

## **BAB II**

### **LANDASAN TEORI**

#### **II.1 Pengertian Sistem Informasi Kampus**

Sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi [6]. Kampus adalah daerah lingkungan bangunan utama perguruan tinggi (universitas, akademi) tempat semua kegiatan belajar mengajar dan administrasi berlangsung [7].

#### **II.2 *Augmented Reality***

Definisi *Augmented Reality* (AR) sebagai sistem yang memiliki karakteristik menggabungkan lingkungan nyata dan *virtual*, berjalan secara interaktif dalam waktu nyata serta integrasi dalam tiga dimensi (3D) [8]. *Augmented Reality* (AR) adalah penggabungan antara benda *virtual* (maya) dan benda nyata secara alami melalui sebuah proses komputerasi, sehingga akan terlihat *real* seperti ada dihadapan pengguna. AR merupakan teknologi yang menggabungkan benda maya tiga dimensi (3D) ke dalam sebuah lingkungan nyata tiga dimensi dan menampilkannya dalam waktu nyata (*real time*) [9].

*Augmented Reality* (AR) pengembangan dari *Virtual Reality* memiliki konsep yang berbeda. Ketika *Virtual Reality* menarik pengguna seakan masuk ke dalam lingkungan 3 dimensi, maka *Augmented Reality* menambahkan realita yang ada dan nyata di dunia kita dengan *object* yang terangkat (*Augmented*), dimana teknologi ini seakan menghilangkan dunia maya 3 dimensi, menyatu dengan dunia nyata. Kegunaan AR bisa diaplikasikan pada berbagai bidang, baik itu untuk keperluan hiburan, kesehatan, robotika, pendidikan, manufaktur lain sebagainya [10]. AR bertujuan untuk menggunakan dunia nyata sebagai dasar dengan kombinasi beberapa teknologi *virtual* dan menambahkan data kontekstual agar pemahaman manusia sebagai penggunanya menjadi semakin jelas. Data kontekstual ini

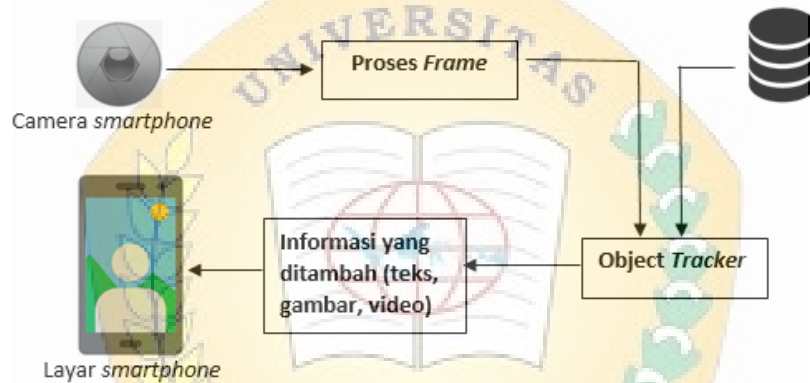
berupa komentar audio, data lokasi, konteks sejarah, atau dalam bentuk lainnya. AR berfungsi untuk meningkatkan persepsi pengguna dari dunia yang ada disekitarnya dan menjadikan sebagian dunia *virtual* dan nyata sebagai antarmuka yang baru yang yang mampu menampilkan informasi sesuai kebutuhan pengguna [11].

*Augmented Reality* (AR) diciptakan dengan berbagai cara, tantangan utama adalah bagaimana membuat kombinasi dari dunia nyata dan maya sehalus mungkin. Berdasarkan apa yang digunakan untuk memicu elemen *virtual* untuk muncul di dunia nyata, AR dapat diklasifikasikan sebagai berikut [12]:

1. *GPS Coordinates*, menggunakan koordinat GPS, kompas dan Akselerometer (alat pengukuran posisi) untuk menemukan posisi yang tepat dari pengguna, termasuk titik koordinat yang pengguna lihat dan beberapa obyek atau lainnya dari posisi yang sama
2. *Black and White Markers*, sebagai contoh yaitu gambar sederhana kode QR hitam putih untuk memproyeksikan obyek *virtual* pada pengguna
3. *Image Markers*, menggunakan kamera perangkat seluler untuk menemukan gambar yang telah ditentukan (target) kemudian memproyeksikan obyek *virtual* di atasnya. Jenis AR ini telah menggantikan *Black And White Makers*
4. *Real-time Markers*, pengguna membuat dan mendefinisikan gambar mereka dengan kamera ponsel untuk memproyeksikan *object virtual* apapun di dalamnya
5. *Facial Recognition*, melalui kamera menangkap gerakan wajah untuk melakukan tindakan tertentu dalam sebuah permintaan, misalnya untuk memberikan ekspresi wajah ke avatar *virtual*
6. *SLAM (Simulation Location and Mapping)*, teknologi ini memahami dunia fisik melalui titik fitur, sehingga memungkinkan aplikasi AR untuk mengenali objek dan pemandangan 3D, serta melacak dunia secara instan dan melapisi augmentasi interaktif digital
7. *Beacons*, eBeacons, RFID, dan NFC adalah sistem identifikasi yang menggunakan frekuensi radio atau bluetooth, mirip dengan koordinat GPS untuk memicu elemen AR.

### II.3 Cara Kerja *Augmented Reality*

Pada dasarnya, prinsip kerja *Augmented Reality* adalah pelacakan (*tracking*) dan dan rekonstruksi (*reconstruction*). Pada mulanya *marker* dideteksi menggunakan kamera. Cara deteksi dapat melibatkan berbagai macam algoritma misalnya *edge detection*, atau algoritma *image processing*. Data yang diperoleh dari proses pelacakan digunakan dalam rekonstruksi sistem koordinat di dunia nyata. Disamping menambahkan obyek ke dalam lingkungan nyata, *Augmented Reality* juga dapat menghilangkan obyek nyata dalam bentuk *virtual*. Dengan menutupi obyek nyata tersebut dengan desain grafis sesuai lingkungannya, maka obyek nyata akan tersembunyi dari pengguna [13]. Adapun alur cara kerja *Augmented Reality* dapat dilihat pada gambar berikut [14]:



**Gambar 2. 1** Alur Kerja *Augmented Reality*

Alur kerja aplikasi AR pada gambar 2.1 secara umum dimulai dari pengambilan *marker* (gambar) dengan kamera. Gambar tersebut dikenali berdasarkan *feature* yang dimiliki, kemudian masuk ke dalam *object tracker* yang disediakan oleh *Android Software Development Kit* (SDK). Di sisi lain, gambar tersebut telah didaftarkan dan disimpan ke dalam *database*. *Object tracker* selanjutnya akan melacak dan mencocokkan gambar tersebut agar dapat menampilkan informasi yang sesuai. Hasil keluaran pelacakan gambar segera ditampilkan ke dalam layar ponsel. Informasi yang ditampilkan melekat pada gambar bersangkutan secara *real time*.

### II.3.1 Komponen Pembuatan *Augmented Reality*

Adapun komponen yang diperlukan dalam pembuatan dan pengembangan aplikasi AR yaitu [11]:

1. Komputer berfungsi sebagai perangkat yang digunakan untuk tempat penginstalan aplikasi yang dibutuhkan dalam pembuatan *Augmented Reality*
2. Kamera *smartphone* Android merupakan perangkat yang berfungsi sebagai *recording sensor*, kamera *smartphone* Android digunakan menampilkan obyek
3. *Marker* (gambar) digunakan untuk proses *tracking* pada saat aplikasi dijalankan.

### II.4 ARCore

Google memperkenalkan platform AR Android baru pada Maret 2018, yang disebut ARCore [15]. ARCore dibangun dengan 3 kemampuan mendasar yaitu [16]:

1. Pelacakan (*Tracking*), ARCore memungkinkan untuk mengetahui dan melacak posisi dengan mengidentifikasi titik fitur / visual dari *object* atau gambar dengan perangkat kamera *smartphone*. Saat menggunakan ARCore dapat menentukan posisi dan orientasi *smartphone* saat bergerak.
2. Memahami lingkungan (*Understanding the Environment*), mendeteksi berbagai permukaan (vertikal, horizontal dan miring) dengan menggunakan *smartphone*
3. Estimasi pencahayaan (*Brightness Estimation*), estimasi cahaya yang memungkinkan *smartphone* untuk memperkirakan cahaya sekitar untuk mengintegrasikan elemen 3D dengan sempurna ke dunia nyata. Pencahayaan hanya memungkinkan posisi pengguna dengan lingkungan posisi obyek *virtual* dengan cara membaca tingkat cahaya.

### II.5 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) untuk platform Android. Android studio bersifat *free* dibawah Apache License 2.0 dan didesain khusus untuk *android development* serta dapat di download untuk windows, Mac OS X, Linux[17]. Android Studio merupakan

pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio direncanakan untuk menggantikan Eclipse sebagai IDE resmi untuk pengembangan aplikasi Android. Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan *Ant*, Android Studio menggunakan *Gradle* sebagai *build environment*. Fitur-fitur lainnya adalah sebagai berikut [18]:

1. Menggunakan Gradle-based build sistem yang fleksibel
2. Bisa mem-build *multiple* APK
3. *Template support* untuk *Google Services* dan berbagai macam tipe perangkat.
4. Layout editor yang lebih bagus.
5. Build-in *support* untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
6. *Import library* langsung dari *Maven repository*.

## II.6 Android Software Development Kit (SDK)

Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan tutorial. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk desktop modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin *Android Development Tools* (ADT), dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan pengendalian perangkat Android (misalnya, *reboot*, menginstal paket perangkat lunak dengan jarak jauh) [19].

## II.7 Sistem Operasi Android

Android adalah sistem operasi Linux yang dirancang untuk perangkat layar sentuh seperti telepon pintar dan komputer *tablet*. Android awalnya dikembangkan

oleh Android Inc. dengan dukungan pendanaan dari Google yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007. Saat ini sistem operasi ini telah berkembang sangat cepat sebagai telepon pintar yang memanjakan penggunanya. Banyak aplikasi yang dapat diunduh pada sistem operasi ini untuk melengkapi kebutuhan penggunanya [20]. Minimum SDK android yang mendukung ARCore sebagai berikut:

**Tabel 2. 1 Minimum SDK yang Mendukung ARCore**

No	Minimum SDK	Deskripsi
1.	Android versi 7.0	Perangkat harus menjalankan Android 7.0 atau yang lebih versi yang baru

## II.8 Java

Bahasa pemrograman java adalah bahasa pemrograman berorientasi objek tingkat tinggi yang memungkinkan pengembangnya untuk menulis kode WORA (tulis sekali, jalankan di mana saja) [21]. Java adalah bahasa pemrograman yang pertama kali dirilis oleh Sun Microsystems pada tahun 1995. Java dapat ditemukan di berbagai jenis perangkat mulai dari ponsel pintar, hingga komputer mainframe. Java tidak mengkompilasi ke kode prosesor asli melainkan bergantung pada "mesin virtual" yang memahami format perantara yang disebut Java *bytecode*. Setiap platform yang menjalankan Java membutuhkan implementasi mesin virtual (VM). Di android, VM asli disebut Dalvik. Google juga telah mulai mempratinjau VM generasi berikutnya yang disebut ART. Tugas mesin virtual ini adalah menafsirkan *bytecode* yang sebenarnya hanyalah sekumpulan instruksi yang mirip dengan kode mesin yang ditemukan di CPU, dan menjalankan program pada prosesor. VM menggunakan berbagai teknologi termasuk kompilasi *just-in-time* (JIT) dan kompilasi sebelumnya (AOT) untuk mempercepat proses. Ini semua berarti bahwa kita dapat mengembangkan aplikasi Android di Windows, Linux atau OS X dan kompiler Java mengubah kode sumber menjadi *bytecode*. Ini pada gilirannya dijalankan pada VM bawaan Android. Java digunakan untuk pengembangan Android karena, itu adalah bahasa yang terkenal di antara para pengembang, tidak memiliki komplikasi aritmatika pointer. Karena ini berjalan di VM, jadi tidak perlu



mengkompilasi ulang kode untuk setiap perangkat tempat kode itu digunakan. Meskipun kecepatan adalah masalah bagi JAVA, namun popularitas dan keunggulannya melebihi kecepatan [22].

## II.9 Blender 3D

*Software* Blender 3D berawal pada tahun 1988 di Dutch didirikan oleh Ton Roosendaal yang menjadi salah satu pendiri studio animasi “Neo Geo”. Pada tanggal 13 Oktober 2002, *software* blender 3D dirilis di bawah lisensi GNU GPL [23]. Blender merupakan sebuah *software* yang bersifat *open source* digunakan untuk membuat film visual, animasi 3D, dan efek khusus. Blender juga *software* yang multiplatform bisa digunakan pada berbagai sistem operasi diantaranya windows, linux, dan mac. Walau bisa digunakan di berbagai *platform file* yang di buat dengan menggunakan Blender tetap tidak akan berubah akan tetap sama dan bisa dibuka dari *platform* satu ke *platform* lainnya. Blender memiliki banyak fitur untuk mendukung pembuatan desain 3D diantaranya 3D *modeling*, *UV Unwrapping*, *texturing*, *raster graphics editing*, *rigging and skinning*, *fluid and smoke simulation*, *particle simulation*, *soft body simulation*, *sculpting*, *animating*, *match moving*, *camera tracking*, *rendering*, *video editing and compositing* [24].

## II.10 Augmented Image Database

*Augmented Image* adalah salah satu *tools* yang digunakan dalam membuat *Augmented Reality* yang dikembangkan oleh Google di dalam ARCore. *Augmented Image* pada ARCore dapat melacak hingga 20 gambar secara bersamaan. Akan tetapi tidak dapat terdeteksi atau melacak gambar contoh yang sama secara bersamaan. Setiap *database Augmented Image* dapat menyimpan informasi hingga 1.000 gambar referensi. Tidak ada batasan jumlah *database*, tetapi hanya satu *database* yang dapat aktif pada waktu tertentu. *Augmented Image* dalam ARCore dapat melacak gambar yang diletakkan di tempat seperti gambar yang ada di dinding, berupa majalah yang ada di atas meja, iklan di bus yang lewat atau gambar di obyek datar yang dipegang oleh pengguna sambil menggerakkan tangan.

Setelah mulai melacak gambar, ARCore memberikan perkiraan untuk posisi dan orientasi gambar pada setiap frame. ARCore terus menyempurnakan perkiraan

ini karena mengumpulkan lebih banyak data. Setelah gambar terdeteksi, ARCore melanjutkan "pelacakan" posisi dan orientasi gambar, meskipun gambar tersebut keluar sementara dari tampilan kamera karena pengguna telah memindahkan perangkat. Dengan demikian, ARCore menganggap bahwa posisi dan orientasi gambar bersifat statis, dan gambar tersebut tidak berpindah melalui lingkungan. Semua pelacakan terjadi di perangkat. Tidak perlu koneksi internet untuk mendeteksi dan melacak gambar[25].


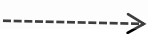
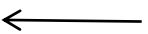
## II.11 *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* merupakan pemodelan yang mampu memodelkan berorientasi obyek dan desain. Selain itu UML sangat populer untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan sistem perangkat lunak yang telah terbukti memiliki kinerja yang baik [26]. Adapun beberapa diagram dalam UML yang penulis gunakan antara lain:

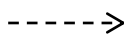



### a. Diagram *Use Case*

Diagram *Use Case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dan sistem [27].

**Tabel 2. 2 Simbol *Use Case***

Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasi himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use Case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang akan bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana <i>object</i> anak ( <i>descendent</i> ) berbagi perilaku dengan struktur data dari <i>object</i> yang ada di dalamnya <i>object</i> induk.



	<i>include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan obyek yang satu dengan yang lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

b. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram (ERD)* merupakan diagram yang terdiri dari elemen dasar yang diwakili oleh entitas, atribut dan hubungan antar entitas [28].


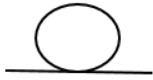




**Tabel 2. 3 Simbol *Entity Relationship Diagram***

Simbol	Nama	Keterangan
	Entitas	Persegi panjang yang menyatakan himpunan entitas.
	Relasi	Belah ketupat, menyatakan himpunan relasi.
	Atribut	Elips, menyatakan atribut (atribut berfungsi sebagai <i>key</i> ).
	Garis	Sebagai penghubung antara entitas, relasi dan atribut.

c. *Diagram Sequence*

*Diagram sequence* merupakan pendekatan pemodelan untuk memvisualisasikan eksekusi perilaku dari sebuah sistem [29].

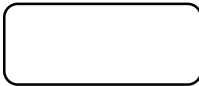




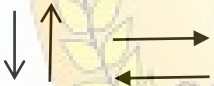
Tabel 2. 4 Simbol Diagram *Sequence*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
	<i>Boundary Class</i>	Berupa tepi dari sistem, seperti user interface atau suatu alat yang berinteraksi dengan sistem yang lain.
	<i>Control Class</i>	element mengatur aliran dari informasi untuk sebuah skenario. Obyek ini umumnya mengatur perilaku dan perilaku bisnis.
	<i>Lifeline</i>	Menggambarkan tempat mulai dan berakhirnya <i>message</i> .
	<i>A message</i>	Menggambarkan pengiriman pesan.

d. Diagram *Activity*

Diagram *activity* yang dirancang digunakan dalam pemodelan bisnis dan untuk mengembangkan sistem perangkat lunak. Diagram ini mewakili aktivitas alur kerja dari sistem perangkat lunak. Aktivitas yang merupakan tindakan yang digambarkan sebagai aliran eksekusi atau perilaku sistem perangkat lunak [30].

Tabel 2. 5 Simbol Diagram Activity

Simbol	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
	<i>Initial Node</i>	Bagaimana <i>object</i> dibentuk atau diawali
	<i>Activity Final Node</i>	Bagaimana <i>object</i> dibentuk atau diakhiri
	<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
	<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.