Implementasi Algoritma *Greedy* Dalam Menentukan Langkah Pergerakan Bidak Ludo

Fikri Ihsan Fadhiilah 13520148
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13520148@std.stei.itb.ac.id

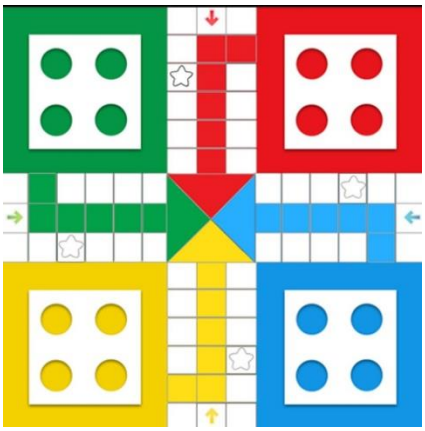
Abstract—Ludo adalah sebuah adalah sebuah permainan papan yang dapat dimainkan dua atau empat orang. Dalam permainan Ludo ini, setiap pemain berlomba untuk memasukkan bidak ke garis *finish*. Algoritma *greedy* akan digunakan diharapkan dapat membawa bidak bidak pemain ke garis finish dengan waktu yang lebih cepat atau giliran paling sedikit. Algoritma *greedy* yang digunakan ada tiga yaitu *greedy by hitter, greedy by no one left behind, dan greedy by quick move.*

Keywords—greedy,Ludo,greedy by hitter, greedy by no one left behind, dan greedy by quick move

I. PENDAHULUAN

Algoritma *greedy* adalah algoritma yang sangat baik dalam menyelesaikan sebuah masalah dengan pendekatan mencari nilai keuntungan maksimum yang didapatkan dari setiap langkah. Salah satu permainan yang dapat diselesaikan dengan algoritma *greedy* ini adalah Ludo. Ludo dapat dimainkan oleh dua atau empat pemain dengan tujuan Akhir memasukkan bidak bidak setiap pemain ke dalam garis *finish*.

Permainan ludo ini membutuhkan strategi yang digunakan untuk menentukan pergerakan pada setiap giliran pemain. Walaupun memiliki faktor keberuntungan dalam penentuan angka dadu, namun jika kita salah dalam mengambil langkah pada saat angka dadu itu muncul kita bisa mendapatkan kerugian.



Gambar 1.1 : Logo Genshin Impact
Sumber:
https://www.kompasiana.com/asih88904/5ebc7366097f361dd21dd0d7/pembelajaran-matematika-menggunakan-permainan-ludo-dan-spingo.svg

II. LANDASAN TEORI

A. Algoritma Greedy

Algoritma Greedy merupakan metode yang paling populer dan sedehana untuk memecahkan persoalan optimasi. Dalam pengertian lain yaitu algoritma yang memecahkan persoalan secara langkah per langkah sedemikian sehingga, pada setiap langkah :

1. Mengambil pilihan yang terbaik pada saat itu tanpa memperhatikan apa yang akan terjadi kedepannya.
2. Dengan memilihin optimum lokal, kita berharap mencapai optimum global.

Terdapat dua macam persoalan optimasi , yaitu maksimasi dan minimasi.

Elemen elemen yang ada pada algoritma *greedy* :

1. Himpunan kandidat (C)

   Himpunan kandidat adalah himpunan dengan elemen pembentuk solusi

2. Himpunan solusi (S)

   Himpunan Solusi merupakan himpunan yang berisi kandidat-kandidat yang terpilih sebagai solusi persoalan

3. Fungsi Solusi

   Fungsi Solusi digunakan untuk menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi

4. Fungsi Seleksi

   Fungsi seleksi adalah fungsi untuk memilih kandidat yang paling memungkinkan untuk mencapai solusi optimal. Namun kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

5. Fungsi Kelayakan

   Fungsi kelayakan merupakan fungsi untuk memeriksa kelayakan dari solusi yang diberikan oleh suatu kandidat yang telah dipilih atau dengan kata lain, kandidat yang telah dipilih setelah ditambahkan pada

himpunan solusi tidak akan melanggar kendala yang ada. Kandidat yang layak, akan dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak akan dibuang dan tidak pernah dipertimbangkan lagi

6. Fungsi Objektif

   Fungsi obyektif merupakan fungsi yang memaksimumkan atau meminimumkan nilai solusi

Berikut ini adalah beberapa contoh persoalan yang dapat diselesaikan dengan Algoritma *greedy* :

1. Persoalan penukaran uang
2. Persoalann memilih aktivitas
3. Minimasi 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

```
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 2.1 : Skema Umum Algoritma Greedy
Sumber:
Informatika Rinaldi Munir

Dalam beberapa kasus, strategi menggunakan Algoritma *greedy* tidak menghasilkan solusi optimum, tetapi dengan *greedy* kita bisa mencapai solusi optimum lokal yang dimana solusi tersebut dapat mengarahkan kita ke solusi optimum global. Pada setiap Langkah dengan Algoritma *greedy* kita harus membuat keputusan yang terbaik tanpa harus memperhatikan konskuensi yang akan datang. Ada empat Algoritma *greedy* yang akan diterapkan dalam menentukan langkah yang tepat pada permainan Ludo ini yaitu :

1. *Greedy by hitter*

   Bidak pemain pada giliran tersebut akan berjalan mengikuti peta sesuai dengan angka yang muncul pada dadu dengan pertimbangan apabila ada bidak pemain lain pada petak langkah yang akan di tempuh, maka bidak akan melempat bidak pemain lain tersebut kembali ke tempat asal.

2. *Greedy by No One Left Behind*

   Apabila bidak pemain berada pada urutan paling belakang dan angka dadu yang muncul adalah 5 atau 6, maka bidak tersebut akan menjadi prioritas untuk melakukan pergerakan

3. *Greedy by quick move*

   Jika bidak pemain berada pada posisi yang memungkinkan untuk berada pada daerah aman akan menjadi prioritas untuk melakukan pergerakan sesuai dengan angka dadu yang muncul

4. *Greedy by blocking*

   Jika didepan bidak yang ingin pemain jalankan Terdapat bidak lain dengan warna sama dan mendapatkan dadu yang tepat untuk bergabung pada petak bidak yang sama tersebut, maka akan dilakukan *greedy by blocking*.

Keuntungan penggunaan Algoritma Greedy :

1. Sederhana

   Algoritma *greedy* lebih mudah dipahami dan diterapkan dibandingkan dengan Algoritma lain

2. Efisien

   Dengan adanya konsep optimum lokal dan optimum global membuat Algoritma ini lebih efisien dan mudah menganalisis waktu yang dibutuhkan sebuah Algoritma *greedy*.

Kerugian penggunaan Algoritma Greedy :

1. Sulit dibuktikan

   Sebuah persoalan sulit untuk dibuktikan benar atau salah algoritma *greedy* dalam memecahkan persoalan

Berikut contoh pengaplikasian dari algoritma *greedy* pada persoalan penukaran uang. Pada persoalan uang, diberikan sebuah uang senilai A dan uang tersebut akan ditukarkan menggunakan koin koin uang dengan nominal-nominal tertentu, akan dicari Jumlah minimum koin dibutuhkan untuk menukar uang A. Berikut penyelesaian secara *greedy* :

1. A bernilai 32
2. Koin koin yang tersedia yaitu 1,5,10,dan 25
3. Strategi *greedy* yang diterapkan, pada setiap langkah pilihlah koin dengan nilai terbesar dari himpunan koin yang tersisa.
4. Contoh :

   Langkah 1 : pilih 1 buah koin 25 (total = 25)

   Langkah 2 : pilih 1 buah koin 5 (total = 30)

   Langkah 3 : pilih 1 buah koin 1 (total = 31)

   Langkah 4 : pilih 1 buah koin 1 (total = 32)

Solusi Jumlah koin minimal yang didaptkan sejumlah 4 koin. (Solusi Optimal)

5. Elemen elemen pada algoritma *greedy* :

   a. Himpunan kandidat

      Himpunan koin atau pecahan uang yang merepresentasikan nilai 1,5,10,25 paling sedikit mengandung satu koin untuk setiap nilai

   b. Himpunan solusi

      Koin koin yang terpilih

   c. Fungsi solusi

      Memeriksa apakah total nilai koin yang dipilih tepat sama jumlahnya dengan nilai uang A

   d. Fungsi seleksi

      Pilihlah koin yang bernilai tertinggi dari himpunan koin yang tersisa

   e. Fungsi kelayakan

      Fungsi yang memeriksa apakah koin yang baru dipilih apabila dijumlahkan dengan semua koin yang sudah ada berada dalam himpunan, tidak melebihi Jumlah dari uang A

   f. Fungsi obyektif

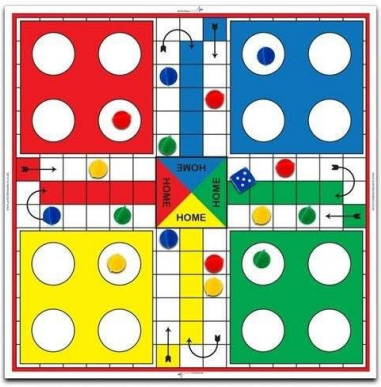
      Fungsi untuk menghasilkan solusi dengan Jumlah koin yang ditukarkan dengan seminimum mungkin.

Solusi dari persoalan penukaran uang dengan algoritma *greedy* diatas menghasilak solusi optimum dengan jumla minimal koin yang diperlukan untuk menukar uang sejumlah A adalah 4 buah koin.

B. *Board game* Ludo

Ludo adalah salah satu dari banyak *Board game* yang dapat dimainkan sebanyak 2-4 orang, dimana setiap pemain harus memindahkan semua bidak dari garis *start* ke garis *finisih* dengan waktu secepat mungkin.

Board game ludo memiliki bentuk peta persegi, disetiap sudut papan, terdapat rumah setiap pemain / garis *start* . Setiap pemain biasanya dibedakan dengan warna yang berbeda yaitu biru,merah,kuning, dan hijau. Setiap rumah, pemain memiliki 4 bidak yang harus dijalankan hingga garis *finish*.



Gambar 2.2 : Arah permainan Ludo
Sumber:
https://www.indiamart.com/proddetail/ludo-board-23255003191.html

Gambar 2.3 : Kondisi masuk garis *finish*
Sumber:
https://www.amazon.co.uk/KandyToys-Ludo-Traditional-Board-Game/dp/B07WH5HRPP

Arah langkah pada bidak harus mengikuti petak-petak yang sudah tersedia pada papan ludo. Pergerakan bidak mengikuti arah jarum jam yang ditandai dengan arah panah pada gambar 2.2 diatas. Petak “home” pada gambar 2.2 menunjukan garis *finish* yang harus dicapai seperti gambar 2.3 dimana salah satu bidak pemain berwarna biru sudah memasuki garis *finish*.

C. Peraturan permainan Ludo

Setiap pemain akan mendapatkan 4 bidak pada awal permainan dan akan berjalan sesuai dengan petak yang ada untuk mencapat petak *finish* pada tengah-tengah papan. Peraturan Ludo umumnya sebagai berikut :

1. Setiap pemain diberi kebebasan untuk memilih daerah yang tersedia di ujung ujung papan permainan ludo
2. Pada awal permainan, para pemain menentukan urutan gerak setiap pemainnya.
3. Setiap giliran diawali dengan pelemparan dadu oleh pemain.
4. Untuk mengeluarkan bidak setiap pemain dari garis *start* untuk menuju petak petak jalur permainan, pelemparan dadu pemain tersebut harus mengahasilkan angka 6. Selain itu, saat pemain mendapatkan dadu dengan angka 6, pemain dapat diperbolehkan memilih untuk menjalankan bidak yang sudah pada jalur permainan atau mengeluarkan bidak ke jalur permainan.
5. Jika pelemparan dadu pada pemain menghasilkan angka selain 6, maka pemain tersebut hanya bisa menjalankan bidak yang sudah ada pada petak jalur permainan dan di pindahkan sesuai angka yang muncul pada dadu tersebut. Jika pemaint tersebut tidak memiliki bidak pada petak jalut permainan, maka giliran akan dilanjutkan ke pemain berikutnya tanpa melakukan aksi lain.
6. Saat pelemparan dadu dan pemain mendapatkan angka dadu 6, maka pemain tersebut mendapatkan kesempatan tambahan untuk bermain lagi untuk kedua kalinya. Jika pemain tersebut mendapatkan angka 6

pada dadu lagi, maka giliran akan dilanjutkan ke pemain berikutnya karena pemain tersebut telah mendapatkan angka dadu 6 sebanyak 3x secara berurutan.

7. Penempatan bidak yang sudah keluar dari garis *start* berada pada petak dengan warna yang sama dan yang paling dekat dengan garis *start*, pada gambar 2.2 ditandai dengan panah lurus. Pada saat bidak berada pada petak tersebut, bidak tersebut tidak dapat diserang oleh bidak milik lawan. Setelah itu, bidak akan menyusuri petak jalur permainan hingga menuju petak garis finish yang ditandai dengan "Home" dan perjalanan bidak selesai sehingga pemain tidak lagi dapat menggerakkan bidak yang sudah berada pada petak "home" atau garis *finish*.
8. Pada saat bidak digerakkan dan ternyata pada petak tujuan terdapat bidak lain dengan warna yang sama, maka pemain tersebut dapat membuat sebuah "halangan" untuk pemain lain. Hal ini biasa disebut dengan "*blob"* atau bermain tumpukan. Ketika lawan mendapatkan dadu yang diimana diharuskan melewati *blob* tersebut, maka giliran akan berlanjut pada pemain berikutnya. Untuk menghancurkan *blob* tersebut, musuh harus mendapatkan angka dadu yang tepat sehingga ada bidak yang dapat menempati petak yang tepat dimana *blob* tersebut berada, jika hal itu terjadi maka biidak pemain yang menciptakan *blob* tersebut akan dikembalikan ke garis *start* (diluar petak jalur permainan) begitu pula bidak yang menghancurkan *blob* tersebut.
9. Untuk skema penyerangan dapat dilakukan ketika pemain A mendapatkan contoh angka dadu 4 dan petak didepan bidang yang akan digerakkan terdapat bidak musuh, maka bidak pemain A dapat menyerang bidak musuh dan akibatnya bidak musuh kembali ke garis *start* diluar petak jalur permainan.
10. Ketika akan mendekati petak "home", jika sebuah bidak hanya butuh tiga langkah untuk masuk ke petak "home" dan dadu menghasilkan angka empat, maka bidak tersebut tidak dapat masuk ke petak "home" dan harus melanjutkan sisa satu langkah dengan mundur 1 petak. Pada intinnya, untuk memasuki petak "home" dibutuhkan angka dadu yang memang dibutuhkan, tidak lebih dan tidak kurang.
11. Daerah aman adalah 5 petak sebelum "petak" home karena pemain lain tidak dapat memasuki daerah aman tersebut karena 5 petak tersebut adalah jalur khusus untuk kembali ke petak "home" masing masing warna

III. APLIKASI ALGORITMA *GREEDY* DALAM PEMILIHAN KARAKTER LANGKAH YANG TEPAT PADA PERMAINAN LUDO

Pada permainan ludo, ada 2 faktor yang dapat membuat kemungkinan menang semakin besar yaitu keberuntungan dan langkah yang diambil pada tiap giliran. Algoritma *greedy* dengan solusi optimum lokal dapat memaksimalkan pergerakan pada setiap gilirannya.

A. Elemen Algoritma *Greedy*

1. Himpunan kandidat (C)
   Himpunan kandidat adalah bidak-bidak yang akan dijalankan. Terdapat 4 warna dan masing masing memiliki 4 bidak untuk dimainkan. C = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}
   Keterangan :
   C = {1,2,3,4} berwarna hijau
   C = {5,6,7,8} berwarna merah
   C = {9,10,11,12} berwarna biru
   C = {13,14,15,16} berwarna kuning

2. Himpunan solusi (S)
   Himpunan solusi adalah petak jalur pada permainan yang akan dilewati oleh bidak. Banyaknya kotak jalur yang akan dilewati bidak sebanyak 56 petak jalur.

3. Fungsi solusi
   Jika angka yang dihasilkan pada pelemparan dadu sesuai dengan petak tujuan bidak tersebut, maka bidak akan bergerak sesuai dengan angka pada dadu lemparan.

4. Fungsi seleksi
   Memilih langkah mana yang menjadi prioritas untuk dilakukan pada giliran tersebut.

5. Fungsi kelayakan
   Memeriksa apakah ada bidak pada kandidat yang dipilih berada pada petak jalur permainan. Dilakukan dengan sistem perulangan untuk memeriksa apakah ada bidak yang layak untuk digerakkan atau tidak. Jika ada maka bidak tersebut akan bergerak dan jika tidak maka akan mencari kandidat lain untuk di gerakkan.

6. Fungsi Obyektif
   Jika petak yang akan dituju terdapat bidak milik lawan maka dilakukan Greedy by Hitter, jika petak jalur yang dituju kosong namun pion berada paling belakang maka dilakukanlah Greedy No One Left Behind, jika petak jalur yang dituju kosong namu bidak berada dekat dengan finish maka dilakukanlah Greedy by Quick Move, jika petak jalur yang dituju terdapat bidak yang berwarna sama maka dilakukanlah Greedy by Blocking. Berapapun angka yang dihasilkan oleh dadu membuka kemungkinan

B. Analisis Algoritma

- *Greedy by Hitter*
  Langkah-langkah menjalankan algoritma sebagai Berikut :
  1) Sistem memeriksa bidak mana yang akan bermain
  2) Contoh, bidak Merah sedang dalam giliran bermain
  3) Pemain dengan bidak merah mengocok dadu dan didapatkan angka N bukan 6.

4) Sistem memeriksa apakah ada bidak merah yang dapat digerakkan dan pada petak tujuan terdapat bidak milik musuh atau tidak.
5) Jika terdapat bidak musuh, maka bidak merah tersebut menjadi prioritas untuk dijalankan.
6) Bidak merah bergerak dan menyingkirkan bidak musuh untuk kembali ke garis *start*

- *Greedy No One Left Behind*
  Langkah-langkah menjalankan algoritma sebagai Berikut :
  1) Sistem memeriksa bidak mana yang akan bermain.
  2) Contoh bidak merah sedang dalam giliran bermain
  3) Pemain dengan bidak merah mengocok dadu dan didapatkan angka 5 atau 6
  4) Sistem memeriksa apakah terdapat bidak merah dengan posisi paling belakang pada petak jalur permainan atau tidak
  5) Jika terdapat bidak dengan paling belakang, maka bidak itu menjadi prioritas untuk di jalankan
  6) Bidak merah maju sesuai dengan angka yang terdapat pada dadu.

- *Greedy by Quick Move*
  Langkah-langkah menjalankan algoritma sebagai Berikut :
  1) Sistem memeriksa bidak mana yang akan bermain
  2) Contoh bidak merah sedang dalam giliran bermain
  3) Pemain dengan bidak merah mengocok dadu dan didapatkan angka N
  4) Sistem memeriksa dengan perulangan apakah ada bidak merah yang memiliki tujuan petak berada pada daerah aman dengan N langkah tersebut
  5) Jika ya, bidak tersebut menjadi prioritas dalam *greedy by Quick Move*.
  6) Bidak tersebut bergerak sesuai dengan angka N pada dadu

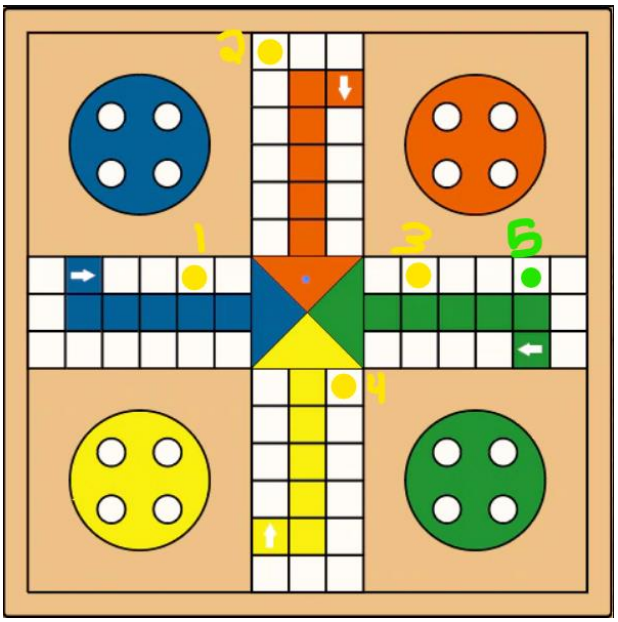
- *Greedy by Blocking*
  Langkah-langkah menjalankan algoritma sebagai Berikut:
  1) Sistem memeriksa bidak mana yang akan bermain
  2) Contoh bidak merah sedang dalam giliran bermain
  3) Pemain dengan bidak merah mengocok dadu dan didapatkan angka N bukan 6
  4) Sistem memeriksa dengan perulangan apakah pada jarak N dari sebuah bidak terdapat bidak lain dengan warna yang sama atau tidak dan apakah ada lawan terdekat yang terletak di belakang *blob* yang akan terbentuk atau tidak.
  5) Jika ya, maka bidak tersebut menjadi priortas dalam *greedy by Blocking*
  6) Bidak merah bergerak lalu membentuk sebuah blob pada petak yang dituju dengan dengan tujuan menghambat gerakan pemain lain

Dari ke empat algoritma tersebut, jika sebuah pemain mendapatkan angka 6 dan masih ada bidak yang terkurung pada garis *start* maka prioritas utama pemain tersebut adalah tetap mengeluarkan semua bidak yang masih berada pada garis *start*. Jika keempat algoritma tersebut terpenuhi oleh bidak-bidak, maka prioritas yang diurutkan adalah

1. *Greedy by Quick Move*
   Karena targe kita dalam permainan ini adalah menjadi paling cepat untuk mengembalikan semua bidak ke garis *finish*
2. *Greedy No One Left Behind*
   Untuk mengejar ketertinggalan bidak, kita Perlu menempatkan *greedy No One Left Behind* pada posisi ke 2.
3. *Greedy by Hitter*
   Pada waktu-waktu tertenu kita juga ingin menghambat kesempatan menang lawan.
4. *Greedy by blocking*
   Strategi ini membutuhkan 2 bidak yang berada di satu tempat dan kita Perlu memeriksa apakah ada lawan terdekat yang dapat kita haling pergerakannya.

C. Contoh Kasus

Berikut adalah contoh kasus ketika semua bidak berwarna kuning berada pada petak jalur permainan dan diasumsikan dadu yang muncul yaitu angka 3.



Gambar 3.1 : Contoh kasus
Sumber:
https://cdn.shopify.com/s/files/1/0876/1176/files/i993_pimgpsh_fullsize_distr.png?v=1525147634

Pada kasus tersebut kita dapat melihat bahwa terdapat 4 bidak kuning dan 1 bidak hijau yang terdapat pada petak jalur permainan. Dalam penerapan Algoritma *greedy* ini kita mengevaluasi pada tiap bidak. Asumsikan pada saat ini giliran kuning bermain. Pada bidak 1 kita periksa apakah memenuhi persyaratan untuk melakukan salah satu dari algoritma diatas atau tidak. Bidak 1 tidak memenuhi *greedy by hitter* karena tidak ada bidak musuh pada 3 petak didepannya, tidak memenuhi *greedy by no one left behind* karena dadu tidak menunjukan angka 5 atau 6, tidak memenuhi *greedy by quick move* karena tidak memungkinkan untuk memasukii daerah aman, dan tidak

memenuhi *greedy by blocking* karena 3 petak didepan tidak terdapat bidak dengan warna yang sama. Begitu pula pada bidak 2 dan 4. Saat kita memeriksa bidak 3, terdapat bidak musuh pada 3 petak setelahnya, hal ini mendukung untuk bidak 3 melakukan *greedy by hitter* karena dapat membuat bidak lawan kembali ke garis *start*.

IV. KESIMPULAN

Berdasarkan analisis algoritma *greedy* yang telah dilakukan, *Board game* ludo ini dapa diselesaikan dengan 4 algoritma diatas yaitu *greedy by hitter,greedy by quick move,greedy by no one left behind* , dan *greedy by blocking*. Penerapan algoritma *greedy* pada permainan ludo tidak akan selalu memberikan solusi yang optimum dikarenakan pada permainan ini dipengaruhi juga faktor keberunungan saat kita mengocok dadu. Secara keseluruhan jika dadu yang muncul sesuai dengan apa yang kita butuhkan saat giliran tersebut, algoritma *greedy* ini menghsilkan solusi yang optimum untuk mempercepat kemenangan pemain.

V. UCAPAN TERIMAKASIH

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT. karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan malah ini dengan baik dan tepat waktu. Ucapan terima kasih juga saya ucapkan kepada Dosen Strategi Algoritma Dr. Ir. Rinaldi Munir, M.T., Dra. Masayu Leylia Khodra, S.T, M.T, dan Dr. Nur Ulfa Maulidevi, S. T, M. Sc. karena atas ilmu yang telah diberikan selama satu semester ini penulis dapat mengaplikasikannya dalam bentuk makalah.

VIDEO LINK AT YOUTUBE

Penulis melampirkan video penjelasan terkait penjelasan permainan dan algoritma greedy yang digunakan dengan tautan https://youtu.be/vTL3todQMhs

REFERENCES

[1] https://gilbertssouthern.com/cara-bermain-ludo-game/#:~:text=Aturan%20main%20ludo%20sangat%20sederhana,Anda%20harus%20mencari%20lawan%20main, diakses pada 22 Mei 2022
[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf diakses pada 22 Mei 2022
[3] https://repository.unikom.ac.id/28648/ diakses pada 22 Mei 2022
[4] https://urbandigital.id/trik-ampuh-menang-ludo-king/ diakses pada 22 mei 2022

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, 22 Mei 2022

Fikri Ihsan Fadhiilah 13520148