

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

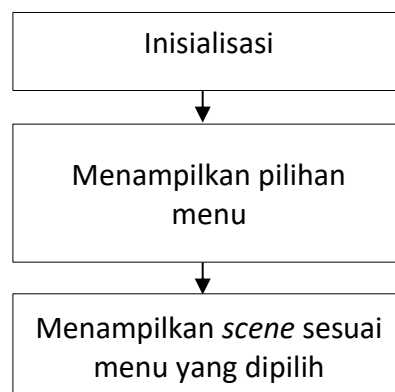
Pada bagian perancangan aplikasi akan dijelaskan perancangan aplikasi VR alat pemantau cuaca BMKG berbasis *Surround Screen Projection*. Perancangan terdiri dari perancangan *work flow*, antarmuka (UI), perancangan obyek 3D, *sequence* dan *class diagram*. Bab ini juga akan membahas mengenai proses implementasi dari aplikasi *virtual reality* alat pemantau cuaca BMKG, implementasi algoritme dan juga antarmuka (UI).

### 5.1 Deskripsi Aplikasi

Aplikasi *virtual reality* alat pemantau cuaca BMKG ini dinamakan “Aplikasi VR BMKG”. Cara kerja aplikasi ini adalah pengguna memilih menu untuk mengakses *environment* taman alat yang berisi obyek 3D alat pemantau cuaca BMKG. Aplikasi lalu merespon dengan menampilkan obyek 3D kepada penggunanya. Narasi berupa audio dimunculkan untuk mempermudah pengguna dalam mengetahui nama alat dan kegunaan dari obyek 3D alat pemantau cuaca yang telah dipilih. Obyek 3D dan *virtual environment* dibuat mendekati kondisi sebenarnya seperti taman alat pemantau cuaca yang dimiliki oleh BMKG Karangploso, Malang.

### 5.2 Perancangan Work Flow

Perancangan *work flow* menjelaskan alur kerja dari aplikasi VR alat pemantau cuaca BMKG. Pertama, pengguna mengakses menu yang tersedia dan memilih menu “Mulai” dengan menggunakan *gaze pointer* dan *trigger gamepad* untuk mengakses *environment* taman alat yang berisi obyek 3D alat pemantau cuaca. Selanjutnya, aplikasi akan menampilkan obyek 3D alat pemantau cuaca. Obyek 3D yang ditampilkan kemudian dapat dipilih dan memunculkan narasi berupa informasi nama alat, kegunaan dan cara kerjanya. Jika pengguna ingin mengakses alat pemantau cuaca yang lain, dapat mengarahkan *gaze pointer* ke obyek 3D lainnya dan memicu dengan *trigger gamepad*. Gambar 5.1 menjelaskan *work flow* dari aplikasi VR BMKG.



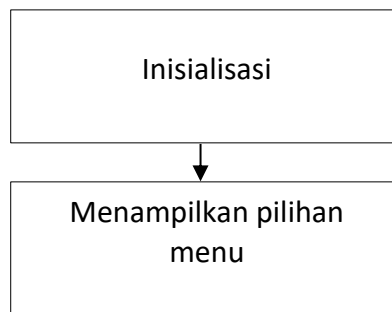
**Gambar 5.1** Diagram Perancangan *Work Flow*

### 5.2.1 Proses Inisialisasi

Proses inisialisasi pada aplikasi VR alat pemantau cuaca BMKG dimulai ketika aplikasi diakses pertama kali. Aplikasi memunculkan *splash screen* berupa logo Unity. Pada proses inisialisasi, semua komponen aplikasi seperti tampilan menu, *environment* taman alat pemantau cuaca, obyek 3D dan audio narasi di-load ke dalam sistem aplikasi.

### 5.2.2 Menampilkan Pilihan Menu

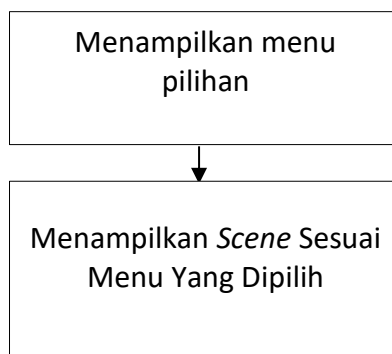
Proses menampilkan pilihan menu dilakukan setelah aplikasi selesai melakukan proses inisialisasi. Selanjutnya aplikasi akan menampilkan pilihan menu, yaitu menu Mulai dan menu Tentang. Kedua menu ini dapat dipilih oleh pengguna dengan menggunakan *gaze pointer* sebagai pengganti kursor dan *trigger gamepad* untuk mengeksekusinya. Gambar 5.2 menjelaskan *work flow* dari proses menampilkan pilihan menu.



**Gambar 5.2** Menampilkan Pilihan Menu

### 5.2.3 Menampilkan Scene Sesuai Menu Yang Dipilih

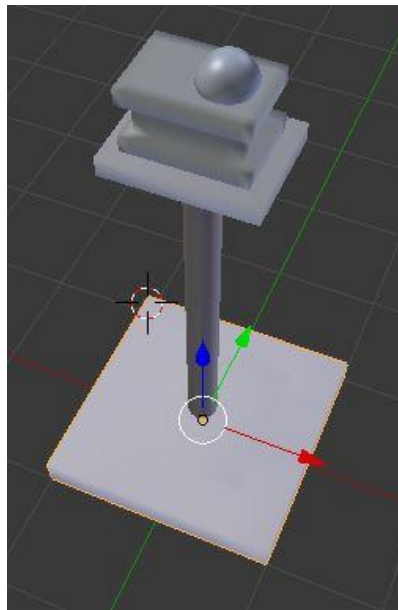
Proses menampilkan *scene* sesuai menu yang dipilih dilakukan setelah pengguna memilih menu yang ingin ditampilkan *scene*-nya dengan menggunakan *gaze pointer* dan *trigger gamepad*. Aplikasi akan menampilkan *scene* sesuai dengan pilihan pengguna. *Scene* yang ditampilkan bisa berupa *environment* taman alat pemantau cuaca beserta obyek 3D alat pemantau cuaca atau *scene* lain yaitu *scene* yang menampilkan profil pengembang aplikasi. Gambar 5.3 menjelaskan *work flow* dari proses menampilkan *scene* sesuai menu yang dipilih.



**Gambar 5.3** Menampilkan Scene Sesuai Menu Yang Dipilih

### 5.3 Perancangan Obyek 3D

Perancangan obyek 3D yang digunakan dalam aplikasi VR alat pemantau cuaca BMKG menggunakan *software* Blender 3D. Langkah pertama yang diperlukan dalam proses perancangan obyek 3D adalah mencari sumber gambar alat pemantau cuaca yang dilakukan dengan cara observasi dan dokumentasi langsung ke taman alat stasiun klimatologi BMKG Karangploso, Malang. Setelah didapatkan data yang cukup mengenai foto obyek alat pemantau cuaca, nama alat serta kegunaannya, selanjutnya foto obyek dibuat modelnya dalam bentuk 3D menggunakan *software* Blender 3D. Gambar 5.4 menjelaskan salah satu foto obyek alat pemantau cuaca yang sudah dimodelkan dalam bentuk obyek 3D menggunakan *software* Blender 3D.



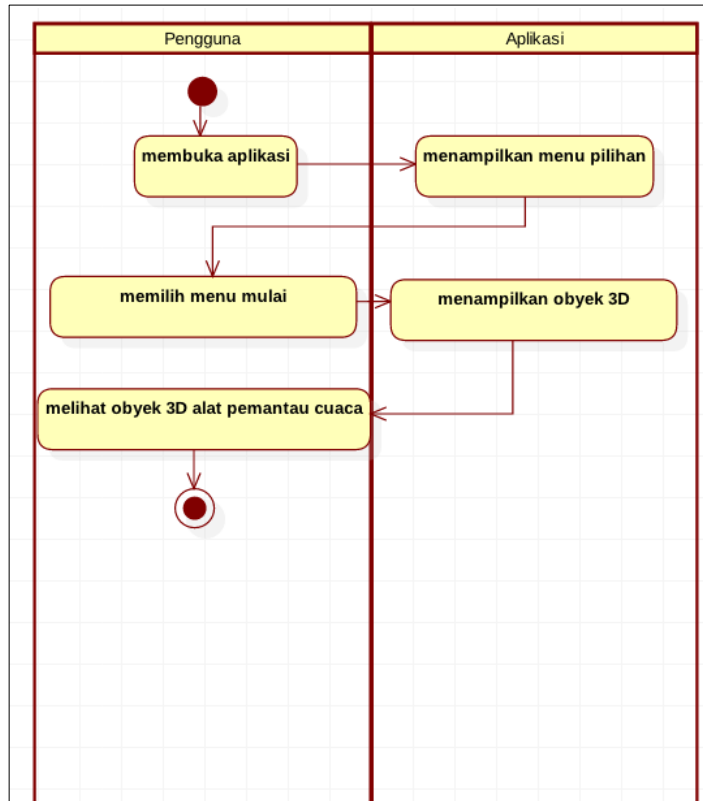
**Gambar 5.4** Obyek 3D Alat Pemantau Cuaca BMKG

Gambar 5.4 adalah wujud dari salah satu obyek 3D alat pemantau cuaca BMKG yang bernama “Aktinograf”. Aktinograf dalam bentuk obyek 3D ini dibuat dengan menggunakan bantuan *software* Blender 3D. Hasil observasi dalam bentuk foto *real* lalu dimodelkan dan disesuaikan ukurannya untuk kemudian diubah ke bentuk obyek 3D.

### 5.4 Perancangan *Activity* dan *Sequence Diagram*

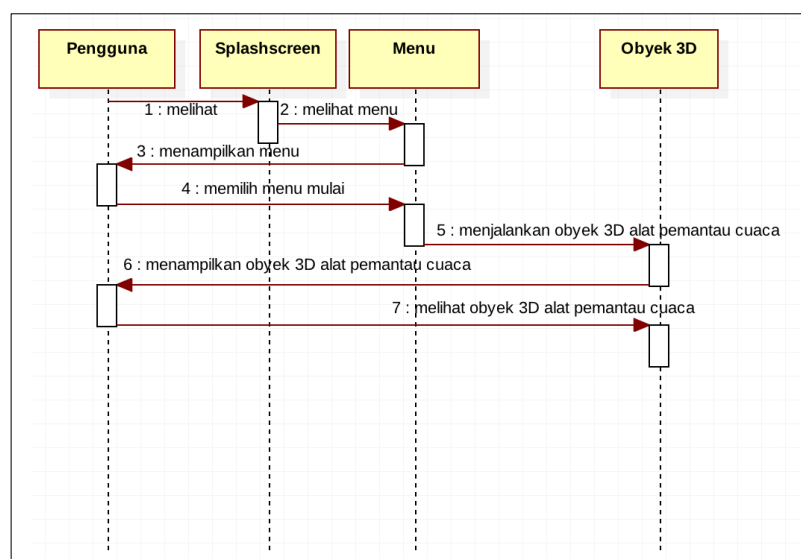
Perancangan *activity diagram* dibuat berdasarkan skenario *use case*. *Activity diagram* menunjukkan aktifitas antara pengguna dengan sistem, respon yang diberikan sistem ketika pengguna mengakses sistem tersebut. Sedangkan *sequence diagram* menunjukkan hubungan antara pengguna dengan sistem. *Sequence diagram* dibuat bertujuan untuk memperjelas fungsi yang dijalankan oleh aplikasi dan mengetahui keluaran apa yang diberikan oleh sistem ketika diakses oleh pengguna.

### 5.4.1 Activity Diagram Melihat Obyek 3D Alat Pemantau Cuaca



**Gambar 5.5** Activity Diagram Melihat Obyek 3D Alat Pemantau Cuaca

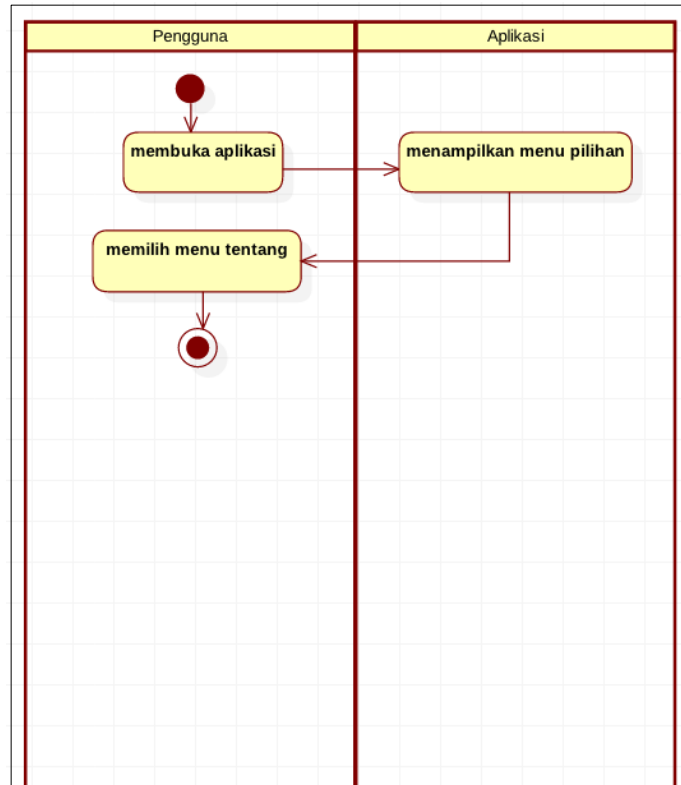
Activity diagram pada Gambar 5.5 menjelaskan ketika pengguna melakukan akses untuk memilih menu alat pemantau cuaca, aplikasi memberikan respon kepada pengguna berupa tampilan obyek 3D alat pemantau cuaca BMKG beserta dengan *environment*-nya. Selanjutnya pengguna dapat melihat tampilan obyek 3D alat pemantau cuaca.



**Gambar 5.6** Sequence Diagram Melihat Obyek 3D Alat Pemantau Cuaca

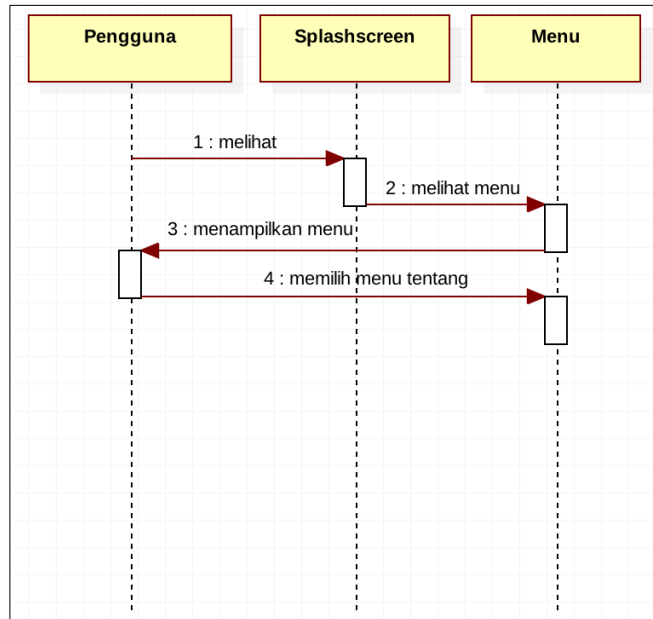
Gambar 5.6 menjelaskan proses pengguna melihat obyek 3D alat pemantau cuaca BMKG. Aplikasi akan menampilkan obyek 3D alat pemantau cuaca yang telah dipilih oleh pengguna. Obyek 3D yang ditampilkan berupa satu alat pemantau cuaca yang diperjelas tampilannya dengan cara mendekatkan jarak antara pengguna dengan obyek 3D alat pemantau cuaca.

#### 5.4.2 Activity Diagram Memilih Menu Tentang



**Gambar 5.7** Activity Diagram Memilih Menu Tentang

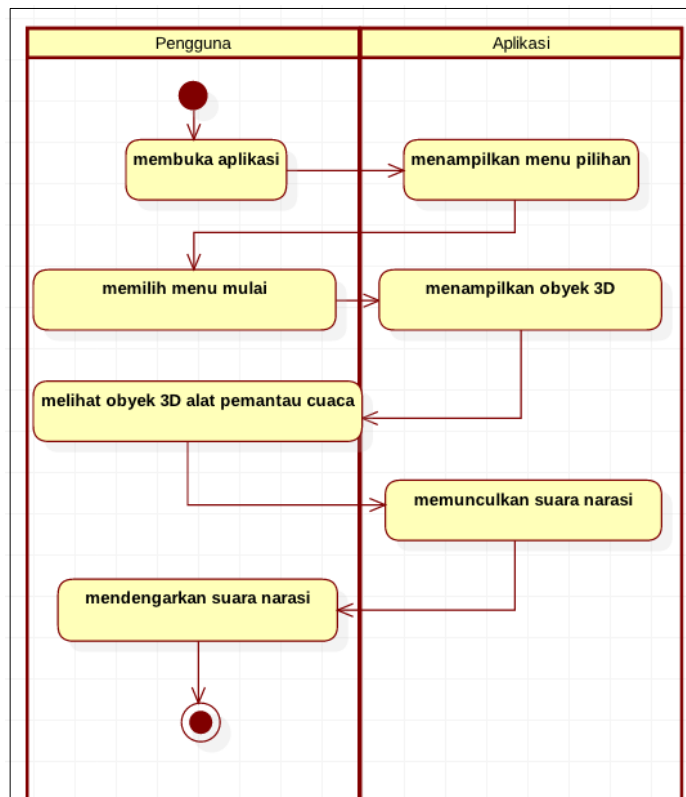
Activity diagram Gambar 5.7 menunjukkan proses yang dilakukan pengguna kepada aplikasi yaitu memilih menu “Tentang”. Sebelum pengguna memilih menu “Tentang”, aplikasi telah menampilkan menu pilihan kepada pengguna.



**Gambar 5.8** *Sequence Diagram* Memilih Menu Tentang

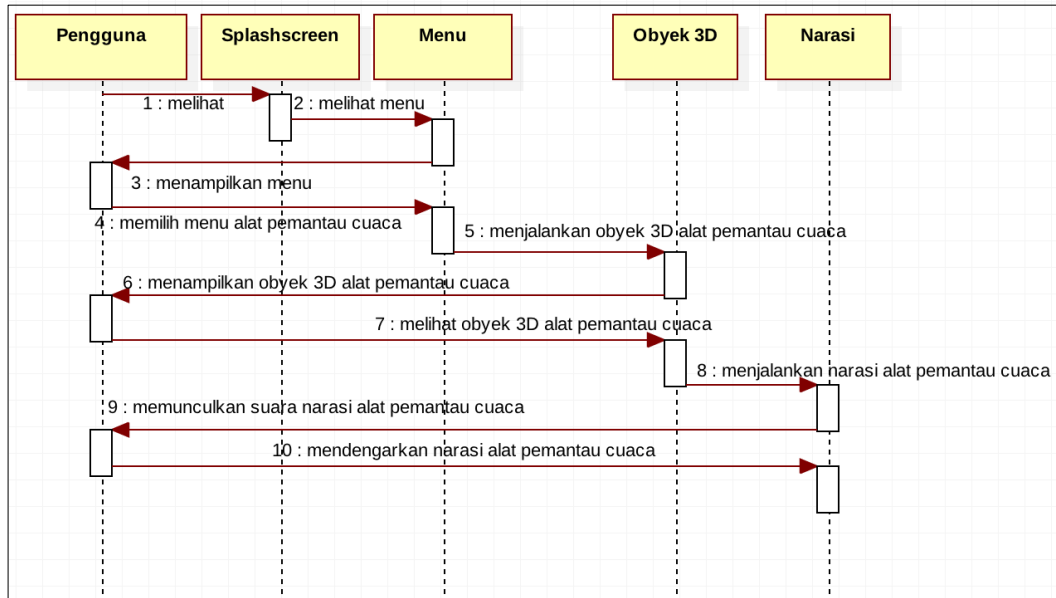
Gambar 5.8 menjelaskan *sequence diagram* pengguna memilih menu “Tentang”. Akses menu ditampilkan oleh aplikasi ketika *splashscreen* berhasil di-*load*, setelah itu pengguna dapat memilih menu tentang pada pilihan menu.

### 5.4.3 *Activity Diagram* Mendengarkan Suara Narasi



**Gambar 5.9** *Activity Diagram* Mendengarkan Suara Narasi

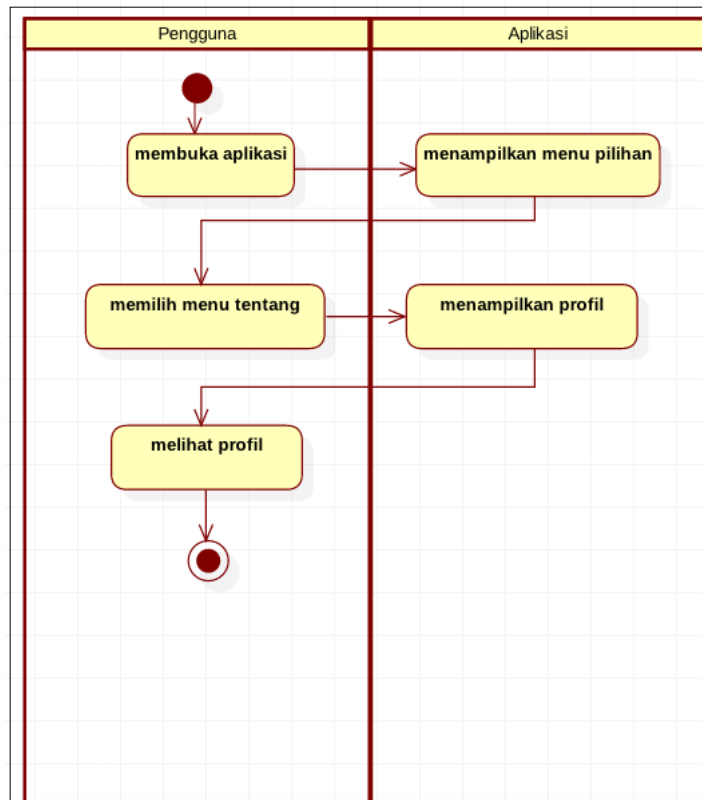
Activity diagram Gambar 5.9 menunjukkan proses yang dilakukan pengguna kepada aplikasi yaitu mendengarkan suara narasi. Pengguna sebelumnya telah memilih menu “Mulai” dan melihat obyek 3D alat pemantau cuaca yang dipilih. Lalu aplikasi meresponnya dengan memutar suara narasi.



**Gambar 5.10** Sequence Diagram Mendengarkan Suara Narasi

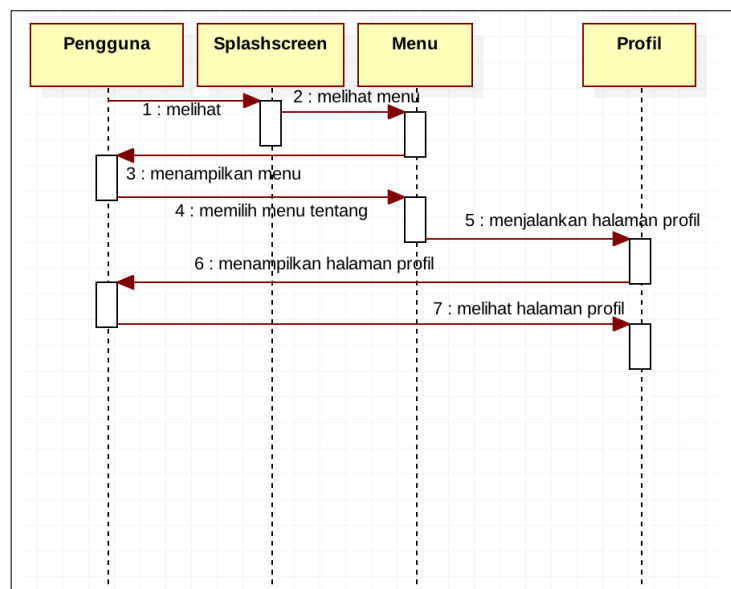
Gambar 5.10 menjelaskan *sequence diagram* pengguna mendengarkan suara narasi. Aplikasi akan menjalankan suara narasi ketika mendapatkan *trigger* dari pengguna yang memilih obyek 3D alat pemantau cuaca yang ingin didengar narasinya. Pengguna dapat mengetahui nama, fungsi dan cara kerja dari alat pemantau cuaca tersebut.

#### 5.4.4 Activity Diagram Melihat Profil Pengembang Aplikasi



**Gambar 5.11** Activity Diagram Melihat Profil Pengembang Aplikasi

Activity diagram Gambar 5.11 menunjukkan proses yang dilakukan pengguna kepada aplikasi yaitu melihat profil pengembang aplikasi. Pengguna sebelumnya telah memilih menu “Tentang” untuk bisa melihat profil pengembang aplikasi.



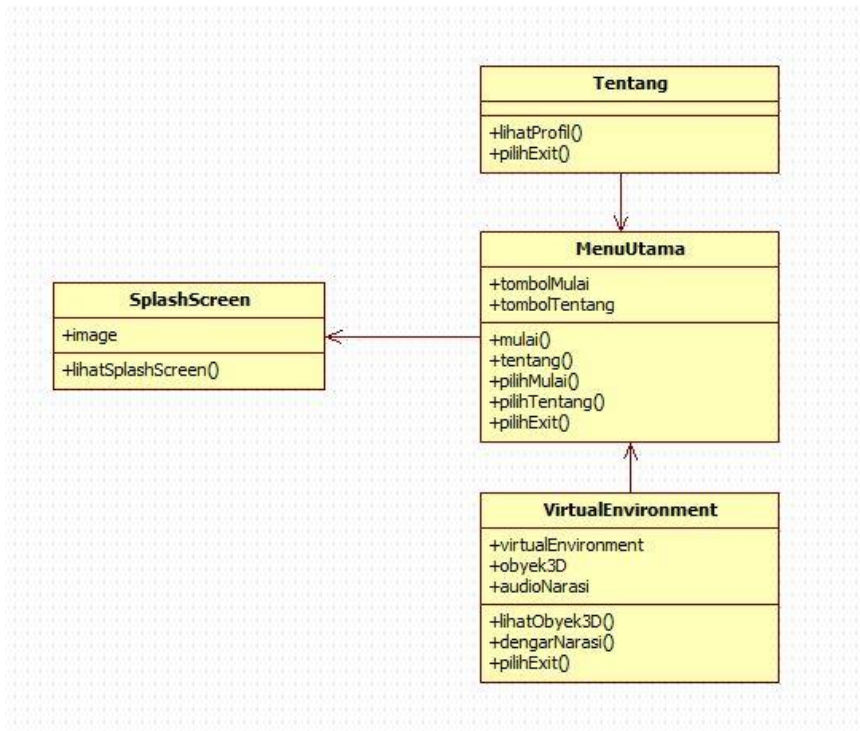
**Gambar 5.12** Sequence Diagram Melihat Profil Pengembang Aplikasi



Gambar 5.12 menjelaskan *sequence diagram* pengguna melihat halaman profil pengembang aplikasi. Aplikasi akan menjalankan halaman profil dan menampilkannya kepada pengguna. Pengguna dapat melihat deskripsi singkat mengenai profil pengembang aplikasi.

## 5.5 Perancangan *Class Diagram*

Perancangan *class diagram* aplikasi *virtual reality* alat pemantau cuaca BMKG terdiri dari sekumpulan *class* yaitu *Splashscreen*, *MenuUtama*, *VirtualEnvironment* dan *class* *Tentang*. *Class* *Splashscreen* berfungsi ketika pertama kali mengakes aplikasi, maka akan muncul gambar kepada pengguna. *Class* *MenuUtama* berfungsi sebagai kendali menu dari aplikasi. Terdapat tombol-tombol yang berguna untuk mengakses menu *VirtualEnvironment* dan *Tentang*. *Class* *VirtualEnvironment* bertugas menampilkan lingkungan *virtual* yang berisi obyek 3D alat pemantau cuaca dan narasi berupa audio. *Class* *Tentang* berguna untuk menampilkan profil pengembang aplikasi.



**Gambar 5.13** *Class Diagram* Aplikasi

*Class diagram* pada Gambar 5.13 memiliki penjelasan, antara lain:

### a. *Class SplashScreen*

*Splashscreen* berfungsi ketika pengguna pertama kali mengakses aplikasi. Sebuah tampilan gambar akan ditampilkan sekilas ke hadapan pengguna. Memiliki *method* `lihatSplashScreen()` yang berguna untuk menjalankan gambar *splashscreen*.

### b. Class MenuUtama

Menu utama berfungsi sebagai kendali pengguna terhadap aplikasi. Terdapat dua buah tombol untuk navigasi aplikasi. *Method* yang berada pada *class* ini antara lain `mulai()`, `tentang()`, `pilihMulai()`, `pilihTentang()` dan `pilihExit()`.

### c. Class VirtualEnvironment

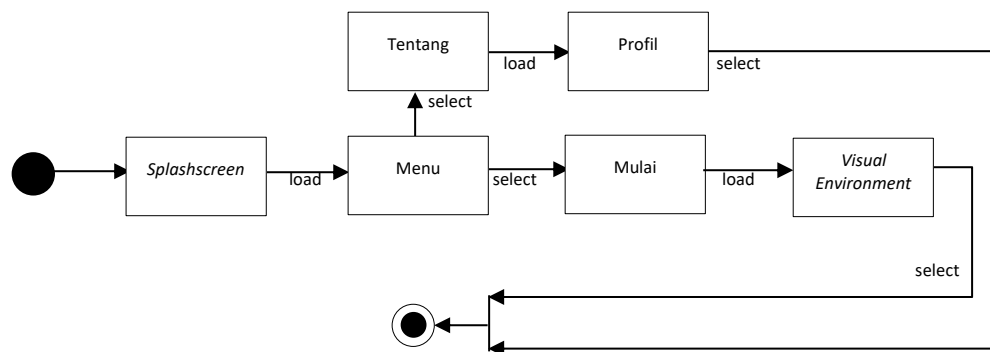
*VirtualEnvironment* berperan menampilkan lingkungan visual yang berisi obyek 3D dan narasi audio dari alat pemantau cuaca BMKG. *Method* yang terdapat di *class VirtualEnvironment* adalah `lihatObyek3D()`, `dengarNarasi()` dan `pilihExit()`.

### d. Class Tentang

*Class* Tentang akan menampilkan informasi singkat mengenai profil pengembang aplikasi. *Method* di *class* Tentang antara lain `lihatProfil()` dan `pilihExit()`.

## 5.6 Perancangan Antarmuka (UI)

Perancangan antarmuka aplikasi *virtual reality* alat pemantau cuaca BMKG menjelaskan bagaimana tampilan antarmuka yang akan ditampilkan kepada pengguna. Tampilan antarmuka dibuat sederhana agar mudah dipahami oleh pengguna, tanpa mengurangi sisi estetika. Gambar 5.14 menjelaskan bagaimana interaksi layar satu sama lain dijelaskan melalui *screen flow diagram* aplikasi. Dimulai dari pengguna membuka aplikasi *virtual reality* alat pemantau cuaca BMKG, lalu muncul layar *splashscreen*. Setelah layar *splashscreen*, kemudian ditampilkan pilihan menu berupa menu “mulai” dan “tentang”. Jika menu yang dipilih adalah “mulai”, pengguna lalu memilih obyek 3D yang ingin dilihat. Aplikasi akan menjalankan obyek 3D alat pemantau cuaca dan narasi dari obyek 3D yang telah dipilih pengguna. Jika menu yang dipilih adalah “tentang” maka akan menampilkan profil tentang pembuat aplikasi.



Gambar 5.14 Screen Flow Diagram Aplikasi

Perancangan antarmuka dari aplikasi *virtual reality* alat pemantau cuaca BMKG dijelaskan pada Gambar 5.15 sampai dengan Gambar 5.18.

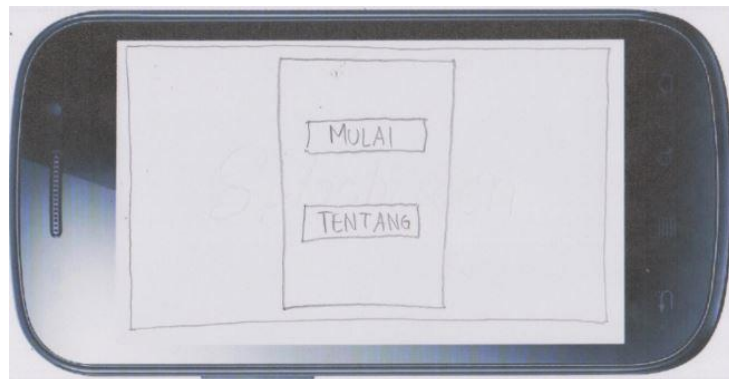
#### 1. Tampilan Antarmuka *Splashscreen*



**Gambar 5.15** Tampilan Antarmuka *Splashscreen* Aplikasi

Antarmuka *splashscreen* aplikasi pada Gambar 5.15 menampilkan tampilan awal ketika pengguna mengakses aplikasi *virtual reality* alat pemantau cuaca BMKG.

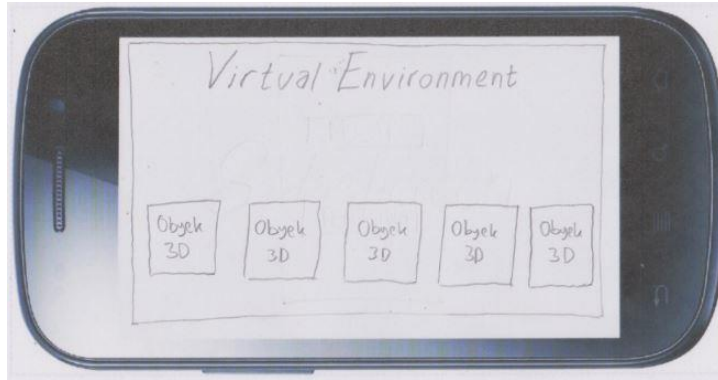
#### 2. Tampilan Antarmuka Menu



**Gambar 5.16** Tampilan Antarmuka Menu

Gambar 5.16 menjelaskan tampilan antarmuka menu aplikasi *virtual reality* alat pemantau cuaca BMKG. Ketika pengguna memilih menu "Mulai", maka aplikasi akan memproses dan menampilkan ke langkah selanjutnya yaitu obyek 3D alat pemantau cuaca beserta narasinya. Sedangkan ketika pengguna memilih pilihan "Tentang", aplikasi akan memproses ke halaman yang berisi profil pengembang aplikasi.

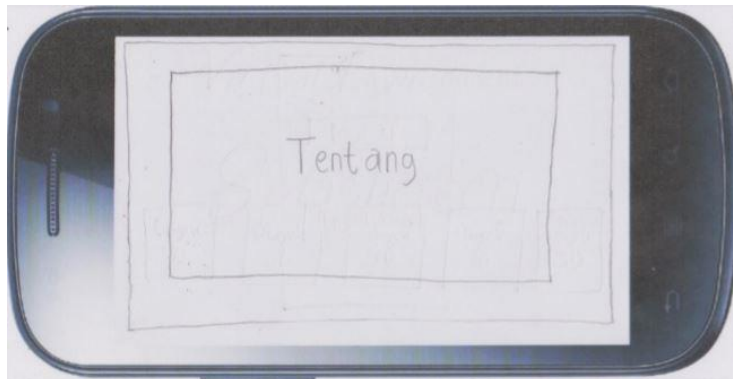
### 3. Tampilan Antarmuka *Virtual Environment* dan Obyek 3D



**Gambar 5.17** Tampilan Antarmuka *Virtual Environment* dan Obyek 3D

Pada Gambar 5.17 menjelaskan tampilan antarmuka *virtual environment* dan obyek 3D alat pemantau cuaca BMKG. *Virtual environment* dan obyek 3D ini akan muncul apabila pengguna memilih mengakses pilihan “Mulai” pada bagian menu. Obyek 3D akan ditampilkan secara horizontal dan pengguna akan memilih obyek 3D mana yang ingin dilihat informasinya. Selanjutnya aplikasi akan menampilkan obyek 3D yang telah dipilih secara lebih mendetail dan memunculkan narasi nama serta kegunaan dari alat pemantau cuaca tersebut.

### 4. Tampilan Antarmuka Tentang



**Gambar 5.18** Tampilan Antarmuka *Tentang*

Tampilan “Tentang” pada Gambar 5.18 akan muncul apabila pengguna memilih menu “Tentang”. Pada halaman ini hanya akan ditampilkan profil mengenai pengembang aplikasi *virtual reality* alat pemantau cuaca BMKG.

## 5.7 Perancangan Algoritme

Bagian perancangan algoritme akan menjelaskan langkah-langkah yang diperlukan untuk menjalankan aplikasi *virtual reality* alat pemantau cuaca BMKG. Perancangan algoritme tersebut dijelaskan dalam *pseudocode* yang menggunakan bahasa manusia.

Algoritme aplikasi dimulai ketika pengguna memulai aplikasi saat pertama kali ditandai dengan *method* `start` yang ditunjukkan pada Tabel 5.1. Jika pengguna memilih untuk menu “Mulai”, aplikasi akan menampilkan halaman *virtual environment*, sedangkan jika memilih menu lainnya maka halaman profil yang akan ditampilkan ke hadapan pengguna.

**Tabel 5.1 Pseudocode Pilihan Menu**

```
start
    if pengguna memilih menu "mulai"
        print virtual environment
        print obyek 3D
    else
        print halaman tentang
        print halaman profil
end
```

Algoritme aplikasi memutar audio narasi dimulai ketika pengguna memilih obyek 3D alat pemantau cuaca yang ingin didengarkan narasinya. Aplikasi lalu mengecek apakah ada audio narasi lain yang masih dijalankan di dalam sistem. Jika masih ada audio narasi lain yang dijalankan di dalam sistem, maka aplikasi akan menghentikannya agar audio narasi yang dijalankan tidak memakan banyak memori dan diputar secara bersamaan. Tabel 5.2 menjelaskan mengenai *pseudocode* dari memutar audio narasi.

**Tabel 5.2 Pseudocode Menjalankan Narasi**

```
start
    if pengguna memilih obyek 3D
    alat pemantau cuaca
        print audio narasi
    else
        print cek audio narasi lain
        yang menyala
        print hentikan audio narasi
        lain yang menyala
end
```

## 5.8 Lingkungan Implementasi

Bagian lingkungan implementasi aplikasi akan menjelaskan bagaimana proses pembuatan aplikasi setelah fase perancangan selesai. Dibahas juga mengenai batasan-batasan dan alur dalam proses implementasi aplikasi.

### 5.8.1 Batasan Implementasi

Batasan implementasi menjelaskan batasan-batasan yang digunakan dalam proses implementasi. Selain batasan-batasan implementasi, dijelaskan juga mengenai perangkat lunak dan perangkat keras yang digunakan dalam proses implementasi. Dijelaskan secara rinci mengenai batasan dan perangkat lunak serta perangkat keras yang digunakan.

1. Aplikasi hanya menampilkan informasi berupa nama, kegunaan dan cara kerja dari alat pemantau cuaca yang dimiliki BMKG
2. Proses implementasi aplikasi menggunakan *software* Unity 5.5 dan Blender 3D
3. Bahasa pemrograman yang dipakai adalah bahasa pemrograman C#
4. Aplikasi diterapkan pada *smartphone* yang menggunakan minimal OS Android versi 4.1 ke atas

Spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam proses implementasi aplikasi alat pemantau cuaca BMKG antara lain:

#### 1. Spesifikasi Perangkat Lunak

Beberapa perangkat lunak yang digunakan dalam proses implementasi antara lain:

- a. Sistem Operasi : Windows 10 Pro
- b. *VR Engine* : Unity 5.6.0f3 Personal
- c. IDE : MonoDevelop
- d. *3D Builder* : Blender 2.78a
- e. *Sound editor* : Audacity

#### 2. Spesifikasi Perangkat Keras

Beberapa perangkat keras yang digunakan dalam proses implementasi antara lain:

- a. Prosesor : Intel<sup>®</sup> Core<sup>™</sup> i5-3210M CPU @ 2.50 GHz
- b. RAM : 8 GB
- c. Kartu Grafis : Nvidia GeForce GT 630M
- e. Harddisk : 750 GB
- f. *VR glasses* : Xiaomi Mi VR Headset Box 3D *Virtual Reality for Smartphone*
- g. *VR device* : Xiaomi Redmi 3S Prime, Android Version 6.0.1 (Marshmallow)

## 5.9 Implementasi Algoritme

Implementasi algoritme dilakukan berdasarkan perancangan yang telah dibuat sebelumnya. Implementasi algoritme menerjemahkan perancangan yang dibuat dalam bahasa manusia ke dalam bahasa pemrograman agar bisa dijalankan dalam bentuk aplikasi. Proses implementasi algoritme menggunakan MonoDevelop IDE dan bahasa pemrograman C#.

### 5.9.1 Implementasi Pilihan Menu

Implementasi pilihan menu menjelaskan bagaimana aplikasi menampilkan pilihan menu kepada pengguna dan respon aplikasi ketika pengguna memberikan *trigger* pilihan dengan menggunakan *gaze pointer*. Aplikasi akan *me-load scene* sesuai dengan *trigger* pilihan pengguna. Tabel 5.3 menjelaskan mengenai implementasi kode dari pilihan menu.

**Tabel 5.3** Skrip Kode Pilihan Menu

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using UnityEngine.SceneManagement;
6  public class Menu : MonoBehaviour {
7      GameObject buttonLooked;
8      public void Mulai () {
9          SceneManager.LoadScene ("scene_vr_bmkg");
10     }
11     public void setButtonLooked(GameObject button){
12         buttonLooked = button;
13     }
14     public void resetButtonLooked(){
15         buttonLooked = null;
16     }
17     public void Tentang () {
18         SceneManager.LoadScene ("scene_vr_bmkg_tentan
19 g");
20     }
21     private void loadScene(){
22         switch(buttonLooked.name){
23             case "MulaiBtn":
24                 {
25                     Mulai ();
26                     break;
27                 }
28             case "TentangBtn":
29                 {
30                     Tentang ();
31                     break;
32                 }
33         }
34     void Update () {
35         if(Input.GetButtonDown("Fire1") && buttonLook
```

```

36 ed != null){
37     loadScene ();
38 }
39 }
40
41 }

```

Penjelasan skrip kode Pilihan Menu:

1. Baris 8-10 merupakan proses menampilkan *scene* dengan nama *scene\_vr\_bmkg*.
2. Baris 11-13 merupakan proses menyimpan pilihan *button* yang telah dipilih pengguna.
3. Baris 14-16 merupakan proses me-*reset* obyek *button* sehingga bernilai *null*.
4. Baris 17-19 merupakan proses menampilkan *scene* dengan nama *scene\_vr\_bmkg\_tentang*.
5. Baris 20-33 merupakan proses seleksi untuk me-*load scene* berdasarkan *trigger* dari pengguna.
6. Baris 34-39 adalah prosedur ketika pengguna menekan *button* "Fire1" maka akan menjalankan *method* untuk me-*load scene*.

### 5.9.2 Implementasi Menjalankan Narasi Berdasarkan *Trigger* Pengguna

Implementasi menjalankan narasi berdasarkan *trigger* pengguna dimulai dengan mendeklarasikan *array* untuk menyimpan alat cuaca apa yang sedang dilihat oleh pengguna. Kemudian jika terdapat *trigger* dari pengguna, sistem akan mengecek audio per alat dan menjalankan audio narasi sesuai *trigger* yang dilakukan oleh pengguna. *Trigger* tersebut didapatkan dari masukan pengguna ketika menekan tombol "Fire1" pada alat kontrol *gamepad*. Tabel 5.5 menjelaskan mengenai implementasi kode dari menjalankan narasi.

**Tabel 5.4** Skrip Kode Menjalankan Narasi

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 public class AlatCuacaObserver : MonoBehaviour {
5     static AlatCuacaObserver instance = null;
6     ObjectTrigger[] alatCuaca;
7     ObjectTrigger alatDilihat;
8     void Awake(){
9         alatCuaca = FindObjectsOfType<ObjectTrigger> ()
10 ;     }
11
12     public static AlatCuacaObserver Instance{
13         get{
14             if (instance == null) {
15                 instance = FindObjectOfType<AlatCuacaOb
16 server> ();
17             }

```



```

18         return instance;
19     }
20 }
21 void Start () {
22 }
23 private void checkAudioPerAlat(){
24     foreach(ObjectTrigger alat in alatCuaca){
25         if (alat.isPlayingAudio ()) {
26             alat.stopAudio ();
27         }
28     }
29     public void OnAlatClicked(ObjectTrigger alat){
30         checkAudioPerAlat ();
31         alat.playAudio ();
32     }
33     public void setLookedAlat(ObjectTrigger alat){
34         alatDilihat = alat;
35     }
36     public void resetLookedAlat(){
37         alatDilihat = null;
38     }
39     void Update () {
40         if (Input.GetButtonDown ("Fire1")) {
41             if (alatDilihat != null) {
42                 checkAudioPerAlat ();
43                 alatDilihat.playAudio ();
44             }
45         }
46     }
47 }

```

Penjelasan skrip kode Menjalankan Narasi:

1. Baris 9-10 merupakan proses mencari obyek dengan tipe `ObjectTrigger`.
2. Baris 12-20 merupakan proses untuk mencari obyek `AlatCuacaObserver` dan mengembalikan obyek yang telah dicari kepada *instance*.
3. Baris 21—28 merupakan proses untuk mengecek audio per alat. Apabila ada alat yang sedang menjalankan audio, maka akan dijalankan *method* untuk menghentikan audio yang sedang dimainkan.
4. Baris 29-32 adalah proses ketika pengguna memberi *trigger* kepada obyek 3D dan menjalankan audio pada obyek 3D.
5. Baris 33-35 merupakan proses untuk mengecek *gaze* pada obyek 3D yang dilihat.
6. Baris 36-38 adalah proses untuk me-*reset gaze* pada alat yang dilihat sehingga bernilai *null*.
7. Baris 39-46 merupakan proses ketika pengguna menekan *button* "Fire1" maka akan menjalankan audio pada obyek 3D yang telah dipilih.

## 5.10 Implementasi Antarmuka (UI)

Implementasi antarmuka menjelaskan proses implementasi yang berkaitan dengan antarmuka dari aplikasi VR alat pemantau cuaca BMKG. Proses implementasi antarmuka dibagi menjadi tiga *scene* yaitu *scene* pilihan menu, *scene environment* dan obyek 3D alat pemantau cuaca, serta *scene* halaman profil. Pembuatan antarmuka aplikasi VR alat pemantau cuaca BMKG pada fase ini dibuat berdasarkan perancangan antarmuka yang telah dilakukan sebelumnya. Antarmuka aplikasi telah bersifat *high fidelity*, sehingga *button*, *environment* dan obyek 3D yang dibuat telah berjalan sesuai fungsinya.



**Gambar 5.19** Tampilan Implementasi Antarmuka Menu

Gambar 5.19 menjelaskan tentang tampilan implementasi antarmuka menu. Terdapat dua buah *button* menu yang bisa dipilih oleh pengguna, yaitu *button* mulai dan tentang. Peletakan dua buah *button* ini telah disesuaikan agar pengguna bisa mengaksesnya dengan mudah ketika menggunakan perangkat VR *glasses*.



**Gambar 5.20** Tampilan Implementasi Antarmuka Obyek 3D Alat Pemantau Cuaca

Tampilan implementasi antarmuka obyek 3D alat pemantau cuaca pada gambar 5.20 dibuat semirip mungkin dengan taman alat yang dimiliki oleh stasiun BMKG Karangploso, Malang. Peletakan pohon dan jalan setapak disesuaikan dengan kondisi sebenarnya. Obyek 3D alat pemantau cuaca yang ditampilkan sebanyak enam buah dari total 15 alat yang dimiliki oleh stasiun BMKG Karangploso. Peletakan enam buah obyek 3D tersebut didasari oleh kenyamanan pengguna ketika menggunakan aplikasi VR alat pemantau cuaca BMKG. Selain itu, keenam alat tersebut dianggap telah mewakili dari beragam alat pemantau cuaca yang terdapat di stasiun BMKG Karangploso, Malang.

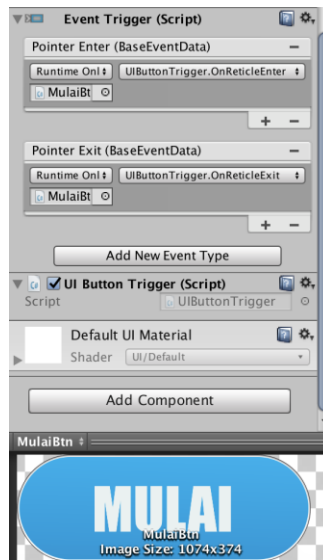


**Gambar 5.21** Tampilan Implementasi Antarmuka Profil Pengembang

Gambar 5.21 menjelaskan tentang tampilan implementasi antarmuka profil pengembang. Halaman ini bisa diakses oleh pengguna ketika memilih menu tentang pada pilihan menu di awal. Halaman profil pengembang menjelaskan tentang profil singkat dari pengembang aplikasi VR alat pemantau cuaca BMKG.

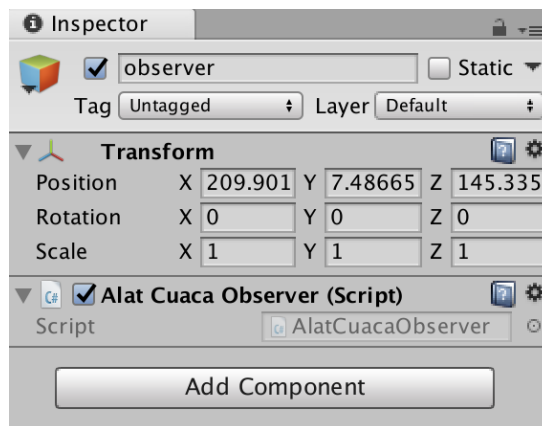
### 5.11 Implementasi *Virtual Reality* (VR)

Implementasi *virtual reality* (VR) menggabungkan antara kode pemrograman yang telah dibuat dengan *gameobject* yang terdapat pada Unity. Untuk bisa menjalankan dan berinteraksi dengan aplikasi, kode pemrograman yang telah dibuat harus diintegrasikan dengan *gameobject*. Apabila tidak diintegrasikan, *gameobject* yang terdapat pada Unity tidak akan bereaksi ketika dijalankan. Kode pemrograman perlu ditambahkan ke dalam *gameobject* yang telah menjadi target.



**Gambar 5.22** Integrasi Kode Pemrograman dengan *GameObject Button*

Gambar 5.22 menunjukkan kode pemrograman yang telah dibuat ditandai dengan tulisan *Script* diintegrasikan dengan *gameObject Button* yang terdapat pada Unity. Skrip kode Even Trigger berfungsi untuk mendeteksi *trigger gaze pointer* dari pengguna. MulaiBt adalah *gameObject* target yang disisipi skrip kode Even Trigger, sehingga *gameObject* ini akan bereaksi ketika mendapat *trigger* dari pengguna. Sedangkan skrip kode UI Button Trigger berisi tentang *method* yang mengatur langkah yang harus dijalankan oleh sistem setelah mendapatkan *trigger* pengguna.



**Gambar 5.23** Integrasi Kode Pemrograman dengan Obyek 3D Alat Pemantau Cuaca

Gambar 5.23 menunjukkan integrasi kode pemrograman dengan nama skrip Alat Cuaca Observer dengan *gameObject* alat pemantau cuaca yang terdapat pada *environment taman* alat pemantau cuaca BMKG. Skrip Alat Cuaca Observer berfungsi untuk mengatur audio narasi dijalankan satu persatu sesuai pilihan pengguna. Skrip ini juga memastikan bahwa *trigger* pengguna tepat mengenai obyek 3D yang dituju dan bukan mengenai obyek yang lain.