Penerapan Algoritma *Greedy* dalam Permainan Kartu Blade

Melita 13519063
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: teresamelita1501@gmail.com

***Abstrak*—Masalah zaman sekarang banyak diselesaikan dengan komputer menggunakan berbagai macam algoritma. Salah satu strategi algoritma yang digunakan adalah algoritma *greedy* yang biasa digunakan untuk masalah optimasi, yaitu memaksimasi atau meminimasi sesuatu. Blade adalah sebuah permainan kartu dalam *game-game* Nihon Falcom yang pemenangnya ditentukan dari skor tertinggi antara dua pemain. Makalah ini akan membahas penerapan algoritma *greedy* yang dapat digunakan untuk memenangkan permainan kartu tersebut.**

Kata kunci—algoritma greedy; blade; nihon falcom; permainan kartu

I. PENDAHULUAN

Zaman sekarang, komputer sering kali digunakan untuk menyelesaikan masalah dalam berbagai bidang. Hal ini dapat dilakukan dengan menggunakan algoritma, yaitu urutan langkah untuk memecahkan persoalan. Terdapat bermacam-macam strategi algoritma, di antaranya adalah algoritma *brute-force*, *greedy*, *divide and conquer*, *decrease and conquer*, *backtracking*, *branch and bound*, dan *dynamic programming*. Makalah ini akan membahas aplikasi dari algoritma *greedy* untuk memenangkan sebuah permainan kartu bernama Blade.

II. DASAR TEORI

A. Algoritma Greedy

Algoritma *greedy* adalah algoritma yang sering digunakan untuk memecahkan persoalan optimasi, yaitu persoalan tentang memaksimasi atau meminimasi sesuatu. Algoritma ini memecahkan persoalan langkah demi langkah sehingga pada setiap langkahnya akan dipilih satu langkah terbaik (optimum lokal) dengan harapan akhirnya akan mencapai optimum global. Beberapa contoh persoalan yang dapat diselesaikan dengan algoritma *greedy* antara lain sebagai berikut.

1. Penukaran uang *(coin exchange)*
2. Memilih aktivitas *(activity selection)*
3. Minimisasi waktu dalam sistem
4. Persoalan *knapsack*
5. Penjadwalan *job* dengan tenggat waktu
6. Pohon merentang minimum
7. Lintasan terpendek
8. Kode Huffman
9. Pecahan Mesir *(Egyptian fraction)*

Algoritma *greedy* juga memiliki beberapa elemen yang dijelaskan sebagai berikut.

1. Himpunan kandidat (C), yaitu himpunan yang berisi kandidat solusi yang dapat dipilih pada setiap langkah.
2. Himpunan solusi (S), yaitu himpunan yang berisi kandidat yang sudah dipilih.
3. Fungsi solusi, yaitu fungsi yang menentukan apakah himpunan kandidat yang dipilih memberikan solusi.
4. Fungsi seleksi, yaitu fungsi yang digunakan untuk memilih kandidat berdasarkan strategi *greedy* tertentu.
5. Fungsi kelayakan, yaitu fungsi yang memeriksa apakah kandidat yang dipilih dapat dimasukkan ke himpunan solusi.
6. Fungsi objektif, yaitu fungsi yang menyatakan hal yang dimaksimumkan atau diminimumkan.

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

Algoritma:
  S ← {}      { inisialisasi 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 1. Skema Umum Algoritma *Greedy*

(Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf)

Walaupun algoritma ini memilih optimum lokal di setiap langkahnya, solusi yang didapat belum tentu merupakan solusi terbaik. Hal ini terjadi karena algoritma *greedy* tidak memeriksa semua kemungkinan solusi seperti algoritma *brute-force*. Akan tetapi, algoritma dapat digunakan untuk menghasilkan solusi hampiran karena membutuhkan waktu yang lebih singkat daripada algoritma *brute-force.*

B. Blade

Blade (ブレード) adalah sebuah permainan kartu yang muncul sebagai *minigame* pada beberapa *game* Nihon Falcom. Beberapa *game* tersebut adalah Brandish VT, Brandish: The Dark Revenant, dan Trails of Cold Steel. Versi kedua dari Blade, disebut Blade II, muncul dalam *game* Trails of Cold Steel II dan Tokyo Xanadu dengan perbedaan beberapa jenis kartu tambahan dari Blade. *Minigame* ini dapat dimaikan pada beberapa poin dalam *game-game* tersebut, dengan salah satu bonus yang didapat dari memainkannya adalah *link experience* dengan lawan.

Beberapa istilah yang digunakan dalam Blade adalah sebagai berikut.

1. *Deck*, yaitu kartu milik pemain yang tidak dapat dilihat oleh pemiliknya sendiri. Setiap pemain memiliki tepat satu *deck*, dan satu *deck* berisi antara 5 buah kartu (Blade) sampai 6 buah kartu (Blade II).
2. *Hand*, yaitu kartu milik pemain yang dapat dilihat oleh pemiliknya. Setiap pemain memiliki tepat satu *hand* yang masing-masing berisi 10 kartu yang diambil di awal permainan. Kartu *hand* seorang pemain tidak dapat dilihat oleh lawannya.
3. *Field*, yaitu tempat meletakkan kartu yang sudah digunakan. Skor pemain pada saat tertentu dapat dihitung dari semua kartu miliknya di *field*. Jika pada suatu ronde dicapai nilai skor yang seri, *field* akan dikosongkan sehingga skor kedua pemain kembali menjadi nol.

Gambar 2. *Minigame* Blade dalam Trails of Cold Steel

(Sumber: Tangkapan layar *game*)

Permainan Blade memiliki beberapa jenis kartu yang dapat dijelaskan dengan tabel berikut.

TABEL I. JENIS KARTU DALAM BLADE

Nama dan Gambar Kartu	Deskripsi	Jumlah dalam Set
 Rod atau 1	Memiliki 2 fungsi. Sebagai Rod: Membalikkan kartu yang dibalik oleh Bolt, terjadi jika digunakan setelah lawan menggunakan Bolt. Kartu Rod tidak ditambah ke *field.* Sebagai 1: Menambah skor pemain sebanyak 1 poin.	2
 2	Menambah skor pemain sebanyak 2 poin	3 (Blade) 5 (Blade II)
 3	Menambah skor pemain sebanyak 3 poin	4 (Blade) 5 (Blade II)
 4	Menambah skor pemain sebanyak 4 poin	4 (Blade) 5 (Blade II)
 5	Menambah skor pemain sebanyak 5 poin	4
 6	Menambah skor pemain sebanyak 6 poin	3

 7	Menambah skor pemain sebanyak 7 poin	2
 Bolt	Kartu spesial. Membalikkan kartu terakhir yang digunakan lawan pada *field*, sehingga skor lawan berkurang sebanyak kartu tersebut. Kartu Bolt tidak ditambah ke *field.*	6 (Blade) 4 (Blade II)
 Mirror	Kartu spesial. Menukar semua kartu *field* pemain dengan semua kartu *field* lawan sehingga skor akan tertukar. Kartu Mirror tidak ditambah ke *field.*	4
 Blast	(Khusus Blade II) Kartu Spesial. Menghapus 1 kartu dari *hand* milik lawan. Kartu Blast tidak ditambah ke *field.*	0 (Blade) Tidak diketahui (Blade II)
 Force	(Khusus Blade II) Kartu spesial. Menggandakan skor total kartu pemain yang ada di *field* saat itu. Kartu Force ditambah ke *field* dan dapat dibalik dengan Bolt.	0 (Blade) Tidak diketahui (Blade II)

(Sumber gambar: https://kiseki.fandom.com/wiki/Blade)

Tata cara permainan Blade adalah sebagai berikut.

1. Pertama, seluruh kartu (30 untuk Blade dan 32 untuk Blade II) akan diacak dan dibagi kepada kedua pemain.
2. Lalu, setiap pemain mengambil 10 kartu dari kartu yang dibagikan tersebut sebagai *hand* dan sisanya menjadi *deck.*
3. Kemudian, kedua pemain mengembil satu kartu dari *deck* masing-masing untuk diletakkan pada *field.* Pada tahap ini, jika diambil kartu spesial (Bolt, Mirror, Force, Blast) dari *deck*, maka kartu tersebut dianggap bernilai 1. Pemain dengan nilai kartu lebih rendah akan memulai gilirannya terlebih dahulu.
4. Jika terjadi skor seri, maka *field* dibersihkan dan kedua pemain mengambil satu kartu lagi dari *deck* untuk menentukan giliran selanjutnya. Jika *deck* kosong, pemain dapat memilih satu kartu dari *hand.* Jika *hand* juga kosong, permainan dinyatakan berakhir seri.
5. Jika skor setelah langkah ketiga tidak sama, pemain yang memiliki giliran memiliki kesempatan untuk menggunakan satu kartu pilihannya dari *hand.* Jika kartu yang dipilih tidak menyebabkan skor total pemain tersebut melebihi atau sama dengan pemain lawan, atau jika pemain tidak lagi memiliki kartu untuk dimainkan, pemain akan kalah. Jika total skor *field* menjadi sama, *field* dibersihkan sesuai aturan keempat. Jika skor *field* pemain menjadi melebihi lawan, pemain lawan akan mendapat giliran selanjutnya.
6. Permainan berlanjut sampai salah satu pemain kalah atau permainan mencapai akhir seri.
7. Jika pemain memainkan kartu spesial (Bolt, Mirror, Force, Blast) sebagai kartu terakhirnya, pemain tersebut dinyatakan kalah karena *foul.*
8. Jika pemain menggunakan kartu Blast (khusus untuk Blade II), pemain mendapat giliran lagi karena kartu Blast tidak mengubah skor pemain tersebut.

III. PEMBAHASAN

Permainan yang akan dibahas strategi *greedy*-nya pada makalah ini adalah Blade yang hanya memiliki 2 kartu spesial, yaitu Bolt dan Mirror. Jumlah kartu dan pembagiannya tidak dimasukkan ke dalam pertimbangan karena jumlah kartu yang tidak pasti dan berbeda-beda antara versi *game* seperti yang dapat dilihat pada Tabel I. Selain jumlah kartu yang dituliskan pada tabel, permainan Blade II versi Tokyo Xanadu memiliki jumlah kartu yang berbeda juga, yaitu 4 untuk masing-masing kartu.

Dengan memanfaatkan elemen-elemen algoritma *greedy* yang telah dijelaskan sebelumnya, masalah permainan Blade dapat diuraikan sebagai berikut.

1. Himpunan kandidat: semua kartu yang ada dalam *hand* pemain.
2. Himpunan solusi: kartu yang sudah dipilih pada tiap langkah algoritma.
3. Fungsi solusi: memeriksa apakah pemain memiliki skor akhir yang lebih tinggi dari lawan.
4. Fungsi seleksi: strategi *greedy* yang digunakan, akan dijelaskan kemudian.
5. Fungsi kelayakan: memeriksa apakah kartu yang dipilih ada dalam *hand* pemain.
6. Fungsi objektif: *cost* kartu yang digunakan untuk menang dalam tiap langkah minimum.

Untuk menguji strategi *greedy* yang akan dibuat, penulis membuat simulasi sederhana permainan Blade menggunakan Python 3. Karena jumlah kartu tidak ditentukan, persebaran kartu pemain dimasukkan secara manual ke dalam program,

dan kartu yang dipilih oleh lawan juga perlu dimasukkan secara manual. Strategi *greedy* akan digunakan untuk pemilihan kartu mana yang harus digunakan oleh pemain saat mendapat giliran. Berikut adalah strategi *greedy* yang digunakan dalam bahasa Python.

```
# ALGORITMA GREEDY
def greedy():
    # Algoritma greedy yang digunakan
    # Perlu set manual_input menjadi False
    global hand
    global field_player
    global field_enemy
    global last_player
    global last_enemy
    global player_bolted
    global enemy_bolted

    nBolt = hand.count("bolt")
    nMirror = hand.count("mirror")
    nRod = hand.count("1")
    deltaSkor = field_enemy - field_player

    if(player_bolted and nRod>0):
        # Jika player baru terkena bolt
        # Gunakan kartu Rod jika ada
        return "1"

    if(deltaSkor>4 and nMirror>0):
        # Jika perbedaan skor > 4
        # Gunakan mirror jika ada
        return "mirror"

    if(last_enemy == "6" or last_enemy == "7"):
        # Jika lawan kartu terakhirnya 6 atau 7
        # Gunakan Bolt jika ada
        if(nBolt>0):
            return "bolt"

    if(nBolt+nMirror==len(hand)-1):
        # Jika kartu non-spesial tinggal 1
        # Gunakan kartu spesial terlebih dahulu
        # jika masih ada
        if(nMirror>0):
            return "mirror"
        if(nBolt>0):
            return "bolt"

    # Pilih kartu yang menjadikan skor
    # player sesedikit mungkin di atas musuh
    choose = deltaSkor + 1
    while(choose<=7):
        nCard = hand.count(str(choose))
        if(nCard>0):
            return str(choose)
        choose = choose + 1

    if(nBolt>0):
        return "bolt"
    if(nMirror>0):
        return "mirror"

    # Jika tidak ada kartu memenuhi,
    # gunakan kartu pertama pada list
    if(len(hand)>0):
        return hand[0]

    # Jika kartu sudah habis:
    return "none"
```

Terdapat beberapa variabel yang digunakan untuk menyimpan data permainan. Hand berisi larik seluruh kartu yang dimiliki oleh pemain pada saat itu, field_player berisi total skor pemain sekarang, field_enemy berisi total skor lawan sekarang, last_player berisi kartu terakhir yang dimainkan oleh pemain, last_enemy berisi kartu terakhir yang dimainkan oleh lawan, player_bolted berisi status jika pemain terkena Bolt dari lawan pada giliran sebelumnya, dan enemy_bolted berisi status jika lawan terkena Bolt pada giliran sebelumnya dari pemain.

Di dalam fungsi *greedy* yang dibuat juga dibuat beberapa variabel baru sebagai perhitungan langkah yang akan dipilih. nBolt berisi jumlah Bolt yang ada pada hand, nMirror berisi jumlah Mirror pada hand, nRod berisi jumlah kartu 1 atau Rod pada hand, dan terakhir deltaSkor berisi selisih jumlah skor lawan dan skor pemain saat ini.

Dari peraturan permainan Blade yang telah dijelaskan, dapat disimpulkan nilai dari setiap kartu. Nilai kartu dalam kasus ini didefinisikan sebagai seberapa besar perbedaan poin yang disebabkan oleh penggunaan kartu tersebut. Kartu biasa bernilai 2 sampai 7 memiliki nilai sesuai namanya. Sementara itu, kartu 1 atau Rod memiliki nilai yang berkisar dari 1 sampai 7 yang bergantung dari penggunaan kartu tersebut. Jika digunakan sebagai kartu biasa, kartu tersebut akan bernilai 1, sedangkan jika digunakan sebagai Rod yang membalikkan kartu, kartu tersebut dapat bernilai 1 hingga 7 bergantung pada kartu yang terkena Bolt pada giliran sebelumnya. Begitu juga dengan kartu spesial Bolt dan Mirror. Kartu Bolt memiliki nilai dari 1 hingga 7 bergantung pada kartu yang terkena Bolt. Kartu Mirror memiliki nilai yang sama dengan selisih skor pemain dan lawan pada saat digunakan.

Dengan panduan hasil pengamatan tersebut, dibuatlah sebuah algoritma *greedy* untuk memenangkan permainan Blade dengan langkah-langkah sebagai berikut.

1. Cek apakah lawan menggunakan Bolt pada giliran sebelumnya. Jika ya, cek apabila pemain memiliki kartu 1 atau Rod. Jika ya, gunakan kartu tersebut. Hal ini dilakukan karena nilai kartu tersebut akan sama dengan atau lebih dari 1 jika digunakan sebagai Rod dengan *cost* yang sama jika digunakan sebagai kartu biasa.
2. Cek apakah deltaSkor lebih dari 4 dan apakah pemain memiliki Mirror. Jika ya untuk keduanya, gunakan Mirror. Hal ini dilakukan karena untuk *cost* yang sama, yaitu 1 kartu Mirror, nilai kartu tersebut bergantung pada selisih skor pemain dan lawan seperti yang dijelaskan sebelumnya. Selisih lebih dari 4 dianggap sudah cukup besar untuk mendapat keuntungan maksimal dari kartu tersebut.
3. Cek apakah pada giliran terakhir, lawan menggunakan kartu 6 atau 7. Jika ya, cek apakah pemain memiliki Bolt. Jika ya, gunakan kartu Bolt. Hal ini dilakukan karena nilai kartu Bolt bergantung pada kartu yang terkena Bolt, yaitu kartu terakhir yang dimainkan oleh lawan.
4. Cek apakah pemain hanya memiliki 1 kartu normal tersisa. Jika ya, coba untuk menggunakan seluruh kartu spesial yang dimiliki. Hal ini dilakukan karena jika

kartu spesial digunakan pada giliran terakhir, pemain akan kalah karena *foul.*

5. Pilih kartu yang akan membuat skor pemain lebih tinggi dari lawan, tetapi dengan selisih sesedikit mungkin. Hal ini dilakukan untuk antisipasi lawan yang menggunakan Bolt atau Mirror pada kartu pemain, sehingga menggunakan kartu yang bernilai besar atau menyebabkan selisih besar pada awal permainan kurang ideal.
6. Jika tidak ada kartu biasa dengan nilai yang memenuhi, gunakan kartu spesial Bolt atau Mirror.
7. Jika masih tidak ada kartu yang memenuhi, gunakan kartu apa pun yang tersisa. Jika kartu habis, dikembalikan nilai "none" yang berarti pemain tidak bisa melakukan giliran lagi dan kalah.

Setelah algoritma *greedy* yang akan digunakan selesai, dilakukan pengujian terhadap algoritma tersebut dalam permainan. Pada makalah ini, penulis menggunakan Blade versi Trails of Cold Steel sebagai bahan uji masukan pada program. Keluaran program yang dihasilkan adalah sebagai berikut.

```
Masukkan kartu pada hand: 1 2 2 4 4 4 5 7 bolt bolt
Masukkan kartu field pertama pemain: 4
Masukkan kartu field pertama lawan : 3
Skor pemain: 4
Skor lawan : 3
Kartu hand pemain:
['1', '2', '2', '4', '4', '4', '5', '7', 'bolt',
'bolt']

Masukkan kartu lawan: 3
Skor pemain: 4
Skor lawan : 6
Kartu hand pemain:
['1', '2', '2', '4', '4', '4', '5', '7', 'bolt',
'bolt']
Kartu player yang dipilih: 4
Skor pemain: 8
Skor lawan : 6
Kartu hand pemain:
['1', '2', '2', '4', '4', '5', '7', 'bolt', 'bolt']

Masukkan kartu lawan: 5
Skor pemain: 8
Skor lawan : 11
Kartu hand pemain:
['1', '2', '2', '4', '4', '5', '7', 'bolt', 'bolt']
Kartu player yang dipilih: 4
Skor pemain: 12
Skor lawan : 11
Kartu hand pemain:
['1', '2', '2', '4', '5', '7', 'bolt', 'bolt']

Masukkan kartu lawan: 6
Skor pemain: 12
Skor lawan : 17
Kartu hand pemain:
['1', '2', '2', '4', '5', '7', 'bolt', 'bolt']
Kartu player yang dipilih: bolt
Skor pemain: 12
Skor lawan : 11
Kartu hand pemain:
['1', '2', '2', '4', '5', '7', 'bolt']

Masukkan kartu lawan: 1
```

```
Skor pemain: 12
Skor lawan : 17
Kartu hand pemain:
['1', '2', '2', '4', '5', '7', 'bolt']
Kartu player yang dipilih: bolt
Skor pemain: 12
Skor lawan : 11
Kartu hand pemain:
['1', '2', '2', '4', '5', '7']

Masukkan kartu lawan: bolt
Skor pemain: 8
Skor lawan : 11
Kartu hand pemain:
['1', '2', '2', '4', '5', '7']
Kartu player yang dipilih: 1
Skor pemain: 12
Skor lawan : 11
Kartu hand pemain:
['2', '2', '4', '5', '7']

Masukkan kartu lawan: bolt
Skor pemain: 8
Skor lawan : 11
Kartu hand pemain:
['2', '2', '4', '5', '7']
Kartu player yang dipilih: 4
Skor pemain: 12
Skor lawan : 11
Kartu hand pemain:
['2', '2', '5', '7']

Masukkan kartu lawan: mirror
Skor pemain: 11
Skor lawan : 12
Kartu hand pemain:
['2', '2', '5', '7']
Kartu player yang dipilih: 2
Skor pemain: 13
Skor lawan : 12
Kartu hand pemain:
['2', '5', '7']

Masukkan kartu lawan: mirror
Skor pemain: 12
Skor lawan : 13
Kartu hand pemain:
['2', '5', '7']
Kartu player yang dipilih: 2
Skor pemain: 14
Skor lawan : 13
Kartu hand pemain:
['5', '7']

Masukkan kartu lawan: mirror
Skor pemain: 13
Skor lawan : 14
Kartu hand pemain:
['5', '7']
Kartu player yang dipilih: 5
Skor pemain: 18
Skor lawan : 14
Kartu hand pemain:
['7']

Masukkan kartu lawan: 6
Skor pemain: 18
Skor lawan : 20
Kartu hand pemain:
['7']
Kartu player yang dipilih: 7
Skor pemain: 25
Skor lawan : 20
```

```
Kartu hand pemain:
[]

Masukkan kartu lawan: none

[Player menang]
```

Gambar 3. Hasil Pengujian Strategi *Greedy* pada *Game*

(Sumber: Tangkapan layar *game*)

Setelah pengujian tersebut, dilakukan 2 kali lagi pengujian dengan hasil pemain berhasil menang. Kartu *hand* yang didapatkan pemain pada tes kedua adalah 3 3 4 4 4 5 6 Bolt Bolt Mirror, sedangkan kartu *hand* pada pengujian ketiga adalah 2 3 3 4 4 5 5 7 Bolt Mirror. Pengujian kedua berakhir dengan skor 17-16, sedangkan pengujian ketiga berakhir dengan skor 16-12 (16-14 jika kartu terakhir lawan dihitung).

Gambar 4. Hasil Pengujian Ketiga Strategi *Greedy* pada *Game*

(Sumber: Tangkapan layar *game*)

IV. KESIMPULAN

Ada berbagai strategi algoritma yang dapat digunakan untuk memecahkan masalah, salah satunya adalah algoritma *greedy* yang biasa digunakan untuk memecahkan persoalan-persoalan optimasi. Algoritma *greedy* dapat digunakan untuk memenangkan permainan kartu Blade dengan mencari langkah terbaik pada setiap gilirannya. Pada makalah ini, digunakan algoritma *greedy* yang meminimasi *cost* sambil memaksimasi nilai dari kartu pada setiap langkahnya. Berdasarkan beberapa hasil pengujian yang dilakukan, algoritma *greedy* yang dibuat sudah cukup efektif dalam memenangkan permainan tersebut.

LINK VIDEO YOUTUBE

Gambaran lebih jelas tentang makalah ini dapat dilihat dalam bentuk video pada *link* https://youtu.be/aSm38CytqGc.

UCAPAN TERIMA KASIH

Puji syukur dan terima kasih penulis ucapkan kepada Tuhan Yang Maha Esa karena atas segala berkat-Nya penulis dapat menyelesaikan tugas makalah ini. Penulis juga mengucapkan terima kasih kepada seluruh dosen IF2211 Strategi Algoritma karena telah atas bimbingannya dalam kegiatan akademik. Terakhir, penulis juga mengucapkan terima kasih kepada keluarga dan teman-teman penulis yang telah mendukungnya selama pembuatan makalah.

REFERENSI

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf diakses pada 10 Mei 2021 pukul 12.41 WIB.

[2] https://kiseki.fandom.com/wiki/Blade diakses pada 10 Mei 2021 pukul 14.07 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, 10 Mei 2021

Melita 13519063