

Penerapan Algoritma *Greedy* untuk Menyelesaikan Permainan *Threes*

M. Ferdi Ghozali/13515014

Program Studi Teknik Informatika, Fakultas Teknik Elektro dan Informatika
Institut Teknologi Bandung (ITB)
Bandung, Indonesia 13515014@std.stei.itb.ac.id

Abstraksi—Algoritma *greedy* adalah salah satu algoritma optimasi dengan kecepatan tinggi yang selalu mengambil langkah dengan keuntungan maksimal lokal disetiap kesempatan nya dengan harapan mendapat solusi maksimum global. *Greedy* yang memiliki arti “tamak” sangat menggambarkan bagaimana algoritma ini bekerja. Algoritma *greedy* mengambil apa yang paling menguntungkan saat itu tanpa mempertimbangkan dampak atau konsekuensi kedepannya. Algoritma *greedy* cocok digunakan untuk menyelesaikan game yang butuh optimasi, namun sangat sulit untuk mencari optimum global nya atau bahkan optimum global nya tidak bisa diperkirakan. Permainan *Threes* merupakan jenis permainan puzzle yang memiliki misi untuk mendapatkan poin sebanyak mungkin selama permainan belum berakhir. Permainan bergenre puzzle ini di buat oleh development sirvo untuk platform iOS pada 2014 dan berhasil mendapat 2 penghargaan dari Apple Design Award. Dan karena reputasinya ayng mendukung, game ini kembali dibuat untuk platform android dan XboxOne. Game ini memiliki aturan yang sederhana, membutuhkan optimasi, dan tidak bisa diperkirakan skor maksimalnya.

Keywords—*greedy, threes, skor.*

I. PENDAHULUAN

Semakin majunya teknologi membuat ketergantungan masyarakat terhadap teknologi semakin besar. Manusia yang mulanya bergantung pada handphone untuk sarana berkomunikasi saja, namun setelah adanya smartphone manusia hamper menggantungkan semua hal pada smartphone ini. Hal ini mendorong semakin pesatnya industry smartphone dan industry pembuatan software mobile.

Melihat antusias masyarakat terhadap software smartphone, Sirvo sebagai developer game indie tidak tinggal diam. Sirvo menciptakan game *Threes* untuk para penggemar game puzzle pada platform iOS. Ide game ini sangat menarik, dan fresh pada saat itu. Setelah menemukan ide pembuatan game, tim Sirvo langsung mengerjakan game ini dalam waktu hanya semalam. Namun untuk menyempurnakan game *Threes* menjadi sebagus sekarang perlu waktu 13 bulan lamanya. Para developer mendedikasikan perhatian penuh selama setahun lebih untuk game ini. Setiap keluhan yang diterima akan langsung berusaha perbaiki. tak heran game *Threes* berhasil menerima 2 penghargaan dari Apple Design Award sebagai Excellence Design dan game of the yer pada tahun 2014. Setelah itu Sirvo memutuskan untuk membuat game nya pada platform android dan XboxOne



Figure 1. Logo dari Game *Threes*

Source : <http://ocremix.org/game/95154/threes-ios>

Peraturan Game ini hampir sama dengan game puzzle pada umumnya, game ini memiliki peraturan yang sederhana, pemain harus mendapatkan poin sebanyak banyak nya sebelum dinyatakan selesai. Game selesai apabila pemain sudah tidak bisa melakukan langkah/move apapun.

Walaupun terkesan gampang saat dibaca, namun game ini tergolong sangat sulit karena memerlukan pertimbangan yang matang tiap putaran permainan agar mendapatkan skor semaksimal mungkin. Saking sulitnya, diinternet banyak orang yang membuat program solver nya untuk para pemain pemula yang ingin merasakan mendapatkan skor tinggi pada game *Threes*.

Alasan kenapa game ini terbilang sangat sulit adalah karena adanya faktor random ayng akan dijelaskan pada bab berikut nya. Faktor random ini tidak bisa diperkirakan dan dapat merubah kondisi permainan secara terus menerus. Game ini juga memiliki state awal yang selalu berubah sehingga tidak ada solusi optimum global yang tetap untuk game ini. Jadi bisa dibilang faktor keberuntungan memegang peran yang cukup penting dalam memainkan game ini.

Makalah ini akan membahas tentang penggunaan algoritma *greedy* untuk membantu kita memainkan game ini. Pemilihan algoritma didasari oleh adanya faktor random yang membuat pemain sulit untuk menerka kondisi game untuk putaran permainan selanjutnya. Alhasil kita hanya bisa berusaha yang terbaik untuk saat ini, dan berharap keberuntungan untuk putaran selanjut nya. Nah karena itu saya memutuskan untuk memainkan game ini secara *Greedy*.

II. DASAR TEORI

2.1 STRATEGI ALGORITMA

Strategi Algoritma adalah pendekatan umum untuk memecahkan persoalan secara algoritmis yang dapat diterapkan pada bermacam-macam persoalan dari berbagai bidang komputasi [Levitin, 2003]. Tujuan penggunaan strategi algoritma adalah agar kita dapat memberikan panduan untuk merancang algoritma untuk persoalan baru dan dapat mengklasifikasikan algoritma berdasarkan rancangan pembuatan algoritmanya [Levitin, 2003]. Dengan menggunakan algoritma yang tepat maka kita dapat memecahkan berbagai macam persoalan dengan cepat dan hemat memory efektif dan efisien[1].

Berikut beberapa contoh Algoritma yang telah kita pelajari selama kuliah STIMA:

A. Algoritma Brute Force

Brute Force adalah obat sapu jagat untuk hampir semua penyelesaian masalah. Pendekatan ini memiliki konsep yang sederhana dan sangat mudah dipahami. Seperti arti dari katanya, algoritma Brute Force memiliki mekanisme mencoba segala kemungkinan secara "brute", brute disini maksudnya algoritma ini akan mencoba satu per satu kemungkinan walaupun jumlah percobaannya sangat banyak. Tentunya algoritma ini hanya digunakan jika tidak ada algoritma lain yang lebih efektif. Algoritma ini lumrahnya tidak efektif dan tidak efisien.

B. Algoritma Greedy

Greedy adalah algoritma yang pemecahan masalah dengan selalu mengambil optimum lokal di setiap kesempatan dengan harapan akan mengarah ke optimum global. Lebih banyak mengenai algoritma Greedy akan dibahas dalam subbab berikutnya.

a) Algoritma Divide and Conquer

Divide and Conquer adalah metode penyelesaian masalah dengan cara melakukan dekomposisi masalah besar menjadi yang upa-upa masalah yang lebih kecil, lalu diselesaikan dan digabung kembali untuk menjadi penyelesaian/solusi secara keseluruhan, dengan syarat hasil dekomposisi masalah harus memiliki karakteristik yang sama dengan karakteristik masalah asal.

D. Algoritma Decrease and Conquer

Decrease and conquer adalah metode desain algoritma dengan mereduksi persoalan menjadi beberapa sub-persoalan yang lebih kecil.

Berbeda dengan divide and conquer yang memproses semua subpersoalan dan menggabungkan semua solusi setiap subpersoalan, Metode ini hanya memproses 1 sub persoalan saja dan tidak ada proses combine dalam penyelesaiannya.

E. Algoritma BFS dan DFS

Breadth First Search dan Depth First Search merupakan metode pencarian solusi dengan bantuan Graf, bedanya BFS mencari secara melebar sehingga simpul dari graf yang dikunjungi pun dihampiri secara melebar/horizontal terlebih dahulu sedangkan DFS mencari langsung ke simpul Graf yang paling dalam/vertikal terlebih dahulu.

F. Algoritma Backtracking / Runut Balik Backtracking

Metode ini merupakan fase dari DFS. Backtracking juga merupakan perbaikan dari algoritma Exhaustive Search karena metode Backtracking akan mengeleminasi setiap kemungkinan yang sudah tidak mungkin menjadi solusi.

G. Algoritma Branch and Bound

Branch and Bound adalah metode pencarian solusi dengan memperkecil Search Tree/pohon pencarian menjadi sekecil mungkin dengan cara mematikan cabang hidup yang sudah melanggar syarat batas kendala atau sering disebut bounds.

H. Algoritma KMP dan Boyer Moore

Knuth-Morris-Pratt dan Boyer Moore merupakan metode yang digunakan untuk menyelesaikan masalah pencocokan kata (String Matching). KMP sangat efektif untuk kata-kata string matching dengan pola kata yang relative pendek. sedangkan BM sangat efektif dalam mencari pola kata yang relative panjang.

2.2 ALGORITMA GREEDY

Algoritma greedy adalah algoritma yang pemecahan masalah dengan selalu mengambil optimum lokal di setiap kesempatan dengan harapan akan mengarah ke solusi optimum global. greedy yang memiliki arti "tamak" sangat menggambarkan bagaimana algoritma ini bekerja. Algoritma greedy mengambil apa yang paling menguntungkan saat itu tanpa mempertimbangkan dampak atau konsekuensi kedepannya.

Kelebihan dari algoritma ini adalah kecepatan proses karena hanya melihat apa yang ada sekarang saja, dan kecil kerap mendapatkan hasil yang baik, walaupun bukan yang terbaik karena setidaknya kita selalu mengambil setiap kesempatan semaksimal mungkin. Oleh karena itu algoritma ini cocok untuk masalah optimisasi sebuah nilai baik itu memaksimalkan keuntungan atau meminimalkan cost.

Sedangkan Kelemahan Algoritma ini adalah sangat rentan tidak mendapatkan solusi optimal. Oleh karena itu algoritma ini cocok untuk permasalahan yang memang tidak diharapkan untuk mendapatkan solusi yang paling optimal. Sebagai contoh untuk game yang solusi optimum nya tidak dapat dicari.

Jika digabungkan kecocokan dari kelebihan dan kekurangan, maka akan didapatkan bahwa algoritma greedy sangat cocok untuk mengoptimalkan sebuah nilai yang tidak diwajibkan mendapat solusi yang optimal. Contoh : game Threes

Algoritma greedy memiliki 5 buah elemen umum yaitu :

- Himpunan Kandidat, C
Himpunan kandidat berisi elemen-elemen yang dapat membentuk solusi.
- Himpunan Solusi. S
Himpunan yang berisi elemen dari himpunan kandidat yang telah dipilih sebagai solusi dari persoalan.
- Fungsi seleksi
Fungsi yang pada setiap langkah akan memilih elemen dari himpunan kandidat yang paling memungkinkan mencapai solusi optimal.
- Fungsi kelayakan (feasible)
Fungsi yang memeriksa apakah elemen kandidat yang telah dipilih jika dimasukkan ke dalam himpunan solusi tidak melanggar constraint yang ada.
- Fungsi objektif
Fungsi untuk meminimumkan atau memaksimalkan solusi yang telah dipilih.

2.3 GAME THREES

Threes merupakan game puzzle ciptaan Sirvo yang dapat kalian unduh di Apple store, Playstore dan XboxOne. Threes awalnya merupakan game yang berbayar yang hanya rilis di AppleStore. Threes rilis pada tanggal 6 february 2014. Tak lama setelah rilis, game ini langsung booming dan langsung mendapat 2 penghargaan dari Apple Design Award sebagai game of the year di tahun yang sama game ini dirilis. Karena kepopulerannya Threes langsung dibuat versi android dan XboxOne nya. Dan sekarang Sirvo merilis Threes versi gratis di playstore.[2]

Gameplay permainan ini cukup sederhana. Misi utama permainan ini adalah mendapatkan poin sebanyak banyaknya sebelum permainan berakhir. Permainan berakhir jika sudah tidak ada move/langkah yang bisa pemain lakukan.

Pada awal permainan, akan disediakan sebuah board dengan 4x4 petak/tiles. Pada awal permainan petak akan berisi 9 kotak yang telah terisi angka dengan nilai 1-3 dan 7 kotak kosong. Setiap permainan memiliki situasi board yang berbeda beda.



Figure 2. Contoh Salah Satu Keadaan Awal Game Threes

Game ini menggunakan system turn dimana pemain akan melakukan langkah terlebih dahulu lalu diikuti oleh langkah AI, hal ini dilakukan sampai tidak ada langkah yang dapat kita lakukan atau yang berarti permainan telah berakhir.

Langkah yang dapat kita lakukan adalah dengan menggeser, menyapu, atau bahasa yang sering kita gunakan nya adalah swipe pada layar. Saat melakukan swipe, asalkan jari anda belum lepas dari layar maka program belum akan melakukan aksi, dan anda dapat membatalkan langkah anda. Ketika anda melakukan swipe, katakanlah dari arah kanan ke kiri. Maka semua angka akan berpindah ke petak kosong sebelah kirinya. Dan apabila petak kirinya terdapat angka maka angka tidak akan berpindah. Urutan perpindahan angka berlawanan arah anda melakukan swipe. Contoh : jika anda melakukan swipe dari kanan ke kiri, maka angka yang bergerak adalah angka yang paling kanan terlebih dahulu, lalu diikuti angka yang lebih kanan. Dan jika suatu angka tidak bisa swipe karena ada angka lain, dan angka itu adalah angka pasangannya, maka angka tersebut akan tetap bergeser dan membentuk sebuah angka baru hasil pertambahan dari angka yang bergeser dengan angka pasangannya yang di tubruk. Dan setiap angka berpasangan dengan angka yang bernilai sama kecuali angka 2 yang hanya boleh berpasangan dengan angka satu, begitu juga sebaliknya.

Berikut contoh gambaran move :

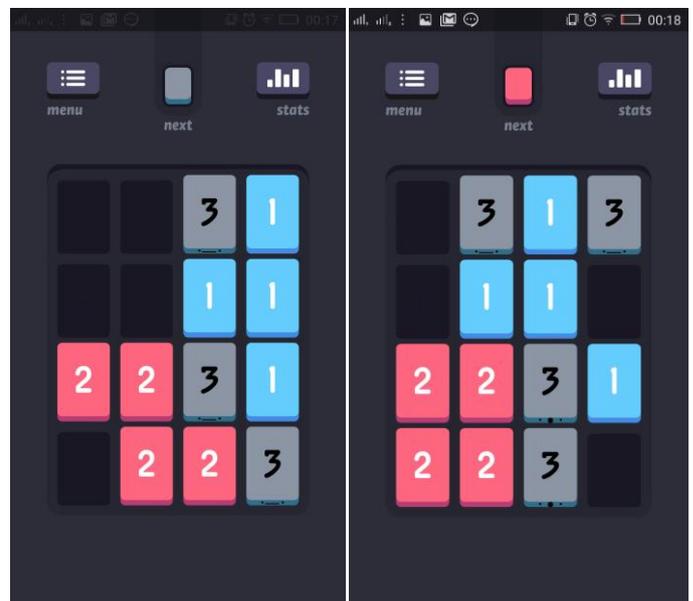


Figure 3 (yang kiri) : Keadaan Sebelum Melakukan Swipe
Figure 4 (yang kanan) : Keadaan Setelah Melakukan Swipe dari kanan ke kiri

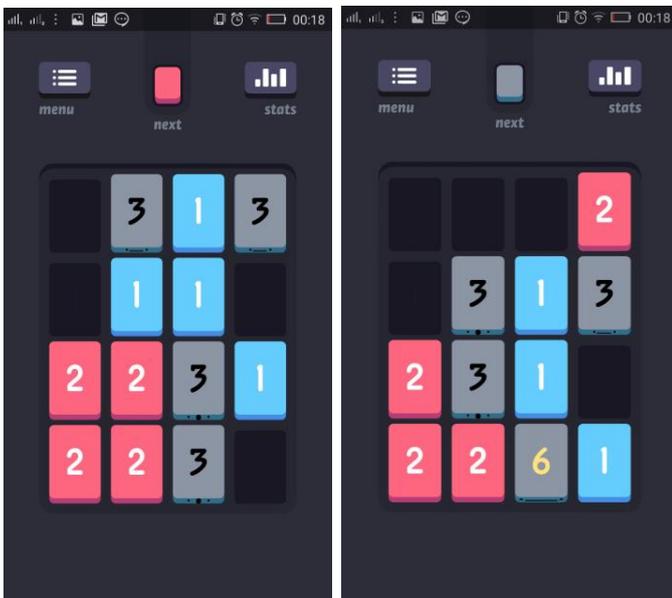


Figure 5 (yang kiri) : Keadaan Sebelum Melakukan Swipe
 Figure 6 (yang kanan) : Keadaan Setelah Melakukan
 Swipe dari atas ke bawah dan terjadi penggabungan antara
 angka 1 & 2 dan 3 & 3.

Langkah dianggap valid/bisa dilakukan apabila jika kita melakukan sebuah langkah maka akan terjadi suatu perubahan pada permainan. Jika suatu langkah tidak menggerakkan angka apapun atau tidak menggabungkan angka apapun maka langkah tersebut tidak dapat dilakukan.

Game akan selesai apabila tidak ada langkah lagi yang dapat kita lakukan. Dan setelah game selesai maka akan dilakukan pengecekan poin akhir permainan. Penilaian berdasarkan angka angka yang ada pada field saat game berakhir. Misalkan : dalam board ada 10 angka 3, 1 angka 96, dan 5 angka 1. Maka total poin yang didapatkan permainan itu adalah $(3 \times 10) + (729 \times 1) + (0 \times 5)$.

Berikut table poin yang di dapatkan berdasarkan angka :

Angka	Poin yang di dapat
1	0
2	0
3	3
6	9
12	27
24	81
48	243
96	729
192	2187
384	6561
768	19683
1536	59049
3072	177147

Table 1 : di kolom kiri terdapat angka angka yang mungkin ada di board, sedangkan kolom kanan berisi poin yang akan dapat apabila terdapat angka itu di board anda

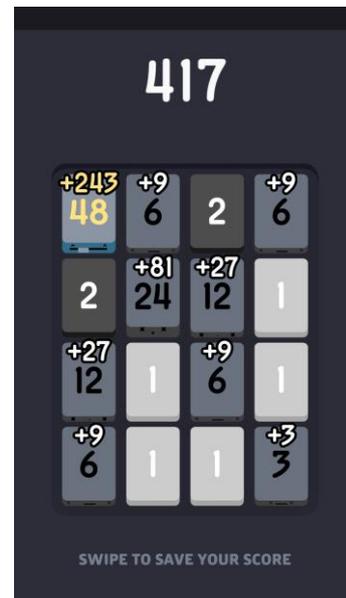


Figure 7 : program saat menghitung poin akhir.

Langkah yang dilakukan oleh AI adalah memasukan angka baru dari petak paling "X", dimana X merupakan arah asal swipe. Misalkan : pemain melakukan swipe dari kanan ke kiri, maka setelah anda melakukan swipe, AI akan menaruh salah satu dari angka yang mungkin ada di board yang nilai nya random namun pasti lebih kecil nilai nya dari angka maximum kita di board itu. Ai akan menaruh angka tersebut ke kolom paling kiri,, AI akan mencari petak paling kiri yang kosong, dan apabila ada 2 lebih yang kosong maka akan dipilih salah satu secara random.

Hal-hal random inilah yang membuat game ini sulit di prediksi dan ditentukan nilai paling optimumnya.

III. PENERAPAN ALGORITMA GREEDY DALAM PERMAINAN THREE

Sebelum Menerapkan Algoritma Greedy dalam Permainan Threes, kita perlu melengkapi elemen elemen dari algoritma greedy terlebih dahulu.

3.1 Analisis Algoritma Greedy dalam Permainan Three

a. Himpunan Kandidat.

Himpunan kandidat untuk permainan ini adalah langkah-langkah yang mungkin dilakukan setiap turn nya. Dalam konteks ini, langkah yang bisa kita ambil adalah swipe atas, swipe bawah, swipe kanan, dan swipe kiri. Jadi Himpunan kandidat Algoritma Greedy kita adalah $C = \{\text{swipe atas, swipe bawah, swipe kiri, swipe kanan}\}$.

b. Himpunan Solusi.

Himpunan solusi untuk permainan ini adalah langkah-langkah yang telah kita ambil sampai permainan berakhir. Karena ada faktor acak/random (angka baru yang nilai dan letak kemunculan nya acak setiap putaran permainan) maka himpunan solusi game ini akan berbeda beda.

c. Fungsi Seleksi.

Fungsi seleksi pada permainan ini kita bergantung pada Poin yang bisa di dapat dan jarak terdekat suatu angka dengan pasangannya. Langkah yang akan diambil adalah langkah dengan poin greedy terbanyak. Dan berikut rumus penentuan poin greedy dari suatu langkah :

$$P = N + U$$

Ket :
 P = total poin greedy
 N = Poin yang didapatkan jika mengambil langkah itu.
 U = Total nilai harapan Nilai Harapan

$$U = \sigma(n)$$

$$n = X/4Y$$

Ket :
 U = Total Nilai Harapan
 n = nilai harapan tiap petak
 X = poin suatu angka jika digabungkan
 Y = jarak petak yang bisa dilalui untuk ke petak pasangan

Dengan rumus seperti diatas maka nilai greedy tidak akan mungkin bernilai negative.

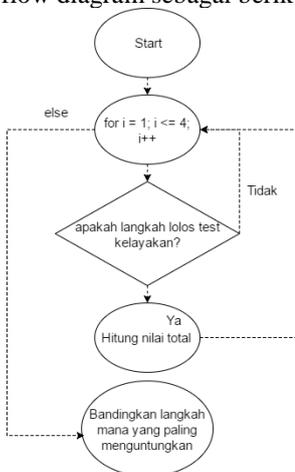
d. Fungsi kelayakan.

Fungsi kelayakan dalam game ini adalah apabila jika dilakukan langkah itu maka kondisi board game ada yang berubah. Apabila langkah tersebut tidak melakukan perubahan apapun baik nilai maupun posisi pada board game. Jika fungsi ini dilanggar maka suatu kandidat tidak bisa dijadikan solusi jadi sudah tidak perlu melalui fungsi seleksi.

e. Fungsi Obyektif.

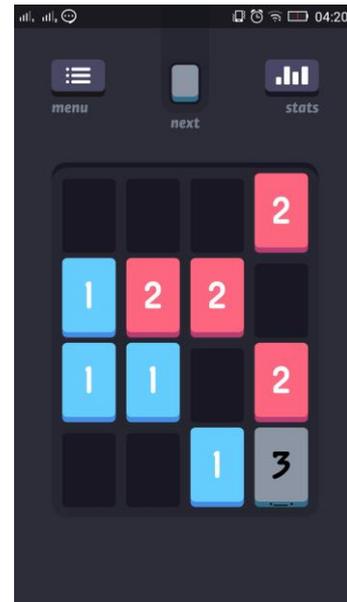
Fungsi obyektif tidak terlalu dipakai dalam permainan ini, karena tidak ada batasan sebuah solusi disebut optimal atau tidak.

Dengan penjelasan di atas, algoritma dapat dibuat dengan flow diagram sebagai berikut:



Ket :
 setiap angka melambangkan move.
 misal :
 nomor 1 -> swipe atas
 nomor 2 -> swipe bawah
 nomor 3 -> swipe kiri
 nomor 4 -> swipe kanan

3.2 Contoh implementasi Greedy pada Trees



$C = \{\text{Swipe atas, Swipe Bawah, Swipe kanan, Swipe kiri}\}$

a. Cek semua himpunan kandidat dengan fungsi kelayakan :

Jika melakukan Swipe atas, maka kondisi board game akan berubah, maka swipe atas lolos fungsi kelayakan.

Jika melakukan Swipe bawah, maka kondisi board game akan berubah, maka swipe atas lolos fungsi kelayakan.

Jika melakukan Swipe kanan, maka kondisi board game akan berubah, maka swipe atas lolos fungsi kelayakan.

Jika melakukan Swipe kiri, maka kondisi board game akan berubah, maka swipe atas lolos fungsi kelayakan.

b. seleksi semua himpunan kandidat yang memenuhi fungsi kelayakan dengan fungsi seleksi

swipe atas :

$$P = N+U$$

$$N = 0$$

$$U = \sigma(n)$$

$$n = X/4Y$$

$$n1 = \text{taka da jalan}$$

$$n2 = 3/4$$

$$n3 = 3/4$$

$$n4 = 3/8$$

$$n5 = \text{tidak ada jalan}$$

$$n6 = 3/4$$

$$n7 = \text{tidak ada jalan}$$

$$n8 = 3/8$$

$$n9 = 0$$

$$U = 3$$

$$P = 0 + 3 = 3$$

Swipe bawah :

$$P = N+U$$

$$N = 0$$

$$U = \sigma(n)$$

$$n = X/4Y$$

$$n1 = 3/16$$

$$n2 = 3/4$$

$$n3 = 3/4$$

$$n4 = 3/4$$

$$n5 = \text{tidak ada jalan}$$

$$n6 = 3/4$$

$$n7 = \text{tidak ada jalan}$$

$$n8 = 3/4$$

$$n9 = \text{tidak ada jalan}$$

$$U = 3 + 31/16$$

$$P = 0 + 3 + 1/16 = 3 + 31/16$$

Swipe kiri ;

$$P = N+U$$

$$N = 3$$

$$U = \sigma(n)$$

$$n = X/4Y$$

$$n1 = \text{tidak ada jalan}$$

$$n2 = \text{tidak ada jalan}$$

$$n3 = 3/4$$

$$n4 = 3/4$$

$$n5 = \text{tidak ada jalan}$$

$$n6 = 3/4$$

$$n7 = 3/4$$

$$n8 = \text{tidak ada jalan}$$

$$n9 = \text{tidak ada jalan}$$

$$U = 3 + 3$$

$$P = 3+3 = 6$$

Swipe kanan ;

$$P = N+U$$

$$N = 0$$

$$U = \sigma(n)$$

$$n = X/4Y$$

$$n1 = 3/12$$

$$n2 = 3/4$$

$$n3 = 3/4$$

$$n4 = \text{tidak ada jalan}$$

$$n5 = \text{tidak ada jalan}$$

$$n6 = 3/4$$

$$n7 = 3/4$$

$$n8 = \text{tidak ada jalan}$$

$$n9 = \text{tidak ada jalan}$$

$$U = 0 + 3 + 3/12$$

$$P = 0 + 3 + 3/12 = 3 + 3/12$$

Jadi setelah dibandingkan, langkah yang paling tepat menurut algoritma greedy adalah swipe ke kiri dengan poin 6.

IV. HASIL PENGUJIAN

a. Hasil Percobaan

Untuk mengetahui kinerja Algoritma Greedy, kami melakukan percobaan 5 kali memainkan game ini dengan algoritma greedy. Dan sebagai pembandingan, kami melakukan percobaan dengan memainkan game ini tanpa memakai algo greedy sebanyak 5 kali.

No Percobaan	Angka Tertinggi Yang Tercapai	Skor Akhir
1	384	9063
2	384	7635
3	192	5808
4	384	7764
5	192	3009

Table 2 : Hasil dari 5 kali percobaan memainkan game Threes menggunakan Algoritma Greedy

No Percobaan	Angka Tertinggi Yang Tercapai	Skor Akhir
1	192	3807
2	384	6084
3	192	2535
4	96	819
5	192	2919

Table 3 : Hasil dari 5 kali percobaan memainkan game Threes TANPA menggunakan Algoritma Greedy

Dari Tabel 2 diatas dapat dilihat bahwa skor maximum yang dapat dicapai permainan dengan greedy adalah 9063 dengan angka tertinggi pada board mencapai 384.

b. Analisis Hasil

Dari Tabel 2 dapat kita lihat bahwa dengan algoritma sama pun nilai yang di dapatkan bermacam macam bahkan jarak antara nilai tertinggi yang pernah di dapat dan nilai terendah yang pernah di dapat sampai 8 x lipatnya. Namun bisa dilihat bahwa kisaran skor yang seringkali muncul adalah kisaran 7000, dan rata rata skor nya mencapai 6655,6. Rata-rata dan modus yang paling sering muncul sama sama mendekati angka 7000an.

Dari Tabel 3 dapat kita lihat bahwa tanpa algoritma kisaran skor yang seringkali muncul adalah kisaran 3000, dan rata rata skor nya mencapai 3232,8. Rata-rata dan modus yang paling sering muncul sama sama mendekati angka 3000an.

V. KESIMPULAN

Setelah melakukan percobaan secara langsung dengan Algoritma greedy dgn metode yang sama, dapat disimpulkan bahwa algoritma greedy cukup baik dipakai dalam game Trees ini karena rata rata skor nya bisa sampai 5000an lebih, hal ini cukup baik karena rata rata skor saya sebelum memakai algoritma ini adalah 3000an lebih.

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.

REFERENCES

- [1] Slide Kuliah STIMA Algoritma Greedy
- [2] <https://en.wikipedia.org/wiki/Threes>



Bandung, 19 Mei 2017

ttd

M.Ferdi Ghozali 13515014