

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Shalat**

Shalat merupakan salah satu kewajiban bagi kaum muslimin yang sudah mukallaf. Shalat merupakan rukun Islam kedua setelah syahadat. Islam didirikan atas lima sendi (tiang) salah satunya adalah shalat, sehingga barang siapa mendirikan shalat, maka ia mendirikan agama (Islam), dan barang siapa meninggalkan shalat, maka ia meruntuhkan agama (Islam).

Shalat harus didirikan dalam satu hari satu malam sebanyak lima kali, berjumlah 17 rakaat. Shalat tersebut merupakan wajib yang harus dilaksanakan tanpa kecuali bagi muslim mukallaf baik sedang sehat maupun sakit. Selain shalat wajib ada juga shalat – shalat sunah. Untuk membatasi bahasan penulisan dalam permasalahan ini, maka penulis hanya membahas tentang shalat wajib kaitannya dengan kehidupan sehari – hari.

##### **2.1.1 Pengertian Sholat**

Secara *etimologi* shalat berarti do'a dan secara *terminologi* atau istilah, para ahli fiqih mengartikan secara lahir dan hakiki. Secara lahiriah shalat berarti beberapa ucapan dan perbuatan yang dimulai dengan takbir dan diakhiri dengan salam, yang dengannya kita beribadah kepada Allah menurut syarat – syarat yang telah ditentukan.

Adapun secara hakikinya ialah berhadapan hati (jiwa) kepada Allah, secara yang mendatangkan takut kepada-Nya serta menumbuhkan di dalam jiwa rasa kebesaran-Nya dan kesempurnaan kekuasaan-Nya.

Dari beberapa pengertian di atas dapat disimpulkan bahwa shalat adalah merupakan ibadah kepada Tuhan, berupa perkataan dengan perbuatan yang diawali dengan takbir dan diakhiri dengan salam menurut syarat dan rukun yang telah ditentukan. Juga shalat merupakan penyerahan diri (lahir dan batin) kepada Allah dalam rangka ibadah dan memohon ridho-Nya.

### **2.1.2 Tata Cara Sholat**

Sebelum kita melaksanakan ibadah shalat baik wajib maupun sunnah, ada beberapa syarat yang harus dipenuhi terlebih dahulu diantaranya :

1. Islam.
2. Sudah balig.
3. Berakal.
4. Suci dari hadas kecil dan besar.
5. Suci dari najis seluruh badan, pakaian dan tempat shalat.
6. Menutup aurat (aurat lelaki antara pusar sampai lutut, sedangkan aurat wanita seluruh anggota badan, kecuali muka dan kedua telapak tangan).
7. Menghadap kiblat.
8. Sudah masuk waktu shalat yang akan dikerjakan.
9. Mengetahui kefarduan shalat.

Tata cara pelaksanaan ibadah sholat diantaranya sebagai berikut :

1. Berdiri tegak dan lurus menghadap kiblat.
2. Membaca niat sholat.
3. Mengucap takbiratul ihram (Allahu Akbar) sambil mengangkat kedua tangan setentang bahu atau telinga dan bersedekap.
4. Membaca doa iftitah.
5. Membaca Surat Al-Fatihah.
6. Membaca Surat Pendek.
7. Melakukan ruku'.
8. Melakukan i'tidal (bangkit dari ruku').
9. Melakukan sujud
10. Duduk diantara dua sujud.
11. Tasyahud (awal dan akhir).
12. Salam.

## **2.2 Object Oriented Analisis and Design**

OOAD (*Object Oriented Analisis and Design*) adalah suatu pendekatan rekayasa perangkat lunak dari sebuah sistem yang terdiri dari sekelompok objek yang saling berinteraksi, dan setiap objek itu mewakili beberapa entitas. Yang ditandai dengan adanya sebuah kelas, elemen data dan perilaku dari objek tersebut.

### 2.2.1 Pengertian OOAD

OOAD adalah metode analisis yang memeriksa kebutuhan dari sudut pandang kelas-kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem. OOAD merupakan 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.

### 2.2.2 Unified Modeling Language

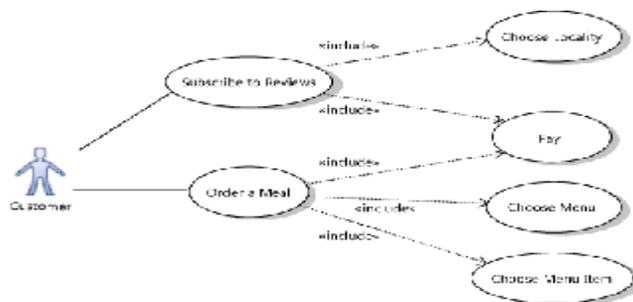
UML(*Unified Modeling Language*) adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artefak dari proses analisis dan desain berorientasi objek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu sistem. UML menyediakan 10 macam diagram untuk memodelkan suatu sistem atau aplikasi berorientasi objek diantaranya :

- Use Case Diagram : untuk memodelkan proses bisnis.
- Conceptual Diagram : untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
- Sequence Diagram : untuk memodelkan pengiriman pesan (*message*) antar objek.
- Collaboration Diagram : untuk memodelkan interaksi antar objek.
- State Diagram : untuk memodelkan perilaku objek di dalam sistem.

- Activity Diagram : untuk memodelkan perilaku Use Case dan objek di dalam sistem.
- Class Diagram : untuk memodelkan struktur kelas.
- Object Diagram : untuk memodelkan struktur objek.
- Component Diagram : untuk memodelkan komponen objek.
- Deployment Diagram : untuk memodelkan distribusi aplikasi.

### 2.2.3 Use Case Diagram

*Use case* adalah abstraksi dari interaksi antara sistem dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.



Gambar 2.1 Contoh use case diagram

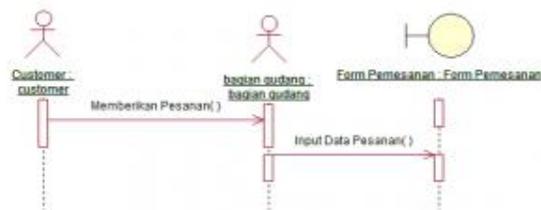
Diagram *Use Case* berguna dalam tiga hal :

1. Menjelaskan fasilitas yang ada (*requirement*).
2. Komunikasi dengan klien.
3. Membuat tes dari kasus-kasus secara umum.

Tabel 2.1 Simbol - simbol Use Case

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
3		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
5		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

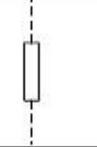
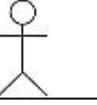
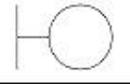
#### 2.2.4 Sequence Diagram



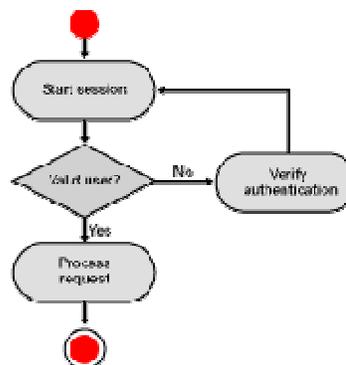
Gambar 2.2 Contoh sequence diagram

*Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.

Tabel 2.2 Penjelasan simbol sequence diagram

NO	GAMBAR	NAMA	KETERANGAN
1		LifeLine	Objek entity, antarmuka yang saling berinteraksi.
		Actor	Digunakan untuk menggambarkan user / pengguna.
2		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		Boundary	Digunakan untuk menggambarkan sebuah form.
4		Control Class	Digunakan untuk menghubungkan boundary dengan tabel.
5		Entity Clas	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

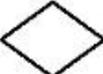
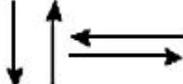
### 2.2.5 Activity Diagram



Gambar 2.3 Contoh activity diagram

*Activity diagram* menyediakan analisis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. *Activity diagram* dapat digunakan untuk alur kerja model, *use case individual*, atau logika keputusan yang terkandung dalam metode individual. *Activity diagram* juga menyediakan pendekatan untuk proses pemodelan paralel.

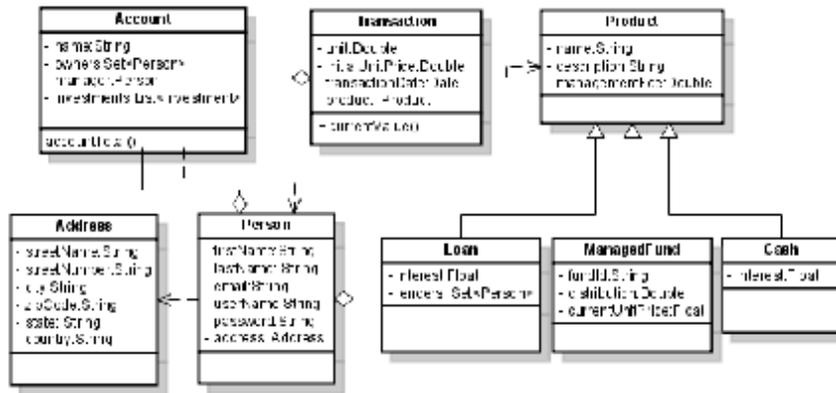
Tabel 2.3 Penjelasan tentang activity diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas akan saling berinteraksi satu sama lain
2		Initial Node	Bagaimana objek dibentuk atau diawali.
3		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
4		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
5		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

### 2.2.6 Class Diagram

*Class* adalah dekripsi kelompok objek-objek dengan properti, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class* diagram dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem

biasanya mempunyai beberapa class diagram. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

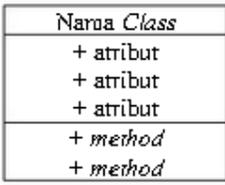


Gambar 2.4 Contoh class diagram

Kelas merupakan sebuah objek dimana dalam kelas diagram, kelas diwakilkan dengan kotak yang berisi tiga bagian diantaranya yaitu :

- Nama Kelas
- Atribut Kelas : merupakan informasi mengenai objek atau kelas yang disimpan atau setidaknya dipertahankan sementara.
- Method : merupakan hal – hal atau operasi yang dapat dilakukan oleh suatu objek atau kelas
- Visibilitas, merupakan *visibility* dari tiap atribut atau operator pada suatu kelas.

Tabel 2.4 Penjelasan tentang class diagram

Nama Komponen	Keterangan	Simbol
<i>Class</i>	<i>Class</i> adalah blok - blok pembangun pada perorograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .	
<i>Association</i>	Sebuah asosiasi merupakan sebuah relationship paling ururu antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menarupikan hukuru-hukuru multiplisitas pada sebuah <i>relationship</i> . (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i> ).	
<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisísolid.	
<i>Dependency</i>	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Ururunya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.	

### 2.3 Computer Vision

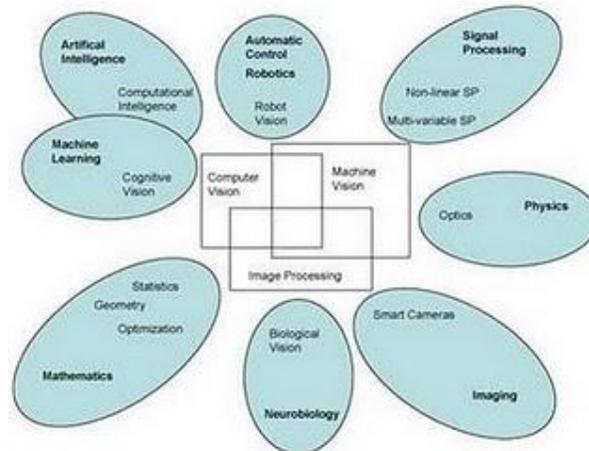
*Computer Vision* sering didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati/diobservasi. Cabang ilmu ini bersama *Artificial Intelligence* akan mampu menghasilkan *Visual Intelligence System*.

### 2.3.1 Pengertian Computer Vision

*Computer Vision* adalah ilmu dan teknologi mesin yang melihat, di mana mesin mampu mengekstrak informasi dari gambar yang diperlukan untuk menyelesaikan tugas tertentu. *Computer Vision* adalah cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati. Cabang ilmu ini bersama *Artificial Intelligence* akan mampu menghasilkan *Visual Intelligence System*. Perbedaannya adalah *Computer Vision* lebih mempelajari bagaimana komputer dapat mengenali objek yang diamati. Namun komputer grafik lebih ke arah pemanipulasian gambar secara digital. Bentuk sederhana dari grafik komputer adalah grafik komputer 2D yang kemudian berkembang menjadi grafik komputer 3D, pemrosesan citra, dan pengenalan pola.

Computer Vision adalah kombinasi antara :

- Pengolahan Citra (*Image Processing*), bidang yang berhubungan dengan proses transformasi citra/gambar (*image*). Proses ini bertujuan untuk mendapatkan kualitas citra yang lebih baik.
- Pengenalan Pola (*Pattern Recognition*), bidang ini berhubungan dengan proses identifikasi objek pada citra atau interpretasi citra. Proses ini bertujuan untuk mengekstrak informasi/pesan yang disampaikan oleh gambar/citra.



Gambar 2.5 Kombinasi Image Processing dan Pattern Recognition dalam  
Computer Vision

Untuk menunjang tugas *Computer Vision*, terdapat beberapa fungsi pendukung ke dalam sistem ini, yaitu :

1. Proses penangkapan citra (*Image Acquisition*)
  - *Image Acquisition* pada manusia dimulai dengan mata, kemudian informasi visual diterjemahkan ke dalam suatu format yang kemudian dapat dimanipulasi oleh otak.
  - Senada dengan proses di atas, computer vision membutuhkan sebuah mata untuk menangkap sebuah sinyal visual.
  - Umumnya mata pada computer vision adalah sebuah kamera video.
  - Kamera menerjemahkan sebuah scene atau image.
  - Keluaran dari kamera adalah berupa sinyal analog, dimana frekuensi dan amplitudonya (frekuensi berhubungan dengan jumlah sinyal dalam satu detik, sedangkan amplitudo berkaitan dengan tingginya

sinyal listrik yang dihasilkan) merepresentasikan detail ketajaman (*brightness*) pada scene.

- Kamera mengamati sebuah kejadian pada satu jalur dalam satu waktu, memindainya dan membaginya menjadi ratusan garis horizontal yang sama.
- Tiap-tiap garis membuat sebuah sinyal analog yang amplitudonya menjelaskan perubahan brightness sepanjang garis sinyal tersebut.
- Kemudian sinyal listrik ini diubah menjadi bilangan biner yang akan digunakan oleh komputer untuk pemrosesan.
- Karena komputer tidak bekerja dengan sinyal analog, maka sebuah *analog-to-digital converter* (ADC), dibutuhkan untuk memproses semua sinyal tersebut oleh komputer.
- ADC ini akan mengubah sinyal analog yang direpresentasikan dalam bentuk informasi sinyal tunggal ke dalam sebuah aliran (*stream*) sejumlah bilangan biner.
- Bilangan biner ini kemudian disimpan di dalam memori dan akan menjadi data *raw* yang akan diproses.

## 2. Proses pengolahan citra (*Image Processing*)

- Tahapan berikutnya *computer vision* akan melibatkan sejumlah manipulasi utama (*initial manipulation*) dari *data binary* tersebut.
- *Image processing* membantu peningkatan dan perbaikan kualitas gambar, sehingga dapat dianalisa dan diolah lebih jauh secara lebih efisien.

- Image processing akan meningkatkan perbandingan sinyal terhadap *noise*.
- Sinyal-sinyal tersebut adalah informasi yang akan merepresentasikan objek yang ada dalam gambar.
- Sedangkan *noise* adalah segala bentuk interferensi yang terjadi pada sebuah objek.

### 3. Analisa data citra (*Image Analysis*)

- *Image analysis* akan mengeksplorasi *scene* ke dalam bentuk karakteristik utama dari objek melalui suatu proses investigasi.
- Sebuah program komputer akan mulai melihat melalui bilangan biner yang merepresentasikan informasi visual untuk mengidentifikasi fitur-fitur spesifik dan karakteristiknya.
- Lebih khusus lagi program image analysis digunakan untuk mencari tepi dan batas-batasan objek dalam image.
- Sebuah tepian (*edge*) terbentuk antara objek dan latar belakangnya atau antara dua objek yang spesifik.
- Tepi ini akan terdeteksi sebagai akibat dari perbedaan level brightness pada sisi yang berbeda dengan salah satu batasnya.

### 4. Proses pemahaman data citra (*Image Understanding*)

- Ini adalah langkah terakhir dalam proses *computer vision*, yang mana spesifik objek dan hubungannya diidentifikasi.
- Pada bagian ini akan melibatkan kajian tentang teknik-teknik *artificial intelligent*.

- *Understanding* berkaitan dengan *template matching* yang ada dalam *sebuah scene*.
- Metode ini menggunakan program pencarian (*search program*) dan teknik penyesuaian pola (*pattern matching techniques*).

### 2.3.2 Kamera

Kamera merupakan seperangkat perlengkapan yang memiliki fungsi untuk mengabadikan suatu objek menjadi sebuah gambar yang merupakan hasil proyeksi pada sistem lensa. Kamera disebut juga dengan *kamera obscura*. Kata ini berasal dari bahasa latin yang artinya “ruang gelap”. *Kamera obscura* adalah sebuah alat yang terdiri dari ruang gelap atau kotak, yang bisa memantulkan cahaya dengan menggunakan dua buah lensa konveks, setelah itu menempatkan gambar objek eksternal itu pada sebuah kertas/film. Penempatan film tersebut ada pada pusat fokus dari lensa. Sebuah kamera minimal terdiri atas:

- Kotak yang kedap cahaya (badan kamera)
- Sistem lensa
- Pemantik potret (*shutter*)
- Pemutar film

Salah satu fungsi yang menunjang tugas *computer vision* adalah proses penangkapan citra (*image acquisition*), dimana kamera berperan seperti layaknya mata manusia dalam melihat suatu objek.

## 2.4 Object Recognition

*Object Recognition* adalah kemampuan untuk merasakan sifat fisik objek (seperti bentuk, warna dan tekstur) dan menerapkan atribut semantik ke objek, yang meliputi pemahaman penggunaannya, pengalaman sebelumnya dengan objek dan bagaimana kaitannya dengan orang lain.

### 2.4.1 Pengertian Object Recognition

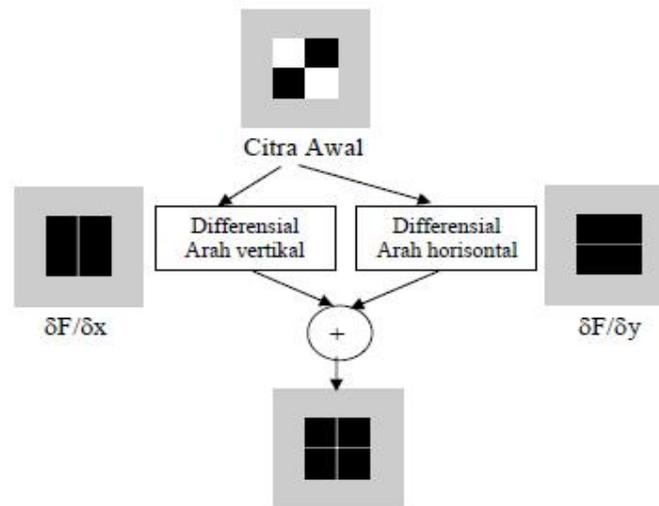
Pengertian *Object Recognition* adalah kemampuan (*computer vision*) dalam menemukan dan mengidentifikasi benda-benda di dalam urutan gambar atau video. Manusia mengenali banyak objek dalam gambar dengan sedikit usaha, meskipun fakta bahwa citra objek mungkin agak berbeda di berbagai titik pandang, dalam berbagai ukuran dan skala atau bahkan ketika mereka diterjemahkan atau diputar. Benda bahkan dapat dikenali ketika benda sebagian terhalang dari pandangan.

### 2.4.2 Edge Detection

*Edge* adalah beberapa bagian dari citra di mana intensitas kecerahan berubah secara drastis. Sedangkan deteksi tepi (*edge detection*) pada suatu citra adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda. Atau secara umum, *edge detection* dalam citra dinyatakan sebagai titik yang nilai warnanya berbeda cukup besar dengan titik yang ada disebelahnya.

Tujuan dari edge detection ini adalah:

- Untuk menandai bagian yang menjadi detail citra.
- Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra.
- Mengubah citra 2D menjadi bentuk kurva. Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.



Gambar 2.6 Proses Deteksi Tepi Citra

### 2.4.3 Grayscale Matching

Dalam komputasi, suatu citra digital *grayscale* atau *greyscale* adalah suatu citra dimana nilai dari setiap pixel merupakan sample tunggal. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra "hitam-putih", dimana pada konteks

komputer, citra hitam putih hanya terdiri atas dua warna saja yaitu "hitam" dan "putih" saja. Citra *grayscale* disimpan dalam format delapan bit untuk setiap sample pixel, yang memungkinkan sebanyak 256 intensitas. Nilai intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih. Citra *grayscale* digunakan untuk menyederhanakan model citra. Citra berwarna terdiri dari tiga layer matrik yaitu R-layer, G-layer dan B-layer. Sehingga untuk melakukan proses-proses selanjutnya tetap diperhatikan tiga layer tersebut. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, berarti dilakukan tiga perhitungan yang sama.

Konsep tersebut diubah dengan mengubah tiga layer menjadi satu layer matrik *grayscale* dan hasilnya adalah citra *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing  $r$ ,  $g$  dan  $b$  menjadi citra *grayscale* dengan nilai  $s$ , maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai  $r$ ,  $g$  dan  $b$  sehingga dapat dituliskan pada persamaan berikut :

$$s = \frac{r + g + b}{3}$$

Gambar 2.7 merupakan contoh konversi citra berwarna menjadi citra *grayscale* dengan menggunakan persamaan di atas.



Gambar 2.7 Hasil konversi citra berwarna menjadi citra grayscale.

#### 2.4.4 Template Matching

*Template matching* adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi sampel (*template*). Teknik ini banyak digunakan dalam bidang industri sebagai bagian dari *quality control*. Prinsip metode ini adalah membandingkan antara citra *streaming* yang akan dikenali dengan citra sampel (*template*).

Citra objek yang akan dikenali mempunyai tingkat kemiripan sendiri terhadap masing-masing citra sampel (*template*). Pengenalan dilakukan dengan melihat nilai tingkat kemiripan tertinggi dan nilai batas ambang pengenalan dari citra objek tersebut. Bila nilai tingkat kemiripan berada di bawah nilai batas ambang maka citra objek tersebut dikategorikan sebagai objek tidak dikenal. Pada prinsipnya metode *template matching* memiliki karakteristik antara lain:

1. Relatif mudah untuk diaplikasikan dalam teknik pengolahan citra digital.
2. Hasilnya relatif sangat akurat karena mendeteksi kesalahan hingga ukuran *pixel*.

3. Metode ini cukup rentan terhadap perbedaan orientasi antara citra sampel (*template*) dengan citra yang akan diidentifikasi, yang meliputi: ukuran, posisi dan kualitas citra.
4. Untuk mendapatkan hasil yang maksimal maka metode ini sangat tergantung pada teknik pengolahan citra digital yang lain seperti *enhancement*, *color filtering*, dan lain-lain. (Wardhana, 2008).

## 2.5 Augmented Reality

Realitas tertambah atau kadang dikenal dengan *augmented reality* 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. Tidak seperti realitas maya yang sepenuhnya menggantikan kenyataan, realitas tertambah sekedar menambahkan atau melengkapi kenyataan. Benda-benda maya menampilkan informasi yang tidak dapat diterima oleh pengguna dengan inderanya sendiri. Hal ini membuat realitas tertambah sesuai sebagai alat untuk membantu persepsi dan interaksi penggunaanya dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata. Realitas tertambah dapat diaplikasikan untuk semua indera, termasuk pendengaran, sentuhan, dan penciuman. Selain digunakan dalam bidang-bidang seperti kesehatan, militer, industri manufaktur, realitas tertambah juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam.

### 2.5.1 Pengertian Augmented Reality

Ronald T. Azuma (1997) mendefinisikan *augmented reality* sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antarbenda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat *input* tertentu, dan integrasi yang baik memerlukan penjejukan yang efektif.

Selain menambahkan benda maya dalam lingkungan nyata, realitas ditambah juga berpotensi menghilangkan benda-benda yang sudah ada. Menambah sebuah lapisan gambar maya dimungkinkan untuk menghilangkan atau menyembunyikan lingkungan nyata dari pandangan pengguna.

### 2.5.2 Arsitektur Augmented Reality

Arsitektur teknologi *augmented reality* dapat dilihat pada gambar di bawah ini:



Gambar 2.8 Arsitektur teknologi augmented reality

- Input  
Input dapat berupa apa saja, contoh marker, gambar 2D, gambar 3D, sensor wifi, sensor gerakan, GPS, dan sensor-sensor yang lain.
- Kamera  
Kamera disini sebagai perantara untuk input yang berupa gambar, baik itu marker, gambar 2D maupun 3D.
- Processor  
Processor dibutuhkan untuk memproses input yang masuk dan kemudian memberikannya ke tahapan output.
- Output  
Dapat berupa HMD, monitor, seperti monitor TV, LCD, monitor ponsel, dan lain-lain

## 2.6 Blender



Gambar 2.9 Logo Blender

Blender adalah program 3D dan animasi yang bersifat *open source*, bebas untuk dikembangkan oleh penggunanya dan dapat didistribusikan kembali dan bersifat Legal. Blender memiliki *video compositor* dan *intergrated game engine*.

Karya yang dihasilkan tidak ada sifat royalti kepada developer, dan dapat dipublikasikan baik free maupun untuk dikomersilkan. Blender merupakan salah satu program *Modeling* 3D dan *Animation*, tapi Blender mempunyai kelebihan sendiri dibandingkan program *modeling* 3D lainnya. Kelebihan yang dimiliki Blender adalah dapat membuat game tanpa menggunakan program tambahan lainnya, karena Blender sudah memiliki Game Engine sendiri dan menggunakan Python sebagai bahasa pemrograman yang lebih mudah dibandingkan dengan menggunakan C++, C, dan lain-lain. Blender menggunakan OpenGL sebagai render grafiknya yang dapat digunakan pada berbagai macam sistem operasi seperti Windows, Linux dan Mac OS X.

## 2.7 Vuforia



Gambar 2.10 Logo Vuforia

Vuforia adalah Augmented Reality Software Development Kit (SDK) untuk perangkat mobile yang memungkinkan pembuatan aplikasi AR. Vuforia SDK juga tersedia untuk digabungkan dengan unity yaitu bernama Vuforia AR Extension for Unity. Vuforia merupakan SDK yang disediakan oleh Qualcomm untuk membantu para developer membuat aplikasi-aplikasi *Augmented Reality* (AR) di mobile phones (iOS, Android).

## 2.8 XML

XML (*eXtensible Markup Language*) merupakan bahasa web turunan dari SGML (*Standart Generalized Markup Language*) yang ada sebelumnya. XML hampir sama dengan HTML, dimana keduanya sama-sama turunan dari SGML. Teknologi XML dikembangkan mulai tahun 1966 dan mendapatkan pengakuan dari *World Wide Web Consortium* (W3C) pada bulan Februari 1998. Sedangkan SGML sendiri telah dikembangkan pada awal tahun 1980-an. Pada saat HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting SGML dan dengan berpedoman pada pengembangan HTML menghasilkan bahasa markup yang tidak kalah hebatnya dengan SGML.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa *meta-markup* yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa *markup* adalah mekanisme untuk mengenal secara terstruktur di dokumen. XML adalah suatu aplikasi profil dari SGML. Seperti yang didefinisikan dari ISO 8879, SGML adalah cara standart dan vendor-independent.

XML tidak mempunyai definisi secara tepat karena ada yang berpendapat bahwa XML bukanlah suatu bahasa pemrograman, melainkan XML merupakan sintaks yang digunakan untuk menjelaskan bahasa *markup* lain (Dournaee,2002), sehingga dinamakan *meta-language*. Meskipun demikian pendapat yang XML bukan merupakan bahasa *markup*, didasarkan bahwa XML merupakan bahasa

*markup* terpisah untuk tujuan terpisah. Selain itu XML bukanlah solusi semua hal untuk tujuan semua user. Sedangkan peran dari markup itu sendiri berupa:

1. *Markup* dapat menambah maksud arti (*semantic*) suatu data.
2. Dapat memisahkan data.
3. Dapat mendefinisikan peran data.
4. Dapat mendefinisikan batasan data.
5. Dapat menfenisikan keterhubungan data.

XML merupakan sebuah himpunan bagian (subset) dari SGML yang bertujuan agar SGML secara generik dapat melayani, menerima, dan memproses di dalam web dengan cara seperti yang dimungkinkan HTML saat ini. XML didesain untuk kemudahan implementasi dan interoperabilitas dengan SGML maupun HTML. XML adalah bahasa markup yang dirancang untuk penyampaian informasi melalui World Wide Web (www) atau sering disebut web saja. (W3C,2000)

```
<?xml version="1.0" ?>
- <product barcode="2394287410">
  <manufacturer>Verbatim</manufacturer>
  <name>DataLife MF 2HD</name>
  <quantity>10</quantity>
  <size>3.5"</size>
  <color>black</color>
  <description>floppy disks</description>
</product>
```

Gambar 2.11 Contoh format penulisan XML

## 2.9 C Sharp (C#)

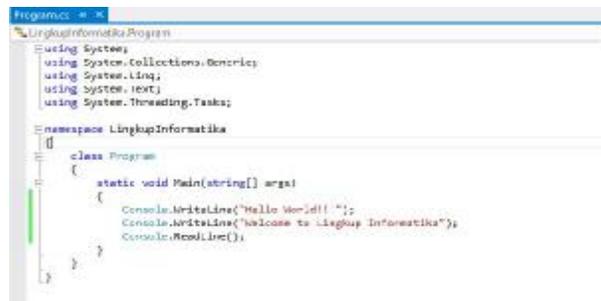
C# (*C sharp*) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# Language Specification, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar.

Menurut standar ECMA-334 C# Language Specification ini, nama bahasa C# ini terdiri dari sebuah huruf latin yaitu C (U+0043) dan # (U+0023). Tanda kres (#) ini sama dengan tanda kres pada music. Kesederhanaan yang dimiliki oleh bahasa pemrograman ini antara lain :

1. Pointer telah dihilangkan dari C#.
2. Operasi-operasi yang “tidak aman” seperti misalnya akses memori langsung sudah tidak diizinkan.
3. Tidak ada lagi penggunaan operator “::” atau “->”.
4. Karena C# masih ada di lingkungan .NET, maka C# memiliki turunan manajemen memori otomatis dan *Garbage Collection*.

5. Dukungan tipe data primitive yang lebih banyak seperti misalnya integer, floats, dan sebagainya.
6. Nilai integer 0 dan 1 tidak diterima lagi sebagai alternative nilai boolean. Nilai boolean yang murni yang dipakai adalah murni true and false.

Karena C# merupakan pengaplikasian dari bahasa program pendahulunya, C# didasarkan pada trend yang berkembang saat ini dan sangat ampuh serta mudah digunakan untuk membangun aplikasi yang bersifat interoperable, skalabilitas dan cepat.



```
Program.cs - C#
LingkupInformatika.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LingkupInformatika
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            Console.WriteLine("Welcome to Lingkup Informatika");
            Console.ReadKey();
        }
    }
}
```

Gambar 2.12 Contoh format penulisan C#

## 2.10 Bluestack



Gambar 2.13 Logo Bluestack

Bluestacks merupakan salah satu emulator Android yang dapat Anda install di komputer yang menggunakan sistem operasi Windows Vista, Windows XP, Windows 7, dan Windows 8. Menariknya, software ini dapat Anda peroleh secara gratis di situs BlueStack. BlueStack versi terbaru masih menggunakan sistem operasi Android versi Gingerbread. Sampai saat ini, BlueStack merupakan emulator Android terbaik jika dibandingkan dengan emulator lainnya.

Perkembangan bluestacks ini sudah lama karena sudah sejak 2010 aplikasi bluestack di kembangkan. Software yang bernama bluestacks di buat oleh seorang dari India yang bernama Rosen Sharma.

Dengan anda menggunakan bluestacks di komputer anda otomatis anda bisa memasang dan mengoperasikan aplikasi yang berbasis android dengan sistem operasi windows maupun Mac. Bluestack memiliki beberapa kelebihan dan kekurangan diantaranya :

#### Kelebihan

1. Bluestacks tidak memerlukan aplikasi lainnya untuk berjalan mulus. Tidak seperti Genymotion yang mensyaratkan adanya Oracle Virtual Box atau Eclipse.
2. Bluestacks mendukung instalasi dari penyedia aplikasi Android, seperti Play Store, Amazon Store, dan 1Ready. Hal ini sangat membantu sekali dibandingkan harus susah-susah mencari berkas APK secara mandiri.
3. Tampilan Bluestacks sangat sederhana dan mudah ditelusuri.

### Kekurangan

1. Bluestacks memerlukan minimal RAM 4 GB untuk berjalan mulus.
2. Bluestacks lebih cepat menghabiskan baterai.
3. Tidak semua aplikasi/permainan Android bisa berjalan di Bluestacks.

Umumnya yang bisa berjalan permainan yang mengandalkan klik. Jika sudah mengandalkan fitur Android seperti kemiringan, getaran, dan sebagainya, Bluestacks tidak bisa mengatasinya.

## **2.11 Android**

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

### **2.11.1 Pengenalan Android**

Android adalah sistem operasi yang digunakan di smartphone dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS. Android tidak terikat ke satu merek handphone saja, beberapa

vendor terkenal yang sudah memakai Android antara lain Samsung, Sony Ericsson, HTC, Nexus, Motorola, dan lain-lain.

Android pertama kali dikembangkan oleh perusahaan bernama Android Inc., dan pada tahun 2005 di akuisisi oleh raksasa Internet Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap rilisnya diberi kode nama berdasarkan nama hidangan makanan.

Keunggulan utama Android adalah gratis dan *open source*, yang membuat *smartphone* Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone meski fitur yang ditawarkan Android lebih baik.

Beberapa fitur utama dari Android antara lain WiFi hotspot, multi-touch, multitasking, GPS, accelerometers, support java, mendukung banyak jaringan (GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE & WiMAX) serta juga kemampuan dasar handphone pada umumnya. Versi Android yang beredar saat ini diantaranya :

- **Gingerbread (2.3)**



Gambar 2.14 Logo Gingerbread

Merupakan versi penerus dari versi android sebelumnya yaitu Froyo, yang saat ini sudah tidak dikembangkan lagi oleh pihak google. Perubahan utama di versi 2.3 ini termasuk update UI, peningkatan fitur *soft keyboard & copy/paste*, *power management*, dan *support Near Field Communication*.

- **Honeycomb (3.0, 3.1 dan 3.2)**



Gambar 2.15 Logo Honeycomb

Merupakan versi Android yang ditujukan untuk *gadget / device* dengan layar besar seperti Tablet PC. Fitur baru Honeycomb yaitu dukungan terhadap processor multicore dan grafis dengan *hardware acceleration*.

Google memutuskan untuk menutup sementara akses ke source code Honeycomb, hal ini dilakukan untuk mencegah vendor pembuat handphone menginstal Honeycomb pada smartphone. Karena di versi sebelumnya banyak perusahaan menginstal Android ke dalam tablet PC, yang menyebabkan pengguna tidak puas dan akhirnya membuat citra Android tidak bagus.

- **Ice Cream Sandwich (4.0)**



Gambar 2.16 Logo Ice Cream Sandwich

Android Ice Cream Sandwich diumumkan pada 10 Mei 2011 di ajang Google I/O Developer Conference (San Francisco) dan resmi dirilis pada tanggal 19 Oktober 2011 di Hongkong. “Android Ice Cream Sandwich” dapat digunakan baik di smartphone ataupun tablet. Fitur utama Android ICS 4.0 ialah Face Unlock , Android Beam (NFC), perubahan major *User Interface*, dan ukuran layar standar (*native screen*) beresolusi 720p (*high definition*).

- **Android Jelly Bean (4.1, 4.2 dan 4.3)**



Gambar 2.17 Logo Jelly Bean

Android Jelly Bean lebih menfokuskan fiturnya ke peningkatan *User Interface* yang lebih lancar dan responsif. Di versi ini juga menandai hadirnya fitur Google Now yang memberikan saran dan rekomendasi berdasarkan data-data yang tersimpan (kontak, kalender, lokasi, dan lain-lain) di telepon genggam.

- **Android KitKat (4.4)**



Gambar 2.18 Logo Kitkat

Android KitKat menfokuskan pada manajemen *memory* sehingga *device low-end* yang memiliki memory rendah tetap dapat berjalan lancar dan minim *lagg*. Selain itu hadir beberapa fitur lain seperti kemampuan untuk menggunakan semua tampilan layar (*immersive mode*), dukungan *mobile printing*, penambahan dukungan sensor baru, dan lain-lain.

- **Android Lollipop (5.0)**



Gambar 2.19 Logo Lollipop

Android Lollipop merupakan versi android yang terbaru yang diluncurkan oleh google di akhir tahun 2014 lalu. Android Lollipop memberikan pengalaman komputasi yang lebih cepat dan lebih kuat. Android sekarang berjalan secara eksklusif pada runtime ART baru, dibangun dari bawah ke atas untuk mendukung campuran *Ahead-of-Time* (AOT), *Just-In-Time* (JIT), dan kode yang diinterpretasikan. Ini didukung pada ARM, x86, dan arsitektur MIPS dan kompatibel pada 64-bit.

### 2.11.2 Arsitektur Android

Android dibangun dengan menggunakan asas *object oriented*, dimana elemen-elemen penyusun sistem operasinya berupa objek yang dapat kita gunakan kembali atau *reusable*. Agar bisa membuat aplikasi dengan baik, tentunya kita harus mengetahui arsitektur OS Android beserta elemen elemennya. Gambar dibawah merupakan skema pembagian elemen pada arsitektur Android. Secara garis besar arsitektur android terdiri dari empat layer komponen, yaitu:



Gambar 2.20 Skema Arsitektur Android

- **Layer Applications dan Widget**

Inilah layer pertama pada OS Android, biasa dinamakan layer Applications dan Widget. Layer ini merupakan layer yang berhubungan dengan aplikasi-aplikasi inti yang berjalan pada Android OS. Seperti klien email, program SMS, kalender, browser, peta, kontak, dan lain-lain. Semua aplikasi ini dibuat dengan menggunakan bahasa Java. Apabila kalian membuat aplikasi, maka aplikasi itu ada di layer ini.

- **Layer Applications Framework**

Applications Framework merupakan layer dimana para pembuat aplikasi menggunakan komponen-komponen yang ada di sini untuk membuat aplikasi mereka. Beberapa contoh komponen yang termasuk di dalam *Applications Framework* adalah sebagai berikut:

1. Views
2. Content Provider
3. Resource Manager
4. Notification Manager
5. Activity Manager

- **Layer Libraries**

*Libraries* merupakan *layer* tempat fitur-fitur android berada. Pada umumnya *libraries* diakses untuk menjalankan aplikasi. Beberapa *library* yang terdapat pada android diantaranya adalah *libraries Media* untuk memutar media

video atau audio, *libraries Graphic* untuk menjalankan tampilan, *libraries SQLite* untuk dukungan database, dan masih banyak *library* lainnya.

- **Android RunTime**

*Android RunTime* merupakan *layer* yang membuat aplikasi android bisa dijalankan. *Android RunTime* dibagi menjadi dua bagian yaitu:

- a) *Core Libraries* : berfungsi untuk menerjemahkan bahasa Java/C
- b) *Dalvik Virtual Machine* : sebuah mesin virtual berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi pada android secara efisien.

- **Linux Kernel**

*Linux Kernel* merupakan *layer* tempat keberadaan inti dari *operating system* android. Layer ini berisi *file-file system* yang mengatur *system processing*, *memory*, *resource*, *drivers*, dan sistem android lainnya. Inilah yang membuat *file system* pada Android mirip dengan *file system* pada sistem operasi berbasis Linux. *Kernel* yang digunakan adalah *kernel Linux* versi 2.6, dan versi 3.x pada Android versi 4.0 ke atas. *Kernel* ini berbasis *monolithic*.

## 2.12 Penelitian Sebelumnya

Adapun penelitian terdahulu yang berkaitan dengan *augmented reality* antara lain :



Gambar 2.21 Tampilan Utama pembelajaran doa agama islam pada anak.

- Ahmad (2015) dalam penelitiannya menerapkan teknologi *augmented reality* pada perancangan aplikasi pembelajaran doa agama islam pada anak. Menurut Ahmad (2015), proses pembelajaran dengan menerapkan teknologi augmented reality sangat efektif dalam proses pembelajaran karena materi yang selama ini diberikan kepada siswa berbentuk teks yang sifatnya baku dan formal, kemungkinan adanya siswa yang jenuh dalam melakukan proses pembelajaran tersebut.



Gambar 2.22 Tampilan Aplikasi Augmented Reality Katalog Rumah.

- Rifa'i, Muhammad (2014) dalam penelitiannya menerapkan teknologi *augmented reality* pada aplikasi katalog rumah berbasis android. Menurut Rifa'i, Penggunaan *augmented reality* saat ini telah melebar ke berbagai aspek dalam kehidupan kita dan diproyeksikan akan mengalami

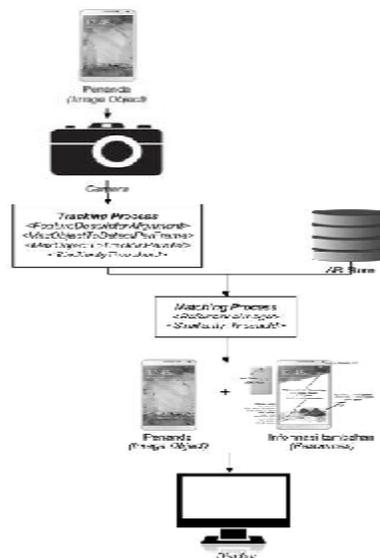
perkembangan yang sangat signifikan. Hal ini dikarenakan penggunaan AR sangat menarik dan memudahkan penggunaannya dalam mengerjakan sesuatu hal, salah satunya pada strategi pemasaran penjualan rumah kepada konsumen. Menurut teknologi *augmented reality* ini dapat membantu para pembeli untuk mengetahui dengan baik rumah yang akan dibeli, serta akan mempermudah *seller* rumah dalam mempromosikan produknya kepada konsumen.



Gambar 2.23 Penerapan AR-Book dalam proses pembelajaran.

- Safaat, Nazruddin (2014) dalam penelitiannya menerapkan teknologi *augmented reality* dalam aplikasi pembelajaran berbasis teknologi augmented reality pada smartphone (studi kasus: Materi Sistem Tata Surya Kelas IX). Di dalam penelitiannya, Safaat (2014) menggabungkan antara buku dengan teknologi *augmented reality* dan menciptakan media baru yang bernama buku berbasis *augmented reality* (AR-Book). Agar dapat bekerja dengan sempurna AR-Book secara garis besar memiliki dua komponen utama, yaitu buku yang dilengkapi dengan *marker*, dan yang

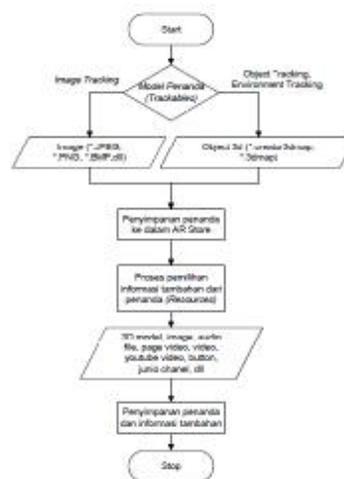
kedua yaitu peralatan untuk menangkap *marker* dan menampilkan hasilnya. Alat tersebut dapat berupa *handheld* (HHD), *head mounted display* (HMD), *virtual retinal display* (VRD), atau bahkan tampilan berbasis layar seperti layar *monitor*.



Gambar 2.24 Alur kerja Aplikasi Augmented Reality

- Setiawan, Yudi (2014) dalam penelitiannya yang berjudul pemodelan pengenalan penanda *augmented reality* dengan *metaio creator*. Setiawan (2014) memanfaatkan *metaio creator* dalam pembuatan *marker* (penanda) dalam penerapan teknologi *augmented reality*. Proses perancangan aplikasi AR menggunakan *software Metaio Creator* digambarkan pada *flowchart* gambar 2.25. Dimulai dengan pemilihan model penanda (*trackables*). Model penanda terbagi menjadi 3 bentuk pelacakan, yaitu; *image tracking*, *object tracking* dan *environment tracking*. Kemudian penanda yang telah dipilih disimpan kedalam *AR Store*. Proses pemilihan

informasi tambahan (*resources*) yang ditampilkan untuk penanda tersebut. Informasi tambahan tersebut dapat berupa *3D Model*, *image*, *audio file*, *page video*, *video*, *youtube video*, *button*, *Junio channel*, dan lain-lain. Setelah ditambahkan informasi, maka proses terakhir yaitu penyimpanan informasi tambahan dari penanda tersebut. Untuk melakukan penambahan penanda dan informasinya, alur kerja tersebut dapat diulang kembali.



Gambar 2.25 Alur Kerja Perancangan Aplikasi AR.



Gambar 2.26 Tampilan Aplikasi ARca

- Wahyudi, Andria Kusuma (2013) dalam penelitiannya yang berjudul “ARca: Perancangan Buku Interaktif Berbasis Augmented Reality pada Pengenalan dan Pembelajaran Candi Prambanan dengan Smartphone Berbasis Android” sangat efektif dalam proses pemahaman materi pelajaran dikarenakan siswa dapat berinteraksi langsung dengan objek 3D dalam materi tersebut sehingga dapat meningkatkan motivasi siswa dalam proses belajar serta dengan menambahkan beberapa unsur multimedia seperti audio, video, dan sebagainya dalam proses belajar juga meningkatkan pemahaman siswa dalam menerima materi yang diajarkan.