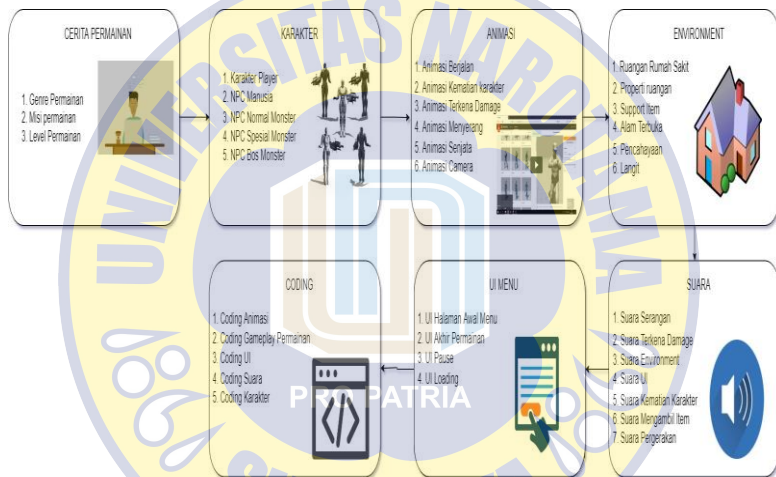


## BAB 4 HASIL DAN PEMBAHASAN

### 4.1 Hasil Penelitian

Proses pembuatan *Game* dalam proses penelitian secara rinci masing masing pekerjaan dan jenis pekerjaan yang dilakukan penulis tunjukkan pada gambar 4.1



Gambar 4. 1 Detail Proses Pembuatan permainan

Tahap pertama sebelum membuat *Game* adalah penulis memulai dengan membuat cerita permainan berGenre FPS yaitu tampilan permainan menggunakan perspektif orang pertama sebagai sudut pandang orang pertama. Setelah menentukan *Genre* permainan yang akan digunakan, selanjutnya penulis menentukan akan menggunakan lima model karakter yaitu satu model karakter PC dan empat

model Karakter NPC yang dijadikan data agent dalam cerita permainan yang ditampilkan dalam tabel 4.1

Tabel 4. 1 Data Agent

Jenis Agent	Deskripsi model
PC	Tentara dengan sudut pandang orang pertama
NPC	Tentara yang harus diselamatkan
NPC	Musuh dalam bentuk zombie
NPC	Musuh khusus dalam bentuk kepala melayang
NPC	Boss Musuh dalam bentuk monster

Tahap selanjutnya adalah memberikan kemampuan pada setiap agen dan aktifitas dari tingkah laku yang dilakukan pada 5 jenis agent seperti pada Tabel 4.2

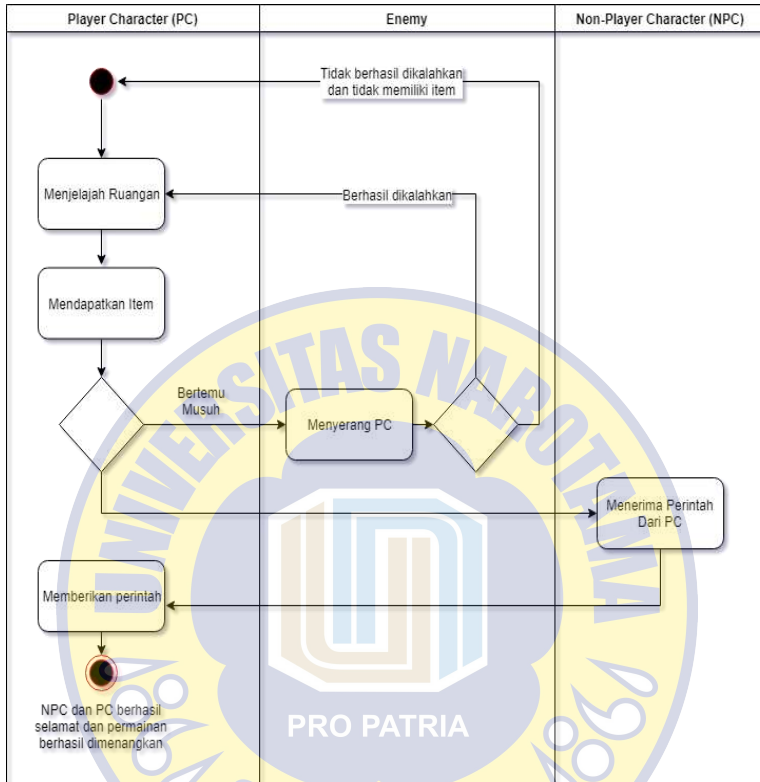
Tabel 4. 2 Daftar Kemampuan dan Tingkah Laku

Jenis Agent	Kemampuan	Tingkah Laku
PC Tentara	<ol style="list-style-type: none"> <li>1. Mampu menunjukkan perilaku yang diarahkan</li> <li>2. Mampu merasakan lingkungannya</li> <li>3. Mampu berinteraksi dengan lingkungan</li> <li>4. Mampu berinteraksi antar agen</li> </ol>	<ol style="list-style-type: none"> <li>1. Menyerang saat dikendalikan</li> <li>2. Mengambil item saat dikendalikan</li> <li>3. Hidup kembali saat mati</li> <li>4. Memberikan perintah kepada agen lain</li> </ol>

NPC Tentara	<ol style="list-style-type: none"> <li>1. Mampu menunjukkan perilaku yang diarahkan</li> <li>2. Mampu merasakan lingkungannya</li> <li>3. Mampu berinteraksi antar agen</li> </ol>	<ol style="list-style-type: none"> <li>1. Menjawab perintah agent PC</li> <li>2. Mengikuti jalan dari agent PC</li> </ol>
NPC Zombie	<ol style="list-style-type: none"> <li>1. Mampu merasakan lingkungan</li> <li>2. Mampu berinteraksi antar agen</li> <li>3. Beraktivitas secara mandiri tanpa kendali</li> </ol>	<ol style="list-style-type: none"> <li>1. Menyerang agen PC saat memasuki suatu trigger</li> <li>2. Mampu mengikuti agent PC</li> <li>3. Jumlah agent yang cukup banyak</li> </ol>
Jenis Agent	Kemampuan	Tingkah Laku
NPC Kepala	<ol style="list-style-type: none"> <li>1. Mampu merasakan lingkungan</li> <li>2. Mampu berinteraksi antar agen</li> <li>3. Beraktivitas secara mandiri tanpa kendali</li> </ol>	<ol style="list-style-type: none"> <li>1. Menyerang agen PC dengan suara</li> <li>2. Mampu mengikuti agent PC</li> <li>3. Memiliki ukuran yang kecil</li> </ol>
NPC Monster	<ol style="list-style-type: none"> <li>1. Mampu merasakan lingkungan</li> </ol>	<ol style="list-style-type: none"> <li>1. Menyerang agen PC saat memasuki suatu trigger</li> </ol>

	2. Mampu berinteraksi antar agen 3. Beraktivitas secara mandiri tanpa kendali	2. Mampu mengikuti agent PC 3. Memiliki jumlah HP cukup besar
--	--	--

Setelah agent dan data kemampuan serta tingkah laku telah di buat maka tahap selanjutnya adalah merancang misi yang harus diselesaikan pemain yaitu dimulai dengan menjelajahi lingkungan *Game* untuk menyelamatkan serta membimbing karakter non-pemain (NPC) keluar dari gedung. Dalam proses bermain, pemain akan dihadapkan dengan musuh yang akan menyerang, sehingga pemain disarankan untuk menjelajahi setiap ruangan dalam permainan untuk mengumpulkan berbagai jenis item mulai dalam bentuk *item* senjata tambahan, item amunisi senjata, dan item penyembuhan yang ditambahkan langsung ke karakter pemain *hitpoint* (HP). Jika seorang pemain dalam kondisi jumlah HP mendekati 0 dan tidak mengambil item penyembuhan serta ketika itu karakter pemain diserang oleh musuh sehingga menyebabkan karakter pemain mati maka karakter pemain akan dibawa kembali ke posisi awal sampai pemain berhasil menemukan karakter NPC dan membimbingnya keluar dari bangunan ke titik penyelamatan di lingkungan terbuka. Alur flowchart permainan diilustrasikan pada Gambar 4.2.



Gambar 4. 2 Flowchart Permainan

Tahap kedua dalam rincian proses kerja penulis merancang desain karakter yang akan digunakan antara lain karakter pemain , NPC manusia, Robot pendukung dan tiga jenis karakter musuh yakni *normal monster*, *spesial monster*, *boss monster* dan dibangun dengan konsep karakter 3D seperti pada Gambar 4.4 Lowpoly 3D Karakter.



Gambar 4. 3 Lowpoly 3D Character

Pembuatan item pendukung juga dilakukan pada tahap ini dengan konsep 3D yang sama item pendukung ini memiliki 3 jenis yaitu item senjata berjenis pistol M-11 dan senapan Avtomat Kalashnikova model 1947 atau dikenal dengan AK-47 beserta item *magazine* peluru yang berfungsi sebagai *trigger* untuk menambah jumlah peluru yang telah didapatkan, dan yang terakhir item penambah nyawa dari karakter pemain yang berbentuk kotak obat. Setelah dibangun karakter dalam konsep 3D maka tahap ketiga ditambahkan animasi kepada setiap karakter antara lain animasi berjalan ditambahkan pada karakter pemain, NPC, *normal monster*, *boss monster*. Pada *special monster* tidak ditambahkan animasi berjalan karena karakter *special monster* bergerak dengan terbang dan karakter robot bersifat dinamis hanya berdiri pada satu tempat. Animasi berikutnya adalah animasi kematian karakter yang dimiliki oleh semua karakter kecuali robot karena fungsi robot hanyalah media pembuka pintu saat didekati oleh karakter pemain. Animasi

terkena *damage* dimiliki oleh setiap karakter yang memiliki animasi kematian karakter karena animasi terkena *damage* ini berjalan sebelum animasi kematian karakter aktif. Saat dalam proses bertarung juga memiliki animasi tersendiri yaitu animasi menyerang yang saling terkait dengan animasi terkena *damage* dan animasi kematian karakter. Animasi senjata ini akan berjalan saat pemain mengambil senjata, mengisi ulang amunisi dan menembak. Pada karakter pemain sebagai sudut pandang orang pertama maka memiliki animasi khusus yaitu animasi kamera yang akan aktif jika beberapa pemicu animasi lain berjalan misalkan saat karakter terkena *damage* maka kamera akan bergoyang sesuai arah datangnya serangan dan saat karakter pemain mati maka kamera akan terjatuh dan muncul bercak merah darah pada tampilan.

*Environment* yang digunakan penulis dalam *Game* ini terdapat dua jenis yaitu lingkungan dalam bangunan dan lingkungan diluar bangunan yang memiliki konsep lingkungan 3D yang dikerjakan pada tahap keempat. Lingkungan dalam bangunan membawa tema rumah sakit yang telah rusak dan ditinggalkan yang berisi berbagai properti layaknya rumah sakit dan beberapa item pendukung seperti item senjata, item amunisi dan item penambah nyawa pemain karakter.



#### Gambar 4. 4 Lingkungan indoor permainan

Pada Lingkungan luar bangunan penulis mengambil tema sebuah lingkungan perbukitan dengan suasana malam dan suasana cahaya yang menyorot dari bulan. Semua karakter yang telah dibuat juga ditempatkan kedalam lingkungan dalam bangunan sesuai dengan jalan cerita.



Gambar 4. 5 Lingkungan outdoor permainan

Semua karakter yang telah dibuat juga ditempatkan di lingkungan di gedung sesuai dengan alur cerita. Setelah item dan karakter yang telah dibuat ditambahkan ke lingkungan, ikuti jalan cerita yang dibuat di awal, kemudian untuk menambah kesan hidup dan merasakan kehadiran karakter kemudian menambahkan suara ke *Game* tahap lima. Pada tahap ini, suara dibagi menjadi dua proses permainan dan suara suara dalam proses interaksi. Dalam permainan ada berbagai jenis suara suara dari kegiatan yang dilakukan oleh proses animasi seperti suara serangan animasi, suara ketika terkena kerusakan, kematian suara karakter, suara ketika



karakter pemain mengambil item pendukung, gerakan karakter terdengar seperti saat bermain. Lingkungan suara juga ditambahkan untuk meningkatkan kesan nyata dari lingkungan yang digunakan dan juga suara UI sedang dipersiapkan sehingga permainan menjadi menarik. UI (antarmuka pengguna) Menu pada tahap keenam antarmuka yang berisi opsi yang dapat digunakan oleh *Game* bagi pemain untuk melakukan aktivitas di luar cerita *Game* sebagai jeda aktivitas yang akan menghentikan *Game* sejenak untuk melanjutkan tombol fungsi untuk kembali ke permainan, tekan. Ada juga opsi perintah *Play* untuk memulai permainan dan *Exit* untuk keluar dari permainan. Di halaman beranda, menu Putar dan Keluar UI memiliki opsi yang dapat dipanggil melalui perintah suara. UI terakhir dari *Game* ini dibuat sebagai tanda bahwa seorang pemain telah menyelesaikan misi permainan. Saat ini, ada juga proses menjalankan *Game* UI yang ditulis sehingga perintah *pause* ditekan pada *keyboard*, jeda UI, ada tiga tombol yang memproses *input* menggunakan kontrol *keyboard* dan mouse adalah tombol *reset*, tombol restart digunakan untuk *restart* permainan dan tombol Keluar. Memuat UI juga ditambahkan ke setiap proses latar belakang yang berjalan, fungsi UI sehingga pemain tahu bahwa proses latar belakang sedang berlangsung.

Tahap terakhir dari proses kerja adalah tahap ketujuh dari proses pengkodean *Game*. Pada tahap ini letakkan permainan logika atau kecerdasan buatan yang terdiri dari lima jenis proses pengkodean animasi, yaitu dalam pengkodean animasi terdapat urutan di mana animasi mulai berjalan secara otomatis sehingga ada pemicu yang terjadi misalnya ketika pemain menyerang dan berlari kode animasi serangan akan menjalankan animasi yang terkena kerusakan karakter yang

menerima serangan. Dalam proses pengkodean permainan, logika permainan akan berfungsi sebagai contoh ketika seorang pemain mengambil item yang meningkatkan masa hidup, jumlah nyawa atau karakter HP pemain akan meningkat, serta amunisi bahwa jumlah tersebut akan meningkat sesuai dengan barang yang diambil. Pengkodean UI berisi perintah untuk memanggil UI ketika kondisi tertentu digunakan sebagai contoh ketika seorang pemain memasukkan kata *Play* untuk memulai permainan kemudian memuat UI akan muncul. Proses pengkodean suara dibagi menjadi dua *Game* pengkodean suara dan pengkodean suara dalam proses interaksi, pengkodean suara berisi kondisi saat aktivitas dalam permainan terjadi ketika karakter pemain menjalankan jejak kaki aktif hingga kondisi aktivitas berhenti, sedangkan pengkodean suara di proses interaksi menggunakan mesin dari *Unity Windows Speech Library* sebagai alat untuk mengenali ucapan pemain. Proses identifikasi suara di *Unity3D 5.5* dimulai dengan pemain mengatakan kata kunci ke mikrofon dan perintah suara akan diproses oleh kode sumber *Unity3D 5.5* yang berisi perkenalan, proses kerja di *Unity3D 5.5* diikuti dengan membuat kamus di sistem yang akan digunakan untuk menyimpan kata kunci dan kegiatan, langkah selanjutnya adalah menambahkan kata kunci dan kegiatan yang terjadi ketika kata kunci dipanggil, setelah kata kunci dibuat dan kemudian diproses dengan mendaftarkannya ke dalam kamus sehingga sistem dapat mengenali kata kunci dan aktivitas kegiatan terjadi. Setelah pemrosesan suara berhasil, karakter akan melakukan kegiatan sesuai dengan perintah suara yang diberikan oleh pemain. Dalam permainan, ada empat jenis kata kunci dan acara yang telah menentukan penulis dan tercantum dalam Tabel 1 Kata Kunci dan Aktivitas *Game*.

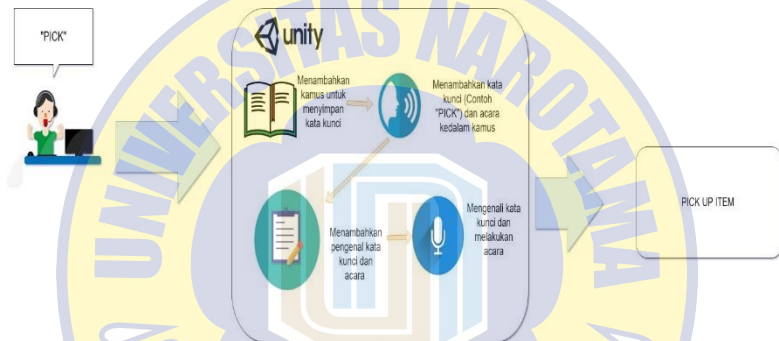
setelah kata kunci dibuat, diproses dengan mendaftarkannya ke dalam kamus sehingga sistem dapat mengenali kata kunci dan kegiatan yang terjadi. Setelah pemrosesan suara berhasil, karakter akan melakukan kegiatan sesuai dengan perintah suara yang diberikan oleh pemain. Dalam permainan, ada empat jenis kata kunci dan acara yang telah menentukan penulis dan tercantum dalam Tabel 1 Kata Kunci dan Aktivitas *Game*. setelah kata kunci dibuat, diproses dengan mendaftarkannya ke dalam kamus sehingga sistem dapat mengenali kata kunci dan kegiatan yang terjadi.

Tabel 4. 3 Daftar Kata Kunci dan Aktivitas

<b>Kata Kunci</b>	<b>Aktivitas yang terjadi</b>
<i>PLAY</i>	Menjalankan permainan pada halaman UI awal permainan.
<i>EXIT</i>	Keluar atau menutup dari permainan pada halaman UI awal permainan.
<i>PICK</i>	Karakter pemain mengambil item yang tersedia seperti item pemulihan, item amunisi, item senjata.
<i>FOLLOW</i>	Karakter memberikan perintah kepada non-pemain (NPC) agar karakter non-pemain (NPC) mengikuti karakter pemain kemanapun karakter pemain bergerak.
<i>OPEN</i>	Karakter membarikan perintah untuk membuka pintu yang ada dalam game

<i>PAUSE</i>	Pemain memberikan perintah untuk melakukan jeda pada permainan
--------------	--

Proses kerja pemrosesan suara terjadi dalam pengenalan suara Unity3D 5.5 menggunakan kode Unity Windows Speech Library dengan perangkat pendukung mikrofon seperti yang ditunjukkan pada Gambar 4.7



Gambar 4. 6 Proses kerja pengenalan suara

Kode program dalam kesatuan digunakan untuk memproses input suara yang diucapkan oleh pemain menjadi fungsi yang digunakan untuk pengontrol *Game*. Langkah pertama dalam kode yang dibuat adalah pernyataan Kode *Plugin Windows Speech* dengan konten dalam Gambar 4.8, kemudian buat Kamus Kode Kelas seperti pada Gambar 4.9, langkah ketiga adalah memasukkan aktivitas ke dalam kode perintah suara, contohnya satu dari perintah "Pick" ditunjukkan pada gambar 10, Selanjutnya adalah menyimpan kata kunci dengan aktivitas dengan kode pada gambar 11, dan yang terakhir adalah mendaftarkan program kata kunci program seperti yang ditunjukkan pada gambar 12.

```
using UnityEngine;
using System.Collections;
using UnityEngine.Windows.Speech;
using System.Collections.Generic;
using System.Linq;
```

Gambar 4. 7 *Windows Speech Plugin Code Statement*

Setelah pernyataan telah dibuat maka dilanjutkan dengan *code class* yang digunakan untuk membuat kamus sebagai media menyimpan kata kunci dan aktivitas seperti yang tercantum pada Gambar 4.6 *Code Class Kamus*.

```
public class FPSPlayer : MonoBehaviour {
    KeywordRecognizer keywordRecognizer;
    Dictionary<string, System.Action>
    keywords = new Dictionary<string, System.Action>();
}
```

Gambar 4. 8 *Class Code Dictionary*

Proses berikutnya dalam Gambar 4.7 adalah membuat kata kunci dan aktivitas yang akan digunakan dan menambahkannya kedalam *class* kamus yang dibuat (contoh kata kunci “*Pick*” dengan aktivitas yang terjadi mengambil item yang tersedia)

```

keywords.Add("Pick", () =>
{
    hit.collider.SendMessageUpwards
    ("PickUpItem", transform.gameObject, SendMessageOptions.DontRequireReceiver);

    hit.collider.SendMessageUpwards
    ("ActivateObject", SendMessageOptions.DontRequireReceiver);
    pickupBtnState = true;
    FPSWalkerComponent.cancelSprint = true;
    usePressTime = -8f;
    objToPickup = null;
});

```

Gambar 4. 9 Proses membuat kata kunci dan aktivitas

Kata kunci yang telah dibuat didaftarkan kedalam *code event OnPhraseRecognized* untuk memulai mengenali kata kunci dan aktivitas yang terjadi, *code event OnPhraseRecognized* ditunjukkan pada Gambar 4.8.

```

keywordRecognizer = new KeywordRecognizer
(keywords.Keys.ToArray());
keywordRecognizer.OnPhraseRecognized +=
KeywordRecognizer_OnPhraseRecognized;
keywordRecognizer.Start();

```

Gambar 4. 10 *Code Event OnphraseRecognized*

Tahap akhir *source code* pengolahan suara yang terjadi dalam Unity3D 5.5 adalah setelah mendaftarkan kata kunci kedalam *event OnPhraseRegonized* maka dibuat *handler* untuk memberikan informai bahwa *event OnPhraseRecognized* dikenali dalam bentuk teks, seperti dalam Gambar 4.9 *Handler Event OnPhraseRecognized*

```

private void KeywordRecognizer_OnPhraseRecognized
(PhraseRecognizedEventArgs args)
{
    Debug.Log("Keyword: " + args.text);
    System.Action keywordAction;

    if (keywords.TryGetValue
(args.text, out keywordAction))
    {
        keywordAction.Invoke();
    }
}

```

Gambar 4. 11 *Event Handler OnPhraseRecognized*

Pada tahap ini, beberapa elemen dalam *Game* diberi kecerdasan buatan melalui pengkodean sehingga setiap elemen dalam *Game* seperti UI, Karakter, Animasi, Lingkungan, dan Suara dapat berinteraksi satu sama lain untuk mengontrol pemain. Di awal *Game* UI, menyandikan perintah suara "Play" dan "Exit" seperti yang ditunjukkan pada Tabel 1. Pengkodean karakter pemain dilakukan dalam kegiatan seperti menyerang kontrol animasi musuh, karakter, dan kondisi kamera ketika terkena serangan musuh, kontrol suara, gerakan, dan kontrol UI. Dalam NPC musuh yang ditanamkan, kecerdasan buatan yang berfungsi ketika karakter pemain melewati pemicu yang telah dipasang akan langsung menyerang pemain NPC musuh. Pengkodean item digunakan untuk menambah nilai status yang baik dari status kehidupan atau jumlah amunisi yang dimiliki oleh karakter pemain. Intelijen buatan NPC memiliki pengkodean jika pemain memberikan perintah suara pada Tabel 1 bahwa NPC manusia akan berjalan mengikuti karakter pemain yang bergerak. Di lingkungan ada juga karakter pendukung adalah robot yang berfungsi jika karakter pemain mendekati pintu tertutup akan terbuka. Di akhir permainan, ada aktivitas di mana karakter memasuki sorotan pesawat, permainan