**LAPORAN STUDI KASUS**

**INFORMATIKA**

**ANALISIS ALGORITMA DAN LOGIKA**

**PADA PERMAINAN CAPSA ATAU BIG TWO**



Dosen Pengampu:

Merryana Lestari, S.Kom., M.Kom

Disusun oleh :

**Varel Gabriel Wungow** – 32230062

Nathan Raffael Simatupang - 32230078

**UNIVERSITAS BUNDA MULIA**

**SERPONG**

**2024**

**PENDAHULUAN**

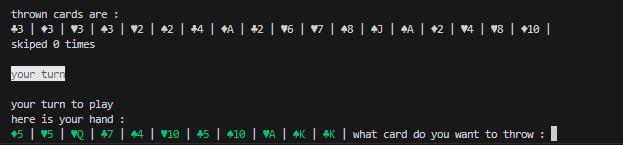
Dalam laporan ini, kelompok kami akan mempersembahkan hasil dari studi kasus tentang penerapan algoritma *Greedy* sebagai strategi untuk memainkan game Capsa atau *Big two.* Dalam laporan ini, kelompok kita memberikan penjelasan singkat tentang cara kerja kodingan yang kami buat, bagaimana cara untuk menjalankan program, dan bagaimana cara user dapat ikut serta bermain permainan capsa menggunakan kodingan yang kelompok kita berikan.

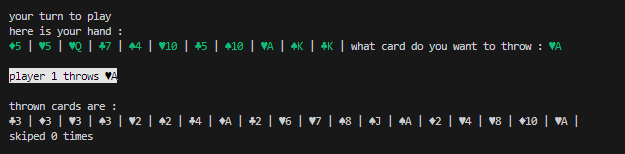
**BAGAIMANA KODINGAN DIBUAT**

Kodingan yang kelompok kita buat pada dasarnya adalah prototipe dari game capsa, alasan mengapa kodingan tersebut hanyalah prototipe dari game capsa karena ada satu aturan main yang tidak kami implementasikan kedalam kode yaitu penggunaan kombinasi kartu atau *Hand Rankings* seperti : *pairs, threes, straight, flush, full house, four of a kind, straight flush* dan *royal flush.* Alasan mengapa penggunaan kombinasi kartu tidak diimplementasikan kedalam kodingan ada dua faktor yaitu, pertama, kodingan akan menjadi makin rumit dan waktu pengerjaan project tidaklah cukup untuk membuat permainan capsa yang utuh. Faktor kedua adalah, untuk mengimplementasikan dan menganalisis algoritma *Greedy* sebagai strategi bermain permainan kartu capsa tidaklah perlu permainan capsa yang utuh, dan prototipe yang kolompok kami buat sudahlah cukup untuk menganalisis jalannya algoritma yang ingin dianalisa.

**PERAN PLAYER 1-4 DAN PENERAPAN ALGORITMA *GREEDY* DALAM KODINGAN.**

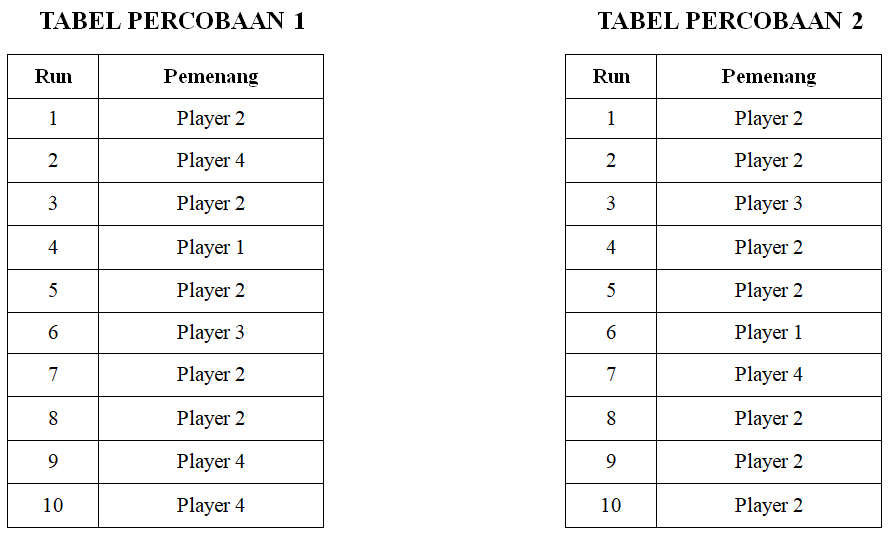
Lalu bagaimana algoritma *greedy* diimplementasikan kedalam program dan apa peran beserta dampaknya seiring berjalannya program? Untuk mengetahui itu, maka harus diketahui terlebih dahulu tentang cara kerja para pemain dalam program. Dalam program prototipe permainan capsa yang kami buat, ada empat pemain yang bergilir untuk membuang kartu. Peran algoritma *greedy* di situ adalah untuk menentukan kartu yang akan dibuang oleh “player 2”. Sedangkan untuk “player 3” dan “player 4”, cara mereka menentukan kartu apa yang dibuang adalah dengan hanya mengambil sembarang kartu dari tangan mereka (selama kartu yang ditentukan dapat mengalahkan kartu yang saat itu ada di meja). Untuk “player 1” cara dia menentukan kartu apa yang akan dibuang itu sedikit unik, karena yang menentukan kartu apa yang akan dikeluarkan “player 1” adalah user, dengan kata lain, user menggunakan inputan untuk menentukan kartu apa yang ingin dikeluarkan.

gambar contoh: terminal meminta inputan user untuk membuang kartu apa agar mengalahkan *diamond 10*

 gambar contoh: user menginput *heartens ace* maka kartu yang dimeja(thrown cards) ditambahkan *heartens ace*

**HASIL EKSPERIMEN DAN ANALISIS EFEKTIFITAS ALGORITMA *GREEDY***

Dibagian ini akan membahas hasil running program dan melihat seberapa efektif jika algoritma *greedy* dijadikan sebagai strategi untuk memainkan permainan capsa. Perlu diingatkan lagi bahwa algoritma *greedy* memang bisa dipakai untuk strategi bermain kartu capsa, tetapi karena sifat permainan yang agak bertolak belakang dengan cara *greedy* menyelesaikan masalah, maka algoritma *greedy* bukanlah strategi paling optimal untuk memainkan permainan kartu capsa.



Catatan: di test running pada tabel percobaan satu dan dua, player 1 bukanlah inputan user, melainkan disengaja menggunakan cara pengambilan kartu seperti player 3 dan 4 yaitu mengambil secara acak. Ini dilakukan untuk melihat perbandingan yang jelas se-efektif apa algoritma *greedy* yang diterapkan untuk menghadapi lawan yang seharusnya lebih inferior darinya.

Bisa dilihat dari tabel percobaan satu dan dua bahwa pemain yang menggunakan *greedy* (Player 2) memiliki kesempatan menang lebih tinggi bahkan bisa mencapai 70% tingkat kesuksesan melawan pemain-pemain yang hanya membuang kartu secara asal dan sembarang. Hasil ini cukup sesuai dengan prediksi kelompok kita yang awalnya beranggapan bahwa strategi algoritma *greedy* memang tidak akan menang setiap saat, tapi tentunya memiliki probabilitas menang yang lebih tinggi. Tapi kelompok kita hanya memperkirakan *outcome* tersebut akan terjadi jika player 2 hanya melawan player-player yang mengeluarkan kartu secara sembarang, jika player 2 yang menggunakan algoritma *greedy* dihadapi dengan algoritma yang dapat mengkalkulasikan strategi sampai beberapa langkah kedepan, maka kemungkinan besar player 2 akan kalah lebih banyak dibandingkan player dengan strategi yang lebih matang tersebut.