BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi merupakan tahap kelanjutan dari kegiatan perancangan sistem. Wujud dari hasil implementasi ini nantinya adalah sebuah sistem yang siap untuk diuji dan digunakan.

4.1.1 Impelementasi Perangkat Keras

Berikut adalah spesifikasi perangkat keras (*hardware*) yang digunakan untuk menjalankan *game save your ricefiled*.

Tabel 4. 1 Perangkat Keras yang Digunakan

Nama Perangkat	Spesifikasi
Processor	CPU 1.80 Ghz
Memory	1 GB
HDD	1 GB Free Hardisk
Display	Resolusi 1024 x 768
Mouse	1 Unit
Keyboard	1 Unit

4.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak (*software*) yang digunakan untuk menjalankan *game save your ricefield* dapat dilihat pada tabel 4.2.

Tabel 4. 2 Perangkat Lunak yang Digunakan

Nama Perangkat Lunak	Keterangan
Windows	Sebagai sistem operasi
Flash Player 9/10/11	Sebagai *player*
Adobe Flash CS 5.5	Sebagai program pembangun

4.1.3 Impelementasi Instalasi *Game*

Sebelum dapat memainkan *game save your ricefield*, *game* harus tersimpan di direktori komputer. Petunjuk dalam instalasi *game* ini adalah sebagai berikut:

1. *Copy file Save Your Rice Field*.swf pada direktori komputer
2. *Install flash player* untuk menjalankan *game*
3. Buka *file Save Your Rice Field*.swf untuk menjalankan *game*

4.1.4 Implementasi Antar Muka

Implementasi antar muka merupakan tahap dimana sebuah sistem siap dioperasikan pada tahap yang sebenarnya, sehingga bisa diketahui apakah sistem yang telah di buat ini sudah sesuai dengan apa yang sudah di rancang sebelumnya.

1. Tampilan Menu Utama *Game Save Your Ricefield*

Saat pemain membuka *game save your ricefield* pertama kali, maka antar muka yang ditampilkan oleh sistem adalah antar muka menu utama. Pemain dapat memilih menu *play now* untuk dapat memainkan permainan. Selain itu pemain bisa mengatur *sound* dan *music* pada *game*. Tampilan menu utama *game save your ricefield* dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Tampilan Menu Utama *Game Save Your Ricefield*

2. Tampilan Pembuatan Profil Pemain

Pembuatan profil bertujuan menyimpan data pemain dan data permainan yang nantinya dapat di mainkan kembali. Tampilan pembuatan profil merupakan tampilan pesan yang muncul ketika pemain memilih menu *play now*. Berikut adalah tampilan pembuatan profil *game save your ricefield*.



Gambar 4. 2 Tampilan Pembuatan Profil Pemain

3. Tampilan Pesan Intruksi Permainan

Tampilan intruksi permainan tampil ketika pemain pertama kali memulai permainan yang berisikan petunjuk-petunjuk dalam memainkan *game save your ricefield*. Tampilan intruksi permainan dapat dilihat pada Gambar 4.3 dibawah ini.



Gambar 4. 3 Tampilan Pesan Intruksi Permainan

4. Tampilan Lahan Permainan

Setelah sistem selesai menampilkan pesan intruksi, maka tampilan lahan pertanian dapat dimainkan. Lahan permainan berisikan semua *resource* yang dapat digunakan pemain untuk memenangkan permainan. Pada lahan permainan pemain dapat menanam dan merawat padi, serta membasmi hama yang akan menyerang padi. Berikut adalah tampilan lahan permainan pada *game save your ricefield*.



Gambar 4. 4 Tampilan Lahan Permainan

5. Tampilan Pesan Info Predator

Tampilan info predator merupakan tampilan pesan yang berisikan keterangan predator yang dapat digunakan pemain untuk memburu hama. Tampilan info predator akan muncul sekali untuk setiap jenis predator dalam setiap permainan. Berikut adalah tampilan pesan info jenis predator yang ditunjukkan pada Gambar 4.5.



Gambar 4. 5 Tampilan Pesan Info Predator

6. Tampilan Pesan Info Hama

Tampilan info hama berisikan keterangan jenis hama yang muncul dilahan permainan. Sistem akan menampilkan tombol *new_hama* pada kiri lahan permainan sebagai tanda bahwa ada jenis hama baru yang muncul dilahan permainan. Pemain dapat menampilkan pesan info hama tersebut dengan mengklik tombol *new_hama* tersebut. Berikut adalah tampilan pesan info hama yang ditunjukkan pada Gambar 4.6.



Gambar 4. 6 Tampilan Pesan Info *New Hama*

7. Tampilan Pesan Info *Tips*

Tampilan pesan info *tips* berisikan *tips* atau saran yang berikan kepada pemain untuk dapat memenangkan permainan. Pesan info *tips* akan muncul beberapa saat setelah pemain memulai permainan yang ditandakan dengan muncul tombol disisi kiri atas lahan permaian. Pemain dapat membuka pesan tersebut dengan mengklik tombol tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 4.7.



Gambar 4. 7 Tampilan Pesan Info *Tips*

8. Tampilan *Option* (Pengaturan)

Tampilan *option* dapat digunakan pemain untuk mengaktifkan dan menonaktifkan *sound* dan *music game*. Selain itu tampilan *option* juga dapat digunakan pemain untuk kembali ke menu pembuka atau mengulang permainan. Berikut tampilan *option game save your ricefield*.



Gambar 4. 8 Tampilan *Option Game Save Your Ricefield*

9. Tampilan Pesan *Victory* (Kondisi Menang)

Tampilan pesan *victory* akan muncul ketika pemain berhasil menyelesaikan jumlah misi yang ditentukan pada *level* tersebut dalam waktu kurang dari waktu permainan.Tampilan *victory* dapat dilihat pada Gambar 4.9.



Gambar 4. 9 Tampilan Pesan *Victory*

10. Tampilan Pesan *Lose* (Kondisi Kalah)

Tampilan pesan *lose* tampil ketika pemain tidak berhasil menyelesaikan sejumlah misi pada *level* tersebut. Pada pesan *lose* sistem akan memberikan pilihan kepada pemain yaitu mengulang permainan pada *level* tersebut atau mengakhiri permainan dengan memilih tombol *quit*. Berikut adalah tampilan pesan *lose* pada *game save your ricefield*.



Gambar 4. 10 Tampilan Pesan *Lose*

4.2 Pengujian Sistem

Pengujian sistem dilakukan bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

Pengujian yang akan digunakan dalam menguji *game save your ricefield* ini adalah dengan menggunakan pengujian *alpha* (fungsional) dan pengujian *beta*.

4.2.1 Pengujian Alpha

Pengujian dilakukan terhadap *game* untuk memastikan bahwa *game* dapat berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. Pengujian alpha berfokus pada persyaratan fungsional perangkat lunak.

4.2.1.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game save your ricefield*. Berikut ini merupakan skenario pengujian dari aplikasi *game* dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Skenario Pengujian *Game Save You Ricefield*

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
1	Menu Utama	Pilih Menu *Play Now*	*Black Box*
		Pilih *New Game*	*Black Box*
		Isi profil baru	*Black Box*
		Pilih Tombol *Close*	*Black Box*
		Pilih Tombol *Sound* dan *Music*	*Black Box*
2	Lahan Permainan	Tampil *Intructions*	*Black Box*
		Pilih Tombol *Next Intructions*	*Black Box*
		Pilih Tombol *Skip Intructions*	*Black Box*
		Pilih Kotak Predator	*Black Box*
		Pilih Kotak Biji Benih	*Black Box*
		Tampil Info Perendaman Selesai	*Black Box*

		Pilih Tombol *OK* Pada Info Perendaman Selesai	*Black Box*
		Tampil Penambahan Jumlah Benih Bibit	*Black Box*
		Pilih Kotak Cangkul	*Black Box*
		Tampil Tanah Semai Terolah	*Black Box*
		Pilih Kotak Benih Bibit	*Black Box*
		Tampil Bibit di Tanah Semai	*Black Box*
		Tampil Info Bibit Butuh Pupuk	*Black Box*
		Bibit Setelah Pemupukan	*Black Box*
		Tampil Info Pembibitan Selesai	*Black Box*
		Pilih Tombol *OK* Pada Info Pembibitan Selesai	*Black Box*
		Tampil Penambahan Jumlah Bibit Tanam	*Black Box*
		Pilih Kotak Kerbau	*Black Box*
		Tampil Tanah Tanam Terolah	*Black Box*
		Pilih Kotak Bibit Tanam	*Black Box*
		Tampil Padi di Tanah Tanam	*Black Box*
		Tampil Info Padi Bubuh Pupuk	*Black Box*
		Padi Setelah Pemupukan	*Black Box*
		Tampil Info Padi Siap Panen	*Black Box*
		Padi Pasca Panen	*Black Box*
		Tampil Jumlah Misi (*Quest*) Bertambah	*Black Box*
		Tampil Uang Bertambah Hasil Panen Padi	*Black Box*
		Tampil Uang Bertambah Hasil Pembasmian Hama	*Black Box*
		Tampil Uang Bertambah Hasil	*Black Box*

		Penjualan Pestisida	
		Pilih Kotak Pupuk	*Black Box*
		Pilih Kotak Celurit	*Black Box*
		Pilih Tombol Jerigen	*Black Box*
		Pilih Kotak Pestisida	*Black Box*
		Pilih Info *Tips*	*Black Box*
		Pilih Info *New* Hama	*Black Box*
		Pilih Tombol *Options*	*Black Box*
		Pilih Tombol *Sound On/Off*	*Black Box*
		Pilih Tombol *Music On/Off*	*Black Box*
		Pilih Tombol *Resume*	*Black Box*
		Pilih Tombol *Quit*	*Black Box*
		Pilih Tombol *Restart*	*Black Box*
		Tampil Kondisi Menang	*Black Box*
		Pilih Tombol *Continue* Pada Kondisi Menang	*Black Box*
		Pilih Tombol *Quit* Pada Kondisi Menang	*Black Box*
		Tampil Kondisi Kalah	*Black Box*
		Pilih Tombol *Restart* Pada Kondisi Kalah	*Black Box*
		Pilih Tombol *Quit* Pada Kondisi Kalah	*Black Box*
3	Algoritma A*	Algoritma A* Pada Hama dan Predator	*White Box*
4	Finite State Machines (FSM)	FSM Hama	*White Box*
		FSM Predator	*White Box*

4.2.1.2 Kasus Hasil Pengujian Pengujian (*Black Box Testing*)

Black box bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan. Berikut ini merupakan kasus untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box testing* berdasarkan skenario pengujian aplikasi yang terdapat pada Tabel 4.3 yang sudah dijabarkan. Hasil pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada Tabel 4.4 :

Tabel 4. 4 Pengujian *Black Box*

No	Kasus yang akan di uji	Skenario dan Hasil Uji		
		Skenario Uji	**Hasil yang Diharapkan**	**Kesimpulan**
1	Menu Utama	Pilih Menu *Play Now*	Sistem menampilkan *movieclip* pembuatan profil/*load* profil	[√] Berhasil [ ] Tidak Berhasil
		Pilih *New Game*	Sistem menampilkan *form* isi nama pemain	[√] Berhasil [ ] Tidak Berhasil
		Isi profil baru	Sistem menyimpan data nama pemain	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Close*	Sistem kembali ke tampilan menu utama	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Sound* dan *Music*	Sistem mengantifkan *sound*/*music* jika *sound/ music* dalam keadaan mati, atau sistem menonaktifkan *sound/ music* jika *sound music* dalam keadaan hidup	[√] Berhasil [ ] Tidak Berhasil
2	Lahan Permainan	Tampil *Intructions*	Sistem menampilkan petunjuk permainan	[√] Berhasil [ ] Tidak Berhasil

			saat pemain pertama kali memulai permainan	
		Pilih Tombol *Next Intructions*	Sistem menampilkan petunjuk permainan berikutnya	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Skip Intructions*	Sistem menghilangkan tampilan petunjuk dan memulai permainan	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Predator	Sistem memberikan hak *drag* pada predator	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Biji Benih	Sistem memberikan hak *drag* pada biji benih	[√] Berhasil [ ] Tidak Berhasil
		Tampil Info Perendaman Selesai	Sistem menampilkan pesan perendaman selesai	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *OK* Pada Info Perendaman Selesai	Sistem mengakhiri proses perendaman	[√] Berhasil [ ] Tidak Berhasil
		Tampil Penambahan Jumlah Benih Bibit	Sistem menampilkan penambahan jumlah benih bibit pada kotak benih bibit	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Cangkul	Sistem memberikan hak *drag* pada cangkul	[√] Berhasil [ ] Tidak Berhasil
		Tampil Tanah Semai Terolah	Sistem menampilkan kondisi tanah yang telah terolah	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Benih Bibit	Sistem memberikan hak *drag* pada benih bibit	[√] Berhasil [ ] Tidak Berhasil
		Tampil Bibit di	Sistem menampilakan	[√] Berhasil

		Tanah Semai	bibit di tanah semai	[ ] Tidak Berhasil
		Tampil Info Bibit Butuh Pupuk	Sistem memberikan pesan bibit butuh pupuk	[√] Berhasil [ ] Tidak Berhasil
		Bibit Setelah Pemupukan	Sistem menampilkan fase selanjutnya dari bibit	[√] Berhasil [ ] Tidak Berhasil
		Tampil Info Pembibitan Selesai	Sistem memberikan pesan proses pembibitan selesai	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *OK* Pada Info Pembibitan Selesai	Sistem mengakhiri proses pembibitan di tanah semai	[√] Berhasil [ ] Tidak Berhasil
		Tampil Penambahan Jumlah Bibit Tanam	Sistem menampilkan penambahan jumlah bibiti tanam di kotak bibit tanam	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Kerbau	Sistem memberikan hak *drag* pada kerbau	[√] Berhasil [ ] Tidak Berhasil
		Tampil Tanah Tanam Terolah	Sistem menampilkan kondisi tanah yang telah terolah	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Bibit Tanam	Sistem memberikan hak *drag* pada bibit tanam	[√] Berhasil [ ] Tidak Berhasil
		Tampil Padi di Tanah Tanam	Sistem menampilkan padi di tanah tanam	[√] Berhasil [ ] Tidak Berhasil
		Tampil Info Padi Bubuh Pupuk	Sistem menampilkan pesan padi butuh pupuk	[√] Berhasil [ ] Tidak Berhasil
		Padi Setelah Pemupukan	Sistem merubah fase tumbuh padi ke fase selanjutnya	[√] Berhasil [ ] Tidak Berhasil

		Tampil Info Padi Siap Panen	Sistem menampilkan pesan padi siap untuk di panen	[√] Berhasil [ ] Tidak Berhasil
		Padi Pasca Panen	Sistem menampilkan padi pasca panen dan mengakhiri proses penanaman padi	[√] Berhasil [ ] Tidak Berhasil
		Tampil Jumlah Misi (*Quest*) Bertambah	Sistem menambahkan jumlah misi permainan	[√] Berhasil [ ] Tidak Berhasil
		Tampil Uang Bertambah Hasil Panen Padi	Sistem menampilkan penambahan uang permainan	[√] Berhasil [ ] Tidak Berhasil
		Tampil Uang Bertambah Hasil Pembasmian Hama	Sistem menampilkan penambahan uang permainan	[√] Berhasil [ ] Tidak Berhasil
		Tampil Uang Bertambah Hasil Penjualan Pestisida	Sistem menampilkan penambahan uang permainan	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Pupuk	Sistem memberikan hak *drag* pada pupuk	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Celurit	Sistem memberikan hak *drag* pada celurit	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol Jerigen	Sistem menampilkan kotak-kotak pestisida	[√] Berhasil [ ] Tidak Berhasil
		Pilih Kotak Pestisida	Sistem memberikan hak *drag* pada pestisida	[√] Berhasil [ ] Tidak Berhasil
		Pilih Info *Tips*	Sistem menampilkan pesan *tips* atau petunjuk bantuan	[√] Berhasil [ ] Tidak Berhasil
		Pilih Info *New* Hama	Sistem menampilkan	[√] Berhasil

			pesan jenis hama yang muncul di lahan permainan	[ ] Tidak Berhasil
		Pilih Tombol *Options*	Menampilkan tampilan *options* atau pengaturan	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Sound On/Off*	Sistem mengaktifkan dan menonaktifkan *sound game*	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Music On/Off*	Sistem mengaktifkan dan menonaktifkan *music game*	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Resume*	Sistem keluar dari tampilan *options* dan kembali melanjutkan permainan	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Quit*	Sistem mengakhiri permainan dan menampilkan menu utama	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Restart*	Sistem mengulang permainan di *level* tersebut	[√] Berhasil [ ] Tidak Berhasil
		Tampil Kondisi Menang	Sistem menampilkan tampilan kondisi menang	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Continue* Pada Kondisi Menang	Sistem melanjutkan permainan ke *level* selanjutnya	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Quit* Pada Kondisi	Sistem mengakhiri permainan dan	[√] Berhasil [ ] Tidak Berhasil

		Menang	menampilkan menu utama	
		Tampil Kondisi Kalah	Sistem menampilkan tampilan kondisi kalah	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Restart* Pada Kondisi Kalah	Sistem mengulang permainan di *level* tersebut	[√] Berhasil [ ] Tidak Berhasil
		Pilih Tombol *Quit* Pada Kondisi Kalah	Sistem mengakhiri permainan dan menampilkan menu utama	[√] Berhasil [ ] Tidak Berhasil

4.2.1.3 Kasus dan Hasil Pengujian (*White Box Testing*)

White Box adalah teknik meramalkan cara kerja perangkat lunak secara rinci, karenanya *logical path* (jalur logika) perangkat lunak akan di *test* dengan menyediakan *test case* yang akan mengerjakan kumpulan kondisi dan atau pengulangan secara spesifik. Secara sekilas dapat diambil kesimpulan *white box testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

1. Pengujian *White Box* Algoritma A*

Pengujian algoritma A* hama merupakan proses dimana akan di uji implementasi algoritma A* yang digunakan hama terhadap padi dan predator terhadap hama sebagai pencarian jalur terpendek.

DEKLARASI :

1. OpenNodes : array [ ] of integer *{* inisialisasi *open node* kosong *}*
   ClosedNodes : array [ ] of integer *{* inisialisasi *close node* kosong *}*
   currentNode = firstNode *{* set node awal ke node open *}*
   l , I : integer
   const travelCost = 1.0 *{* inisialisasi biaya perjalanan *}*
   g, h, f : real
   procedure euclidianHeuristic(node, destinationNode)

{ menentukan nilai *heuristic* dengan rumus *pythagoras }*

currentNode.g = 0

currentNode.h = euclidianHeuristic(currentNode, destinationNode)

currentNode.f = currentNode.g + currentNode.h

ALGORITMA :

2. while currentNode ≠ destinationNode

{ Hentikan loop jika currentNode telah sampai ke rute tujuan *}*

3. for i ← 0 to connectedNodes

testNode ← connectedNodes + i

4. if (testNode = currentNode) or (testNode.traversable = false) then

5. g ← currentNode.g + travelCost

h ← euclidianHeuristic(testNode, destinationNode);

{ fungsi *euclidianHeuristic* adalah fungsi utk menghitung jarak dari node A ke B.. dlm A *star* ini, digunakan rumus *phytagoras }*

f ← g + h

6. endif

7. if (isOpen(testNode, openNodes) or isClosed (testNode,closedNodes) then

*{*Jika tidak dapat dilalui atau sudah ada dalam *close list*, abaikan*}*

8. if testNode.f > f then

9. *{* Jika sudah ada di *open list*, cek bila *node* tetangga ini lebih baik, menggunakan nilai *g* sebagai ukuran. Jika lebih baik ganti *parent* dari *node* ini di *open list* menjadi *current node*, lalu kalkulasi ulang nilai *g* dan *f* dari *node* ini *}*

testNode.f ← f

testNode.g ← g

testNode.h ← h

testNode.parentNode ← currentNode

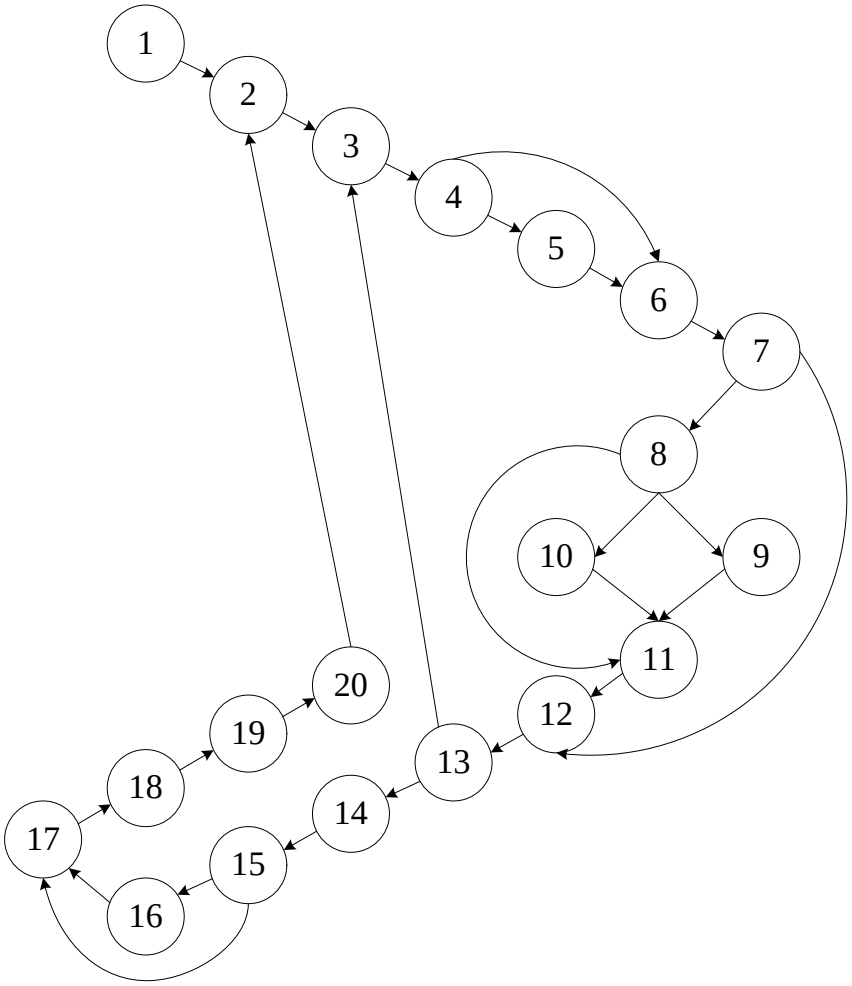
10. else

testNode.f ← f

```
                testNode.g ← g
                testNode.h ← h
                testNode.parentNode ← currentNode
                { Jika belum ada di open list . Buat current node parent dari
                node tetangga ini. Simpan nilai f,g dan h dari node ini }
                openNodes.push(testNode)
11.           endif
12.         endif
13.         endfor
14.         closedNodes.push (currentNode)
            { Keluarkan current node dari open list dan masukan ke close list }
15.         if  openNodes.length = 0 then
16.           return null
17.         endif
18.       openNodes.sortOn ("f", Array.NUMERIC)
          { Cari node (n) dengan nilai f(n) yang paling rendah dalam open list.
          Node ini sekarang menjadi current node }
19.       currentNode ← openNodes
          { Keluarkan current node dari open list }
20.     endwhile
```

a. *Flow Graph*

Berikut ini merupakan *flow graph* algoritma A* berdasarkan *psedocode* di atas yang dapat dilihat pada Gambar 4.11.



Gambar 4. 11 *Flow Graph* Algoritma A* (*A Star*)

Keterangan:

= Menggambarkan Kondisi

= Menggambarkan Aksi

b. *Cyclomatic Complexity* V(G)

Dari diagram alir di atas, dapat dihitung *cyclomatic complexity*, yakni *Region* = 8 yang di dapat dari perhitungan dibawah ini:

V(G) = E – N + 2

V(G) = 26 – 20 + 2

V(G) = 6 + 2 = 8

Dimana:

E = jumlah *edge* pada diagram alir

N = jumlah *node* pada diagram alir

c. *Independent Path*

Dari hasil perhitungan *cyclomatic complexity* terdapat 8 *independent path* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20

Path 2 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20

Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 17 – 18 – 19 – 20

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20

Path 5 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 12 – 13 – 14 – 15 – 17 – 18 – 19 – 20

Path 6 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20

Path 7 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 3

Path 8 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 2

d. *Graph Matric*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	n(E) – 1
1		1																			1 – 1 = 0
2			1																		1 – 1 = 0
3				1																	1 – 1 = 0
4					1	1															2 – 1 = 1
5						1															1 – 1 = 0
6							1														1 – 1 = 0
7								1				1									2 – 1 = 1
8									1	1	1										3 – 1 = 2
9											1										1 – 1 = 0
10											1										1 – 1 = 0
11												1									1 – 1 = 0
12													1								2 – 1 = 1
13			1											1							1 – 1 = 0
14															1						1 – 1 = 0
15																1	1				2 – 1 = 1
16																	1				1 – 1 = 0
17																		1			1 – 1 = 0
18																			1		1 – 1 = 0
19																				1	1 – 1 = 0
20		1																		1	2 – 1 = 1
																	sum (E) +1				7 + 1 = 8

Gambar 4. 12 *Graph Matric* Algoritma A* Penerapan dihama dan Predator

Keterangan:

1. Baris dan kolom merepresentasikan simpul (*node*)
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul

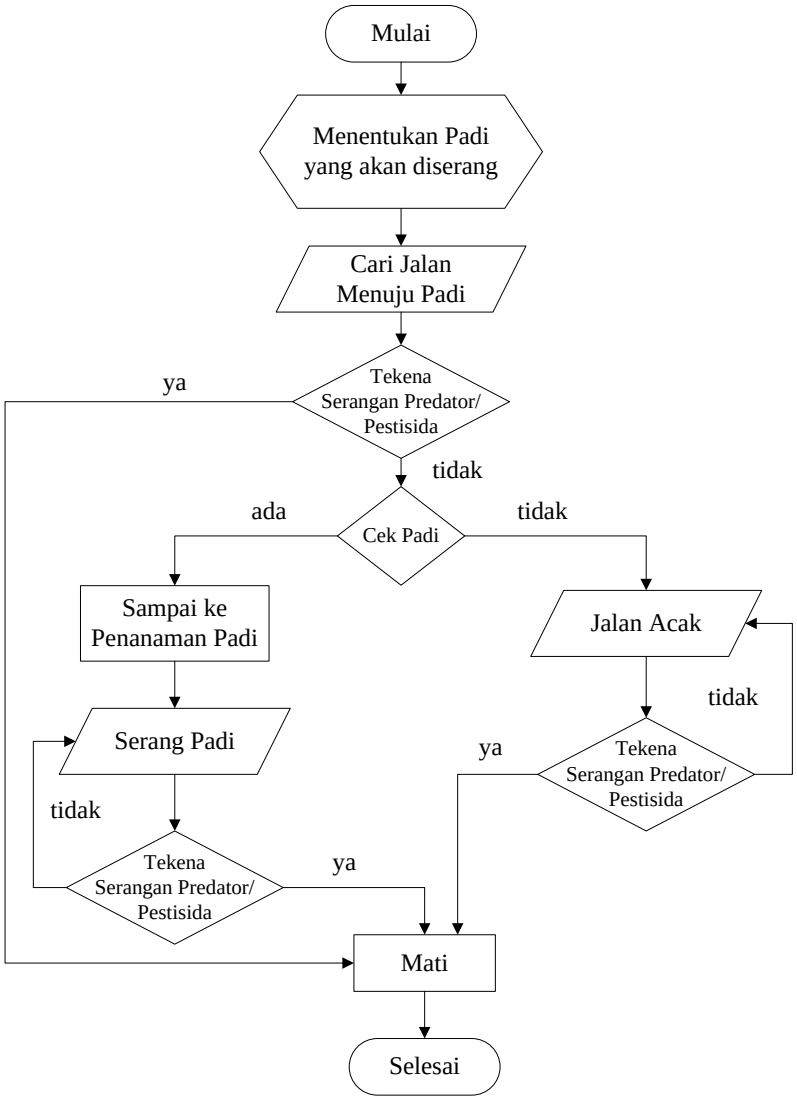
e. Kesimpulan

Dari hasil pengujian algoritma A* yang dilakukan, maka didapat kesimpulan bahwa seluruh *node* yang ada pada setiap *path* telah dikerjakan, serta tidak terjadi pengulangan.

2. Pengujian *White Box Finite State Machine* Hama

a. *Flow Chart*

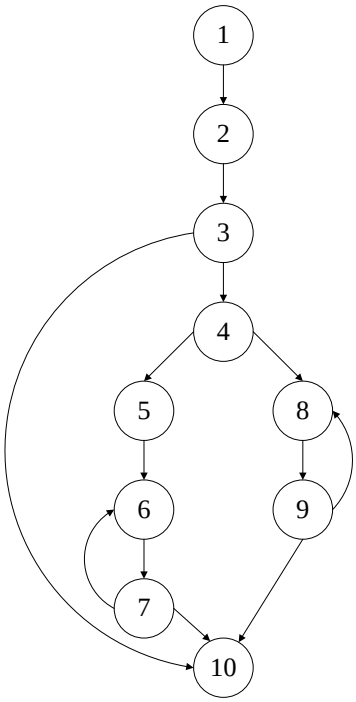
Berikut adalah *flow chart* pengujian *white box finite state machine* penerapan pada kelas hama yang dapat dilihat pada Gambar 4.13.



Gambar 4. 13 *Flow Chart Finite State Machine* Hama

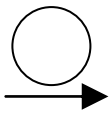
b. *Flow Graph*

Berikut ini merupakan *flow graph finite state machine* hama berdasarkan *flow chart* di atas yang dapat dilihat pada Gambar 4.14.



Gambar 4. 14 *Flow Graph Finite State Machine* Hama

Keterangan



= Menggambarkan Kondisi

= Menggambarkan Aksi

c. *Cyclomatic Complexity* V(G)

Dari diagram alir di atas, dapat dihitung *cyclomatic complexity*, yakni *Region* = 5 yang di dapat dari perhitungan dibawah ini:

V(G) = E – N + 2

V(G) = 13 – 10 + 2

V(G) = 3 + 2 = 5

Dimana:

E = jumlah *edge* pada diagram alir

N = jumlah *node* pada diagram alir

d. *Independent Path*

Dari hasil perhitungan *cyclomatic complexity* terdapat 5 *independent path* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 10

Path 2 = 1 – 2 – 3 – 4 –8 – 9 – 10

Path 3 = 1 – 2 – 3 – 4 – 8 – 9 – 8 – 9 – 10

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 6 – 7 – 10

Path 5 = 1 – 2 – 3 – 10

e. *Graph Matriks*

	1	2	3	4	5	6	7	8	9	10	n(E) – 1
1		1									1 – 1 = 0
2			1								1 – 1 = 0
3				1						1	2 – 1 = 1
4					1			1			2 – 1 = 1
5						1					1 – 1 = 0
6							1				1 – 1 = 0
7						1				1	2 – 1 = 1
8									1		1 – 1 = 0
9								1		1	2 – 1 = 1
10										1	1 – 1 = 0
								sum (E) +1			4 + 1 = 5

Gambar 4. 15 *Graph Matric Finite State Machine* Hama

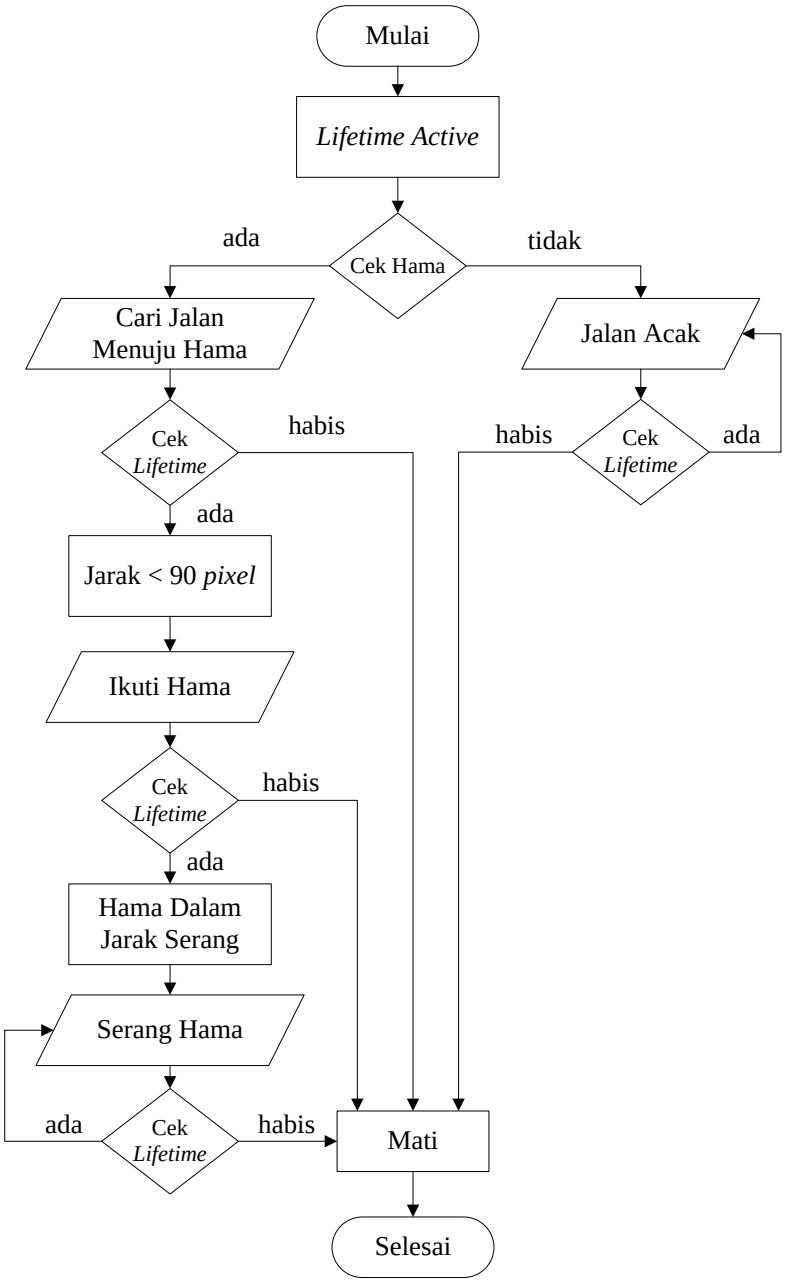
f. Kesimpulan

Dari hasil pengujian *finite state machine* hama yang dilakukan, maka didapat kesimpulan bahwa seluruh *node* yang ada pada setiap *path* telah dikerjakan, serta tidak terjadi pengulangan.

3. Pengujian *White Box Finite State Machine* Predator

a. *Flow Chart*

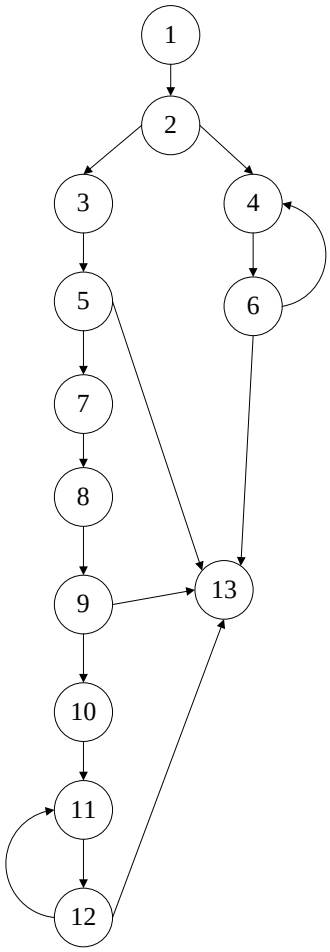
Berikut adalah *flow chart* pengujian *white box finite state machine* penerapan pada kelas predator yang dapat dilihat pada Gambar 4.16.



Gambar 4. 16 *Flow Chart Finite State Machine* Predator

b. *Flow Graph*

Berikut ini merupakan *flow graph finite state machine* predator berdasarkan *flow chart* di atas yang dapat dilihat pada Gambar 4.17.



Gambar 4. 17 *Flow Graph Finite State Machine* Predator

c. *Cyclomatic Complexity* V(G)

Dari diagram alir di atas, dapat dihitung *cyclomatic complexity*, yakni *Region* = 6 yang di dapat dari perhitungan dibawah ini:

V(G) = E – N + 2

V(G) = 17 – 13 + 2

V(G) = 4 + 2 = 6

Dimana:

E = jumlah *edge* pada diagram alir

N = jumlah *node* pada diagram alir

d. *Independent Path*

Dari hasil perhitungan *cyclomatic complexity* terdapat 6 *independent path* yaitu:

Path 1 = 1 – 2 – 3 – 5 – 13

Path 2 = 1 – 2 – 3 – 5 – 7 – 8 – 9 – 13

Path 3 = 1 – 2 – 3 – 5 – 7 – 8 – 9 – 10 – 11 – 12 – 13

Path 4 = 1 – 2 – 3 – 5 – 7 – 8 – 9 – 10 – 11 – 12 – 11- 12 – 13

Path 5 = 1 – 2 – 4 – 6 – 13

Path 6 = 1 – 2 – 4 – 6 – 4 – 6 – 13

e. *Graph Matrics*

	1	2	3	4	5	6	7	8	9	10	11	12	13	n(E) – 1
1		1												1 – 1 = 0
2			1	1										2 – 1 = 1
3					1									1 – 1 = 0
4						1								1 – 1 = 0
5							1						1	2 – 1 = 1
6				1									1	2 – 1 = 1
7								1						1 – 1 = 0
8									1					1 – 1 = 0
9										1			1	2 – 1 = 1
10											1			1 – 1 = 0
11												1		1 – 1 = 0
12											1		1	2 – 1 = 1
13													1	1 – 1 = 0
												sum (E) +1		5 + 1 = 6

Gambar 4. 18 *Graph Matrics Finite State Machine* Predator

f. Kesimpulan

Dari hasil pengujian *finite state machine* predator yang dilakukan, maka didapat kesimpulan bahwa seluruh *node* yang ada pada setiap *path* telah dikerjakan, serta tidak terjadi pengulangan.

4.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan dengan menggunakan kuesioner mengenai tanggapan pengguna terhadap *game* yang telah dibangun. Kuesioner disebarkan terdiri dari 14 pertanyaan dan ditanggapi oleh 41 orang dengan rentang

usia antara 14 tahun sampai 18 tahun. Berdasarkan data hasil kuesioner tersebut, dapat dicari persentase masing-masing jawaban dengan menggunakan rumus.

P =f/N x 100%.

P : Persentase

f : Frekuensi data

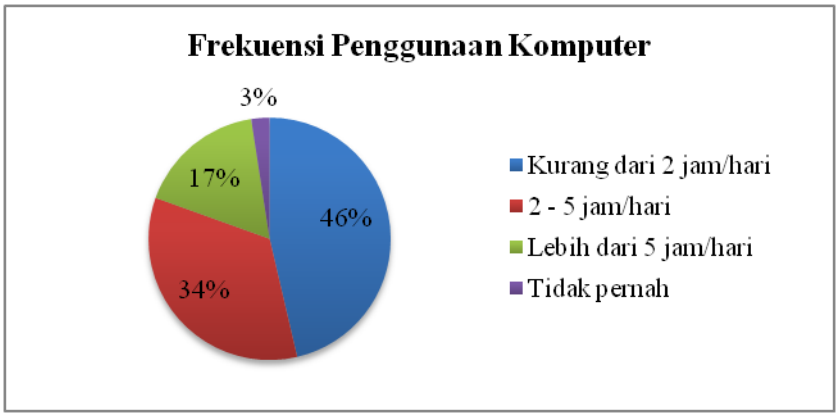
N : Jumlah sampel yang di olah

Berikut adalah data-data yang didapat dari hasil kuesioner yang berkaitan dengan *game save your ricefield* dan tujuan penelitian.

1. Pertanyaan 1: Seberapa lama Anda menggunakan komputer dalam sehari?

Tabel 4. 5 Hasil Kuesioner Frekuensi Waktu Penggunaan Komputer perhari

No	Jawaban	Responder	Persentase
1	Kurang dari 2 jam/hari	19	46%
2	2 -5 jam/hari	14	34%
3	Lebih dari 5 jam/hari	7	17%
4	Tidak pernah	1	3%
Total Responder		41	100%



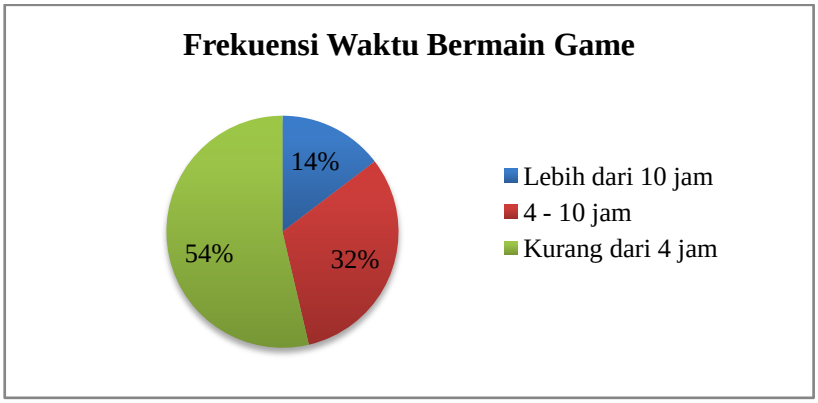
Gambar 4. 19 Diagram Frekuensi Waktu Penggunaan Komputer perhari

Berdasarkan hasil jawaban kuesioner yang digambarkan pada diagram di atas, diketahui bahwa sekitar 46% responder menggunkan komputer kurang dari 2 jam dalam sehari.

2. Pertanyaan 2: Seberapa lama Anda bermain *game* dalam seminggu?

Tabel 4. 6 Hasil Kuesioner Frekuensi Waktu Bermain *Game* Perminggu

No	Jawaban	Responder	Persentase
1	Lebih dari 10 jam	6	14%
2	4 – 10 jam	13	32%
3	Kurang dari 4 jam	22	54%
Total Responder		41	100%



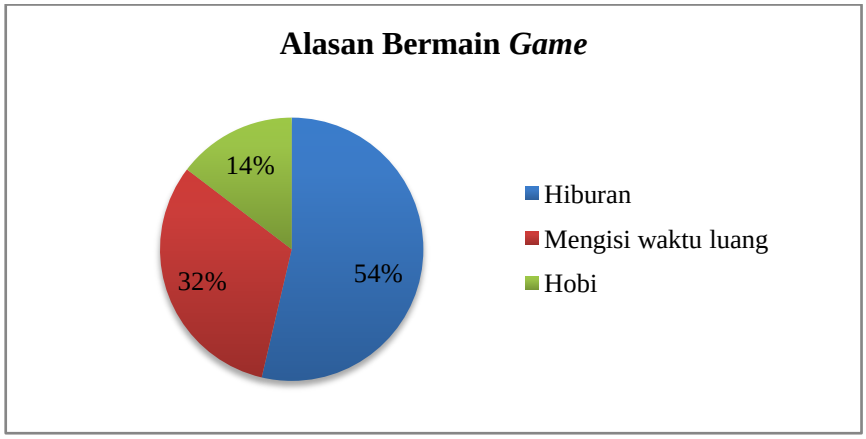
Gambar 4. 20 Diagram Frekuensi Waktu Bermain *Game* Perminggu

Berdasarkan hasil jawaban kuesioner yang digambarkan pada diagram di atas, diketahui bahwa sekitar 54% responder bermain *game* selama kurang dari 4 jam dalam satu minggu.

3. Pertanyaan 3: Apa alasan Anda bermain *game*?

Tabel 4. 7 Hasil Kuesioner Alasan Bermain *Game*

No	Jawaban	Responder	Persentase
1	Hiburan	22	54%
2	Mengisi waktu luang	13	32%
3	Hobi	6	14%
Total Responder		41	100%



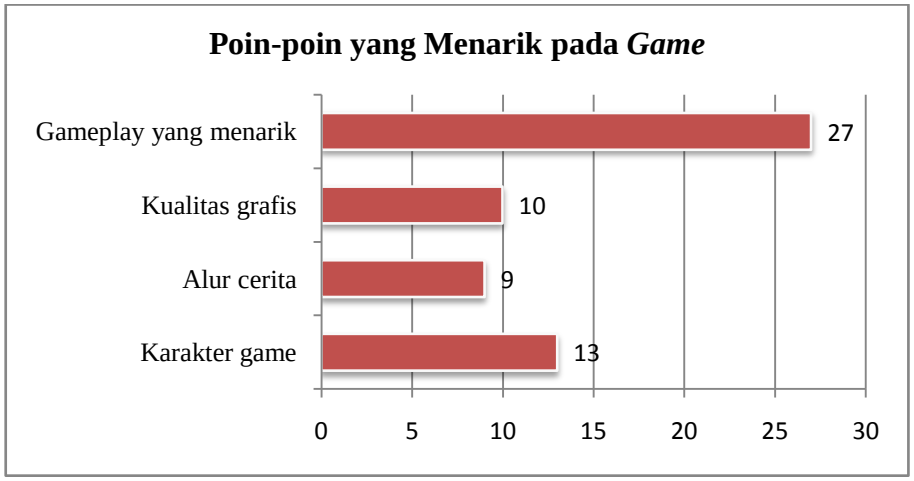
Gambar 4. 21 Diagram Alasan Bermain *Game*

Berdasarkan hasil jawaban kuesioner yang digambarkan pada diagram di atas, diketahui bahwa 54% responder bermain *game* untuk mencari hiburan dan 32% responder bermain *game* untuk mengisi waktu luang. Sedangkan yang menjadikan *game* sebagai hobi adalah sekitar 14%.

4. Pertanyaan 4: Apa yang membuat Anda tertarik dari sebuah *game*? (Jawaban boleh lebih dari satu)

Tabel 4. 8 Hasil Kuesioner Poin-poin yang Menarik pada *Game*

No	Jawaban	Responder
1	Karakter *game*	13
2	Alur cerita	9
3	Kualitas grafis	10
4	Gameplay yang menarik	27



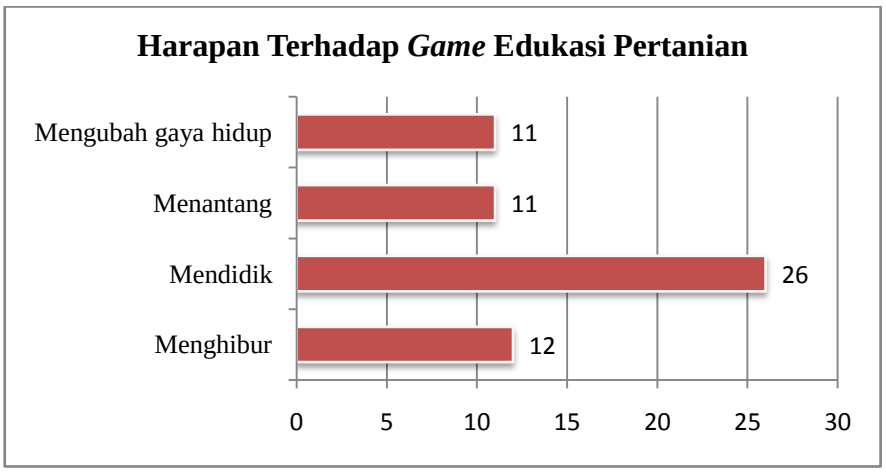
Gambar 4. 22 Diagram Poin-poin yang Menarik pada *Game*

Berdasarkan hasil jawaban kuesioner yang digambarkan pada diagram di atas, diketahui bahwa sebagian besar responder memilih bermain *game* karena *gameplay* yang menarik dengan 27 responder dan diikuti dengan karakter *game*.

5. Pertanyaan 5: Jika ada *game* edukasi bertemakan tentang pertanian yang ada di Indonesia, apa yang Anda harapkan setelah bermain *game* tersebut? (jawaban boleh lebih dari satu)

Tabel 4. 9 Hasil Kuesioner Harapan Terhadap *Game* Edukasi Pertanian

No	Jawaban	Responder
1	Menghibur	12
2	Mendidik	26
3	Menantang	11
4	Mengubah gaya hidup	11



Gambar 4. 23 Diagram Harapan Terhadap *Game* Edukasi Pertanian

Berdasarkan hasil jawaban kuesioner di atas, sebanyak 26 reponder mengharapkan *game* yang mendidik jika dibuat sebuah *game* edukasi bertemakan pertanian di Indonesia. Sementara 12 responder mengharapkan *game* edukasi bertemakan pertanian dapat mengubah gaya hidup mereka.

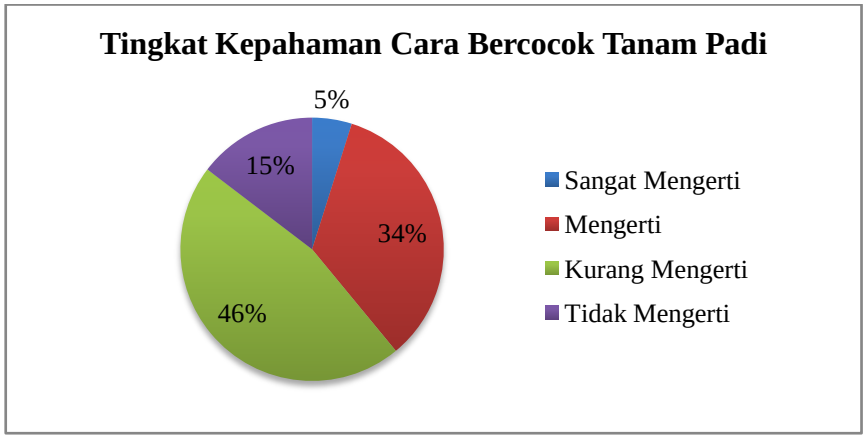
6. Pertanyaan 6: Apakah Anda mengerti cara bercocok tanam padi?

Tabel 4. 10 Hasil Kuesioner Tingkat Kepahaman Cara Bercocok Tanam Padi

No	Kode Jawaban	Bobot	Frekuensi Jawaban	Skor
1	SM	4	2	(4x2) = 8
2	M	3	14	(3x14) = 42
3	KM	2	19	(2x19) = 38
4	TM	1	6	(1x6) = 6
Jumlah Skor				94
Persentase				(94/164) x 100% = 57,3%

Keterangan:

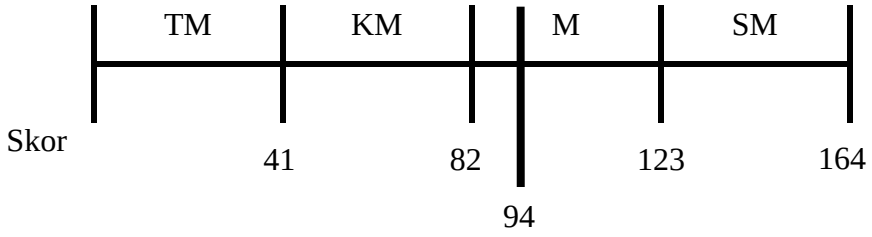
SM = Sangat Mengerti
M = Mengerti
KM = Kurang Mengerti
TM = Tidak Mengerti



Gambar 4. 24 Diagram Tingkat Kepahaman Cara Bercocok Tanam Padi

Jumlah skor ideal (kriterium) untuk seluruh item adalah 4 x 41 responder = 164, sedangkan jumlah skor yang diperoleh dari penelitian = 94. Jadi berdasarkan data pada Tabel 4.10 maka didapat persentasenya yaitu 57,3%.

Berikut adalah garis kontinum tingkat kepahaman generasi muda terhadap cara bercocok tanam padi.



Gambar 4. 25 Garis Kontinum Tingkat Kepahaman Cara Bercocok Tanam Padi

Berdasarkan hasil tersebut maka dapat disimpulkan bahwa tingkat kepahaman generasi muda tentang bercocok tanam padi adalah 94 dari yang diharapkan 164 atau dikategorikan mengerti.

7. Pertanyaan 7: Apakah Anda mengetahui jenis hama apa saja yang sering meyerang pertanian di Indonesia

Tabel 4. 11 Hasil Kuesioner Tingkat Pengenalan Jenis Hama

No	Kode Jawaban	Bobot	Frekuensi Jawaban	Skor
1	ST	4	1	(4x1) = 4

2	T	3	16	(3x16) = 48
3	KT	2	19	(2x19) = 38
4	TT	1	5	(1x5) = 5
Jumlah Skor				95
Persentase				(95/164) x 100% = 58%

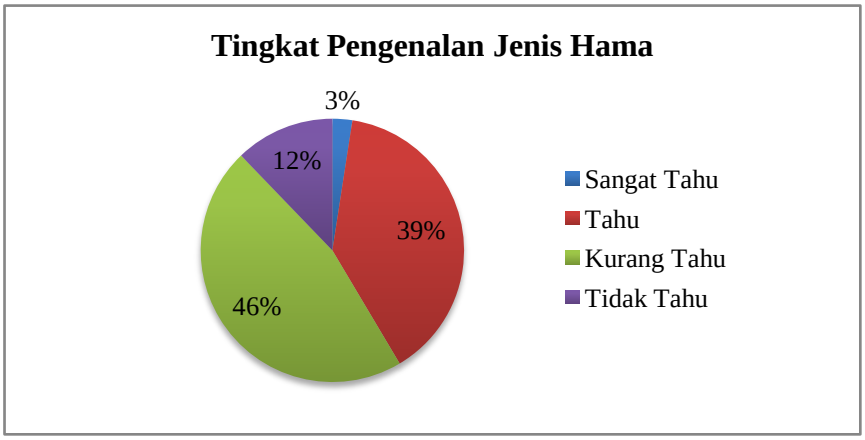
Keterangan:

ST = Sangat Tahu

T = Tahu

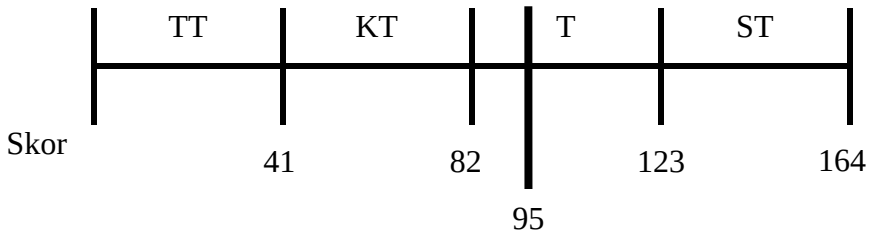
KT = Kurang Tahu

TT = Tidak Tahu



Gambar 4. 26 Diagram Tingkat Pengenalan Jenis Hama

Jumlah skor ideal (kriterium) untuk seluruh item adalah 4 x 41 responder = 164, sedangkan jumlah skor yang diperoleh dari penelitian = 95. Jadi berdasarkan data pada Tabel 4.11 maka didapat persentasenya yaitu 58%. Berikut adalah garis kontinum tingkat pengenalan generasi muda terhadap jenis-jenis hama yang sering menyerang pertanian di Indonesia.



Gambar 4. 27 Garis Kontinum Tingkat Pengenalan Jenis Hama

Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa tingkat pengenalan generasi muda terhadap jenis-jenis hama yang sering menyerang pertanian di Indonesia adalah 95 dari yang diharapkan 90 atau dapat dikategorikan tahu.

8. Pertanyaan 8: Apakah Anda mengetahui jenis predator/hewan yang dapat membantu petani dalam membasmi hama?

Tabel 4. 12 Hasil Kuesioner Tingkat Pengenalan Jenis Predator

No	Kode Jawaban	Bobot	Frekuensi Jawaban	Skor
1	ST	4	2	(4x2) = 8
2	T	3	21	(3x21) = 63
3	KT	2	12	(2x12) = 24
4	TT	1	7	(1x7) = 7
Jumlah Skor				102
Persentase				(102/164) x 100% = 62%

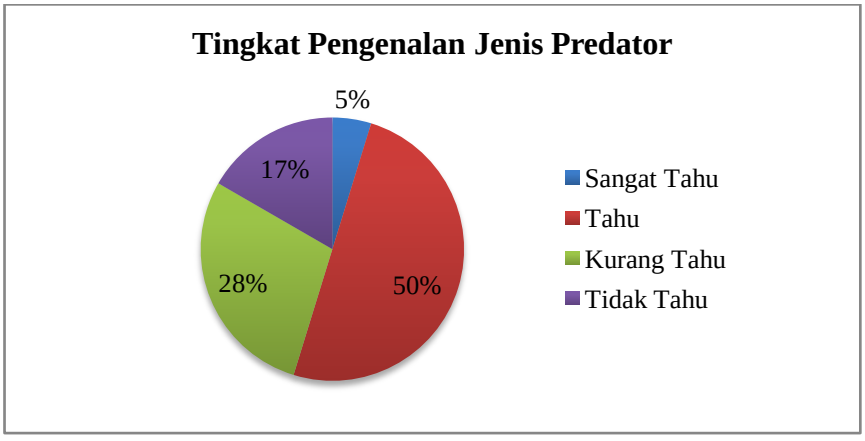
Keterangan:

ST = Sangat Tahu

T = Tahu

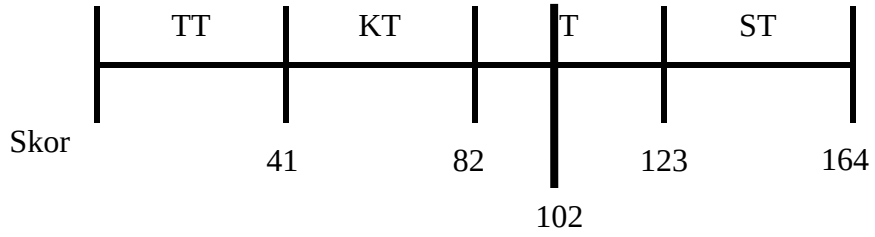
KT = Kurang Tahu

TT = Tidak Tahu



Gambar 4. 28 Diagram Tingkat Pengenalan Jenis Predator

Jumlah skor ideal (kriterium) untuk seluruh item adalah 4 x 41 responder = 164, sedangkan jumlah skor yang diperoleh dari penelitian = 102. Jadi berdasarkan data pada Tabel 4.12 maka didapat persentasenya yaitu 62%. Berikut adalah garis kontinum tingkat pengenalan generasi muda terhadap jenis-jenis predator yang dapat membantu petani membasmi hama.



Gambar 4. 29 Garis Kontinum Tingkat Pengenalan Jenis Predator

Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa tingkat pengenalan generasi muda terhadap jenis-jenis predator yang membantu petani membasmi hama adalah 102 dari yang diharapkan 164, atau dapat dikategorikan tahu.

9. Pertanyaan 9: Karena Indonesia dikenal sebagai negara agraris, menurut Anda perlukah masyarakatnya mengenal dunia pertanian?

Tabel 4. 13 Hasil Kuesioner Tingkat Pengenalan Dunia Pertanian

No	Kode Jawaban	Bobot	Frekuensi Jawaban	Skor
1	SP	4	18	(4x18) = 72
2	P	3	18	(3x18) = 54
3	CP	2	5	(2x5) = 10
4	TP	1	0	(1x0) = 0
Jumlah Skor				136
Persentase				(136/164) x 100% = 83%

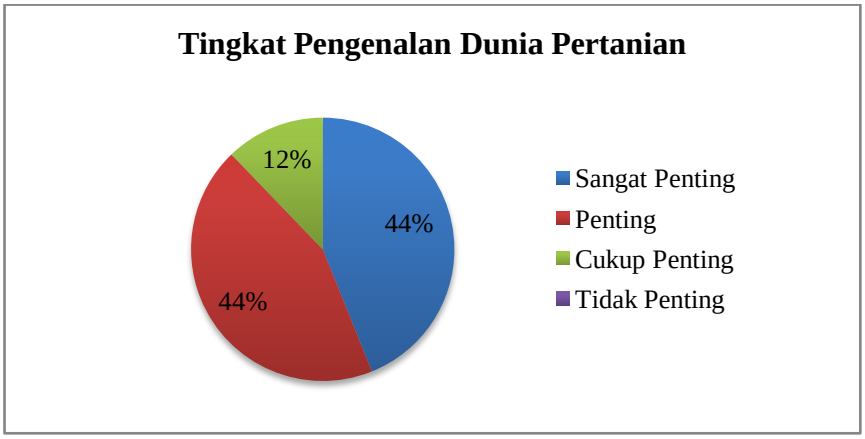
Keterangan:

SP = Sangat Penting

P = Penting

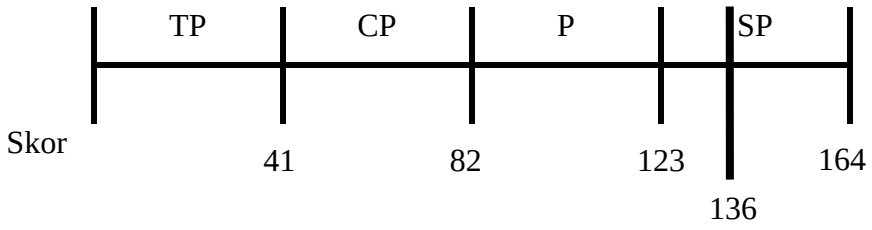
CP = Cukup Penting

TP = Tidak Penting



Gambar 4. 30 Diagram Tingkat Pengenalan Dunia Pertanian

Jumlah skor ideal (kriterium) untuk seluruh item adalah 4 x 41 responder = 164, sedangkan jumlah skor yang diperoleh dari penelitian = 136. Jadi berdasarkan data pada Tabel 4.13 maka didapat persentasenya yaitu 83%. Berikut adalah garis kontinum tingkat kepentingan generasi muda mengenal dunia pertanian, khususnya pertanian di Indonesia.



Gambar 4. 31 Garis Kontinum Tingkat Pengenalan Dunia Pertanian

Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa tingkat minat generasi muda mengenal dunia pertanian sebesar 136 dari yang diharapkan adalah 164, atau dapat dikategorikan sangat penting bagi generasi muda mengenal dunia pertanian.

10. Pertanyaan 10: Bagaimana pendapat Anda terhadap *gameplay* dari *game Save Your Ricefield* ini?

Tabel 4. 14 Hasil Kuesioner Pendapat Tentang *Gameplay*

No	Kode Jawaban	Bobot	Frekuensi Jawaban	Skor
1	SM	4	1	(4x1) = 4
2	M	3	23	(3x23) = 69
3	CM	2	15	(2x15) = 30
4	KM	1	2	(1x2) = 2
Jumlah Skor				105
Persentase				(105/164) x 100% = 64%

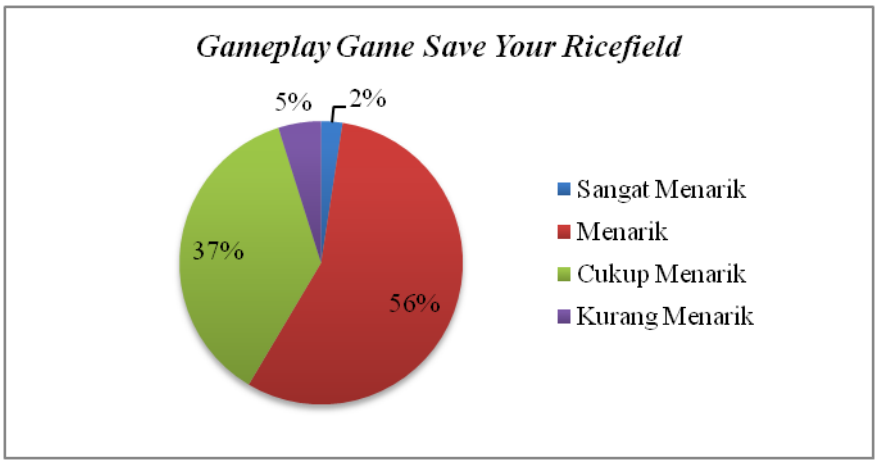
Keterangan:

SM = Sangat Menarik

M = Menarik

CM = Cukup Menarik

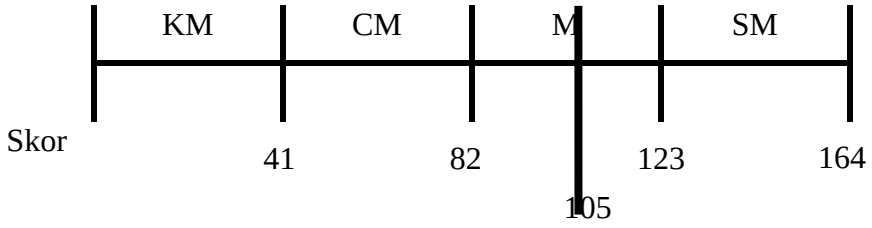
KM = Kurang Menarik



Gambar 4. 32 Diagram Pendapat Tentang *Gameplay*

Dari diagram diatas 56% responder mengatakan *gameplay game save your ricefield* menarik, dan 37% mengatakan cukup menarik.

Jumlah skor ideal (kriterium) untuk seluruh item adalah 4 x 41 responder = 164, sedangkan jumlah skor yang diperoleh dari penelitian = 105. Jadi berdasarkan data pada Tabel 4.14 maka didapat persentasenya yaitu 64%. Berikut adalah garis kontinum pendapat *tentang gameplay game save your ricefield*.



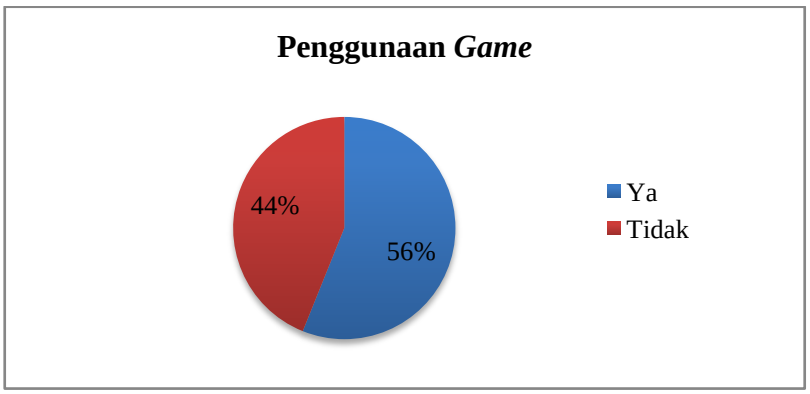
Gambar 4. 33 Garis Kontinum Pendapat Tentang *Gameplay*

Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa *gameplay game save your ricefield* menarik, yaitu dengan skor 105 dari yang diharapkan adalah 164.

11. Pertanyaan 11: Apakah *game Save Your Ricefield* ini mudah untuk dimainkan?

Tabel 4. 15 Hasil Kuesioner Penggunaan *Game*

No	Jawaban	Responder	Persentase
1	Ya	23	56%
2	Tidak	18	44%
Total Responder		41	100%



Gambar 4. 34 Diagram Penggunaan *Game*

Berdasarkan hasil jawaban kuesioner yang digambarkan pada diagram di atas, diketahui bahwa *game save your ricefield* 56% responder mengatakan *game* ini mudah untuk dimainkan.

12. Pertanyaan 12: Setelah memainkan *game Save Your Ricefield* Anda menjadi mengerti cara dasar menanam padi?

Tabel 4. 16 Hasil Kuesioner Tingkat Kesetujuan Cara Bercocok Tanam Padi Setelah Memainkan *Game*

No	Kode Jawaban	Bobot	Frekuensi Jawaban	Skor
1	SS	4	3	(4x3) = 12
2	S	3	29	(3x29) = 87
3	KS	2	8	(2x8) = 16
4	TS	1	1	(1x1) = 1
Jumlah Skor				116
Persentase				(116/164) x 100% = 71%

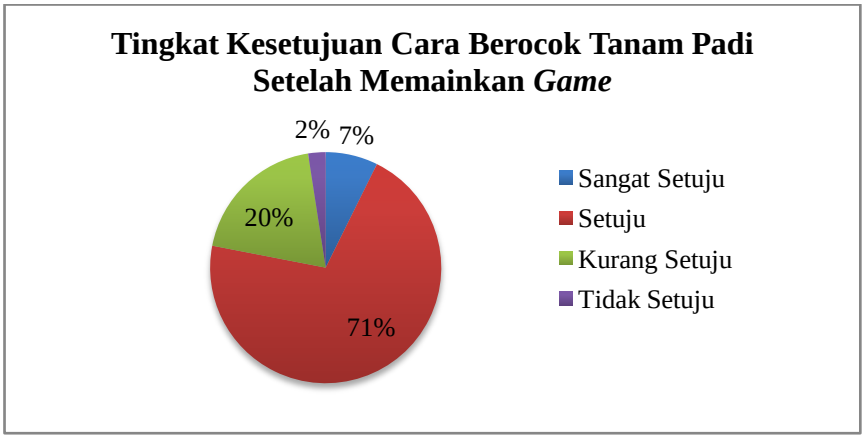
Keterangan:

SS = Sangat Setuju

S = Setuju

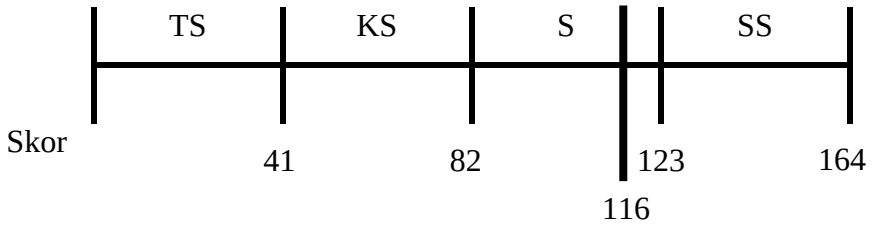
KS = Kurang Setuju

TS = Tidak Setuju



Gambar 4. 35 Diagram Tingkat Kesetujuan Cara Bercocok Tanam Padi Setelah Memainkan *Game*

Jumlah skor ideal (kriterium) untuk seluruh item adalah 4 x 41 responder = 164, sedangkan jumlah skor yang diperoleh dari penelitian = 116. Jadi berdasarkan data pada Tabel 4.16 maka didapat persentasenya yaitu 71%. Berikut adalah garis kontinum tingkat kesetujuan cara bercocok tanam padi setelah memainkan *game save your ricefield*.



Gambar 4. 36 Garis Kontinum Tingkat Kesetujuan Cara Bercocok Tanam Padi Setelah Memainkan *Game*

Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa setelah responder bermain *game save your ricefield* responder setuju bahwa *game save your ricefield* dapat membuat pemainnya mengerti cara dasar bercocok tanam padi. Skor yang didapat adalah 116 dari yang diharapkan sebesar 164.

13. Pertanyaan 13: Apakah *game Save Your Ricefield* ini membantu Anda mengenal jenis-jenis hama yang sering menyerang pertanian di Indonesia?

Tabel 4. 17 Hasil Kuesioner Tingkat Mengenal Hama Setelah Memainkan *Game*

No	Kode Jawaban	Bobot	Frekuensi Jawaban	Skor
1	SM	4	6	(4x6) = 24
2	M	3	29	(3x29) = 87
3	KM	2	6	(2x6) = 12
4	TM	1	0	(1x0) = 0
Jumlah Skor				123
Persentase				(123/164) x 100% = 75%

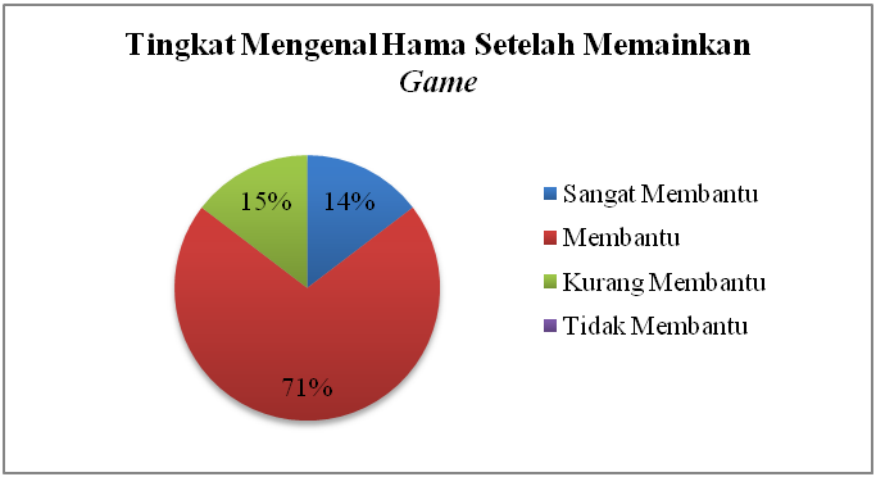
Keterangan:

SM = Sangat Membantu

M = Membantu

KM = Kurang Membantu

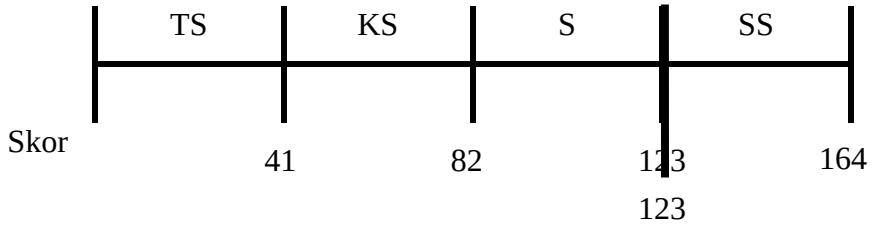
TM = Tidak Membantu



Gambar 4. 37 Diagram Tingkat Mengenal Hama Setelah Memainkan *Game*

Jumlah skor ideal (kriterium) untuk seluruh item adalah 4 x 41 responder = 164, sedangkan jumlah skor yang diperoleh dari penelitian = 123. Jadi

berdasarkan data pada Tabel 4.17 maka didapat persentasenya yaitu 75. Berikut adalah garis kontinum tingkat mengenal hama setelah responder memainkan *game save your ricefield*.



Gambar 4. 38 Garis Kontinum Tingkat Mengenal Hama Setelah Memainkan *Game*

Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa *game save your ricefield* sangat membantu pemainnya untuk dapat mengenal jenis-jenis hama yang sering menyerang pertanian di Indonesia. Skor yang didapat dari hasil kuesioner adalah 123 dari 164 yang diharapkan.

14. Pertanyaan 14: Apakah *game Save Your Ricefield* ini membantu Anda mengenal jenis hewan/predator yang membantu petani dalam membasmi hama?

Tabel 4. 18 Hasil Kuesioner Tingkat Mengenal Predator Setelah Memainkan *Game*

No	Kode Jawaban	Bobot	Frekuensi Jawaban	Skor
1	SM	4	3	(4x3) = 12
2	M	3	30	(3x30) = 90
3	KM	2	6	(2x6) = 12
4	TM	1	2	(1x2) = 2
Jumlah Skor				116
Persentase				(116/164) x 100% = 71%

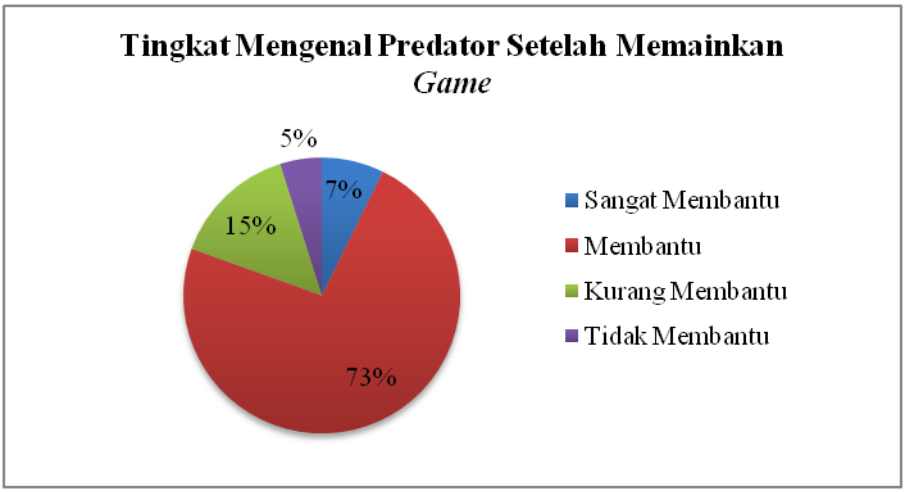
Keterangan:

SM = Sangat Membantu

M = Membantu

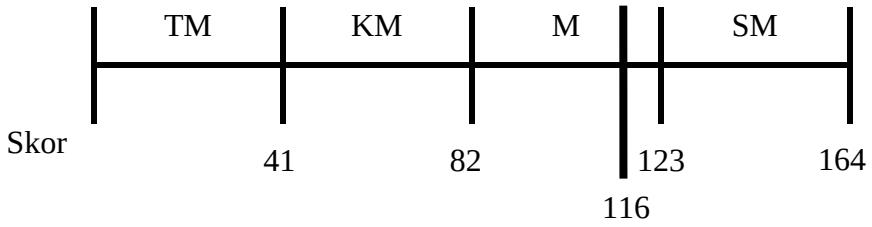
KM = Kurang Membantu

TM = Tidak Membantu



Gambar 4. 39 Diagram Tingkat Mengenal Predator Setelah Memainkan *Game*

Jumlah skor ideal (kriterium) untuk seluruh item = 4 x 41 responder = 164, sedangkan jumlah skor yang diperoleh dari penelitian = 116. Jadi berdasarkan data pada Tabel 4.18 maka didapat persentasenya yaitu 71%. Berikut adalah garis kontinum tingkat mengenal predator setelah responder memainkan *game save your ricefield*.



Gambar 4. 40 Garis Kontinum Tingkat Mengenal Predator Setelah Memainkan *Game*

Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa *game save your ricefield* membantu pemainnya untuk dapat mengenal jenis-jenis predator atau hewan dapat membantu petani dalam membasmi hama.

4.2.3 Kesimpulan Pengujian Beta

Berdasarkan pengujian beta yang telah didapat dari hasil kuesioner yang telah dijawab oleh responder, maka dapat ditarik kesimpulan sebagai berikut:

1. Secara umum responder menghabiskan waktu didepan komputer kurang dari 2 jam perharinya
2. Sebagian besar responder memainkan *game* kurang dari 4 jam perminggu nya yaitu sebesar 54%. Hanya 14% responder yang menghabiskan waktu untuk bermain *game* selama lebih dari 10 jam perminggunya.
3. Alasan responder bermain *game* kebanyakan adalah untuk mencari hiburan persentase yang memilih mencari hiburan mencapai 54%.
4. Sebagian besar responder memilih *game* karena *gameplay* nya dibandingkan alur cerita. Dari data kuesioner perbandingan yang memilih *gameplay* dan alur cerita yaitu 27 berbanding 9.
5. Dengan adanya *game* edukasi bertemakan pertanian, responder mengharapkankan *game* tersebut menjadi sarana edukasi tentang pertanian itu sendiri dibanding dengan *game* yang hanya menghibur.
6. Dari diagram tentang tingkat kepahaman cara bercocok tanam padi, 46% responder kurang mengerti cara bercocok tanam padi.
7. Tingkat pengenalan responder terhadap jenis hama perusak padi cukup kurang, sekitar 46% mengatakan kurang tahu.
8. Responder yang mengenal jenis-jenis hewan yang menjadi predator hama cukup besar yaitu sekitar 50%, tetapi responder yang tidak mengetahui nya juga mengkhawatirkan yaitu sekitar 17%.
9. Dari hasil pertanyaan pentingkah mengenal dunia pertanian sebagian besar responder menjawab penting dan sangat penting masing masing 44%.
10. *Gameplay game save your ricefield* dianggap menarik bagi sebagian besar responder.

11. *Game save your ricefield* cukup mudah dimainkan.
12. Setelah memainkan *game save your ricefield* 71% responder setuju *game* ini responder menjadi mengerti cara dasar dalam menanam padi.
13. Sebagian besar responder yaitu 71%, mengatakan *game save your ricefield* membantu pemainnya mengenal jenis-jenis hama pertanian.
14. Sebagian besar responder juga mengatakan bahwa *game* s*ave your ricefield* membantu pemainnya mengenal hewan apa saja yang menjadi predator bagi hama pertanian.