

## BAB V. IMPLEMENTASI DAN PENGUJIAN

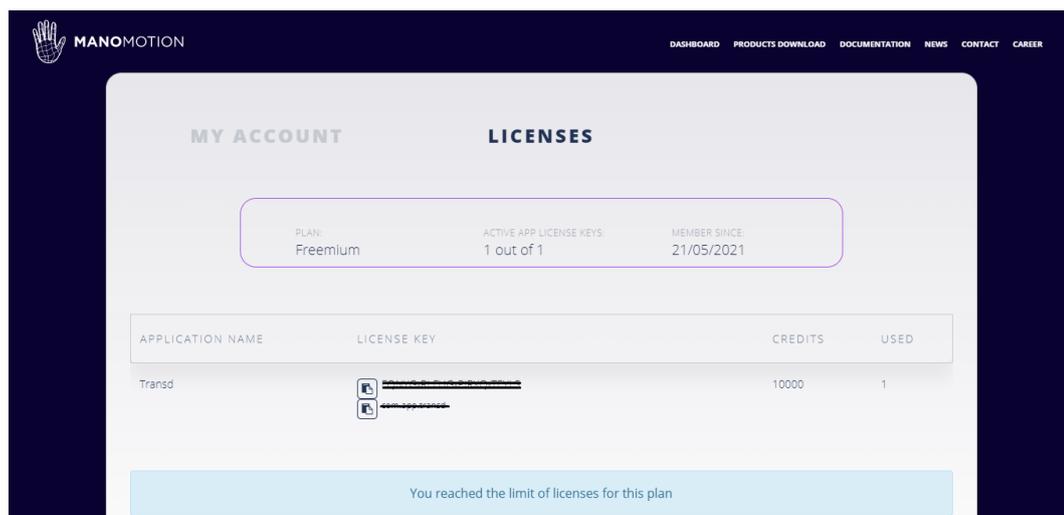
### 5.1 Implementasi Sistem

Pada tahap implementasi sistem berikut ini akan dijelaskan mengenai bagaimana aplikasi dibuat dan berjalan berdasarkan analisis dan perancangan yang dilakukan sebelumnya pada bagian perancangan sistem. Berdasarkan pada perancangan sistem sebelumnya maka dalam penerapan *Hand Recognition* dengan *Augmented Reality* diperlukan perangkat lunak Unity Hub 2018.2.5f1 yang didapatkan dari tautan <https://unity3d.com/get-unity/download/archive>, dimana perangkat lunak tersebut sudah disediakan oleh unity dan sudah terpasang pada perangkat laptop.



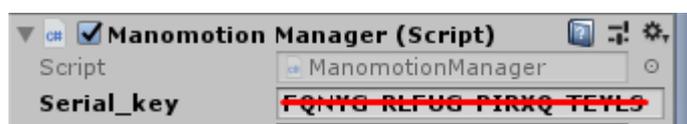
Gambar 5.1 Perangkat Lunak Unity

Kemudian diperlukan perangkat lunak Manomotion SDK yang didapatkan dari tautan <https://www.manomotion.com/products/>, dimana perangkat lunak tersebut sudah disediakan oleh manomotion dengan tiga macam SDK. Pada penelitian ini penulis menggunakan manomotion SDK dengan versi SDK Manomotion 2.0. SDK tersebut didapatkan dengan cara *register* pada *website* milik manomotion kemudian membuat *licence key* pada bagian *licenses*.



Gambar 5.2 *License* Perangkat Lunak Manomotion SDK

Pada Gambar 5.2 terdapat peringatan “*You reached the limit of licenses for this plan*”. Hal tersebut dikarenakan SDK yang digunakan merupakan SDK yang bersifat gratis dan dapat diunduh untuk penggunaannya. Penulis menggunakan SDK versi tersebut dimaksudkan agar SDK dapat berjalan pada perangkat mobile yang tidak memiliki fitur ARCore. Pembuatan *license key* ini digunakan untuk menerapkan *hand recognition* pada *augmented reality* dengan cara memasukkan *lisence key* kedalam Unity. Sebelum memasukkan *license key*, diharuskan *import Manomotion Unity package* yang sudah terunduh kedalam *project* Unity. Setelah berhasil melakukan *import*, maka *license key* dapat dimasukkan kedalam *Game Object* ManomotionManager pada bagian *Serial\_key* seperti pada Gambar 5.3 di bawah ini kemudian SDK siap digunakan.



Gambar 5.3 Memasukkan *License Key* pada ManomotionManager

Dalam penerapannya, Manomotion SDK digunakan pada bagian menu Materi dan menu Permainan pada aplikasi. Dimana pada menu Materi, Manomotion SDK digunakan untuk merotasi objek 3D alat transportasi secara virtual yang telah diproyeksikan melalui layar *smartphone*. Objek yang diproyeksikan pada layar *smartphone* akan berputar sesuai dengan gerakan tangan dari *user* atau pengguna. Gerakan tangan *user* dapat menggunakan *gesture* seperti pada pilihan memegang yaitu, *grab*, *pinch*, dan *point*. Tetapi terdapat perbedaan dengan *gesture* untuk memegang objek 3D, dimana pada rotasi objek 3D akan berputar sesuai gerakan tangan *user* ke kanan atau ke kiri dengan objek 3D yang tetap pada posisi tertentu. Sedangkan pada bagian memegang objek 3D menggunakan pilihan *gesture* dapat merubah posisi objek 3D sesuai gerakan tangan *user* atau pengguna. Posisi objek 3D dapat berubah dari segi sumbu X, Y, dan Z. dengan memanfaatkan *Scene* yang sudah disediakan oleh Manomotion sudah terdapat *Hand Tracker Manager* dimana digunakan untuk mengakses SDK sehingga gerakan tangan dapat dideteksi (Lam and Nilsson 2019).

```
public float speed = 40f;
void DisplayRotationGizmo(TrackingInfo tracking_info)
{
    if (Show_rotation)
    {
```

```

        float angle =
Mathf.LerpAngle(rotationIconTransform.rotation.z,
tracking_info.rotation, 0.1f);
        ....
        rotationObjectTransform.eulerAngles = new Vector3(0,
-angle * speed, 0);
        rotationValueText.text =
tracking_info.rotation.ToString();      }
        else
        {
            float angle =
Mathf.LerpAngle(rotationIconTransform.rotation.z, 0, 0.1f);
            ....
            rotationObjectTransform.eulerAngles = new Vector3(0,
angle * speed, 0);
            rotationValueText.text = "0";      }
    }

void SetRotationGizmoParts()
{
    ....
    rotationObjectTransform = Object.transform;
    ....
}

```

*Source code* diatas merupakan potongan dari fungsi yang digunakan untuk memutar objek 3D alat transportasi sesuai dengan gerakan tangan *user* atau pengguna. Fungsi tersebut memiliki parameter dengan tipe *TrackingInfo* dimana *TrackingInfo* merupakan *library* yang sudah disediakan oleh Manomotion SDK yang berisikan informasi mengenai posisi dan pelacakan tangan. Kemudian parameter tersebut akan diisi dengan informasi dan pelacakan tangan ketika *user* menggunakan aplikasi. Sehingga ketika objek dari tangan user terdeteksi oleh kamera perangkat (*smartphone*) dengan menggunakan pilihan *gesture grab*, *pinch*, atau *point*, maka objek 3D alat transportasi dapat diputar mengikuti gerakan tangan user ke arah kanan atau kiri. Keseluruhan dari *script rotation* akan diletakkan dalam *game object RotationManager* kemudian *parent* dari objek 3D alat transportasi dimasukkan kedalam *script rotation*. Untuk merotasi dari arah atas dan bawah berikut *source code* nya :

```

public float speed = 40f;
void DisplayRotationGizmo(TrackingInfo tracking_info)
{
    if (Show_rotation)
    {
        float angle =
Mathf.LerpAngle(rotationIconTransform.rotation.x,
tracking_info.rotation, 0.1f);
        ....
        rotationObjectTransform.eulerAngles = new Vector3(
-angle * speed, 0, 0);
    }
}

```

```

        rotationValueText.text =
tracking_info.rotation.ToString();      }
        else
        {
            float angle =
Mathf.LerpAngle(rotationIconTransform.rotation.x, 0, 0.1f);
            ....
            rotationObjectTransform.eulerAngles = new Vector3(
angle * speed, 0, 0);
            rotationValueText.text = "0";      }
        }

void SetRotationGizmoParts()
{
    ....
    rotationObjectTransform = Object.transform;
    ....
}

```

Sedangkan pada menu Permainan, Manomotion SDK digunakan untuk memegang objek 3D alat transportasi secara virtual yang telah diproyeksikan melalui layar *smartphone* dengan menggunakan pilihan memegang antara lain, *grab*, *pinch*, dan *point*. Manomotion SDK telah mengkategorikan *hand gesture* dalam tiga grup yang berbeda yang dinamakan Manoclasses, yaitu *grab*, *pinch*, dan *point* seperti pada Gambar 5.4. Pada menu Permainan ini ketiga *hand gesture* dibedakan menggunakan *button*. Apabila *user* atau pengguna ingin menggunakan *gesture grab*, maka diharuskan menekan *button grab* terlebih dahulu, begitu dengan *gesture pinch* dan *point*.



Gambar 5.4 Hand Gesture (a) Pinch (b) Grab (c) Point

```

public class InstantiateOnPalmCentre : MonoBehaviour
{
    public GameObject objectPrefab;
    void Update()
    {
        spwanOnPalmCentre (ManomotionManager.Instance.Hand_infos[0].hand_
info.tracking_info);
    }

    public void spwanOnPalmCentre (TrackingInfo trackingInfo)
    {
        Vector3 normalizedPalmCentre = trackingInfo.palm_center;
        float depth = trackingInfo.relative_depth;
    }
}

```

```

        Vector3          relativePalmCentrePosition          =
ManoUtils.Instance.CalculateNewPosition(normalizedPalmCentre,
depth);
        float smoothingVariable = 0.85f;

        objectPrefab.transform.position                    =
Vector3.Lerp(transform.position,          relativePalmCentrePosition,
smoothingVariable);
    }
}

```

Potongan *source code* diatas digunakan untuk menginisiasi *game object cube* yang digunakan untuk menggerakkan objek 3D alat transportasi pada menu Permainan. *Game object cube* akan muncul pada *palm center* atau posisi tengah dari tangan ketika tangan diarahkan ke depan kamera. Potongan *source code* diatas akan memunculkan *game object cube* pada *palm center* sesuai dengan informasi tangan (*hand info*) dan informasi dari pindaian tangan (*tracking info*) yang ditangkap oleh kamera dan diproses oleh *library*, kemudian *game object cube* tersebut akan berpindah sesuai gerakan tangan tetapi tetap pada posisi *palm center*. Posisi *palm center* digunakan dalam *hand gesture grab* atau menggenggam. Setelah *cube* muncul dan ketika *cube* diarahkan ke objek 3D alat transportasi, maka objek alat transportasi dapat bergerak mengikuti gerakan tangan.

```

public class GrabAndMove : MonoBehaviour
{
    void OnTriggerStay(Collider trigCol)
    {
        TrackingInfo          myTrackingInfo          =
ManomotionManager.Instance.Hand_infos[0].hand_info.tracking_info
;

        GestureInfo          myGestureInfo          =
ManomotionManager.Instance.Hand_infos[0].hand_info.gesture_info;

        grabMove(myTrackingInfo, myGestureInfo, trigCol);
    }
    void grabMove(TrackingInfo trackingInfo, GestureInfo
gesture_info, Collider trigCol)
    {
        // mengecek apakah tangan menggenggam
        if (gesture_info.state >= 10)

```

```

        {
            // mendapatkan posisi dari palm center
            Vector3 normalizedPalmCentre =
trackingInfo.palm_center;
            float depth = trackingInfo.relative_depth;
            Vector3 relativePalmCentrePosition =
ManoUtils.Instance.CalculateNewPosition(normalizedPalmCentre,
depth);
            // menggerakkan objek dengan tangan
            float smoothingVariable = 0.01f;
            gameObject.transform.position =
Vector3.Lerp(gameObject.transform.position,
relativePalmCentrePosition, smoothingVariable);
        }
    }
}

```

Berbeda dengan *tracking info* yang digunakan dalam *hand gesture grab*, pada *hand gesture pinch* menggunakan *tracking info* berupa *finger tips* atau ujung jari. Gerakan *pinch* menggunakan jari telunjuk dan ibu jari, sehingga *finger tips* yang digunakan yaitu *index* ke-4 dan ke-3 yang mempresentasikan *finger tips* dari jari telunjuk dan ibu jari sesuai dengan *library* Manomotion SDK. Sehingga *game object cube* akan muncul di ujung ketika jari telunjuk dan ibu jari terbentuk seperti gerakan mencubit. Kemudian dengan munculnya *game object cube*, objek 3D dapat digerakkan ketika *cube* menyentuh objek 3D alat transportasi.

```

public class InstantiateOnPinch : MonoBehaviour
{
    ....
    void spwanOnPinch(TrackingInfo trackingInfo)
    {
        Vector3 normalizedPinch = trackingInfo.finger_tips[4] =
trackingInfo.finger_tips[3];
        ....
    }
}
public class PinchAndMove : MonoBehaviour
{
    ....
}

```

```

void pinchMove(TrackingInfo trackingInfo, GestureInfo
gesture_info, Collider trigCol)
{
    // mengecek apakah gerakan tangan mencubit
    if (gesture_info.state >= 5)
    {
        // mendapatkan posisi dari ujung gerakan cubit
        Vector3 normalizedPinch = trackingInfo.finger_tips[4]
= trackingInfo.finger_tips[3];
        float depth = trackingInfo.relative_depth;
        Vector3 relativePinchPosition =
ManoUtils.Instance.CalculateNewPosition(normalizedPinch, depth);
        // menggerakkan objek dengan tangan
        float smoothingVariable = 0.85f;
        gameObject.transform.position =
Vector3.Lerp(gameObject.transform.position,
relativePinchPosition, smoothingVariable);
    }
}
}

```

Sama dengan *tracking info* yang digunakan dalam *hand gesture pinch*, pada *hand gesture point* menggunakan *tracking info* berupa *finger tips* atau ujung jari. Gerakan *point* hanya menggunakan jari telunjuk, sehingga *finger tips* yang digunakan yaitu *index* ke-3 saja yang mempresentasikan *finger tips* dari jari telunjuk sesuai dengan *library* Manomotion SDK. Sehingga *game object cube* akan muncul di ujung jari telunjuk. Kemudian dengan munculnya *game object cube* pada ujung jari telunjuk, objek 3D dapat digerakkan ketika *cube* menyentuh objek 3D alat transportasi.

```

public class InstantiateOnPoint : MonoBehaviour
{
    ....
    void spwanOnFinger(TrackingInfo trackingInfo)
    {
        Vector3 normalizedFinger = trackingInfo.finger_tips[3];
        float depth = trackingInfo.relative_depth;
    ....
    }
}

```

```

}
public class PointAndMove : MonoBehaviour
{
    ....
    void pointMove(TrackingInfo trackingInfo, GestureInfo
gesture_info, Collider trigCol)
    {
        // mengecek apakah gerakan tangan hanya jari telunjuk
        if (gesture_info.state >= 3)
        {
            // mendapatkan posisi dari ujung jari
            Vector3 normalizedFinger =
trackingInfo.finger_tips[3];
            float depth = trackingInfo.relative_depth;
            Vector3 relativeFingerPosition =
ManoUtils.Instance.CalculateNewPosition(normalizedFinger,
depth);
            // menggerakkan objek dengan tangan
            float smoothingVariable = 0.85f;
            gameObject.transform.position =
Vector3.Lerp(gameObject.transform.position,
relativeFingerPosition, smoothingVariable);
        }
    }
}

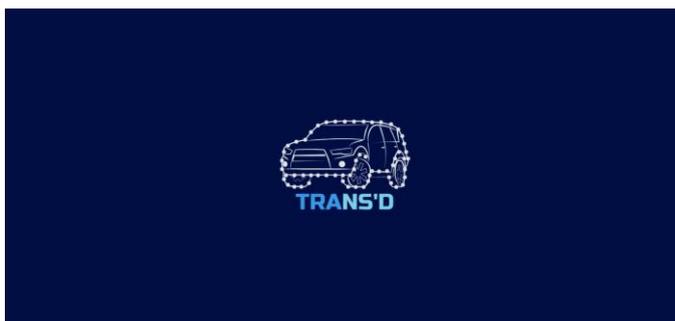
```

Seperti yang dijelaskan sebelumnya bahwa ketiga *hand gesture* dibedakan menggunakan *button*. Sehingga apabila *button grab* yang ditekan dan aktif, maka hanya *source code grab* yang aktif, sedangkan *source code* dari *pinch* dan *point* tidak aktif. Begitu ketika *button pinch* yang ditekan dan aktif, maka hanya *source code pinch* yang aktif, sedangkan *source code* dari *grab* dan *point* tidak aktif. Dan ketika *button point* yang ditekan dan aktif, maka hanya *source code point* yang aktif, sedangkan *source code* dari *grab* dan *pinch* tidak aktif. Hal tersebut dilakukan karena ketika ketiga *hand gesture* aktif secara bersamaan akan menimbulkan munculnya *game object cube* yang tidak akurat sesuai pada *hand gesture* yang digunakan.

## 5.2 Implementasi *User Interface*

Pada tahap implementasi *user interface* atau antarmuka berikut ini akan dijelaskan mengenai bagaimana aplikasi dibuat dan berjalan berdasarkan analisa dan perancangan yang dilakukan sebelumnya pada bagian perancangan *user interface* atau antarmuka.

### 1. *Splash Screen*



Gambar 5.5 Hasil Tampilan *Splash Screen*

Tampilan *Splash Screen* pada Gambar 5.5 akan muncul ketika aplikasi pertama kali dibuka oleh user (siswa). Tampilan tersebut memuat logo aplikasi dan nama aplikasi.

### 2. Menu Utama atau *Main Menu*



Gambar 5.6 Hasil Tampilan Menu Utama

Tampilan Menu Utama atau *Main Menu* pada Gambar 5.6 akan muncul setelah tampilan *splash screen* ditampilkan. Tampilan ini memuat tiga menu utama dengan tiga tombol yaitu tombol menu Materi, tombol menu Permainan, dan tombol Keluar dengan nama aplikasi di atasnya. Apabila *user* menekan tombol menu Materi, maka akan diarahkan oleh sistem untuk menampilkan menu Materi. Apabila *user* menekan tombol menu Permainan, maka akan diarahkan oleh sistem untuk menampilkan menu Permainan. Apabila *user* menekan

tombol menu Keluar, maka akan diarahkan oleh sistem untuk menampilkan menu Keluar. Kemudian terdapat tombol info pada pojok kanan bawah tampilan, ketika ditekan akan memunculkan *popup* berisikan informasi mengenai penulis seperti Gambar 5.7.



Gambar 5.7 Tampilan *Button* Info

### 3. Menu Materi



Gambar 5.8 Tampilan Menu Materi

Tampilan Menu Materi pada Gambar 5.8 akan muncul ketika *user* menekan tombol menu Materi pada menu utama. Sebelum tampilan pada Gambar 5.8 akan ditampilkan terlebih dahulu *popup* peringatan untuk menyalakan fitur kamera pada *smartphone* dengan menekan tombol *start* atau mulai pada pojok kanan atas tampilan. *Popup* peringatan tersebut ditunjukkan pada Gambar 5.9 di bawah ini.



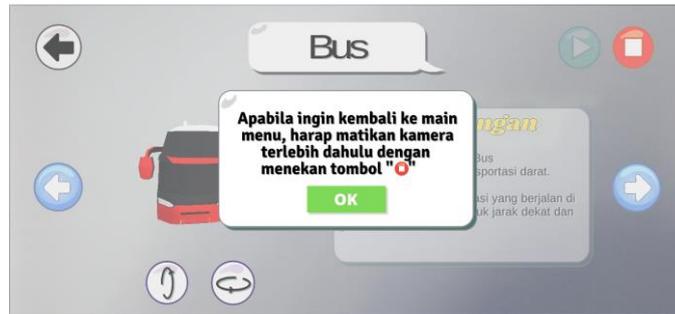
Gambar 5.9 Tampilan *Popup* Peringatan untuk *Start* Kamera

Setelah user menekan tombol “OK” pada popup tersebut, maka user akan diarahkan untuk menekan tombol *start* terlebih dahulu. Karena tombol *start* berfungsi untuk menyalakan fitur kamera pada *smartphone* serta memberikan instruksi lebih lanjut melalui *popup*. Berikut tampilan ketika tombol *start* ditekan oleh user.



Gambar 5.10 Tampilan Instruksi pada Menu Materi

Pada tampilan tersebut memiliki informasi berupa instruksi lebih lanjut untuk mengoperasikan menu Materi dan memiliki dua tombol yaitu tombol “Lewati” yang berfungsi untuk menutup *popup* instruksi dan mengarahkan *user* pada tampilan Gambar 5.8 sehingga *user* bisa mengakses menu Materi dan tombol “Selanjutnya” untuk instruksi lebih lanjut. Gambar 5.8 juga memiliki tombol *stop* dan tombol kembali, dimana tombol *stop* berfungsi untuk mematikan fitur kamera pada *smartphone* dan tombol kembali berfungsi untuk mengarahkan *user* kembali ke menu utama. Ketika *user* menekan tombol kembali sebelum mematikan fitur kamera pada *smartphone*, maka *user* akan ditampilkan *popup* peringatan untuk menekan tombol *stop* terlebih dahulu sebelum menekan tombol kembali seperti di bawah ini.



Gambar 5.11 Tampilan *Popup* Peringatan untuk *Stop* Kamera

Setelah *user* menekan tombol “OK” pada *popup* tersebut, *user* diharuskan menekan tombol *stop* untuk dapat menekan tombol kembali dan menuju ke menu utama. Pada Gambar 5.8 terdapat *button* selanjutnya dan *button* sebelumnya yang direpresentasikan dengan *button* panah ke arah kanan untuk selanjutnya dan ke arah kiri untuk sebelumnya yang keduanya berfungsi untuk memindah objek 3D beserta materi yang ada dalam Menu Materi seperti pada Gambar 5.12 berikut.



Gambar 5.12 Tampilan Menu Materi Setelah Menekan *Button* Selanjutnya

*Button start camera* dan *stop camera* selain digunakan untuk menyalakan dan mematikan fitur kamera pada perangkat *smartphone*, tetapi juga digunakan agar kamera dari perangkat *smartphone* tidak menyala terus menerus yang dapat mengakibatkan *force close* pada aplikasi ketika berpindah menu atau *scene*. Karena ketika tidak diberikan *button start* dan *stop* kamera, kamera *device* akan otomatis menyala ketika *scene* pertama kali dibuka atau ketika saat masuk ke menu Materi atau Permainan dan mengecek apakah Manomotion SDK terinisialisasi atau tidak kemudian menghasilkan *true*, kemudian ketika

keluar dari *scene*, kamera *device* tetap memiliki status menyala dan menghasilkan *true*. Hal tersebut menyebabkan *force close* karena setelah keluar dari *scene* kemudian masuk *scene* lainnya akan diarahkan bahwa Manomotion SDK menghasilkan *false* untuk inisialisasi. Sehingga penulis menambahkan potongan *source code* untuk menghindari hal tersebut terjadi seperti di bawah ini:

```

public class ManomotionManager : MonoBehaviour
{
    ....
    protected void Start()
    {
        PauseWebCamTexture();
        InstantiateSession();
        StartCamera();

        PickResolution(STARTING_WIDTH, STARTING_HEIGHT);
        SetUnityConditions();
        _initialized = true;
    }
    ....
    protected void StartWebCamTexture()
    {
        PickResolution(STARTING_WIDTH, STARTING_HEIGHT);
        _mCamera = new
WebCamTexture(WebCamTexture.devices[0].name, _width, _height);
        _mCamera.requestedFPS = 60;
        _mCamera.Play();
        Debug.Log("Webcam is playing");
    }
    protected void PauseWebCamTexture()
    {
        _mCamera = new
WebCamTexture(WebCamTexture.devices[0].name, _width, _height);
        //_mCamera.requestedFPS = 60;
        _mCamera.Stop();
        Debug.Log("Webcam is pausing");
    }
    public void StartCamera()

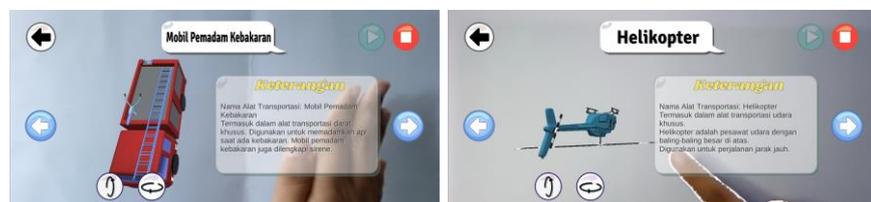
```

```

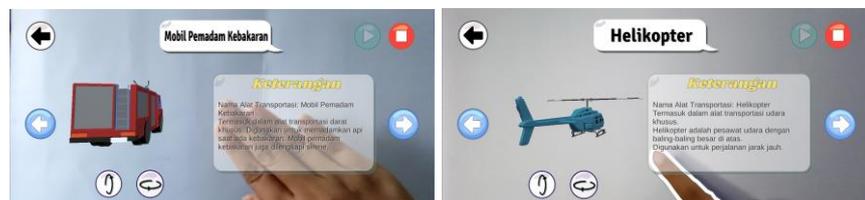
{
    startPanel.SetActive(true);
}
public void StopCamera()
{
    stopPanel.SetActive(true);
}
}

```

Menu materi juga memiliki dua tombol untuk merotasi objek dari atas dan bawah serta kanan dan kiri yang berada di bawah objek 3D dari alat transportasi. Berikut hasil apabila merotasi objek 3D sesuai pada Gambar 5.13 dan 5.14:



Gambar 5.V.13 Tampilan Rotasi Atas dan Bawah



Gambar 5.14 Tampilan Rotasi Kanan dan Kiri

#### 4. Menu Permainan



Gambar 5.15 Tampilan Menu Permainan

Tampilan Menu Permainan pada Gambar 5.15 akan muncul ketika *user* menekan tombol menu Permainan pada menu utama. Sebelum tampilan pada Gambar 5.15 akan ditampilkan terlebih dahulu *popup* peringatan untuk menyalakan fitur kamera pada *smartphone* dengan

menekan tombol *start* atau mulai pada pojok kanan atas tampilan. *Popup* peringatan tersebut ditunjukkan pada Gambar 5.16 di bawah ini.



Gambar 5.16 Tampilan *Popup* Peringatan untuk *Start* Kamera

Setelah user menekan tombol “OK” pada popup tersebut, maka user akan diarahkan untuk menekan tombol *start* terlebih dahulu. Karena tombol *start* berfungsi untuk menyalakan fitur kamera pada *smartphone* serta memberikan instruksi lebih lanjut melalui popup. Berikut tampilan ketika tombol *start* ditekan oleh user.



Gambar 5.17 Tampilan Instruksi pada Menu Permainan



Gambar 5.18 Tampilan Instruksi pada Menu Permainan Lanjutan

Pada Gambar 5.17 tersebut memiliki informasi berupa instruksi lebih lanjut untuk mengoperasikan menu Permainan dan memiliki satu tombol yaitu tombol “Lewati” yang berfungsi untuk menutup *popup* instruksi dan mengarahkan *user* pada tampilan Gambar 5.15 sehingga

*user* bisa mengakses menu Permainan. Pada Gambar 5.17 terdapat *button* Selanjutnya yang akan mengarahkan ke instruksi lanjutan dalam Menu Permainan seperti pada Gambar 5.18. Ketika *user* mengakses menu Permainan ketika menggerakkan objek 3D alat transportasi kemudian objek tersebut sudah untuk digerakkan maka *user* dapat menekan *button Refresh* pada pojok kanan atas untuk mengembalikan posisi objek 3D ke posisi awal seperti pada Gambar 5.19 dan Gambar 5.20.



Gambar 5.19 Tampilan Ketika *User* Menggerakkan Objek 3D



Gambar 5.20 Tampilan Ketika *User* Menekan *Button Refresh*

Apabila jawaban *user* benar dari pertanyaan yang ada maka akan muncul *popup* yang menandakan bahwa jawaban tersebut benar serta apabila jawaban *user* salah dari pertanyaan yang ada maka akan muncul *popup* yang menandakan bahwa jawaban tersebut salah seperti pada Gambar 5.21 dan Gambar 5.22 di bawah ini.



Gambar 5.21 Tampilan *Popup* Jawaban Benar



Gambar 5.22 Tampilan *Popup* Jawaban Salah

Munculnya *popup* tersebut dikarenakan ketika *collider* dari objek jawaban menyentuh *collider* dari objek kotak. Ketika *game object* jawaban dengan *tag Correct* menyentuh *collider* kotak, maka *popup* jawaban benar akan muncul. Ketika *collider* dari *game object* jawaban dengan *tag Wrong* menyentuh *collider* kotak, maka *popup* jawaban salah akan muncul.

```
public class DetectCollision : MonoBehaviour
{
    public GameObject correctFeedback;
    public GameObject wrongFeedback;

    void OnTriggerEnter(Collider col)
    {
        if (col.gameObject.tag == "Correct")
        {
            Destroy(col.gameObject);
            correctFeedback.SetActive(true);
            wrongFeedback.SetActive(false);
        }
        else if (col.gameObject.tag == "Wrong")
        {
            Destroy(col.gameObject);
            correctFeedback.SetActive(false);
            wrongFeedback.SetActive(true);
        }
    }
    void OnTriggerExit(Collider col)
    {
        if (col.gameObject.tag == "Correct")
```

```

{
    correctFeedback.SetActive(true);
    wrongFeedback.SetActive(false);
}
else if (col.gameObject.tag == "Wrong")
{
    correctFeedback.SetActive(false);
    wrongFeedback.SetActive(true);
}
}
}

```

Gambar 5.21 dan Gambar 5.22 memiliki masing-masing dua tombol pada *popup*, yaitu tombol “Main Lagi” dan tombol “Keluar”. Dimana tombol “Main Lagi” akan mengarahkan *user* kembali ke menu Permainan dengan soal yang berbeda, sedangkan tombol “Keluar” mengarahkan *user* kembali ke menu utama.

Gambar 5.15 juga memiliki tombol stop dan tombol kembali, dimana tombol *stop* berfungsi untuk mematikan fitur kamera pada *smartphone* dan tombol kembali berfungsi untuk mengarahkan *user* kembali ke menu utama. Ketika *user* menekan tombol kembali sebelum mematikan fitur kamera pada *smartphone*, maka *user* akan ditampilkan *popup* peringatan untuk menekan tombol *stop* terlebih dahulu sebelum menekan tombol kembali seperti di bawah ini.



Gambar 5.23 Tampilan *Popup* Peringatan untuk *Stop* Kamera

Setelah *user* menekan tombol “OK” pada *popup* tersebut, *user* diharuskan menekan tombol *stop* untuk dapat menekan tombol kembali dan menuju ke menu utama. *Button start* dan *stop* kamera digunakan

untuk menghindari *force close* dari aplikasi seperti penjelasan sebelumnya.

## 5. Menu Keluar



Gambar 5.24 Tampilan Popup Peringatan pada Menu Keluar

Tampilan menu keluar pada Gambar 5.24 akan ditampilkan ketika *user* menekan tombol menu Keluar pada menu utama, dimana ketika tombol ditekan akan memunculkan *popup* peringatan apakah ingin keluar dari aplikasi atau tidak dengan memiliki dua tombol, yaitu tombol “Tidak” dan “Ya”. Ketika tombol “Tidak” ditekan, maka *user* akan diarahkan kembali pada menu utama, sedangkan ketika tombol “Ya” ditekan, maka *user* akan diarahkan keluar aplikasi.

## 5.3 Pengujian

Untuk mengetes kelayakan aplikasi ini penulis menggunakan dua metode, yaitu metode pengujian *alpha* (*alpha test*) dan metode pengujian beta (*beta test*). Dimana pengujian *alpha* dimaksudkan untuk menguji fungsionalitas dari aplikasi yang telah dibuat. Pada pengujian *alpha* ini penulis menggunakan metode *black-box testing* untuk menguji apakah aplikasi dapat berjalan dengan baik tanpa gangguan atau tidak. Aplikasi dinyatakan lolos dalam pengujian *alpha* apabila semua fungsi dalam aplikasi berjalan dengan baik dan benar sesuai dengan fungsinya. Sedangkan pengujian beta dimaksudkan untuk mendapatkan umpan balik dari pengguna terhadap aplikasi yang telah dibuat. Pengujian beta dilakukan dengan pengambilan data menggunakan metode *survey* berupa kuesioner kepada target pengguna yaitu siswa kelas 3 SDN Lawang 05 dan siswa kelas 3 SD dari tempat les Bu Lis. Metode kuesioner yang digunakan yaitu metode kuesioner tertutup dimana responden sudah dipersiapkan jawaban ya dan tidak dari

pertanyaan singkat mengenai keakuratan tiga pilihan memegang benda dan pertanyaan mengenai pilihan memegang mana yang lebih disukai.

### 5.3.1 Pengujian *Alpha* (*Alpha Test*)

Pengujian *alpha* pada tahap pengujian ini akan dilakukan dengan menguji fungsionalitas dari aplikasi dengan metode *black-box*.

Tabel 5.1 Tabel *Test Scenario*

<b>Test Scenario #</b>	<b>Test Scenario ID</b>	<b>Test Scenario Description</b>
1.	TS011	Membuka Menu Materi
2.	TS012	Kembali ke <i>Main Menu</i> dalam Menu Materi
3.	TS013	Menyalakan Kamera dalam Menu Materi
4.	TS014	Mematikan Kamera dalam Menu Materi
5.	TS015	Menampilkan Objek 3D Selanjutnya dalam Menu Materi
6.	TS016	Menampilkan Objek 3D Sebelumnya dalam Menu Materi
7.	TS017	Memutar Objek 3D dalam Menu Materi dengan Tangan
8.	TS018	Menampilkan Instruksi dalam Menu Materi
9.	TS021	Membuka Menu Permainan
10.	TS022	Kembali ke <i>Main Menu</i> dalam Menu Permainan
11.	TS023	Menyalakan Kamera dalam Menu Permainan
12.	TS024	Mematikan Kamera dalam Menu Permainan
13.	TS025	Menampilkan Instruksi dalam Menu Permainan
14.	TS031	Memindahkan Objek 3D Menggunakan <i>Grab</i>
15.	TS032	Memindahkan Objek 3D Menggunakan <i>Pinch</i>
16.	TS033	Memindahkan Objek 3D Menggunakan <i>Point</i>
17.	TS034	Mengembalikan Posisi Objek 3D ke Posisi Semula
18.	TS035	Menampilkan Popup Jawaban Benar
19.	TS036	Menampilkan Popup Jawaban Salah

Test Scenario #	Test Scenario ID	Test Scenario Description
20.	TS037	Kembali ke Permainan Setelah Bermain
21.	TS038	Kembali ke <i>Main Menu</i> Setelah Bermain
22.	TS041	Membuka Menu Keluar
23.	TS042	Keluar dari Aplikasi
24.	TS043	Kembali ke Aplikasi

Pada sub bab analisis masalah dijelaskan bahwa faktor cahaya mempengaruhi keakuratan dari *hand gesture*, sehingga penulis melakukan pengujian terhadap faktor cahaya tersebut dengan bantuan aplikasi *lux meter* yang dapat menilai intensitas dari cahaya. Berikut merupakan tabel dari pengujian faktor cahaya :

Tabel 5.2 Tabel Pengujian Faktor Cahaya

Cahaya (lx)	Keakuratan <i>Hand Gesture</i>		
	<i>Grab</i>	<i>Pinch</i>	<i>Point</i>
0 lx (Sangat Redup)	Tidak Baik	Tidak Baik	Tidak Baik
5 lx (Sangat Redup)	Tidak Baik	Tidak Baik	Tidak Baik
25 lx (Redup)	Tidak Baik	Tidak Baik	Tidak Baik
50 lx (Redup)	Tidak Baik	Kurang Baik	Kurang Baik
100 lx (Terang)	Kurang Baik	Sangat Baik	Baik
550 lx (Terang)	Kurang Baik	Sangat Baik	Sangat Baik
1000 lx (Sangat Terang)	Tidak Baik	Baik	Baik

Pada Tabel 5.2 bagian keakuratan *hand gesture* skala “Tidak Baik” mengacu pada *hand gesture* yang tidak dapat berjalan dan tidak dapat memindahkan objek dengan akurat. Skala “Kurang Baik” mengacu pada *hand gesture* yang dapat berjalan tetapi perpindahan objek tidak stabil dan tidak akurat. Skala “Baik” mengacu pada *hand gesture* yang dapat berjalan dan dapat memindahkan objek tetapi sedikit tidak akurat. Skala “Sangat Baik” mengacu pada *hand gesture* yang dapat berjalan dengan baik dan dapat memindahkan objek dengan stabil dan akurat.

Berikut merupakan tabel statistik dari keakuratan serta kesuksesan dari *hand gesture* yang telah diuji dengan jumlah 10 kali percobaan menghasilkan bahwa

*hand gesture grab* memiliki tingkat kesuksesan sebesar 60% dengan 6 kali percobaan berhasil dan 4 kali percobaan gagal. Kemudian *hand gesture pinch* memiliki tingkat kesuksesan sebesar 100% dengan 10 kali percobaan berhasil dan tanpa percobaan yang gagal. *Hand gesture point* memiliki tingkat kesuksesan sebesar 80% dengan 8 kali percobaan berhasil dan 2 kali percobaan gagal sesuai dengan Tabel 5.3.

Tabel 5.3 Tabel Statistik Keakuratan dan Tingkat Kesuksesan *Hand Gesture*

<b>Hand Gesture</b>	<b>Jumlah Percobaan</b>	<b>Frekuensi Kesuksesan</b>	<b>Frekuensi Kegagalan</b>	<b>Tingkat Kesuksesan</b>
Grab	10	6	4	60%
Pinch	10	10	0	100%
Point	10	8	2	80%

Sedangkan untuk pengujian fungsionalitas aplikasi atau *Test Case* dapat dilihat pada Lampiran 1 Tabel 5.4 Tabel Pengujian Fungsionalitas Aplikasi (*Test Case*) .

### 5.3.2 Pengujian Beta (*Beta Test*)

Pengujian beta pada tahap ini dilakukan dengan pengambilan data menggunakan metode *survey* berupa kuesioner kepada target pengguna yaitu siswa kelas 3 SDN Lawang 05 dan dari tempat les Bu Lis. Berikut kuesioner yang diberikan kepada target pengguna yaitu siswa kelas 3 SDN Lawang 05 dan dari tempat les Bu Lis:

Tabel 5.5 Kuesioner Pengujian Beta (*Feedback*)

No.	Pertanyaan	Pilihan Jawaban	
		Ya	Tidak
1.	Apakah aplikasi dapat berjalan dengan baik?		
2.	Apakah anda menyukai aplikasi ini sebagai media pembelajaran?		
3.	Apakah anda sudah mengetahui tentang alat transportasi darat dan udara? (Setidaknya 5 alat transportasi darat dan udara)		

No.	Pertanyaan	Pilihan Jawaban	
		Ya	Tidak
4.	Setelah menggunakan aplikasi ini apakah pengetahuan anda tentang alat transportasi darat dan udara menjadi bertambah?		
5.	Apakah pilihan memegang benda dengan nama “Grab” dapat berjalan dengan baik?		
6.	Apakah pilihan memegang benda dengan nama “Pinch” dapat berjalan dengan baik?		
7.	Apakah pilihan memegang benda dengan nama “Point” dapat berjalan dengan baik?		
8.	Apakah anda menyukai pilihan memegang benda dengan nama “Grab”?		
9.	Apakah anda menyukai pilihan memegang benda dengan nama “Pinch”?		
10.	Apakah anda menyukai pilihan memegang benda dengan nama “Point”?		

Tabel 5.6 Kuesioner Tambahan Pengujian Beta

No.	Pertanyaan	Pilihan Jawaban	
		Ya	Tidak
1.	Dari menu materi berapa alat transportasi yang sudah anda ketahui? (1 - 11)		
2.	Jumlah jawaban benar setelah 11 kali percobaan pada menu permainan?		
3.	Jumlah jawaban salah setelah 11 kali percobaan pada menu permainan?		