

Dynamic Environment Augmented Reality Menggunakan Metode SLAM untuk Simulasi Penempatan Furnitur

Andrew Sebastian Lehman¹, Benny Budiawan Tjandrasa², Joseph Sanjaya^{*3}

¹Program Studi Sistem Komputer Fakultas Teknik, Universitas Kristen Maranatha,

²Program Studi Manajemen Fakultas Ekonomi, Universitas Kristen Maranatha,

³Program Studi Magister Ilmu Komputer Fakultas IT, Universitas Kristen Maranatha

E-mail: ¹andrewsebastianl@gmail.com, ²benny.tjandrasa@gmail.com,

^{*3}sanjaya josep@gmail.com

Abstrak

Seiring berkembangnya teknologi dibidang komputer yang semakin pesat, mesin-mesin vision banyak dikembangkan seperti penggunaan Augmented Reality untuk simulasi penyimpanan objek visual. Mebel adalah objek interior yang pada saat pembeliannya memerlukan penggambaran visual. Masih banyak orang yang tidak bisa memvisualisasikan estetika penempatan objek interior yang dapat menyebabkan penyesalan pasca pembelian. Berdasarkan fakta tersebut, dikembangkan dan dibuat Software Augmented Reality yang dapat mensimulasikan penempatan mebel menggunakan bidang 3D untuk memvisualisasikan penempatan mebel.

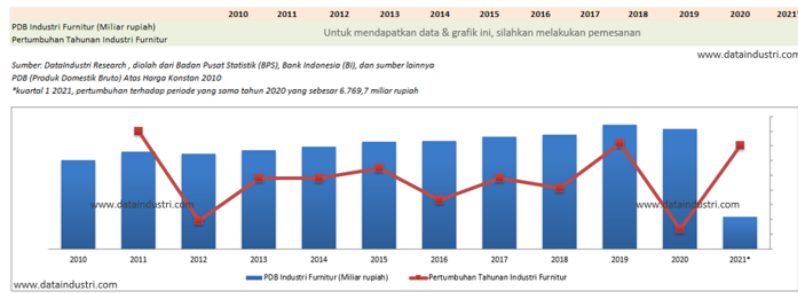
Kata Kunci— *Augmented Reality, Furniture, Mesin Vision*

1. PENDAHULUAN

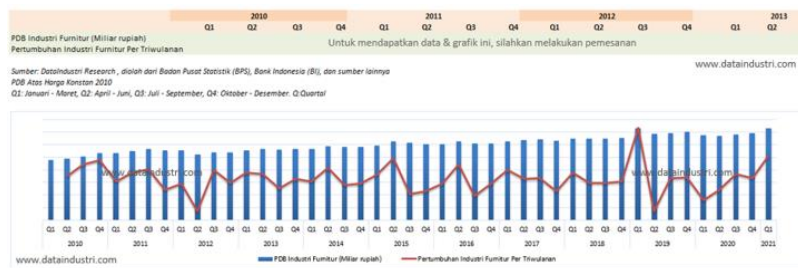
Kehidupan manusia saat ini banyak dibantu oleh berbagai mesin *vision* karena dipengaruhi oleh semakin cepatnya perkembangan teknologi. Mesin *vision* ini sudah diintegrasikan ke bentuk simulasi. *Vision* sudah menjadi konsep yang dapat divisualisasikan. Ketidaksesuaian dengan ekspektasi saat pembelian mebel sering terjadi sehingga menimbulkan kekecewaan. Masih banyak orang yang tidak bisa memvisualisasikan estetika penempatan mebel saat mendesain interior rumahnya, sehingga simulasi penempatan mebel dan desain interior secara *real time* dibutuhkan untuk membantu manusia memvisualisasikan penempatan mebel pasca pembelian.

Pertumbuhan penduduk yang terus meningkat hingga mencapai 270 jiwa dan tujuh puluh persen dari jumlah penduduk tersebut merupakan penduduk dengan usia produktif. Hal ini berdampak positif bagi pertumbuhan industri mebel seiring dengan kebutuhan perumahan dan gedung yang semakin meningkat. Di tengah kondisi pandemi COVID-19 ternyata, menurut Abdul Sobur yang merupakan Ketua Presidium Himpunan Industri Mebel dan Kerajinan Indonesia (HIMKI) industri mebel mengalami pertumbuhan ekspor sebesar 5%-6% ke zona pasar Amerika Serikat [1].

Pandemi ini mendorong dalam produksi sosial, dengan pergeseran dari pengeluaran untuk alat transportasi dan pariwisata menjadi ke arah perbaikan rumah, penyempurnaan tata letak rumah, produk teknologi, dan hiburan di dalam rumah yang sangat ketergantungan pada gadget dan internet. Setelah sebelumnya mengalami kontraksi, industri mebel pada triwulan I 2021 kini bangkit dan tumbuh positif sebesar 8.04% [2].



Gambar 1. Pertumbuhan Tahunan Industri Furnitur [2].



Gambar 2. Pertumbuhan Industri Furnitur Per Triwulan [2].

Bisnis toko online dan *e-commerce* di Indonesia saat ini sedang *booming* disebabkan karena meningkatnya jumlah pengguna *gadget* untuk keperluan transaksi dan perdagangan serta peningkatan kualitas jaringan internet di Indonesia. Pemanfaatan teknologi *Augmented Reality* untuk promosi mebel belum banyak diteliti, sehingga penelitian ini ingin mengambil ceruk pasar yang besar tersebut. *Augmented Reality* untuk mendukung pemahaman dan interaksi pengguna dengan dunia nyata. Kegiatan dalam dunia nyata dapat dibantu dengan informasi yang dimunculkan oleh benda maya [3].

Berdasarkan fakta tersebut *software* dikembangkan untuk membantu memvisualisasikan penempatan mebel dan desain interior dengan memanfaatkan teknologi *SLAM* dalam implementasinya terhadap *dynamic environment AR*. *Software* ini bekerja dengan mengambil data seluruh ruang sebagai marker untuk *Augmented Reality*, selanjutnya menggunakan *image processing* untuk mendapatkan hasil grafik yang sesuai dengan ketentuan.

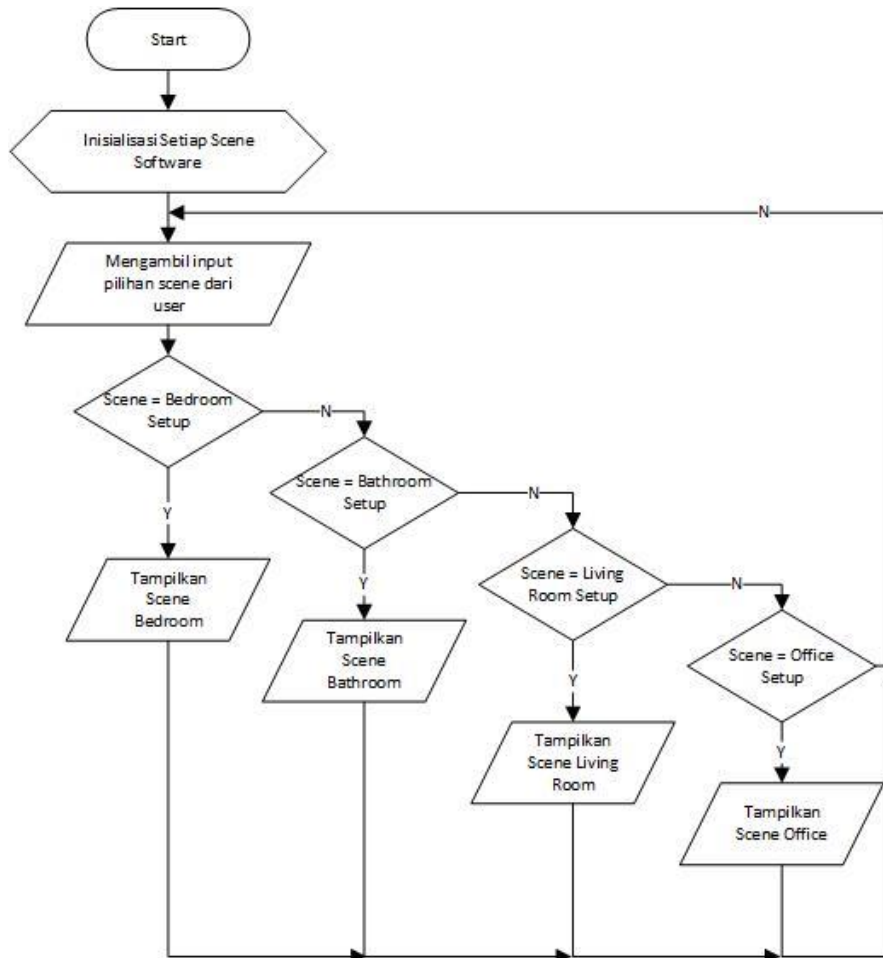
2. METODE PENELITIAN.

2.1. Tahapan Review

Perancangan Sistem *Dynamic Environment AR* untuk simulasi penempatan mebel akan mengaplikasikan model proses *waterfall*. Metode ini menawarkan pendekatan pada masalah *software* yang sistematis dan sekuensial, dimulai dari tingkat sistem dan berlanjut melalui analisis, desain, *coding*, pengujian dan *maintenance* [4].

1. Analisis: proses pengumpulan sumber data penelitian, melakukan strategi untuk mengatur waktu saat penelitian, dan penjabaran alat yang digunakan untuk penelitian.
2. Desain: rancangan yang dibuat dari representasi aplikasi.
3. Coding: kode yang diterjemahkan kedalam bahasa yang bisa dibaca.
4. Pengujian: modul yang ada di sistem akan dilakukan pengujian, dan jika tidak sesuai dengan yang diharapkan maka akan dilakukan perbaikan.

2.2. Flowchart Menu Utama



Gambar 3. Flowchart Menu Utama

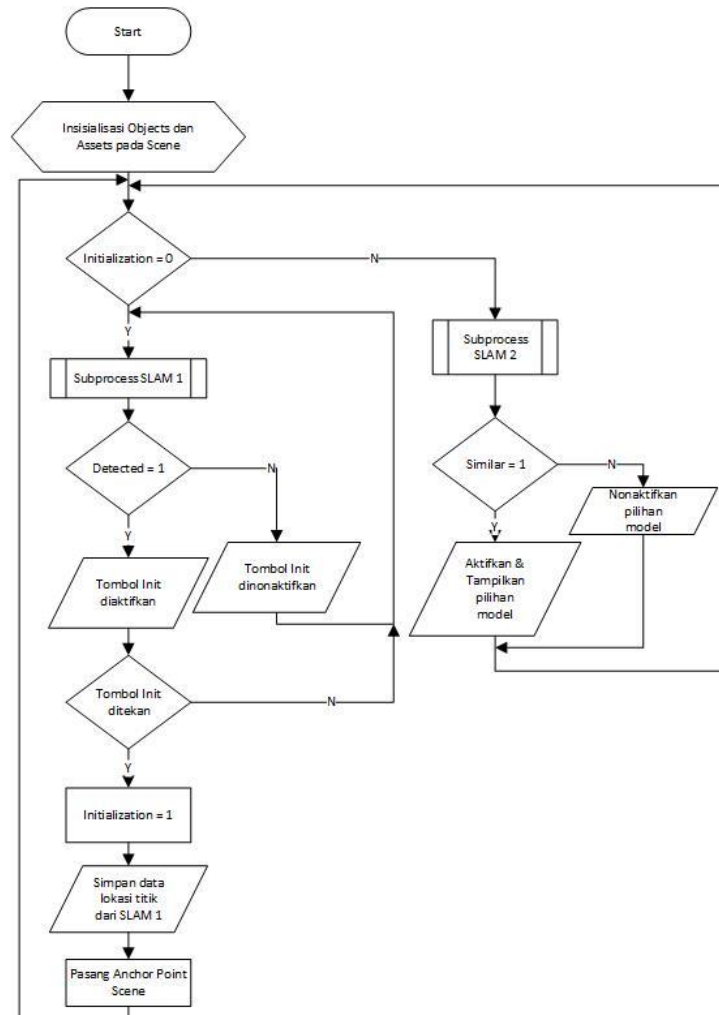
Flowchart menu utama dari *software Augmented Reality* ini menjelaskan *scene setup* dipersiapkan saat *software* terbuka. Setelah inisialisasi tersebut, *software android* menunggu pilihan *setup* yang telah dipersiapkan sebelumnya. Ketika pilihan diterima *software* akan membuka *scene setup* yang sesuai dengan pilihan *user*.

2.3. Flowchart Scene

Flowchart scene menjelaskan proses yang terjadi saat satu scene dibuka. Inisialisasi dilakukan untuk mempersiapkan objects dan asset yang dibutuhkan saat scene dibuka. Subprocess SLAM 1 akan dijalankan dengan menghasilkan variable detected. Variable detected bernilai 1 tombol initialize akan diaktifkan dan variable detected bernilai 0 tombol initialize akan dinonaktifkan. Proses ini dilakukan sampai tombol initialize ditekan. Anchor point scene dipasang pada koordinat kamera saat tombol initialize ditekan. Anchor point ini digunakan sebagai penentu posisi scene sehingga ketika 3D model diletakkan, posisi 3D model akan tetap pada posisinya walaupun kamera bergerak.

Subprocess SLAM 2 dijalankan untuk memastikan anchor point tidak berpindah dan memastikan user tidak dapat menambahkan 3D model pada saat anchor point tidak terdeteksi.

Subprocess ini akan menghasilkan variable similar. Variable similar bernilai 0 maka user tidak dapat menyimpan 3D model baru, dan variable similar bernilai 1 maka tombol pilihan 3D model akan dimunculkan lalu diaktifkan sehingga 3D model baru dapat ditambahkan.



Gambar 4. Flowchart Scene

2.3.1. Subprocess Metode SLAM 1

Subprocess ini akan mengambil gambar pada kamera, mengubah gambar menjadi *monochrome*, lalu mengkalkulasi titik *identifier* pada gambar tersebut. Jika titik *identifier* lebih dari 5 maka variable *detected* akan diberi nilai 1, jika titik *identifier* kurang dari 5 maka nilai variable *detected* akan diberi nilai 0.

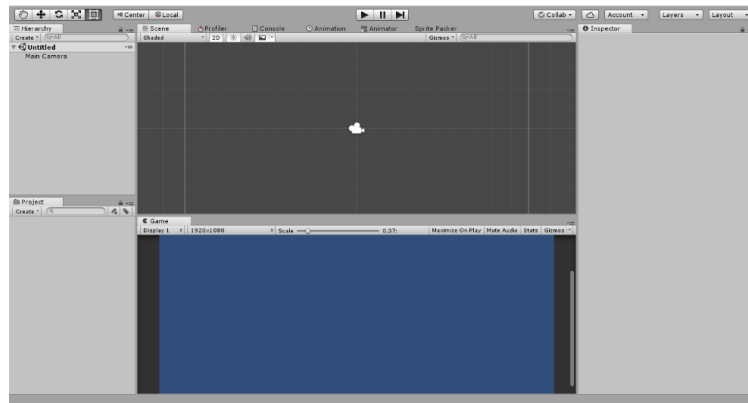
2.3.2. Subprocess Metode SLAM 2

Subprocess ini akan mengambil gambar pada kamera, mengubah gambar menjadi *monochrome*, lalu mengkalkulasi titik *identifier* pada gambar tersebut. Setelah kalkulasi dilakukan hasilnya dibandingkan dengan data titik yang sebelumnya disimpan dari Subprocess

SLAM 1. Jika data titik mirip maka variable similar akan diberi nilai 1 dan jika data berbeda maka variable similar akan diberi nilai 0.

2.4. Pembuatan project

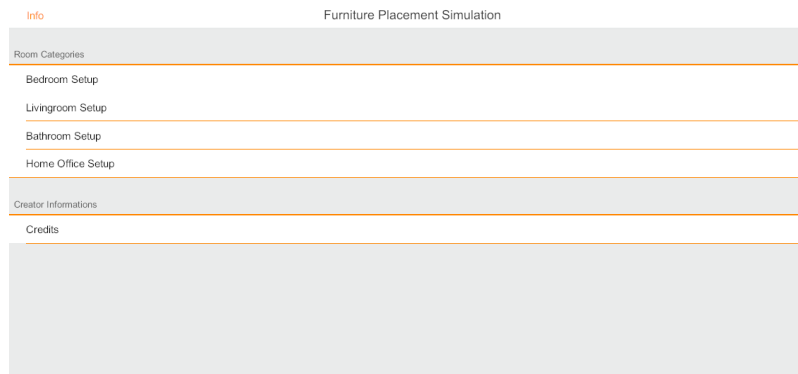
Pengembang harus membuat project terlebih dahulu untuk mengembangkan *software* pada Unity. Setelah Unity selesai membuat *project* maka akan muncul Unity Editor dengan *scene default* bernama *Untitled* seperti pada Gambar 5



Gambar 5. Tampilan Unity Editor Setelah *Project* Dibuat

2.5. Scene Main Menu

Scene Main Menu adalah *scene* yang dibuka pada saat *software* dijalankan. *Scene* ini adalah tempat *user* diberikan pilihan berupa *list* untuk memilih *scene setup* ruangan yang diinginkan, *scene* yang dipersiapkan antara lain kamar tidur, kamar mandi, ruang tamu, dan kantor. Desain *interface* pada *scene* ini adalah seperti pada gambar 6.



Gambar 6. Desain *User Interface* pada *Scene Main Menu*

2.6. Pembuatan 3D model

Proses pembuatan *3D model* untuk *software* ini dibutuhkan referensi yang cukup realistis karena tujuan utama dari *software* ini adalah untuk mensimulasikan penyimpanan mebel secara *realtime*.

3D model dibuat dengan cara setup awal pada blender, menambahkan gambar referensi melalui *3D View Properties*, membentuk *base mesh*, menambahkan detail *sculpting*. Saat *sculpting* dilakukan, terdapat *Dynamic Topology* pada *Tools Shelf*. *Dynamic Topology* berguna untuk menambahkan jumlah *vertices* setiap *sculpting* dilakukan untuk menambahkan *detail* pada *mesh*. Setelah itu membuat *UV wrapping Map* untuk *texture 3D model*. Lalu membuat *texture* berdasarkan *UV map* dengan menggunakan *adobe photoshop* dan membuat *materials*. *Materials* adalah peningkatan pemetaan tekstur (dan prasyarat untuk efek bayangan canggih) yang memungkinkan *object* dalam pemodelan *3D* dan *video games* untuk mensimulasikan berbagai jenis bahan dalam kehidupan nyata.

Setelah *3D model* selesai dibuat, pengembang dapat menguji *3D model* dengan memasuki *mode Viewport Shading Rendered*. Pengembang juga dapat membuat gambar atau *video* dengan melakukan *rendering* setelah *3D model* selesai. Hasil *render* akhir terdapat pada gambar 7.



Gambar 7. Hasil Render Akhir 3D Model

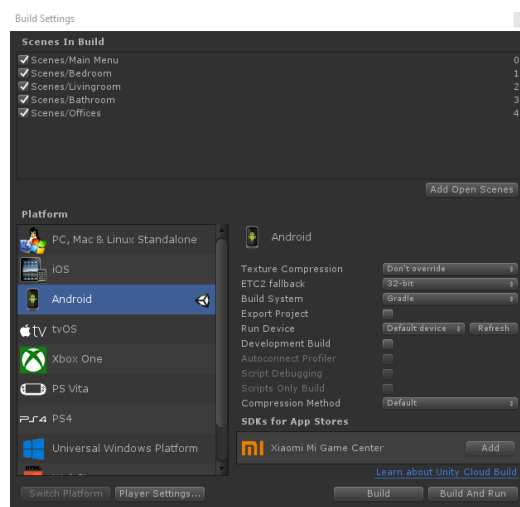
2.7. Scene Simulasi

Scene simulasi merupakan tempat *user* melakukan simulasi penempatan mebel yang temanya disesuaikan berdasarkan pilihan *user* pada *scene main Menu*. Tahapan untuk membuat *scene* simulasi yaitu dengan *setting main camera* pada *scene* simulasi, *scene controller* dan *user interface* pada *scene* simulasi.

2.8. Build Aplikasi

Bagian ini akan menjelaskan cara mengakses *build settings* dan cara membuat *project*-nya menjadi *software* yang siap dipakai dalam berbagai platform yang disediakan oleh Unity. Cara mengakses *build settings* adalah dengan menekan menu *file* pada *header* Unity dan memilih menu *build settings*. *Build settings* adalah menu yang berguna untuk memunculkan daftar *scene* yang akan disertakan pada *software*. *Player Settings* didapat dari *build settings* yang berisi opsi untuk mendefinisikan *behaviour* dari produk akhir *software*. Gambar menu *build settings* terdapat pada gambar 8.

Pengembang dapat memilih target platform untuk *software* yang dibuat dan dapat dengan mudah membuat *multiplatform software* yang tertera di menu tampilan ini. *Platform* yang disediakan oleh Unity saat ini yakni PC, iOS, Android, tvOS, Xbox One, PS Vita, PS4, Universal Windows Platform, WebGL, dan Facebook [5].

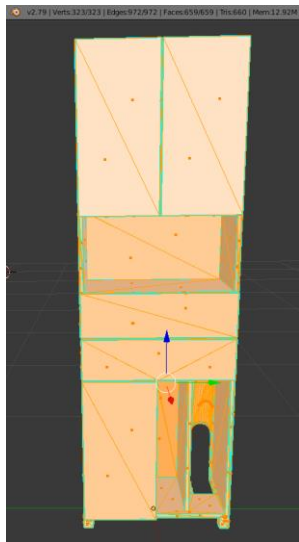
Gambar 8. Tampilan *Build Settings*

3. HASIL DAN PEMBAHASAN

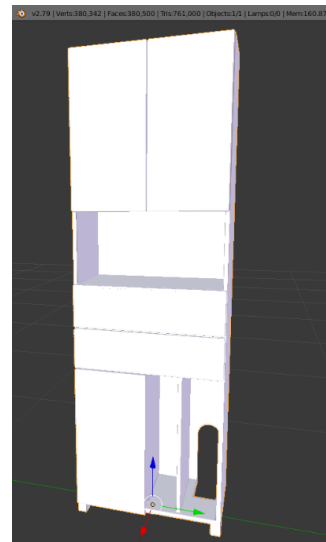
3.1. Teknik Pemodelan 3D

Perkembangan teknologi saat ini muncul berbagai macam teknik yang memudahkan pembuatan *3D model*. Terdapat 2 teknik pembuatan *3D model* yang akan diamati yaitu secara manual menggunakan Blender dan *3D scanning* menggunakan *meshroom*.

3.1.1. Pembuatan Manual Menggunakan Blender dan Meshroom 3D Scan (Fotogrametri)



Gambar 9. Hasil Akhir Pembuatan Manual



Gambar 10. Hasil Akhir 3D scan yang sudah diperbaiki

3D model yang dibuat secara manual menggunakan blender memiliki jumlah vertices yang lebih efektif dan fleksible hasil akhir manual dilihat pada gambar 9. *3D model* yang dibuat

melalui fotogrametri memiliki detail yang lebih baik tetapi jumlah *vertices* tidak efektif sehingga perlu dilakukan *remesh*. Perbedaan data *3D model* dari masing masing teknik tertera pada tabel 1

Tabel 1. Data *3D model* masing - masing teknik

Teknik	<i>Vertices</i>	<i>Edges</i>	<i>Faces</i>	<i>Memory (MB)</i>	Waktu Pembuatan
Manual	323	972	659	12,92	2 Jam 47 Menit
Fotogrametri	380342	760847	380500	160,87	4 Jam 15 Menit

Pembuatan *3D model* menggunakan fotogrametri merupakan *workload* yang cukup berat bagi sistem. Waktu yang dibutuhkan untuk membuat satu *3D model* menggunakan fotogrametri juga cukup lama karena *user* harus menangkap puluhan foto dari berbagai sisi dan memperbaiki *mesh*.

3.1.2. Pengujian *3D model* dalam Software

Tabel 2. Data performa *devices* pada saat pemunculan *3D model*

Teknik	Devices	AnTuTu Score	Jumlah <i>3D model</i>	FPS	Load Time
Manual	GX501VIK	856926	1	60	Instant
			2	60	Instant
			3	60	Instant
			4	60	Instant
			5	60	Instant
Fotogrametri			1	60	Instant
			2	60	Instant
			3	60	Instant
			4	58	Instant
			5	49	Delayed
Manual	Samsung Note 9	235456	1	60	Instant
			2	60	Instant
			3	60	Instant
			4	60	Instant
			5	60	Instant
Fotogrametri			1	60	Instant
			2	50	Instant
			3	42	Delayed
			4	33	Delayed
			5	24	Delayed
Manual	Google Nexus 5	57006	1	45	Instant
			2	39	Instant
			3	35	Delayed
			4	31	Delayed
			5	29	Delayed
Fotogrametri			1	25	Delayed
			2	19	Delayed
			3	14	Delayed

Teknik	Devices	AnTuTu Score	Jumlah 3D model	FPS	Load Time
			4	11	Delayed
			5	10	Delayed

Pengujian ini dilakukan dengan memunculkan *3D model* kedalam scene hingga berjumlah 5 buah. Setiap satu *3D model* ditambahkan, dapat diamati perubahan performa dari beberapa *devices* yang memiliki *score AnTuTu Benchmark* dari tinggi hingga rendah. *Devices* yang dipakai untuk pengujian yaitu Laptop GX501VIK, Samsung Note 9, dan Google Nexus 5. Contoh dari pemunculan *3D model* dari *software* terdapat pada gambar 11.



Gambar 11. Contoh Pemunculan 3D Model

Dilihat dari tabel 2 dapat disimpulkan bahwa pembuatan *3D model* menggunakan fotogrametri tidak efektif digunakan dalam pembuatan *software* berbasis *Augmented Reality*. Hal ini dikarenakan *workload* yang digunakan pada saat proses *tracking Augmented Reality* sudah tinggi sehingga *system resources* untuk melakukan *rendering* lebih rendah. Hal ini dapat ditanggulangi dengan melakukan online *tracking* pada proses *Augmented Reality* sehingga *tracking workload* dapat dilakukan oleh server, tetapi proses ini membutuhkan dana yang cukup besar.

3.2. Pembacaan Titik untuk Metode SLAM

3.2.1. Warna Gambar

Tabel 3. Gambar sebelum dan sesudah proses berdasarkan warna

Warna	Gambar Awal	Gambar Hasil
RGB		
Grayscale		
Black & White		

Akan digunakan tiga *sample* gambar dengan warna *RGB*, *grayscale*, dan *black & white*. Kondisi gambar sebelum dan sesudah proses terdapat pada tabel 3.

Didapatkan data penting pada pengujian untuk menemukan langkah yang paling efektif untuk memproses gambar sebelum pencarian titik. Data pengujian ini terdapat pada tabel 4.

Tabel 4. Data pengujian warna gambar

Warna	Jumlah Titik	Waktu Proses
RGB	198	28 Menit 42 Detik
Grayscale	225	16 Menit 33 Detik
<i>Black & White</i>	375	8 Menit 38 Detik

Gambar RGB terdapat semua kemungkinan warna sehingga proses *tracking* harus membandingkan 256 irisan kubus warna RGB. Hal ini menyebabkan *workload* dalam proses *tracking* lebih tinggi dan membutuhkan waktu yang lama, tetapi menghasilkan banyak titik yang lebih sedikit karena perbandingan warna pada RGB tidak signifikan.

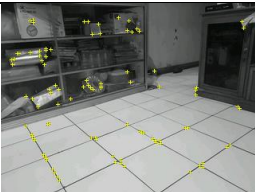
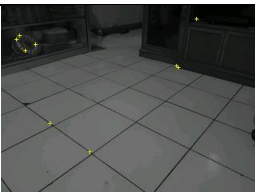
Gambar *grayscale*, nilai piksel adalah angka tunggal yang mewakili kecerahan piksel. Format piksel yang paling umum adalah gambar *byte* yang nilainya disimpan sebagai bilangan bulat 8-bit yang memberikan kisaran nilai yang mungkin dari 0 hingga 255. Biasanya nol diambil menjadi hitam, dan 255 dianggap putih. Jangkauan warna yang harus dibandingkan lebih sedikit sehingga *workload* lebih ringan dan waktu yang dibutuhkan untuk menyelesaikan berkurang.

Gambar dengan warna *black & white*, Titik yang didapatkan lebih banyak karena perbedaan antara hitam dan putih sangat jelas. Dengan hasil pengujian ini, gambar yang didapatkan pada saat penangkapan *realtime* diproses terlebih dahulu menjadi *black & white* sebelum melakukan proses *tracking*

3.2.2. Intensitas Cahaya

Proses ini akan digunakan tiga *sample* gambar dengan tingkat intensitas cahaya terang, redup, dan gelap. Kondisi gambar sebelum dan sesudah proses terdapat pada tabel 5.

Tabel 5. Gambar sebelum dan sesudah proses berdasarkan cahaya

Cahaya	Gambar Awal	Gambar Hasil
Terang		
Redup		
Gelap		

Gambar dimasukkan ke dalam *software* untuk pengujian menemukan titik *identifier*. pengujian ini didapatkan *data* penting untuk menemukan pencahayaan yang paling efektif untuk memproses gambar sebelum pencarian titik. *Data* tersebut terdapat pada tabel 6.

Tabel 6. Data pengujian cahaya ruangan

Cahaya	Jumlah Titik	Waktu Proses
Terang	89	11 Menit 42 Detik
Redup	9	14 Menit 33 Detik
Gelap	2	16 Menit 38 Detik

3.2.3. Tingkat Kompleksitas Ruangan

Pengujian dilakukan menurut kondisi kompleksitas ruangan pada saat *tracking* dilakukan. Pada proses ini akan digunakan dua *sample* gambar dengan tingkat kompleksitas ruangan penuh dan kosong. Kondisi gambar sebelum dan sesudah proses dapat dilihat pada tabel 7.

Tabel 7. Gambar sebelum dan sesudah proses berdasarkan kompleksitas

Tingkat Kompleksitas	Gambar Awal	Gambar Akhir
Ruangan Penuh		
Ruangan Kosong		

Didapatkan data penting untuk menemukan kompleksitas ruangan yang paling efektif untuk memproses gambar sebelum pencarian titik. *Data* tersebut dapat dilihat pada tabel 8.

Tabel 8. Data pengujian kompleksitas ruangan

Tingkat Kompleksitas	Jumlah Titik	Waktu Proses
Penuh	289	11 Menit 42 Detik
Kosong	6	7 Menit 33 Detik

3.3. Pengujian Setiap Scene Software

Pengujian dilakukan pada keseluruhan *scene software* yang telah diselesaikan. Pada *software* ini terdiri dari 4 *scene* yaitu *bedroom*, *bathroom*, *living room* dan *office* yang memiliki 6 pilihan *3D model* yang siap dimunculkan.

Tabel 9. Pengujian Setiap Scene Software

Scene	Variant	Gambar
<i>Bedroom Scene</i>	Meja belajar, meja lampu tidur, meja rias, kasur dan lemari baju	
<i>Bathroom Scebe</i>	<i>Bathtub</i> , wastafel, toilet, lemari kamar mandi dan bangku kamar mandi	
<i>Living room Scene</i>	Sofa, meja kaca, kursi kecil, lemari elektronik, dan meja bunga	
<i>Office Scene</i>	Kabinet, meja kerja, 2 variant kursi, dan projection screen	

Terlihat pada seluruh scene, semua 3D model berhasil dimunculkan kedalam lingkungannya. Semua 3D model juga berhasil digerakan, diperbesar, dan dirotasi menggunakan *single touch* dan *multi touch*.

4. KESIMPULAN

Perancangan serta pengujian pada software yang telah dibuat didapatkan kesimpulan sebagai berikut:

1. Simulasi Penempatan Mebel telah berhasil diimplementasikan melalui Teknologi *Instant Tracking Augmented Reality*.
2. Saat proses mencari titik *identifier* warna hitam dan putih saja yang paling cepat dan efektif.
3. Pembacaan data titik *identifier* sangat dipengaruhi oleh intensitas cahaya dan kompleksitas ruangan.

Saran yang bisa dilakukan agar dapat mengembangkan *software* yang sudah dibuat untuk penelitian selanjutnya yaitu:

1. Proses akan berjalan lebih cepat dan *system resources* dapat digunakan untuk *rendering* yang tampak realistis jika saat proses *tracking* sebaiknya dilakukan secara *online* sehingga *workload* dapat diolah oleh server yang mempunyai *processing power* yang lebih kuat.

2. Agar bisa dipakai untuk bidang marketing, sebaiknya *3D model* diintegrasikan dengan produk asli.
3. Jika ingin *software* dapat diperbaharui tanpa harus melakukan *build* ulang, maka sebaiknya *software* memiliki pilihan atau opsi untuk menambahkan pilihan 3D model baru pada tampilan.

PENGHARGAAN

Penulis menyampaikan ucapan terima kasih kepada Universitas Kristen Maranatha yang telah membiayai penelitian ini.

DAFTAR PUSTAKA

- [1] idxchannel. "Ekspor Industri Furniture Tumbuh 5 Persen di Tengah Pandemi Covid-19." <https://www.idxchannel.com/economics/ekspor-industri-furniture-tumbuh-5-persen-di-tengah-pandemi-covid-19> (accessed 19 Jun, 2021).
- [2] B. I. BPS, dan Kementerian Perdagangan. "Pertumbuhan Industri Furnitur 2011 - 2021 | Pusat Data Industri Indonesia." <https://www.dataindustri.com/produk/tren-data-pertumbuhan-industri-furnitur-di-indonesia/> (accessed 19 Jun, 2021).
- [3] E. Pangilinan, S. Lukas, and V. M. eds., E. Pangilinan, S. Lukas, and V. Mohan, Eds. *Creating augmented and virtual realities: theory and practice for next-generation spatial computing*, First edition ed. Sebastopol, CA: O'Reilly Media, 2019, p. 340.
- [4] M. Kuhrmann *et al.*, "Hybrid software and system development in practice: waterfall, scrum, and beyond," in *ICSSP 2017: International Conference on the Software and Systems Process 2017*, 2017-07-05 2017, Paris France: ACM, pp. 30-39, doi: 10.1145/3084100.3084104.
- [5] P. Buttfield-Addison, J. Manning, and T. Nugent, *Unity game development cookbook: essentials for every game*, First edition ed. Sebastopol, CA: O'Reilly Media, 2019, p. 391.