BAB 2
LANDASAN TEORI

2.1 Sekolah Dasar

Sekolah dasar (disingkat SD) adalah jenjang paling dasar pada pendidikan formal di Indonesia. Sekolah dasar ditempuh dalam waktu 6 tahun, mulai dari kelas 1 sampai kelas 6. Saat ini murid kelas 6 diwajibkan mengikuti Ujian Nasional (dahulu Ebtanas) yang memengaruhi kelulusan siswa. Lulusan sekolah dasar dapat melanjutkan pendidikan ke sekolah menengah pertama (atau sederajat).

Pelajar sekolah dasar umumnya berusia 7-12 tahun. Di Indonesia, setiap warga negara berusia 7-15 tahun tahun wajib mengikuti pendidikan dasar, yakni sekolah dasar (atau sederajat) 6 tahun dan sekolah menengah pertama (atau sederajat) 3 tahun.

Sekolah dasar diselenggarakan oleh pemerintah maupun swasta. Sejak diberlakukannya otonomi daerah pada tahun 2001, pengelolaan sekolah dasar negeri (SDN) di Indonesia yang sebelumnya berada di bawah Departemen Pendidikan Nasional, kini menjadi tanggung jawab pemerintah daerah kabupaten/kota. Sedangkan Departemen Pendidikan Nasional hanya berperan sebagai regulator dalam bidang standar nasional pendidikan. Secara struktural, sekolah dasar negeri merupakan unit pelaksana teknis dinas pendidikan kabupaten/kota.[1]

2.2 Buku Tematik Terpadu

Buku Tematik Kurikulum 2013 SD, Pembelajaran tematik adalah pembelajaran terpadu yang menggunakan tema untuk mengaitkan beberapa mata pelajaran sehingga dapat memberikan pengalaman bermakna kepada peserta didik. Tema adalah pokok pikiran atau gagasan pokok yang menjadi pokok pembicaraan.

Kurikulum 2013 adalah kurikulum berbasis kompetensi. Di dalamnya dirumuskan secara terpadu kompetensi sikap, pengetahuan, dan keterampilan

yang harus dikuasai peserta didik. Juga dirumuskan proses pembelajaran dan penilaian yang diperlukan peserta didik untuk mencapai kompetensi yang diinginkan itu. Buku yang ditulis dengan mengacu pada kurikulum 2013 ini dirancang dengan menggunakan proses pembelajaran yang sesuai untuk mencapai kompetensi yang sesuai dan diukur dengan proses penilaian yang sesuai.

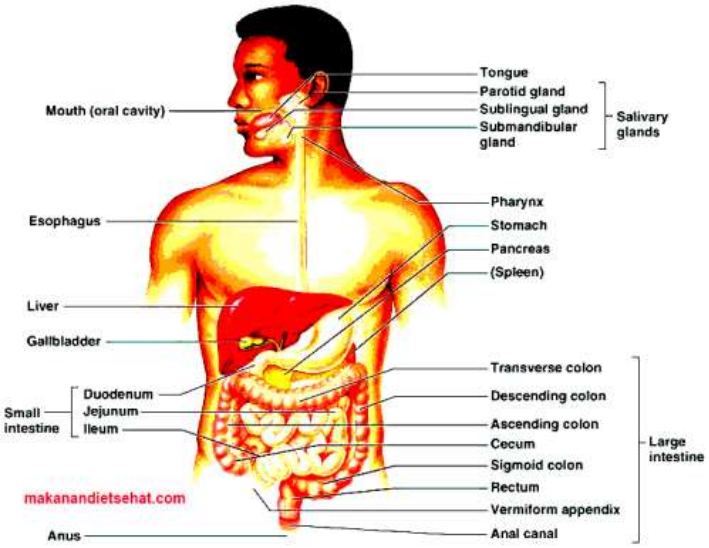
Sejalan dengan itu, kompetensi yang diharapkan dari seorang lulusan SD/MI adalah kemampuan pikir dan tindak yang produktif dan kreatif dalam ranah abstrak dan konkret. Kemampuan itu diperjelas dalam kompetensi inti, yang salah satunya, "menyajikan pengetahuan dalam bahasa yang jelas, logis dan sistematis, dalam karya yang estetis, atau dalam tindakan yang mencerminkan perilaku anak sehat, beriman, berakhlak mulia". Kompetensi itu dirancang untuk dicapai melalui proses pembelajaran berbasis penemuan *(discovery learning)* melalui kegiatan-kegiatan berbentuk tugas *(project based learning)*, dan penyelesaian masalah *(problem solving based learning)* yang mencakup proses mengamati, menanya, mengumpulkan informasi, mengasosiasi, dan mengomunikasikan.[2]

2.3 Organ Tubuh Manusia Dan Hewan

Organ adalah gabungan dari berbagai jenis jaringan yang terorganisasi dalam fungsi tertentu. Sistem organ adalah gabungan dari berbagai organ untuk melakukan fungsi tertentu di dalam tubuh. Setiap organ memegang peranan yang sama penting dalam menjalankan fungsinya.

1. Sistem Pencernaan Manusia

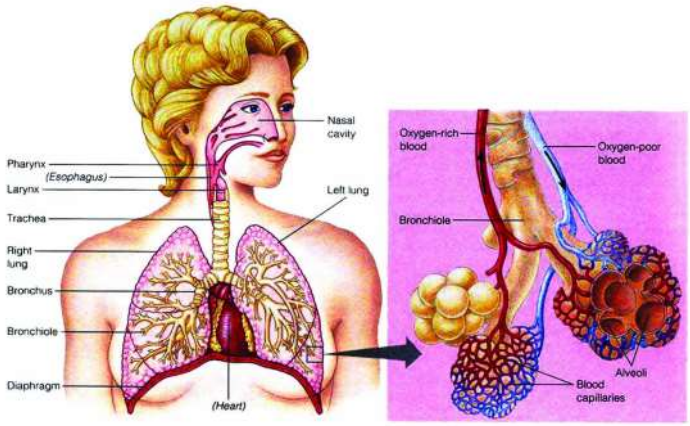
   Sistem pencernaan merupakan sistem yang memproses mengubah makanan dan menyerap sari makanan yang berupa nutrisi-nutrisi yang dibutuhkan oleh tubuh. Sistem pencernaan juga akan memecah molekul makanan yang kompleks menjadi molekul yang sederhana dengan bantuan enzim sehingga mudah dicerna oleh tubuh.[3]



Gambar 2.1 Organ Sistem Pencernaan Manusia[4]

2. Sistem Pernafasan Manusia

Sistem pernapasan pada manusia adalah sistem menghirup oksigen dari udara serta mengeluarkan karbon dioksida dan uap air. Dalam proses pernapasan, oksigen merupakan zat kebutuhan utama. Oksigen untuk pernapasan diperoleh dari udara di lingkungan sekitar.[3]

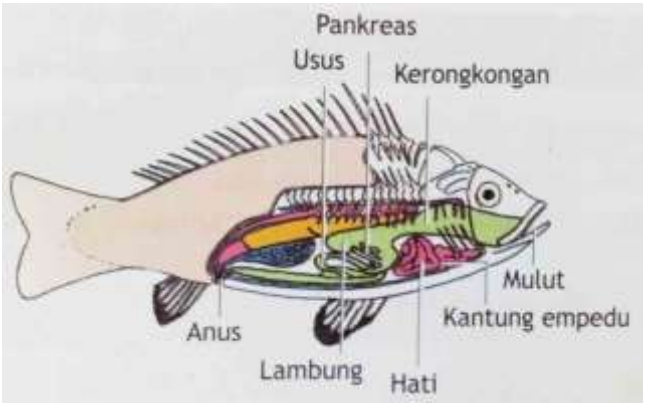


Gambar 2.2 Organ Sistem Pernafasan Manusia[5]

3. Sistem Pencernaan Hewan

Sistem ini berfungsi mengolah dan mengubah makanan, berupa molekul organik kompleks menjadi molekul yang lebih sederhana (sari pati makanan) agar dapat diserap tubuh. Organ hewan yang terkait dengan

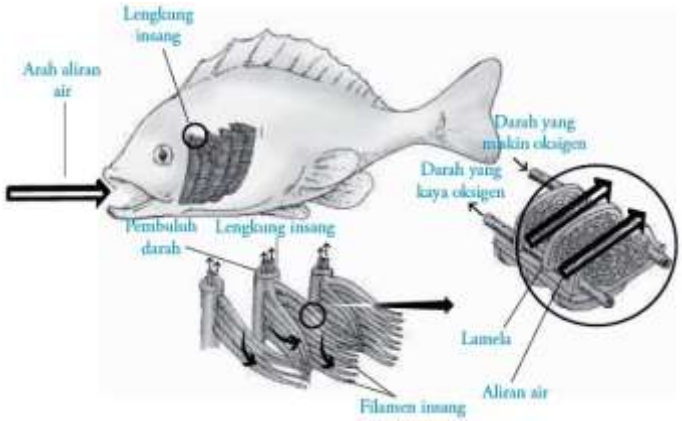
fungsi sistem ini, antara lain mulut (kelenjar ludah, gigi, dan lidah), esofagus, lambung, usus halus, dan usus besar.[3]



Gambar 2.3 Organ Sistem Pencernaan Hewan[4]

4. Sistem Pernafasan Hewan

Sistem ini berfungsi menyediakan oksigen dan mengeluarkan sisa metabolisme yang berbentuk CO2. Sistem pernapasan tersusun oleh beberapa organ hewan, di antaranya saluran-saluran pernapasan yang meliputi faring, laring, dan trakea serta paru-paru yang meliputi sistem bronkus dan alveolus.[3]



Gambar 2.4 Organ Sistem Pernafasan Hewan[4]

2.4 *Augmented Reality*

Augmented reality (AR) sebuah tampilan *real-time* langsung atau tidak langsung dari sebuah fisik dari sebuah objek nyata ditambah dengan menambahkan objek pada dunia maya sehinggan menghasilkan informasi tambahan pada objek yang ada. *Augmented reality* ini menggabungkan benda-benda nyata dan virtual objek yang ada, virtual objek ini hanya bersifat menambahkan bukan menggantikan objek nyata, sedangkan tujuan dari *augmented reality* ini adalah menyederhanakan objek nyata dengan membawa objek maya sehingga informasi tidak hanya untuk pengguna secara langsung *(user interface),* tetapi juga untuk setiap pengguna yang tidak langsung berhubungan dengan *user interface* dari objek nyata, seperti *live-streaming* video.[5] Perangkat utama untuk *augmented reality* adalah *display*, perangkat *input*, *tracking*, dan komputer.

Menurut Jacko dkk (2003) Realitas tertambah, atau kadang dikenal dengan singkatan bahasa Inggrisnya AR *(Augmented Reality)*, adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata.[6]

Menurut Ronald T. Azuma (1997) *Augmented Reality* adalah penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antarbenda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata.[7]



Gambar 2.5 *Augmented Reality*[8]

2.4.1 Sejarah *Augmented Reality*

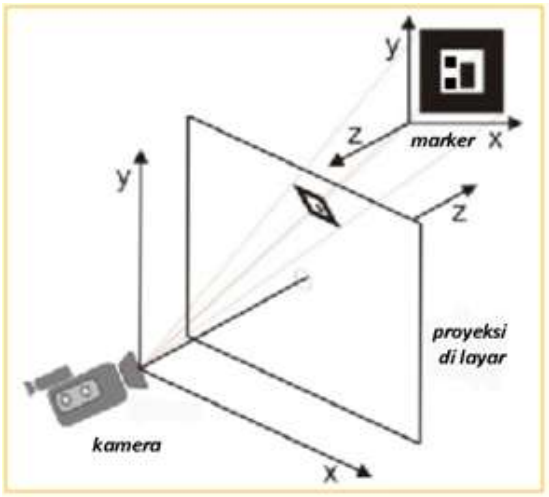
Sejarah tentang *Augmented Reality* dimulai dari tahun 1957-1962, ketika seorang penemu yang bernama Morton Heilig, seorang sinematografer, menciptakan dan memapatenkan sebuah simulator yang disebut Sensorama dengan visual, getaran dan bau. Pada tahun 1966, Ivan Sutherland menemukan *head-mounted display* yang dia claim 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, memperkenalkan *Virtual Reality* dan menciptakan bisnis komersial pertama kali di dunia maya, Tahun 1992 mengembangkan *Augmented Reality* untuk melakukan perbaikan pada pesawat boeing, dan pada tahun yang sama, LB Rosenberg mengembangkan salah satu fungsi sistem 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 ARToolkit di HITLab dan didemonstrasikan di SIGGRAPH, pada tahun 2000, Bruce.H.Thomas, mengembangkan ARQuake, sebuah *Mobile Game AR* yang ditunjukan di International Symposium on Wearable Komputers. Pada tahun 2008, Wikitude AR Travel Guide, memperkenalkan Android G1 Telephone yang berteknologi AR, tahun 2009, Saqoosha memperkenalkan FLARToolkit yang merupakan perkembangan dari ARToolkit. FLARToolkit memungkinkan kita memasang teknologi AR di sebuah 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 AR pada I-Phone 3GS.[9]

2.4.2 Lingkungan *Augmented Reality*

Pada sistem AR, sistem koordinat yang dipakai adalah model *pinhole* camera atau kamera lubang jarum. Dimana pada model ini sumbu z positif berada didepan dan yang menjadi acuan adalah posisi *marker* jika dilihat dari kamera.

Jika dilihat pada gambar 2.6, terlihat *marker* dan kamera masing-masing memiliki orientasi posisi yang berbeda. Baik *marker* kamera menggunakan sistem *right handed* (sumbu z positif didepan) dan hasil penangkapan gambar dari kamera diproyeksikan ke *viewplane* menggunakan proyeksi perspektif.



Gambar 2.6 Sistem Koordinat Lingkungan AR[9]

Dalam menampilkan objek 3D yang sesuai dengan posisi dan orientasi *marker*, perlu diperhitungkan hasil proyeksi yang diterima *viewplane* (bidang proyeksi dilayar) untuk kemudian ditampilkan. Selain proyeksi pada bidang 2D, dalam pergeseran *marker* maupun kamera perlu diperhatikan perubahan posisi dan rotasi dalam sistem koordinat 3D. Posisi dan orientasi dari *marker* didapat dari hasil *tracking marker* yang ditransformasikan dengan operasi translasi dan rotasi, sedangkan posisi dan orientasi yang ada pada proyeksi dilayar didapat dari perhitungan transformasi proyeksi perspektif.[9]

2.4.3 *Mixed Reality*

Paul Milgram and Fumio Kishino merumuskan kerangka kemungkinan penggabungan dan peleburan dunia nyata dan dunia maya yang disebut *Milgram's Reality-Virtuality Continuum* pada tahun 1994. Dalam Gambar 2.7, sisi yang paling kiri adalah lingkungan nyata yang hanya berisi benda nyata, dan sisi paling kanan adalah lingkungan maya yang berisi benda maya dalam *Augmented Reality*, yang lebih dekat ke sisi kiri, lingkungan bersifat nyata dan benda bersifat maya, sementara dalam *Augmented Virtuality*, yang lebih dekat ke sisi kanan, lingkungan bersifat maya dan benda bersifat nyata.[9]



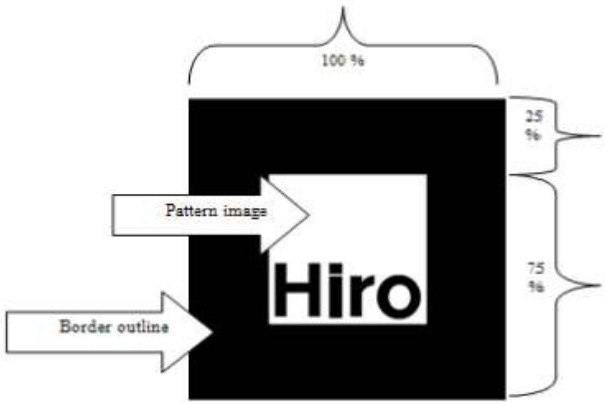
Gambar 2.7 *Mixed Reality*[9]

2.4.4 *Marker*

Marker adalah lingkungan nyata berbentuk objek nyata yang akan menghasilkan *Virtual Reality*, *marker* ini digunakan sebagai tempat *Augmented Reality* muncul. Berikut beberapa jenis *marker* yang digunakan pada teknologi *Augmented Reality*:

2.4.4.1 *Fiducial Marker*

Marker adalah pola yang dibuat dalam bentuk gambar yang telah dicetak dengan printer yang akan dikenali oleh kamera. *Marker* merupakan gambar yang terdiri atas *border outline* dan *pattern image* seperti terlihat pada gambar 2.8.



Gambar 2.8 *Fiducial Marker*[9]

Marker biasanya dengan warna hitam dan putih. Cara pembuatannya pun sederhana tetapi harus diperhatikan ketebalan *Marker* yang akan dibuat, ketebalan *Marker* jangan kurang dari 25 % dari panjang garis tepi agar pada saat proses deteksi *Marker* dapat lebih akurat. Nama *Hiro* yang ada pada gambar 2.8 merupakan sebuah pembeda saja. Sedangkan objek warna putih sebagai *background*, yang nantinya akan digunakan sebagai tempat objek yang akan di*render*.

Ciri-ciri yang umum digunakan untuk mengenali satu atau beberapa obyek di dalam citra adalah ukuran, posisi atau lokasi, dan orientasi atau sudut kemiringan obyek terhadap garis acuan yang digunakan. *Marker* terdapat dua intensitas warna yaitu warna hitam dan putih atau sering disebut sebagai citra biner. Citra biner memisahkan daerah *(region)* dan latar belakang dengan tegas, walaupun potensi munculnya kekeliruan selalu ada. Kekeliruan di sini adalah kesalahan mengelompokan *pixel* ke dalam golongannya, apakah *pixel* milik suatu daerah dikelompokan sebagai latar belakang atau sebaliknya. Kesalahan seperti ini sering disebut dengan *noise*. Warna putih pada *Marker* menunjukan warna sebuah objek, sedangkan warna hitam menunjukan latar belakang. Intensitas warna pada suatu objek memiliki warna yang lebih rendah (gelap), sedangkan latar belakang mempunyai intensitas yang lebih tinggi (terang). Namun pada kenyataannya dapat saja berlaku kebalikannya, yaitu objek mempunyai intensitas tinggi dan latar belakang mempunyai intensitas rendah. Kombinasi ini biasanya tergantung pada sifat latar belakang pada saat citra tidak tampil terang sekali

(putih) atau gelap sekali (hitam), melainkan di antaranya dengan demikian suatu objek yang sama dapat tampil lebih terang atau lebih gelap daripada latar belakangnya dalam citra, tergantung pada gelap atau terangnya warna yang melatar belakanginya.



Gambar 2.9 Contoh *Fiducial Marker*[9]

Ukuran *Marker* yang digunakan dapat mempengaruhi penangkapan pola *Marker* oleh kamera. Semakin besar ukuran *Marker* semakin jauh jarak yang bisa ditangkap oleh kamera dalam mendeteksi *Marker*. Namun disinilah masalahnya, ketika *Marker* bergerak menjauhi kamera, jumlah *pixel* pada layar kamera menjadi lebih sedikit dan ini bisa mengakibatkan pendeteksian tidak akurat.[9]

2.4.4.2 *Markerless Marker*

Markerless Marker mempunyai fungsi sama dengan *fiducial marker* namun bentuk *Markerless Marker* tidak harus kotak hitam dan putih, *Markerless* ini bisa berbentuk gambar yang mempunyai banyak warna. Pada *markerless* yang digunakan dan dikembangkan oleh *IN2AR*, 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). Contoh *Markerless Marker* pada gambar 2.10.



Gambar 2.10 *Markerless Marker*[8]

Markerless ini salah satu metode *Augmented Reality* yang saat ini sedang berkembang. Dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan objek. Dalam perancangan nya, seolah-olah *markerless* menggabungkan objek maya dengan objek nyata, dalam hal ini objek maya berupa objek 2D atau 3D dan objek nyatanya berupa gambar dengan pola tertentu *(markerless).* Secara garis besarnya dalam perancangan aplikasi ini ada 3 bagian penting yaitu :[9]

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

Adapun *markerless* yang sudah 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 algoritma 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.[10]



Gambar 2.11 *Face Tracking*[10]

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.[10]

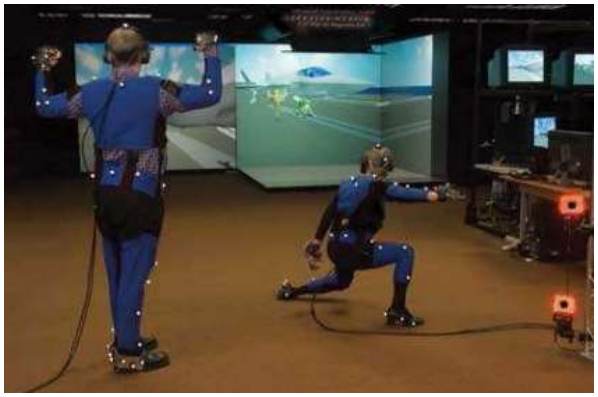


Gambar 2.12 *3D Object Tracking*[11]

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*.[10]



Gambar 2.13 *Motion Tracking*[10]

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. Salah satu pelopor *GPS Based Tracking* adalah aplikasi yang bernama *Wikitude*.[10]



Gambar 2.14 *GPS Based Tracking*[12]

2.4.5 *Multi Marker*

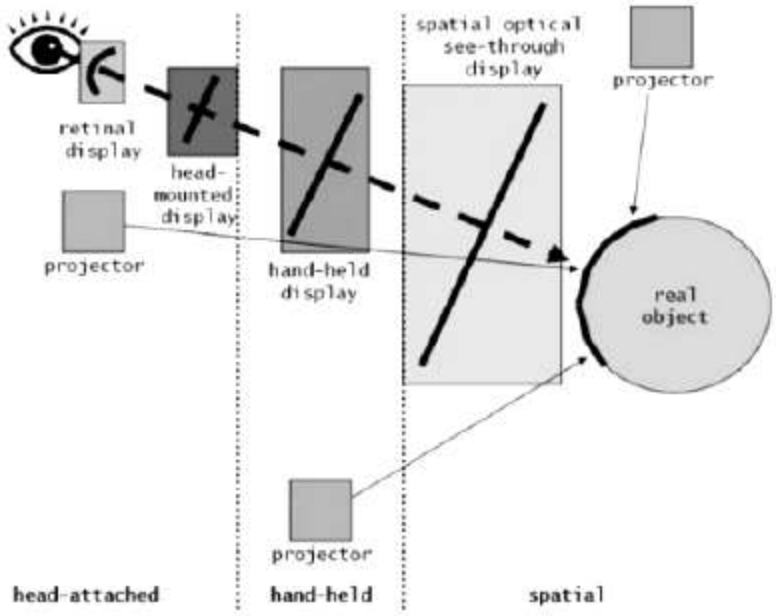
Multi Marker adlah merupakan sebuah metode perkembangan dari *single marker*, dimana proses pencocokan objek yag ditangkap lebih dari satu. Dalam implementasinya dapat dilakukan dengan menggunakan beberapa pendekatan metode yang dapat dilakukan seperti pelabelan komponen serta *corner detection* sebagai pengelan sudut dari beberapa bentuk *marker*.[10]



Gambar 2.15 *Multi Marker*[13]

2.4.6 Teknik *Display Augmented Reality*

Sistem *display AR* merupakan sistem manipulasi citra yang menggunakan seperangkat optik, elektronik, dan komponen mekanik untuk membentuk citra dalam jalur optik antara mata pengamat dan objek fisik yang akan digabungkan dengan teknik AR. Bergantung kepada optik yang digunakan, citra bisa dibentuk pada sebuah benda datar atau suatu bentuk permukaan yang kompleks (tidak datar). Gambar 2.16 mengilustrasikan kemungkinan citra akan dibentuk untuk mendukung AR, peletakan *display* bergantung dari pandangan pengguna dan objek, dan tipe citra seperti apa yang akan dihasilkan *(planar atau curved)*.[9]



Gambar 2.16 Pembentukan Citra *Display Augmented Reality*[9]

Secara garis besarnya ada tiga teknik display AR, yaitu sebagai berikut:

1. *Head-Attached Display*

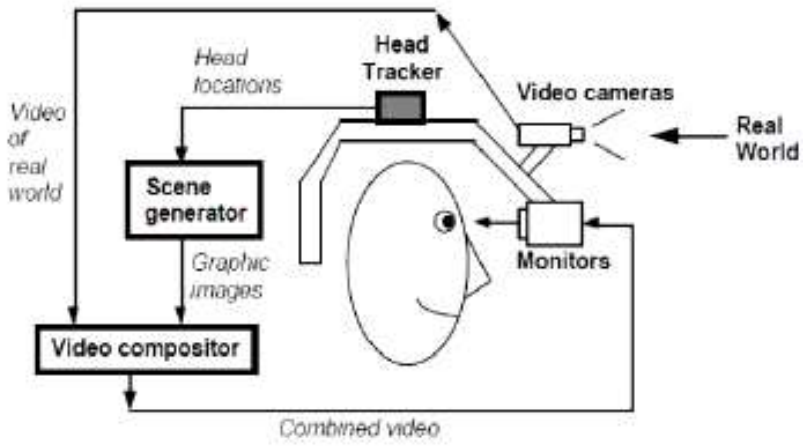
   Head-Attached Display merupakan teknik *display* yang mengharuskan penggunanya untuk memakai sistem ini di kepala pengguna. Berdasarkan teknik citra yang terbentuk, *Head-Attached Display* terbagi tiga, yaitu sebagai berikut:

a. *Head-Mounted Display*

Head-Mounted Display (HMD) menggabungkan citra dari objek virtual dan objek nyata dan menampilkannya langsung ke mata pengguna melalui suatu alat yang dipasang di kepala pengguna. Terdapat dua tipe utama perangkat HMD yang digunakan dalam aplikasi realitas tertambah, yaitu *video see-through HMD* dan *optical-see-through HMD*. Keduanya digunakan untuk berbagai jenis pekerjaan dan memiliki keuntungan dan kerugian masing masing. Dengan *optical-see-through HMD*, lingkungan nyata dilihat melalui cermin semi transparan yang diletakkan didepan mata pengguna. Cermin tersebut juga digunakan untuk merefleksikan citra yang dibentuk oleh komputer ke mata pengguna, menggabungkan lingkungan nyata dan virtual. Dengan *video see-through HMD*, lingkungan nyata direkam mengunakan dua kamera video yang terintegrasi ke alat, seperti gambar 2.18, dan citra yang dibentuk komputer digabung dengan video tadi untuk merepresentasikan lingkungan yang akan dilihat pengguna.

1) *Video-see-through Head-Mounted Display*

Video see-through HMD bekerja dengan menggabungkan sebuah *closedview HMD* dengan satu atau dua head-mounted kamera video, melalui kamera video tersebut pengguna melihat ke lingkungan nyata. Video dari kamera dikombinasikan dengan citra yang dibuat oleh *scene generator*, dunia nyata dan virtual digabungkan. Hasilnya dikirimkan ke monitor yang terletak di depan mata pengguna. Gambar 2.19 menunjukkan konsep dari *Video see-through HMD*, gambar 2.19 adalah contoh *Video see through HMD*, dengan dua video terintegrasi di bagian atas Helm.



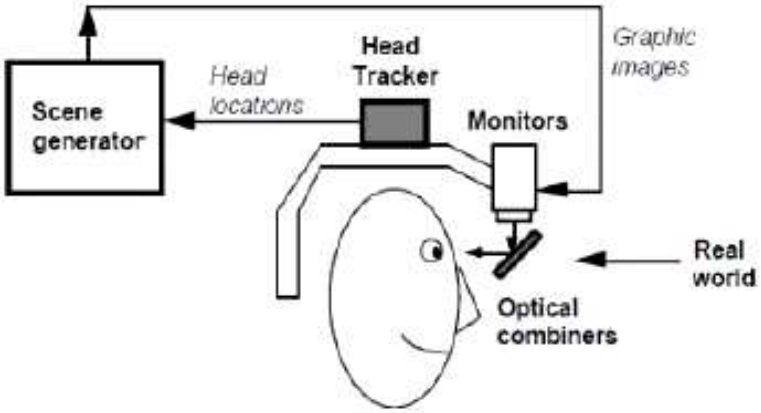
Gambar 2.17 Diagram *Opaque HMD*[9]

2) *Optical see-through Head-Mounted Display*

Tidak seperti penggunaan *video see-through HMD*, *optical seethrough HMD* menyerap cahaya dari lingkungan luar, sehingga memungkinkan pengguna untuk secara langsung mengamati dunia nyata dengan mata (gambar 2.19). Selain itu, sebuah sistem cermin yang diletakkan di depan mata pengguna memantulkan cahaya dari pencitraan grafis yang dihasilkan komputer. Pencitraan yang dihasilkan merupakan gabungan optis dari pandangan atas dunia nyata dengan pencitraan grafis.



Gambar 2.18 Contoh *Opaque HMD*[9]



Gambar 2.19 Diagram *see-trough HMD*[9]

b. *Head-Mounted Projectors*

Head-Mounted Projectors Menggunakan proyektor atau panel LCD kecil dan mempunyai cahaya sendiri untuk menampilkan citra langsung ke lingkungan nyata. Seperti yang ditunjukkan oleh gambar 2.20 dan gambar 2.21.



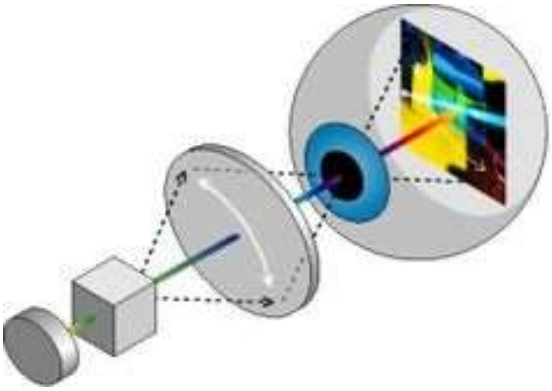
Gambar 2.20 Contoh *see-through HMD*[9]



Gambar 2.21 Ilustrasi penggunaan dua jenis perangkat HMD[9]

c. *Virtual Retina Display*

Virtual retina display (VRD), atau disebut juga dengan *retinal scanning display (RSD)*, memproyeksikan cahaya langsung kepada retina mata pengguna. VRD dapat menampilkan proyeksi citra yang penuh dan juga tembus pandang tergantung pada intensitas cahaya yang dikeluarkan, sehingga pengguna dapat menggabungkan realitas nyata dengan citra yang diproyeksikan melalui sistem penglihatannya. VRD dapat menampilkan jarak pandang yang lebih luas daripada HMD dengan citra beresolusi tinggi. Keuntungan lain VRD adalah konstruksinya yang kecil dan ringan. Namun, VRD yang ada kini masih merupakan prototipe yang masih terdapat dalam tahap perkembangan, sehingga masih belum dapat menggantikan HMD yang masih dominan digunakan dalam bidang AR. Gambaran sederhana VRD ini dapat dilihat pada gambar 2.22.



Gambar 2.22 Diagram sederhana *virtual retina display*[15]

Kelebihan teknik *display Head-Attached Display* ini adalah lebih nyaman ke pengguna, karena citra yang terbentuk mengikuti sudut pandang pengguna.

2. *Handheld Display*

Teknik ini menggunakan alat dengan *display* yang dengan mudah dapat di genggam pengguna (Tablet PC, PDA dan telepon genggam)

seperti yang ditunjukkan pada gambar 2.23. Sensor dapat berupa GPS, kompas digital ataupun kamera yang ada pada *Handheld* tersebut. Semua penerapan AR pada perangkat genggam menggunakan kamera untuk menggabungkan citra digital dengan lingkungan nyata, *Handheld AR* sangat menjanjikan untuk tujuan komersial. Dua kelebihan utama dari *Handheld* AR adalah mobilitas perangkat yang mudah dan salah satu perangkat genggam yang banyak digunakan (telepon genggam) telah banyak dilengkapi kamera.



Gambar 2.23 Contoh *Augmented Reality* Dengan *Smartphone*[16]

3. *Spatial Display*

Dalam *Spatial Augmented Reality (SAsR),* objek nyata digabungkan langsung dengan citra yang terintegrasi langsung ke lingkungan nyata. Contohnya, citra diproyeksikan ke lingkungan nyata menggunakan proyektor digital atau tergabung dengan lingkungan menggunakan *panel display*. Perbedaan utama pada SAR dibanding teknik *display* sebelumnya adalah *display*nya terpisah dengan pengguna. SAR memiliki kelebihan dari HMD dan *Handheld*, sistem ini bisa digunakan oleh banyak orang pada waktu bersamaan tanpa perlu mengenakan suatu alat. Ada tiga teknik *display* dalam SAR, yaitu sebagai berikut:

a. *Screen-Based Video See-Through Displays*

Screen-based AR menggabungkan citra dan lingkungan nyata yang ditampilkan ke sebuah monitor, seperti yang ditunjukkan pada gambar 2.24.



Gambar 2.24 Contoh *Screen-Based Video See-Through Displays*[10]

b. *Spatial Optical See-Through Displays*

Sistem ini menghasilkan citra yang ditampilkan langsung ke lingkungan nyata. Komponen yang penting dalam sistem ini meliputi *Spatial optical combiners* (*planar* atau *curved beam combiners*), layar transparan atau *hologram.*

c. *Projection-Based Spatial Displays*

Sistem ini memproyeksikan citra secara langsung pada permukaan objek fisik daripada menampilkannya pada sebuah bidang pencitraan dalam penglihatan pengguna. Sistem ini menggunakan banyak *proyektor* yang digunakan untuk meningkatkan wilayah tampilan serta meningkatkan kualitas citra.[10]

2.4.7 Penerapan *Augmented Reality*

Seiring berjalannya waktu, teknologi *Augmented Reality* berkembang sangat pesat sehingga memungkinkan pengembangan aplikasi ini di berbagai bidang contoh nya sebagai berikut :

1. Pendidikan

   Sehubungan dengan penerapan *Augmented Reality* di Indonesia, teknologi ini sebenarnya sudah bisa kita aplikasikan dalam dunia pendidikan. Beberapa siswa cukup banyak yang masih merasa kesulitan dalam memahami peristiwa sejarah, terutama sejarah bangsanya sendiri, Bangsa Indonesia. Dalam hal ini, teknologi A*ugmented Reality* dipandang mempunyai kesempatan besar untuk membantu memvisualisasikan sejarah yang masih "*abstrak*" menjadi lebih nyata dalam pandangan para siswa. Harapannya, visualisasi seperti ini dapat memberikan *feel* perjuangan yang telah diusahakan oleh para pendahulu dan pejuang-pejuang sejarah kemerdekaan Indonesia.

2. Medis

   Bidang ini merupakan salah satu bidang yang paling penting bagi sistem realitas tertambah. Contoh penggunaannya adalah pada pemeriksaan sebelum operasi, seperti *CT Scan* atau *MRI*, yang memberikan gambaran kepada ahli bedah mengenai anatomi internal pasien. Dari gambar-gambar ini kemudian pembedahan direncanakan. *Augmented Reality* dapat diaplikasikan sehingga tim bedah dapat melihat data *CT Scan* atau *MRI* pada pasien saat pembedahan berlangsung. Penggunaan lain adalah untuk pencitraan ultrasonik, di mana teknisi ultrasonik dapat mengamati pencitraan *fetus* yang terletak di *abdomen* wanita yang hamil.

3. Hiburan

   Bentuk sederhana dari *Augmented Reality* telah dipergunakan dalam bidang hiburan dan berita untuk waktu yang cukup lama. Contohnya adalah pada acara laporan cuasa dalam siaran televisi dimana wartawan ditampilkan berdiri di depan peta cuaca yang berubah. Dalam studio, wartawan tersebut sebenarnya berdiri di depan layar biru atau hijau. Pencitraan yang asli digabungkan dengan peta buatan komputer menggunakan teknik yang bernama *chroma-keying*.

4. Robot

   Dalam robotika, seorang operator robot, menggunakan pengendali pencitraan visual dalam mengendalikan robot itu. Jadi, penerapan *Augmented Reality* dibutuhkan didunia robot.

5. Desain

   Seorang desainer membutuhkan *Augmented Reality* untuk menampilkan hasil desain mereka secara lebih nyata terhadap klien. Dengan AR klien dapat mengetahui tentang spesifikasi yang lebih detail tentang desain.

6. Militer

   Kalangan militer telah bertahun-tahun menggunakan tampilan dalam kokpit yang menampilkan informasi kepada pilot pada kaca pelindung kokpit atau kaca depan helm penerbangan mereka. Ini merupakan sebuah bentuk tampilan *Augmented Reality*. SIMNET, sebuah sistem permainan simulasi perang, juga menggunakan teknologi *Augmented Reality*. Dengan melengkapi anggota militer dengan tampilan kaca depan helm, aktivitas unit lain yang berpartisipasi dapat ditampilkan.[17]

2.5 *Motion Detection*

Motion detection adalah proses untuk mengidentifikasi adanya suatu gerakan dengan melakukan proses pengurangan nilai- nilai intensitas setiap *pixel* yang ada pada *background* terhadap nilai-nilai intensitas pada suatu *image* yang diambil secara *continyu*.

Penanganan masukan dalam perangkat lunak menerapkan metode *motion detection* atau pelacakan gerakan. Pelacakan gerakan ini berjalan dengan melakukan deteksi pada ruang lingkup yang dapat dijangkau oleh perangkat kamera. Sistem dapat mengetahui gerakan yang terjadi dengan mula-mula membagi citra yang tertangkap oleh kamera menjadi blok-blok yang lebih kecil. Blok-blok yang lebih kecil ini masing-masing memiliki kondisi awal *(initial state)* berupa blok yang memiliki properti warna dan citra tertentu. Kondisi awal inilah

yang digunakan sebagai acuan oleh sistem bahwa saat itu semua benda sedang diam dan tidak terdapat gerakan. Pembagian blok dan penentuan properti awal citra tersebut dapat dilakukan dengan mengubah citra yang tertangkap oleh kamera menjadi citra biner. Proses pengubahan citra ini lazim disebut dengan *Image Binarization*.

Citra yang telah diubah menjadi citra biner menjadi memungkinkan untuk melacak gerakan dengan lebih mudah. Gerakan yang terjadi dapat terdeteksi ketika sistem, melalui penangkapan citra oleh kamera, mendapati bahwa properti citra awal suatu blok tidak sama dengan properti citra saat ini. Dengan adanya hal ini, sistem dapat mengetahui bahwa sedang terjadi gerakan pada area yang tertangkap oleh kamera. Dalam hal ini, area yang diam direpresentasikan dengan citra berwarna hitam, dan area yang terjadi gerakan di dalamnya direpresentasikan dengan citra yang berwarna putih. Dengan pembedaan warna citra ini, sistem dapat mendeteksi secara tepat area mana saja yang sedang mengalami perubahan citra akibat terjadinya gerakan.[18]

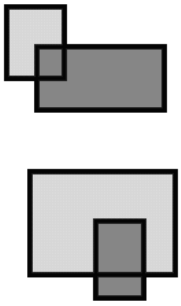


Gambar 2.25 *Motion Detection*[18]

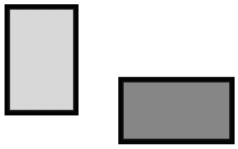
2.5.1 *Collusion Detection*

Collision Detection adalah proses pendeteksian apakah terjadi tumbukan antara dua buah objek atau lebih. Secara umum, proses ini dilakukan dengan melakukan deteksi koordinat antara objek-objek dalam satu bidang. Ketika suatu objek dengan luasan tertentu mengenai area objek lain baik sebagian ataupun seluruhnya maka sistem akan memberikan peringatan bahwa bagian objek saling menindih (*overlapping*). Proses pendeteksian tumbukan ini menjadi penting

karena digunakan pada aplikasi untuk mendeteksi apakah posisi tangan yang digerakkan telah menyentuh bidang dari objek atau suatu area tertentu pada layar. Dengan menggunakan algoritma pendeteksian tumbukan, perangkat lunak yang dibangun mejadi mungkin melakukan tindakan tertentu ketika dua buah benda mengalami saling sentuh. Dengan begitu, gerakan tangan yang berhasil dilacak dengan menggunakan metode *motion detection* sebelumnya dapat digunakan sebagai masukan dengan menerapkan pendeteksian tumbukan antara objek dan posisi area gerakan. Kondisi dua buah objek dengan status tumbukan tertentu dalam suatu waktu dapat dilihat pada Gambar 2.26.[18]



Gambar 2.26 *Overlapping*



Gambar 2.27 Tidak *Overlapping*

2.6 *IN2AR*

IN2AR adalah plugin *ActionScript3* dan Adobe Ekstensi Perpustakaan asli atau Unity3D yang memungkinkan untuk mendeteksi gambar dan memperkirakan *tracking* dengan menggunakan *Webcam* standar/kamera atau kamera ponsel. Informasi dapat digunakan untuk menempatkan benda-benda dan video 3D dalam bentuk gambar dan dapat membuat aplikasi *Augmented Reality* atau untuk membuat *game* unik yang dapat dikendalikan oleh gerakan gambar atau telepon.[19]

Pelacakan rekursif menjadi teknik model yang digunakan oleh pihak *IN2AR* untuk melakukan teknik *markerless tracking system.* Sifat pelacakan yang digunakan dengan teknik berbasis model *(model based techniques)* dapat diklasifikasikan dalam 2 bagian:

1. Pelacakan rekursif *(Recursive tracking*), dimana pada saat sebelum objek dimunculkan terlebih dahulu dilakukan perkiraan untuk menghitung pelacakan yang terjadi.
2. Pelacakan dengan pendeteksian *(Tracking by detection)*, pada klasifikasi ini dilakukan perhitungan pelacakan tanpa perkiraan sebelumnya, yang memungkinkan otomatis inisialisasi dan pemulihan dari kegagalan.

Kelebihan *IN2AR* jika dibandingkan dengan *toolkit* atau *library* lain adalah:

1. Selain menyediakan lisensi berbayar (2212,50 EURO per proyek), juga terdapat lisensi gratis. Perbedaan kedua lisensi tersebut adalah bahwa aplikasi AR yang dibuat dengan lisensi gratis harus selalu menyertakan logo *IN2AR* di dalam aplikasi.
2. Dukungan pengembang di forum *IN2AR* sangat kuat. Pengembang aktif menjawab pertanyaan seputar teknis *IN2AR*.
3. *IN2AR* dapat mengenali hampir semua jenis gambar sebagai *marker*. Berbeda dengan *ARToolkit* dan *FLARToolkit* yang hanya mengenali gambar dengan bingkai hitam sebagai *marker*.
4. Distribusi aplikasi yang dibuat dengan *IN2AR* cukup mudah.

Aplikasi didistribusikan menggunakan file .swf yang dapat ditampilkan di halaman *web*. *User* yang memiliki *web browser* dengan *flash player* dan komputer *webcam* serta terkoneksi internet akan dapat mengakses aplikasi AR tersebut.

Selain memiliki beberapa kelebihan, *IN2AR* juga memiliki kelemahan yaitu:

1. Pengguna harus mengenal bahasa pemrograman, terutama bahasa pemrograman *Action Script 3*.0. Hal ini karena segala macam perubahan dalam penggunaan *IN2AR* adalah dengan cara mengetikkan kode dalam bahasa *pemrograman Action Script 3.0*. *IN2AR* tidak memiliki dokumentasi yang lengkap dan sistematis. Pengguna *IN2AR* harus melakukan eksplorasi sendiri untuk dapat menggunakan *IN2AR* sesuai kebutuhan. Aktivitas di forum *IN2AR* akan sangat membantu jika pengguna mengalami kesulitan.

2. Pembuatan *marker* untuk lisensi gratis harus melalui tim, Gambar yang ke tim *IN2AR* gambar yang akan dijadikan *marker* harus dikirim terlebih dahulu ke tim *IN2AR* untuk dibatkan file .ass.



Gambar 2.28 *IN2AR*[19]

2.7 *Papervision 3D*

Papervision 3D merupakan salah satu library *open source* untuk menampilkan gambar 3 dimensi di dalam *flash*. Ada enam unsur yang terlibat ketika bekerja dengan *Papervision3D* yaitu *scene, viewport, camera, 3D object, material,* dan *render engine*.

1. *Scene*

Scene merupakan ruang untuk meletakkan suatu objek, sehingga objek tersebut dapat terlihat. Scene ini serupa dengan stage pada *Flash* atau canvas pada Photoshop. 3D *engine renderer*, *scene* harus bersifat tiga dimensi, dengan (x, y, z) sebagai aksisnya.

2. *Camera*

Camera adalah suatu alat yang berfungsi untuk merekam aktifitas objek-objek di dalam *scene*. Di perlukan keberadaan *camera* ini di ruang 3D, tapi tidak membutuhkan *camera* tersebut masuk ke dalam *scene*. Kelebihan yang dimiliki *camera* adalah kemampuannya untuk tidak *me-render* objek di dalam *scene* yang terletak terlalu jauh atau terlalu dekat dengan *camera*. Kemampuan ini dibuat agar performa *flash movie* tetap terjaga.

3. *Viewport*

Container sprite pada *stage*, itulah *viewport*. Tidak ada yang lebih tepat untuk menganalogikan *viewport* dengan lensa pada sebuah kamera. *Viewport* dapat dikatakan sebagai indera penglihatan *camera*. Kita dapat mengatur besar kecil

ruang pandang penglihatan kita dengan menyipitkan atau membuka lebar-lebar mata begitu pula dengan *viewport*.

4. *3D Objects*

3D objects adalah bentuk atau bangun ruang *3D* yang berada di ruang *3D*, tentu saja. *3D objects* dalam *Papervision3D* dikenal juga sebagai *DisplayObject3D*, karena *3D objects* di sini bisa berupa *sprites* atau *movie clips*.

5. *Material*

Tekstur atau kulit pembungkus *3D objects* dalam *Papervision3D* disebut sebagai *material*. *Material* yang paling sederhana yang dapat gunakan adalah warna atau *still images*. Tapi, live streaming video pun sebenarnya dapat gunakan sebagai *material*.

6. *Render Engine*

Pada intinya, *render engine* itu seperti *rolling camera*. Selama masih membiarkan tombol ada pada posisi *on*, *render engine* akan menyampaikan informasi-informasi yang terrekam di dalam *scene*, oleh camera, untuk kemudian disampaikan atau dimunculkan pada *viewport*. Ketika tombol berada di posisi *off*, *camera* tidak akan merekam dan *viewport* tidak akan menampilkan informasi-informasi baru yang ada dalam *scene*.

2.8 *FlashDevelop*

Flashdevelop adalah sebuah aplikasi *open source*. *Flashdevelop* dibuat pada tahun 2005 oleh *Flash developers* yang memiliki motivasi yang baik. *FlashDevelop* sebagai *software* utama untuk melakukan *coding* dan *Adobe Flash* sebagai visual *IDE* untuk membuat dan menaruh aset-aset yang akan digunakan nanti, sekaligus sebagai compilernya.

Selain itu *Flashdevelop* juga mendukung *Actionscript* 2.0 dan 3.0 dan perkembangan *HaXe*. Aplikasi ini sangat membantu untuk pada *Flash programmer* untuk generalisasi kode atau sintaks yang dapat digunakan dalam suatu proyek.[20]



Gambar 2.29 *FlashDevelop*[20]

2.9 *Adobe Flash*

Adobe Flash (dulunya *Macromedia Flash*) adalah platform multimedia yang aslinya dibuat oleh *Macromedia* dan saat ini dikembangkan dan didistribusikan oleh *Adobe Systems*. Sejak pengenalannya pada Tahun 1996, *Flash* telah menjadi metode yang popular untuk menambahkan animasi dan interaktivitas ke halaman web. Komponen *Flash* untuk mengintegrasikan video ke halaman web, dan yang terbaru saat ini, untuk mengembangkan *RIAs*.

Flash dapat memanipulasi vector dan raster grafik, serta mendukung streaming dua arah audio dan video. *Flash* menggunakan bahasa *script* yang disebut *Action Script*. Banyak produk *software*, sistem dan *device* dapat menampilkan konten *Flash*, contohnya Adobe Flash player, yang tersedia gratis bagi sebagian besar web browser. Beberapa ponsel dan alat elektronik lainnya juga dapat menampilkan konten Flash, menggunakan Flash lite. File dalam format SWF, biasanya disebut *"ShockWave Flash movies"*, *"Flash movies"* atau *"Flash games"*, yang biasanya memiliki sebuah ekstensi .swf dan dapat menjadi objek di halaman web. File tersebut pada dasarnya dijalankan dengan Flash Player itu sendiri atau digabungkan dengan "*Projector*" (video flash yang dapat berjalan sendiri dengan ekstensi .exe di Microsoft Windows atau .hqx untuk Macintosh). File Flash Video memiliki ekstensi .flv dan juga digunakan dalam .swf atau dijalankan melalui aplikasi yang dapat menjalankan file .flv.[21]



Gambar 2.30 *Adobe Flash*[21]

2.9.1 *ActionScript*

ActionScript merupakan bahasa pemrograman berorientasi objek yang berdasarkan *ECMAScript* (bahasa yang distandarisasi oleh *Ecma International* dalam spesifikasi *ECMA-262* dan *ISO/IEC 16262*). *ActionScript* terutama digunakan untuk pengembangan *website* dan *software* menggunakan *Adobe Flash Player* (dalam bentuk file SWF yang diintegrasikan ke halaman web), *ActionScript* juga digunakan pada beberapa aplikasi untuk database (seperti *Alpha Five*). *ActionScript* pada awalnya didesain untuk mengatur animasi vektor 2D sederhana yang dibuat di *Adobe Flash*, dengan berkembangnya. Versi terakhir dari *ActionScript* menambahkan kemungkinan penggunaan untuk pembuatan web berbasis game dan *RIAs* dengan media streaming (seperti video dan audio).

ActionScript 3.0 mempunyai beberapa tipe data primitif. selain itu *actionscript* juga mempunyai beberapa tipe data kompleks yang dibangun dari tipe data prmitif. bahkan kita bisa membangun tipe data sendiri, dalam bahasa pemograman obyek sering dikenal dengan nama *class* obyek.

Salah satu kelebihan *Actionscript 3.0* adalah dukungannya terhadap pemograman menggunakan *class-class*. Dengan penggunan *class*, anda dapat menerapakan konsep pemograman berorientasi obyek dalam pengkodean program. Sebuah *class* adalah cetak biru dari sebuah obyek. *Class* mendefinisikan obyek secara penuh, meliputi data obyek operasi-operasinya. Untuk membuat *class* perlu diperhatikan beberapa hal berikut ini:

a. Pembuatan *class* dalam *actionscript* harus diawali dengan sebuah *package*. Sebuah *packet* adalah grup dari *class* (bisa hanya satu *class* atau beberapa *class*). Bisa juga menambahkan nama folder di belakang kata *package* yang menunjukan direktori tempat anda menyimpan *class* anda. Misalnya anda bisa membuat *class* Bola yang akan anda taruh dalam folder *lib* dengan menambahkan kata *lib* di belakang kata *package*.
b. Dokumen dari *class* harus disimpan dengan nama yang sama dengan *class*, misalnya nama *Class.as.* Misalnya *class* Bola yang anda buat harus anda simpan dengan nama bola.as.
c. Sebaliknya *class* yang dibuat merujuk ke *class* yang sudah dimiliki Flash, misalnya *Sprite* atau *MovieClip* gunakan tambahan kata *extends Sprite* pada lanjutan nama *class* nya, misalnya *public class* bola *extends Sprite*.
d. Menambahkan *variable* dengan ruang lingkup *public*, internal ataupun yang lainnya sesuai kebutuhan. Misalnya menambahkan variabel-variabel yang bersifat *public*, seperti : radius yang menunjukan jari-jari bola, warna yang menunjukan warna bola, vy yang menunjukan kecepatan bola searah sumbu-x, dan vy yang menunjukan bola searah sumbu-y.
e. Menambahkan *method* berupa fungsi dengan ruang lingkup yang di inginkan didalam fungsi. Salah satu fungsi yang sebaiknya ada adalah fungsi dengan nama fungsi sama dengan nama *class*.[22]

2.9.2 Sejarah *Action Script*

ActionScript 1.0 dirilis pertama kali pada *Flash* 5 yang merupakan pengembangan dari *Action* di *Flash* 4, dan masih digunakan hingga *Flash Mx* atau *Flash* 6. Kemudian muncul *ActionScript* 2.0 yang dirilis pada *Flash* MX 2004 atau *Flash* 7, *Flash* 8, sampai *Flash* CS3 atau *Flash* 9. Sebagai generasi penerus *ActionScript* 1.0 dengan kelebihannya yang mempunyai kemampuan *compile time checking, strict-typing* pada *variable, class-based syntax* (yang sebelumnya adalah *prototype base*). *ActionScript* 3.0 digunakan pada *Flash* CS3 atau *Flash* 9, Flex 2, *Flex* 3. *ActionScript* versi 3.0 ini berupa sebuah restrukturisasi fundamental dari model pemrograman sebelumnya. Penggunaannya yang luas terutama dalam

pengembangan *Rich Internet Application* atau sering disebut *RIA*, dengan hadirnya *Flex* yang menawarkan hal serupa seperti *AJAX, JavaFX*, dan *Microsoft Silverlight.*[10]

2.10 *3D Stuido Max*

3D Studio Max biasa juga disebut *3ds Max* atau hanya *MAX* adalah sebuah perangkat lunak grafik *vektor* 3D dan animasi, ditulis oleh *Autodesk Media* & *Entertainment* (dulunya dikenal sebagai *Discreet* and *Kinetix*). Perangkat lunak ini dikembangkan dari pendahulunya *3D Studio fo DOS*, tetapi untuk *platform Win32*. *Kinetix* kemudian bergabung dengan akuisisi terakhir *Autodesk, Discreet Logic*. Versi terbaru *3Ds Max* pada Juli 2005 adalah 7. *3Ds Max Autodesk 8* diperkirakan akan tersedia pada akhir tahun. Hal ini telah diumumkan oleh *Discreet* di *Siggraph* 2005.

3ds Max adalah salah satu paket perangkat lunak yang paling luas digunakan sekarang ini, karena beberapa alasan seperti penggunaan platform *Microsoft Windows*, kemampuan mengedit yang serba bisa, dan arsitektur *plugin* yang banyak.

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

1) Koordinat Cartesian

   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)

2) 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)

3) Koordinat spherical

   Cara ini menggabungkan antara jarak dan dua sudut, dan masing-masing besaran dipisahkan dengan tanda<, yaitu:

   (jarak)<(sudut)<(sudut), penulisannya: (1.03<60.95<0); (1<43<55).[22]



Gambar 2.31 *3D Studio Max*[23]

2.11 Pemrograman Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespodensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas, sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dimiliki oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

Sistem berorientasi objek merupakan sebuah sistem yang komponennya dienkapsulasi menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, dan komponen lainnya yang dapat berinteraksi satu sama lain.

Terdapat beberapa konsep utama pada metodologi berorientasi objek, diantaranya:[24]

1. Kelas (*class*), kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dari himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*), dan arti. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dari kelas semula dapat diwariskan ke kelas yang baru.
2. Objek (*object*), abstraksi sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, datau hal-hal

lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

3. Abstraksi (*abstraction*), prinsip untuk merepresntasikan dunia nyata yang kompleks menjadi suatu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.
4. Enkapsulasi (*encapsulation*), pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek untuk menyembunyikan impelemntasi dari objek sehingga objek lain tidak mengetahui cara kerjanya.
5. Pewarisan (*inheritance*), mekanisme yang memugkinkan suatu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.
6. Polimorfisme (*polymorphism*), kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

2.12 *Unified Modeling Language (UML)*

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*)[5]. UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software. *Unified Modelling Language* (UML) juga merupakan 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.

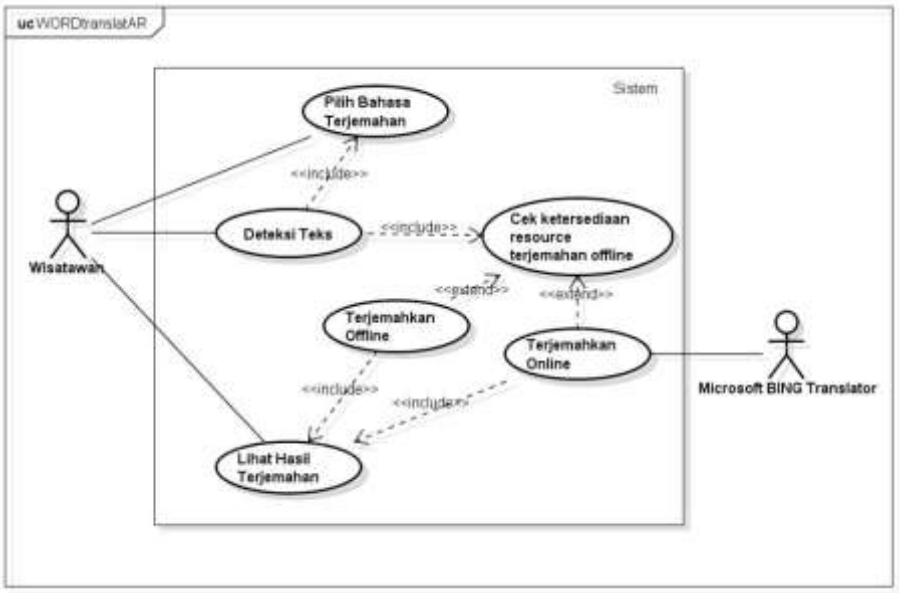
Terdapat beberapa diagram yang umum digunakan dalam pemodelan UML diantaranya [24]:

1. *Use Case Diagram,* merupakan pemodelan untuk kelakuan (*behavior*) sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.
2. *Class Diagram,* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode (operasi). Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode (operasi) fungsi-fungsi yang dimiliki oleh suatu kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki kelas *main*, kelas *view*, kelas *controller*, serta kelas *model*.
3. *Sequence Diagram,* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui onjek-objek yang terlibat dalam sebuah *use cse* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.
4. *Activity Diagram,* menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, melainkan aktivitas yang dapat dilakukan oleh sistem.
5. *Component Diagram,* dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komonen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah komponen *user interface* yang menagani tampilan, komponen *bussiness processing* yang menangani fungsi-fungsi proses bisnis, komponen *data* yang menangani manipulasi data, dan komponen *security*

yang menangani keamanan sistem. Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan.

2.13 *Use Case Diagram*

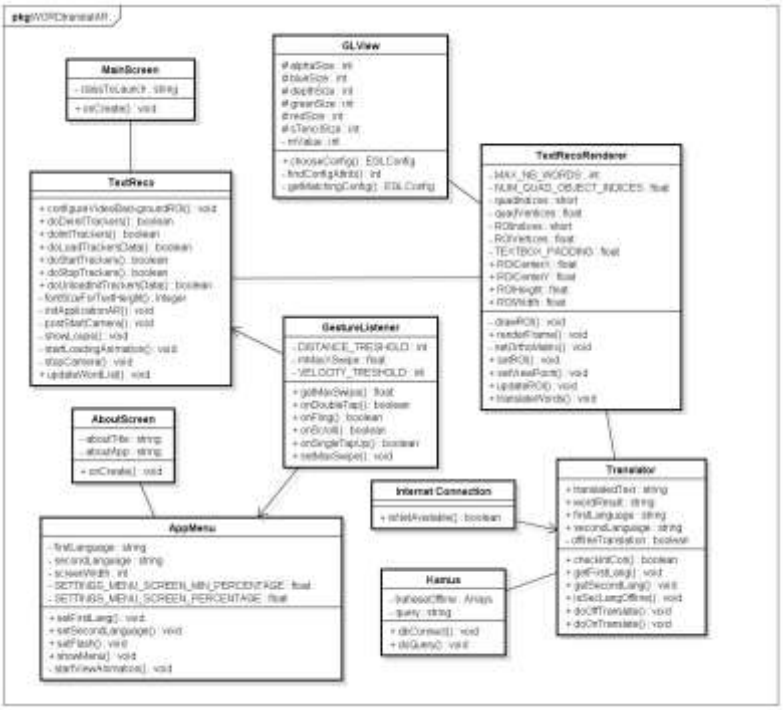
Use Case Diagram menjelakan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (aktor). Diagram ini menunjukan fungsionalitas suatu sistem yang berinteraksi dengan dunia luar. *Use Case* Diagram dapat digunakan selama proses analisis untuk menangkap *requirement* sistem dan untuk memahami bagaimana sistem bekerja seperti pada gambar 2.32.



Gambar 2.32 *Use Case Diagram*

2.14 *Class Diagram*

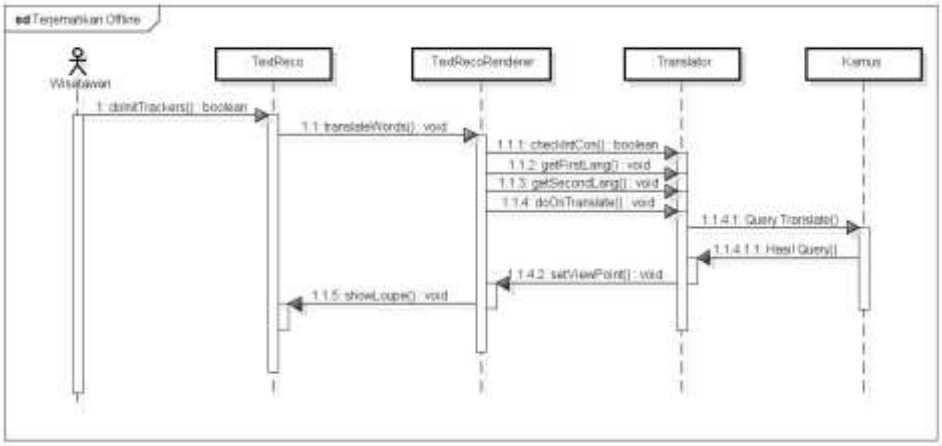
Class Diagram menjelaskan dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class Diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem. Selama proses analisis, class diagram memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan prilaku sistem. Selama tahap desain, class diagram berperan dalam menagkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat seperti pada gambar 2.33.



Gambar 2.33 *Class Diagram*

2.15 *Sequnce Case Diagram*

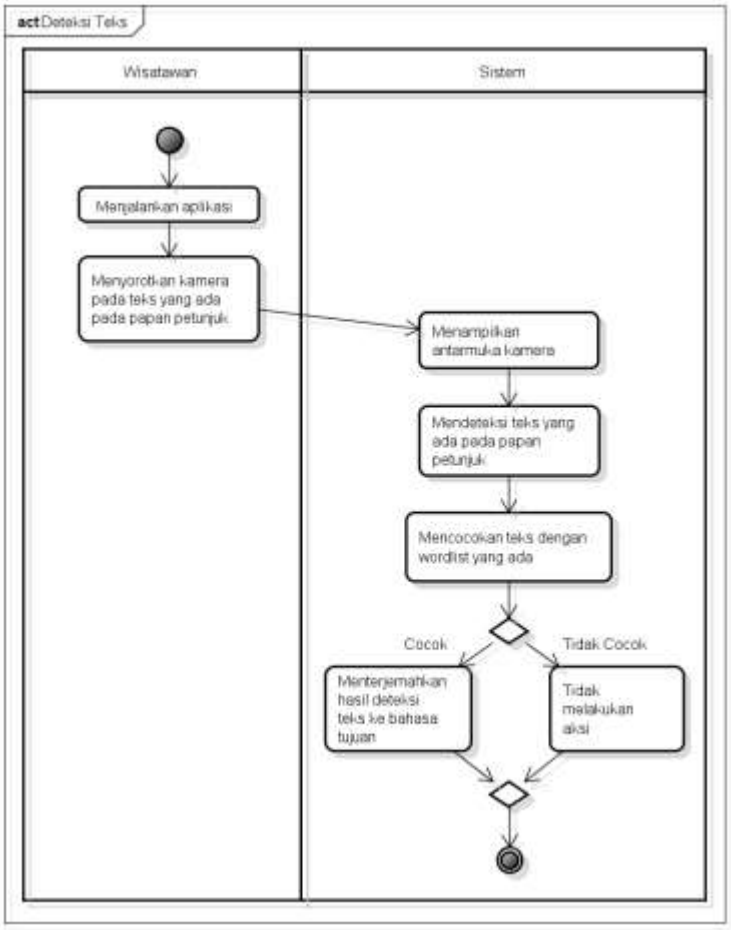
Menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem seperti pada gambar 2.34.



Gambar 2.34 *Sequence Diagram*

2.16 *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainya seperti *use case* atau interaksi seperti pada gambar 2.35.



Gambar 2.35 *Activity Diagram*