

# Tugas Besar 1 IF2211 Strategi Algoritma

## Semester 2 tahun 2018/2019

### Aplikasi Permainan Kartu 24 dengan Algoritma Greedy

**Batas pengumpulan :** Rabu, 13 Februari 2019 sebelum pukul 13.00 WIB

**Arsip pengumpulan :**

- Kode *source* yang bisa dijalankan disertai *readme.txt*, diupload di dropbox
- Laporan (*hard copy*)

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

Dalam permainan kartu 24, terdapat dek (tumpukan) 52 kartu remi. Permainan akan memilih 4 kartu secara acak, lalu setiap pemain akan mencari solusi 24 game dari ke-4 kartu tersebut. Nilai yang mungkin dari sebuah kartu adalah 1 (as), 2, ..., 10, 11 (jack), 12 (queen), dan 13 (king). Operator yang dapat dipilih + - \* / ( ), dan hasil akhir sedekat mungkin dengan nilai 24. Selisih nilai ekspresi solusi dengan 24 akan menjadi pengurang.

Dalam tugas besar ini, setiap tim wajib merancang dan mengimplementasikan strategi greedy untuk memberikan solusi dalam permainan ini. Karena algoritma greedy membentuk solusi langkah per langkah (step by step), harus ditentukan urutan pemilihan operand, urutan pemilihan operator, dan penggunaan variasi kurung. **Tidak boleh** menggunakan strategi lain selain greedy.

Fungsi objektif persoalan ini adalah memaksimalkan skor utk ekspresi solusi yang dihasilkan. Seperti scrabble, setiap operator akan memiliki skor. Semakin kompleks operatornya, skor semakin kecil. Skor setiap operator didefinisikan 5 untuk +, 4 untuk -, 3 untuk \*, dan 2 untuk /, serta -1 untuk setiap pasang kurung (). Selain skor, operator \* dan / memiliki derajat lebih tinggi dibandingkan + dan -, artinya operator berderajat lebih tinggi akan diproses terlebih dahulu. Ekspresi  $a+b*c-d$  akan diproses seperti  $(a+(b*c))-d$ . Skor akhir ekspresi yang sama dengan 24 tetapi dengan nilai total operator yang lebih rendah akan memiliki bobot yang lebih tinggi daripada ekspresi dengan nilai total operator yang lebih tinggi tetapi tidak menghasilkan nilai 24. Semakin dekat skor akhir ekspresi dengan 24, dengan nilai total operator yang sama, bobotnya akan semakin tinggi.

Setiap tim akan membuat satu engine backend yang menghasilkan ekspresi solusi berdasarkan masukan 4 angka, dan dua front-end. Front-end pertama berupa GUI yang mendemokan proses pengambilan 4 kartu untuk memberikan input, dan menampilkan hasilnya. Visualisasi kartu untuk demo boleh menggunakan library. Front-end kedua membaca file masukan, memproses 4 angka dari file masukan, dan menghasilkan file keluaran. Front-end kedua akan berinteraksi dengan lingkungan permainan untuk kompetisi antar tim.

Lingkungan permainan akan mengeluarkan 4 kartu secara acak. Setiap pemain akan memberikan jawaban masing-masing dan mendapatkan total skor berdasarkan operator yang digunakan. Jika tidak bisa diselesaikan, keempat kartu dikembalikan ke deck dengan urutan

acak. Jika pemain memberikan ekspresi yang salah, akan diberikan nilai -10. Permainan diulang sampai dengan dek habis, dan tim pemenang adalah tim dengan skor tertinggi. Aplikasi pemain tidak diperbolehkan membuat cache solusi dari kombinasi supaya lebih cepat, karena akan dibandingkan waktu eksekusi dari implementasi strategi greedy.

Spesifikasi pada tugas besar ini adalah sebagai berikut:

1. Terdapat satu backend engine, dan dua front end:
  - a. Backend menghasilkan ekspresi solusi berdasarkan masukan 4 angka,
  - b. Front-end pertama berupa GUI yang mendemokan proses pengambilan 4 kartu untuk memberikan input, dan menampilkan hasilnya. Visualisasi kartu untuk demo boleh menggunakan library.
  - c. Front-end kedua membaca file masukan, memproses 4 angka dari file masukan, dan menghasilkan file keluaran. Front-end kedua akan berinteraksi dengan lingkungan permainan untuk kompetisi antar tim.
2. Terdapat satu lingkungan permainan yang akan disiapkan oleh **asisten** untuk kompetisi, yang akan memanggil sejumlah program 24game solver dari tim pemain yang berbeda. Lingkungan permainan ini memiliki engine yang mengatur pemilihan 4 kartu secara acak, memberi masukan ke setiap program pemain, menerima solusi dari setiap program pemain, dan menghitung skor setiap pemain. Proses ini diulangi sampai dek 52 kartu habis. Aturan tambahan permainan adalah:
  - a. Pemain dengan skor tertinggi akan menang.
  - b. Pemain yang tidak menggunakan strategi greedy akan didiskualifikasi.
  - c. Pemain yang melakukan kecurangan juga akan didiskualifikasi.
  - d. Setiap kelompok hanya memiliki satu pemain.
  - e. Pemenang akan dikompetisikan kembali dengan tim pemenang lainnya, sampai keluar 3 tim pemenang dari tim semua kelas (K1-K3).
3. Environment akan memanggil keempat program dengan argument nama file input dan nama file output. Contoh:

```
"python batchXX_kelompokYY.py AA BB "
```

XX dan YY adalah nomor yang akan diberikan kepada setiap kelompok pada saat pengisian sheet kelompok. AA adalah nama file (termasuk ekstensi file) yang harus dibaca program, dan BB adalah nama file untuk menuliskan ekspresi yang dihasilkan. Perhatikan bahwa nama file berupa variabel, bukan nama file sesungguhnya. Jangan melakukan hardcode untuk melakukan operasi pada file "AA" dan "BB".

- a. File input terdiri dari 1 baris berisi 4 buah integer yang dipisahkan oleh whitespace. Integer memiliki domain [1..13] inklusif.
  - b. File output berupa 1 baris ekspresi matematika yang diminta sesuai spek.
4. Strategi *greedy* yang diimplementasikan tiap kelompok harus mempertimbangkan fungsi objektifnya yaitu berusaha memaksimalkan skor. Strategi ini harus dituliskan secara eksplisit pada laporan, karena akan diperiksa pada saat demo apakah strategi yang dituliskan sesuai dengan yang diimplementasikan. Tiap kelompok dapat menggunakan kreativitas mereka dalam menyusun strategi *greedy* untuk memenangkan permainan.

5. Implementasi pemain 24game solver harus dapat dijalankan pada lingkungan permainan yang telah disediakan oleh asisten.

Spesifikasi teknis lainnya:

1. Tugas dikerjakan per kelompok dengan jumlah anggota minimal 2 orang dan maksimal 3 orang. Anggota kelompok diperbolehkan lintas kelas.
2. Program dibuat dalam Bahasa **Python 3**.
3. Program harus mengandung komentar yang jelas, dan untuk setiap strategi Greedy yang disebutkan, harus dilengkapi dengan kode sumber yang dibuat.
4. Beri nama pemain anda tersebut dengan nama-nama yang menarik dan mudah diingat.
5. Mahasiswa dilarang menggunakan kode program lain yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, tetapi belajar dari program yang sudah ada tidak dilarang.
6. Pengumpulan tugas besar adalah hari **Rabu, 13 Februari 2019** sebelum pukul **13.00 WIB**. Keterlambatan akan mengurangi nilai.
7. Kode program yang dibuat disimpan pada folder StrAlgo1-xxxxx. Lima digit terakhir adalah NIM anggota terkecil.
8. **Bonus** (nilai maksimal 10): Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke YouTube. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video YouTube tersebut dan memutarinya di depan asisten.
9. Semua pertanyaan menyangkut tugas ini harus dikomunikasikan melalui milis agar dapat dicermati oleh semua peserta kuliah IF2211.
10. Demo program akan dilaksanakan pada tanggal **18–22 Februari 2019**, kecuali ada pemberitahuan lebih lanjut dari asisten. Peserta mengisi jadwal demo yang disediakan.
11. Tiap anggota harus memahami proses pembuatan program, karena akan ada pertanyaan-pertanyaan yang harus dijawab per individu.
12. Pada saat demo, asisten akan memanggil kelompok yang diminta demo. 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” ganessa.
- **Bab 1:** Deskripsi tugas (dapat menyalin spesifikasi tugas ini).
- **Bab 2:** Dasar teori (algoritma *greedy*), persoalan 24 game.
- **Bab 3:** Pemanfaatan strategi *greedy*. Langkah-langkah strategi *greedy* yang dipilih oleh kelompok, dikaitkan dengan proses memperoleh nilai semaksimal mungkin, dituliskan pada bab ini. Usulan strategi *greedy* diletakkan di sini, beserta contoh ilustrasi (jika ada), termasuk juga di dalamnya struktur data, dan spesifikasi program.

- **Bab 4:** Implementasi dan pengujian, berisi implementasi program dalam *game engine* yang digunakan (*pseudocode* yang cukup detil dengan komentar untuk pembaca kode agar mudah dipahami), misalnya menampilkan nama file program, dll. Di dalam pengujian dituliskan hasil pengujian, lalu dianalisis hasil pengujian yang dilakukan, misal apakah strategi *greedy* berhasil mendapatkan nilai optimal, lalu jika tidak, dalam kondisi seperti apa strategi *greedy* tidak berhasil mendapatkan nilai optimal, dll.
- **Bab 5:** Kesimpulan dan saran.
- **Daftar Pustaka.**

#### **Keterangan laporan:**

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Laporan tidak perlu memakai *cover* mika atau dijilid. Cukup dibuat agar laporan tidak akan tercecer bila dibaca.
3. Laporan boleh menggunakan kertas rius, 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. Kesesuaian strategi Greedy yang dituliskan dengan implementasi dan saat demo (50%)
2. Demo – pemahaman Anda dalam pembuatan program (30%)
3. Laporan (20%)
4. Beberapa kelompok pemenang (akan ditentukan jumlahnya nanti) pada saat kompetisi akan mendapatkan bonus nilai 5 (selain bonus dari membuat video kelompok).

*-selamat mengerjakan-*

