

**Laporan**  
**Tugas Besar IF2211 Strategi Algoritma**  
**Penyelesaian 24 Game dengan Algoritma Greedy**



oleh :

Raihan Lutfhi Haryawan / 13517016  
Haris Salman Al-Ghifary/ 13517052

PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2019

## I. DESKRIPSI PERSOALAN

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 setiap ekspresi adalah total skor dari operator dikurangi jumlah pasang kurung dan selisih dengan 24.

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.

## II. DASAR TEORI

Algoritma greedy merupakan metode penyelesaian yang paling populer untuk memecahkan persoalan optimasi. Greedy sendiri diambil dari Bahasa Inggris yang artinya rakus, tamak atau serakah. Banyak orang berkata algoritma greedy itu seperti istilah “Ambilah apa yang bisa kamu dapatkan sekarang!”. Algoritma greedy sendiri membentuk solusi step by step yang artinya adalah harus ada tahap-tahap dalam penyelesaiannya. Terdapat banyak pilihan solusi, oleh karena itu harus dibuat langkah yang terbaik untuk mendapat keputusan terbaik (good move). Setiap keputusan yang telah diambil tidak dapat diubah lagi atau di backtracking pada langkah-langkah berikutnya.

Dalam penyelesaian persoalan optimasi pada algoritma greedy dibagi menjadi dua macam yaitu maksimasi dan minimasi. Pada persoalan kasus 24 game solver ini, kami menggunakan algoritma greedy untuk mencari kemungkinan terbaik dari penyusunan 4 kartu untuk dapat mencapai nilai 24. Kami menggunakan pendekatan dengan cara sorting seluruh masukan dari yang terbesar ke yang terkecil. Kemudian melakukan pengecekan (fungsi seleksi) kemudian dilakukan dengan uji kelayakan dan dilanjutkan dengan fungsi obyektif yaitu mencari optimasi nilai atau hampiran dari nilai 24.

### III. PEMANFAATAN STRATEGI GREEDY

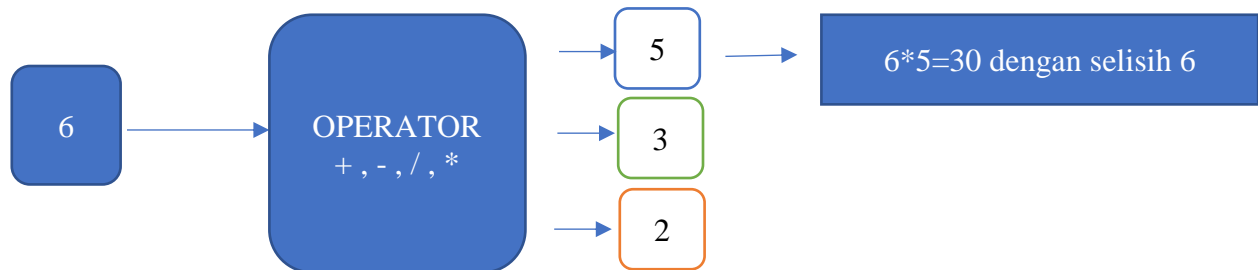
Pada persoalan 24 game solver ini, tim kami menggunakan strategi greedy yang bertujuan untuk mendapat nilai 24 atau yang paling mendekati nilai 24. Ide yang tim kami temukan adalah dengan mensorting semua nilai yaitu 4 kartu yang menjadi inputan. Kemudian untuk langkah selanjutnya adalah dengan mengambil nilai terbesar yang ada pada keempat masukan yang ada, setelah itu kita membandingkan nilai terbesar dengan ketiga masukan yang lain dengan semua operator yang telah kami deklarasikan. Dari perbandingan itu kita cari hasil selisih yang paling dekat dengan 24, jika selisih sama kami menggunakan masukan yang nilai nya lebih besar. Setelah mendapat nilai yang baru kemudian ulangi seperti langkah sebelumnya, bandingkan nilai hasil yang baru dengan dua masukan setelahnya. Proses itu dilakukan sampai habis, maka nilai yang diinginkan telah didapatkan. Kami menyadari strategi yang kami gunakan masih kurang mangkus mengingat kita tidak dapat melakukan backtracking.

5	3	6	2
---	---	---	---

Pada ilustrasi tersebut terdapat 4 masukan awal yaitu 5,3,6,2. Dari keempat masukan tersebut kita sorting dari yang terbesar ke terkecil. Maka hasilnya sebagai berikut :

6	5	3	2
---	---	---	---

Kemudian setelah kita dapat data yang terurut, pilih angka terbesar yaitu 6.



Kita lakukan pengecekan dengan angka selanjutnya dan memberikan hasil selisih paling dekat yaitu 6. Karena selisih yang paling dekat dengan 24 adalah operator \* yaitu  $6*5$  dengan selisish 6. Maka dihasilkan keputusan terbaik  $6*5$ , setelah itu lakukan dengan cara yang sama sampai habis maka akan didapat hasil  $6*5-3-2 = 25$ . Hasil tersebut mendekati 24, meskipun dengan cara bruteforce dapat dihasilkan hasil 24 akan tetapi strategi greedy kami tidak mendapatkannya.

Struktur data yang ada pada program ini adalah menggunakan list yang mana untuk handle masukan dan keluaran. Spesifikasi program ini adalah dapat merandom 4 kartu dari deck, akan tetapi tidak dapat untuk dimainkan lagi sampai habis hanya sekali turn saja. Program juga dapat mencari solusi optimasi dari 4 operan masukan dengan 3 operator dengan strategi algoritma greedy. Terdapat 3 program yang kami buat dalam penyelesaian 24 game solver ini.

#### IV. IMPLEMENTASI DAN PENGUJIAN

##### Pseudo Code

Program backend;

//Fungsi untuk membuat list solusi angka dan operator

Function solve ( listnb, listop )

    //Sorting list input mulai dari yang terbesar ke terkecil

    nb <= sorted(listNb, reverse=True)

    op <= solusiOp(nb, listOp)

    solusi <= [str(nb[0]),op[0],str(nb[1]),op[1],str(nb[2]),op[2],str(nb[3])]

=> solusi

//Fungsi untuk membuat list operator yang dipilih melalui algoritma greedy

function solusiOp(sortedListNb, listOp)

    //Inisialisasi list solusi operator

    op <= []

    //Algoritma greedy untuk memasukkan solusi ke dalam list

    x <= sortedListNb[0]

    for i in sortedListNb[1:]

        closestNb <= 999

        for j in listOp

            Nb <= eval(str(x) + j + str(i))

            diff <= abs(Nb - 24)

            if (diff <= abs(closestNb - 24)):

                closestNb <= Nb

                selectedOp <= j

    x <= closestNb

    op.append(selectedOp)

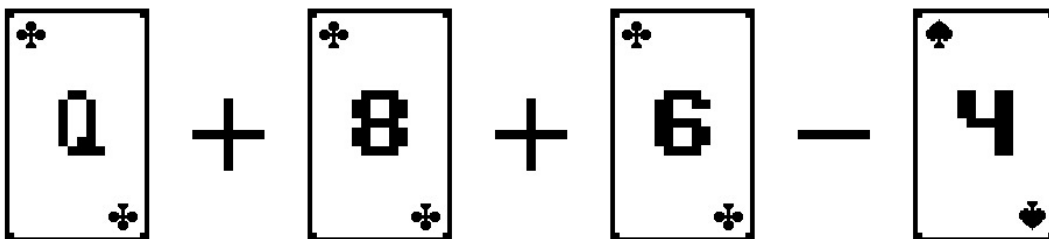
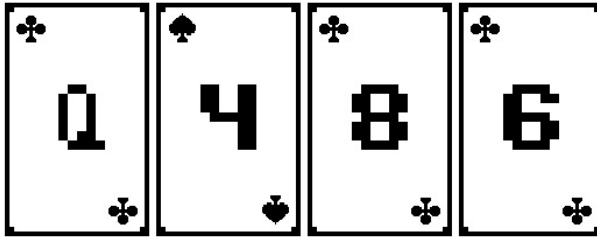
=> op

Contoh Hasil program yang telah dijalankan :

Solusi tidak ditemukan

$$12+8+6-4 = 22$$

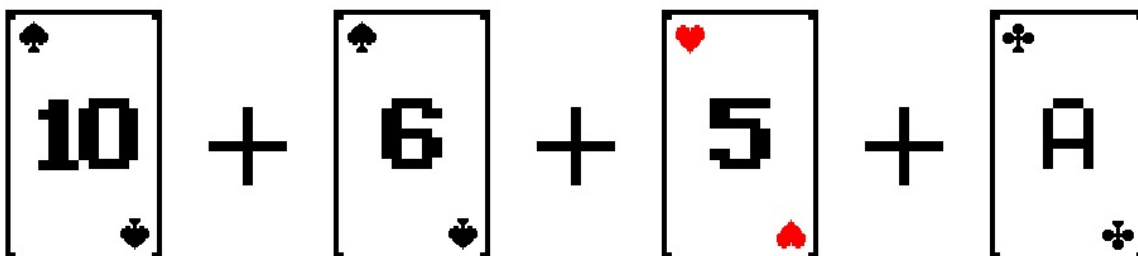
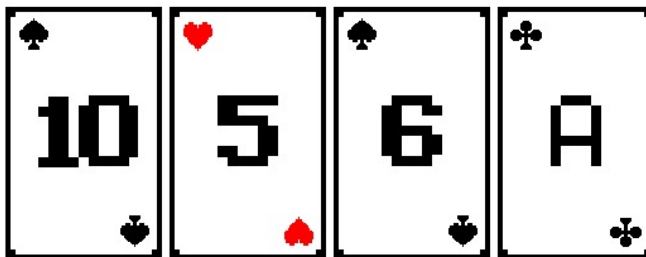
---



Solusi tidak ditemukan

$$10+6+5+1 = 22$$

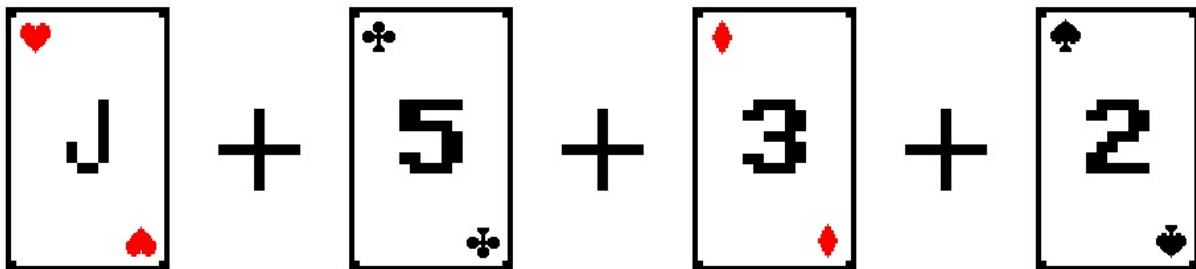
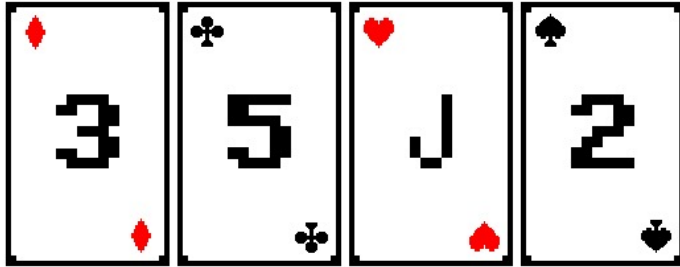
---



Solusi tidak ditemukan

$$11+5+3+2 = 21$$

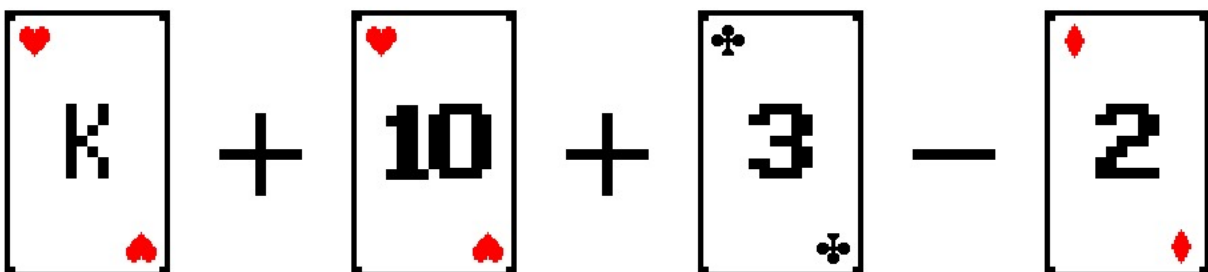
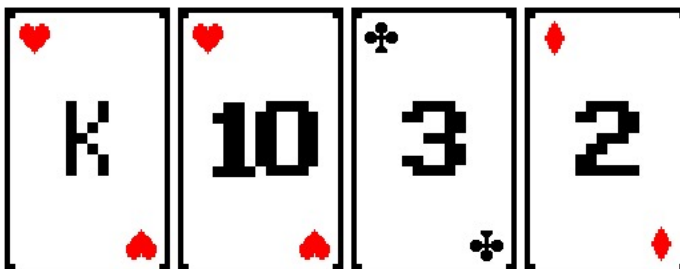
---



Solusi ditemukan

$$13+10+3-2 = 24$$

---





Solusi Kami dapat optimal apabila dalam pengurutan list dapat mencapai 24, akan tetapi jika dalam pengurutan hasilnya tidak bisa 24 maka solusi yang kami punya tidak optimum. Kita ambil contoh yang ketiga yaitu inputan 3 5 11 2, mula-mula kita urutkan secara descending maka hasilnya adalah 11 5 3 2. Algoritma kami akan menghasilkan  $11+5+3+2$  padahal dengan algoritma bruteforce dapat dihasilkan 24 yaitu  $5*3+11-2$ . Algoritma kami tidak dapat menghasilkan hasil 24 karena menurut kami jika kami mengecek semua kemungkinan yang ada maka algoritma tersebut tidak greedy murni yang mana harus dilakukan step by step.

## **V. KESIMPULAN DAN SARAN**

### **5.1 SIMPULAN**

1. Dengan adanya tugas besar strategi algoritma yang mana membuat penyelesaian 24 game solver dengan algoritma greedy dapat membuat mahasiswa lebih paham dengan algoritma greedy.
2. Melalui program ini mahasiswa dapat mendapat ilmu lebih yaitu penguasaan bahasa pemrograman python ditambah dengan GUI yang ada.

### **5.2 SARAN**

1. Untuk kedepannya dapat ditambah variasi yang lebih seru
2. Kepada asisten, dimohon untuk dapat membantu mahasiswa IF yang bukan angkatan 2017 untuk dapat menemukan kelompoknya. Karena mungkin saja mereka canggung dan sungkan terima kasih.

## DAFTAR PUSTAKA

<https://www.w3schools.com/python/default.asp>

<https://www.geeksforgeeks.org/python-gui-tkinter/>