

Penerapan Algoritma Greedy dalam Strategi Permainan Kartu UNO

Louis Leslie 13516087

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516087@std.stei.itb.ac.id

Abstrak—Pemainan kartu merupakan salah jenis permainan yang disenangi berbagai kalangan usia. Dalam permainan kartu, dapat dijumpai berbagai macam strategi atau teknik permainan. Tidak jarang pula bila strategi permainan kartu yang dipakai hanya berupa intuisi. UNO merupakan salah satu permainan kartu yang memiliki peraturan dan cara permainan yang unik.

Algoritma *greedy* merupakan algoritma untuk memecahkan masalah-masalah terkait optimasi. Dalam makalah ini, akan dibahas berbagai strategi dalam permainan kartu UNO dengan algoritma *greedy* untuk mengoptimasi performa pemain dalam permainan ini.

Kata Kunci—permainan kartu; strategi permainan; algoritma *greedy*; optimasi

I. PENDAHULUAN

Permainan Kartu UNO merupakan permainan kartu klasik yang digemari berbagai kalangan usia, mulai dari anak-anak sampai dewasa. Peraturan permainan UNO tidaklah rumit, sehingga mudah dimengerti bagi anak-anak, tetapi tetap menantang untuk dimainkan oleh kalangan dewasa. Kartu untuk permainan merupakan kartu yang dicetak secara khusus. Permainan ini dikembangkan oleh **Merle Robbins** pada tahun 1971 dan menjadi produk perusahaan **Mattel, Inc.**

Permainan ini dimainkan oleh beberapa pemain dan dengan strategi atau teknik permainan yang beragam. Permainan kartu ini dimainkan secara *shedding* (pemain mengeluarkan/meletakkan kartu pada satu tumpukan kartu bersama). Meski tidak ada cara bermain yang paling efektif, strategi untuk permainan ini dapat mengikuti sebuah pola atau algoritma. Salah satu algoritma yang cocok dalam memodelkan strategi permainan ini yaitu algoritma *greedy*.



Gambar 1 – Kartu Permainan UNO (sumber: <https://www.walmart.ca/en/ip/uno-card-game/6000019435148>)

II. TEORI DASAR

A. Permainan UNO

1. Kartu Permainan

Dalam satu dek kartu permainan UNO terdapat 108 kartu. Secara umum, kartu dalam permainan UNO terbagi atas empat warna, yaitu merah, biru, hijau, kuning dan *wildcard*. Berikut jenis-jenis kartu dalam permainan UNO.

1) Kartu angka

Dalam permainan ini terdapat kartu bertuliskan angka 0 sampai dengan 9. Untuk setiap warna kartu terdapat satu kartu angka 0 dan dua kartu untuk masing-masing angka lainnya (angka 1 s.d. 9). Kartu angka memiliki nilai sebesar angka yang terdapat pada kartu tersebut.



Gambar 2 – Kartu angka (sumber: <http://besttoys4toddlers.com>)

2) Kartu aksi

Ada tiga jenis kartu aksi yakni *reverse*, *skip*, dan *draw 2*. Untuk masing-masing warna kartu, terdapat dua kartu untuk masing-masing dari tiga jenis kartu aksi ini (2 kartu *reverse*, 2 kartu *skip*, dan 2 kartu *draw two*). Setiap kartu aksi memiliki nilai 20.

i. Reverse

Kartu *reverse* berfungsi untuk memutar arah permainan bagi pemain (searah jarum jam ke arah berlawanan jarum jam dan sebaliknya).

ii. Skip

Kartu *skip* berfungsi untuk meloncati/ mengabaikan giliran pemain selanjutnya.

iii. Draw Two

Kartu *draw two* berfungsi agar pemain selanjutnya menarik 2 kartu dan gilirannya diloncati.



Gambar 3 – Kartu Aksi

Dari kiri ke kanan : kartu *skip*, kartu *reverse*, kartu *draw two*
(sumber : [https://en.wikipedia.org/wiki/Uno_\(card_game\)](https://en.wikipedia.org/wiki/Uno_(card_game)))

3) Kartu *wildcard*

Kartu *wildcard* ditandai dengan warna latar belakang hitam dan tidak tergolong ke empat warna kartu utama. Ada dua jenis *wildcard* yaitu *wild card (change color)* dan *wild draw four*. Setiap kartu *wildcard* memiliki nilai 50.

i. *Wild Card (change color)*

Kartu ini berfungsi untuk mengubah warna terakhir dalam permainan sesuai dengan keinginan pengguna.



Gambar 4 – Kartu Wild Card

(sumber: [https://en.wikipedia.org/wiki/Uno_\(card_game\)](https://en.wikipedia.org/wiki/Uno_(card_game)))

ii. *Wild Draw Four*

Kartu ini berfungsi agar pemain selanjutnya menarik empat kartu, meloncati gilirannya, dan mengubah warna terakhir dalam permainan sesuai dengan keinginan pengguna.



Gambar 5 – Kartu Wild Draw Four

(sumber: [https://en.wikipedia.org/wiki/Uno_\(card_game\)](https://en.wikipedia.org/wiki/Uno_(card_game)))

2. Peraturan Permainan

1) Permainan

Satu permainan UNO terdiri dari satu atau lebih set permainan. Satu permainan dimainkan oleh 2 sampai 10 orang pemain. Dalam setiap set permainan, pemain akan mendapatkan poin. Permainan berakhir jika salah satu pemain telah mendapatkan poin minimal 500.

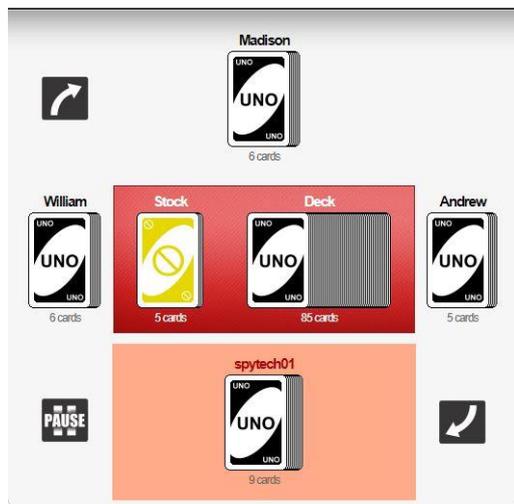
2) Set Permainan

Satu set permainan ini dimulai dengan membagikan 7 kartu ke tangan setiap pemain. Sisa kartu ditempatkan ke tumpukan secara tertutup sebagai tumpukan bersama. Kartu teratas dari tumpukan dibuka dan ditempatkan terbuka ke tumpukan kartu keluaran.

Setiap pemain secara bergilir harus mengeluarkan kartu yang 'cocok' terhadap kartu teratas pada tumpukan kartu keluaran dan meletakkannya pada tumpukan tersebut. Sebuah kartu 'cocok' dengan kartu lainnya apabila memiliki warna yang sama dengan kartu tersebut atau memiliki simbol yang sama dengan kartu tersebut. Misalnya kartu pada tumpukan teratas kartu keluaran adalah kartu 5 hijau, maka pemain dapat meletakkan kartu berwarna hijau apapun atau kartu angka 5 dengan warna apapun. Selain mengeluarkan kartu yang 'cocok', pemain juga dapat memainkan kartu jenis *wildcard* tanpa menghiraukan kartu teratas pada tumpukan kartu keluaran. Jika pemain tidak dapat mengeluarkan kartu apapun, maka ia harus mengambil satu kartu dari tumpukan teratas tumpukan bersama. Kartu yang diambil tersebut dapat dimainkan langsung jika memang dapat dimainkan. Jika tidak, maka permainan dilanjutkan ke giliran pemain lain.

Pemain yang telah memiliki banyak kartu di tangan sebanyak 1 harus mengucapkan "UNO" (artinya satu). Jika tertangkap pemain lain tidak melakukan hal ini, pemain tersebut harus menarik dua kartu dari tumpukan bersama.

Satu set permainan berakhir jika salah satu pemain berhasil menghabiskan/ memainkan semua kartu di tangannya. Poin pemain tersebut kemudian dihitung dengan mengakumulasikan nilai semua kartu yang masih tersisa di tangan semua pemain yang lain. Permainan kemudian akan dilanjutkan ke set selanjutnya.



Gambar 6 – Contoh tatanan permainan
(sumber: www.play-uno.com)



Gambar 7 – Contoh tatanan kartu tangan pemain
(sumber: www.play-uno.com)

Contoh tatanan permainan dapat dilihat pada Gambar 6 dan Gambar 7. Pada Gambar 6, terlihat bahwa kartu teratas pada tumpukan bersama/ *stock* adalah kartu *skip* kuning dan pada Gambar 7 terlihat semua kartu pada tangan pemain dengan kartu yang berwarna kontras menunjukkan kartu yang dapat dimainkan dan kartu berwarna buram menunjukkan kartu yang tidak dapat dimainkan. Perhatikan bahwa kartu 5 dan *skip* kuning dapat dimainkan karena berwarna sama dengan kartu teratas tumpukan. Kartu *skip* merah dapat dimainkan karena memiliki simbol yang sama dengan kartu teratas tumpukan. Dua jenis kartu *wildcard* dapat dimainkan kapan saja.

B. Algoritma Greedy

1. Prinsip Dasar

Algoritma *greedy* merupakan algoritma yang menerapkan prinsip *take what you can get now/* ambil apa yang dapat diambil sekarang. Algoritma ini biasanya digunakan untuk menemukan solusi atas permasalahan optimasi. Ada dua jenis masalah optimasi, yaitu maksimalisasi dan minimalisasi. Maksimalisasi bertujuan untuk mendapatkan nilai atau hasil maksimal dari suatu persoalan dan sebaliknya minimalisasi bertujuan untuk mendapatkan nilai atau hasil minimal dari suatu persoalan.

Algoritma *greedy* mengevaluasi pilihan langkah demi langkah. Dalam setiap langkah, akan diambil solusi yang merupakan keputusan optimum pada langkah tersebut. Keputusan optimum dalam setiap langkah akan mengarah ke nilai/ hasil optimum lokal yang jika diteruskan untuk setiap langkah selanjutnya, nilai/ hasil tersebut diharapkan akan mengarah menuju nilai/ hasil optimum global.

Nilai/ hasil optimum global dalam algoritma ini tidak selalu merupakan solusi optimum terbaik karena algoritma ini tidak mengevaluasi semua alternatif solusi yang ada dari suatu persoalan, melainkan hanya dicari solusi yang berupa taksiran. Oleh karena itu, algoritma ini sering digunakan untuk mencari solusi taksiran/ hampiran dari suatu masalah.

2. Elemen Algoritma

- Himpunan Kandidat, C

Himpunan ini berisi elemen pembentuk solusi

- Himpunan Solusi, S

Himpunan ini berisi elemen-elemen dari himpunan kandidat yang terpilih sebagai solusi permasalahan. S adalah subset dari C.

- Fungsi Seleksi

Fungsi yang digunakan untuk memilih keputusan yang akan diambil pada setiap langkah. Keputusan yang diambil tidak memperhatikan keputusan yang akan diambil pada langkah selanjutnya dan keputusan tersebut tidak dapat diubah lagi pada langkah selanjutnya. Keputusan yang diambil biasanya berupa nilai optimum lokal dari permasalahan.

- Fungsi Kelayakan

Fungsi yang memeriksa apakah kandidat yang terpilih dengan solusi yang sudah didapat pada langkah sebelumnya dapat memberikan solusi yang layak, yakni tidak melanggar kendala yang ada dalam permasalahan. Kandidat yang layak akan diambil dan dimasukkan ke himpunan solusi. Jika tidak, maka kandidat tidak dipertimbangkan.

- Fungsi Objektif

Fungsi yang mengoptimasi solusi.

3. Skema Umum Algoritma Greedy

```

function greedy(input C:
himpunan_kandidat) → himpunan_solusi
{Mengembalikan solusi dari persoalan
optimasi dengan algoritma greedy}
Deklarasi
x : kandidat
S : himpunan_kandidat
Algoritma
S ← {} {inisialisasi}
while (not SOLUSI(S) and C ≠ {}) do
  x ← SELEKSI(C)
  C ← C - {x}
  if LAYAK(S U {x}) then
    S ← S U {x}
  endif
endwhile
if SOLUSI(S) then
  return S {Didapat solusi greedy}
else
  output ("Tidak ada solusi")
endif

```

4. Contoh Persoalan yang Diselesaikan dengan Algoritma Greedy

1) Minimasi waktu tunggu

Persoalan

Sebuah server mempunyai n kerja yang harus dilayani/ dikerjakan. Waktu tunggu setiap kerja i adalah t_i . Diperlukan suatu langkah untuk meminimalkan waktu total kerja dan waktu tunggu untuk semua kerja T .

Contoh

Terdapat tiga kerja dengan waktu masing-masing 5, 7, dan 3 satuan waktu. Simpulkan urutan kerja mana yang harus dikerjakan oleh server agar waktu total kerja server minimal.

Urutan	T
1, 2, 3	$5 + (5+7) + (5+7+3) = 32$
1, 3, 2	$5 + (5+3) + (5+3+7) = 28$
2, 1, 3	$7 + (7+5) + (7+5+3) = 34$
2, 3, 1	$7 + (7+3) + (7+3+5) = 32$
3, 1, 2	$3 + (3+5) + (3+5+7) = 26$ (optimal)
3, 2, 1	$3 + (3+7) + (3+7+5) = 28$

Tabel 1 – Contoh persoalan waktu tunggu

Algoritma

- Urutkan kerja berdasarkan waktu kerja dengan urutan menaik.
- Pada setiap langkah, pilih pelanggan yang membutuhkan waktu terkecil di antara pelanggan lain yang belum dilayani

Elemen

C : $\{t_1, t_2, t_3\}$
 S : $\{\}$

Seleksi : kerja dengan waktu kerja terkecil
 Kelayakan : tidak ada
 Objektif : waktu kerja total minimum

Kesimpulan

Dengan algoritma *greedy* persoalan waktu tunggu ini selalu dapat memenuhi fungsi objektif/ mendapat solusi minimum.

2) Integer Knapsack

Persoalan

Terdapat sebuah kantong (*knapsack*) yang akan diisi dengan beberapa barang yang memiliki berat (*weight*) dan keuntungan (*profit*). *Knapsack* yang digunakan mempunyai kapasitas berat yang terbatas. Diperlukan suatu cara untuk memaksimalkan keuntungan dari barang-barang yang akan dimasukkan ke dalam *knapsack*.

Algoritma

C : Himpunan semua barang yang ada
 S : $\{\}$

Ada beberapa pendekatan algoritma *greedy* untuk persoalan *integer knapsack*.

1. Greedy by profit

Seleksi : barang dengan *profit* terbesar

2. Greedy by weight

Seleksi : barang dengan *weight* terkecil

3. Greedy by density

Seleksi : barang dengan rasio *profit/weight* terbesar

Contoh

Terdapat *knapsack* dengan kapasitas 100 dan terdapat beberapa barang sebagai berikut.

$w_1 = 100; p_1 = 40; w_2 = 50; p_2 = 35; w_3 = 45; p_3 = 18; w_4 = 20; p_4 = 4; w_5 = 10; p_5 = 10; w_6 = 5; p_6 = 5$

Properti Objek				Greedy by			Solusi Optimal
i	w_i	p_i	p_i/w_i	<i>profit</i>	<i>weight</i>	<i>density</i>	
1	100	40	0.4	1	0	0	0
2	50	35	0.7	0	0	1	1
3	45	18	0.4	0	1	0	1
4	20	4	0.2	0	1	1	0
5	10	10	1.0	0	1	1	0
6	5	2	0.4	0	1	1	0
Total Bobot				100	80	85	95
Total Keuntungan				40	34	51	55

Tabel 2 – Contoh persoalan *integer knapsack*

Angka 1/0 menunjukkan barang diambil/ tidak diambil

Kesimpulan

Ketika pendekatan algoritma *greedy* tersebut tidak dapat menemukan solusi optimal untuk masalah *integer knapsack*, tetapi dapat menghasilkan solusi hampiran terhadap solusi optimal tersebut.

III. PENERAPAN ALGORITMA GREEDY DALAM STRATEGI PERMAINAN

Tujuan permainan ini adalah menjadi pemain pertama yang mendapatkan poin sebanyak 500. Untuk mencapai tujuan ini, pemain harus memenangkan sebanyak mungkin set permainan. Selain itu, pemain juga harus berusaha meminimalkan poin yang didapat oleh pemain lain yang memenangkan set permainan dengan cara meminimalkan banyak kartu di tangan. Jadi, secara umum, pemain harus meminimalkan banyak kartu di tangan dalam satu set permainan atau bahkan menghabiskan semua kartu di tangan di akhir setiap set permainan.

Algoritma *greedy* akan diterapkan dalam setiap set permainan. Dari peraturan permainan, diketahui bahwa untuk mengeluarkan/ memainkan kartu dari tangan, pemain perlu mencocokkan kartu yang akan dimainkan dengan kartu teratas tumpukan bersama. Jadi, perlu adanya upaya agar selalu ada kartu yang 'cocok' di tangan pemain. Dalam suatu set permainan, pemain akan menghadapi kondisi ketika ada lebih dari satu kartu di tangan yang 'cocok'/ dapat dimainkan.

Elemen untuk algoritma *greedy* terhadap persoalan ini adalah sebagai berikut.

Kandidat : Himpunan kartu di tangan pemain.

Solusi : {}

Kelayakan :

Kecocokan kartu yang dimainkan dengan kartu teratas tumpukan bersama atau kartu yang dimainkan berjenis *wildcard*.

Dalam menentukan pilihan kartu yang akan dikeluarkan untuk meminimalkan, ada tiga pendekatan penerapan algoritma *greedy* yang akan dibahas penulis.

A. Greedy by Number

Prinsip dari pendekatan *greedy by number* adalah mengeluarkan kartu angka dengan angka yang paling banyak dikeluarkan dalam set permainan yang sama. Dengan cara demikian, diharapkan kartu dengan angka tersebut memiliki kemunculan yang lebih kecil di giliran/ putaran permainan selanjutnya.

Jadi, fungsi seleksinya adalah

Seleksi :

Kartu angka dengan angka yang paling banyak dikeluarkan dalam satu set permainan.

Sedangkan, fungsi objektifnya adalah

Objektif:

Meminimalkan banyak kartu di tangan pemain di akhir satu set permainan.

Diketahui bahwa ada empat kartu 0 dan delapan kartu untuk masing-masing angka 1 s.d. 9. Untuk menerapkan strategi ini, akan dibuat tabel yang berisi banyak kartu untuk

masing-masing kartu angka yang telah dimainkan dalam satu set permainan.

Angka	Banyak Keluaran
0	1
1	4
2	6
3	1
4	2
5	4
6	7
7	2
8	0
9	3

Tabel 3 – Contoh tabel *greedy by number*

Dapat dilihat pada Tabel 3 bahwa kartu angka 6 memiliki kemunculan terbanyak, yaitu sebanyak 7 kali. Maka, jika seorang pemain memiliki kartu angka 6 di tangan, dengan strategi ini, ia harus memainkan kartu angka 6 tersebut.

B. Greedy by Color

Prinsip dari pendekatan *greedy by color* mirip dengan pendekatan *greedy by number*. Pendekatan ini memiliki prinsip untuk mengeluarkan kartu dengan warna yang paling banyak dikeluarkan dalam satu set permainan yang sama. Dengan cara demikian, diharapkan kartu dengan warna tersebut memiliki kemunculan yang lebih kecil di giliran/ putaran permainan selanjutnya.

Jadi fungsi seleksinya adalah

Seleksi :

Kartu dengan warna yang paling banyak dikeluarkan dalam satu set permainan.

Sedangkan, fungsi objektifnya adalah

Objektif:

Meminimalkan banyak kartu di tangan pemain di akhir satu set permainan.

Diketahui bahwa terdapat 25 kartu untuk setiap warna kartu. Untuk menerapkan strategi ini, akan dibuat tabel banyak kartu yang telah dimainkan dalam satu set permainan.

Warna	Banyak Keluaran
Merah	10
Kuning	3
Hijau	15
Biru	13

Tabel 4 – Contoh tabel *greedy by color*

Dapat dilihat pada Tabel 4 bahwa kartu dengan warna hijau memiliki kemunculan terbanyak, yaitu sebanyak 15 kali. Maka dengan strategi ini, pemain berusaha untuk mengeluarkan kartu dengan warna hijau.

C. Greedy by Points

Untuk menerapkan pendekatan *greedy by points*, setiap jenis kartu dikelompokkan berdasarkan nilai yang dimiliki. Dalam permainan ini, kartu aksi dan kartu *wildcard* memiliki nilai yang jauh lebih tinggi dari kartu angka (kartu *wildcard*

bernilai 50, kartu aksi bernilai 20). Oleh karena itu, dalam strategi ini, pemain berusaha untuk mengurangi total nilai kartu pada tangan pemain dengan cara mengeluarkan jenis kartu dengan bobot tertinggi terlebih dahulu, yaitu kartu *wildcard*, kartu *aksi*, kemudian kartu angka. Dengan demikian diharapkan bahwa pada akhir permainan, pemain tersebut memiliki nilai total minimal pada kartu di tangan.

Jadi, fungsi seleksinya adalah

Seleksi:

Kartu dengan jenis kartu yang memiliki bobot minimum.

Sedangkan, fungsi objektifnya adalah

Objektif:

Meminimalkan total nilai kartu di tangan pemain pada akhir satu set permainan.

Berikut *pseudo-code* algoritma yang diterapkan sebagai strategi permainan ini.

```

function selectByNumber(K: himpunan_kartu) →
kartu
{Fungsi memilih kartu di tangan untuk
dimainkan sesuai prinsip greedy by number}

function selectByColor(K: himpunan_kartu) →
kartu
{Fungsi memilih kartu di tangan untuk
dimainkan sesuai prinsip greedy by color}

function selectByPoints(K: himpunan_kartu) →
kartu
{Fungsi memilih kartu di tangan untuk
dimainkan sesuai prinsip greedy by points}

procedure play()
{permainan satu set oleh satu pemain}
C ← drawCards() {Tarik kartu ke tangan}
S ← {}
{Himpunan kartu yang dimainkan pemain}
while not END_GAME and IS_PLAYER_TURN do
x ← selectCard()
{panggil fungsi select sesuai strategi
number, color, atau points}
if x != null then
S ← S ∪ {x}

```

IV. ANALISIS

Penerapan algoritma *greedy by number* sebagai strategi permainan UNO dapat mengurangi perputaran kartu angka dalam satu set permainan sehingga kemunculan kartu angka tertentu berkurang. Dengan berkurangnya kemunculan kartu angka tertentu, maka pemain terhindar dari keharusan untuk mencocokkan kartu dengan kartu angka tersebut. Selain itu, dengan strategi ini pemain menyimpan kartu aksi dan *wildcard* sebagai alternatif kartu keluaran terakhir yang dapat membantu pemain di pertengahan sampai akhir set permainan.

Penerapan algoritma *greedy by color* memiliki kesamaan sifat dengan algoritma *greedy by number* yang pada dasarnya mengurani kemunculan sebuah warna dalam permainan. Dibandingkan algoritma *greedy by number*, algoritma ini unggul karena diperlukan lebih sedikit pengelompokan (empat

warna kartu dibandingkan dengan sepuluh angka). Dengan lebih sedikit pengelompokan, pemain lebih mudah untuk menerapkan algoritma ini dalam permainan.

Penerapan algoritma *greedy by points* memiliki pendekatan berbeda dengan dua pendekatan algoritma *greedy* sebelumnya. Pendekatan *greedy by points* justru mengutamakan pemain untuk memainkan kartu dengan nilai tertinggi, yaitu kartu *wildcard*, kartu aksi, kemudian kartu angka. Pendekatan algoritma *greedy by points* juga dapat dikombinasi dengan pendekatan *greedy by number* dan *greedy by color*, yaitu ketika di tangan pemain tersisa kartu angka karena kartu angka memiliki nilai terendah.

V. KESIMPULAN

Penerapan algoritma *greedy* dalam strategi permainan UNO dapat menjadi salah satu alternatif strategi yang dipakai pemain. Kekurangan dari strategi yang diterapkan ini adalah bahwa strategi-strategi tersebut bersifat kaku dan tidak mempertimbangkan semua kemungkinan solusi yang ada. Selain itu, dalam penerapan algoritma *greedy* ini, masih banyak faktor permainan yang belum dipertimbangkan dalam strategi. Strategi-strategi yang diterapkan ini masih dapat dikombinasikan dengan strategi lain maupun dengan intuisi pemain dalam permainan.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang terlibat dan yang membantu kelancaran pembuatan makalah ini, terlebih kepada Dr. Ir. Rinaldi Munir, M.T. selaku pemberi tugas pembuatan makalah ini sekaligus selaku dosen mata kuliah IF2211 Strategi Algoritma.

DAFTAR PUSTAKA

- [1] <http://www.mattelgames.com/en-us/cards/uno> diakses pada Sabtu, 12 Mei 2018 11.00.
- [2] <http://www.thehouseofcards.com/games/uno.html> diakses pada Sabtu, 12 Mei 2018 11.30.
- [3] Munir, Rinaldi. "Diktat Kuliah IF2211 Strategi Algoritma". Bandung: Teknik Informatika ITB 2007.

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, 13 Mei 2018



Louis Leslie - 13516087