Aplikasi *gameplay* edukasi pencegahan obesitas dengan menggunakan algoritma *astar* dan *greedy* pada pencarian jalur makanan

M Abdul Fatah Z[*1)], Adi Wibowo[*2)]

[**]Jurusan Ilmu Komputer/Informatika, Fakultas Sains dan Matematika,
Universitas Diponegoro
[1)]vata_hero@yahoo.com, [2)]bowo.adi@undip.ac.id

Abstrak

Edukasi untuk obesitas sangatlah penting dilakukan. Salah satunya dapat dilakukan dengan game edukasi berbasis TD. Untuk dapat mengedukasi masyrakat tentang obesitas maka dibutuhkan gameplay yang memperlihatkan tentang makanan sehat, makanan tidak sehat dan kebutuhan olahraga. Pada tugas akhir ini kami menggunakan algoritma Astar dan Greedy sebagai gameplay yang diimplementasikan pada makanan sehat dan makanan tidak sehat. Aplikasi ini dibangun dengan dengan bahasa pemrograman C#, software Unity 3D dan metodologi XP. Hasil eksperimen menunjukkan bahwa waktu yang dibutuhkan algoritma Astar dari kotak start ke kotak finish lebih cepat dibandingkan dengan algoritma Greedy. Oleh sebab itu, pada makanan sehat akan digunakan pencarian jalur algoritma Greedy dan makanan tidak sehat menggunakan algortima Astar. Hasil survei menunjukkan responden lebih memahami edukasi tentang cara untuk mencegahan obesitas setelah memainkan game ini.

***Kata kunci** : Obesitas, Game, Edukasi, Astar, Greedy, TD, XP, Unity 3D.*

Abstract

Education for obesity is very important. One of them can be done with TD-based educational game. To be able to educate the public about obesity then it takes gameplay that shows about healthy food, unhealthy foods and sports needs. In this final project we use Astar and Greedy algorithm for the gameplay that is implemented on healthy food and unhealthy food. This application was built using the XP methodology and was created using Unity 3D software with C # programming language. EXPerimental results show that Astar algorithm's time cost is cheaper than Greedy algorithm's time cost. Based on that, on healthy food pathfinding we use Greedy algorithms as methode and Astar algorithm for an unhealthy foods pathfinding. In addition, results show after playing this game the public more understand on how to overcome obesity through the management of food and sports.

Keywords : *Obesity, Game, Education, Astar, Greedy, TD, XP, Unity 3D.*

1. PENDAHULUAN

Obesitas merupakan salah satu isu kesehatan terbesar yang kerap ditemui oleh masyarakat Indonesia bahkan dunia. Jumlah penderita obesitas di dunia naik dari 857 juta pada 1980 menjadi 2,1 milyar pada tahun 2013 (Lancet, 2014). Menurut Riskesnas (2016), penduduk Indonesia dewasa berusia diatas 18 tahun yang

mengalami kegemukan atau obesitas sebesar 20,7 persen. Sebagaimana dikutip dari CNN (2017), obesitas yang mengendap cukup lama menyebabkan banyak penyakit serius, seperti penyakit jantung, hipertensi, gangguan pernapasan, diabetes, stroke dan kemandulan.

Dari fakta diatas diperlukan cara untuk mencegah serta menanggulangi wabah obesitas. Beberapa solusi penanganan obesitas yang sudah ada diantaranya adalah dengan edukasi pengaturan pola makan serta olahraga teratur (Supriyanto, 2013). Salah satu cara edukasinya adalah dengan membuat *game* tentang pengelolaan pola makan untuk mencegah obesitas.

Game edukasi kesehatan sudah pernah dibuat oleh peneliti lainnya, diantaranya paper karya Brich (2015) tentang *game* edukasi *liver defense* untuk mengedukasi pemain tidak mengkonsumsi zat-zat yang dapat merusak hati seperti *ammonia* dan alkohol. Peneliti lain Bassilious (2011) merancang *game power defense* untuk mengedukasi tentang pencegahan dan pemantauan diabetes yang disebabkan kadar glukosa yang tinggi dalam darah sehingga memacu pankreas untuk memproduksi insulin terus menerus hingga akhirnya diabetes. Akan tetapi belum ada paper yang mengembangkan mengenai *game* edukasi untuk mengenalkan edukasi pengelolaan pola makan dan olahraga untuk mencegah serta mengangani obesitas. Sehingga pada penelitian ini akan dibuat *game* untuk mengedukasi tentang pengelolaan pola makan dan olahraga untuk menangani obesitas.

TD merupakan salah satu jenis *game* yang banyak diminati dan dikembangkan untuk edukasi. Hal ini dikarenakan *game* tersebut mudah difahami dan sesuai untuk berbagai usia. Pada *game TD* diperlukan implementasi algoritma pada *monster* untuk mencari jalan dari titik awal ke titik *finish*. Beberapa algoritma yang dapat diterapkan untuk pencarian jalur adalah algortima *Astar* dan *Greedy*.

Berdasarkan latar belakang diatas, pada penelitian ini akan dibuat *game* edukasi obesitas mencakup pengelolaan pola makan dan olahraga. Selain itu pada *game* ini akan dibandingkan dua algoritma yakni *Astar dan Greedy*. Hasil keluaran yang diinginkan adalah algoritma yang membutuhkan waktu lebih lama pada pencarian jalur akan digunakan pada *gameplay* makanan sehat. Sebaliknya algoritma yang lebih cepat akan digunakan pada *gameplay* makanan sehat.

2. LANDASAN TEORI

2.1 PENGERTIAN OBESITAS

Kegemukan atau obesitas adalah suatu kondisi medis berupa kelebihan lemak tubuh yang terakumulasi sedemikian rupa sehingga menimbulkan dampak merugikan bagi kesehatan, yang kemudian menurunkan harapan hidup dan/atau meningkatkan masalah kesehatan (WHO, 2000). Seseorang dianggap menderita kegemukan bila *BMI*, yaitu ukuran yang diperoleh dari hasil pembagian berat badan dalam kilogram dengan kuadrat tinggi badan dalam meter, lebih dari 30 kg/m^2 (WHO, 2000).

Terdapat berbagai cara untuk menentukan apakah seseorang disebut obesitas atau tidak. Salah satu diantaranya dengan pengaturan gizi makanan dan teknik antrophometrik yakni berdasarkan *BMI*. Menurut Hartley (2018) perhitungan dalam pencarian *BMI* dengan persamaan:

$$BMI = \frac{Berat\ Badan}{Tinggi\ Badan \times Tinggi\ Badan} \quad (1)$$

Definisi indeks tabel obesitas yang paling sering dipakai adalah yang dikutip dari WHO (2000) seperti yang tertera pada tabel 1:

Tabel 1 Indeks *BMI* (WHO, 2000)

BMI	**Klasifikasi**
< 18.5	Berat badan kurang
18.5–24.9	Normal
25.0–29.9	Berat badan lebih
30.0–34.9	Kegemukan kelas I
35.0-39.9	Kegemukan kelas II
≥ 40.0	Kegemukan kelas III

2.2 CARA MENCEGAH OBESITAS

Menurut Wijaya (2015) bahwa dalam pencegahan dan penanganan obesitas dapat dilakukan tindakan diantaranya seperti :

a) Mengurangi makanan berkalori, berlemak dan beralkohol.
b) Perbanyak sayuran, buah-buahan.
c) Mengkonsumsi nasi merah, kacang-kacangan, daging tanpa lemak dalam jumlah cukup.
d) Mengurangi makanan bergaram.
e) Melakukan aktivitas fisik dan berolahraga seperti jalan kaki 1 jam minimal 3 kali seminggu.
f) Menghindari penurunan berat badan yang terlalu instan.

2.3 PENGERTIAN *GAME*

Game adalah salah satu bentuk hiburan yang dapat dijadikan sebagai penyegar pikiran dari kepenatan akibat dari padatnya aktivitas sehari-hari (Fauzia, 2014). Dalam Kamus Besar Bahasa Indonesia, *game* diartikan sebagai permainan. Permainan adalah melakukan sesuatu dimana kondisi akhirnya menghasilkan menang atau kalah. Permainan yang dimaksud disini dalah permainan yang mampu mengedukasi para pemainnya.

2.4 *GAME* EDUKASI

Game edukasi adalah salah satu jenis media yang digunakan dalam memberikan pengajaran melalui permainan dengan tujuan untuk merangsang daya pikir dan meningkatkan konsentrasi melalui media yang unik dan menarik (Handriyantini, 2009).

2.5 *TOWER DEFENSE*

Menurut Philipa Avery dalam jurnalnya yang berjudul *Computational Intelligence and Tower Defence Game, TD* adalah salah satu *game* ber-*genre* strategi yang berfokus pada pengaturan sumber daya berupa *tower* serta penempatannya. Dalam *game* TD, *player* ditugaskan untuk membeli dan mengorganisir menara pertahanan yang nantinya akan menyerang satu set musuh (*creeps*) yang terdiri dari beberapa jenis yang berbeda. Untuk setiap *creep* yang mati, *player* akan mendapatkan sumber daya yang nantinya akan digunakan untuk membuat menara pertahanan yang lain. Apabila *player* berhasil menahan *creep* dengan menaranya agar tidak sampai tempat yang dilindungi dalam waktu tertentu, maka *player* dinyatakan menang (Avery, 2011).

2.6 *ASTAR PATHFINDING*

Sebagaimana yang dirangkum dari sumber yang dikutip dari Wenderlich (2011) bahwa iterasi yang dilakukan pada pencarian jalur dengan algoritma *Astar* adalah sebagai berikut :

1) Menambahkan kotak awal ke *openlist*.
2) Mengambil kotak dengan nilai F terkecil dan diset menjadi kotak sekarang.
3) Menambah kotak sekarang ke *closelist*
4) Menghapus kotak sekarang dari *openlist*.
5) Apabila *closelist* mencapai kotak akhir maka pencarian jalur dihentikan.

6) Mencari kotak tetangga disekitar kotak sekarang.
7) Apabila kotak tetangga sudah *closelist* maka diabaikan.
8) Apabila kotak tetangga *openlist* maka dihitung apakah nilai G kotak sekarang kurang dari nilai G tetangga. Apabila kurang dihitung nilai G, H dan F nya.
9) Kotak tetangga selain itu ditambahkan ke *openlist* dan dihitung nilai G, H dan F nya.
10) Diulangi ke langkah (1) sampai sudah tidak terdapat *openlist* (pencarian jalur selesai).

2.7 *OPENLIST* DAN *CLOSELIST*

Definisi *openlist* adalah *list* untuk menyimpan lantai akan dipakai untuk menemukan jalur terpendek. Definisi *closelist* adalah *list* untuk menyimpan lantai sudah dilewati sebagai jalur terpendek.

2.8 DEFINISI NILAI F, G DAN H

G adalah harga perpindahan yang dibutuhkan dari kotak awal ke kotak berikutnya. Nilai G akan bertambah apabila kotak semakin menjauh dari kotak awal.

H adalah harga perkiraan perpindahan dari titik saat ini ke titik akhir. Disebut sebagai nilai heuristik karena kita tidak tau nilai pastinya (perkiraan). Biasanya digunakan metode *"manhattan distance"* untuk menghitung jumlah kotak *horizontal* dan *vertical* untuk mencapai titik akhir tanpa memperhitungkan halangan yang ada.

F adalah nilai aktual atau jalur sebenarnya yang diperoleh dengan menambahkan nilai G dan nilai H.

2.9 *GREEDY PATHFINDING*

Algoritma *Greedy* merupakan algoritma dengan mencari nilai LM, yakni nilai maksimum sementara untuk menemukan solusi optimal. Namun seringkali hasilnya justru bukan merupakan hasil yang optimal. Berikut adalah iterasi yang dilakukan pada algoritma *Greedy* (Anugerah, 2016) :

1. Memeriksa titik awal dan memeriksa apabila titik sekarang adalah titik tujuan. Apabila titik sekarang adalah titik tujuan maka iterasi dihentikan.
2. Periksa titik sekitar yang terhubung dengan titik saat ini.
3. Dicari nilai LM dari titik sekarang ke titik selanjutnya.
4. Pilih titik dengan LM terkecil dan menandai titik sekarang sebagai titik yang telah dilalui. Apabila titik selanjutnya sudah pernah dilalui maka diabaikan.
5. Kembali ke langkah (1) sampai titik tujuan ditemukan.

2.10 *AGILE SOFTWARE DEVELOPMENT*

Agile software development atau disebut juga *agile software process* merupakan sebuah proses pengembangan perangkat lunak yang efektif dan responsif bila terjadi perubahan secara cepat pada perangkat lunak yang sedang dibangun. Menurut (Pressman, 2005), penggunaan *agile software process* didasari oleh 3 asumsi yang biasanya terjadi dalam sebagian besar proyek perangkat lunak, yaitu (Pressman, 2005) :

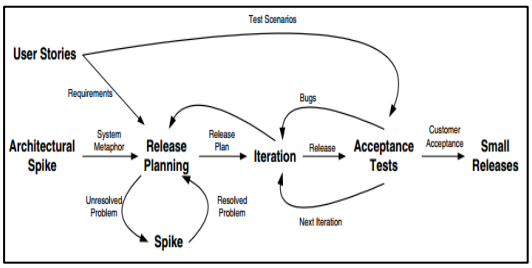
1. Sulit memprediksi persyaratan perangkat lunak di awal dan yang akan berubah.
2. Membutuhkan kegiatan desain dan konstruksi yang bersamaan, dalam artian agar model desain yang di ajukan benar-benar terbukti diciptakan.
3. Analisis, desain, konstruksi, dan pengujian yang tidak dapat diprediksi (dari sudut pandang perencanaan) seperti yang diinginkan.

2.11 *EXTREME PROGRAMMING*

Menurut Beck (1999), XP adalah sebuah pendekatan atau model pegembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel.

2.12 *XP LIFECYCLE*

Menurut Hunt (2006) dalam bukunya berjudul *"Agile Software Construction"* menggambarkan *lifecyle XP* pada gambar 1.



Gambar 1 *XP Lifecycle*

3. METODE PENELITIAN

Metode penelitian yang digunakan adalah *Extreme Programming (XP)*. Langkah-lagkah metodologinya ada 5 tahap utama, yaitu:

1. *User story*, tujuan dari *user stories* seperti halnya *use case* yakni untuk menerangkan bagaimana *user* akan menggunakan sistem yang akan dibuat.
2. *Release plan*, merupakan proses menentukan *user stories* mana yang akan diimplementasikan pada tiap iterasi.
3. *Iterasi*, iterasi menentukan seberapa cepat pengembang *software* merespon perubahan. Semakin kecil iterasi maka semakin mudah dalam mengaplikasikan perubahan yang terjadi. Jumlah iterasi dapat diperoleh dengan membagi jumlah seluruh *story point* dengan nilai *velocity* yang ada.
4. *Acceptance testing*, *Acceptance Test* dibuat dari *user stories* yang ditulis pada awal *project*. Tiap iterasi mengimplementasikan satu atau lebih *user stories*. *Stories* akan dijabarkan menjadi *acceptance test* pada iterasi.
5. *Small release*, apabila ada hasil yang bermanfaat maka di-*release* ke *end user* tanpa menunggu seluruh sistem selesai terlebih dahulu.

4. DEFINISI KEBUTUHAN DAN *RELEASE PLANNING*

Pada bab ini dijelaskan definisi kebutuhan melalui *user story card* (gambar 2). Pada user story terdapat 4 poin dasar user story (Beck, 1999):

a. Nama *user story*
b. Deskripsi tujuan *user story* berupa paragraf pendek
c. Estimasi waktu berapa lama *user story* akan diimplementasikan
d. Tingkat kepentingan *user story (*seperti *must have, should have)*.

***Story* Name :** Tampilan Awal *Game*
***Story* :**
Saya ingin ada tampilan awal *game* sebelum masuk kedalam *game*. Saya juga ingin setelah itu dapat melihat *tutorial* bermain sebelum bermain *game.*
***Priority* :** *Could have*
***Estimate* :** 3
***Acceptance Test Criteria* :**
1) Halaman awal mempunyai tombol yang mengarahkan ke permainan
2) Halaman awal mempunyai tombol yang mengarahkan ke *tutorial*
3) Halaman awal mempunyai tombol yang mengarahkan keluar *game*
4) Menampilkan halaman awal

Gambar 2 *User Story Card*

Kemudian direncanakan *relase planning* yang meliputi perencanaan *estimasi story*, *priority story* dan pembagian iterasi untuk setiap *story*.

Pada penelitian ini, direncanakan implementasi *program* dapat terselesaikan dalam waktu 3 bulan atau 60 hari aktif kerja.

Hari aktif kerja disini mencakup hari *weekday*. *Velocity* merupakan waktu yang diperlukan untuk menyelesaikan tiap iterasi. Nilai *velocity* yang digunakan adalah 15. Estimasi *story* dapat dilihat pada tabel 2.

Tabel 2 Estimasi *Story*

No.	*User Story*	Deskripsi	Estimasi
1.	US – 01	Tampilan Awal *Game*	3
2.	US – 02	*Tutorial Game*	3
3.	US – 03	*Input Data* Awal *Game*	3
4.	US – 04	Fungsi Menghitung *BMI*	3
5.	US – 05	Menu *Loading*	3
6.	US – 06	Lantai *Game*	3
7.	US – 07	*Portal* Keluar dan *Portal* Masuk Makanan	3
8.	US – 08	*Index* Penunjuk *BMI*, Bobot dan Energi	3
9.	US – 09	*Tower*	6
10.	US – 10	Energi sebagai Mata Uang *Game*	3
11.	US – 11	Makanan sebagai Musuh	6
12.	US – 12	Fungsi Pencarian Jalur Makanan	4
13.	US – 13	Peluru untuk Menyerang Makanan	9
14.	US – 14	*Level Game*	3
15.	US – 15	Pengkondisian Menang dan Kalah	5
Total Story Points			60

Story yang sudah dibuat kemudian diprioritaskan berdasarkan tingkat kepentingannya. Untuk *story* yang harus ada akan dikelompokkan dalam kelas *must have*. Kemudian *story* penunjang dapat dikelompokkan dalam kelas *should have*. Sedangkan *story* yang tidak terlalu penting dikelompokkan dalam kelas *could have*. Pembagian prioritas *story* dapat dilihat pada tabel 3.

Tabel 3 *Priority Story*

No.	Kode *Story*	Deskripsi	Estimasi
Must Have			
1.	US – 03	*Input Data* Awal *Game*	3
2.	US – 06	Lantai *Game*	3
3.	US – 07	*Portal* Keluar dan *Portal* Masuk Makanan	3
4.	US – 09	*Tower*	6
5.	US – 11	Makanan sebagai Musuh	6
6.	US – 12	Fungsi Pencarian Jalur Makanan	4
7.	US – 13	Peluru untuk Menyerang Makanan	9
8.	US – 15	Pengkondisian Menang dan Kalah	5
Should Have			
9.	US – 02	*Tutorial Game*	3
10.	US – 10	Energi sebagai Mata Uang *Game*	3
11.	US – 14	*Level Game*	3
Could Have			
12.	US – 01	Tampilan Awal *Game*	3
13.	US – 04	Fungsi Menghitung *BMI*	3
14.	US – 05	Menu *Loading*	3
15.	US – 08	*Index* Penunjuk *BMI*, Bobot dan Energi	3

Dalam mencari jumlah iterasi digunakan dengan membagi total *story* dengan *velocity*. Penelitian ini memiliki *story points* yang berjumlah 60 dan nilai *velocity* adalah 15, maka dapat diketahui jumlah iterasi adalah 60/15 = 4 iterasi. Tiap iterasi pada pembuatan *game* edukasi obesitas ini akan diselesaikan dalam kurun waktu 3 minggu atau 15 hari aktif kerja. Pembagian iterasi dan alokasi *story* dapat dilihat pada tabel 4.

Tabel 4 Pembagian Iterasi *Story*

No.	Kode *Story*	Deskripsi	Estimasi
Iteration – 1			
1.	US – 01	Tampilan Awal *Game*	3
2.	US – 02	*Tutorial Game*	3
3.	US – 03	*Input Data* Awal *Game*	3
4.	US – 04	Fungsi Menghitung *BMI*	3
5.	US – 05	Menu *Loading*	3
Velocity			15
Iteration – 2			
6.	US – 06	Lantai *Game*	3
7.	US – 07	*Portal* Keluar dan *Portal* Masuk Makanan	3
8.	US – 09	*Tower*	6
9.	US – 10	Energi sebagai Mata Uang *Game*	3
Velocity			15
Iteration – 3			
10.	US – 13	Peluru untuk Menyerang Makanan	9
11.	US – 11	Makanan sebagai Musuh	6
Velocity			
Iteration – 4			
12.	US – 12	Fungsi Pencarian Jalur Makanan	4
13.	US – 08	*Index* Penunjuk *BMI*, Bobot dan Energi	3
14.	US – 15	Pengkondisian Menang dan Kalah	5
15.	US – 14	*Level Game*	3
Velocity			15

5. HASIL DAN PEMBAHASAN

Pada subbab ini dijelaskan implementasi pembangunan *game* pada setiap iterasi yang telah dibuat. Selain itu akan dilihat hasil

perbandingan algoritma *Astar* dan *Greedy* sebagai *gameplay* pada game edukasi obesitas ini. Pembahasan perancangan game meliputi implementasi *acceptance test*. Pada implementasi *acceptance test* menggunakan metode *Acceptance Test Driven Development* (TDD). TDD *Acceptance Test* memiliki siklus, yaitu:

1. Penulisan *unit test* dari *story*,
2. Implementasi kode *test*,
3. Implementasi *story*,
4. Implementasi antarmuka eksekusi *story*.

Penulisan *unit test* / *acceptance test* adalah penjabaran dari *story* yang dibuat *user* oleh tim developer. Tiap *story* dapat memiliki satu atau lebih *unit test*. Contoh *unit test* tampilan awal *game* dapat dilihat pada tabel 5.

Setelah didefinisikan unit test kemudian diimplementasikan fungsi pada tiap unit test yang ada. Contoh implementasi kode *test* tampilan awal *game* dapat dilihat pada tabel 6.

Kemudian setelah pengimplementasian tiap unit sudah dilakukan dibuat implementasi total dari *story* tampila awal *game*. Pengimplementasian *story* dapat dilihat pada tabel 7.

Tabel 5 Penulisan *Unit Test*

No	*Acceptance Test Story* US-01
1	Halaman awal mempunyai tombol yang mengarahkan ke permainan.
2	Halaman awal mempunyai tombol yang mengarahkan ke *tutorial*.
3	Halaman awal mempunyai tombol yang mengarahkan keluar *game.*
4	Menampilkan halaman awal

Tabel 6 Implementasi Kode *Test*

No	Kutipan Kode *Unit Test Story* US-01	Deskripsi
1	`public void ShowMenu`*`Input`*`(){` `  menu`*`Game`*`.SetActive(false);`	Fungsi untuk menampilkan menu *input*
	`}`	

2	`public void ShowTutor()`
	`{`
	`  menuTutor.SetActive(true);`
	`}`
		Fungsi untuk menampilkan menu *tutorial*

3	`public void Exit`*`Game`*`(){`
	`  Application.Quit();`
	`}`
		Fungsi untuk keluar *game*

4	`public void ShowMain(){`
	`menu`*`Game`*`.SetActive(true);`
	`  menuInfo`*`BMI`*`.SetActive(false);`
	`  menu`*`Input`*`.SetActive(false);`
	`  `*`video`*`Tutor.SetActive(false);`
	`  `*`Video`*`Manager.Instance.`*`videoPlayer`*`.Stop ();`
	`  Manager`*`Input`*`.Instance.Reset`*`Data`*` ();`
	`}`
		Fungsi untuk menampilkan menu awal

Tabel 7 Pengimplementasian *Story*

Kutipan Implementasi Story Membuat Tampilan Awal Game
`public class MenuManager:Singleton<MenuManager>{` `    private `*`Game`*`Object manu`*`Input`*`;` `    public void ShowMenu`*`Input`*`(){` `        menu`*`Game`*`.SetActive(false);` `        menuInfo`*`BMI`*`.SetActive(false);` `        menu`*`Input`*`.SetActive(true);` `        menuTutor.SetActive(false);` `    }` `    private `*`Game`*`Object menuTutor;` `    public void ShowTutor(){` `        menu`*`Game`*`.SetActive(false);` `        menuInfo`*`BMI`*`.SetActive(false);` `        menu`*`Input`*`.SetActive(false);` `        menuTutor.SetActive(true);` `    }` `    public void Exit`*`Game`*`(){` `        Application.Quit();` `    }` `    private `*`Game`*`Object mainMenu;` `    public void ShowMain(){` `        infoMenu.SetActive(false);` `        tutorMenu.SetActive(false);` `        mainMenu.SetActive(true);` `*`pregame`*`Menu.SetActive(false);` `   }` `}`

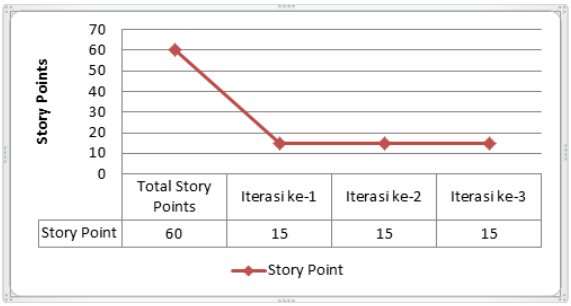
Setelah dilakukan pengimplementasian story kemudian

diimplementasikan antarmuka story. Contoh pengimplementasian antarmuka *story* dapat dilihat pada gambar 3.

Gambar 3 Implentasi Antarmuka *Story*

5.1 *ITERATION BURNDOWN CHART*

Iteration Burndown chart ini merangkum perkembangan implementasi *story* pada keseluruhan iterasi. Gambar 4 menunjukkan grafik *burndown* pada *iteration planning* yang disusun.



Gambar 4 *Iteration Burndown Chart*

5.2 HASIL PERBANDINGAN ALGORITMA *ASTAR* DAN *GREEDY* PADA PENCARIAN JALUR MAKANAN

Berdasarkan perbandingan kedua algoritma pada tabel 8, maka dapat disimpulkan bahwa algoritma *Astar* lebih efisien dalam melakukan pencarian jalur makanan sehat dan makanan tidak sehat dibandingkan algoritma *Greedy*. Oleh karenanya, algoritma *Greedy* digunakan pada *gameplay* makanan sehat dan algoritma Astar digunakan pada *gameplay* makanan sehat.

Tabel 8 Perbandingan waktu pencarian jalur algoritma *Astar* dan *Greedy*

No	*Stage*	Makanan Sehat (detik)		Makanan Tidak Sehat (detik)	
		Astar	*Greedy*	*Astar*	*Greedy*
1	*Normal*	20.33	21.43	20.33	21.43
2	*Easy1*	12.97	14.19	12.97	14.19
3	*Easy2*	14.83	15.81	14.83	15.81
4	*Easy3*	9.43	10.65	9.43	10.65
5	*Medium1*	11.29	12.29	11.29	12.29
6	*Medium2*	9.81	10.63	9.81	10.63
7	*Medium3*	8.45	8.90	8.45	8.90
8	*Hard1*	4.89	5.73	4.89	5.73
9	*Hard2*	9.53	10.44	9.53	10.44
10	*Hard3*	8.69	9.93	8.69	9.93
Total		110.2	120	129.11	110.2
Rata-rata		11.02	12	11.02	12

6. KESIMPULAN DAN SARAN

Dalam tugas akhir ini dihasilkan *game* edukasi obesitas dan perbandingan algoritma *Astar* dan *Greedy* pada pencarian jalur makanan sehat dan makanan tidak sehat menuju ke titik akhir (makanan dikonsumsi). Hasilnya algoritma *Astar* lebih optimal dalam melakukan pencarian jalur dibandingkan algoritma *Greedy*. Sehingga disini digunakan algoritma *Astar* sebagai *gameplay* pada makanan tidak sehat dan algoritma *Greedy* sebagai *gameplay* pada makanan sehat.

Hasil analisis perbandingan algoritma *Astar* dan *Greedy* menunjukkan bahwa algoritma *Astar* lebih cepat menuju *finish* dibandingkan algoritma *Greedy* dalam pencarian jalur berdasarkan tabel perbandingan. Pada pengembangan tugas akhir ini digunakan empat iterasi dalam penyelesaiannya dan menghasilkan aplikasi yang memenuhi *user story* yang telah ditentukan.

Beberapa saran untuk pengembangan game ini diantaranya adalah menambahkan detail kategori lain seperti karbohidrat, lemak, *protein* sebagai bahan pertimbangan pemilihan status gizi. Selain itu dapat juga ditambahkan senjata pada makanan

sehingga makanan dapat menghancurkan *tower*. Dalam pencaroan jalurnya dapat digunakan algoritma lain seperti *Greedy Best First Search* dimana nilai F / nilai sebenarnya sama dengan nilai H / heuristik.

7. DAFTAR PUSTAKA

ANUGERAH, M. J., HASBI, M., PRIBADI, R. P., YOANNITA, 2016. Penerapan Algoritma Greedy Untuk Pencarian Jalur Terpendek Pada Jakabaring Sport City. Palembang*:* STMIK GI MDP.

AVERY, P., 2011. Computational Intelligance and Tower Defence Games. USA: University of Nevada.

BASSILIOUS, E., DECHAMPLAIN, A., MCCABE, I., STEPHAN, M., KAPRALOS, B., MAHMUD, F., et al., 2011. Power Defense : A Video Game for Improving Diabetes Numeracy. In Proceedings of the IEEE Congress on Evolutionary Computation, December 29, 2011, Orange, CA, USA.

BECK, K, ANDRES, C., 1999. Extreme Programming Explained: Embrace Change. Boston: Addison-Wesley Longman Publishing Co.

BRICH, J., ROGERS, K., FROMMEL, J., WEIDHAAS, M., BRUCKNER, A., DORN, S. M., et al., 2015. LiverDefense: Using A TD Game As A Customisable Research Tool. IEEE Congress on Evolutionary Computation. Skovde, Sweden, 16-18 October 2011.

CNN. 2017. Semakin Banyak Orang Indonesia Alami Obesitas. CNN Indonesia, [online] Tersedia di: < https://www.cnnindonesia.com/ gaya-hidup/20170302223030-255- 197500/ semakin-banyak-orang-indonesia-alami-obesitas > [Diakses 3 Agustus 2018]

FAUZIA. 2014. Pengertian Game Menurut Para Ahli. Mandala Maya, [online] Tersedia di: < http://www.mandalamaya.com/pengertian-*game*-menurut-para-ahli/> [Diakses 10 Maret 2017]

HANDRIYANTINI, E., 2009. Permainan Edukatif (Educational Games) Berbasis Komputer untuk Siswa Sekolah Dasar. Konferensi dan Temu Nasional Teknologi Informasi dan Komunikasi untuk Indonesia, Bandung, Jawa Barat, 24-25 June 2009.

HUNT, J., 2006. Agile Software Construction. London: Springer-Verlag.

KOSKELA, L., 2008. Test Driven: Practical TDD And Acceptance TDD For Java Developers. London: Manning Publications.

LANCET. 2014. Global, Regional and Native Prevalence of Overweight and Obesity in Children and Adultsduring 198-2013: A Sysmatic Analysis for the Global Burden of Disease Study.The Lancet, [online] Tersedia di: < http://www.thelancet.com/journals/lancet/ article/PIIS0140-6736(14)60460-8/abstract> [Diakses 9 April 2017]

PRESSMAN, R. S., 2005. Software Engineering: A Practitioner's Approach. New York: McGraw-Hill.

SUPRIYANTO, A., 2013. Obesitas, Faktor Penyebab dan Bentuk-bentuk Terapinya. Yogyakarta: Universitas Negeri Yogyakarta.

WENDERLICH, R., 2011. Raywenderlich.com, [online] Tersedia di: < https://www.raywenderlich.com/4946/introduction-to-a-pathfinding > [Diakses 1 Juli 2009]

WHO. 2000. Obesity: preventing and managing the global epidemic. Geneva: World Health Organization.

WIJAYA, R., 2015. Cara Mengatasi Obesitas. ALODOKTER, [online]

Tersedia di: < ply/73658/> [Diakses 10 Maret 2017]
http://www.alodokter.com/kom*unit*as/re