# BAB II TINJAUAN PUSTAKA

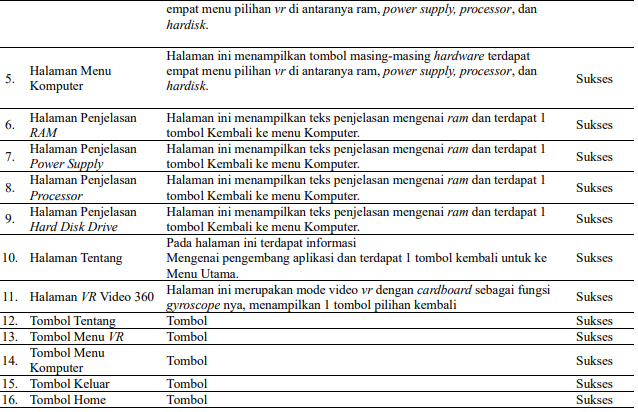
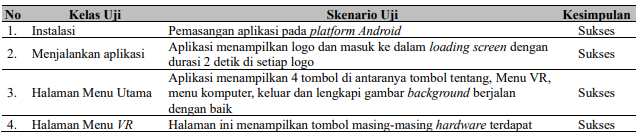
## 2.1 Penelitian Terdahulu

Berikut merupakan penelitian terdahulu yang mengimplementasikan media pembelajaran dalam bentuk aplikasi 3D ke beberapa studi kasus :

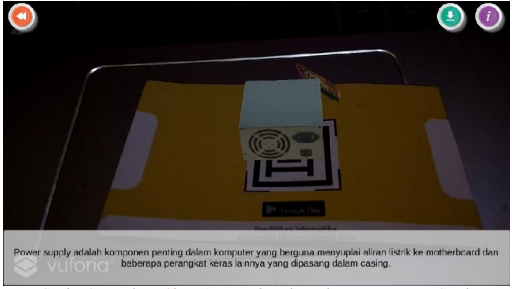
 **Aplikasi Perakitan Komputer untuk Media Ajar Menggunakan Teknologi *Virtual Reality* Berbasis Android.** Penelitian ini ditulis oleh Ayu Latifah, Eri Satria, dan Fajar Shidiq Gimnastiar dari Institut Teknologi Garut. Fokus penelitian ini adalah membangun aplikasi perakitan komputer sebagai media pembelajaran untuk SMK jurusan TKJ dengan teknologi *Virtual Reality* berbasis android. Penelitian ini menggunakan pengujian Blackbox dengan hasil yang didapat dari penelitian ini adalah aplikasi yang dibangun dapat merubah cara belajar yang berawal menggunakan media buku, artikel dan lainnya, menjadi aplikasi multimedia.

##### Gambar 2.1 Preview Aplikasi Penelitian Pertama

##### Gambar 2.2 Tabel Hasil Penelitian Pertama

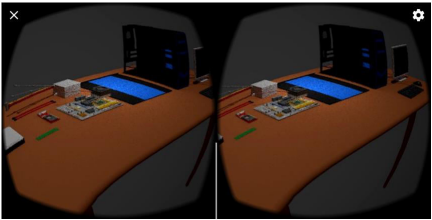


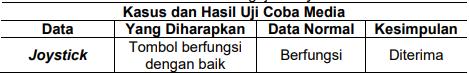
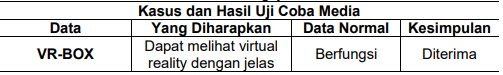
**Simulasi Pengenalan dan Instalasi PC berbasis *Augmented reality* dengan Metode Single marker.** Penelitian ini ditulis oleh Pratama Benny Herlandy, Agus Satria, Edi Ismanto dari FKIP Universitas Muhammadiyah Riau. Fokus dari penelitian ini adalah untuk menghasilkan sebuah media simulasi yang dapat membantu peserta didik SMK untuk memahami materi tentang pengenalan komponen dan teknik perakitan komputer dengan menggunakan teknologi *Augemented Reality*. Penelitian ini menggunakan uji coba *alpa testing* dan *beta testing* untuk aspek fungsionalitas*,* dengan hasil peserta didik tertarik dengan media simulasi yang telah dikembangkang.



##### Gambar 2.3 Preview Aplikasi Penelitian Kedua

##### Gambar 2.4 Tabel Hasil Penelitan Kedua

**Implementasi Teknologi *Virtual Reality* Pada Media Pembelajaran Perakitan Komputer.** Penelitian ini ditulis oleh Hari Antoni Musril, Jasmienti, dan Mifta Hurrahman dari FTIK IAIN Bukittinggi. Fokus dari penelitian ini adalah perancangan media pembelajaran yang bisa menggantikan peran dari alat peraga asli. Penelitian ini menggunakan uji coba *blackbox* dengan hasil berfungsi semua, dan disimpulkan bahwa siswa merespon aplikasi ini dengan sangat baik dan antusias, meskipun membutuhkan perangkat keras berupa VR Box.



##### Gambar 2.5 Preview Aplikasi Penelitan Ketiga

##### Gambar 2.6 Tabel Hasil Penelitian Ketiga

**Media Pembelajaran Simulasi Perakitan Komputer Menggunakan Unity 3D.** Penelitian ini ditulis oleh Bayu Ari Pranata, Ulfah MediatyArief, dan Agus Suryanto dari Universitas Negeri Semarang. Fokus penelitan ini adalah membangung aplikasi simulasi perakitan komputer sebagai media belajar pada mata pelajaran “Perakitan Komputer” di SMK. Penelitian ini menggunakan pengujian *blackbox* dan uji kelayakan yang dilakukan oleh ahli media dan ahli materi, dengan hasil kelayakan 93% dari ahli media dan 95% ahli materi, dengan kategori layak sebagai media pembelajaran.



##### Gambar 2.7 Preview Aplikasi Penelitian Keempat

##### Gambar 2.8 Tabel Hasil Penelitian Keempat

**Perancangan Game Simulasi Rakit Komputer Menggunakan Unity.** Penelitian ini ditulis oleh Muh.Yusuf Zahir, Muhammad Sabiq Al Mubaraq, Muh.said Misbahuddin, dan Rismayani dari Universitas Dipa Makassar. Fokus penelitian ini adalah merancang sebuah *game* edukasisimulasi perakitan komputer. Penelitian ini menggunakan pengujian *blackbox* dengan penilaian berdasarkan kuesioner, dan didapat hasil pengujian *blackbox* adalah semua berfungsi dengan normal dan game ini layak untuk digunakan.

##### Gambar 2.9 Preview Aplikasi Penelitian Kelima

##### Gambar 2.10 Tabel Hasil Penelitian Kelima

**Kesimpulan dari penelitian terdahulu :**

Berdasarkan kelima jurnal penelitian di atas yang telah diulas oleh peneliti sebagai referensi penelitian, kesimpulan yang dapat diambil adalah penelitian ini akan menggunakan teknologi sensor gyroscope yang terdapat pada *smartphone*, karena *smartphone* merupakan benda yang cukup umum dan mayoritas dimiliki oleh siswa SMK, dan sensor gyroscope mudah didapatkan pada *smartphone* harga menengah kebawah hingga yang mahal sehingga sensor ini cukup umum dimiliki pada mayoritas *smartphone*. Dengan sensor gyroscope pengguna tidak perlu memiliki VR box beserta *controller-*nya, dan tentunya tidak perlu memiliki perangkat komputer untuk menjalankan aplikasi ini. Dengan objek 3D *low-poly* maka akan mengurangi beban *hardware* dalam memproses aplikasi pada *smartphone* sehingga ramah spesifikasi menengah ke bawah.

## 2.2 Teori Terkait

Berikut merupakan teori yang berkaitan dengan penelitan yang berguna untuk menunjang perancangan aplikasi.

### 2.2.1 *Hardware*

*Hardware* atau dalam bahasa Indonesia disebut dengan perangkat keras adalah komponen dari sebuah komputer yang bersifat dapat dilihat dan diraba secara langsung atau yang berwujud nyata, yang berfungsi untuk mendukung proses komputerisasi. *Hardware* adalah perangkat komputer yang terdiri atas susunan komponen-komponen elektronik berbentuk fisik (berupa benda). *Hardware* atau perangkat keras adalah sebuah alat atau benda yang bisa dilihat, sentuh, pegang dan memiliki fungsi tertentu. Peralatan yang secara fisik terlihat dan bisa diraba atau dipegang (Rizky Dhanta, 2009:58).

*Hardware* terbagi berdasarkan fungsi, menurut Dudi Misky, (2005:12), Berikut adalah *hardware* yang dibagi berdasarkan fungsi :

1. *Input-Output Device*

*Input Device* adalah perangkat yang berfungsi untuk memasukan data ke dalam CPU. Contoh : *mouse*, *keyboard*, *joystick*, dan lain-lain.*Ouput device* adalah perangkat yang menjadi sarana untuk menampilkan hasil proses sebuah *processor*. Contoh : *Printer*, *Display* *Monitor*, *Sound Speaker*, pengontrolan lewat *interface* dan lain-lain.

##### Gambar 2.8 *Mouse* Sebagai *Input* *Device*

*(Sumber : https://blog.dimensidata.com/macam-jenis-tipe-mouse-komputer-dan-kelebihannya/)*

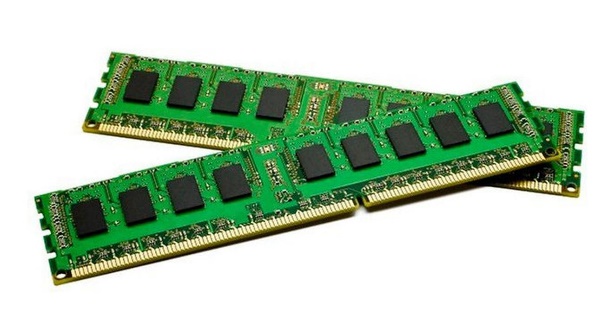
1. *Central Processing Unit* (CPU)

*Central Processing Unit* adalah merupakan pusat dari semua pemrosesan data yang diinputkan ke komputer. CPU terbagi menjadi 2 bagian yaitu Satuan Kendali (*Control Unit*) dan Satuan Logika Aritmatika (*Aritmethic Logical Unit*).

##### Gambar 2.9 CPU Intel

*(Sumber : https://techno.okezone.com/read/2020/08/08/16/2259018/cpu-intel-generasi-ke-12-alder-lake-s-dukung-memori-ddr5)*

1. *Memory*

Memori yang menjadi pendukung CPU dalam melakukan proses, dimana memori bertugas untuk menampung data yang akan diproses serta data hasil proses dan dari CPU.

##### Gambar 2.10 *Memory* pada komputer

*(Sumber : https://www.liputan6.com/tekno/read/2969231/3-tips-mudah-meningkatkan-kinerja-ram-pada-android)*

1. *Interface*

*****Interface* adalah sarana penghubung antara CPU dengan perangkat *input* dan *output*.

##### Gambar 2.11 *Interface Motherboard*

*(Sumber : https://haloedukasi.com/komponen-motherboard)*

1. *Secondary Storage*

*Secondary storage* adalah alat yang digunakan untuk menyimpan program dan data yang tidak dilibatkan dalam proses yang aktif pada suatu saat, alat ini terletak atau terpisah diluar *main memo*ry.

##### Gambar 2.12 *Harddisk* Sebagai Media *Storage*

*(Sumber : https://computory.com/fungsi-hard-disk/)*

### 2.2.2 Media Pembelajaran

Menurut Nasution (2000), media pengajaran adalah sebagai alat bantu mengajar, yakni penunjang penggunaan metode mengajar yang dipergunakan guru. Manfaat media pembelajaran sebagai alat bantu dalam proses pembelajaran ada empat yaitu.

1. pengajaran akan lebih menarik perhatian siswa sehingga dapat menumbuhkan motivasi belajar.
2. bahan pengajaran akan lebih jelas maknanya, sehingga dapat lebih dipahami siswa, serta memungkinkan siswa menguasai tujuan pengajaran dengan baik.
3. metode pembelajaran akan bervariasi, tidak semata-mata hanya komunikasi verbal melalui penuturan kata-kata lisan pengajar, siswa tidak bosan, dan pengajar tidak kehabisan tenaga.
4. siswa lebih banyak melakukan kegiatan belajar, sebab tidak hanya mendengarkan penjelasan dari pengajar saja, tetapi juga aktivitas lain yang dilakukan seperti mengamati, melakukan, mendemonstrasikan dan lain-lainya.

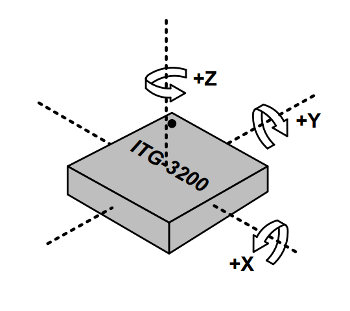
### 2.2.3 Simulasi

Simulasi merupakan suatu metode latihan yang menirukan sesuatu dalam bentuk tiruan yang serupa dengan situasi dunia nyata. Simulasi menggambarkan sistem dengan peragaan model aktor dan objek. Pembelajaran melalui simulasi dapat meningkatkan keterampilan berinteraksi dan berkomunikasi peserta didik. Pada pembelajaran melalu simulasi peserta didik diminta untuk melakukan serangkaian peran atau perilaku yang dianggap sesuai dengan tujuan pembelajaran (Hasbullah, 2021).

Menurut Wahyuni (2012), Pembelajaran melalui simulasi memiliki tujuan tertentu yakni:

1. Melatih kemampuan yang bersifat profesional dan kehidupan sehari-hari.
2. Memahami tentang suatu konsep atau prinsip.
3. Meningkatkan aktivitas belajar dengan mempelajari situasi yang mirip dengan situasi dunia nyata.
4. Meningkatkan motivasi anak melalui media interaktif dan menghibur.

### 2.2.4 Sensor Gyroscope

Gyrosope merupakan sebuah perangkat keras yang berfungsi untuk mengukur atau mempertahankan orientasi, dengan prinsip ketetapan momentum sudut. Analogi dari gyroscope adalah sebuah roda yang berputar dengan piringan di dalam piringan tersebut yang tetap stabil. Sebuah gyroscope beroperasi dengan kebebasan terhadap tiga sumbu, yakni sumbu x, sumbu y, dan sumbu z. Nilai sudut yang dihasilkan akan disesuaikan dengan perubahan gerakan pada sudut tertentu.

##### Gambar 2.13 Sensor Gyroscope

*(Sumber : https://* *https://learn.sparkfun.com/tutorials/gyroscope/all/)*

### 2.2.5 Android

Android adalah sistem operasi berbasis linux yang dimodifikasi untuk perangkat bergerak (*mobile device*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi inti (Susanti & Aifan, 2019). Android adalah sistem operasi untuk *smartphone* yang dibuat oleh Google Corporation. Sistem operasi ini dikembangkan dengan memanfaatkan linux kernel (Subari, 2014). Android bersifat *Open* *Source*, sehingga pengguna dapat membuat aplikasi milik mereka sendiri dengan bantuan Android SDK, dan bahkan Google memperbolehkan pengguna untuk memodifikasi sistem operasi.

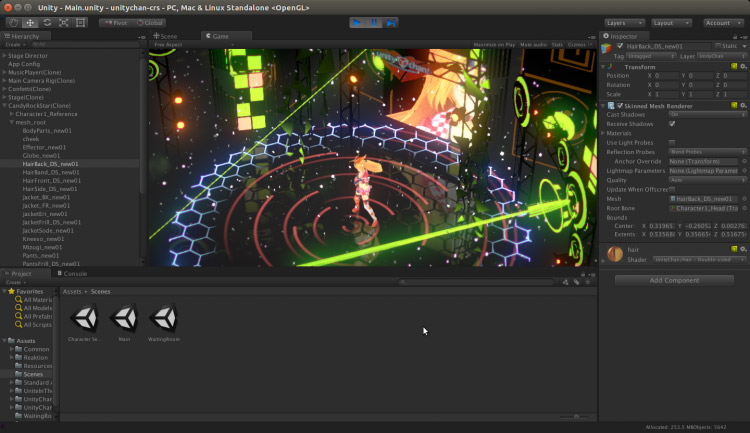
##### Gambar 2.14 Versi Android Beserta *Codename*

*(Sumber : https://salamadian.com/tingkatan-urutan-versi-android/)*

### 2.2.6 Unity

Unity adalah sebuah *game engine* yang berbasis *cross-platform*. Unity dapat digunakan untuk membuat sebuah *game* yang bisa digunakan pada perangkat komputer, *android*, iPhone, PS3, dan bahkan X-BOX. Unity adalah sebuah sebuah tool yang terintegrasi untuk membuat *game*, arsitektur bangunan dan simulasi (Kusuma & Wirawan, 2017).

Unity juga dapat menghasilkan permainan untuk *browser* dengan menggunakan *plugin* Unity Web Player. Unity 3D, juga memiliki kemampuan untuk mengekspor permainan yang dibangun untuk fungsionalitas Adobe Flash 3D (Rizki, 2012).



##### Gambar 2.15 Tampilan *Software* Unity

*(Sumber : https://www.omgubuntu.co.uk/2015/08/unity3d-linux-build)*

### 2.2.7 Blender

Blender merupakan perangkat lunak yang bersifat *open source* digunakan untuk membuat animasi tiga dimensi. Selain itu, perangkat lunak ini juga dapat digunakan untuk membuat animasi, efek visual, model cetak 3D, aplikasi 3D interaktif dan pendukung dalam pembuatan *game* 3D. Blender memiliki beberapa fitur termasuk *rigging, animation, simulation, rendering, compositing and motion tracking, video editing dan game creation*. (Pranata & Santyadiputra, 2017)

Blender memiliki ukuran instalasi yang relatif kecil dan dapat diimplementasikan di semua platform komputer. Walaupun sering didistribusikan tanpa adanya dokumentasi yang cukup atau tanpa contoh yang jelas, *software* ini mengandung beberapa fitur yang hampir sama dengan *software modelling* terbaru. Beberapa kemampuan dari blender adalah mendukung keanekaragaman dari bentuk *geometri primitiv*e, termasuk *polygon* yang tak beraturan, *fast subdivision*, *surfaced modeling,* *kurva bezi*er, *meatballs* dan lain lain (Moloeng, 2013).

Didukung dengan *keyframed animation tools* termasuk *kinematic invers*, *armature (skeleton), shape keys (morphing*), animasi nonlinier, pemberian bobot pada *vertex*, pendeteksian *mesh colution*, *particle based hair*, dan partikel sistem dengan *collution dete*ction. Didukung oleh *phyton* scripting untuk menciptakan tools baru dan *prototyping*, *game logic, import dan export* dari format lain seperti OBJ, FBX, DFX dan *task automation*. Memiliki kemampuan untuk *video editing* atau *audio* yang nonlinier dan masih banyak lagi fitur yang lain yang merupakan teknologi *high-end* (Moleong, 2013).

##### Gambar 2.16 Tampilan *Software* Blender

*(Sumber : https://medium.com/imeshup/blender-pros-cons-quirks-and-links-3a9bf803826f)*

### 2.2.8 Unified Modeling Language

UML (Unified Modeling Language) adalah bahasa spesifikasi standar yang dipergunakan untuk membangun, mendokumentasikan dan menspesifikasikan perangkat lunak (Windu & Grace, 2013). Unified Modeling Language merupakan metodologi dalam mengembangkan sistem berorientasi objek, merupakan alat dalam mendukung pengembangan sistem aplikasi. Unifield Modeling Language memungkinkan pengembang aplikasi untuk melakukan permodelan secara visual, yaitu melakukan penekanan saat penggambaran model, bukan didominasi oleh teks narasi penjelasan. Permodelan visual dapat membantu pembaca dalam menangkap struktur dari objek,

Pada pengerjaan projek aplikasi ini, penulis menggunakan Unified Modeling Language dengan tipe *behavior diagrams* untuk menampilkan perilaku apa yang harus terjadi dalam sistem yang dimodelkan. Diagram ini digunakan secara luas untuk menggambarkan fungsionalitas sistem perangkat lunak. Berikut ini merupakan tiga jenis diagram perilaku yang digunakan penulis dalam penelitian ini:

1. *Use Case Diagram*

*Use case diagram* adalah bentuk utama dari persyaratan sistem untuk program perangkat lunak. *Use case diagram* dapat menentukan hal yang dapat dilakukan dan bukan penjelasan cara kerjanya. *Use case diagram* yang telah ditentukan dapat dilambangkan dengan menggunakan representasi tekstual dan visual. *Use case diagram* menggambarkan *external view* dari sistem yang akan dibuat atau dirancang. Berikut ini adalah simbol-simbol yang umum digunakan untuk menggambar *use case diagram*:

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

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | *Actor* | Pelaku atau orang yang berperan dalam suatu sistem atau perangkat lunak. |
|  | *Dependency* | Hubungan antara suatu elemen mandiri kemudian mempengaruhi elemen yang tidak mandiri. |
|  | *Generalization* | Hubungan antara objek anak dengan objek induk. |
| <<include>> | *Include* | Menspesifikasikan suatu use case sumber eksplisit. |
| <<extend>> | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
|  | *Association* | Penghubung antar objek. |
|  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
|  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem. |
|  | *Collaboration* | Interaksi antar elemen sehingga menjadi elemen yang lebih besar. |
|  | *Note* | Elemen fisik yang ada saat aplikasi dijalankan dan mencerminkan sumber daya komputasi. |

1. *Activity Diagram*

*Activity diagram* adalah diagram yang menggambarkan suatu proses atau urutan aktivitas dalam sebuah proses. *Activity diagram* dipakai pada *business* modeling untuk memperlihatkan urutan aktifitas proses bisnis. Memiliki struktur diagram yang hampir sama atau mirip dengan *flowchart* yang ada pada perancangan terstruktur. Berikut ini adalah simbol-simbol yang umum digunakan untuk menggambar *activity* *diagram*:

#### Tabel 2.2 Simbol *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | *Activity* | Memperlihatkan masing-masing kelas yang saling berinteraksi. |
|  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi. |
|  | *Initial Node* | Awal dari dibentuknya objek. |
|  | *Activity Final Node* | Akhir dari dibentuknya objek. |
|  | *Decision* | Menggambarkan suatu keputusan atau tindakan dari suatu kondisi. |
|  | *Line Connection* | Menghubungkan antar simbol. |

1. *Sequence Diagram*

*Sequence diagram* adalah diagram yang menjelaskan bagaimana suatu proses dijalankan dalam suatu program. *Sequence diagram* digunakan untuk menggambarkan langkah-langkah yang dilakukan untuk mencapai suatu hasil yang diinginkan. Berikut ini adalah simbol-simbol umum yang biasa digunakan dalam penggambaran *sequence diagram*:

#### **Tabel 2.3** Simbol *Sequance Diagram*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | *Actor* | Orang atau sistem yang melakukan sebuah aksi. |
|  | *Object* | Objek yang terlibat dalam sistem. |
|  | *Lifeline* | Garis yang menggambarkan masa aktif dari sebuah objek. |
| Message | *Line Message* | Pengiriman pesan. |
|  | *Entity Class* | Menggambarkan hubungan kegiatan yang akan dilakukan. |
|  | *Boundary Class* | Menggambarkan sebuah *form*. |
|  | *Control Class* | Menghubungkan *boundary* dengan tabel. |