

# Sistem Penentuan Rute Yang Tepat Dalam Sebuah Labirin Dengan Menerapkan Algoritma Prim

Sayed Fachrurrazi

Sistem Informasi Fakultas Teknik Universitas Malikussaleh  
Jl. Cot Tgk Nie-Reulet, Aceh Utara, 141 Indonesia  
email :sayedfachrurrazi@gmail.com

## ABSTRAK

Matriks adalah tabel atau array berdimensi 2 sehingga mempermudah pemrograman dalam mendesain programnya. Labirin adalah sebuah permainan yang direpresentasikan sebagai tempat yang terdiri dari kumpulan jalan yang rumit, bercabang, dan buntu yang dirancang membentuk sebuah teka-teki dimana pejalan harus mencari jalan keluar dari tempat tersebut. Semakin besar ukuran suatu labirin maka akan meningkatkan kerumitan dalam mencari jalan keluarnya. Tujuan dari penelitian ini adalah untuk membangun sebuah sistem pengambilan keputusan yang membantu menentukan rute terpendek dalam sebuah labirin dengan menggunakan metode algoritma prim.

**Kata kunci :** Matriks, Labirin, Algoritma Prim, Rute Terpendek

## PENDAHULUAN

### 1.1 Latar Belakang

Banyak permasalahan yang dapat dimodelkan dengan menggunakan prim, salah satunya adalah masalah dalam pencarian pohon merentang minimum (*minimum spanning tree*). Prim mempunyai peruntukan yang cukup luas dalam kehidupan nyata, diantaranya adalah penggunaan prim dalam melakukan Perutean yaitu kegiatan membuat rute atau jalur dengan tujuan tertentu.

Dengan penggunaannya akan didapatkan lintasan dengan keunggulan-keunggulan tertentu misalnya lintasan dengan biaya paling murah, lintasan dengan waktu tempuh paling cepat, lintasan dengan jarak paling pendek, dan lintasan dengan tingkat efisiensi paling tinggi. Pada bidang yang lebih luas, jika dianalogikan lintasan yang dibuat dengan alur kerja yang harus dilakukan, maka akan didapatkan efisiensi kerja dan hasil yang optimal.

Labirin adalah sebuah jaringan dari jalur jalur yang saling berhubungan untuk dilalui dari awal hingga akhir yang dimaksudkan untuk sebuah tantangan, manusia mungkin masih dapat menyelesaikan masalah pencarian ruang terdekat yang sederhana, tetapi jika jumlah rute yang ada sudah sedemikian banyaknya, maka kita akan mengalami kesulitan dan akan memakan waktu yang lama untuk menyelesaikannya.

Ada beberapa cara yang lazim digunakan dalam memecahkan masalah perutean salah satunya algoritma prim. Algoritma prim merupakan cara yang paling umum digunakan untuk membentuk pohon merentang minimum algoritma ini terbukti mampu menghasilkan pohon merentang minimum.

Pada perancangan ini penulis menggunakan algoritma prim dikarenakan algoritma ini mampu menyelesaikan masalah terhadap penentuan rute yang dapat dilalui dalam sebuah tempat. Dikarenakan dalam algoritma tersebut disaat 1 titik didapatkan dia tidak akan mengulangi ke belakang malah dia akan mencari titik terdekat selanjutnya sehingga proses kalkulasi dan observasinya menjadi cepat tepat dan efisien. Oleh karena itulah penulis merasa cocok untuk menerapkan algoritma prim dalam penentuan rute yang tepat dalam labirin.

Pada prinsipnya penelitian bertujuan untuk menerapkan algoritma prim dalam pemecahan rute kompleks dalam sebuah Labirin dan untuk memberikan informasi jalur yang tepat yang ada pada labirin.

Penelitian ini diharapkan dapat Memberikan manfaat untuk dapat Membantu dalam mencari rute yang tepat untuk dilalui. Dengan adanya sistem penentuan rute ini dapat mempermudah dalam mencari rute mana yang tepat dalam labirin. Dapat mengetahui kemungkinan rute yang salah dilalui dan jalan menuju titik akhir cuma 1 yang lainnya akan tersesat Sesuai dengan ilustrasi yang diuraikan pada pendahuluan, maka rumusan masalah Bagaimana membuat sebuah sistem dapat memecahkan rute dengan tepat. Bagaimana mengimplementasi sistem dalam proses penetapan rute dalam sebuah labirin.

## **2. TINJAUAN PUSTAKA**

### **2.1 MATRIK**

Matriks adalah susunan kumpulan bilangan yang diatur dalam baris dan kolom berbentuk persegi panjang. Matriks dicitikan dengan elemen-elemen penyusun yang diapit oleh tanda kurung siku [ ] atau tanda kurung biasa ( ).

Menurut ruminta (2012) Matriks adalah susunan bilangan (elemen) yang disusun menurut baris dan kolom sehingga berbentuk persegi panjang. Sering dikatakan bahwa matriks adalah tabel atau array berdimensi 2. Pengertian 2 dimensi ini hanya untuk mempermudah pemrograman dalam mendesain programnya.

Matriks merupakan kumpulan elemen yang disusun menurut baris dan kolom sehingga berbentuk empat persegi panjang, yang panjang dan lebarnya ditunjukkan oleh banyaknya kolom dan baris.

Matriks dinotasikan dengan huruf kapital A, B, K dan sebagainya. banyaknya baris dan banyaknya kolom suatu matriks menentukan ukuran dari matriks tersebut, yang disebut ordo matriks Secara umum,

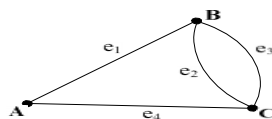
Matriks A  $m \times n$ , Perhatikan bahwa elemen matriks A tersebut berindeks rangkap, misalnya  $a_{23}$  menyatakan elemen matriks A pada baris ke-2 dan kolom ke-3 sedangkan matriks A berordo  $m \times n$  dan ditullis  $A_{m \times n}$

## 2.2 GRAF

Sejak pertama kali diperkenalkan oleh seorang matematikawan Swiss bernama Leonhard Euler pada tahun 1736 (Nur Insani , 2012), teori graf mulai berkembang pesat bahkan hingga saat ini. Teori graf merupakan salah satu cabang dari ilmu matematika diskrit yang meskipun usianya sudah cukup tua namun masih banyak mendapat perhatian terutama oleh para ilmuan.

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut (Nur Insani, 2012). Representasi visual dari graf adalah dengan menyatakan objek sebagai noktah, bulatan atau titik, sedangkan hubungan antara objek dinyatakan dengan garis.

Graf adalah suatu struktur diskrit yang terdiri dari verteks-verteks dan busur-busur yang menghubungkan verteks-verteks tersebut (Nur Insani, 2012). Suatu graf G terdiri dari dua himpunan, yaitu himpunan V dan himpunan E. Himpunan V adalah himpunan simpul (verteks) yang terbatas dan tidak kosong sedangkan himpunan E adalah himpunan busur (*edge*) yang menghubungkan sepasang simpul yang terdapat dalam himpunan V. Oleh karena itu, graf G dinotasikan sebagai  $G(V,E)$  yang artinya graf G mempunyai V simpul dan E busur. Dapat dikatakan graf adalah kumpulan dari simpul-simpul yang dihubungkan oleh sisi-sisi



**Gambar 2.1 Graf**

Pada gambar graf G diatas, graf terdiri dari himpunan V dan E yaitu:

$$V = (A, B, C) \dots\dots\dots (1)$$

$$E = (e1,e2,e3,e4);$$

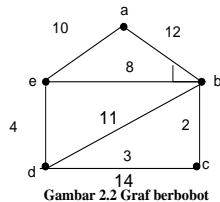
$$\text{Bisa ditulis}\{(A,B),(B,C),(B,C),(A,C)\} \dots\dots (2)$$

Graf G disebut graf terhubung jika untuk setiap pasang simpul  $u$  dan  $v$  di dalam himpunan  $V$  terdapat lintasan dari  $u$  ke  $v$ . Jika tidak, maka graf G disebut graf tak terhubung (*disconnected graph*) (Lucia Ratna Sari, 2012). Keterhubungan dua buah simpul adalah penting di dalam graf. Jika dua buah simpul terhubung maka pasti simpul yang pertama dapat dicapai dari simpul yang kedua.

Misalkan  $u$  dan  $v$  adalah titik yang berbeda pada graf  $G$ . Maka titik  $u$  dan  $v$  dapat dikatakan terhubung (*connected*), jika terdapat lintasan  $u-v$  di  $G$ . Sedangkan suatu graf  $G$  dapat dikatakan terhubung (*connected*), jika untuk setiap titik  $u$  dan  $v$  di  $G$  terhubung. Keterhubungan adalah sifat yang dimiliki graf. Graf terhubung dapat dilihat atau dibuktikan dari keterhubungan antara  $u$  dan  $v$ . Untuk lebih menguatkan kondisi ( $u, v$ ) (Dr. Ibrahim, M.Pd, 2013)

**A. Graf Berbobot (Weighted Graph)**

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga. Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf (Nur Insani, 2012). Bobot dapat menyatakan jarak antara dua buah tiang listrik, kapasitas, biaya perjalanan antara dua buah kota, waktu tempuh pesan (*message*) dari sebuah simpul komunikasi ke simpul komunikasi lain, ongkos produksi, dan sebagainya.



Gambar 2.2 Graf berbobot

**2.3 ALGORITMA**

Algoritma adalah deskripsi langkah-langkah penyelesaian masalah yang tersusun secara logis atau urutan logis pengambilan keputusan untuk pemecahan suatu masalah. Algoritma dengan notasi khusus, notasi mudah dimengerti dan ditulis notasi dapat diterjemahkan menjadi sintaks suatu bahasa pemrograman (Abdul Kadir, 2012). Pengertian Algoritma lainnya menurut Moh Sjukani, dalam bukunya "Dari Algoritma dan Struktur Data dengan C, C++, dan Java " adalah sebuah alur pemikiran dalam memecahkan suatu pekerjaan yang dituangkan secara tertulis.

Algoritma dapat berupa kalimat, gambar, atau table tertentu. Algoritma merupakan sebuah pemikiran, artinya dimungkinkan adanya algoritma yang berbeda dari setiap orang. (M.Shalahuddin Rosa A.S., 2010)

Algoritma merupakan salah satu cabang ilmu komputer yang membahas prosedur penyelesaian suatu permasalahan. Dengan algoritma yang baik maka komputer bisa menyelesaikan perhitungan dengan cepat dan benar. Sebaliknya jika algoritma kurang baik maka penyelesaian lambat dan bahkan tidak didapat solusi yang diharapkan.

Suatu algoritma akan memerlukan masukan (input) tertentu untuk memulainya, dan akan menghasilkan keluaran (output) tertentu pada akhirnya. Hal-hal yang perlu diperhatikan dalam algoritma adalah mencari langkah-langkah yang paling sesuai untuk penyelesaian suatu masalah, karena setiap algoritma memiliki karakteristik tertentu yang memiliki kelebihan dan kekurangan. Beberapa hal yang harus dipahami dalam mencari algoritma antara lain:

1. Masalah seperti apa yang hendak diselesaikan.
2. Gagasan apa yang ada pada algoritma tersebut.
3. Berapa lama yang diperlukan untuk menyelesaikan masalah.
4. Berapa jumlah data yang dapat ditangani oleh suatu algoritma.

Berdasarkan penjelasan diatas, dapat disimpulkan bahwa Algoritma adalah seperangkat alur instruksi dari awal sampai selesai yang bertujuan untuk memecahkan sesuatu, dapat berupa kalimat, gambar, ataupun table tertentu dan bersifat logis ( bernilai salah atau benar )

## 2.4 ALGORITMA PRIM

Algoritma prim adalah sebuah algoritma dalam teori graf untuk mencari pohon rentang minimum untuk sebuah graf berbobot yang saling terhubung. Ini berarti bahwa sebuah himpunan bagian dari edge yang membentuk suatu pohon yang mengandung node, di mana bobot keseluruhan dari semua edge dalam pohon diminimalisasikan. Bila graf tersebut tidak terhubung, maka graf itu hanya memiliki satu pohon rentang minimum untuk satu dari komponen yang terhubung.

Algoritma Prim digunakan untuk mencari pohon pembangkit minimum dari graf terhubung berbobot dengan cara mengambil sisi atau ruas garis yang memiliki bobot terkecil dari graf, dimana ruas garis tersebut bersisian dengan pohon terentang yang telah dibuat dan yang tidak membentuk siklus. Metode ini digunakan untuk menemukan suatu *subset* dari sisi yang membentuk suatu pohon yang melibatkan tiap-tiap titik, dimana total bobot dari semua sisi di dalam pohon adalah minimum.

Misalkan T adalah pohon merentang yang sisi-sisinya diambil dari graf G. Algoritma prim membentuk pohon merentang minimum langkah perlangkah. Pada setiap langkah kita mengambil sisi e dari graf G yang mempunyai bobot minimum dan bersisian dengan bagian simpul-simpul di dalam T tetapi e tidak membentuk sirkuit didalam.

Algoritma ini ditemukan pada 1930 oleh matematikawan Vojtěch Jarník dan kemudian secara terpisah oleh computer scientist Robert C. Prim pada 1957 dan ditemukan kembali oleh dijkstra pada 1959. Karena itu algoritma ini sering dinamai algoritma DJP atau algoritma Jarnik. (Deny Wiria Nugraha)

Algoritma ini dimulai dari suatu titik sembarang, secara bertahap semakin bertambah jumlah nilainya hingga usai ketika telah menelusuri semua titik atau *vertex* yang ada. Berikut adalah langkah-langkahnya:

1. Buat sebuah pohon yang terdiri dari satu node, dipilih secara acak dari graf
2. Buat sebuah himpunan yang berisi semua cabang di graf
3. Loop sampai semua cabang di dalam himpunan menghubungkan dua node di pohon
4. Hapus dari himpunan satu cabang dengan bobot terkecil yang menghubungkan satu node di pohon dengan satu node di luar pohon
5. Hubungkan cabang tersebut ke pohon

### 3. METODOLOGI PENELITIAN

#### 3.1 Tahapan Penelitian

1. Studi kepustakaan bertujuan untuk mengumpulkan data dan informasi ilmiah, berupa teori-teori, metode atau pendekatan yang pernah berkembang dan telah didokumentasikan dalam bentuk buku, majalah, naskah, dan lain sebagainya. Melakukan studi kepustakaan terhadap berbagai referensi yang berkaitan dengan penelitian yang dilakukan. Seperti membaca dan mempelajari buku-buku yang berkaitan dengan algoritma prim.

2. Merancang Program/ Aplikasi

Pada tahapan ini penulis melakukan perancangan sebuah aplikasi/program. Disini penulis merancang program agar dapat menyelesaikan sistem penentuan rute yang tepat dalam sebuah labirin. Langkah pertama dalam perancangan program ini adalah merancang proses kerja sistem dengan menggunakan DFD yang menjelaskan secara rinci proses-proses yang akan dilakukan program dalam menghasilkan

sebuah keputusan. Selanjutnya merancang bentuk tampilan program (*User Interface*).

### 3.2 Analisis Alat Penelitian

#### a. Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan sistem adalah :

- Sistem Operasi : Microsoft Windows 7 Home Standar
- Program Aplikasi : Borland Delphi 7.0

#### b. Perangkat Keras

Untuk mendukung sistem operasi aplikasi yang dirancang maka perangkat yang digunakan adalah :

Microprocessor : Intel ® core™ CPU N570  
RAM/Memory : 3 GB  
Harddisk : 300 GB

#### c. Perangkat Keras

Untuk mendukung sistem operasi aplikasi yang dirancang maka perangkat yang digunakan adalah :

Microprocessor : Intel ® core™ CPU N570  
RAM/Memory: 3 GB Harddisk : 300 GB

### 3.3 Membuat Coding

Program ini menggunakan bahasa pemrograman Delphi7

### 3.4 Implementasi

Pada tahap ini dilakukan pengkodean untuk mengimplementasikan perancangan aplikasi sistem pendukung keputusan kedalam bahasa pemrograman Delphi 7.

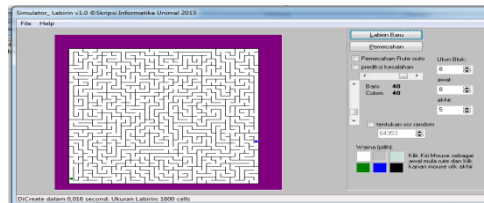
### 3.5 Pengujian Sistem

Pengujian dilakukan dengan mencoba secara detail aplikasi yang ditampilkan oleh interface.

## 4. HASIL DAN PEMBAHASAN

### 4.1 Implementasi Sistem

Program Utama merupakan tampilan yang pertama kali muncul saat program dijalankan. Pada tampilan utama program terdapat menu yang terdiri dari tombol *File* dan tombol *About* .

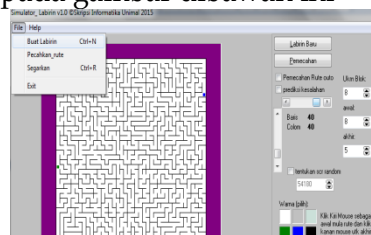


**Gambar 4.1 Form Pengujian Sistem**

Dalam form utama ini semua jenis kegiatan sistem terjadi di form ini, seperti

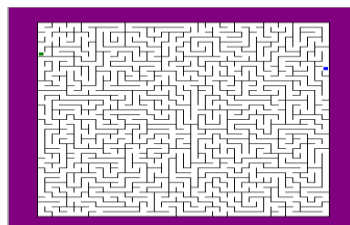
a. Proses Menciptakan labirin

Dalam form ini yang pertama sekali user menciptakan jalur labirin dengan bantuan sistem seperti pada gambar dibawah ini



**Gambar 4.2 Form Proses Menciptakan Labirin**

Pada saat user mengklik menu file dan popup menyunya akan keluar dan memberikan beberapa menu kita memilih buat labirin. Setelah kita klik buat labirin, maka labirin secara acak dengan menentukan matrik secara langsung adalah 4x4 sehingga akan menghasilkan labirin sebagai berikut



**Gambar 4.3 Matrik 4x4**

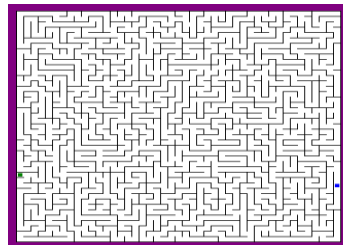
Adapun selain sistem dapat membuat labirin secara acak kita juga bisa menentukan batas kolom dan baris layaknya sebuah matrik yang akan dipolakan sistem menjadi sebuah labirin sesuai yang diinginkan user dengan memodifikasi nomor atau jumlah kolom dan baris yang diinginkan user. Seperti gambar dibawah ini





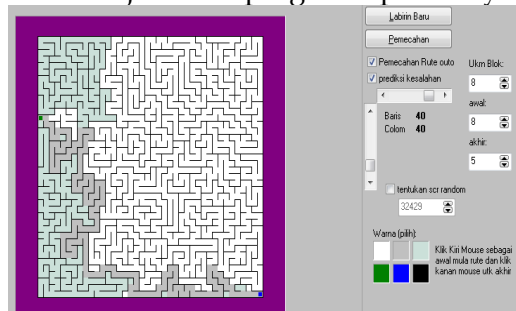
**Gambar 4.4 Form Modifikasi Nomor Dan Colom**

Dari gambar diatas merupakan menu dimana sistem memberi kesempatan pada user untuk membuat labirin sesuai keinginan dari user dimana kolom dan baris ditentukan sendiri oleh user dengan catatan semakin besar kolom dan baris di buat semakin besar dan banyaknya rintangan dari labirin yang akan terbentuk. Adapun hasil dari penentuan kolom dan baris adalah sebagai berikut:



**Gambar 4.5 Form Hasil Modifikasi Nomor Dan Kolom**

Setelah proses pembuatan jalur labirin maka tahap selanjutnya menentukan titik dari awal perjalanan dan titik dimana akhir perjalanan didalam labirin sehingga memiliki jarak tempuh tertentu sesuai keinginan user. Untuk titik awal ditentukan dengan menekan kiri mouse di area labirin akan muncul titik yang berwarna hijau itu merupakan awal dari perjalanan yang ditempuh. Selanjutnya untuk menentukan titik akhir perjalanan maka user menekan bagian kanan mouse di area sekitar labirin sehingga muncul titik biru itu merupakan tujuan akhir dari perjalanan dilabirin. Maka sistem menjalankan program aplikasinya.



**Gambar 4.6 Form Pemecahan Rute**

Setelah penentuan titik awal dan titik akhir oleh user maka secara otomatis sistem akan langsung merekam jarak kolom dan baris antara 1 titik dengan titik yang lain dan menghitungnya sehingga terjadi beberapa simultan yang simultan tersebut dikatakan sebagai prediksi dari perjalanan yang dilalui sebelum laluan yang tepat maka disebut dengan visited yang dihitung dengan jumlah kemungkinan ruangan yang di lewati. Yang jalur tersebut terilustrasi dengan baik kedalam labirin dalam berbagai warna sesuai keinginan user. Baik jalur yang dikunjungi juga jalur yang mentok ketitik akhir yang telah ditentukan user.

Adapun proses algoritma primnya dalam aplikasi ini terdapat source code yang mencerminkan proses algoritma prim dalam memberi solusi dalam memecahkan rute labirin. Prinsip algoritma prim dimana sistem akan merecord titik awal yang ditentukan user sebagai titik start awal dan start akhir yang kemudian di confers ke dalam rosedur solve maze yang kemudian solve maze tersebut diincrement kedalam nilai yang di temporer atau nilai sementara yang kemudian menggunakan fasilitas or dan xor dalam mengekstraksikan nilai tersebut menjadi sebuah patokan dalam penyelesaian rute labirin tersebut.

Secara umum algoritma prim memiliki langkah penyelesaiannya diantaranya:

1. Diberikan sebuah Graph pada himpunan R yang dinyatakan dalam source yaitu temp R, dan himpunan C yang diilustrasikan dalam temp C Himpunan R berisi titik-titik yang telah menjadi jalur di dalam Labirin. Sedangkan himpunan C berisi titik-titik yang belum menjadi jalur di dalam labirin, tetapi bersebelahan dengan titik-titik anggota himpunan R
2. Pilih sebuah titik dari R secara acak dan masukkan ke dalam himpunan C (menentukan titik start)
3. Masukkan titik-titik yang bersebelahan dengan titik-titik yang ada di himpunan R, tetapi bukan anggota himpunan R, ke dalam himpunan C
4. Pilih salah satu titik dari himpunan R secara acak, sebut saja titik i. Buat jalan diantara titik i dengan titik yang bersebelahan dengan titik i dan merupakan anggota himpunan R
5. Ulangi langkah ke-2 dan 3 sampai semua titik telah menjadi anggota himpunan R

Maka terciptalah jalur labirin yang kemudian sistem akan mengkalkulasikan jalur labirin tersebut sehingga tercapai sebuah jalur yang memecahkan labirin tersebut.

Setelah sistem menganalisa posisi titik awal yang dianggap sebagai titik start dari rute labirin dan dari perpaduan nilai R dan C maka akan dibandingkan dengan titik-titik rute yang dianggap sebagai rute yang dianggap sebagai rute utama dan menganalisa tepi hingga membentuk satu jalur labirin yang menghubungkan antara titik awal hingga titik akhir dan terciptalah jalur atau rute yang menjadi solusi dalam memecahkan sebuah labirin.

**Analisis Algoritma Prim Terhadap Sistem.**

Pengujian sistem dilakukan dengan beberapa model matrik yang menjadi dasar pembentukan labirin dan waktu dibutuhkan untuk menyelesaikan labirin dan juga jumlah ruang atau sel yang terbentuk serta jumlah solusi yang dihasilkan dalam aplikasi

Dalam aplikasi ada beberapa model matrik yang ditentukan secara standar yang telah di tentukan oleh aplikasi. Dari hasil pengujian maka didapat hasil table sebagai berikut:

Tabel 4.2 titik awal dan titik akhir berarah vertikal

Matrik	Time create	Solution time	Maze size	Solution	Cell invite
20x20	0.00	0.00	400 ruang	57 laluan	207 laluan
20x20	0.00	0.00	400 ruang	85 laluan	176 laluan
20x20	0.00	0.00	400 ruang	71 laluan	278 laluan
30x30	0.00	0.00	900 ruang	109 laluan	573 laluan
30x30	0.016	0.00	900 ruang	112 laluan	468 laluan
30x30	0.016	0.00	900 ruang	145 laluan	611 laluan
40x40	0.00	0.00	1600 ruang	137 laluan	1289 laluan
40x40	0.00	0.00	1600 ruang	198 laluan	1209 laluan
40x40	0.015	0.00	1600 ruang	182 laluan	496 laluan

Tabel 4.3 titik awal dan titik akhir berarah horizontal

Matrik	Time create	Solution time	Maze size	Solution	Cell invite
20x20	0.00	0.00	400 ruang	67 laluan	353 laluan
20x20	0.00	0.00	400 ruang	85 laluan	190 laluan
20x20	0.016	0.00	400 ruang	71 laluan	296 laluan
30x30	0.016	0.00	900 ruang	161 laluan	504 laluan
30x30	0.15	0.00	900 ruang	255 laluan	1501 laluan
30x30	0.16	0.00	900 ruang	144 laluan	1439 laluan
40x40	0.16	0.00	1600 ruang	156 laluan	411 laluan
40x40	0.00	0.00	1600 ruang	704 laluan	144 laluan
40x40	0.16	0.00	1600 ruang	154 laluan	737 laluan

Dari table diatas kondisi titik awal dan titik akhir itu berarah vertical untuk kesemua matrik dan memberi hasil seperti diatas dengan ukuran labirin yang berbeda dan kemungkinan ruang yang dilalui juga berbeda selaras dengan jumlah ruang yang tercipta menurut matrik yang ditentukan. Begitu pula jika di posisikan titik laluan secara horizontal maka

hanya solusi dan ruang yang dikunjungi berbeda. Ini membuktikan bahwa sistem simulasi labirin ini dari proses penciptaan labirin sampai dengan menghasilkan solusi menurut letak titik awal dan titik akhir dari sebuah perjalanan atau rute.

### 1. Form Login

*Formlogin* digunakan oleh admin untuk masuk kedalam sistem. Jika *username* dan *password* yang dimasukkan sesuai dengan yang ada di *database*, maka akan dialihkan ke halaman utama admin. Jika tidak sesuai, maka sistem akan menampilkan pesan kesalahan kepada *user*. Berikut ini tampilannya.



Gambar 4.7 Form Login

### 2. Form Utama

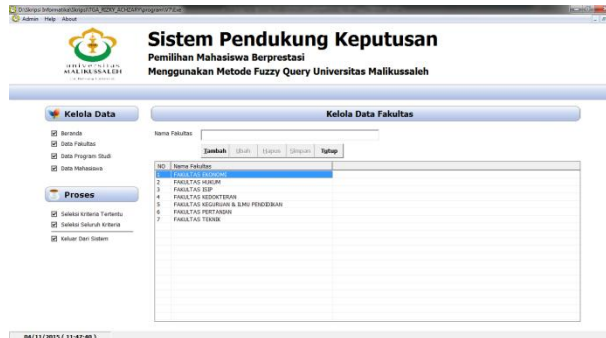
Berikut ini tampilan dari *form* utama setelah pengguna berhasil *login* kedalam sistem. Form ini menampilkan / berisi menu untuk mengelola konten sistem. Berikut ini tampilan form utama:



Gambar 4.8 Form Utama

### 3. Form Kelola Data Fakultas

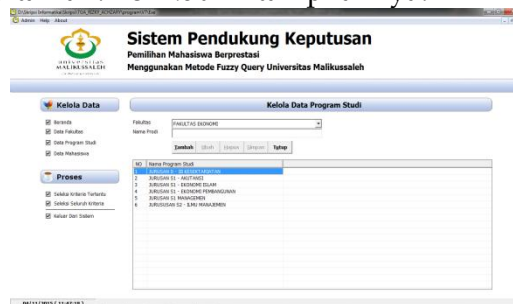
*Form* kelola data fakultas digunakan untuk mengelola data fakultas. Berikut ini tampilannya.



Gambar 4.9 Form Kelola Data Fakultas

#### 4. Form KelolaData Program Studi

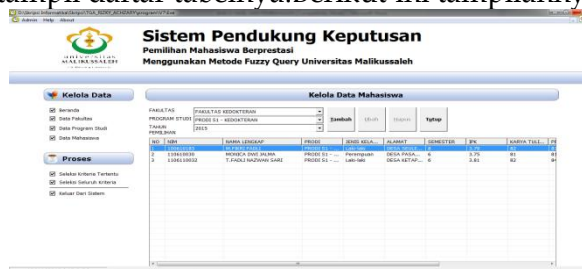
Form ini digunakan untuk mengelola data-data program studi yang ada di fakultas pertanian. Berikut ini tampilannya.



Gambar 4.10 Form Data Program Studi

#### 5. Form Kelola Data Mahasiswa

Form ini digunakan untuk mengelola data mahasiswa tiap tahunnya untuk tiap program studi. Pengguna cukup memilih program studi yang ada, dan akan tampil daftar tabelnya. Berikut ini tampilannya.



Gambar 4.11 Form Kelola Data Mahasiswa

#### 6. Form Pengisian Data Mahasiswa

Form ini digunakan untuk menambah atau mengubah data mahasiswa. Berikut ini tampilannya.

**Sistem Pendukung Keputusan**  
Pemilihan Mahasiswa Berprestasi  
Menggunakan Metode Fuzzy Query Universitas Malikussaleh

**Kelola Data Mahasiswa**

Beranda  
 Data Fakultas  
 Data Program Studi  
 Data Mahasiswa

**Proses**

Seleksi Kriteria Tertentu  
 Seleksi Seluruh kriteria  
 Keluar Dari Sistem

FAKULTAS: FAKULTAS EKONOMI  
 PROGRAM STUDI: JURUSAN D - III KESEKRETARIATAN  
 TAHUN: 2015  
 NIM: \_\_\_\_\_  
 NAMA LENGKAP: \_\_\_\_\_  
 JENIS KELAMIN: Laki-laki  
 ALAMAT: \_\_\_\_\_  
 SEMESTER: \_\_\_\_\_  
 IP KUMULATIF (3.00 - 4.00): \_\_\_\_\_  
 KARVA TULIS ELMISAH (50 - 100): \_\_\_\_\_  
 PRESTASI / KEMAMPUAN YANG DININGGULKAN (5.5 - 100): \_\_\_\_\_  
 BAHASA INGGRIS / ASING (310 - 677): \_\_\_\_\_  
 KEPRIBADAN (60 - 100): \_\_\_\_\_

04/11/2015 ( 11:44:40 )

Gambar 4.12 Form Pengisian Data Mahasiswa

### 7. Form Seleksi Mahasiswa Berprestasi

Dalam *Form* ini terbagi atas 2 kategori seleksi mahasiswa berprestasi yakni Seleksi kriteria tertentu dan Seleksi seluruh kriteria.

**Sistem Pendukung Keputusan**  
Pemilihan Mahasiswa Berprestasi  
Menggunakan Metode Fuzzy Query Universitas Malikussaleh

**Seleksi Kriteria Tertentu**

Beranda  
 Data Fakultas  
 Data Program Studi  
 Data Mahasiswa

**Proses**

Seleksi Kriteria Tertentu  
 Seleksi Seluruh Kriteria  
 Keluar Dari Sistem

Tahun: 2014  
 Fakultas - Jurusan: SEMUA FAKULTAS  
 Kriteria: IP Kumulatif  
 Nilai Kriteria: Sangat Baik

NO	NIM	NAMA LENGKAP	FAKULTAS - JURUSAN	IPK	Kry Tulis	Kemampuan	Bhs Asing
1	120510057	ALFAMOLI	FAKULTAS HUKUM - PRODI SI - B...	3.89	90	90	598
2	120510105	ANDRIANTA BANGUN	FAKULTAS HUKUM - PRODI SI - B...	3.85	90	88	578
3	110510061	ABRIL MAULIDA	FAKULTAS HUKUM - PRODI SI - B...	3.81	87	80	523
4	1106110032	T. FADLI HAZWAN SARI	FAKULTAS KEDOKTERAN - PRODI S...	3.81	82	84	517
5	100610183	M.FIKRI FADLI	FAKULTAS KEDOKTERAN - PRODI S...	3.79	82	81	518
6	110610030	MONICA DWI JALMA	FAKULTAS KEDOKTERAN - PRODI S...	3.75	81	81	510
7	100310014	MURHAFI	FAKULTAS PERTANIAN - JURUSAN ...	3.73	80	81	516
8	110310161	MIRZA PUTRA	FAKULTAS PERTANIAN - JURUSAN ...	3.71	80	80	515
9	110170078	RAHMI PUTRI	FAKULTAS TEKNIK - JURUSAN SI - ...	3.69	80	80	511
10	110310123	M.ZULFARMI WAHYUDI	FAKULTAS PERTANIAN - JURUSAN ...	3.65	80	80	514
11	110110086	AGUNG SUPRIADI	FAKULTAS TEKNIK - JURUSAN SI - ...	3.63	80	80	459

12/11/2015 ( 15:2:20 )

Gambar 4.13 Form Seleksi Kriteria Tertentu

### 8. Laporan Hasil Seleksi Mahasiswa Berprestasi

Berikut ini laporan hasil seleksi pemilihan mahasiswa berprestasi.

**DAFTAR HASIL SELEKSI MAHASISWA BERPRESTASI  
 UNIVERSITAS MALIKUSSALEH  
 TAHUN AJARAN 2015**

NO	NIM	NAMA LENGKAP	FAKULTAS	PRODI	IPK	Karya Tulis	Kemampuan	Bhs Asing	Kepribadian	NILAI
1	100510057	ALFIANDI	FAKULTAS HUKUM	PRODI S1 - ILMU HUKUM	3.89	90	90	598	95	876.89
2	120510105	ANDRIANTA BANGUN	FAKULTAS HUKUM	PRODI S1 - ILMU HUKUM	3.85	90	88	578	90	849.85
3	110510061	ABRIL MAULIDA	FAKULTAS HUKUM	PRODI S1 - ILMU HUKUM	3.81	87	80	523	89	782.81
4	1106110032	T.FADLI NAZWAN SARI	FAKULTAS KEDOKTERAN	PRODI S1 - KEDOKTERAN	3.81	82	84	517	87	773.81
5	100610183	M.FIKRI FADLI	FAKULTAS KEDOKTERAN	PRODI S1 - KEDOKTERAN	3.79	82	81	518	88	772.79
6	100310014	MURHAFI	FAKULTAS PERTANIAN	JURUSAN S1 - AGROTEKNOLOGI	3.73	80	81	516	80	760.73
7	110310161	MIRZA PUTRA	FAKULTAS PERTANIAN	JURUSAN S1 - AGROTEKNOLOGI	3.71	80	80	515	81	759.71
8	110610030	MONICA DWI JALMA	FAKULTAS KEDOKTERAN	PRODI S1 - KEDOKTERAN	3.75	81	81	510	83	758.75
9	110310123	M.ZULFAHMI WAHYUDI	FAKULTAS PERTANIAN	JURUSAN S1 - AGROTEKNOLOGI	3.65	80	80	514	80	757.65
10	110170078	RAHMI PUTRI	FAKULTAS TEKNIK	JURUSAN S1 - TEKNIK INFORMATIKA	3.69	80	80	511	80	754.69
11	110110086	AGUNG SUPRIADI	FAKULTAS TEKNIK	JURUSAN S1 - TEKNIK SIPIL	3.63	80	80	459	80	702.63

**Gambar 4.14 Laporan Hasil Seleksi Pemilihan Mahasiswa Berprestai**

## 9. Interpretasi

Aplikasi ini dibuat untuk menentukan mahasiswa/i yang berprestasi dengan beberapa kriteria yakni : IP Kumulatif, Karya Tulis Ilmiah, Prestasi Yang Diunggulkan, Bahasa Asing, Kepribadian. Pada setiap nilai kriteria akan di *fuzzy* kan dengan cara mencari nilai keanggotaannya setelah itu akan dimulai proses *query* yakni pemanggilan data, lalu tahapan selanjutnya mencetak laporan dari hasil pemilihan mahasiswa yang terpilih. Pada aplikasi ini terpilih mahasiswa berprestasi yakni Alfiandi mahasiswa Fakultas Hukum dengan rincian nilai, IP Kumulatif : 3.89, Karya Tulis Ilmiah : 90, Prestasi/Kemampuan yang Diunggulkan : 90, Bahasa Inggris/Asing : 598, Kepribadian : 95 sehingga *firestenghyang* didapat 876.

## 5.KESIMPULAN DAN SARAN

### Kesimpulan

Setelah melakukan penelitian dan pembahasan mengenai sistem pencarian rute terpendek menggunakan algoritma prim, maka dapat diambil kesimpulan sebagai berikut :

1. Dengan adanya sistem ini penjelajah dengan mudah dapat mengetahui rute mana yang akan dilalui untuk menempuh titik tujuan dengan dapat menganalisa kemungkinan kesalahan dalam rute tersebut.
2. Hasil dari pencarian rute kondisi titik awal dan titik akhir berarah vertical untuk ukuran matrik 20x20 solutionnya 57 laluan. Ukuran

matrik  $30 \times 30$  solutionnya 109 laluan. Ukuran matrik  $40 \times 40$  solutionnya 137 laluan.

3. Jika di posisikan titik laluan secara horizontal maka hanya solusi dan ruang yang dikunjungi maka hasil pencariannya adalah ukuran matrik  $20 \times 20$  solutionnya 67 laluan. Ukuran matrik  $30 \times 30$  solutionnya 161 laluan. Ukuran matrik  $40 \times 40$  solutionnya 156 laluan.

## Saran

Berikut ini adalah saran yang mungkin dapat digunakan untuk pengembangan sistem ini lebih lanjut :

- a. Sistem penentuan rute terpendek pada labirin dibangun dengan menggunakan algoritma prim, akan lebih baik sistem ini dicoba dengan menggunakan metode yang lain sehingga dapat diketahui kekurangan dan kelebihan dari masing-masing metode.
- b. Perancangan berikutnya diharapkan dapat menyempurnakan bagian desain agar tampak lebih menarik.

## DAFTAR PUSTAKA

- Aswin, M, 2012. *Pencarian Rute Pada Labirin Menggunakan Metode A\**
- Didik, E, W, 2012. *Pemodelan Sistem dengan UML*. Teknik Sistem Komputer, Universitas Diponegoro
- Insani Nur, 2012. *Penerapan Teori Graf Pada Analisis Jejaring Sosial Dengan Menggunakan Microsoft Nodexl*. Jurusan Pendidikan Matematika Fakultas Matematika Ilmu Pengetahuan Alam Universitas Yogyakarta.
- Kadir Abdul, 2014. *Buku Pintar Pemrograman Delphi Untuk Pemula*. Yogyakarta, Mediacom.
- Manongga Denny, 2013. *Matematika Diskrit*. Prenada Media Group. Bandung
- Martina, Inge, 2010. *36 Jam Belajar Komputer Pemograman Visual Borland Delphi 7*, Jakarta, Elex Media Komputindo
- M Musafi Saif Noor, Ibrahim DR, 2013. *Pengantar Kombinatorika & Teori Graf*. Graha Ilmu. Yogyakarta
- Ratna sari Lucia, 2012. *Energi Derajat Maksimal pada Graf Terhubung*. Program Studi Matematika, Jurusan Matematika, Universitas Diponegoro Semarang



Jurnal Sistem Informasi ISSN P : 2598-599X; E: 2599-0330  
Vol 2.No.1 2018

Ruminta DR, 2014. "*Matriks Persamaan Linier dan Pemrograman Linier (Edisi Revisi)*", Informatika.

Shalahuddin, M. Rosa, AS, 2010. *Algoritma Dan Pemograman*. Jakarta. Informatika.

Wiria nugraha deny, 2011. *Implementasi Algoritma Prim Pada Jaringan Distribusi Listrik Primer Dengan Menggunakan Program Berbasis Gis*. Jurusan Teknik Elektro Fakultas Teknik Universitas Taduloka Palu