

RANCANG BANGUN APLIKASI VISUALISASI 3D *FURNITURE INTERIOR* RUMAH MENGGUNAKAN *AUGMENTED REALITY* DENGAN METODE *MARKERLESS* BERBASIS *ANDROID*

^[1]Muhammad Qadriyanto, ^[2]Syamsul Bahri

^[1]^[2]Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jalan Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963

e-mail:

^[1]qadriyanto2793@gmail.com^[2]syamsul.bahri@siskom.untan.ac.id

Abstrak

*Sebuah ruangan yang nyaman dapat dipastikan sebagai ruangan yang tertata dengan baik dan sedap dipandang. Dibutuhkan kepiawaian pemilik rumah untuk menata interior ruangan agar terlihat menawan sekaligus nyaman untuk ditempati. Dengan berkembangnya teknologi informasi, sarana pengenalan furniture dapat dilakukan menggunakan perangkat berupa laptop maupun smartphone. Masyarakat pada umumnya masih mengalami kendala dalam menentukan pilihan furniture ruangan karena masyarakat tersebut berpikir apakah barang tersebut cocok bila ditempatkan dalam ruangan rumahnya. Berdasarkan permasalahan tersebut, pada penelitian ini dibuat sebuah aplikasi yang dapat membantu masyarakat merealisasikan pandangannya dalam menentukan furniture pada suatu ruangan. Dengan *Augmented Reality*, masyarakat tersebut dapat melihat gambaran nyata furniture di dalam lingkungan yang nyata secara *real time*. Aplikasi *augmented reality* ini dibuat menggunakan software *unity 3D* dan *vuforia* dengan menerapkan metode *FAST corner detection*. Hasil pengujian pendeteksian marker menunjukkan bahwa intensitas cahaya, jarak dan sudut pendeteksian mempengaruhi proses pendeteksian marker. Ketika intensitas cahaya rendah proses pendeteksian marker semakin sulit. Jarak terjauh yang dapat dicapai untuk mendeteksi marker mencapai 70cm. Hasil pengujian sudut pandang marker dapat terdeteksi diantara range 0° - 75°. Kemudian berdasarkan hasil pengujian pada sistem, didapat tingkat akurasi pendeteksian marker sebesar 70%.*

Kata Kunci : *Augmented Reality, Furniture, Markerless, Vuforia, FAST Corner Detection.*

1. PENDAHULUAN

Bagi mereka yang mempunyai selera tinggi, proses pemilihan *furniture* biasanya menjadi hal yang sulit untuk ditentukan. Akan ada banyak pertimbangan yang mengiringi pemilihan *furniture*, baik berdasarkan selera maupun kebutuhan. Beberapa pertimbangan terkait pemilihan *furniture* juga ditentukan sesuai kondisi ruang. Menentukan ukuran suatu *furniture* dengan ukuran ruangan merupakan salah satu cara yang umum dilakukan oleh masyarakat pada umumnya dalam menentukan suatu *furniture*. Penelitian tentang *Augmented Reality* dalam bidang *furniture* pernah dilakukan oleh Christian Patrik, yaitu visualisasi 3 dimensi desain

interior perabotan rumah berbasis *augmented reality*[1]. Penelitian tersebut menghasilkan sebuah aplikasi visualisasi objek 3 dimensi dengan output objek 3 dimensi *furniture* yang muncul pada tiap – tiap *marker*.

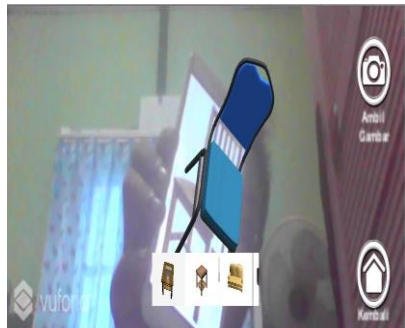
2. LANDASAN TEORI

2.1 *Augmented Reality*

Augmented Reality merupakan cara yang baru dan menyenangkan sebagai sarana interaksi antara manusia dan lingkungan sekitarnya karena dapat membawa, menggunakan bahkan menambahkan objek virtual (tidak nyata) ke dalam lingkungan yang nyata secara *real time*.

Definisi *Augmented Reality* sebagai sistem yang memiliki karakteristik sebagai berikut :[2]

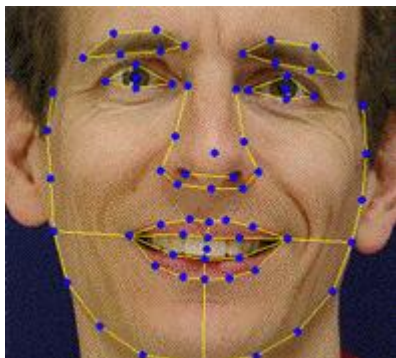
1. Menggabungkan lingkungan nyata dan virtual
2. Berjalan secara interaktif dalam waktu nyata
3. Integrasi dalam tiga dimensi (3D)



Gambar 1. *Augmented Reality*

2.2 *Markerless Tracking*

Metode *Augmented Reality* yang saat ini sedang berkembang adalah metode *markerless tracking*. Metode ini mempermudah pengguna untuk tidak perlu lagi menggunakan sebuah *marker* hitam putih ataupun barcode untuk menampilkan objek – objek *virtual*[3].



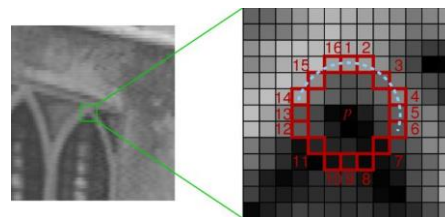
Gambar 2. *Face Tracking*

2.3 *Algoritma FAST (Feature from Accelerate Segment Test)*

FAST (Features from Accelerated Segment Test) Corner Detection merupakan algoritma penentuan titik sudut yang ditemukan oleh Edward Rosten. Proses penentuan titik sudutnya adalah dengan cara

merubah gambar menjadi warna hitam-putih dan menjalankan algoritmanya. Algoritma ini menentukan titik sudut dengan menentukan sebuah titik acuan atau calon titik sudut dengan asumsi variabel p , dari gambar masukan dengan 16 piksel disekitar p diperiksa. Ada 3 kasus berbeda yang ditetapkan untuk masing-masing perbandingan yakni :[4]

$$C = \begin{cases} I_n < I_p - t & \text{gelap} \\ I_p - t < I_n < I_p + t & \text{normal} \\ I_p + t < I_n & \text{terang} \end{cases} \quad (1)$$



Gambar 3. *Algoritma FAST*

2.4 *Pengolahan Citra Digital*

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. *Input* dari pengolahan citra adalah citra, sedangkan *output*-nya adalah citra hasil pengolahan. Agar dapat diolah dengan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit[5].

2.5 *Furniture*

Keindahan sebuah rumah bukan hanya terletak pada tampilan luarnya. Keindahan sesungguhnya justru terletak pada ruang – ruang di dalamnya. Sebuah rumah dapat

dirasakan sebagai tempat yang indah jika ruang – ruang yang ada di dalamnya dapat memberikan rasa nyaman saat penghuni menggunakannya untuk beraktivitas[6].



Gambar 4. Furniture

2.6 Vuforia Software Development Kit (SDK)

Vuforia adalah *Augmented Reality Software Development Kit (SDK)* untuk perangkat *mobile* yang memungkinkan pembuatan aplikasi *Augmented Reality*. Dulunya lebih dikenal dengan *QCAR (Qualcomm Company Augmentend Reality)*. Vuforia menggunakan teknologi *Computer Vision* untuk mengenali dan melacak gambar target (*Target Image*) dan objek 3D sederhana, seperti kotak, secara *real time*[3].

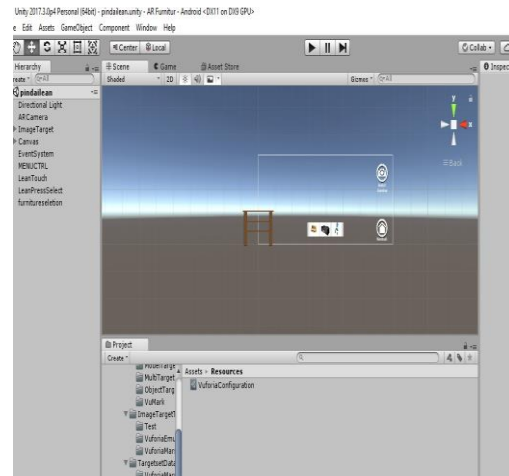
2.7 C#(C Sharp)

C# (dibaca C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif kerangka *.NET Framework*. Tanda pagar # yang digunakan memang bukan tanda kres dalam seni usic (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni usic tidak terdapat di dalam *keyboard* standar[7].

2.8 Unity 3D

Unity merupakan sebuah *software* yang terintegrasi untuk membuat game, arsitektur

bangunan dan simulasi. Bahasa pemrograman yang digunakan bermacam-macam, mulai dari Javascript, C#, dan BooScript[7].



Gambar 5. Layar Kerja Unity 3D

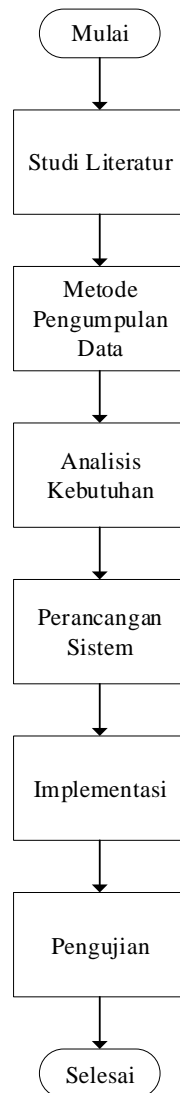
2.9 Blender 3D

Blender 3D adalah *software* visualisasi 3D yang mempunyai fitur yang cukup lengkap, gratis dan populer. Meskipun *software* ini gratis atau *free*, kualitas pencitraan digital tidak kalah dengan *software-software* grafis 3D lainnya. Blender dapat digunakan untuk membuat animasi 3D dan ada fitur tambahan yang membuat *software* ini semakin menarik yaitu bisa membuat sebuah *game* dengan *game engine* yang ada pada *software* ini[8].



Gambar 6. Layar Kerja pada Aplikasi Blender 3D

3. METODE PENELITIAN



Gambar 7. Diagram Alir Penelitian

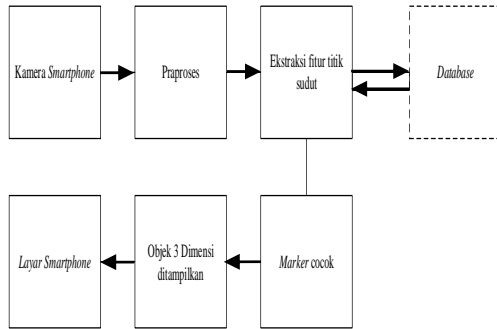
Penelitian ini dimulai dengan studi literatur yang bertujuan untuk menghimpun informasi yang relevan dengan topik penelitian. Dibutuhkan teori-teori pendukung untuk membuat sistem aplikasi visualisasi 3D *furniture interior* rumah menggunakan *augmented reality*. Teori-teori pendukung tersebut diperoleh dari beberapa sumber, yaitu buku, jurnal ilmiah dan berbagai sumber tertulis lainnya yang berkaitan dengan sistem pada penelitian ini. Tahap berikutnya pengumpulan data, dilakukan dengan mengumpulkan 8 gambar sampel *furniture*

interior rumah yang nantinya akan dijadikan objek 3D. Gambar – gambar tersebut meliputi gambar sofa, kursi, tempat tidur dan meja. Selanjutnya tahap analisis kebutuhan terhadap alat dan bahan yang diperlukan untuk melaksanakan penelitian ini. Analisis kebutuhan dapat dibagi menjadi dua bagian, yaitu analisis kebutuhan perangkat lunak dan analisis kebutuhan perangkat keras. Kemudian proses perancangan sistem dimulai dari membuat model 3D objek *furniture* dengan menggunakan *software blender* 3D, kemudian model objek yang sudah jadi, diekspor kedalam format .FBX agar bisa dipanggil kedalam *library Unity*. Tahapan selanjutnya adalah membuat *marker*. *Marker* berfungsi sebagai penanda agar objek 3D dapat muncul di dunia nyata. Pembuatan *marker* dimulai dengan *upload file* gambar berformat JPG ke *website Vuforia Qualcomm*. Setelah *file* gambar di upload maka secara otomatis *marker engine* dari *website vuforia Qualcomm* akan memberikan *rating* kelayakan untuk gambar yang diunggah. Setelah itu pihak *Vuforia Qualcomm* akan menyediakan *file* dalam bentuk *.Package* yang nantinya dijadikan masukan pada *coding* untuk mendeteksi gambar yang sudah dijadikan *marker*. *File .Package* yang diberikan oleh pihak *Vuforia Qualcomm* ini akan dijadikan sebagai *library* pada saat proses *rendering* di *unity* 3D menggunakan kamera *webcam*. Selanjutnya Implementasi sistem dilakukan berdasarkan rancangan program serata rancangan antarmuka aplikasi. Aplikasi dibangun dengan bahasa pemrograman C# dan dengan bantuan *library Vuforia*. Kemudian pengujian sistem dilakukan bertujuan untuk mengetahui apakah perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Dalam penelitian ini pengujian yang dilakukan terhadap sistem yaitu pengujian *black box*.

4. PERANCANGAN SISTEM

4.1 Diagram Blok Perancangan Sistem

Langkah yang dilakukan pada perancangan sistem adalah membuat diagram blok perancangan sistem.



Gambar 8. Diagram Blok Perancangan Sistem

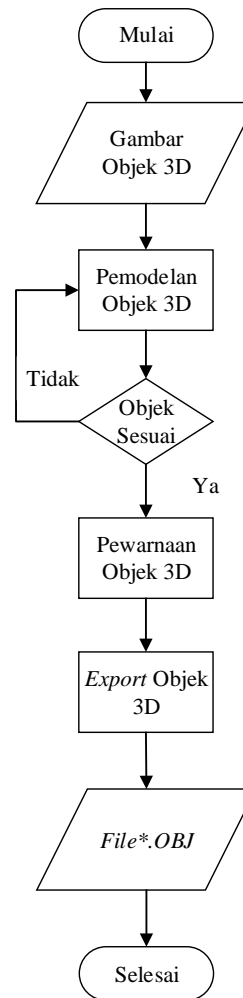
Pada Gambar 8 menjelaskan bahwa proses pertama yang dilakukan adalah *user* mengarahkan kamera dari *smartphone* ny gambar *marker*. Kemudian sistem akan melakukan praproses pada *marker*. Setelah gambar *marker* di *grayscale* kemudian masuk ke proses ekstraksi fitur titik sudut menggunakan algoritma *FAST* yang terdapat pada *library Vuforia*. Ketika titik titik sudut terekstraksi , maka system akan melakukan pencocokan pola titik sudut pada *database*. Ketika pola pada gambar *marker* cocok maka objek 3 dimensi *furniture* akan di tampilkan pada layar *smartphone user*.

4.2 Flowchart Perancangan Aplikasi

4.2.1 Inisialisasi Objek 3D

Model 3D yang akan ditampilkan dipanggil terlebih dahulu pada *software Unity 3D*. Agar aplikasi dapat menampilkan objek 3D tertentu tanpa merubah atau membangun ulang aplikasi, diperlukan sebuah file format **.3DS*, **.OBJ* atau **.FBX* untuk menentukan objek 3D yang akan dipanggil ke *Unity 3D*. Dalam inisialisasi objek 3D terdapat beberapa tahapan. Tahap pertama yaitu pemodelan objek 3D. Pemodelan objek 3D disesuaikan dengan acuan gambar yang akan di jadikan objek 3D. Objek 3D dibentuk sedemikian rupa agar terlihat sama dengan gambar yang dijadikan acuan. Setelah itu tahapan kedua yaitu pewarnaan objek 3D. Pewarnaan objek 3D disesuaikan dengan kebutuhan. Masuk ketahapan selanjutnya yaitu *rendering* objek. *Rendering* objek 3D berfungsi untuk menampilkan hasil akhir dari pengerjaan pemodelan objek 3D. Tahapan terakhir yaitu

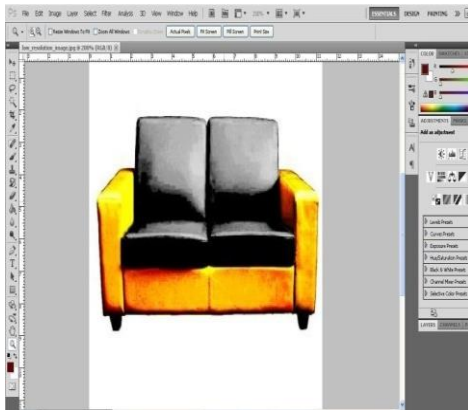
meng-*export* objek 3D ke dalam bentuk file berformat **.OBJ* ataupun **.FBX*.



Gambar 9. Flowchart Pembuatan Objek 3D

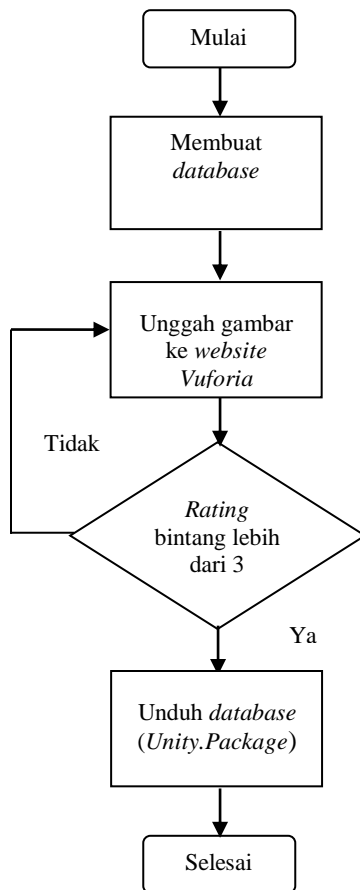
4.2.2 Inisialisasi Marker

Setelah objek tiga dimensi diunggah maka selanjutnya adalah pembuatan *marker*. *Marker* yang digunakan didapatkan dari hasil gambar buatan sendiri menggunakan *Photoshop*. Gambar yang digunakan pada penelitian ini sebanyak 2 macam jenis gambar, meliputi sofa dan kursi. Gambar yang akan di unggah memiliki ukuran tidak lebih dari 2 *Mega Byte*. Pada Gambar 10.



Gambar 10. Pembuatan *Marker* pada Aplikasi *Photoshop*

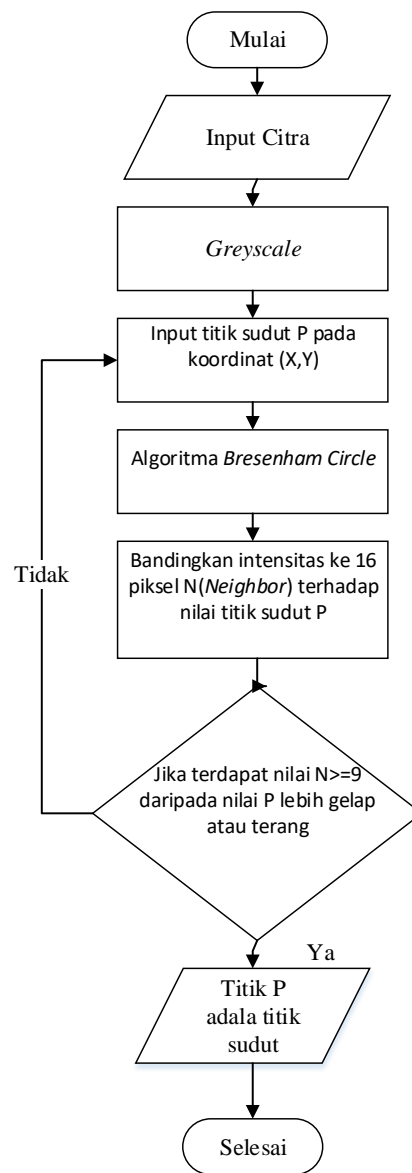
Setelah gambar *marker* dibuat tahapan selanjutnya adalah meregistrasi gambar *marker* dengan cara mengunggah gambar *marker* ke *website Vuforia*. Pada Gambar 11.



Gambar 11. Flowchart Pembuatan *Marker*

4.2.3 Ekstraksi Fitur Titik Sudut

Vuforia Qualcomm menggunakan algoritma *FAST (Feature from Accelerate Segment Test)* untuk proses pendeteksian *marker*. Hal ini dapat dilihat pada *web API vuforia* dimana terdapat *feature* atau titik sudut. Jika diaktifkan *feature*-nya, maka akan terlihat gambar *marker* yang diunggah warnanya akan berubah menjadi *grayscale*. Selanjutnya gambar tersebut akan diproses dengan menentukan *corner point*-nya. Pada Gambar 12.



Gambar 12. Flowchart Algoritma *FAST*

Gambar 12. menunjukkan alir sistem algoritma *FAST* pada *vuforia*. Beberapa penjelasan lebih lanjut adalah sebagai berikut :

1. Langkah pertama proses algoritma *FAST* adalah *input* citra.
2. *Grayscale* merupakan salah satu tahap dalam praproses citra, yaitu merubah citra asli menjadi skala keabuan dengan intensitas warna 0-255.
3. Pilih koordinat awal untuk menentukan calon titik sudut P.
4. Lakukan proses algoritma *Bresenham Circle* untuk mendapatkan 16 piksel *Neighbor* yang nantinya akan dijadikan pembanding terhadap calon titik sudut P.
5. Bandingkan nilai 16 piksel *Neighbor* terhadap nilai piksel calon titik sudut P menggunakan parameter

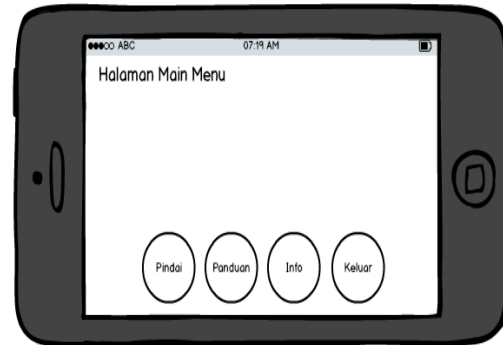
$$c = \begin{cases} I_p < I_p - t & \text{gelap} \\ I_p - t < I_p < I_p + t & \text{normal} \\ I_p + t < I_p & \text{terang} \end{cases}$$

6. Jika terdapat setidaknya 9 *Neighbor* bernilai lebih gelap atau lebih terang daripada nilai calon titik sudut P maka calon titik sudut P dapat dikatakan titik sudut dan proses akan diulang dengan menempatkan titik koordinat untuk calon titik sudut yang baru hingga semua koordinat dengan jari jari 3 di uji. Jika terdapat < 9 *Neighbor* yang bernilai lebih gelap atau lebih terang maka calon titik sudut P dinyatakan gagal sebagai titik sudut dan proses diulang dengan mencari koordinat titik sudut yang baru.

4.3 Perancangan Antarmuka

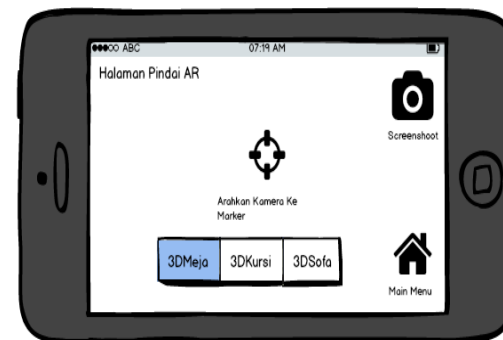
Perancangan antarmuka aplikasi terdiri dari empat halaman, yaitu halaman menu, halaman pindai, halaman panduan dan halaman info. Halaman menu berfungsi sebagai halaman utama yang memiliki beberapa fitur tombol seperti pindai, panduan, info dan keluar. Halaman pindai sebagai halaman yang memiliki fungsi untuk mendeteksi *marker* dan dapat memunculkan objek 3D *furniture*. Halaman panduan yang memiliki fungsi memunculkan cara

menggunakan aplikasi. Halaman info yang memiliki fungsi memunculkan halaman informasi riwayat hidup penulis. Berikut adalah rancangan antarmuka aplikasi *augmented reality interior* rumah.



Gambar 13. Perancangan Antarmuka Halaman Menu

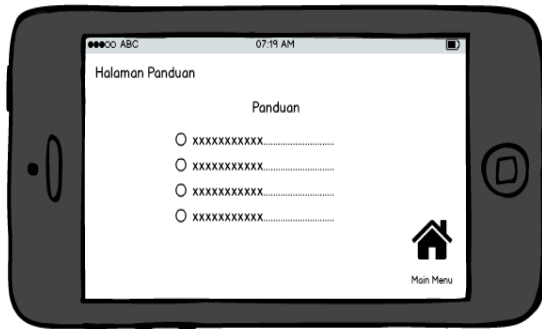
Gambar 13. menunjukkan perancangan tampilan halaman utama dari aplikasi. Disini pengguna dapat memilih 4 pilihan tombol pada layar halaman utama.



Gambar 14. Perancangan Antarmuka Halaman Pindai

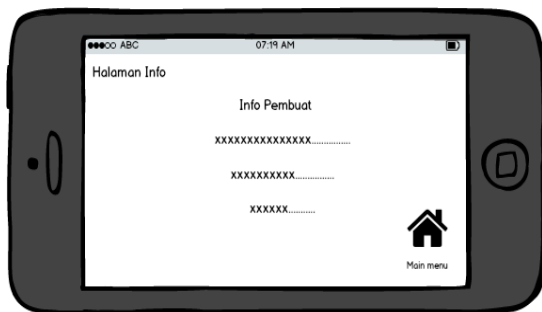
Gambar 14. menunjukkan tampilan halaman antarmuka pindai AR. Pada halaman ini kamera belakang pada *Smartphone* akan aktif dan melakukan pendeteksian marker. Ketika marker dikenali maka secara otomatis objek 3D *furniture* akan muncul pada layar. Terdapat fitur pilih objek 3D *furniture*. Fitur ini memungkinkan pengguna untuk memilih objek 3D *furniture* sesuai yang diinginkan. Kemudian pada bagian pojok kanan atas

halaman terdapat fitur *screenshot*. Fitur ini berfungsi sebagai penyimpan gambar ketika posisi objek 3D *furniture* sesuai keinginan.



Gambar 15. Perancangan Antarmuka Halaman Panduan

Gambar 4.15 menunjukkan tampilan halaman panduan. Halaman panduan AR berisi tentang petunjuk cara menggunakan aplikasi AR seperti memindai *marker*.



Gambar 16. Perancangan Antarmuka Halaman Info

5. PENGUJIAN DAN ANALISA

Pengujian sistem dilakukan bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang telah diuji. Pengujian dimaksudkan untuk mengetahui perangkat lunak yang telah dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Dalam penelitian ini pengujian yang dilakukan terhadap sistem yaitu pengujian secara fungsionalitas. Pengujian yang dilakukan terhadap fungsionalitas perangkat lunak ini menggunakan metode *black box*. Tujuan dari metode *black box* ini adalah untuk

menemukan ada tidaknya kesalahan fungsi pada perangkat lunak.

5.1 Pengujian Fungsional Aplikasi

Pengujian fungsional aplikasi merupakan proses dimana pengguna menyentuh/memilih menu-menu yang ada pada tampilan aplikasi. Hasil dari proses pengujian aplikasi dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian Fungsional Aplikasi

Kasus dan Hasil Uji			
Masukan	Keluaran yang diharapkan	Hasil Pengamatan	Keterangan
Memilih Menu Pindai	Sistem Dapat Menampilkan Halaman Pindai	Sistem Menampilkan Halaman Pindai	[<input checked="" type="checkbox"/>] Berhasil [] Gagal
Memilih Menu Panduan	Sistem Dapat Menampilkan Halaman Panduan	Sistem Menampilkan Halaman Panduan	[<input checked="" type="checkbox"/>] Berhasil [] Gagal
Memilih Menu Info	Sistem Dapat Menampilkan Halaman Info	Sistem Menampilkan Halaman Info	[<input checked="" type="checkbox"/>] Berhasil [] Gagal
Memilih Menu Keluar	Sistem Dapat Menampilkan Halaman Keluar	Sistem Menampilkan Halaman Keluar	[<input checked="" type="checkbox"/>] Berhasil [] Gagal
Menekan Tombol <i>Screen Capture</i>	Sistem Dapat Menyimpan Gambar	Sistem Menyimpan Gambar	[<input checked="" type="checkbox"/>] Berhasil [] Gagal
Menekan Tombol <i>Home</i>	Sistem Dapat Kembali ke Halaman Menu Utama	Sistem Kembali ke Halaman Menu Utama	[<input checked="" type="checkbox"/>] Berhasil [] Gagal

5.2 Pengujian Render Objek 3D

Pengujian *render* objek merupakan pengujian yang dilakukan terhadap teknologi *augmented reality* pada aplikasi 3D *furniture* interior rumah. Pengujian dilakukan dengan mengukur jarak dan sudut antara kamera pada perangkat *smartphone* dengan *marker* yang telah dicetak pada kertas dengan ukuran A4.

Pengujian dianggap berhasil jika *marker* terdeteksi dan dapat memunculkan objek 3D *furniture*. Sebaliknya, pengujian dianggap gagal apabila *marker* tidak terdeteksi dan tidak dapat memunculkan objek 3D. Aturan logika dalam penentuan keberhasilan atau kegagalan dalam pengujian adalah sebagai berikut:

1. if *tracking* = target and hasil = sesuai then Terdeteksi.

2. if *tracking* = target and hasil \neq sesuai then Gagal.

Tabel 5.2 *render* objek 3D

Kasus dan Hasil Uji				
Masukan		Keluaran yang Diharapkan	Keterangan	Hasil Uji
Jarak	Sudut			
20cm	75°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	60°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	45°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	30°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	15°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
30cm	75°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	60°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	45°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	30°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	15°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
40cm	75°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	60°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	45°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	30°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	15°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
50cm	75°	Memunculkan objek 3D	Objek 3D Muncul	[] Terdeteksi [√] Gagal
	60°	Memunculkan objek 3D	Objek 3D Muncul	[] Terdeteksi [√] Gagal
	45°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	30°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	15°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
60cm	75°	Memunculkan objek 3D	Objek 3D Muncul	[] Terdeteksi [√] Gagal
	60°	Memunculkan objek 3D	Objek 3D Muncul	[] Terdeteksi [√] Gagal
	45°	Memunculkan objek 3D	Objek 3D Muncul	[] Terdeteksi [√] Gagal
	30°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	15°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
70cm	75°	Memunculkan objek 3D	Objek 3D Muncul	[] Terdeteksi [√] Gagal
	60°	Memunculkan objek 3D	Objek 3D Muncul	[] Terdeteksi [√] Gagal
	45°	Memunculkan objek 3D	Objek 3D Muncul	[] Terdeteksi [√] Gagal
	30°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal
	15°	Memunculkan objek 3D	Objek 3D Muncul	[√] Terdeteksi [] Gagal

$$\text{Persentase_Keberhasilan} = \frac{21}{30} \times 100\% = 70\%$$

Berdasarkan hasil pengujian dari Tabel 5.5 *render* objek yang dilakukan sebanyak 30


kali, terdapat 21 pengujian yang berhasil dan 9 pengujian yang mengalami kegagalan dalam pendeteksian. Persentase keberhasilan aplikasi dalam mendeteksi *marker* adalah sebesar 70%.


Jarak yang diuji antara *marker* dengan kamera adalah 20cm, 30cm, 40cm, 50cm, 60cm dan 70cm, sedangkan untuk sudut yang diuji adalah 75°, 60°, 45°, 30° dan 15°. Jarak terdekat antara kamera dan *marker* agar sistem dapat merender objek yaitu 20cm, sedangkan untuk jarak terjauh mencapai 70cm. Sedangkan dari hasil pengujian 4 sudut, sudut 15° dengan jarak 20cm, 30cm, 60cm dan 70cm semua nya berhasil di deteksi hal ini dikarenakan hampir seluruh permukaan *marker* masih tertangkap oleh kamera *smartphone*. Sedangkan untuk pengujian sudut diatas 30,° jarak sangat mempengaruhi pendeteksian. Untuk sudut diatas 30° dengan jarak diatas 50cm *marker* gagal dideteksi. Hal ini dikarenakan permukaan *marker* tidak semuanya tertangkap oleh kamera *smartphone*.

5.3 Pengujian Intensitas Cahaya

Pengujian intensitas cahaya merupakan pengujian yang dilakukan terhadap intensitas cahaya pada ruangan pengujian. Parameter yang digunakan untuk pengujian intensitas cahaya meliputi ruangan terang dan ruangan redup. Hasil dari proses pengujian intensitas cahaya dalam mendeteksi *marker* dapat dilihat pada Tabel 5.3

Tabel 5.3 Pengujian Intensitas Cahaya

Kondisi	Hasil Pengujian	
	Output yang dihasilkan	Keterangan
Ruangan Terang		Deteksi sangat cepat. Objek 3D <i>furniture</i> dapat ditampilkan dengan baik.

Ruangan Gelap		Tidak Terdeteksi. Karena <i>marker</i> sulit dikenali dalam keadaan gelap
---------------	---	--

6. PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian dari rancang bangun aplikasi visualisasi 3D *furniture interior* rumah menggunakan *augmented reality* dengan metode *markerless* berbasis android didapatkan hasil sebagai berikut :

1. Berdasarkan pengujian pendeteksian *marker* untuk jarak terdekat adalah 20cm dan untuk jarak terjauh mencapai 70cm.
2. Berdasarkan pengujian pendeteksian *marker* untuk sudut pandang memiliki rentang antara sudut 30° hingga 75°, semakin tegak lurus sudut pandang terhadap *marker* maka semakin mudah *marker* untuk dideteksi .
3. Berdasarkan pengujian, penentuan titik sudut pada *marker* dilakukan membandingkan nilai intensitas 16 *neighbor* terhadap nilai intensitas titik pusat (sudut). Titik sudut tercipta apabila setidaknya memiliki 9 nilai intensitas *Brightness* atau *Darkness* dari nilai intensitas titik sudut.
4. Berdasarkan pengujian, cahaya ruangan mempengaruhi pendeteksian *marker*. Ruangan yang gelap tidak dapat digunakan untuk melakukan uji coba pendeteksian *marker* .
5. Pada penelitian ini telah berhasil diuji aplikasi visualisasi 3D *furniture interior* rumah menggunakan metode *markerless*, *marker* yang digunakan tidak diharuskan berwarna hitam putih atau barcode.
6. Persentase keberhasilan dari pengujian 30 data *render* objek adalah sebesar 70%.

6.2 Saran

Berdasarkan hasil penelitian ”Rancang Bangun Aplikasi Visualisasi 3D Furniture Interior Rumah Menggunakan Augmented Reality dengan Metode Markerless Berbasis

Android” terdapat beberapa saran yang bisa digunakan untuk penelitian yang lebih lanjut, antara lain :

1. Aplikasi *augmented reality* dapat dikembangkan untuk bidang lain, misalnya pada bidang kesehatan dan pembelajaran.
2. Pada penelitian selanjutnya disarankan dapat mengembangkan aplikasi *augmented reality* dengan menggunakan metode *Ground Plane*.

DAFTAR PUSTAKA

- [1] Patrik, Christian dan Sabloak, Rohit. 2013. *Visualisasi 3 Dimensi Desain Interior Perabotan Rumah Berbasis Augmented Reality Pada Mobile Phone Dengan Sistem Operasi Android*. STMIK MDP : Palembang.
- [2] Azuma, Ronald T. 1997. *A Survey of Augmented Reality*. Presence: Teleoperators and Virtual Environments.
- [3] Santoso, R. 2014. *Pembangunan Augmented Reality Denah Museum Geologi Bandung Menggunakan Metode Markerless Berbasis Android*. Skripsi. Universitas Komputer Indonesia : Bandung.
- [4] Akbar, R. 2016. *Implementasi Teknologi Augmented Reality Pada Majalah Interaktif Berbasis Android Menggunakan Metode Algoritma Features From Accelerated Segment Test Corner Detection*. Universitas Jember: Jember.
- [5] Sutoyo, T. d. 2009. *Teori Pengolahan Citra Digital*. Semarang: Andi Offset.
- [6] Noorrahmah, Nia dan Kartiwa, An An. 2009. *Tips Kreatif Menata Interior Rumah*. Depok.
- [7] Aryana, F. 2014. *Implementasi Augmented Reality Untuk Informasi Tiga Dimensi Pada Bangunan Rumah*. Skripsi. Universitas Komputer Indonesia : Bandung.
- [8] Pratama, Wahyu. 2014. *Game Adventure Misteri Kotak Pandora*. Puwekerto: STMIK AMIKOM.