|  |
| --- |
| Program A\* |
| Laporan Tugas Program Kecerdasan Artifisial |
| Dosen Pengampu: Mahmud Dwi Sulistyo S.T, M.T  Nama: Salim ‘Imaduddiyan  NIM: 1103134378 |

2/28/2016

**2016**

# Deskripsi Kasus

Pada Tugas Program 1 ini kita diminta membantu Flavius Aetius salah satu ksatria Romawi dan pasukannya untuk menemukan jalur sungai yang paling pendek dari Bobbia menuju Revenna. Agar dapat menemukan jalur sungai yang paling optimal dan complete dari Bobbia menuju Revenna, kita dapat menggunakan Algoritma A\*. Dimana telah disediakan peta jalur sungai beserta jarak dan nilai heuristik masing-masing kota yang dilalui untuk menuju Revenna.

# Data yang Digunakan

Data yang harus kita miliki peta jalur sungai untuk mengetahui keterkaitan antar kota, jarak antar kota serta nilai heuristik setiap kota untuk menuju Revenna.

# Desain Metode dan Implementasi Algoritma

1. Struktur data pada implementasi di program

Program yang saya rancang menggunakan :

* Dua Kelas utama yaitu Kota dan jalur
  + Kota

Kota merepresentasikan kota mana saja yang dilalui. Pada Kelas ini terdapat beberapa atribut dan method, yaitu :

Atribut :

* + - Nama bertipe string untuk menyimpan nama kota.
    - List bertipe ArrayList dari kelas Jalur, untuk menyimpan data kota tujuan.
    - H\_ke\_revenna bertipe double, untuk menyimpan nilai heuristik ke kota Revenna.
    - F bertipe double, untuk menyimpan nilai F kota tersebut.
    - G bertipe double, untuk menyimpan nilai G kota tersebut.
    - Kota\_asal bertipe kota, untuk menyimpan data kota parent-nya.

Method :

* + - Setter, untuk memberi nilai pada atribut yang terdapat dalam kelas tersebut.
    - Getter, untuk mengambil nilai dari atribut kelas tersebut.
    - GenerateF, untuk menghitung nilai F dari kota tersebut.
    - AddJalur, untuk menambahkan daftar kota suksessor.
    - Constructor, untuk menginstansiasi Object kelas tersebut.
  + Jalur

Jalur merepresentasikan daftar kota suksessor beserta jaraknnya. Pada kelas ini terdapat beberapa atribut dan method, yaitu :

Atribut :

* + - Kota\_tujuan bertipe Kota, untuk menyimpan data kota tujuan.
    - Jarak bertipe double, untuk menyimpan jarak antara kota asal dengan kota tujuan.

Method :

* + - Setter, untuk memberi nilai pada atribut yang terdapat dalam kelas tersebut.
    - Getter, untuk mengambil nilai dari atribut kelas tersebut.
    - Constructor, untuk menginstansiasi Object kelas tersebut.

1. Algoritma/psudocode A\*

Ulangi ketika ArrayList Open tidak kosong dan bestnode bukanlah kota tujuan{

Jika, ArrayList Open kosong, maka gagal

Jika tidak maka

Cari bestnode dalam ArrayList Open

Masukkan bestnode ke dalam ArrayList Closed

Hapus bestnode dari ArrayList Open

Jika bestnode adalah kota tujuan, maka hentikan perulangan

Jika tidak, maka

Untuk setiap suksessor bestnode lakukan

Tampung nilai g bestnode dalam suatu variabel

Cek apakah bestnode terdapat dalam ArrayList Open

Jika iya,

Cek apakah nilai g yang lama lebih buruk dari pada yang baru,

Jika iya,

Rubah alur melalui jalur yang baru dan perbarui nilai G dan F

Cek apakah bestnode terdapat dalam ArrayList Closed

Jika iya,

Cek apakah nilai g yang lama lebih buruk dari pada yang baru,

Jika iya,

Rubah alur melalui jalur yang baru dan lakukan propagasi,

Jika kota seksessornya terdapat dalam ArrayList Open, maka teruskan propagasi.

Jika nilai G kota suksessor lebih buruk dari nilai G baru, maka teruskan propagasi

Jika bestnode tidak terdapat dalam ArrayList Open maupun ArrayList Closed, maka

Tambahkan suksessor bestnode ke dalam ArrayList Open

Update nilai G dan F suksessor bestnode

Atur agar bestnode menjadi parent dari suksessor bestnode

}

1. Cara menentukan bestnode pada setiap iterasi

Kita dapat menentukan bestnode untuk setiap iterasi dengan mengecek nilai F terkecil setiap kota yang terdapat dalam ArrayList Open (ArrayList yang berguna menampung data kota yang berstatus open).

1. Penjelasan bagaimana iterasi dalam proses pencarian dihentikan

Iterasi dalam proses pencarian dapat berhenti apabila bestnode merupakan kota yang dituju yaitu Revenna.

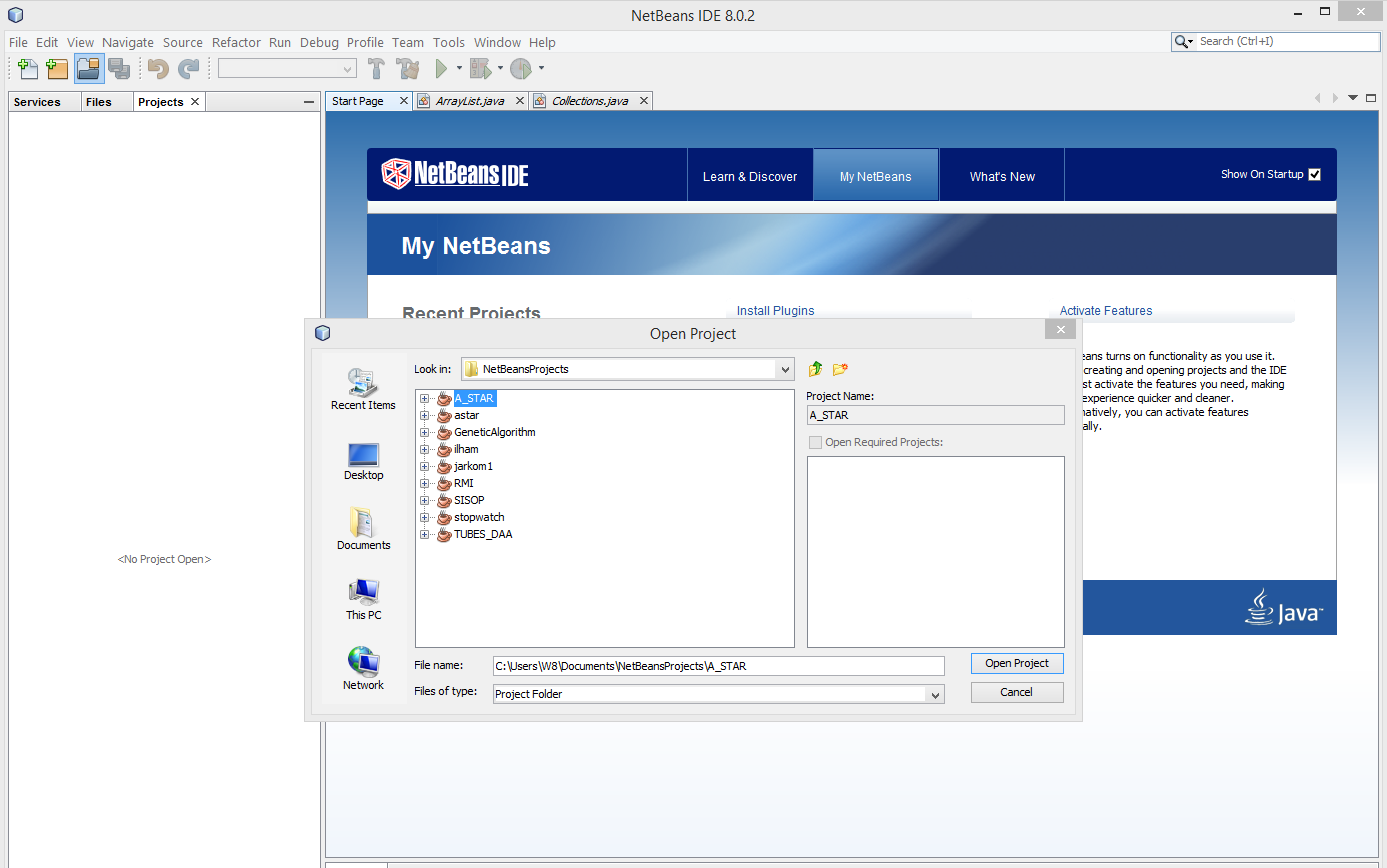
# Petunjuk Penggunaan Program

Program dibuat dengan menggunakan bahasa pemrogaman java dengan menggunakan Netbeans sebagai Integrated Development Environment(IDE). Sehingga **disarankan** untuk menjalankan program pada IDE tersebut. Tata cara running program :

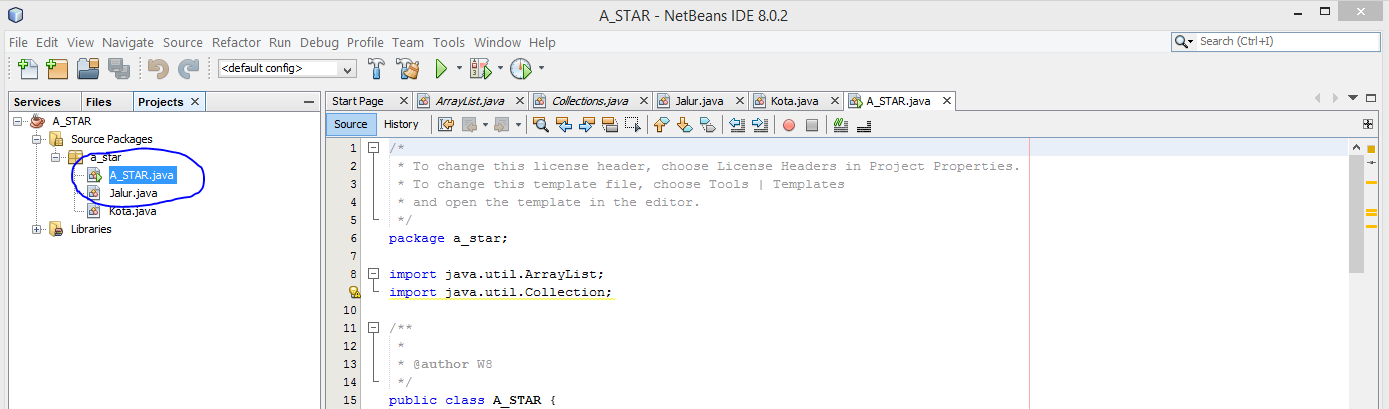
1. Buka aplikasi Netbeans.



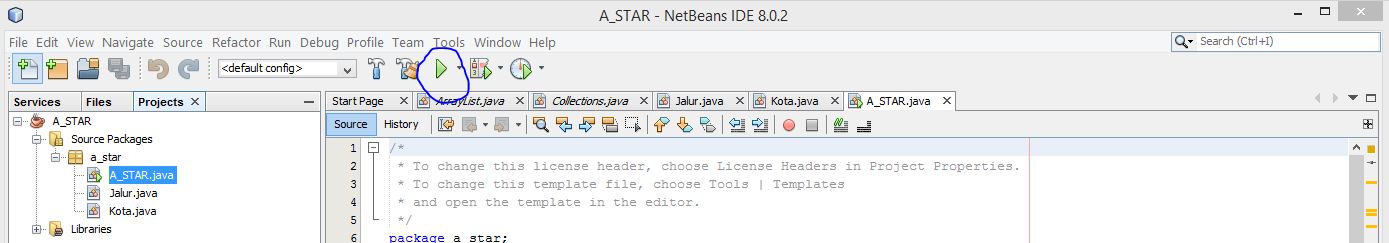
1. Open Project, pada direktori dimana file berada.



1. Klik pada kelas A\_STAR



1. Compile dan jalankan Pogram



# Luaran Program

Output yang dihasilkan dalam program ini adalah urutan langkah pencarian jalur terpendek menggunakan algoritma A\* beserta solusi (Daftar kota yang dilalui) dan jarak optimalnya.

# Screenshot Program

