

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Augmented reality adalah sebuah teknologi mediasi yang menggabungkan objek yang berada di dunia nyata dan dunia 3 dimensi yang ditingkatkan oleh persepsi yang dihasilkan komputer. *Augmented reality* dapat diartikan sebagai sistem yang memiliki dasar kombinasi dunia nyata serta virtual, interaksi waktu nyata, dan 3 dimensi yang akurat dari objek dan dunia nyata.

2.2 Taman Purbakala Pugung Raharjo

Taman Wisata Purbakala Pugung Raharjo adalah situs arkeologi yang terletak di Desa Pugung Raharjo, Kecamatan Sekampung Udik, Kabupaten Lampung Timur Provinsi Lampung. Saat ini Taman Wisata Purbakala Pugung Raharjo sering ditawarkan dalam paket wisata yang ada di Lampung (Praleka, I. G. P. C. 2019).

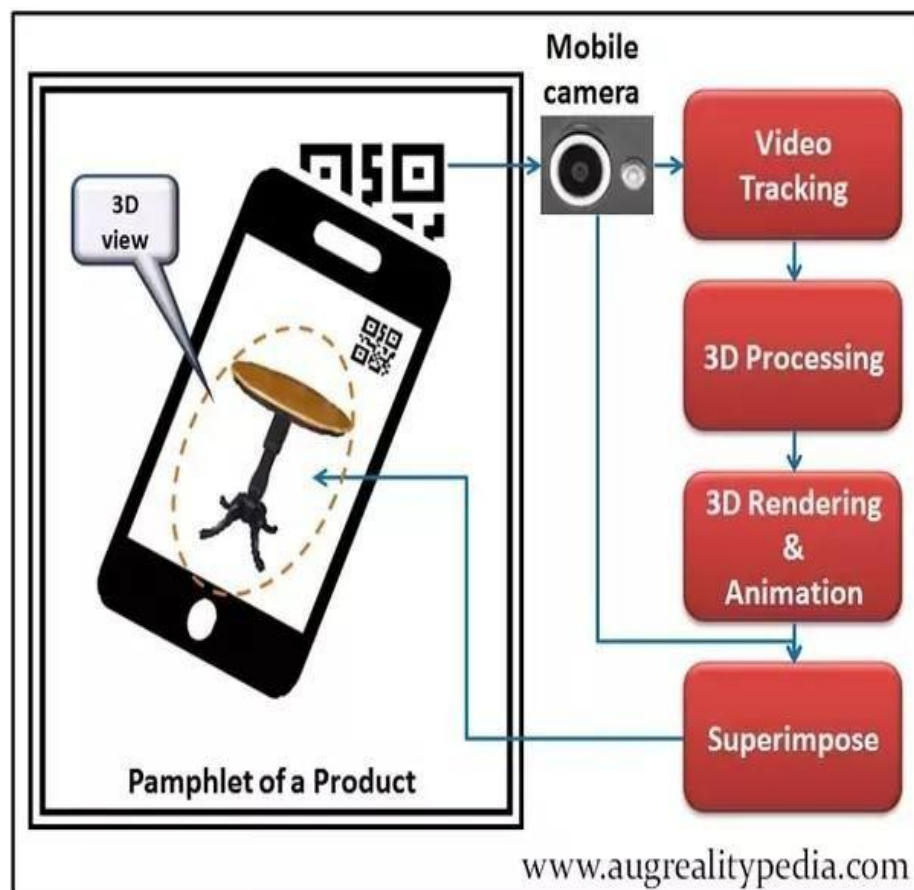
Pengguna media digital dan internet di Indonesia cukup besar yaitu 171,17 juta dari total populasi 264,16 juta orang atau setara dengan 64,8% yang merupakan pengguna aktif. Terbuka luasnya era digital saat ini memberikan akses, dampak, serta peluang yang sangat besar dalam mengembangkan aplikasi Taman Purbakala Pugung Raharjo yang merupakan salah satu cagar budaya yang sangat penting untuk dilestarikan dan merupakan warisan budaya untuk jati diri dan dikaitkan dengan fungsi pendidikan, (Azima, M. F., & Laila, S. N. 2020).

2.3 Augmented reality

Augmented reality digunakan untuk membantu penelitian arkeologi. Dengan menambah fitur objek arkeologi ke dalam ruang lingkup modern sehingga *Augmented reality* memungkinkan arkeolog untuk merumuskan segala bentuk yang terkait penelitian menjadi struktur yang dapat dipelajari. Berdasarkan hal tersebut dapat dijelaskan bahwa teknologi *Augmented reality* dapat diterapkan dalam banyak hal guna membantu pekerjaan.

2.3.1 Prinsip Kerja Augmented reality

Didalam *Augmented reality* terdapat beberapa prinsip kerja yang menjelaskan alur kerja pada sistem *Augmented reality*, gambar dibawah ini menjelaskan prinsip kerja pada sistem *Augmented reality*.



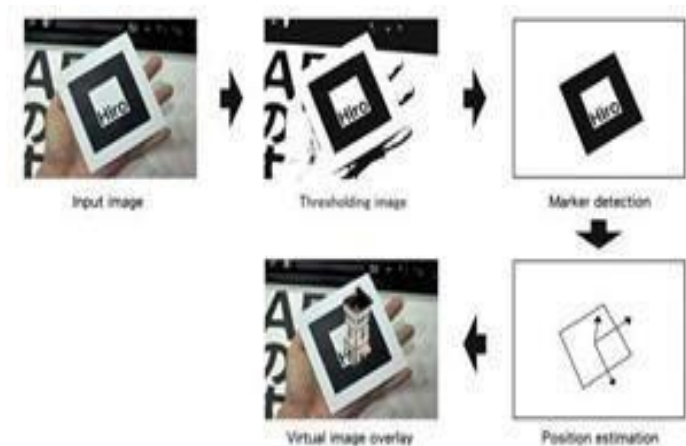
Gambar 2.1 Gambar Prinsip Kerja *Augmented reality* (sumber gambar: www.augrealitypedia.com)

Deskripsi gambar adalah sebagai berikut:

1. Kamera menangkap data dari marker dalam dunia nyata.
2. Mengirimkan informasi ke komputer.
3. *Software* pada komputer akan melacak bentuk kotak dari *marker* dan mendeteksi beberapa *videoframe*.
4. Bila kotak telah ditemukan, maka *software* menggunakan perhitungan matematis untuk menghitung posisi dari kamera relative terhadap kotak hitam pada *marker*.
5. Setelah dilakukan maka model grafis akan dimunculkan pada posisi yang sama dan berada di dalam lingkup kotak hitam lalu ditampilkan ke layar untuk melihat grafis dalam dunia nyata.

2.3.2 Marker

Marker merupakan sebuah gambar berpola khusus yang sudah dikenali oleh template memory artoolkit. Dimana marker tersebut berfungsi untuk dibaca dan dikenali oleh kamera lalu dicocokkan dengan template artoolkit. Setelah itu, baru kamera akan melakukan render objek 3d diatas marker. Pada umumnya marker yang bisa dikenali artoolkit hanya marker dengan pola berbentuk kotak dengan bingkai hitam didalamnya. Akan tetapi seiring berkembangnya zaman banyak pengembang augmented reality yang dapat membuat marker tanpa bingkai hitam(Yulsilviana, E., Basrie, B., & Saputra, A. W.2017)



Gambar 2.2 Gambar Contoh *Marker*

Marker merupakan bagian yang sangat penting pada aplikasi berbasis *Augmented reality* karena objek yang dibuat hanya dapat dipanggil lewat proses scan pada *marker*.

2.4 Android

Android merupakan sebuah sistem operasi yang digunakan *smartphone* dan tablet. Sistem operasi dapat digunakan sebagai penghubung antara piranti dan pengguna, sehingga pengguna dapat berinteraksi dengan piranti (*device*) dan dapat menjalankan aplikasi yang tersedia pada *device* (Sari dan Ali, 2019).

Dari uraian diatas, dapat disimpulkan bahwa *Android* merupakan sistem operasi yang dikembangkan guna meningkatkan performa *smartphone* dan tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai macam piranti bergerak.

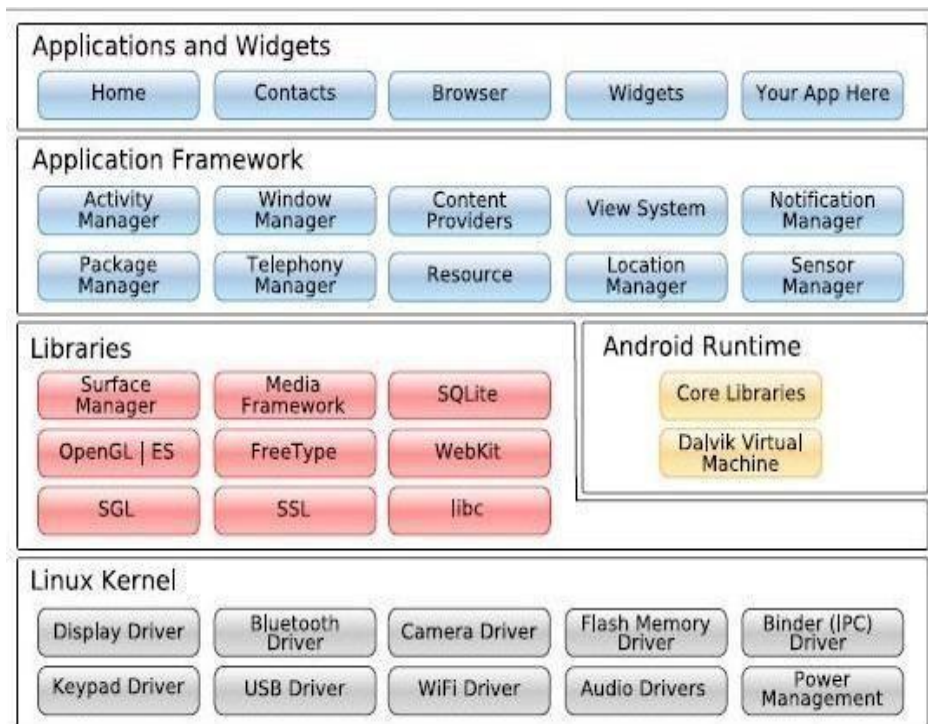
Android merupakan suatu sistem operasi untuk perangkat *mobile* yang berbasis linux mencakup *middleware*, aplikasi dan sistem operasi, (Safaat 2019). Sebagai *OpenSource* dan bebas dalam memodifikasi, didalam *Android* tidak memiliki ketentuan yang tetap didalam konfigurasi *Hardware* dan *Software*. Fitur-fitur yang terdapat di dalam *Android* antara lain:

1. *Storage*-Menggunakan SQLite, *database*
2. Chrome'sV8JavaScript.

3. *Mediasupport*–Termasuk mendukung untuk beberapa media berikut: H.263, H.264 (dalam bentuk 3GP or MP4), MPEG-4SP, AMR, AMRWB (dalam bentuk 3GP), AAC, HE-AAC (dalam bentuk MP4 atau 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, GIF, dan BMP.
4. *Hardware support*– Sensor akselerasi, Kamera, Kompas Digital, Sensor Kedekatan, GPS.
5. *Multi-touch*–Mendukung multi-touchscreens.
6. *Multi-tasking*–Mendukung aplikasi multi-tasking.
7. *Flash-support*–Android 2.3 mendukung *Flash* 10.1.
8. *Tethering*–Mendukung pembagian dari koneksi Internet sebagai *wired/ wireless hotspot*.
9. *Playstore*– katalog aplikasi yang dapat di-*download* dan di-*install* pada telepon seluler secara *online*, tanpa menggunakan PC (*Personal Computer*).
10. Lingkungan pengembangan yang kaya, termasuk *emulator*, peralatan *debugging*, dan *plugin* untuk Eclipse IDE.
11. *Connectivity*-Mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS.
12. *Bluetooth* (termasuk A2DP dan AVRCP), WiFi, LTE, dan WiMax.
13. *Messaging*–Mendukung SMS dan MMS.
14. *Web Browser*–Berbasiskan *open-source* Web Kit, bersama mesin.

2.4.1 **Arsitektur Android**

Secara garis besar Arsitektur *Android* dapat dijelaskan dan digambarkan seperti pada gambar 2.2 (Safaat 2019):



Gambar 2.3 Gambar Arsitektur *Android*

a. Application dan Widgets

Application dan *Widgets* merupakan *layer* pengguna yang berhubungan dengan aplikasi saja, *download* aplikasi melakukan instalasi dan menjalankan aplikasi tersebut. Di *layer* ini terdapat aplikasi inti termasuk program SMS, klien email, peta, kalender, *browser*, kontak, dan lain-lain. Hampir semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b. *Application Frameworks*

Android memberi atau menawarkan kepada pengembang kemampuan untuk membangun aplikasi yang inovatif dan bagus, *Android* adalah "*Open Development Platform*" Pengembang bebas untuk mengatur alarm, background service, mengakses perangkat keras, informasi resource akses, notification, menambah status, dan sebagainya. Akses Pengembang memiliki akses penuh *API framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang

supaya kita dapat dengan mudah menggunakan kembali *reuse* (komponen yang sudah di gunakan).

Dapat disimpulkan bahwa *Application Framework* merupakan lapisan bagi pengembang aplikasi untuk mengembangkan aplikasi yang akan dijalankan pada sistem operasi *Android*, karena pada lapisan ini aplikasi dapat dirancang dan dibuat, seperti penyedia konten berupa SMS dan telepon. Komponen-komponen yang termasuk didalam *Application Frameworks* adalah sebagaiberikut:

1. Views
2. ResourceManager
3. NotificationManager
4. ContentProvider
5. ActivityManager

c. Libraries

Ini merupakan *layer* dimana fitur-fitur *libraries Android* berada, biasanya para pembuat aplikasi mengakses untuk menjalankan aplikasinya *libraries Layer* ini berjalan diatas Kernel, yang meliputi berbagai *library* inti C/C++, seperti:

1. *Libraries* untuk manajemen tampilan.
2. *Libraries Graphics* mencakup SGL dan *Open GL* untuk grafis 2D dan 3D.
3. *Libraries media* untuk pemutaran media audio dan *video*.
4. *Libraries SQLite* untuk dukungan *database*.
5. *Libraries Live Webcore* mencakup modern web browser dengan engine embedded webview.
6. *Libraries 3D* yang mencakup implementasi *Open GLES1.0API's*.
7. *Libraries SSL* dan *Web Kit* terintegrasi dengan *web browser* dan *security*.

d. *Android Run Time*

Layer inilah yang membuat aplikasi *Android* bisa dijalankan, di mana dalam prosesnya memakai implementasi Linux. DVM (Dalvik Virtual Machine) ialah mesin yang membentuk dasar kerangka aplikasi *Android*. Di dalam *Android Run Time* dipecah jadi 2 bagian, ialah:

1. *Core Libraries*: Aplikasi *Android* dibentuk dalam bahasa Java, sedangkan DVM selaku virtual mesin, sehingga dibutuhkan suatu *libraries* yang berfungsi untuk menterjemahkan bahasa Java/ C yang ditangani oleh *Core Libraries*.
2. *Dalvik Virtual Machine*: Virtual mesin berbasis register yang dimaksimalkan untuk melaksanakan fungsi- fungsi secara efisien, di mana ialah pengembangan yang sanggup membuat Linux Kernel buat melaksanakan *threading* serta manajemen tingkatan rendah.

e. *LinuxKernel*

Linux Kernel merupakan *layer* di mana inti dari operating sistem dari *Android* itu terletak. Berisi file- file sistem yang mengendalikan *processing*, *drivers*, *resource*, *memory*, serta sistem- sistem pembedahan *Android* yang lain. *Linux Kernel* yang digunakan *Android* merupakan *Linux Kernel release2.6*(Safaat 2012).

2.4.2 Fundamental Aplikasi

Aplikasi *Android* dengan bahasa pemrograman Java. Kode Java dikomplikasi bersama dengan data *fileresource* yang dibutuhkan oleh aplikasi, dimana proses nya di-*package* oleh *tools* yang dinamakan “*apttools*” kedalam paket *Android* sehingga menghasilkan *file* dengan ekstensi *apk*. Ada empat jenis komponen pada aplikasi *Android*,

1. *Activities*

Suatu *activity* menampilkan *userinterface* (UI) kepadapengguna, sehingga pengguna dapat melakukan interaksi. Aplikasi *Android*

dapat memiliki satu *activity* saja, tetapi pada umumnya memiliki banyak *activity* tergantung dengan tujuan dan desain dari aplikasi tersebut. Satu *activity* biasanya dapat digunakan untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi ditunjukkan kepada pengguna. Untuk berpindah dari satu *activity* ke *activity* lain dapat menggunakan satu *event*, contohnya *click* tombol, opsi atau memakai *triggers* tertentu. Secara hirarki sebuah *window* *sactivity* dinyatakan menggunakan *method* *Activity*. *Content View* adalah objek yang berada pada *root* hirarki.

2. *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan seperti *background*, contoh dalam memainkan musik, *service* memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam induknya. Misalnya, *media player* memutar lagu dari *list*, aplikasi akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu, atau menulis sms dan *player* sedang berjalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada *thread* utama dari proses aplikasi.

3. *BroadcastReceiver*

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *broadcast* seperti notifikasi zona waktu berubah baterai lemah, gambar telah selesai diambil oleh *camera*, atau perubahan referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi *broadcast* seperti memberikan informasi kepada aplikasi lain bahwa data yang telah diunduh ke perangkat serta siap untuk digunakan. *BroadcastReceiver* tidak mempunyai *userinterface* (UI), namun memiliki sebuah *activity* untuk merespon yang mereka terima, atau memakai *NotificationManager* untuk

memberitahu pengguna, seperti lampu latar atau *vibrating*(getaran), dan lain sebagainya.

4. *ContentProvider*

Contentprovider membentuk kumpulan aplikasi data secara khusus mampu digunakan oleh perangkat lunak lain. Data disimpan sistem seperti *Database SQLite*. *Contentprovider* menyediakan akses data yang dibutuhkan oleh suatu activity, contohnya saat kita menggunakan perangkat lunak yang membutuhkan peta (*Map*), atau aplikasi yang membutuhkan akses data kontak dan navigasi, maka disinilah fungsi *contentprovider*.

2.4.3 **Android SDK (Software development kit)**

Android SDK adalah *tool* Api adalah bahanyang di perlukan untuk membuat suatu aplikasi *platformAndroid* dengan bahasa pemrograman adalah java script sehingga menghasilkan perangkat lunak.

2.4.4 **ADT (AndroidDevelopment kit)**

Android Development Tools adalah *ADT* akronim dari *Android Development Tools* atau sering juga disebut dengan *PluginsEclipse*, yang merupakan penghubung antara IDE Eclipse dengan *Android SDK*. *ADT* memungkinkan Eclipse untuk digunakan dalam membuat aplikasi *Android* baru, membuat *UserInterface*, menambahkan komponen berdasarkan *framework API Android*, debug aplikasi, dan pemaketan aplikasi Aplikasi. Menurut Ir. Yuniar Supardi (2015)

2.5 **Metode Pengembangan Perangkat Lunak**

Metode penembangan perangkat lunak ialah metode yang digunakan dalam pembangunan sistem *Augmented reality*, penjelasan mengenai metode pengembangan pada penelitian ini adalah sebagai berikut:

2.6 Metode Pengembangan Multimedia

Penelitian ini bertujuan untuk membuat sebuah aplikasi *virtual reality* pengenalan taman purbakakala pugung raharjo Sebagai Media Edukasi dengan menggunakan *software* Unity 3D, dengan metode *Multimedia Development Life Cycle*. Pengembangan metode multimedia ini dilakukan berdasarkan enam tahap, yaitu *concept* (pengonsepan), *design* (perancangan), *material collecting* (pengumpulan bahan), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Keenam tahap ini tidak harus berurutan dalam praktiknya, tahap-tahap tersebut dapat saling bertukar posisi. Meskipun begitu, tahap *concept* memang harus menjadi hal yang pertama kali dikerjakan (Mustika dkk, 2018). Pada metode ini terdapat 6 tahapan yaitu:

1. Konsep

Tahapan ini digunakan untuk menentukan tujuan dan siapa yang akan menjadi pengguna program (identifikasi *Audience*). Selain itu pada tahapan ini juga menentukan macam aplikasi dan tujuan dari aplikasi.

2. Desain atau Perancangan

Tahapan ini dilakukan pembuatan spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan material atau bahan untuk program.

3. Pengumpulan Bahan

Tahapan ini dilakukan pengumpulan bahan yang sesuai dengan kebutuhan, pada tahapan ini dikerjakan secara paralel dengan tahapan pembuatan atau *Assembly*.

4. *Assembly* atau Pembuatan

Tahapan ini adalah tahapan dimana pembuatan semua objek dilakukan. Tahapan pembuatan aplikasi didasarkan pada tahapan desain.

5. Pengujian atau *Testing*

Setelah semua tahapan selesai kemudian akan dilakukan pengujian agar dapat meminimalisir kesalahan pada aplikasi atau program.

6. Pendistribusian

Tahapan dimana aplikasi disimpan dalam suatu media penyimpanan, pada tahapan ini jika media tidak cukup maka akan dilakukan kompresi terhadap aplikasi tersebut.

2.7 Pengertian Animasi

Animasi dari bahasa Latin yakni *anima* yang artinya jiwa, hidup, nyawa dan semangat. Dalam bahasa Inggris, animasi diambil dari kata *animate* yang memiliki arti menjiwai, menghidupkan serta *animation* yang memiliki arti semangat atau gelora (Aditya, 2009: 2). Animasi merupakan film yang dihasilkan dari gambar yang diolah hingga menjadi suatu gambar bergerak (Bambi Bambang Gunawan, 2013: 26). Kata animasi sering diartikan menghidupkan dari suatu benda mati atau benda statis yang kemudian dapat menciptakan gerak atau kesan bergerak (Gatot Prakosa, 2010: 23).

2.8 3D Animasi

Menurut Zaharuddin G. Djalle (2018), animasi 3D merupakan animasi yang menggunakan animasi dalam wujud yang memperhitungkan karakter sifat bahan animasi. Animasi 3D adalah animasi pada bidang yang memiliki 3 sumbu, yaitu sumbu X, Y, dan Z. Meskipun bukan dalam wujud 3D yang sebenarnya, yaitu bukan sebuah obyek 3D yang dapat disentuh dan dirasakan wujud fisiknya, namun dalam wujud 3D dalam layar kaca. Pengertian animasi 3D dapat disimpulkan sebagai teknik membuat kesan hidup obyek diam yang memiliki 3 sumbu, yaitu x, y, dan z.

2.9 Blender 3D

Blender merupakan sebuah *software* animasi 3D berbasis *open source*. 3D Blender adalah aplikasi lintas platform, berjalan di Linux, macOS, dan sistem MS-Windows. Blender juga merupakan aplikasi yang memungkinkan anda memproduksi animasi berkualitas tinggi dengan menggunakan geometri tiga dimensi. (Walhidayat, W., Yuhelmi, Y., & Devega, M.2019)

2.10 Metode Image tracking Vuforia

Image tracking Vuforia ialah penerapan gambar penanda menggunakan *tools* vuforia yang akan disinkronisasikan pada unity. *Tools* vuforia sendiri memiliki konsep penilaian kualitas pengenalan gambar penanda, ini dapat membantu penelitian untuk menentukan gambar penanda yang tepat. Deteksi setiap tepi line yang terdapat pada objek penanda yang merupakan penentu dari penilaian objek penanda itu sendiri, semakin rumit tepi line yang dibentuk akan semakin mendapatkan nilai yang baik, nantinya akan berpengaruh pada saat pengenalan dan sudut kamera dalam pengenalan objek 3D yang dihasilkan.

2.11 Unified modeling language (UML)

Unified Modelling Language (UML) Definisi UML menurut Rosa A.S, M. Shalahuddin (2015), “Teknik pemrograman yang berorientasi objek dan merupakan bahasa pemodelan untuk pembangunan perangkat lunak.” Ada empat jenis diagram dalam UML yakni:

1. *Use case diagram*
2. *Activity diagram*
3. *Class diagram*
4. *Sequence diagram*

2.11.1 Use case diagram

Merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada dua hal utama pada use case yaitu pendeskripsian apa itu actor dan use case. Berikut adalah simbol – simbol yang ada pada diagram use case. Penjelasan simbol pada tabel 2.1

Tabel 2.1 Tabel *Use case diagram*

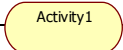
No.	Simbol	Keterangan
-----	--------	------------

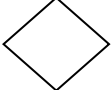
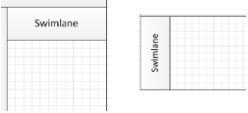
1		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor kerja di awal
2		Orang, proses, atau sistem lain yang saling berinteraksi. Walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
3		Komunikasi antar aktor dan <i>usecase</i> yang berpartisipasi pada <i>usecase</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		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.
5		Relasi <i>usecase</i> tambahan ke sebuah <i>usecase</i> di mana <i>usecase</i> yang ditambahkan memerlukan <i>usecase</i> ini untuk menjalankan fungsinya.
6		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

2.11.2 Activity diagram

Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2 Tabel *Activity diagram*

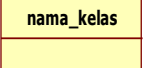
No	Simbol	Keterangan
1	 Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	 Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

3	Decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.11.3 Class diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2.3 Tabel *Class diagram*

No.	Simbol	Keterangan
1	Kelas 	Kelas pada struktur <i>system</i>
2	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3	Asosiasi 	Relasi antara kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity

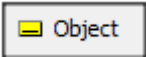
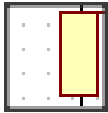
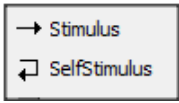
4	Asosiasi berarah 	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus)
6	Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas
7	Agregasi 	Relasi antar kelas dengan makna semua bagian

2.11.4 Sequence diagram

Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada suatu sistem dengan mendeskripsikan waktu hidup objek dan pesan yang dikirim dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada diagram sequence:

Tabel 2.4 Tabel *Sequence diagram*

No.	Symbol	Keterangan
1	Actor 	Menggambarkan orang yang berinteraksi dengan <i>system</i>
2	Entity class 	Menggambarkan hubungan kegiatan yang akan dilakukan
3	Boundary class 	Menggambarkan sebuah penggambaran dari form

4	Control class 	Menggambarkan penghubung antara boundary dengan table
5	Lifeline 	Menggambarkan tempat mulai dan berakhirnya sebuah pesan
6	Line message 	Menggambarkan pengiriman pesan
7	Object 	Menyatakan objek yang berinteraksi pesan
8	Garis hidup 	Menyatakan kehidupan suatu objek
9	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya actor tidak memiliki waktu aktif.
10	Stimulus 	Menyatakan suatu objek mengirimkan pesan untuk menjalankan operasi yang ada pada objek lain

2.12 Penelitian Terkait

Menurut Pressman, (2012) Black-Box Testing atau Pengujian Kotak Hitam atau juga disebut Behavioral Testing, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik Black-Box Testing memungkinkan untuk

mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program.

Black-Box Testing bukan merupakan alternatif dari pengujian White-Box Testing. Sebaliknya, Black-Box Testing adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode White-Box Testing. Black-Box Testing mencoba untuk menemukan kesalahan dalam kategori berikut:

3. Fungsi tidak benar atau hilang.
4. Kesalahan *interface* atau antarmuka.
5. Kesalahan dalam struktur data atau akses *database* eksternal.
6. Kesalahan kinerja atau perilaku.
7. Kesalahan inisialisasi dan terminasi.

Berikut ini merupakan beberapa Penelitian yang berhubungan dengan laporan skripsi ini yaitu:

Tabel 2.5 Tabel Referensi Penelitian

No	Nama	Judul	Keterangan	Sumber
1	Rosandy, T., & Zaini, T. M	<i>Augmented reality</i> Wisata Monumen Bersejarah Lampung Berbasis <i>Mobile</i>	Implementasi penggunaan Teknologi AR (<i>Augmented reality</i>) ini dapat di terapkan sebagai salah satu sarana promosi wisata ‘monumen bersejarah’ di provinsi lampung dan	Rosandy, T., & Zaini, T. M. (2019). <i>AUGMENTED REALITY WISATA MONUMEN BERSEJARAH LAMPUNG BERBASIS MOBILE. Jurnal Informatika</i> , 19(1), 31-35.

			menjadikan lebih menarik dan interaktif. Dimana objek tiga dimensi (3d) dapat di tampilkan dengan diwakilkan oleh objek 'image' sehingga user yang memiliki Handphone dapat berinteraksi langsung dan dapat melihat tampilan yang lebih interaktif yang mewakili tentang sarana dan aktifitas yang ada pada objek bersejarah.	
2	Kusumawati, D. A. (2017)	Pengembangan Virtual Reality Pada Lawang Sewu Sebagai Pengenalan Objek Wisata Jawa Tengah	Penelitian ini mengimplementasikan virtual reality sebagai Membangun aplikasi	Kusumawati, D. A. (2017). <i>Pengembangan Virtual Reality pada Lawang Sewu sebagai Pengenalan Objek</i>

			virtual reality pada Lawang Sewu sebagai pengenalan objek wisata Jawa Tengah	<i>Wisata Jawa Tengah</i> (Doctoral dissertation, Universitas Negeri Semarang).
3	Wahyudi, A., Adam, P., & Jesyrviano, G. (2017).	Virtual Reality of Historical Places in North Sulawesi	Pada penelitian ini dibahas bagaimana pengimplementasian virtual reality sebagai pengenalan tempat wisata bersejarah di Sulawesi utara	Wahyudi, A., Adam, P., & Jesyrviano, G. (2017). Virtual Reality of Historical Places in North Sulawesi. <i>CogITO Smart Journal</i> , 3(1), 32-41.