BAB 2
LANDASAN TEORI

Dalam implementasi teknologi *Augmented reality* untuk Informasi 3D Peralatan Multimedia dibutuhkan landasan teori yang dapat membantu dalam proses pembuatan aplikasi serta objek penelitian . Berikut beberapa teori yang digunakan dalam perancangan aplikasi ini :

2.1 Profil SMK Mandalasari Cikancung

Objek penelitian dalam penelitian ini yaitu dilakukan di SMK Mandalasari Cikancung Kecamatan Mandalasari Kabupaten Bandung, Berikut Profil SMK Mandalasari Cikancung;

Nama Instansi	: SMK Mandalasari Cikancung
Alamat Intansi	: Jalan Raya Majalaya-Cicalengka Kp. Wirama Rt. 03 Rw. 10 Mandalasari Cikancung Bandung 40396.
Kepala Sekolah	: Usman Fauzi, S.Pd.I
Jurusan	: Multimedia
Akreditas	: C
Kelas	: 3 Rombel
Jumlah Siswa/i	: 135 orang

2.1.1 Logo SMK Mandalasari Cikancung

Gambar 2.1 Logo SMK Mandalasari Cikancung

2.1.2 Visi dan Misi SMK Mandalasari Cikancung

Visi adalah suatu pandangan jauh tentang program instansi di masa depan, tujuan –tujuan instansi dan apa yang harus dilakukan untuk mencapai tujuan tersebut pada masa yang akan datang, sedangkan Misi adalah pernyataan tentang apa yang harus dikerjakan oleh instansi dalam usahanya mewujudkan Visi.

2.1.2.1 Visi

Mewujudkan lingkungan sekolah yang bersih, warga sekolah yang santun, kehidupan yang religius serta berprestasi dalam bidang imtaq dan iptek.

2.1.2.2 Misi

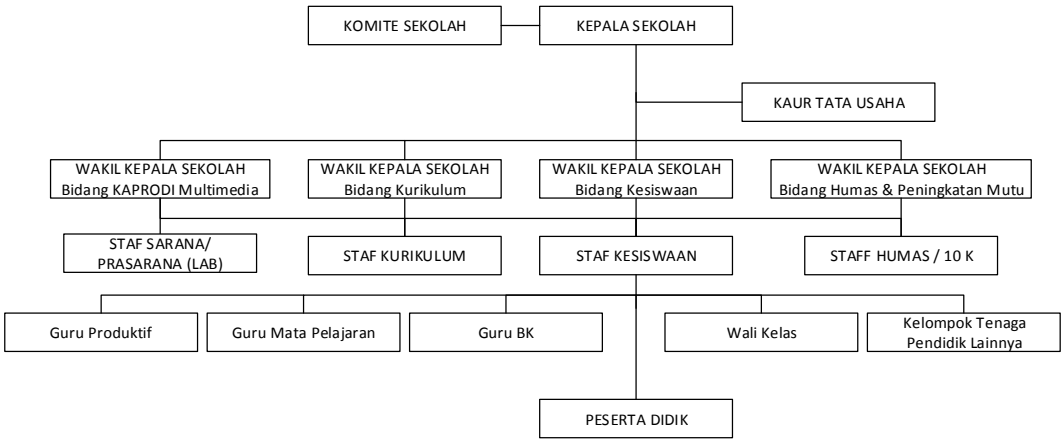
1. Menciptakan lingkungan sekolah yang bersih, sehat, indah dan nyaman
2. Membiasakan salam, senyum, sapa dan silaturahmi agar terbentuk suasana kekeluargaan yang kondusif bagi setiap warga sekolah
3. Menanamkan nilai-nilai keagamaan dalam kehidupan sehari-hari
4. Mampu berkompetisi dalam bidang imtaq dan iptek

2.1.3 Struktur Organisasi SMK Mandalasari Cikancung

Setiap organisasi mempunyai stuktur yang berbeda disesuaikan dengan karateristik masing-masing organisasi tersebut. Dengan adanya stuktur organisasi setiap anggota organisasi akan mengetahui dimana posisinya sehingga mereka dapat melakukan aktifitas sesuai dengan tugas, wewenang, dan tanggungjawab masing-masing.

Penyusunan stuktur organisasi harus fleksibel dalam arti memungkinkan adanya penyesuaian-penyesuaian tanpa harus mengadakan perubahan secara total. Stuktur organisasi juga harus dapat menunjukan garis wewenang dan tanggung jawab secara jelas, dan pemisahan fungsi-fungsi operasionalnya, sehingga memungkinkan tidak terjadinya overlapping dari fungsi masing-masing bagian. Stuktur organisasi yang digunakan SMK Mandalasari Cikancung adalah bentuk susunan organisasi garis, stasf dan fungsional dimana pimpinan memiliki tanggung

jawab yang penuh terhadap keputusan yang dikeluarkan dan bawahan harus mematuhi dan menjalankan sesuai prosedur. Untuk lebih jelasnya mengenai stuktur organisasi SMK Mandalasari Cikancung dapat dilihat pada Gambar 2.2.



Gambar 2.2 Struktur Organisasi SMK Mandalasari Cikancung

2.2 Multimedia

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi, audio dan video dengan alat bantu (tool) dan koneksi (link) sehingga pengguna dapat melakukan navigasi, berinteraksi, berkarya dan berkomunikasi. Multimedia sering digunakan dalam dunia informatika. Selain dari dunia informatika, multimedia juga diadopsi oleh dunia game, dan juga untuk membuat website.

Di dunia pendidikan, multimedia digunakan sebagai media pengajaran, baik dalam kelas maupun secara sendiri-sendiri atau otodidak. Di dunia bisnis, multimedia digunakan sebagai media profil perusahaan, profil produk, bahkan sebagai media kios informasi dan pelatihan dalam sistem e-learning.

Pada awalnya multimedia hanya mencakup media yang menjadi konsumsi indra penglihatan (gambar diam, teks, gambar gerak video, dan gambar gerak rekaan/animasi), dan konsumsi indra pendengaran (suara) dan juga berupa (berwujud). Dalam perkembangannya multimedia mencakup juga kinetik (gerak) dan bau yang merupakan konsumsi indra penciuman. [4].

2.3 Augmented Reality

2.3.1 Definisi Augmented Reality

Augmented reality (AR) atau dalam bahasa Indonesia disebut realitas tertambah adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. Bendabenda maya berfungsi menampilkan informasi yang tidak dapat diterima oleh manusia secara langsung. Hal ini membuat realitas tertambah berguna 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. [2]

Menurut definisi Ronald Azuma (1997), ada tiga prinsip dari *augmented reality*. Yang pertama yaitu *Augmented reality* merupakan penggabungan dunia nyata dan virtual, yang kedua berjalan secara interaktif dalam waktu nyata (*realtime*), dan yang ketiga terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Dalam perkembangannya saat ini *Augmented reality* tidak hanya bersifat visual saja, tapi sudah dapat diaplikasikan untuk semua indera, termasuk pendengaran, sentuhan, dan penciuman. Selain digunakan dalam bidang-bidang seperti kesehatan, militer, industri manufaktur, *Augmented reality* juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam.[2]

Ada banyak definisi dari *Augmented reality* tetapi asumsi umum adalah bahwa *Augmented reality* memungkinkan perspektif yang diperkaya dengan melapiskan objek virtual pada dunia nyata dengan cara yang mengajak penonton bahwa objek virtual adalah bagian dari lingkungan nyata. Oleh karena itu *Augmented reality* adalah perpaduan antara dunia nyata dan dunia virtual, sebagaimana diilustrasikan oleh diagram terkenal *Reality-Virtuality Continuum*. [2]

Dalam pembuatan AR menggunakan webcam sebagai perangkat untuk menangkap citra. Sebelum citra diubah ke dalam bentuk digital maka proses manipulasi citra digital tidak bisa dilakukan. Citra digital (f(x,y)) mempunyai dua

unsur. Unsur yang pertama merupakan kekuatan sumber cahaya yang melingkupi pandangan kita terhadap objek (illumination). Unsur yang kedua merupakan besarnya cahaya yang direfleksikan olah objek ke dalam pandangan mata kita atau disebut juga *reflectance components*. Kedua unsur tersebut dituliskan sebagai fungsi i(x,y) dan r(x,y). [1]

2.3.2 Perkembangan Augmented Reality

Penemuan tentang *Augmented reality* (AR) berawal dari tahun 1957-1962. Seorang sinematografer, bernama Morton Heilig, menciptakan dan mempatenkan sebuah alat simulator yang disebut Sensorama dengan visual, getaran dan bau, kemudian tahun 1966, Ivan Sutherland menemukan *head-mounted display* yang dia klaim adalah, jendela ke dunia virtual.

Tahun 1975 ilmuwan bernama Myron Krueger menciptakan *Videoplace* yang memungkinkan pengguna, dapat berinteraksi dengan objek virtual untuk pertama kalinya. Tahun 1989, Jaron Lanier, memeperkenalkan *Virtual Reality* kepada publik dan menciptakan bisnis komersial pertama kalinya. Tahun 1989, Jaron Lanier, memeperkenalkan *Virtual Reality* kepada publik dan menciptakan bisnis komersial pertama kali di dunia maya, Tahun 1992 *Augmented reality* dikembangkan untuk dapat melakukan perbaikan pada pesawat boeing, di tahun yang sama, LB Rosenberg mengembangkan Sistem *Augmented reality* yang digunakan di Angkatan Udara AS yang disebut Virtual Fixtures, dan 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 ditunjukkan 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* disebuah website, karena *output* yang dihasilkan FLARToolkit berbentuk Flash. Ditahun yang sama, Wikitude Drive meluncurkan sistem navigasi berteknologi AR di Platform Android. Tahun 2010, Acrossair menggunakan teknologi *Augmented reality* pada I-Phone 3GS. [2]

Berikut merupakan contoh dari penerapan teknologi *Augmented reality* sederhana dapat dilihat pada Gambar 2.3.



Gambar 2.3 Contoh Augmented reality Sederhana [1]

Augmented reality dapat di klasifikasikan menjadi dua berdasarkan ada tidaknya penggunaan marker yaitu:

1. MarkerAugmented Reality

Sebuah metode yang memanfaatkan marker yang biasanya berupa ilustrasi hitam dan putih berbentuk persegi atau lainnya dengan batas hitam tebal dan latar belakang putih. Melalui posisi yang dihadapkan pada sebuah kamera komputer atau smartphone, maka komputer atau smartphone tersebut akan melakukan proses menciptakan dunia virtual 2D atau 3D. Marker Based Tracking ini sudah lama dikembangkan sejak 1980 dan pada awal 1990 mulai dikembangkan untuk pengguna Augmented Reality.[2]

2. Markerless Augmented Reality

Salah satu metode *Augmented reality* yang saat ini sedang berkembang adalah metode Markerless Augmented Reality. Dengan menggunakan metode ini maka

pengguna tidak perlu menggunakan sebuah marker untuk menampilkan objek 3D atau yang lainnya. Sekalipun dengan markerless namun aplikasi tetap berjalan dengan melakukan pemindaian terhadap objek, namun ruang lingkup yang dipindai lebih luas disbanding dengan *Marker Based Tracking*.

Saat ini *Augmented reality* dikembangkan oleh perusahaan terbesar di dunia yaitu Total Imersion, yang telah membuat berbagai macam teknik Markerless Tracking sebagai teknologi andalan perusahaan, seperti Image Target, Face Tracking, 3D Object Tracking, dan Motion Tracking.

a. *Image Target*

   Image Target adalah sebuah metode dimana marker yang dibaca oleh sistem AR adalah sebuah Gambar, baik Gambar hitam putih atau Gambar berwarna.

b. *Face Tracking*

   Dengan menggunakan algoritma yang telah dikembangkan, komputer dapat mengenali wajah manusia secara umum dengan cara mengenali semua komponen wajah manusia kemudian akan mengabaikan objek- objek lain di sekitarnya seperti pohon, rumah, dan benda-benda lainnya.

c. *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 sebagainya.

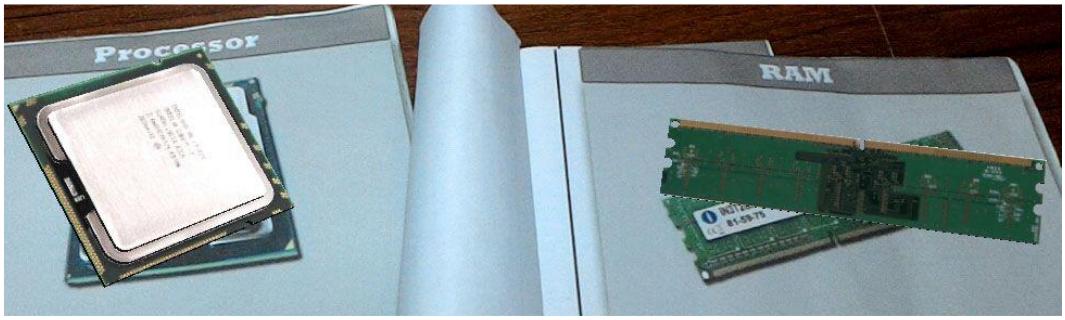
d. *Motion Tracking*

   Teknik ini merupakan teknik yang membuat komputer dapat menangkap gerakan. Motion Tracking telah mulai digunakan secara ekstensif untuk memproduksi film-film yang mencoba mensimulasikan gerakan. Contohnya pada film Avatar, dimana James Cameron menggunakan teknik ini untuk membuat film tersebut dengan menggunakannya secara *real-time*.

2.3.3 Multi Marker

Multi marker adalah merupakan sebuah metode perkembangan dari single marker, dimana kamera men-*tracking* objek lebih dari satu.

Dalam implementasinya dapat dilakukan dengan menggunakan beberapa pendekatan metode yang dapat dilakukan seperti pelabelan komponen serta corner detection sebagai pengenalan sudut dan beberapa bentuk marker.



Gambar 2.4 Contoh Multi Marker [2]

2.3.4 Display Augmented Reality

Dari semua modalitas pada input sensorik manusia, penglihatan, suara, dan sentuhan adalah indera yang saat ini berlaku pada sistem AR. Dibagian ini terutama berfokus pada *visual displays*, namun *aural (suara) displays* dijelaskan secara singkat di bawah ini.

1. *Aural Display* (Suara)

Aplikasi *aural display* pada AR kebanyakan terbatas pada *mono* (0-dimensi), *stereo* (1-dimensi), atau *surround* (2-dimensi) headphone dan *loudspeaker*. Tiga dimensi *aural display* yang sebenarnya saat ini ditemukan dalam simulasi yang lebih mendalam dari lingkungan *virtual* atau masih dalam tahap percobaan.

Haptic audio mengacu pada suara yang dirasakan daripada didengar dan telah digunakan pada perangkat konsumen seperti headphone *Turtle Beach* untuk meningkatkan rasa pengaruh dan kenyataan, tetapi juga untuk meningkatkan antarmuka pengguna misalnya *mobile phone*. Perkembangan terakhir di area ini disajikan dalam workshop seperti workshop internasional *Haptic Audio Visual Environments* dan *Haptic and Audio Interaction Design*.

2. *Visual Display*

Pada dasarnya ada tiga cara untuk menyajikan secara visual sebuah AR. Paling dekat dengan *virtual reality* (VR) adalah video seethrough, dimana lingkungan virtual digantikan oleh sebuah video feed realitas dan *augmented reality* (AR) dilapisi atas gambar digital. Cara lain yang mencakup pendekatan Sutherland adalah *optical see-through* dan meninggalkan persepsi dunia nyata tetapi menampilkan hanya hamparan AR melalui cermin dan kamera. Pendekatan ketiga adalah memproyeksikan hamparan AR ke objek nyata itu sendiri sehingga menghasilkan tampilan proyektif.

3. *Video See-Through*

Selain menjadi yang termurah dan termudah dalam implementasi, teknik display ini menawarkan keuntungan sebagai berikut. Sejak realitas di-digital-kan, ini menjadi lebih mudah untuk dimediasikan atau menghapus objek dari kenyataan. Ini termasuk menghapus dan mengganti marker *fiducial* atau penampung dengan objek-objek virtual. Dan juga, brightness dan contrast dari objek virtual dicocokkan dengan mudah dengan lingkungan nyata. Mengevaluasi kondisi cahaya dari suasana luar ruangan yang statis adalah penting ketika konten yang dihasilkan komputer telah berbaur halus.

Gambar digital memungkinkan pelacakan gerakan kepala untuk registrasi yang lebih baik. Ini juga menjadi mungkin untuk mencocokkan persepsi delay dari yang nyata dan yang virtual. Kekurangan *video seethrough* termasuk resolusi rendah realitas, *field-of-view* yang terbatas (meskipun bisa dengan mudah ditingkatkan), dan disorientasi pengguna karena paralaks (*eye-offset*) karena posisi kamera pada jarak dari lokasi mata pengamat, menyebabkan upaya penyesuaian yang signifikan bagi pengamat. Masalah ini dipecahkan di laboratorium *mixed reality* dengan menyelaraskan video capture. Kelemahan terakhir adalah jarak fokus dari teknik yang cocok pada kebanyakan tipe display, menyediakan akomodasi *poor-eye*. Beberapa pengaturan *head-mounted* bagaimanapun bisa menggerakkan display

(atau lensa di depannya) untuk melingkupi jarak 0,25 meter hingga tidak terbatas dalam 0,3 detik. Seperti masalah paralaks, *biocular* display (dimana kedua mata melihat gambar yang sama) karena secara signifikan lebih tidak nyaman daripada *monocular* atau *binocular* display, keduanya dalam ketegangan dan kelelahan mata.

4. *Optical See-Through*

Teknik *optical see-through* dengan *beam-splitting* holographic optical elements (HOEs) dapat diterapkan pada *head-worndisplay, hendheld display*, dan pengaturan spatial dimana hamparan AR tercermin baik dari layar planar atau melalui layar curve. Display ini tidak hanya meninggalkan resolusi dunia nyata utuh, mereka juga memiliki keuntungan menjadi lebih murah, lebih aman, dan bebas paralaks (tidak ada *eye-offset* karena posisi kamera). Teknik optikal lebih aman karena pengguna masih dapat melihat saat power fails, membuat teknik ini ideal untuk tujuan militer dan medis. Namun, perangkat input lainnya seperti kamera diperlukan untuk interaksi dan registrasi. Dan juga, menggabungkan objek virtual secara holografik melalui cermin dan lensa transparan menciptakan kerugian yaitu berkurangnya kecerahan dan kontras kedua gambar dan persepsi dunia nyata, membuat teknik ini kurang cocok untuk digunakan di luar ruangan.Semua *field of view* yang penting terbatas untuk teknik ini dan dapat menyebabkan *clipping* gambar virtual pada ujung cermin atau lensa.Akhirnya,*occlusion* (saling menutupi) *or mediation* dari objek nyata menjadi sulit karena cahaya mereka selalu bergabung dengan gambar virtual.Kiyowaka dkk memecahkan masalah ini untuk *head-worn display* dengan menambahkan lapisan buram menggunakan panel LCD dengan pixel yang memburamkan area menjadi tertutupi.

Virtual retina displays atau *retinal scanning displays* (RSDs) memecahkan masalah brightness dan *field-of-view* yang rendah pada (head-worn) *optical see-through display*. Sebuah laser berdaya rendah menarik gambar virtual langsung ke retina yang menghasilkan brightness yang tinggi dan *field-of-view* yang luas. Kualitas RSD tidak dibatasi oleh ukuran pixel tetapi hanya oleh difraksi dan penyimpangan (*diffraction and abberrations*) pada sumber cahaya, sehingga

memungkinkan resolusi yang (sangat) tinggi. Bersama dengan konsumsi daya yang rendah display ini sangat cocok untuk penggunaan luar ruangan.

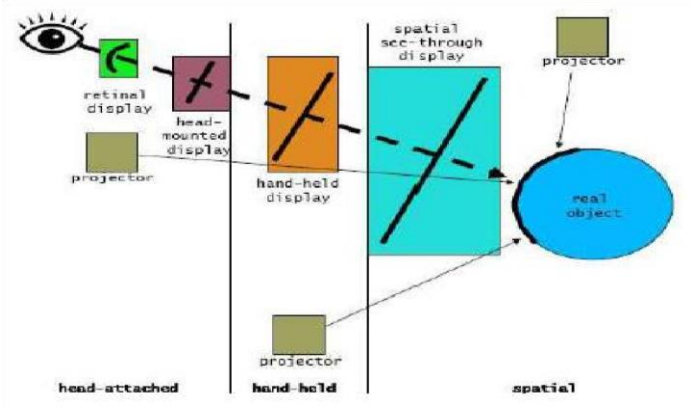
5. *Projective*

Alat display ini memiliki keuntungan tidak memerlukan *eye-wear* khusus sehingga mengakomodasikan mata pengguna selama fokus, dan bisa menutupi permukaan yang besar untuk sebuiah *field-of-view* yang luas. Permukaan proyeksi dapat berkisar dari datar, dinding berwarna datar, hingga model skala kompleks.

Zhou dkk mendaftarkan beberapa pikoproyektor yang ringan dan rendah konsumsi daya untuk integrasi yang lebih baik. Namun, seperti *optical see-through displays*, perangkat input lainnya dibutuhkan untuk (tidak langsung) interaksi. Dan juga, proyektor harus dikalibrasi setiap kali lingkungan atau jarakke permukaan proyeksi berubah. Untungnya, kalibrasi dapat diotomatiskan menggunakan kamera pada contohnya sebuah *cave automatic virtual environment*(CAVE) berdinding banyak dengan permukaan tidak teratur. Selain itu, jenis display ini terbatas pada pnggunaan dalam ruangan (indoor) karena brighness dan kontras yang rendah dari gambar yang diproyeksikan. *Oklusi* atau *mediasi* objek juga cukup lemah, tapi untuk *head-worn* proyektor ini dapat ditingkatkan dengan menutupi permukaan dengan material *retro-reflective*. [2]

6. Display Positioning

Display AR dapat diklasifikasikan ke dalam tiga kategori berdasarkan pada posisi mereka diantara pengamat dan lingkungan nyata, yaitu head-worn, hand-held, dan spatial.



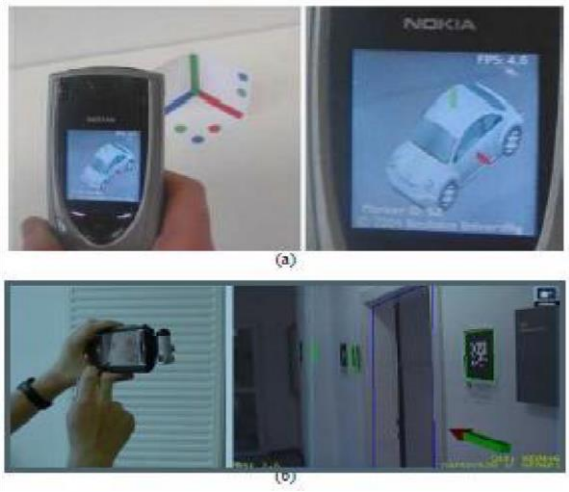
Gambar 2.5 Teknik Posisi Visual Display

7. Head-Worn

Visual display yang dilekatkan pada kepala termasuk *video/optical see-through* HMD (*head-mounted display*), *virtual retinal display* (VRD), dan *head-mounted projective display* (HMPD). Cakmakci dan Rolland [9] memberikan sebuah detil review terakhir dari teknologi *head-worn display*. Kelemahan saat ini dari *head-worn displays* adalah kenyataan bahwa merekaharus terhubung ke komputer grafis seperti laptop yang membatasi mobilitas karena terbatasnya daya baterai. Daya baterai dapat diperpanjang dengan memindahkan perhitungan ke lokasi yang jauh (*clouds*) dan menyediakan koneksi (wireless) menggunakan standar seperti IEEE 802.11 atau BlueTooth.

8. *Hand-held*

Kategori ini termasuk *video/optical see-through* genggam serta proyektor genggam. Meskipun kategori display ini lebih besar dari *headworn display*, saat ini merupakan kinerja terbaik untuk memperkenalkan AR ke pasar karena biaya produksi yang rendah dan mudah digunakan. Misalnya, *video see-through* genggam AR bertindak sebagai kacamata pembesar mungkin didasarkan pada produk konsumen yang ada sepeti telepon genggam (Gambar 2.6a) yang menunjukkan objek 3D, atau *personal digital assistant* (PDA) (Gambar 2.6b) dengan misalnya informasi navigasi.



Gambar 2.6 Hand-held video see-through display

9. *Spatial*

Kategori terakhir display AR adalah ditempatkan secara statis di dalam lingkungan dan termasuk *video see-through display* berbasis layar, *spatial optical see-through display*, dan *projective display*. Teknik ini baik untuk presentasi dan pameran besar dengan interaksi terbatas. Cara awal membuat AR adalah didasarkan pada layar konvensional (komputer atau televisi) yang menunjukkan hasil tangkapan kamera dengan hamparan AR. Teknik ini sekarang sedang diterapkan di dunia televisi olahraga dimana lingkungan seperti kolam renang dan trek balapan didefinisikan dengan baik dan mudah untuk ditambahkan. Head-up displays (HUDs) di kokpit militer adalah bentuk dari *spatial optical seethrough*dan menjadi sebuah tambahan standar untuk mobil produksi untuk memproyeksikan arah navigasi di kaca depan mobil. Sudut pandang pengguna relatif terhadap hamparan AR hampir tidak berubah dalam kasusu ini karena ruang terbatas. *Spatial see-through display* dapat bagaimanapun muncul sejajar ketika pengguna bergerak di ruang terbuka, misalnya saat hamparan AR disajikan pada sebuah layar transparan.

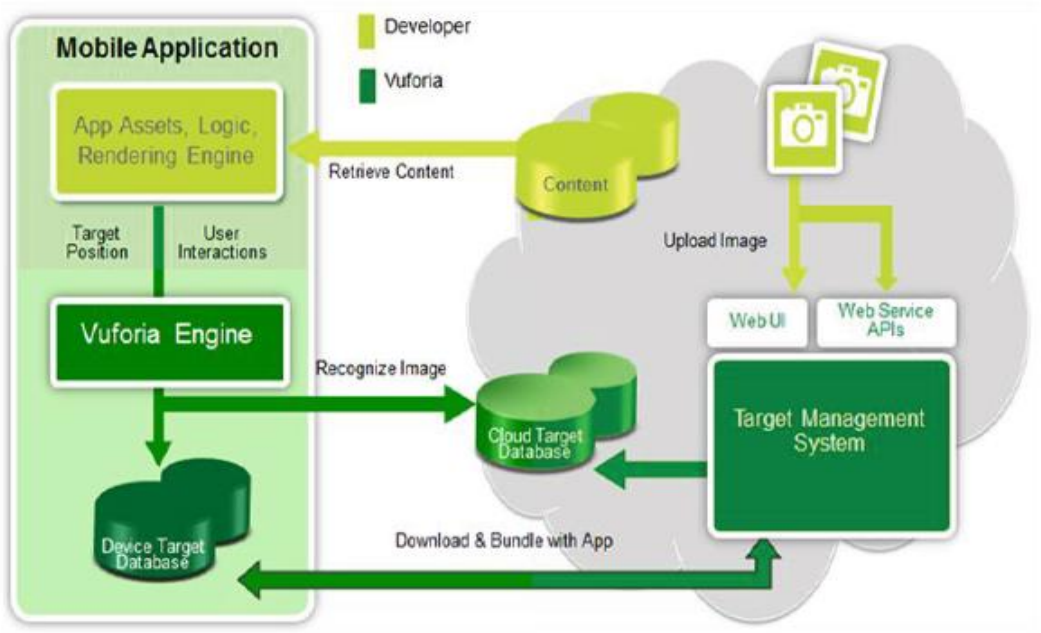
2.4 Vuforia

Vuforia merupakan *library* untuk *Augmented Reality* yang dikembangkan oleh Qualcomm, yang menggunakan sumber yang konsisten mengenai *Computer Vision* yang fokus pada *image recognition.* Vuforia mempunyai banyak fitur-fitur

dan kemampuan, yang dapat membantu pengembang untuk mewujudkan pemikiran mereka tanpa adanya batas secara teknikal, dengan *support* untuk kios, Android dan Unity3D, *platform* Vuforia mendukung para pengembang untuk membuat aplikasi yang dapat digunakan di hampir seluruh jenis *device smartphone, dekstop* dan *tablet*.

Pengembang juga diberikan kebebasan untuk mendesain dan membuat aplikasi yang mempunyai kemampuan antara lain :

1. Teknologi *Computer Vision* tingkat tinggi yang mengijinkan *developer* untuk membuat efek khusus pada *mobile Service*.
2. Terus-menerus mengenali *multiple image.*
3. *Tracking* dan *Detection* tingkat lanjut.
4. Solusi pengaturan *database* gambar yang fleksibel



Gambar 2.7 Struktur Vuforia

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 dan kartu ucapan.

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

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

2.4.1 Arsitektur Vuforia

Vuforia SDK memerlukan beberapa komponen penting agar dapat bekerja dengan baik. Komponen-komponen tersebut antara lain:

a. Kamera

Kamera dibutuhkan untuk memastikan bahwa setiap *frame* ditangkap dan diteruskan secara efisian ke *tracker*. Para developer hanya tinggal memeberi tahu kamera kapan mereka mulai menangkap dan berhenti.

b. *Image Converter*

Mengkonfersi format kamera (misalnya YUV12 kedalam format yang dapat dideteksi oleh OpenGL (misalnya RGB565) untuk *tracking* (misalnya *luminance*).

c. *Tracker*

Mengandung algoritma *computer vision*untuk melaukan pelacakan objek nyata yang diambil dari kamera. Objek tersebut dievaluasi dan hasilnya akan disimpan dalam *statae object* yang akan digunakan oleh video *background renderer* dan dapat dikases dari *application code*.

d. *Video Background Render*

Me-*render* gambar dari kamera yang tersimpan di dalam *state object*. Performa dari *video background render* sangat bergantung pada *device* yang digunakan.

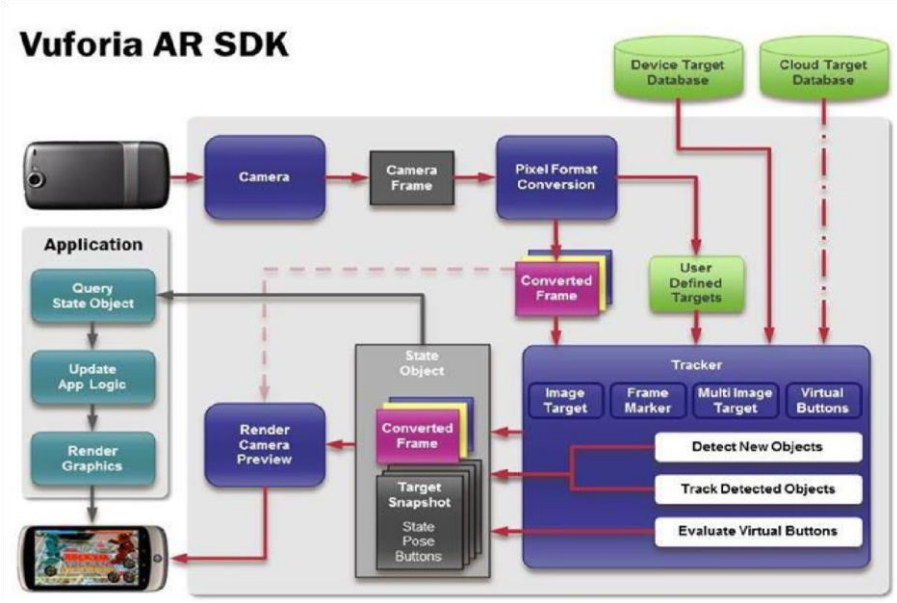
e. *Application Code*

Melibatkan analisasi dari semua komponen diatas. Selama objek yang dikehendaki diubah prosesnya, maka *application code* harus diubah berdasarkan lokasi objek virtual.

f. *Target Resource*

Dibuat menggunakan online *Target Management System*. Assets di unduh berisi sebuah konfigurasi xml config.xml yang memungkinkan developer untuk mengkonfigurasi beberapa fitur dalam trackable dan binary file yang berisi database trackable.

Berikut adalah gambaran dari diagaram alir data Vuforia, dapat dilihat pada Gambar 2.8.



Gambar 2.8 Diagram Alir Data Vuforia

2.4.2 Vuforia API Reference

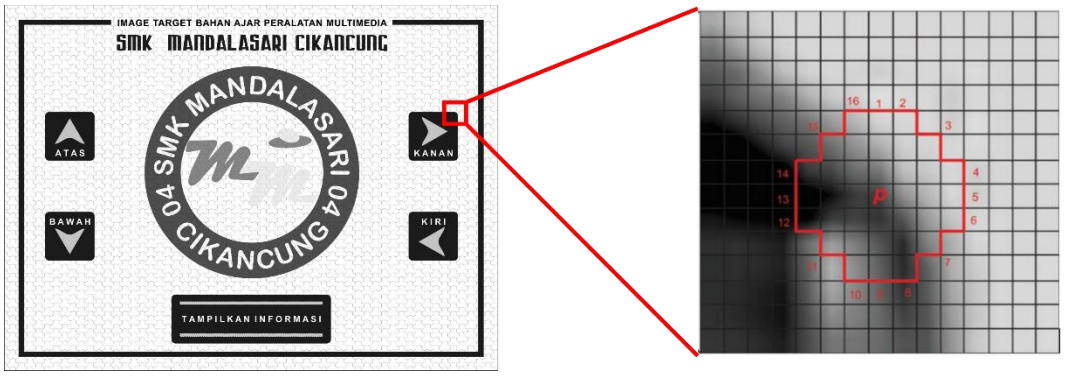
API reference berisi informasi tentang hierarki kelas dan fungsi ember dari QCAR SDK. Sistem dari QCAR SDK menyediakan callback event. Contoh : sebuah image baru yang tersedia.

a. High-level access ke perangkat keras. Contoh : kamera start / stop.
b. Multiple trackables.
c. Interaksi secara langsung dengan dunia nyata.

2.4.3 Metode FAST Corner Detection

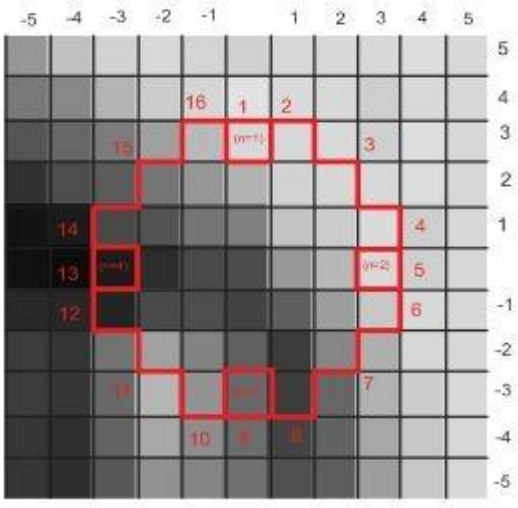
Inisialisasi *feature* pada *library Qualcomm* Vuforia ini menggunakan metode *FAST Corner Detection* untuk pembuatan marker sehingga membantu *developer* membuat sebuah penanda berupa *image target*. Berikut merupakan cara kerja metode *FAST Corner Detection*.

1. Hal yang pertama dilakukan metode *Fast Corner Detection* adalah menentukan sebuah titik *p* pada citra denga posisi awal (x_p,y_p).



Gambar 2.9 Titik awal nilai p

2. Tentukan keempat titik. 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 terletak pada koordinat (*n*=3) terletak pada koordinat (x_p, y_{p-3}), titik keempat (*n*=4) terletak pada koordinat (x_{p-3},y_p).



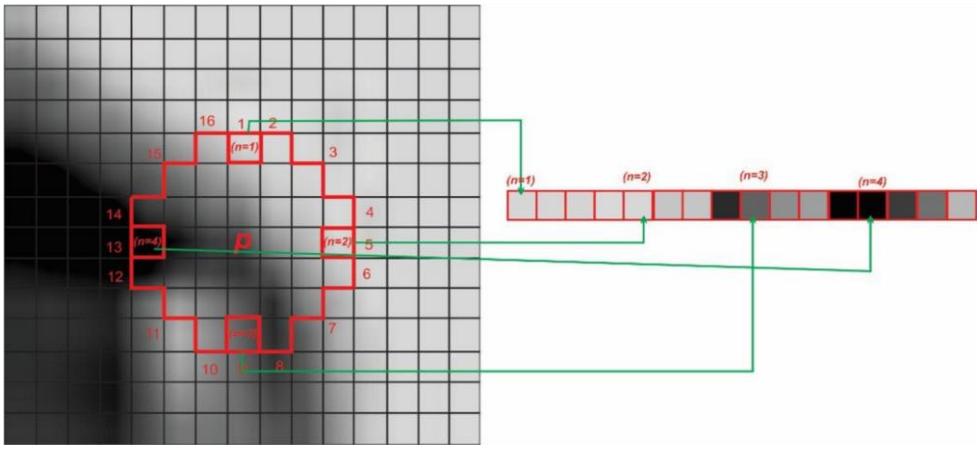
Gambar 2.10 Empat Titik Koordinat

3. Bandingkan intensitas titik pusat *p* dengan keempat titik disekitar. Jika nilai intensitas dititik *p* bernilai lebih besar atau lebih kecil daripada intensitas tiga titik disekitarnya ditambah dengan suatu intensitas batas ambang, maka dapat dikatakan bahwa titik *p* adalah suatu sudut.

$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \le I_p - t \quad \text{(darker)} \\ s, & I_p - t < I_{p\to x} < I_p + t \quad \text{(similar)} \\ b, & I_p + t \le I_{p\to x} \quad \text{(brighter)} \end{cases}$$

Keterangan:

Sp→x = Keputusan titik p sebagai sudut, nilai 1 menunjukan bahwa titik merupakan suatu sudut, dan nilai 0 menunjukan bahwa titik bukanlah sebuah sudut. *Ip* = Nilai intensitas titik *p t* = Batas ambang nilai intensitas yang ditoleransi.



Gambar 2.11 Menentukan Titik atau Sudut

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

2.5 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan

mendokumentasikan sistem piranti lunak.UML menawarkan sebuah standar untuk merancang model sebuah sistem.

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.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memilikimakna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). [11]

2.5.1 Diagram UML

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

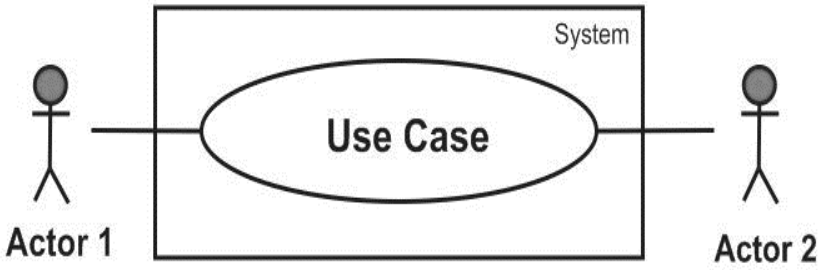
1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (message)
4. antar objek.
5. *Collaboration Diagram* untuk memodelkan interaksi antar objek.
6. *State Diagram*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

9. *Objek Diagram* untuk memodelkan struktur objek.
10. *Component Diagram* untuk memodelkan komponen objek.
11. *Deployment Diagram* untuk memodelkan distribusi aplikasi. Berikut akan dijelaskan 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case diagram, Activity diagram, sequence diagram* dan *class diagram.*

2.5.1.1 Use Case Diagram

Use case diagram adalah gambaran *graphical* dari beberapa atau semua *actor, use-case* dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. *Use-case diagram* menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.

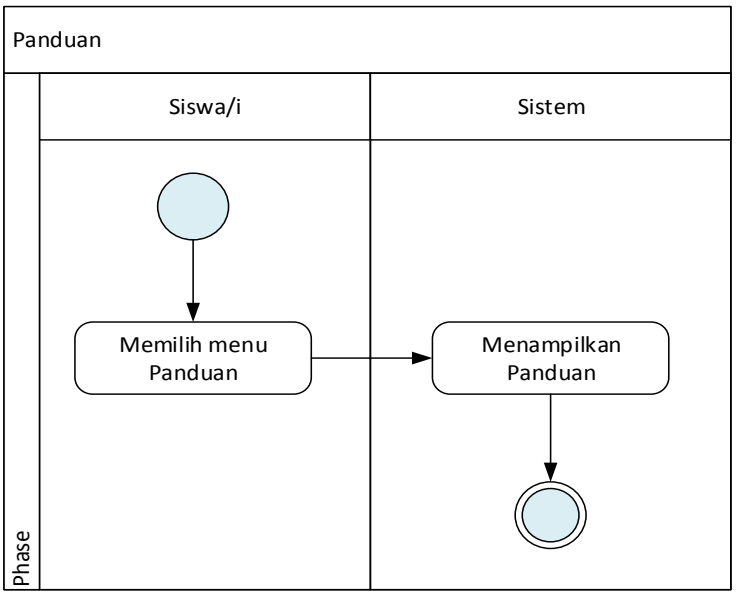
Use-case diagram dapat digunakan selama proses analisis untuk menangkap *requirement system* dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use-case diagram* berperan untuk menetapkan perilaku sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use-case diagram*. Kebutuhan atau *requirement system* adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan pada model *use-case* yang menggambarkan fungsi sistme yang diharapkan, dan yang mengelilinginya, serta hubungan antara actor dengan use-case itu sendiri. Berikut merpuakan contoh gambar *Use Case* diagram dapat dilihat pada Gambar 2.12.



Gambar 2.12 Contoh Use Case Diagram

2.5.1.2 Activity Diagram

Diagram yang menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat digunakan untuk aktivitas lainnya seperti *Use Case*. Berikut merupakan contoh activity diagram dapat dilihat pada Gambar 2.13.



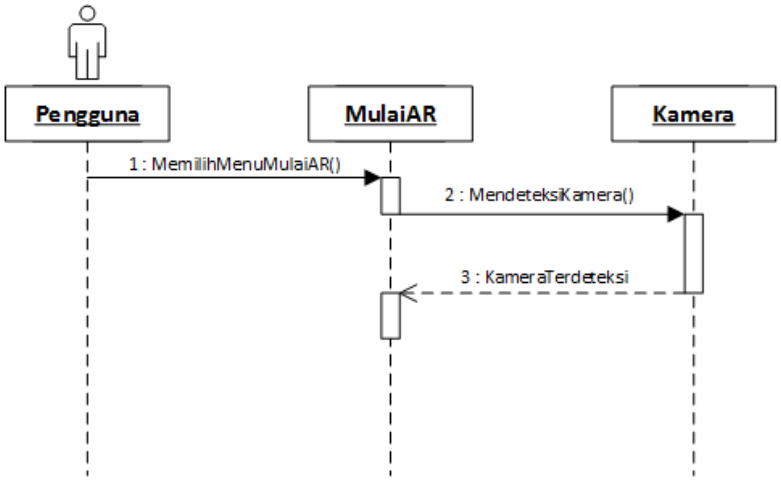
Gambar 2.13 Contoh Activity Diagram

2.5.1.3 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut

kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

Dimensi vertikal merepresentasikan waktu. Bagian paling atas dari diagram menjadi titik awal dan waktu berjalan ke bawah sampai dengan bagian dasar dari diagram. Garis vertical, disebut *lifeline*, dilekatkan pada setiap obyek atau aktor. Kemudian, *lifeline* tersebut digambarkan menjadi kotak ketika obyek melakukan suatu operasi, kotak tersebut disebut *activationbox.* Obyek dikatakan mempunyai *live activation* pada saat tersebut. Pesan yang dipertukarkan antar obyek digambarkan sebagai sebuah anak panah antara activation box pengirim dan penerima. Kemudian diatasnya diberikan label pesan. Berikut merupakan contoh gambar *squence* diagram dapat dilihat pada Gambar 2.14



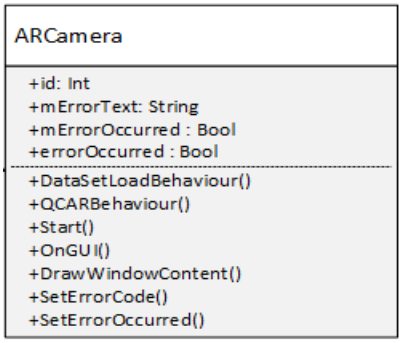
Gambar 2.14 Contoh Sequence Diagram

2.5.1.4 Class Diagram

Class diagram adalah alat perancangan terbaik untuk tim pengembang. Diagram tersebut membantu pengembang mendapatkan struktur sistem sebelum kode ditulis, dan membantu untuk memastikan bahwa sistem adalah desain terbaik.

Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam *system*. *Class diagram* memberikan gambaran *system* secara statis dan relasi antar mereka. Biasanya, dibuat beberapa *class diagram* untuk sistem tunggal. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya. Dapat dibuat beberapa diagram sesuai dengan yang diinginkan untuk mendapatkan

gambaran lengkap terhadap *system* yang dibangun. Berikut merupakan contoh *class* diagram dapat dilihat pada Gambar 2.15.



Gambar 2.15 Contoh Class Diagram

2.5.1.5 Library

Library adalah tempat dimana kita menyimpan dan mengelola symbol yang dibuat dalam Flast, seperti juga file-file yang diimpor, termasuk gambar bitmap, soud file, dan video klip. Panel library memberikan kita kebebasan untuk mengelola banyak item kedalam folderfolder, melihat beberapa sering sebuah item digunakan dalam dokumen, dan menyortir item berdasarkan tipenya.

2.6 Unity 3D

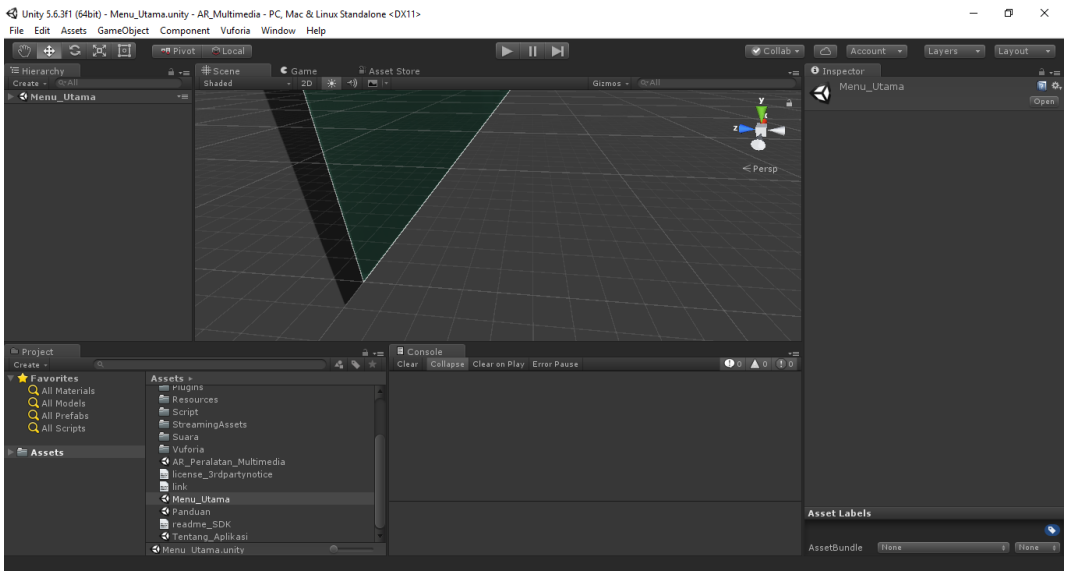
Unity 3D adalah sebuah *engine* untuk mengembangkan *game* berbasis 3D dan *multiplatform*. Dengan Unity, *developer* dapat melakukan implementasi dan bisa dipakai untuk berbagai *platform* seperti Unity Web, Windows, Mac, Android, iOS, XBox, Playstation 3 dan Wii. Sejak Unity 3D merilis versi 4.3, banyak penambahan fitur pada rilisnya, salah satunya fitur 2D. Jika sebelumnya kebanyakan *developer* menggunakan *plug-in* tambahan untuk membuat *game* 2D dengan Unity, kali ini *developer* bisa memilih *workspace* sendiri dalam pembuatan projek baru, 2D atau 3D [8].

Unity 3D dapat mengolah beberapa data seperti gambar tiga dimensi, suara, tekstur, dan lain sebagainya. Keunggulan dari *unity engine* ini yaitu, dapat menangani grafik dua dimensi dan tiga dimensi. Namun *engine* ini lebih konsentrasi

pada pembuatan grafik tiga dimensi. Dari beberapa *game engine* yang sama-sama menangani grafik tiga dimensi.

Editor Unity 3D menyediakan beberapa alat untuk memepermudah pengembangan yaitu *Unity Tree* dan *Terrain Creator* atau mempermudah pembuatan *vegetasi* dan *terrain* serta *MonoDevelop* untuk proses pemrograman.

Tampilan dari software Unity3D *engine* dapat dilihat pada Gambar 2.16.



Gambar 2.16 Tampilan Unity 3D

2.7 Autodesk 3Ds Max 2014

Autodesk 3Ds Max adalah sebuah software yang dikhususkan dalam pemodelan 3 dimensi ataupun untuk pembuatan animasi 3 dimensi. Selain terbukti andal untuk digunakan dalam pembuatan objek 3 dimensi, Autodesk 3Ds Max juga banyak digunakan dalam pembuatan desain furnitur, konstruksi, maupun desain interior. Selain itu, Autodesk 3Ds Max juga sering digunakan dalam pembuatan animasi atau film kartun [12].

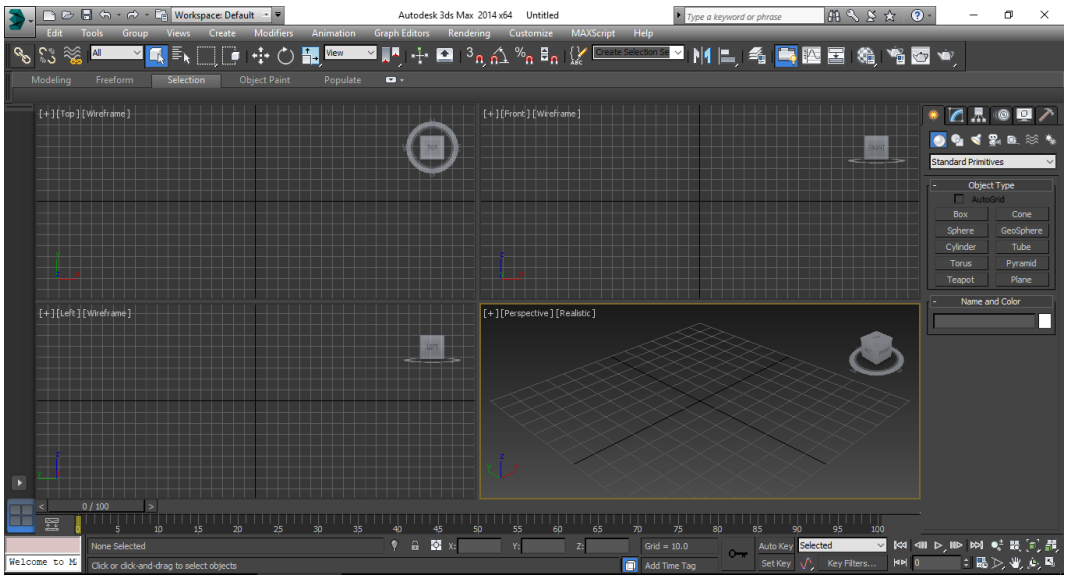
Autodesk 3Ds Max yang dilengkapi dengan bahasa scripting (MaxScript) juga terbukti ampuh untuk membuat game 3 dimensi, mulai dari yang sederhana hingga yang rumit sekalipun. Dengan kemampuan tersebut, banyak orang maupun instansi memanfaatkan software Autodesk 3Ds Max untuk membuat suatu desain atau iklan yang berguna sebagai media publikasi produk atau karya mereka kepada

publik. Autodesk 3Ds Max memungkinkan pengguna untuk membuat tampilan 3 dimensi yang sangat menarik.

Autodesk 3ds Max memberikan tiga kemungkinan untuk menentukan sistem koordinat sebuah titik dalam ruang, yaitu dengan memperlihatkan terhadap sumbu-sumbu x, y, z dan sudut yang terjadi. Ketiga kemungkinan sistem koordinat itu ialah:

a. Koordinat *Cartesian* (*rectangular coordinat*).
   Menentukan koordinat dengan menggunakan sumbu-sumbu x, y, z. yaitu (x), (y), (z). Penulisannya (0.5,0.9,0.0); (0.42,0.39,0.82)
b. Koordinat *cylindrical*
   Cara ini mengabungkan antara jarak, sudut dan koordinat sumbu z yaitu: (jarak)< (sudut), (z) Penulisannya: (.03<60.95,0.0);(0.57<43,0.82)
c. Koordinat *spherical*
   Cara ini menggabungkan antara jarak dan dua sudut, dan masingmasing besaran dipisahkan dengan tanda<, yaitu: (jarak)<(sudut)<(sudut), penulisannya: (1.03<60.95<0); (1<43<55). [12]

Tampilan dari software Autodesk 3Ds Max dapat dilihat pada Gambar 2.17.



Gambar 2.17 Tampilan Autodesk 3Ds Max 2014

2.8 Alpha Testing

Didalam pengembangan software, pengujian pada umumnya sangat diperlukan sebelum perangkat lunak tersebut dipublikasikan. Pengujian yang di kenal dengan *alpha* testing sering dilakukan di bawah suatu *debugger* atau dengan *hardware-assisted* yang *debugging* untuk menangkap *bugs* dengan cepat. Teknik ini juga dikenal sebagai *white-box testing*.

Selanjutnya diserahkan staff pengujian untuk pemeriksaan tambahan di dalam lingkungan yang serupa. Teknik ini dikenal juga dengan *black-box testing* dan sering di sebut langkah lanjutan dari *white-box testing/alpha testing*.

Pengujian alpha merupakan pengujian fungsional yang digunakan untuk menguji sistem yang baru. Pengujian alpha berfokus pada persyaratan fungsional perangkat lunak.

2.9 Beta Testing

Pengujian *beta* merupakan pengujian langsung kepada pengguna untuk mencoba aplikasi yang baru. Pengujian *beta* diadakan dilingkungan *live* (sebenarnya), dalam pengujian *beta end user* mencatat kemudian menyampaikan pada pihak devloper.

Pengujian dilakukan pada satu atau lebih pelanggan oleh pemakai akhir perangkat lunak dalam lingkungan yg sebenarnya, pengembang biasanya tidak ada pada pengujian ini. Pelanggan merekan semua masalah (*real* atau *imajiner*) yg ditemui selama pengujian dan melaporkan pada pengembang pada interval waktu tertentu. Pengujian bertujuan untuk meningkatkan jumlah para pemakai di masa yang akan datang.

2.10 Pengujian Skala Likert

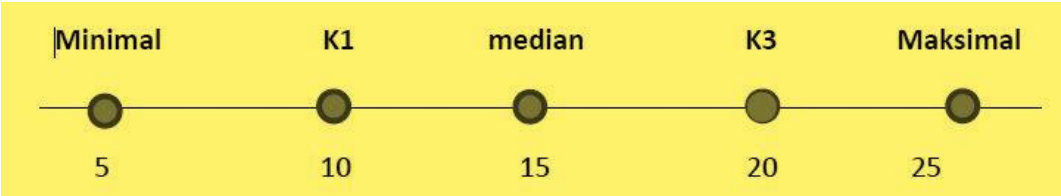
Skala *likert* adalah suatu skala psikometrik yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. *Rensis likert* telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama skala *likert*. Skala *likert* ini merupakan skala yang dapat dipergunakan

untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena.

Dalam skala *likert* terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif dan sikap negatif yang berfungsi untuk mengukur sikap negative objek. Contoh skor pernyataan positif dimulai dari 1 untuk tidak setuju (STS), 2 untuk tidak setuju (TS), 3 untuk raguragu (R), 4 untuk setuju (S), dan 5 untuk sangat setuju (SS). Skor pernyataan negative dimulai dari 1 untuk sangat setuju (SS), 2 untuk setuju (S), 3 untuk raguragu (R), 4 untuk tidak setuju (TS), dan 5 untuk sangat tidak setuju (STS). Beberapa menghilangkan option "Ragu-ragu" dalam instrument untuk memudahkan dalam melihat angket yang responden isikan. Skala *likert* digunakan untuk mengukur kesetujuan dan ketidak setujuan seseorang terhadap sesuatu rencana program, pelaksanaan program ataupun tingkat keberhasilan suatu program.

Berikut adalah tahapan untuk mengetahui bagainana sikap tiap responden:

1. Menentukan skor maksimal, yaitu skor jawaban terbesar di kali banyak pertanyaan. 5 x 5 = 25
2. Menentukan skor minimal, yaitu skor jawaban terkecil dikali banyak pertanyaan. 1 x 5 = 5
3. Menentukan nilai median, yaitu hasil penjumlahan skor maksimal dengan skor minimal dibagi dua. (25+5) : 2 = 15
4. Menentukan nilai kuartil 1, yaitu hasil penjumlahan skor minimal dengan median dibagi dua. (5+15) : 2 = 10
5. Menentukan nilai kuartil 3, yaitu hasil penjumlahan skor maksimal dengan median dibagi dua. (25+15) : 2 = 20
6. Buatlah skala yang menggambarkan skor minimal, nilai kuartil 1, median, kuartil 3 dan skor maksimal.



Gambar 2.18 Contoh Skala Likert