BAB III

METODOLOGI PENELITIAN

3.1 Analisis Masalah

Analisis sistem bertujuan untuk melakukan identifikasi persoalan - persoalan yang muncul dalam pembuatan sistem, selain itu hal ini juga dilakukan agar mampu meminimalisir kesalahan pada saat proses perancangan program simulasi pencarian rute terpendek sehingga dalam pelaksanaannya bisa sesuai dengan waktu yang telah ditentukan serta dapat berjalan dengan baik. Dalam analisis sistem ini, sistem yang akan di analisa meliputi, analisis kebutuhan sistem, spesifikasi aplikasi, dan lingkungan operasi.

3.1.1 Analisis Kebutuhan Sistem

Pada tahapan Analisis kebutuhan sistem, penulis memaparkan secara garis besar tentang kebutuhan sistem yang akan dibangun. Dari tahapan perencanaan, analisis, perancangan, dan pembangunan yang telah dilakukan nanti, Sistem yang dihasilkan nanti diharapkan mampu memberikan suatu penyeleseian masalah dengan menampilkan gambar peta disertai animasi gambar bergerak dari taxi sesuai rute yang diinginkan dengan jarak terpendek dan waktu tempuh. Berikut gambaran dari alur proses sistem yang akan dibuat gambar 3.1.



Gambar 3.1 Rancangan Sistem

Dalam tahapan proses perancangan sistem yang akan dibuat ada langkah – langkah yang dilakukan yaitu mulai dari menginput node akhir/tujuan, jarak, waktu tempuh, dan waktu keberangkatan gambar peta yang sudah di analisis, dan animasi gambar dari proses pencarian rute. Kemudian dari keseluruhan data tersebut akan dicari beberapa rute terdekat dengan node awal, selanjutnya pengelompokan rute dari awal sampe ke tujuan, dan kemudian dilakukan penjumlahan dari jarak dan waktu untuk selanjutnya dilakukan pemilihan hasil data dengan jumlah jarak terpendek, dan waktu terdekat, serta banyak jalur kemacekan. 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 Taksi dari Terminal Leuwi Panjang – Dipati Ukur di Kota Bandung menentukan rute terpendek dengan

menggunakan Algoritma *Floyd Warshall* 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 disertai animasi bergerak dari gambar takai dari jarak yang ditempuh.

3.1.3 Lingkungan Operasi

Dalam proses pembuatan program simulasi jalur taksi terminal leuwi panjang menuju Dipati Ukur pencarian rute terpendek didukung dengan lingkup operasi yang sesuai dengan kebutuhan. Berikut lingkungan operasi :

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 *Floyd Warshall*

Untuk mengetahui suatu metode yang akan digunakan dalam proses penelitian maka terlebih dahulu harus dilakukan analisis terhadap metode yang akan digunakan guna mendukung kesesuaian dalam hal implementasi terhadap masalah yang ada. Dalam hal ini penulis mencoba menganalisis algoritma Floyd Warshall yg dimana digunakan penulis sebagai metode untuk pencarian rute terpendek.

3.2.1 Algoritma *Floyd Warshall*

Algoritma Floyd Warshall Merupakan salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu.

3.2.2 Definisi Strategi Algoritma Floyd Warshall

Hal yang membedakan pencarian solusi menggunakan pemrograman dinamis (Warshall) dengan algoritma *greedy* adalah, bahwa keputusan yang diambil pada tiap tahap pada algoritma *greedy* hanya berdasarkan pada informasi yang terbatas, sehingga hanya nilai optimum yang diperoleh pada saat itu. Jadi pada algoritma greedy, kita tidak memikirkan konsekuensi yang akan terjadi seandainya kita memilih suatu keputusan pada suatu tahap.

Jika dibandingkan dengan jenis algoritma *Greedy* dalam beberapa kasus, Algoritma *Greedy* gagal memberikan solusi terbaik karena kelemahan yang dimilikinya tadi. Di sinilah peran pemrograman dinamis yang mencoba untuk

memberikan solusi yang memiliki pemikiran terhadap konsekuensi yang ditimbulkan dari pengambilan keputusan pada suatu tahap.

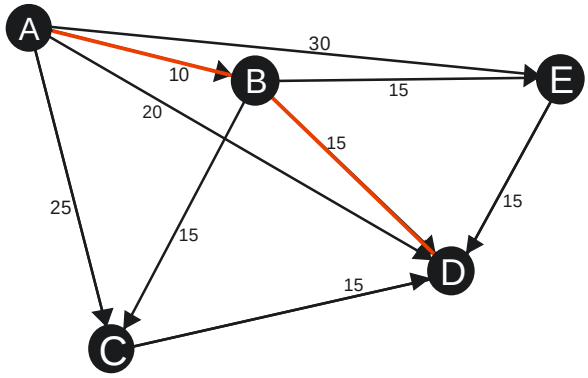
Pemrograman dinamis mampu :

- Mengurangi pengenumerasian (Pendaftaran) keputusan yang tidak mengarah ke solusi.
- Prinsip yang dipegang oleh pemrograman dinamis adalah prinsip optimalitas, yaitu jika solusi total optimal, maka bagian solusi sampai suatu tahap (misalnya tahap ke-i) juga optimal.

3.2.3 Cara Kerja Algoritma *Floyd Warshall*

Algoritma *Floyd-Warshall* membandingkan semua kemungkinan lintasan pada graf untuk setiap sisi dari semua simpul. Hal tersebut bisa terjadi karena adanya perkiraan pengambilan keputusan (pemilihan jalur terpendek) pada setiap tahap antara dua simpul, hingga perkiraan tersebut diketahui sebagai nilai optimal.

Salah satu contoh apabila kita berada dari suatu tempat di titik A akan menuju tempat yang berada di titik D di mana kita harus melewati minimal satu titik titik antara b,c, dan e.



Gambar 3.2 Contoh implementasi jalur pada node

Apabila kita memakai floyd warshall maka ada beberapa tahapan kerja dari algoritmanya yaitu :

1. Mencari node mana saja yang bisa dilalui untuk menuju ke node tujuan atau D.
2. Menjumlahkan nilai edge pada node dengan edge pada node yang akan dilalui mulai dari node awal menuju node tujuan.

   A-E-D = 30+15 km
   A-B-E-D = 10+15+15 km
   A-B-D = 10+15 km
   A-B-C-D = 10+15+15 km
   A-C-D = 25+15 km

3. Mencari nilai terkecil dari hasil penjumlahan edge pada node-node yang bisa dilalui. Dan dari hasil penjumlahan di atas didapat nilai terkecilnya yaitu pada jalur a-d-e dengan jumlah total edge 25Km

3.2.4 *Pseudocode Algoritma Floyd Warshall*

Pseudo-code algoritma *Floyd Warshall* adalah sebagai berikut:

```
function fw(int[1..n,1..n] graph) {
    // Inisialisasi
    var int[1..n,1..n] jarak := graph
    var int[1..n,1..n] sebelum
    for i from 1 to n
        for j from 1 to n
            if jarak[i,j] < Tak-hingga
                sebelum[i,j] := i
    // Perulangan utama pada algoritma
    for k from 1 to n
        for i from 1 to n
            for j from 1 to n
                if jarak[i,j] > jarak[i,k] + jarak[k,j]
                    jarak[i,j] = jarak[i,k] + jarak[k,j]
                    sebelum[i,j] = sebelum[k,j]
    return jarak
}
```

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.