

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Penelitian Terdahulu**

Studi literatur dilakukan oleh penulis dengan memahami penelitian yang pernah dibuat lebih dahulu dan penelitian penulis yang bersangkutan sehingga bisa dipakai sebagai referensi dan perimbangan pada metode penelitian dan perolehan yang diraih dari penelitian ini.

Penelitian mengenai aplikasi pembelajaran tentang tokoh pahlawan berbasis *augmented reality* pernah dilakukan oleh beberapa peneliti. Pertama, Risman et al. (2018) yang membahas tentang media pembelajaran yang menjadikan patung di pantai Losari Makassar sebagai *marker* menggunakan teknologi *augmented reality*. Kedua, Putra et al. (2021) membahas media pembelajaran yang menjadikan uang kertas sebagai marker yang kemudian muncul video uang yang berbicara. Ketiga, Rochmah et al. (2019) membahas media pembelajaran yang menjadikan gambar-gambar yang ada pada buku LKS sebagai marker yang kemudian muncul video deskripsi singkat tentang pahlawan.

Kesimpulan didapat dari bahasan diatas, bahwa aplikasi yang akan dibangun dengan yang ada dalam penelitian terkait terdapat perbedaan dan juga ada kesamaan. Persamaannya adalah peneliti tersebut sama-sama membangun aplikasi *Augmented Reality* berbasis android sebagai media informasi yang dikemas dengan menarik dan unik.

Rangkuman penelitian terkait aplikasi pembelajaran tentang tokoh pahlawan berbasis *augmented reality* dapat dilihat pada tabel 2.1

Tabel 2.1 Rangkuman Hasil Penelitian Terdahulu

No	Penelitian	Fungsi Aplikasi	Marker Objek	Kesimpulan penelitian	Keterangan	
					Animatif	Quiz
1	(Risman et al., 2018)	Perancangan Aplikasi Pengenalan Tokoh Pahlawan dan Kebudayaan Melalui Patung di Pantai Losari Makassar Berbasis Teknologi <i>Augmented Reality</i> (RARHA PANLOS)	Patung di Pantai Losari Makassar	Memunculkan objek 3D berupa pahlawan, informasi singkat tentang pahlawan, kebudayaan yang ada di pantai Losari Makassar	-	-
2	(Putra et al., 2021)	Aplikasi <i>Augmented Reality</i> Media Pembelajaran Pengenalan Gambar Tokoh Pahlawan Nasional Pada Uang Kertas Berbasis Android	Uang kertas	Memunculkan objek 3D yang berbicara namun animasi tidak bergerak dan tidak ada teks pada tampilan uang berbicara	-	-
3	(Rochmah et al., 2019)	Pengembangan Media Berbasis Teknologi <i>Augmented Reality</i> Bermuatan Wawasan Kebangsaan pada Tokoh Kepahlawanan	Buku LKS pelajaran IPS kelas 5 SD	Menampilkan objek 3D pada wajah pengguna dan video deskripsi pahlawan. Media ini berbasis desktop	-	-

Tabel 2.1 (Lanjutan)

No	Penelitian	Fungsi Aplikasi	Marker Objek	Kesimpulan penelitian	Keterangan	
					Animatif	Quiz
1	Penelitian yang diusulkan	Aplikasi Pembelajaran Bertema Wawasan Kebangsaan Berbasis <i>Augmented Reality</i> pada <i>Platform Android</i>	Buku tematik kurikulum 2013 tentang pahlawanku pada kelas 4 SD	Menampilkan objek 2D yang berbicara dengan menceritakan deskripsi singkat tentang pahlawan	√	√

## 2.2 Wawasan Kebangsaan

Sikap kecintaan terhadap tanah air merupakan salah satu makna dari wawasan kebangsaan. Menurut Rochmah et al.(2019) wawasan kebangsaan merupakan pemahaman berwarganegaraan, yaitu perasaan yang muncul secara alamiah lantaran memiliki solidaritas kemasyarakatan yang hadir menurut sejarah, adat, dan tekad kerja keras zaman dahulu, dan solidaritas pada menemui tantangan sejarah masa kini. Wawasan kebangsaan juga identik menggunakan nilai patriotisme. Jiwa patriotisme anak didik yang memperlihatkan perilaku cinta tanah air dan bangsa dapat dicermati menggunakan wawasan kebangsaan (Rahayuningsih, 2009).

Sikap radikalisme yang adalah lawan dari perilaku nasionalis. Mencegah keluarnya perilaku radikalisme merupakan menggunakan menambah wawasan kebangsaan buat mengurangi konduite radikalisme. Persepsi yang positif terhadap negara merupakan hal krusial yang wajib dimiliki sang pemuda pemudi waktu ini. Hal ini dikarenakan persepsi pula bisa berpengaruh dalam terbentuknya perilaku

fundamentalis. Fundamentalis yang positif seharusnya mulai pada bangun dalam anak buat mencegah keluarnya perilaku radikalisme (Ariyanto et al., 2019).

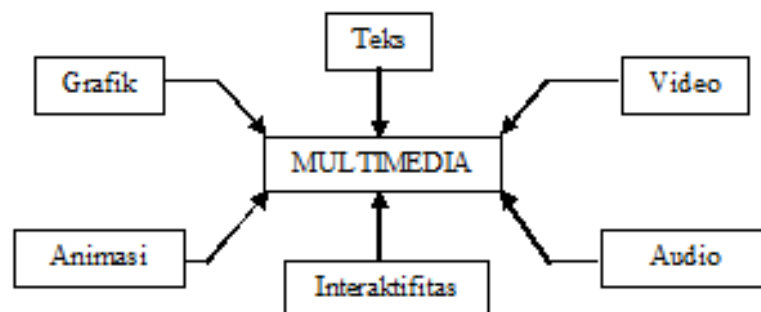
### **2.3 Media Pembelajaran**

Menurut Binanto (2010), secara luas pembelajaran merupakan sarana dukung pada mekanisme belajar mengajar. Suatu hal yang bisa digunakan buat mendorong gagasan, ketertarikan, pandangan, dan kemahiran atau kemampuan peran pelajar tersebut sehingga akibatnya bisa mendorong terjadinya proses belajar atau aktivitas pembelajaran. Media pembelajaran ini menurut batasan relatif umum dan mendetail yang meliputi arti sumber, manusia dan lingkungan serta tujuan pembelajaran atau pelatihan menurut metode yang dimanfaatkan (Saputro & Saputra, 2015).

### **2.4 Multimedia**

Multimedia merupakan menggunakan komputer dalam menciptakan serta mengumpulkan teks, grafik, audio dan gambar bergerak (video dan animasi) yang memungkinkan pemakai berhubungan, berkarya, berkoneksi serta menimbulkan teks yang melibatkan bunyi, gambar, musik, animasi, dan video. Sistem penunjang kepastian yang berbasis komputer ditafsir berupa interaktif. Interaktif menyimpan persepsi apa yang bisa dibuat murid terhadap media. Pada multimedia interaktif terjadi hubungan antara murid menggunakan fasilitas multimedia yang tersedia (Rahmawati, n.d.).

Menurut Vaughan dalam Binanto (2010), gabungan dari teks, seni, suara gambar, video serta animasi yang dibuat menggunakan komputer atau diberi trik secara digital dan bisa diberikan atau interaktif secara dikontrol merupakan arti dari multimedia (Gambar 2.1). Multimedia memiliki 3 jeni, yaitu: (1) Multimedia interaktif: Pemakai bisa mengamati apa dan kapan unsur-unsur multimedia akan diantarkan atau ditampakkan. (2) Multimedia hiperaktif: Suatu struktur dari unsur-unsur terkait menggunakan pemakai yang bisa mengarahkannya, ini dimiliki multimedia jenis ini. Dapat dijelaskan bahwa multimedia jenis ini memiliki banyak tautan (*link*) yang menyambung unsur-unsur multimedia yang ada. (3) Multimedia linear: Pemakai hanya sebagai pengamat dan menikmati hasil multimedia yang diberikan dari awal sampai akhir (Saputro & Saputra, 2015).

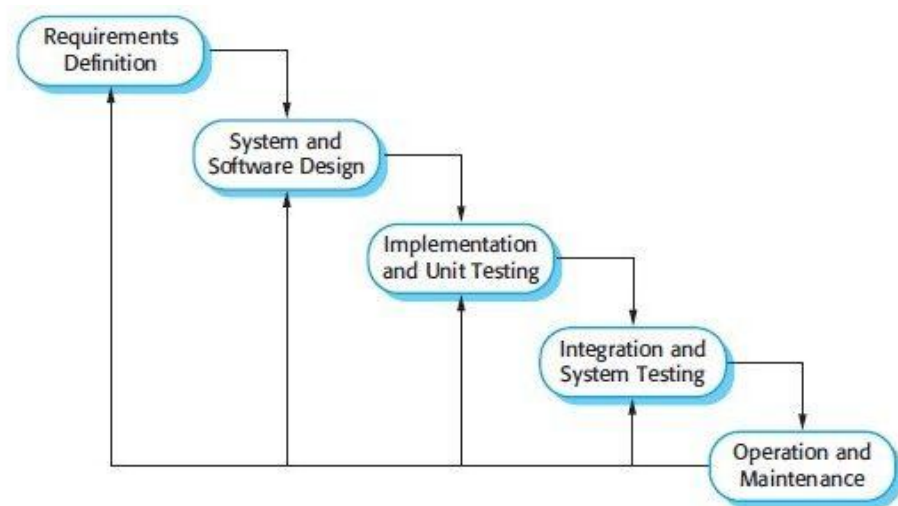


Gambar 2.1 Gambaran Definisi Multimedia  
Sumber : (Saputro & Saputra, 2015)

## 2.5 Metode Waterfall

Penelitian ini menggunakan metode pengembangan perangkat lunak yaitu model air terjun (*Waterfall*). Menurut Pressman dalam Moenir & Yuliyanto (2017) bahwa model *waterfall* yaitu paradigma unggul yang berupa berurut dalam menciptakan aplikasi. Menurut Sukamto dan Shalahuddin dalam Firmansyah &

Udi (2017) menjelaskan bahwa model sekuensi linear atau alur hidup klasik disebut juga model air terjun. Sistem dikembangkan secara berurutan sedari analisis, desain, pengkodean, pengujian dan tahap pendukung.



Gambar 2.2 Model *Waterfall*  
Sumber : (Jayanti & Hendini, 2021)

### 1. Analisis Kebutuhan Perangkat Lunak Proses

Pengumpulan kebutuhan dikerjakan secara mendalam buat lebih detail keperluan *software* agar muda diasumsi seperti apa yang diperlukan oleh pengguna. Pada tahap ini, spesifikasi kebutuhan *software* perlu di perbahankan. Menerima penjelasan tentang *software* yang selaras dengan keperluan pengguna dalam pengupulan data. Wawancara dan observasi adalah teknik pengumpulan data yang kerap kali dilakukan.

### 2. Desain

Prosedur banyak tahap yang berpusat pada desain penciptaan program *software* termuat struktur data, arsitektur perangkat lunak, representasi antar muka dan mekanisme pengkodean merupakan arti dari desain perangkat lunak. Langkah ini memindahkan keperluan perlengkapan dari tahap analisis keperluan ke

perwakilan desain untuk bisa di terapkan program di langkah berikutnya. Pada tahap ini juga perlu di bahankan dari hasil desain perangkat lunak. Pada tahap ini, desain sistem model *UML (Unified Modeling Language)* menggunakan desain aritektur lunak dan desain antar muka media dengan papan cerita.



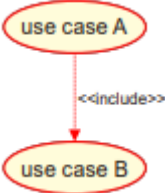
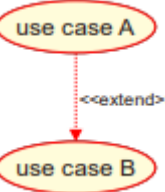





a. *UML (Unified Modeling Language)*

Menurut Martin Fowler (2005: 1), *Unified Modeling Language (UML)* adalah kelompok notasi grafis yang dibantu oleh meta-model tunggal, yang membantu penggambaran dan desain sistem perangkat lunak, pemrograman berorientasi objek yang dibangun menggunakan sistem. Menurut Rosa (2014: 133) tolak ukur bahasa yang banyak dipakai di dunia industri buat mengartikan *requirement*, membangun analisis dan desain, dan pemrograman berorientasi objek yang di deskripsikan dalam aritektur adalah *UML* (Burhanudin, 2017).

b. *Use Case Diagram*

Menurut Martin Fowler (2005: 141), Sebuah metode untuk mendokumentasikan persyaratan fungsional dari sebuah sistem yaitu arti dari *use case*. *Use case* mengabadikan hubungan tipikal antara para pemakai sistem dan sistem itu sendiri, dengan bagaimana sistem digunakan dengan memberikan narasi. Simbol diagram *use case* ditampilkan pada Tabel 2.2 Notasi *Use Case Diagram*. Secara sederhana *use case diagram* merupakan gambaran fungsionalitas dari sistem yang dapat digunakan oleh pemakai.

Tabel 2.2 Notasi *Use Case Diagram*

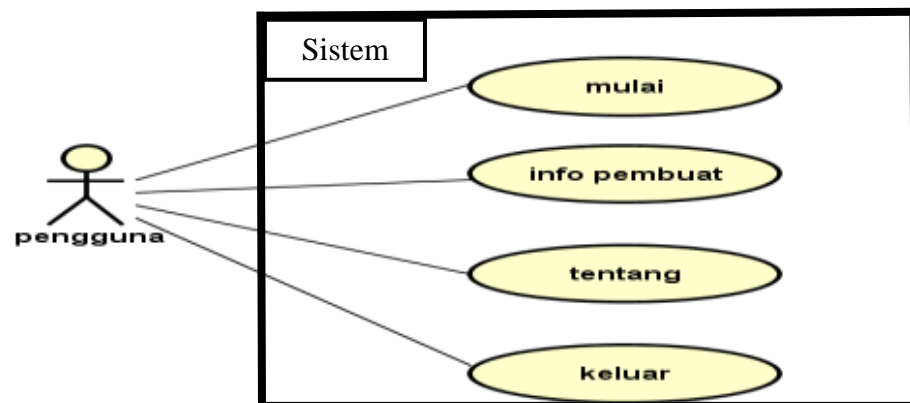
Simbol	Nama	Kegunaan
	<i>Actor</i>	Sesuatu diluar sistem (manusia, sistem, atau perangkat) yang akan menggunakan sistem untuk mengerjakan sesuatu
	<i>Dependency</i>	Perhubungan menampilkan tentang perubahan dalam satu komponen mempengaruhi komponen lain
	<i>Include</i>	Menampilkan bahwa bagian dari komponen (yang digaris bawahhi tanpa panah) membuat eksekusi bagian dari komponen lain (yang ada digaris dengan panah). Jika <i>Use case B</i> telah dieksekusi setidaknya sekali, maka <i>use case A</i> dapat dieksekusi
	<i>Extend</i>	Menampilkan bahwa bagian dari komponen pada garis tanpa panah bisa dimasukkan komponen yang berada dalam garis dengan panah. <i>Use case A</i> mendatangkan <i>use case B</i> dalam kondisi tertentu
	<i>Generalization</i>	Menampilkan koneksi antara komponen yang lebih umum ke komponen yang lebih spesifik
	<i>Association</i>	Pengenalan hubungan antara pengguna tertentu dan setiap <i>use case</i> tertentu. Sebagai garis antara pengguna dan <i>use case</i> yang bersangkutan yang digambarkan
	<i>System</i>	Menentukan batasan sistem dalam interaksi dengan pengguna- pengguna yang menggunakannya (di luar sistem) dan yang harus disediakan dalam fitur (dalam sistem)
	<i>Use case</i>	Menunjukkan tujuan yang ingin dicapai oleh sistem dan dinamai sesuai tujuannya
	<i>Interaction</i>	Menampilkan aliran intruksi atau informasi antar objek dan hubungan antar objek

Contoh *use case diagram* pada penelitian (Putra et al., 2021) tentang

Aplikasi Augmented Reality Media Pembelajaran Pengenalan Gambar

Tokoh Pahlawan Nasional Pada Uang Kertas Berbasis Android



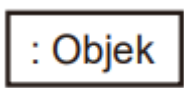




Gambar 2.3 Contoh *Use Case Diagram*  
 Sumber : (Putra et al., 2021)



c. *Sequence Diagram*

Menurut John Satzinger (2010) diagram yang difungsikan untuk mendefinisikan *input* dan *output* dari *use case* dan urutan interaksi antara *user* dan sistem merupakan arti dari *sequence diagram*. Menurut Martin Fowler (2005: 81), *sequence diagram* merupakan bentuk diagram hubungan yang menggambarkan sekelompok objek saling berkerjasama dalam beberapa perlakuan.

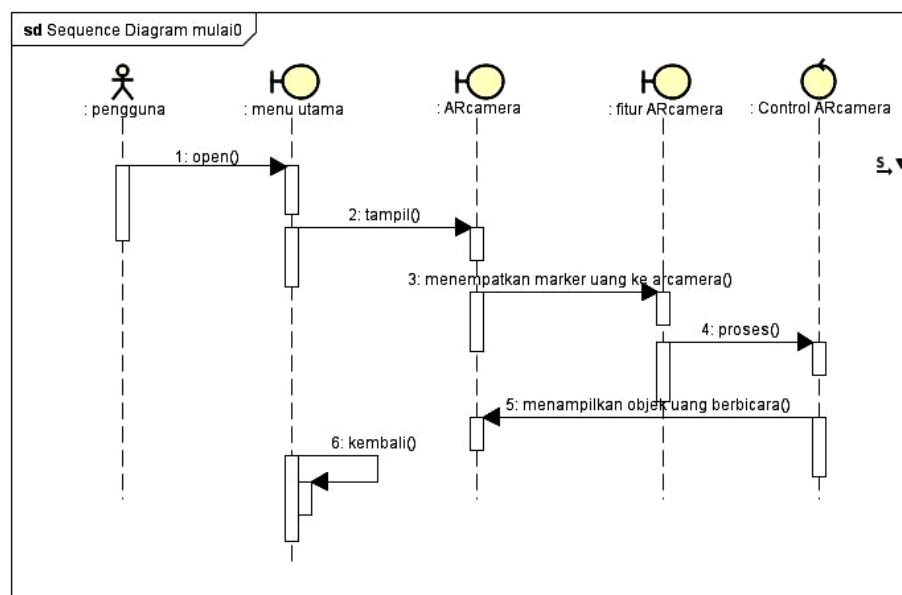
Tabel 2.3 Notasi *Sequence Diagram*

Simbol	Nama	Kegunaan
	<i>Object</i>	Sebuah contoh dari kelas, ditulis secara mendatar. Dipresentasikan sebagai kelas (kotak) yang berisi nama objek yang dimulai dengan titik koma
	<i>Actor</i>	Aktor adalah hal-hal diluar sistem (orang, sistem, atau perangkat) yang akan melakukan sesuatu dengan sistem
	<i>Lifeline</i>	<i>Lifeline</i> menunjukkan keberadaan suatu objek berdasarkan waktu. Notasi <i>lifeline</i> adalah garis putus-putus lurus tegak yang ditarik dari suatu objek

Tabel 2.3 (Lanjutan)

Simbol	Nama	Kegunaan
	<i>Activation</i>	<i>Activation</i> dicatatkan sebagai kotak persegi panjang pada <i>lifeline</i> . <i>Activation</i> menentukan objek untuk melakukan sebuah aksi
	<i>Message</i>	<i>Message</i> didokumentasikan dengan anak panah mendarat antara <i>Activation</i> . Pesan ini menunjukkan koneksi antara objek

Contoh *sequence diagram* pada penelitian (Putra et al., 2021) tentang Aplikasi Augmented Reality Media Pembelajaran Pengenalan Gambar Tokoh Pahlawan Nasional Pada Uang Kertas Berbasis Android.

Gambar 2.4 Contoh *Sequence Diagram*







Sumber : (Putra et al., 2021)

#### d. *Activity Diagram*

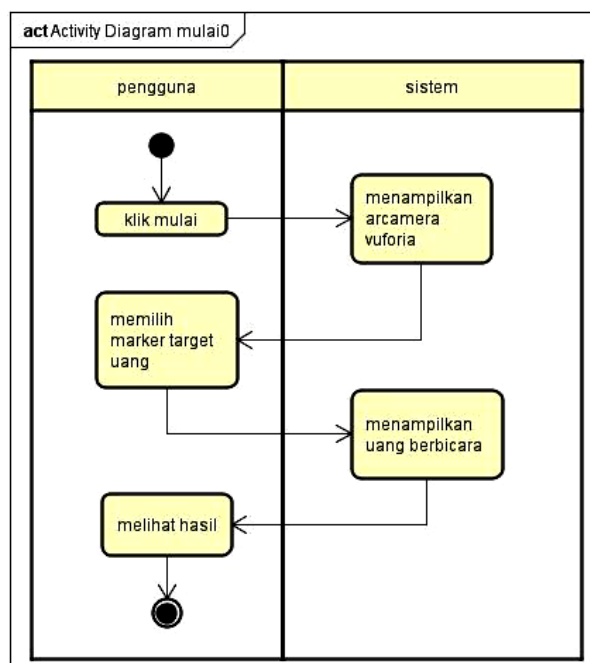
Menurut Martin Fowler (2005: 163), metode dalam mendeskripsikan logika prosedural, proses bisnis dan jalur kerja merupakan arti dari *activity diagram*. Diagram ini bertindak seperti diagram alir yang membawa

perilaku sejalan. Sedangkan menurut Henderi (2008), Gambaran urutan aktifitas *use case* secara garis besar menggunakan *activity diagram*.

Tabel 2.4 Notasi *Activity Diagram*

Simbol	Nama	Kegunaan
	<i>Start Point</i>	Memberitahukan posisi awal alur aktivitas
	<i>End Point</i>	Memberitahukan di mana alur aktivitas berhenti
	<i>Activities</i>	Memberitahukan aktivitas dalam alur kerja
	<i>Dicision</i>	Memberitahukan di mana dalam aliran aktivitas keputusan harus dibuat
	<i>Join</i>	Memberitahukan persimpangan pada aliran aktivitas
	<i>Fork</i>	Memberitahukan pencampuran dalam aliran aktivitas

Contoh *activity diagram* pada penelitian (Putra et al., 2021) tentang Aplikasi Augmented Reality Media Pembelajaran Pengenalan Gambar Tokoh Pahlawan Nasional Pada Uang Kertas Berbasis Android.



Gambar 2.5 Contoh *Activity Diagram*

Sumber : (Putra et al., 2021)

e. *Storyboard*

Penggambaran dari ide aplikasi yang akan dibangun untuk memberikan gambaran tentang aplikasi yang akan dibangun ialah arti dari *storyboard*. *Storyboard* juga dikenal sebagai *visualisasi script* yang digunakan sebagai *outline* dari sebuah proyek, ditampilkan *shot by shot* yang biasa disebut sebagai *scene*.

3. Pembuatan Kode Program

Program perangkat lunak harus menerjemahkan ke dalam desain. Program komputer sesuai dengan desain yang dibuat pada tahap desain merupakan hasil dari langkah ini.

4. Pengujian

Pengujian fokus pada *software* dari segi logik dan fungsional serta menetapkan bahwa seluruh komponen sudah diuji. Hal ini dibuat untuk meminimalkan kesalahan dan untuk memastikan bahwa *output* yang diperoleh sesuai dengan yang di mau. Teknik pengujian *black box* untuk menguji fungsionalitas perangkat lunak digunakan penulis untuk mengembangkan media pembelajaran *Augmented Reality*.

Menurut Rosa (2014: 275), pengujian perangkat lunak terhadap spesifikasi fungsional tanpa memeriksa desain dan kode program merupakan arti dari pengujian *black box*. Tujuan pengujian adalah untuk menentukan apakah fungsionalitas, *input* dan *output* perangkat lunak sesuai dengan spesifikasi yang diperlukan. Kasus uji yang menggunakan perangkat lunak untuk menguji semua

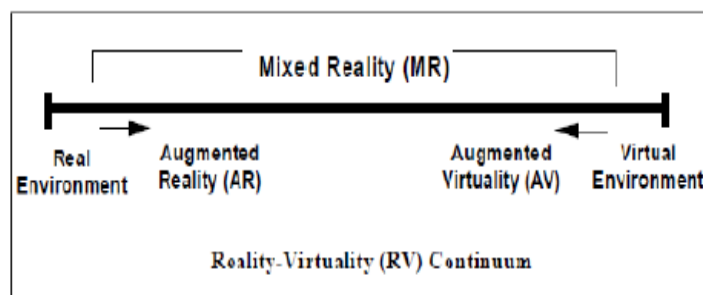
fungsi dan memastikan bahwa memenuhi spesifikasi yang diperlukan, dibuat dengan melakukan pengujian *black box*.

*Black box testing* mencoba menemukan kesalahan dalam beberapa kategori, termasuk:

1. fungsi yang rusak atau hilang,
2. kekeliruan antarmuka,
3. kekeliruan dalam struktur data atau akses *database external*,
4. kekeliruan kinerja,
5. kekeliruan inisialisasi dan terminasi.

## 2.6 Augmented Reality

*Augmented Reality* adalah upaya komputer untuk memadukan dunia nyata dan dunia maya, meninggalkan garis di antara keduanya sangat tipis. *Augmented Reality* (AR) adalah jenis dari *Virtual Environment* (VE) yang lebih dikenal dengan *Virtual Reality* (VR).



Gambar 2.6 Diagram Ilustrasi *Augmented Reality*  
Sumber : (Saputro & Saputra, 2015)

*Augmented Reality* menggunakan dua metode yaitu: *marker based tracking* dan *markerless*. *Marker based tracking* adalah AR, yang menggunakan penanda atau penanda pada objek dua dimensi, yang polanya dibaca oleh komputer melalui

*webcam* atau kamera yang terhubung ke komputer, biasanya ilustrasi kotak hitam putih dengan batas hitam. Metode *Markerless Augmented Reality*, di mana pengguna tidak memerlukan penanda untuk melihat komponen digital. Teknologi *Markerless AR* berbasis perangkat *android* menjanjikan implementasi AR yang lebih efisien, bermanfaat, menarik, dan dapat diakses oleh siapa saja, dimana saja, kapan saja, tanpa harus mencetak *marker* (Saputro & Saputra, 2015).

## **2.7 Android**

Sistem operasi untuk perangkat mobile berbasis Linux, termasuk sistem operasi, *middleware* dan aplikasi, arti dari *android*. *Android* menyajikan *platform* terbuka bagi *developer* untuk membuat aplikasi mereka sendiri. Awalnya, Google Inc. beli Android Inc. Ini adalah hal baru dalam menciptakan perangkat lunak untuk ponsel atau *smartphone*. Kemudian datanglah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia*, untuk mengembangkan *Android*.

## **2.8 Easy AR**

*EasyAR* adalah perpustakaan perangkat lunak untuk menciptakan *Augmented Reality (AR)*. Program ini merupakan aplikasi yang menampilkan gambar virtual di dunia nyata. Untuk melakukan ini, *EasyAR* menggunakan penelusuran video untuk mengukur letak sesungguhnya dari kamera dan menyelaraskan pola pada penanda secara *realtime*. Setelah posisi tepat kamera

yang benar sudah diketahui, letakkan kamera virtual di posisi yang sama dan gambar objek 3D pada penanda. Jadi *EasyAR* menyelesaikan persoalan *Augmented Reality* dalam hal sudut pandang penelusuran objek dan hubungan objek virtual.

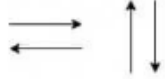




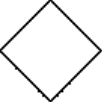
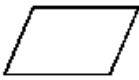
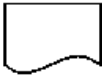
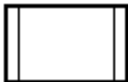

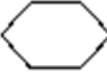
## 2.9 Unity 3D

*Unity* adalah salah satu *game engine* atau mesin pencipta game dan *software* lainnya. *Unity 3D* adalah *software* tidak berbayar, namun ada beberapa fitur di dalam *Unity 3D* tidak tersedia tanpa membeli lisensi berbayar. Perangkat lunak ini membangun *game* sendiri lebih cepat dan sederhana. Hebatnya, *Unity 3D* membantu penciptaan permainan dan *software* lain di beragam *platform*, seperti *Unity Web*, *Windows*, *Mac*, *Android*, *iOS*, *XBox*, *Playstation 3* dan *Wii* (Putra et al., 2021).

## 2.10 Flowchart

*Flowchart* merupakan sketsa yang menjabarkan alur pemrosesan rancangan. Saat membangun sebuah program, *flowchart* memainkan tugas penting dalam menerjemahkan proses eksekusi program dengan cara yang lebih mudah untuk dipahami.

Tabel 2.5 Simbol-simbol *flowchart*

No	Simbol	Nama	Fungsi
1		<i>Flow</i>	Simbol digunakan untuk menghubungkan simbol dengan simbol lainnya
2		<i>Connector Symbol</i>	Lambang untuk mengakhiri atau bergabung dengan proses pada lembar atau halaman yang sama
3		<i>Connector Symbol</i>	Lambang untuk input dan output atau koneksi proses pada lembar/ halaman yang berbeda
4		<i>Terminator Symbol</i>	Lambang untuk memulai atau mengakhiri aktivitas
5		<i>Processing Symbol</i>	Lambang yang menampilkan pemrosesan oleh komputer
6		Simbol <i>Decision</i>	Lambang pemilihan proses berdasarkan situasi yang ada
7		Simbol <i>input output</i>	Lambang yang menampakkan proses masukan dan keluaran tanpa tergantung dengan jenis perlengkapannya
8		Simbol <i>Manual Operation</i>	Lambang yang menampilkan pemrosesan non-komputer
9		Simbol <i>Dokumen</i>	Lambang yang menampakkan masukan berasal dari dokumen kertas atau keluaran dicetak di atas kertas
10		Simbol <i>Predefine Proses</i>	Lambang untuk pengoperasian suatu bagian (sub-program)/ prosedur
11		Simbol <i>Display</i>	Lambang yang menunjukkan perangkat keluaran yang dipakai, seperti layar, plotter, atau printer
12		<i>Preparation</i>	Lambang yang menunjukkan penyiapan nilai awal dan alokasi area memori untuk pemrosesan inisialisasi