

BAB V. IMPLEMENTASI DAN PENGUJIAN

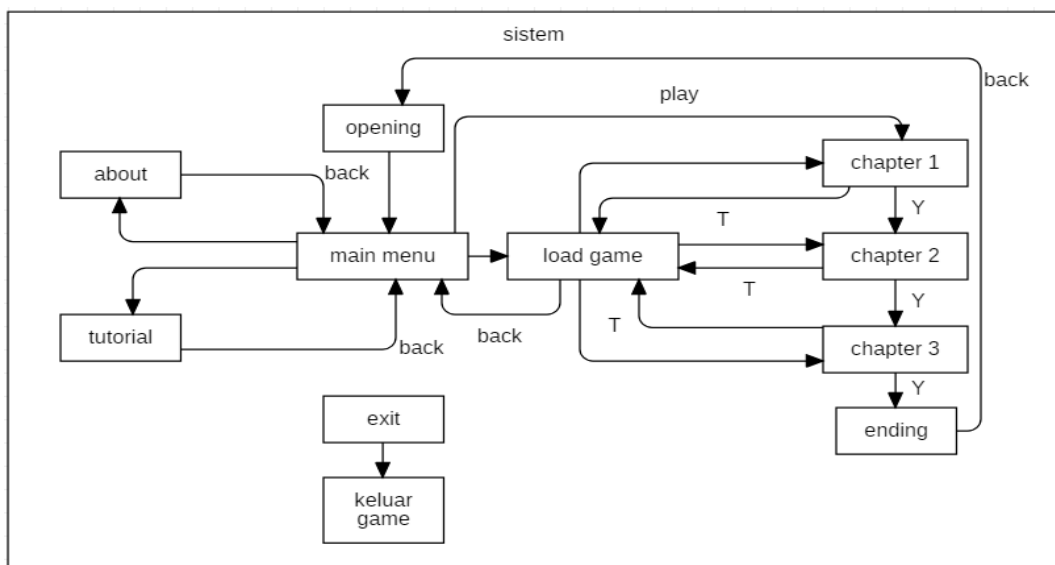
Menjelaskan tentang proses pembuatan sistem berdasarkan teori dan rancangan yang digunakan dalam pembuatan sistem. Penjelasan berisi langkah-langkah hingga hasil sistem seperti apa, serta pengujian terhadap sistem/hipotesis.

Penggunaan *tools* pada penelitian ini yang digunakan adalah *software* unity versi 2020 dan visual studio 2019 menggunakan bahasa pemrograman c sharp, hal ini dapat dilihat dan sudah dijelaskan pada tahapan metode penelitian di bab 3.

Menjelaskan proses yang akan dilakukan pertama implementasi sistem, kedua menggambarkan desain antarmuka pengguna seluruh sistem dan memaparkan kode yang berhubungan dengan implementasi sistem yaitu menulis kode, ketiga masuk ke pengujian yaitu menjelaskan *system testing* dan *UX testing*/Kuesioner ini telah dijelaskan pada bab 3 bagian uji coba dan akan dilakukan pengujian menggunakan *Confusion Matrix* agar terlihat hasilnya metode *fuzzy* Mamdani benar adanya.

1.1 Implementasi Sistem

Memaparkan gambaran sistem dan dapat menjelaskan semua fungsi pada rancangan permainan serta metode yang digunakan dalam penelitian.



Gambar 5. 1 Cara Kerja Seluruh Sistem

Hal tersebut menggambarkan fungsi yang dapat dilakukan oleh pemain/pengguna permainan apa saja yang dapat diakses saat bermain dan saat selesai bermain.

Pada saat pemain masuk permainan akan tampil opening, main menu, bisa mengakses about, tutorial, load game, exit seperti yang dijelaskan pada bab 4 gambar flowchart cara kerja sistem, load game memiliki 3 level dan ini dapat diakses setelah menyelesaikan tiap chapter, jika kalah akan kembali ke load game, dan jika sudah menyelesaikan semua gameplay akan kembali pada opening sebagaimana sistem berjalan.

1.2 Menulis kode (*Writing code*)

Pada tahap ini akan menjelaskan fungsi berupa kode bahasa pemrograman bagaimana sistem gambar di implementasi sistem bisa berjalan.

Tabel 5. 1 Fungsi Dari Load Scene

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class MainMenuControl : MonoBehaviour
{
    public void LoadGame(int level)
    {
        SceneManager.LoadSceneAsync(level);
    }

    public void Quit()
    {
        Application.Quit();
    }
}
```

Tabel diatas menjelaskan bahwa setiap fungsi gambaran implementasi dapat berjalan karena kode program dari tabel tersebut dan untuk LoadGame pada script tersebut dapat digunakan diberbagai tombol.

Tabel 5. 2 Salah 1 Hasil Perhitungan Fuzzy Mamdani

```
void FuzzyLogic()
{ if (myHealth.health <= 50)
  {
    defaultAttakTimer = 0.75f;
  }
}
```

```
        movementSpeed = 6f;  
        slashDamage = 15f;  
    }  
}
```

Tabel ini menjelaskan bahwa kondisi akan berubah jika hp kurang dari sama dengan 50% seperti yang sudah dihasilkan dari perhitungan logika fuzzy Mamdani, demikian juga untuk kedua bos memiliki kondisi yang sama perbedaannya hanya pada hp, bos kedua 200 hp dan bos ketiga 400 hp.

1.3 Desain Antarmuka

Desain antarmuka ini adalah tampilan permainan yang sudah dijalankan dan diimplementasikan dapat dilihat gambar seperti dibawah ini.



Gambar 5. 2 UI Opening

Tampilan awal dengan judul permainan dengan menekan tombol apapun dapat masuk dalam menu utama.



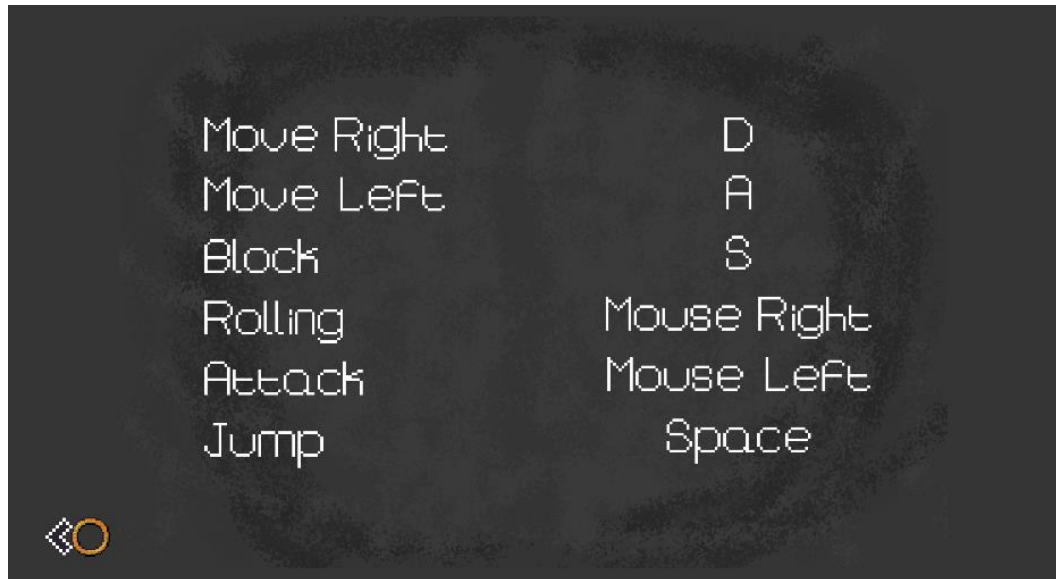
Gambar 5. 3 UI Menu utama

Menu utama dengan fitur-fitur tombol yang ada dapat dijelaskan pada gambar 4.4 struktur menu awal.



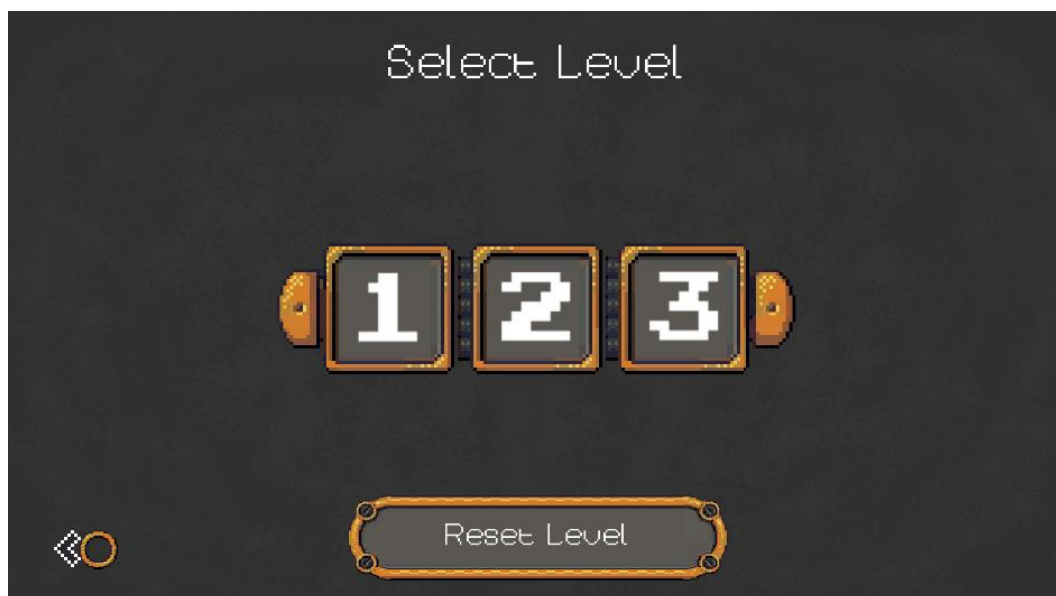
Gambar 5. 4 Tentang permainan

Keterangan tentang deskripsi permainan.



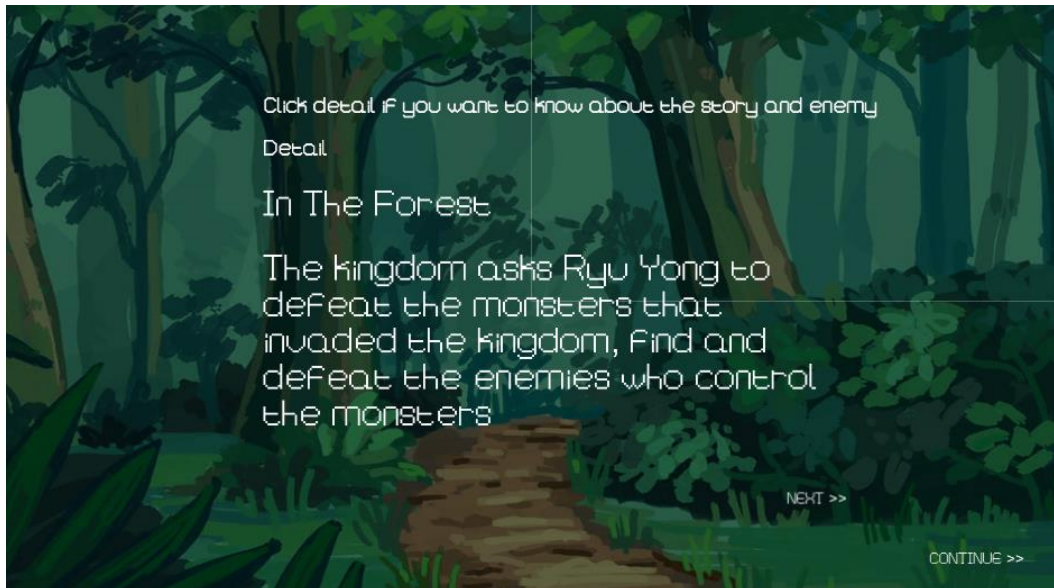
Gambar 5. 5 UI tutorial

Tampilan Tutorial berisi penjelasan bagaimana menjalankan permainan.



Gambar 5. 6 UI level

Tampilan *load game*, dapat memainkan permainan dan memilih *level* yang ingin dimainkan ulang.



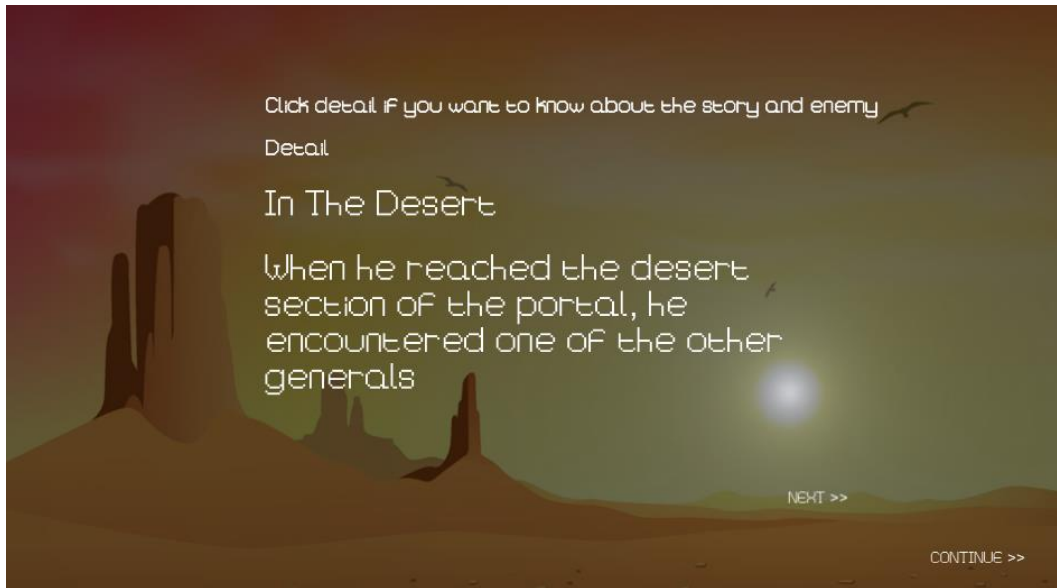
Gambar 5. 7 Intro dialog

Tampilan intro pertama/bos pertama memiliki story/dialog dijelaskan dan dipaparkan pada bab 4.



Gambar 5. 8 Bos pertama

Tampilan scene/bos pertama memiliki dialog dipaparkan di bab 4.



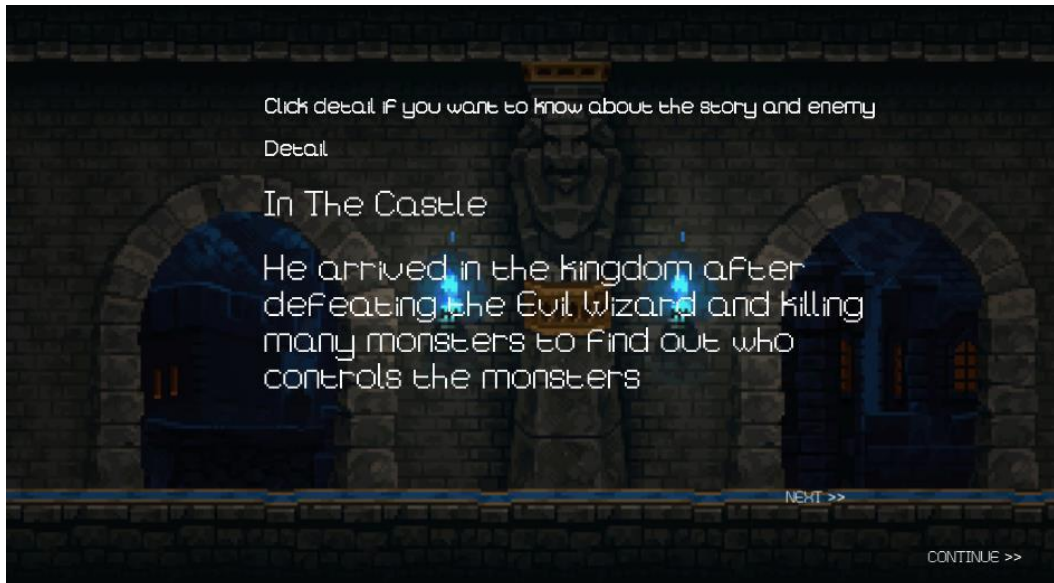
Gambar 5. 9 Intro bos kedua

Intro kedua/bos kedua memiliki story/dialog di bab 4.



Gambar 5. 10 Bos kedua

Scene bos kedua memiliki dialog dapat dilihat di bab 4.



Gambar 5. 11 Intro bos ketiga

Intro ketiga/bos ketiga memiliki story/dialog dipaparkan di bagian bab 4.



Gambar 5. 12 Bos ketiga

Scene bos ketiga memiliki dialog dapat dilihat di bab 4.



Gambar 5. 13 *Ending*

Akhir permainan memiliki beberapa teks dialog dapat dilihat di akhir dialog bagian bab 4.



Gambar 5. 14 Portal

Setiap level memiliki portal hanya berbeda warna ini akan muncul pada saat mengalahkan musuh, ini bertujuan untuk melanjutkan ke level berikutnya.

1.4 Pengujian Sistem

Pengujian sistem dilakukan setelah implementasi pembuatan, tahap ini memiliki tujuan untuk mengetahui apakah sistem sudah sesuai dengan yang diharapkan dan apakah ada kesalahan yang terjadi dengan implementasi program.

2.1.1 Metode Pengujian (*System Testing*)

Metode pengujian yang pertama pada sistem ini yaitu, pengujian yang didasarkan pada detail prosedur dan alur logika kode program. Pada kegiatan *system testing*, tester melihat *source code* program dan menemukan *bugs* dari kode program yang diuji. Intinya pengujian yang mencoba untuk menemukan cacat yang properti dari seluruh sistem.

2.1.2 Tujuan Pengujian

Pengujian dari keseluruhan sistem yang ada apakah fungsi *game* sudah sesuai dengan rancangan yang sudah dibuat.

2.1.3 Proses Pengujian

Tabel 5. 3 Proses Pengujian

Proses pengujian diperlihatkan dalam bentuk script atau code programnya dalam bentuk tabel dibawah ini.

No.	Script
Opening	<pre> Press Any Key void Update () { Invoke (nameof (ButtonShowSkip), 3f); if (Input.anyKeyDown) { if (!oneShootfx) { AudioSource.PlayClipAtPoint (fxButton, Vector3.zero); Invoke (nameof (LoadScene), 0.5f); oneShootfx = true; } } } </pre>

	<pre> void LoadScene () { AudioListener.pause = false; SceneManager.LoadScene ("MainMenu"); } </pre>
Menu utama	<pre> public void StartGame () { SceneManager.LoadScene ("Intro1"); } public void LoadGame (int level) { SceneManager.LoadSceneAsync (level); } public void ControlGame () { SceneManager.LoadScene ("Control"); } public void OpenPanelAbout () { if (activePanelAbout != null) { bool isActive = activePanelAbout.activeSelf; activePanelAbout.SetActive (!isActive); } } public void Quit () { Application.Quit (); } </pre>
Dialog intro	<pre> public void TriggerDialogue () { FindObjectOfType<DialogManager> ().StartDialogue (dialogue); } public void StartDialogue (Dialogue dialogue) { animator.SetBool ("isOpen", true); nameText.text = dialogue.name; sentences.Clear (); foreach (string sentence in dialogue.sentences) { sentences.Enqueue (sentence); } DisplayNextSentence (); } public void DisplayNextSentence () { if (sentences.Count == 0) </pre>

	<pre> { EndDialog(); return; } string sentence = sentences.Dequeue(); StopAllCoroutines(); StartCoroutine(TypeSentence(sentence)); } IEnumerator TypeSentence (string sentence) { dialogueText.text = ""; foreach(char letter in sentence.ToCharArray()) { dialogueText.text += letter; yield return null; } } void EndDialog() { animator.SetBool("isOpen", false); } public void LoadGame(int level) { SceneManager.LoadSceneAsync(level); } </pre>
Nyawa	<pre> Bos 1 if (myHealth.health > 50) { defaultAttakTimer = 2f; movementSpeed = 4f; slashDamage = 7f; } if (myHealth.health <= 50) { defaultAttakTimer = 0.75f; movementSpeed = 6f; slashDamage = 15f; } Bos 2 if (myHealth.health > 100) { defaultAttakTimer = 2f; movementSpeed = 4f; evilAttack1 = 7f; evilAttack2 = 7f; } if (myHealth.health <= 100) { defaultAttakTimer = 0.75f; movementSpeed = 6f; } </pre>

	<pre> evilAttack1 = 15f; evilAttack2 = 15f; } Bos 3 if (myHealth.health > 200) { defaultAttakTimer = 2f; movementSpeed = 4f; punchDamage = 7f; slashDamage = 7f; fireDamage = 7f; } if (myHealth.health <= 200) { defaultAttakTimer = 0.75f; movementSpeed = 6f; punchDamage = 15f; slashDamage = 15f; fireDamage = 15f; } </pre>
<i>Timer</i>	<pre> Bos 1 if(GetPlayer.infoTimers > 45f) { defaultAttakTimer = 4.14f; movementSpeed = 1.37f; slashDamage = 3.45f; } Bos 2 if(GetPlayer.infoTimers > 95) { defaultAttakTimer = 4.11f; movementSpeed = 1.42f; evilAttack1 = 3.55f; evilAttack2 = 3.55f; } </pre>
Daya serang	<code>public float slashDamage = 2;</code>
Item	<pre> Penambah serangan private void OnTriggerEnter2D(Collider2D take) { if(take.tag == Tags.Tambah_Attack) { SoundManagerScript.PlaySound("Collectable"); slash.slashDamage += 5; Destroy(take.gameObject); } Penambah health if(take.tag == Tags.Tambah_Health) { SoundManagerScript.PlaySound("Collectable"); healt.health += 20; Destroy(take.gameObject); } } </pre>

	<pre> } } Penambah Waktu if(take.tag == Tags.Tambah_Waktu) { SoundManagerScript.PlaySound("Collectable"); slash.infoTimers -= 20; Destroy(take.gameObject); } </pre>
Portal	<pre> Portal next level 1 private void OnTriggerEnter2D(Collider2D collision) { if(collision.tag == Tags.Player_Tag) { LoadingScreenControl.instance.LoadScene(3); LevelControlScripts.instance.youWin(); } } Portal next level 2 public class EnterTriggerScript2 : MonoBehaviour { private void OnTriggerEnter2D(Collider2D collision) { LoadingScreenControl.instance.LoadScene(5); LevelControlScripts.instance.youWin(); } } Portal next level 3 private void OnTriggerEnter2D(Collider2D collision) { LoadingScreenControl.instance.LoadScene(8); LevelControlScripts.instance.youWin(); } </pre>
Tombol skip	<pre> void Update() { Invoke(nameof(ButtonShowSkip), 24f); } void ButtonShowSkip() { skip.gameObject.SetActive(true); } public void BackToOpeing() { SceneManager.LoadScene("OpeningGame"); } </pre>

2.1.4 Analisa Hasil Pengujian

Dari pengujian implementasi kode program dan menjalankan permainan mendapatkan hasil yang baik dan kode berjalan dengan baik tidak ada masalah yang terjadi.

Disini juga sudah termasuk dalam pemeliharaan sistem karena menguji dan menjalankan dan mengecek berbagai kode program sehingga mendapatkan hasil yang sesuai dengan rancangan sistem.

2.1.5 Metode Pengujian (*Usability Testing*)

Pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi yaitu, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan rancangan aplikasi. Pengujian ini tidak melihat dan menguji source code program dan responden yang menguji sudah mengetahui cara bermain *game*.

Usability testing merupakan sebuah metode untuk mengevaluasi user experience (UX) terhadap *software* ataupun website yang dibuat. Pada umumnya, metode ini dilakukan oleh para *UX developer* dengan melibatkan beberapa *user* (pengguna) tertentu untuk diteliti bagaimana proses mereka selama berinteraksi dengan website.

2.1.6 Tujuan Pengujian

Menguji permainan dengan memberikan pengalaman terhadap pengguna dan mendapatkan umpan balik, pengujian ini dilakukan dengan melakukan penyebaran kuesioner.

Penyebaran diserahkan kepada 6 Mahasiswa/Mahasiswi dan 14 diserahkan kepada masyarakat umum dengan *range* umur 20-30 Tahun, pengujian ini diberikan bukan kepada orang awam melainkan kepada orang yang telah mengetahui apa itu permainan dan bagaimana cara bermain *game*.

2.1.7 Proses Pengujian

Tabel 5. 4 Pertanyaan Kuesioner

No.	Pertanyaan	Pilihan
1	Deskripsi mengenai link permainan:	-

	Memberikan link download permainan yang akan dicoba oleh responden							
2	Nama responden	Teks bebas						
3	Jenis Kelamin	<table border="1"> <tr> <td>Laki-Laki</td> </tr> <tr> <td>Perempuan</td> </tr> </table>	Laki-Laki	Perempuan				
Laki-Laki								
Perempuan								
4	Bagaimana permainan menurut responden?	<table border="1"> <tr> <td>Tidak menarik</td> </tr> <tr> <td>menarik</td> </tr> <tr> <td>Sangat menarik</td> </tr> <tr> <td>Membosankan</td> </tr> <tr> <td>Tidak membosankan</td> </tr> </table>	Tidak menarik	menarik	Sangat menarik	Membosankan	Tidak membosankan	
Tidak menarik								
menarik								
Sangat menarik								
Membosankan								
Tidak membosankan								
5	Bagaimana desain antarmuka yang terlihat?	<table border="1"> <tr> <td>Tidak Bagus</td> </tr> <tr> <td>Bagus</td> </tr> <tr> <td>Menarik</td> </tr> <tr> <td>Tidak Menarik</td> </tr> <tr> <td>Kacau</td> </tr> <tr> <td>Acak tidak sesuai dengan judul</td> </tr> </table>	Tidak Bagus	Bagus	Menarik	Tidak Menarik	Kacau	Acak tidak sesuai dengan judul
Tidak Bagus								
Bagus								
Menarik								
Tidak Menarik								
Kacau								
Acak tidak sesuai dengan judul								
6	Apakah permainan memiliki kendala/bug?	Teks bebas						
7	<p>Deskripsi</p> <p>Fungsi ini yang difokuskan dalam penelitian yaitu mengenai kecerdasan buatan untuk musuh atau biasa disebut non-player character, kecerdasan yang diterapkan pada musuh yaitu tingkat kesulitannya.</p> <p>Pernyataan = Di setiap level, tingkat kesulitan akan berubah pada kondisi hp 50% dari 100%</p> <p>Pertanyaan</p> <p>Dari pernyataan diatas apakah kondisi tersebut terpenuhi dan sesuai pernyataan?</p>	<table border="1"> <tr> <td>Terpenuhi</td> </tr> <tr> <td>Tidak Terpenuhi</td> </tr> </table>	Terpenuhi	Tidak Terpenuhi				
Terpenuhi								
Tidak Terpenuhi								

8	Tingkat kesulitan pada boss Level 1	Mudah	
		Sulit	
		Sangat Sulit	
9	Tingkat kesulitan pada boss Level 2	Mudah	
		Sulit	
		Sangat Sulit	
10	Tingkat kesulitan pada boss Level 3	Mudah	
		Sulit	
		Sangat Sulit	
11	Tolong berikan kesimpulan dan saran apa yang kurang dari penelitian terutama pada tingkat kesulitan musuh dan apa yang perlu ditingkatkan, terimakasih atas partisipasinya.	Teks bebas	

2.1.8 Analisa Hasil Pengujian

Dari pertanyaan-pertanyaan diatas maka akan diketahui hasil pengujian berupa pilihan jawaban yang sudah tersedia, link download yang sudah tersedia juga. Untuk jawaban menurut pendapat responden mengenai kesimpulan dan saran, itu akan tersedia setelah pengujian kuesioner telah selesai.