

PENERAPAN ALGORITMA *GREEDY* PADA PERMAINAN MATEMATIKA

Fathirul Ilmi¹, Septi Maisari², M. Havis Irfani³, Kgs Achmad Siddik⁴

STMIK GI MDP; JL. Rajawali No. 14 Palembang, 0711-376400

Jurusan Teknik Informatika, STMIK GI MDP, Palembang

e-mail: fathirul@mhs.mdp.ac.id¹, dc_nr@mhs.mdp.ac.id²,

haviz.irfani@mdp.ac.id³, siddiqachmad@mdp.ac.id⁴

Abstrak

Permainan merupakan salah satu media yang tidak hanya memberikan hiburan, tetapi juga sebagai media pembelajaran yang dapat meningkatkan perkembangan kognitif anak dalam membaca, berhitung, dan pemecahan masalah, seperti permainan matematika. Permainan matematika dalam penelitian ini, yaitu permainan meminimalkan pecahan uang. Masalah yang dihadapi pada penelitian ini adalah diperlukannya sebuah algoritma yang dapat memberikan solusi dalam meminimalkan pecahan uang dengan seminimum mungkin. Salah satu algoritma yang dapat meminimalkan pecahan uang adalah algoritma greedy. Algoritma greedy merupakan algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara pada setiap langkahnya. Langkah yang didapat menggunakan algoritma greedy merupakan langkah yang mendekati pilihan terbaik. Oleh karena itu, pada permainan ini mengimplementasikan algoritma greedy untuk memberikan solusi dalam meminimalkan pecahan uang. Permainan ini dibangun dengan menggunakan metodologi iterasi dan dibuat dengan menggunakan software Unity 3D serta bahasa pemrograman C#. Dari hasil implementasi dan pengujian terhadap Algoritma greedy, didapat kesimpulan bahwa algoritma greedy dapat meminimalkan pecahan uang. Sedangkan berdasarkan hasil pengujian dengan menggunakan metode black box testing, fitur-fitur yang terdapat pada permainan dapat berfungsi sesuai dengan yang diinginkan.

Kata kunci : Algoritma Greedy, Permainan, Unity 3D.

Abstract

Game is one of media that not only deliver entertainment, but also a learning media for increasing children cognitive development in reading, counting and problem solving such as mathematical game. Math games in this study, namely minimizing game denominations. Problems encountered in this study was the need for an algorithm can provide solutions to minimize denominations with a minimum. One algorithm that can minimize denominations are greedy algorithm. Mathematical games in this research is a fractional currency game that implements the greedy algorithm to give solution for denomination. Greedy algorithm is the algorithm that use a problem solving approach by search the temporal maximum value in every step. Step obtained use a greedy algorithm is a step closer to the best choice. By Therefore, in this game implements a greedy algorithm to provide solution in minimizing denominations. Step that can be used by using greedy algorithm is the closest toward the best step. This game was developed by using iteration methodology and made by Unity 3D software with C# programming language. The conclusion from the implementation result and testing on greedy algorithm is that greedy algorithm can do the denomination task. Otherwise, based on the testing result by using the black box testing, the features in the game works as expected.

Keyword : Greedy Algorithm, Games, Unity 3D.

1. PENDAHULUAN

Perkembangan teknologi yang semakin modern dapat dijadikan sebagai alat atau sarana permainan yang baik. Salah satunya, yaitu dengan menggunakan teknologi komputer. Teknologi komputer dapat menjadi sarana permainan belajar yang positif dengan pendekatan sesuai tahap perkembangan anak dalam belajar.

Menumbuhkan semangat belajar anak memang merupakan masalah tersendiri yang harus diselesaikan para orang tua. Mengingat belajar merupakan salah satu sarana yang efektif untuk membantu anak agar tetap berprestasi, minimal anak tidak akan ketinggalan pelajaran dan nilainya tidak terlalu jatuh. Salah satu penyebab anak malas belajar adalah waktu bermain yang sedikit. Anak yang lebih banyak menghabiskan waktu di sekolah cenderung akan malas untuk belajar. Beban pikiran di sekolah membuat anak harus mencari hiburan tersendiri. Cara meningkatkan dan menumbuhkan semangat anak untuk belajar dapat dilakukan dengan menggunakan permainan. Permainan merupakan salah satu media hiburan yang dapat memberikan rasa senang pada anak dan seseorang yang bermain *game* akan lebih fokus terhadap apa yang terjadi di sekitarnya dan kemampuan visualnya lebih meningkat bila dibandingkan dengan orang yang tidak bermain *game* [1].

Penelitian sebelumnya yang berjudul “Penerapan Algoritma *Greedy* Pada *Game* Monopoli Indonesia Berbasis Unity 3D”. Penelitian tersebut menerapkan algoritma *greedy* untuk membeli rumah yang profitnya tinggi. Hasil dari penelitian tersebut, yaitu pemain yang memainkan permainan dengan metode *greedy* bisa memenangkan permainan [2].

Dari penjelasan di atas, maka dibuatlah suatu penelitian permainan matematika, yaitu permainan meminimalkan pecahan uang dengan menerapkan algoritma *greedy* yang berfungsi untuk memberikan solusi yang tepat dalam meminimalkan pecahan uang. Digunakannya algoritma *greedy*, karena algoritma *greedy* merupakan algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum pada setiap langkahnya dan algoritma *greedy* akan menghasilkan solusi yang dapat meminimumkan nilai. Oleh karena itu algoritma *greedy* sangat cocok diterapkan pada permainan ini untuk meminimumkan pecahan uang. Dengan permainan ini diharapkan dapat membantu dalam meningkatkan semangat anak untuk belajar dan kemampuan berhitung anak serta dapat meminimalkan pecahan uang dengan metode *greedy*.

2. METODE PENELITIAN

2.1 Pengertian Matematika

Matematika berasal dari bahasa latin *mathanein* atau *mathemta* yang berarti “belajar atau hal yang dipelajari”. Dalam bahasa Belanda disebut *wiskunde* atau ilmu pasti, yang kesemuanya berkaitan dengan penalaran [3]. Matematika adalah studi besaran, struktur, ruang, dan perubahan. Para matematikawan mencari berbagai pola, merumuskan konjektur baru, dan membangun kebenaran melalui metode deduksi yang kaku dari aksioma-aksioma dan definisi-definisi yang bersesuaian.

2.2 Algoritma *Greedy*

Algoritma *greedy* adalah algoritma yang selalu memilih solusi terbaik pada suatu saat tertentu tanpa memikirkan apakah pilihan solusi itu akan berakibat buruk terhadap pilihan lain di masa yang akan datang”. Algoritma *greedy* secara umum mengindikasikan bahwa implementasi akan menggunakan sederetan pilihan sementara yang terbaik, yang akan memicu pilihan terbaik yang bersifat final.

Algoritma *greedy* disusun oleh elemen-elemen sebagai berikut:

1. Himpunan kandidat: Berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi: Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi: Memilih kandidat yang paling memungkinkan mencapai solusi optimal.

4. Fungsi kelayakan (*feasible*): Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak.
5. Fungsi obyektif: Merupakan fungsi yang memaksimumkan atau meminimumkan nilai solusi [4].

2.3 Game Engine

Game Engine adalah sebuah perangkat lunak yang dirancang untuk membuat sebuah *game*. Sebuah *game engine* biasanya dibangun dengan mengenkapsulasi beberapa fitur standar yang umum digunakan dalam pembuatan sebuah *game*. Misalnya, fungsi *rendering*, pemanggilan suara, *network*, atau pembuatan partikel untuk *special effect*. Sebagian besar *game engine* umumnya berupa *library* atau sekumpulan fungsi-fungsi yang penggunaannya dipadukan dengan bahasa pemrograman [5].

2.4 Sekilas Unity 3D

Unity merupakan *game engine* yang dikembangkan oleh Unity Technology. *Software* ini pertama kali diluncurkan pada tahun 2005 dan menjadi salah satu dari sekian banyak *game engine* yang dipakai oleh banyak pengembang *game* profesional di dunia. Unity merupakan alat bantu pengembang *game* dengan kemampuan *rendering* yang terintegrasi di dalamnya. Dengan menggunakan kecanggihan fitur-fiturnya dan juga kecepatan kerja yang tinggi, Unity dapat menciptakan sebuah program interaktif tidak hanya dalam 2 dimensi, tetapi juga dalam bentuk 3 dimensi [6].

2.5 Bahasa Pemrograman C#

Bahasa pemrograman C# (baca : C-sharp) dirancang oleh Microsoft Corp sebagai bahasa pemrograman yang sangat berdaya-guna, aman, serta mudah digunakan. Sebagai bagian dari *platform .NET*, bahasa pemrograman C# dirancang sedemikian rupa untuk bekerja dengan sangat baik diatas *framework .NET*. Kode yang ditulis menggunakan bahasa pemrograman C# tidak secara langsung diterjemahkan ke bahasa mesin atau komputer. Kompiler bahasa pemrograman C# mengonversi kode yang ditulis menggunakan bahasa C# menjadi berkas MSIL (*Microsoft Intermediate Language*), yang dinamakan *assembly*. Berkas MSIL dapat dibuat menggunakan semua bahasa pemrograman yang dikembangkan di atas CLR (*Common Language Runtime*), misalnya Visual C++, Visual J#, Visual Basic, dan sebagainya[7].

2.6 Database SQLite

SQLite adalah paket perangkat lunak yang menyediakan sistem manajemen *database* relasional. Sistem *database* relasional yang digunakan untuk menyimpan *record* yang didefinisikan oleh user dalam tabel yang berskala besar. Selain penyimpanan dan manajemen data, mesin *database* dapat memproses perintah *query* yang kompleks dengan mengkombinasikan data dari beberapa tabel untuk menghasilkan laporan dan ringkasan data. *SQLite* sendiri terkenal sebagai *database* yang *opensource*, *stand alone SQL database*, berukuran kecil, tidak membutuhkan administrasi, tanpa server, tanpa file konfigurasi dan juga telah digunakan pada banyak aplikasi terkenal seperti Browser Firefox dan Iphone [8].

2.7 Metodologi Iterasi

Metode iterasi merupakan gabungan dari model *waterfall* dan model prototipe. Metode iterasi sangat cocok digunakan jika staf yang dimiliki memiliki pergantian (*turnover*) yang tinggi sehingga staf tidak dapat terus ikut dalam pengembangan perangkat lunak. Mekanisme tahapan iterasi perlu direncanakan terlebih dahulu agar hasil produk dan pengerjaan setiap tahapan iterasi menjadi lebih baik. Metode iterasi memiliki tahapan atau fase pengembangan sistem dengan teknik pengulangan, dimana suatu proses dapat dilaksanakan secara berulang-ulang sampai mendapatkan hasil yang maksimal [9].

3. HASIL DAN PEMBAHASAN

3.1 Pemodelan Use Case Diagram

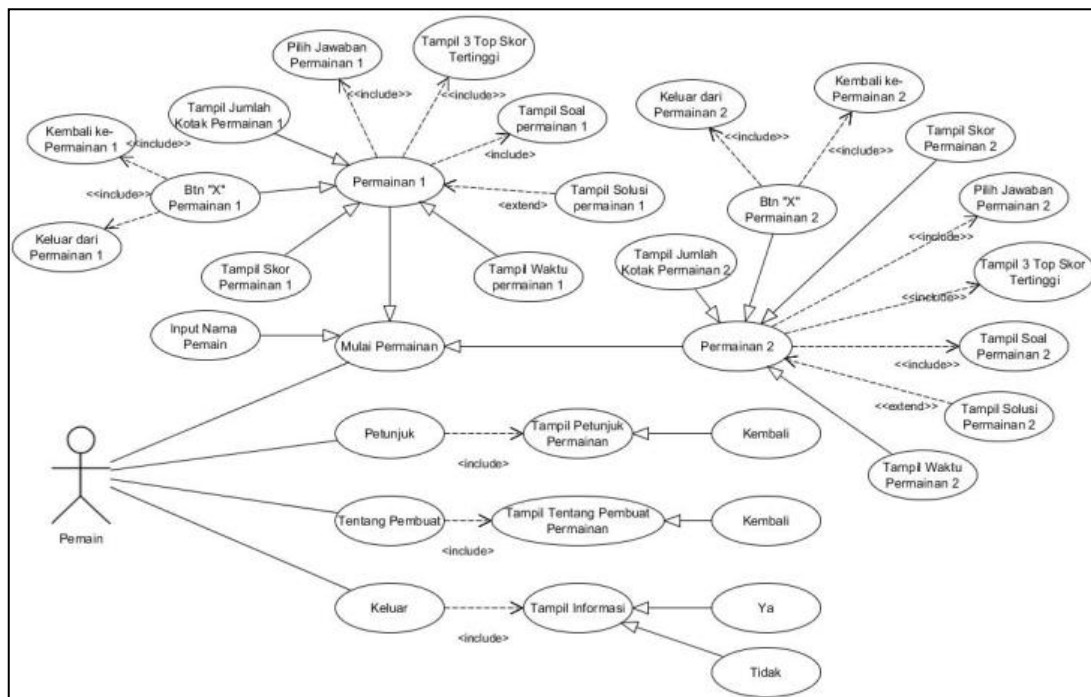
Gambar 3.1 merupakan diagram *use case* dari permainan matematika, terdapat satu aktor pada aplikasi yang akan dibangun, yaitu pemain. Pemain adalah orang yang berperan dalam pemilihan menu dan menjalankan permainan.

Use case yang berhubungan dengan mulai permainan terdiri dari *input* nama pemain, yaitu memasukan nama pemain ke dalam *database*. *Use case* permainan 1 memiliki aktivitas-aktivitas untuk menampilkan soal permainan, memilih jawaban dari soal yang di berikan, menampilkan solusi dari soal permainan, menampilkan waktu permainan, menampilkan nama pemain, menampilkan skor yang diperoleh pemain, menampilkan jumlah kotak harta karun yang telah terbuka, menampilkan tiga top skor tertinggi, dan tombol X untuk kembali ke permainan dan keluar dari permainan. Sedangkan *use case* pada permainan 2 terdapat aktivitas-aktivitas untuk menampilkan soal permainan, pilih jawaban dari soal yang diberikan, menampilkan solusi dari soal yang diberikan, menampilkan jumlah skor yang didapat pemain menampilkan waktu permainan, menampilkan jumlah kotak harta karun yang telah terbuka, menampilkan nama pemain, menampilkan 3 skor tertinggi, dan tombol X untuk kembali ke permainan dan keluar dari permainan.

Use case yang berhubungan dengan petunjuk akan menampilkan informasi mengenai cara memainkan permainan dan memiliki *use case* kembali, yaitu untuk kembali ke halaman utama.

Use case yang berhubungan dengan tentang pembuat akan menampilkan data tentang pembuat permainan dan memiliki *use case* kembali, yaitu untuk kembali ke halaman utama.

Use case yang berhubungan dengan keluar akan menampilkan informasi dan memiliki *use case* “Ya”, yaitu untuk keluar dari permainan dan “Tidak”, yaitu untuk tetap di dalam permainan.



Gambar 3.1 Diagram Use Case Permainan Matematika

3.2 Pseudocode Algoritma Greedy

Berikut ini merupakan *pseudocode* dari algoritma *greedy* pada permainan matematika yang dapat dilihat pada Gambar 3.3.

```

Function greedy(x)

y ← {}
looper ← 0
x ← 0
z ← 0
a ← 0

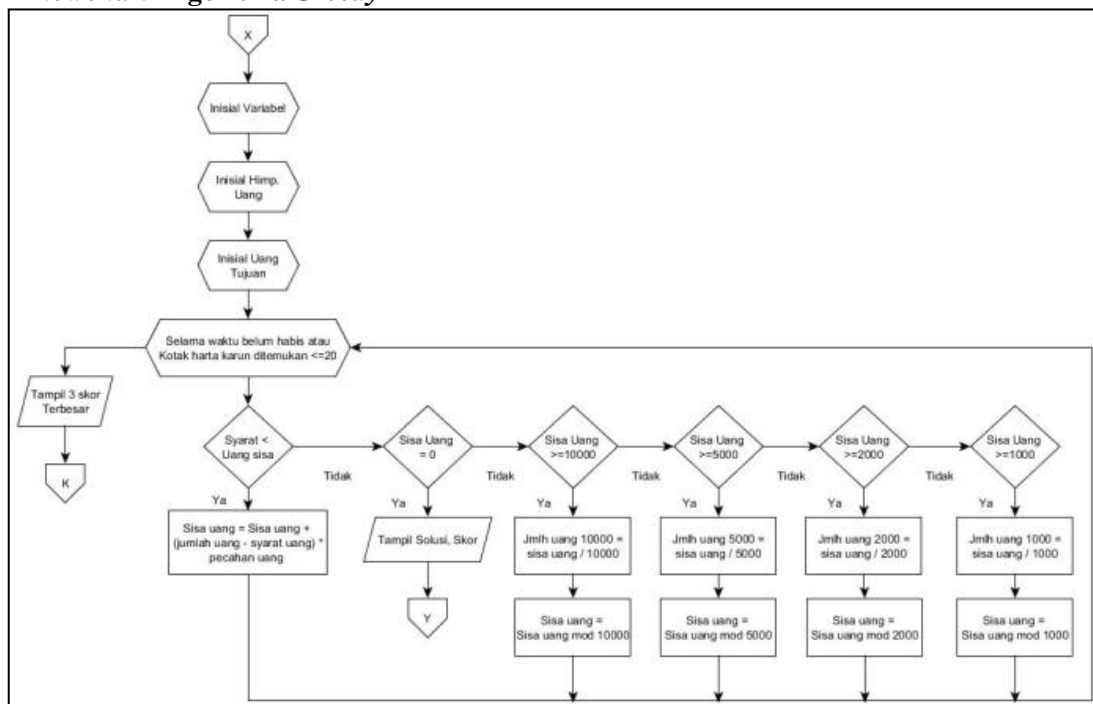
While (x != 0 || looper < Length(y))
    z[looper] ← x / y[looper]
    x ← x mod y[looper]

    If (a < z[looper])
        x ← x + (z[looper] - a) * y
        z ← a [looper]
    End if

    looper ← looper + 1
End while
Return z
    
```

Gambar 3.2 Pseudocode Algoritma Greedy

3.3 Flowchart Algoritma Greedy



Gambar 3.3 Flowchart Algoritma Greedy pada Permainan Matematika

Gambar 3.3 merupakan *flowchart greedy* dari permainan matematika. Langkah pertama melakukan inisial variabel dan inisial himpunan uang (1000, 2000, 5000, 10000) serta inisial uang tujuan (misal 27000). Jika syarat (uang yang dimiliki) lebih kecil dari uang sisa, maka sisa uang sama dengan sisa uang ditambah jumlah uang dikurang syarat uang lalu dikali pecahan uang. Jika sisa uang tujuan tidak sama dengan nol, yaitu lebih dari atau sama dengan 10000, maka sisa uang tujuan akan di bagi 10000 sehingga sisa uang tujuan sama dengan sisa dari hasil pembagian tersebut. Apabila sisa uang tujuan lebih dari

atau sama dengan 5000, maka sisa uang tujuan akan di bagi 5000 sehingga sisa uang tujuan sama dengan sisa dari hasil pembagian tersebut. Apabila sisa uang tujuan lebih dari atau sama dengan 2000, maka sisa uang tujuan akan di bagi 2000 sehingga sisa uang tujuan sama dengan sisa dari hasil pembagian tersebut. Apabila sisa uang tujuan lebih dari atau sama dengan 1000, maka sisa uang tujuan akan di bagi 1000 sehingga sisa uang tujuan sama dengan sisa dari hasil pembagian tersebut. Langkah ini dilakukan terus menerus sampai sisa uang sama dengan 0. Apabila seluruh kotak harta karun telah ditemukan, maka akan tampil halaman tiga top skor terbesar.

3.4 Pencarian Manual Algoritma *Greedy*

Berikut ini merupakan pencarian manual dari algoritma *greedy* pada permainan matematika yang dapat dilihat pada Gambar 3.4.

Tanpa Syarat	Dengan Syarat
Jumlah Uang = 27000	Jumlah Uang = 32000
→10000 (27000-10000=17000)	Uang yang dimiliki : 1 Buah 10000 7 Buah 2000 5 Buah 5000 3 Buah 1000
→10000 (17000-10000=7000)	→10000 (32000-10000=22000)
→5000 (7000-5000=2000)	→5000 (22000-5000=17000)
→2000 (2000-2000=0)	→5000 (17000-5000=12000)
	→5000 (12000-5000=7000)
	→5000 (7000-5000=2000)
	→2000 (2000-2000=0)

Gambar 3.4 Pencarian Manual Algoritma *Greedy*

3.5 Coding Algoritma *Greedy*

Berikut ini merupakan *coding* dari algoritma *greedy* pada permainan matematika yang dapat dilihat pada Gambar 3.5.

```
int[] greedy2(int uang_dicari)
{
    int uang_dipakai = uang_dicari;
    int uang_sisa = uang_dipakai;
    int uang_pelengkap = 0;
    looper = 0;
    while (uang_sisa!=0 || looper<pecahan_uang.Length)
    {
        jumlah_uang[looper] = uang_sisa/pecahan_uang[looper];
        Debug.Log("jumlah Uang:"+jumlah_uang[looper]);
        uang_sisa=uang_sisa%pecahan_uang[looper];

        if (gameCorePermainan1.syarat_uang[looper]<jumlah_uang[looper] &&
            gameCorePermainan1.syaratye)
        {
            uang_sisa = uang_sisa + (jumlah_uang[looper]-
                gameCorePermainan1.syarat_uang[looper])*
                pecahan_uang[looper];
            jumlah_uang[looper]= gameCorePermainan1.syarat_uang[looper];
        }
        looper++;
    }
    return jumlah_uang;
}
```

Gambar 3.5 Coding Algoritma *Greedy* pada Permainan

3.6 Tampilan *Game*

3.6.1 Tampilan Halaman Utama

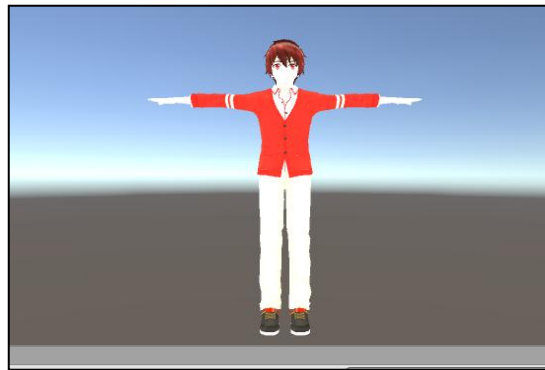
Tampilan halaman utama merupakan tampilan awal saat permainan dijalankan. Terdapat empat *button* pilihan, yaitu mulai permainan, petunjuk, tentang pembuat, dan *butto* keluar. Tampilan halaman utama dapat dilihat pada Gambar 3.6.



Gambar 3.6 Tampilan Halaman Utama

3.6.2 Karakter Permainan

Gambar 3.7 merupakan karakter dari permainan matematika yang di *import* dari *asset store*.



Gambar 3.7 Karakter Permainan

3.6.3 Tampilan Soal Permainan 1

Gambar 3.8 merupakan tampilan soal pada permainan 1. Soal pada permainan 1 tidak memiliki syarat (tidak ada uang yang dimiliki). Algoritma *greedy* berfungsi untuk memberikan solusi dalam meminimalkan pecahan uang dan solusi dari permainan akan selalu tampil saat pemain berhasil ataupun tidak berhasil dalam meminimalkan pecahan uang dengan metode *greedy*.



Gambar 3.8 Tampilan Soal Permainan 1

3.3.4 Tampilan Soal Permainan 2

Gambar 3.9 merupakan tampilan dari soal yang terdapat pada permainan 2. Soal pada permainan 2 memiliki syarat (terdapat uang yang dimiliki). Algoritma *greedy* akan memberikan solusi dalam meminimumkan pecahan uang. Solusi akan tampil saat pemain tidak berhasil meminimumkan pecahan uang dan akan tetap

tampil juga saat pemain berhasil meminimumkan pecahan uang dengan metode algoritma *greedy*.



Gambar 3.9 Tampilan Soal Permainan 2

3.4 Pengujian Dengan Metode dan Pengujian Algoritma *Greedy*

Permainan yang telah dibuat akan diuji dengan menggunakan metode *black box testing* dan akan dilakukan juga pengujian terhadap algoritmanya, yaitu algoritma *greedy*.

Tabel 3.1 Pengujian Permainan dengan Metode *Black Box Tetsing*

No.	Kasus	Hasil yang Diharapkan	Hasil Uji
1.	Pilih <i>button</i> Mulai Permainan	Tampil halaman <i>input</i> nama setelah <i>button</i> mulai permainan diklik.	Berhasil
2.	Pilih <i>button</i> Petunjuk	Tampil halaman petunjuk setelah <i>button</i> petunjuk diklik. Halaman tersebut berisi informasi mengenai cara memainkan <i>game</i> .	Berhasil
3.	Pilih <i>button</i> Tentang	Tampil halaman tentang setelah <i>button</i> tentang diklik. Halaman tersebut memberikan data tentang pembuat <i>game</i> .	Berhasil
4.	Pilih <i>button</i> Keluar	<i>Message box</i> akan tampil setelah <i>button</i> keluar di pilih.	Berhasil
5.	Pilih <i>button</i> Permainan 1	Halaman permainan 1 yang akan tampil setelah pemain berhasil <i>input</i> nama	Berhasil
6.	Pilih <i>Button</i> Permainan 2	Halaman permainan 2 yang akan tampil setelah pemain berhasil <i>input</i> nama	Berhasil
7.	Pilih <i>Button</i> Kembali	Tampilan halaman utama yang akan tampil setelah <i>button</i> kembali diklik.	Berhasil
8.	Menekan tombol panah atas pada <i>keyboard</i>	Karakter permainan akan maju berjalan ke depan ketika menekan tombol panah atas pada <i>keyboard</i> .	Berhasil
9.	Menekan tombol panah bawah pada <i>keyboard</i>	Karakter permainan akan maju berjalan ke belakang ketika menekan tombol panah bawah pada <i>keyboard</i> .	Berhasil
10.	Menekan tombol panah kanan pada <i>Keyboard</i>	Karakter permainan akan maju berjalan ke kanan ketika menekan tombol panah kanan pada <i>keyboard</i> .	Berhasil
11.	Menekan tombol panah kiri pada <i>keyboard</i>	Karakter permainan akan maju berjalan ke kiri ketika menekan tombol panah kiri pada <i>keyboard</i> .	Berhasil
12.	Karakter permainan menemukan atau mendekati kotak harta karun	Kotak harta karun akan menghilang dan akan tampil soal permainan.	Berhasil
13.	Jawaban pemain dalam meminimalkan pecahan uang benar, yaitu sesuai dengan algoritma <i>greedy</i> .	Skor akan bertambah 10 dan akan tampil solusi dari algoritma <i>geedy</i> serta pemberitahuan bahwa pemain berhasil.	Berhasil
14.	Jawaban pemain salah dalam meminimalkan pecahan uang.	Skor tidak akan bertambah 10 dan akan tampil solusi dari algoritma <i>greedy</i> serta pemberitahuan bahwa jawaban pemain salah.	Berhasil
15.	Waktu	Saat waktu dalam permainan = 0, maka permainan akan selesai dan akan tampil top skor tertinggi dari pemain yang pernah memainkan permainan.	Berhasil

16.	Pemain menutup solusi permainan dari algoritma <i>greedy</i> .	<i>Label</i> kotak akan selalu bertambah 1 saat pemain menutup solusi dari permainan (algoritma <i>greedy</i>).	Berhasil
17.	Menemukan atau berhasil membuka 20 kotak harta karun.	Permainan akan selesai dan akan tampil tampilan halaman tiga top skor tertinggi dari pemain yang pernah memainkan permainan dan skor dari pemain yang baru saja memainkan permainan.	Berhasil
18.	Klik <i>button</i> X	Waktu dalam permainan akan berhenti dan Tampil pilihan <i>button</i> lanjutkan permainan dan <i>button</i> keluar permainan.	Berhasil
19.	Klik <i>button</i> lanjutkan permainan setelah mengklik <i>button</i> X.	Waktu dalam permainan akan berjalan lagi dan pemain dapat melanjutkan permainan.	Berhasil
20.	Klik <i>button</i> keluar permainan setelah mengklik <i>button</i> X.	Permainan yang sedang dimainkan akan menutup dan tampilan akan beralih atau menuju ke halaman pilihan permainan.	Berhasil

Tabel 3.2 Pengujian Algoritma *Greedy* dengan Uang yang Dimiliki

No	Jawaban Pemain	Algoritma <i>Greedy</i>	Hasil
1	Uang tujuan : Rp. 32000 Jika uang yang dimiliki : a. 2 lmbr. uang 10000 b. 3 lmbr. uang 5000 c. 0 lmbr. uang 2000 d. 5 lmbr. uang 1000 Penyelesaian : a. 2 lmbr. uang 10000 b. 2 lmbr. uang 5000 c. 0 lmbr. uang 2000 d. 2 lmbr. uang 1000	Uang tujuan : Rp. 32000 Jika uang yang dimiliki : a. 2 lmbr. uang 10000 b. 3 lmbr. uang 5000 c. 0 lmbr. uang 2000 d. 5 lmbr. uang 1000 Penyelesaian : a. 2 lmbr. uang 10000 b. 2 lmbr. uang 5000 c. 0 lmbr. uang 2000 d. 2 lmbr. uang 1000	Solusi dari algoritma <i>greedy</i> dengan uang yang dimiliki, berhasil 100% meminimalkan pecahan uang
2	Uang tujuan : Rp. 26000 Jika uang yang dimiliki : a. 0 lmbr. uang 10000 b. 7 lmbr. uang 5000 c. 1 lmbr. uang 2000 d. 3 lmbr. uang 1000 Penyelesaian : a. 0 lmbr. uang 10000 b. 5 lmbr. uang 5000 c. 0 lmbr. uang 2000 d. 1 lmbr. uang 1000	Uang tujuan : Rp. 21000 Jika uang yang dimiliki : a. 0 lmbr. uang 10000 b. 7 lmbr. uang 5000 c. 1 lmbr. uang 2000 d. 3 lmbr. uang 1000 Penyelesaian : a. 0 lmbr. uang 10000 b. 5 lmbr. uang 5000 c. 0 lmbr. uang 2000 d. 1 lmbr. uang 1000	Solusi dari algoritma <i>greedy</i> dengan uang yang dimiliki, berhasil 100% meminimumkan pecahan uang.

Tabel 3.3 Pengujian Algoritma *Greedy* Tanpa Syarat

No	Jawaban Pemain	Algoritma <i>Greedy</i>	Hasil
1	Uang tujuan : Rp. 32000 Penyelesaian : a. 3 lmbr. uang 10000 b. 0 lmbr. uang 5000 c. 1 lmbr. uang 2000 d. 0 lmbr. uang 1000	Uang tujuan : Rp. 32000 Penyelesaian : a. 3 lmbr. uang 10000 b. 0 lmbr. uang 5000 c. 1 lmbr. uang 2000 d. 0 lmbr. uang 1000	Solusi dari algoritma <i>greedy</i> tanpa syarat, berhasil 100% meminimalkan pecahan uang.
2	Uang tujuan : Rp. 26000 Penyelesaian : a. 2 lmbr. uang 10000 b. 1 lmbr. uang 5000 c. 0 lmbr. uang 2000 d. 1 lmbr. uang 1000	Uang tujuan : Rp. 21000 Penyelesaian : a. 2 lmbr. uang 10000 b. 1 lmbr. uang 5000 c. 0 lmbr. uang 2000 d. 1 lmbr. uang 1000	Solusi dari algoritma <i>greedy</i> tanpa syarat, berhasil 100% meminimumkan pecahan uang.

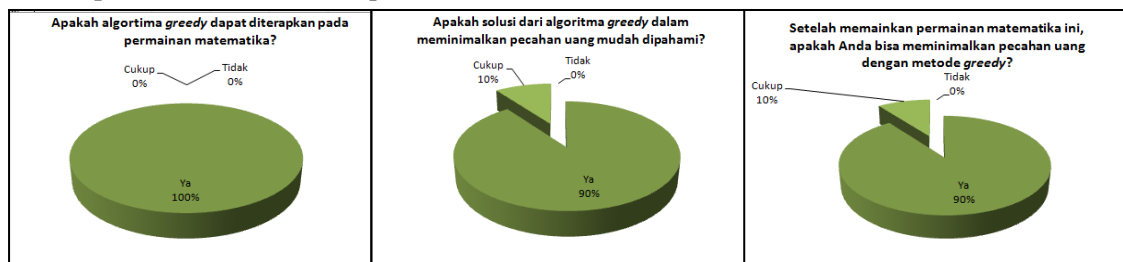
3.5 Pengujian Kuesioner Terhadap Kepuasan Pengguna Permainan

Kuesioner diberikan kepada 30 pengguna atau pemain untuk memberikan penilaian terhadap permainan yang telah dibuat dengan menerapkan algoritma *greedy*. Berikut ini adalah tabel hasil pengujian kuesioner yang telah diberikan yang dapat dilihat pada Tabel 3.4.

Tabel 3.4 Hasil Pengujian Kuesioner

No	Pernyataan	Jawaban		
		Ya	Cukup	Tidak
1	Apakah algoritma <i>greedy</i> dapat diterapkan pada permainan matematika?	30	0	0
2	Apakah solusi dari algoritma <i>greedy</i> dalam meminimalkan pecahan uang mudah dipahami?	27	3	0
3	Setelah memainkan permainan matematika ini, apakah Anda bisa meminimalkan pecahan uang dengan metode <i>greedy</i> ?	27	3	0

Dari hasil kuesioner yang telah dilakukan, dimana pada pernyataan 1 semua responden memilih “Ya”. Pernyataan 2 sebanyak 27 responden memilih “Ya” dan 3 responden memilih “Cukup”. Pernyataan 3 sebanyak 27 responden memilih “Ya” dan 3 responden memilih “Cukup”.



Gambar 3.10 Grafik Kuesioner Permainan

Gambar di atas menunjukkan grafik apakah algoritma *greedy* dapat diterapkan pada permainan matematika dengan responden memilih 100% “Ya”, 0% “Cukup” dan 0% “Tidak”. Menunjukkan grafik apakah solusi dari algoritma *greedy* dalam meminimalkan pecahan uang mudah dipahami dengan responden memilih 90% “Ya”, 10% “Cukup” dan 0% “Tidak”. Menunjukkan grafik, setelah memainkan permainan matematika ini, apakah Anda bisa meminimalkan pecahan uang dengan metode *greedy* dengan responden memilih 90% “Ya”, 10% “Cukup” dan 0% “Tidak”. Berdasarkan data-data tersebut dapat disimpulkan bahwa algoritma *greedy* dapat diterapkan pada permainan matematika dan solusi algoritma *greedy* dalam meminimalkan pecahan uang mudah dipahami serta setelah memainkan permainan matematika pengguna bisa meminimalkan pecahan uang dengan metode *greedy*.

3.6 Hasil Pengujian Metode *Black Box Testing*

Berdasarkan pengujian yang telah dilakukan dengan menggunakan metode *black box testing* pada Tabel 3.1, maka dapat diambil kesimpulan bahwa fitur-fitur yang ada pada permainan matematika dapat berfungsi sesuai dengan yang diinginkan.

3.7 Hasil Pengujian Algoritma *Greedy*

Berdasarkan pengujian algoritma yang telah dilakukan pada Tabel 3.2 dan 3.3 dengan cara menguji coba antara jawaban pemain dengan solusi dari algoritma *greedy* dengan syarat (dengan uang yang dimiliki) dan tanpa syarat (tanpa uang yang dimiliki). Dapat disimpulkan bahwa solusi dari algoritma *greedy* dengan syarat (dengan uang yang dimiliki) dan tanpa syarat (tanpa uang yang dimiliki) 100% dapat meminimalkan pecahan uang dengan seminimal mungkin.

4 KESIMPULAN

Berdasarkan hasil proses implementasi dan uji coba yang dilakukan, kesimpulan dari pengerjaan skripsi ini, yaitu :

1. Berdasarkan hasil pengujian terhadap algoritma, algoritma Algoritma *greedy* dapat meminimalkan pecahan uang.
2. Penelitian ini memiliki fitur untuk melihat petunjuk permainan, fitur lihat tentang pembuat, fitur untuk *input* nama pemain, fitur untuk mulai permainan, dan lain sebagainya.
3. Berdasarkan hasil pengujian dengan menggunakan metode *black box testing*, fitur-fitur yang terdapat pada permainan dapat berfungsi sesuai dengan yang diinginkan.
4. Berdasarkan hasil kuesioner, Solusi yang diberikan pada permainan (algoritma *greedy*) dalam meminimalkan pecahan uang sangat membantu, karena pengguna bisa mengetahui solusi atau jawaban yang benar dan sekaligus pengguna bisa belajar serta Permainan yang dibangun bersifat mendidik dan menyenangkan.

5 SARAN

Saran yang dapat direkomendasikan dalam menyelesaikan skripsi ini adalah :

1. Diberikannya peta untuk mengetahui letak atau posisi antara kotak harta karun dengan karakter permainan.
2. Penelitian selanjutnya dapat ditambahkan lagi nilai uang yang lebih besar dari Rp. 50000, karena pada permainan matematika ini nilai uang sebesar Rp. 20000 sampai Rp. 50000.
3. Permainan dapat dibuat versi *mobile* yang dapat dimainkan pada *smartphone*.

DAFTAR PUSTAKA

- [1] Harsono, Maruf 2014, *Pengaruh Permainan Game Terhadap Perkembangan Remaja*, Surya University, Tangerang
 - [2] Wijaya, Frans Setia, Nugroho, M. Adi, dkk, *Penerapan Algoritma Greedy pada Game Monopoli Indonesia Berbasis Unity 3D*, STMIK GI MDP, Palembang
 - [3] Supatmono, Catur 2009, *Matematika Asyik*, Grasindo
 - [4] Nugroho, Adi 2009, *Algoritma dan Struktur Data dengan C#*, Andi Offset, Yogyakarta
 - [5] Roedavan, Rickman 2014, *Unity : Tutorial Game Engine*, Informatika, Bandung
 - [6] Pranata, Baskara Arya 2015, *Mudah Membuat Game dan Potensi Finansialnya dengan Unity 3D*, Elex Media Komputindo, Jakarta
 - [7] Nugroho, Adi 2010, *Mengembangkan Aplikasi Basis Data Menggunakan C# dan SQL Server*, Andi Offset, Yogyakarta
 - [8] Mulyadi 2010, *Membuat Aplikasi untuk Android*, Multimedia Center Publishing, Yogyakarta
 - [9] A.S., Rosa dan Shalahuddin, M. 2013, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Informatika, Bandung
-