

IMPLEMENTASI *AUGMENTED REALITY* UNTUK BROSUR INTERAKTIF DENGAN METODE FAST CORNER DETECTION

Nelly Otavia Adiwijaya¹, Yuni Wahyu Hapdini², Antonius C.P³

Sistem Informasi Program Ilmu Komputer Universitas Jember
Jl. Kalimantan 37, Jember 68121
E-mail: nelly.oa@unej.ac.id

Abstrak

Penerapan teknologi *augmented reality* (AR) dapat dimanfaatkan untuk berbagai hal salah satunya untuk memberi *user experience* (UX) yang berbeda pada suatu aplikasi. Penelitian ini menambahkan teknologi AR pada sebuah brosur untuk bahan promo. Aplikasi brosur interaktif ini dibuat dengan menggunakan software Unity 3D dengan bahasa pemrograman C#. Aplikasi yang dibangun mampu memberikan visualisasi 3D, gambar, teks maupun video sesuai dengan *marker* yang terdapat pada brosur PSSI UNEJ versi cetak. Selama dilakukan penelitian ini ditemukan bahwa penggunaan aplikasi harus memperhatikan jarak dan fokus kamera dengan *marker*, dimana semakin dekat jaraknya semakin baik mendeteksi *marker* dan semakin jauh akan mengganggu kinerja aplikasi. Semakin tinggi *rating* dari target gambar maka semakin mudah mendeteksi *marker* tetapi objek tidak tampil apabila ada pantulan cahaya atau bayangan *smartphone* saat mendeteksi *marker*. Persentase maksimum rata-rata *occlusion* agar objek tetap tampil pada *marker* yaitu 50%. Kecepatan dan waktu akses video secara *online* tergantung pada lokasi, bandwidth, dan penyedia jaringan yang digunakan oleh pengguna.

Key word : maksimum 6 keyword

1. Pendahuluan

Penerapan teknologi *Augmented Reality* saat ini banyak dipakai dalam bidang periklanan, karena memang memberikan *user experience* yang menarik. Sarana promosi sebaiknya dapat berupa media cetak, atau media elektronik dan juga media digital. Melalui teknologi AR, sarana promosi dapat mengkombinasikan media cetak dengan media digital [4].

Brosur merupakan salah satu alat publikasi suatu instansi atau perusahaan yang berupa cetakan dan berisi informasi produk, layanan, atau program yang ditujukan kepada sasaran tertentu. Brosur cetakan tersebut berisi rangkuman keterangan lengkap sehingga mudah dipahami dalam waktu singkat. Brosur ditujukan untuk memperkenalkan secara terperinci mengenai produk, layanan, serta program dalam membantu promosi.

Desain brosur yang kreatif dan unik dapat menarik dan merebut perhatian penerima. Informasi produk, layanan, atau program yang terdapat dalam brosur dapat menginformasikan, mengedukasi, dan membujuk atau mempengaruhi penerima untuk membeli atau mengadopsi pesan yang disampaikan.

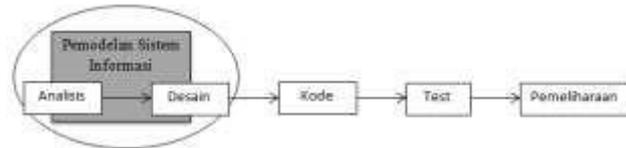
Implementasi teknologi AR pada penelitian ini bekerja berdasarkan deteksi *marker* yang digunakan sebagai objek acuan pada dunia nyata yaitu brosur dan animasi sebagai objek dari dunia virtual yang ditambahkan pada lingkup dunia nyata. Penggabungan teknik AR pada brosur dapat memberikan *user*

experience yang berbeda pada penggunaannya. Hal ini juga dapat menjadi nilai tambah pada brosur untuk lebih menarik perhatian karena AR memungkinkan pengguna untuk melihat lingkungan nyata, dengan objek virtual yang ditambahkan atau tergabung dengan lingkungan nyata. Pada layar *smartphone* ditampilkan objek 3D maupun video yang dapat membuat promosi lebih menarik karena penyampaian informasi tidak hanya dengan tulisan, suara, maupun objek 2D. Misalnya, dengan menampilkan objek 3D dari gedung beserta ruangan dan fasilitas didalamnya yang dapat dilihat dari dekat dan nyata sesuai dengan keadaan aslinya, serta video berbagai kegiatan.

Brosur sebagai sarana promosi juga dimanfaatkan oleh Program Studi Sistem Informasi (PSSI) Universitas Jember. Biasanya brosur dibagikan kepada siswa-siswi sekolah menengah atas didalam maupun diluar Kabupaten Jember. Brosur yang digunakan berupa gambar dua dimensi disertai tulisan. Pengembangan brosur interaktif dengan menerapkan teknologi AR dapat menjadi alat promosi yang lebih menarik dan interaktif. Brosur juga dapat digunakan untuk menunjukkan produk teknologi yang dihasilkan oleh PSSI. Tujuan dikembangkan brosur interaktif adalah untuk menarik dan memberikan contoh penerapan teknologi terkini pada siswa – siswi sekolah menengah atas. Harapannya siswa lebih tertarik untuk melanjutkan studi di Program Studi Sistem Informasi Universitas Jember yang selalu *update* dengan penerapan teknologi terkini.

2. Metodologi

Pengembangan yang digunakan dalam penelitian ini yaitu SDLC (*System Development Life Cycle*) dengan model *waterfall*. Model *waterfall* merupakan metode yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, test, dan pemeliharaan[9]. Tahapan model *waterfall* digambarkan pada gambar 1.



Gambar 1. Tahapan Model Waterfall[14]

Proses *Augmented Reality* (AR) dari awal inialisasi, *tracking object*, sampai dengan proses memunculkan informasi teks dan obyek 3D. Dalam perancangan aplikasi dengan teknologi AR, menggabungkan objek virtual dengan objek nyata, dalam hal ini objek virtual berupa objek 3D dan objek nyatanya berupa marker yang terdapat pada brosur yang ditangkap oleh kamera ponsel dengan pola tertentu (*markerless*)[7]. Analisis alur aplikasi dapat dilihat pada gambar 2.



Gambar 2. Diagram Alur Aplikasi *Augmented Reality*

Berdasarkan gambar 3 alur aplikasi pada analisis aplikasi ini sebagai berikut:

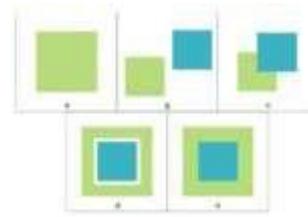
- Inisialisasi feature dan rating pada *marker*
- Tracking image target*
- Menampilkan objek
- Aksi pada objek

Metode yang bekerja pada implementasi teknologi AR brosur ini adalah *Occlusion detection* dan *Fast Corner Detection*.

Occlusion adalah hubungan antara suatu benda dengan benda lain jika kita lihat dari suatu sudut pandang. Hal ini tentunya mengurangi informasi antar objek dalam lingkungan 3D, karena jika dilihat dari sudut pandang maka lingkungan 3D akan diproyeksikan kepada suatu bidang sehingga seolah – olah menjadi lingkungan 2D. Pengurangan dimensi ini menyebabkan informasi interaksi antar objek seperti keadaan bersinggungan, beririsan atau berapa jarak antara objek akan menjadi ambigu [1].

Occlusion detection adalah metode untuk mendeteksi ada tidaknya *occlusion* dalam penampilan objek 3D. Pada [2] secara sederhana *occlusion detection* hanya mendefinisikan keadaan dimana suatu marker tidak terdeteksi karena tertutup oleh benda lain.

Sedangkan pada [8] menggunakan *occlusion detection* berdasarkan posisi koordinat 2D dari dua objek yang ada.



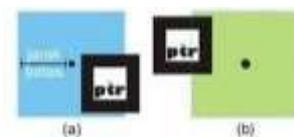
Gambar 3. *Occlusion* yang Terjadi Karena Interaksi Antara Objek [9] (a)None (b)Proximity (c) Intersection (d) Enclosure (e) Containment

Pada gambar 3 jika ada n objek yang diwaliki matrik maka akan dihasilkan matrik O_1, O_2, \dots, O_n yang merupakan posisi proyeksi objek-1, objek-2, ..., objek-n dilayar. Deteksi *occlusion* akan dilakukan dengan pengecekan 2 objek misal dipilih objek-1 terhadap objek-2 maka akan dilakukan pengecekan syarat pertidaksamaan *point of clipping* berikut [5]:

$$O_2x - batas \leq O_1x \leq O_2x + batas \quad (1)$$

$$O_2y - batas \leq O_1y \leq O_2y + batas \quad (2)$$

Hasil deteksi ini berupa nilai kebenaran yang merupakan dasar pendefinisian *event* dari interaksi *occlusion based* jika persamaan 1 dan 2 terpenuhi.



Gambar 4. (a) Terjadi Deteksi (b) Tidak Terjadi Deteksi

Pada gambar 4 apabila titik biru ditengah *marker ptr* adalah objek O_1 dan titik hitam adalah objek O_2 maka gambar (a) dikatakan terjadi *event* karena memenuhi pertidaksamaan 1 dan 2 yaitu koordinat $O_1(x,y)$ ada di dalam batas area O_2 . Sedangkan gambar (b) tidak terjadi *event* karena hanya memenuhi pertidaksamaan 2 (nilai O_1y ada dalam batas O_2y) namun tidak memenuhi pertidaksamaan 1.

Persentasi nilai maksimum *occlusion* pada marker agar objek pada bidang AR *lost tracked* karena terhalang oleh suatu benda padat [6].

$$\frac{A_{occlusion}}{A_{total}} \times 100 = \%occlusion \quad (3)$$

Dimana:

$A_{occlusion}$ = persentasi *marker* terhalang oleh benda

A_{total} = luas pada *marker*

Algoritma *FAST Corner detection*

FAST (Feature Form Accelerated segment Test)

Corner Detection adalah suatu algoritma yang dikembangkan oleh Edward Rosten, Reid Porter, and Tom Drummond. *FAST Corner detection* ini dibuat dengan tujuan mempercepat waktu komputasi secara real-time dengan konsekuensi menurunkan tingkat akurasi pendeteksian sudut [8].

Pada FAST Corner detection, proses penentuan corner point-nya adalah dengan cara merubah gambar menjadi warna hitam-putih dan menjalankan algoritmanya. Algoritma ini menentukan corner point dengan menentukan sebuah titik yakni p dari gambar masukan dengan 16 pixel disamping p diperiksa seperti yang tertera pada gambar. Ada 3 kasus berbeda yang ditetapkan untuk masing-masing perbandingan yakni dijelaskan pada formula 4 dengan perumusan titik p .

$$C = \begin{cases} |I_p - I_n| < t & \text{normal} \\ I_n - I_p > t & \text{brighter} \\ I_p - I_n > t & \text{darker} \end{cases} \quad 4)$$

I_p pada gambar 1 menunjukkan intensitas titik pusat. I_n adalah titik intensitas tetangga ke- n . Poin p ditandai sebagai sudut, jika setidaknya 9, 10, 11 atau 12 piksel yang berdekatan lebih terang atau lebih gelap dari p .

3. Desain dan Perancangan Aplikasi

Penerapan teknologi *Augmented Reality* (AR) pada aplikasi brosur interaktif Program Studi Sistem Informasi Universitas Jember berbasis android digambarkan secara garis besar dalam kebutuhan fungsional dan non- fungsional, *business process*, *usecase diagram*.

1. Kebutuhan Fungsional

Kebutuhan fungsional aplikasi berisi fitur-fitur inti yang harus dipenuhi dalam aplikasi. Kebutuhan fungsional aplikasi adalah sebagai berikut:

- Aplikasi mampu memberikan visualisasi 3D dari brosur cetak Program Studi Sistem Informasi Universitas Jember sesuai perspektif arah kamera mobile Android.
- Aplikasi mampu melakukan control object 3D yaitu rotasi, *scale* dan *move*.
- Aplikasi mampu menampilkan video profil Program Studi Sistem Informasi Universitas Jember.
- Aplikasi mampu menampilkan video dari kegiatan Program Studi Sistem Informasi Universitas Jember.
- Aplikasi mampu mengunduh video.
- Aplikasi mampu memberikan informasi berupa gambar dan teks.
- Aplikasi mampu membuka *URL website*.

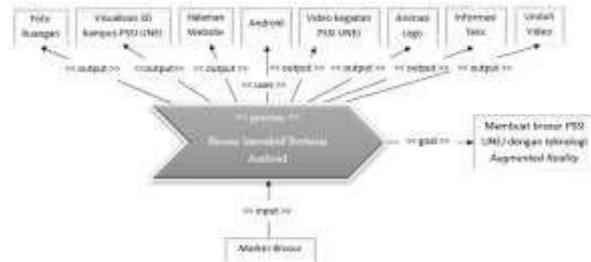
2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan

kebutuhan tambahan untuk melengkapi aplikasi. Kebutuhan non fungsional aplikasi sebagai berikut:

- Aplikasi dijalankan pada perangkat seluler berbasis android.
 - Aplikasi menggunakan *library vuforia*.
3. *Business process*

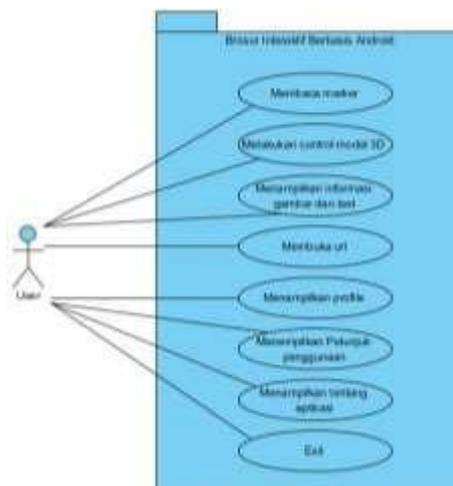
Business process digunakan untuk menggambarkan sebuah proses suatu aplikasi secara keseluruhan, lengkap dengan *resources* dan informasi yang dibutuhkan seperti *input* dan *output* aplikasi, sehingga mendorong terjadinya proses dan tujuan yang telah ditentukan, seperti pada gambar 5



Gambar 5. Business Proses

4. Use case diagram

Use case diagram merupakan model atau diagram yang menggambarkan fungsi atau tugas yang dilakukan user, baik manusia maupun mesin/komputer. *Use case* digambarkan dari beberapa aktor, *use-case*, dan interaksi diantara komponen tersebut yang dapat memberikan informasi dari aplikasi yang akan dibangun. Pada gambar 6 digambarkan *use case diagram* aplikasi brosur interaktif yang terdiri dari satu aktor dengan delapan *use case*.



Gambar 6. Use Case Diagram

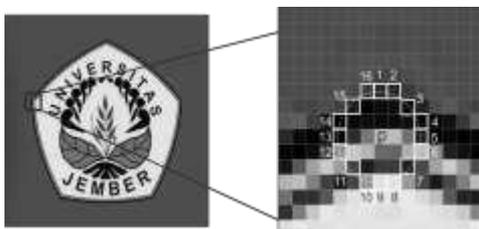
4. Pembahasan

Hasil pembuatan sistem dalam penelitian ini

menghasilkan penerapan teknologi *Augmented Reality* (AR) pada aplikasi brosur interaktif Program Studi Sistem Informasi Universitas Jember berbasis android. Teknologi AR memungkinkan user untuk melihat lingkungan nyata, dengan objek virtual yang ditambahkan atau tergabung dengan lingkungan nyata. Pada layar smartphone ditampilkan objek 3D maupun video yang dapat membuat promosi lebih menarik karena penyampaian informasi tidak hanya dengan tulisan, suara, maupun objek 2D tetapi dengan menampilkan objek 3D dari gedung beserta ruangan dan fasilitas didalamnya yang dapat dilihat dari dekat dan nyata sesuai dengan keadaan aslinya, menampilkan informasi kompetensi lulusan, serta menampilkan video profil dan video kegiatan mahasiswa.

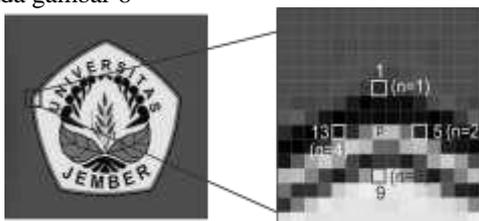
1. Implementasi Algoritma FAST *Corner detection*

- a) Menentukan sebuah titik p pada citra dengan posisi awal (x_p, y_p) seperti pada gambar 7.



Gambar 7. Titik p pada Citra dengan Posisi Awal (x_p, y_p)

- b) Menentukan keempat titik. Titik pertama $(n=1)$ terletak pada koordinat (x_p, y_p+3) , titik kedua $(n=2)$ terletak pada koordinat (x_p+3, y_p) , titik ketiga terletak pada koordinat (x_p, y_p-3) , titik keempat $(n=4)$ terletak pada koordinat (x_p-3, y_p) seperti pada gambar 8



Gambar 8. Keempat Titik Koordinat

- c) Bandingkan intensitas titik pusat p dengan keempat titik disekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat persamaan 5, maka titik pusat p adalah titik sudut.

$$C_p = \begin{cases} 1, & I_n < I_p - t \text{ or } I_n > I_p + t \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Dengan:

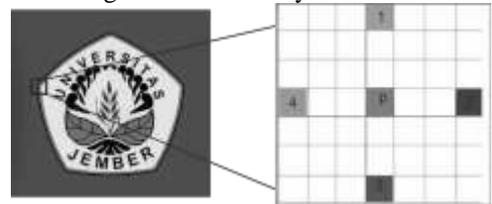
C_p = Keputusan titik p sebagai sudut, nilai 1 menunjukkan bahwa titik merupakan suatu sudut, dan nilai 0 menunjukkan bahwa titik bukanlah sudut

I_n = nilai intensitas piksel ke-n I_p =

nilai intensitas titik p

t = batas ambang nilai intensitas yang ditoleransi

- d) Ulangi proses sampai seluruh titik pada citra sudah dibandingkan intensitasnya



Gambar 9. Deteksi Tepi Pertitik Intensitas

Pada gambar 9 merupakan proses pendeteksian tepi pertitik intensitas, proses ini terus diulang sampai seluruh titik pada sebuah citra dipenuhi dan *point marker* akan didapat.

2. Proses Pembuatan Marker

Marker yang digunakan pada analisis ini didapatkan dari gambar yang terdapat dalam brosur seperti pada lampiran G. Proses pembuatan gambar *marker* menjadi *marker* dilakukan oleh pihak *vuforia* dengan cara mengunggah gambar yang akan di *convert*, melalui *marker engine* yang disediakan oleh pihak *vuforia*. Setelah gambar di *convert* menghasilkan file dengan format *unitypackage*. File tersebut kemudian dijadikan masukan pada *Engine Unity 3D* untuk mendeteksi gambar yang dijadikan *marker*. Berikut gambar 10 yaitu tahapan atau alur pembuatan *marker*.



Gambar 10. Alur Pembuatan Marker

Pada gambar 10 menjelaskan alur *flowchart* pada pembuatan *marker*, proses ini memiliki satu masukan, tujuh tahap proses dan satu keluaran yaitu gambar yang sudah menjadi *marker*.

- a) *Resize*

Setelah mengunggah gambar ke situs *vuforia* maka system secara otomatis akan memperkecil gambar ke ukuran 320*320, dijelaskan pada gambar 11.

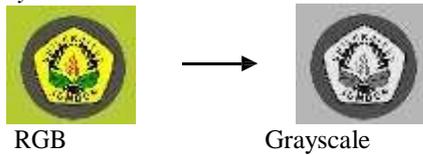


Gambar 11. Proses *Resize Marker*

Gambar 11 merupakan proses *resize marker* ke ukuran 320*320 dimana ukuran ini sangat ideal untuk pembacaan *point marker*, dari ukuran berapapun setelah gambar masuk kedalam tahap *converting* akan mengalami perubahan ukuran.

b) *Grayscale*

Grayscale merupakan proses mengubah gambar menjadi monokrom dengan entitas 0 untuk hitam, 255 untuk putih, dan abu-abu untuk nilai yang berada pada rentang nilai 0 sampai 255. Dimana rumus secara garis besarnya adalah $new\ pixel = (Red + Green + Blue) / 3$, dijelaskan pada gambar 12 *converting* dari RGB ke *grayscale*.

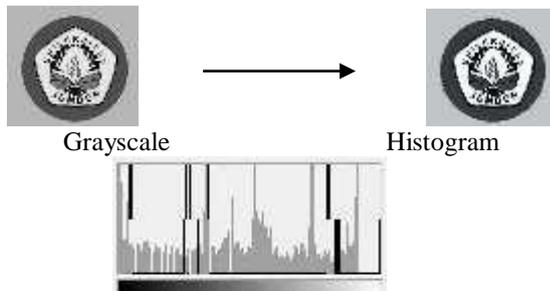


Gambar 12. Proses *Grayscale Marker*

Gambar 12 menjelaskan proses *grayscale* pada gambar, proses ini berawal dari RGB dan di ubah menjadi warna hitam putih bertujuan untuk memperhalus gambar sehingga memudahkan pencarian titik *point*. Setelah selesai *convert* ke ukuran yang diperlukan lalu *vuforia* merubah pewarnaan awal yaitu *grayscale*, perubahan warna ini merupakan awal dari tiga langkah berikutnya.

Histogram

Proses *Histogram* adalah suatu proses perataan, dimana distribusi derajat keabuan pada suatu gambar dibuat rata. Gambar 13 menjelaskan *converting grayscale ke histogram*.



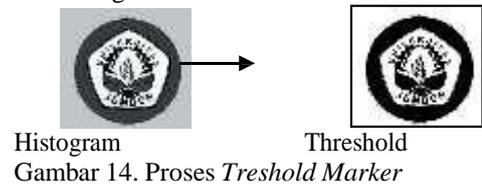
Gambar 13. Proses *Histogram Marker*

Histogram pada gambar 13 adalah proses

penyakinan gambar abu-abu, proses ini untuk mempertajam suatu *marker* sehingga titik point menjadi lebih konsisten dalam penitikkannya. Perubahan *histogram* merupakan langkah kedua untuk proses meratakan gambar dimana distribusi gambar keabuan menjadi rata sehingga penitikan *marker* awal menjadi terlihat.

d) *Ambang Batas (threshold)*

Dalam library *Vuforia* sebelum objek di *convert* keadaan *point - point marker* yang dapat menghasilkan titik sudut *marker*, *marker* diubah kedalam bentuk *threshold*. Proses *threshold* ini yang menjadikan acuan untuk proses selanjutnya, yaitu proses penentu titik *marker*. Berikut contoh gambar 14 menunjukkan perubahan dari gambar asli ke *threshold*.



Gambar 14. Proses *Threshold Marker*

Pada proses terakhir terdapat *convert* dari *histogram* ke *threshold* dimana proses ini merupakan *finishing vuforia* terhadap *marker* sehingga titik dan *point* ditemukan.

e) *Penentuan Point marker*

Gambar *Threshold* kemudian diproses untuk menentukan letak *point marker* dan pencocokan pola menggunakan metode *Fast Corner detection*. Dari bentuk gambar *threshold* maka selanjutnya menyimpan titik *point* ke dalam gambar *threshold* tersebut yang bertujuan untuk menentukan banyaknya jumlah dari titik *point*, berikut contoh gambar 15 tentang penentuan *point marker*.



Gambar 15. Penentuan *Point marker*

Semakin banyak bintang (*marker detection*) terdapat pada *marker*, maka semakin baik pola tersebut dijadikan *marker*, sehingga kamera ponsel dapat dengan cepat mendeteksi *marker*. Terlihat pada gambar 5.9 bintang yang didapatkan pada *marker* menunjukkan banyak bintang sehingga *ARCamera* dapat membacanya dengan jelas dan memudahkan *user* untuk menampilkan 3D pada *marker*.

3. Implementasi Teknologi *Augmented Reality*

Proses pembuatan program merupakan proses pengaplikasian tahap desain ke dalam bahasa pemrograman agar menjadi aplikasi yang mempunyai fungsionalitas sesuai dengan yang diharapkan. Dalam tahap pembuatan program ini terdapat beberapa langkah yang dilakukan. Langkah-langkah tersebut antara lain sebagai berikut.

a) Pembuatan Desain Interface

Dalam tahap ini *software* yang digunakan adalah Corel Draw X5. *Software* ini digunakan untuk mendesain semua *asset* dalam media yang meliputi *background*, tombol, isi halaman tutorial, isi halaman profil, isi halaman tentang, dan lain-lain. Fitur-fitur yang terdapat di dalam aplikasi brosur interaktif yaitu halaman awal (*splash screen*), menu utama, AR *brochure*, *profile*, *how to use*, *about*, dan *exit* seperti pada gambar 16.



Gambar 16. Halaman Menu Utama

b) Inisialisasi *Feature* dan *Rating Marker*

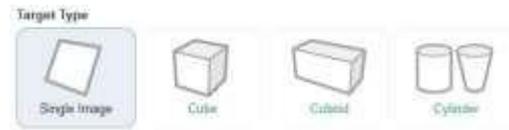
Tahapan sebuah gambar yang akan dijadikan *marker* menggunakan library *vuforia* sebagai pendukung untuk membuat aplikasi *Augmented Reality* dan *unity3D*. Pada dasarnya *vuforia* sudah menyediakan *tools* sebagai inisialisasi *feature* pada *marker* yang bernama *target manager system*. Berfungsi memberikan *feature* dan *rating* pada gambar yang dijadikan *marker*. Fungsi utama dari *target manager* ialah memudahkan *developer* yang menggunakan library *vuforia* untuk membangun sebuah aplikasi AR. Pihak *vuforia* memberikan kemudahan pada *developer* untuk membuat sebuah *marker* yang akan digunakan sebagai acuan munculnya sebuah objek pada *marker*. Gambar 17 menunjukkan alur dari bagaimana membuat sebuah penanda pada *target manager*.



Gambar 17. Alur Inisialisasi *Feature* dan *Rating Marker*

Deskripsi dari alur pada gambar 17 adalah sebagai berikut:

1. *Developer* memilih gambar dengan ekstensi JPG dan PNG dengan maksimal gambar 2.25MB.
2. Melakukan *upload* pada *target management system* dan memilih tipe target yang digunakan dapat dilihat pada gambar 18.



Gambar 18. Tipe Target yang Dapat Digunakan

3. *Developer* mengunduh ekstensi editor yang digunakan sebagai penggunaan pada IDE. Ekstensi yang disediakan oleh *vuforia* ada 2 yaitu *unitypackage* dan *zip*.

Berikut merupakan implementasi inisialisasi *feature* dan *rating* pada gambar yang digunakan pada aplikasi ini. Hasil dari *feature* pada gambar yang digunakan dapat dilihat pada gambar 19 dan gambar 20.



Gambar 19. *Feature* dan *Rating*

Target Name	Type	Rating	Created	Last Modified
Image001	Single Image	5.0	Apr 22, 2016 10:22	Apr 22, 2016 10:22
Image002	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image003	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image004	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image005	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image006	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image007	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image008	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image009	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image010	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image011	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image012	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image013	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image014	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image015	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image016	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image017	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image018	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image019	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21
Image020	Single Image	5.0	Apr 22, 2016 10:21	Apr 22, 2016 10:21

Gambar 20. *Rating Marker*

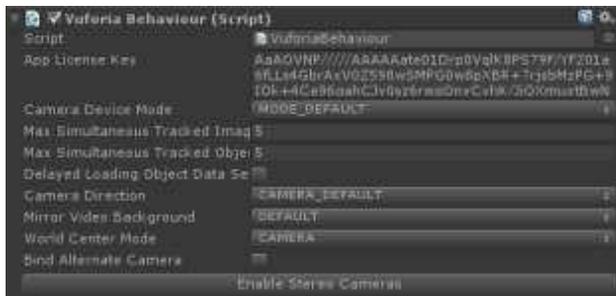
Rating marker dapat berkisar dari nol sampai lima untuk setiap gambar yang diberikan. Semakin tinggi *rating* dari target gambar, semakin kuat kemampuan deteksi dan pelacakan yang dikandungnya. Sebuah *rating* dari nol menunjukkan bahwa target tidak dilacak sama sekali, sedangkan *rating* bintang lima menunjukkan bahwa sebuah gambar dengan mudah dilacak oleh aplikasi *Augmented Reality*. Terdapat tiga belas *marker* pada aplikasi brosur interaktif, delapan *marker* memiliki *rating* bintang empat dan lima *marker* memiliki *rating* bintang lima.

Tingginya *rating* sebuah *marker* dipengaruhi

oleh beberapa kategori Semakin besar *contrast* pada citra tersebut maka semakin baik. Semakin seimbang pembagian fitur dalam gambar maka semakin baik gambar dapat dideteksi dan dilacak. *Feature* yang tidak sama dengan yang lainnya. Biasanya tidak terdapat background atau background gambar hanya mempunyai warna hitam dan putih. *Keypoint* yang tidak berulang, apabila berulang dapat menyebabkan proses *tracking* tidak baik karena persamaan posisi *keypoint*.

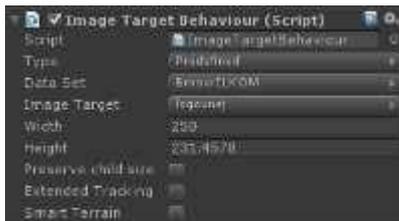
c) Pembuatan Program

Tahapan pembuatan program merupakan implementasi tahap desain ke dalam bahasa pemrograman. Dalam penelitian ini peneliti menggunakan bahasa pemrograman C#. Dalam tahap pembuatan program ini terdapat beberap langkah yang dilakukan. Pembuatan menu diawali dengan pembuatan *project* baru pada lembar kerja Unity 3D kemudian menyiapkan *plane* objek sebagai tempat background, dan membuat tombol-tombol. Selanjutnya *GameObject* ditambahkan sebagai *action* dari tombol untuk mengatur posisi dan tindakan yang terjadi jika tombol ditekan. Langkah pembuatan menu AR *Brochure* yaitu pada saat *ARCamera* pada Unity3D mengarah pada sebuah *image target*. *VuforiaBehaviour* akan melakukan pengecekan dengan nilai maksimal *image target* yang akan di deteksi. Apabila nilai yang ditentukan tidak sesuai, maka *ARCamera* hanya akan mendeteksi sesuai dengan nilai yang ditentukan. Tampilan *inspector* *VuforiaBehaviour* dapat dilihat pada gambar 21.



Gambar 21. Tampilan *Inspector VuforiaBehaviour* pada Unity3D

Selanjutnya *ARCamera* akan mengecek *image target* yang aktif sebagai acuan munculnya nya sebuah objek pada *marker*. Apabila sudah aktif sesuai gambar 22 selanjutnya penerapan gambar untuk dijadikan *image target* diaktifkan.



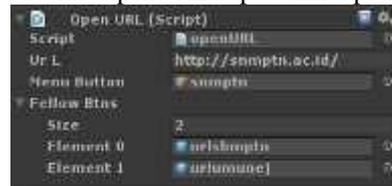
Gambar 22. Menentukan Objek pada Unity3D Sebagai Sasaran Image Target

Apabila gambar yang dijadikan *image target* telah diaktifkan proses selanjutnya adalah menampilkan objek pada *class TrackableEventHandler* merupakan tahapan dimana sebuah objek akan akan muncul pada *image target* yang sudah ditentukan. Menampilkan sebuah objek pada *image target* yang sudah ditentukan dapat dilihat pada gambar 23.



Gambar 23. Objek diatas *Image Target*

Proses membuka url pada browser merupakan tahapan *button url* akan tampil pada *image target* yang sudah ditentukan untuk membuka url pada browser yaitu pada *class openURL* dapat dilihat pada gambar 24



Gambar 24. Tampilan *Inspector openURL (Script)*

Menampilkan informasi pada browser merupakan tahapan dimana gambar dan teks akan tampil pada *image target* yang sudah ditentukan yaitu pada *class informasi* dapat dilihat pada gambar 25.



Gambar 25. Tampilan *Inspector Informasi (Script)*

3. Hasil Pengujian Aplikasi

Pengujian program dalam penelitian ini dilakukan dengan menggunakan dua metode pengujian aplikasi, yaitu metode pengujian *white box* dan *black box*. Pada pengujian *white box* menggunakan *cyclometric complexity* untuk menghitung kompleksitas suatu method. Pengujian *white box* yang dilakukan

pada penelitian ini meliputi *listing program*, grafik alir, kompleksitas siklomatis, basis set dan *test case*. Kode program sudah berjalan dan menghasilkan respon yang diharapkan. Kompleksitas suatu method dapat diuji dengan mengetahui berapa jumlah *test case* yang diperlukan untuk mengevaluasi method tersebut sehingga setiap percabangan didalam method tersebut pernah dilalui. Semakin banyak jumlah *test case* maka method dapat dikatakan semakin kompleks.

Dalam aplikasi brosur interaktif dengan menerapkan teknologi *Augmented Reality* (AR) ini jumlah *test case* yang diperlukan adalah 7 buah. Sehingga aplikasi ini tingkat kompleksitasnya dapat dikatakan cukup tinggi. Metode pengujian *black box* digunakan untuk mengetahui apakah fungsi-fungsi, inputan, dan keluaran aplikasi sesuai dengan spesifikasi yang dibutuhkan oleh pengguna. Maka berdasarkan pengujian *black box* yang telah dilakukan, dapat diketahui bahwa aplikasi ini telah berhasil dibangun sesuai dengan kebutuhan fungsional.

Pengujian dilakukan dengan menggunakan perangkat *smartphone* dengan spesifikasi O.S Android v4.4.4 (KitKat), CPU: Quad-core 1.2 GHz Cortex-A53, Memori: 1 GB RAM, Kamera: *dual camera* (rear 8MP + front 5MP). Pengujian selanjutnya adalah pengujian akurasi aplikasi serta waktu yang dibutuhkan untuk mendeteksi *marker* dan menampilkan objek. Pengujian akurasi merupakan sebuah pengujian mendeteksi *marker* pada sudut tertentu (0, 15, 30,

dan 45 derajat), jarak tertentu (10 cm, 15 cm, 20 cm, 30cm) dan keadaan cahaya tertentu (cahaya lampu, cahaya matahari mendung dan terang). Pengujian ini dilakukan untuk mengetahui apakah setiap *marker* yang diarahkan ke kamera terdeteksi dengan baik atau tidak.

Pada pengujian akurasi semakin dekat jarak *marker* dengan kamera akan mengakibatkan ukuran *marker* yang terdeteksi semakin besar, sehingga bisa tertangkap dengan baik. Namun ketika jarak kamera dengan *marker* semakin jauh maka ukuran *marker* yang tertangkap kamera semakin kecil, sehingga pola *marker* menjadi tidak jelas dan mengakibatkan *marker* tidak terdeteksi. Salah satu permasalahan dari jarak antara kamera dan *marker* adalah tingkat kefokusan dari gambar yang ditangkap oleh kamera. Kualitas kamera yang memiliki fitur *autofocus* maka deteksi *marker* akan berjalan dengan baik.

Jarak minimum *marker* terhadap kamera 10 cm dan jarak maksimum *marker* terhadap kamera yaitu 30 cm dengan sudut minimum 0° dan sudut maksimum 45°. Pada jarak 20 cm tidak semua *marker* dapat terdeteksi dengan baik, sedangkan pada jarak 30 cm hanya dua buah *marker* yang dapat terdeteksi. *Marker* yang mampu terdeteksi pada jarak tersebut hanya *marker* yang tercetak dengan ukuran besar serta mempunyai rating maksimal yaitu bintang lima. Jarak terbaik kamera terhadap *marker* agar semua *marker* dapat terdeteksi dengan baik yaitu 15 cm. Berdasarkan hasil pengujian akurasi jarak minimum dan

maksimum serta sudut kemiringan kamera minimum – maksimum dipengaruhi dengan *rating*, ukuran setiap *marker* yang tercetak pada brosur dan keadaan cahaya saat pendeteksian *marker*. Terdapat tiga belas *marker* pada aplikasi brosur interaktif, delapan *marker* memiliki *rating* bintang empat dan lima *marker* memiliki *rating* bintang lima. Semakin besar *rating* dan ukuran *marker* maka *marker* akan tetap terdeteksi pada jarak dan sudut maksimum, apabila ada pantulan cahaya atau bayangan *smartphone* pada *marker* maka objek tidak tampil.

Pengujian *occlusion* dilakukan untuk mengetahui seberapa besar maksimum sebuah *marker* terhalang oleh benda agar objek yang muncul pada *marker* tidak *lost*.



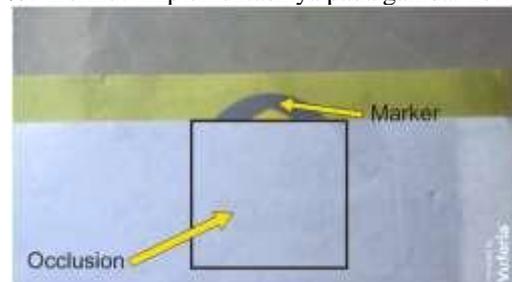
Gambar 26. *Occlusion* yang Terjadi Pada *Marker*

$$\frac{A_{occlusion}}{A_{total}} \times 100 = \%occlusion$$

$$\frac{2,5 \times 3,06}{3,31 \times 3,06} \times 100 = \%occlusion$$

$$\frac{7,65}{10,1286} \times 100 = 75,5\%$$

Persentase yang terjadi akibat *occlusion* berdasarkan benda yang menghalangi *marker* dengan perumpamaan seperti gambar 26 berwarna hitam ialah 75,5%. Berikut implementasinya pada gambar 27.



Gambar 27. *Occlusion* pada *Marker* Sehingga Objek Tidak Muncul

Terdapat satu *marker* dengan persentase 70%, dua *marker* dengan persentase 60%, tujuh *marker* dengan persentase 50%, dua *marker* dengan persentase 40%. Persentase maksimum rata - rata *occlusion* agar objek tetap tampil pada *marker* yaitu 50%. Pengujian selanjutnya yang dilakukan adalah pengujian waktu akses video. Pengujian menggunakan alat penghitung waktu untuk mengetahui waktu yang dibutuhkan untuk aplikasi menampilkan video yang diakses secara *online*. Ukuran file video profil sebesar

2.9 MB dan video kegiatan mahasiswa sebesar 3.4 MB. Pengujian dilakukan di dalam maupun di luar

bangunan dan berlokasi di (1) kampus PSSI UNEJ dan (2) Jl. Belitung 2 No.15, Jember. Hasil pengujian dapat dilihat pada tabel 1.

Tabel 1. Pengujian Waktu Akses Video Online

Nama Akses Video	Lokasi 1 (WiFi)							
	Dalam Bangunan (Indoor)				Luar Bangunan (Outdoor)			
	Pi ng	Downl oad	Upl oad	Waktu Akses	Pin g	Downl oad	Upl oad	Waktu Akses
Profil PSSI UNEJ	28 m	1.00 Mbps	1.73 Mb	02.38 s	24 ms	1.04 Mbps	2.14 Mb	01.76 s
Kegiatan Mahasiswa	s		ps	02.63 s			ps	02.07 s
Nama Akses Video	Lokasi 2 (Tri)							
	Dalam Bangunan (Indoor)				Luar Bangunan (Outdoor)			
	Pi ng	Downl oad	Upl oad	Waktu Akses	Pin g	Downl oad	Upl oad	Waktu Akses
Profil PSSI UNEJ	110 m	6.32 Mbps	0.86 Mb	02.47 s	89 ms	3.59 Mbps	0.76 Mb	02.11 s
Kegiatan Mahasiswa	s		ps	03.21 s			ps	02.41 s

Hasil pengujian menunjukkan bahwa waktu akses relatif tergantung pada lokasi dan jaringan *internet* yang digunakan oleh *user*. Pengambilan data pada *user* yang berada di dalam bangunan memiliki waktu akses yang relatif lama. Jika pengguna berada di dalam bangunan pengambilan data dari *server* menggunakan WiFi akan melambat, sedangkan dengan jaringan provider Tri pengambilan data dari *server* akan lebih lambat baik di dalam maupun di luar bangunan.

5. Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan oleh peneliti adalah sebagai berikut:

- Penerapan *Augmented Reality* (AR) pada aplikasi brosur interaktif berbasis android dilakukan dengan beberapa tahap yaitu analisis kebutuhan, desain sistem, penulisan kode program, pengujian kode program dan pengujian fungsionalitas dari aplikasi. Aplikasi brosur interaktif ini dibuat dengan menggunakan software Unity 3D 5.2.3f1 dengan bahasa pemrograman C#.
- Dalam penggunaan aplikasi harus diperhatikan jarak dan fokus kamera dengan *marker*. Semakin dekat jaraknya semakin baik mendeteksi *marker* namun semakin jauh akan mengganggu performa aplikasi.
- Semakin tinggi *rating* dari target gambar maka semakin mudah mendeteksi *marker*.
- Objek tidak tampil apabila ada pantulan cahaya atau bayangan *smartphone* saat mendeteksi *marker*.
- Persentase maksimum rata - rata *occlusion* agar objek tetap tampil pada *marker* yaitu 50%.

Waktu akses video secara *online* tergantung pada lokasi dan jaringan *provider* yang digunakan oleh *user*.

Daftar Pustaka

- Walesa, D. (2011). *Analisis Metode Occlusion Based pada Augmented Reality Studi Kasus : Interaksi dengan Objek Virtual Secara Real Time Menggunakan Gerakan Marker*. Repost dokumen karya ilmiah Institut Teknologi Telkom. Diambil kembali dari <https://nyargreen.files.wordpress.com/2011/02/jurn-al-ta.pdf>
- Gun A., L., Mark, B., Gerard, K. J. (2011). *Analisis Metode Occlusion Based pada Augmented Reality Studi Kasus : Interaksi dengan Objek Virtual Secara Real Time Menggunakan Gerakan Marker*. Repost dokumen karya ilmiah Institut Teknologi Telkom. Diambil kembali dari <https://nyargreen.files.wordpress.com/2011/02/jurn-al-ta.pdf>
- Volkert, B., Stephen, V., Mark, B. (2011). *Analisis Metode Occlusion Based pada Augmented Reality Studi Kasus : Interaksi dengan Objek Virtual Secara Real Time Menggunakan Gerakan Marker*. Repost dokumen karya ilmiah Institut Teknologi Telkom. Diambil kembali dari <https://nyargreen.files.wordpress.com/2011/02/jurn-al-ta.pdf>
- Elmqvist, N., Tsigas, P. (2011). *Analisis Metode Occlusion Based pada Augmented Reality Studi Kasus : Interaksi dengan Objek Virtual Secara Real Time Menggunakan Gerakan Marker*. Repost dokumen karya ilmiah Institut Teknologi Telkom. Diambil kembali dari <https://nyargreen.files.wordpress.com/2011/02/jurn-al-ta.pdf>
- Donald, H., M. Pauline, B. (2011). *Analisis Metode Occlusion Based pada Augmented Reality Studi Kasus : Interaksi dengan Objek Virtual Secara Real Time Menggunakan Gerakan Marker*. Repost dokumen karya ilmiah Institut Teknologi Telkom. Diambil kembali dari <https://nyargreen.files.wordpress.com/2011/02/jurn-al-ta.pdf>
- Ibañez, A. S., Figueras, J. P. (2013). *Vuforia v1.5 SDK: Analysis and evaluation of capabilities*. Universitat Politecnica De Catalunya. Diambil kembali dari <http://upcommons.upc.edu/bitstream/handle/2099.1/17769/memoria.pdf?sequence=4>
- Fernando, M. (2013). *Membuat Aplikasi Android Augmented Reality Menggunakan Vuforia SDK dan Unity*. Surakarta: Buku AR Online.
- Harris, C., Stephens, M. (1988). *A Combined Corner and Edge Detector*. United Kingdom: Plessey Research Roke Manor.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi Buku 1*. Yogyakarta: Andi.