

## BAB 4 PERANCANGAN DAN PENGUJIAN GAME

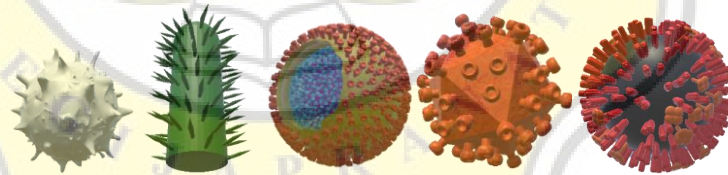
### 4.1 Penedukasian tentang Virus dalam Game “Orbiter Salutis”

Dalam bab ini akan dijelaskan mengenai pemahaman mengenai edukasi kepada masyarakat lewat pembuatan *game*, perancangan *game* serta perancangan aset *game* dalam *game* edukasi virus “Orbiter Salutis”.

#### 4.1.1 Edukasi mengenai Virus di dalam game “Orbiter Salutis”

Perancangan *game* bernama “Orbiter Salutis” yang telah dibuat oleh penulis bertujuan untuk mengedukasi masyarakat Indonesia agar dapat lebih mengerti dan memahami tentang berbagai jenis virus yang sering dijumpai di sekitar masyarakat dalam kehidupan sehari-hari. Hal ini dapat meningkatkan pemahaman dan pengetahuan serta langkah pencegahan yang dapat diantisipasi oleh masyarakat agar dapat melakukan tindakan pencegahan dari terjangkitnya suatu penyakit oleh virus.

Virus yang diambil penulis untuk dimasukkan ke dalam permainan “Orbiter Salutis” merupakan virus yang sering dijumpai dalam lingkungan sehari-hari. Virus ini dapat ditemukan dalam penyakit seperti Covid-19, Rabies, HIV, Hepatitis-B, dan Influenza.

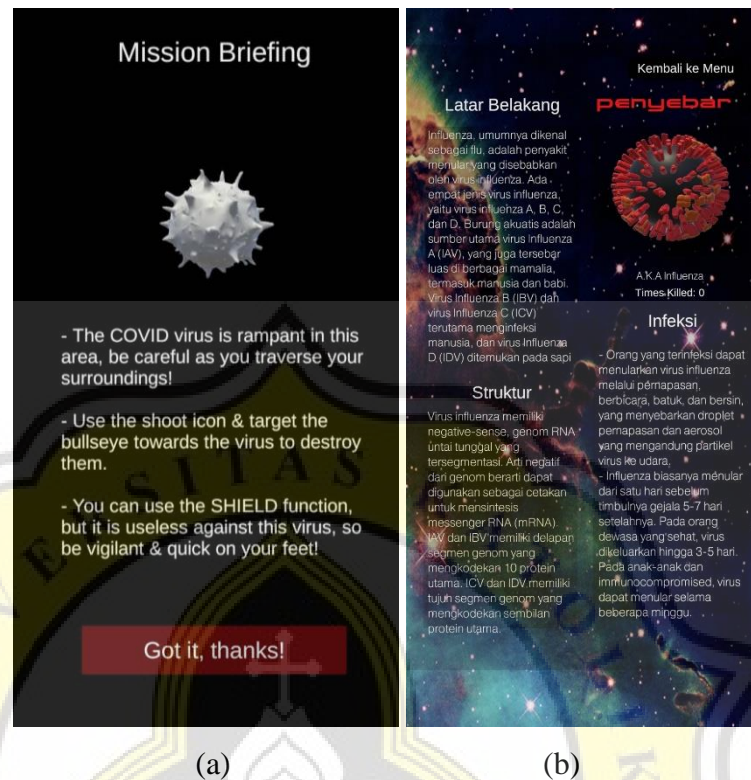


*-The Crown -The Maddening -The Sickness -The Weakened -The Viral*  
(Pemahkota) (Penggila) (Pewabah) (Pelemah) (Penyebar)

*Gambar 4. 1 Jenis Virus yang digunakan dalam game "Orbiter Salutis"*

Kelima jenis virus pilihan ini dikemas ke dalam pembuatan *game* “Orbiter Salutis” yang dalam alurnya pemain harus menembakkan poin laser ke arah musuh yang datang menghampiri pemain serta melindungi diri dari serangan musuh sesuai level yang dimainkan pemain. Penjelasan

lebih lanjut mengenai virus dan penempatan levelnya dijelaskan dalam sub-bab 4.1.3 pada tabel 4.3.



Gambar 4.2 Capture Permainan yang memberikan edukasi virus kepada pemain (a) Intro sebelum bermain dalam level di Game Orbiter Salutis (b) Tampilan menu Help

Sebelum bermain, pemain dipandu dan dijelaskan terlebih dahulu mengenai jenis virus yang akan dimainkan pemain. Hal ini agar pemain tidak hanya asal bermain dan tidak canggung saat bermain. Dalam menu 'help' pemain dapat memahami lebih jauh setiap jenis virus yang dimainkannya. Seperti permainan 'Pokemon Go' dimana pemain menjadi hapa karakter tiap *Pokemon* yang dimilikinya, *Orbiter Salutis* juga mengajak agar para pemain dapat mengingat setiap virus yang dimainkannya beserta detail yang perlu diketahui untuk mencegah penyebaran virus. Secara tidak langsung pemain menjadi hapa karena kebiasaannya memainkan virus game pada "Orbiter Salutis".

#### 4.1.2 Perancangan Game Edukasi Virus "Orbiter Salutis"



Gambar 4. 3 Logo "Orbiter SalutiS"

Penggunaan nama dalam *Game* Edukasi Virus "*Orbiter SalutiS*" diambil dari Bahasa Latin. Kata "*Orbiter*" berarti pengorbit atau pengitar, dan "*SalutiS*" berarti kesahatan atau keselamatan. Sehingga setelah digabung memiliki arti sebagai 'seorang pengitar yang menyelamatkan'. Berikut merupakan penjelasan mengenai *platform* yang digunakan penulis, *genre* yang diusung dalam pembuatan *game* "*Orbiter SalutiS*", *art style* yang dipakai, dan dasar alur permainan yang digunakan dalam permainan "*Orbiter SalutiS*" ini :

1. *Platform*

Dengan pertimbangan penulis, *game* "*Orbiter SalutiS*" dibuat dengan menggunakan platform *Android* karena pangsa pasarnya yang lebih mendominasi dan merupakan *Operating System* yang umum ditemukan dalam kalangan masyarakat Indonesia.

2. *Genre*

*Genre* atau kategori *game* yang digunakan dalam *game* "*Oribter SalutiS*" adalah *First Person Shooter* (FPS) dengan menggunakan teknologi *Augmented Reality* (AR) melalui kamera *smartphone* yang kompatibel dengan *Augmented Reality* (AR). Hal ini dikarenakan seiringnya perkembangan zaman, banyak orang menginginkan *game* yang interaktif dan dapat membuat pemain aktif dalam bermain.

3. *Art Style*

Dengan menggunakan teknologi *Augmented Reality* sebagai fondasi dalam pembuatan *game* "*Orbiter SalutiS*". Gaya dimensi yang digunakan adalah tiga dimensi (3D) agar selaras dengan lingkungan sekitar pemain. Model yang digunakan seperti untuk musuh dan

lingkungan *virtual* akan menggunakan gaya yang menyondong ke arah kartun.

#### 4. Dasar Alur Permainan

Pada *game* “*Orbiter Salutis*” terdapat lima level (terdapat dalam menu “*Start Game*”) yang dapat dimainkan oleh pemain dan dalam setiap level terdapat kumpulan musuh yang berbeda, secara bentuk, perlakuan, jenis dan mekanisme terhadap pemain. Sebelumnya masuk kedalam permainan, *game* akan bertanya (kepada pemain baru) ‘apakah pemain ingin melihat Tutorial bermain?’ dan jika menjawab ‘Iya’ maka akan dibawa ke bagian Tutorial dimana pemain dapat memahami mekanisme permainan dan setelah selesai pemain akan dibawa kembali untuk memulai level pertama. Jika pemain menjawab ‘Tidak’ maka pemain akan langsung diarahkan untuk bermain pada level pertama. Setiap sebelum bermain dalam level terdapat “*Mission Briefing*” yang berisi informasi tentang musuh yang akan ditemui dan apa yang harus dilakukan pemain agar dapat membasmi mereka secara optimal. Setelah mengerti, pemain akan diminta untuk menempatkan sebuah “*Portal*” pada lantai yang sudah terpindai oleh kamera pemain (berbentuk jala persegi-persegi) dengan cara mengetuknya sesuai lokasi yang diinginkan pemain, maka permainan baru dimulai. Dalam “*Portal*” setiap level dapat dijalani masuk oleh pemain dan akan terdapat berbagai lingkungan virtual didalamnya;

- Level 1 akan mendapati sebuah lingkungan garasi kendaraan dengan berbagai macam peralatan yang terkontaminasi dengan virus Corona. Level ini lebih terarah pada pemula yang baru bermain agar dapat terbiasa dengan teknik mainnya.
- Level 2 terdapat lingkungan rumah yang terdiri dari ruang tengah, kamar tidur serta lorong kecil. Pada level ini jumlah jenis musuh sudah meningkat menjadi dua agar kesulitan permainan lebih menarik pemain untuk meningkatkan performanya.

- Level 3 menempatkan pemain dalam lingkungan sebuah pesawat luar angkasa yang futuristik. Pemain akan disajikan tiga jenis musuh dalam level ini dan untuk level-level selanjutnya agar pemain selalu merasa tertantang untuk membasmi virus secara optimal.
- Level 4 akan ada lingkungan virtual dimana terdapat sebuah gubuk peralatan dan sebelahny ada sebuah truk yang ditinggalkan. Akan dikenalkan sebuah jenis virus baru dalam level ini dan menggantikan salah satu virus yang terdapat pada level sebelumnya.
- Level 5 adalah level terakhir dalam game “*Orbiter Salutis*”, dimana pemain akan ditempatkan dalam sebuah lingkungan virtual berupa sebuah ruang altar terbuka terbuat dari bebatuan dan dapat melihat bulan purnama diatas pemain. Level ini juga memperkenalkan jenis virus baru yang menggantikan salah satu virus yang terdapat pada level sebelumnya.

Untuk membuka level-level selanjutnya, pemain diharuskan untuk membasmi sejumlah virus yang terdapat dalam setiap level, terutama virus-virus yang baru dikenalkan kepada pemain (dapat dilihat jumlah setiap virus yang telah dibasmi oleh pemain pada menu “*Help*” di *Main Menu*). Karena dalam permainan tidak ada ‘*Win Condition*’ alias mode permainan ‘*Endless*’ (tidak terdapat batasan waktu atau target), maka pemain harus ‘mati’ atau *Game Over* untuk menyelesaikan permainan dan akan ditunjukkan berbagai *Highscore* yang telah diraih pada level tersebut dan skor yang telah diperoleh pemain saat bermain.

#### **4.1.3 Perancangan Aset dalam Game Edukasi Virus “*Orbiter Salutis*”**

Perancangan aset-aset dua dimensi (2D) menggunakan *software Adobe Illustrator* serta *Adobe Photoshop*. Kedua software tersebut pantas digunakan dalam pembuatan aset seperti *User Interface* (UI), logo *game*, dan hal-hal lain yang bersangkutan dengan elemen grafis karena utilitas yang sangat memfasilitasi. Gaya yang diutamakan dalam pembuatan aset

2D adalah minimalis serta futuristik untuk menyesuaikan tema judul yang diungkitkan penulis.

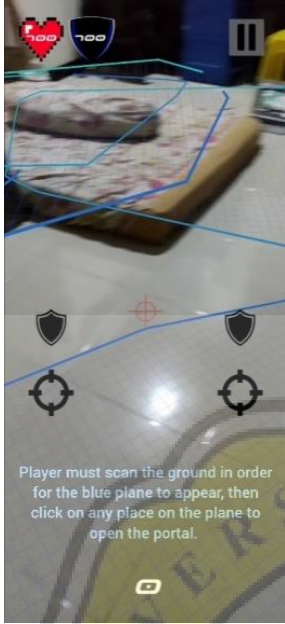
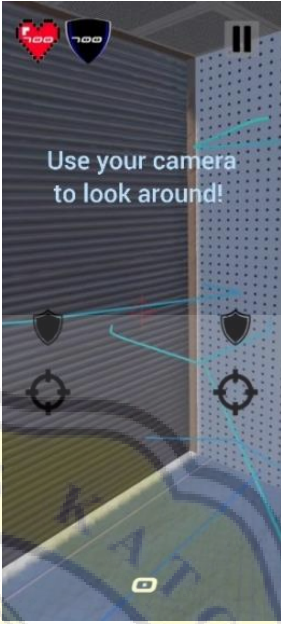

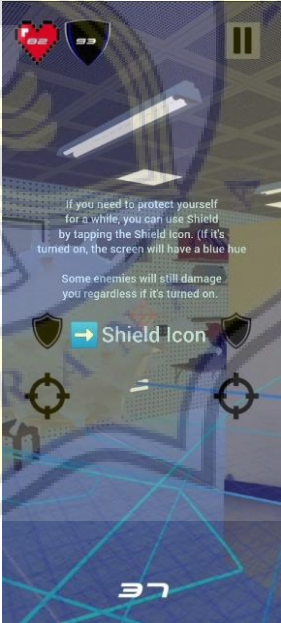
Tabel 4. 1 Tabel elemen UI dalam game "Orbiter Salutis"

Icon/UI	Penjelasan	Icon/UI	Penjelasan
	Untuk memulai permainan pada menu 'Level Select'. Terdapat di Main Menu.		Untuk menayangkan halaman 'Informasi'. Terdapat di Main Menu.
	Untuk memindahkan pemain pada menu 'Option'. Ada di Main Menu.		Digunakan untuk mematikan aplikasi atau game. Terdapat di Main Menu.
 Icon "Pause"	Untuk menghentikan game sementara. Terdapat saat bermain level.	 Icon "Resume"	Untuk mengembalikan pemain dalam game. Terdapat saat bermain level.
	Icon "Home" digunakan untuk mengembalikan pemain pada Main Menu. Terdapat saat bermain level.		Icon "Shoot" digunakan untuk menembak dan mematikan musuh. Terdapat saat bermain level.
	Icon "Shield" digunakan untuk mengaktifasi / mematikan fungsi Shield (perisai dari serangan musuh). Terdapat saat bermain level.		
	Bendera Indonesia digunakan pada menu "Option" untuk menggantikan sistem Bahasa.		Bendera U.K digunakan pada menu "Option" untuk menggantikan sistem Bahasa.

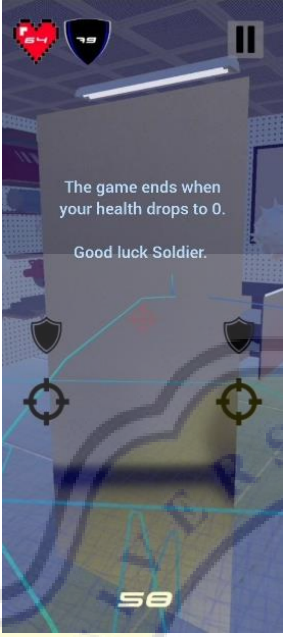
Selain terdapatnya *icon* yang difungsikan sebagai tombol dalam UI/UX *game*, terdapat juga beberapa gambar-gambar lain seperti gambar tutorial yang digunakan untuk mengedukasi pemain tentang cara bermain dalam *level*, gambar “*Mission Briefing*” yang menunjukkan musuh yang terdapat dalam *level* serta beberapa petunjuk agar pemain mengerti apa yang harus mereka persiapkan sebelum bermain dalam *level*, serta gambar *background* luar angkasa untuk menyesuaikan *backstory* awal *game* “*Orbiter Salutis*”, yaitu seseorang luar angkasa yang datang ke Bumi untuk mencoba membasmi berbagai virus yang masih berkeliaran dan merombak kehidupan di Bumi.

Tabel 4. 2 Tabel Tutorial pada Gem Orbiter Salutis

Gambar	Penjelasan	Gambar	Penjelasan
 <p>Gambar <i>background</i></p>	<p>Digunakan selain <i>Main Menu</i>, bernuansa luar angkasa. Dapat ditemukan dalam menu <i>Help</i>, <i>Options</i>, <i>Level Select</i> dan <i>Tutorial</i>.</p>		<p>Tampilan <i>Main Menu</i> (Menu Utama) pada <i>game</i> “<i>Orbiter Salutis</i>”. <i>Background</i> adalah lingkungan luar angkasa yang tiga dimensi.</p>


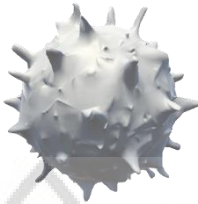
Gambar	Penjelasan	Gambar	Penjelasan
<p style="text-align: center;"><b>Tutorial 1.</b></p> 	<p>Pemain diharuskan untuk memindai lantai disekitarnya terlebih dahulu agar dapat meletakkan <i>portal</i> untuk memulai permainan.</p>	<p style="text-align: center;"><b>Tutorial 2.</b></p> 	<p>Pemain diarahkan untuk menggerakkan perangkatnya untuk memasuki arena bermain serta untuk melacak musuh.</p>
<p style="text-align: center;"><b>Tutorial 3</b></p> 	<p>Pemain dapat membidik musuh dengan bantuan gambar sasaran (<i>Bullseye</i>) di tengah-tengah layar serta menembak dengan menekan tombol dengan <i>icon</i> yang tertera pada gambar.</p>	<p style="text-align: center;"><b>Tutorial 4</b></p> 	<p>Pemain dapat memakai fungsi '<i>Shield</i>' untuk melindungi musuh. Saat <i>Shield</i> mencapai nilai 0, maka fungsi tersebut menjadi non-aktif dan tidak dapat digunakan dalam permainan atau <i>level</i> tersebut lagi.</p>





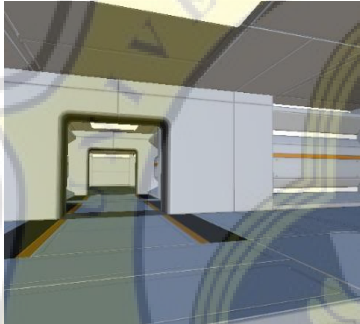
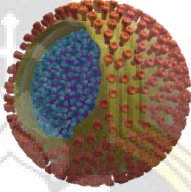

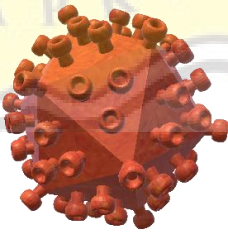
Gambar	Penjelasan	Gambar	Penjelasan
<p style="text-align: center;"><b>Tutorial 5</b></p> 	<p>Memberitahu pemain bahwa permainan akan selesai atau <i>Game Over</i> saat musuh berhasil menurunkan <i>Health</i> atau nyawa pemain (diindikasikan dengan gambar hati pada kiri atas) sampai nol.</p>		


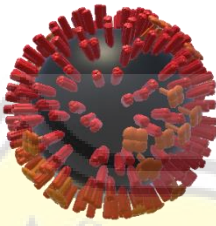
Untuk aset tiga dimensi (3D) seperti musuh serta lingkungan virtual dibuat dengan program pembuat model 3D seperti Blender dan ada juga beberapa yang diambil secara gratis dengan 'free license' melalui *Asset Store* pada aplikasi *Unity* sendiri (<https://assetstore.unity.com/>).

Tabel 4. 3 Model 3D lingkungan dan musuh

Preview lingkungan virtual 3D	Preview model 3D musuh	Penjelasan musuh pada tiap level
 <p data-bbox="379 748 692 786">Level 1 <i>“Garage Spoil”</i></p>	 <p data-bbox="767 685 943 763"><i>“The Crown”</i> (Pemahkota)</p>	<p data-bbox="991 383 1361 833">Morfologi musuh ini disesuaikan dengan salah satu bentuk virus dari keluarga coronavirus. Karakteristik yang sering dijumpai pada badan virus ini adalah berisinya tonjolan-tonjolan pada permukaan virus dan ujung-ujungnya yang berbentuk mahkota.</p>



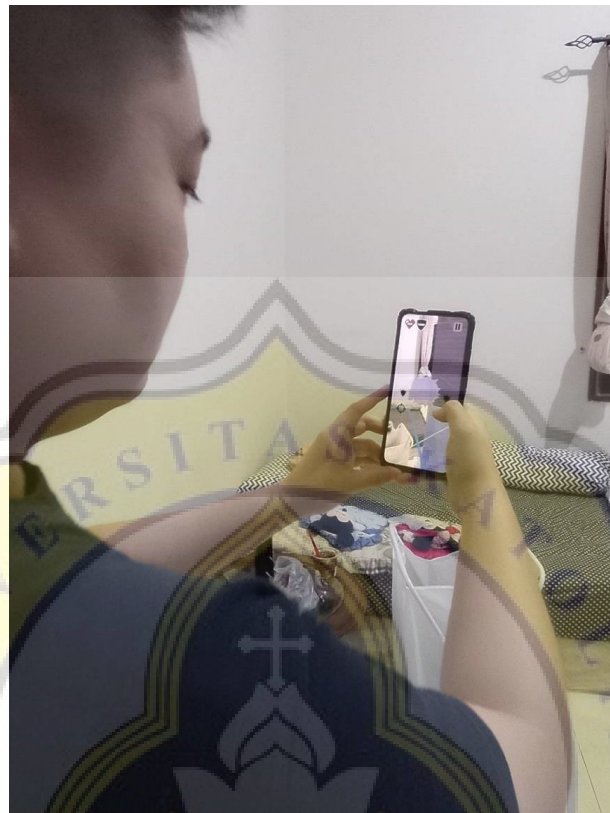
Preview lingkungan virtual 3D	Preview model 3D musuh	Penjelasan musuh pada tiap level
 <p data-bbox="379 712 708 748">Level 2 “<i>Home Invaded</i>”</p>	 <p data-bbox="767 689 991 770">“<i>The Maddening</i>” (Penggila)</p>	<p data-bbox="1023 315 1353 815">Morfologi musuh ini disesuaikan mengikuti bentuk virus rabies yang sering ditemukan dalam hewan dan manusia. Karakteristik fisik dari musuh yang satu ini adalah bentuknya yang lonjong seperti tabung dan <i>glycoprotein</i>, duri tersebar pada badan virus.</p>
 <p data-bbox="363 1218 724 1254">Level 3 “<i>Spaceship Susse'd</i>”</p>	 <p data-bbox="783 1151 979 1232">“<i>The Sickness</i>” (Pewabah)</p>	<p data-bbox="1023 822 1353 1321">Morfologi musuh ini disesuaikan dengan bentuk virus hepatitis B (HBV). Karakteristik yang dapat ditemukan dalam virus ini adalah bagian luarnya terdiri dari selubung protein dan <i>lipid</i> guna untuk melindungi bagian dalam virus yang berisi asam nukleat.</p>
 <p data-bbox="389 1783 702 1818">Level 4 “<i>Surrounde'd</i>”</p>	 <p data-bbox="772 1724 991 1805">“<i>The Weakened</i>” (Pelemah)</p>	<p data-bbox="1023 1328 1353 1953">Morfologi musuh ini disesuaikan dengan bentuk <i>Human Immunodeficiency Virus</i> (HIV). Karakteristik yang dapat ditemukan dalam virus ini adalah bentuknya yang bundar dikelilingi oleh berbagai <i>glycoprotein</i> dan selubung viral melindungi bagian dalam virus yang terdapat kapsida, RNA serta berbagai enzim</p>

Preview lingkungan virtual 3D	Preview model 3D musuh	Penjelasan musuh pada tiap level
 <p data-bbox="427 853 671 927">Level 5 “<i>Dungeon Destruction</i>”</p>	 <p data-bbox="807 786 954 860">“<i>The Viral</i>” (Penyebar)</p>	<p data-bbox="1023 374 1359 1034">Morfologi dari musuh ini disesuaikan dengan bentuk virus Influenza. Karakteristik yang dapat ditemukan dalam virus ini adalah bagian luar yang dikelilingi oleh <i>Neuraminidase</i> serta <i>Hemagglutinin</i> yang berfungsi untuk melindungi virus dari berbagai serangan luar, protein matriks serta <i>lipid bilayer</i> sebagai pelindung struktur dalam yang berisi RNA virus.</p>

#### 4.2 Penggunaan dan Pemrograman Kombinasi Sistem FPS (*First Person Shooter*) dan AR (*Augmented Reality*) dalam Game “*Orbiter Salutis*”

Kombinasi sistem permainan dari genre *First Person Shooter* dengan teknologi *Augmented Reality* bukanlah hal yang baru dalam dunia *video games*, tetapi masih merupakan salah satu kombinasi yang belum tergalai potensinya secara penuh. Sebagai referensi, salah satu *game* AR-FPS yang sempat mendapat perhatian secara internasional adalah *game* yang dirilis oleh Proxy42 Inc. untuk Android dan iOS bernama *Father.io*, dimana para pemain dilengkapi oleh berbagai macam senjata api dan perlengkapan perang untuk bertarung dengan pemain lain dalam aksi *warfare* dengan latar belakang dunia nyata, menggunakan teknologi *Augmented Reality* serta sebuah komponen laser yang diletakan pada belakang smartphone bernama *Inceptor* sebagai penanda pemain menembak dan ditembak. Selain itu juga terdapat sebuah *game* yang dirilis pada iOS oleh Sebastian Armand bernama *W.AR – Augmented Reality*, dimana pemain dilengkapi oleh senjata api untuk membasmi berbagai musuh

zombie yang mengelilingi pemain, dengan latar dunia nyata yang ditambahi berbagai macam aset dunia virtual untuk membantu pemain.



*Gambar 4. 4 Pemain memainkan sebuah level*

Hal yang membedakan atau fitur yang ditambahkan oleh penulis dalam merancang game Orbiter Salutis adalah adanya penambahan dunia virtual kecil melalui sebuah pintu portal, dimana letaknya menyesuaikan lokasi riil pada dunia nyata yang ditetapkan oleh pemain seperti yang ditunjukkan pada gambar 4.4. Dengan hal tersebut, melalui layar perangkat smartphone, pemain dapat melihat sebuah lingkungan lain dari dunia nyata. Mekanisme permainan dalam Orbiter Salutis tidaklah kompleks, bahkan terdapat tutorial yang menjelaskan cara bermain untuk pemula yang baru memainkan pertama kali sebelum level pertama dimulai. Pemain membidik musuh yang ingin dibasmi melalui *bullseye* ditengah layar lalu menekan ikon tembak (dibawah ikon perisai) untuk menembak virus agar menghilang. Fitur perisai digunakan untuk melindungi player dari berbagai macam musuh virus dan membasminya saat mendekati pemain jika ia merasa tidak mungkin untuk menembak seluruh musuh dalam

waktu yang singkat, tetapi ada juga beberapa musuh yang mengelabui perlindungan perisai yang membuat pemain harus lebih waspada dalam mengatur perisainya dengan musuh-musuh tersebut saat bermain.

Setelah semua aset dan perancangan *game* sudah selesai dan siap digunakan, maka *development* masuk pada fase pemrograman. *Game engine* yang digunakan adalah *Unity* dengan versi *Editor* 2020.3.28f1, bahasa pemrograman yang digunakan adalah *C#* dengan menggunakan software IDE yaitu *Microsoft Visual Studio* 2019. Juga terdapat kode *PHP* yang diletakan pada sebuah database website untuk pemindaian, membentuk, menghapus dan meng-*update* data pemain. Berikut adalah beberapa *script* yang dibentuk untuk keberlangsungan *game Orbiter Salutis*:

#### 4.2.1 Pada menu Login/Register

##### a. Login Pemain

Sistem login terdapat beberapa variabel yang digunakan untuk memverifikasi kredensial pemain, antara lain username dan password. Untuk memverifikasi data tersebut, maka data yang telah diketik akan dikirimkan menggunakan fungsi *WWWForm* dan *string* URL kepada sebuah file *PHP* pada *database website* yang mengatur segala pengoperasian *back-end*.

```

1. public class ServerLogin : MonoBehaviour
2. {
3.     [SerializeField] Text user;
4.     [Space]
5.     [SerializeField] InputField username;
6.     [SerializeField] InputField password;
7.
8.     [SerializeField] Text errorMessages;
9.
10.    [SerializeField] Button loginButton;
11.
12.    public StateManager stateManager;
13.    public GameObject StateManagerObj;
14.
15.
16.    string url = "https://projectsteps.000webhostapp.com/Build/orbsalut_mp/SUFoL.php";
17.
18.    WWWForm form;
19.
20.    private void Start()
21.    {
22.        InitializeCred();
23.        StateManagerObj = GameObject.FindWithTag("StateManager");
24.        if (StateManagerObj != null)
25.            stateManager = StateManagerObj.GetComponent<StateManager>();
26.    }
27.
28.    public void OnLoginButtonClicked()
29.    {
30.        loginButton.interactable = false;
31.        StartCoroutine(Login());
32.    }
33.
34.    IEnumerator Login()

```

*Gambar 4. 5 Script Login*

```

1.    {
2.        form = new WWWForm();
3.
4.        form.AddField("username", username.text);
5.        form.AddField("password", password.text);
6.
7.        WWW w = new WWW(url, form);
8.        yield return w;
9.
10.       if (w.error != null)
11.       {
12.           errorMessages.text = "404 not found!";
13.           Debug.Log("<color=red>" + w.text + "</color>");
14.       }
15.       else
16.       {
17.           if (w.isDone)
18.           {
19.               if (w.text.Contains("error"))
20.               {
21.                   errorMessages.text = "Invalid username or
22.                   password!";
23.                   Debug.Log("<color=red>" + w.text +
24.                   "</color>");
25.               }
26.               else
27.               {
28.                   errorMessages.text = "";
29.                   //user.text = "Success!" + "(" +
30.                   Debug.Log("<color=green>" + w.text +
31.                   "PlayerPrefs.SetString("playerName",
32.                   stateManager.LoadOnLogin();
33.                   SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 2);
34.               }
35.           }
36.           loginButton.interactable = true;
37.
38.           w.Dispose();
39.       }
40.    }

```

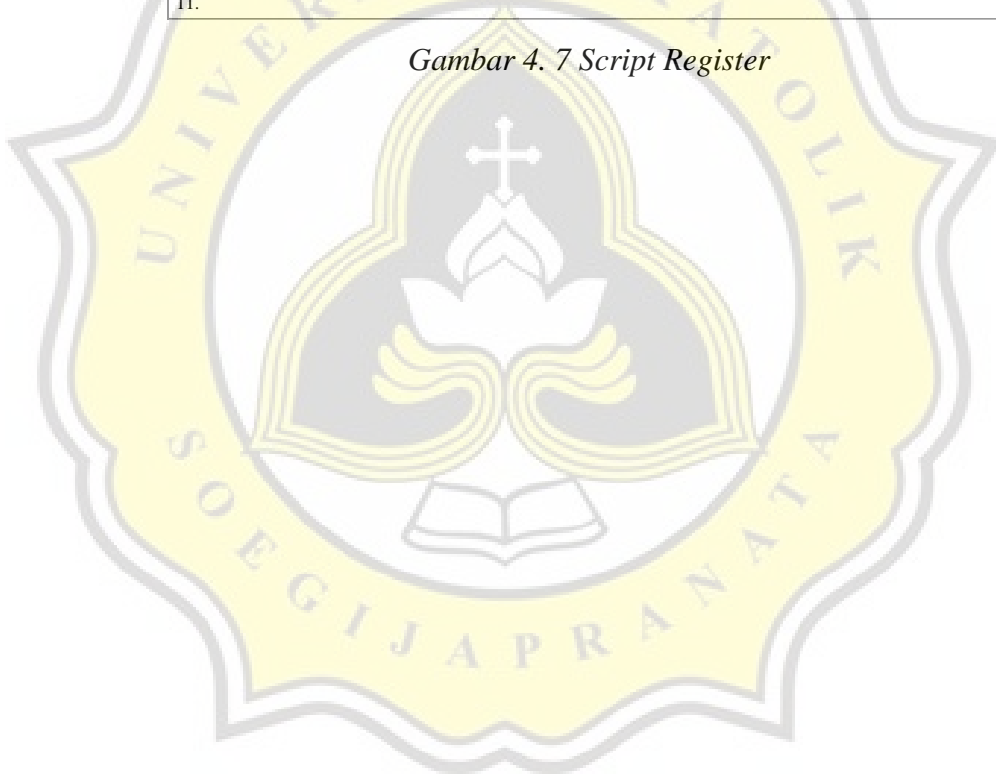
*Gambar 4. 6 Script Login (lanjutan)*

b. *Register* data pemain baru

Sistem registrasi pemain baru hampir sama dengan sistem *login*, seperti adanya variabel *username* dan *password*, yang membedakan adalah sistem ini merupakan pembuatan data baru ke dalam *database*. Terdapat juga beberapa pengecekan, seperti apakah pemain pernah menggunakan *username* yang sama atau konfirmasi *password* tidak cocok.

```
1. public class ServerRegister : MonoBehaviour
2. {
3.     [SerializeField] Text user;
4.     [SerializeField] InputField username;
5.     [SerializeField] InputField password;
6.     [SerializeField] InputField conf_password;
7.     [SerializeField] Text errorMessages;
8.
9.     [SerializeField] Button loginButton;
10.
11. }
```

Gambar 4. 7 Script Register





```

1.         string url = "https://projectsteps.000webhostapp.com/Build/orbsalut_mp/insDataUserUts.php";
2.         WWWForm form;
3.
4.     private void Start()
5.     {
6.         InitializeCred();
7.     }
8.
9.     public void OnLoginButtonClicked()
10.    {
11.        loginButton.interactable = false;
12.        if (username.text != "" && password.text != "")
13.            StartCoroutine(Login());
14.        else
15.        {
16.            errorMessages.text = "Please enter Username/Password!";
17.            loginButton.interactable = true;
18.        }
19.    }
20.
21.
22.    IEnumerator Login()
23.    {
24.        form = new WWWForm();
25.
26.        form.AddField("nama", username.text);
27.        form.AddField("password", password.text);
28.
29.        WWW w = new WWW(url, form);
30.        yield return w;
31.
32.        if (password.ToString() != conf_password.ToString())
33.        {
34.            errorMessages.text = "Password does not match!";
35.            Debug.Log("<color=red>" + w.text + "</color>");
36.        }
37.
38.        if (w.error != null)
39.        {
40.            errorMessages.text = "404 not found!";
41.            Debug.Log("<color=red>" + w.text + "</color>");
42.        }
43.        else
44.        {
45.            if (w.isDone)
46.            {
47.                if (w.text.Contains("error"))
48.                {
49.                    errorMessages.text = "User already
50.                    created!";
51.                    Debug.Log("<color=red>" + w.text +
52.                    "</color>");
53.                }
54.                else
55.                {
56.                    errorMessages.text = "";
57.                    Debug.Log("<color=green>" + w.text +
58.                    "</color>");
59.                    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex - 1);
60.                }
61.            }
62.            w.Dispose();
63.        }
64.    }
65. }

```

Gambar 4. 8 Script Register (lanjutan)

#### 4.2.2 Pada menu game

##### a. Script untuk button UI pada game (Menuing)

Script ini terdapat beberapa fungsi yang tujuannya adalah menggabungkan berbagai aspek menu sehingga tidak diperlukannya banyak script untuk berpindah ke berbagai scene dalam game.

```

1. public class MenuScript : MonoBehaviour
2. {
3.     public StateManager stateManager;
4.     public GameObject StateManagerObj, tutorialCanvas;
5.     public bool isStateManagerActive;
6.
7.     private void Start()
8.     {
9.         if (isStateManagerActive)
10.        {
11.            StateManagerObj = GameObject.FindWithTag("StateManager");
12.            if (StateManagerObj != null) {
13.                stateManager = StateManagerObj.GetComponent<StateManager>();
14.                stateManager.LoadFile();
15.            }
16.        }
17.    }
18.
19.    public void RegisterButton()
20.    {
21.        SceneManager.LoadScene(1);
22.    }
23.
24.    public void MainMenu()
25.    {
26.        SceneManager.LoadScene(2);
27.    }
28.
29.    public void StartGame2D()
30.    {
31.        SceneManager.LoadScene("Level Select (2D)");
32.    }
33.
34.    public void HelpMenu()
35.    {
36.        SceneManager.LoadScene("Help");
37.    }
38.
39.
40.    public void OptionsMenu()
41.    {
42.        SceneManager.LoadScene("Options");
43.    }
44.
45.    public void LoadGarage()
46.    {
47.        if (stateManager.myPlayer.isTutorialDone == false) {
48.            tutorialCanvas.SetActive(true);
49.        }
50.        else {
51.            SceneManager.LoadScene("Demo");
52.        }
53.    }
54.
55.    public void LoadHouse()
56.    {
57.        SceneManager.LoadScene("Demo 1");
58.    }
59.

```

*Gambar 4. 9 Script untuk pindah Menu*

```

1. public void LoadSpace()
2. {
3.     SceneManager.LoadScene("Demo 2");
4. }
5.
6. public void LoadDead()
7. {
8.     SceneManager.LoadScene("Demo 3");
9. }
10.
11. public void LoadDungeon()
12. {
13.     SceneManager.LoadScene("Demo 4");
14. }
15.
16. public void YesTutorial()
17. {
18.     if (stateManager.myPlayer.isTutorialDone == false)
19.     {
20.         stateManager.myPlayer.isTutorialDone = true;
21.         stateManager.SaveData();
22.         SceneManager.LoadScene("Tutorial");
23.     }
24. }
25.
26. public void NoTutorial()
27. {
28.     if (stateManager.myPlayer.isTutorialDone == false)
29.     {
30.         stateManager.myPlayer.isTutorialDone = true;
31.         stateManager.SaveData();
32.         SceneManager.LoadScene("Demo");
33.     }
34. }
35.
36. public void QuitGame()
37. {
38.     Application.Quit();
39. }
40. }

```

Gambar 4. 10 Script untuk pindah Menu (lanjutan)

#### b. Sistem pemilihan level

Pemilihan *level* dalam *scene Level Select* dilakukan dengan sistem korsel, dimana gambar *preview level* dan teks yang mendampinginya akan muncul secara bergantian dengan gerakan menggeser.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class SlideControl : MonoBehaviour
6. {
7.     public GameObject LeftSlide, CenterSlide, RightSlide;
8.     public GameObject Object1, Object2, Object3, Object4, Object5;
9.     public GameObject LeftButton, RightButton;
10.    public GameObject StartGame;
11.    public bool isTutorial;
12.    public int count = 0;
13.    bool check = true;
14.    bool check1 = false;
15.    bool check2 = false;
16.    bool check3 = false;
17.    bool check4 = false;
18.
19.    void Start()

```

Gambar 4. 11 Script pemilihan level

```

1.  {
2.      LeftButton.SetActive(false);
3.      if (isTutorial)
4.          StartGame.SetActive(false);
5.  }
6.
7.  void Update()
8.  {
9.      if (chck == true)
10.     {
11.         Object1.transform.position = Vector3.Lerp(Object1.transform.position,
CenterSlide.transform.position, 15f * Time.deltaTime);
12.         Object2.transform.position = Vector3.Lerp(Object2.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
13.         Object3.transform.position = Vector3.Lerp(Object3.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
14.         Object4.transform.position = Vector3.Lerp(Object4.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
15.         Object5.transform.position = Vector3.Lerp(Object5.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
16.     }
17.     else if (chck1 == true)
18.     {
19.         Object1.transform.position = Vector3.Lerp(Object1.transform.position, LeftSlide.transform.position,
15f * Time.deltaTime);
20.         Object2.transform.position = Vector3.Lerp(Object2.transform.position,
CenterSlide.transform.position, 15f * Time.deltaTime);
21.         Object3.transform.position = Vector3.Lerp(Object3.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
22.         Object4.transform.position = Vector3.Lerp(Object4.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
23.         Object5.transform.position = Vector3.Lerp(Object5.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
24.     }
25.     }
26.     else if (chck2 == true)
27.     {
28.         Object2.transform.position = Vector3.Lerp(Object2.transform.position, LeftSlide.transform.position,
15f * Time.deltaTime);
29.         Object3.transform.position = Vector3.Lerp(Object3.transform.position,
CenterSlide.transform.position, 15f * Time.deltaTime);
30.         Object4.transform.position = Vector3.Lerp(Object4.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
31.     }
32.     else if (chck3 == true)
33.     {
34.         Object3.transform.position = Vector3.Lerp(Object3.transform.position, LeftSlide.transform.position,
15f * Time.deltaTime);
35.         Object4.transform.position = Vector3.Lerp(Object4.transform.position,
CenterSlide.transform.position, 15f * Time.deltaTime);
36.         Object5.transform.position = Vector3.Lerp(Object5.transform.position, RightSlide.transform.position,
15f * Time.deltaTime);
37.     } else if (chck4 == true)
38.     {
39.         Object4.transform.position = Vector3.Lerp(Object4.transform.position, LeftSlide.transform.position,
15f * Time.deltaTime);
40.         Object5.transform.position = Vector3.Lerp(Object5.transform.position,
CenterSlide.transform.position, 15f * Time.deltaTime);
41.     }
42. }
43.

```

Gambar 4. 12 Script pemilihan level (lanjutan)

```
1. public void clickRight()
2. {
3.     count++;
4.     if (count == 1)
5.     {
6.         LeftButton.SetActive(true);
7.         chck = false;
8.         chck1 = true;
9.         chck2 = false;
10.        chck3 = false;
11.        chck4 = false;
12.    }
13.    else if (count == 2)
14.    {
15.        chck = false;
16.        chck1 = false;
17.        chck2 = true;
18.        chck3 = false;
19.        chck4 = false;
20.    }
21.    else if (count == 3)
22.    {
23.        chck = false;
24.        chck1 = false;
25.        chck2 = false;
26.        chck3 = true;
27.        chck4 = false;
28.    }
29.    else
30.    {
31.        count = 4;
32.    }
33. }
34. public void clickLeft()
35. {
36.     count--;
37.     if (count == 0)
38.     {
39.         LeftButton.SetActive(false);
40.         chck = true;
41.         chck1 = false;
42.         chck2 = false;
43.         chck3 = false;
44.         chck4 = false;
45.     }
46.     if (count == 1)
47.     {
48.         chck = false;
49.         chck1 = true;
50.         chck2 = false;
51.         chck3 = false;
52.         chck4 = false;
53.     }
54.     else if (count == 2)
55.     {
56.         chck = false;
57.         chck1 = false;
58.         chck2 = true;
59.         chck3 = false;
60.         chck4 = false;
61.     }
62.     else
63.     {
64.         count = 0;
65.     }
66. }
```

Gambar 4. 13 Script pemilihan level (lanjutan)

### c. State Manager

Yaitu sistem kode yang mengurus data *player* (seperti skor, *level* yang dibuka) yang sukses *login* agar dapat dibaca maupun ditulis ulang oleh aplikasi sesuai kondisi.

```

1.  public class StateManager : MonoBehaviour
2.  {
3.      public bool isLevelSelect, isEndRank, isHelpPage, isMainMenu;
4.      public GameObject[] buttons;
5.      public GameObject[] buttonText;
6.      public Text[] text;
7.      private string path = "";
8.      private string persistentPath = "";
9.
10.     [System.Serializable]
11.     public class Player
12.     {
13.         public string username, defaultLanguage;
14.         public int level1enemies, level2enemies, level3enemies, level4enemies, level5enemies;
15.         public bool isTutorialDone;
16.
17.         public Player(string username, int level1enemies, int level2enemies, int level3enemies, int
18.         level4enemies, int level5enemies, bool isTutorialDone, string defaultLanguage)
19.         {
20.             this.username = username;
21.             this.level1enemies = level1enemies;
22.             this.level2enemies = level2enemies;
23.             this.level3enemies = level3enemies;
24.             this.level4enemies = level4enemies;
25.             this.level5enemies = level5enemies;
26.             this.isTutorialDone = isTutorialDone;
27.             this.defaultLanguage = defaultLanguage;
28.         }
29.     }
30.     public Player myPlayer = new Player("null", 0, 0, 0, 0, 0, false, "English");
31.
32.     public void CreatePlayerData()
33.     {
34.         myPlayer = new Player(PlayerPrefs.GetString("playerName"), 0, 0, 0, 0, 0, false, "English");
35.     }
36.
37.     public void SetPaths()
38.     {
39.         path = Application.dataPath + Path.AltDirectorySeparatorChar + PlayerPrefs.GetString("playerName") +
40.         ".json";
41.         persistentPath = Application.persistentDataPath + Path.AltDirectorySeparatorChar +
42.         PlayerPrefs.GetString("playerName") + ".json";
43.     }
44.     public void SaveData()
45.     {
46.         string savePath = persistentPath;
47.         Debug.Log("Saving Data at " + savePath);
48.         string json = JsonUtility.ToJson(myPlayer);
49.         Debug.Log(json);
50.
51.         using StreamWriter writer = new StreamWriter(savePath);
52.         writer.Write(json);
53.     }
54.     public void LoadFile()
55.     {
56.
57.         using StreamReader reader = new StreamReader(persistentPath);
58.         string json = reader.ReadToEnd();
59.
60.         myPlayer = JsonUtility.FromJson<Player>(json);
61.     }

```

Gambar 4. 14 Script StateManager

```

1.  public void LoadOnLogin()
2.  {
3.      SetPaths();
4.      if (!File.Exists(path))
5.      {
6.          CreatePlayerData();
7.          SaveData();
8.      }
9.      LoadFile();
10. }
11. public void LoadDataLevel()
12. {
13.
14.     using StreamReader reader = new StreamReader(persistentPath);
15.     string json = reader.ReadToEnd();
16.
17.     myPlayer = JsonUtility.FromJson<Player>(json);
18.
19.     if (myPlayer.level1enemies < 15)
20.     {
21.         buttons[0].GetComponent<Image>().color = new Color32(0, 0, 0, 255);
22.         buttonText[0].GetComponent<Text>().text = "Locked";
23.         buttonText[0].GetComponent<Text>().color = new Color32(255, 255, 255, 255);
24.         buttons[0].GetComponent<Button>().enabled = false;
25.     }
26.
27.     if (myPlayer.level2enemies < 15)
28.     {
29.         buttons[1].GetComponent<Image>().color = new Color32(0, 0, 0, 255);
30.         buttonText[1].GetComponent<Text>().text = "Locked";
31.         buttonText[1].GetComponent<Text>().color = new Color32(255, 255, 255, 255);
32.         buttons[1].GetComponent<Button>().enabled = false;
33.     }
34.
35.     if (myPlayer.level3enemies < 15)
36.     {
37.         buttons[2].GetComponent<Image>().color = new Color32(0, 0, 0, 255);
38.         buttonText[2].GetComponent<Text>().text = "Locked";
39.         buttonText[2].GetComponent<Text>().color = new Color32(255, 255, 255, 255);
40.         buttons[2].GetComponent<Button>().enabled = false;
41.     }
42.
43.     if (myPlayer.level4enemies < 15)
44.     {
45.         buttons[3].GetComponent<Image>().color = new Color32(0, 0, 0, 255);
46.         buttonText[3].GetComponent<Text>().text = "Locked";
47.         buttonText[3].GetComponent<Text>().color = new Color32(255, 255, 255, 255);
48.         buttons[3].GetComponent<Button>().enabled = false;
49.     }
50. }
51.
52. public void LoadKills()
53. {
54.     using StreamReader reader = new StreamReader(persistentPath);
55.     string json = reader.ReadToEnd();
56.
57.     myPlayer = JsonUtility.FromJson<Player>(json);
58.
59.     text[0].text = "Times Killed: " + myPlayer.level1enemies.ToString();
60.     text[1].text = "Times Killed: " + myPlayer.level2enemies.ToString();
61.     text[2].text = "Times Killed: " + myPlayer.level3enemies.ToString();
62.     text[3].text = "Times Killed: " + myPlayer.level4enemies.ToString();
63.     text[4].text = "Times Killed: " + myPlayer.level5enemies.ToString();
64.     Debug.Log("Loaded Kills");
65. }

```

*Gambar 4. 15 Script StateManager (lanjutan)*

```

1. void Start()
2. {
3.     if (isLevelSelect)
4.     {
5.         SetPaths();
6.         LoadDataLevel();
7.     }
8.
9.     if (isHelpPage)
10.    {
11.        SetPaths();
12.        LoadKills();
13.    }
14.
15.    if (isEndRank)
16.    {
17.        SetPaths();
18.        LoadFile();
19.
20.        myPlayer.level1enemies = myPlayer.level1enemies + PlayerPrefs.GetInt("kill1");
21.        myPlayer.level2enemies = myPlayer.level2enemies + PlayerPrefs.GetInt("kill2");
22.        myPlayer.level3enemies = myPlayer.level3enemies + PlayerPrefs.GetInt("kill3");
23.        myPlayer.level4enemies = myPlayer.level4enemies + PlayerPrefs.GetInt("kill4");
24.        myPlayer.level5enemies = myPlayer.level5enemies + PlayerPrefs.GetInt("kill5");
25.
26.        SaveData();
27.    }
28.
29.    if (isMainMenu)
30.    {
31.        SetPaths();
32.        LoadFile();
33.
34.        LocalizationManager.Language = myPlayer.defaultLanguage;
35.        Debug.Log("Default Language is: " + myPlayer.defaultLanguage);
36.    }
37. }
38. }

```

Gambar 4. 16 Script StateManager (lanjutan)

#### d. Script Game Over

Digunakan untuk meng-*update* skor pada *database* dan memunculkan *Highscore*.



```

1. public class EndRank : MonoBehaviour
2. {
3.     public Text[] textRankName;
4.     public Text[] textRankScore;
5.     public string[] items;
6.     public Text playerScore;
7.     WWWForm form;
8.
9.     IEnumerator Start()
10.    {
11.        playerScore.text = PlayerPrefs.GetInt("playerScore").ToString();
12.
13.        form = new WWWForm();
14.        form.AddField("nama", PlayerPrefs.GetString("playerName"));
15.        form.AddField("poinhistory", PlayerPrefs.GetInt("playerScore"));
16.
17.        if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo") { form.AddField("highscore", "level1hs");
form.AddField("virus1", PlayerPrefs.GetInt("kill1")); form.AddField("virus1name", "kill1");
form.AddField("ammountVir", 1); }
18.        else if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo 1") { form.AddField("highscore",
"level2hs"); form.AddField("virus1", PlayerPrefs.GetInt("kill1")); form.AddField("virus2",
PlayerPrefs.GetInt("kill2")); form.AddField("virus1name", "kill1"); form.AddField("virus2name", "kill2");
form.AddField("ammountVir", 2); }
19.        else if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo 2") { form.AddField("highscore",
"level3hs"); form.AddField("virus1", PlayerPrefs.GetInt("kill1")); form.AddField("virus2",
PlayerPrefs.GetInt("kill2")); form.AddField("virus3", PlayerPrefs.GetInt("kill3")); form.AddField("virus1name",
"kill1"); form.AddField("virus2name", "kill2"); form.AddField("virus3name", "kill3");
form.AddField("ammountVir", 3); }
20.        else if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo 3") { form.AddField("highscore",
"level4hs"); form.AddField("virus1", PlayerPrefs.GetInt("kill2")); form.AddField("virus2",
PlayerPrefs.GetInt("kill3")); form.AddField("virus3", PlayerPrefs.GetInt("kill5")); form.AddField("virus1name",
"kill2"); form.AddField("virus2name", "kill3"); form.AddField("virus3name", "kill5");
form.AddField("ammountVir", 3); }
21.        else if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo 4") { form.AddField("highscore",
"level5hs"); form.AddField("virus1", PlayerPrefs.GetInt("kill1")); form.AddField("virus2",
PlayerPrefs.GetInt("kill2")); form.AddField("virus3", PlayerPrefs.GetInt("kill4")); form.AddField("virus1name",
"kill1"); form.AddField("virus2name", "kill2"); form.AddField("virus3name", "kill4");
form.AddField("ammountVir", 3); }
22.
23.        WWW w = new
WWW("https://projectsteps.000webhostapp.com/Build/orbsalut_mp/insDataScoreUts.php", form);
24.        yield return w;
25.
26.        w.Dispose();

```

*Gambar 4. 17 Script Game Over*

```

1.     form = new WWWForm();
2.     if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo") { form.AddField("highscore",
"level1hs"); }
3.     else if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo 1") { form.AddField("highscore",
"level2hs"); }
4.     else if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo 2") { form.AddField("highscore",
"level3hs"); }
5.     else if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo 3") { form.AddField("highscore",
"level4hs"); }
6.     else if (PlayerPrefs.GetString("chosenPlayerScene") == "Demo 4") { form.AddField("highscore",
"level5hs"); }
7.     WWW itemsData = new
WWW("https://projectsteps.000webhostapp.com/Build/orbsalut_mp/index.php", form);
8.     yield return itemsData;
9.
10.    items = itemsData.text.Split(';');
11.
12.    for (int i = 0; i < items.Length; i++)
13.    {
14.        textRankName[i].text = (i + 1) + ". " + GetDataValue(items[i], "Nama:");
15.        textRankScore[i].text = GetDataValue(items[i], "Point:");
16.    }
17. }
18. string GetDataValue(string data, string index)
19. {
20.     string value = data.Substring(data.IndexOf(index) + index.Length);
21.     if (value.Contains("&"))
22.         value = value.Remove(value.IndexOf("&"));
23.     return value;
24. }
25. public void Retry()
26. {
27.     PlayerPrefs.SetInt("playerScore", 0);
28.     SceneManager.LoadScene(PlayerPrefs.GetString("chosenPlayerScene"));
29. }
30. public void QuitMM()
31. {
32.     PlayerPrefs.SetInt("playerScore", 0);
33.     PlayerPrefs.SetString("chosenPlayerScene", null);
34.     SceneManager.LoadScene(2);
35. }
36. }

```

Gambar 4. 18 Script Game Over (lanjutan)

### 4.2.3 Pada saat bermain dalam level

#### a. Sistem nyawa pemain

Terdapat variabel untuk mengatur dan menghitung nyawa pemain saat permainan dimulai dan berlangsung yaitu *LivesCount* dan fungsi *PlayerPrefs* untuk terus melacak pergantian variabel nyawa yang berganti-ganti. Ada juga sistem untuk mendeteksi tumbukan antar pemain dengan musuh menggunakan *OnTriggerEnter* dan membandingkan nama *tag* dari setiap virus agar tumbukan bisa terdeteksi dan mengurangi nyawa sesuai dengan jenis musuh. Jika nyawa pemain mencapai angka nol, maka akan memunculkan *scene Game Over*.

```

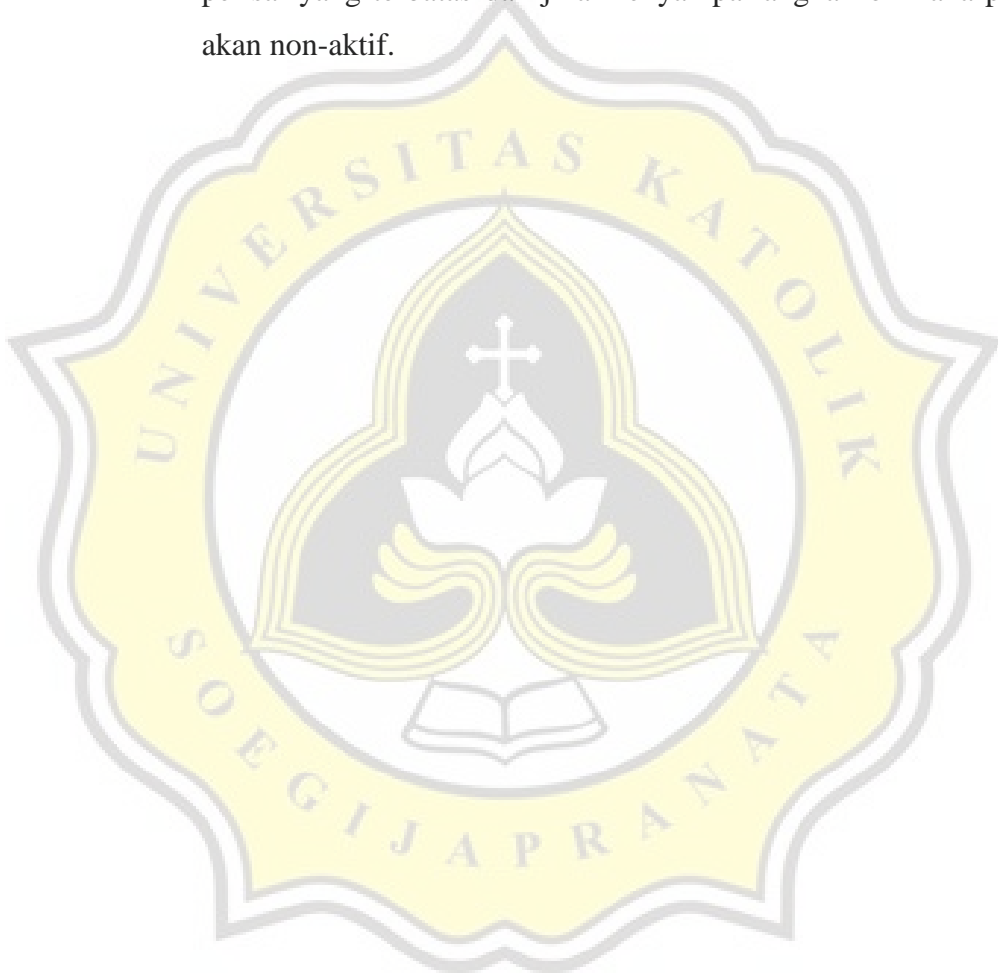
1. public class PlayerLives : MonoBehaviour
2. {
3.     private int LivesCount = 100;
4.     private int curLives;
5.
6.     public void Start()
7.     {
8.         PlayerPrefs.SetString("chosenPlayerScene", SceneManager.GetActiveScene().name);
9.
10.        PlayerPrefs.SetInt("PlayerLives", 100);
11.        curLives = PlayerPrefs.GetInt("PlayerLives");
12.    }
13.
14.    public void OnTriggerEnter(Collider other)
15.    {
16.        if (other.tag == "EnemyCOVID")
17.        {
18.            PlayerPrefs.SetInt("PlayerLives", curLives -= Random.Range(5, 15));
19.            Debug.Log(PlayerPrefs.GetInt("PlayerLives"));
20.            Destroy(other.gameObject);
21.            if (PlayerPrefs.GetInt("PlayerLives") <= 0)
22.            {
23.                SceneManager.LoadScene("GameOver");
24.            }
25.        }
26.
27.        if (other.tag == "EnemyRabies")
28.        {
29.            PlayerPrefs.SetInt("PlayerLives", curLives -= Random.Range(15, 20));
30.            Debug.Log(PlayerPrefs.GetInt("PlayerLives"));
31.            Destroy(other.gameObject);
32.            if (PlayerPrefs.GetInt("PlayerLives") <= 0)
33.            {
34.                SceneManager.LoadScene("GameOver");
35.            }
36.        }
37.
38.        if (other.tag == "EnemyHepaB")
39.        {
40.            PlayerPrefs.SetInt("PlayerLives", curLives -= Random.Range(20, 25));
41.            Debug.Log(PlayerPrefs.GetInt("PlayerLives"));
42.            Destroy(other.gameObject);
43.            if (PlayerPrefs.GetInt("PlayerLives") <= 0)
44.            {
45.                SceneManager.LoadScene("GameOver");
46.            }
47.        }
48.
49.        if (other.tag == "EnemyHIV")
50.        {
51.            PlayerPrefs.SetInt("PlayerLives", curLives -= Random.Range(30, 35));
52.            Debug.Log(PlayerPrefs.GetInt("PlayerLives"));
53.            Destroy(other.gameObject);
54.            if (PlayerPrefs.GetInt("PlayerLives") <= 0)
55.            {
56.                SceneManager.LoadScene("GameOver");
57.            }
58.        }
59.
60.        if (other.tag == "EnemyInf")
61.        {
62.            PlayerPrefs.SetInt("PlayerLives", curLives -= Random.Range(30, 35));
63.            Debug.Log(PlayerPrefs.GetInt("PlayerLives"));
64.            Destroy(other.gameObject);
65.            if (PlayerPrefs.GetInt("PlayerLives") <= 0)
66.            {
67.                SceneManager.LoadScene("GameOver");
68.            }
69.        }

```

Gambar 4. 19 Script sistem nyawa

b. Sistem penembakan pemain

Cara pemain menembak adalah menggunakan kamera AR yang digunakan oleh aplikasi, dengan membidik musuh pada tengah layar dan mengetuk ikon tembak pada layar. Selain menembak, pemain juga dapat menggunakan fitur '*Shield*', dimana pemain akan mendapatkan perisai disekitarnya untuk melindungi diri dari beberapa jenis musuh, tetapi pemain hanya mempunyai nyawa perisai yang terbatas dan jika menyampai angka nol maka perisai akan non-aktif.



```

1. public class PlayerShooting : MonoBehaviour
2. {
3.     public GameObject arCamera;
4.     public GameObject playerShield;
5.     public int curShield;
6.     public Text PointsText;
7.     public Text ShieldText;
8.     private int pointsAccumulated = 0;
9.     public float lastUpdate;
10.    public GameObject ShieldPanel;
11.
12.    void Start()
13.    {
14.        PlayerPrefs.SetInt("playerScore", 0);
15.        PlayerPrefs.SetInt("playerShield", 100);
16.        PlayerPrefs.SetInt("kill1", 0);
17.        PlayerPrefs.SetInt("kill2", 0);
18.        PlayerPrefs.SetInt("kill3", 0);
19.        PlayerPrefs.SetInt("kill4", 0);
20.        PlayerPrefs.SetInt("kill5", 0);
21.        curShield = PlayerPrefs.GetInt("playerShield");
22.        pointsAccumulated = 0;
23.        arCamera = GameObject.Find("/AR Session Origin/AR Camera");
24.        playerShield = GameObject.Find("/AR Session Origin/Shield");
25.        playerShield.SetActive(false);
26.        PointsText.text = pointsAccumulated.ToString();
27.        ShieldText.text = PlayerPrefs.GetInt("playerShield").ToString();
28.    }
29.
30.
31.    public static class PlayerPrefsHelper
32.    {
33.        public static void IncInt(string key)
34.        {
35.            PlayerPrefs.SetInt(key, PlayerPrefs.GetInt(key) + 1);
36.        }
37.    }
38.
39.    IEnumerator ShieldDropping()
40.    {
41.        while (playerShield.activeInHierarchy == true)
42.        {
43.            yield return new WaitForSeconds(0.5f);
44.            curShield = PlayerPrefs.GetInt("playerShield");
45.            PlayerPrefs.SetInt("playerShield", --curShield);
46.            ShieldText.text = PlayerPrefs.GetInt("playerShield").ToString();
47.            if (PlayerPrefs.GetInt("playerShield") <= 0)
48.            {
49.                PlayerPrefs.SetInt("playerShield", 0);
50.                ShieldText.text = PlayerPrefs.GetInt("playerShield").ToString();
51.                StopCoroutine(ShieldDropping());
52.                playerShield.SetActive(false);
53.                ShieldPanel.SetActive(false);
54.            }
55.        }
56.    }
57.

```

Gambar 4. 20 Script PlayerShooting

```

1. public void Shield()
2. {
3.     curShield = PlayerPrefs.GetInt("playerShield");
4.     if (playerShield.activeInHierarchy == false && curShield > 0)
5.     {
6.         playerShield.SetActive(true);
7.         StartCoroutine(ShieldDropping());
8.         ShieldPanel.SetActive(true);
9.         if (curShield <= 0)
10.        {
11.            StopCoroutine(ShieldDropping());
12.            playerShield.SetActive(false);
13.            ShieldPanel.SetActive(false);
14.        }
15.    } else
16.    {
17.        StopCoroutine(ShieldDropping());
18.        playerShield.SetActive(false);
19.        ShieldPanel.SetActive(false);
20.    }
21. }
22.
23. public void Shooting()
24. {
25.     RaycastHit hit;
26.     if (Physics.Raycast(arCamera.transform.position, arCamera.transform.forward, out hit))
27.     {
28.         if (hit.transform.name == "VirusCOVID(Clone)")
29.         {
30.             Destroy(hit.transform.gameObject);
31.             Debug.Log("HIT");
32.             PlayerPrefsHelper.IncInt("kill1");
33.             pointsAccumulated += Random.Range(5, 10);
34.             PlayerPrefs.SetInt("playerScore", pointsAccumulated);
35.             PointsText.text = pointsAccumulated.ToString();
36.         }
37.     }
38. }
39. }

```

Gambar 4. 21 Script PlayerShooting (lanjutan)

- c. Saat pemain yang menggunakan fungsi *shield* bertumbukan dengan musuh, menggunakan *PlayerPrefs* untuk terus melacak pergantian keadaan perisai.

```

1. public class ShieldCollision : MonoBehaviour
2. {
3.     private int curShield;
4.
5.     public void Start()
6.     {
7.         curShield = PlayerPrefs.GetInt("playerShield");
8.     }
9.
10.    public void OnTriggerEnter(Collider other)
11.    {
12.
13.        if (other.tag == "EnemyRabies")
14.        {
15.            PlayerPrefs.SetInt("playerShield", curShield -= Random.Range(10, 15));
16.            curShield = PlayerPrefs.GetInt("playerShield");
17.            Destroy(other.gameObject);
18.        }
19.
20.        if (other.tag == "EnemyHIV")
21.        {
22.            PlayerPrefs.SetInt("playerShield", curShield -= Random.Range(20, 25));
23.            curShield = PlayerPrefs.GetInt("playerShield");
24.            Destroy(other.gameObject);
25.        }
26.
27.        if (other.tag == "EnemyInflu")
28.        {
29.            PlayerPrefs.SetInt("playerShield", curShield -= Random.Range(25, 30));
30.            curShield = PlayerPrefs.GetInt("playerShield");
31.            Destroy(other.gameObject);
32.            PlayerShooting.PlayerPrefsHelper.IncInt("kills");
33.        }
34.    }
35. }

```

Gambar 4. 22 Script ShieldCollision

d. Salah satu mekanisme musuh dalam *level*

Saat musuh muncul dalam area permainan, mereka akan terus mencari posisi pemain secara berkala dan bergerak terus ke arahnya dengan memperbarui koordinatnya menurut pergerakan fungsi *Quaternion* dan *transform*.

```

1. public class VirusBehaviour2 : MonoBehaviour
2. {
3.     public float Speed = 0.5f;
4.     public GameObject Player;
5.     private Vector3 _newPosition;
6.
7.     void Start()
8.     {
9.         Player = GameObject.Find("/AR Session Origin/AR Camera");
10.    }
11.
12.    // Update is called once per frame
13.    void Update()
14.    {
15.        transform.LookAt(Player.transform.position);
16.        Quaternion targetRot = Quaternion.LookRotation(Player.transform.position - transform.position);
17.        transform.rotation = Quaternion.Slerp(transform.rotation, targetRot, 1 * Time.deltaTime);
18.        _newPosition = transform.position;
19.        _newPosition.y += Mathf.Sin(Time.time) * Time.deltaTime;
20.        transform.position = _newPosition;
21.        transform.position += transform.forward * Speed * Time.deltaTime;
22.    }
23. }

```

Gambar 4. 23 Script VirusBehaviour

e. Sistem *Lyceum-Portal* yang dimodifikasi.

Digunakan untuk memperlihatkan dunia virtual melalui sebuah pintu yang muncul pada layar perangkat. Secara sederhana, kamera AR akan melakukan pemindaian pada lantai disekitar pemain dan saat selesai akan memunculkan *mesh* kotak-kotak untuk menandakan bahwa terdapat daerah untuk meletakkan pintu dan ruang *virtual*. Saat pemain melalui pintu dan masuk ke dalam dunia virtual, maka pemandangan sekitar secara visual dan posisi akan menyesuaikan dari titik munculnya pintu *virtual*.

```

1. public class PortalController : MonoBehaviour {
2.
3.     public float cameraForwardActivationPointOffset = 0.01f;
4.
5.     [Tooltip("Seconds to reactive portal when enter it")]
6.     [SerializeField] private float secondsToReactivePortal = 0.1f;
7.     [Tooltip("Find inside portal object to inside portal objects list")]
8.     [SerializeField] private bool getInsidePortalObjectsChilds;
9.     [Tooltip("Objects that are inside portal")]
10.    public List<GameObject> insidePortalObjects = new List<GameObject>();
11.    private List<GameObject> insidePortalObjectsChilds = new List<GameObject>();
12.    [Tooltip("Layer of objects when user is inside portal")]
13.    public LayerMask layerWhenInside;
14.    [Tooltip("Layer of objects when user is outside portal")]
15.    public LayerMask layerWhenOutside;
16.    [SerializeField] private Transform insidePortalParent;
17.    private MeshRenderer meshRenderrer;
18.

```

Gambar 4. 24 Script PortalController



```

1.     public GameObject gameGUI;
2.
3.     private Vector3 camPostionInPortalSpace;
4.
5.     private bool wasInFront;
6.     private bool inOtherWorld;
7.
8.     private bool hasCollided;
9.     private bool frontTraspassed;
10.    private bool cameraBacking;
11.
12.    private bool isInFront;
13.
14.
15.    void Start() {
16.        if (getInsidePortalObjectsChilds) {
17.            GetInsidePortalObjectsChilds();
18.        }
19.        ChangeInsidePortalObjectsLayer(false);
20.        meshRenderrer = GetComponent<MeshRenderer>();
21.    }
22.
23.    private void GetInsidePortalObjectsChilds() {
24.        insidePortalObjects.ForEach(insidePortalObject => {
25.            foreach (Transform child in insidePortalObject.transform) {
26.                insidePortalObjectsChilds.Add(child.gameObject);
27.            }
28.        });
29.        insidePortalObjects.AddRange(insidePortalObjectsChilds);
30.    }
31.
32.    void ChangeInsidePortalObjectsLayer(bool fullRender) {
33.        insidePortalObjects.ForEach(insidePortalObject => {
34.            insidePortalObject.layer = LayermaskToLayer(fullRender ? layerWhenInside : layerWhenOutside);
35.        });
36.    }
37.
38.    public static int LayermaskToLayer(LayerMask layerMask) {
39.        return (int)Mathf.Log(layerMask.value, 2);
40.    }
41.
42.    private bool GetsInFront() {
43.        Vector3 worldPos = Camera.main.transform.position + Camera.main.transform.forward *
44.        (Camera.main.nearClipPlane + cameraForwardActivationPointOffset);
45.        camPostionInPortalSpace = transform.InverseTransformPoint(worldPos);
46.        return camPostionInPortalSpace.y >= 0 ? true : false;
47.    }
48.
49.    private void OnTriggerEnter(Collider collider) {
50.        if (collider.transform != Camera.main.transform)
51.            return;
52.        wasInFront = GetsInFront();
53.        hasCollided = true;
54.    }
55.
56.    private void OnTriggerExit(Collider collider) {
57.        if (collider.transform != Camera.main.transform)
58.            return;
59.        hasCollided = false;
60.    }
61.
62.

```

*Gambar 4. 25 Script PortalController (lanjutan)*

```

1. private void RunPortal() {
2.     if (!hasCollided) {
3.         return;
4.     }
5.     isInFront = GetIsInFront();
6.     if ((isInFront && !wasInFront) || (wasInFront && !isInFront)) {
7.         inOtherWorld = !inOtherWorld;
8.         ChangeInsidePortalObjectsLayer(inOtherWorld);
9.         StartCoroutine(PortalCoroutine());
10.        frontTraspassed = true;
11.    }
12.    CheckTraspassed();
13.    wasInFront = isInFront;
14. }
15.
16. private void OnDestroy() {
17.     ChangeInsidePortalObjectsLayer(true);
18. }
19.
20. private void Update() {
21.     RunPortal();
22. }
23.
24. private bool IsCameraInMid() {
25.     bool cameraCenterIsInFront =
transform.InverseTransformPoint(Camera.main.transform.position).y >= 0 ? true : false;
26.     return isInFront != cameraCenterIsInFront;
27. }
28.
29. private void CheckTraspassed() {
30.     if (!IsCameraInMid()) {
31.         frontTraspassed = false;
32.     }
33.     if (!frontTraspassed && IsCameraInMid()) {
34.         cameraBacking = true;
35.         meshRenderer.enabled = false;
36.     }
37. }
38.
39. private IEnumerator PortalCoroutine() {
40.     if (cameraBacking) {
41.         cameraBacking = false;
42.         meshRenderer.enabled = true;
43.         yield return null;
44.     } else {
45.         meshRenderer.enabled = false;
46.         yield return new WaitForSeconds(secondsToReactivePortal);
47.         meshRenderer.enabled = true;
48.     }

```

*Gambar 4. 26 Script PortalController (lanjutan)*

#### 4.2.4 Pengaturan data pemain pada bagian website

##### a. Definisi dan koneksi ke *database*

Melakukan koneksi MySQL menggunakan kode dibawah ini untuk menghubungkan data permainan ke *database* yang ada. Sehingga permainan bisa mengakses data dari suatu *database engine*.

```

1. <?php
2. define('HOSTNAME','localhost');
3. define('DB_USERNAME','CENSORED');
4. define('DB_PASSWORD','CENSORED');
5. define('DB_NAME','CENSORED');
6.
7. $conn = mysqli_connect(HOSTNAME, DB_USERNAME, DB_PASSWORD, DB_NAME) or die("err");
8.
9. if(mysqli_connect_errno($conn)) echo "Failed to Connect: ".mysqli_connect_error();
10. ?>

```

Gambar 4. 27 Script koneksi database

##### b. Pembuatan data pemain baru

Melalui SQL Query, data pemain akan dibuatkan jika lolos penyaringan pada tahap sebelumnya. Terdapat variabel *username*, *password*, MD5 hash, variabel skor dan musuh yang sudah dibasmi pada setiap *level*.

```

1. <?php
2.
3. include('db.php');
4. if($conn->connect_error){
5.     die("Conn Failed: " . $conn->connect_error);
6. }
7.
8. $nama = $_POST['nama'];
9. $password = $_POST['password'];
10. $passwordmd = password_hash($_POST['password'], PASSWORD_DEFAULT);
11.
12. $sql = "SELECT * FROM user_cred WHERE username = '$nama'";
13. $query = mysqli_query($conn,$sql) or die (mysqli_error($conn));
14.
15. if(mysqli_num_rows($query) == 0){
16.     $sql = "INSERT INTO
17.         user_cred(username, password, passmd, level1hs, level2hs, level3hs, level4hs, level5hs, kill1,
18.         kill2, kill3, kill4, kill5)VALUES
19.         ('".$nama."','".$password."','".$passwordmd."', 0, 0, 0, 0, 0, 0, 0, 0)";
20.     if(mysqli_query($conn, $sql)) {
21.         echo "New Record Created";
22.     } else {
23.         echo "Error: ".$sql."".mysqli_error($conn);
24.     }
25. } else {
26.     die('phpmyadmin error: Username already created.');
```

Gambar 4. 28 Script register PHP

##### c. Pembaruan skor pemain pada *database*

Setiap selesai bermain pada suatu level maka skor atau poin yang didapatkan akan terbaru dalam *database* dalam variabel seperti *virusname*, *pointhistory* dan *highscore* dan jika mendapatkan skor tertinggi (*highscore*) maka itupun akan ter-*update* dalam *database*.

```

1.  <?php
2.
3.  include('db.php');
4.  if($conn->connect_error){
5.      die("Conn Failed: " . $conn->connect_error);
6.  }
7.
8.  $nama = $_POST['nama'];
9.  $pointhistory = $_POST['pointhistory'];
10. $highscore = $_POST['highscore'];
11.
12. $virus1 = $_POST['virus1'];
13. $virus1name = $_POST['virus1name'];
14.
15. $ammountvir = $_POST['ammountVir'];
16.
17.
18. $sql = "SELECT ". $highscore. " FROM user_cred WHERE username = '". $nama. "'";
19. $query = mysqli_query($conn,$sql) or die (mysqli_error($conn));
20.
21. if(mysqli_num_rows($query) == 0){
22.     die('phpmyadmin error: Error updating score. ');
23. } else {
24.     while($row = mysqli_fetch_row($query)){
25.         if($row[0] < $pointhistory){
26.             $sql = "UPDATE user_cred SET ". $highscore. " = ". $pointhistory. "
WHERE username='". $nama. "'";
27.             if(mysqli_query($conn, $sql)) {
28.                 echo "Record Updated";
29.             } else {
30.                 echo "Error: ". $sql. ".mysqli_error($conn);
31.             }
32.         }
33.     }
34. }
35. if($ammountvir == 1){
36.     $sql = "UPDATE user_cred SET ". $virus1name. " = ". $virus1name. " + ". $virus1. " WHERE
username='". $nama. "'";
37. } else if ($ammountvir == 2){
38.     $virus2 = $_POST['virus2'];
39.     $virus2name = $_POST['virus2name'];
40.
41.     $sql = "UPDATE user_cred SET ". $virus1name. " = ". $virus1name. " + ". $virus1. ", ". $virus2name. " =
". $virus2name. " + ". $virus2. " WHERE username='". $nama. "'";
42. } else {
43.     $virus2 = $_POST['virus2'];
44.     $virus3 = $_POST['virus3'];
45.     $virus2name = $_POST['virus2name'];
46.     $virus3name = $_POST['virus3name'];
47.
48.     $sql = "UPDATE user_cred SET ". $virus1name. " = ". $virus1name. " + ". $virus1. ", ". $virus2name. " =
". $virus2name. " + ". $virus2. ", ". $virus3name. " = ". $virus3name. " + ". $virus3. " WHERE username='". $nama. "'";
49. }
50.
51. if(mysqli_query($conn, $sql)) {
52.     echo "Record Updated";
53. } else {
54.     echo "Error: ". $sql. ".mysqli_error($conn);
55. }
56.
57. $conn->close();
58.
59. ?>

```

Gambar 4. 29 Script pembaruan skor PHP

- d. Pengecekan kredensial pemain saat *login* dengan membandingkan data yang diinput dengan *database*

```

1. <?php
2. include('db.php');
3. $username = $_REQUEST['username'];
4. $password = $_REQUEST['password'];
5. $sqlhash = mysqli_query($conn, "SELECT passmd FROM user_cred WHERE
6.                                     username='".$username.'"
AND password='".$password.'"");
7. $shash = mysqli_fetch_array($sqlhash);
8. $sql = "SELECT * FROM user_cred WHERE username = '$username'";
9. $query = mysqli_query($conn,$sql) or die (mysqli_error($conn));
10.
11. if(mysqli_num_rows($query) == 0){
12.     die('phpmyadmin error: User does not exist.');
```

Gambar 4. 30 Script pengecekan kredensial pemain PHP

- e. Pengambilan *Highscore* untuk tampilan dalam *game*

Saat Game Over, game akan memanggil database untuk menunjukkan nilai Highscore tertinggi pada level tersebut

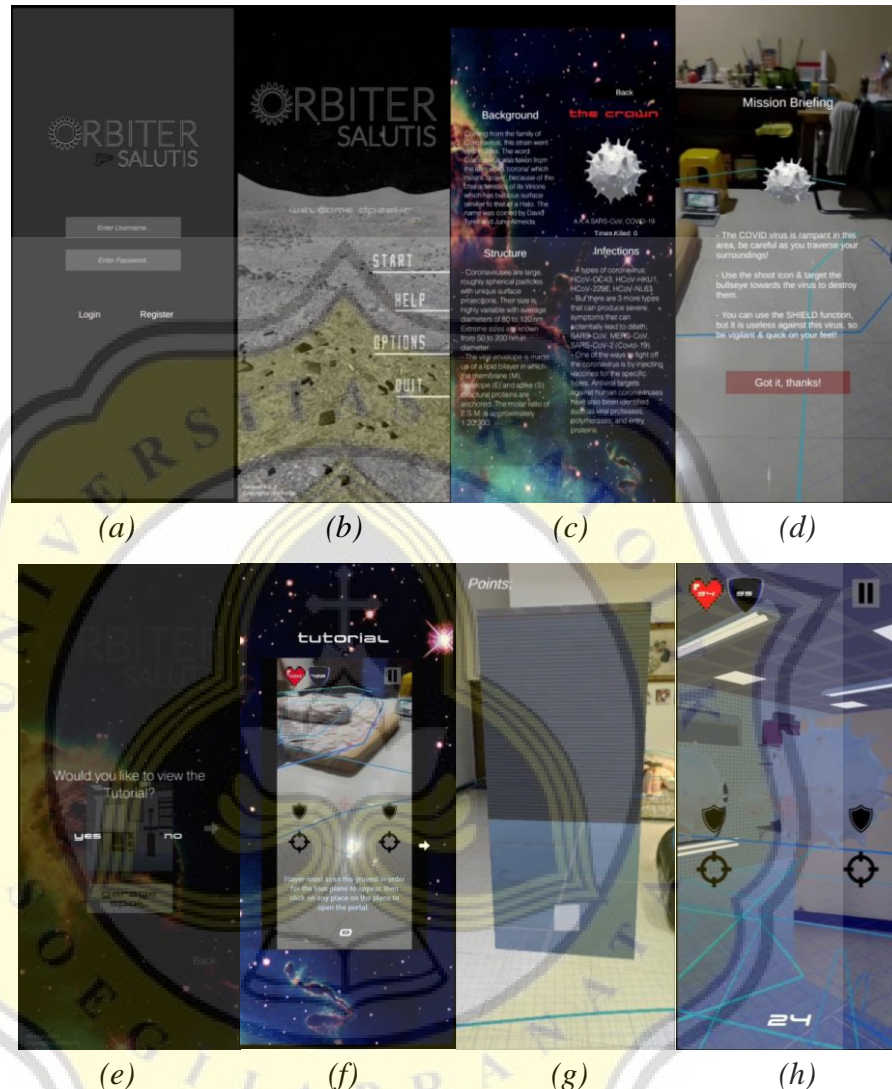
```

1. <?php
2. include('db.php');
3. $levelhs = $_REQUEST['highscore'];
4. $sql = "SELECT * FROM user_cred ORDER BY ".$levelhs." DESC";
5. $query = $conn->query($sql) or die (mysqli_error($conn));
6.
7. while($data = mysqli_fetch_array($query)){
8.     echo "Id:".$data['userid']. "&Nama:".$data['username']. "&Point:".$data[$levelhs]. ";";
9. }
10. ?>
```

Gambar 4. 31 Script pengambilan highscore PHP

#### 4.2.5 Screenshot dari permainan

Berikut merupakan data visual berupa *Screenshot* permainan yang telah dibuat oleh penulis.



Gambar 4. 32 Capture dari Tampilan Game "Orbiter Saluti" (a. Tampilan Login, b. Tampilan Menu, c. Tampilan Help, d. Tampilan Mission Briefing, e. Tampilan Tutorial (1), f. Tampilan Tutorial (2), g. Tampilan Pintu Virtual, h. Tampilan in Game)

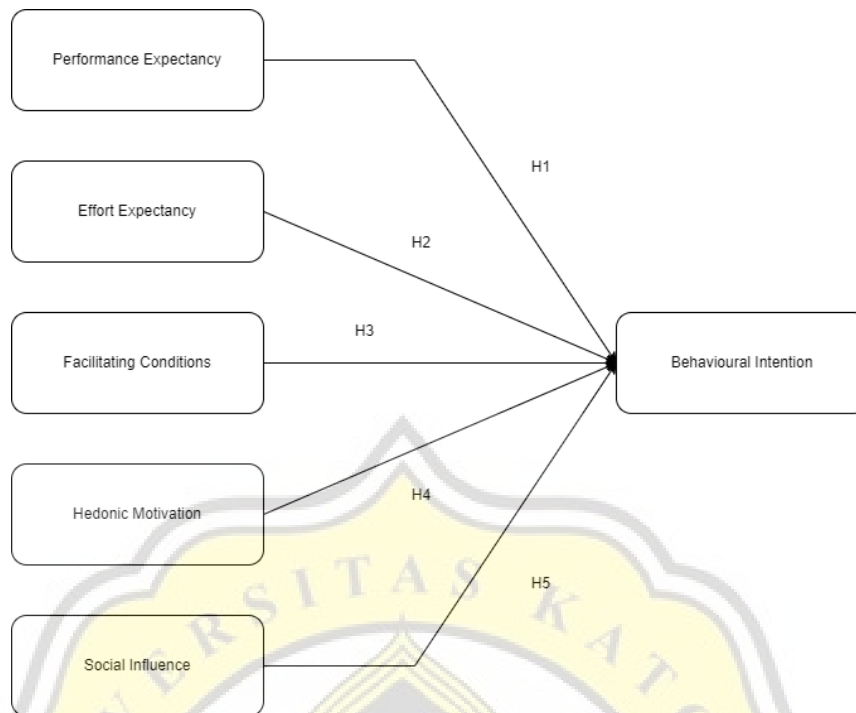
Berikut penjabaran mengenai tampilan dalam permainan "Orbiter Saluti":

1. Tampilan *Login* : Digunakan oleh pemain untuk memasukan *username* dan *password* untuk masuk kedalam game "Orbiter Saluti"

2. Tampilan Menu : Pemain dapat memilih untuk Bermain (*Start*), melihat informasi virus (*Help*), mengganti bahasa (*Option*), atau meninggalkan *game* (*Quit*)
3. Tampilan *Help* : Laman Help digunakan untuk membaca informasi tentang virus yang akan dihadapi pemain.
4. Tampilan *Mission Briefing* : Digunakan untuk memberi tahu misi yang harus dilakukan pemain agar bisa memenangkan permainan. Misi ini memberitahu mengenai calon monster virus yang akan diserang pemain nantinya dalam permainan.
5. Tampilan Tutorial : Ditunjukkan cara bermain (terutama pada pemain pertama kali) agar mengerti mekanik permainannya.
6. Tampilan Pintu Virtual : Digunakan untuk pemain memasuki arena, pemain boleh memainkan permainan di area real-life maupun di dalam ruangan yang telah dibuat oleh penulis.
7. Tampilan in Game : Merupakan tampilan real-game saat pemain bermain game “Orbiter Salutis” dengan cara menembakkan peluru ke target virus untuk membasmi virus yang ada di dalam level yang dimainkan pemain.

### 4.3 Hasil Penelitian

Setelah tahap pembangunan dan pengembangan *game* dijalani dan tereksekusi, maka dilanjutkan dengan penelitian dengan hasilnya diperoleh setelah penulis merangkai serta menyebarkan kuesioner sebagai bahan uji kepada responden-responden yang bersedia dan telah mencoba memainkan *game* ‘*Orbiter Salutis*’.



Gambar 4. 33 Bentuk model sebelum Uji Hipotesa

Gambar 4.33 menunjukkan model hipotesis yang diuji dalam penelitian ini dengan berbagai variabel yang diliat, juga didukung oleh hipotesis peneliti yang dimulai sebagai Hipotesis Alternatif yang muncul dalam penelitian:

- H1 : *Performance Expectancy* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain Orbiter Salutis
- H2 : *Effort Expectancy* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain Orbiter Salutis
- H3 : *Facilitating Conditions* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain Orbiter Salutis
- H4 : *Hedonic Motivation* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain Orbiter Salutis
- H5 : *Social Influence* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain Orbiter Salutis

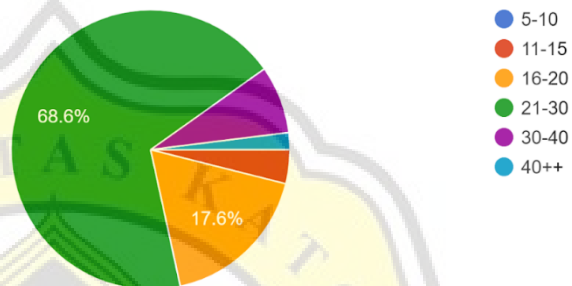


#### 4.3.1 Profil Responden dari Survey

Jumlah responden yang terkumpul dalam kuesioner yang disebar terdapat 51 responden dengan 33 diantaranya telah memainkan *game Augmented Reality*, berikut adalah profil-profil para responden yang didapat:

##### 1. Usia responden

Usia  
51 responses

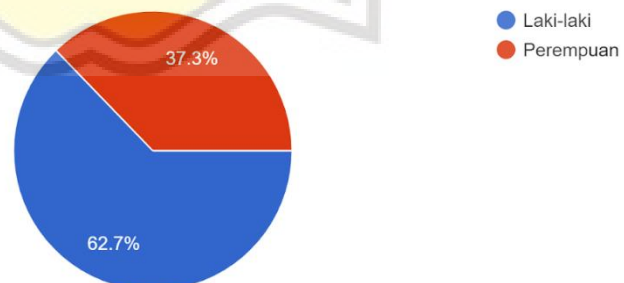


Gambar 4. 34 Hasil usia responden

Dapat dilihat dari gambar 4.34 bahwa mayoritas responden yang ada pada survei ini adalah dari jenjang 21-30 tahun dengan jenjang 16-20 tahun sebagai kedua terbesar. Terdapat juga responden dari umur 30-40 tahun menjadi ketiga terbesar, dan yang sisanya adalah dari umur 11-15 tahun serta umur diatas 40 tahun sebagai yang terkecil.

##### 2. Jenis Kelamin responden

Jenis Kelamin  
51 responses

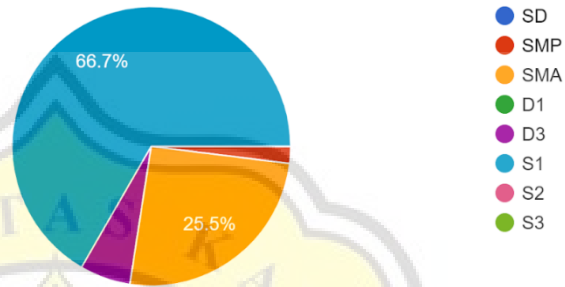


Gambar 4. 35 Hasil jenis kelamin responden

Dilihat dari gambar 4.35, jenis kelamin responden yang dominan pada penelitian ini adalah laki-laki dengan persentase 62.7% dan perempuan meraih 37.3%.

3. Pendidikan Terakhir responden

Pendidikan Terakhir  
51 responses

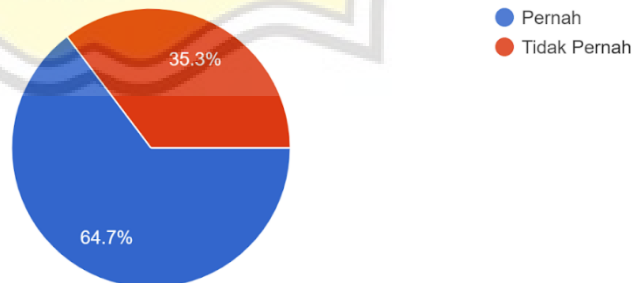


Gambar 4. 36 Hasil pendidikan terakhir responden

Dilihat dari gambar 4.36, responden yang mengisi survei penelitian ini mayoritas memegang pendidikan terakhir S1, dengan pendidikan SMA menjadi kedua terbesar. Terdapat juga beberapa yang memegang pendidikan D3 dan SMP.

4. Persentase Resoponden yang sudah memainkan game Augmented Reality

Apakah anda pernah memainkan game genre Augmented Reality?  
51 responses



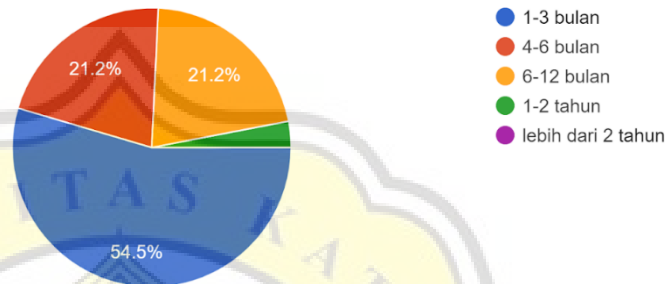
Gambar 4. 37 Hasil "Apakah responden pernah memainkan game AR sebelumnya?"

Mayoritas responden menjawab ‘pernah’ sebanyak 64.7% diikuti dengan responden yang menjawab ‘tidak’ sebanyak 35.3%.

#### 5. Durasi responden yang sudah pernah bermain *game Augmented Reality*

Jika iya, sudah berapa lama memainkan game genre Augmented Reality?

33 responses



Gambar 4. 38 Hasil responden dari berapa lama bermain game AR

Mayoritas responden menjawab bahwa mereka sudah memainkan game genre *Augmented Reality* sekitar 1-3 bulan, diikuti juga responden-responden yang menjawab mereka sudah memainkan sekitar 4-6 bulan dan 6-12 bulan dengan jumlah yang sama. Data terakhir adalah yang terkecil yaitu terdapat responden yang memainkan sudah lebih dari 1-2 tahun.

#### 6. Permainan *game Augmented Reality* yang sudah dimainkan responden

Pokemon Go  
 Pokemon GO  
 Pokemon go  
 pokemon go  
 Pokemon GO, Wizards Unite  
 Wizards Unite  
 Pokemon Go, Ingress  
 pokemon go  
 POKEMON GO  
 Pokemon go

Pokemon GO  
 Pokemen Go  
 Vroid  
 pokemon Go  
 pokemon Go  
 Pokemon  
 Minecraft World, Pokemon GO

Mayoritas responden telah memainkan *game Augmented Reality* yang bernama Pokemon GO. Terdapat juga game lain seperti *Wizards Unite*, *Vroid*, *Ingress* dan juga *Minecraft World*.

#### 4.3.2 Uji Statistik

Bagian berikut dari survei adalah dimana para responden menjawab beberapa rangkaian pertanyaan yang menyangkut pengujian hipotesis yang telah tertera pada bagian sebelum. *Performance Expectancy* (PE) menyinggung masalah keefektivitasan *Orbiter Salutis* dalam mengajarkan responden berbagai virus yang terdapat dalam *game*, *Effort Expectancy* (EE) menyinggung keterampilan responden dalam pengopersian *game Orbiter Salutis*, *Hedonic Motivation* (HM) menyinggung rasa kesenangan responden dalam bermain *game Orbiter Salutis*, *Social Influence* (SI) meyinggung pengaruh sosial dalam bermain *game Augmented Reality* seperti *Oribter Salutis*, dan *Behavioural Intention* (BI) menyinggung kesediaan responden dalam memainkan *Orbiter Salutis* dan/atau *game* yang setara kedepannya jika teknologi, fitur lebih dikembangkan.

##### 1. Hasil *Outer Loading*

Tabel 4. 4 Hasil *Outer Loading*

	BI	EE	FC	HM	PE	SI
BI1	0.930					
BI2	0.861					
BI3	0.837					
EE1		0.867				
EE2		0.761				
EE3		0.910				

FC1	0.813		
FC2	0.836		
FC3	0.704		
HM1		0.817	
HM2		0.914	
HM3		0.881	
PE1			0.705
PE2			0.916
PE3			0.835
SI1			0.853
SI2			0.917
SI3			0.820

Dilihat dari hasil pada Tabel 4.4, nilai dari masing-masing indikator nilai *loading outer*-nya lebih besar dari 0.7 pada setiap kolomnya.

## 2. Hasil *Discriminant Validity* (Fornell-Lacker)

Tabel 4. 5 Hasil *Discriminant Validity* (Fornell-Lacker)

	BI	EE	FC	HM	PE	SI
BI	0.877					
EE	0.683	0.848				
FC	0.728	0.569	0.787			
HM	0.719	0.565	0.540	0.872		
PE	0.262	0.166	0.172	0.179	0.823	
SI	0.654	0.441	0.427	0.492	0.157	0.864

Hasil dari *Fornell-Lacker* jika dilihat pada Tabel 4.5 adalah semua variabel jika dibandingkan dengan dirinya dengan variabel lain pada setiap kolomnya lebih unggul secara nilai. Maka dari itu hasil validitas diskriminan dapat diterima.

## 3. Hasil *Discriminant Validity* (Cross Loadings)

Tabel 4. 6 Hasil *Discriminant Validity* (Cross Loadings)

	BI	EE	FC	HM	PE	SI
B1	0.930	0.762	0.689	0.679	0.254	0.608
B2	0.861	0.501	0.561	0.585	0.263	0.644
B3	0.837	0.509	0.661	0.624	0.169	0.464

<i>EE1</i>	0.521	0.867	0.441	0.557	0.232	0.408
<i>EE2</i>	0.504	0.761	0.478	0.426	0.112	0.288
<i>EE3</i>	0.687	0.910	0.524	0.466	0.094	0.417
<i>FC1</i>	0.717	0.527	0.813	0.497	0.154	0.387
<i>FC2</i>	0.543	0.454	0.836	0.480	0.148	0.349
<i>FC3</i>	0.364	0.305	0.704	0.226	0.089	0.235
<i>HM1</i>	0.584	0.373	0.429	0.817	0.183	0.301
<i>HM2</i>	0.618	0.442	0.435	0.914	0.121	0.470
<i>HM3</i>	0.672	0.642	0.541	0.881	0.165	0.503
<i>PE1</i>	0.142	-0.011	0.037	0.307	0.705	0.110
<i>PE2</i>	0.300	0.259	0.229	0.118	0.916	0.182
<i>PE3</i>	0.102	0.000	0.051	0.031	0.835	0.017
<i>SII</i>	0.481	0.387	0.350	0.237	0.285	0.853
<i>SI2</i>	0.702	0.474	0.453	0.567	0.113	0.917
<i>SI3</i>	0.460	0.246	0.271	0.419	0.016	0.820

Jika dilihat hasil dari Tabel 4.6, nilai-nilai yang dihasilkan kurang lebih sama seperti pada Tabel 4.4.

#### 4. Hasil Construct Reliability (Alpha & Composite Reliability)

Tabel 4. 7 Construct Reliability (Alpha & Composite Reliability)

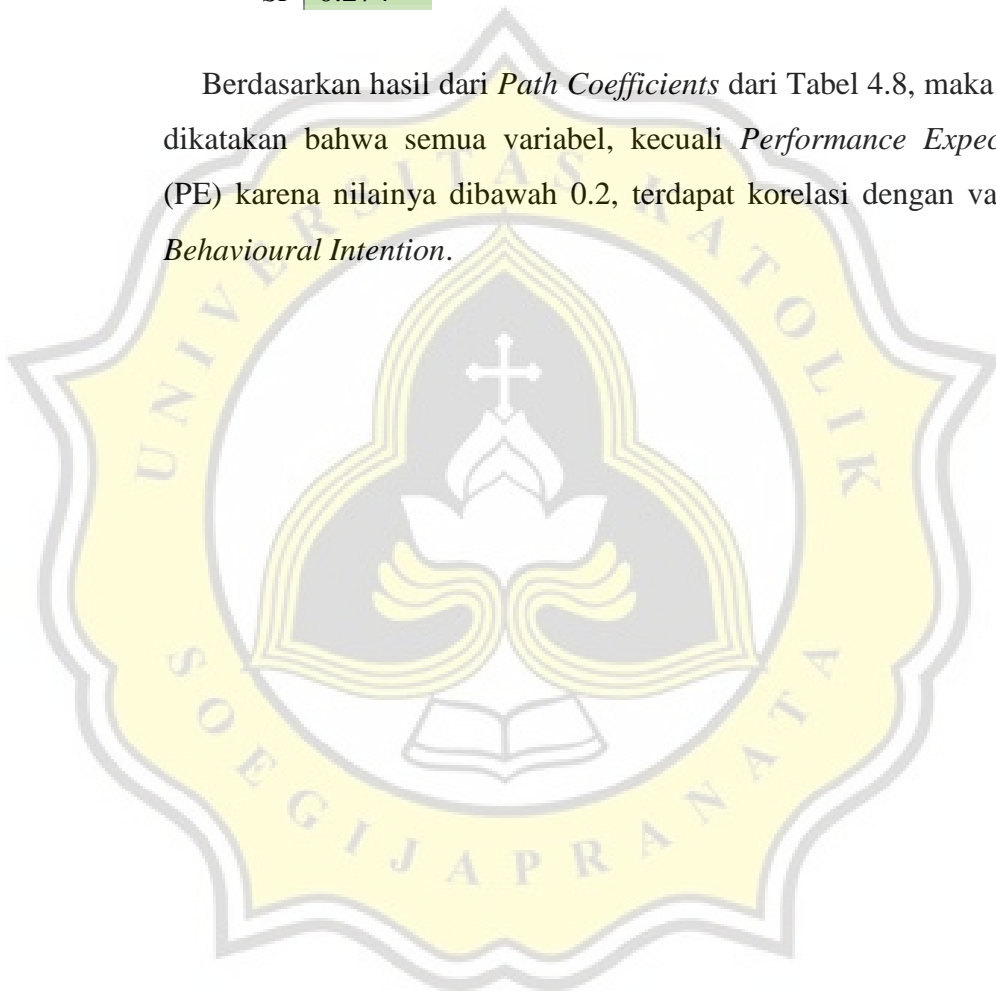
	<i>Cronbach's Alpha</i>	<i>rho_A</i>	<i>Composite Reliability</i>	<i>Average Variance Extracted (AVE)</i>
<i>BI</i>	0.849	0.861	0.909	0.769
<i>E</i>	0.804	0.836	0.885	0.720
<i>E</i>				
<i>F</i>	0.709	0.749	0.829	0.619
<i>C</i>				
<i>H</i>	0.841	0.846	0.904	0.760
<i>M</i>				
<i>P</i>	0.790	1.057	0.862	0.678
<i>E</i>				
<i>SI</i>	0.833	0.890	0.898	0.747

Terlihat dari Tabel 4.7, *Cronbach Alpha* pada setiap variabel terdapat diatas 0.7 dan dapat dikatakan bahwa data yang didapatkan adalah konsisten dan dapat digunakan serta nilai AVE pada seluruh variabel memiliki nilai diatas 0.5, dapat diartikan (dengan hasil Tabel 4.4) bahwa model memenuhi syarat *Convergent Validity*.

5. Hasil *Path Coefficients*Tabel 4. 8 Hasil *Path Coefficients*

	<i>BI</i>	<i>EE</i>	<i>FC</i>	<i>HM</i>	<i>PE</i>	<i>SI</i>
<i>BI</i>						
<i>EE</i>	0.205					
<i>FC</i>	0.332					
<i>HM</i>	0.274					
<i>PE</i>	0.079					
<i>SI</i>	0.274					

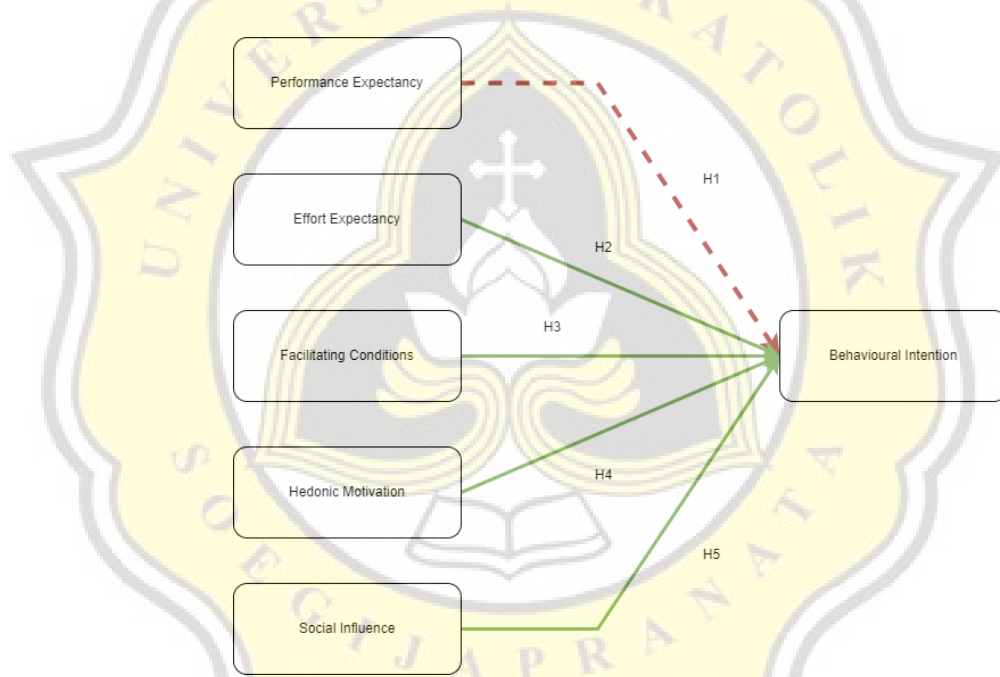
Berdasarkan hasil dari *Path Coefficients* dari Tabel 4.8, maka dapat dikatakan bahwa semua variabel, kecuali *Performance Expectancy* (PE) karena nilainya dibawah 0.2, terdapat korelasi dengan variabel *Behavioural Intention*.



## 6. Uji Hipotesis

Tabel 4. 9 Hasil dari perhitungan nilai P (two-tailed)

	Original Sample (O)	Sample Mean (M)	Standard Deviation (STDEV)	T Statistics ( O/STDEV )	P Values
EE -> BI	0.205	0.197	0.091	2.253	0.024
FC -> BI	0.332	0.330	0.090	3.691	0.000
HM -> BI	0.274	0.276	0.089	3.068	0.002
PE -> BI	0.079	0.083	0.082	0.961	0.337
SI -> BI	0.274	0.275	0.105	2.602	0.009



Gambar 4. 39 Bentuk model setelah Uji Hipotesa

Jika dilihat dari nilai P yang tertera pada Tabel 4.9, hipotesis yang diterima adalah H2 yaitu *Effort Expectancy* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain *Orbiter Salutis*, H3 yaitu *Facilitating Condition* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain *Orbiter Salutis*, H4 yaitu *Hedonic Motivation* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain *Orbiter Salutis* dan H5 yaitu *Social*

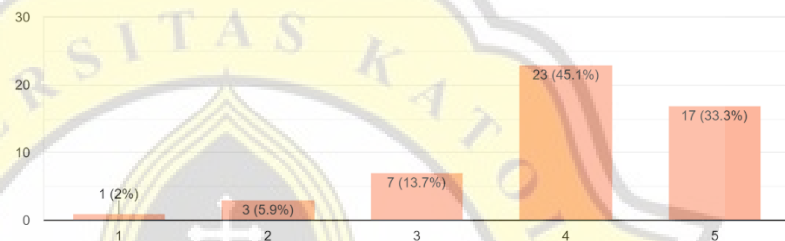


*Influence* mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain *Orbiter Salutis*. Terdapat satu hipotesis yang ditolak dari uji statistik yaitu H1, dimana *Performance Expectancy* ternyata tidak mempunyai pengaruh yang signifikan terhadap *Behavioural Intention* saat bermain *Orbiter Salutis*. Model hipotesis mengikuti perubahan sesuai pengujian seperti yang dapat dilihat pada gambar 4.39.

#### 4.3.3 Hasil survey pemahaman responden terkait virus setelah bermain

(1) Game 'Orbiter Salutis' dapat membantu saya lebih memahami berbagai macam jenis virus.

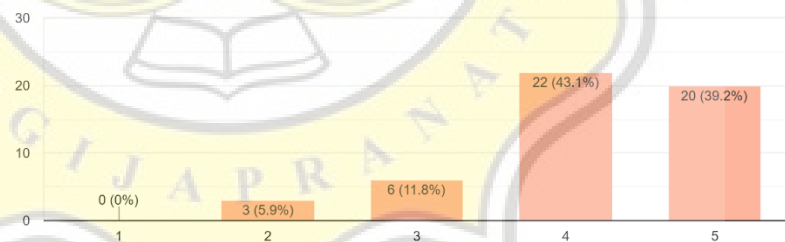
51 responses



Gambar 4. 40 Bagan hasil pernyataan; game dapat membantu saya lebih memahami berbagai macam jenis virus

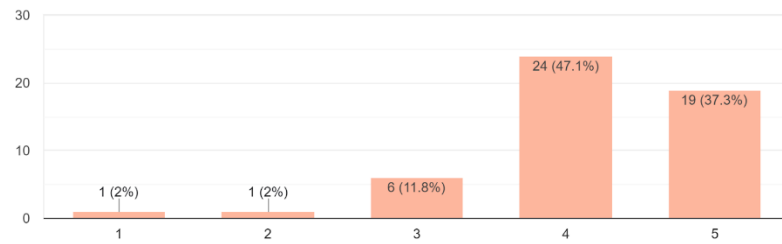
(2) Melalui game 'Orbiter Salutis' memungkinkan saya untuk lebih mengerti berbagai macam jenis virus.

51 responses



Gambar 4. 41 Bagan hasil pernyataan; game dapat memungkinkan saya lebih mengerti berbagai macam jenis virus

(3) Saya rasa game edukasi seperti 'Orbiter Salutis' berguna dalam fungsinya.  
51 responses



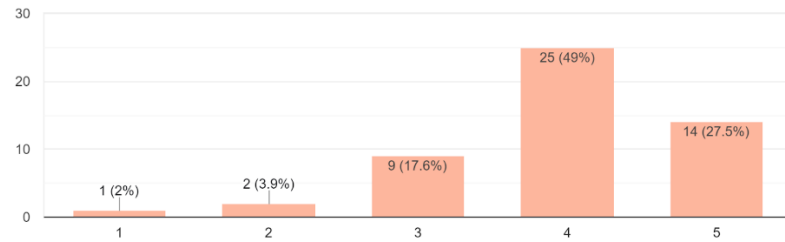
*Gambar 4. 42 Bagan hasil pernyataan; bahwa game seperti Orbiter Salutis berguna dalam fungsinya*

Jika dilihat secara keseluruhan dari hasil *survey* pada gambar 4.40 dimana responden ditanya apakah mereka jadi lebih memahami tentang virus, pada gambar 4.41 ditanyakan apakah memungkinkan dapat menambah ilmu tentang virus dan gambar 4.42 dimana ditanyakan apakah game seperti Orbiter Salutis berguna dalam fungsinya (dalam mendidik), secara garis besar, mayoritas responden (setelah dirata-rata sekitar 84% bercondong setuju, 13% netral dan 3% bercondong tidak setuju) mengatakan bahwa adanya suatu efektivitas dalam mendidik melalui media game seperti Orbiter Salutis dan memungkinkan pemain untuk lebih memahami tentang berbagai macam virus.

#### **4.3.4 Hasil survey responden terkait penggunaan sistem First Person Shooter dalam permainan Augmented Reality**

Salah satu keberhasilan yang dapat dilihat setelah pembuatan dan peluncuran sebuah game ialah berhasilnya implementasi alur bermain (gameplay). *Orbiter Salutis* dikembangkan dengan menggabungkan sistem atau genre *First Person Shooter* dengan teknologi dan *engine Augmented Reality* sebagai sebuah media permainan. Berikut adalah beberapa hasil survey responden dari penggabungan teknologi tersebut:

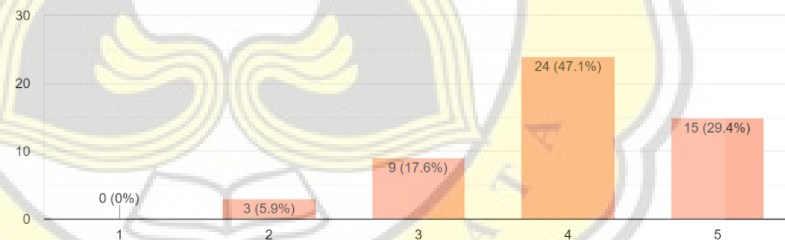
(1) Bermain game Augmented Reality seperti 'Orbiter Salutis' memberi saya perasaan senang.  
51 responses



*Gambar 4. 43 Bagan hasil pernyataan;bermain Orbiter Salutis memberikan responden perasaan senang*

Hasil pada gambar 4.43 menunjukkan bahwa sebagian besar responden (sekitar 76.5%) setuju bahwa dengan bermain game seperti Orbiter Salutis yang menggabungkan FPS dengan teknologi AR mereka merasa senang, dengan sisanya menjawab secara netral (17.6%) atau bahkan tidak setuju atau tidak merasa senang saat bermain (5.9%)

(2) Belajar untuk mengoperasikan game 'Orbiter Salutis' mudah dan dapat dilakukan bagi saya.  
51 responses

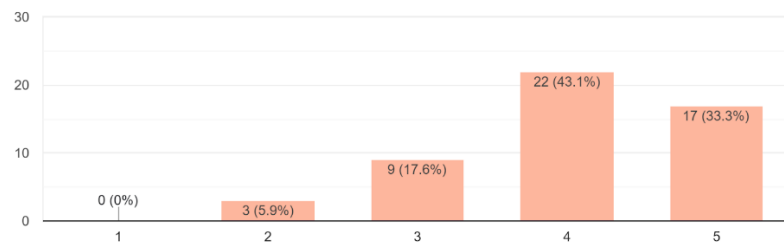


*Gambar 4. 44 Bagan hasil pernyataan; belajar untuk mengoperasikan Orbiter Salutis mudah dan dapat dilakukan*

Hasil pada gambar 4.44 menunjukkan bahwa sekitar 76.5% responden setuju bahwa belajar untuk mengoperasikan game Orbiter Salutis (seperti mengklik User Interface, memainkan permainan inti) mudah dan dapat dilakukan, dengan 17.6% responden menjawab netral (tidak mudah ataupun susah) dan 5.9% menjawab bahwa mereka agak kesusahan dalam pengoperasiannya.

(3) Sebagian besar orang disekitar saya saya harus mengikuti perkembangan teknologi (seperti Augmented Reality, Virtual Reality, Metaverse, dsb).

51 responses



Gambar 4. 45 Bagan hasil pernyataan; sebagian besar orang sekitar responden mengikuti perkembangan teknologi

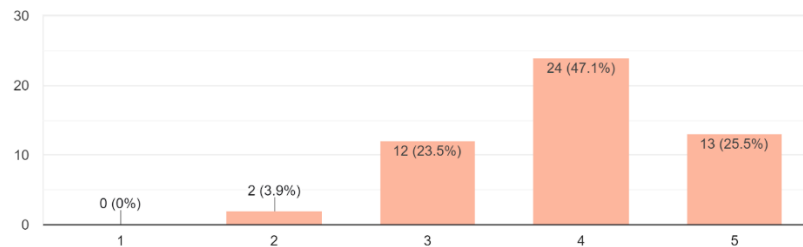
Hasil pada gambar 4.45 menunjukkan bahwa mayoritas responden setuju bahwa sebagian besar orang disekitarnya harus mengikuti perkembangan teknologi seperti penggunaan Augmented Reality, Virtual Reality, dan lain sebagainya dengan beberapa responden menjawab netral dan hanya 5.9% yang menjawab bahwa mereka tidak harus mengikuti perkembangan teknologi.

Dari hasil yang telah didapatkan, maka dapat disimpulkan bahwa ada kepantasan penggunaan genre First Person Shooter dalam teknologi Augmented Reality, baik dilihat dari sisi hiburan atau edukatif dapat diimplementasikan untuk pengembangan *game* yang sudah ada dan yang akan datang.

#### 4.3.5 Hasil *Post-Survey* dan keinginan responden melanjutkan bermain

Pada bagian akhir *survey* juga terdapat bagian dimana responden memberikan nilai kepuasan menurut pendapat mereka terhadap *game Orbiter Salutis* dengan nilai 1 (Kurang Puas) sampai 5 (Sangat Puas) dilihat dari gambar 4.46.

Secara keseluruhan, bagaimana pendapat Anda, mengenai Game Orbiter Salutis?  
51 responses



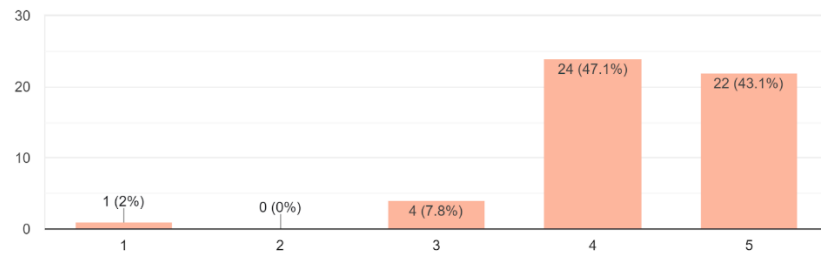
Gambar 4. 46 Pendapat responden terhadap Orbiter Salutis

Sekitar 72.6% responden (jumlah nilai 4 dan 5 digabungkan) lebih condong pada perasaan puas terhadap prototipe *game Orbiter Salutis* yang telah dimainkan, 23.5% menjawab netral dan 3.9% responden menyondong pada ketidakpuasan pada *game*. Selain itu, responden juga dapat memberikan kritik dan saran terhadap pengembangan *game Orbiter Salutis* (opsional), dan hasil yang didapatkan disingkat sebagai berikut:

- 1 Tutorial direvisi karena terdapat bagian yang masih berbahasa asing.
- 2 Warna UI masih kurang bagus, terdapat font yang kurang jelas dibaca.
- 3 Terdapat kejadian model Virus tiba-tiba berubah menjadi hitam (tidak bertekstur).
- 4 Gerakan musuh bisa ditambah variasinya dan lebih dioptimisasi.
- 5 Ada kejadian *Sprite UI* tidak muncul.
- 6 Lebih dikembangkan stabilitas/performa *game* untuk kedepannya.

Berikut adalah keinginan responden dalam melanjutkan bermain *game Orbiter Salutis* jika terus dikembangkan:

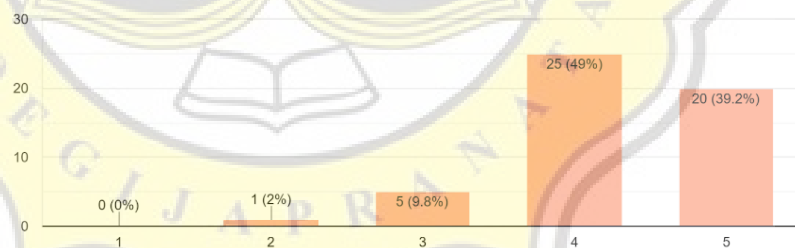
(4) Saya berniat untuk memainkan game 'Orbiter Salutis' jika dilanjutkan pengembangan gamenya.  
51 responses



*Gambar 4. 47 Bagan untuk hasil pernyataan keinginan responden untuk melanjutkan bermain game Orbiter Salutis*

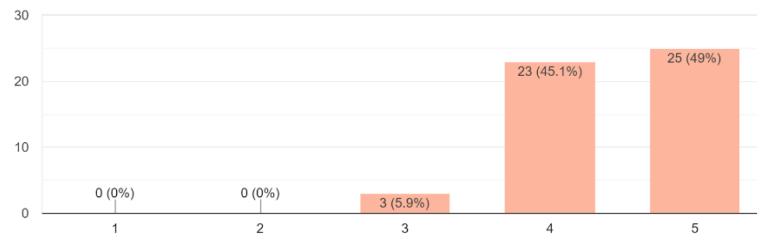
Dapat dilihat gambar 4.47 bahwa mayoritas responden, sekitar 43.1% sangat setuju dan 47.1% setuju akan terus memainkan jika *game Orbiter Salutis* terus dikembangkan. Para responden juga menyatakan pendapatnya bahwa saat bermain mereka juga merasakan ada elemen edukasi saat bermain sehingga jadi lebih menyenangkan dan lebih berniat untuk mencoba game edukatif lainnya yang juga mengedepankan perkembangan teknologi (dapat dilihat pada gambar 4.48 dan 4.49)

(5) Saya rasa bermain game seperti 'Orbiter Salutis' dapat membantu pengedukasian dalam bentuk yang menyenangkan  
51 responses



*Gambar 4. 48 Bagan hasil pernyataan; game dapat membantu pengedukasian jika dibuat dalam bentuk yang menyenangkan*

(6) Menggunakan media game seperti 'Orbiter Salutis' dapat membuat saya lebih berniat untuk mencoba game edukatif yang juga mengedepankan perkembangan teknologi.  
51 responses



*Gambar 4. 49 Bagan hasil pernyataan; game seperti Orbiter Salutis dapat membuat responden lebih berniat untuk mencoba game sejenis lain*

