

# Aplikasi Algoritma *Greedy* sebagai Strategi Permainan Kartu Remi “41”

M. Ibnu Syah Hafizh (13519177)  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail (gmail): 13519177@std.stei.itb.ac.id

**Abstract**— Permainan kartu memang menjadi pilihan yang tepat ketika ingin melepas penat dan menghilangkan rasa bosan. Meski ada banyak macam permainan kartu di luar sana, permainan kartu remi 41 merupakan jenis permainan kartu yang sering dimainkan dan tidak memakan banyak waktu untuk memainkan permainan ini. Pada permainan kartu remi 41 dibutuhkan strategi untuk memenangkannya, salah satu strategi yang dapat digunakan untuk memenangkan permainan ini adalah algoritma *greedy*

**Kata Kunci**—Permainan Kartu Remi; 41; strategi; Algoritma *Greedy*

## I. PENDAHULUAN



Gambar 1.1. Kartu Remi  
Sumber:

[https://cdn.shopify.com/s/files/1/1901/3435/products/pc108-caspari-plaid-playing-cards-2-decks-included-11566565425199\\_1024x1024.jpg?v=1567601450](https://cdn.shopify.com/s/files/1/1901/3435/products/pc108-caspari-plaid-playing-cards-2-decks-included-11566565425199_1024x1024.jpg?v=1567601450) (diakses pada 7 Mei 2021, 01.44)

Kartu permainan atau lebih dikenal dengan kartu remi adalah sekumpulan kartu seukuran tangan yang digunakan untuk permainan kartu. Kartu ini sering juga digunakan untuk hal-hal lain, seperti sulap, enkripsi, permainan papan, dan pembuatan rumah kartu. Kata “remi” itu sendiri sebenarnya adalah nama salah satu permainan kartu.

Ada 1001 macam permainan kartu. Setiap negara, bahkan wilayah suatu negara, memiliki jenis permainannya sendiri. Di Indonesia, akrab dengan istilah permainan “41”, truf, “remi”, “cangkulan”, dsb. Namun, yang populer di banyak negara misalnya poker, canasta, blackjack, casino, solitaire, bridge, dengan jumlah pemain yang bisa berbeda-beda.

Seperti yang kita kenal sekarang, satu pak kartu remi berisi 52 lembar. Kartu remi dibagi menjadi 4 suit atau jenis kartu yaitu:

- ♠ Spade atau sekop (juga disebut "waru")
- ♥ Heart atau hati
- ♦ Diamond atau berlian (juga disebut "wajik")
- ♣ Club atau keriting

Masing-masing terdiri atas 13 kartu (2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King, dan Ace/As) ditambah dua kartu Joker berwarna hitam dan merah, biasanya kartu Joker jarang dipakai dalam permainan.

Algoritma adalah suatu metode khusus yang tepat dan terdiri dari serangkaian langkah yang terstruktur dan dituliskan secara matematis yang akan dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer. Jadi berdasarkan definisi ini, dapat dikatakan bahwa algoritma merupakan langkah penyelesaian suatu masalah yang menghasilkan solusi dalam bentuk program komputer. Namun penting diketahui bahwa algoritma tidaklah tergantung oleh suatu bahasa pemrograman tertentu, artinya suatu algoritma harus dapat diwujudkan oleh bahasa pemrograman komputer apapun.

Terdapat berbagai jenis algoritma, yaitu Algoritma BruteForce, Algoritma Greedy, Algoritma Divide and Conquer, Dynamic Programming, dan lain-lain.

Algoritma dapat disajikan dalam dua bentuk yaitu dalam bentuk tulisan/bahasa dan dalam bentuk gambar. Penyajian algoritma dalam bentuk tulisan haruslah menggunakan bahasa yang dapat dimengerti manusia dalam menyajikan langkah-langkah algoritma. Penyajian algoritma dalam bentuk tulisan juga dapat dilakukan menggunakan pseudocode.

Algoritma memiliki beberapa fungsi, yaitu algoritma bisa digunakan untuk memecahkan program yang rumit, bisa menjadikan program yang besar menjadi program yang lebih sederhana, bisa digunakan secara berulang atau lebih dari satu kali penggunaan, dan bisa memecahkan masalah dengan logika dan urutan.

Ketika bermain permainan kartu, tentunya setiap pemain berlomba-lomba untuk mendapatkan kemenangan. Untuk mendapatkan kemenangan dibutuhkan strategi yang cerdas dan tepat. Pada kondisi ini algoritma yang tepat dapat dimanfaatkan sebagai strategi salah satu permainan kartu. Contohnya,

permainan kartu remi 41 dapat menerapkan algoritma *greedy* sebagai strateginya.

Algoritma *greedy* serta pengimplementasiannya terhadap *integer knapsack problem* dapat diaplikasikan sebagai strategi salah satu permainan kartu remi yaitu permainan “41”. Dengan menggunakan algoritma *greedy*, pemain dapat memaksimalkan nilai dari kartu-kartu tersebut sehingga memperoleh nilai sempurna yakni 41 dan menjadi pemenang tanpa melebihi bobot yang telah ditentukan.

## II. TEORI DASAR

### A. Algoritma Greedy

Algoritma *greedy* merupakan metode yang paling populer untuk memecahkan persoalan optimasi (persoalan mencari solusi optimum). Persoalan optimasi hanya memiliki dua macam, yakni maksimasi dan minimasi. Solusi optimum adalah solusi yang bernilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin. Dalam setiap persoalan optimasi memiliki dua elemen, yakni kendala (constraints) dan fungsi objektif (fungsi optimasi).

Algoritma ini memiliki prinsip: “Take what you can get now!”. Artinya, pada setiap langkah dalam algoritma *greedy* akan diambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan, kemudian berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Algoritma ini disusun oleh elemen-elemen berikut:

- Himpunan kandidat : elemen-elemen pembentuk solusi
- Himpunan solusi : kandidat-kandidat yang terpilih sebagai solusi persoalan
- Fungsi seleksi : memilih kandidat yang paling memungkinkan untuk mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya
- Fungsi kelayakan : memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (constraints) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi
- Fungsi obyektif : fungsi yang memaksimalkan atau meminimumkan nilai solusi

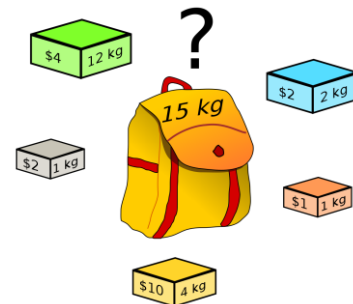
Adapun skema umum dari algoritma *greedy* adalah sebagai berikut:

```

function greedy(input C: himpunan_kandidat)→
himpunan_kandidat
Deklarasi
  x : kandidat
  S : himpunan_kandidat
Algoritma:
S ← {} { inisialisasi S dengan kosong }
while (not SOLUSI(S)) and (C ≠ {} ) do
  x ← SELEKSI(C) { pilih sebuah kandidat dari C }
  C ← C - {x} { elemen himpunan kandidat berkurang 1 }
  if LAYAK(S ∪ {x}) then
    S ← S ∪ {x}
  endif
endwhile
{ SOLUSI(S) or C = {} }
if SOLUSI(S) then
  return S
else
  write('tidak ada solusi')
endif

```

Salah satu persoalan optimasi yang dapat diselesaikan melalui algoritma ini adalah persoalan integer knapsack. Diberikan sebuah knapsack dengan kapasitas M. Terdapat n buah objek, setiap objek memiliki bobot  $w_i$  dan keuntungan  $p_i$ . Objektif dari persoalan ini adalah bagaimana cara memilih objek-objek yang dimasukkan ke dalam knapsack sehingga total keuntungan yang diperoleh maksimal.



Gambar 2.1. Knapsack Problem

Sumber: [https://en.wikipedia.org/wiki/Knapsack\\_problem](https://en.wikipedia.org/wiki/Knapsack_problem) (diakses pada 8 Mei 2021, 02.26)

Terdapat beberapa strategi *greedy* yang heuristik yang dapat digunakan untuk memilih objek yang akan dimasukkan ke dalam knapsack. Adapun strategi *greedy* yang dilakukan dapat dibagi menjadi 3 objektif, yakni *greedy by profit*, *greedy by weight*, dan *greedy by density*.

- *Greedy by profit*
  - a. Pada setiap langkah, pilih objek yang mempunyai keuntungan terbesar
  - b. Mencoba memaksimalkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu

- *Greedy by weight*
  - Pada setiap langkah, pilih objek yang mempunyai berat teringan
  - Mencoba memaksimumkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam knapsack
- *Greedy by density*
  - Pada setiap langkah, pilih objek yang memiliki density ( $p_i/w_i$ ) terbesar
  - Mencoba memaksimumkan keuntungan dengan memilih objek yang mempunyai keuntungan per unit berat terbesar

Berikut ini adalah contoh masalah integer knapsack dengan jumlah  $n = 4$ ,

$$w_1 = 6; p_1 = 12;$$

$$w_2 = 5; p_2 = 15;$$

$$w_3 = 10; p_3 = 50;$$

$$w_4 = 5; p_4 = 10;$$

dengan kapasitas knapsack  $K = 16$ , maka apabila ditulis dalam tabel adalah sebagai berikut:

Tabel 1. *Integer Knapsack Problem*

Properti Objek				<i>Greedy by</i>			Solusi Optimal
i	$w_i$	$p_i$	$p_i/w_i$	<i>profit</i>	<i>weight</i>	<i>density</i>	
1	6	12	2	0	1	0	0
2	5	15	3	1	1	1	1
3	10	50	5	1	0	1	1
4	5	10	2	0	1	0	0
Total bobot				15	16	15	15
Total keuntungan				65	37	65	65

Solusi optimal dari masalah ini akan memberikan keuntungan atau total profit sebesar 65. *Greedy by profit* dan *density* dapat mencapai solusi optimal namun *greedy by weight* tidak. Kesimpulannya, algoritma *greedy* tidak selalu memberikan solusi yang optimal.

### B. Permainan Kartu Remi 41

Permainan kartu remi 41 merupakan permainan kartu yang menggunakan satu set kartu remi berisi 52 kartu tanpa kartu joker. Biasanya permainan ini dimainkan oleh empat orang. Di dalam permainan ini, pemain harus mengumpulkan empat kartu yang sama dengan nilai tertinggi. Sesuai dengan nama permainannya, 41 adalah skor paling sempurna di permainan ini. Dalam peraturan permainan kartu remi 41, masing-masing kartu memiliki poin yang berbeda, jika pemain mengumpulkan empat kartu, menang atau tidaknya ditentukan oleh skor dari masing-

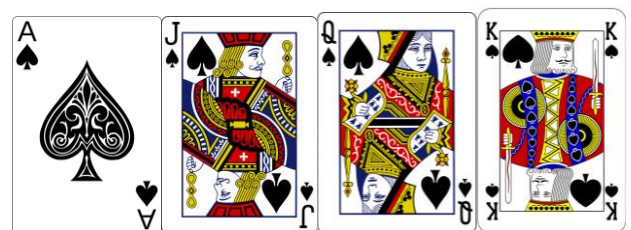
masing kartu yang didapat. Berikut tabel perhitungan poin pada masing-masing kartu:

Tabel 2. *Nilai Poin Kartu*

Kartu	Poin
As/Ace	11
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
Jack	10
Queen	10
King	10

Namun, jangan sembarang mengumpulkan kartu, karena untuk mendapat skor yang baik, pemain harus mengambil kartu dari jenis dan warna yang sama, jika ada kartu yang berbeda jenisnya di antara empat kartu yang dimilikinya, maka jumlahnya akan dikurangi dengan jumlah poin dari kartu yang berbeda.

Awal permainan hampir sama dengan cara bermain cangkulan, ada seseorang yang mengocok kartu lalu membagikan empat kartu untuk tiap pemain, lalu menyimpan sisa stok kartu di tengah-tengah. Pemain pertama akan memulai mengambil kartu di stok kartu tersebut dan bergiliran sesuai dengan arah jarum jam. Para pemain boleh membuang kartunya jika ada yang tidak cocok atau tidak sama jenisnya, namun tidak diperbolehkan mengambil kartu yang sudah dibuang. Permainan berakhir ketika salah satu pemain mendapatkan skor sempurna atau stok kartu habis. Skor sempurna adalah empat kartu berjenis sama yang nilainya 41, contohnya:



Gambar 2.2 Susunan kartu yang menghasilkan poin 41

Sumber:

[https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Ace\\_of\\_spades.svg/1200px-Ace\\_of\\_spades.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Ace_of_spades.svg/1200px-Ace_of_spades.svg.png) (diakses pada 8 Mei 2021, 02.53)

Permainan berakhir apabila terjadi salah satu diantara dua kondisi berikut:

1. Salah satu pemain telah berhasil mendapatkan skor 41  
Pemain yang terlebih dahulu mendapatkan skor 41 adalah pemenangnya.
2. Kartu ditumpukan telah habis  
Apabila kartu di tumpukan telah habis, maka otomatis permainan akan berakhir. Penentuan pemenang dilihat dari jumlah poin yang dikumpulkan masing-masing pemain. Nilai kartu yang sejenis dijumlahkan dan dikurangi nilai kartu yang berbeda jenis. Pemain dengan nilai tertinggi menjadi pemenangnya.

Seperti yang telah disebutkan pada paragraf sebelumnya, jika terdapat kartu yang tak sama, maka kartu yang tidak sama tersebut akan menjadi nilai negatif. Contohnya, jika pemain A mendapat 4 kartu yang terdiri dari King hati, Jack hati, 10 sekop dan 4 sekop, maka perhitungannya menjadi  $10+10-10-4 = 6$ . Namun, bukan berarti skor dari kartu yang sama jenisnya sudah pasti akan menang. Misalnya, jika kartu pemain A King-6-5-3 dan semuanya berjenis hati, berarti skor pemain A adalah  $10+6+5+3 = 24$ , pemain A bisa dikalahkan oleh pemain lain yang kartunya As-King-10 berjenis sekop dan angka 3 berjenis wajik, maka pemain lain tersebut masih bisa menang dengan skor 28.

### III. ALGORITMA GREEDY PADA PERMAINAN KARTU REMI 41

Permainan kartu remi 41 mencari pemenang dengan skor sempurna yakni 41 atau pemain dengan nilai kartu yang paling tinggi dari kombinasi empat buah kartu jika kartu pada tumpukan habis. Setiap pemain menentukan strategi yang tepat untuk dapat memenangkan permainan ini. Tujuan dari strategi ini adalah mendapatkan nilai semaksimal mungkin dari kombinasi empat buah kartu.

Salah satu strategi yang dapat digunakan yaitu dengan memanfaatkan algoritma *greedy*. Algoritma *greedy* adalah algoritma yang memaksimalkan atau meminimumkan pilihan pada setiap langkah (optimum lokal) dengan harapan memperoleh nilai optimal pada akhirnya (optimum global) tanpa memikirkan konsekuensi ke depan. Pada permainan kartu remi 41, pemain harus memaksimalkan nilai kartu yang dimilikinya pada setiap langkah agar mendapatkan skor 41 tercepat atau mendapatkan skor terbesar dari kombinasi kartu yang dimilikinya. Permainan ini tidak dapat mengetahui secara pasti kartu apa yang didapat pada langkah-langkah selanjutnya karena kartu pada tumpukan di tengah dalam keadaan acak sehingga pemain harus memaksimalkan nilai kartunya pada setiap langkah. Oleh karena itu, Algoritma *greedy* menjadi salah satu alternatif strategi yang cocok untuk permainan kartu remi 41.

Untuk mengaplikasikan algoritma *greedy* dalam menentukan keputusan yang diambil dalam setiap langkah pada permainan kartu remi 41, perlu ditentukan elemen-elemen penyusun algoritma ini terlebih dahulu.

- Himpunan kandidat: lima kartu yang terdiri atas empat kartu yang dimiliki pemain sekarang dan satu kartu ambilan.

- Himpunan solusi: empat kartu yang dipilih dari lima kartu yang setelah diseleksi.
- Fungsi seleksi: memilih kartu dengan nilai tertinggi atau menghasilkan kombinasi nilai tertinggi.
- Fungsi kelayakan: kartu yang dipilih berpotensi untuk membentuk nilai maksimum pada himpunan solusi.
- Fungsi obyektif: memiliki nilai maksimum setiap kartunya, atau kombinasi nilai dari empat kartu yang dihasilkan bernilai maksimum.

Melalui elemen-elemen yang sudah ditulis diatas, algoritma *greedy* dalam masalah ini dapat diimplementasikan.

#### A. Implementasi Integer Knapsack Problem dalam Permainan Kartu Remi 41

Pada persoalan *integer knapsack*, *knapsack* memiliki batasan sebesar berat maksimal yang dapat dibawa oleh tas. Apabila dianalogikan dalam persoalan permainan kartu remi 41 ini, dapat dimisalkan bahwa 'berat maksimal tas' adalah jumlah maksimal kartu yang dapat dimiliki oleh pemain yakni empat kartu sehingga pemain harus membuang sebuah kartu setelah mengambil kartu dari tumpukan kartu agar kartu yang dimilikinya tidak melebihi batas. Setiap kartu memiliki bobot yang sama yakni berbobot satu.

*Integer knapsack problem* menentukan objektif dari algoritma *greedy* berdasarkan *weight*, *profit*, dan *density*. Oleh karena setiap kartu memiliki bobot yang sama, maka *greedy* berdasarkan *weight* tidak akan berpengaruh (tidak dapat mengoptimasi nilai kartu) sehingga *greedy* berdasarkan *weight* atau *greedy* berdasarkan bobot tidak digunakan pada persoalan ini. Untuk persoalan pemilihan kartu agar membentuk kombinasi nilai maksima, digunakan dua objektif yakni *greedy* berdasarkan poin dan *greedy* berdasarkan jenis kartu.

#### B. Greedy berdasarkan poin



Gambar 3.1. Pemilihan kartu berdasarkan nilai  
Sumber : <https://n2.sdlcdn.com/imgs/b/8/j/Bharatham-Stores-Others-Plastic-Playing-SDL375130543-1-c8682.jpg> (diakses pada 9 Mei 2021, 03.19)

Seperti namanya, *greedy* berdasarkan poin adalah penentuan empat kartu berdasarkan poin tertingginya. Setiap mengambil kartu dari tumpukan kartu, diurutkan berdasarkan poin kartunya kemudian 4 kartu dengan poin tertinggi dimasukkan ke dalam himpunan solusi dan kartu yang memiliki poin terkecil dibuang tanpa melihat jenisnya. Jika terdapat dua atau lebih kartu dengan poin yang sama maka yang dipilih adalah kartu dengan jenis yang mendominasi. Terdapat kelebihan dan kekurangan dengan metode ini.

Kelebihannya adalah pemain mendapatkan kartu-kartu dengan nilai tertinggi sehingga harapannya dari kartu-kartu tersebut dapat membentuk skor 41. Sedangkan kekurangannya ialah ketika hingga kartu tumpukan habis belum mendapatkan semua kartu dengan jenis yang sama sehingga pengurangan poinnya akan cukup besar. Seperti contoh pada gambar 3.1 terdapat kartu Jack wajik, Queen sekop, King hati, dan As sekop. Total skor dari empat kartu tersebut adalah  $10-10-10+11=1$ . Meskipun tidak negatif tetapi nilai 1 adalah nilai yang kecil.

Oleh karena itu, *greedy* berdasarkan poin tidak menjamin hasil akhir (nilai kombinasi empat kartu) optimal walaupun setiap langkahnya dipilih optimal lokal (berdasarkan nilai tiap kartu). Alternatif solusi lainnya adalah *greedy* berdasarkan jenis.

C. *Greedy* berdasarkan jenis



Gambar 3.2. Pemilihan kartu berdasarkan jenis

Sumber :

[https://www.thesprucecrafts.com/thumb/ByLCuYSqYeFZ65K87B4FOHsHq0=/1335x1001/smart/filters:no\\_upscale\(\)/Hearts-5c195fc746e0fb000103d252.jpg](https://www.thesprucecrafts.com/thumb/ByLCuYSqYeFZ65K87B4FOHsHq0=/1335x1001/smart/filters:no_upscale()/Hearts-5c195fc746e0fb000103d252.jpg) (diakses pada 9 Mei 2021, 03.35)

*Greedy* berdasarkan jenis adalah pemilihan empat kartu berdasarkan jenis yang paling banyak tanpa memikirkan poin yang dikandungnya. Apabila empat kartu yang dimiliki sekarang sudah sama jenisnya dan kartu yang diambil dari tumpukan adalah kartu yang sama jenisnya dengan empat kartu tersebut maka yang dimasukkan ke himpunan solusi dimulai dari yang terbesar hingga yang terkecil. Kartu terkecil akan dibuang karena bobot yang dapat ditampung hanya empat kartu. Sebagai contoh, apabila empat kartu yang dimiliki sekarang tidak ada yang berjenis sama maka ketika mengambil kartu, jenis yang dipilih terlebih dahulu untuk dimasukkan ke dalam himpunan

solusi yaitu semua kartu yang berjenis sama dengan jenis kartu yang diambil. Hal ini karena total kartu dengan jenis yang sama dengan jenis kartu yang diambil adalah dua yaitu kartu yang diambil tersebut dan salah satu kartu dari empat kartu yang dimiliki sekarang.

*Greedy* berdasarkan jenis bertujuan untuk meminimumkan pengurangan nilai yang disebabkan oleh jenis kartu yang berbeda dan memaksimalkan total nilai dari kombinasi keempat kartu. Jika terdapat tiga kartu berjenis sama dan dua kartu berjenis berbeda maka yang dipilih untuk dimasukkan ke dalam himpunan solusi adalah kartu 3 berjenis sama (tanpa melihat nilai lainnya) dan 1 kartu berjenis berbeda dengan poin yang minimum. Pemilihan ini digunakan untuk meminimumkan pengurangan poin. Apabila dari empat kartu yang dimiliki sekarang berjenis sama maka tidak terdapat pengurangan poin sehingga nilai terkecil dari empat kartu yang berjenis sama adalah  $2+3+4+5 = 14$ . Nilai 14 tidak terlalu buruk dibandingkan dengan nilai terburuk dari *greedy* by nilai yang bisa saja poinnya bernilai negatif.

Namun terdapat kekurangan dalam algoritma ini, *greedy* by jenis kartu tidak menjamin hasil yang optimal atau mendapatkan nilai sempurna. *Greedy* berdasarkan jenis kartu juga bisa saja tidak berhasil. Pada akhir permainan, empat kartu yang dimiliki tidak berjenis sama. Hal ini bisa saja terjadi. salah satu faktornya yaitu jika setiap pengambilan kartu selalu mendapatkan kartu dengan jenis yang berbeda dari pengambilan sebelum-sebelumnya sehingga empat kartu berjenis sama tidak akan terbentuk.

IV. STUDI KASUS

Permainan kartu remi 41 menggunakan satu set kartu remi tanpa joker yang berisi 52 kartu dan dimainkan oleh empat orang. Setiap orang menerima empat buah kartu awal secara acak dan sisa kartu ditumpuk di tengah-tengah keempat pemain.

Sebagai contoh, mari kita tempatkan diri kita sebagai salah satu pemain. Kita mendapatkan empat kartu pertama yaitu 3 sekop, 7 wajik, 4 hati, dan Jack keriting. Selanjutnya, kita akan mengambil kartu setiap putaran sebagai berikut:

Tabel 3. Pengambilan kartu setiap putaran

Putaran ke-	kartu	jenis
1	9	Hati
2	King	Keriting
3	6	Keriting
4	As	Hati
5	Queen	Wajik
6	10	Hati
7	8	Wajik
8	2	Sekop
9	Jack	Hati

10	As	Keriting
----	----	----------

Dengan asumsi permainan berhenti karena kartu pada tumpukan kartu habis atau kita mendapat skor 41. Selanjutnya Tinjau berdasarkan algoritma *greedy* berdasarkan nilai kartu dan *greedy* berdasarkan jenis kartu.

#### A. Greedy Berdasarkan Nilai Kartu

Pada putaran pertama, kita mendapatkan 9 hati sehingga sekarang kita mempunyai 5 kartu pada himpunan kandidat yaitu empat kartu sebelumnya ditambah 9 hati. Kartu yang dimasukkan ke dalam solusi dengan algoritma *greedy* berdasarkan nilai kartu adalah jack keriting, 9 hati, 7 wajik, dan 4 hati, kartu yang dibuang adalah 3 sekop karena bernilai paling kecil. Sekarang semua kartu sudah dalam optimum lokalnya. Selanjutnya dengan cara yang sama akan dicari himpunan solusi setiap putarannya. Berikut adalah tabel himpunan solusi pada tiap putaran:

Tabel 4. Himpunan Solusi Greedy berdasarkan Nilai Kartu

Putaran ke-	Himpunan solusi
0	{3 sekop, 7 wajik, 4 hati, Jack keriting}
1	{Jack keriting, 9 hati, 7 wajik, 4 hati}
2	{King keriting, Jack keriting, 9 hati, 7 wajik}
3	{King keriting, Jack keriting, 9 hati, 7 wajik}
4	{As hati, King keriting, Jack keriting, 9 hati}
5	{As hati, King keriting, Jack keriting, Queen wajik}
6	{As hati, King keriting, Jack keriting, 10 hati}
7	{As hati, King keriting, Jack keriting, 10 hati}
8	{As hati, King keriting, Jack keriting, 10 hati}
9	{As hati, King keriting, Jack hati, 10 hati}
10	{As hati, As keriting, Jack hati, 10 hati}

Hingga akhir permainan, kartu tidak terbentuk skor 41. Kartu pada putaran terakhir adalah As keriting, As hati, Jack hati, dan 10 hati. Total poin dari 4 kartu tersebut secara berturut-turut yakni  $11+11+10+10=20$ . As keriting bernilai negatif karena berbeda jenis dengan 3 kartu lainnya yaitu hati. Walaupun setiap langkah kartu dipilih berdasarkan poin terbesarnya tetapi pada akhirnya tidak membentuk nilai optimum global. Pada putaran ke-10 didapat himpunan kandidat adalah {As hati, king keriting, jack hati, 10 hati, As keriting}, seharusnya kartu yang tidak dimasukkan ke dalam himpunan solusi adalah As keriting karena jika yang dipilih bukan As keriting melainkan king keriting, total poin yang didapat adalah  $11-10+10+10=21$  (lebih baik daripada As keriting yang dipilih). Namun karena menggunakan algoritma *greedy* berdasarkan nilai, maka yang dipilih adalah As keriting. Walaupun pada algoritma ini

mendapatkan kartu dengan poin yang tinggi, tetapi berpotensi untuk mendapatkan pengurangan poin yang tinggi pula.

#### B. Greedy Berdasarkan Jenis Kartu

Pada putaran pertama, kita mendapatkan 9 hati sehingga sekarang kita mempunyai lima kartu pada himpunan kandidat yaitu empat kartu sebelumnya ditambah 9 hati. Kartu yang dimasukkan ke dalam solusi dengan algoritma *greedy* berdasarkan jenis kartu adalah 9 hati, 4 hati, 7 wajik, dan 3 sekop, kartu yang dibuang adalah Jack keriting karena berjenis berbeda dengan yang lainnya dan nilainya paling besar. Sekarang semua kartu sudah dalam optimum lokalnya. Selanjutnya dengan cara yang sama akan dicari himpunan solusi setiap putarannya. Berikut adalah tabel himpunan solusi pada tiap putaran:

Tabel 5. Himpunan Solusi Greedy berdasarkan Jenis Kartu

Putaran ke-	Himpunan solusi
0	{3 sekop, 7 wajik, 4 hati, Jack keriting}
1	{9 hati, 4 hati, 7 wajik, 3 sekop}
2	{9 hati, 4 hati, 7 wajik, 3 sekop}
3	{9 hati, 4 hati, 6 keriting, 3 sekop}
4	{As hati, 9 hati, 4 hati, 3 sekop}
5	{As hati, 9 hati, 4 hati, 3 sekop}
6	{As hati, 10 hati, 9 hati, 4 hati}
7	{As hati, 10 hati, 9 hati, 4 hati}
8	{As hati, 10 hati, 9 hati, 4 hati}
9	{As hati, 10 hati, Jack hati, 9 hati}
10	{As hati, 10 hati, Jack hati, 9 hati}

Hingga akhir permainan, kartu tidak terbentuk skor 41. Kartu pada putaran terakhir adalah As hati, 10 hati, Jack hati, dan 9 hati. Total poin dari 4 kartu tersebut secara berturut-turut yakni  $11+10+10+9=40$ . Poin 40 adalah poin yang mendekati poin sempurna. Salah satu faktor kenapa tidak mendapatkan poin sempurna adalah karena tidak mendapat kartu hari dengan poin 10 dari tumpukan kartu sehingga tidak dapat membentuk kombinasi poin sempurna dari 4 kartu yang dimiliki. Namun, dengan memaksimalkan nilai kartu yang dimiliki dan meminimalkan pengurangan poin, algoritma *greedy* berdasarkan jenis merupakan salah satu alternatif yang cukup baik. Walaupun pada studi kasus dengan algoritma *greedy* berdasarkan jenis ini memberikan skor yang lebih tinggi daripada algoritma *greedy* berdasarkan nilai, bukan berarti *greedy* berdasarkan jenis mutlak lebih baik daripada *greedy* berdasarkan nilai.

## V. KESIMPULAN

Kesimpulan yang dapat diambil dari makalah ini adalah:

1. Algoritma *greedy* dapat diterapkan sebagai salah satu alternatif strategi dalam permainan kartu remi 41.
2. Dua alternatif strategi dengan menggunakan algoritma *greedy* yaitu *greedy* berdasarkan nilai dan *greedy* berdasarkan jenis kartu.
3. Algoritma *greedy* tidak menjamin hasil optimal karena terdapat strategi-strategi lain sebagai penunjang kemenangan (contoh: peluang) yang tidak dibahas pada makalah ini dan faktor-faktor lain yang menghambat kemenangan (contoh: kartu yang didapat tiap putaran dalam keadaan acak)

### TAUTAN VIDEO DI YOUTUBE

<https://youtu.be/UezDuLtEK7E>

### UCAPAN TERIMA KASIH (*Heading 5*)

Segala puji bagi Allah SWT yang telah memberi kemudahan dan kelancaran dalam menyusun makalah ini sehingga dapat diselesaikan dengan baik dan tepat waktu. Terima kasih kepada kedua orang tua dan teman-teman yang memberi dukungan secara moral maupun doa selama penyusunan makalah. Terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T selaku dosen mata kuliah Strategi Algoritma yang telah memberi ilmu yang menjadi pondasi dasar dari makalah ini. Disadari masih terdapat kekurangan dan kesalahan kata dalam makalah ini, diharapkan makalah ini dapat berguna dan bermanfaat serta dikembangkan lebih jauh sehingga berdampak untuk masyarakat luas.

## REFERENSI

- [1] Algoritma *Greedy*. Dipetik pada 7 Mei 2021, pukul 02.07 dari Homepage Rinaldi Munir : [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Algoritma-Greedy-\(2020\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Algoritma-Greedy-(2020).pdf)
- [2] Permainan Kartu Remi 41. Dipetik pada 8 Mei 2021, pukul 02.45 dari Website Permainan Seru dan Menarik: <https://gamerhandal.wordpress.com/2018/11/27/mengenal-permainan-kartu-remi-41/#:~:text=Permainan%20kartu%20remi%2041%20merupakan,paling%20sempurna%20di%20permainan%20ini.>
- [3] N. Erzed dan A. Rospita. *Permainan Kartu Remi 41 dengan Menggunakan Algoritma Linier Congruential Generator (LCG) sebagai Pembangkit Angka Random*, Jurnal Sekolah Tinggi Teknologi Informasi NIIT I-Tech, vol. 10(2), 2014.

## 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.

Bogor, 11 Mei 2021



M. Ibnu Syah Hafizh  
13519177