BAB III

ANALISA DAN PERANCANGAN

3.1. Analisa Kebutuhan Software

Analisis ini merupakan penguraian dari suatu sistem yang utuh ke dalam bagianbagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan solusi atau perbaikan. Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien. Sistem yang dibuat merupakan aplikasi untuk mendeteksi marker dan menampilkan objek tiga dimensi yang telah dibuat dengan menggunakan *software* tiga dimensi (Autodesk Maya). Objek yang dibuat merupakan bangunan apartemen, seolah-olah pengguna berinteraksi langsung dengan objek virtual dalam dunia nyata yang disajikan dalam bentuk brosur.

3.1.1. Identifikasi Permasalahan

1. Bagaimana merancang dan membuat aplikasi augmented reality brosur interaktif

berbasis android?

2. Bagaimana membuat aplikasi *augmented reality* berbasis *Android* agar nanti dapat digunakan sebagai media pemasaran *interaktif* dan menarik perhatian pelanggan dan dapat membantu menyampaikan informasi kepada pelanggan secara lengkap?

3. Apa yang pertama kali menarik minat pelanggan ketika melihat sebuah brosur?

4. Masalah apa saja yang anda hadapi ketika melihat sebuah gambar produk pada sebuah

brosur?

3.1.2. Rumusan Masalah

Berdasarkan latar belakang masalah dan identifikasi masalah yang telah di jelaskan di bab sebelumnya maka dapat dirumusan masalah sebagai berikut : perlu dibangun media pemasaran *interaktif* dimana dengan media pemasaran ini bisa menarik minat seseorang pembeli, dan dengan media ini penjual lebih mudah menyampaikan informasi seputar produk lengkap dengan bentuk-bentuk produk yang ditawarkan kepada pelanggan

3.1.3. Analisa Kebutuhan

1. Kebutuhan Perangkat Keras

Spesifikasi perangkat keras (*hardware*) yang direkomendasikan untuk menggunakan aplikasi *augmented reality*, yaitu :

- a. Processor: Quad-core 1.3GHz Cortex-A7
- b. Random Access Memory(RAM): 1 GB
- c. Memori: microSD 32 GB
- d. Input: LCD capacitive touchscreen, 16M colors
- e. Output: Layar 540 x 960 pixels(200ppi pixel density)
- f. Kamera: 13MP, 4128 x 3096 pixels

2. Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak (*software*) yang direkomendasikan untuk menggunakan aplikasi AR adalah sistem operasi android 4.4.2 Kitkat

3.2. Desain

3.2.1. Rancangan Algoritma pada kasus

using UnityEngine; using System.Collections; using System.IO;

public class UserInterfaceButtons : MonoBehaviour

{

```
public float scalingSpeed = 0.03f;
public float rotationSpeed = 70.0f;
public float translationSpeed = 5.0f;
```

```
// public GameObject Model;
bool repeatScaleUp = false;
bool repeatScaleDown = false;
bool repeatRotateLeft = false;
bool repeatRotateRight = false;
bool repeatPositionUp = false;
bool repeatPositionDown = false;
bool repeatPositionLeft = false;
bool repeatPositionRight = false;
```

```
void Update ()
{
    if (repeatScaleUp) {
        ScaleUpButton ();
    }
```

```
if (repeatScaleDown) {
              ScaleDownButton ();
       }
       if (repeatRotateRight) {
              RotationRightButton();
       }
       if (repeatRotateLeft) {
              RotationLeftButton();
       }
       if (repeatPositionUp) {
              PositionUpButton();
       }
       if (repeatPositionDown) {
              PositionDownButton();
       }
       if (repeatPositionLeft) {
              PositionLeftButton();
       }
       if (repeatPositionRight) {
              PositionRightButton();
       }
public void CloseAppButton ()
```

}

{

```
Application.Quit ();
```

```
}
```

```
public void RotationRightButton ()
```

{

```
// transform.Rotate (0, -rotationSpeed * Time.deltaTime, 0);
```

```
GameObject.FindWithTag ("Model").transform.Rotate (0, -rotationSpeed *
```

Time.deltaTime, 0);

```
GameObject.FindWithTag ("Modell").transform.Rotate (0, -rotationSpeed * Time.deltaTime, 0);
```

}

```
public void RotationLeftButton ()
```

{

// transform.Rotate (0, rotationSpeed * Time.deltaTime, 0);

GameObject.FindWithTag ("Model").transform.Rotate (0, rotationSpeed * Time.deltaTime, 0);

```
GameObject.FindWithTag ("Modell").transform.Rotate (0, rotationSpeed * Time.deltaTime, 0);
```

```
}
```

```
public void RotationRightButtonRepeat ()
```

{

```
// transform.Rotate (0, -rotationSpeed * Time.deltaTime, 0);
repeatRotateRight=true;
```

```
}
```

```
public void RotationLeftButtonRepeat ()
```

```
// transform.Rotate (0, rotationSpeed * Time.deltaTime, 0);
repeatRotateLeft=true;
```

```
}
```

{

```
public void ScaleUpButton ()
```

{

```
// transform.localScale += new Vector3(scalingSpeed, scalingSpeed,
```

scalingSpeed);

```
GameObject.FindWithTag ("Model").transform.localScale += new
Vector3 (scalingSpeed, scalingSpeed);
```

GameObject.FindWithTag ("Modell").transform.localScale += new Vector3 (scalingSpeed, scalingSpeed);

```
}
```

```
public void ScaleUpButtonRepeat ()
{
    repeatScaleUp = true;
```

```
Debug.Log ("Up");
```

}

{

}

{

}

}

{

```
public void ScaleDownButtonRepeat ()
```

```
repeatScaleDown = true;
```

```
Debug.Log ("Down");
```

public void PositionDownButtonRepeat ()

```
repeat Desition Down - to
```

```
repeatPositionDown = true;
```

```
public void PositionUpButtonRepeat ()
```

```
{
```

```
repeatPositionUp = true;
```

```
public void PositionLeftButtonRepeat ()
```

```
repeatPositionLeft = true;
```

```
}
public void PositionRightButtonRepeat ()
```

40

```
{
       repeatPositionRight = true;
}
public void ScaleUpButtonOff ()
{
       repeatScaleUp = false;
       Debug.Log ("Off");
}
public void ScaleDownButtonOff ()
{
       repeatScaleDown = false;
       Debug.Log ("Off");
}
public void RotateLeftButtonOff ()
{
       repeatRotateLeft = false;
       Debug.Log ("Off");
}
public void RotateRightButtonOff ()
{
       repeatRotateRight = false;
       Debug.Log ("Off");
}
public void PositionRightButtonOff ()
{
       repeatPositionRight = false;
       Debug.Log ("Off");
}
public void PositionLeftButtonOff ()
```

```
repeatPositionLeft = false;
             Debug.Log ("Off");
       }
      public void PositionUpButtonOff ()
       {
             repeatPositionUp = false;
             Debug.Log ("Off");
       }
      public void PositionDownButtonOff ()
      {
             repeatPositionDown = false;
             Debug.Log ("Off");
       }
      public void ScaleDownButton ()
      {
             // transform.localScale += new Vector3(-scalingSpeed, -scalingSpeed, -
scalingSpeed);
             GameObject.FindWithTag ("Model").transform.localScale += new Vector3 (-
scalingSpeed, -scalingSpeed);
             GameObject.FindWithTag ("Modell").transform.localScale += new Vector3 (-
scalingSpeed, -scalingSpeed);
      }
      public void PositionUpButton ()
      {
             GameObject.FindWithTag ("Model").transform.Translate (0, 0, -
translationSpeed * Time.deltaTime);
       }
      public void PositionDownButton ()
       {
```

```
GameObject.FindWithTag ("Model").transform.Translate (0, 0,
translationSpeed * Time.deltaTime);
       }
       public void PositionRightButton ()
       {
              GameObject.FindWithTag ("Model").transform.Translate (-translationSpeed *
Time.deltaTime, 0, 0);
       }
       public void PositionLeftButton ()
       {
              GameObject.FindWithTag ("Model").transform.Translate (translationSpeed *
Time.deltaTime, 0, 0); // backward
       }
       public void ChangeScene (string a)
       {
              Application.LoadLevel (a);
       }
       public void AnyButton ()
       {
              Debug.Log ("Any");
       }
}
```

3.2.2. Database

Vuforia adalah *Augmented Reality Software Development Kit* (SDK) untuk perangkat mobile yang memungkinkan pembuatan aplikasi AR. SDK Vuforia juga tersedia untuk digabungkan dengan Unity 3D adalah bernama Vuforia AR Extension for Unity. Vuforia merupakan SDK yang disediakan oleh Qualcomm untuk membantu para *developer* membuat aplikasi-aplikasi *Augmented Reality* (AR) di *mobile phones* (iOS, Android).

Terdapat dua jenis workflow dengan dasar database yang dapat dipilih oleh developer, adalah

- 1. Cloud Database
- 2. *Device* Database

seperti ditunjukan pada Gambar III.1 Unity 3D dibawah adalah sebuah *game engine* yang berbasis cross platform. Unity3D dapat digunakan untuk membuat sebuah *game* yang bisa digunakan pada perangkat komputer, smartphone, iPhone, PS3, dan bahkan X-BOX. Unity3D *Game Engine* tidak hanya merupakan sebuah *game Engine*, tapi juga merupakan sebuah editor.



Gambar III.1. Diagram Database Vuforia

Sumber : (Domhan dan Tobias, 2010)

3.2.3. Software Architecture

a. Pseudocode Algorithma

transform.Rotate (0, -rotationSpeed * Time.deltaTime, 0);

GameObject.FindWithTag ("Model").transform.Rotate (0, -rotationSpeed * Time.deltaTime, 0);

GameObject.FindWithTag ("Modell").transform.Rotate (0, -rotationSpeed * Time.deltaTime, 0);

}

// Melakukan Scale

transform.localScale += new Vector3(-scalingSpeed, -scalingSpeed);

GameObject.FindWithTag ("Model").transform.localScale += new Vector3 (scalingSpeed, -scalingSpeed);

GameObject.FindWithTag ("Modell").transform.localScale += new Vector3 (-

scalingSpeed, -scalingSpeed);

}

//Melakukan screen shot

mainCamera = Camera.main.GetComponent<Camera> (); // for Taking Picture

renderTex = new RenderTexture (width, height, 24);

mainCamera.targetTexture = renderTex;

RenderTexture.active = renderTex;

mainCamera.Render ();

screenshot = new Texture2D (width, height, TextureFormat.RGB24,

false);

screenshot.ReadPixels (new Rect (0, 0, width, height), 0, 0);

screenshot.Apply (); //false

RenderTexture.active = null;

mainCamera.targetTexture = null

// Gesture tangan

}

Lean.LeanTouch.MoveObject(transform, Lean.LeanTouch.DragDelta);

}

b. Mekanisme Sistem



Gambar III.2. Mekanisme sistem

Pada gambar diatas dapat dijelaskan bahwa pertama kali user melakukan scanning marker yang ada pada brosur dengan menggunakan aplikasi yang terdapat pada *smartphone* Android miliknya, selanjutnya aplikasi akan mengenali pola atau point tracker yang ada pada marker serta mencocokkan pola marker tersebut terhadap gambar 3D yang telah diupload ke server qualcomm sebelumnya, apabila pola markernya yang di scan cocok dengan pola gambar 3D maka aplikasi akan memberikan respon berupa penampilan gambar 3D pada *smartphone user*

b. Diagram UML

Perancangan UML terdiri dari *use case* yang menjelaskan fungsi sistem dari sudut pandang *user* kemudian dijelaskan dalam bentuk *activity diagram* yang menggambarkan alur proses dari setiap *use case, class diagram* yang menggambarkan objek atau kelas dari sistem serta hubungan antar objek-objek kelas dan *sequence diagram* yang menggambarkan interaksi *message* dengan objek-objek sesuai urutan.

1. Use Case Diagram



Gambar III.3. use case diagram

Sumber : hasil penelitian

Pada gambar dijelaskan bahwa user bisa menggunakan aplikasi dengan cara:

- a. Untuk memulai user membuka aplikasi
- b. User melakukan deteksi marker dengan menggunakan smartphone android
- c. *User* bisa melihat secara langsung bentuk sebuah gedung apartemen dan melihat bentukbentuk interior ruangan
- d. User bisa menggeser position objek kearah yang kita inginkan
- e. User bisa memutar objek sampai 360 derajat
- f. User bisa melakukan screen capture dengan menggunakan button yang telah disediakan
- g. User bisa melakukan scale
- h. User bisa melakukan penutupan aplikasi

2. Activity Diagram

a. Menampilkan Objek



Gambar III.4. Activity Diagram Menampilkan Objek

Sumber : Hasil penelitian

Untuk menjalankan aplikasi *user* terlebih dahulu harus membuka aplikasi dengan cara menekan *icon* aplikasi setelah itu aplikasi akan mengaktifkan kamera *smartphone, User* bisa melakukan deteksi *marker* dengan meletakkan *marker* secara tegak lurus atau memenggang secara tegak lurus dan mengarahkan kamera *smartphone* pada *marker* selanjutnya kamera yang sudah aktif bisa mendeteksi objek dan aplikasi akan menampilkan objek 3D pada layar *smartphone*

b. Rotate Objek



Gambar III.5. Activity Diagram Rotate Objek

Sumber : Hasil penelitian

User bisa me-rotate objek dengan menekan button rotate yang telah ada, jika user mau me-rotate objek kekiri atau kekanan user cukup dengan menekan button rotate kanan atau *button rotate* kiri selanjutnya aplikasi yang akan memutar objek kekiri atau kekanan sesuai button yang ditekan

e. Scale Objek



Gambar III.6. Activity Diagram Scale Objek

Sumber : Hasil penelitian

User bisa me-*scale* objek dengan menekan *button zoom* yang telah ada, jika *user* mau memperbesar objek *user* cukup dengan menekan *button zoom in* dan begitu juga sebaliknya jika *user* mau memperkecil objek *user* cukup dengan menekan *button zoom out*

f. Position objek



Gambar III.7. Activity Diagram Position objek

Sumber : Hasil penelitian

User bisa merubah *position* objek dengan melakukan *gesture* tangan lalu aplikasi akan mendeteksi tangan selanjutnya *user* menggeser *position* objek kesegala arah berdasarkan keinginan *user* maka aplikasi akan merespon dengan pergerakan objek kearah yang diinginkan oleh *user*

g. Melakukan Screen Capture



Gambar III.8. Activity Diagram Melakukan Screen Capture

Sumber : Hasil Penelitian

User melakukan *screen capture* dengan menekan *button screen capture* lalu aplikasi mengambil gambar objek lalu menyimpan pada *smartphone*

h. Keluar dari aplikasi



Gambar III.9. Activity Diagram Keluar Dari Aplikasi

Sumber : Hasil penelitian

User bisa menutup atau keluar dari aplikasi dengan menekan *button exit* lalu aplikasi akan menutup atau keluar dengan sendirinya secara langsung

3. Sequence Diagram

a. Memilih Rotate



Gambar III.10. Sequence Diagram Memilih Rotate

Sumber : Hasil Penelitian

User berada di halaman main menu, lalu *user* menekan *button rotate* yang terdapat pada aplikasi setelah itu *system* menerima *signal* atau perintah, kemudian *user* me-*rotate* objek kekanan dan kekiri maka secara otomatis *system* akan merespon perintah dengan cara me-*rotate* objek 3D sesuai arah yang kita inginkan

b. Melakukan scale objek



Gambar III.11. Sequence Diagram Melakukan Scale Objek

Sumber : Hasil penelitian

Menjelaskan tentang proses user me-*scale* objek, untuk melakukannya *user* harus menekan *button scale* yang sudah disediakan setelah itu *system* akan merespon dengan cara membesar atau memperkecil objek sesuai dengan *button* yang *user* tekan

c. Memindahkan position objek



Gambar III.12. Sequence Diagram Memindahkan Position Objek

Sumber : Hasil penelitian

User mengarahkan *gesture* tangan ke layar *smartphone* lalu sistem akan mendeteksi *gesture* tangan, kemudian *user* memindahkan *position* objek maka sistem akan memindahkan *position* objek sesuai arah tangan *user*



d. Melakukan screen capture atau print screen

Gambar III.13. Sequence Diagram Melakukan Screen Capture

Sumber : Hasil penelitan

Menjelaskan tentang proses *user* melakukan *screen capture* atau *screen shot* pada objek yang muncul dengan cara menekan *button Screen capture* yang sudah disediakan setelah itu sistem akan merespon dengan mengambil potongan gambar dari objek tersebut

e. Keluar dari aplikasi



Gambar III.14. Sequence Diagram Keluar Dari Aplikasi

Sumber : Hasil Penelitian

User bisa keluar dari aplikasi dengan menekan *button exit*, setelah menekan aplikasi akan tertutup dengan sendirinya tanpa ada konfirmasi selanjutnya

3.2.4. User Interface



Gambar III.15. User Interface Splash Image

Sumber : Hasil Penelitian

Layar splash screen terdiri dari logo aplikasi AR dan nama aplikasi.



Gambar III.16. User Interface Layout Aplikasi

Sumber : Hasil Penelitian

Pada gambar diatas dijelaskan bahwa aplikasi menggunakan beberapa button diantaranya:

1. Button scale

berfungsi untuk memperbesar atau memperkecil gambar

2. Button screen capture

berfungsi untuk mengambil gambar

3. Button rotate

berfungsi untuk memutar gambar atau objek pada layar smartphone

4. Button exit

berfungsi untuk keluar dari aplikasi

3.3 Implementasi



Gambar III.17. Implementasi Splash Image

Sumber : Hasil Penelitian

Layar splash screen terdiri dari logo aplikasi AR dan nama aplikasi.



Gambar III.18. Implementasi Tampilan Aplikasi

Sumber : Hasil Penelitian

Tampilan interface pada aplikasi AR beserta button dimana semua button telah berfungsi



Gambar III.19. Implementasi Tampilan Gedung

Sumber : Hasil Penelitian

Pada gambar diatas adalah gambar hasil pengimplementasian terhadap aplikasi augmented reality, pada gambar tersebut bisa dilihat bentuk bangunan apartemen beserta lingkungan secara interaktif



Gambar III.20. Implementasi Tampilan Interior Ruangan

Sumber : Hasil Penelitian

Pada gambar diatas adalah gambar hasil pengimplementasian terhadap aplikasi *augmented reality*, pada gambar tersebut bisa dilihat bentuk interior ruangan apartemen secara *interaktif*

3.4 Testing

3.4.1. Pengujian White Box

Hasil pengujian *white box* testing aplikasi sebagai berikut:



Gambar III.21. Flowchart Pengujian White Box

Sumber : Penelitian langsung



Gambar III.22. Flowchart Angka Pengujian White Box

Sumber : Penelitian langsung

Kompleksitas iklomatis dari grafik alir *white box* dapat diperoleh dengan perhitungan sebagai beikut :

V(G) = E - N + 2

Dimana

E =Jumlah *Edge* yang ditentukan gambar panah

N = Jumlah simpul grafik alir ditentukan dengan gambar lingkaran

Sehingga didapat :

V(G) = 20 - 15 + 2 = 7

V(G) < 10 berarti memenuhi kekomplesitas siklomatisnya

Baris set yang dihasilkan dari jalur independent adalah sebagai berikut :

a. 1-2-3-4-5-6-15
b. 1-2-3-4-5-6-7-14-6-15
c. 1-2-3-4-5-6-7-8-13-6-15
d. 1-2-3-4-5-6-7-8-9-12-6-15
e. 1-2-3-4-5-6-7-8-9-10-11-6-15
f. 1-2-3-4-5-6-7-8-9-10-15
g. 1-2-3-4-3-4-5-6-15

Ketika aplikasi dijalankan, maka terlihat bahwa satu set baris yang dihasilkan adalah 1-2-3-4-5-6-1-2-3-4-5-6-7-14-6-1-2-3-4-5--7-8-13-6-1-2-3-4-5-6-7-8-9-12-6-1-2-3-4-5-7-8-9-10-11-6-1-2-3-4-5-6-7-8-9-10-6-1-2-3-4-3-4-5-6-15 dan terlihat bahwa simpul telah dieksekusi satu kali

3.4.2. Pengujian *Black box*

No	Skenario uji	Test Case	Hasil yang	Keterangan
			diharapkan	
1	Scan marker	Menampilkan objek 3d bangunan	Tampil objek 3d bangunan	Valid
2	button Rotate	Menekan button rotate	Objek 3D bangunan berputar	Valid
3	Button scale	Menekan button scale	Objek membesar dan mengecil	Valid
4	Screen capture	Menekan button Screen	Gambar tesimpan	Valid
5	Memilih keluar	Keluar dari menu utama	Tampil halamankeluar	Valid

Tabel III.1. Pengujian Black Box

3.5. Support

Pada aplikasi ini penulis menggunakan *Smartphone Android* yang akan digunakan menjalankan aplikasi yang telah dibuat dan menguji aplikasi ini agar aplikasi tersebut sudah benar-benar dapat dioperasikan sebagaimana fungsinya. Seperti halnya pada perangkat PC pada *Smartphone Android* juga memiliki spesifikasi minimal agar aplikasi berjalan optimal. Berikut spesifikasi minimal pada perangkat *smartphone android* yang digunakan.

Kebutuhan	Keterangan
Tipe	LG G3 Stylus
SistemOperasi	android 4.4.2 Kitkat
Processor	Quad-core 1.3GHz Cortex-A7
RAM	1 GB

Tabel III.2. Support