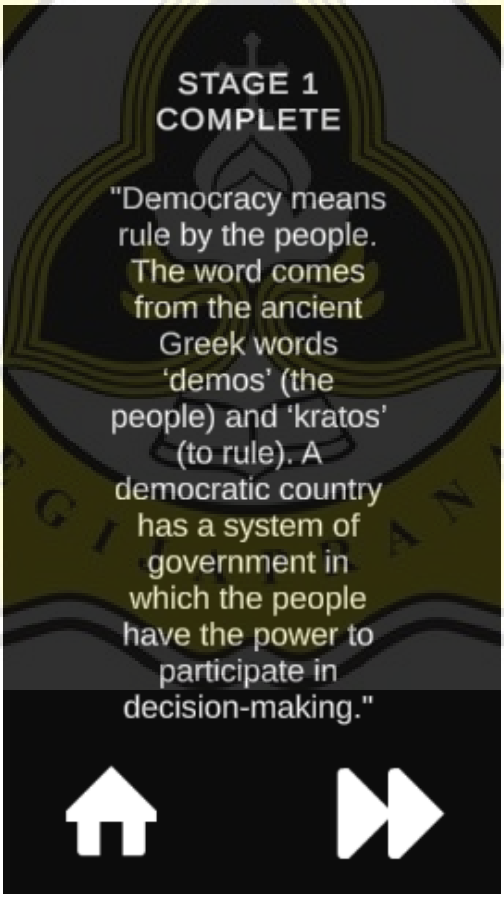
Bab IV

Hasil Penelitian dan Pembahasan

4.1 Implementasi Pembelajaran Sistem Pemerintahan

Dalam *game* "Governman", tujuan utama dari pembuatan *game* tersebut adalah untuk mengenalkan pada pengguna tentang sistem pemerintahan di Indonesia. Untuk mencapai tujuan tersebut perlu dilakukan proses perancangan yang harus memasukan konsep sistem pemerintahan di Indonesia. Dalam pengembangan *game* "Governman" pengembang akan mengambil konsep dasar serta implementasi dari sistem pemerintahannya,



Gambar 4. 1 Implementasi pembelajaran sistem pemerintahan

Seperti gambar 4.1, teknik implementasi sistem pemerintahan dalam *game* "Governman" melalui tutorial dan ketika pengguna menyelesaikan stage yang ada didalamnya. Setiap pengguna atau *player* menyelesaikan *game* akan selalu muncul penjelasan tentang sistem pemerintahan tersebut, dan setiap stagenya berbeda-beda, dan ketika *player* sampai pada *stage* 6 atau *stage* terakhir, sebelum memulai stage pengguna akan melakukan yang namanya pemilu.

Gambar 4. 2 Implementasi referendum untuk simulasi kampanye pemilu

Seperti pada gambar 4.2, pemain akan mendapatkan kesempatan untuk terpilih kembali sebagai pemimpin di kota tersebut atau dengan kata lain referendum yang dimaksud disini lebih kepada pertanyaan untuk pengguna (sesi wawancara), pengguna akan dihadapkan beberapa pertanyaan yang menyinggung permasalahan di kotanya. Sehingga jika pengguna memilih jawaban yang salah maka reputasi dari pengguna akan berkurang dan memberikan efek untuk tidak dipilih oleh masyarakat. Pemberian edukasi yang ada dalam *game* "Governman" ada beberapa poin, berikut poin poin yang ada dalam pemberian edukasi tersebut.

1. Dasar sistem pemerintahan

   Dasar sistem pemerintahan yang di edukasikan seperti apa itu sistem pemerintahan, bagaimana sistem pemerintahan berjalan dan sebagainya.

2. Sistem pemerintahan di Indonesia

   Yang menjadi pembelajaran dalam sistem pemerintahan di Indonesia merupakan, sistem pemerintahan presidensial.

3. Pemecahan masalah

   Pemain diberikan edukasi mengenai bagaimana memecahkan suatu masalah dan menanganinya dengan baik.

4.2 Perancangan Game

Pembuatan *game* "Governman" menggunakan model *game* simulasi untuk mengenalkan sistem pemerintahan yang ada di Indonesia, *game* ini menggunakan perspektif 2D dan menggunakan sistem *staging* dimana pemain harus menyelesaikan setiap *stage* yang memberikan pengajaran tentang sistem pemerintahan tersebut. Dalam *game* ini setiap stagenya memiliki permasalahan yang berbeda-beda terutama pada beberapa stage tertentu seperti pada stage terakhir. Dalam stagenya pemain diharuskan untuk mengambil *item* atau benda tertentu untuk mendapatkan uang sehingga dapat menyelesaikan stagenya. lalu setelah selesai mengambil *item* dan mengumpulkan uang dengan jumlah tertentu, pemain harus membuat upgrade untuk menyelesaikan stage tersebut. Agar *game* "Governman" tetap menarik dan tidak membosankan, pada stage tertentu diberikan *minigames* dan tingkat kesulitan yang berbeda-beda pula.

4.3 Game Design

Game "Governman" merupakan *game* 2D dengan sistem simulasi sebagai sistem utamanya, dengan tujuan untuk memberikan pembelajaran tentang sistem pemerintahan. "Governman" sendiri dibagi menjadi beberapa stage dengan sistem yang sama tetapi berbeda tingkat kesulitan, dengan menggunakan sistem tapping untuk menyelesaikan stagenya.

4.4 Pre Production

Dalam masa *Pre - Production*, pengembang *game* mengumpulkan data dan ide serta membuat model produk. Pada titik ini, pengembang juga memutuskan bagaimana membuat bentuk akhir dari produk tersebut. Masa *pre-production* ini dibagi menjadi dua yang pertama adalah *prototype* atau desain *awal* dan yang kedua merupakan desain akhir atau final.

4.4.1 Prototype

Dalam masa desain awal atau *prototype game* "Governman" mengalami banyak perubahan, dari awal lebih pada tipe *open world*, sampai sekarang dibagi menjadi beberapa *stage*. Pembuatan *prototype* terinspirasi dari game "simcity" dengan sistem mendapatkan uang secara *overtime* atau per detik.

Gambar 4. 3 Prototype Governman

Seperti Gambar 4.3, prototype dari *game* "Governman" merupakan *game* dengan map yang dapat digeser sehingga pengguna dapat melihat kota yang dimiliki dan dapat meng-*upgrade* kota tersebut seperti di *game* "Simcity". Dalam *prototype* awal map yang akan digunakan mempunyai beberapa daerah yang terpisah dan memiliki keberagamannya sendiri serta memiliki permasalahannya sendiri. Pada Gambar 4.3 bentuk dari *map* yang ada berupa kotak agar sesuai dengan layar *handphone* yang *potrait*.

Gambar 4. 4 Tab upgrade city

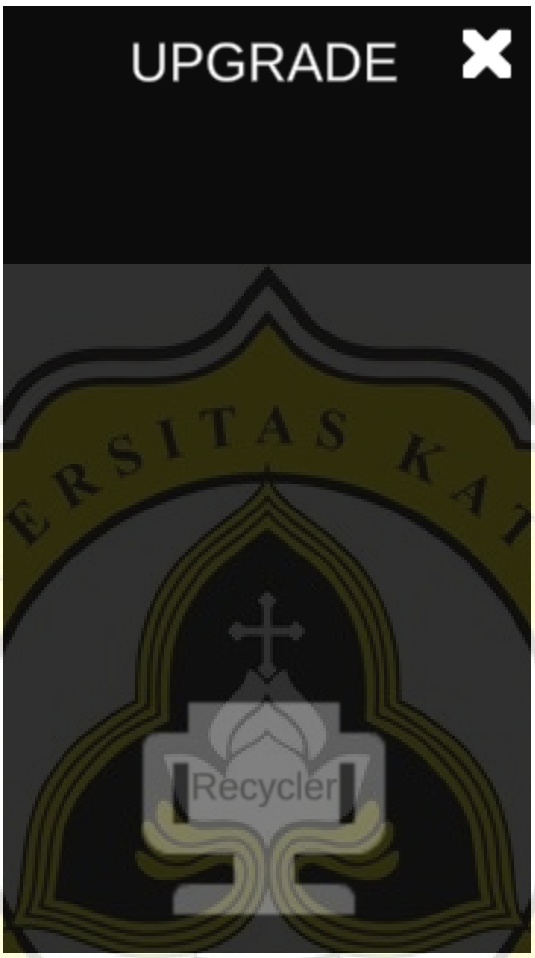
Pada Gambar 4.4 yang merupakan tab upgrade city, digunakan untuk melakukan upgrade kota yang dimiliki oleh pemain. Setiap bangunan yang diupgrade dapat menambahkan value dari kota sehingga *income* yang didapatkan meningkat, beberapa upgrade juga berfungsi untuk meningkatkan kebahagiaan masyarakatnya.

4.4.2 Desain final

Pada desain akhir atau final dari *game* "Governman", *gameplay* yang awalnya lebih leluasa kini menjadi lebih rapat dengan pembagian stage yang ada. Pembagian stage ini bertujuan agar tujuan utama dari pembuatan *game* ini tercapai, yaitu memberikan edukasi tentang sistem pemerintahan.

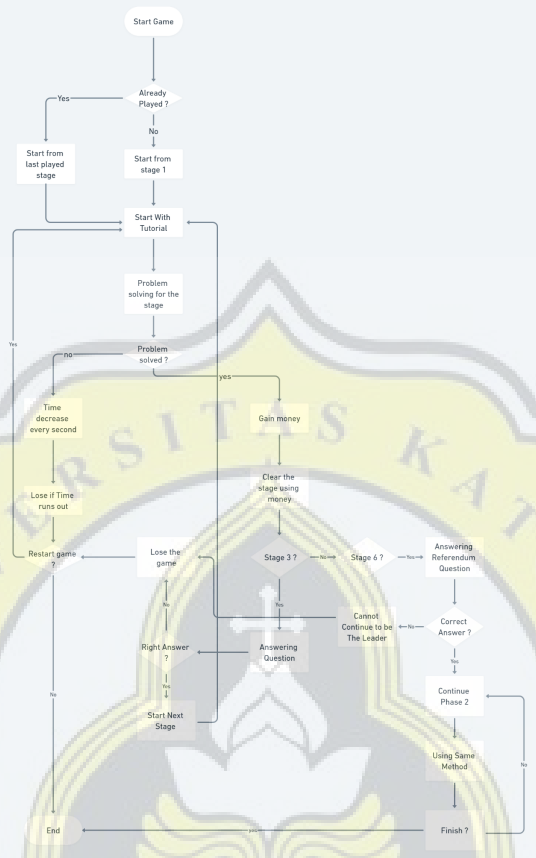
Gambar 4. 5 Level menu

Gambar 4.5 menunjukan pembagian *stage* yang ada di *game* "Governman", masing-masing *stage* memberikan pengajaran yang berbeda-beda tentang sistem pemerintahan.



Gambar 4. 6 Fitur upgrade

Gambar 4.6 yang merupakan fitur upgrade dari desain final merupakan fitur yang diambil untuk memberikan kesan simulasi di dalam game "Governman". Fitur upgrade ini juga merupakan kunci untuk menyelesaikan stage yang ada, ketika uang yang dimiliki pemain cukup maka dapat membeli upgrade yang langsung meningkatkan kemakmuran masyarakat sehingga mencapai batas kemakmuran yang ada. Gambar 4.7 dibawah merupakan *game flow* dari *game* "Governman" dimulai dari awal *player* masuk *menu* hingga selesai *game*.



Gambar 4. 7 Game Flow

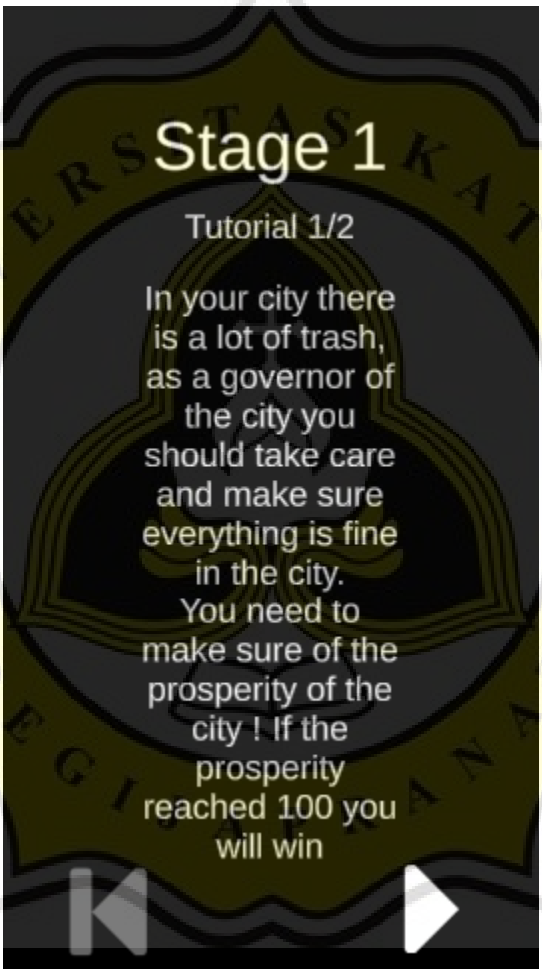
Gameplay dari game "Governman" lebih berfokus pada *stage*, pada bagian awal yaitu bagian menu, dibuat sangat simpel dengan button start dan exit. Tutorial untuk pemain akan diberikan ketika pemain sudah memilih stage yang ingin dimainkan.

4.5 Implementasi Pembuatan

Implementasi pembuatan game akan disesuaikan dengan perancangan yang sudah dijelaskan sebelumnya, ada beberapa point seperti *tutorial*, *gameplay*, dan kondisi untuk menyelesaikan gamenya.

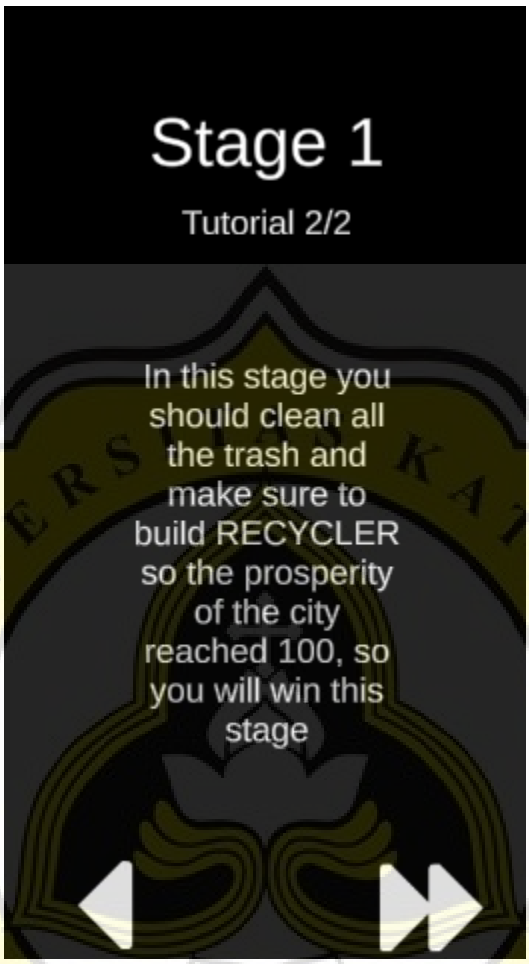
4.5.1 Implementasi Tutorial

Pada awal memulai *game*, pemain diberikan instruksi bagaimana cara bermain "Governman". Tampilan dari instruksi tersebut berupa *scene* yang berbeda, instruksi atau *tutorial* tersebut menunjukan bagaimana cara bermain dan juga menceritakan permasalahan yang ada di kota yang dimainkan oleh pemain.



Gambar 4. 8 Tutorial page 1

Gambar 4.8 menunjukan bagaimana kota pemain memiliki masalah dan bagaimana cara menyelesaikan stage tersebut.



Gambar 4. 9 Tutorial page 2

Pada gambar 4.9, diberikan petunjuk juga bagaimana pemain harus menyelesaikan stagenya, contohnya pada stage 1 pemain harus membuat *recycler* untuk menyelesaikan stage tersebut.

Dalam tab tutorial yang ada pada gambar 4.8 pemain harus masuk ke halaman selanjutnya untuk masuk ke *main stage*, sementara itu di gambar 4.9 memberikan pilihan untuk kembali ke halaman pertama atau masuk ke dalam stagenya.

```
public void Panel1()
{
 tutor01.SetActive(false);
 tutor02.SetActive(true);
}
 public void Panel2()
{
 tutor02.SetActive(false);
 tutor01.SetActive(true);
}

 public void Scene1() {
    SceneManager.LoadScene("Stage1");
 }
```

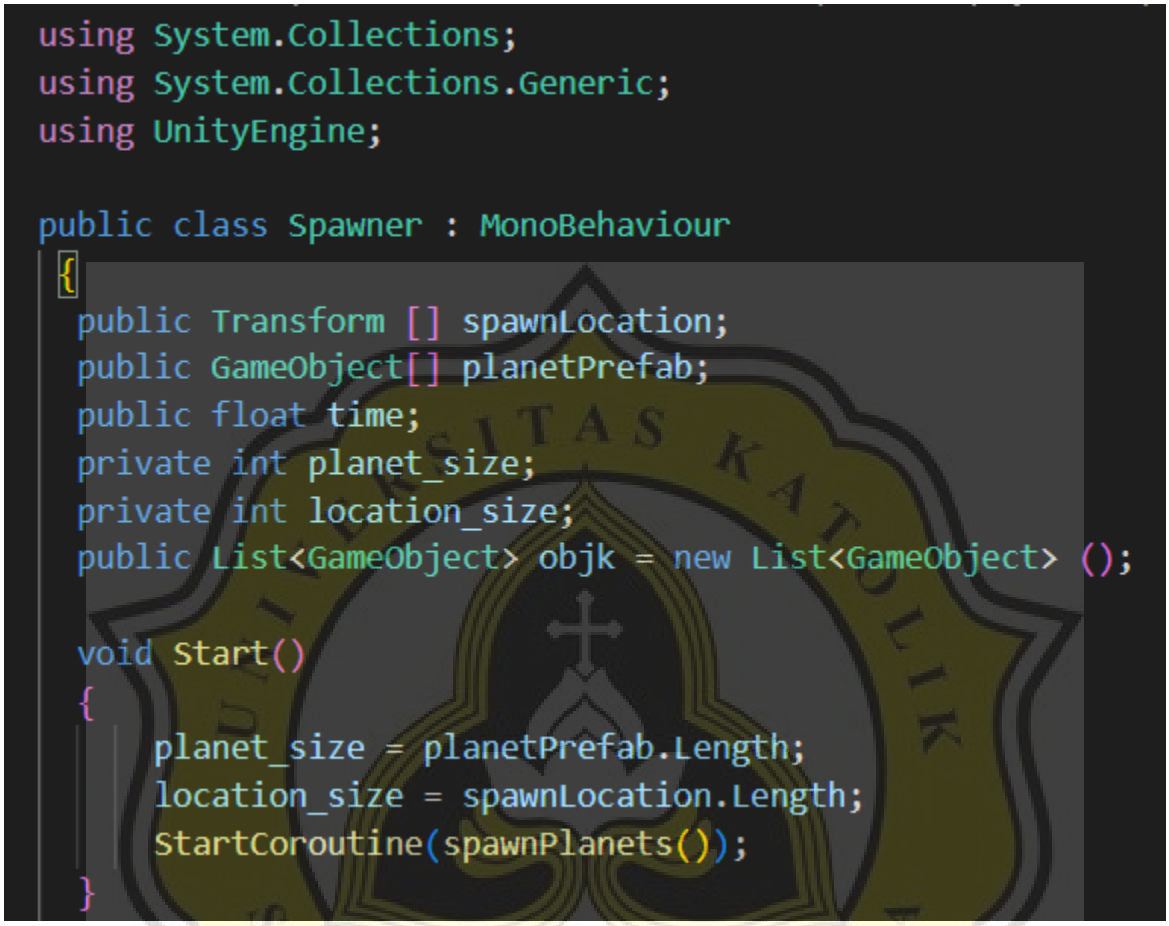
Gambar 4. 10 Script di tab tutorial

Pada gambar 4.10, fungsi utama dari *script* tersebut adalah untuk membuka panel pertama atau panel kedua. Fungsi *panel1()* merupakan fungsi untuk masuk ke halaman tutorial selanjutnya, sedangkan fungsi *panel2()* merupakan fungsi untuk kembali ke halaman sebelumnya. Lalu fungsi terakhir yang ada dalam script ini berfungsi untuk membuka scene stage dari tutorial tersebut.

4.5.2 Implementasi Gameplay

- Item Generator

Gameplay "Governman" merupakan *tap* pada *item* yang muncul untuk mendapatkan uang dan menambahkan *prosperity*, *item* yang muncul dalam *game* "Governman" merupakan *designated spawn* atau tempat spawn yang sudah di desain untuk menjadi tempat munculnya *item* tersebut.



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

public class Spawner : MonoBehaviour
{
    public Transform [] spawnLocation;
    public GameObject[] planetPrefab;
    public float time;
    private int planet_size;
    private int location_size;
    public List<GameObject> objk = new List<GameObject> ();

    void Start()
    {
        planet_size = planetPrefab.Length;
        location_size = spawnLocation.Length;
        StartCoroutine(spawnPlanets());
    }
```

Gambar 4. 11 Spawner.cs

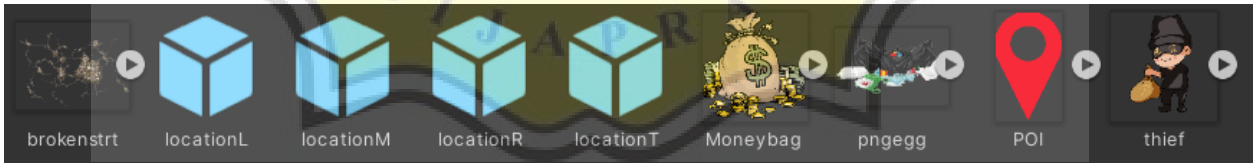
```
private IEnumerator spawnPlanets()
{
    objk.Clear();
    while (true)
    {
        GameObject spawned;
        int planet_index = Random.Range(0, planet_size);
        int location_index = Random.Range(0, location_size);

        spawned = Instantiate(planetPrefab[planet_index],  spawnLocation[location_index].position, Quaternion.identity) as GameObject;
        spawned.transform.parent = spawnLocation[location_index];
        objk.Add(spawned);
        yield return new WaitForSeconds(time);
    }
}
}
```

Gambar 4.11 Spawner.cs (lanjutan)

Gambar 4.11 merupakan *script* untuk *spawn item* yang diinginkan, dalam *script* tersebut pengembang mengambil *transform* yang merupakan lokasi dari spawner lalu mengambil prefab yang digunakan sebagai *item*.

Fungsi *spawner* ini juga membuat *list* atau *array* baru sebagai fungsi utama dari spawn-nya. Data dari *array* tersebut diambil dari kombinasi antara *transform* dan prefab objek yang digunakan untuk menjadi item.

Gambar 4. 12 Prefab untuk menyimpan transform



Gambar 4. 13 Prefab item yang di spawn

Pada Gambar 4.12, prefab yang sudah disiapkan (yang berada di *Hierarchy*) merupakan prefab untuk mengambil lokasi atau *transform*, dari hal itu prefab pada gambar 4.13 dapat *spawn* ke lokasi yang sudah ditentukan melalui gambar 4.12 tersebut.

- Game Manager

  Game Manager merupakan *script* yang menampung fungsi-fungsi utama dalam sebuah *game*. Fungsi lain dari *game manager* adalah dapat dipanggil oleh *script* lain sehingga *game* dapat berjalan dengan lancar.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class GameManager : MonoBehaviour
{
    private static GameManager _instance;

    public static GameManager Instance {
        get{
            return _instance;
        }
    }

    public int mahkneek = 0 ;
    public TMP_Text moneitxt;
    public int prospery = 0;
    public TMP_Text prosprtxt;
    public GameObject wins;
    public Button btn;
    public GameObject building1;
    public GameObject building2;
    public int goalm = 50;
    public int goalp = 100;

    void Awake ()
    {
        if(_instance != null && _instance != this)
        {
            Destroy(this.gameObject);
        }
        else
        {
            _instance = this;
        }
```

Gambar 4. 14 Script Game Manager

```
}
// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{
    moneitxt.text = $"Money: {mahkneek}";
    prosprtxt.text = $"Prosper: {prospery}";

    if (prospery >= goalp)
    {
        wins.SetActive(true);
    }
     if(mahkneek >= goalm)
    {
       btn.GetComponent<Button>().interactable =true ;
    }
}

public void AddMoney()
{
    mahkneek += 5;
    prospery += 5;
}

public void Buys()
{
   building1.SetActive(true);
   prospery += 100;

}
}
```

Gambar 4.14 Script Game Manager (lanjutan)

Pada gambar 4.14, fungsi utamanya adalah untuk menambah uang dan *prosperity* setelah mengambil item dengan menggunakan tools, lalu setelah uangnya cukup pemain dapat membangun suatu bangunan untuk menyelesaikan *game*-nya. Kemudian setelah membangun bangunan, ketika *prosperity* sudah mencukupi maka akan menampilkan panel *wins*.

Gambar 4. 15 Gameobject Game Manager

Game objek yang ada pada gambar 4.15 merupakan *game* objek yang menampung *script game* manager, supaya dapat mengambil *point-point* penting seperti *text uang*, *text prosperity*, tombol untuk *upgrade*, *goals* uang dan prosperity, *game* objek untuk bangunan, dan juga panel *wins*, Semua aset-aset penting dimasukan kedalam *game* objek ini.

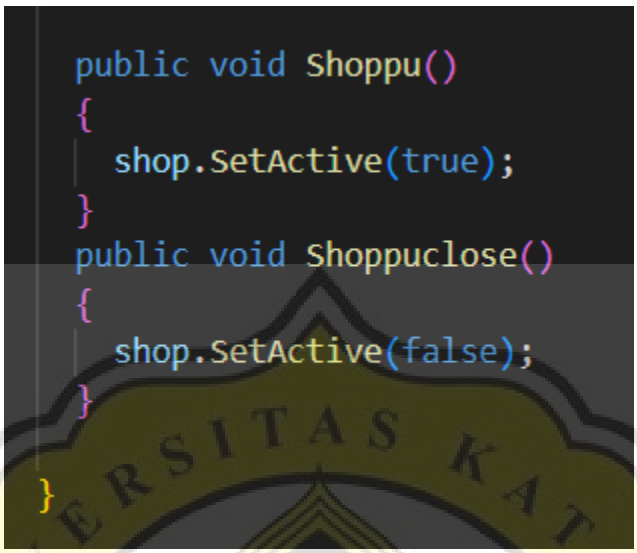
- Sistem ambil item

  Sistem yang ada disini digunakan untuk mengambil *item* dengan menggunakan button yang ada pada bagian bawah kiri layar. Ketika pemain ingin mengambil item, pemain harus

memencet tombol tersebut. Pemain yang belum memencet tombol tersebut tidak dapat memencet *item* yang sudah ter-*spawn*.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class Systemm : MonoBehaviour
{
  public GameObject obj;
  public Button brm;
  public Spawner spawner;
  public Spawner1 spawner1;
  public TMP_Text money;
  public TMP_Text Prosper;
  public int mahknee = 0;
  public int prospert = 0;
  public GameObject shop;

  public void Broom ()
  {
    foreach (GameObject obk in spawner.objk)
    {
      obk.GetComponent<Button>().interactable = true;

    }
  }
   public void Shovel ()
  {
    foreach (GameObject obk1 in spawner1.objk1)
    {
      obk1.GetComponent<Button>().interactable = true;

    }
  }
```

Gambar 4. 16 Systemm.cs



Gambar 4.16 Systemm.cs (lanjutan)

Pada gambar 4.16, merupakan *script* yang digunakan untuk mengaktifkan sifat dari *item* yang merupakan sebuah tombol untuk dapat berinteraksi dengan pemain. Jika pemain memencet tombol tools atau pada gambar 4.16 merupakan *Broom()* maka *gameobject* obk dari script spawner akan menjadi *interactable* atau dapat berinteraksi dengan *item* tersebut.

Gambar 4. 17 Prefab item dalam inspector

Pada gambar 4.17, menunjukan bagian *interactable* tidak di centang, yang mengindikasikan bahwa ketika *item* tersebut ter-*spawn* maka tidak dapat berinteraksi dengan *item* tersebut.

4.6 Pengujian Statistik

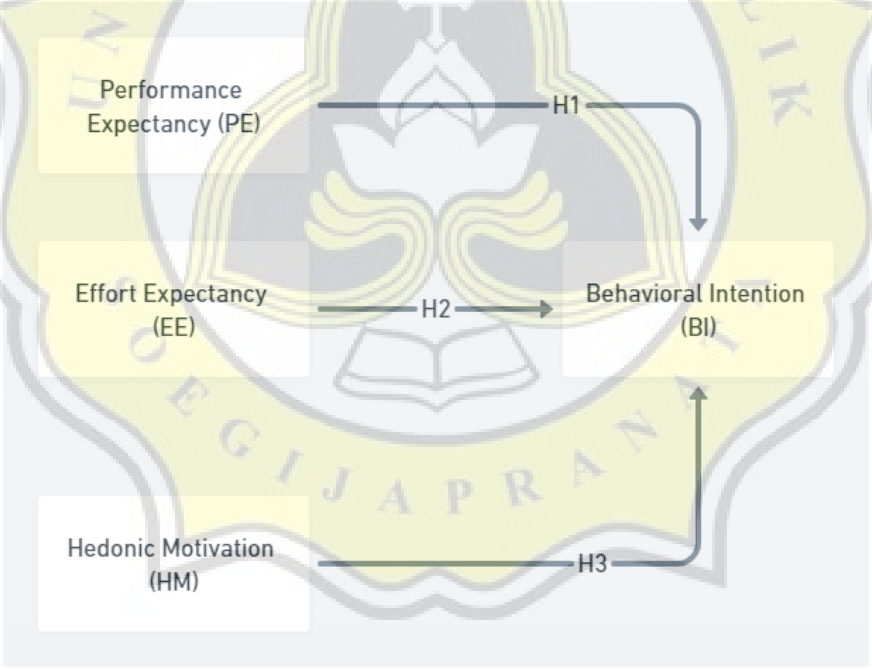
Setelah selesai dalam mengembangkan *game* "Governman" maka akan ditentukan hipotesis yang berguna untuk memprediksi hasil akhir dari penelitian dalam mengedukasi sistem pemerintahan.

Berikut merupakan hipotesis yang sudah disusun:

H1: Pengedukasian sistem pemerintah melalui *game* simulasi "Governman" menarik dan mengedukasi

H2: Pengedukasian sistem pemerintah melalui *game* simulasi "Governman" mudah dipahami.

H3: Pengedukasian sistem pemerintah melalui *game* simulasi "Governman" menarik dan menantang untuk dimainkan.



Gambar 4. 18 Hipotesa relasi antar variabel

Sesuai dengan gambar 4.18 menunjukan korelasi antar variabel. Korelasi ini berupa hubungan antara *Performance Expectancy* dan *Behavioral Intention*, *Effort Expectancy* dan *Behavioral Intention*, serta yang terakhir *Hedonic Motivation* dan *Behavioral Intention*.

4.6.1 Rancangan pertanyaan kuesioner

Untuk mendapatkan hasil yang terbaik dari responden, peneliti merancang kuesioner berdasarkan hipotesis yang telah disiapkan. Berikut merupakan daftar pertanyaan yang dijadikan tolak ukur dalam penelitian.

- *Performance Expectancy*

PE1: Ketika memainkan *game* "Governman" dapat membantu saya dalam memahami sistem pemerintahan

PE2: Dengan memainkan *game* "Governman" dapat membantu saya dalam melihat dan menganalisa sebuah masalah.

PE3: Dengan memainkan *game* "Governman" saya dapat melihat permasalahan dan membentuk strategi untuk menyelesaikan masalah.

- *Effort Expectancy*

EE1: Memahami cara bermain *game* "Governman" mudah bagi saya

EE2: Menurut saya, instruksi bermain di *game* "Governman" sangat jelas dan mudah dipahami

EE3: Konsep permainan dalam *game* "Governman" mudah untuk dipahami.

EE4: Saya sangat cepat memahami cara bermain *game* "Governman".

- *Hedonic Motivation*

HM1: Menurut saya, permainan yang ada dalam *game* "Governman" menarik.

HM2: Menurut saya, permainan "Governman" sangat menantang.

HM3: Memainkan *game* "Governman" dapat membuat saya merasa senang.

- *Behavioral Intention*

BI1: Saya berencana untuk tetap memainkan *game* "Governman" di kemudian hari.

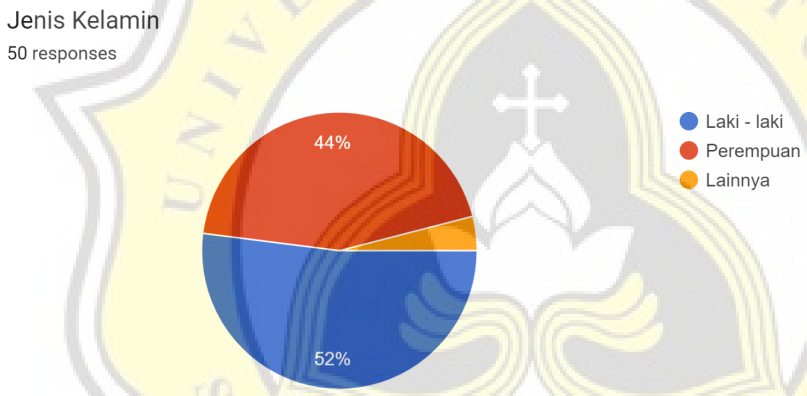
BI2: Saya akan terus mencoba memainkan *game* "Governman" di kehidupan saya sehari - hari karena mengedukatif.

BI3: Saya berencana memainkan *game* "Governman" lebih sering karena menantang.

4.6.2 Data Responden

1. Jenis Kelamin

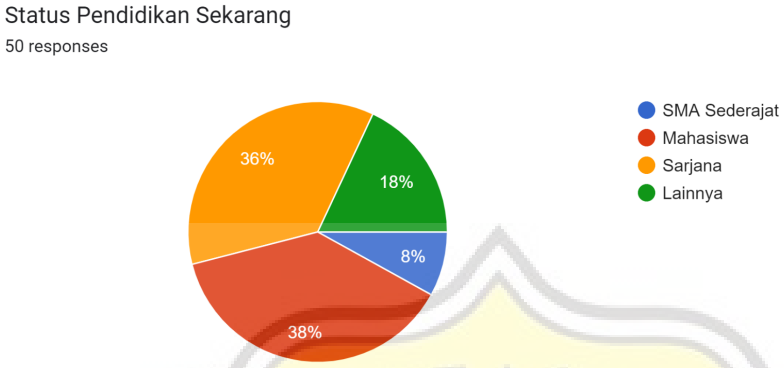
Dari total 50 responden, jenis kelamin terbanyak merupakan laki-laki dengan nilai persentase 52% atau sebanyak 26 responden, sedangkan responden perempuan memiliki persentase sebesar 44% atau sebanyak 22 responden, dan yang terakhir dengan persentase 4% atau sebanyak 1 orang merupakan jenis kelamin lainnya dikarenakan responden tidak ingin diketahui jenis kelaminnya. Persentase tersebut dapat dilihat dari diagram yang sesuai dengan gambar 4.19.



Gambar 4. 19 Persentase Jenis Kelamin

2. Status Pendidikan Sekarang

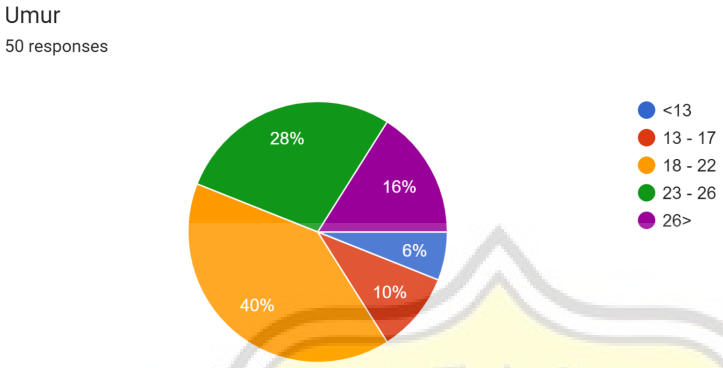
Pada gambar 4.20 dari total 50 responden, responden paling banyak dengan persentase 38% atau sebanyak 19 responden merupakan mahasiswa, lalu responden dengan persentase sebanyak 36% atau 18 orang merupakan sarjana, selanjutnya responden dengan persentase sebanyak 18% atau 9 responden merupakan lainnya, dan yang terakhir dengan persentase 8% atau sebanyak 4 responden adalah SMA sederajat.



Gambar 4. 20 Persentase Status Pendidikan Sekarang

3. Usia

Dalam gambar 4.21 dari total 50 responden, mayoritas responden dengan persentase 40% atau 20 responden memiliki rentang usia 18-22 tahun, sedangkan rentang usia 23-26 tahun memiliki persentase sebesar 28% atau sebanyak 14 responden, lalu dengan rentang usia lebih dari 26 tahun memiliki persentase 16% atau sebanyak 8 responden, kemudian dengan rentang usia 13-17 tahun memiliki persentase sebanyak 10% atau sebanyak 5 responden, dan yang terakhir dengan persentase 6% atau sebanyak 3 responden memiliki rentang usia kurang dari 13 tahun.



Gambar 4. 21 Persentase usia responden

4.6.3 Statistik Deskriptif

Statistik Deskriptif dapat dilihat pada Tabel 4.1, tabel ini berisikan tentang data keseluruhan responden, dengan total 50 responden dan dengan ketentuan pada tabel gender nomor 1 adalah laki - laki dan nomor 2 adalah perempuan. Pada tabel Education nomor 1, responden saat ini sedang menempuh pendidikan SMA Sederajat, nomor 2, responden merupakan seorang Mahasiswa, nomor 3, responden kini sudah memiliki status pendidikan Sarjana, dan angka 4, responden memiliki status pendidikan lainnya. Pada tabel umur nomor 1 menunjukan usia dibawah 13 tahun, lalu nomor 2 menunjukan usia 13-17 tahun, nomor 3 menunjukan usia 18-22 tahun, nomor 4 menunjukan usia 22-26, dan nomor 5 menunjukan responden yang berusia diatas 26 tahun. Dalam keterangan tabel gender, memiliki rata-rata sebanyak 1.46 dengan nilai tengah 1, lalu dalam keterangan tabel education, memiliki nilai rata-rata sebanyak 2.62 dengan nilai tengah 3, dan yang terakhir keterangan tabel umur, memiliki nilai rata rata sebanyak 3.38 dengan nilai tengah 3.

Tabel 4. 1 Statistik Deskriptif

Statistics

		gender	education	Umur
N	Valid	50	50	50
	Missing	0	0	0
Mean		1.46	2.62	3.38
Median		1.00	3.00	3.00
Mode		1	2	3
Sum		73	131	169

4.6.4 Uji Validasi

Uji validasi diperlukan untuk mengetahui apakah alat yang diujikan merupakan data yang valid atau tidak valid. Data yang digunakan adalah data yang diperoleh dari kuesioner yang dibagikan kepada responden. Pada *game* "Governman" variabel yang diujikan merupakan variabel PE, EE, HM, dan BI.

Tabel 4. 2 Uji Validasi

Rotated Component Matrix[a]

	Component	
	1	2
PE1	.826	.214
PE2	.827	.235
PE3	.860	.193
EE1	.392	.707
EE2	.179	.769
EE3	.193	.836
EE4	.303	.727
HM1	.618	.501
HM2	.825	.319
HM3	.707	.332
BI1	.637	.549
BI2	.685	.490
BI3	.810	.294

Extraction Method: Principal Component Analysis.
Rotation Method: Equamax with Kaiser Normalization.

a. Rotation converged in 3 iterations.

Pada tabel 4.2 hasil uji validasi dari seluruh variabel adalah valid dikarenakan memiliki pola yang serupa dan memiliki nilai diatas 0.4.

4.6.5 Uji Reliabilitas

Uji reliabilitas berfungsi untuk melihat konsistensi dari sebuah data, tingkat konsistensi ini tercermin dari hasil yang diperoleh dan akan tetap sama ketika responden mengulang kuesioner. Pada titik ini diperlukan uji validitas untuk melihat seberapa konsisten suatu data, dengan menggunakan uji validitas untuk melihat variabel mana yang valid untuk uji reliabilitas. Pada tabel 4.2, hasil dari uji validitas semua variabel merupakan variabel yang valid, maka dari itu uji reliabilitas dapat dilakukan dan berikut hasil dari uji reliabilitas.

Tabel 4. 3 Uji Reliabilitas PE

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.902	.903	3

Pada tabel 4.3 variabel PE mempunyai hasil 0.903 pada uji reliabilitas.

Tabel 4. 4 Uji Reliabilitas EE

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.833	.833	4

Pada tabel 4.4 variabel EE mempunyai hasil 0.833 pada uji reliabilitas.

Tabel 4. 5 Uji Reliabilitas HM

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.836	.835	3

Pada tabel 4.5 variabel HM mempunyai hasil 0.835 pada uji reliabilitas.

Tabel 4. 6 Uji Reliabilitas BI

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.872	.873	3

Pada tabel 4.6 variabel BI memiliki hasil 0.873 pada uji reliabilitas.

Dalam menentukan nilai dari uji reliabilitas maka diperlukan keterangan mengenai rentang nilai atau standar nilai yang digunakan untuk menilai suatu variabel. Pada tabel 4.7 merupakan rentang nilai standar untuk uji reliabilitas.

Tabel 4. 7 Rentang Nilai Uji Reliabilitas

Cronbach's Alpha	**Internal Consistency**
$\alpha \geq 0.9$	Excellent
$0.9 > \alpha \geq 0.8$	Good
$0.8 > \alpha \geq 0.7$	Acceptable
$0.7 > \alpha \geq 0.6$	Questionable
$0.6 > \alpha \geq 0.5$	Poor
$0.5 > \alpha$	Unacceptable

Dari rentang nilai tersebut dapat diketahui bahwa variabel PE memiliki hasil *excellent*, lalu variabel EE, HM, BI memiliki hasil *good*. Seperti yang ditunjukan oleh tabel 4.8 berikut.

Tabel 4. 8 Hasil Uji Reliabilitas

Variabel	Cronbach's Alpha	Internal Consistency
PE	0.903	Excellent
EE	0.833	Good
HM	0.835	Good
BI	0.873	Good

4.6.6 Uji Korelasi

Dalam tabel 4.9, hasil yang didapat dari uji korelasi mempunyai hubungan yang bagus dengan memiliki tanda bintang sebagai petunjuk bahwa korelasi antar variabel kuat. Hasil korelasi antara rata-rata variabel PE, EE, HM dan BI diubah menjadi RPE, REE, RHM, dan RBI sehingga memiliki kesimpulan bahwa.

H1 terbukti variabel PE dan BI memiliki korelasi yang kuat dengan nilai korelasi diatas 0.4 dan memiliki tanda bintang.

H2 terbukti variabel EE dan BI memiliki korelasi yang kuat dengan nilai korelasi diatas 0.4 dan memiliki tanda bintang.

H3 terbukti variabel HM dan BI memiliki korelasi yang kuat dengan nilai korelasi diatas 0.4 dan memiliki tanda bintang.

Tabel 4. 9 Hasil Uji Korelasi

Correlations

		RPE	REE	RHM	RBI
RPE	Pearson Correlation	1	.554**	.793**	.764**
	Sig. (2-tailed)		.000	.000	.000
	N	50	50	50	50
REE	Pearson Correlation	.554**	1	.650**	.688**
	Sig. (2-tailed)	.000		.000	.000
	N	50	50	50	50
RHM	Pearson Correlation	.793**	.650**	1	.869**
	Sig. (2-tailed)	.000	.000		.000
	N	50	50	50	50
RBI	Pearson Correlation	.764**	.688**	.869**	1
	Sig. (2-tailed)	.000	.000	.000	
	N	50	50	50	50

**. Correlation is significant at the 0.01 level (2-tailed).