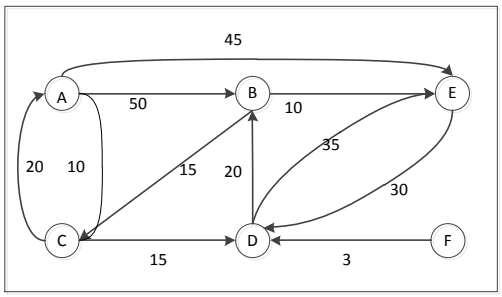
1. Periksa semua sisi yang langsung bersesuaian dengan simpul a. Pilih sisi yang bobotnya terkecil. Sisi ini menjadi lintasan terpendek pertama, sebut saja L(1).
2. Tentukan lintasan terpendek ke dua dengan cara sebagai berikut :
   a. Hitung d(i) = panjang L(1) + bobot sisi dari simpul akhir L(1) ke simpul i yang lain. Pilih d(i) yang terkecil.

b. Bandingkan d(i) dengan bobot sisi (a, i) lebih kecil daripada d(i), maka L(2) = L(1) U (sisi dari simpul akhir L(i) ke simpul i)

3. Dengan cara yang sama, ulangi langkah (2) untuk menentukan lintasan terpendek berikutnya.

Contoh Kasus :

Suatu Graph dengan 6 simpul dan masing – masing ruasnya mempunyai nilai tersendiri,



Gambar 3.2 Graf Berarah dan berbobot dengan 6 simpul

Jalur terpendek dari A ke semua tempat tujuan pada graph G (gambar 3.2) adalah :

Tabel 3.1
Jalur dari simpul A ke semua tempat tujuan

Jalur	Panjang
1. A – C	10
2. A – C – D	25
3. A – C – D – B	45
4. A – E	45

Proses untuk mencapai simpul tujuan dari simpul awal (A) dari setiap jalur pada tabel 3.1 adalah :

1. A – C diperoleh melalui jalur – jalur

   A – C : 10 atau

   A – B – C : 50 + 10 = 60 atau

A – B – E – D – B – C : 50 + 10 + 30 + 20 + 15 = 125 atau

A – E – D – B – C : 45 + 30 + 20 + 15 = 110

Solusi optimal = A – C dengan panjang 10

2. A – D diperoleh melalui jalur – jalur

A – C – D : 10 + 15 = 65 atau

A – B – E – D : 50 + 10 + 30 = 90 atau

A – B – C – D : 50 + 15 + 15 = 80 atau

A – E – D : 45 + 30 = 75

Solusi optimal = A – C – D dengan panjang 65

3. A – B diperoleh melalui jalur – jalur

A – B : 50 atau

A – C – D – B : 10 + 15 + 20 = 45 atau

A – E – D – B : 45 + 30 + 20 = 95

Solusi optimal = A – C – D – B dengan panjang 45

4. A – E diperoleh melalui jalur – jalur

A – E : 45 atau

A – B – E : 50 + 10 = 60 atau

A – C – D – B – E : 10 + 15 + 20 + 10 = 55 atau

A – C – D – E : 10 + 15 + 35 = 60

Solusi optimal = A – E dengan panjang 45

3.2.3 *Pseudocode* Algoritma *Greedy*

Pseudo-code algoritma *greedy* adalah sebagai berikut:

```
procedure greedy(input C: himpunan_kandidat;
                 output S : himpunan_solusi)
{ menentukan solusi optimum dari persoalan optimasi dengan
  algoritma greedy
  Masukan: himpunan kandidat C
  Keluaran: himpunan solusi S
}
Deklarasi
   x : kandidat;
Algoritma:
   S←{}  { inisialisasi S dengan kosong }
   while (belum SOLUSI(S)) and (C ≠ {} ) do
      x←SELEKSI(C);    { pilih sebuah kandidat dari C}
      C← C - {x}        { elemen himpunan kandidat berkurang
   satu }
     if LAYAK(S ∪ {x}) then
        S←S ∪ {x}
     endif
    endwhile
   {SOLUSI(S) sudah diperoleh or C = {} }
```

3.3 Analisis Perangkat Lunak dan Perangkat Keras

3.3.1 Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan program simulasi pencarian rute terpendek yaitu :

Sistem Operasi : *Windows 7*

Bahasa pemrograman : *JavaScript*

3.3.2 Perangkat Keras

Perangkat keras yang dipergunakan untuk pembuatan program simulasi pencarian rute terpendek ini adalah sebagai berikut :

1. Unit komputer lengkap (laptop).
2. *Processor* Amd Turion – X2.
3. RAM 1 Gb.
4. VGA 256 Mb.
5. Kapasitas *Harddisk* 250 Gb.