



Implementasi Algoritma Greedy pada Pencarian Langkah Optimal Permainan *Mahjong Solitaire*

Alvin Juvianto^a, Halim Agung^b

^aTeknik Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia, email : alvinjuvianto@gmail.com

^bTeknik Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia, email : hagung@bundamulia.ac.id

Abstract

The game of mahjong solitaire is a difficult tough teaser and takes a long time to complete when manually played by the user. The greedy algorithm is applied to find the optimal step in completion of this solitaire mahjong game. The algorithm used in this research is greedy algorithm. The result of this research is game of mahjong solitaire with 5 model of tile arrangement where game can be completed manually or automatically by greedy algorithm. The greedy algorithm is applied to compare the acquisition of the most tile pairs in the 1 step forward when 1 pair of tiles is removed. The conclusion of this study is from 250 times testing of 5 levels of the game where the test at each level as much as 50 times, obtained the victory as much as 168 times and defeat as much as 82 times. So the percentage of success of greedy algoritma in completion of game of solitaire mahjong is 67.2%. While in manual testing, from 250 times the test of 5 levels of the game where the test at each level as much as 50 times by 5 players, obtained 161 victories and defeat 89 times. So the percentage of success of greedy algoritma in game settlement is 64.4%.

Keywords: *Mahjong Solitaire, game, greedy, artificial intelligence*

Abstrak

Permainan mahjong solitaire merupakan suatu permainan asah otak yang cukup sulit dan membutuhkan waktu cukup lama untuk diselesaikan apabila dimainkan oleh user secara manual. Algoritma greedy diterapkan untuk mencari langkah optimal dalam penyelesaian permainan mahjong solitaire ini. Algoritma yang digunakan dalam penelitian ini adalah algoritma greedy. Hasil dari penelitian ini adalah permainan mahjong solitaire dengan 5 model susunan tile dimana permainan dapat diselesaikan secara manual atau secara otomatis oleh algoritma greedy. Algoritma greedy diterapkan untuk membandingkan perolehan pasangan tile terbanyak pada 1 step kedepan apabila 1 pasang tile dihapus. Hasil dari penelitian ini adalah dari 250 kali pengujian terhadap 5 level permainan dimana pengujian pada setiap level sebanyak 50 kali, diperoleh kemenangan sebanyak 168 kali dan kekalahan sebanyak 82 kali. Jadi persentase keberhasilan algoritma greedy dalam penyelesaian permainan mahjong solitaire adalah 67.2 %. Sedangkan pada pengujian manual, dari 250 kali pengujian terhadap 5 level permainan dimana pengujian pada setiap level sebanyak 50 kali oleh 5 pemain, diperoleh kemenangan sebanyak 161 kali dan kekalahan sebanyak 89 kali. Jadi persentase keberhasilan algoritma greedy dalam penyelesaian permainan adalah 64.4 %.

Kata kunci: *mahjong solitaire, permainan, greedy, kecerdasan buatan*

© 2017 Jurnal RESTI

1. Pendahuluan

Game mahjong solitaire merupakan pengembangan dari *game mahjong tradisional* China yang dimainkan 4 pemain. *Game mahjong solitaire* hanya dimainkan oleh 1 pemain. Batu *mahjong* akan disusun dan saling bertumpuk kemudian pemain tinggal mencocokkan batu *mahjong* yang sama. Tantangan dalam permainan ini adalah pemain tidak dapat sembarangan mencocokkan batu *mahjong*. Hanya batu *mahjong* yang berada ditepian dan tidak dihimpit oleh batu *mahjong* lain yang dapat dicocokkan.

Tujuan dari permainan *mahjong solitaire* ini adalah mencocokkan semua batu *mahjong* dengan waktu secepat mungkin. *Game* ini biasanya membutuhkan waktu yang cukup lama untuk diselesaikan apabila dimainkan secara manual oleh *user* dimana telah dilakukan 50 kali percobaan permainan secara random dan mendapatkan hasil waktu permainan rata-rata 4 - 7 menit. Salah satu algoritma yang digunakan dalam penelitian ini untuk penyelesaian *game mahjong solitaire* adalah algoritma *greedy*. Algoritma *greedy* memiliki kelebihan dalam pengambilan keputusan yang cepat [1]. Algoritma *greedy* merupakan algoritma yang

dapat membantu dalam mencari langkah yang optimal untuk menyelesaikan *game mahjong solitaire* dengan waktu yang cepat. Algoritma Greedy juga diterapkan pada penelitian Noor Saif dan Muhammad Mussafi [2] yang menghasilkan sebuah analisis yang menunjukkan bahwa banyaknya warna minimum yang dapat dibuat yaitu 4 warna, maka bilangan kromatik pada graf tersebut $X(G) = 4$. Hasil ini berkesesuaian dengan teorema 4 warna pada suatu bidang.

2. Tinjauan Pustaka

2.1 Game

Game adalah kegiatan penyelesaian masalah, didekati dengan sikap yang menyenangkan, *game* juga sesuatu yang saat membuat pemain menemukan kesenangan dalam memainkannya. *Game* yang bagus adalah *game* yang dapat membuat pengguna berpartisipasi secara aktif dan mempunyai jumlah tantangan yang tepat, tidak terlalu sedikit atau terlalu banyak. Sikap orang ketika sedang bermain *game*, bisa saja berbeda ketika orang itu sedang tidak bermain *game*, karena ketika orang tersebut sedang bermain *game* maka dia akan merasa sedang berada di “dunia” yang *game* tersebut ciptakan [3].

2.2 Mahjong Solitaire

Mahjong solitaire adalah pembawaan yang relatif modern dari permainan *mahjong* yang tumbuh dalam popularitas selama abad ke-20, terutama karena permainan komputer mulai berkembang pada 1980-an. *Activision* membuat versi dari permainan ini pada tahun 1986 untuk platform *Apple* disebut *Shanghai*, dan *Microsoft* termasuk ulasan versi mereka sendiri dari permainan *Taipei* disebut komputer pribadi sebagai bagian dari *Windows 3.x Entertainment Pack* pada tahun 1990.

Mahjong solitaire melibatkan menghapus batu dari papan dengan mengidentifikasi pasangan yang cocok. Permainan dapat diubah menjadi kegiatan *mnemonic* yang kuat dan adiktif dengan mengharuskan pemain untuk mengidentifikasi pasangan batu dengan konsep serupa [4].

2.3 Algoritma Greedy

Algoritma *greedy* merupakan algoritma yang lazim untuk memecahkan persoalan optimasi meskipun hasilnya tidak selalu merupakan solusi yang optimum. Sesuai arti harafiah, *greedy* berarti tamak. Prinsip utama dari algoritma ini adalah mengambil sebanyak mungkin apa yang dapat diperoleh sekarang [1].

Metode ini melakukan ekspansi node yang memiliki jarak terdekat dengan goal. Namun, ekspansi yang dilakukan pada metode ini menggunakan evaluasi node hanya dengan melihat kepada fungsi heuristiknya. Dengan kata lain, yang dibandingkan untuk penentuan ekspansi node adalah nilai estimasi/prediksinya saja.

Untuk memecahkan persoalan dengan algoritma *Greedy*, kita memerlukan elemen-elemen sebagai berikut:

- a. Himpunan Kandidat (C)
Himpunan ini berisi elemen-elemen pembentuk solusi.
- a. Himpunan Solusi, (S)
Himpunan ini berisi kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.
- b. Fungsi Seleksi
Fungsi seleksi merupakan fungsi yang ada pada setiap langkah memilih kandidat yang paling memungkinkan guna mencapai solusi optimal.
- c. Fungsi Kelayakan (*Feasible*)
Fungsi kelayakan adalah fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak dan tidak melanggar batasan atau *constraints* yang ada.
- d. Fungsi Objektif
Fungsi objektif adalah fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Skema umum algoritma *greedy* adalah sebagai berikut:

- a. Inisialisasi S dengan kosong.
- b. Pilih sebuah kandidat C dengan fungsi seleksi.
- c. Kurangi C dengan kandidat yang sudah dipilih dari langkah (b) di atas.
- d. Periksa apakah kandidat yang dipilih tersebut bersama-sama dengan himpunan solusi membentuk solusi yang layak atau *feasible* (dengan fungsi kelayakan).
- e. Periksa apakah himpunan solusi sudah memberikan solusi yang lengkap serta optimal (dengan fungsi objektif) [5].

3. Metodologi Penelitian

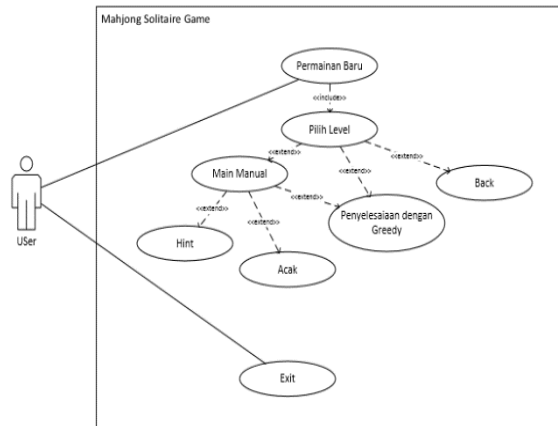
3.1 Pengumpulan Data

Melakukan studi pustaka dari buku, jurnal dan *internet* serta melakukan riset terhadap permainan *mahjong solitaire*.

3.2 Analisis dan desain

Tahapan analisis diperlukan untuk mengetahui kebutuhan apa saja yang terdapat pada aplikasi yang digunakan untuk mengimplementasikan algoritma *greedy*. Kebutuhan aplikasi ini dapat digambarkan menggunakan salah satu diagram UML, yaitu *use case diagram*. *Use case diagram* memberikan cara berkomunikasi kepada *user* dengan detail mengenai apa yang sistem akan lakukan dengan simpel dan langsung. *Use case diagram* menggambarkan cara yang sangat simpel dengan fungsi utama sistem dan beberapa jenis *user* yang berbeda yang akan berinteraksi dengan

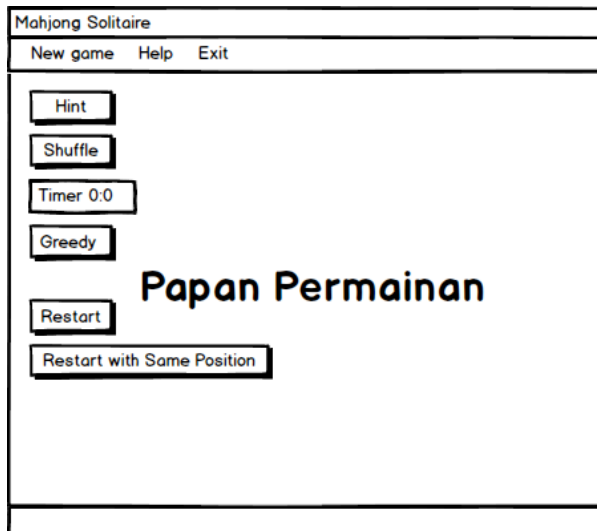
sistem [6]. Use case diagram aplikasi ini dapat dilihat pada Gambar 1.



Gambar 1: Use Case Diagram Permainan Mahjong Solitaire

Interaksi antara pemain dengan aplikasi yang terlihat pada gambar diatas adalah pemain memulai permainan baru dengan memilih level terlebih dahulu. Kemudian dapat memilih untuk memainkan secara manual atau mencari penyelesaiannya dengan algoritma *greedy*.

Setelah menganalisis kebutuhan sistem, berikutnya adalah merancang tampilan aplikasi. Tampilan aplikasi dapat dilihat pada Gambar 2.



Gambar 2: Rancangan Halaman Permainan

Gambar 2 merupakan halaman untuk memainkan *mahjong solitaire*. Terdapat menu bar yang berisi *New game* untuk memulai permainan baru, *help* untuk menginformasikan aturan bermain *mahjong solitaire* dan *exit* untuk menutup aplikasi. Bagian kiri atas terdapat *timer* yang menunjukkan lamanya waktu yang dihabiskan untuk menyelesaikan permainan ini. Terdapat 2 *button* bantuan, yaitu *hint* untuk membantu mencari batu *mahjong* yang dapat dicocokkan dan *shuffle* untuk mengacak posisi batu *mahjong*. *Button greedy* berfungsi untuk menyelesaikan permainan

dengan algoritma *greedy*. *Button restart* untuk memulai permainan dari awal dengan posisi batu *mahjong* yang berbeda dan yang terakhir *button restart with same position* untuk memulai permainan dari awal dengan posisi batu *mahjong* yang sama dengan posisi pada permainan sebelumnya

3.3 Pengkodean

Proses pembuatan aplikasi atau pengkodean (*coding*) dengan pemrograman berorientasi objek sesuai dengan analisis dan antar muka yang telah dibuat. Aplikasi ini berbasis *desktop application* yang dikembangkan menggunakan bahasa pemrograman *vb.net*.

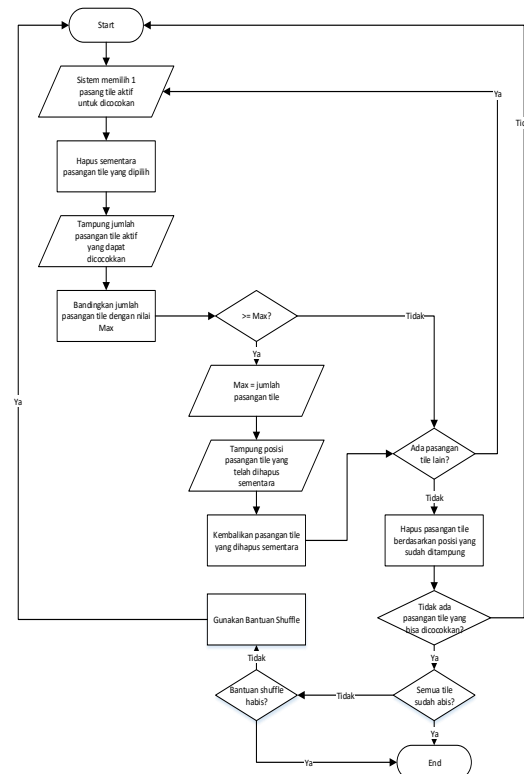
3.4 Pengujian

Melakukan pengujian terhadap aplikasi yang telah dibuat agar implementasi algoritma berjalan dengan benar dan memastikan tidak ada kesalahan pada aplikasi. Pengujian algoritma dilakukan untuk mengukur tingkat keberhasilan algoritma dalam menyelesaikan permainan serta waktu yang diperlukan untuk menyelesaikan permainan tersebut dibandingkan permainan secara manual.

4. Hasil dan Pembahasan

4.1 Flowchart Algoritma

Proses algoritma *greedy* dalam mencari langkah optimal pada permainan *mahjong solitaire* dapat digambarkan dengan flowchart diagram pada Gambar 3.



Gambar 3: Flowchart Algoritma Greedy pada Mahjong Solitaire

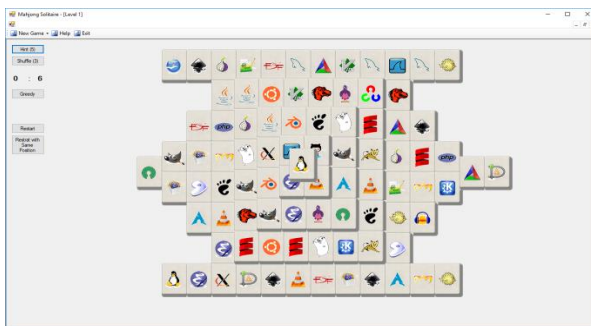
Flowchart adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Jadi, setiap simbol flowchart melambangkan pekerjaan dan instruksinya [7].

Proses yang terjadi adalah pertama sistem memilih semua pasangan *tile* aktif yang dapat dicocokkan. Pilih salah satu pasang *tile*. Hapus sementara pasangan *tile* tersebut lalu cek lagi jumlah pasangan *tile* aktif yang dapat dicocokkan dan tampung pada sebuah variabel *Max*. Kembalikan *tile* yang telah dihapus sementara. Lakukan pada semua pasangan *tile*. Pasangan *tile* yang memiliki nilai *Max* paling besar akan dihapus permanen dari papan. Lakukan semua proses sampai semua *tile* dipapan habis.

4.2 Tampilan Aplikasi

Berikut merupakan tampilan aplikasi yang sudah dibuat:

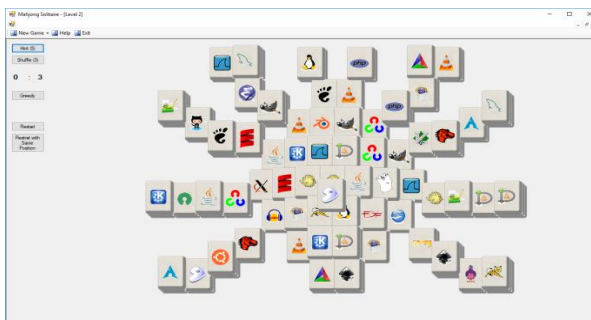
a. Halaman Permainan Model 1



Gambar 4: Halaman Permainan Model 1

Menampilkan halaman permainan model pertama. Pada halaman permainan, *user* tinggal mencocokkan pasangan *tile* sampai semua *tile* habis. Terdapat bantuan *hint* dan *shuffle* untuk memudahkan dalam menyelesaikan permainan. Untuk penyelesaian oleh *artificial intelligence* *user* dapat menekan tombol *greedy*. Terdapat juga tombol *restart* untuk memulai permainan dari awal dengan posisi batu *mahjong* yang berbeda, sedangkan tombol *restart with same position* untuk memulai permainan dari awal dengan posisi batu *mahjong* yang sama dengan posisi pada permainan sebelumnya.

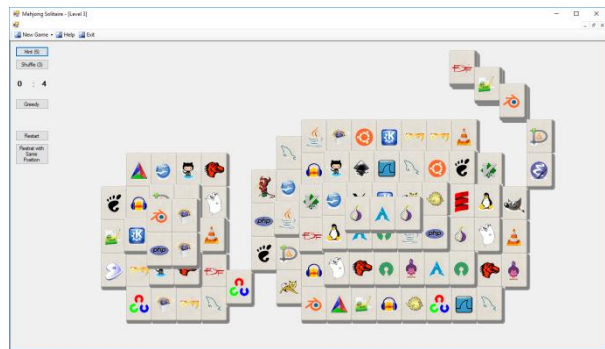
b. Halaman Permainan Model 2



Gambar 5: Halaman Permainan Model 2

Menampilkan halaman permainan model kedua.

c. Halaman Permainan Model 3



Gambar 6: Halaman Permainan Model 3

Gambar 6, menampilkan halaman permainan model ketiga.

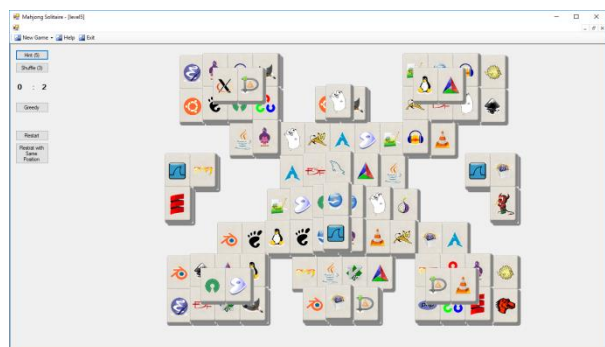
d. Halaman Permainan Model 4



Gambar 7: Halaman Permainan Model 4

Gambar 7, menampilkan halaman permainan model keempat.

e. Halaman Permainan Model 5

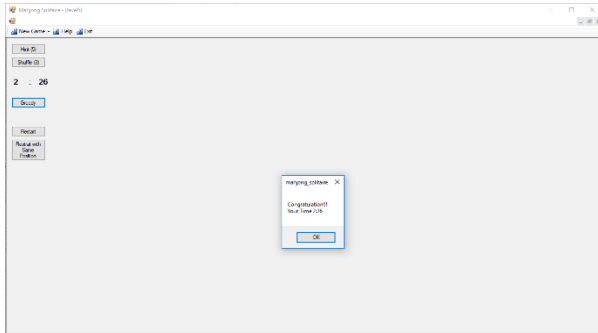


Gambar 8: Halaman Permainan Model 5

Gambar 5, menampilkan halaman permainan model kelima.

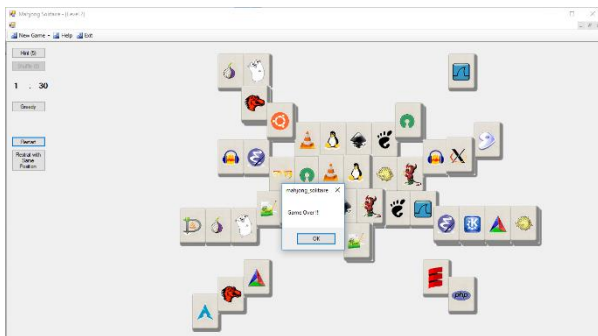
f. Kondisi Menang

Gambar 9 adalah kondisi ketika permainan *mahjong solitaire* berhasil dimenangkan atau saat *tile* sudah terhapus semua dari papan permainan.



Gambar 9: Kondisi Menang

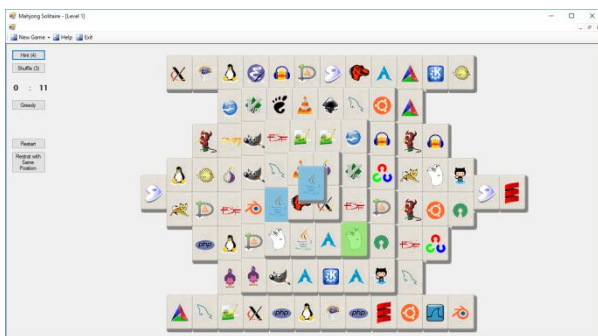
g. Kondisi Kalah



Gambar 10: Kondisi Kalah

Kondisi pada Gambar 10 diatas adalah kondisi kalah atau saat 3 bantuan *shuffle* telah digunakan dan masih ada *tile* tersisa yang tidak dapat dicocokkan.

h. Select dan Hint



Gambar 11: Select dan Hint

Pada Gambar 11 terlihat 1 *tile* yang diberi *highlight* hijau menandakan bahwa *tile* tersebut sedang dipilih (*select*) untuk kemudian dicocokkan dengan *tile* yang sama. Sedangkan 2 *tile* yang diberi *highlight* biru merupakan bantuan *hint* untuk mencari 1 pasang *tile* yang dapat dicocokkan. Kedua fitur tersebut hanya dapat digunakan pada permainan secara manual.

4.3 Pengujian

a. Pengujian Model 1

Pengujian algoritma *greedy* untuk menyelesaikan permainan *mahjong solitaire* model 1 dilakukan

sebanyak 50 kali dan memperoleh tingkat keberhasilan sebesar 66 % dengan waktu tercepat 2 menit 15 detik dan waktu terlama 2 menit 27 detik. Sedangkan pengujian secara manual oleh 5 pemain pada level 1 yang dilakukan sebanyak 50 kali, memperoleh tingkat keberhasilan sebesar 64 % dengan waktu tercepat 4 menit 32 detik dan waktu terlama 8 menit 22 detik.

b. Pengujian Model 2

Pengujian algoritma *greedy* untuk menyelesaikan permainan *mahjong solitaire* model 2 dilakukan sebanyak 50 kali dan memperoleh tingkat keberhasilan sebesar 54 % dengan waktu tercepat 2 menit 17 detik dan waktu terlama 2 menit 28 detik. Sedangkan pengujian secara manual oleh 5 pemain pada level 2 yang dilakukan sebanyak 50 kali, memperoleh tingkat keberhasilan sebesar 54 % dengan waktu tercepat 4 menit 53 detik dan waktu terlama 8 menit 3 detik.

c. Pengujian Model 3

Pengujian algoritma *greedy* untuk menyelesaikan permainan *mahjong solitaire* model 3 dilakukan sebanyak 50 kali dan memperoleh tingkat keberhasilan sebesar 74 % dengan waktu tercepat 2 menit 17 detik dan waktu terlama 2 menit 27 detik. Sedangkan pengujian secara manual oleh 5 pemain pada level 3 yang dilakukan sebanyak 50 kali, memperoleh tingkat keberhasilan sebesar 70 % dengan waktu tercepat 4 menit 27 detik dan waktu terlama 6 menit 33 detik.

d. Pengujian Model 4

Pengujian algoritma *greedy* untuk menyelesaikan permainan *mahjong solitaire* model 4 dilakukan sebanyak 50 kali dan memperoleh tingkat keberhasilan sebesar 58 % dengan waktu tercepat 2 menit 17 detik dan waktu terlama 2 menit 27 detik. Sedangkan pengujian secara manual oleh 5 pemain pada level 4 yang dilakukan sebanyak 50 kali, memperoleh tingkat keberhasilan sebesar 50 % dengan waktu tercepat 4 menit 38 detik dan waktu terlama 7 menit 21 detik.

e. Pengujian Model 5

Pengujian algoritma *greedy* untuk menyelesaikan permainan *mahjong solitaire* model 5 dilakukan sebanyak 50 kali dan memperoleh tingkat keberhasilan sebesar 84 % dengan waktu tercepat 2 menit 14 detik dan waktu terlama 2 menit 27 detik. Sedangkan pengujian secara manual oleh 5 pemain pada level 5 yang dilakukan sebanyak 50 kali, memperoleh tingkat keberhasilan sebesar 78 % dengan waktu tercepat 4 menit 1 detik dan waktu terlama 6 menit 43 detik.

f. Rangkuman Pengujian

Dari 250 kali pengujian terhadap 5 level permainan *mahjong solitaire* dimana pengujian pada setiap level sebanyak 50 kali, diperoleh kemenangan sebanyak 168 kali dan kekalahan sebanyak 82 kali. Jadi persentase keberhasilan algoritma *greedy* dalam

penyelesaian permainan *mahjong solitaire* adalah 67.2%.

Sedangkan pada pengujian manual, dari 250 kali pengujian terhadap 5 level permainan *mahjong solitaire* dimana pengujian pada setiap level sebanyak 50 kali oleh 5 pemain, diperoleh kemenangan sebanyak 161 kali dan kekalahan sebanyak 89 kali. Jadi persentase keberhasilan algoritma *greedy* dalam penyelesaian permainan *mahjong solitaire* adalah 64.4%.

Dapat dilihat juga bahwa, penyelesaian dengan algoritma *greedy* lebih cepat dibandingkan permainan secara manual dimana dengan algoritma *greedy* memperoleh waktu sekitar 2 menit sedangkan permainan manual memperoleh waktu diatas 4 menit.

5. Kesimpulan

5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan penulis tentang penerapan algoritma *greedy* untuk mencari langkah optimal pada permainan *mahjong solitaire*, maka diperoleh kesimpulan sebagai berikut:

1. Algoritma *greedy* dapat digunakan untuk mencari langkah optimal dalam penyelesaian permainan *mahjong solitaire*. Hal ini dibuktikan dari pengujian sebanyak 250 kali terhadap 5 level permainan *mahjong solitaire*, diperoleh persentase keberhasilan pengujian sebesar 67.2 %.

2. Algoritma *greedy* dapat menyelesaikan permainan *mahjong solitaire* lebih cepat dibandingkan dengan permainan secara manual oleh *user*, dimana dengan algoritma *greedy* memperoleh waktu sekitar 2 menit sedangkan permainan manual memperoleh waktu diatas 4 menit.

6. Daftar Rujukan

- [1] Rachmawati, D dan Candra, A., 2013. Implementasi Algoritma Greedy untuk Menyelesaikan Masalah Knapsack Problem. *Jurnal Saintikom*, Vol. 12.
- [2] Mussafi, Noor Saif Muhammad. (2015), *Penerapan Greedy Coloring Algorithm pada Peta Kotamadya Yogyakarta Berbasis Four-Colour Theorem*, Jurnal Kaunia, Vol. 9, No. 1, ISSN: 1829-5266 (print), ISSN: 2301-8550 (online), Universitas Islam Negeri Sunan Kalijaga, Yogyakarta
- [3] Singkoh, R. T., et al., 2016. Perancangan Game FPS (First Person Shooter) Police Personal Training. *E-Journal Teknik Elektro dan Komputer*, Vol. 5
- [4] Cossairt, T. J. dan Grubbs, W. T., 2011. Chemical Mahjong. *Journal of Chemical Education*, 88.
- [5] Budiharto, W. dan Suhartono, D., 2014. *Artificial Intelligence Konsep dan Penerapannya*. Yogyakarta: Andi.
- [6] Dennis, A., et al., 2013. *Systems Analysis and Design with UML*. 4th ed. New York: John Wiley & Sons.
- [7] Community, E., 2012. *Indeks Lengkap Syntax*. Yogyakarta: Mediakom.