

Penerapan Algoritma *Greedy* dalam Permainan Kartu Capsa Banting (*Big Two*)

Hilda Carissa Widelia - 13520164
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13520164@std.stei.itb.ac.id

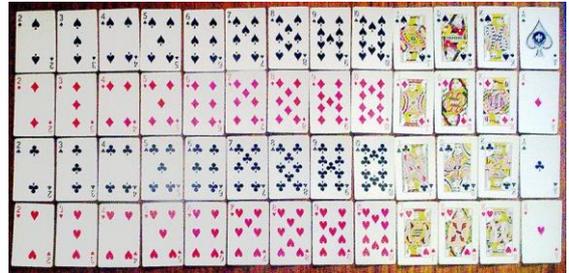
Abstrak—Kartu permainan (*playing cards*) atau yang umum disebut kartu remi adalah sekumpulan kartu yang biasa digunakan untuk bermain bersama dengan teman dan keluarga. Terdapat banyak sekali permainan yang dapat dimainkan dengan menggunakan kartu remi. Salah satu dari permainan itu adalah Capsa Banting. Dalam permainan Capsa Banting, pemenang adalah pemain yang berhasil mengeluarkan seluruh kartu di tangannya pertama kali. Untuk menjadi pemenang, dibutuhkan strategi saat permainan berlangsung. Oleh karena itu, dalam makalah ini akan dibahas mengenai penerapan strategi/algoritma *Greedy* dalam permainan kartu Capsa Banting ini.

Kata kunci—*Greedy, Capsa Banting, Kartu, Strategi, Algoritma*

I. PENDAHULUAN

Kartu permainan (*playing cards*) atau yang biasa disebut kartu remi merupakan sekumpulan kartu seukuran tangan yang digunakan untuk permainan kartu. Selain permainan, kartu ini juga seringkali digunakan untuk sulap, permainan papan, untuk membuat rumah kartu, dan hal-hal lainnya. Permainan yang dapat dimainkan dengan menggunakan kartu remi sendiri ada banyak. Beberapa permainan yang cukup populer di Indonesia contohnya seperti capsa banting, 41, cangkulan, hantu-hantuan, dan masih banyak lainnya. Sedangkan permainan yang populer di banyak negara lainnya ada seperti *poker*, *blackjack*, *bridge*, *solitaire* dan masih banyak lagi.

Kartu remi sendiri dibagi menjadi 4 jenis dengan masing-masing jenis terdapat 13 kartu berbeda. Empat jenis kartu yang ada adalah sekop (*Spade*), keriting (*Club*), wajik (*Diamond*), dan hati (*Heart*). Masing-masing jenis terdiri dari kartu dengan angka 2 hingga 10, lalu ada J (*Jack*), Q (*Queen*), K (*King*), yang pada beberapa permainan dianggap sebagai 11 dan di beberapa permainan lain K dianggap lebih tinggi dari Q dan Q lebih tinggi dari J. Terakhir ada A atau As yang pada beberapa permainan dianggap sebagai 1 dan di beberapa permainan lain dianggap sebagai kartu tertinggi, melebihi K.



Gambar 1.1 Kartu remi. Sumber : https://id.wikipedia.org/wiki/Kartu_remi

Pada permainan kartu remi, jumlah pemain yang bisa bermain berbeda-beda tergantung jenis permainannya. Permainan *solitaire* dapat dimainkan oleh 1 orang, sedangkan permainan seperti cangkulan, 41, dan capsa banting, membutuhkan minimal 2 orang pemain. Untuk capsa banting sendiri jumlah pemain maksimal adalah sebanyak 4 orang. Untuk bermain permainan kartu, dibutuhkan strategi dan pemikiran yang baik untuk dapat memenangkan permainan sehingga permainan kartu ini cukup populer. Tentu saja, selain strategi dan pemikiran, dibutuhkan juga faktor keberuntungan. Beberapa permainan bahkan tidak memerlukan strategi dan hanya membutuhkan keberuntungan seperti pada permainan *poker*.

Untuk bermain capsa banting, ke-52 kartu yang ada akan dibagikan kepada semua pemain hingga habis. Jumlah pemain optimal adalah 4, dan memang biasanya dimainkan bersama oleh 4 orang, dan oleh karena itu masing-masing pemain akan mendapat 13 kartu. Untuk memenangkan permainan ini, pemain harus mengeluarkan kartunya, dan orang pertama yang berhasil mengeluarkan ke-13 kartu mereka akan memenangkan permainan. Tentu saja dibutuhkan strategi agar dapat memenangkan permainan ini secepatnya. Oleh karena itu untuk penyusunan strategi digunakan algoritma *Greedy*.

Algoritma *Greedy* sendiri adalah salah satu metode yang paling populer dan sederhana untuk memecahkan permasalahan optimasi. Pada intinya, algoritma ini akan memecahkan persoalan secara langkah per langkah dengan mengambil pilihan terbaik yang dapat diperoleh pada setiap langkahnya. Dalam makalah ini, akan dibahas mengenai penerapan algoritma *Greedy* dalam permainan Capsa Banting agar pemain dapat menyusun ke-13 kartunya dengan menggunakan strategi *Greedy* dan kemudian memenangkan permainan setelah berhasil mengeluarkan semua kartunya.

II. TEORI DASAR

A. Algoritma Greedy

Algoritma *Greedy* adalah algoritma yang memecahkan persoalan secara langkah per langkah dengan memilih pilihan terbaik dalam setiap langkahnya dan mengharapkan bahwa rangkaian pilihan yang diambil dari setiap langkah akan menghasilkan optimum global atau penyelesaian permasalahan yang paling efektif. Algoritma *Greedy* sendiri dapat diaplikasikan pada persoalan-persoalan optimasi yaitu maksimasi dan minimasi. Beberapa contoh persoalan yang dapat diselesaikan dengan Algoritma *Greedy* adalah :

1. Persoalan penukaran uang (*coin exchange problem*)
2. Persoalan memilih aktivitas (*activity selection problem*)
3. Minimisasi waktu dalam sistem
4. Persoalan *Knapsack* (*Knapsack problem*)
5. Penjadwalan pekerjaan dengan tenggat waktu (*Job scheduling with deadlines*)
6. Pohon merentang minimum (*minimum spanning trees*)
7. Lintasan terpendek (*shortest path*)
8. Kode Huffman (*Huffman code*)
9. Pecahan Mesir (*Egyptian fraction*)

Dan masih banyak lainnya.

Dalam penyelesaian permasalahan dengan menggunakan Algoritma *Greedy*, terdapat beberapa elemen algoritma *greedy*, yaitu :

1. Himpunan kandidat, C : kandidat-kandidat yang dipilih pada setiap langkahnya. Contohnya ada *job*, koin, benda, simpul dalam graf, dan lain-lain.
2. Himpunan solusi, S : kandidat yang sudah dipilih
3. Fungsi solusi : memastikan bahwa kandidat yang dipilih sudah memberikan solusi
4. Fungsi seleksi (*selection function*) : memilih kandidat berdasarkan strategi *greedy* tertentu, biasanya bersifat heuristik.
5. Fungsi kelayakan (*feasible*) : memeriksa apakah kandidat yang sudah dipilih dapat dimasukkan ke himpunan solusi, apakah melanggar *constraint* tertentu atau tidak.
6. Fungsi obyektif : memaksimalkan atau meminimumkan.

Berikut adalah skema umum dari algoritma *Greedy* :

```
function greedy(C: himpunan_kandidat) → himpunan_solusi
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy }
Deklarasi
x : kandidat
S : himpunan_solusi

Algoritma:
S ← {} { inialisasi S dengan kosong }
while (not SOLUSI(S) and (C ≠ {})) do
x ← SELEKSI(C) { pilih sebuah kandidat dari C }
C ← C - {x} { buang x dari C karena sudah dipilih }
if LAYAK(S ∪ {x}) then { x memenuhi kelayakan untuk dimasukkan ke dalam himpunan solusi }
S ← S ∪ {x} { masukkan x ke dalam himpunan solusi }
endif
endwhile
{ SOLUSI(S) or C = {} }

if SOLUSI(S) then { solusi sudah lengkap }
return S
else
write("tidak ada solusi")
endif
```

Gambar 2.1 Skema umum Algoritma *Greedy*. Sumber : [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf)

Pada intinya, dalam algoritma *Greedy* yang akan dilakukan adalah :

1. Inialisasi S (himpunan solusi) kosong
2. Pilih sebuah kandidat dari C (himpunan kandidat)
3. Buang kandidat terpilih dari C
4. Cek apakah kandidat terpilih memenuhi kelayakan untuk masuk kedalam himpunan solusi
5. Jika layak, maka masukkan kandidat terpilih ke himpunan solusi.
6. Jika isi dari himpunan solusi belum memberikan solusi lengkap dan himpunan kandidat C belum kosong, ulangi lagi dari langkah kedua.
7. Jika solusi yang dimiliki pada himpunan solusi sudah lengkap, berarti solusi sudah didapatkan.

Pada setiap akhir iterasi, solusi yang didapat adalah optimum lokal. Pada akhir iterasi, diperoleh optimum global. Akan tetapi, pada algoritma ini, optimum global yang dihasilkan belum tentu merupakan hasil optimal (hasil terbaik) karena algoritma *Greedy* tidak mengecek semua kemungkinan yang ada seperti pada *exhaustive search*. Selain itu, terdapat beberapa fungsi seleksi berbeda sehingga untuk menghasilkan solusi optimal kita harus memilih fungsi yang tepat.

Oleh karena itu, solusi dengan menggunakan algoritma *Greedy* biasanya menghasilkan solusi hampiran. Maka untuk permasalahan yang tidak perlu menemukan solusi terbaik mutlak, dapat digunakan algoritma *Greedy* daripada algoritma lain yang waktunya lebih panjang untuk kebutuhan menghasilkan solusi eksak.

B. Kartu Remi (Playing Cards)

Kartu remi adalah sekumpulan kartu yang diberi angka atau gambaran (atau keduanya) yang digunakan untuk memainkan sebuah permainan, untuk edukasi, untuk sulap, dan banyak hal lainnya. Biasanya kartu remi berbentuk persegi panjang, dengan ukuran sama pada setiap kartunya, Ukurannya beragam, namun rata-rata ukurannya cukup untuk sebuah tangan memegang beberapa kartu sekaligus. Salah

satu sisi dari kartu akan berisi angka / gambar yang berbeda sedangkan sisi lainnya dibiarkan kosong atau berisi gambaran pola yang sama untuk semua kartu.

Permainan kartu remi biasanya memanfaatkan fakta bahwa setiap pemain hanya dapat mengidentifikasi kartu yang ia miliki sedangkan kartu lawannya tidak. Tentu saja, terdapat juga beberapa permainan yang tidak menggunakan fakta ini seperti pada permainan gebrakan.

Kartu remi sendiri terdiri dari 4 jenis kartu. Pertama ada sekop (*spade*) ♠, lalu ada kriting (*club*) ♣, selanjutnya ada hati (*heart*) ♥, dan terakhir ada wajik (*diamond*) ♦. Untuk masing-masing jenis kartu, terdapat 13 kartu berbeda. Ada kartu dengan angka biasa yaitu 2, 3, 4, 5, 6, 7, 8, 9, dan 10, dan ada juga kartu J (*Jack*), Q (*Queen*), K (*King*) dan A / As. Pada sebagian besar permainan, As merupakan kartu tertinggi, diikuti dengan King, Queen, Jack, 10, 9, 8, hingga 2 sebagai kartu terkecil.

C. Capsa Banting

Permainan kartu capsa banting diperkirakan berasal dari negara China, tepatnya mulai muncul pada tahun 1980-an. Permainan ini kemudian menjadi sangat populer di Shanghai, Hongkong, Taiwan, juga di Filipina, Singapur dan tentu saja, Indonesia. Permainan ini juga sudah tersebar ke negara-negara barat dan memiliki beberapa nama alternatif.

Permainan ini optimalnya dimainkan oleh 4 pemain, namun masih mungkin dimainkan oleh 3 dan 2 pemain dengan sedikit perubahan pada aturannya. Kartu yang digunakan adalah deck kartu standar dengan 52 kartu, dengan urutan kartu dari tinggi ke rendah adalah 2-A-K-Q-J-10-9-8-7-6-5-4-3. Jenis kartu juga mempengaruhi rankingnya, dengan urutan tinggi ke rendah adalah sekop-hati-keriting-wajik.

Objektif permainan ini adalah untuk mengeluarkan ke-13 kartu yang ada di tangan. Pemenang adalah yang tercepat mengeluarkan semua kartunya. Kartu dapat dikeluarkan satu per satu, atau dengan kombinasi tertentu. Jika tidak dapat menjadi yang pertama mengeluarkan kesemua kartu, maka minimalisasi kartu yang dipunya sampai salah satu pemain lain berhasil mengeluarkan kesemua kartunya.

Terdapat 4 tipe kombinasi kartu yang dapat dikeluarkan. Pertama ada *single*, yaitu pemain mengeluarkan kartu yang ada di tangannya satu per satu. Setiap giliran selanjutnya, kartu yang dikeluarkan harus memiliki ranking lebih besar. Ranking ditentukan pertama berdasarkan angkanya, dengan 2 sebagai paling tinggi lalu King, Queen, hingga 3. Jika pemain mau mengeluarkan angka yang sama, mana harus diperhatikan jenis kartunya. Jenis kartu juga memiliki rankingnya dengan sekop menjadi ranking tertinggi, dilanjutkan hati, keriting, dan paling rendah adalah wajik.

Selanjutnya adalah *double / pair* yaitu pasangan kartu yang memiliki angka yang sama, dengan sama seperti pada *single*, 2 menjadi kartu tertinggi dan 3 menjadi kartu terendah. Kartu dengan angka lebih tinggi akan mengalahkan kartu dengan angka lebih rendah. Jika kartu memiliki angka yang sama, maka kartu yang memiliki jenis lebih tinggi akan menang. Misalnya pemain A mengeluarkan 3♠ dan 3♦, selanjutnya pemain B ingin mengeluarkan 3♣ dan 3♥, maka pemain B tidak dapat mengeluarkan kartunya, karena pemain B harus

mengeluarkan kartu yang lebih tinggi dari kartu pemain A sedangkan kartu pemain A memiliki jenis sekop dan jenis sekop merupakan jenis tertinggi. Pemain B akan dapat mengeluarkan kartunya jika kartunya berupa 4♣ dan 4♦ karena 4 dapat mengalahkan 3.

Ada juga kombinasi 3 kartu dengan angka yang sama biasa disebut *Triples*, dan sama juga seperti sebelumnya, kartu tertinggi adalah 2 dan terendah adalah 3. Kombinasi terakhir adalah kombinasi 5 kartu. Untuk kombinasi 5 kartu terdapat 5 jenis kombinasi. Berikut kombinasi 5 kartu diurutkan dari ranking paling rendah :

1) *Straight*

Straight berisi kombinasi 5 kartu yang berurutan. Misalnya 3-4-5-6-7. Untuk kartu dengan angka yang sama, maka jenis kartu pada angka terbesar akan menentukan yang mana yang lebih tinggi. Pada kombinasi ini, jika pemain menggunakan kartu dengan angka 2, maka kartu tersebut hanya dapat digunakan sebagai kartu kecil, maksudnya kartu 2 hanya dapat digunakan dalam kombinasi 2-3-4-5-6, bukan J-Q-K-A-2. Oleh karena itu, kombinasi *straight* tertinggi adalah 10-J-Q-K-A. As dapat juga digunakan sebagai kartu kecil dalam kombinasi A-2-3-4-5 yang kemudian dapat dikalahkan oleh 2-3-4-5-6 yang dapat dikalahkan oleh 3-4-5-6-7.

2) *Flush*

Flush berisi kombinasi 5 kartu yang berjenis sama. Kombinasi *flush* yang memiliki jenis yang lebih tinggi akan mengalahkan yang lebih rendah dengan aturan jenis seperti yang sudah disebutkan sebelumnya. Untuk 2 kombinasi kartu dengan jenis yang sama, kombinasi lebih tinggi adalah yang memiliki angka lebih tinggi. Contohnya untuk kombinasi 3♥-7♥-9♥-4♥-8♥ akan mengalahkan kombinasi 2♦-K♦-10♦-7♦-4♦ yang mengalahkan kombinasi A♦-Q♦-J♦-9♦-5♦.

3) *Full House*

Full House berisi kombinasi 5 kartu dengan 3 kartu berjenis sama, dan 2 kartu berjenis sama. Bisa dibilang kartu ini berisi kombinasi dari *Triple* dan *Double*. Pada kombinasi ini, penentuan tinggi rendahnya kombinasi adalah berdasarkan kartu *Triple*. Maka jika terdapat kartu kombinasi 3-3-3-2-2 akan dapat dikalahkan dengan kartu kombinasi 4-4-4-5-5 karena yang dipentingkan hanya jenis kartu dari *triple*-nya.

4) *Four of a kind*

Kombinasi kartu ini terdiri dari 4 kartu berjenis sama dengan kartu kelimanya bebas. Misalnya 4-4-4-4-3. Tentu saja kombinasi ini akan ditentukan tinggi rendahnya berdasarkan jenis dari 4 kartu yang sama.

5) *Straight Flush*

Kombinasi kartu ini terdiri dari 5 kartu yang berurutan mirip seperti kombinasi *Straight*. Akan tetapi, pada *Straight Flush*, jenis kelima kartu harus sama. Contohnya adalah 4♣-5♣-6♣-7♣-8♣. Sama seperti pada *straight*, angka tertinggi dari kombinasi kelima kartu akan menentukan mana kombinasi yang lebih kuat, dan pada kombinasi kartu dengan angka yang

sama, jenis kartulah yang digunakan sebagai penentu kombinasi mana yang lebih kuat. Maka, kombinasi *straight flush* terkuat adalah *Royal Flush* yang terdiri dari 10♠-J♠-Q♠-K♠-A♠.

Kombinasi-kombinasi ini hanya dapat dikalahkan oleh kombinasi dengan jumlah kartu yang sama. Hal ini berarti untuk kombinasi *pair* hanya dapat dikalahkan oleh *pair* lagi, dan kombinasi *triple* hanya dapat dikalahkan oleh *triple* lagi. Akan tetapi untuk kombinasi lima kartu, dapat dikalahkan sesuai dengan urutan diatas. Misalnya terdapat *straight*, maka dapat dikalahkan oleh *flush* dan *flush* dapat dikalahkan oleh *full house*.

Ketika awal dimulai permainan, ke-52 kartu akan dibagikan hingga habis. Kemudian, pemain yang memiliki kartu 3♦ dapat mengeluarkan kartunya terlebih dahulu, dengan harus menggunakan kartu 3♦ tadi. Pada beberapa versi permainan, pemain hanya dapat mengeluarkan satu kartu 3♦ saja. Akan tetapi, pada versi lain, pemain dapat mengeluarkan kartu 3♦ nya di dalam kombinasi, misalnya dengan *pair* yaitu 3♦ dan 3♣ atau bisa juga dalam kombinasi 5 kartu, misalnya kombinasi *straight* yaitu 3♦-4♣-5♣-6♥-7♠.

Setelah pemain pertama mengeluarkan kartunya, pemain selanjutnya dapat mengeluarkan kartunya dengan jumlah kartu yang sesuai dan tentu saja harus dapat mengalahkan pemain sebelumnya. Urutan pemain biasanya searah jarum jam, namun tentu saja dapat bervariasi sesuai pemain.

Seorang pemain tidak diharuskan melawan kartu yang ada di papan, jika pemain tidak ingin mengeluarkan kartunya atau tidak memiliki kartu yang lebih tinggi dari kartu yang terakhir berada di papan, maka pemain dapat memilih untuk "pass". Ketika semua pemain sudah melakukan "pass", maka pemain yang terakhir mengeluarkan kartu, dapat mengeluarkan kartu lagi dengan kombinasi kartu yang berbeda, sesuai dengan kemauan pemain. Jadi misalnya sebelumnya sedang bermain *single*, dan pemain A terakhir mengeluarkan 2♠, yaitu kartu paling tinggi dalam satu deck, maka pemain A dapat mengeluarkan lagi kombinasi kartu lain, misalnya *triple* 7♦-7♥-7♠.

Pemain pertama yang berhasil mengeluarkan semua kartunya akan keluar sebagai pemenangnya. Sedangkan pemain lain akan diberikan penalti sesuai dengan sisa kartu pada tangan, dengan 1 poin per kartu untuk jumlah kartu dibawah 9, 2 poin per kartu untuk jumlah kartu diatas 9 dan dibawah 13, serta 3 poin per kartu untuk jumlah kartu 13 atau berarti pemain tidak pernah mengeluarkan kartunya.

Peraturan permainan capsia banting ini beragam dan pada akhirnya kembali lagi kepada pemain yang sedang bermain. Terdapat aturan-aturan lain yang berbeda, misalnya *four of a kind* dapat mengalahkan 2♠, atau tidak ada kartu kombinasi *triple*, dan lain-lain. Akan tetapi pada makalah ini, aturan yang digunakan adalah yang sudah disebutkan tadi.

III. ANALISIS DAN PEMBAHASAN

Penerapan prinsip *Greedy* dalam permainan ini ada pada peraturan utamanya. Pemain akan menjadi pemenang ketika pemain berhasil mengeluarkan semua kartu di tangannya. Oleh karena itu digunakan algoritma *Greedy by* jumlah kartu, dan *Greedy by* nilai kartu terendah. *Greedy by* jumlah kartu akan

berlaku ketika pemain menjadi pemain pertama atau dapat mengeluarkan kombinasi kartu baru atau bisa dibilang memulai giliran baru. Sedangkan *Greedy by* nilai kartu terendah akan berlaku ketika pemain melawan pemain-pemain lain. Dalam algoritma ini, pemain diusahakan untuk mengeluarkan kartu sebanyak-banyaknya dan sebisa mungkin menghindari kemungkinan "Pass" karena dengan "Pass", musuh dapat terus mengeluarkan kartu sementara pemain harus menyimpan kartunya sehingga kemungkinan untuk menang semakin kecil.

A. Pemetaan elemen-elemen algoritma *Greedy*

Berikut pemetaan elemen-elemen algoritma *Greedy* dengan permasalahan permainan kartu ini pada tahap penyusunan kartu :

1. Himpunan Kandidat (C) : himpunan kumpulan kartu yang berada di tangan pemain
2. Himpunan Solusi (S) : himpunan kombinasi-kombinasi kartu terpilih
3. Fungsi solusi : pada permasalahan tahap ini, fungsi ini berfungsi untuk memastikan bahwa kombinasi yang ada merupakan salah satu dari kombinasi yang ada pada peraturan
4. Fungsi seleksi : fungsi seleksi pada tahap ini akan berguna untuk menentukan ada berapa banyak kartu yang memiliki angka sama dan jenis yang sama, kemudian akan membuat kombinasi-kombinasi sesuai dengan data yang dimiliki. Dengan mengutamakan kombinasi 5 kartu, kemudian 3 kartu, 2 kartu, dan 1 kartu
5. Fungsi kelayakan : fungsi ini harus memastikan bahwa semua kombinasi yang sudah dibentuk adalah kombinasi yang dapat diterima
6. Fungsi Obyektif : fungsi ini berfungsi untuk memastikan bahwa kartu yang dikeluarkan akan membutuhkan giliran terpendek dengan masing-masing giliran mengeluarkan jumlah kartu terbanyak.

Sedangkan pada tahap permainan, berikut pemetaan elemen anggota *Greedy* :

1. Himpunan Kandidat (C) : himpunan kumpulan kombinasi kartu di tangan pemain
2. Himpunan Solusi (S) : himpunan kandidat kartu yang terpilih untuk dikeluarkan oleh pemain
3. Fungsi solusi : Memastikan bahwa kombinasi kartu yang sudah dipilih memiliki nilai lebih tinggi dibanding kartu musuh
4. Fungsi Seleksi : Fungsi seleksi pada tahap ini akan memilih sesuai dengan keadaan. Jika dalam keadaan pemain mendapat giliran pertama atau giliran untuk mengeluarkan kartu baru (setelah sebelumnya semua pemain "Pass") maka akan dipilih kombinasi kartu yang mengeluarkan jumlah terbanyak dengan nilai terkecil. Sedangkan, jika pemain memiliki giliran untuk melawan pemain lain, pertama-tama akan dipilih kombinasi kartu dengan jumlah kartu yang sesuai

dengan kombinasi kartu terakhir yang dimainkan oleh pemain sebelumnya. Setelah itu, dipilih kombinasi kartu dengan nilai terendah apabila ada lebih dari satu, dan jika tidak ada sama sekali, akan dilakukan “Pass”.

5. Fungsi kelayakan : memastikan bahwa kombinasi kartu terpilih memiliki jumlah yang sama dengan jumlah kombinasi kartu musuh
6. Fungsi Obyektif : kombinasi kartu yang dikeluarkan memiliki total nilai terendah dari semua kombinasi kartu yang ada di tangan, namun lebih tinggi dari nilai kombinasi kartu musuh.

B. Penerapan Algoritma Greedy

Dalam permainan Capsa banting, penerapan dapat dilakukan seperti berikut :

1. Ketika pertama kali semua kartu dibagikan, pemain dapat langsung mengelompokkan kartu di tangan dengan algoritma *Greedy* by jumlah kartu. Sehingga pemain akan langsung mencoba membuat kombinasi 5 kartu, entah itu *straight*, *flush*, *full house*, *four of a kind*, atau *straight flush*. Kemudian jika sudah tidak dapat dibuat menjadi kombinasi 5 kartu, maka usahakan cari kombinasi untuk *triple*, dan kemudian untuk *double* dan *single*. Jika dalam pembuatan kombinasi 5 kartu ternyata terdapat 6 kartu yang dapat menjadi kandidat untuk kombinasi 5 kartu, pilih kartu yang lebih rendah. Contohnya terdapat kartu 3, 4, 5, 6, 7, 8. Maka sebenarnya bisa membuat *straight* dengan 3-4-5-6-7 atau 4-5-6-7-8. Pemain dapat memilih kombinasi kartu yang pertama, yaitu 3-4-5-6-7. Karena jika ada kelebihan 1, kemungkinan besar pada akhirnya akan menjadi kartu *single* dan agar dapat memenangkan pertandingan dengan pemain lain, dibutuhkan kartu yang lebih besar, sehingga dipilih kartu lebih kecil untuk kombinasi.
2. Jika pemain memiliki kartu 3♦, maka pemain dapat mengeluarkan kartu tersebut. Jika kartu tersebut berada di dalam paket, maka keluarkan saja satu paket tersebut.
3. Selanjutnya, akan dilanjutkan oleh pemain lain. Ketika giliran sudah kembali pada pemain, maka pilih kombinasi kartu dengan jumlah yang sama, kemudian jika terdapat lebih dari 1 kombinasi, pilih kombinasi terendah yang masih lebih tinggi dari kartu lawan. Pemilihan kombinasi terendah didasarkan pemikiran bahwa jika kemudian pemain lain masih memiliki kombinasi kartu yang lebih tinggi, masih ada nada kemungkinan untuk kombinasi kartu kita dapat keluar lagi seandainya kombinasi kartu lawan lebih rendah dari kombinasi terakhir kita. Namun, jika ternyata kombinasi yang dibutuhkan tidak ada, pemain dapat melakukan “Pass”.
4. Ketika pemain telah mengeluarkan kombinasi kartunya dan ternyata pemain lain semuanya melakukan “Pass”, maka pemain dapat memilih kombinasi kartu yang memiliki jumlah terbanyak, dari 5, 3, 2 baru 1 kartu. Tentu saja, kombinasi yang

dikeluarkan sebisa mungkin merupakan kombinasi dengan nilai terendah dari kumpulan kombinasi dengan jumlah kartu yang sama.

C. Pseudo-code algoritma

Untuk fungsi seleksi dari penyusunan kartu diawal adalah sebagai berikut :

```

function getCombination (hK :
himpunanKartu) → array of himpunanKartu
{kamus}

kombinasiKartu : array of himpunanKartu
function isThere(jeniskombinasi) :
boolean

kombinasiKartu ← []
while isihK 0 do
    if isThere(straightFlush) then
        kombinasiKartu ← straightFlush
        hK ← hK - straightFlush
    else if isThere(fourOfAKind) then
        kombinasiKartu ← fourOfAKind
        hK ← hK - fourOfAKind
    else if isThere(fullHouse) then
        kombinasiKartu ← fullHouse
        hK ← hK - fullHouse
    else if isThere(flush) then
        kombinasiKartu ← flush
        hK ← hK - flush
    else if isThere(straight) then
        kombinasiKartu ← fourOfAKind
        hK ← hK - fourOfAKind
    else if isThere(triples) then
        kombinasiKartu ← triples
        hK ← hK - triples
    else if isThere(doubles) then
        kombinasiKartu ← doubles
        hK ← hK - doubles
    else
        kombinasiKartu ← sisahK
        hK ← hK - sisahK
endwhile
→kombinasiKartu
    
```

Dalam fungsi ini, pemain akan diutamakan untuk pertama membuat kombinasi dari 5 kartu dengan urutan prioritas sebagai berikut :

1. *Straight Flush*
2. *Four of A Kind*
3. *Full House*
4. *Flush*
5. *Straight*

Selanjutnya, apabila kelima kombinasi tersebut sudah tidak bisa dibuat lagi, maka akan dibuat kombinasi dengan prioritas *triples*, *doubles* dan terakhir *singles*. Selanjutnya, akan ditunjukkan *pseudo-code* untuk fungsi seleksi pada tahap permainan, berikut *pseudo-code*-nya :

```

Function getCard(cards : array of
himpunanKartu) → himpunanKartu
{kamus}
kartuSolusi : himpunanKartu
kartuSolusi ← { }
if giliranPertama and have3Diamond then
    kartuSolusi ← paketDengan3Diamond
dalam cards
    cards ← cards - paket3Diamond
else
    if jumlahKartuMusuh = 5 then
        if cardsHave(Kombinasi5) then
            if jumlahKombinasi > 1 then
                kartuSolusi ← Kombinasi5
nilai terendah yang lebih besar dari
musuh
                cards ← cards -
paket3Diamond
            else
                kartuSolusi ← Kombinasi5
                cards ← cards -
paket3Diamond
        else
            kartuSolusi ← pass
        else if jumlahKartuMusuh = 3 then
            if cardsHave(Kombinasi3) then
                if jumlahKombinasi > 1 then
                    kartuSolusi ← Kombinasi3
nilai terendah yang lebih besar dari
musuh
                    cards ← cards -
paket3Diamond
                else
                    kartuSolusi ← Kombinasi3
                    cards ← cards -
paket3Diamond
            else
                kartuSolusi ← pass
            else if jumlahKartuMusuh = 2 then
                if cardsHave(Kombinasi2) then
                    if jumlahKombinasi > 1 then
                        kartuSolusi ← Kombinasi2
nilai terendah yang lebih besar dari
musuh
                        cards ← cards -
paket3Diamond
                    else
                        kartuSolusi ← Kombinasi2
                        cards ← cards -
paket3Diamond
                else
                    kartuSolusi ← pass
            else if jumlahKartuMusuh = 1 then
                if cardsHave(Kombinasi1) then
                    if jumlahKombinasi > 1 then
                        kartuSolusi ← Kombinasi1

```

```

nilai terendah yang lebih besar dari
musuh
                cards ← cards -
paket3Diamond
            else
                kartuSolusi ← Kombinasi1
                cards ← cards -
paket3Diamond
        else
            kartuSolusi ← pass

```

Pada fungsi seleksi diatas, dilakukan filterisasi yang sesuai dengan fungsi seleksi pada pemetaan di bab 3 bagian A ini.

D. Pengujian

Dalam pengujian digunakan website [Play Big Two Cards Game Online \(onlinesolgames.com\)](http://Play Big Two Cards Game Online (onlinesolgames.com)). Pada pembagian pertama, pemain mendapatkan kartu sebagai berikut



Gambar 3.1 Kartu di tangan pemain

Dari kartu yang didapat dibuatlah kombinasi sebagai berikut :

- 1 kombinasi 5 kartu *full house* 6-6-6-8-8
- 1 kombinasi 2 kartu *double* A-A
- 6 kartu satuan yaitu 4-7-10-J-K

Pada giliran pertama, pemain Jen mengeluarkan kartu *pair* 3♦-3♥, dilanjutkan oleh pemain Sal dengan kartu *double* juga yaitu 3♠-3♣. Maka sesuai strategi *Greedy*, pemain akan mengeluarkan kartu *double* A-A, yang kemudian dilawan oleh pemain Ric dengan kartu 2-2. Selanjutnya pemain Ric mengeluarkan kombinasi 5 kartu 9-9-9-Q-Q, yang dibalas oleh pemain Sal dengan kartu 10-10-10-K-K. Karena pemain tidak memiliki kombinasi kartu tepat untuk melawan, maka pemain memilih *pass*. Selanjutnya, giliran pemain Sal. Pemain Sal mengeluarkan kartu 4 wajik, yang kemudian dilawan oleh pemain dengan kartu 4 sekop. Selanjutnya pemain ric melawan dengan kartu 5 tahu yang dibalas oleh pemain Jen dengan 5 hati dan oleh pemain Sal dengan 9 hati dan pemain dengan 10 kriting. Selanjutnya, pemain Ric mengeluarkan J hati, pemain Jen mengeluarkan Q sekop dan pemain Sal mengeluarkan K hati. Pemain kemudian mengeluarkan K sekop dan pemain Ric memilih untuk *pass*. Sedangkan pemain Jen mengeluarkan A kriting, dan pemain Sal mengeluarkan A sekop. Karena pemain sudah tidak memiliki kartu lebih tinggi lagi, maka pemain memilih *pass*. Pemain Jen kemudian mengeluarkan kartu 2 kriting yang dibalas oleh pemain Sal dengan 2 hati. Karena 2 sekop sudah keluar sebelumnya, maka semua pemain *pass*. Pemain Sal tinggal memiliki 1 kartu lagi dan karena semua pemain lain sudah *pass*, pemain Sal memenangkan permainan ini. Didapatkan bahwa pada

percobaan ini, pemain kalah dengan menyisakan cukup banyak kartu. Hal ini bisa terjadi karena beberapa faktor, yang utama adalah karena pemain kurang beruntung dalam mendapatkan kartu-kartu satuan sedangkan pemain lain banyak menggunakan kartu satuan untuk bermain.

Dilakukan beberapa pengujian lagi, dan didapat bahwa pada beberapa kesempatan, penggunaan algoritma *Greedy* dapat menghasilkan hasil terbaik dimana pemain menjadi pemenang. Namun karena permainan ini juga memiliki faktor keberuntungan, maka tidak setiap percobaan algoritma *Greedy* ini akan berhasil.

IV. KESIMPULAN

Setelah pengujian dilakukan, didapatkan bahwa ternyata penggunaan algoritma *Greedy* tidak selalu menghasilkan kemenangan. Akan tetapi, hal ini juga dipengaruhi oleh berbagai macam faktor lainnya. Misalnya, terkadang akan lebih baik untuk melakukan pass dahulu untuk menyimpan kartu sehingga kemudian dapat digunakan dengan lebih efektif. Akan tetapi dalam algoritma ini, karena dengan *Greedy* maka sebisa mungkin mengeluarkan kartu dalam setiap gilirannya.

Selain itu, didapat juga bahwa ternyata faktor keberuntungan juga berpengaruh karena bisa saja ketika mendapat kartu kecil, maka dengan algoritma apapun akan kemudian selalu kalah dengan pemain lain.

LINK VIDEO YOUTUBE

Video mengenai makalah ini dapat dilihat di pranala berikut <https://hmif.link/VideoMakalahStima13520164>

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Esa karena hanya dengan bimbingannya penulis dapat menyelesaikan tugas makalah ini. Penulis mengucapkan terima kasih sebesar-

besarnya kepada dosen pengampu mata kuliah IF2211 Strategi Algoritma, Dr. Nur Ulfa Maulidevi, S.T., M.Sc., Dr. Ir. Rinaldi Munir, M.T., dan Dr. Masayu Leylia Khodra, S.T., M.T. atas bimbingannya selama semester 4 ini. Juga mengucapkan terima kasih kepada asisten mata kuliah IF2211 Strategi Algoritma yang sudah membantu saya dan teman-teman selama keberlangsungan kuliah IF2211 Strategi Algoritma di semester 4 ini. Tidak lupa juga berterima kasih kepada teman-teman yang membantu dan menyemangati selama semester 4 ini.

REFERENSI

- [1] Munir, Rinaldi. 2021. Algoritma *Greedy* (Bagian 1). [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf). Diakses pada 20 Mei 2022 pukul 19.33 WIB
- [2] <https://www.pagat.com/climbing/bigtwo.html>. Diakses pada 21 Mei 2022 pukul 20.36 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Mei 2022



Hilda Carissa Widelia 13520164