BAB 2
LANDASAN TEORI

2.1 Sistem

Sistem adalah beberapa komponen yang memiliki fungsi yang berbeda dan saling bekerja sama untuk mencapai suatu tujuan [5]. Sebagai gambaran, jika dalam sebuah system terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari system. Berikut adalah beberapa elemen yang membentuk sebuah sistem :

a. Tujuan

   Setiap sistem pasti memiliki tujuan. Tujuan inilah yang mengarahkan sebuah sistem dalam proses kerjanya. Tanpa Tujuan, sistem menjadi tak terarah dan tidak dapat dikendalikan.

b. Input

   Input suatu sistem adalah segala sesuatu yang masuk ke dalam database dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak berwujud

c. Proses

   Proses merupakan bagian yang melakukan perubahan atau transformasi dari input menjadi output.

d. Output

   Output merupakan hasil pemrosesan. Pada sistem informasi, output bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

e. Mekanisme Pengendalian dan Umpan Balik

   Mekanisme pengendalian diwujudkan dengan menggunakan umpan balik, yang mencuplik output. Umpan balik 11ini digunakan untuk mengendalikan baik input maupun proses. Tujuannya adalah untuk mengatur agar system berjalan sesuai dengan tujuan.

2.2 Aplikasi

Program adalah sekumpulan instruksi diberikan untuk mengendalikan perangkat keras Komputer [5]. Secara lebih umum, program komputer inilah yang disebut aplikasi. Program biasa dikelompokan menjadi beberapa kelompok, diantaranya :

a. Program Sistem

   Program sistem adalah program yang digunakan untuk mengontrol sumber daya komputer, seperti CPU dan peranti masukan/keluaran. Kedudukan program ini adalah sebagai perantara antara program aplikasi dan perangkat keras komputer. Itulah sebabnya, peran program sistem sering kali tidak terlihat langsung.

b. Program Aplikasi

   Program Aplikasi adalah program yang dibuat oleh pemakai dengan ditunjukan untuk melakukan suatu tugas khusus. Program seperti ini biasa dikelompokan menjadi dua, yaitu program aplikasi serbaguna dan program aplikasi spesifik.

2.3 Basis Data dan DBMS

2.3.1 Basis Data

Basis Data (Database) adalah suatu perorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi [5]. Basis Data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas data. Definisi dasar struktur database adalah :

a. Data

   Data adalah sekumpulan informasi mengenai suatu objek yang dinyatakan dengan angka, huruf, gambar, video, suara dan sebagainya yang berkaitan dan belum mempunyai arti.

b. Informasi

   Informasi adalah hasil pengolahan data yang konkrit dan sudah mempunyai arti untuk mencapai suatu tujuan tertentu.

c. Field

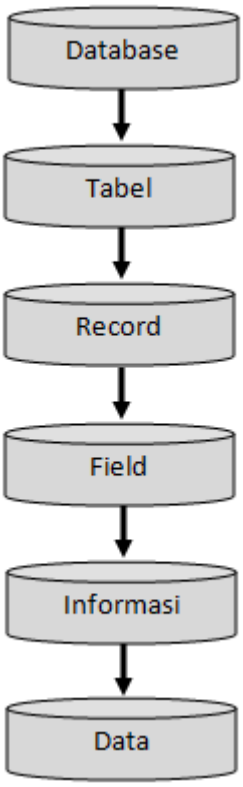
Field adalah elemen dari tabel yang berisikan informasi tertentu yang spesifik tentang subjudul tabel pada sebuah item data.

d. Record

Record adalah kumpulan filed yang digabungkan menjadi satu kesatuan.

e. Tabel

Tabel adalah hal yang paling mendasar dalam hal penyimpanan data yang terdiri dari field dan record.



Gambar 2. 1 Struktur Database

2.3.2 DBMS

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memlihara, mengontrol dan mengakses data dengan cara yang praktis dan efisien. DBMS dapat digunakan untuk mengakomodasi berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda. Umumnya DBMS menyediakan fitur-fitur sebagai berikut :

a. Independensi data-program

Karena basis data ditangani oleh DBMS, program dapat ditulis sehingga tidak tergantung pada struktur data dalam basis data.

b. Keamanan

Keamanan dimaksudkan untuk mencegah pengaksesan data oleh orang yang tidak berwenang.

c. Integritas

Hal ini ditunjukan untuk menjaga agar data selalu dalam keadaan yang valid dan konsisten.

d. Konkurensi

Konkurensi memungkinkan data dapat diakses oleh banyak pemakai tanpa menimbulkan masalah.

e. Pemulihan

DBMS menyediakan mekanisme untuk mengembalikan basis data ke keadaan semula yang konsisten sekiranya terjadi gangguan perangkat keras atau kegagalan perangkat lunak.

f. Katalog Sistem

Katalog sistem adalah deskripsi tentang data yang terkandung dalam basis data yang dapat diakses oleh pemakai.

g. Perangkat produktivitas

Untuk menyediakan kemudahan bagi pemakai dan meningkatkan produktivitas, DBMS menyediakan sejumlah perangkat produktivitas seperti pembangkit query dan pembangkit laporan.

2.3.2.1 Kelebihan DBMS

Kelebihan DBMS diantaranya :

a. Mengendalikan atau mengurangi duplikasi data.
b. Menjaga konsistensi dan integrasi data.
c. Memudahkan pemeroleh informasi yang lebih banyak dari data yang sama disebabkan data dari berbagai bagian dalam organisasi dikumpulkan menjadi satu.
d. Meningkatkan keamanan data dari orang yang tak berwenang.
e. Dapat menghemat biaya karena data dapat dipakai oleh banyak departemen.

2.3.2.2 Kekurangan DBMS

Kekurangan DBMS diantaranya :

a. Kompleksitas yang tinggi membuat administrator dan pemakai akhir harus benar-benar memahami fungsi-fungsi dalam DBMS agar dapat diperoleh manfaat optimal.
b. Ukuran penyimpanan yang dibutuhkan oleh DBMS sangat besar dan memerlukan memori yang besar agar bisa bekerja secara efektif.
c. Rata-rata harga DBMS yang handal sangat mahal.
d. Terkadang DBMS meminta kebutuhan hardware dengan spesifikasi tertentu sehingga diperlukan biaya tambahan.

2.3.2.3 Pengelolaan DBMS

Semua DBMS memiliki dua macam bahasa yang digunakan untuk mengelola dan mengorganisasi data, yaitu :

a. Bahasa Definisi Data (Data Definition Language atau DDL).

   DDL adalah perintah-perintah yang biasa digunakan oleh administrator basis data untuk mendefinisikan skema basis data dan juga subskema. Pernyataan SQL seperti CREATE TABLE dan DROP TABLE merupakan contoh perintah yang termasuk dalam kategori DDL.

b. Bahasa Manipulasi Data (DML)

   DML adalah perintah-perintah yang digunakan untuk melakukan hal-hal seperti berikut :

   1. Mengambil data pada basis data.
   2. Menambahkan data pada basis data.
   3. Mengubah data pada basis data.
   4. Menghapus data pada basis data.
   5. Mengexport data

2.4 Jaringan Komputer

Jaringan komputer (network) adalah sebuah sistem operasi yang terdiri atas sejumlah komputer dan perangkat jaringan lainnya yang bekerja bersama-sama untuk mencapai suatu tujuam yang sama atau suatu jaringan kerja yang

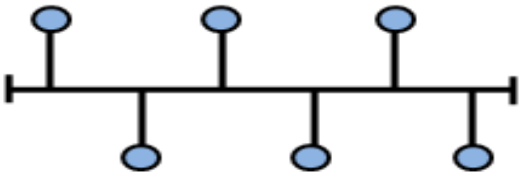
terdiri dari titik-titik (nodes) yang terhubung satu sama lain, dengan atau tanpa kabel [5].

2.4.1. Topologi Jaringan

Tujuan dari suatu jaringan adalah menghubungkan jaringan- jaringan yang telah ada dalam jaringan tersebut sehingga informasi dapat ditransfer dari satu lokasi ke lokasi yang lain [5]. Struktur *Geometric* ini disebut dengan LAN Topologies. Terdapat 6 jenis topologi yaitu :

1. Bus

Konfigurasi lainnya dikenal dengan istilah bus-network, yang cocok digunakan untuk daerah yang tidak terlalu luas. Setiap komputer (setiap simpul) akan dihubungkan dengan sebuah kabel komunikasi melalui sebuah interface. Setiap komputer dapat berkomunikasi langsung dengan komputerataupun peralatan lainnya yang terdapat didalam network, dengan kata lain, semua simpul mempunyai kedudukan yang sama. Dalam hal ini, jaringan tidak tergantung kepada komputer yang ada dipusat, sehingga bila salah satu peralatan atau salah satu simpul mengalami kerusakan, sistem tetap dapat beroperasi. Setiap simpul yang ada memiliki address atau alam sendiri. Sehingga untuk meng-access data dari salah satu simpul, user atau pemakai cukup menyebutkan alamat dari simpul yang dimaksud. Keunggulan topologi Bus adalah pengembangan jaringan atau penambahan workstation baru dapat dilakukan dengan mudah tanpa mengganggu workstation lain. Kelemahan dari topologi ini adalah bila terdapat gangguan di sepanjang kabel pusat maka keseluruhan jaringan akan mengalami gangguan.

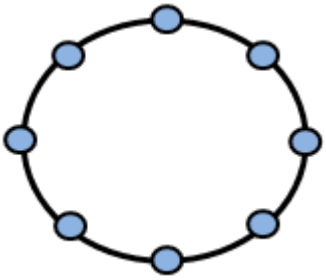


Gambar 2. 2 Topologi Jaringan Bus

2. Ring

Pada jaringan ini terdapat beberapa peralatan saling dihubungkan satu dengan lainnya dan pada akhirnya akan membentuk bagan seperti halnya sebuah cincin. Jaringan cincin tidak memiliki suatu titik yang bertindak sebagai pusat ataupun pengatur lalu lintas data, semua simpul mempunyai tingkatan yang sama.

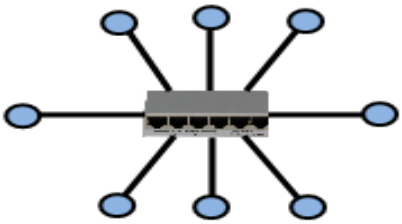
Data yang dikirim akan berjalan melewati beberapa simpul sehingga sampai pada simpul yang dituju. Dalam menyampaikan data, jaringan bisa bergerak dalam satu ataupun dua arah. Walaupun demikian, data yang ada tetap bergerak satu arah dalam satu saat. Pertama, pesan yang ada akan disampaikan dari titik ketitik lainnya dalam satu arah. Apabila ditemui kegagalan, misalnya terdapat kerusakan pada peralatan yang ada, maka data yang ada akan dikirim dengan cara kedua, yaitu pesan kemudian ditransmisikan dalam arah yang berlawanan, dan pada akhirnya bisa berakhir pada tempat yang dituju. Konfigurasi semacam ini relative lebih mahal apabila dibanding dengan konfigurasi jaringan bintang. Hal ini disebabkan, setiap simpul yang ada akan bertindak sebagai komputer yang akan mengatasi setiap aplikasi yang dihadapinya, serta harus mampu membagi sumber daya yang dimilikinya pada jaringan yang ada. Disamping itu, sistem ini lebih sesuai digunakan untuk sistem yang tidak terpusat (decentralized-system), dimana tidak diperlukan adanya suatu prioritas tertentu.



Gambar 2. 3 Topologi Jaringan Ring

3. Star

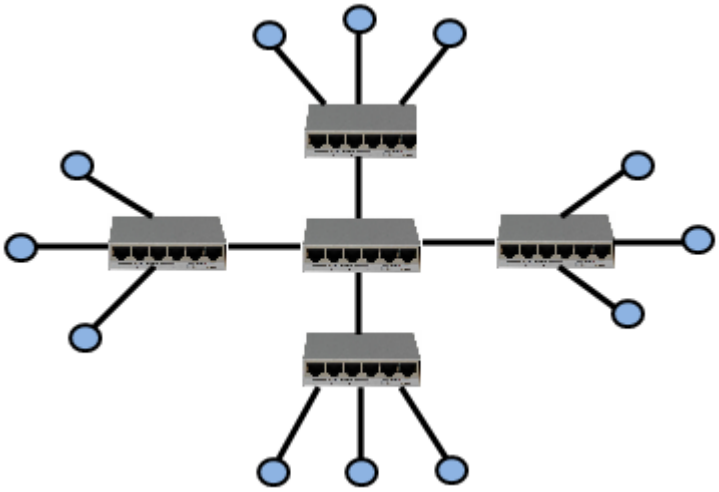
Dalam konfigurasi bintang, beberapa peralatan yang ada akan dihubungkan kedalam satu pusat komputer. Kontrol yang ada akan dipusatkan pada satu titik, seperti misalnya mengatur beban kerja serta pengaturan sumber daya yang ada. Semua link harus berhubungan dengan pusat apabila ingin menyalurkan data kesimpul lainnya yang dituju. Dalam hal ini, bila pusat mengalami gangguan, maka semua terminal juga akan terganggu. Model jaringan bintang ini relatif sangat sederhana, sehingga banyak digunakan oleh pihak per-bank-kan yang biasanya mempunyai banyak kantor cabang yang tersebar diberbagai lokasi. Dengan adanya konfigurasi bintang ini, maka segala macam kegiatan yang ada di-kantor cabang dapatlah dikontrol dan dikoordinasikan dengan baik.



Gambar 2. 4 Topologi Jaringan Star

4. Extended Star

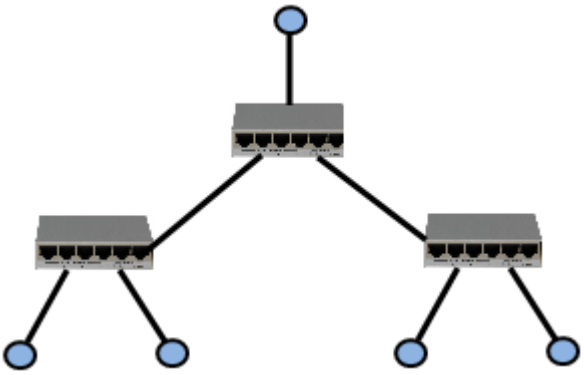
Merupakan perkembangan dari topologi *star*. Memiliki beberapa titik yang terhubung ke satu titik utama.



Gambar 2. 5 Topologi Jaringan Extended Star

5. Hierarchical topology

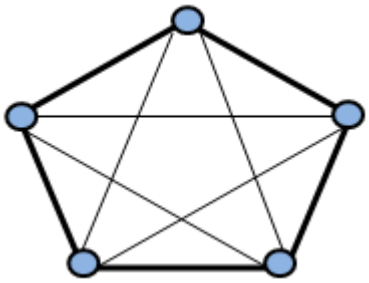
Dibuat mirip dengan topologi *extended star*. Sistem dihubungkan ke komputer yang mengendalikan jalur penghubung pada topologi.



Gambar 2. 6 Topologi Jaringan Hierarchical

6. Mesh

Topologi Mesh adalah suatu topologi yang memang didisain untuk memiliki tingkat restorasi dengan berbagai *alternatif* rute atau penjaluran yang biasanya disiapkan dengan dukungan perangkat lunak atau *software*. Komponen utama yang digunakan dalam topologi mesh ini adalah *Digital Cross Connect* (DXC) dengan satu atau lebih dari dua sinyal *aggregate*, dan tingkat *cross connect* (koneksi persilangan) yang beragam pada level sinyal SDH. Topologi jaringan mesh ini menerapkan hubungan antar sentral secara penuh. Jumlah saluran ini harus disediakan untuk membentuk suatu jaringan topologi mesh adalah jumlah sentral dikurangi 1 (n-1, dengan n adalah jumlah sentral). Tingkat kerumitan yang terdapat pada jaringan mesh ini sebanding dengan meningkatnya jumlah sentral yang terpasang. Dengan demikian disamping kurang ekonomis juga relatif mahal dalam pengoperasiannya.



Gambar 2. 7 Topologi Jaringan Mesh

2.4.2. Tipe Jaringan

Dalam jaringan terdapat tiga buah peran yang dijalankan. Yang pertama adalah client. Peran ini hanya sebatas pengguna tetapi tidak menyediakan sumber daya (sharing), informasi, dan lain-lain. Peran kedua adalah sebagai peer, yaitu client yang menyediakan sumber daya untuk dibagi kepada client lain sekaligus memakai sumber daya yang tersedia pada client yang lain (peer to peer). Sedangkan peran yang terakhir adalah sebagai server, yaitu menyediakan sumber daya secara maksimal untuk digunakan oleh client tetapi tidak memakai sumber daya yang disediakan oleh client [5]. Dibawah ini akan dijelaskan beberapa tipe jaringan.

1. Jaringan Berbasis Server

Jaringan berbasis server atau client-server diartikan dengan adanya server didalam sebuah jaringan yang menyediakan mekanisme pengamanan dan

pengelolaan jaringan tersebut. Jaringan ini terdiri dari banyak client dari satu atau lebih server. Client juga biasa disebut front-end meminta layanan seperti penyimpanan dan pencetakan data ke printer jaringan, sedangkan server yang sering disebut back-end menyampaikan permintaan tersebut ke tujuan yang tepat.

2. Jaringan Peer to Peer

Setiap computer di dalam jaringan peer mempunyai fungsi yang sama dan dapat berkomunikasi dengan computer lain yang telah member izin. Jadi, secara sederhana setiap komputer pada jaringan peer berfungsi sebagai client dan server sekaligus. Jaringan peer digunakan di sebuahkantor kecil dengan jumlah computer sedikit, dibawah sepuluh workstation.

3. Jaringan Hybrid

Jaringan hybrid memiliki semua yang terdapat pada tiga tipe jaringan di atas. Ini berarti pengguna dalam jaringan dapat mengakses sumber daya yang dishare oleh jaringan peer, sedangkan diwaktu bersamaan juga dapat memanfaatkan sumber daya yang disediakan oleh server.Keuntungan jaringan hybrid adalah sama dengan keuntungan menggunakan jaringan berbasis server dan berbasis peer. Jaringan hybrid memiliki kekurangan seperti pada jaringan berbasis server.

2.5 Internet

Internet adalah sistem informasi global berbasis komputer. Internet terbentuk dari jaringan-jaringan komputer-komputer yang terkoneksi satu sama lain. Setiap jaringan mungkin menghubungkan puluhan, ratusan, hingga ribuan komputer, yang memungkinkan komputer-komputer itu saling berbagi data dan informasi satu sama lain dan untuk saling berbagi sumber daya komputasi yang mahal dan langka seperti komputer-komputer super (*SuperComputer*) yang sangat kuat dan sistem basis data yang berisi informasi yang bermanfaat [5].

2.5.1 Perkembangan Internet

Penggunaan internet berkembang pesat sejak penemuannya. Alih- alih menghubungkan jaringan-jaringan secara terbatas pada tipe komputer tertentu, teknologi internet memungkinkan koneksi terjadi diantara berbagai jenis komputer. Antar berbagai sistem operasi (baca:tidak terikat *platform* perangkat keras maupun perangkat lunak). Tidak ada jaringan yang terlalu cepat atau lambat,

terlalu besar atau terlalu kecil sehingga tidak bisa dikoneksikan. Internet dapat menghubungkan jaringan-jaringan sederhana yang terdiri dari beberapa komputer di suatu ruangan hingga jaringan- jaringan canggih yang merentang antar benua dan menghubungkan ribuan bahkan jutaan komputer.

2.5.2 Sumber Daya Internet

Internet memiliki dua sumber daya diantaranya :

a. Surat Elektronis.

Sumber daya internet yang cukup banyak dimanfaatkan oleh para pemakaii adalah surat elektronis (e-mail). Transfer e-mail yang cepat menggunakan protol yang disebut SMTP (Simple Mail Transfer Protocol). Dalam hal ini mail server segera menghubungi tujuan dan kemudian mengirimkan surat. Model pengiriman e-mail yang lain adalah simpan dan teruskan. Cara ini diterapkan jika server tujuan tidak selalu terhubung ke internet.

b. World-Wide Web

Sistem pengaksesan informasi dalam internet yang paling terkenal adalah World Wide Web (WWW) atau biasa dikenal web. Web menggunakan protocol yang disebut HTTP (Hypertext Transfer Protocol) yang berjalan pada TCP/IP. Informasi yang terdapat pada Web disebut halaman web. Untuk mengakses sebuah halaman web dari browser, pemakai perlu menyebutkan URL (Uniform Resource Locator).

2.6 Unified Modelling Language

Unified Modelling Language (UML) adalah sebuah model yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentukbentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya, yaitu Grady Booch OOD (*Object- Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*) [6].

2.6.1 Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

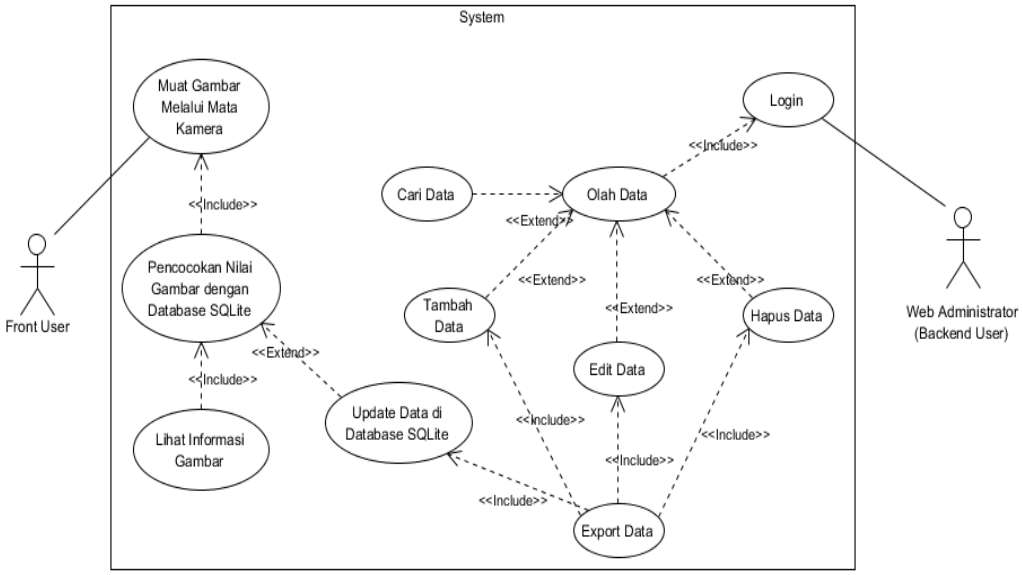
1. Use Case Diagram untuk memodelkan proses bisnis.
2. Conceptual Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. Sequence Diagram untuk memodelkan pengiriman pesan (message) antar objek.
4. Collaboration Diagram untuk memodelkan interaksi antar objek.
5. State Diagram untuk memodelkan perilaku objek di dalam sistem.
6. Activity Diagram untuk memodelkan perilaku userdan objek di dalam sistem.
7. Class Diagram untuk memodelkan struktur kelas.
8. Objek Diagram untuk memodelkan struktur objek.
9. Component Diagram untuk memodelkan komponen objek.
10. Deployment Diagram untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan tiga macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case diagram, sequence diagram,* dan *class diagram.*

a. Use Case Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. Actormerepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor.Use case* digambarkan berbentuk elips dengan

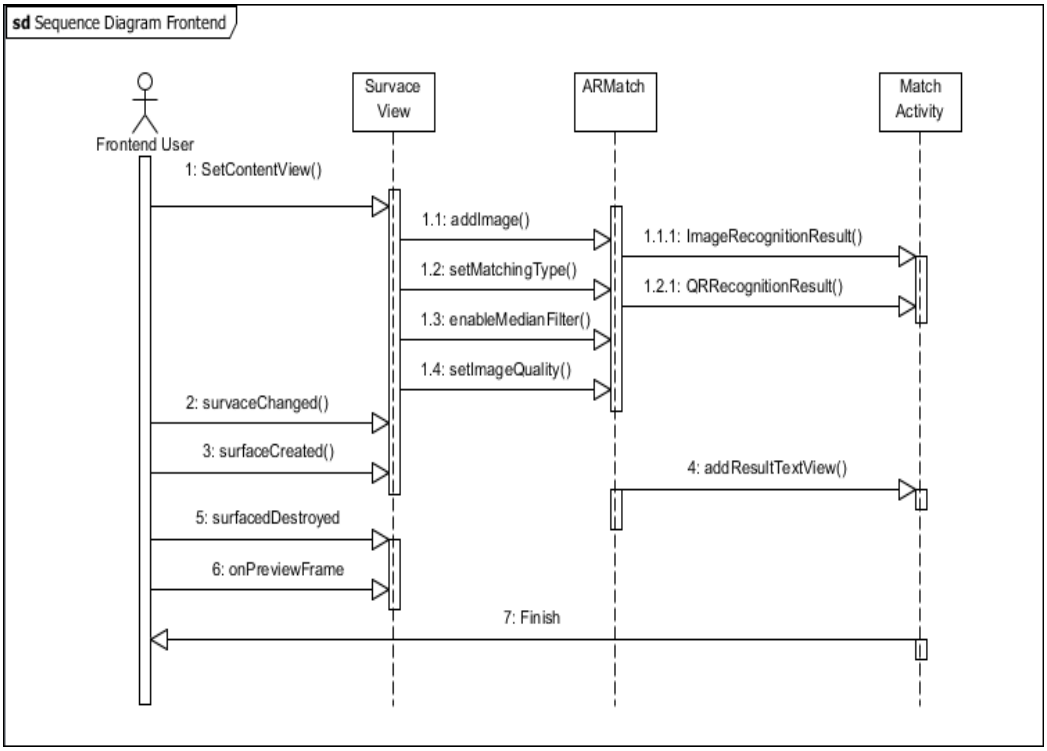
nama operasi dituliskan didalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke *use case.*



Gambar 2. 8 Contoh Use Case Diagram

b. Sequence Diagram

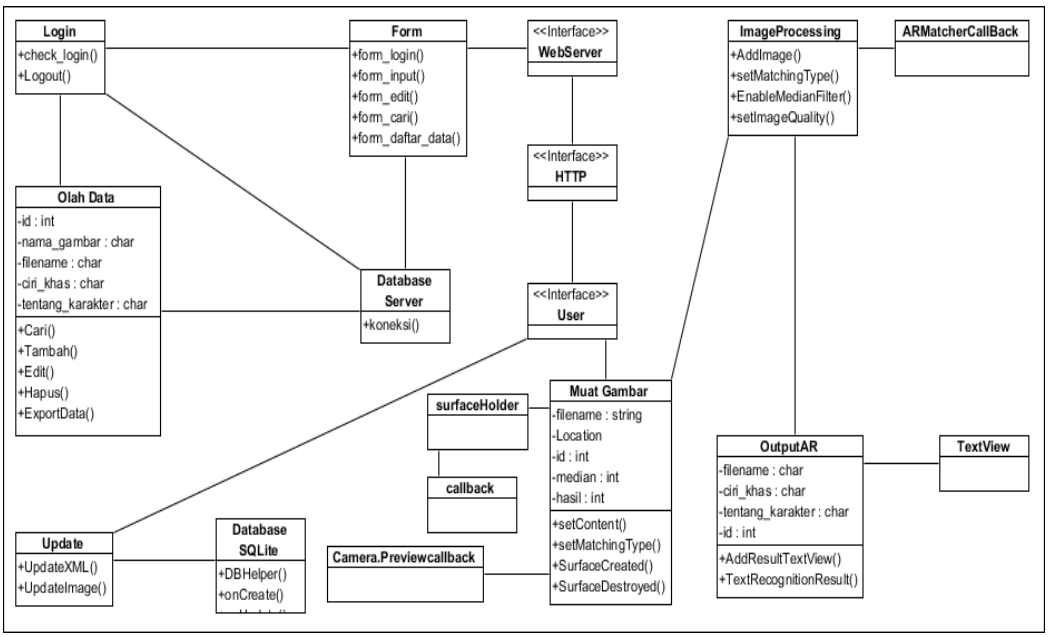
Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case.* Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi [7].



Gambar 2. 9 Contoh Sequence Diagram

c. Class Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan [7].

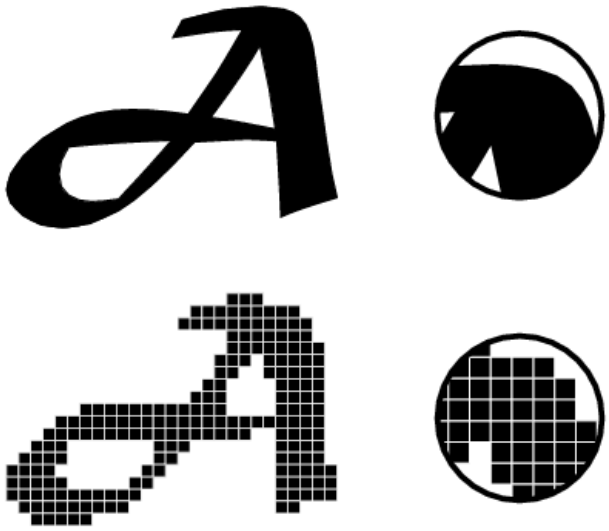


Gambar 2. 10 Contoh Class Diagram

2.7 Gambar Digital

Citra digital didefinisikan sebagai fungsi f(x,y) dua dimensi, dimana x dan y adalah koordinat spasial dan f(x,y) adalah disebut dengan intensitas atau tingkat keabuan gambar pada koordinat x dan y. Jika x, y, dan nilai f terbatas dalam diskrit, maka disebut dengan gambar digital. Citra digital dibentuk dari sejumlah elemen terbatas, yang masing-masing elemen tersebut memiliki nilai dan koordinat tertentu. *Pixel* adalah elemen gambar yang memiliki nilai yang menunjukkan intensitas warna.

Citra digital dapat dibagi menjadi dua jenis. Jenis pertama adalah citra digital yang dibentuk oleh kumpulan *pixel* dalam array dua dimensi. Citra jenis ini disebut citra bitmap (*bitmap image*) atau citra raster (*raster image*). Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik vektor (*vector graphics*).



Gambar 2. 11 Vektor image dan Bitmap Image

2.7.1 Pengolahan Citra Digital

Pengolahan citra merupakan proses pengolahan dan analisis gambar yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk gambar. Istilah pengolahan gambar digital secara umum didefinisikan sebagai pemrosesan gambar dua dimensi dengan komputer. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu.

Umumnya citra digital berbentuk persegi panjang atau bujur sangkar dengan lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau *pixel* sehingga ukuran citra selalu bernilai bulat. Setiap *pixel* memiliki koordinat sesuai posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 tergantung pada sistem yang digunakan. Setiap titik juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili oleh titik tersebut. Beberapa pemrosesan citra yang bisa dilakukan adalah sebagai berikut [8]:

- Memperbaiki kualitas gambar,
- Dilihat dari aspek radiometrik (peningkatan kontras, transformasi warna, restorasi gambar) dan dari aspek geometrik (rotasi, translasi, skala, transformasi geometrik),
- Melakukan pemilihan ciri gambar (*feature images*) yang optimal untuk tujuan analisis,
- Melakukan proses penarikan informasi atau deskripsi obyek atau pengenalan objek yang terkandung pada citra,
- Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data.

Istilah pengolahan citra digital secara umum dapat didefinisikan sebagai pemrosesan citra dua dimensi dengan menggunakan perangkat komputer. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu. Umumnya citra digital berbentuk persegi panjang atau bujur sangkar dengan lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau *pixel* sehingga ukuran citra selalu bernilai bulat. Setiap *pixel* memiliki koordinat sesuai posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 tergantung pada sistem yang digunakan. Setiap titik juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili oleh titik tersebut.

2.7.1.1. Sejarah Pengolahan Citra Digital

Minat pada bidang pengolahan citra secara digital (*digital image processing*) tercatat dimulai pada awal tahun 1921, pada waktu itu sebuah foto

untuk pertama kalinya berhasil ditransmisikan secara digital melalui kabel laut dari kota New York ke kota London (*Bartlane Cable Picture Trasmision System*).

Keuntungan utama yang dirasakan pada waktu itu adalah pengurangan waktu pengiriman foto dari sekitar satu minggu menjadi kurang dari 3 jam. Foto tersebut dikirim dalam bentuk kode digital, selanjutnya diubah kembali oleh suatu printer telegraph pada sisi penerima. Masalah yang muncul pada saat itu berkisar pada teknik transmisi data secara digital serta teknik reproduksi pada sisi penerima untuk mendapatkan satu resolusi gambar yang baik. Walaupun minat dalam bidang ini telah dimulai sejak tahun 1921, tetapi perkembanganya secara pesat baru tercatat pada sekitar tahun 1960. Pada saat itu teknologi komputer telah dianggap memenuhi suatu kecepatan proses serta kapasitas memori yang dibutuhkan oleh berbagai algoritma pengolahan citra. Sejak itulah berbagai jenis aplikasi mulai dikembangkan, yang secara umum dapat dikelompokan dalam dua jenis kegiatan yaitu:

1. Memperbaiki kualitas suatu gambar sehingga dapat lebih mudah diinterpretasikan oleh mata manusia.
2. Mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis oleh suatu mesin.

Bidang aplikasi sangat erat hubungannya dengan ilmu pengenalan pola (*pattern recognition*) yang umumnya bertujuan untuk mengenali suatu objek dengan cara mengekstraksi informasi penting dalam suatu citra. Rinaldi Munir (2004:12) dalam bukunya mengemukakan beberapa contoh apliksi bidang ini di berbagai disiplin ilmu yaitu:

1. Dalam bidang kedokteran
   Sistem untuk mendeteksi diagnosa suatu kelainan dalam tubuh manusia melalui gambar yang dihasilkan oleh suatu gambar scanner.
2. Dalam bidang industri
   Sistem untuk memeriksa kualitas suatu produk melalui kamera video.
3. Dalam bidang perdagangan
   Sistem untuk mengenal huruf / angka pada suatu formulir secara automatis oleh suatu mesin pembaca.

4. Dalam bidang militer

   Sistem pengenalan target peluru kendali melalui sensor visual.

5. Dalam bidang biologi

   Sistem pengenalan jenis kromosom melalui gambar mikroskop.

Keikutsertaan berbagai disiplin ilmu dalam kegiatan pengolahan citra dimulai dari pembentukan model matematik suatu objek sampai dengan teknik analisis dan teknik klasifikasi berbagai jenis objek.

2.7.2 Elemen-elemen Citra Digital

Citra digital mengandung sejumlah elemen-elemen dasar. Elemen-elemen dasar tersebut dimanipulasi dalam pengolahan citra dan dieksploitasi lebih lanjut dalam komputer vision. Elemen - elemen dasar citra digital diantaranya:

1. Kecerahan *(brightness)*

Kecerahan adalah kata lain untuk intensitas cahaya.Sebagai mana telah dijelaskan pada bagian sampling,kecerahan pada sebuah titik (*pixel*) didalam citra bukanlah intensitas yang riil,tetapi sebenarnya adalah intensitas rata-rata dari suatu area yang melingkupinya. Sistem visual manusia mampu menyesuaikan dirinya dengan tingkatan kecerahan(*brightness level*) mulai dari yang paling rendah sampai yang paling tinggi dengan jangkauan 1010.

2. Kontras *(contrast)*

Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) didalam sebuah gambar. Citra dengan kontras rendah dicirikan sebagai besar komposisi citranya adalah terang sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

3. Kontur *(contour)*

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas cahaya pada pixel-pixel yang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendeteksi tepi-tepi (*edge*) objek didalam citra.

4. Warna (*color*)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna

mempunyai panjang gelombang (λ) yang berbeda. Warna merah mempunyai panjang gelombang yang paling tinggi, sedangkan warna ungu (violet) mempunyai panjang gelombang yang paling rendah. Warna-warna yang diterima oleh mata (sistem visual mata) merupakan hasil kombinasi cahaya dengan panjang gelombang yang berbeda. Penelitian memperlihatkankombinasi warna yang memberikan rentang warna yang paling lebar adalah merah (*red*), hijau (*green*), biru (*blue*).

Penyesuaian warna pada visual kita tidak jarang dapat menimbulkan "cacat" warna (distorsi) yang dilihat. Ada dua jenis distorsi, yakni distorsi warna terhadap ruang (misal bercak abu-abu yang berada disekitar warna hijau akan berkesan ungu), dan distorsi terhadap waktu (misalnya setelah melihat warna hijau kita langsung melihat warna abu-abu, maka warna ungulah yang berkesan pada mata kita).

5. Bentuk (*shape*)

Pada umumnya citra yang dibentuk oleh mata merupakan citra dua dimensi, sedangkan objek yang diamati biasanya adalah 3 dimensi telah diproyeksikan kebidang dua dimensi dan kelihatannya sama. Misalnya, suatu ruangan terlihat berbentuk trapesium pada gambar dua dimensi. Didalam hal ini kita tahu apakah hal ini memang disebabkan oleh bentuk ruangan yang panjang ataukah memang ruangan tersebut berbentuk trapesium.

6. Tekstur (*texture*)

Pada hakikatnya sistem visual manusia tidak menerima informasi citra secara terpisah pada setiap titik, tetapi sesuatu citra dianggapnya sebagai suatu kesatuan, jadi definisi kesamaan suatu objek perlu dinyatakan dalam bentuk kesamaan dari suatu himpunan parameter citra (*brightness*, *color*, *size*) atau dengan kata lain dua buah citra tidak dapat disamakan dari satu parameter saja [10].

2.8 FAST Corner Detection

Library Vuforia SDK menggunakan algoritma *FAST Corner detection* untuk melacak dan mendeteksi keberadaan *object target*. FAST *(Features from Accelerated Segment Test)* adalah suatu algoritma yang dikembangkan oleh Edward Rosten, Reid Porter, and Tom Drummond. FAST *corner detection* ini

dibuat dengan tujuan mempercepat waktu komputasi secara *real-time* dengan konsekuensi menurunkan tingkat akurasi pendeteksian sudut [3].

FAST *corner detection* dimulai dengan menentukan suatu titik *p* pada koordinat (x_p , y_p) pada citra dan membandingkan intensitas titik *p* dengan 4 titik di sekitarnya.

- Titik pertama terletak pada koordinat (x_p, $y_{p-3)}$,
- Titik kedua terletak pada koordinat (x_{p+3}, y_p),
- Titik ketiga terletak pada koordinat (x_p, y_{p+3}), dan
- Titik keempat terletak pada koordinat (x_{p-3}, y_p).

Jika nilai intensitas di titik *p* bernilai lebih besar atau lebih kecil daripada intensitas sedikitnya tiga titik disekitarnya ditambah dengan suatu intensitas batas ambang, maka dapat dikatakan bahwa titik *p* adalah suatu sudut. Setelah itu titik *p* akan digeser ke posisi(x_{p+1},y_p) dan melakukan perbandingan intensitas di keempat titik di sekitarnya lagi. Iterasi ini terus dilakukan sampai semua titik pada citra sudah dibandingkan.

FAST *corner detection* bekerja pada suatu citra sebagai berikut [3]:

1. Tentukan sebuah titik *p* pada citra dengan posisi awal (x_p,y_p),

2. Tentukan lokasi keempat titik, dengan ketentuan sebagai berikut :
   - Titik pertama (*n*=1) terletak pada koordinat (x_p,y_{p-3}),
   - Titik kedua (*n*=2) terletak pada koordinat (x_{p+3},y_p),
   - Titik ketiga (*n*=3) terletak pada koordinat (x_p,y_{p+3}),
   - Titik keempat (*n*=4) terletak pada koordinat (x_{p-3},y_p)

3. Bandingkan intensitas titik pusat *p* dengan keempat titik di sekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat berikut, maka titik pusat *p* adalah sudut

$$C_p = \begin{cases} 1, & In<Ip-t \quad or \quad In>Ip+t \\ 0, & otherwise \end{cases} \tag{2.1}$$

Keterangan :

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

I_n : Nilai intensitas piksel ke-n

I_p : Nilai intensitas titik p

t : Batas ambang nilai intensitas yang ditoleransi (Threshold)

4. Ulangi proses sampai seluruh titik pada citra sudah dibandingkan intensitasnya.

2.9 Augmented Reality

Augmented Reality adalah teknologi yang menggabungkan benda maya tiga dimensi ke dalam sebuah lingkungan nyata dan menampilkannya dalam waktu nyata (*real time*). Tidak seperti realitas maya (*Virtual Reality*) yang sepenuhnya menggantikan kenyataan, realitas tertambah (*Augmented Reality*) hanya sekedar menambahkan atau melengkapi kenyataan. Benda-benda maya menampilkan informasi yang tidak dapat diterima oleh pengguna dengan inderanya sendiri. Hal ini membuat *Augmented Reality* sesuai sebagai alat untuk membantu persepsi dan interaksi penggunanya dengan dunia nyata.

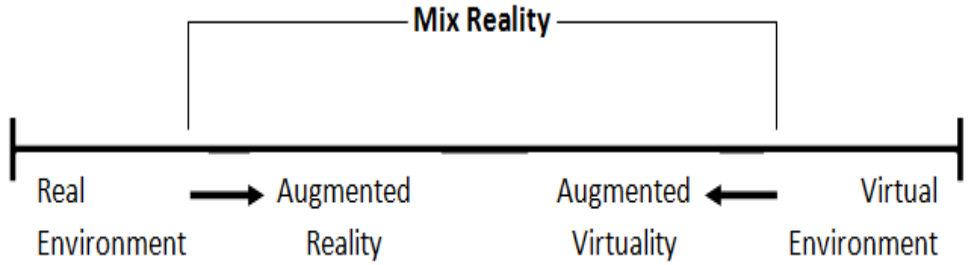
Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata. *Augmented Reality* dapat diaplikasikan untuk semua indera, termasuk pendengaran, sentuhan, dan penciuman. Biasanya teknologi *Augmented Reality* digunakan dalam bidang-bidang seperti kesehatan, militer dan industri manufaktur. *Augmented Reality* juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada perangkat berbasis *mobile smartphone* [11].

2.9.1 Pengertian Augmented Reality

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 *augmented reality* dengan tampilan yang sesuai,

interaktivitas dimungkinkan melalui perangkat-perangkat *input* tertentu, dan integrasi yang baik memerlukan penjejakan yang efektif.

Milgram dan Kishino (1994) merumuskan kerangka kemungkinan penggabungan dan penyatuan dunia nyata dan dunia maya ke dalam sebuah *continuum virtualitas.*



Gambar 2. 12 Continuum virtualitas

Sisi yang paling kiri adalah lingkungan nyata yang hanya berisi benda nyata, dan sisi paling kanan adalah lingkungan maya yang berisi benda maya. Dalam *augmented reality*, yang lebih dekat ke sisi kiri, lingkungan bersifat nyata dan benda bersifat maya, sementara dalam *virtual reality*, yang lebih dekat ke sisi kanan, lingkungan bersifat maya dan benda bersifat nyata. *Augmented reality* dan *virtual reality* digabungkan menjadi *mixed reality* atau realitas campuran.

Augmented Reality atau yang dalam bahasa indonesianya adalah realitas tertambah adalah suatu macam teknologi yang pada intinya adalah menggabungkan benda nyata (3 dimensi) dengan benda maya (2 dimensi) lalu mempekerjakan teknologi tersebut dalam waktu yang dibatasi oleh rentang waktu dan memiliki tenggat waktu yang jelas. Pada intinya *Augmented Reality* berfungsi untuk melengkapi atau menambah objek pada dunia nyata. *Augmented Reality* sebenarnya sudah dapat dikenal orang banyak, karena pemakainnya yang sudah banyak digunakan, seperti pada bidang kesehatan, militer ataupun iklan pada televisi. contohnya, jika kita sering melihat pertandingan sepak bola inggris di televisi, kita bisa melihat iklan minuman yang menjadi sponsor pertandingan tersebut, dan itu terlihat seperti nyata padahal itu adalah teknologi *Augmented Reality*.

Selain menambahkan benda maya dalam lingkungan nyata, *augmented reality* juga berpotensi menghilangkan benda-benda yang sudah ada. Menambah sebuah lapisan gambar maya dimungkinkan untuk menghilangkan atau menyembunyikan lingkungan nyata dari pandangan pengguna. Misalnya, untuk menyembunyikan sebuah meja dalam lingkungan nyata, perlu digambarkan lapisan representasi tembok dan lantai kosong yang diletakkan di atas gambar meja nyata, sehingga menutupi meja nyata dari pandangan pengguna [11].

2.9.2 Sejarah Augmented Reality

Secara umum, *Augmented Reality* (AR) adalah penggabungan antara objek *virtual* dengan objek nyata. Sebagai contoh, adalah saat stasiun televisi, menyiarkan pertandingan sepak bola, terdapat objek *virtual*, tentang skor pertandingan yang sedang berlangsung. Menurut Ronald Azuma pada tahun 1997, *Augmented Reality* adalah menggabungakan dunia nyata dan virtual, bersifat interaktif secara *real time*, dan merupakan animasi 2D maupun 3D. Sejarah tentang *Augmented Reality* dimulai dari tahun 1957-1962, ketika seorang penemu yang bernama Morton Heilig, seorang sinematografer, menciptakan dan memapatenkan sebuah simulator yang disebut Sensorama dengan *visual*, getaran dan bau. Pada tahun 1966, Ivan Sutherland menemukan *head-mounted display* yang dia klaim adalah jendela ke dunia virtual.

Tahun 1975 seorang ilmuwan bernama Myron Krueger menemukan *Videoplace* yang memungkinkan pengguna, dapat berinteraksi dengan objek *virtual* untuk pertama kalinya. Tahun 1989, Jaron Lanier, memeperkenalkan *Virtual Reality* dan menciptakan bisnis komersial pertama kali di dunia maya, Tahun 1992 mengembangkan *Augmented Reality* untuk melakukan perbaikan pada pesawat boeing, dan pada tahun yang sama, LB Rosenberg mengembangkan salah satu fungsi sistem *Augmented Reality*, yang disebut *Virtual Fixtures*, yang digunakan di Angkatan Udara Amerika Serikat, Armstrong Labs, dan menunjukan manfaatnya pada manusia. Pada tahun 1992 juga, Steven Feiner, Blair Maclntyre dan dorée Seligmann, memperkenalkan untuk pertama kalinya *Major Paper* untuk perkembangan *Prototype Augmented Reality*.

Pada tahun 1999, Hirokazu Kato, mengembangkan *ArToolkit* di *HITLab* dan didemonstrasikan di *SIGGRAPH*, pada tahun 2000, Bruce.H.Thomas,

mengembangkan *ARQuake*, sebuah *Mobile Game Augmented Reality* yang ditunjukan di *International Symposium on Wearable Computers*.

Pada tahun 2008, *Wikitude AR Travel Guide*, memperkenalkan *Android G1 Telephone* yang berteknologi *Augmented reality*. Tahun 2009, Saqoosha memperkenalkan *FLARToolkit* yang merupakan perkembangan dari *ArToolkit. FLARToolkit* memungkinkan kita memasang teknologi *Augmented reality* di sebuah *website*, karena *output* yang dihasilkan *FLARToolkit* berbentuk *Flash*. Ditahun yang sama, *Wikitude Drive* meluncurkan sistem navigasi berteknologi *Augmneted Reality* di *Platform Android*. Tahun 2010, *Acrossair* menggunakan teknologi *Augmented Reality* pada *I-Phone 3GS* [11].

Bidang-bidang yang pernah menerapkan teknologi *Augmented Reality* adalah [12]:

1. Kedokteran (*Medical*):

   Teknologi pengambaran sangat dibutuhkan di dunia kedokteran, seperti misanya, untuk simulasi operasi, simulasi pembuatan vaksin virus, dll. Untuk itu, bidang kedokteran menerapkan *Augmented Reality* pada visualisasi penelitian mereka.

2. Hiburan (*Entertainment*)

   Dunia hiburan membutuhkan *Augmented Reality* sebagai penunjang efek-efek yang akan dihasilkan oleh hiburan tersebut. Sebagai contoh, ketika sesorang wartawan cuaca memperkirakan ramalan cuaca, dia berdiri di depan layar hijau atau biru, kemudian dengan teknologi *Augmented Reality*, layar hijau atau biru tersebut berubah menjadi gambar animasi tentang cuaca tersebut, sehingga seolah-olah wartawan tersebut, masuk ke dalam animasi tersebut.

3. Latihan Militer (*Military Training*)

   Militer telah menerapkan *Augmented Reality* pada latihan tempur mereka. Sebagai contoh, militer menggunakan *Augmented Reality* untuk membuat sebuah permainan perang, dimana prajurit akan masuk kedalam dunia game tersebut, dan seolah-olah seperti melakukan perang sesungguhnya.

4. *Engineering Design*

   Seorang *engineering design* membutuhkan *Augmented Reality* untuk menampilkan hasil *design* mereka secara nyata terhadap klien. Dengan

Augmented Reality klien akan tahu, tentang spesifikasi yang lebih detail tentang *desain* mereka.

5. *Robotics* dan *Telerobotics*

   Dalam bidang robotika, seorang operator robot, mengunnakan pengendari pengambaran *visual* dalam mengendalikan robot itu. Jadi, penerapan *Augmented Reality* dibutuhkan di dunia robot.

6. *Consumer Design*

   Virtual reality telah digunakan dalam mempromsikan produk. Sebagai contoh, seorang pengembang menggunakan brosur *virtual* untuk memberikan informasi yang lengkap secara 3D, sehingga pelanggan dapat mengetahui secara jelas, produk yang ditawarkan.

2.9.3 Tujuan Augmented Reality

Tujuan utama dari sistem *Augmented Reality* akan berwujud sebagai sebuah kacamata atau proyektor retina yang akan menyediakan tampilan informasi yang relevan, dipetakan ke lingkungan sekitarnya secara *realtime*. Misalnya, saat melihat sebuah restoran dengan kacamata *Augmented Reality*, maka satu panggilan otomatis langsung ke *database review* atau menu dari *website restoran* tersebut. Seorang ilmuwan yang bekerja pada perusahaan farmasi bisa menggunakan kacamata untuk menampilkan model *3D* dari berbagai molekul dan menggunakannya untuk memvisualisasikan obat yang lebih baik. Anak-anak mungkin menggunakan jaringan yang terhubung kacamata *Augmented Reality* untuk bermain *video game* kehidupan nyata yang memungkinkan menembakkan “laser” dari tangan mereka, meski kemungkinan agak terbatas.

Augmented Reality bergantung pada kemajuan teknologi miniaturisasi dan komputasi bergerak (*mobile computing*). Saat ini, teknologi sistem proyeksi yang efektif dan komputer kecil dan cepat masih kurang matang untuk benar-benar dapat membuat antarmuka *Augmented Reality* yang efektif, walaupun kita sudah semakin dekat pada kesempurnaan dengan riset yang dilakukan terus menerus oleh para ahli.

Purwarupa telah banyak dibuat selama beberapa dekade, tetapi produk utama yang benar-benar layak belum ada. Salah satu teknologi yang menjanjikan adalah proyeksi retina, sehingga kacamata tidak digunakan lagi. Sistem proyeksi

retina komersial sebenarnya sudah ada, namun resolusi dan *palet* warnanya masih sangat rendah.

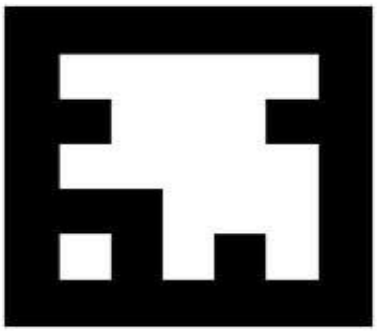
Augmented Reality saat ini sebenarnya sudah ada dalam bentuk yang belum sempurna. Misalnya, komentator olahraga sering menggunakan pena cahaya untuk "menggambar" di lapangan sepak bola untuk memberikan bantuan visual untuk rekan komentatornya. Contoh lain yaitu gambar iklan yang sering tampil dilapangan bulutangkis ketika ditayangkan di TV, seakan gambar iklan ini melekat pada lapangan. Gambar iklan sponsor ini terus berganti-ganti selama pertandingan. Ini menunjukkan bukti dari konsep *Augmented Reality* meski masih terbatas [19].

2.9.4 Macam-macam Metode *Augmented Reality*

Augmented Reality memiliki berbagai jenis Metode yang semakin berkembang sampai saat ini. Berikut macam-macam metoda *Augmented Reality* :

2.8.3.1 Marker Based Tracking

Marker adalah suatu pola yang dibuat dalam bentuk gambar yang akan dikenali oleh *webcam*. *Marker* berfungsi sebagai pendefinisi dari *Augmented Reality*. Informasi *marker* akan digunakan untuk mendefinisikan dan menampilkan sebuah objek *Augmented reality*. *Marker* juga merupakan gambar yang terdiri atas border *outline* dan *pattern image* seperti terlihat pada gambar dibawah ini.



Gambar 2. 13 Contoh Marker

Marker biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi marker dan menciptakan dunia *virtual 3D* yaitu titik (0,0,0) dan 3 sumbu yaitu X,Y,dan Z. Cara pembuatannya pun sederhana, tetapi harus diperhatikan ketebalan *Marker* yang akan dibuat, ketebalan *Marker* jangan kurang dari 25% dari panjang garis tepi agar pada saat proses deteksi *Marker* dapat lebih akurat. Sedangkan objek warna putih sebagai *background*, yang nantinya akan digunakan sebagai tempat objek yang akan di-*render*.

Warna putih pada *Marker* menunjukan warna sebuah objek, sedangkan warna hitam menunjukan latar belakang.Intensitas warna pada suatu objek memiliki warna yang lebih rendah (gelap), sedangkan latar belakang mempunyai intensitas yang lebih tinggi (terang). Adapun beberapa aturan umum dalam pembuatan pola marker, yaitu seperti harus kotak berbingkai hitam dan ini adalah rahasia dari pelacakan sebuah *marker*, ukuran tidak lebih dari 631x634 pixel. Warna selain hitam putih juga bisa dikenali oleh sistem *Marker* membantu komputer dimana letak objek akan ditamplikan.

Ukuran *Marker* yang digunakan dapat mempengaruhi penangkapan pola *Marker* oleh kamera. Semakin besar ukuran *Marker,* maka semakin jauh jarak yang dapat ditangkap oleh kamera dalam proses pendeteksian *Marker*. Namun masalahnya, ketika *Marker* bergerak menjauhi kamera, jumlah *pixel* pada layar kamera menjadi lebih sedikit dan ini bisa mengakibatkan pendeteksian tidak akurat.

2.8.3.2 Markerless Based Tracking

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, 3D Object Tracking,* dan *Motion Tracking.*

1. *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.



Gambar 2. 14 Metode Face Tracking

2. 3D *Object Tracking*

Berbeda dengan *Face Tracking* yang hanya mengenali wajah manusia, secara umum teknik *3D Object Tracking* dapat mengenali semua bentuk benda yang ada disekitar, seperti mobil, meja, televisi, dan lain-lain.



Gambar 2. 15 Metode 3D Object Tracking

3. *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*.



Gambar 2. 16 Metode Motion Tracking

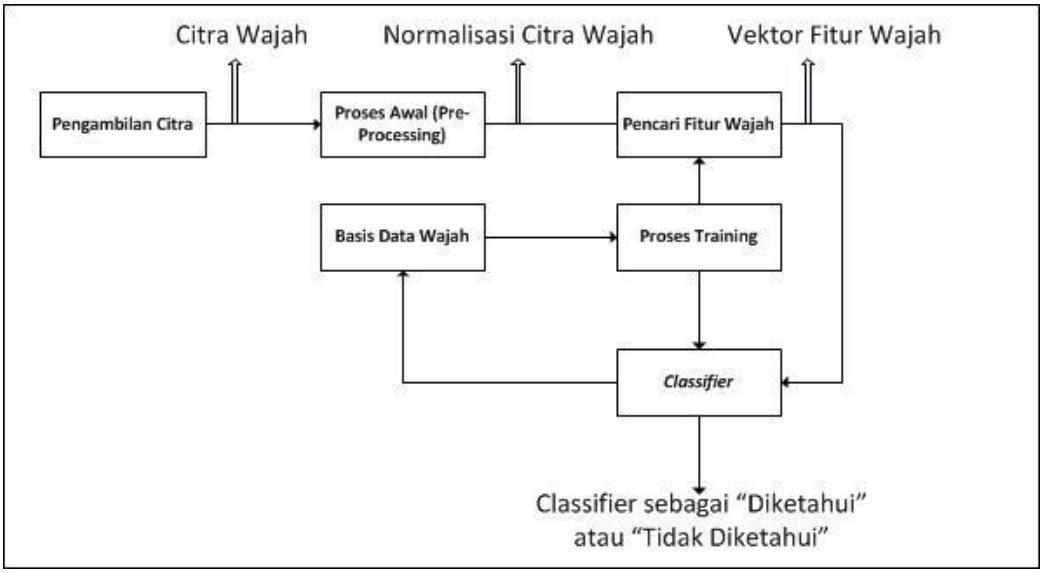
2.8.3.2.1. Sistem Pengenalan Wajah

Sistem pengenalan wajah (*face recognition*) merupakan suatu sistem yang dirancang pada komputer yang bertujuan untuk mengidentifikasi wajah seseorang melalui citra atau video digital. Salah satu metode yang sering digunakan dalam

sistem pengenalan wajah ini adalah dengan cara membandingkan keunikan wajah (*facial features*) dari suatu citra dengan *database* wajah yang telah diambil sebelumnya. Beberapa algoritma dalam suatu sistem identifikasi wajah didasarkan pada ekstraksi dari *landmark* maupun keunikan yang didapatkan pada citra dengan subjek wajah seperti posisi atau ukuran mata, hidung, pipi, dan rahang [13].

Beberapa pendekatan yang digunakan dalam sistem pengenalan wajah diantaranya :

a. *Geometric (Feature Based Matching)* adalah sebuah pendekatan yang mana sebuah wajah dapat dikenali dengan menggunakan *geometrical feature* seperti lebar hidung, posisi pipi, posisi mata dengan cara melakukan ekstraksi posisi relatif dari parameter pembeda pada wajah seperti mata, mulut, hidung, dan pipi [14].
b. *Template Matching* merupakan sebuah pendekatan dalam sistem pengenalan wajah yang direpresentasikan ke dalam larik dua dimensi, kemudian dibandingkan dengan matriks yang sesuai ke dalam satu *template* yang telah mempresentasikan seluruh bagian wajah [14].



Gambar 2. 17 Alur Kerja Sistem Pengenalan Wajah [7]

Gambaran mengenai bagaimana alur kerja umum dari suatu system pengenalan wajah terdapat dalam Gambar 2.6. Secara menyeluruh system pengenalan wajah dapat dilakukan dengan beberapa cara [13], yaitu :

1. **Detection**

Detection merupakan proses pengambilan data berupa citra yang dapat diperoleh dengan melakukan proses pemindaian (*scanning*) dari foto 2D atau dapat pula menggunakan citra video yang didapatkan dari *live recording* (3D).

2. **Alignment**

Setelah wajah telah terdeteksi, sistem kemudian menentukan posisi wajah, ukuran, dan pose.

3. **Measurement**

Sistem kemudian melakukan komputasi berupa pembentukan kurva dari wajah dan membuat suatu *template* untuk menampung hasil komputasi dari kurva tersebut.

4. **Representation**

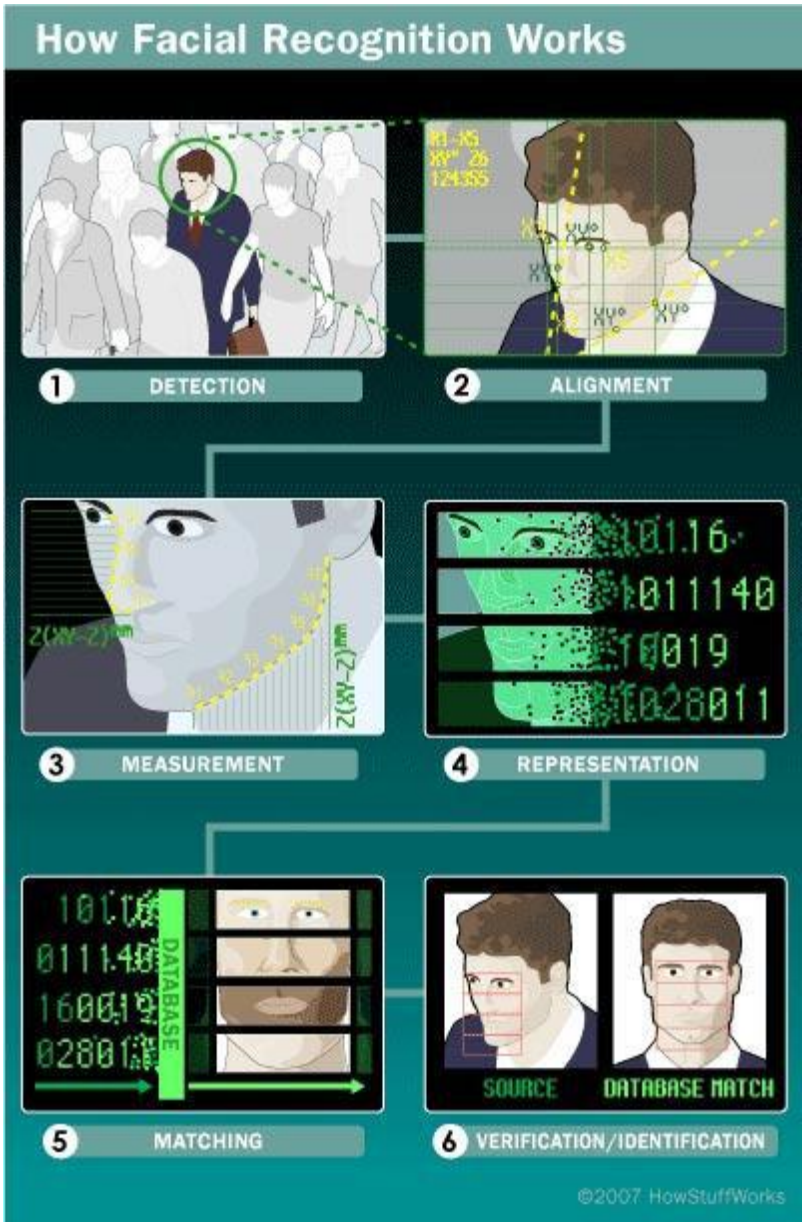
Setelah melakukan *measurement*, sistem menerjemahkan *template* yang telah dibuat dalam bentuk kode unik (*unique code*). Pengkodean inilah yang akan memberikan representasi dari keunikan wajah dalam suatu citra.

5. **Matching**

Citra hasil pengkodean kemudian dibandingkan dengan *database* citra yang telah dibuat sebelumnya.

6. **Verification atau Identification**

Setelah itu, sistem dapat memberikan hasil pengenalannya terhadap citra yang ingin dikenali identitasnya.



Gambar 2. 18 Proses Dalam Sistem Pengenalan Wajah [13]

2.10 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc.membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel [13]. 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 saat perilisan perdana *Android*, 5

November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka(*open source*) pada perangkat seluler. Di lain pihak, *Google* merilis kode – kode *Android* dibawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Terdapat dua jenis distributor sistem operasi *Android*, pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services (GMS)* dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution(OHD)*.

2.9.1 Sejarah Android

Android pertama kali dikembangkan oleh perusahaan bernama AndroidInc. Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White.Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. versi android terbaru yaitu versi 3.0.

Pada tahun 2005 Google mengakusisi Android Inc yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Crish White. Yang kemudian pada tahun itu juga memulai membangun platform Android secara lebih intensif. Kemudian pada tanggal 12 November 2007 Google bersama Open Handset Aliance (OHA) yaitu konsorium perangkat lunak mobile terbuka, merilis Google Android SDK, setelah mengumumkannya seminggu sebelumnya. Dan sambutannya sangat luar biasa , hampir semua media berita tentang IT dan Programming memberitakan tentang dirilisnya Android SDK (*Software Developtment Kit*). Ini dikarenakan ide dari flatform Android sangat menarik untuk developer dan programmer di seluruh penjuru dunia. Di sisi lain, produsen pembuat handset ponsel juga segera berlomba-lomba membuat handset dengan

platform Android ini. Google bersama dengan OHA merilis paket software SDK yang lengkap untuk mengembangkan aplikasi pada perangkat mobile, yaitu : Sistem Operasi, Middleware dan Aplikasi utama untuk perangkat mobile. Sebagai Programmer atau Developer kita bisa melakukan segalanya, mulai dari membuat aplikasi pengiriman SMS hanya dengan dua baris kode, hingga mengganti even pada Home Screen perangkat Android. Selain itu, bahkan dengan mudah kita bias membuatdan mengkostumisasi Sistem Operasinya, atau mengganti semua aplikasi default dari google.

Semua aplikasi yang dibuat untuk android akan memiliki akses setara dalam mengakses seluruh kemampuan handset, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan platform android ini, Programer atau Developer secara penuh akan bias mengkustomisas perangkat androidnya. Android built in pada Linux kernel (Open Linux Kernel), dengan sebuah mesin virtual yang telah didesain dan untuk mengoptimalkan penggunaan sumberdaya memori dan handware pada lingkungan perangkat mobile (*Mobile Environment*). Dalvik adalah nama dari Android Virtual Mesin, yang merupak interpreter virtual mesin yang akan mengeksekusi file kedalam format Dalvik Executable (*.dex). Sebuah format yang dirancang untuk ruang penyimpanan yang efisien dan eksekusi memori yang terpetakan (memory-mappable execution). Dlavik Virtual Mesun (Dalvik VM) berbasis register dan dapat mengeksekusi kelas (class) yang telah terkompilasi pada compier bahasa java, kemudian di transformasikan kedalam native format dengan menggunakan Tool “dx” yang telah terintegrasi. Kita mungkin telah mengenal JavaVM (Java Virtual Mesin), yang saat ini bisa kita temukan pada setiap komputer desktop. Berbeda dengan DalvikVM, JavaVM berbasis stack (Stack-based Virtual Machine), DalvikVM memiliki keunggulan dengan menggunakan Registred Based, ini karena pada prosesor perangkat genggam telah dioptimasi untuk eksekusi berbasis register . Android saat ini tidak hanya berjalan pada handphone, beberapa vendor menanamkan Android pada tablet, Internet Tablet, E-Book Reader, Laptop, dan gadget lainnya.

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi

oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

Beberapa versi dari android yang sudah diluncurkan diantaranya:

1. Andorid versi 1.1

   Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 (Cupcake)

   Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. Android versi 1.6 (Donut)

   Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan control applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. Android versi 2.0/2,1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* – aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

5. Android versi 2.2 (Froyo: Frozen Youghurt)

   Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. Android versi 2.3 (Gingerbread)

   Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahanperubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0 / 3.1 (HoneyComb)

   Android *Honeycomb* dirancang khusus untuk tablet.Android versi ini mendukung ukuran layar yang lebih besar. *User nterface* pada *Honeycomb* juga berbeda karena sudah didesain untuktablet. *Honeycomb* juga mendukung multi prosesor dan jugaakselerasi perangkat keras (*hardware*) untuk grafis. Versi ini dirilis pada 22 Februari 2011.

8. Android versi 4.0 (Ice Cream Sandwich)

   Android versi 4.0 Ice Cream Sandwich atau disingkat ICS adalah android pertama yang mempunyai fitur baru membuka kunci dengan pengenalan wajah.android ini juga dikalim mampu menghasilan interface yang lebih halus dan bersih. terdapat juga beberapa penambahan font baru pada android versi ICS ini.

9. Android versi 4.1 (Jelly Bean)

   Jelly Bean-Android versi 4.1 yang diluncurkan pada acara Google I/O membawa fitur fitur baru yang menawan, beberapa fitur yang diperbaharui dalam sistem operasi ini antara lain, pencarian dengan menggunakan Voice Search yang lebih cepat, informasi cuaca, lalu lintas, hasil pertandingan olahraga yang cepat dan tepat, selain itu versi 4.1 ini juga mempunyai fitur keyboard virtual yang lebih baik. Permasalahan umum yang sering ditemui pengguna Android adalah baterai, namun Baterai dalam sistem Android Jelly Bean versi 4.1 ini diklaim cukup hemat.

10. Android versi 4.2 (Jelly Bean)

    Android versi 4.2 ini merupakan versi terbaru dari versi versi Android sebelumnya. Jelly Bean V.4.2 diklaim lebih pintar dan inovatif dibandingkan dengan pendahulunya. Beberapa fitur yang diperbaharui dalam sistem operasi ini antara lain Notifications, Google Assistant, Face unlock dengan Liveness Check, Barrel Roll, Smart Widget, Google Now, Tehnologi Project Butter untuk meningkatkan responsifitas yang sangat baik, Sistem operasi yang cepat dan Ringan, Full Chrome browser menjejalah internet menggunakan Google Chrome seperti pada PC.

11. Android versi 4.3 (Jelly Bean)

Beberapa bulan kemarin google mengeluarkan versi terbaru android jelly bean 4.3, terdapat beragam pembaruan fitur pada versi ini diantaranya adalah teknologi smart bluetooth yang membuat bluetooth aktif tanpa menguras baterai, Mendukung Open GL ES 3.0, performa grafis yang lebih bagus dan realistis. kecepatan kinerja yang luar biasa cepat serta grafis memukau atau lebih halus.

12. Android versi 4.4 (Kitkat)

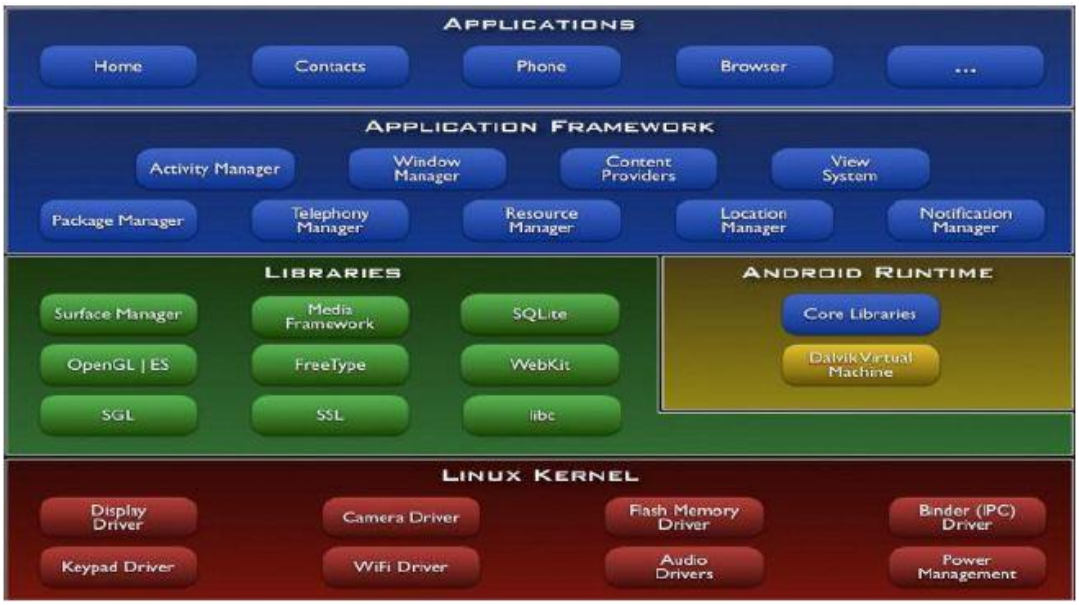
    Fitur-fitur baru Android versi 4.4 (Kitkat) ini antara lain fitur SMS yang terintegrasi langsung kedalam Aplikasi Google Hangouts. Terdapat fasilitas Could Printing, dimana pengguna dapat Printing secara nirkabel / mengirim perintah ke Laptop / PC yang terhubung dengan printer.Desain ikon dan tema yang lebih unik dan realistik. Mendengarkan perintah suara dari Google Now tanpa menguras daya baterai. Navigasi dan statusbar yang mengalami pembaruan. Interface yang sangat halus. Bisa mengakses aplikasi kamera dari layar yang terkunci.

13. Android versi 5.0 (Lolipop)

    Kelebihan yang di dapat pada Lollipop ialah pada sisi pengaturan untuk notifikasi, pengguna dapat mengatur apa saja yang ingin di tampilkan. Daya baterai menjadi perhatian khusus oleh Google di OS Lollipop, Google mengklaim dengan fitur penghemat baterai yang baru, umur dari baterai dapat bertahan hingga 90 lebih lama dibanding sistem operasi sebelumnya “KitKat 4.4″. Dari sektor keamanan, selain kunci PIN, password, dan pattern, Google menambahkan fitur baru yaitu keamanan Smart Lock. Kerja dari fitur ini ialah menjadikan objek-objek lain sebagai kunci pembuka ponsel, seperti jam tangan, mobil dan sebagainya.

2.9.2 Arsitektur Android

Berikut ini menunjukkan komponen utama dari sistem operasi Android.



Gambar 2. 19 Arsitektur Android

1. Linux Kernel

   Linux Kernal adalah layer dimana inti dari operating sistem dari Android itu berada. Berisi file – file system yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem –sistem operasi Android lainnya. Linux kernel yang digunakan Android adalah linux kernel release 2.6.

2. *Libraries*

   Libraries ini adalah layer dimana fitur – fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas kernel, Layer ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL, serta :

   a. *Libraries* media untuk pemutaran media audio dan *video*
   b. *Libraries* untuk manajemen tampilan
   c. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D
   d. *Libraries* SQLite untuk dukungan *database*
   e. *Libraries* SSL dan Webkit terintegrasi dengan web browser dan security
   f. *Libraries* LiveWebcore mencakup modern web browser dengan engine embedded web view
   g. *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API‟s.

3. Android*Runtime*

Android terdiri dari satu *set core libraries* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core libraries* bahasa pemrograman *Java*. Setiap aplikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari mesin virtual Dalvik (Dalvik VM). Dalvik dirancang agar perangkat dapat menjalankan multiple VMs secara efisien. Mesin Virtual Dalvik mengeksekusi file dalam Dalvik executable (.dex), sebuah format yang dioptimalkan untuk memori yang kecil. Dalvik VM berbasis, berjalan, dan dikompilasi oleh compiler bahasa *Java* yang telah ditransformasikan ke dalam .dex format oleh tool "dx" yang telah disertakan. Dalvik VM bergantung pada kernel Linux untuk berfungsi seperti threading dan manajemen memori tingkat rendahnya.

4. *Framework* Aplikasi

   Pengembang memiliki akses penuh menuju API *framework* yang sama dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali *(reuse)* dengan mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini (sesuai dengan batasan keamanan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna. Semua aplikasi merupakan rangkaian set layanan dan sistem, termasuk:

   a. *Views* yang kaya dan *extensible* yang dapat digunakan untuk membangun aplikasi, termasuk *list,grids*, kotak teks, tombol, dan bahkan sebuah *embedded web*.
   b. *Content Provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak telepon) atau membagi data tersebut.
   c. *Resource Manager*, yang menyediakan akses ke kode nonsumber lokal seperti string, gambar, dan tata letak file.
   d. *Notifications Manager* yang memungkinkan semua aplikasi menampilkan *alert* yang bisa dikostumisasi di dalam status bar.
   e. *Activity Manager*yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack*.

5. *Fitur Applications*

Android telah menyertakan aplikasi inti seperti email *client*, SMS, kalender, peta, *browser*, kontak, dan lainnya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman *Java*. Pada layer inilah *developer* atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada Android semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan pada layer aplikasi dengan menggunakan *library* API yang sama. Ini berarti semua aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses seluruh kemampuan handset, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan platform Android ini, *Programmer* atau *Developer* secara penuh akan bisa mengkostumisasi perangkat Androidnya.

2.9.3 Fitur Aplikasi Android

Android telah menyertakan aplikasi inti seperti email elient, SMS, kalender, peta, browser, kontak, dan lain-nya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman Java. Pada layer inilah developer atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada android semua aplikasi baik aplikasi inti (native) maupun aplikasi pihak ketiga berjalan pada layer aplikasi yang dibuat untuk android akan memiliki akses yang setara dalam mengakses seluruh kemampuan handset, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan platform android ini, Programmer atau developer secara penuh akan bisa mengkostumisasi perangkat androidnya.

2.11 C#

C# (dibaca: **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.

2.11.1. Sejarah C#

Pada akhir dekade 1990-an, *Microsoft* membuat program *Microsoft Visual J++* sebagai sebuah langkah percobaan untuk menggunakan *Java* di dalam sistem operasi *Windows* untuk meningkatkan antarmuka dari *Microsoft Component Object Model (COM)*. Akan tetapi, karena ada masalah dengan pemegang hak cipta bahasa pemrograman *Java, Sun Microsystems, Microsoft* pun menghentikan pengembangan *J++*, dan beralih untuk membuat pengganti *J++*, *compiler* dan mesin virtualnya sendiri dengan menggunakan sebuah bahasa pemrograman yang bersifat *general-purpose*. Untuk menangani proyek ini, *Microsoft* merekrut Anders Helsberg, yang merupakan mantan karyawan *Borland* yang membuat bahasa *Turbo Pascal*, dan *Borland Delphi*, yang juga mendesain *Windows Foundation Classes (WFC)* yang digunakan di dalam *J++*. Sebagai hasil dari usaha tersebut, *C#* pun pertama kali diperkenalkan pada bulan Juli 2000 sebagai sebuah bahasa pemrograman modern berorientasi objek yang menjadi sebuah bahasa pemrograman utama di dalam pengembangan di dalam *platform Microsoft .NET Framework.*

Pengalaman Helsberg sebelumnya dalam pendesain bahasa pemrograman seperti *Visual J++, Delphi, Turbo Pascal*) dengan mudah dilihat dalam sintaksis bahasa *C#*, begitu pula halnya pada inti *Common Language Runtime* (CLR). Dari kutipan atas *interview* dan makalah-makalah teknisnya ia menyebutkan kelemahan-kelemahan yang terdapat pada bahasa pemrograman yang umum digunakan saat ini, misalnya *C++, Java, Delphi,* ataupun *Smalltalk*. Kelemahan-kelemahan yang dikemukakannya itu yang menjadi basis CLR sebagai bentukan baru yang menutupi kelemahan-kelemahan tersebut, dan pada akhirnya memengaruhi desain pada bahasa *C#* itu sendiri. Ada kritik yang menyatakan *C#* sebagai bahasa yang berbagi akar dari bahasa-bahasa pemrograman lain. Fitur-fitur yang diambilnya dari bahasa *C++* dan *Java* adalah desain berorientasi objek, seperti *garbage collection*, *reflection*, akar kelas (*root class*), dan juga penyederhanaan terhadap pewarisan jamak (*multiple inheritance*). Fitur-fitur tersebut di dalam *C#* kini telah diaplikasikan terhadap iterasi, properti, kejadian (*event*), *metadata*, dan konversi antara tipe-tipe sederhana dan juga objek.

C# didisain untuk memenuhi kebutuhan akan sintaksis *C++* yang lebih ringkas dan *Rapid Application Development* yang 'tanpa batas' (dibandingkan

dengan RAD yang 'terbatas' seperti yang terdapat pada *Delphi* dan *Visual Basic*). Agar mampu mempromosikan penggunaan besar-besaran dari bahasa *C#*, *Microsoft,* dengan dukungan dari *Intel Corporation* dan *Hewlett-Packard*, mencoba mengajukan standardisasi terhadap bahasa *C#*. Akhirnya, pada bulan Desember 2001, standar pertama pun diterima oleh *European Computer Manufacturers Association* atau *Ecma International* (ECMA), dengan nomor standar ECMA-334. Pada Desember 2002, standar kedua pun diadopsi oleh ECMA, dan tiga bulan kemudian diterima oleh *International Organization for Standardization* (ISO), dengan nomor standar ISO/IEC 23270:2006

2.11.2. Kelebihan C#

1. *Flexible: C#* program dapat dieksekusi di mesin computer sendiri atau di transmiskan melalu web dan di eksekusi di komputer lainnya.
2. *Powerful*: *C#* memiliki sekumpulan perintah yang sama dengan *C++* yang kaya akan fitur yang lengkap, tetapi dengan gaya bahasa yang lebih diperhalus sehingga memudahkan penggunanya.
3. *Easier to use*: *C#* memodifikasi perintah yang sepenuhnya sama dengan *C++* dan memberitahu dimana letak kesalahan bila ada *error* dalam aplikasi, hal ini dapat mengurangi waktu dalam mencari *error.*
4. *Visually oriented*: *.NET library code* yang digunakan oleh *C#* menyediakan bantuan yang dibutuhkan untuk membuat tampilan yang *complicated* dengan *frames, dropdown , tabbed windows, group button , scroll bar , background image ,* dan lainnya
5. *Secure*: semua bahasa pemprograman yang digunakan untuk kebutuhan internet harus memiliki *security* yang benar-benar aman untuk menghindari aksi kejahatan dari pihak lain seperti hacker , *C#* memiliki segudang fitur untuk menanganinya.
6. *Memory management* lebih mudah karena adanya *garbage collector,* yang membebaskan *memory* secara otomatis sehingga dapat mencegah *memory leak. Type safe*, konversi implisit dari tipe data hanya *support* pada turunan dan operasi dari tipe data yang melebar (misal dari *int* ke *long*, tapi kalau *int* ke *short* tidak bisa) dan ini dideteksi *pas compile*.

7. Banyak fungsi yang tersedia di *Base Class Library .NET Framework*. *.NET Framework* berkembang cepat dan semakin banyak fitur yang membuat produktivitas kita bertambah, misal: *linq*.
8. Untuk pengembangan aplikasi bisnis/umum atau *enteurprise*, penggunaan *C#* akan lebih produktif daripada bila menggunakan *C++*. Bahasa *C#* masih merupakan turunan dari bahasa *C*, tetapi seolah dibuat lebih mudah dan produktif seperti *Visual Basic* dengan tetap mempertahankan fleksibilitas dan "*power*" dari bahasa *C*.
9. Kabarnya *Microsoft* sedang mengembangkan sistem operasi dengan *code name* "Singularity" / "Midori" yang akan menggantikan *Windows*. Sistem operasi ini, *API*-nya ditulis dengan *C#* dan merupakan *managed code* seperti *.Net Framework* saat ini, sehingga tentunya akan mengusung *C#* sebagai bahasa pemprograman standar. Hal ini akan menambah lagi kelebihan *C#*

2.11.3. Kekurangan C#

1. Banyaknya operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
2. Bagi pemula pada umumnya akan kesulitan menggunakan *pointer*.

2.12 WEB Service

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung *interoperabilitas* dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format *XML*, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa *compiler*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *web service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang

terdapat di dalamnya. Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan *object* yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-*upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di *port 80* yang merupakan protokol standar *HTTP*, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

2.12. 1 Arsitektur Web Service

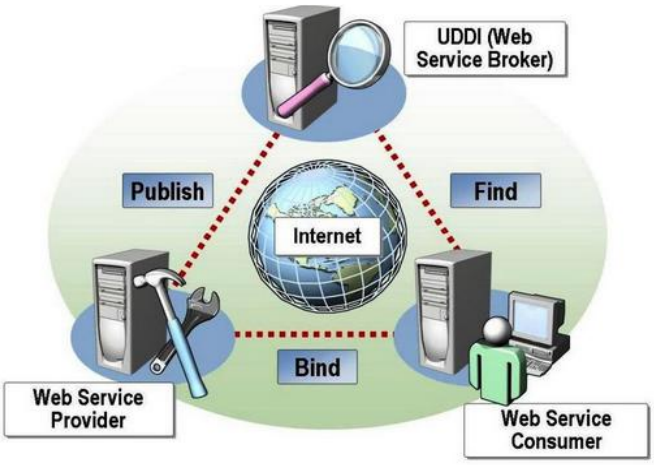
Web service memiliki tiga entitas dalam arsitekturnya [11], yaitu:

1. *Service Requesto* (peminta layanan)

   Service Requestor: Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan dan menggunakan layanan tersebut.
2. *Service Provider* (penyedia layanan)

   Berfungsi untuk menyediakan layanan(*service*) dan mengolah *registry* layanan-layanan tersebut agar dapat tersedia.
3. *Service Registry* (daftar layanan)

   Berfungsi sebagai lokasi *central* yang mendeskripsikan semua layanan setelah di-*register*.



Gambar 2. 20 Arsitektur Web Service

2.12. 2 Operasi-operasi padaWeb Service

Secara umum, *web service* memiliki tiga operasi yang terlibat di dalamnya [11] yaitu :

1. *Publish/Unpublish*: Menerbitkan/menghapus layanan ke dalam atau dari *registry*.
2. *Find*: *Service requestor* mencari dan menemukan layanan yang dibutuhkan.
3. *Bind*: *Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan *binding* ke *service provider* untuk melakukan interaksi dan mengakses layanan(*service*) yang disediakan oleh *service provider*.

2.12. 3 Komponen-komponen Web Service

Web service secara keseluruhan memiliki empat *layer* komponen (lihat tabel 2.1), yaitu:

1. *Layer* 1: Protokol internet standar seperti *HTTP, TCP/IP*
2. *Layer* 2: *Simple Object Access Protocol (SOAP)*, merupakan protokol akses objek berbasis *XML* yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. *Layer* 3: *Web Service Definition Language (WSDL)*, merupakan suatu standar bahasa dalam format *XML* yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.

Tabel 2. 1 *Layer* Komponen *Web Service*

Service Publication and Discovery (UDDI)
Service Description (WSDL)
XML Bassed Messaging (SOAP)
Common Internet Protocol (Http, TCP/IP)

2.13 Dropbox

Dropbox adalah layanan penyedia data berbasis web yang dioperasikan oleh Dropbox, Inc. Dropbox menggunakan sistem penyimpanan berjaringan yang memungkinkan pengguna untuk menyimpan dan berbagi data serta berkas dengan pengguna lain di internet menggunakan sinkronisasi data. Dropbox didirikan pada tahun 2007 oleh lulusan Massachusetts Institute of Technology (MIT) Drew Houston dan Arash Ferdowsi dengan modal awal yang didapat dari Y Combinator [23].

Dropbox menawarkan akun gratis yang dengan kapasitas data sebesar 2 GB dan akun berbayar dengan kapasitas sebesar 50 GB dan 100 GB, dan akun khusus kelompok dengan kapasitas data sebesar 1 TB atau lebih. Baik akun gratis atau berbayar memiliki fasilitas yang persis sama, yang membedakan hanyalah jumlah dari kapasitas penyimpanan yang ditawarkan. Menyediakan akun gratis bagi pengguna memakan banyak biaya bagi Dropbox, namun walaupun begitu Dropbox terus memberikan fasilitas akun gratis, karena hal ini mampu memberikan banyak keuntungan bagi Dropbox.

Bila dibandingkan dengan layanan serupa lainnya, Dropbox menawarkan jumlah pengguna yang relatif besar, dengan penggunaan sistem operasi yang bervariasi, baik untuk perangkat mobile ataupun desktop. Terdapat berbagai versi untuk berbagai sistem operasi, termasuk untuk Microsoft Windows, Mac OS X, dan Linux (resmi atau tidak resmi). Dan tersedia juga berbagai versi untuk

perangkat *mobile*, diantaranya Android, Windows Phone 7, iPhone, iPad, WebOS, dan Blackberry, dan klien yang berbasis web. Dropbox menggunakan model finansial Freemium, dan layanan gratisnya menyediakan 2 GB penyimpanan online gratis. Para pengguna yang menyarankan Dropbox ke orang lain bisa meningkatkan kapasitas penyimpanan hingga 8 GB.[8] Kompetitor utama dari Dropbox antara lain, Box.net, FilesAnywhere, CloudMe, CrashPlan, Egnyte, iCloud, Mozy, SpiderOak, SugarSync, TitanFile, Ubuntu One, Windows Live SkyDrive, Wuala dan ZumoDrive.

2.14 Tools Implementasi

Tools implementasi merupakan peralatan-peralatan yang dibutuhkan dalam membangun perangkat lunak. Adapun *tools* implementasi yang digunakan untuk membuat aplikasi *ARWayangGolek* adalah *VuforiaSDK* sebagai perangkat lunak pembangun (*Library*), *Unity3D* sebagai lingkungan pengembangan perangkat lunak, *Android SDK* sebagai alat bantu dan *API(Application Programming Interface)* untuk mulai mengembangkan aplikasi pada platform android, dan *SQL* sebagai sistem manajemen basis data.

2.14.1 Android SDK (Software Development Kit)

Android SDK(Software Development Kit) adalah tools *API(Application Programming Interface)* yang diperlukan untuk memulai pengembangan suatu aplikasi pada *platform android* menggunakan bahasa pemrograman *Java*. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh *Google.* Saat ini disediakan *Android SDK* sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral, android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan aplikasi bawaan *Handphone/smartphone* [15].

2.14.2 Vuforia SDK

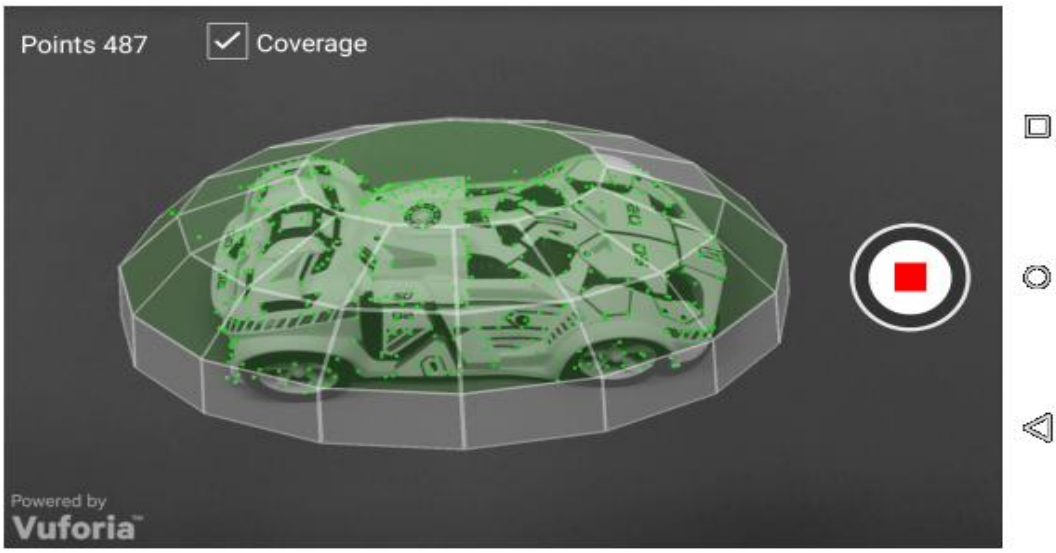
Vuforia adalah *Augmented Reality Software Development Kit (SDK)* untuk perangkat *mobile* yang memungkinkan pembuatan aplikasi *Augmented Reality*. *Vuforia SDK* juga tersedia untuk digabungkan dengan *game engine unity3D*, yaitu

bernama *Vuforia AR Extension for Unity*. *Vuforia* merupakan sebuah *SDK* yang disediakan oleh *Qualcomm* untuk membantu para developer dalam membuat aplikasi-aplikasi *Augmented Reality* (AR) pada perangkat *mobile phones* (*iOS, Android*). *Vuforia SDK* sudah sukses dipakai di beberapa aplikasi-aplikasi *mobile* untuk kedua *platform* tersebut.

Vuforia SDK memberikan cara berinteraksi yang memanfaatkan kamera *mobile phones* untuk digunakan sebagai perangkat masukan, sebagai mata elektronik yang mengenali penanda (*marker*) tertentu, sehingga di layar bisa ditampilkan perpaduan antara dunia nyata dan dunia yang digambar oleh aplikasi (*Augmented Reality*). Dengan kata lain, *Vuforia* adalah sebuah *SDK* untuk computer vision based *Augmented Reality* [16].

2.14.2.1 Vuforia Object Scanner

Vuforia object scanner adalah sebuah aplikasi *mobile smartphone Android* yang digunakan untuk membuat *object target* dari suatu objek di dunia nyata. Dengan aplikasi ini, pengguna dapat mengevaluasi kualitas pelacakan suatu objek di dunia nyata sebelum membangunnya menjadi object target pada aplikasi *augmented reality* [17].



Gambar 2. 21 Object Scanning

Ada 3 langkah untuk membuat sebuah *Object Target* :

1. *Setup*
2. *Scanning Object*

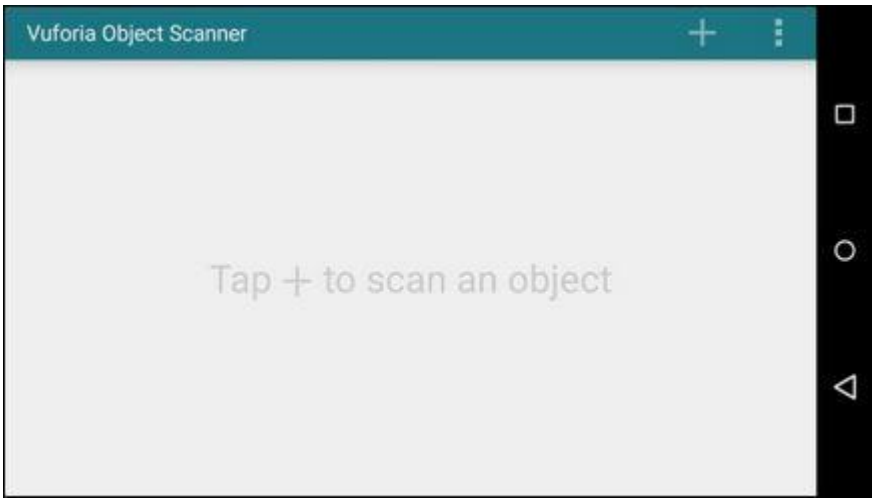
3. *Transfer file data Object*
4. *Upload file data Object*

1. Setup

Langkah pertama yang harus dilakukan untuk membuat sebuah object target adalah melakukan *Setup*. Berikut adalah langkah-langkah yang harus dilakukan [18] :

a. Jalankan Aplikasi Vuforia Object Scanner

Aplikasi akan menampilkan halaman utama dari aplikasi. Halaman utama aplikasi memiliki menu untuk me-*resume* proses *scanning* yang pernah dilakukan sebelumnya atau menu untuk memulai proses *scanning* yang baru.

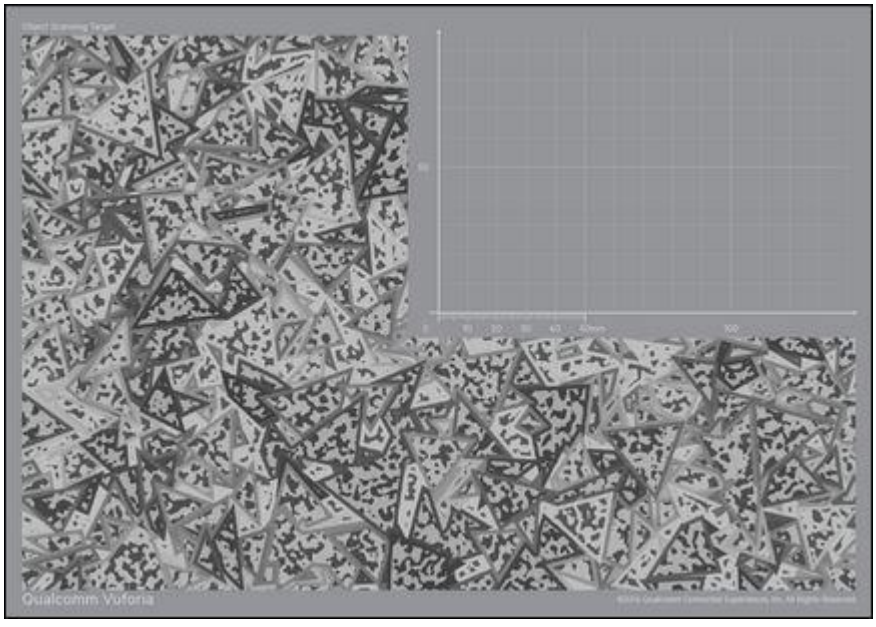


Gambar 2. 22 Tampilan Menu Utama Sebelum Ada Proses Scan



Gambar 2. 23 Tampilan Menu Utama Sesudah Ada Proses Scan

b. Printout *Object Scanning Target*. *Object Scanning Target* digunakan untuk memudahkan aplikasi dalam mengetahui keberadaan objek yang akan di scan dari posisi asal. Posisi asal ini diwakili oleh titik (0,0,0) pada pojok kiri bawah kotak *Object Scanning Target.* Tujuan dari *Object Scanning Target* ini adalah untuk memperkirakan besar skala sebuah objek yang akan di-*scan*.



Gambar 2. 24 Object Scanning Target

2. Scanning Object

Scanning Object adalah sebuah proses untuk mengenali suatu objek dunia nyata yang akan dijadikan *Object Target.* Tujuannya adalah untuk meng-*capture* bagian-bagian sebuah objek dari segala sudut agar dapat dengan mudah dilacak keberadaannya. Berikut langkah-langkah yang dilakukan dalam proses *scanning Object* [19]*:*

a. Letakkan Objek pada kotak *Object Scanning Target.*



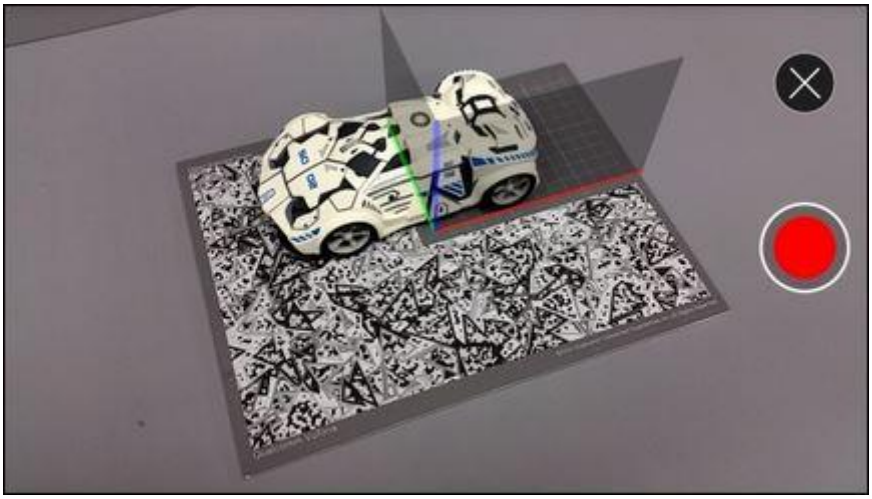
Gambar 2. 25 Cara Meletakkan Sebuah Objek pada Object Scanning Target

Aplikasi *VuforiaObjectScanner* akan mendeteksi keberadaan objek. Posisi asal target diwakili oleh titik (0,0,0) pada pojok kiri bawah kotak *Object Scanning Target.*



Gambar 2. 26 Aplikasi Mendeteksi Keberadaan Objek

Objek yang memasuki daerah fitur target tidak akan dimasukkan kedalam data *Object.*



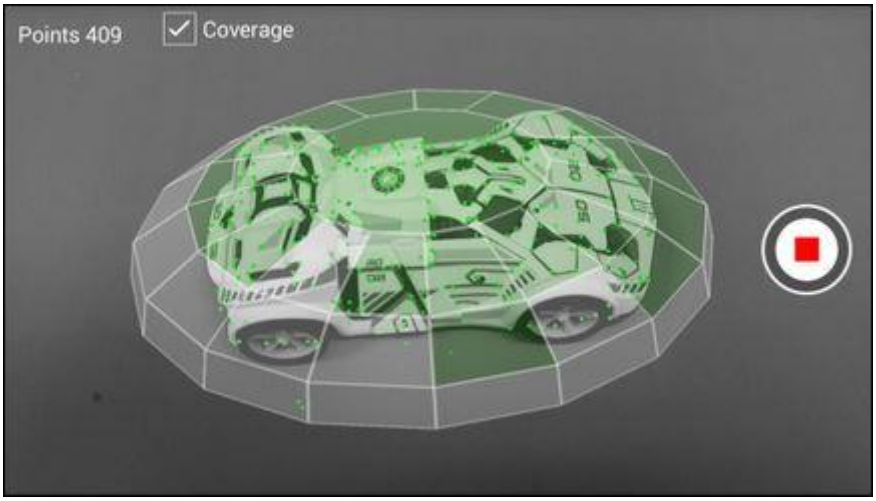
Gambar 2. 27 Objek yang Memasuki Daerah Fitur Target

b. Tekan tombol *Reccord* untuk memulai proses Scanning. Jangan pindah posisi objek ketika Proses scanning sedang berangsung.



Gambar 2. 28 Proses Scanning Object

c. Pindahkan kamera untuk meng-capture bagian-bagian objek. Ketika bagian sebuah objek berhasil di-*capture*, segi yang sesuai akan berubah warna menjadi hijau dan jumlah point akan bertambah.



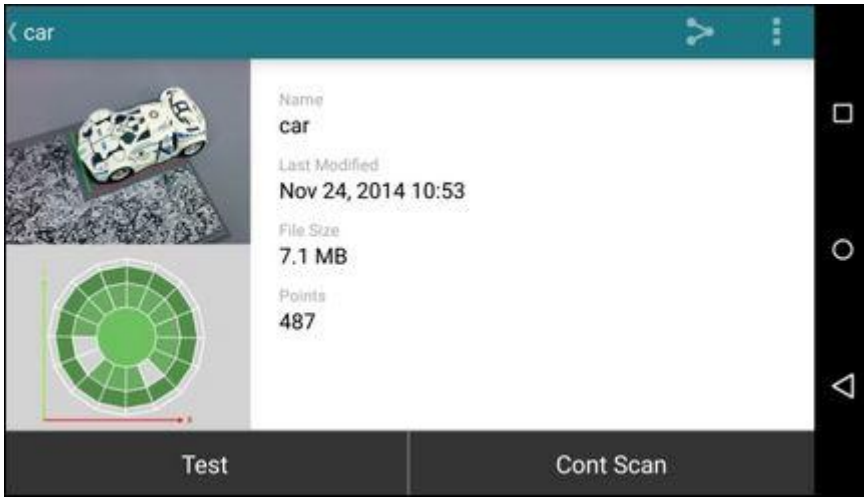
Gambar 2. 29 Proses Scanning Object

d. Ketika proses pemindaian (*Scanning*) objek selesai, tekan kembali tombol *Record* untuk menghentikan proses *Scanning* tersebut. Aplikasi akan menampilkan menu untuk menginputkan nama hasil dari proses *Scanning*. *Input*-kan nama dan tekan tombol Ok.



Gambar 2. 30 Tampilan Menu Untuk Nama Hasil Scanning

e. Aplikasi akan menampilkan hasil *Scanning*, terdapat dua menu pilihan untuk melanjutkan proses scanning atau men-*test* hasil scanning.



Gambar 2. 31 Tampilan Menu Hasil Proses Scanning

f. Tekan Tombol *Test* untuk men-*test*kualitas pelacakan objek hasil *scanning*. Aplikasi akan menampilkan *Object Augmented Reality* berbentuk *cube* (balok) apabila objek berhasil dilacak. Jika aplikasi tidak menampilkan *Object Augmented Reality*, hal ini menunjukan bahwa kualitas pelacakan objek hasil scanning rendah [20]. Untuk meningkatkan kualitas pelacakan objek hasil scanning, maka pilih menu *Cont Scan* untuk melanjutkan proses *Scanning*.



Gambar 2. 32 Test Objek Hasil Scanning

3. Transfer File Data Object

Setelah melakukan proses *scanning object*, langkah yang dilakukan selanjutnya adalah mentransfer data objek hasil proses *scanning*. Transfer data objek hasil *scanning* bertujuan untuk memindahkan data dari aplikasi ke tempat penyimpanan berikutnya sebelum kemudian dilakukan proses *upload*.

Untuk mentransfer data hasil scanning, kembali ke menu utama aplikasi *VuforiaObjectScanner*, pilih data objek hasil *scanning* yang akan ditransfer, lalu tekan tombol *share*. Aplikasi akan menampilkan pilihan kemana file akan di-*share*. *File data object* berformat (*.od). pindahkan *file data object* ke *PC* untuk memudahkan proses *upload*.

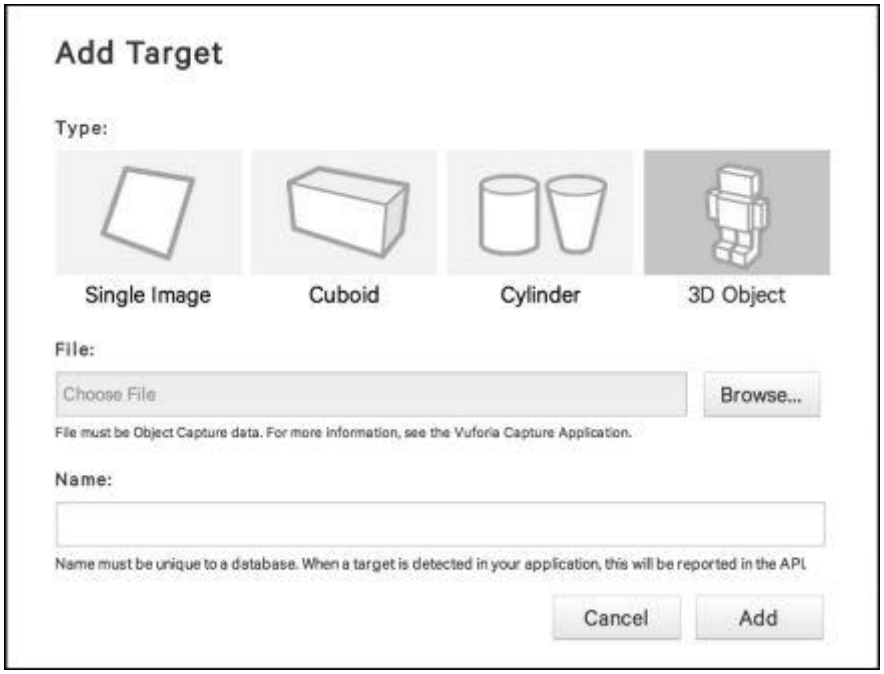


Gambar 2. 33 Tampilan Menu Share pada Aplikasi VuforiaObjectScanner

4. Upload File Data Object

Upload file data object adalah sebuah proses yang dilakukan dengan tujuan untuk membuat sebuah *database* yang berisikan data *object target* yang akan digunakan pada aplikasi. Berikut langkah-langkah untuk meng-*upload file data object* [21]*:*

a. Sebelum meng-*upload file data object* ke situs *vuforia*, terlebih dahulu buat *Account* untuk dapat *login* ke situs vuforia.
b. Pilih menu Develop, pilih Target Manager, lalu buat sebuah *database* untuk menampung *file data object.*
c. Pilih *Add Target*, lalu pilih *3D Object, browse file data object* yang akan di-*upload, input* nama, lalu pilih *Add.*



Gambar 2. 34 Tampilan Menu Add Target di situs Vuforia

d. Setelah *file data object* berhasil di-*upload*, pilih *download databasepilih* format database *(SDK atau Unity)*sesuai dengan kebutuhan, lalu pilih *download.*



Gambar 2. 35 Tampilan Menu Download Database di Situs Vuforia

2.14.3 Unity3D

Unity3D adalah sebuah *game engine* yang berbasis *cross-platform*. *Unity3D* dapat digunakan untuk membuat sebuah *game* yang bisa digunakan pada perangkat komputer, *smartphone*(android, iPhone), *PlayStation3*, dan bahkan *X-BOX*. *Unity3D* adalah sebuah *tool* yang terintegrasi untuk membuat *game*, arsitektur bangunan atau simulasi. *Unity3D* bisa untuk *game* PC dan *game Online*.

Untuk *game Online* diperlukan sebuah *plugin*, yaitu *Unity Web Playe*r, sama halnya dengan *Flash Player* pada *Browser*. *Unity3D* tidak dirancang untuk proses desain atau *modelling*, dikarenakan *unity3D* bukan *tool* untuk mendesain. Jika ingin mendesain, dapat menggunakan *tool 3D editor* seperti *3dsmax* atau *Blender*. Banyak hal yang bisa dilakukan dengan *unity3D*, ada fitur audio *reverb zone*, *particle effect*, dan *Sky Box* untuk menambahkan langit.

Fitur *scripting* yang disediakan *unity3D*, mendukung beberapa bahasa pemrograman seperti *JavaScript*, *C#*, dan *Boo*. Dengan menggunakan *unity3D, efect Flexible and EasyMoving*, *rotating*, dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *Duplicating, removing*, dan *changing properties*. *Visual Properties Variables* yang di definisikan dengan *scripts* ditampilkan pada *Editor*. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker*. *Unity3D* berbasis *.NET*, artinya penjalanan program dilakukan dengan *Open Source .NET platform, Mono* [22].