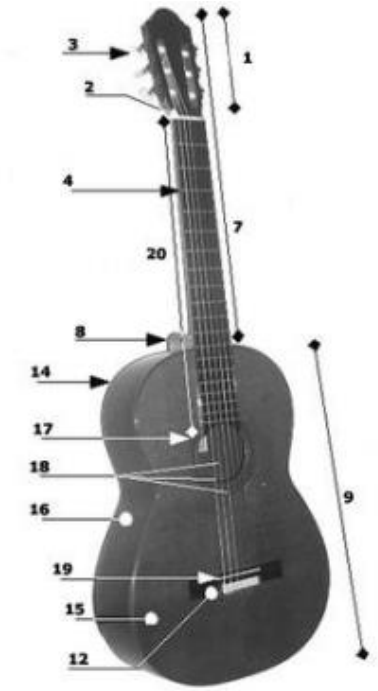
# BAB 2
# LANDASAN TEORI

## 2.1 Gitar

Gitar adalah alat musik yang dimainkan dengan cara dipetik. Suara gitar dihasilkan dari senar yang bergetar. Getaran tersebut bergema di dalam badan gitar yang merupakan sebuah ruang yang terbuat dari kayu, akan tetapi berbeda halnya dengan gitar elektrik yang menghasilkan suara dengan mengandalkan magnet. Berikut ini adalah gambaran dari sebuah gitar[4] :

a. *Headstock* (kepala) (1)
b. *Nut* (2)
c. Alat Pemutar (3)
d. *Frets* (4)
e. *Neck* (Leher) (7)
f. *Heel* (Penghubung) (8)
g. Badan Gitar (9)
h. *Bridge* (12)
i. Bagian Belakang (14)
j. *Soundboard* (Bagian Penghasil Suara / kotak resonansi) (15)
k. Bagian Sisi (Samping) (16)
l. Lubang suara (17)
m. Senar (18)
n. *Saddle* (19)
o. *Fretboard* (*Fingerboard*) (20)



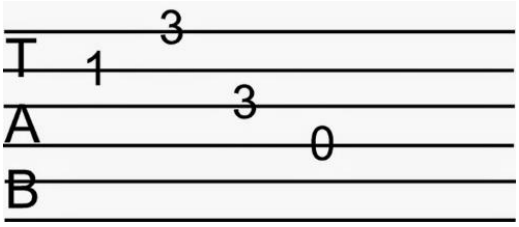

**Sumber Gambar : guitar.about.com[14]**

**Gambar 2. 1 Gitar**

Semua bagian gitar merupakan satu kesatuan, akan tetapi untuk menghasilkan suara yang berbeda, maka bagian yang paling penting adalah *fret*, senar dan *fingerboar*

## 2.2 Tablatur Gitar

Tablatur gitar adalah sebuah media (media cetak) yang menggambarkan bagaimana cara memainkan *melody* gitar.




**Sumber Gambar : http://guitar.about.com[4]**

**Gambar 2. 2 Tablature Gitar**

Berikut adalah penjelasan dari gambar :

a. Angka menunjukan di *fret* berapa kita harus menekan senar. *Fret* merupakan bagian dari gitar yang digunakan sebagai penampang senar saat ditekan.
b. Garis paling atas atau garis pertama menunjukan senar E tinggi
c. Garis ke 2 menunjukan senar B
d. Garis ke 3 menunjukan senar D
e. Garis ke 4 menunjukan senar G
f. Garis ke 5 menunjukan senar A
g. Garis ke 6 atau Garis paling bawah menunjukan senar E Rendah

Berikut istilah dalam tablatur[4]:

a. *pull off* : tempatkan jari bersamaan pada not-not yang akan dibunyikan. Petik not awal, tarik jari pertama anda(yang memetik not awal) untuk membunyikan not berikutnya.

b. *slide* : petik not awal, geser jari pada not berikutnya (lebih rendah atau tinggi). Ada dua slide disini, pertama: tanpa memetik not berikutnya (sekali petik 2 not), kedua: dengan memetik not berikutnya (dua petik, 2 not).
c. *tapping* : dengan ujung jari telunjuk atau tengah ketuklah not yang ingin dibunyikan pada *fretboard.*
*d.* *trill* : selang-seling dengan cepat antara not awal dan kedua memakai *hammer on* dan *pull off.*
e. *pick slide* : bunyikan not dengan memakai sisi *pick,* sehingga muncul suara gerit/seret*(scratch).*
f. *hammer on* : petik not awal(pertama) / not lebih rendah dengan cara *picking*(memetik dengan *pick*) lalu bunyikan not lebih tinggi dengan jari lain dengan tanpa *picking* lagi.
g. *bend* : tekan kuat jari kiri pada *fretboard* kemudian naikkan jari anda hingga nada yang dihasilkan menjadi naik.
h. *bend and release* : petik, bending hingga nadanya naik 1/2 atau 1, kemudian balik lagi ke nada awalnya.
i. *pre bend* : bending dulu hingga nadanya naik 1/2 atau 1, baru dipetik.
j. *pre bend and release* : bending dulu hingga nadanya naik 1/2 atau 1, petik, bending lagi hingga bunyinya balik ke not awal.
k. *unison bend* : bending dua buah not bersamaan.
l. *artificial harmonic* : bunyikan not dengan seperti biasa, hanya saja pada *picking* tangan kanan iringi dengan sisi jempol, atau pucuk jari telunjuk, dengan *volume* atau tingkat efek distorsi tinggi akan menghasilkan bunyi *decit harmonic*.
m. *tremolo bar* : memakai *tremolo*/*whammy bar*, turunkan not pada tablatur sejauh yang ditunjukkan, lalu balikkan lagi ke not awalnya.
n. *palm muting* : diamkan not yang bersuara dengan telapak tangan kanan(bagi *left handed guitarist*)/kiri(bagi *right handed guitarist*).
*o.* *muffled strings* : letakkan jari dengan tanpa menekan terlalu kuat pada *fretboard* sehingga menghasilkan *sound percussive.*

p. *vibrato* : efek getar, dengan bending, getarkan senar dengan bending naik turun secara cepat, bisa dengan tangan kiri atau memakai tremolo/whammy bar
q. *wide atau exaggerated vibrato* : bending naik turun lebih kuat lagi dari vibrato bias
r. *natural harmonic* : bunyikan not dengan menyentil(menyentuh ringan) dengan jari kiri hingga berbunyi seperti lonceng “ting”

## 2.3 Aplikasi

Aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.[5].

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyaraan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*[5].

Aplikasi merupakan satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas. Misalnya termasuk perangkat lunak perusahaan, *software* akuntansi, perkantoran, grafis perangkat lunak dan pemutar media. Dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Program aplikasi merupakan program siap pakai. Program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi akan menggunakan sistem operasi (OS) komputer dan aplikasi yang lainnya yang mendukung.

Klasifikasi aplikasi dapat dibagi menjadi 2 (dua) yaitu:

a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
b. Aplikasi paket, dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

## 2.4 Penerjemahan

Penerjemahan adalah pengalihan pikiran atau gagasan dari suatu bahasa sumber ke dalam bahasa yang lain. Penerjemahan adalah mengubah teks bahasa sumber ke dalam teks bahasa sasaran dengan mempertimbangkan makna kedua bahasa sehingga diusahakan semirip-miripnya, yang tak kalah pentingnya adalah terjemahan harus mengikuti kaidah-kaidah yang berlaku dalam bahasa sasaran[6].

Namun suatu hal yang perlu diperhatikan adalah bahwa struktur suatu bahasa akan sangat berbeda dengan struktur bahasa yang lain, walaupun kadang-kadang memang ada sebagian yang sama.

Penerjemahan juga membutuhkan seni dalam menyusun kembali hasil terjemahan ke dalam bahasa sasaran. Seni yang dimaksud di sini adalah seni agar bahasa yang digunakan mudah dipahami, tidak kaku, dan memiliki gaya bahasa sasaran tepat, sehingga tulisan yang dibaca merupakan hasil terjemahan atau dengan kata lain bahwa seolah-olah hasil terjemahan yang ditulis merupakan tulisan asli.

## 2.5 Sistem Operasi Android

Pada bagian ini dijelaskan hal-hal yang berkaitan dengan sistem operasi Android, diantaranya pengenalan sistem operasi Android, arsitektur Android, dan Android SDK (*Software Development Kit*).

### 2.5.1 Pengenalan Sistem Operasi Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi [7]. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka [7]. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti

lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada masa saat ini, sebagian besar vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis Android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi Android. Hal ini karena Android itu adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun [7].

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan di atas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, aplikasi dan tool pengembangan, *market* aplikasi Android, serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga Android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia [7].

Android telah melalui cukup banyak pembaruan sejak pertama rilis.

**Tabel 2. 1 Android Version**

| Versi Android | Tanggal Rilis | Nama Kode |
|---|---|---|
| 1.0 | 23 September 2008 | - |
| 1.1 | 9 February 2009 | - |
| 1.5 | 30 April 2009 | Cupcake |
| 1.6 | 15 September 2009 | Donut |
| 2.0/2.1 | 26 Oktober 2009 | Éclair |
| 2.2 | 20 Mei 2010 | Froyo |
| 2.3 | 6 Desember 2010 | Gingerbread |
| 3.0/3.1/3.2 | 22 Februari 2011 | Honeycomb |
| 4.0 | 19 Oktober 2011 | Ice Cream Sandwich |
| 4.1/4.2/4.3 | 9 Juli 2012 | Jelly Bean |
| 4.4 | 31 Oktober 2013 | KitKat |
| 5.0 | 3 November 2014 | Lollipop |

| | | |
|---|---|---|
| 6.0 | 2015 | Marshmallow |

### 2.5.2 Arsitektur Android

Secara garis besar, arsitektur Android terdiri atas 5 bagian, diantaranya sebagai berikut [7]:

1. *Application* dan *Widgets*

   Merupakan *layer* (lapis), dimana pengguna berhubungan dengan aplikasi saja.

2. *Application Frameworks*

   Merupakan *Open Development Platform* yang ditawarkan Android untuk dapat dikembangkan guna membangun aplikasi. Pengembang memiliki akses penuh menuju API *Frameworks* seperti yang di lakukan oleh aplikasi kategori inti. Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah *Views*, *Content Provider*, *Resource Manager*, *Notification Manager*, dan *Activity Manager*.

3. *Libraries*

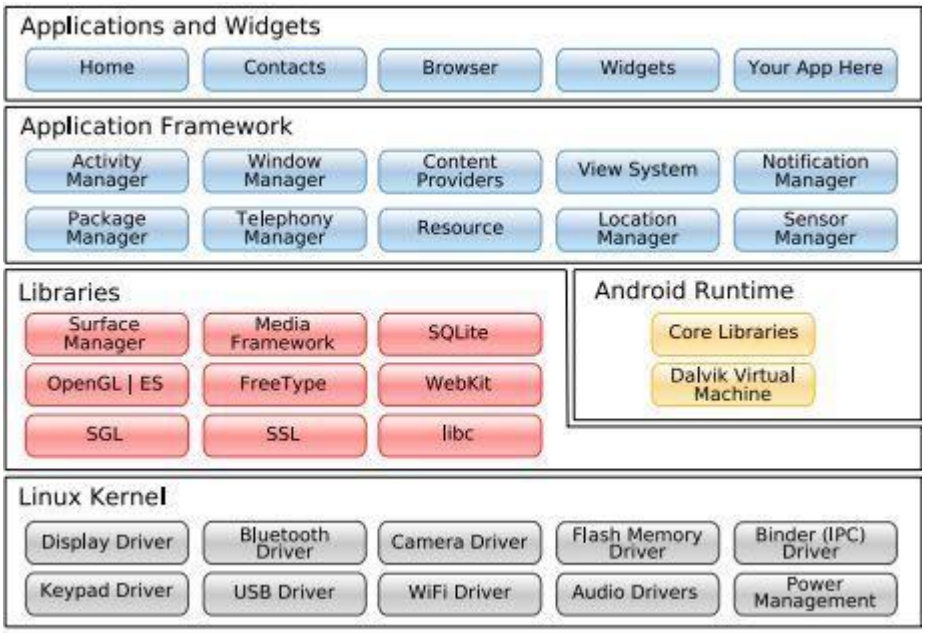
   Merupakan *layer*, dimana fitur-fitur Android berada.

4. *Android Run Time*

   Merupakan *layer* yang membuat aplikasi Android dapat dijalankan, dimana dalam prosesnya menggunakan implementasi Linux.

5. Linux Kernel

   Merupakan *layer* inti dari sistem operasi Android berada.




**Sumer Gambar : Pemrograman Android[7]**

**Gambar 2. 3 Arsitektur Android**

### 2.5.3 Android SDK (Software Development Kit)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middeware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan *handphone*/*smartphone* [7].

## 2.6 Augmented Reality

*Augmented Reality* (*AR*) adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. Ronald Azuma pada tahun 1997 mendenisikan *Augmented Reality* sebagai sistem yang memiliki karakteristik sebagai berikut [2]:

1. Menggabungkan lingkungan nyata dan virtual
2. Berjalan secara interaktif dalam waktu nyata
3. Integrasi dalam tiga dimensi (3D)

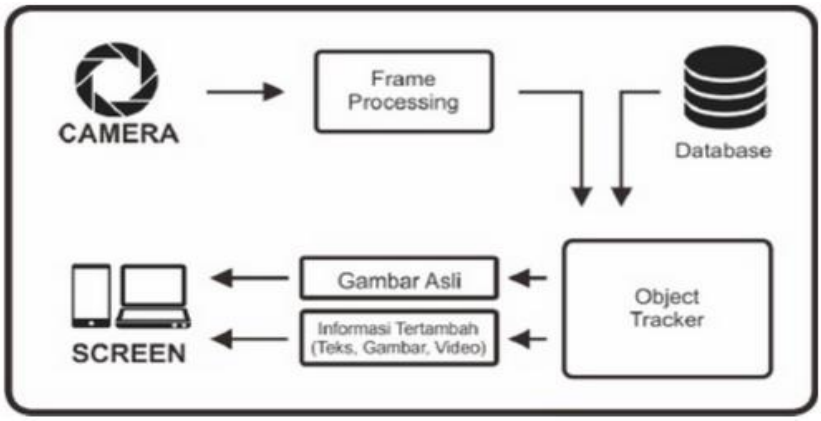
Secara sederhana AR bisa didenisikan sebagai lingkungan nyata yang ditambahkan objek virtual. Penggabungan objek nyata dan virtual dimungkinkan dengan teknologi display yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu.

AR merupakan variasi dari *Virtual Environments* (VE), atau yang lebih dikenal dengan istilah *Virtual Reality* (VR). Teknologi VE membuat pengguna tergabung dalam sebuah lingkungan virtual secara keseluruhan. Ketika tergabung dalam lingkungan tersebut, pengguna tidak bisa melihat lingkungan nyata di sekitarnya. Sebaliknya, AR memungkinkan pengguna untuk melihat lingkungan nyata, dengan objek virtual yang ditambahkan atau tergabung dengan lingkungan nyata. Tidak seperti VR yang sepenuhnya menggantikan lingkungan nyata, AR sekedar menambahkan atau melengkapi lingkungan nyata.

Tujuan utama dari AR adalah untuk menciptakan lingkungan baru dengan menggabungkan interaktivitas lingkungan nyata dan virtual sehingga pengguna merasa bahwa lingkungan yang diciptakan adalah nyata. Dengan kata lain, pengguna merasa tidak ada perbedaan yang dirasakan antara AR dengan apa yang mereka lihat/rasakan di lingkungan nyata. Dengan bantuan teknologi AR (seperti visi komputasi dan pengenalan objek) lingkungan nyata disekitar kita akan dapat berinteraksi dalam bentuk digital (virtual). Informasi tentang objek dan lingkungan disekitar kita dapat ditambahkan kedalam sistem AR yang kemudian informasi tersebut ditampilkan diatas layer dunia nyata secara realtime seolah-olah informasi tersebut adalah nyata. Informasi yang ditampilkan oleh

objek virtual membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata. AR banyak digunakan dalam bidang-bidang seperti kesehatan, militer, industri manufaktur dan juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam [2].

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




**Sumber Gambar : Ronald[2]**

**Gambar 2. 4 Alur kerja *Augmented Reality***

Berikut ini adalah penjelasannya :

a. Camera

Digunakan untuk men-*scan marker* / *markerless.* Camera akan membaca objek untuk kemudian dikenali sebagai marker.

*b. Frame Processing*

Proses perhitungan pixel yang digunakan untuk pendeteksian gambar atau objek. Pada proses iti terjadi prose *Grayscaling* dan deteksi tepi.

c. Database

Digunakan untuk menyimpan data marker

*d. Object Tracker*

Objek yang dilacak yang akan di tampilkan informasi nya jika sesuai dengan data yang ada didalam database. Jika cocok maka objek 2 dimensi atau 3 dimensi akan ditampilkan yang kemudian disebut *Augmented Reality*.

e. Informasi

Menampilkan informasi yang sesuai dengan *object tracker* yang sudah di inisilisasi didatabase. Informasi bias berupa tulisan ataupun suara.

f. Screen

Untuk menampilkan keluaran dari aplikasi yang berupa Teks, Gambar, Video, dan Objek 2d dan 3D. Bagian ini merupakan hasil yang di tampilkan sistem yang dapat kita lihat di mobile maupun PC.

## 2.7 Markerless Augmented Reality

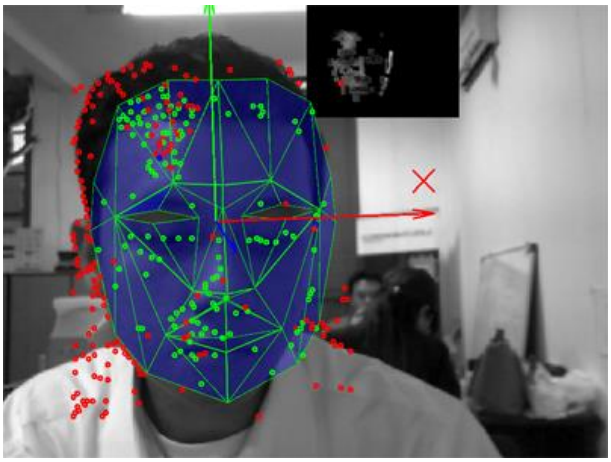
Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode *Markerless Augmented Reality*, dengan metode ini pengguna tidak perlu lagi menggunakan sebuah marker untuk menampilkan elemen-elemen digital.

Dalam perancangannya, seolah-olah markerless menggabungkan objek virtual dengan objek nyata, dalam hal ini objek virtual berupa objek 2D atau 3D dan objek nyatanya berupa gambar dengan pola tertentu (*markerless*). Meski dinamakan dengan markerless namun aplikasi tetap berjalan dengan melakukan pemindaian terhadap object, namun ruang lingkup yang dipindai lebih luas dibanding dengan pemindaian menggunakan metode marker based tracking. Berbagai macam teknik *Markerless Based Tracking* sebagai teknologi yang saat ini terus dikembangkan adalah *Face Tracking*, *Object Tracking*, dan *Motion Tracking*.

a. *Face Tracking*

Dengan menggunakan algoritma yang sedang dikembangkan, komputer dapat mengenali wajah manusia secara umum dengan cara mengenali posisi

mata, hidung, dan mulut manusia, kemudian akan mengabaikan objek-objek lain di sekitarnya seperti pohon, rumah, dan benda-benda lainnya. Contoh *face tracking* dapat dilihat pada gambar 2.5.



**Sumber Gambar : it-jurnal.com[8]**

**Gambar 2. 5** ***Face Tracking***

b. *Object Tracking*

Berbeda dengan *Face Tracking* yang hanya mengenali wajah manusia, secara umum teknik *Object Tracking* dapat mengenali semua bentuk benda yang ada disekitar, seperti mobil, meja, televisi, dan lain-lain. Contoh *Object Tracking* dapat dilihat pada gambar 2.6

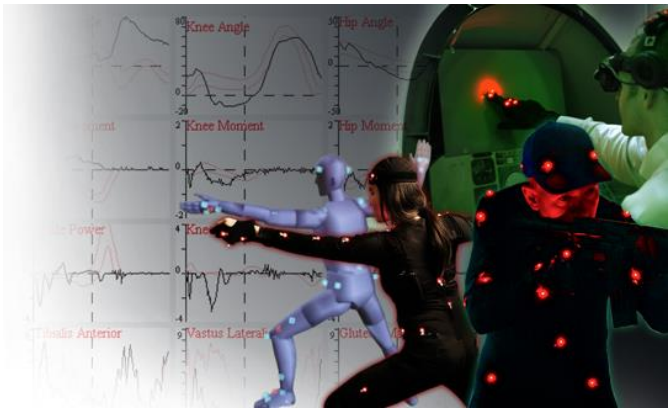


**Sumber Gambar : it-jurnal.com[8]**

**Gambar 2. 6** ***3D Object Tracking***

c. *Motion Tracking*

Pada teknik ini komputer dapat menangkap gerakan. *Motion Tracking* mulai digunakan secara ekstensif untuk memproduksi film-film yang mencoba mensimulasikan gerakan. Contohnya pada film *Avatar The Last Airbender*, di mana James Cameron menggunakan teknik ini untuk membuat film tersebut dan menggunakannya secara *realtime*. Contoh *motion tracking* dapat dilihat pada gambar 2.7




**Sumber Gambar : it-jurnal.com[8]**

**Gambar 2. 7** ***Motion Tracking***

## 2.8 Object Oriented Analysis and Design (OOAD)

Pada bagian ini dijelaskan hal-hal yang berkaitan dengan analisis dan perancangan berorientasi objek, diantaranya pengenalan *Object-Oriented Analysis and Design* (OOAD) dan *Unified Modelling Language* (UML).

### 2.8.1 Pengenalan OOAD

Secara spesifik, pengertian berorientasi objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan

pemrograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, *polymorphism* (polimorfisme), dan *inheritance* (pewarisan) [9].

Analisis dan desain berorientasi objek atau *Object-Oriented Analysis and Design* (OOAD) adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Model berorientasi objek bermanfaat untuk memahami masalah, komunikasi dengan ahli aplikasi, pemodelan suatu organisasi, menyiapkan dokumentasi, serta perancangan program dan basis data. Pertama-tama, suatu model analisis dibuat untuk menggambarkan aspek dasar dari domain aplikasi, dimana model tersebut berisi objek yang terdapat dalam domain aplikasi termasuk deskripsi dari keterangan objek dan perilakunya [9].

Analisis Berorientasi Objek atau *Object-Oriented Analysis* (OOA) dimulai dengan menyatakan suatu masalah, analisis membuat model situasi dari dunia nyata, menggambarkan sifat yang penting. Dalam menganalisa suatu sistem, analisis harus bekerja dengan pihak yang membutuhkan sistem untuk memahami masalah tersebut dengan jelas. Model analisis adalah abstraksi yang ringkas dan tepat dari apa yang harus dilakukan oleh sistem dan bagaimana melakukannya. Objek dalam model harus merupakan konsep domain dari aplikasi dan bukan merupakan implementasi komputer seperti struktur data. Model yang baik harus dipahami dan ditanggapi oleh ahli aplikasi. Empat kesulitan yang merupakan gangguan utama dalam pengembangan sistem adalah memahami *problem* domain, komunikasi antara pihak yang berkaitan, perubahan kontinyu, dan *reuse* (penggunaan kembali) [9].

Desain Berorientasi Objek atau *Object-Oriented Design* (OOD) merupakan tahap lanjutan setelah Analisis Berorientasi Objek dimana tujuan sistem diorganisasi ke dalam sub-sistem berdasar struktur analisis dan arsitektur yang dibutuhkan. Desainer sistem (*system designer*) menentukan karakteristik

penampilan secara optimal, menentukan strategi memecahkan masalah, dan menentukan pilihan alokasi sumber daya. Sebagai contoh, desainer sistem mungkin menentukan perubahan pada *screen* untuk *workstation* yang memerlukan kecepatan serta resolusi lebih tinggi [9].

Desain model digunakan berdasarkan model analisis dengan dilengkapi rincian untuk implementasi. Fokus dari desain objek (*object design*) adalah perencanaan struktur data dan algoritma yang diperlukan untuk implementasi setiap kelas. Objek domain aplikasi dan objek domain komputer dijelaskan dengan menggunakan konsep dan notasi berorientasi objek yang sama [9].

Kelas, objek dan relasi dikembangkan dalam tahap pembuatan desain objek yang pada akhirnya diterjemahkan ke dalam bahasa pemrograman, basis data, dan implementasi perangkat keras. Hal yang penting dalam tahap implementasi adalah mengikuti penggunaan perangkat lunak yang baik [9].

### 2.8.2 Unified Modelling Language (UML)

*Unified Modelling Language* (UML) adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, serta didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI). Standar UML dikelola oleh OMG (*Object Management Group*) [10].

Tujuan utama perancangan UML adalah:

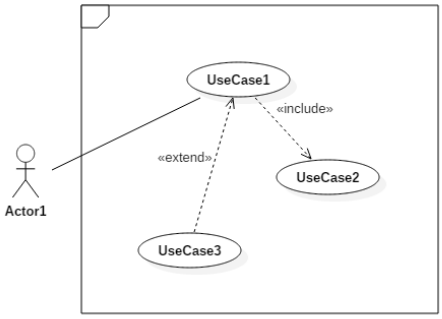
a. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
b. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.
c. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
d. Menyediakan basis formal untuk pemahaman bahasa pemodelan.

e. Mendorong pertumbuhan pasar kakas berorientasi objek.
f. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

UML menyediakan diagram-diagram yang sangat kaya dan dapat diperluas sesuai kebutuhan kita. Diagram adalah representasi secara grafis dari elemen-elemen tertentu beserta hubungan-hubungannya. Diagram penting karena diagram menyediakan representasi secara grafis dari sistem (atau bagiannya). Representasi grafis sangat memudahkan pemahaman terhadap sistem [10].

Diagram di dalam UML terbagi menjadi sebagai berikut :

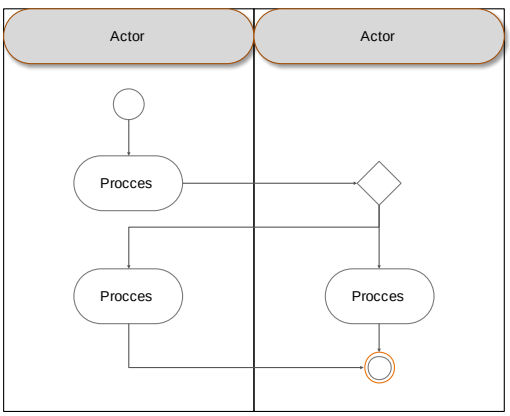
a. Diagram *Use Case* (*Use Case Diagram*)




**Gambar 2. 8** ***Use Case Diagram***

Diagram ini menunjukkan sekumpulan kasus fungsional dan aktor (jenis kelas khusus) dan keterhubungannya.

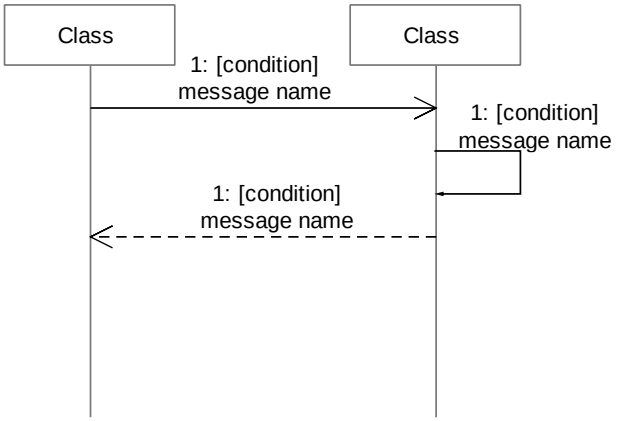
b. Diagram Aktivitas (*Activity Diagram*)



**Gambar 2. 9** ***Activity Diagram***

Diagram ini menunjukan aliran aktivitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem. Diagram ini penting untuk memodelkan fungsi sistem dan menekankan pada aliran kendali di antara objek-objek.

c. Diagram Sekuen (*Sequence Diagram*)




**Gambar 2. 10** ***Sequence Diagram***

Diagram ini menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi.

d. Diagram Kelas (*Class Diagram*)




**Gambar 2. 11** ***Class Diagram***

Diagram ini menunjukkan sekumpulan kelas, interface, serta kolaborasi dan keterhubungannya. Diagram kelas ditujukan untuk pandangan statik terhadap sistem.

## 2.9 Web Service

Perangkat peranti lunak integrasi aplikasi perusahaan sifatnya spesifik terhadap suatu produk, artinya perangkat ini hanya dapat bekerja dengan bagian-bagian peranti lunak dan sistem operasi tertentu. Contoh, suatu perangkat EAI untuk menghubungkan bagian tertentu dari peranti lunak pencatatan penjualan untuk aplikasi produksi, pengangkutan, dan penagihan, tidak dapat bekerja dengan peranti lunak pencatatan penjualan dari vendor lainnya. Program middleware yang dikembangkan oleh sistem BEA, sebuah integrator yang besar, tidak dapat berhubungan dengan aplikasi middleware vendor lainnya yang dibeli pada tahun-tahun sebelumnya tanpa pengeluaran yang besar untuk pemrograman dan perancangan. Layanan web (web services) mencoba memberikan alternatif standar untuk menghadapi masalah-masalah integrasi seperti itu dengan menciptakan sebuah lingkungan komunikasi yang netral [7].

Layanan web (web services) adalah serangkaian komponen peranti lunak yang bertukar informasi antara satu sama lain dengan bebas menggunakan komunikasi web dan bahasa yang standar. Layanan web dapat bertukar informasi antara dua sistem yang berbeda, terlepas dari sistem operasi atau bahasa pemrograman sistem tersebut. Layanan web dapat dipadukan untuk membangun sistem penghubung aplikasi berbasis web standar terbuka dari dua organisasi yang berbeda, dan juga dapat digunakan untuk membuat aplikasi yang menghubungkan sistem terpisah dalam satu perusahaan. Layanan web tidak terikat dengan sistem operasi atau bahasa pemrograman apa pun, dan aplikasi-aplikasi yang berbeda dapat digunakan untuk berkomunikasi satu sama lain dengan cara standar tanpa penulisan kode khusus yang menghabiskan waktu [7].

## 2.10 Tools

*Tools* implementasi merupakan peralatan-peralatan yang dibutuhkan dalam membangun perangkat lunak. Adapun *tools* implementasi yang digunakan untuk membuat Aplikasi menggunakan Teknologi Augmented Reality ini adalah *Library Vuforia, Unity* dan *Android Studio.*

### 2.10.1 Vuforia

Vuforia merupakan software untuk augmented reality yang dikembangkan oleh Qualcomm, yang menggunakan sumber yang konsisten mengenai computer vision yang fokus pada image recognition. Vuforia mempunyai banyak fitur-fitur dan kemampuan, yang dapat membantu pengembang untuk mewujudkan pemikiran mereka tanpa adanya batasan secara teknikal.

Dengan support untuk iOS, Android, dan Unity3D, platform Vuforia mendukung para pengembang untuk membuat aplikasi yang dapat digunakan dihampir seluruh jenis smartphone dan tablet. Pengembang juga diberikan kebebasan untuk mendesain dan membuat aplikasi yang mempunyai kemampuan antara lain (Fernando, 2013):

1. Teknologi computer vision tingkat tinggi yang mengijinkan developer untuk membuat efek khusus pada device.
2. Terus-menerus mengenali multiple image.
3. Tracking dan detection tingkat lanjut.
4. Solusi pengaturan database gambar yang fleksibel.

Target pada vuforia merupakan obyek pada dunia nyata yang dapat dideteksi oleh kamera, untuk menampilkan obyek virtual. Beberapa jenis target pada vuforia adalah :

1. Image targets, contoh : foto, papan permainan, halaman majalah,sampul buku, kemasan produk, poster, kartu ucapan. Jenis target ini menampilkan gambar sederhana dari Augmented.

2. Frame markers, tipe frame gambar 2D dengan pattern khusus yang dapat digunakan sebagai potongan permainan di permainan pada papan.
3. Multi-target, contohnya kemasan produk atau produk yang berbentuk kotak ataupun persegi. Jenis ini dapat menampilkan gambar sederhana Augmented 3D.
4. *Virtual buttons,* yang dapat membuat tombol sebagai daerah kotak sebagai sasaran gambar.

Pada Vuforia, ada 2 jenis workflow dengan dasar database yang dapat dipilih oleh developer, yaitu Cloud Database dan Device Database.

### 2.10.2 Unity

Unity adalah game developing software,dengan built-in IDE yang dikembangkan oleh Unity Technologies. Hal ini digunakan untuk mengembangkan video game untuk plugin web, platform desktop, konsol dan perangkat mobile, dan digunakan oleh lebih dari satu juta pengembang. Unity tumbuh dari OS X didukung permainan alat pengembangan pada tahun 2005 untuk game developing software game multi platform.

Mesin grafis menggunakan Direct3D (Windows , Xbox 360 ) , OpenGL ( Mac , Windows , Linux , PS3 ) , OpenGL ES ( Android , iOS ) , dan kepemilikan API ( Wii ) . Ada dukungan untuk pemetaan mesh , pemetaan refleksi , pemetaan paralaks, bayangan dinamis menggunakan peta bayangan , merender ke tekstur dan efek post-processing layar penuh.

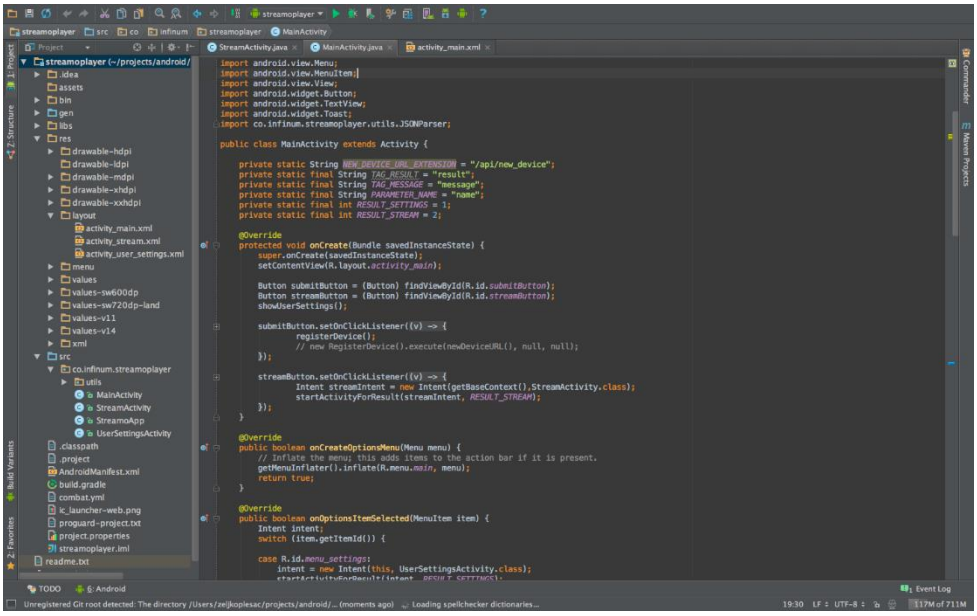
Unity mendukung aset seni dan format file dari 3ds Max , Maya , Softimage , Blender , modo , ZBrush , Cinema 4D , Cheetah3D , Adobe Photoshop , Adobe Fireworks dan Substansi Allegorithmic . Aset ini dapat ditambahkan ke proyek game, dan dikelola melalui antarmuka pengguna grafis Unity.

Bahasa ShaderLab digunakan untuk shader , mendukung kedua deklaratif " pemrograman " dari program tetap fungsi pipa dan shader ditulis dalam GLSL atau Cg . Shader A dapat mencakup beberapa varian dan spesifikasi fallback

deklaratif, memungkinkan Unity untuk mendeteksi varian yang terbaik untuk kartu video saat ini, dan jika tidak ada yang kompatibel , jatuh kembali ke shader alternatif yang mungkin mengorbankan fitur untuk kinerja.

### 2.10.3 Android Studio

Android studio merupakan sebuah *Integrated Development Environment* (IDE) untuk *platform* Android. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi GoogleI/O oleh Produk Manajer Google, Ellie Powers. Android studio bersifat *free* dibawah Apache License 2.0. Android Studio awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudiandibuat versi beta 0.8 yang dirilis pada bulan juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Berbasiskan JetBrainns' IntelliJ IDEA, Studio didesain khusus untuk Android *Development.*



**Sumber gambar : infinum.com**

**Gambar 2. 12 *Android Studio***

Berikut ini beberapa keunggulan yang terdapat pada Android Studio :

a. *Live Layout : WYSIWYG Editor – Live Coding – Real time App Rendering*
b. *Developer Console : optimization tips, assistance for translation, referral tracking,campaigning and promotions – Usage Metrics*
c. *Provision for beta releases and staged rollout*
d. *Gradle-based build support*
e. *Android – specific refactoring and quick fixes*
f. *Lint tools to catch performance, usability, version compatibility and other problems*
g. *ProGuard and app-signing capabilities*
h. *Template-based wizards to create common Android designs and components*
i. *A rich layout editor that allows user to drag-and-drop UI components, option to previewlayouts on multiple screen configurations*
j. *Support for building Android Wear apps*

*Built-in support for Google Cloud Platform, enabling integration with Google Cloud Messaging and App Engine.*

## 2.11 Fast Corner Detection

*FAST (Features from Accelerated Segment Test)* adalah suatu algoritma yang dikembangkan oleh Edward Rosten, Reid Porter, dan Tom Drummond. *FAST corner detection* ini dibuat dengan tujuan mempercepat waktu komputasi secara *real-time* dengan konsekuensi menurunkan tingkat akurasi pendeteksian sudut.

Vuforia menggunakan algoritma FAST Corner Detection untuk mendefenisikan seberapa baik gambar dapat dideteksi dan dilacak menggunakan Vuforia SDK. Peringkat ini ditampilkan dalam *Target Manager* yang telah diupload melalui web API Vuforia. Rating target dapat berkisar dari 0 sampai 5 untuk setiap gambar yang diupload. Semakin tinggi rating target maka akan semakin kuat kemampuan deteksi dan pelacakannya. Sebuah rating 0 menunjukkan bahwa target tidak dilacak sama sekali oleh sistem *Augmented*

*Reality*, sedangkan rating 5 menunjukkan bahwa sebuah gambar akan dengan mudah dilacak sistem *Augmented Reality* yang disediakan oleh Vuforia.

*FAST Corner Detection* bekerja pada suatu citra sebagai berikut :

1. Tentukan sebuat titik *p* pada citra dengan posisi awal ($x_p$,).
2. Tentukan lokasi keempat titik pengujian untuk dibandingkan dengan titik *p*, dengan ketentuan sebagai berikut :
   a. Titik pertama ($n$=1) terletak pada koordinat ($x_p$,$y_{p-3}$),
   b. Titik kedua ($n$=2) terletak pada koordinat ($x_{p+3}$,$y_p$),
   c. Titik ketiga ($n$=3) terletak pada koordinat ($x_p$,$y_{p+3}$),
   d. Titik keempat ($n$=4) terletak pada koordinat ($x_{p-3}$,$y_p$)
3. Intensitas pixels terbagi menjadi tiga subset, yaitu *Much Brigter* (lebih cerah), *Much Darker* (lebih gelap), *Similar* (sama). Bandingkan intensitas titik pusat *p* dengan keempat titik disekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat berikut, maka titik pusat *p* adalah sudut.

$$C_p = \begin{cases} 1, & In<Ip-t \ \ or \ \ In>Ip+t \\ 0, & otherwise \end{cases}$$

   Keterangan :

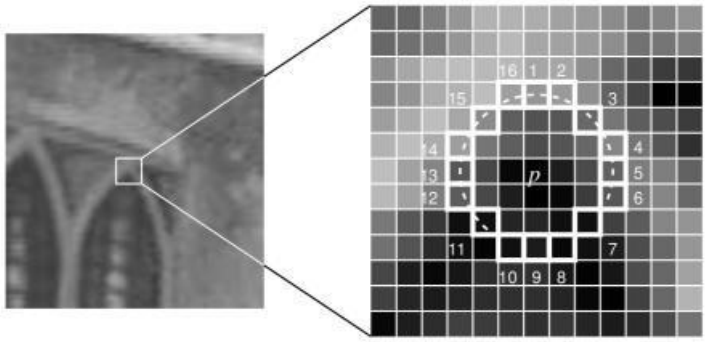
   Cp : Keputusan titik *p* sebagai sudut atau bukan sudut. Nilai 1 menunjukkan bahwa titik *p* merupakan suatu sudut, dan nilai 0 menunjukkan bahwa titip *p* bukan suatu sudut.

   $I_n$ : Nilai intensitas pixel ke-n

   $I_p$ : Nilai intensitas titik *p*

   $t$ : Batas ambang nilai intensitas yang ditoleransi (Threeshold)
4. Representasi hasil proses perbandingan intensitas pixels dalam bentuk vektor. Titk *p* merupakan sebuah sudut karena terdapat 4 titik yang memenuhi syarat.
5. Ulangi proses sampai seluruh titik pada citra sudah dibandingkan intensitasnya.

Alur dari *FAST Corner Detection* dalam menentukan titik sudut pada suatu citra dapat dilihat pada gambar 2.13 berikut.




**Gambar 2. 13 Alur algoritma FAST**

## 2.12 Pengujian Black Box

Pengujian *black box* fokus pada keperluan penelusuran kesalahan fungsional dari *software.* Ujicoba *black box* berusaha untuk menentukan kesalahan dalam beberapa kategori, diantaranya :

a. Fungsi-fungsi yang salah atau hilang
b. Kesalahan *interface*
c. Kesalahan dalam struktur data atau akses *database* eksternal.
d. Kesalahan performa.
e. Kesalahan inisialisasi dan terminasi.

Teknik pengujian *black box* terdiri dari 10 jenis diantaranya *Equivalence Partitioning, Boundary Valu Analysis/Limit Testing, Comparison Testing, Sample Testing, Robustness Testing, Behavior Testing, Requirement Testing, Performance Testing, Endurance Testing, Cause-Effect Relationship Testing. Salah satunya yang akan dibahas adalah Equivalence partitioning.*

*Equivalence partioning* merupakan metode ujicoba *black box* yang membagi domain *input* dari program menjadi beberapa kelas data dari kasus ujicoba yang dihasilkan. Kasus uji penanganan *single* yang ideal menemukan sejumlah kesalahan (misalnya: kesalahan pemrosesan dari seluruh data karakter)

yang merupakan syarat lain dari suatu kasus yang dieksekusi sebelum kesalahan umum diamati.

## 2.13 Pengujian Kuisioner

Kuesioner adalah sebuah daftar pernyataan yang harus diisi oleh orang yang akan dievaluasi (responden). Metode yang digunakan dalam kuesioner pada penelitian ini adalah skala Likert. Dalam skala likert, responden diminta untuk membaca dengan seksama setiap pernyataan yang disajikan, kemudian ia diminta untuk menilai pernyataan-pernyataan tersebut.

Derajat penilaian responden terhadap suatu pernyataan terbagi dalam 5 kategori yang tersusun secara bertingkat, mulai dari Sangat Tidak Setuju (STS), Tidak Setuju (TS), Ragu-Ragu (R), Setuju (S), dan Sangat Setuju (SS). Atau dapat pula sebaliknya. Pernyataan tiap kuesioner dibuat berdasarkan aspek-aspek yang diteliti. Bobot pemberian skor yang digunakan dapat dilihat pada Tabel 2.2.

**Tabel 2.2 Panduan Pemberian Skor**

| **Jenis Pertanyaan** | **Bobot Pendapat** | | | | |
|---|---|---|---|---|---|
| | **SS** | **S** | **R** | **TS** | **STS** |
| Positif | 5 | 4 | 3 | 2 | 1 |
| Negatif | 1 | 2 | 3 | 4 | 5 |

Skor yang telah dihitung pada setiap pernyataan kemudian dikalikan dengan masing-masing bobot tersebut sesuai dengan skenario kuesioner yang telah dibuat. Setelah itu, totalkan seluruh bobot jawaban tersebut kemudian bagi dengan total responden yang nantinya menjadi nilai rata-rata. Nilai rata-rata inilah yang diambil sebagai acuan sikap dimana jika nilai rata-rata kurang dari 3, maka dapat diartikan responden bersikap negatif dan jika nilai rata-rata lebih dari sama dengan 3, maka dapat diartikan responden bersikap positif terhadap

tujuan yang ingin dicapai. Untuk lebih jelasnya dapat dilihat pada rumus dibawah ini [11].

$$\overline{X} = \frac{\sum Total}{n}$$

Dimana,

$\overline{X} \geq 3$ *bersikap positif*

$\overline{X} < 3$ *bersikap negatif*

Keterangan :

$\overline{X}$ = nilai rata-rata

$\sum Total$ = jumlah seluruh nilai setelah dikalikan dengan bobot

N = total responden