



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Judul**

##### **2.1.1 Pengertian Pengenalan**

Menurut Saniansah dan B. Hendi (2011:79) pengenalan adalah nama, istilah, lambang, symbol atau desain, atau kombinasi dari hal-hal tersebut.

Menurut Kamus Besar Bahasa Indonesia (KBBI) menyatakan “Pengenalan adalah proses, cara, perbuatan mengenal atau mengenali: pengenalan nya tentang masalah kehidupan kurang sempurna

Dari pernyataan diatas dapat disimpulkan bahwa pengenalan adalah unsur-unsur kesadaran dalam memberitahu agar dapat dikenali.

##### **2.1.2 Pengertian Kampus**

Menurut Walanda (2018:51) kampus adalah ruang terbuka dan miniatur Negara pertama bagi gerakan mahasiswa untuk berkiprah di publik. Sedangkan Menurut Ciputra (2008:75) kampus adalah terminal utama generasi muda terdidik untuk menjadi tenaga kerja terdidik.

Dari pernyataan diatas dapat disimpulkan bahwa kampus adalah tempat mahasiswa mencari ilmu untuk menunjang di dunia secara luas dalam mencari pekerjaan.

##### **2.1.3 Pengertian Gedung**

Menurut Kamus Besar Bahasa Indonesia (KBBI) menyatakan “Gedung adalah bangunan tembok dan sebagainya yang berukuran besar sebagai tempat kegiatan, seperti perkantoran, pertemuan, perniagaan, pertunjukan, olahraga, dan sebagainya.

#### **2.2 Teori Khusus**

##### **2.2.1 Pengertian *Markerless Marker***

Menurut Riski dalam Hidayat & Nurjayandi (2015:49), “*Markerless Augmented Reality* merupakan salah satu metode *Augmented Reality* tanpa



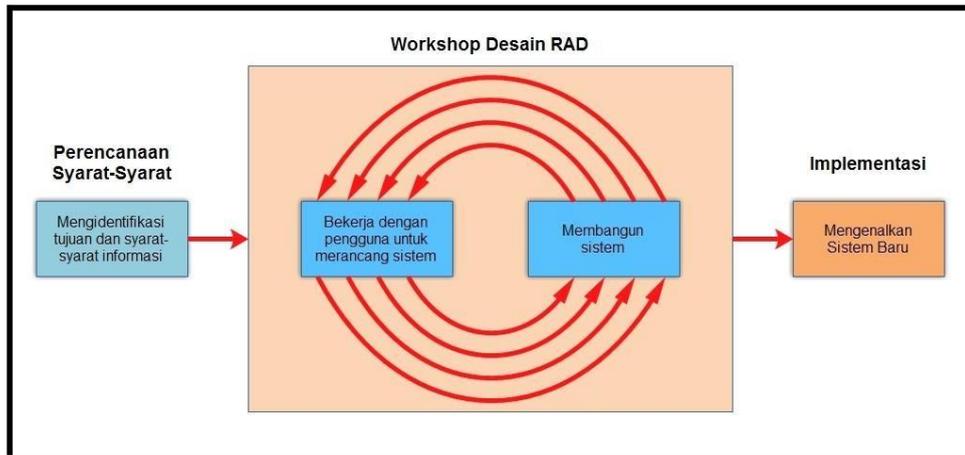
menggunakan *framemarker* sebagai obyek yang dideteksi. Dengan adanya *Markerless Augmented Reality* maka penggunaan *marker* sebagai *tracking object* yang selama ini menghabiskan ruang, akan digantikan dengan gambar, atau permukaan apapun yang berisi dengan tulisan, logo, atau gambar sebagai *tracking object* (obyek yang dilacak) agar dapat langsung melibatkan obyek yang dilacak tersebut sehingga dapat terlihat hidup dan interaktif, juga tidak lagi mengurangi efisiensi ruang dengan adanya *marker*.”

Jadi dengan metode *markerless* untuk menampilkan elemen-elemen digital ataupun model 3D pada aplikasi *augmented reality* tidak perlu menggunakan *marker* dengan bentuk hitam persegi dengan latarbelakang putih sebagai *tracking object* seperti pada metode *markerbased*. yang mana hal ini akan membutuhkan ruang pada *object* yang dilacak sehingga tampilan design dari sebuah *tracking object* menjadi relative kurang menarik.

### **2.2.2 Pengertian *Rapid Application Development (RAD)***

Menurut Kendall dalam Widiyanto (2018:36), “RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi.”

Dari definisi konsep RAD ini, dapat dilihat bahwa pengembangan aplikasi dengan menggunakan metode RAD dapat dilakukan dalam waktu yang relatif lebih cepat. Sesuai dengan metodologi RAD berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi dapat di lihat pada gambar dibawah ini.” (Meidyan Permata Putri dan Hendra Effendi, 2018:131)



**Gambar 2.1. Tahapan RAD**

Tahapan RAD terdiri dari 3 tahap yang terstruktur dan saling bergantung disetiap tahap, yaitu :

- 1) *Requirements Planning* (Perencanaan Persyaratan).
  - a) Pengguna dan analis bertemu untuk mengidentifikasi tujuan dari aplikasi atau sistem
  - b) Berorientasi pada pemecahan masalah bisnis.
  
- 2) *Design Workshop*.
  - a) Fase desain dan menyempurnakan.
  - b) Gunakan kelompok pendukung keputusan sistem untuk membantu pengguna setuju pada desain.
  - c) Programmer dan analis membangun dan menunjukkan tampilan visual desain dan alur kerja pengguna.
  - d) Pengguna menanggapi prototipe kerja aktual.
  - e) Analis menyempurnakan modul dirancang berdasarkan tanggapan pengguna.
  
- 3) *Implementation* (Penerapan).
  - a) Sebagai sistem yang baru dibangun, sistem baru atau parsial diuji dan diperkenalkan kepada organisasi.
  - b) Ketika membuat sistem baru, tidak perlu untuk menjalankan sistem yang lama secara *parallel*. (Meidyan Permata Putri dan Hendra Effendi, 2018:131).



---

### 2.2.3 Pengertian *Location Based Service* (LBS)

Menurut IMAI dalam Agustina, *et al* (2016:54), “mendefinisikan bahwa *Location Based Service* (LBS) mengacu pada “sekumpulan aplikasi yang mengeksplorasi pengetahuan / informasi dari lokasi geografis perangkat mobile untuk mendapatkan layanan berdasarkan informasi tersebut.”

Menurut Beaubrun dalam Fauzi (2015:251), “mendefinisikan bahwa layanan yang diberikan oleh LBS dapat diklasifikasikan menurut fungsi maupun menurut dan lokasi keberadaan pengguna informasi tersebut. Layanan-layanan itu antara lain: (1) *Map Service*; (2) *City Guide Service*; (3) *Yellow Page Service*; (4) *Navigation Service*; (5) *Location/Contextaware Information Service*.”

Dari pernyataan diatas dapat disimpulkan bahwa LBS adalah sekumpulan aplikasi memudahkan pengguna perangkat mobile mengatur dan memilih layanan sesuai kebutuhan dan dapat dimanfaatkan untuk memberikan berbagai layanan seperti informasi kondisi lingkungan (kemacetan lalu lintas, cuaca, lokasi fasilitas umum terdekat), maupun promosi produk dan jasa.

## 2.3 Teori Program

### 2.3.1 *Augmented Reality*

Menurut Milgram & Kishino dalam Utomo, *et al.* (2017:225), “*Augmented Reality* atau disebut juga sebagai *Mixed Reality* (MR) adalah sebuah cabang dari teknologi yang menyangkut *VirtualReality* (VR) dan melibatkan penggabungan antara dunia real dan dunia virtual.”

Menurut Yuen dalam Mustaqim & Kurniawan (2017:137), “*Augmented Reality* apabila diartikan kedalam Bahasa Indonesia adalah “realitas tertambat”. *Augmented reality* adalah proyeksi materi hasil pengolahan komputer, seperti tulisan, gambar, dan video ke dalam perspektif manusia di dunia nyata.”

Dari penjelasan diatas dapat disimpulkan bahwa *Augmented Reality* adalah sebuah teknologi yang menggabungkan dunia nyata dan dunia maya secara interaktif dan *realtime*.



### 2.3.2 Android

Menurut Yudhanto dan Wijayanto (2017:1), “mendefinisikan Android adalah Sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet.”

Menurut Safaat dalam Rizal, *et al.*(2014:82), “mendefinisikan Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.”

Dari pernyataan diatas dapat disimpulkan bahwa android adalah platform mobile berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh digunakan untuk pengembang buat menciptakan aplikasi mereka.

### 2.3.3 Google Maps API

Adi dan Setiadi (2014:1249) menjelaskan bahwa *Google maps* API adalah *library* JavaScript. Dengan menggunakan *Google maps* API jadi kita hanya berkonsentrasi tentang data dan biarkan urusan peta ditangani oleh google, sehingga dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal.

### 2.3.4 Unified Modelling Language (UML)

Menurut Salahuddin (2014:137), “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

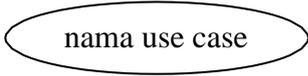
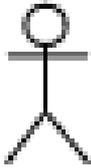
UML hanya berfungsi untuk melakukan pemodelan. Sehingga penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan untuk metodologi berorientasi objek.



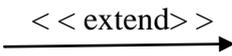
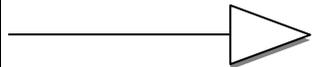
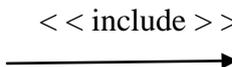
### 2.3.5 Use Case Diagram

Menurut Salahuddin (2014:155), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

**Tabel 2.1** Simbol *Use Case* Diagram

No	Simbol	Deskripsi
1	<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal fase nama <i>use case</i>
2	<p>Aktor / <i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal fase nama aktor
3	<p>Asosiasi / <i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor

Tabel 2.1 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
4	Ekstensi / <i>extend</i>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i>
5	Generalisasi / <i>generalization</i>  	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan / <i>include</i> / <i>uses</i>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

### 2.3.6 Activity Diagram

Menurut Salahuddin (2014:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

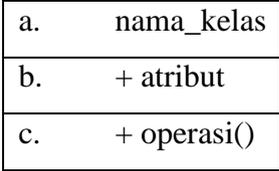
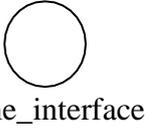
Tabel 2.2 Simbol *Activity Diagram*

No	Simbol	Deskripsi
1	<p><i>Start / status awal (Initial State)</i></p> 	<i>Start</i> atau <i>initial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup
2	<p><i>End / status akhir (final state)</i></p> 	<i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu sistem.
3	<p><i>Event</i></p> 	<i>Event</i> adalah kegiatan yang menyebabkan berubahnya status mesin.
4	<p><i>State</i></p> 	Sistem pada waktu tertentu. <i>State</i> dapat berubah jika ada <i>event</i> tertentu yang memicu perubahan tersebut.

### 2.3.7 Class Diagram

Menurut Salahuddin (2014:143), “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi”.

Tabel 2.3 Simbol *Class Diagram*

No	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem
2	Antarmuka / <i>interface</i> 	sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang saat digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6	Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

### 2.3.8 Sequence Diagram

Menurut Salahuddin (2014:165), “Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan

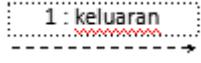


*message* yang dikirimkan dan diterima antar objek”. Untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstantiasi menjadi objek itu dan juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

**Tabel 2.4** Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1.		Garis hidup <i>/ lifeline</i>	Menyatakan kehidupan suatu objek
2.	Atau 	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
3.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
4.		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
5.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode

**Tabel 2.4** Simbol Sequence Diagram

No	Simbol	Nama	Keterangan
6.		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
7.		Pesan tipe keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian

(Sumber : Rosa dan Shalahuddin, 2018:165)