

## Tugas Besar I IF3051 Strategi Algoritma

### Aplikasi Algoritma *Greedy* pada Permainan *Tower Defense*

**Batas pengumpulan :** 28 September 2012

**Arsip pengumpulan :** - CD yang berisi *Source* dan *Exe* program disertai *readme.txt*  
- Laporan (*hard copy*)

**Tempat pengumpulan :** Di atas loker Lab IRK

#### Sekilas Pemainan *Tower Defense*

Permainan *Tower Defense* (TD) merupakan permainan strategi pengaturan *tower* (bangunan, senjata, naga, tanaman, dll) yang bertujuan menghentikan musuh yang akan melintas. *Tower* ini akan menembaki musuh dalam radiusnya. *Tower* memiliki beberapa tipe dan level dengan kemampuan, biaya pembelian, biaya penggunaan, dan biaya upgrade yang berbeda. Untuk setiap musuh yang berhasil dikalahkan, pemain akan mendapatkan uang untuk meng-*upgrade* towers. Strategi dalam permainan ini mencakup pemilihan *tower* yang akan di-*upgrade*.

Permainan ini biasanya memiliki misi untuk meminimalkan jumlah musuh yang lolos atau menjaga sejumlah barang. Pemain dianggap menang jika jumlah musuh yang lolos kurang dari batas yang ditentukan atau barang yang dijaga masih ada.



Kingdom Rush



Xeno Tactic



Lair-defense



Plant vs zombies

Di dalam tugas ini, Anda diminta mengaplikasikan algoritma *greedy* untuk memenangkan permainan TD. Komputer bermain menggunakan algoritma *greedy*, sedangkan manusia bermain menggunakan akal dan intuisinya. Program yang dibuat harus memungkinkan melakukan permainan TD dengan pemainnya adalah:

1. *User* (manusia) lawan komputer
2. Komputer dengan dirinya sendiri.

Jika pemainnya adalah *user* (manusia) versus komputer, maka manusia akan mengatur kemampuan sejumlah *tower* yang ada, dan komputer akan mengaplikasikan algoritma *greedy* untuk melakukan penyusunan rombongan musuh (*wave*) agar semakin banyak musuh yang dapat lolos. Penyusunan rombongan musuh dibatasi oleh jumlah uang yang tersedia. Penyusunan *wave* ini akan dilakukan setelah *tower* selesai diatur.

Jika pemain adalah komputer versus dirinya sendiri, maka pihak komputer pertama mengaplikasikan algoritma *greedy* untuk penyusunan *wave*, dan pihak komputer kedua mengaplikasikan algoritma *greedy* untuk pengaturan kemampuan *tower*. Pengaturan *tower* akan dilakukan setelah hasil penyusunan *wave* diberikan.

Anda harus merancang minimal masing-masing dua buah strategi *greedy* yang berbeda (*greedy by X* dan *greedy by Y*) untuk penyusunan rombongan musuh (*wave*) dan strategi pengaturan kemampuan *tower*. Deskripsi algoritma *greedy* tersebut harus dapat memperlihatkan properti algoritmanya, yaitu himpunan kandidat, himpunan solusi, fungsi seleksi, fungsi kelayakan, dan fungsi obyektif. Setelah algoritma *Greedy* anda rancang, implementasikan menjadi sebuah program yang membolehkan anda berkreaitivitas seperti yang dijelaskan di dalam spesifikasi di bawah ini.

#### **Spesifikasi program :**

1. Program mampu menampilkan peta/background yang memperlihatkan jalur yang akan dilalui musuh, sejumlah *tower* yang tersedia, jumlah uang, dan pintu masuk/keluar.
2. Penyusunan *wave* ataupun pengaturan kemampuan *tower* dibatasi oleh jumlah uang yang telah ditentukan. Aplikasi algoritma *greedy* ditentukan oleh strategi *greedy* yang telah didefinisikan, yaitu minimal dua strategi *greedy* yang berbeda untuk setiap task.
3. Temanya dibebaskan antar kelompok, tetapi setiap kelompok harus menyatakan secara eksplisit goal permainan, spesifikasi setiap unit musuh, dan spesifikasi setiap tipe dan level *tower* yang tersedia termasuk siklus unggul antara unit musuh dan tipe *tower*. Dalam setiap aksi menyerang, setiap unit akan selalu menghasilkan pengurangan sebesar  $x$  poin pada target. Perlu diperhatikan bahwa terdapat pengali untuk beberapa kondisi:
  - Unggul:  $1,5X$  damage yang dihasilkan
  - Kalah:  $0,5X$  damage yang dihasilkan
  - Setara:  $1X$  damage yang dihasilkan

Rancanglah semakin kuat suatu *tower*, *upgrade* level semakin mahal. Definisikanlah untuk setiap level.

4. Program dapat memberikan informasi *tower* yang menyerang setiap unit musuh, dan *health point* (HP) setiap unit musuh. Unit musuh dengan  $HP=0$  akan dihapus. Untuk penyederhanaan, penyerangan dilakukan secara sekuensial. *Tower* diasumsikan selalu menyerang musuh yang melewati depannya.
5. Jika pemain adalah komputer melawan dirinya sendiri, maka kecepatan gerakan pemain dapat diatur (lambat, sedang, cepat). Rombongan musuh yang akan menyerang diketahui dan dapat digunakan untuk menyusun strategi pengaturan kemampuan *tower*.

### Lain – lain :

1. Anda dapat menambahkan *features* lain yang menunjang program yang anda buat (unsur kreatifitas). Jika diperlukan penyederhanaan, jelaskanlah alasannya pada laporan dan deskripsikanlah penyederhanaan yang dilakukan tersebut.
2. Program ini harus Anda buat berbasis *Graphical User Interface* (GUI), kaskas pengembangan aplikasi yang diperbolehkan adalah kaskas klasik seperti *Microsoft Visual C*, *Microsoft Visual C++*, *Borland C++ Builder*, dan *Borland Delphi*, namun tidak boleh menggunakan *Java* dan kaskas lain yang berbasis .net seperti VB .net, MS Visual C# dll.
3. Tugas dikerjakan per kelompok dengan jumlah anggota adalah 3 orang dan boleh lintas kelas.
4. Program harus modular dan mengandung komentar yang jelas.
5. Mahasiswa harus membuat program sendiri, tetapi belajar dari contoh-contoh program *game* serupa yang sudah ada tidak dilarang (tidak boleh mengkopi *source code* dari program orang lain).
6. Pengumpulan paling lambat adalah tanggal 28 September 2012 pukul 18.00. Keterlambatan akan mengurangi nilai.
7. Program disimpan di dalam *folder* StrAlgo1-xxxxx. Lima digit terakhir adalah NIM anggota terkecil. Di dalam folder tersebut terdapat tiga folder *bin*, *src* dan *doc* yang masing-masing berisi :
  - a. Folder *bin* berisi *executable file* (*exe*)
  - b. Folder *src* berisi *source code* dari program
  - c. Folder *doc* berisi dokumentasi program dan *readme*

*Folder* ini disimpan dalam bentuk CD untuk dikumpulkan bersama berkas laporan dimasukan kedalam amplop coklat.
8. Semua pertanyaan menyangkut tugas ini harus dikomunikasikan melalui milis agar dapat dicermati oleh semua peserta kuliah IF3051 (milis [IF3051@students.if.itb.ac.id](mailto:IF3051@students.if.itb.ac.id) dan di cc ke milis asisten : [AsistenIRK@students.if.itb.ac.id](mailto:AsistenIRK@students.if.itb.ac.id)).
9. Demo program akan dilaksanakan setelah pengumpulan. Jadwal demo akan diumumkan pada saat pengumpulan di Lab IRK. Peserta mengisi jadwal demo yang disediakan pada saat pengumpulan tugas.
10. Tiap anggota harus memahami proses pembuatan program, karena akan ada pertanyaan-pertanyaan yang harus dijawab per individu.
11. Pada saat demo, asisten akan memanggil per kelompok sesuai jadwal yang diisi sebelumnya. Kelompok yang tidak berkepentingan dilarang masuk. Demo dilakukan di Lab IRK.

### Isi laporan :

*Cover*: *Cover* laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo “gajah” ganesha.

Bab 1: Deskripsi masalah (dapat meng-copy paste file tugas ini)

Bab 2: Dasar teori (berisi deskripsi singkat algoritma *greedy*).

Bab 3: Analisis Pemecahan Masalah. Di dalam bab ini diuraikan langkah-langkah pemecahan masalah dengan algoritma *greedy* (ada dua strategi *greedy* yang harus anda buat). Anda harus menuliskan apa yang menjadi elemen-elemen algoritma *greedy*-nya pada masalah ini (himpunan kandidat, himpunan solusi, fungsi seleksi, dll). Nilai bonus bagi anda jika dapat membuktikan apakah algoritma *Greedy* anda itu memberikan solusi optimal dan contoh kontranya (contoh yang tidak selalu memberikan solusi optimal).

Bab 4: Implementasi dan pengujian. Bab ini berisi:

- a. Spesifikasi teknis program, termasuk di dalamnya struktur data, fungsi dan prosedur (*header* fungsi dan prosedur saja, tidak perlu *source code*), antarmuka, dan lain-lain yang dianggap perlu.
- b. *Capture layar* yang memperlihatkan beberapa keadaan papan dam dan kondisi akhir papan (akhir permainan).

c. Analisis hasil pengujian.

Bab 5: Kesimpulan dan saran (hasil yang dicapai, saran pengembangan).

Tuliskan juga referensi (buku, web), yang dipakai/diacu di dalam Daftar Referensi.

**Keterangan laporan :**

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar, tidak perlu panjang tetapi tepat sasaran dan jelas.
2. Laporan tidak perlu memakai *cover* mika dan dijilid. Cukup dibuat agar laporan tidak akan tercecer bila dibaca.
3. Laporan boleh menggunakan kertas riu, boleh bolak-balik, boleh dalam satu halaman kertas terdapat dua halaman tulisan asalkan masih terbaca.
4. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

**Penilaian :**

1. Kebenaran program (40%) : program mampu berjalan sesuai dengan spesifikasi yang diberikan.
2. Demo – pemahaman Anda dalam pembuatan program (30%)
3. Laporan (20%)
4. *Interface, features* program, dan unsur kreativitas (20%)

*-selamat mengerjakan-*