BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan SMK Negeri 2 Kota Bandung

SMK Negeri 2 Kota Bandung Merupakan sekolah menengah kejuruan nengeri yang berada di Kota Bandung, Jawa Barat, Indonesia. Masa pendidikan di SMK Neger 2 Kota Bandung ditempuh dalam waktu tiga tahun pelajaran, mulai dari kelas X hingga kelas XII, seperti pada umumnya masa pendidikan sekolah menengah kejuruan di Indonesia. Kejuruan yang ada diantaranya adalah; TKI, Teknik Mesin. [15]

2.1.1.Sejarah SMK Negeri 2 Kota Bandung

Berdasarkan dari Surat Putusan Menteri Pendidikan, Pengadjaran, dan Kebudajaan No.3835/B.II.1951 tertanggal 30 Agustus 1951, diputuskan untuk dibuka : Sekolah Teknik Menengah (STM) Negeri (Bagian Bangunan, Mesin, dan Listrik) di Bandung. Maka berdasarkan surat keputusan tersebut didirikan STM Negeri I Bandung pada tahun 1951 di jalan DR.Wahidin No.2 Bandung dipimpin oleh : W.G.Zweekhorst.

Berdasarkan surat keputusan kepala kanwil departemen pendidikan dan kebudayaan propinsi Jawa Barat No.148/102.Kep/R.77 tertanggal 16-12-1977, memutuskan untuk memindahkan STM Negeri 1 Bandung ke Jalan Ciliwung No.4 Bandung. Pada saat itu Jalan Ciliwung no.4 pada awalnya adalah sebuah ST Negeri 1 yang didirikan sejak zaman belanda (tahun tidak tercatat) dan STM Negeri IV Bandung. Kemudian STM Negeri IV Bandung dipindahkan ke Jalan Buah Batu Bandung.

Kepindahan STM Negeri 1 Bandung secara bertahap mulai tahun ajaran 1978/1979 dan secara keseluruhan berakhir pada tahun ajaran 1979/1980 secara resmi STM Negeri 1 Bandung pindah dari Jalan DR. Wahid No.2 ke Jalan Ciliwung No.4 Bandung. Perubahan nama STM Negeri 1 menjadi SMK Negeri 2 Bandung. Perubahan nama dari STM Negeri 1 menjadi SMK Negeri 2 Bandung berdasarkan SK Mendikbud Nomor 036/O/1997, bulan Oktober 1997, angka indek 1,2,3 diambil dari umur sekolah dihitung dari sejak berdirinya sekolah.[15]

2.1.2 Logo SMK Negeri 2 Kota Bandung

Logo atau lambang adalah suatu identitas dari sebuah sekolah, hampir semua sekolah memiliki logo untuk mempertegas keberadaan, begitu juga SMK Negeri 2 Kota Bandung yang memiliki logo sebagai berikut:



Gambar 2.1 Logo SMK Negeri 2 Kota Bandung

Sumber : http://smkn2bandung.sch.id

2.1.3 Profil Singkat SMK Negeri Kota 2 Kota Bandung

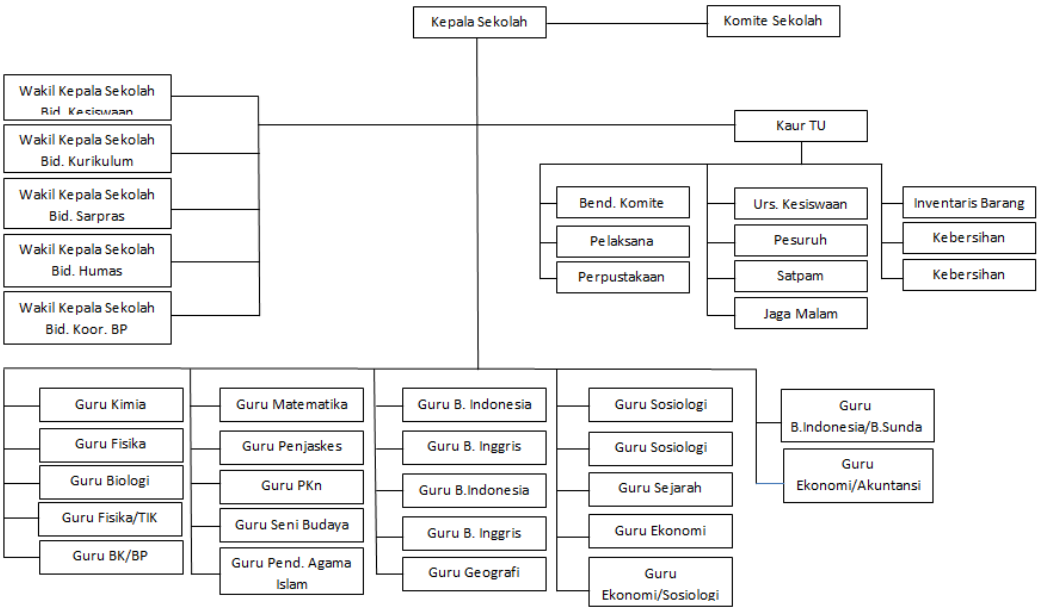
Informasi umum sekolah sebagai berikut:

2.1.4 Struktur Organisasi

Struktur organisasi merupakan suatu kerangka dalam manajemen sekolah agar suatu sekolah dapat berjalan sebagaimana mestinya sesuai dengan tujuan pertaman kali berdirinya perusahaan/organisasi tersebut. Struktur organisasi memuat tatanan kerja yang ada di sekolah tersebut yang menyankut tugas, tanggung jawab dan wewenang. Semuanya ada agar tercipta hubungan yang baik lebih dinamis dan masing – masing mempunyai tugas dan tanggung jawab sendiri juga mempunyai rasa hormat terhadap yang lain baik itu berkedudukan diatas, dibawah atau setingkat. Tidak berjalannya struktur organisasi dengan baik akan

mengarahkan perusahaan tersebut pada kehancuran maka dari itu struktur organisasi merupakan faktor penting dalam suatu sekolah.

Struktur organisasi merupakan suatu bentuk susunan keanggotaan yang membedakan jabatan masing – masing personil didalam suatu perusahaan, manfaat lainnya dengan dibuatnnya struktur organisasi dalam proses kerja. Pegawai perlu mengetahui struktur organisasi perusahaan, agar dapat melakukan kegiatan yang sesuai dengan fungsi dan tidak menyimpang dari yang telah ditentukan, sehingga pegawai mengetahui hak dan kewajibannya didalam sekolah tersebut. Berikut adalah gambar struktur organisasi SMK Negeri 2 Kota Bandung.



Gambar 2.2 Struktur Organisasi SMK Negeri 2 Kota Bandung

2.2 Landasan Teori

Landasan teori dimanfaatkan sebagai pemandu agar fokus penelitian sesuai dengan fakta di lapangan. Selain itu landasan teori juga bermanfaat untuk memberikan gambaran umum tentang latar penelitian dan sebagai bahan pembahasan hasil penelitian. Peneliti mengutip beberapa teori yang berhubungan dengan variabel-variabel penelitian. Dan teori-teori ini merupakan landasan dalam penelitian.

2.2.1 Computer Hardware

Computer Hardware (Perangkat Keras Komputer) adalah beberapa komponen fisik komputer yang bersifat dapat dilihat, diraba, dan berbentuk nyata yang membentuk suatu kesatuan sistem PC (*Personal Computer*). Fungsi dari perangkat – perangkat ini sangat beragam diantaranya sebagai alat *input*, alat *output*, perangkat pemrosesan data, dan juga sebagai alat tambahan dengan beragam fungsi tambahan yang tidak selalu ada di semua sistem PC.

Biasanya perangkat – perangkat ini dirakit, sebagian besar di masukkan ke dalam casing komputer dan sebagian lain berada di luar casing. Berikut merupakan gambar macam – macam perangkat keras komputer dapat dilihat pada gambar 2.3 [1] :



Gambar 2.3 Macam – Macam Perangkat Keras Komputer

2.2.2 Augmented Reality

Augmented Reality (AR) adalah sebuah istilah untuk lingkungan yang menggabungkan dunia nyata dan dunia virtual yang dibuat oleh komputer sehingga batas antara keduanya menjadi sangat tipis. Ronald Azuma pada tahun 1997 mendenisikan *Augmented Reality* sebagai sistem yang memiliki karakteristik sebagai berikut :

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

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

AR merupakan variasi dari *Virtual Environments* (VE), atau yang lebih dikenal dengan istilah *Virtual Reality* (VR). Teknologi VE membuat pengguna tergabung dalam sebuah lingkungan virtual secara keseluruhan. Ketika tergabung dalam lingkungan tersebut, pengguna tidak bisa melihat lingkungan nyata di sekitarnya. Sebaliknya, AR memungkinkan pengguna untuk melihat lingkungan nyata, dengan objek virtual yang ditambahkan atau tergabung dengan lingkungan nyata. Tidak seperti VR yang sepenuhnya menggantikan lingkungan nyata, AR sekedar menambahkan atau melengkapi lingkungan nyata.[2]

Tujuan utama dari AR adalah untuk menciptakan lingkungan baru dengan menggabungkan interaktivitas lingkungan nyata dan virtual sehingga pengguna merasa bahwa lingkungan yang diciptakan adalah nyata. Dengan kata lain, pengguna merasa tidak ada perbedaan yang dirasakan antara AR dengan apa yang mereka lihat/rasakan di lingkungan nyata. Dengan bantuan teknologi AR (seperti visi komputasi dan pengenalan objek) lingkungan nyata disekitar kita akan dapat berinteraksi dalam bentuk digital (*virtual*). Informasi tentang objek dan lingkungan disekitar kita dapat ditambahkan kedalam sistem AR yang kemudian informasi tersebut ditampilkan diatas layer dunia nyata secara realtime seolah-olah informasi tersebut adalah nyata. Informasi yang ditampilkan oleh objek virtual membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata. AR banyak digunakan dalam bidang-bidang seperti kesehatan, militer, industri manufaktur dan juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam. Berikut merupakan contoh dari penerapan teknologi *Augmented Reality* sederhana dapat dilihat pada Gambar 2.4.[2]



Gambar 2.4 Contoh Augmented Reality Sederhana

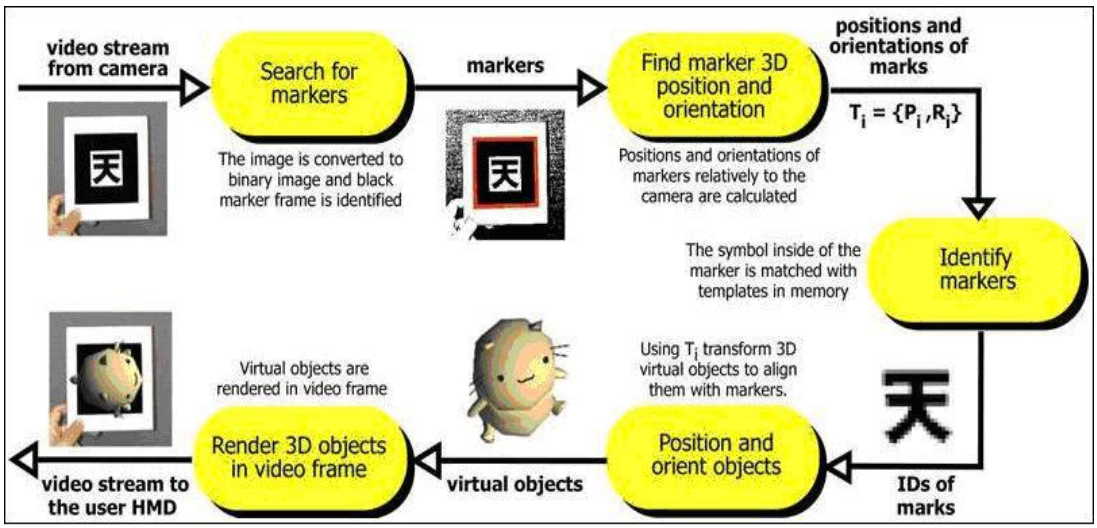
2.2.2.1 Sejarah Augmented Reality

Sejarah tentang *Augmented Reality* (AR) dimulai dari tahun 1957-1962, ketika seorang penemu yang bernama Morton Heilig, seorang sinematografer, menciptakan dan mempatenkan 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 AR untuk melakukan perbaikan pada pesawat boeing, dan pada tahun yang sama, LB Rosenberg mengembangkan salah satu fungsi sistem AR, yang disebut *Virtual Fixtures*, yang digunakan di Angkatan Udara AS Armstrong Labs, dan menunjukan manfaatnya pada manusia, dan pada tahun 1992 juga, Steven Feiner, Blair Maclntyre dan dorée Seligmann, memperkenalkan untuk pertama kalinya Major Paper untuk perkembangan Prototype AR. Pada tahun 1999, Hirokazu Kato, mengembangkan Unity di HITLab dan didemonstrasikan di SIGGRAPH, pada tahun 2000, Bruce. H. Thomas,mengembangkan ARQuake, sebuah mobile games AR yang ditunjukan di international symposium on wearable komputers. Pada tahun 2008, witiude AR Travel Guide, memperkenalkan Android G1 telephone yang berteknologi AR, tahun

2009, Saqoosha memperkenalkan FLUnity yang merupakan perkembangan dari Unity. FLUnity memungkinkan kita memasang teknologi AR di sebuah website, karena output yang dihasikan FLUnity berbentuk Flash. Ditahun yang sama, wikitude Drive meluncurkan sistem navigasi berteknologi AR di platform android. Tahun 2010, Acrossair menggunakan teknologi AR pada I-Phone 3GS [9].

2.2.2.2 Prinsip Kerja Augmented Reality

Sistem *Augmented Reality* bekerja berdasarkan deteksi citra dan citra yang digunakan adalah *marker*. Prinsip kerjanya sebenarnya cukup sederhana. *Camera* yang telah dikalibrasi akan mendeteksi marker yang diberikan, kemudian setelah mengenali dan menandai pola *marker*, *webcam* akan melakukan perhitungan apakah *marker* sesuai dengan *database* yang dimiliki. Bila tidak, maka informasi *marker* tidak akan diolah, tetapi bila sesuai maka informasi *marker* akan digunakan untuk me-*render* dan menampilkan objek 3D atau animasi yang telah dibuat sebelumnya.[3]



Gambar 2.5 Prinsip Kerja *Augmented Reality*

2.2.2.3 Metode Augmented Reality

Metode yang dikembangkan pada *augmented reality* saat ini terbagi menjadi dua metode, yaitu *Marker Based Tracking* dan *Markless Augmented Reality*.

a. *Marker Based Tracking*

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 tiga sumbu yaitu X, Y, dan Z. *Marker Based Tracking* ini sudah lama dikembangkan sejak 1980-an dan pada awal 1990-an mulai dikembangkan untuk penggunaan *Augmented Reality*.

b. *Markless Augmented Reality*

Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode "*Markerless Augmented Reality*", dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan elemen-elemen digital, dengan *tool* yang disediakan Qualcomm untuk pengembangan *Augmented Reality* berbasis *mobile device*, mempermudah pengembang untuk membuat aplikasi yang *markerless*.

Seperti yang saat ini dikembangkan oleh perusahaan *Augmented Reality* terbesar di dunia Total Immersion dan Qualcomm, mereka telah membuat berbagai macam teknik *Markerless Tracking* sebagai teknologi andalan mereka, seperti *Face Tracking*, 3D *Object Tracking*, dan *Motion Tracking*.

1) *Face Tracking*

Algoritma pada komputer terus dikembangkan, hal ini membuat 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 lain - lain.

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.

3) *Motion Tracking*

Komputer dapat menangkap gerakan, *Motion Tracking* telah mulai digunakan secara ekstensif untuk memproduksi film-film yang mencoba mensimulasikan gerakan.

4) GPS *Based Tracking*

Teknik GPS *Based Tracking* saat ini mulai populer dan banyak dikembangkan pada aplikasi *smartphone* (iPhone dan Android), dengan memanfaatkan fitur GPS dan kompas yang ada didalam *smartphone*, aplikasi akan mengambil data dari GPS dan kompas kemudian menampilkannya dalam bentuk arah yang kita inginkan secara *realtime*, bahkan ada beberapa aplikasi menampikannya dalam bentuk 3D. [4]

2.2.3 Markerless

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

Pada *Markerless* yang digunakan dan dikembangkan oleh Vuforia, dalam perancangannya, seolah-olah 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*). Sistem *Augmented Reality Display* yang digunakan adalah teknik *spatial display* dengan *screen display* (bisa menggunakan monitor ataupun proyektor). Secara garis besarnya, dalam perancangan aplikasi ini ada tiga bagian utama yaitu sebagai berikut :

1. Inisialisasi
2. *Tracking Marker*
3. *Rendering* Objek 3D

Markerless AR menggunakan AR tanpa pelacakan atau tracking tanpa penanda khusus . AR browser gravimetri adalah contoh dari *Markerless* AR sejak points of interest (POI) dan data lain yang ditampilkan di jendela kamera namun

data tersebut tidak terpasang atau dilacak untuk setiap objek visual tertentu . Banyak game mobile yang mengaku AR adalah permainan yang hanya menyalakan kamera sebagai latar belakang . Selain kamera menyediakan pengaturan lokal bagi pengguna , penggunaan kamera superfi sosial . Jadi , sekali lagi , beberapa akan mengklaim bahwa ini bukan AR nyata.



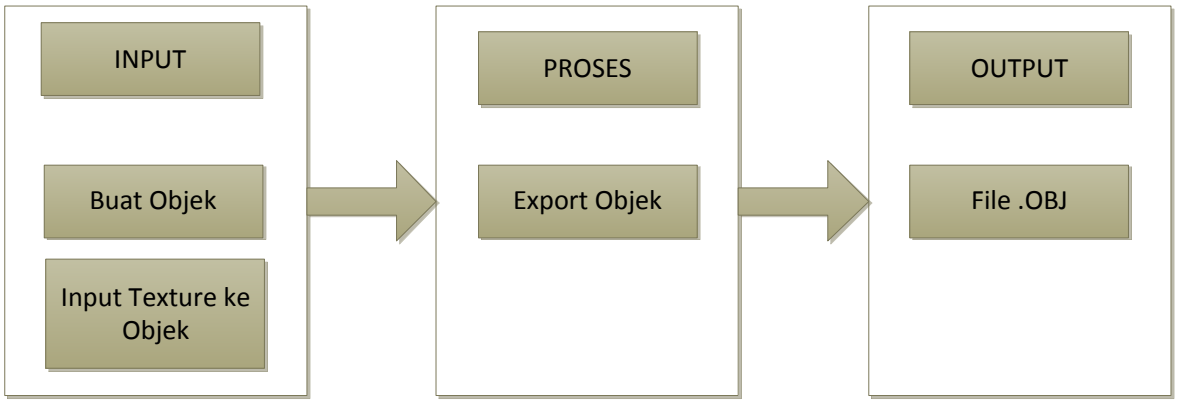
Gambar 2.6 Contoh *Markerless*

Gambar 2.10 menunjukkan menghibur permainan *AR Invaders iPhone* Soulbit7 itu , game ini menambahkan AR memutar ke *Space Invaders* game klasik dengan menempatkan pengguna di lingkungan mereka (courtesy of kemampuan *smartphone* kamera untuk memberikan latar belakang permainan) . Pelacakan *Markerless* adalah di mana AR digunakan untuk melacak benda-benda di dunia nyata tanpa menggunakan penanda khusus. Pengenalan wajah adalah contoh yang sangat baik . Anda mungkin akan terkejut melihat betapa sistem pengenalan wajah canggih adalah untuk perangkat mobile . *Polar Rose* , sebuah perusahaan Swedia , memamerkan prototipe pengenalan wajah untuk perangkat Android pada tahun 2009 . Prototipe , bernama *Augmented ID* , memungkinkan pengguna untuk menunjukkan ponsel Android -nya di wajah seseorang dan aplikasi akan membandingkan wajah ke wajah dalam database dalam upaya untuk fi nd pertandingan . Jika pertandingan ditemukan , aplikasi overlay profil media sosial subjek (*Twitter* , *Facebook* , *LinkedIn* , *LastFM* , dan sebagainya) . Teknologi ini sejak itu telah dibeli oleh *Apple*, sehingga memberikan indikasi dan jenis aplikasi *Markerless* AR kita akan lihat di masa depan . Sejak akuisisi , bebas

face recognition SDK tidak lagi tersedia untuk download . Namun, pada pameran dagang *Mobile World Congress* di Barcelona pada bulan Februari 2011 , sebuah perusahaan baru bernama *Viewdle* (www.viewdle.com) memamerkan gratis *face recognition* SDK mereka untuk pengembang *Android.*[10]

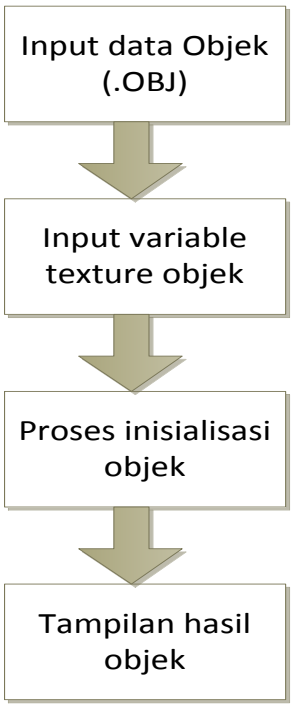
2.2.3.1 Inisialisasi Model 3D

Model 3D yang akan ditampilkan di-load terlebih dahulu. Agar aplikasi dapat menampilkan objek 3D tertentu tanpa merubah atau membangun ulang aplikasi, diperlukan sebuah file konfigurasi untuk menentukan objek 3D yang akan *di-load* sesuai dengan pola *Target* yang dideteksi.



Gambar 2.7 Proses pembentukan data objek 3D

Dalam proses pemodelan terdiri dari 3 langkah. Pertama kita buat objek 3D. Kedua, memasukan teksture sesuai dengan objek 3D dan ketiga mengexport objek yang sudah dirancang dan dibuat kedalam format *objek file* (*.OBJ).



Gambar 2.8 Proses Input Data Objek

a. Input data objek

Proses pertama yaitu menambahkan data *objek .package* ke dalam *library Vuforia*. File package ini berfungsi untuk memanggil bentuk objek yang sudah diexport untuk siap diimportkan oleh Unity.

b. Input variabel objek

Proses ini berfungsi untuk menambahkan tekstur pada objek, tekstur pada objek tidak bisa digunakan atau muncul sebelum ada penambahan variabel dan pengaturan kode yang ditambahkan pada function init 3Dmodel agar tekstur tersebut sesuai dengan model yang telah dibuat.

1) Penambahan *variabel teksture*.

Penambahan *variabel teksture* disesuaikan dengan jumlah dan nama material id pada file .package yang digunakan pada objek.

2) Penambahan kode *class variabel teksture*

Gambar atau tekstur yang ditambah disesuaikan dengan *variabel teksture*.

2.2.3.2 Target Yang Digunakan

Target yang digunakan pada analisis ini yaitu berupa *Target* Hardware Komputer bergambar *pattern* yang sudah ada, ini disebabkan karena aplikasi ini masih bersifat sementara belum di resmikan.

Setelah *Target* telah ada kemudian proses selanjutnya adalah menambahkan gambar *Target* yang telah dikonversi menjadi .package agar *Target* terdeteksi pada saat di-render.

2.2.3.3 Tracking Target

Vuforia memiliki kemampuan untuk mendeteksi gambar dan menghitung posisi gambar tersebut menggunakan webcam standar. Informasi posisi yang didapatkan akan dipergunakan untuk menempatkan objek atau model tiga dimensi atau video ke dalam posisi gambar atau *Markerless*. Ada lima langkah, dalam proses kerja *Tracking Target* di Unity.

1. Mengambil video dari *webcam*

Mendapatkan masukan video dari sebuah webcam adalah langkah awal yang harus dilakukan. Sistem mengolah dan menganalisis frame perframe video yang di-streaming secara real-time dan hasilnya berupa gambar digital yang akan digunakan untuk tahap berikutnya. Sebelum webcam digunakan, webcam harus dikaliberasi terlebih dahulu. Kaliberasi webcam merupakan bagian sangat penting dalam proses pengambilan input atau masukan video. Hal ini disebabkan oleh distorsi pada lensa webcam yang tiap-tiap kamera berbeda karakteristiknya. Tujuan dari kalibrasi webcam adalah untuk menghitung tingkat distorsi dari sebuah lensa webcam yang digunakan agar citra yang dihasilkan mendekati citra ideal.Proses kalibrasi gambar secara real time.

2. Binarisasi citra masukan (*thresholding*).

Langkah pertama pada aplikasi visi komputer yang terletak pada deteksi tepi adalah untuk men-*threshold* sumber citra atau disebut juga binarisasi seperti yang ditunjukkan mengkonversi citra ke citra binari sehingga memudahkan untuk komputasi. Sebuah citra binari dibuat dengan mengubah pixel yang lebih cerah daripada nilai threshold ke suatu warna, dan pixel yang lebih gelap daripada nilai *threshold* ke suatu warna lainnya (didenisikan sebagai *gray-scale* atau hitam-putih).



Gambar 2.9 Thresholding

Nilai *threshold* berada pada angka 0 - 255 dan secara *default*, *threshold* bernilai 100. Fungsi dari proses ini adalah untuk membantu sistem agar dapat mengenali bentuk segi empat dan pola di *Target* pada citra yang diterima. Nilai *threshold* dapat dirubah dan disesuaikan dengan kondisi cahaya disekitar *Target* untuk tetap membuat *Target* terlihat sebagai segi empat, karena ketika cahaya disekitar *Target* berkurang ataupun berlebih pada saat proses thresholding, sistem tidak dapat mendeteksi *Target*.

2.2.3.4 Pencocokan Pola

Setelah semua area persegi dan pola-pola gambar ditandai, Unity menganalisa citra yang berada di dalam persegi dan membandingkan polanya dengan sekumpulan pola yang telah ditentukan (pencocokan pola). Unity mengekstrak pola didalam persegi menggunakan transformasi homography.

Pembuatan *Target* dilakukan oleh pihak Unity dengan cara menconvert melalui *Target* engine yang disediakan oleh pihak Unity, setelah gambar di convert menghasilkan file dengan format .ass. File tersebut kemudian dijadikan masukan pada coding untuk mendeteksi gambar yang di jadikan *Target*.

Spesikasi pola *Target* :

1. Pola *Target* minimum harus memiliki lebar 550 pixels
2. Format gambar yang dikirimkan .jpg

2.2.3.5 Rendering Objek 3D

Transformasi matriks yang dikalkulasikan di step sebelumnya yang digunakan Unity dan menampilkan objek yang sesuai dengan sebuah library 3D, seperti yang ditunjukkan gambar 2.20. Unity menyertakan kelas pendukung yang mengkonversikan transformasi matriks Unity ke setiap kelas matriks internal library 3D tersebut. Sebagai contoh lainnya pada saat ini yang dikembangkan oleh perusahaan Augmented Reality terbesar di dunia Total Immersion, mereka telah membuat berbagai macam teknik *Markerless* Tracking sebagai teknologi andalan mereka, seperti Face Tracking, 3D Object Tracking, dan Motion Tracking.

1) *Face Tracking*

Dengan menggunakan alogaritma yang mereka kembangkan, 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.

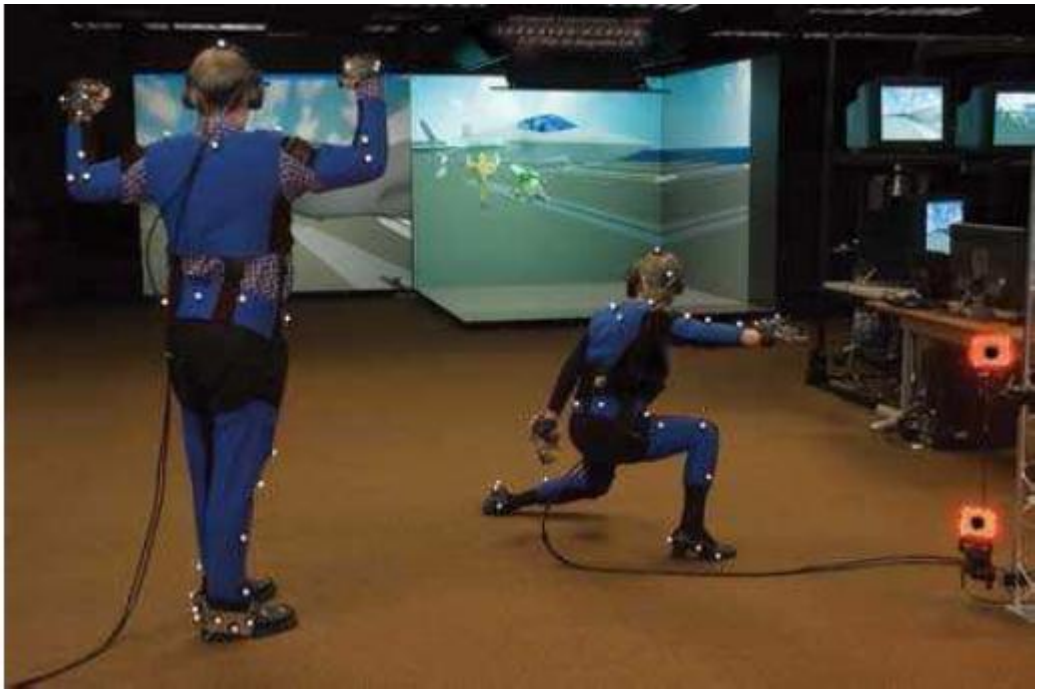
2) *3D Object Tracking*

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.

3) *Motion Tracking*

Pada teknik ini komputer dapat menangkap gerakan, Motion Tracking telah mulai digunakan secara ekstensif untuk memproduksi film - film yang mencoba mensimulasikan gerakan.

Contohnya pada film Avatar, di mana James Cameron menggunakan teknik ini untuk membuat film tersebut dan menggunakannya secara realtime.



Gambar 2.10 Motion Tracking

2.2.3.6 Model Tiga Dimensi (3D)

Pemodelan Tiga Dimensi (3D) (3D modeling atau dikenal juga dengan meshing) adalah proses pembuatan representasi matematis permukaan tiga dimensi dari suatu objek dengan software tertentu. Produk hasil pemodelan itu disebut model 3D. Model 3D tersebut dapat ditampilkan sebagai citra dua dimensi melalui sebuah proses yang disebut 3D rendering. Model 3D direpresentasikan dari kumpulan titik dalam 3D, terhubung oleh berbagai macam entitas geometri, seperti segitiga, garis, permukaan lengkung, dan lain sebagainya. Berdasarkan hal tersebut, model 3D bisa dibuat manual (seperti seni memahat), secara algoritma (pemodelan prosedural), atau scanning. Hasil akhir dari citra 3D adalah sekumpulan poligon. Model dengan jumlah poligon yang lebih banyak memerlukan waktu yang lebih lama untuk direnderer oleh komputer, karena setiap permukaan memiliki tekstur dan shading tersendiri. [11]

2.2.4 Bahasa Pemrograman 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 tellah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat

pada bahasa-bahasa pemrograman lainnya seperti *Java, Delphi, Visula 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.

Standar European Computer Manufacturer Association (ECMA) mendaftarkan beberapa tujuan desain dari bahasa pemrograman C#, sebagai berikut:

a) Bahasa pemrograman C# dibuat sebagai bahasa pemrograman yang bersifat *bahasa pemrograman general-purpose* (untuk tujuan jamak), berorientasi objek, modern, dan sederhana.
b) Bahasa pemrograman C# ditujukan untuk digunakan dalam mengembangkan komponen perangkat lunak yang mampu mengambil keuntungan dari lingkungan terdistribusi.
c) Portabilitas *programmer* sangatlah penting, khususnya bagi *programmer* yang telah lama menggunakan bahasa pemrograman C dan C++.
d) Dukungan untuk internasionalisasi (*multi-language*) juga sangat penting.
e) C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem klien-server (*hosted system*) maupun sistem *embedded* (*embedded system*), mulai dari perangkat lunak yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada perangkat lunak yang sangat kecil yang memiliki fungsi-fungsi terdedikasi.
f) Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan ukuran perangkat lunak yang dibuat dengan menggunakan bahasa pemrograman C dan bahasa rakitan.

g) Bahasa C# harus mencakup pengecekan jenis (*type checking*) yang kuat, pengecekan larik (*array*), pendeteksian terhadap percobaan terhadap penggunaan Variabel-variabel yang belum diinisialisasikan, portabilitas kode sumber, dan pengumpulan sampah (*garbage collection*) secara otomatis.

2.2.5 Bahasa Pemrograman *Java*

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program *Java* tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram *Java* banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas *Java*, yang disebut dengan *Java Application Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket *(package). Java* API telah menyediakan fungsionalitas yang memadai untuk menciptakan *applet* dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam *Java*, yaitu mempelajari bahasa *Java* dan bagaimana mempergunakan kelas pada *Java* API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada *Java* program *Javac* untuk mengkompilasi file kode sumber *Java* menjadi kelas-kelas *bytecode*. File kode sumber mempunyai ekstensi *.*Java*. Kompilator *Javac* menghasilkan file *bytecode* kelas dengan ekstensi *.*class*. *Interpreter* merupakan modul utama sistem *Java* yang digunakan aplikasi *Java* dan menjalankan program *bytecode Java*. [4]

Beberapa keunggulan *Java* yaitu *Java* merupakan bahasa yang sederhana. *Java* dirancang agar mudah dipelajari dan digunakan secara efektif. *Java* tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan *Java* secara otomatis seperti dealokasi memori. Bagi pemrogram yang sudah mengenal bahasa C++ akan cepat belajar susunan bahasa *Java* namun harus waspada karena mungkin *Java* mengambil arah (semantiks) yang berbeda dibanding C++.

Java merupakan bahasa berorientasi objek *(OOP)* yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi ringkas OOP adalah mengorganisasikan program sebagai kumpulan komponen, disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagai modul sekaligus tipe. Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi kelas lain. *Java* tidak mengijinkan pewarisan jamak namun menyelesaikan kebutuhan pewarisan jamak dengan fasilitas antarmuka yang lebih elegan.

Seluruh objek diprogram harus dideklarasikan lebih dulu sebelum digunakan. Ini merupakan keunggulan *Java* yaitu *Statically Typed*. Pemaksaan ini memungkinkan kompilator *Java* menentukan dan melaporkan terjadinya pertentangan (ketidakkompatibelan) tipe yang merupakan barikade awal untuk mencegah kesalahan yang tidak perlu (seperti mengurangkan variabel bertipe integer dengan variabel bertipe string). Pencegahan sedini mungkin diharapkan menghasilkan program yang bersih. Kebaikan lain fitur ini adalah kode program lebih dapat dioptimasi untuk menghasilkan program berkinerja tinggi.

Java menggunakan model pengamanan tiga lapis *(three-layer security model)* untuk melindungi sistem dari *untrusted Java code*. Pertama, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan-aturan dasar bahasa *Java*. Kedua, *class loader* menangani pemuatan kelas *Java* ke *runtime interpreter*. Ketiga, manajer keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem *file*, *port* jaringan, proses eksternal dan sistem *window*.

Platform independence adalah kemampuan program bekerja di sistem operasi yang berbeda. Bahasa *Java* merupakan bahasa yang secara sempurna tidak bergantung *platform*. Tipe variabel *Java* mempunyai ukuran sama di semua *platform* sehingga variabel bertipe integer berukuran sama tidak peduli dimana program *Java* dikompilasi. Begitu telah tercipta *file .class* dengan menggunakan

kompilator *Java* di *platform* manapun, maka *file .class* tersebut dapat dijalankan di *platform* manapun. Jadi "dimanapun dibuat, dimanapun dapat dijalankan". Slogan ini biasa diringkas sebagai *Write Once, Run Anywhere (WORA).*

Java termasuk bahasa *Multithreading*. *Thread* adalah untuk menyatakan program komputer melakukan lebih dari satu tugas di satu waktu yang sama. *Java* menyediakan kelas untuk menulis program *multithreaded,* program mempunyai lebih dari satu *thread* eksekusi pada saat yang sama sehingga memungkinkan program menangani beberapa tugas secara konkuren.

Program *Java* melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh pemrogram dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat di bahasa yang memungkinkan alokasi dinamis.

Java mempunyai mekanisme *exception-handling* yang ampuh. *Exception-handling* menyediakan cara untuk memisahkan antara bagian penanganan kesalahan dengan bagian kode normal sehingga menuntun ke struktur kode program yang lebih bersih dan menjadikan aplikasi lebih tegar. Ketika kesalahan yang serius ditemukan, program *Java* menciptakan *exception*. *Exception* dapat ditangkap dan dikelola program tanpa resiko membuat sistem menjadi turun.

Program *Java* mendukung *native method* yaitu fungsi ditulis di bahasa lain, biasanya C/C++. Dukungan *native method* memungkinkan pemrogram menulis fungsi yang dapat dieksekusi lebih cepat dibanding fungsi ekivalen di *Java*. *Native method* secara dinamis akan di-*link* ke program *Java,* yaitu diasosiasikan dengan program saat berjalan.

Selain itu keuntungan menggunakan bahasa pemrograman *Java* antara lain memori pada *Java* secara otomatis dilengkapi *garbage collector* yang berfungsi mendealokasi memori yang tidak diperlukan. Tidak ada lagi upaya yang dilakukan pemrogram untuk melakukan *dispose*(). Kita tidak lagi dibebani urusan korupsi memori. *Java* menerapkan array sebenarnya, menghilangkan keperluan aritmatika pointer yang berbahaya dan mudah menjadi salah. Menghilangkan pewarisan jamak *(multiple inheritance)* diganti fasilitas antarmuka. Dan mudah dijalankan diberbagai *platform*.

Grafical User Interface (GUI) adalah salah satu kemampuan *Java* dalam mendukung dan manajemen antarmuka berbasis grafis. Tampilan grafis yang akan ditampilkan terhubung dengan program serta tempat penyimpanan data. Elemen dasar di *Java* untuk penciptan tampilan berbasis grafis adalah dua paket yaitu AWT dan *Swing*. *Abstract Windowing Toolkit* (AWT), atau disebut juga *"Another Windowing Toolkit"*, adalah pustaka *windowing* bertujuan umum dan *multiplatform* serta menyediakan sejumlah kelas untuk membuat GUI di *Java*. Dengan AWT, dapat membuat window, menggambar, manipulasi gambar, dan komponen seperti *Button, Scrollbar, Checkbox, TextField,* dan menu *pull-down*.

Penggunaan komponen AWT ditandai dengan adanya instruksi : *import Java.awt.*; Swing* merupakan perbaikan kelemahan di AWT. Banyak kelas *swing* menyediakan komponen alternatif terhadap AWT. Contohnya kelas *JButton swing* menyediakan fungsionalitas lebih banyak dibanding kelas *Button*. Selain itu komponen *swing* umumnya diawali dengan huruf "J", misalnya *JButton, JTextField, JFrame, JLabel, JTextArea, JPanel,* dan sebagainya. Teknologi *swing* menggunakan dan memperluas gagasan-gagasan AWT. Sementara, penggunaan komponen *Swing* ditandai dengan adanya instruksi : *import Javax.swing.*;*

Beberapa perbedaan AWT dan *Swing*, AWT merupakan komponen *heavyweight* (kelas berat) sedangkan *Swing lightweight* (kelas ringan). *Swing* memiliki lebih banyak komponen. Fasilitas *Swing Look and Feel* : *Metal, Windows,* Motif. Komponen *Swing* berdasar *model-view*, yaitu suatu cara pengembangan komponen dengan pemisahan penyimpanan dan penanganan data dari representasi visual data.

Bahasa pemrograman *Java* merupakan salah satu bahasa pemrograman yang umum digunakan untuk mengembangkan aplikasi basis data yang dibuat menggunakan MySQL.[4]

2.2.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung

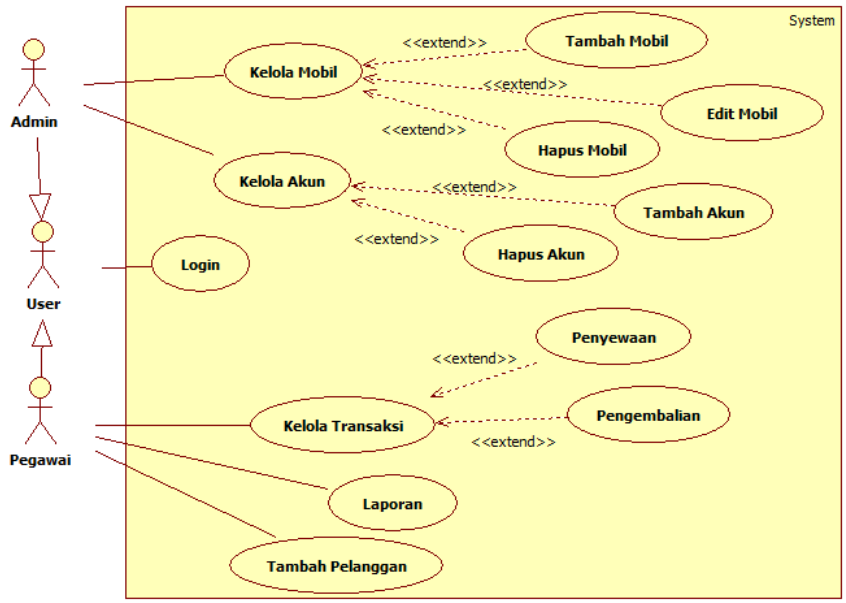
pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group,* sebuah organisasi yang telah mengembangkan model, teknologi dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.[14]

2.2.6.1 Diagram Unified Modeling Language

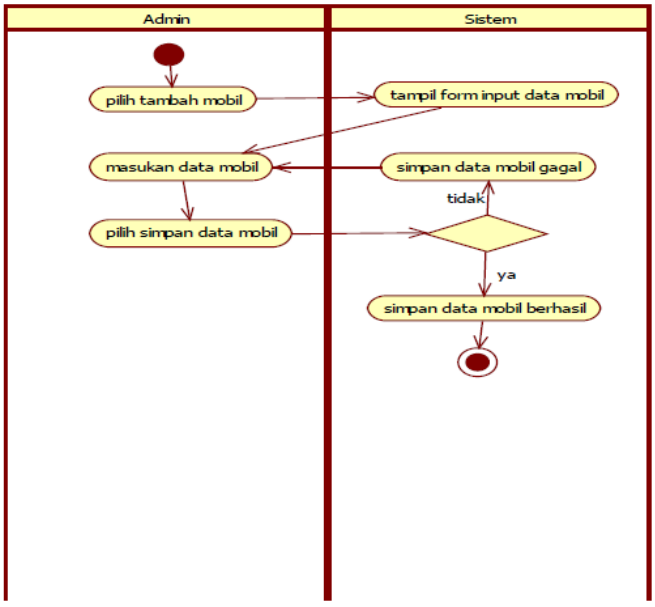
UML terdiri atas pengelompokan diagram-diagram sistem menurut aspek dan sudut pandang tertentu. Diagram merupakan gambaran permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu :

a. ***Use case*** **diagram** digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan *actor*. *Actor* merepresentasikan 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 di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.



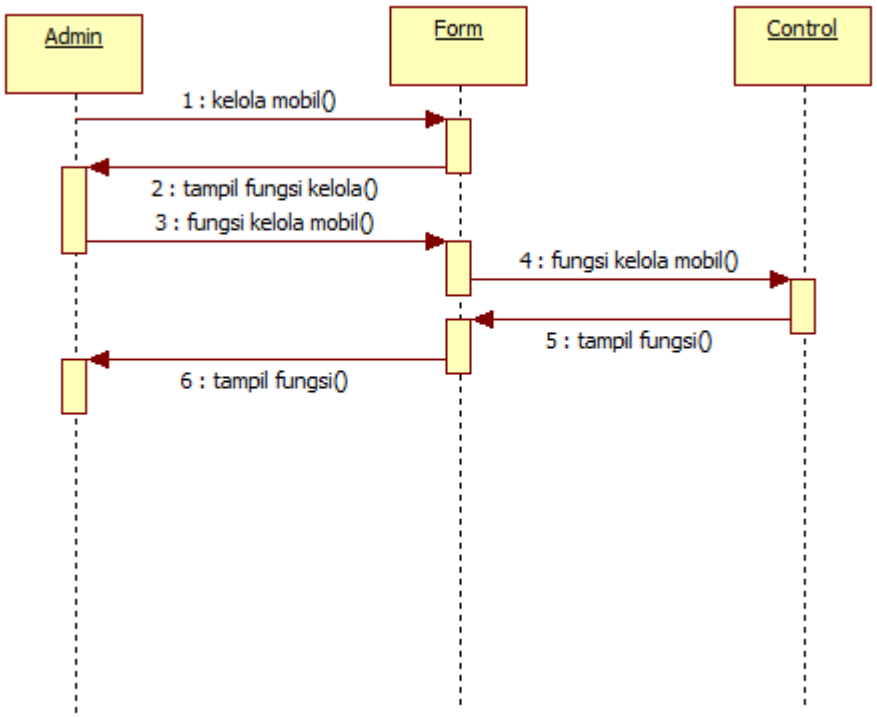
Gambar 2.11 Contoh Use Case Diagram

b. ***Activity* diagram** menggambarkan aliran fungsionalitas sistem. Pada tahap emodelan bisnis, diagrama aktivitas dapat digunakan untuk menujukkan aliran kerja bisnis *business work flow).* Dapat juga digunakan untuk menggambarkan aliran kejadian *(flow of event)* dalam *use case*.



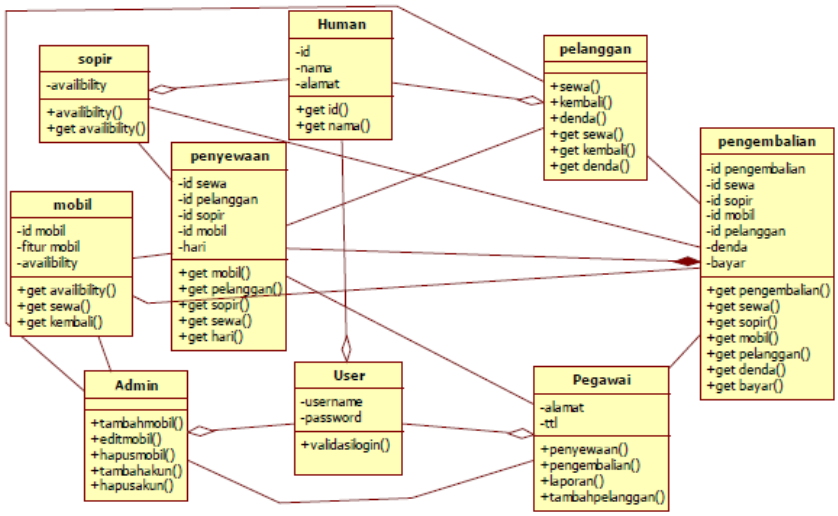
Gambar 2.12 Contoh Activity Diagram

c. ***Sequence* diagram** digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah diagram yang disusun berdasarkan urutan waktu. Kita membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan suatu aliran dari beberapa aliran di dalam *use case*. Kita dapat membaca diagram ini dengan memperhatikan obyek-obyek dan pesan-pesan yang ada di diagram. Obyek yang terlibat dalam aliran ditunjukkan dengan bujur sangkar yang ada di atas diagram.



Gambar 2.13 Contoh Sequence Diagram

d. ***Class* diagram** menggambarkan struktur statis *class* di dalam sistem. *class* merepresentasikan sesuatu yang ditangani oleh sistem. *class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialed* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (*group* bersama sebagai satu unit). sebuah sistem biasanya mempunyai beberapa *class* diagram.



Gambar 2.14 Contoh Class Diagram

e. ***Collaboration* diagram** dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence* diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration* diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

f. ***State* Diagram**, yang memperliatkan semua keadaan (*state*) yang dapat dimiliki oleh kelas dan *event* yang dapat merubah keadaan tersebut.

g. ***Component* Diagram**, yang memperlihatkan struktur fisik dari *source code* dalam *terminology code components*. Komponen berisi informasi tentang logical *class* dapat berupa komponen *source code*, komponen biner atau komponen yang dapat dieksekusi.

h. ***Deployment* Diagram**, yang memperlihatkan arsitektur fisik dari *hardware* dan *software* pada sistem.

i. ***Object* Diagram**, yang merupakan varian dari kelas diagram yang memperlihatkan lebih detail banyaknya obyek yang mengintantiasi (*instances*) kelas.

2.2.7 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika

pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Berikut konsep dasar Pemrograman Berorientasi Objek, yaitu :

a. Kelas adalah kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *'class of dog'* adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen.
b. Objek adalah membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
c. Abstraksi adalah kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
d. Enkapsulasi memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan

bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

e. Polimorfisme tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. [5]

2.2.8 Pengolahan Citra Digital

Data atau informasi tidak hanya disajikan dalam bentuk teks, tapi juga dapat berupa gambar, audio (bunyi, suara, musik) dan video. Keempat macam data atau informasi ini sering disebut multimedia. Citra (image) istilah lain untuk gambar sebagai satu komponen multimedia memegang peranan penting sehingga bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ada sebuah peribahasa yang berbunyi “sebuah gambar akan lebih bermakna dari seribu kata” (a picture is more than a thousand words). Maksudnya tentu sebuah gambar dapat memberikan informasi yang lebih banyak dari pada informasi tersebut disajikan dalam bentuk kata-kata (tekstual).

Istilah citra atau image yang pada umumnya digunakan dalam bidang pengolahan citra diartikan sebagai suatu fungsi kontinu dari intensitas cahaya f(x,y) dalam bidang dua dimensi dengan (x,y) menyatukan suatu koordinat dangan nilai f pada setiap titik menyatukan intensitas atau tingkatan kecerahan atau derajat keabuan (brightness/gray level). Suatu citra digital adalah suatu citra kontinyu yang diubah kedalam bentuk diskrit, baik koordinat maupun intensitas

cahayanya. Kita dapat menganggap suatu citra digital sebagai suatu matriks, dimana indeks baris dan kolomnya menyatakan koordinat sebuah titik pada citra tersebut dan nilai masing-masing elemennya menyatakan intensitas cahaya pada titik tersebut.

Suatu titik pada sebuah citra digital sering disebut sebagai elemen citra (image-elemen), elemen gambar (picture-elemen), piksel (pixel / pel). Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer menjadi citra yang kualitasnya lebih baik.[6]

2.2.8.1 Sejarah Pengolahan Citra

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:

Dalam bidang kedokteran

Sistem untuk mendeteksi diagnosa suatu kelainan dalam tubuh manusia melalui gambar yang dihasilkan oleh suatu gambar scanner.

1. Dalam bidang industri

   Sistem untuk memeriksa kualitas suatu produk melalui kamera video.

2. Dalam bidang perdagangan

   Sistem untuk mengenal huruf / angka pada suatu formulir secara automatis oleh suatu mesin pembaca.

3. Dalam bidang militer

   Sistem pengenalan target peluru kendali melalui sensor visual.

4. Dalam bidang biologi

   Sistem pengenalan jenis kromosom melalui gambar mikroskop.

Keikut sertaan berbagai disiplin ilmu dalam kegiatan pengolahan citra dimulai dari pembentukan model matematik suatu objek sampai dengan teknik analisis dan teknik klasifikasi berbagai jenis objek.[6]

2.2.8.2 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 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 memperlihatkan kombinasi 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.[7]

2.2.9 Data dan Informasi

Hubungan antara data dan informasi sangatlah erat sebagaimana hubungan antara sebab dan akibat. Bahwa data merupakan bentuk dasar dari sebuah informasi, sedangkan informasi merupakan elemen yang dihasilkan dari suatu bentuk pengolahan data.

2.2.9.1 Data

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Data sering kali disebut sebagai bahan mentah informasi.

Berikut adalah kutipan pengertian data dari sudut pandang yang berbeda :

1. Menurut kamus bahasa Inggris-Indonesia, data diterjemahkan sebagai istilah yang berasal dari kata “datum” yang berarti fakta atau bahan-bahan keterangan.
2. Dari sudut pandang bisnis, terdapat pengertian data bisnis sebagai berikut: *“Bussines data is an organization’s description of things* (*resources*) *and events (transactions) that it faces”*. Jadi data, dalam hal ini disebut sebagai data bisnis, merupakan deskripsi organisasi tentang sesuatu (*resource*) dan kejadian (*transaction)* yang terjadi.
3. Menurut (Jogiyanto, HM, 1989 : 8) data merupakan bentuk jamak dari datum (kenyataan) yang berupa fakta-fakta, angka-angka, gambar-gambar yang dapat ditarik kesimpulannya.
4. Gordon B. Davis dalam bukunya *Management Informations System : Conceptual Foundations, Structure, and Development* menyebut data sebagai bahan mentah dari informasi, yang dirumuskan sebagai sekelompok lambang-lambang tidak acak yang menunjukkan jumlah atau tindakan atau hal-hal lain.

Dari keempat pengertian diatas, dapat diambil kesimpulan bahwa data adalah bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda,dan sebagainya. Data terbentuk dari karakter, dapat berupa alphabet, angka, maupun simbol khusus seperti *, $, dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data.[8]

2.2.9.2 Informasi

Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi ada suatu proses transformasi data menjadi suatu informasi (input – proses – output).



Gambar 2.15 Pemrosesan Data Menjadi Informasi

Definisi umum untuk informasi dalam sistem informasi menurut Jogiyanto H.M (1990; 11) : "Informasi adalah bentuk data yang dapat diolah yang lebih berguna dan berarti bagi yang menerimanya". Menurut Robert G. Munik (1973 ; 12) : "Informasi adalah data yang telah diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. Berikut pengertian informasi dari berbagai sumber :

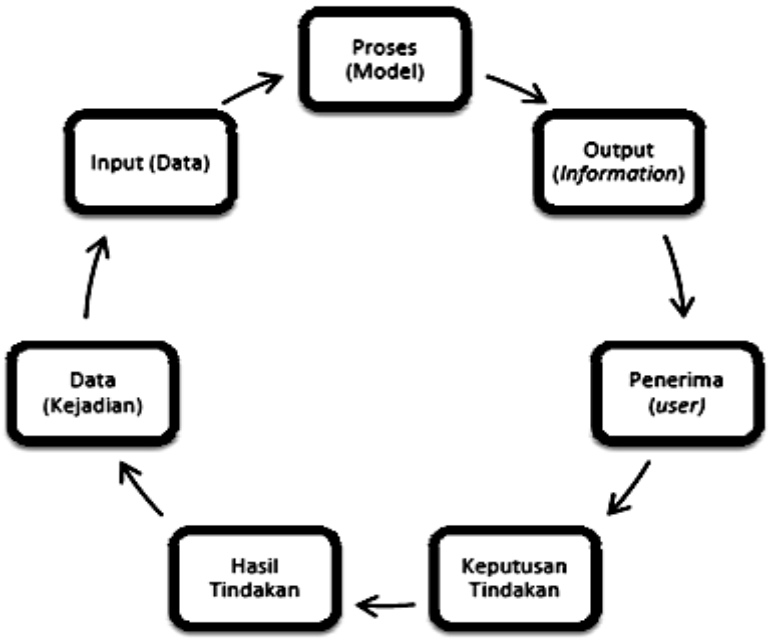
1. Menurut Gordon B. Davis dalam bukunya *"Management Informations System: Conceptual Foundations, Structures, and Development* menyebut informasi sebagai data yang telah diolah menjadi bentuk yang berguna bagi penerimanya dan nyata, berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan.
2. Menurut Berry E. Cushing dalam bukunya*"Accounting Information System and Business Organization"* dikatakan bahwa informasi merupakan sesuatu yang menunjukkan hasil pengolahan data yang diorganisasi dan berguna kepada orang yang menerimanya.
3. Menurut Robert N. Anthony dan John Dearden dalam bukunya *"Managemet Control Systems"*, menyebut informasi sebagai suatu kenyataan, data, item, yang menambah pengetahuan bagi penggunanya.

4. Menurut Stephan A. Moscove dan Mark G. Simkin dalam bukunya *"Accounting Information Systems: Concepts and Practise"* mengatakan informasi sebagai kenyataan atau bentuk-bentuk yang berguna yang dapat digunakan untuk pengambilan keputusan bisnis.

Dari keempat pengertian seperti tersebut di atas dapat disimpulkan bahwa informasi hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan.[8]

2.2.9.3 Sirkulasi Informasi

Untuk memperoleh informasi yang bermanfaat bagi penerimanya, perlu untuk dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi. Pertama-tama data dimasukkan ke dalam model yang umumnya memiliki urutan proses tertentu dan pasti, setelah diproses akan dihasilkan informasi tertentu yang membuat suatu keputusan atau melakukakn tindakan tertentu. Dari keputusan atau tindakan tersebut akan menghasilkan atau diperoleh kejadian-kejadian tertentu yang akan digunakan kembali sebagai data yang nantinya akan dimasukkan ke dalam model (proses), begitu seterusnya. Dengan demikian akan membentuk suatu siklus informasi (*information cycle*) atau siklus pengolahan data (*data processing cycles*), seperti gambar berikut :



Gambar 2.16 Siklus Informasi.

2.2.10 Konsep Basis Data

Basis data dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Data menunjukan sekumpulan data yang dipakai dalam suatu lingkungan perusahaan atau instansi – instansi. Penerapan basis data dalam sistem informasi disebut sistem basis data.[8]

2.2.10.1 Pengertian Basis data

Basis data terdiri dari kata *basis* dan *data*. Basis data dapat diartikan gudang atau tempat bersarang, sedangkan data berarti representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasi.

Dapat disimpulkan bahwa basis data merupakan kumpulan data (arsip) yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Basis data dapat diartikan sebagai kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.[10]

2.2.10.2 Model Basis Data

Model basis data adalah sekumpulan konsep terintegrasi yang dipakai untuk menjabarkan data, hubungan antar data, dan kekangan terhadap data yang digunakan untuk menjaga konsistensi. Kadang model data disebut dengan stuktur data logis.Model data yang umum pada saat ini ada empat macam,diantaranya sebagai berikut :

1. Model data hierarki.
2. Model data jaringan.
3. Model data relasional.
4. Model data berbasis objek.

Tiga model di atas yang selain model data berbasis objek disebut model data yang berbasis rekaman *(record-based data model)*

2.2.10.3 Media Penyimpanan (Storage)

Media penyimpan data *(data storage)* adalah sebuah kemampuan dari sebuah benda yang memungkinkan data tersimpan dalam jumlah besar. Seiring dengan perkembangan dan penelitian para ilmuan-ilmuan dunia mencoba untuk membentuk sebuah media penyimpanan data yang efisien. Dalam artian memiliki ruang penyimpanan yang besar tapi ukuran dengan ukuran yang kecil.

2.2.10.4 XML (Extensible Markup Language)

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet. XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi.

XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang *view database*, tetapi dengan suatu cara yang standar. XML memiliki tiga tipe file :

1. XML, merupakan standar format dari struktur berkas (*file*).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD, merupakan standar yang mendefinisikan struktur database dalam XML.

Keunggulan XML bisa diringkas sebagai berikut :

1. Pintar (*Intelligence*). XML dapat menangani berbagai tingkat (*level*) kompleksitas.
2. Dapat beradaptasi dan mengadaptasi untuk membuat bahasa sendiri. Seperti Microsoft membuat bahasa MSXML atau Macromedia mengembangkan MXML
3. Mudah pemeliharaannya.

4. Sederhana. XML lebih sederhana.
5. Mudah dipindah-pindahkan (*Portability*). XML mempunyai kemudahan perpindahan (portabilitas) yang lebih bagus.

2.2.10.5 DAT (Digital Audio Tape)

Digital Audio Tape (DAT atau R-DAT) adalah sebuah sinyal perekaman dan pemutaran menengah yang dikembangkan oleh Sony dan diperkenalkan pada tahun 1987. Dalam penampilan itu mirip dengan kaset audio kompak, menggunakan 4 mm pita magnetik pelindung tertutup di shell, tetapi kira-kira setengah dari ukuran di 73 mm × 54 mm × 10.5 mm. Seperti namanya, adalah rekaman digital daripada analog.

DAT memiliki kemampuan untuk merekam lebih tinggi, sama atau lebih rendah dari tingkat sampling CD (48, 44,1 atau 32 kHz sampling rate masing-masing) pada 16 bit kuantisasi. Jika sumber digital disalin maka DAT akan menghasilkan klon yang tepat, tidak seperti media digital lain seperti *Digital Compact Cassette* atau non-Hi-MD MiniDisc, baik yang menggunakan kompresi data *lossy*.

DAT adalah ekstensi file VCD yang digunakan dalam Sistem Operasi Microsoft Windows berarti "data video" normal. Hal ini dapat dikodekan dengan MPEG 1.0, 1.1, 2.0 standar.

2.3 Tool Pendukung Pembuatan Aplikasi

Adapun tool pendukung dalam pembangunan aplikasi augmented reality Pengenalan Kupu-kupu yaitu :

2.3.1 Java Development Kit (JDK)

JDK adalah software development kit merupakan seperangkat kode, library, form dan paket yang berguna untuk membuat aplikasi atau applet java. Di dalam JDK terdapat Java Runtime Environment (JRE) yang akan memproses byte code yang dihasilkan dari interpenter java. Tugas dari interpenter java ini adalah untuk mengcompile source.java menjadi source.class yang selanjutnya source.class inilah yang akan dijalankan oleh JRE.

2.3.2 Software Development Kit (SDK)

SDK adalah suatu kumpulan dari library dan toolkit seperti emulator untuk mengembangkan atau menciptakan aplikasi untuk suatu perangkat lunak, perangkat computer, system operasi atau platform. Di dalam SDK terdapat tools yang dibutuhkan dalam pengembangan android, diantaranya yaitu adb shell yang merupakan singkatan dari android development bridge yang dapat menjalankan terminal android seperti terminal pada system operasi linux, dan command yang terdapat dalam adb shell sendiri seperti command linux pada umumnya.

2.3.3 Unity

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

Update terbaru, *Unity 4.2.1*, dirilis September 2013. Saat ini mendukung pengembangan untuk *iOS, Android, Windows, Blackberry 10, OS X, Linux, web browser, Flash, PlayStation 3, Xbox 360, Windows Phone 8,* dan *Wii U*. Dua versi dari *game developing software* tersedia untuk di-download, *Unity* dan *Unity Pro*.

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

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

Bahasa *ShaderLab* digunakan untuk *shader* , mendukung kedua deklaratif " pemrograman " dari program tetap fungsi pipa dan shader ditulis dalam *GLSL* atau *Cg* . *Shader A* dapat mencakup beberapa varian dan spesifikasi fallback deklaratif , memungkinkan *Unity* untuk mendeteksi varian yang terbaik untuk kartu video saat ini , dan jika tidak ada yang kompatibel , jatuh kembali ke shader alternatif yang mungkin mengorbankan fitur untuk kinerja.

Pada 3 Agustus 2013 , dengan rilis 4.2 , *Unity* memungkinkan pengembang untuk menggunakan bayangan *Indie Realtime* hanya untuk lampu *Directional* , juga dukungan dari *DirectX11* ditambahkan , yang memberikan resolusi pixel yang lebih sempurna bayangan , tekstur untuk membuat objek 3d dari *grayscale* , grafis yang lebih wajah , animasi halus dan dorongan untuk FPS .

Scripting permainan mesin ini dibangun di atas *Mono 2.6* , implementasi *open-source* dari *NET* . Kerangka . Pemrogram dapat menggunakan *UnityScript* (bahasa kustom dengan sintaks *ECMAScript*, disebut sebagai *JavaScript* oleh perangkat lunak) , *C #* , atau *Boo* (yang memiliki sintaks *Python*).

Unity juga mencakup *Unity Aset Server* - sebuah solusi kontrol versi untuk aset permainan pengembang dan skrip . Menggunakan *PostgreSQL* sebagai *backend* , sistem audio dibangun di perpustakaan *FMOD* (dengan kemampuan untuk pemutaran *Ogg Vorbis* terkompresi audio) , pemutaran video menggunakan codec *Theora* , medan dan mesin vegetasi (yang mendukung pohon *billboarding* , *Occlusion* Pemusnahan dengan *Umbra*) , *built-in* iluminasi *lightmapping* dan global dengan *Beast* , jaringan *multiplayer* menggunakan *RakNet* , dan *built-in pathfinding* jerat navigasi.

Unity mendukung penyebaran ke berbagai *platform* . Dalam sebuah proyek , pengembang memiliki kontrol atas pengiriman ke perangkat mobile , web browser , desktop , dan konsol. *Unity* juga memungkinkan . Spesifikasi kompresi tekstur dan pengaturan resolusi untuk setiap platform game mendukung.

Platform yang saat ini didukung termasuk *BlackBerry 10* , *Windows 8* , *Windows Phone 8* , *Windows* , *Mac* , *Linux* , *Android* , *iOS* , *Unity Web Player* , *Adobe Flash* , *PlayStation 3* , *Xbox 360* , *Wii U* dan *Wii* .

Diluncurkan pada bulan November 2010, Aset toko Unity adalah sumber daya yang tersedia dalam editor Unity . Toko terdiri dari koleksi lebih dari 4.400 paket aset, termasuk model 3D , tekstur dan bahan , sistem partikel , musik dan efek suara , tutorial dan proyek , paket scripting , ekstensi Editor dan layanan online. Unity memiliki class yang bernama Monobehaviour,class tersebut merupakan dasar dari semua script yang terdapat dalam device,di bawah merupaka penjelasan function yang terdapat di class monobehaviour.[11]

No	Functions	Deskripsi
1	CancelInvoke()	Membatalkan semua perintah Invoke pada Monobehaviour
2	Invoke()	Memanggil method yang bernama methodName
3	InvokeRepeating()	Memanggil method yang bernama methodName,kemudian berulang kali
4	IsInvoking()	Setiap memanggil methodName tertunda
5	StartCoroutine()	Memulai Coroutine
6	StopAllCoroutines()	Memberhentikan semua Coroutine
7	StopCoroutine()	Memberhentikan semua Coroutine bernama methodname yang berjalan.
8	print()	Pesan log ke unity console
9	Awake()	Awake di panggil ketika script sedang di muat
10	FixedUpdate()	Fungsi ini dipanggil setiap frame framerate tetap, jika MonoBehaviour diaktifkan.
11	LateUpdate()	LateUpdate disebut setiap frame, jika Behaviour diaktifkan.
12	Start()	Start disebut pada frame bila sebuah script diaktifkan sebelum salah satu metode pembaruan disebut pertama kalinya.

Tabel 2.1 Tabel Deskripsi function MonoBehaviour

2.3.4 Vuforia Augmented Reality SDK

Vuforia adalah *Augmented Reality Software Development Kit (SDK)* untuk perangkat mobile yang memungkinkan pembuatan aplikasi *Augmented Reality*. Ini menggunakan teknologi *Computer Vision* untuk mengenali dan melacak planar gambar *(Image Target)* dan objek 3D sederhana , seperti kotak , secara *real-time*.

Kemampuan registrasi citra memungkinkan pengembang untuk posisi dan benda-benda *virtual orient* , seperti model 3D dan media lainnya , dalam kaitannya dengan gambar dunia nyata saat ini dilihat melalui kamera perangkat mobile . Virtual obyek kemudian melacak posisi dan orientasi gambar secara *real-time* sehingga perspektif pemirsa pada objek sesuai dengan perspektif mereka pada Sasaran Gambar, sehingga muncul bahwa obyek virtual adalah bagian dari adegan dunia nyata.

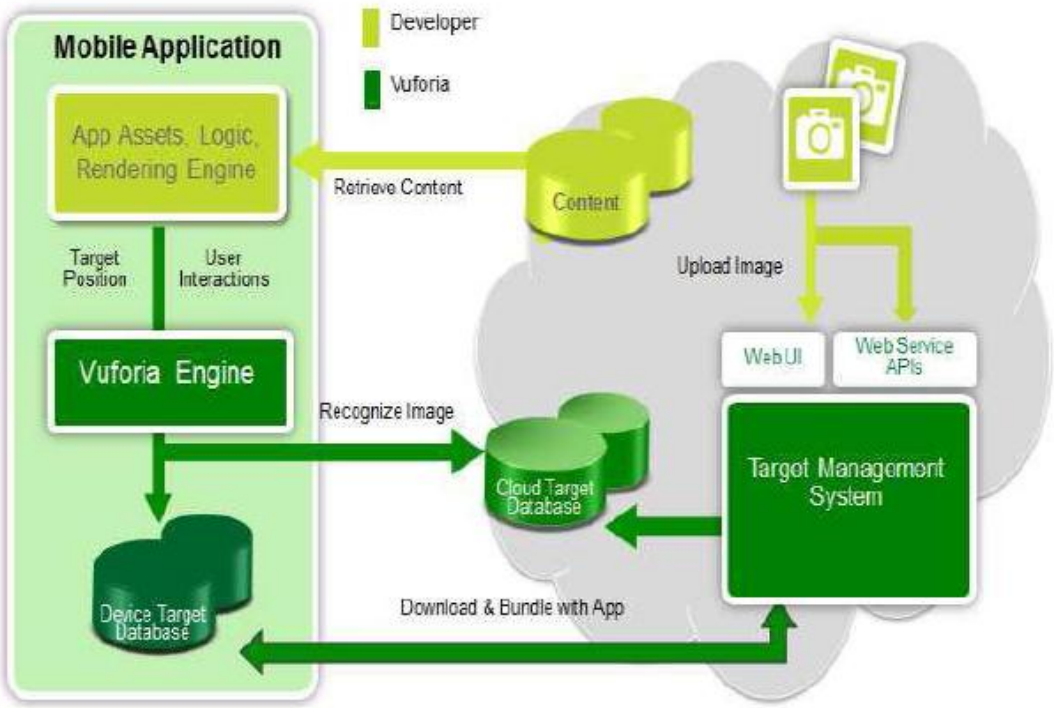
Vuforia SDK mendukung berbagai jenis sasaran 2D dan 3D termasuk '*Markerless* ' Citra Target , 3D konfigurasi *multi - target* , dan bentuk *Target Fidusia* dialamatkan dikenal sebagai *Target Frame*. Fitur tambahan dari *SDK* termasuk Deteksi lokal *Occlusion* menggunakan ' *Buttons Virtual* ' , runtime gambar pemilihan target , dan kemampuan untuk membuat dan mengkonfigurasi ulang sasaran set pemrograman saat *runtime*.

Vuforia menyediakan *Application Programming Interfaces (API)* di *C + +* , *Java* , *Objective- C* , dan bahasa *Net* . Melalui perluasan ke mesin permainan *Unity*. Dengan cara ini , SDK mendukung pengembangan asli untuk IOS dan Android sementara juga memungkinkan pengembangan aplikasi AR dalam Unity yang mudah portabel untuk kedua platform . Aplikasi AR dikembangkan menggunakan Vuforia karena itu kompatibel dengan berbagai perangkat mobile termasuk iPhone (4/4S) , iPad , dan ponsel Android dan tablet yang menjalankan OS Android versi 2.2 atau yang lebih besar dan prosesor ARMv6 dengan FPU atau 7 (*Floating Point Unit*) kemampuan pemrosesan.

Vuforia merupakan software untuk augmented reality, yang menggunakan sumber yang konsisten mengenai computer vision yang fokus pada image

recognition.Vuforia mempunyai banyak fitur-fitur dan kemampuan, yang dapat membantu pengembang untuk mewujudkan pemikiran mereka tanpa adanya batas secara teknikal.

Dengan support untuk Windows, iOS, Android, dan Unity3D, platform Vuforia mendukung para pengembang untuk membuat aplikasi yang dapat digunakan di hampir seluruh jenis Komputer, *smartphone* dan, tablet. Pengembang juga diberikan kebebasan untuk mendesain dan membuat aplikasi yang mempunyai kemampuan antara lain :



Gambar 2.17 Vuforia Structure

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

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

2. *Frame markers*, tipe frame gambar 2D dengan pattern khusus yang dapat digunakan sebagai potongan permainan di permainan pada papan.

3. *Multi-target*, contohnya kemasan produk atau produk yang berbentuk kotak ataupun persegi. Jenis ini dapat menampilkan gambar sederhana Augmented 3D.

4. *Virtual buttons*, yang dapat membuat tombol sebagai daerah kotak sebagai sasaran gambar.

Pada Vuforia, ada 2 jenis workflow dengan dasar database yang dapat dipilih oleh developer, yaitu Cloud Database dan Device Database. Setelah memilih metode yang dipilih, developer dapat mengikuti panduan di bawah ini untuk membuat aplikasi pada Vuforia.[12]

2.3.5 Sketchup

Sketchup merupakan salah satu dari sekian banyak software yang berfungsi untuk membuat gambar 3D (3 dimensi). SketchUp dapat digunakan oleh praktisi-praktisi dibidang arsitektur, sipil, pembuat film, pengembang game, desainer grafis, bahkan ilustrator untuk menciptakan gambar 3d sesuai kebutuhan. Perbedaan SketchUp dibanding *software-software* 3D lain adalah user friendly, artinya SketchUp didesain familiar dan mudah digunakan oleh siapapun tanpa harus menguasai teknik-teknik yang rumit dan penuh perhitungan. Selain itu *software* ini tersedia dalam versi gratis yang dapat didownload. Versi gratis ini pun tidaklah mengecewakan karena memiliki banyak fungsi yang memungkinkan kita untuk membuat gambar dengan kualitas baik dan kerumitan cukup tinggi. Keunggulan lain adalah output yang dapat dihasilkan SketchUp dapat kita ubah menjadi beraneka ragam karakter gambar. Mulai dari gambar 2D vector, 3D realistis, ataupun gambar yang sangat menyerupai sketsa tangan.

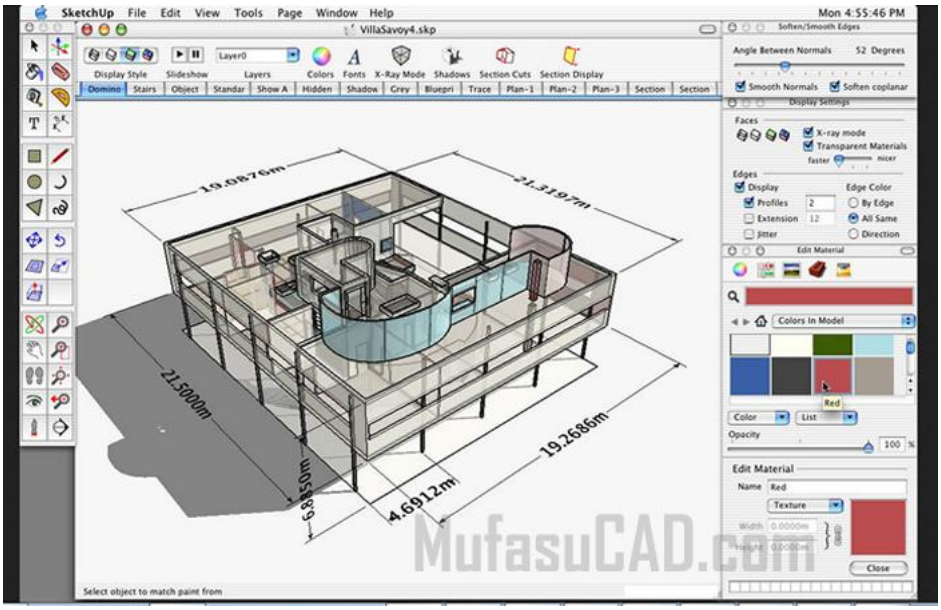
2.3.5.1 Sejarah Sketchup

Awalnya SketchUp dikembangkan oleh Last Software yang berlokasi di Boulder, Colorado yang didirikan pada tahun 1999 oleh founder Brad Schell dan Joe Esch. Rilis SketchUp pertama kali adalah pada Agustus 2000 dengan mengedepankan slogan "*3D for Everyone*" dan mengupayakan agar para desainer profesional dapat menuangkan karyanya semudah menggunakan pensil dan kertas, dan memiliki kebebasan berekspresi melalui gambar dengan interface yang mudah.

Pada *tradeshow* pertama tahun 2000 SketchUp memenangkan penghargaan Community Choice Award yang merupakan awal kesuksesan yang diraih SketchUp dalam waktu cukup singkat dibanding software 3D lainnya.Pada 14 Maret 2006 Google mengakuisisi ketertarikan Google setelah Last Software mengembangkan plugin untuk mengeksport hasil 3D ke Google Earth.

Pada 9 Januari 2007, Google merilis SketchUp 6, dengan berbagai fitur tambahan seperti "follow me" tool, dan SketchUp LayOut. Pada rilis ini terdapat tool yang memungkinkan kita untuk mengupload gambar ke 3D Warehouse, sebuah website yang memberikan sarana bagi pengguna untuk saling berbagi objek 3D. Pada rilis ini juga ditambahkan fitur navigasi yang memungkinkan pengguna untuk mensimulasi tampilan objek seolah-olah kita berjalan di area gambar untuk kemudian melihat sekeliling sesuai sudut pandang manusia.

Pada 9 February 2007, Google merilis update untuk maintenance dan memperbaiki beberapa bug. Kali ini Google tidak menambah satupun fitur baru. Kemudian pada 17 Nopember 2008, SketcUp 7 dirilis, memperkenalkan pengembangan yang berslogan ***"ease-of-use"*** serta Integrasi dengan browser komponen dan 3D Warehouse.



Gambar 2.18 Contoh Gambar Aplikasi Sketchup

2.3.6 Blender

Blender adalah perangkat lunak sumber terbuka grafika komputer 3D. Perangkat lunak ini digunakan untuk membuat film animasi, efek visual, model cetak 3D, aplikasi 3D interaktif dan permainan video. Blender memiliki beberapa fitur termasuk pemodelan 3D, penteksturan, penyunting gambar bitmap, penulangan, simulasi cairan dan asap, simulasi partikel, animasi, penyunting video, pemahat digital, dan rendering.

2.3.6.1 Sejarah Blender

Bermula Pada tahun 1988-an Ton Roosendaal mendanai sebuah perusahaan yang bergerak dibidang animasi yang dinamakan NeoGeo. NeoGeo yang sangat berkembang pesat sehingga menjadi perusahaan animasi terbesar di Belanda dan salah satu perusahaan animasi terdepan di Eropa. Ton Roosendaal selain bertanggung jawab sebagai art director juga bertanggung jawab atas pengembangan perangkat lunak internal.

Pada tahun 1995 muncullah sebuah perangkat lunak yang pada akhirnya dinamakan Blender. Setelah diamati lebih dalam ternyata Blender ini memiliki potensi untuk digunakan oleh artis –artis di luar NeoGeo. Lalu pada tahun 1998 Ton mendirikan perusahaan yang bernama Not a Number (NaN) Untuk mengembangkan dan memasarkan Blender lebih jauh. Cita – cita NaN adalah untuk menciptakan sebuah perangkat lunak animasi 3D yang padat, lintas platform yang gratis dan dapat digunakan oleh masyarakat pengguna komputer yang umum.

Sayangnya cita-cita nya NaN tidak sesuai dengan kenyataan pasar saat itu. pada tahun 2001 NaN dibentuk ulang menjadi perusahaan yang lebih kecil NaN lalu meluncurkan perangkat lunak komersial pertamanya, Blender Publisher. Sasaran pasar perangkat lunak ini adalah untuk web 3D interaktif. Angka penjualan yang rendah dan iklim ekonomi yang tidak menguntungkan saat itu mengakibatkan NaN ditutup. Penutupan ini termasuk penghentian terhadap pengembangan Blender.

Karena tidak ingin Blender hilang ditelan waktu dan zaman begitu saja, Ton Roosendaal mendirikan organisasi non profit yang bernama Blender Foundation. Tujuan utama Blender Foundation adalah terus mempromosikan dan mengembangkan Blender sebagai proyek sumber terbuka. Pada tahun 2002 Blender dirilis ulang di bawah syarat–syarat GNU General Public License. Pengembangan Blender terus berlanjut hingga saat ini.