

BAB II

LANDASAN TEORI

2.1. Augmented Reality (AR)

Augmented Reality adalah sebuah teknologi yang dapat menampilkan objek virtual yang dibuat oleh komputer, dua dimensi atau tiga dimensi ke rana lingkungan nyata di sekitar pengguna secara real time. AR menampilkan objek untuk membantu pengguna dalam menyelesaikan persepsi baru yang dapat memungkinkan berinteraksi dengan rana lingkungan nyata (Rosandy, Rosandy and Komputer, 2019).

2.1.1. Metode pada Augmented Reality

Terdapat dua metode *Augmented Reality* yang dikembangkan sekarang yaitu *Marker Based Tracking* dan *markerless*.

Marker Based Tracking adalah merupakan metode yang memanfaatkan *marker* biasanya berupa ilustrasi hitam putih atau persegi panjang lainnya dengan border tebal hitam dan background putih atau sebuah bias disebut juga gambar. Melalui posisi yang diharapkan pada kamera komputer atau smartphone, komputer atau smartphone akan melakukan proses penampilan dunia virtual 2D atau 3D (Reyhan Achmad Rizal¹, Suardin Gulo¹, Octavriana Della C. Sihombing¹, Ardi Bernandustahi Miduk Napitupulu¹, Amsal Yusuf Gultom¹, 2019).

Markerless Augmented Reality merupakan sebuah teknik dalam menggunakan Vuforia dan OCAR (*Qualcome Augmented Reality*). Yaitu objek citra 3D atau 2 D dapat tampil pada layar perangkat android dan langsung ditempatkan diatas frame Video yang diangkap oleh kamera. Dengan ini teknik ini tidak membutuh sebuah *marker* untuk menampilkan sebuah citra digitak 2D atau 3D.

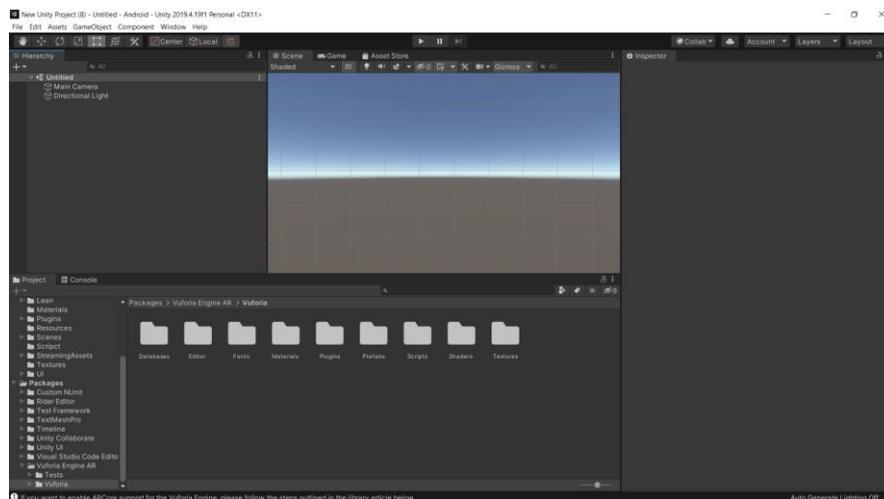
2.2. Android

Android merupakan sebuah sistem operasi yang bersifat open source dengan berbasis kernel linux. Sistem operasi ini dikembangkan oleh Android Inc dan beberapa campur tangan pihak Google. Developer android ini membuat android

bersifat open source agar para programmer dapat membuat aplikasi sedara mudah tanpa batas yang ditentukan (Jubilee enterprise, 2010).

2.3 Unity

merupakan sebuah *engine* yang berjalan di rana *engine game* grafis 2 dimensi maupun 3 dimensi. Unity sendiri menjadi sala satu dari sekian banyak game engine terlaris dalam mengembangkan sebuah game professional. Unity juga tidak hanya digunakan untuk mengembangkan sebuah game engine, melainkan unity dapat juga digunakan untuk mengembangkan sebuah program interaktif 2 dimensi dan 3 dimensi. Keunggulan unity yaitu memilih kemampuan rendering yang terstruktur di dalam sistemnya (Baskara Arya Pranata, 2020).



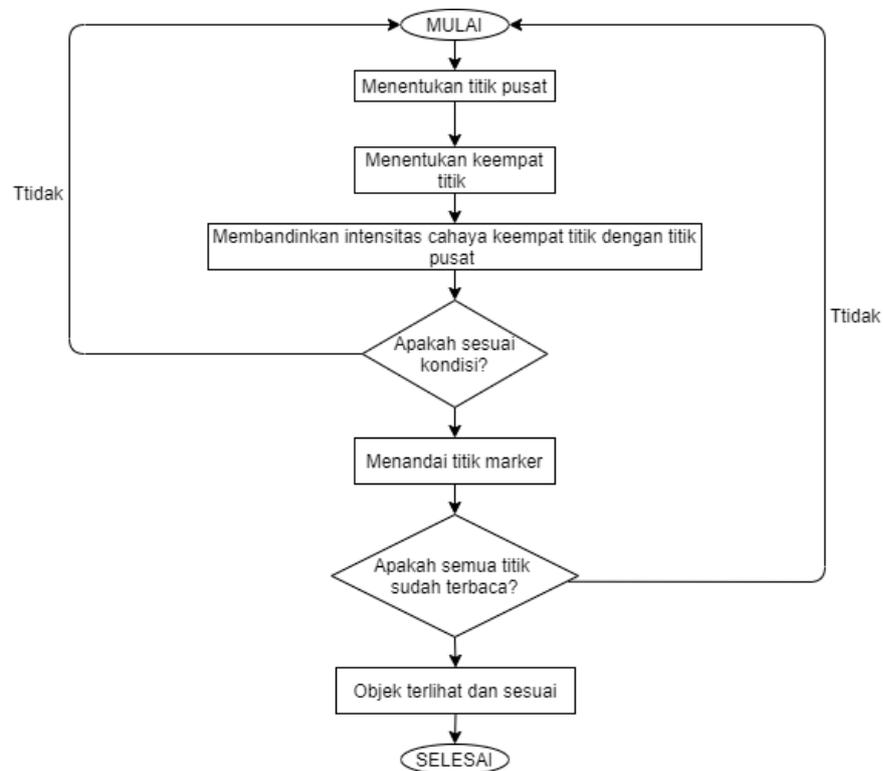
Gambar 2.1 Tampilan Unity

2.4 Algoritma FAST Corner Detection

Corner Detection (deteksi sudut) atau bisa disebut juga intreset point detection merupakan sebuah pendekatan dalam suatu sistem *computer vison* untuk mendeteksi sudut-sudut dari suatu objek. Pendekatan ini biasanya digunakan mendeteksi sebuah objek dan akan ditentukan kekhususan nya pada suatu objek misal dari segi bentuk dan pelacakan objek (Akbar, 2016).

Adapun definisi lain dari deteksi sudut atau ekstraksi titik kunci adalah sebuah pengestrakan informasi paling penting dari data gambar dalam computer vision untuk mencegah pemerosesan semua pixel yang akan dapat menurunkan latensi dan kualitas pemerosesan selanjutnya (Liu, Kao and Delbruck, 2019).

FAST (Features from Accelerated Segmented Test) Corner Detection adalah merupakan algoritma penentu *corner point* yang dikembangkan oleh oleh Edward Rosten, Reid Porter, and Tom Drummond yang bertujuan untuk mempersingkat waktu komputasi secara real time, ada dampak yang terjadi karena mempersingkat waktu komputasi ini yaitu berkurangnya tingkat akurasi pendeteksian sudut. Dalam FAST Corner detection terdapat proses penentuan *corner point*-nya dengan cara merubah *image* menjadi *grayscale*, setelah dirubah maka algoritmanya dapat dijalankan . 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.



Gambar 2.2 Diagram Alir Algoritma FAST Corner Detection

Gambar 2.2 merupakan diagram alir algoritma FAST Corner Detection . Adapun penjelasan tahap-tahapan proses FAST corner detection adalah seperti berikut berikut :

1. Menentukan sebuah titik pusat untuk menampilkan objek.
2. Menentukan keempat titik untuk membandingkan dengan titik pusat.
3. Setelah keempat titik ditemukan, selanjutnya membandingkan dengan intensitas titik pusat dengan keempat titik disekitarnya. Jika titik yang memenuhi syarat paling sedikit 3 titik, maka titik pusat ditemukan.
4. Mengecek apakah kondisi ke empat telah dibandingkan dan sudah sesuai maka dapat dilanjutkan ke proses selanjutnya, jika tidak maka akan diulang ke proses awal.
5. Jika sudah dicek maka selanjutnya adalah menandai titik yang ditampilkan objek, jika pengecekan sudah benar maka dapat dilanjutkan ke proses selanjutnya, jika tidak maka akan diulang ke proses awal.
6. Selanjutnya adalah pengecekan terakhir yaitu mengecek apakah perbandingan semua titik sudah benar.
7. Jika semua proses telah di lalui dengan benar dari awal maka proses terakhir adalah menampilkan objek pada titik yang dituju.

2.5 Bahasa Pemrograman Yang Digunakan

2.5.1 Bahasa pemrograman C#

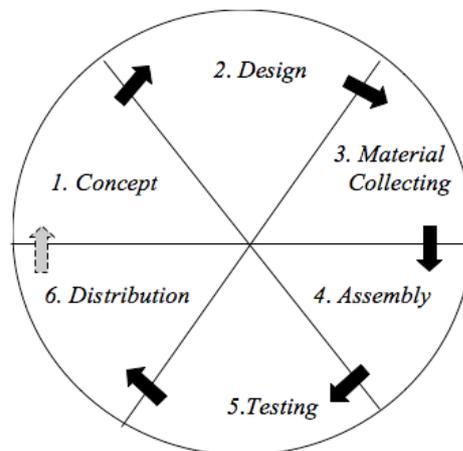
Bahasa pemrograman C# (C sharp) adalah sebuah bahasa pemrograman berorientasi objek yang sederhana untuk penggunaannya. Bahasa pemrograman ini menggunakan editor visual studio. C# juga biasa digunakan untuk membangun aplikasi desktop, mobile, class binary atau bisa juga membangun aplikasi server side pada website (Rahmawati, 2018).

2.6 Metode Pengembangan Perangkat Lunak Yang Digunakan

2.6.1 Multimedia Development Life Cycle

Multimedia Development Life Cycle(MDLC) adalah sebuah model pengembangan perangkat lunak basis multimedia. Metodologi pengembangan multimedia terdiri dari 6 tahap, yaitu *concept*(pengonsepan), *design*(perancangan),

material collecting(pengumpulanmateri), *assembly*(pembuatan), *testing*(pengujian) dan *distribution*(pendistribusian) (Ariesto, 2003).



Gambar 2.3 Multimedia Development Life Cycle

Gambar 2.3 Menggambarkan sebuah diagram alir model pengembangan perangkat lunak basis multimedia yang terdiri dari 6 tahapan yang terdiri dari *concept*, *design*, *material collection*, *assembly*, *testing*, *distribution*. Berikut keterangan tahapan-tahapan dalam model MLDC :

1. *Concept* (Pengkonsepan)

Pengonsepan merupakan tahap awal untuk menentukan dasar dari sebuah proyek multimedia yang akan dibuat dan dikembangkan. Pada tahap ini adalah mengetahui kapasitas pengguna sehingga informasi yang akan disampaikan oleh aplikasi multimedia dapat dengan mudah tersampaikan kepada pengguna. Selain itu, hal yang harus diperhatikan sebelum melakukan tahapan *concept* adalah mentukan jenis penelitian, tempat penelitian, dan lainnya.

2. *Design* (perancangan)

Perancangan merupakan tahap pembuatan rancangan program, tampilan, dan kebutuhan lainnya yang akan dipakai pada tahap selanjutnya yaitu pengumpulan bahan. Karna itu tahapan perancangan harus dilakukan

secara rinci dikarenakan jika tahapan dilakukan secara rinci maka pada tahapan pengumpulan bahan akan mudah dilakukan untuk dilakukan karna desain yang dirancang sudah bagus karna dikerjakan secara rinci dan dapat dilakukan pengumpulan apa saja yang akan dibutuhkan pada tahap selanjutnya. Pada tahap ini biasanya terdapat flowchart dan storyboard untuk menggambarkan alir dan scene satu ke scene lainnya.

3. *Material Collecting*(pengumpulan data)

Pengumpulan bahan merupakan tahapan mengumpulkan bahan-bahan yang akan dipakai pada tahap selanjutnya yaitu pembuatan perangkat lunak. Bahan yang dikumpulkan bisa berupa informasi, gambar, objek 3D, dan lainnya. Tahap ini biasanya dapat dilakukan secara bersamaan dengan tahapan selanjutnya yaitu pembuatan karena kadang ada kemungkinan ada bahan-bahan tambahan yang dibutuhkan pada saat pembuatan perangkat lunak. Tetapi hal ini tidak selalu terjadi tergantung dengan situasi pada saat proses tahapan pembuatan perangkat lunak.

4. *Assembly*(Pembuatan)

Pembuatan merupakan tahap untuk menyatukan semua bahan yang telah dikumpulkan pada tahap pengumpulan bahan menjadi sebuah perangkat lunak berdasarkan pada desain yang telah dibuat sebelumnya yakni *storyboard*.

5. *Testing*(pengujian)

Pengujian merupakan tahap untuk menguji perangkat lunak yang telah dibuat tadi apakah terdapat kekurangan atau bug. Pengujian ini memiliki dua tahap yaitu *alpha* yang dilakukan oleh pembuat perangkat lunak dan *beta* yang melibatkan *end-user*.

6. *Distribution*(pendistribusian)

Pendistribusian merupakan tahap untuk mengevaluasi kembali perangkat lunak yang telah dibuat. Pada tahap ini pembuat akan mengevaluasi kembali apakah kekurangan serta *bug* pada perangkat lunak yang telah dibuat. Tahap pendistribusian ini dapat menjadi acuan untuk tahap concept pada pengembang perangkat lunak selanjutnya dapat dibuat atau diperbaiki menjadi lebih baik.

2.7 Unified Modeling Language(UML)

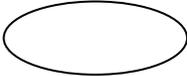
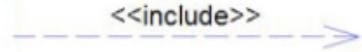
UML adalah sebuah metode untuk pemodelan secara visual sebagai cara untuk merancang perangkat lunak berorientasi objek. UML ini juga merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua komponen dan diagram berbasis pada konteks object oriented (Rahardi, Zaidal and Palaguna, 2019).

UML juga bisa disebut sebagai tool untuk merancang pengembangan sebuah perangkat lunak yang berbasis object oriented. UML juga menyediakan standar penulisan sebuah rancangan atau bisa disebut juga dengan *blue print* (Booch, Rumbaugh and Jacobson, 1996).

1. *Use Case* Diagram

Diagram Use Case merupakan gambaran aktifitas dari pengamatan luar yang dilakukan oleh suatu sistem atau sebuah kejadian-kejadian yang dilakukan sebuah sistem. Yang menjadi personal diagram ini adalah bukan soal bagaimana melakukan tetapi apa yang dilakukan (Purwati and Nz, 2019). Diagram use case memiliki notasi-notasi standar yang telah disediakan seperti yang terlihat pada Tabel 2.1 notasi use case dibawah

Tabel 2.1 Notasi Use Case

| | |
|---|---|
|  | <p>Use Case merupakan sebuah pekerjaan tertentu dalam sebuah sistem.</p> |
|  | <p>Actor merupakan pengguna sistem, atau sistem yang berhubungan dengan sistem lain.</p> |
|  | <p>Assosiasi merupakan sebuah penghubung antara actor dan use case.</p> |
|  | <p>Extend merupakan sebuah relasi yang disediakan use case lain yang digunakan untuk optional</p> |
|  | <p>Include merupakan sebuah relasi yang disediakan use case lain yang digunakan untuk fungsional</p> |

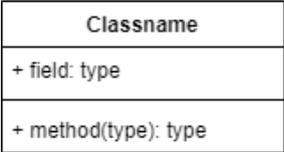
Lanjutan Tabel 2.1 Notasi *Use Case*

| | |
|---|--|
|  | Hubungan antara generalisasi dan spesialisasi antara dua buah use case dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya |
|---|--|

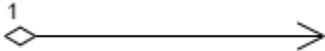
2. *Class* Diagram

Class diagram merupakan sebuah gambaran perangkat lunak atau bisa disebut juga dengan blueprint dari perangkat lunak, yang dimana dalam class diagram ini akan menggambarkan mengenai segala aktivitas pemrosesan dalam perangkat lunak. Dalam class diagram terdapat ketentuan notasi-notasi yang dapat digunakan untuk melakukan sebuah penggambaran (Purwati and Nz, 2019). Berikut notasi class diagram seperti yang terlihat pada Tabel 2.3 Notasi *class* diagram.

Tabel 2. 2 Notasi Class Diagram

| Nama | Keterangan | Simbol |
|--------------------|---|---|
| <i>Class</i> | Class merupakan struktur mendetail mengenai fungsi-sungsi yang terdapat dalam perangkat lunak |  |
| <i>Association</i> | Association merupakan sebuah relasi antar class |  |

Lanjutan Tabel 2.2 Notasi Class Diagram

| | | |
|--------------------|---|---|
| <i>Composition</i> | Relasi antar class yang saling membutuhkan atau bisa di artikan relasi yang bergantung antar class |  |
| <i>Dependency</i> | <i>Dependency</i> digunakan untuk menunjukan operasi pada suatu class |  |
| <i>Aggregation</i> | Aggregation digunakan untuk menjelaskan keseluruhan bagian hubungan antar class atau relasi antar class |  |

3. *Sequence* Diagram

Sequence diagram merupakan sebuah penggambaran fungsional objek perangkat lunak dari waktu ke waktunya sehingga terlihat jalur alur proses fungsi objek dari waktu ke waktu (Purwati and Nz, 2019).

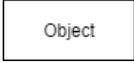
Tabel 2.3 Notasi Sequence Diagram

| Nama | Keterangan | Simbol |
|-------------------|--|---|
| <i>Object</i> | Object merupakan fungsional dari sebuah perangkat lunak |  |
| <i>Actor</i> | Actor merupakan pengguna dari perangkat lunak |  |
| <i>Lifeline</i> | <i>Lifeline</i> digunakan untuk mendikasikan fungsional <i>object</i> dari waktu ke waktu |  |
| <i>Activation</i> | <i>Activation</i> digunakan untuk menggambarkan sebuah fungsional sebuah <i>object</i> dalam waktu |  |
| <i>Message</i> | <i>Message</i> digunakan untuk menjelaskan komunikasi yang akan dilakukan oleh <i>object</i> |  |

4. Activity Diagram

Activity diagram merupakan sebuah penggambaran secara umum dari semua pemrosesan yang terdapat dalam perangkat lunak (Purwati and Nz, 2019).

Tabel 2.4 Notasi Activity Diagram

| Simbol | Keterangan |
|---|--|
|  | Simbol memulai (<i>Start</i>) suatu proses |
|  | Simbol object proses program |
|  | Panah untuk menyatakan alur proses program |
|  | Akhir(<i>End</i>) dari proses program |

2.8. Penelitian Terdahulu

Dalam penyusunan skripsi ini, penulis terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan skripsi ini. Daftar penelitian tersebut ditunjukkan pada Tabel 2.5 Penelitian Terkait dibawah.

Tabel 2. 5 Penelitian Terdahulu

| NO. | NAMA | JUDUL | SUMBER | TAHUN | METODE | KEKURANGAN |
|-----|----------------------------|---|---------------|-------|-----------------------|------------------------------------|
| 1 | Guntur Tiara Wahyu Hidayah | Rancang Bangun Media Ajar Sejarah Pahlawan Pada Uang Kertas Emisi | IIB Darmajaya | 2019 | Marker Based Tracking | Pemerosesan Object 3D yang sedikit |

Lanjutan Tabel 2.5 Penelitian Terdahulu

| | | | | | | |
|---|--------------------------|--|------------------|------|------------|--|
| | | 2016 Menggunakan Metode markerless | | | | |
| 2 | Nadny Yoga Pranata | Augmented Reality Pengenalan Jenis Obat Serta Fungsinya Berdasarkan Android | IIB Darmajaya | 2019 | Markerless | Pemerosesan Object 3D yang sedikit |
| 3 | Wahyu Anggara | Aplikasi Media Promosi Produk Kreatif Dijes Berdasarkan 3D Menggunakan Teknologi Augmented Reality | IIB Darmajaya | 2019 | Markerless | Pemerosesan Object 3D yang sedikit |
| 4 | Adam Fauzan Pratomi | Pengenalan Rumah Adat Provinsi Lampung Dengan Teknologi Markerless Augmented Reality Berbasis Android | IIB Darmajaya | 2019 | Markerless | Pemerosesan Object 3D yang sedikit |
| 5 | M Biondy Dami Pratama | Penerapan Augmented Reality Untuk Media Pembelajaran | IIB Darmajaya | 2019 | Markerless | Pemerosesan Object 3D yang sedikit |

Lanjutan Tabel 2.5 Penelitian Terkait

| | | | | | | |
|---|-------------------------|---|---------------|------|------------|------------------------------------|
| | | Perangkat Keras Komputer Berbasis Android | | | | |
| 6 | Fazriansyah Siregar | Mobile Edukasi Panca Indra Pada Manusia Menggunakan Augmented Reality | IIB Darmajaya | 2018 | Markerless | Pemerosesan Object 3D yang sedikit |
| 7 | Stevanus | Rancang Bangun Aplikasi Visualisi Flora Dan Fauna Pulau Sumatera Menggunakan Teknik Augmented Reality | IIB Darmajaya | 2019 | Markerless | Pemerosesan Object 3D yang sedikit |
| 8 | Ashibilly Satria Gultom | Rancang Bangun Aplikasi Visualisasi 3D Bendera | IIB Darmajaya | 2019 | Markerless | Pemerosesan Object 3D yang sedikit |