# Indoor Routing menggunakan Ruang Tiga Dimensi(Studi Kasus : Telkom University, Bandung)

**Proposal Tugas Akhir** 

Kelas TA 1

Tiara Annisa Dionti 1103134405



Program Studi Sarjana Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2016

## Lembar Persetujuan

## Indoor Routing menggunakan Ruang Tiga Dimensi(Studi Kasus : Telkom University, Bandung)

Indoor Routing in Three Dimensional Spaces (Case Study: Telkom University Bandung)

Tiara Annisa Dionti NIM: 1103134405

Proposal ini diajukan sebagai usulan pembuatan tugas akhir pada
Program Studi Sarjana Informatika
Fakultas Informatika Universitas Telkom

Bandung, 1 April 2016 Menyetujui

Calon Pembimbing I

Calon Pembimbing II

Kiki Maulana A., S. Kom., M. T., PhD.

NIK: 06800352-1

Siti Sa'adah, M. T. NIK: 13861143-1

#### **Abstrak**

Data spatial merupakan data yang menyimpan tipe data geografis. Data ini sering digunakan pada sistem yang memanfaatkan data terkait wilayah suatu daerah, seperti pada sistem routing atau sistem navigasi. Sistem routing telah banyak diimplementasikan pada *outdoor routing*, dan seiring perkembangan zaman mulai dikembangkan ke arah indoor routing. Terdapat perbedaan signifikan yang menjadikan indoor routing lebih kompleks dibanding outdoor routing, yaitu pada outdoor routing hanya mengimplementasikan routing pada ruang satu dimensi, sedang pada indoor routing memungkinkan adanya routing pada ruang tiga dimensi yang merepresentasikan gedung yang bertingkat. Dalam kasus indoor routing ini, menggunakan metode Three Dimensional Spaces yang akan mengidentifikasi suatu objek secara akurat dengan menyimpan data spatial yang direpresentasikan ke bentuk undirected graph atau graph tidak berarah dengan atribut data tiga dimensi di mana x dan y merupakan koordinat suatu titik, dan z merepresentasikan level ketinggian titik tersebut. Indoor routing menggunakan shortest path algorithm dapat diimplementasikan setelah struktur Three Dimensional Spaces dibangun agar dapat memberikan keluaran rute terpendek yang dapat ditempuh antara dua titik lokasi. Tugas akhir ini bertujuan untuk mengimplementasikan sistem indoor routing menggunakan metode Three Dimensional Spaces pada data spatial Fakultas Informatika Universitas Telkom untuk dapat menentukan rute terpendek secara umum dan dengan data spatial yang terfilter berdasarkan atribut yang spesifik (ruang tertutup).

**Kata Kunci :** Spatial, *Indoor Routing*, *Three Dimensional Spaces*, graf, *shortest path algorithm* 

## Daftar Isi

Abstrak	C	i
Daftar Is	Isi	ii
BAB 1	Pendahuluan	1
1.1	Latar belakang	1
1.2	Perumusan Masalah	2
1.3	Tujuan	2
1.4	Hipotesa	2
1.5	Rencana Kegiatan	3
1.6	Jadwal Kegiatan	3
BAB 2	Landasan Teori	4
2.1	Basis Data Spatial	4
2.1	1.1 Representasi Data	4
2.2	Routing System	5
2.2	2.1 Outdoor Routing	5
2.2	2.2 Indoor Routing	7
2.3	Algoritma Menghitung Jarak Terpendek	9
2.3	3.1 Algoritma Djikstra	9
2.3	3.2 Floyd-Warshall	11
2.3	3.3 Bellman-Ford	12
2.3	3.4 Genetic Algorithm (GA)	13
BAB 3	Metodologi dan Desain Sistem	16
3.1	Gambaran Umum Sistem	16
3.1. Din	1.1 Pembangunan Struktur Data <i>Undirected Graph</i> dengan <i>Three</i> imensional Spaces	17
3.1	1.2 Pencarian Rute Terpendek	17
3.1		
3.2	Data Set	17
3.3	Spesifikasi Kebutuhan Sistem	18
BAB 4	Daftar Pustaka	

### BAB 1 Pendahuluan

## 1.1 Latar belakang

Dalam beberapa tahun belakangan ini, sistem navigasi atau sistem outdoor routing seperti yang dimiliki oleh Google Maps menjadi sistem yang sangat besar manfaatnya terutama pada orang-orang yang sering melakukan perjalanan tanpa mengetahui arah yang harus ditempuh untuk mencapai suatu tempat yang dituju. Seiring perkembangan zaman, sistem navigasi ini juga diimplementasikan pada wilayah yang lebih kecil, yaitu pada indoor spaces atau pemetaan ruangan di dalam sebuah gedung. Dengan adanya sistem pemetaan indoor spaces ini, akan mempermudah seseorang dalam menemukan lokasi yang ingin dituju. Pengaplikasian indoor routing ini cukup banyak berkembang di luar negeri untuk tempat-tempat tertentu seperti mall, bandara, perkantoran, dll. Namun di Indonesia sendiri mayoritas masih menggunakan sistem manual dengan menampilkan peta denah ruangan dalam gedung, dan belum banyak yang mengimplementasikan sistem indoor routing ini.

Terdapat perbedaan signifikan yang menjadikan indoor routing lebih kompleks dibanding *outdoor routing*, yaitu pada *outdoor routing* hanya mengimplementasikan routing pada ruang satu dimensi, sedang pada indoor routing memungkinkan adanya routing pada ruang tiga dimensi yang merepresentasikan gedung yang bertingkat. Hal ini menjadi tantangan dalam membangun sistem indoor routing, yaitu bagaimana cara merepresentasikan indoor spaces yang ada dalam sebuah gedung. Sebuah gedung mungkin memiliki sejumlah ruangan dan sejumlah koridor. Dari setiap ruangan memungkinkan untuk memiliki sejumlah pintu yang mehubungkan ruangan tersebut dengan *space* lain. Dan gedung tersebut bisa merupakan gedung bertingkat yang memiliki tangga, lift, atau pun elevator untuk dapat berpindah dari tingkat satu ke tingkat lainnya. Seluruh *space* tersebut harus dapat diidentifikasi label serta keterhubungan antar space tersebut. Metode spatial three dimentional spaces dapat menjadi solusi untuk membangun sistem indoor routing ini. Metode ini akan mengidentifikasi suatu objek secara akurat dengan menyimpan data geografis yang direpresentasikan ke bentuk undirected graph atau graph tidak berarah dengan atribut data tiga dimensi x, y, dan z, di mana x dan y merupakan koordinat suatu titik, dan z merepresentasikan level ketinggian titik tersebut.

Dalam tugas akhir ini, *indoor routing* dengan menggunakan representasi *three dimensional spaces* Query processing akan diimplementasikan untuk studi kasus gedung-gedung perkuliahan Fakultas Teknik Informatika Unversitas Telkom. Dengan adanya sistem ini, diharapkan mampu memberikan rute optimal yang dapat ditempuh untuk mencapai ruangan yang dituju.

#### 1.2 Perumusan Masalah

Adapun rumusan masalah dari latar belakang yang telah dipaparkan adalah:

- 1. Bagaimana representasi *three dimentional spaces* dapat diimplementasikan untuk membangun struktur data ruangan yang ada di wilayah Fakultas Teknik Informatika Universitas Telkom?
- 2. Bagaimana *indoor routing* dapat diimplementasikan pada data *three dimentional spaces*?
- 3. Bagaimana performansi dari sistem *indoor routing* menggunakan *three dimentional spaces*?

Adapun batasan masalah dalam tugas akhir ini adalah :

- 1. Representasi data graph yang digunakan dalam Tugas Akhir ini adalah *undirected graph* atau graph tidak berarah.
- 2. Rute yang terpilih adalah rute terpendek dengan dua pilihan, yaitu rute terpendek pada umunya dan rute dengan jalan yang tertutup (beratap) keseluruhan. Jika tidak ditemukan rute yang tertutup keseluruhan, maka yang dimunculkan hanya rute terpendek saja.
- 3. Data set yang digunakan adalah data lokasi ruangan Fakultas Teknik Informatika Universitas Telkom (sekitar Gedung F, E, D, A, dan B).

## 1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam tugas akhir ini adalah:

- 1. Mengetahui bagaimana representasi *three dimentional spaces* dapat diimplementasikan untuk membangun struktur data ruangan yang ada di wilayah Fakultas Teknik Informatika Universitas Telkom.
- 2. Mengetahui bagaimana *indoor routing* dapat diimplementasikan pada data *three dimentional spaces*.
- 3. Mengetahui bagaimana performansi dari sistem *indoor routing* menggunakan *three dimentional spaces*.

## 1.4 Hipotesa

Metode three dimentional spaces merupakan salah satu metode untuk merepresentasikan data spatial. Sistem yang akan dibangun ini merupakan sistem indoor routing dengan menggunakan struktur data graph sebagai data yang akan diproses. Sistem nantinya dapat menerima dua inputan, inputan pertama adalah lokasi asal dan inputan kedua adalah lokasi tujuan. Kemudian sistem akan mengoutputkan rute terpendek dari tempat asal ke tempat tujuan. Rute terpendek yang dioutputkan adalah rute terpendek pada umunya dan rute dengan jalan yang tertutup (beratap) keseluruhan. Jika tidak ditemukan rute yang tertutup keseluruhan, maka yang dimunculkan hanya rute terpendek saja.

## 1.5 Rencana Kegiatan

Adapun metode penyelesaian yang akan dilakukan untuk penyelesaian tugas akhir ini yaitu :

#### 1. Studi Literatur

Mempelajari sumber-sumber pustaka yang bisa dijadikan referensi mengenai *Indoor Routing* dengan *three dimentional spaces*.

#### 2. Pengumpulan Data dan Analisis

Penulis mengumpulkan dataset berbentuk text yang akan digunakan untuk mengimplementasikan metode *three dimentional spaces* untuk *Indoor Routing*. Pada tahap ini, akan dilakukan pelabelan pada setiap tempat yang termasuk dalam data set.

#### 3. Analisis dan Pembangunan Model

Tahap ini meliputi analisis kebutuhan, analisis pembangunan sistem *Indoor Routing* serta memodelkan sistem dengan model aritmatik.

#### 4. Pembagunan Sistem

Tahap ini meliputi pembangunan perangkat lunak sesuai dengan rancangan pada tahap sebelumnya.

## 5. Pengujian Sistem

Melakukan pengujian terhadap sistem dari segi akurasi dan performansi sistem.

#### 6. Analisis dan Kesimpulan

Dilakukan analisis terhadap hasil akurasi pembangunan struktur *three dimensional spaces*.

#### 1.6 Jadwal Kegiatan

Berikut rencana jadwal kegiatan yang akan dilakukan:

Kegiatan			Ві	ulan ke-		
	1	2	3	4	5	6
Studi						
Literatur						
Pengumpulan						
Data						
Pembangunan						
Model						
Pembangunan						
Sistem						
Analisis Hasil						
Pembuatan	_					
Laporan						

## BAB 2 Landasan Teori

## 2.1 Basis Data Spatial

Basis data spatial adalah basis data yang didesain untuk menyimpan dan memproses data objek dalam sebuah lokasi dengan tipe data spatial seperti titik (point), garis (line), atau pun area (region) yang terdapat dalam *Geographic Information System* (GIS). Data spasial dapat dimanfaatkan pada penampakan 2 dimensi seperti penampakan permukaan bumi, ataupun penampakan 3 dimensi seperti pemodelan otak manusia, rantai molekul protein, dll. Setelah adanya sisten basis data relasional, berkembanglah sistem basis data lainnya, salah satunya adalah basis data spasial. Karakteristik basis data spasial yang dibutuhkan adalah sistem yang mampu menampung data objek geometris sederhana dengan jumlah yang besar untuk menampung seperti 100.000 polygon.

Sistem basis data spasial mulai popular beberapa tahun belakangan ini, terutama pada konferensi "Symposium on Large Spatial Databases (SSD)" yang diadakan dua tahun sekali sejak 1898 terkait dengan basis data yang menyimpan objek pada ruang sebagai pendukung gambar ruang. [10]

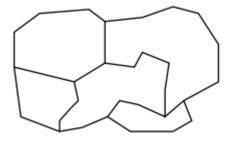
## 2.1.1 Representasi Data

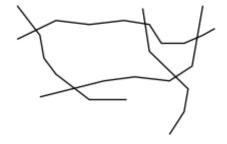
Seperti yang telah disebutkan sebelumnya, data spatial dapat direpresentasikan dengan titik (point), garis (line), atau pun area (region). Misalnya, sebuah kota dapat dimodelkan sebagai titik dalam model yang menggambarkan area geografis yang luas. Sebuah garis biasanya digunakan untuk merepresentasikan fasilitas untuk bergerak atau koneksi dalam ruang seperti jalan, sungai, kabel telepon, listrik, dll. Sebuah wilayah atau *region* merepresentasikan objek yang memiliki batas di ruang dua dimensi, misalnya Negara, danau, dll. Sebuah daerah juga muungkin saja memiliki daerah-daerah potongan. Gambar 2.1 merupakan gambar representasi dasar pada basis data spasial, yaitu titik, garis, dan area.



Gambar 2.1 Tiga representasi dasar spatial: titik, garis, area

Objek-objek spatial yang saling berelasi dapat digambarkan dengan bentuk partisi (*partitions*) atau pun jaringan (*networks*) seperti yang terlihat pada Gambar 2.2 berikut.





Gambar 2.2 Partisi dan jaringan

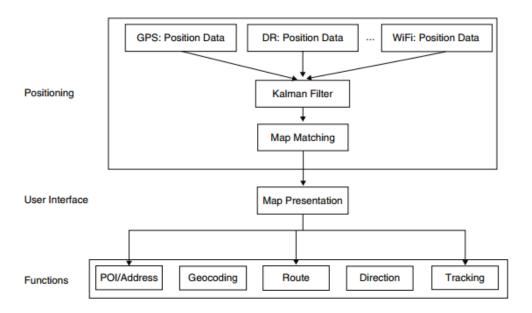
Sebuah partisi dapat digambarkan dengan sekelompok *region* yang terpisah. Yang menjadi penghubung antar *region* tersebut adalah sisi *region* yang sama-sama menjadi batas atau sisi dari region lainnya. Sebagai contoh, partisi dapat digunakan untuk merepresentasikan peta tematik. Sebuah jaringan dapat digambarkan dengan graph yang terhubung dalam sebuah bidang dimana setiap objek titik dianggap sebagai *node*, dan garis-garis sebagai bentuk geometri dari *edges*. Bentuk jaringan ini dapat digunakan untuk merepresentasikan jalan raya, sungai, jalur transportasi umum, atau pun jalur kabel listrik. [6]

## 2.2 Routing System

Routing System merupakan sebuah sistem yang berfungsi untuk memberikan petunjuk jalan atau rute yang dapat ditempuh oleh suatu objek bergerak untuk dapat mencapai suatu lokasi dari tujuan. Sistem ini sering disebut dengan sistem navigasi. Sistem navigasi modern saat ini telah mengintegrasikan antara posisi objek, sensor, komputasi, serta komunikasi antara hardware dan software untuk menunjang fasilitas pada manusia, kendaraan, dan juga objek bergerak lainnya. Selain itu, sistem navigasi modern juga memperhatikan akurasi jarak koordinat geografis lokasi, kecepatan, dan ketinggian objek bergerak. Sistem navigasi ini banyak digunakan pada data outdoor. Namun seiring perkembangan zaman, sistem ini diterapkan pula pada data indoor.

## 2.2.1 Outdoor Routing

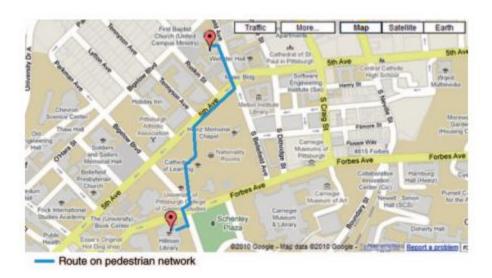
Teknologi *outdoor routing* saat ini sudah sangat berkembang. Gambar 2.3 berikut menjelaskan alur informasi dari sistem *outdoor navigation*.



Gambar 2.3 Alur informasi pada sistem outdoor navigation

Dari gambar di atas, posisi user ditentukan oleh : (a) memperoleh data posisi melalui sensor *geo-positioning* dan (b) menerapkan algoritma *map matching* menggunakan data posisi yang diperoleh. Langkah-langkah ini merupakan langkah umum untuk meningkatkan akurasi, ketersediaan, dan keandalan sistem *outdoor navigation* dengan menggunakan lebih dari satu sensor *geo-positioning* di mana setiap data posisi dapat difilter menggunakan Kalman Filter untuk menemukan estimasi posisi terbaik. Selanjutnya data posisi yang telah terfilter tersebut akan menjadi inputan pada algoritma *map matching* yang menggunakan basis data map untuk *travelling area*, yang mengandung data spatial dan non-spatial untuk menemukan : (a) ruas jalan / trotoar di mana pengguna berada dan (b) lokasi yang tepat dari pengguna di segmen.

Saat lokasi pengguna telah diketahui, lokasi tersebut akan ditandai dalam peta dan ditampilkan kepada user. Pada tahap ini, sistem sedang melakukan *tracking*, dan pengguna memiliki pilihan untuk mencari POIs atau *request* untuk rute optimal antara pasangan alamat. Sistem menggunakan pencarian menggunakan kriteria rute terpendek atau tercepat. [8] Gambar 2.4 merupakan contoh implementasi sistem *outdoor routing*.



Gambar 2.4 Contoh implementasi outdoor routing

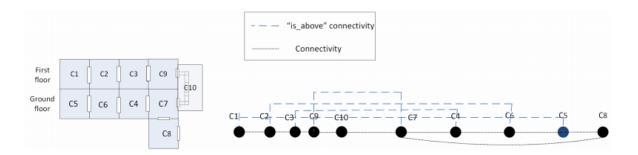
## 2.2.2 Indoor Routing

Indoor routing merupakan implementasi routing system atau sistem navigasi di dalam ruang atau gedung. Dengan menggunakan data spatial, setiap ruang dapat diidentifikasi secara akurat. Terdapat berbagi macam elemen pada indoor spaces seperti kamar, pintu, koridor, lantai, tangga, lift, dan jalur yang menghubungkan antar gedung. Indoor spaces direpresentasikan dengan undirected graph dengan nodes dan edges. Konsep pemodelan indoor spaces dapat dirangkum dalam Tabel 2.1.

domain concept	modeling concepts
room	a cell
door	an edge
corridor	one or more cells with one or more edges
stair	one or more cells with one or more edges
elevator	one cell with several edges
pathway	one or more cells with several edges

Tabel 2.1 Konsep pemodelan indoor spaces

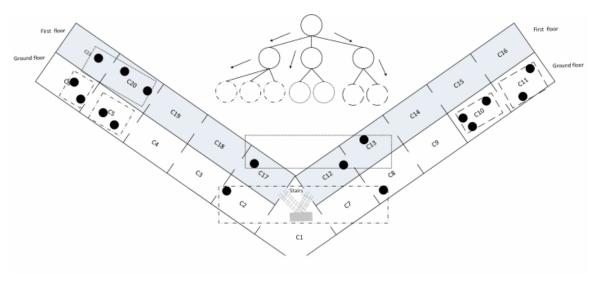
Sebagai ilustrasi, Gambar 2.5 merupakan pemodelan graf pada *indoor spaces* yang digambarkan secara satu level.

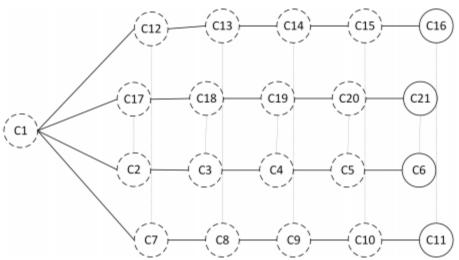


Gambar 2.5 Pemodelan indoor spaces dengan graph satu level

Dari ilustrasi tersebut, dapat dilihat bahwa C10, C9, C8, C7, C4, C6, C5, C3, C2 dan C1 terhubung di mana C10 merupakan tangga yang menghubungkan lantai dasar dan lantai satu. Dengan adanya konsep ini, dapat disimpulkan bahwa :

- (i) *Indoor spaces* adalah sebuah *undirected graph* yang terhubung (nodes, edges) di mana sel = {C1, C2, C3, ..., Cn} merupakan sebuah set sel, dan *edges* = {Ei, E2, E3, ..., En} merupakan sebuah set *edge* yang masing-masing merupakan penghubung dari dua sel yang berbeda.
- (ii) (Nodes bertetanggaan) missal, G = (N, C) merupakan graph. Dan dua node n1, n2  $\in$  N dari G adalah bertetangga jika :  $\exists$  n1,n2 terhubung.
- (iii) (Graf dengan konektivitas multi dimensi atau *Multidimensional Connectivity Graph*) misal, ada satu set sel Ci, Cj, Cf, Cl di lantai dasar, dan satu set sel Cx, Cy,Cr, Ct maka *multidimensional connectivity graph* menunjukkan beberpa *edges* yang menghubungkan setiap ruang pada tiap lantai. Dapat diilustrasikan melalui Gambar 2.6.





Gambar 2.6 Multidimensional Connectivity Tree based pada indoor spaces[2]

## 2.3 Algoritma Menghitung Jarak Terpendek

Terdapat beberapa algoritma yang telah teruji kaurasi dan kompleksitas waktunya untuk menghitung jarak terpendek pada data graph, diantaranya adalah Algoritma Djikstra, Floyd-Warshall, Bellman-Ford, dan Genetic Algorithm (GA). Berikut keterangan dari masing-masig algoritma.

## 2.3.1 Algoritma Djikstra

Untuk setiap simpul dalam graf, label telah ditetapkan yang menentukan panjang minimal dari titik awal s ke simpul lainnya v dari graf. Dalam komputasi kita bisa melakukannya dengan menyatakan array d[]. Algoritma ini bekerja secara berurutan, dan setiap langkah itu dilakukan untuk menurunkan nilai label simpul.

Algoritma berhenti ketika semua simpul telah dikunjungi. Nilai label di titk awal s adalah sama dengan nol (d [s] = 0). Namun, label di simpul lainnya v sama dengan infinity (d  $[v] = \infty$ ), yang berarti bahwa panjang dari titik awal s ke simpul lainnya tidak diketahui. Dalam komputasi kita hanya bisa menggunakan jumlah yang sangat besar untuk mewakili infinity. Selain itu, untuk setiap vertex v kita harus mengidentifikasi apakah sudah dikunjungi atau belum. Untuk itu, kita mendeklarasikan array dari jenis Boolean yang disebut u [v] yang diinisialisasikan sebagai belum dikunjungi (u [v] = false) untuk semua simpul. Algoritma Dijkstra terdiri dari n iterasi. Jika semua simpul yang telah dikunjungi, maka algoritma selesai. Sebaliknya, dari daftar simpul yang belum dikunjungi kita harus memilih titik yang memiliki nilai minimum (terkecil) pada labelnya (awalnya, akan dipilih titik awal s). Setelah itu, akan dipertimbangkan semua tetangga dari simpul ini (Tetangga dari vertex adalah simpul-simpul yang memiliki tepi yang sama dengan vertex awal). Untuk setiap tetangga yang belum dikunjungi kita akan mempertimbangkan panjang label yang baru, yang sama dengan jumlah dari nilai label pada awal vertex v (d [v]) dan panjang tepi l yang menghubungkan mereka. Jika nilai yang dihasilkan kurang dari nilai pada label, maka nilai dalam label harus diubah dengan nilai yang baru diperoleh.

$$d [ neighbors ] = min ( d [ neighbors ], d[ v ] + 1 )$$
 (1)

Setelah mempertimbangkan semua tetangga, selanjutnya vertex awal akan ditetapkan sebagai telah dikunjungi (u [v] = true). Setelah mengulangi langkah ini sebanyak n kali, semua simpul dari graph akan telah dikunjungi dan algoritma selesai atau berakhir. Simpul yang tidak terhubung dengan titik awal akan tetap di*assign* sebagai infinity. Untuk mengembalikan jalur terpendek dari titik awal ke simpul lainnya, kita perlu mengidentifikasi berbagai p [], di mana untuk setiap vertex, di mana  $v \neq s$ , akan disimpan jumlah titik p [v], yang merupakan simpul kedua dari belakang di jalur terpendek. Dengan kata lain, jalur dari s ke v dapat didefinisikan dengan persamaan berikut.

$$P = (s, ..., p[p[p[v]]), p[p[v], v)$$
 (2)

Gambar 2.7 merupakan implementasi Algoritma Djikstra menggunakan bahasa pemrograman Java.

```
for (int i = 0; i < n; i++) {
    // vertex
    int v = -1;
    // finding minimum label among unvisited vertices
    for (int j = 0; j < n; j++) {
        if (u[j] == false && (v == -1 || d[j] < d[v])) {
            v = j;
        }
    }
    // set as visited.
    u[v] = true;

for (int j = 0; j < n; j++) {
        // if there is exists path between v and j vertices
        if (a[v][j] > 0) {
            if (d[v] + a[v][j] < d[j]) {
                 d[j] = d[v] + a[v][j];
            }
        }
    }
}</pre>
```

Gambar 2.7 Kutipan implementasi Algoritma Djikstra menggunakan Bahasa Pemrograman Java

## 2.3.2 Floyd-Warshall

Perhatikan grafik G, di mana simpul diberi nomor dari 1 sampai n. Notasi  $d_{ijk}$  berarti jalur terpendek dari i ke j, yang juga melewati vertex k. Tepi antara simpul i dan j itu akan sama dengan  $d_{ij0}$ , jika tidak ada maka di*assign* sebagai infinity. Namun, untuk nilai-nilai lain dari  $d_{ijk}$  bisa menjadi dua pilihan: (1) Jika jalur terpendek dari i ke j tidak melewati titik k maka nilai  $d_{ijk}$  akan sama dengan  $d_{ijk-1}$ . (2) Jika jalur terpendek dari i ke j melewati vertex k maka pertama ia pergi dari saya untuk k, setelah itu pergi dari k ke j. Didalam kasus nilai dijk akan sama dengan  $d_{ikk-1} + d_{kjk-1}$ . Untuk menentukan jalur terpendek, hanya perlu menemukan minimal dua pernyataan ini :

$$d_{ij0} = panjang tepi antara simpul$$
 (3)

$$d_{ijk} = \min (d_{ijk-1}, d_{ikk-1} + d_{kjk-1})$$
(4)

Gambar 2.8 merupakan implementasi Algoritma Floyd-Warshall menggunakan bahasa pemrograman Java.

Gambar 2.8 Kutipan implementasi Algoritma Floyd-Warshall menggunakan Bahasa Pemrograman Java

#### 2.3.3 Bellman-Ford

Berbeda dengan Algoritma Dijkstra, Algoritma Bellman-Ford memperhatikan tepi dengan bobot negatif. Itu sebabnya, graf dapat berisi siklus bobot negatif, yang akan menghasilkan banyak jumlah jalur dari titik awal ke tujuan akhir, di mana setiap siklus akan meminimalkan panjang jalur terpendek. Dengan adanya fakta ini, diasumsikan bahwa graf tidak mengandung siklus dengan bobot negatif. Array d[] akan menyimpan panjang minimal dari titik awal s ke simpul lainnya. Algoritma ini terdiri dari beberapa fase, di mana di setiap fase perlu untuk meminimalkan nilai semua sisi dengan mengganti d[b] agar dapat mengikuti pernyataan d[a] + c. a dan b adalah simpul dari graf, dan c adalah tepi yang menghubungkan mereka. Dan untuk menghitung panjang dari semua jalur terpendek dalam graf, dibutuhkan n - 1 fase, tetapi bagi simpul-simpul dari graf yang terjangkau, nilai elemen array akan tetap sama dengan infinity.

Gambar 2.9 merupakan implementasi Algoritma Bellman-Ford menggunakan bahasa pemrograman Java.

Gambar 2.9 Kutipan implementasi Algoritma Bellman-Ford menggunakan Bahasa Pemrograman Java

## 2.3.4 Genetic Algorithm (GA)

Algoritma *Intelliget* telah diperkenalkan dalam menemukan jalur terpendek optimal dalam banyak situasi yang memerlukan sistem untuk mencari melalui ruang pencarian yang sangat besar dalam waktu yang terbatas dan juga dalam mengatasi lingkungan yang selalu berubah. Salah satu algoritma ini adalah GA. Definisi Algoritma Genetika adalah kelas atau kelompok stochastic search algorithms yang didasarkan pada evolusi biologis. GA sebagian besar digunakan untuk masalah optimasi. Menggunakan beberapa operasi genetik seperti seleksi, crossover, dan mutasi dalam rangka untuk menghasilkan generasi baru dari populasi, yang mewakili seperangkat solusi (kromosom) untuk masalah saat ini. Selain itu, rata-rata generasi baru ini seharusnya baik dari segi nilai fitness mereka secara keseluruhan dibandingkan dengan populasi sebelumnya. Setiap individu atau kromosom dalam populasi akan diberi nilai fitness yang dihitung berdasarkan fungsi fitness yang telah ditentukan yang mengukur seberapa optimal solusi dalam memecahkan masalah yang ada. Untuk mengatasi kasus jalan terpendek menggunakan GA, kita perlu untuk menghasilkan sejumlah solusi, dan kemudian memilih satu yang paling optimal antara set solusi yang memungkinkan. Untuk mengatasi masalah tersebut, sebuah populasi awal yang membentuk set pertama kromosom yang akan digunakan di GA akan dibangkitkan secara acak. Setiap kromosom mewakili salah satu solusi untuk masalah ini. Setelah itu, kromosom-koromosom tersebut akan diestimasi berdasarkan funsi fitnes tertentu, yang menentukan seberapa baik solusi tersebut. Setelah memperrhitungkan nilai fitness dari setiap solusi atau kromosom, beberapa kromosom atau individu akan dipilih, dan operasi genetik dasar seperti crossover dan mutasi diterapkan pada kromosom tersebut. Kemudian, nilai fitness setiap kromosom dihitung ulang, dan solusi terbaik yang dipilih akan menjadi pertimbangan untuk generasi selanjutnya. Proses ini berlanjut sampai masalah yang diberikan tidak akan tercapai. Dengan demikian, dapat diidentifikasi tahapan GA adalah:

- Langkah 1: Tentukan fungsi fitness, dalam kasus ini perlu dimaksimalkan fungsi berikut f ( $Ch_k$ ) = ( $\Sigma$ edge) 1, di mana  $Ch_k$  adalah k-th kromosom dan  $\Sigma$ edge adalah hasil penjumlahan tepi dari titik awal ke tujuan akhir.
- Langkah 2: Buat populasi awal populasi yang mengandung n individu. Pada tahap ini kita tidak perlu membuat fittest individu, karena besar kemungkinan GA akan mengubah individu ke populasi yang layak. Dalam rangka menciptakan kromosom untuk populasi awal, kita akan menghasilkan jalur acak dari titik awal ke tujuan akhir.
- Langkah 3: Seleksi tahap GA yang digunakan untuk memilih dua kromosom untuk operasi genetik seperti crossover dan mutasi. Ada berbagai jenis metode seleksi. Namun, Roulette Wheel merupakan metode seleksi yang dipilih untuk memecahkan masalah jalan terpendek.
- Langkah 4: Crossover merupakan proses reproduksi di mana keturunan mewarisi beberapa ciri-ciri dari kedua orang tua mereka. Individu yang melakukan reproduksi akan dipilih dari seluruh populasi (bukan dari survivors di iterasi pertama, karena kita perlu menjaga keanekaragaman individu), dengan kata lain, seluruh penduduk akan dicocokkan dengan salinan tunggal satu individu. Ada jenis metode Crossover lain, namun untuk masalah ini akan digunakan metode yang paling sederhana, yang disebut single point crossover.
- Langkah 5: Mutasi proses yang mengubah nilai dari beberapa gen. Mutasi menjaga keragaman genetik dari populasi dengan mengubah gen kromosom yang dipilih.

Jika *fittest* kromosom tidak berubah setelah jumlah iterasi tertentu, yang dijelaskan di atas, maka algoritma akan berakhir. Solusi yang paling optimal otomatis adalah *fittest* kromosom di antara seluruh populasi. Diagram ini adalah sebuah kerangka kerja yang akan digunakan untuk pelaksanaan GA dalam menemukan jalur terpendek atau rute di peta tertentu sebuah kota bernama Almaty di Kazakhstan. GA yang digunakan di sini cukup umum kecuali untuk bagian di mana loop mungkin diperkenalkan saat mengeksekusi GA dalam menemukan solusi yang paling optimal, semua loop harus dihapus karena loop tidak harus ada dalam jalur.

Dari penelitian sebelumnya, didapatkan perbedaan nilai kompleksitas waktu antara *shortest path algorithms* tersebut yang tercantum dalam Tabel 2.2 berikut.

Algorithm	Time complexity
Dijkstra	n² + m
Bellman-Ford	n <sup>3</sup>
Floyd-Warshall	nm

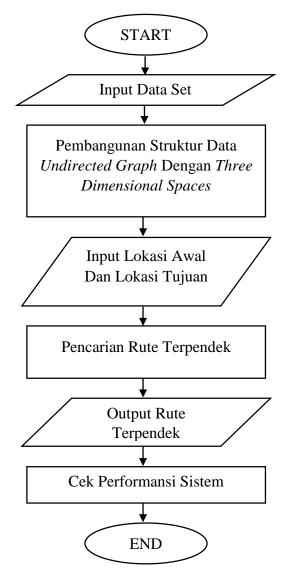
Tabel 2.2 Kompleksitas waktu Shortest Path Algorithms

Algoritma Djikstra, Bellman-Ford, dan Floyd-Warshall memberikan satu keluaran rute terpendek. Berbeda dengan Algoritma GA yang akan memberi beberapa rekomendasi rute terpendek.[9]

## BAB 3 Metodologi dan Desain Sistem

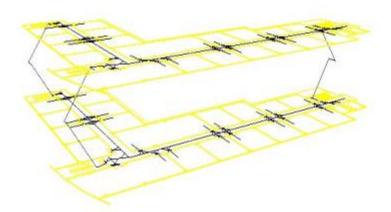
## 3.1 Gambaran Umum Sistem

Gambar 3.1 menggambarkan proses secara umum sistem yang akan dibuat.



## 3.1.1 Pembangunan Struktur Data Undirected Graph dengan Three Dimensional Spaces

Pada tahap ini akan dibangun struktur data graf tidak berarah dengan menggunakan representasi ruang tiga dimensi dari data set yang ada. Berbeda dengan representasi data *indoor routing* yang sudah dijelaskan di BAB 2, tahap ini akan merepresentasikan gedung bertingkat secara tiga dimensi berdasarkan letak koordinat x, y, dan koordinat ketinggian z. Ilustrasi *Three Dimensional Spaces* dapat dilihat pada Gambar 3.1. [7]



Gambar 3.1 Representasi Three Dimensional Spaces pada indoor spaces

## 3.1.2 Pencarian Rute Terpendek

Pada tahap ini akan diimplementasikan algoritma pencarian rute terpendek pada graf tidak berarah, di mana pencarian dilakukan secara umum dan dengan mengimplementasikan teknik filtrasi ruang tertutup.

#### 3.1.3 Cek Performansi Sistem

Performansi yang akan diukur pada sistem addalah performansi terkait dengan akurasi dan kecepatan waktu eksekusi.

#### 3.2 Data Set

Data set yang digunakan adalah data tabel yang merupakan data geografis lokasi gedung serta setiap ruangan dalam gedung Fakultas Teknik Informatika Universitas Telkom. Properti yang akan ada dalam data set antara lain adalah *space label*, koordinat x, koordinat y, koordinat z, keterhubungan antar *space*, serta property-properti penunjang lainnya.

## 3.3 Spesifikasi Kebutuhan Sistem

Dalam pengerjaan tugas akhir ini, dibutuhkan perangkat keras dan perangkat lunak yang menunjang sistem *indoor routing* dapat dibangun dan bekerja dengan semestinya. Berikut adalah spesifikasi kebutuhan sistem :

a. Spesifikasi Perangkat Keras

Berikut spesifikasi perangkat keras yang digunakan dalam pengerjaan tugas akhir ini.

Merk : Lenovo

Model : Intel® Core<sup>TM</sup> i7-3770 CPU

Prosesor : 3.40 GHz Intel Core i7

RAM : 4 GB

BIOS : Lenovo BIOS

OS : Windows 7 Professional 64 bit

## b. Spesifikasi Perangkat Lunak

Berikut spesifikasi perangkat lunak yang digunakan dalam pengerjaan tugas akhir ini.

Aplikasi :

Microsoft Word 2013, Microsoft Excel 2013, Latex, Notepad++, Java Netbeans IDE dengan JDK 1.8, MySQL, XAMPP

## **BAB 4** Daftar Pustaka

- [1] Afyouni, I., Cyril, R., & Christophe, C. (2012). Spatial models for context-aware indoor navigation systems: A survey. *Journal of Spatial Information Science*, *1*(4), 85-123.
- [2] Alamri, S. M. (2014). Adjacency-based indexing for moving objects in spatial databases (Doctoral dissertation, Monash University. Faculty of Information Technology. Clayton School of Information Technology).
- [3] Bian, W., Guo, Y., & Qiu, Q. (2014, November). Research on Personalized Indoor Routing Algorithm. In *Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2014 13th International Symposium on* (pp. 275-277). IEEE.
- [4] Gotlib, D., Gnat, M., & Marciniak, J. (2012, November). The research on cartographical indoor presentation and indoor route modeling for navigation applications. In *Indoor Positioning and Indoor Navigation (IPIN)*, 2012 International Conference on (pp. 1-7). IEEE.
- [5] Gubichev, A., Bedathur, S., Seufert, S., & Weikum, G. (2010, October). Fast and accurate estimation of shortest paths in large graphs. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 499-508). ACM.
- [6] Güting, R. H. (1994). An introduction to spatial database systems. *The VLDB Journal—The International Journal on Very Large Data Bases*, *3*(4), 357-399.
- [7] Han, L., Zhang, T., & Wang, Z. (2014). The design and development of indoor 3D routing system. *Journal of Software*, 9(5), 1223-1228.
- [8] Karimi, H. A. (2011). *Universal navigation on smartphones*. Springer Science & Business Media.
- [9] Magzhan, K., & Jani, H. M. (2013). A review and evaluations of shortest path algorithms. *International journal of scientific & technology research*, 2(6).
- [10] Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (2000). *Spatial tessellations: concepts and applications of Voronoi diagrams* (Vol. 501). John Wiley & Sons.