BAB II
TINJAUAN PUSTAKA

2.1 Profil Singkat Sekolah Dasar Darul Hikam

Sekolah Dasar Darul Hikam Bandung merupakan salah satu sekolah ternama di kota bandung. Sekolah ini banyak berprestasi dalam bidang akademik atau pun non akademik. Sekolah Dasar yang ada saat ini di Darul Hikam ada 2 jenis, yaitu Sekolah Dasar Darul Hikam (SD Darul Hikam) dan *Primay Darul Hikam International School (Primary DHIS)*.

2.1.1 Sejarah Sekolah Dasar Darul Hikam

Cikal bakal yayasan Darul Hikam, dirintis oleh K.H.E. Hasbullah Hafidzi pada tahun 1942 yaitu sesaat setelah beliau selesai di pesantren Al-Ianah Cianjur. Kegiatan pertama yang dilaksanakan adalah menyelenggarakan Madrasah Islam di kampung Cisitu Girang, kota Bandung. Pada saat revolusi fisik, madrasah ini dihentikan karena rakyat harus mengungsi meninggalkan Bandung. Sekembalinya dari pengungsian tahun 1949, madrasah dibuka kembali dan dinaikkan statusnya menjadi Sekolah Dasar Islam dengan nama Sekolah Rakyat Muslimin.

Alhamdulillah tahun 1951, sekolah sudah mempunyai bangunan milik sendiri di atas lahan pinjaman ustadz Abdussalam. Yayasan ini terus berkembang, tahun 1953 kegiatan pendidikan ditingkatkan dengan membuka sekolah tingkat SMP dengan waktu belajar pada siang harti. Dalam waktu singkat SMP tersebut dipindahkan ke jln Puyuh No.5 dengan nama SMP Muslimin.

Selama prosese pertumbuhannya, yayasan Darul Hikam terus didera dengan hambatan dan tantangan yang tidak ringan, sebagai cuplikan sejarah pada tahun 1963 bangunan SD Islam yang berlokasi di Cisitu Girang (Sekarang Jalan Cisitu Indah) mendapat musibah yaitu ambruk diterpa angina kencang. Sehingga para murid dipindahkan ke sekolah-sekolah yang berada di sekitar lokasi.

Setelah bangunan SD Islam di Cisitu Girang hancur, perintisan dimulai lagi pada tahun 1963 dibuka Taman Kakak-kanak Islam yang bertempat di rumah Hj. Dedeh Ruyati Hasbullah di Jalan Ir. H. Juanda 212 Bandung. Sekolah ini bertahan

sampai peristiwa pemberontakan G.30.S/PKI gagal. Atas izin Allah SWT, setelah pemberontakan G.30.S/PKI gagal, perjuangan keras K.H.E. Hasbullah Hafidzi akhirnya berhasil membangun Mesjid Darul Hikam yang berukuran 12m X 8m di Jalan Ir.H.Juanda 285 (lokasi sekarang).

Setelah pembangunan mesjid dianggap selesai, secara bertahap diselenggarakan pendidikan formal yang berorientasi kepada kurikulum Departemen Pendidikan Nasional seperti :

a) Taman Kanak-kanak (TK) Th. 1966.
b) Sekolah Dasar (SD) Th. 1968.
c) Sekolah Menengah Pertama (SMP) Th. 1972.
d) Sekolah Pendidikan Guru (SPG) Th. 1974.
e) Sekolah Menengah Atas (SMA) Th. 1981.
f) Sekolah Menengah Ekonomi Atas (sekarang Sekolah Menengah Kejuaruan) Th. 1987.
g) Taman Kanak-kanak II (TK II) Rancaekek Th. 1991.
h) Diploma 3 Lembaga Ilmu Pengembangan Profesi Indonesia (LIPPI) Th. 1996-1998.
i) Bimbel Muslim Averous (Th. 1991-1997) beekerjasama dengan yayasan Ibnu Rusydi.
j) Sekolah Dasar II (SD II) Rancaekek Th. 2006.

Lembaga-lembaga Sosial Kemasyarakatan :

a) Panti Asuhan Arief Rahman Th. 1968.
b) Badan Usaha Warga Darul Hikam Th. 1970 dan Koperasi Warga Darul Hikam Th. 1984.
c) Ikatan Pengajian Ibu-ibu yang mengkoordinir 72 Majelis Ta’lim di Bandung.
d) Baitul Mal wa Tanwil (BMT Th. 1994-1998).

Pada Tahun 1998, melakukan kerjasama dengan yayasan Al Ihsan dan membentuk Perguruan Darul Hikam wal Ihsan. Tahun 1998 berupaya mempertajam visi dan misi yang jelas, kearah pemeliharaan dan pengembangan fitrah melalui pengembangan system sekolah kecil berprestasi. Tahun 2006 mendapatkan nilai akreditasi A+ (sangat baik).

2.1.2 Visi, Misi, dan Tujuan Sekolah Dasar Darul Hikam

Sekolah Dasar Darul Hikam juga mempunyai visi, misi, dan tujuan yang merupakan target yang akan dicapai untuk kesuksesan sekolah kedepannya. Uraian visi, misi dan tujuan dari sekolah dasar darul hikam akan dijelaskan selanjutnya.

2.1.2.1 Visi

Menjadi sekolah Dasar Islam terbaik melalui budaya (Jati diri, ciri khas, dan keunggulan) berakhlak berpretasi pada level Kota Bandung tahun 2011.

2.1.2.2 Misi

1. Melaksanakan pendidikan sekolah dasar Islam secara utuh terpadu dan sempurna untuk membangun akhlakul karimah, siswa, dan semua civitas akademika.
2. Melaksanakan pendidikan umum secara utuh, terpadu dan sempurna untuk meraih prestasi siswa dan civitas akademika dalam berbagai bidang pendidikan.
3. Membangun citra baik sekolah dasar Islam sebagai bagian dari sistem pendidikan Nasional.
4. Membangun silaturahim dan kerjasama dengan orang tua dalam proses pendidikan Islam bagi putra-putrinya.
5. Terimplementasikanya jati diri budaya berakhlak berprestasi secara bertahap dan sempurna.

2.1.2.3 Tujuan

Tujuan pendidikan dasar adalah meletakkan dasar kecerdasan, pengetahuan, kepribadian, akhlak mulia, serta keterampilan untuk hidup mandiri dan mengikuti pendidikan lebih lanjut. Merujuk pada tujuan pendidikan dasar tersebut, maka tujuan Sekolah Dasar Darul Hikam adalah sebagai berikut:

1. Dapat mengamalkan ajaran agama hasil proses pembelajaran dan kegiatan pembiasaan.
2. Meraih prestasi akademik maupun non akademik minimal tingkat Kabupaten/Kota.
3. Menguasai dasar-dasar ilmu pengetahuan dan teknologi sebagai bekal untuk melanjutkan ke sekolah yang lebih tinggi.
4. Menjadi sekolah pelopor dan penggerak di lingkungan masyarakat sekitar.
5. Menjadi sekolah yang diminati di masyarakat.

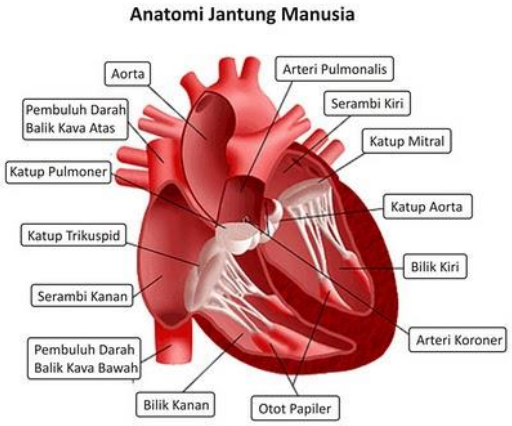


Gambar 2.1 Logo Darul Hikam Bandung [6]

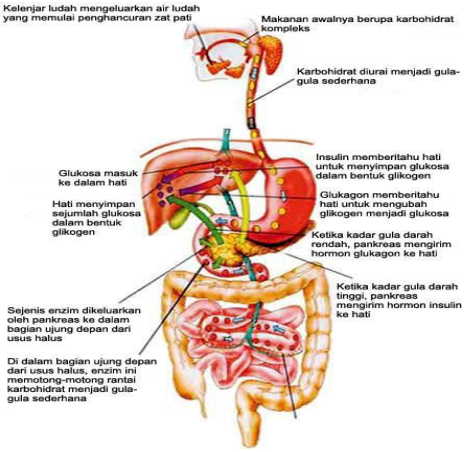
2.2 Pengertian Anatomi

Anatomi adalah cabang dari biologi yang berhubungan dengan struktur dan organisasi dari makhluk hidup. Anatomi berasal dari bahasa Yunani *anatomia* dari *anatemnein* yang berarti memotong.

Bagian paling penting dari anatomi khusus adalah yang mempelajari tentang manusia dengan berbagai macam pendekatan yang berbeda. Dari sudut medis, anatomi terdiri dari berbagai pengetahuan tentang bentuk, letak, ukuran, dan hubungan berbagai struktur dari tubuh manusia sehat sehingga sering disebut sebagai anatomi deskriptif atau topografis. Kerumitan tubuh manusia menyebabkan hanya ada sedikit ahli anatomi manusia profesional yang benar-benar menguasai bidang ilmu ini, sebagian besar memiliki spesialisasi di bagian tertentu seperti otak atau bagian dalam. Contoh anatomi pada gambar 2.2 dan gambar 2.3.



Gambar 2.2 Anatomi Jantung



Gambar 2.3 Anatomi Pencernaan

2.3 Pengertian *Augmented Reality*

Augmented Reality (AR) adalah suatu teknik yang memperluas lingkungan dunia nyata dengan beberapa objek *virtual* yang muncul dalam pandangan *user*. *Augmented Reality* memungkinkan objek 3D *virtual* untuk diintegrasikan kedalam lingkungan nyata secara *real-time*. Bisa juga disebut pencampuran dunia nyata dan dunia maya.

Menurut Stephen Cawood & Mark Fiala bahwa *Augmented Reality* merupakan cara alami untuk mengeksplorasi objek 3D dan data, AR merupakan suatu konsep perpaduan antara *Virtual* Reality dengan *World Reality*. Sehingga objek-objek *Virtual* 2D atau 3D seolah-olah terlihat nyata dan menyatu dengan dunia nyata. Pada teknologi AR, pengguna dapat melihat dunia nyata yang ada di sekelilingnya dengan penambahan objek *virtual* yang dihasilkan oleh komputer. [4]

Sedangkan menurut Ronald T. Azuma mendefinisikan *Augmented Reality* sebagai penggabungan benda-benda nyata dan maya dilingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antar benda dalam 3D, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktifitas dimungkinkan melalui perangkat-perangkat input tertentu, dan integrasi yang baik memerlukan penjejakan yang efektif. [3]

2.3.1 Sejarah *Augmented Reality*

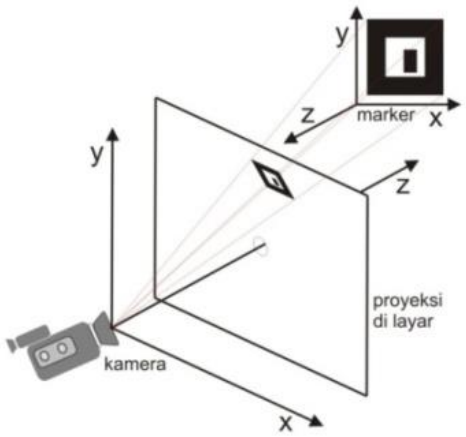
Sejarah tentang *Augmented Reality* dimulai dari tahun 1957-1962, ketika seorang penemu yang bernama Morton Heilig, seorang sinematografer, menciptakan dan memapatenkan sebuah simulator yang disebut Sensorama dengan visual, getaran dan bau. Pada tahun 1966, Ivan Sutherland menemukan *head-mounted display* 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]

2.3.2 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 2.4, 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 2.4 Sistem Koordinat Lingkungan AR

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.

Transformasi Translasi :

$$\begin{bmatrix} Xc' \\ Yc' \\ Zc' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix} \quad (1)$$

Transformasi Rotasi :

$$\begin{bmatrix} Xc' \\ Yc' \\ Zc' \\ 1 \end{bmatrix} = \begin{bmatrix} R11 & R12 & R13 & 0 \\ R21 & R22 & R23 & 0 \\ R31 & R32 & R33 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix} \quad (2)$$

Transformasi Proyeksi Perspektif :

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} fc & 0 & u & 0 \\ 0 & fc & v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix} \quad (3)$$

Transformasi Objek Pada Sistem AR :

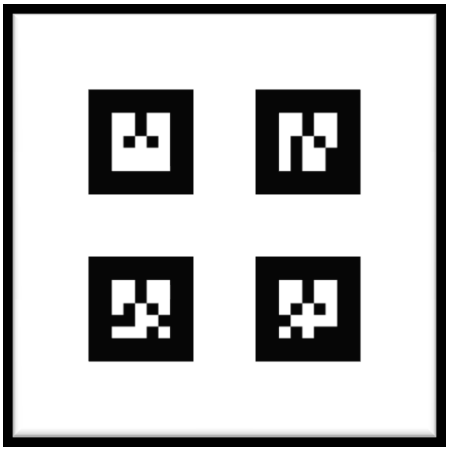
$$\begin{bmatrix} 0x \\ 0y \\ 0z \\ 1 \end{bmatrix} = \begin{bmatrix} fc & 0 & u & 0 \\ 0 & fc & v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R11 & R12 & R13 & Tx \\ R21 & R22 & R23 & Ty \\ R31 & R32 & R33 & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Cx \\ Cy \\ Cz \\ 1 \end{bmatrix} \quad (4)$$

2.3.3 Marker

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

2.3.3.1 *Fiducial Marker*

Fiducial Marker adalah bentuk paling sering digunakan oleh teknologi AR karena *marker* ini digunakan untuk melacak benda-benda di *Virtual Reality* tersebut. Kotak hitam dan putih digunakan sebagai titik referensi atau untuk memberikan skala dan orientasi ke aplikasi. Bila penanda tersebut deteksi dan dikenali maka *Augmented Reality* akan keluar dari marker. Contoh *Fiducial Marker* pada gambar 2.5.



Gambar 2.5 *Fiducial Marker*

2.3.3.2 *Quick Response (QR) Code*

QR Code merupakan 2D *code* yang terdiri dari banyak kotak diatur dalam pola persegi, biasanya *QR* ini berwarna hitam dan putih. *QR Code* diciptakan di Jepang pada awal 1990-an dan digunakan untuk melacak berbagai bagian dalam manufaktur kendaraan. Dan saat ini *QR* digunakan sebagai *link* cepat ke *website*, *dial* cepat untuk nomor telepon, unduh aplikasi, atau bahkan dengan cepat mengirim pesan SMS. Contoh *QR Code* pada gambar 2.6.



Gambar 2.6 *Quick Response (QR) Code*

2.3.3.3 *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. Contoh *Markerless Marker* pada gambar 2.7.



Gambar 2.7 *Markerless Marker*

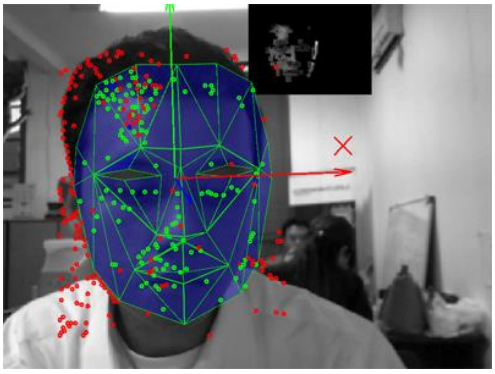
Markerless ini salah satu metode *Augmented Reality* yang saat ini sedang berkembang. Dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan objek. Dalam perancangan nya, seolah-olah *markerless* menggabungkan objek maya dengan objek nyata, dalam hal ini objek maya 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 :

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

Adapun pengembangan markerless yang sudah dikembangkan oleh pengembang didunia seperti :

1) *Face Tracking*

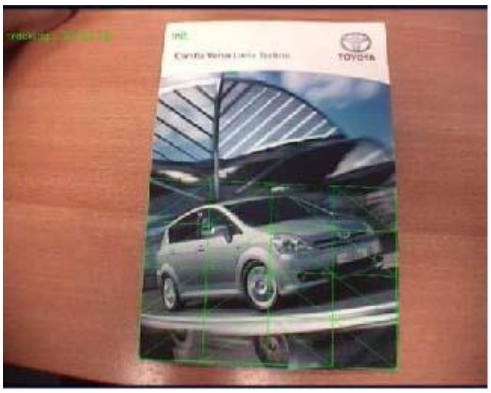
   Dengan menggunakan alogaritma yang mereka kembangkan, komputer dapat mengenali wajah manusia secara umum dengan cara mengenali posisi mata, hidung, dan mulut manusia, kemudian akan mengabaikan objek-objek lain di sekitarnya seperti pohon, rumah, dan benda-benda lainnya.



Gambar 2.8 *Face Tracking*

2) 3D Object Tracking

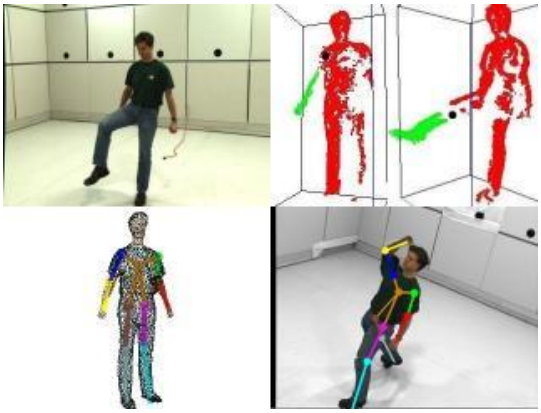
Berbeda dengan Face Tracking yang hanya mengenali wajah manusia secara umum, teknik 3D Object Tracking dapat mengenali semua bentuk benda yang ada disekitar, seperti mobil, meja, televisi, dan lain-lain.



Gambar 2.9 *3D Object Tracking*

3) Motion Tracking

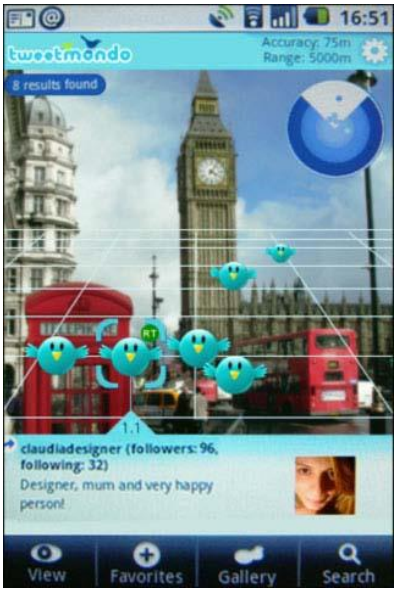
Pada teknik ini komputer dapat menangkap gerakan, Motion Tracking telah mulai digunakan secara ekstensif untuk memproduksi film-film yang mencoba mensimulasikan gerakan. Contohnya pada film Avatar, di mana James Cameron menggunakan teknik ini untuk membuat film tersebut dan menggunakannya secara realtime.



Gambar 2.10 *Motion Tracking*

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 Layar.



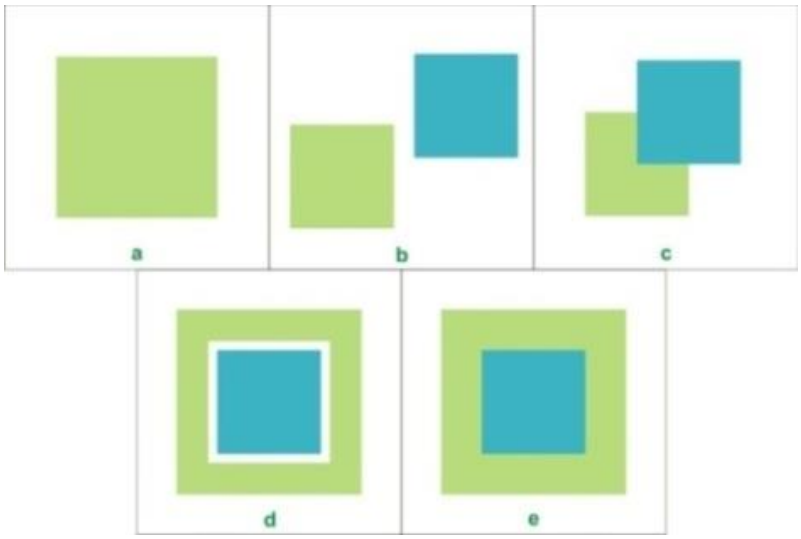
Gambar 2.11 *GPS Based Tracking*

2.3.3.4 *Multi Marker*

Multi Marker adalah 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*.

2.3.4 Occlusion Based Interaction

Occlusion adalah hubungan antara suatu benda dengan benda lain jika kita lihat dari suatu sudut pandang. Hal ini tentunya mengurangi informasi antar objek dalam lingkungan 3D, karena jika dilihat dari satu sudut pandang maka lingkungan 3D akan diproyeksikan kepada suatu bidang sehingga seolah-olah menjadi lingkungan 2D. Pengurangan dimensi ini menyebabkan informasi interaksi antar objek seperti keadaan bersinggungan, atau berapa jarak antar objek akan menjadi ambigu.



Gambar 2.12 : *Occlusion* yang terjadi karena interaksi antar objek (a) *None* (b) *Proximity* (c) *Intersection* (d) *Enclosement* (e) *Containment*

Occlusion Detection adalah metode untuk mendeteksi ada tidaknya *occlusion* dalam penampilan objek 3D. Pada [Gun A, Mark, dan Gerard, 2004] secara sederhana *occlusion detection* hanya mendefinisikan keadaan dimana suatu marker tidak terdeteksi karena tertutup oleh benda lain. Sedangkan pada [Volkert, Stephen, Mark, 2004] menggunakan *occlusion detection* berdasarkan posisi koordinat 2D dari dua objek yang ada.

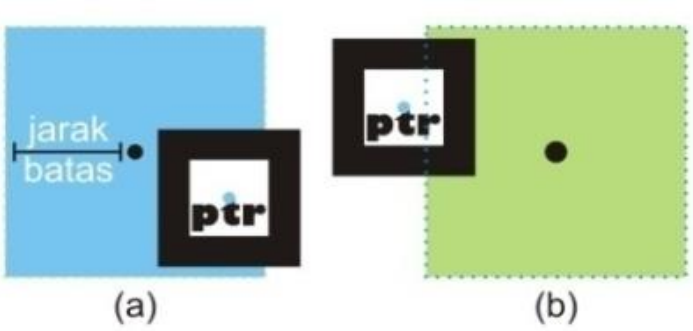
Jika ada n objek yang mewakili matrik O, maka akan dihasilkan matrik O1, O2, ... , On yang merupakan posisi proyeksi objek-1, objek-2, ... , objek-n di layar. Deteksi *occlusion* akan dilakukan dengan pengecekan 2 objek misal dipilih objek-1 terhadap objek-2 maka akan dilakukan pengecekan syarat pertidaksamaan *point clipping* [Donald, dan Baker, 1996] berikut :

$$O2x - batas \leq O1x \leq O2x + batas \quad (5)$$

$$O2y - batas \leq O1y \leq O2y + batas \quad (6)$$

Hasil deteksi ini berupa nilai kebenaran yang merupakan dasar pendefinisian *event* dari interaksi *occlusion based* jika pertidaksamaan 5 dan 6 terpenuhi.

Occlusion Based Interaction adalah sebuah desain interaksi eksosentris [Hannah, Matthew, Rudi, dan Bruce, 2001] dimana dalam mendefinisikan *event* untuk menghasilkan aksinya menggunakan metode *occlusion detection*. Desain interaksi yang menggunakan proyeksi 2D dari objek 3D ini mengurangi kompleksitas yang diperlukan dalam mendesain interaksi dalam sistem AR lain yang menggunakan acuan bidang 3D.



Gambar 2.13 : (a) Terjadi *Event* (b) Tidak Terjadi *Event*

Jika titik biru ditengah *marker* ptr adalah objek O1 dan titik hitam adalah objek O2 maka gambar 2.13 (a) dikatakan terjadi *event* karena memenuhi pertidaksamaan 5 dan 6 yaitu koordinat O1 (x,y) ada di dalam batas area O2. Sedangkan gambar 2.13 (b) tidak terjadi *event* karena hanya memenuhi pertidaksamaan 6 (nilai O1y ada dalam batas O2y) namun tidak memenuhi pertidaksamaan 5. [5]

2.3.5 MetaIO

MetaIO dikenal sebagai salah satu platform teknologi *Augmented Reality* (AR) yang populer saat ini. Beberapa tools AR telah diproduksi untuk keperluan pengembangan aplikasi AR bagi *end user*. Kemudahan dan fleksibilitas *tools* untuk mengembangkan aplikasi AR menjadi hal penting karena kiprah teknologi AR yang cemerlang, dapat digunakan untuk berbagai macam aplikasi. [8]



Gambar 2.14 Logo MetaIO

2.3.5.1 MetaIO Library

MetaIO Library adalah *library engine* yang mendukung untuk membangun *Augmented Reality*. *Metaio* merupakan *engine* AR paling hebat diantara *engine-engine* AR lainnya. Karena pemakaiannya mudah, fitur banyak, mau buat AR apa saja mudah tanpa harus melakukan *Coding*.

Kelas dasar perpustakaan *MetaIO* adalah kelas *MetaObject*. Ini mendefinisikan satu *set* dasar *tag* yang umum untuk semua *metaObjects* seperti *MetaImages*, *MetaTubes*, dll.

2.3.6 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

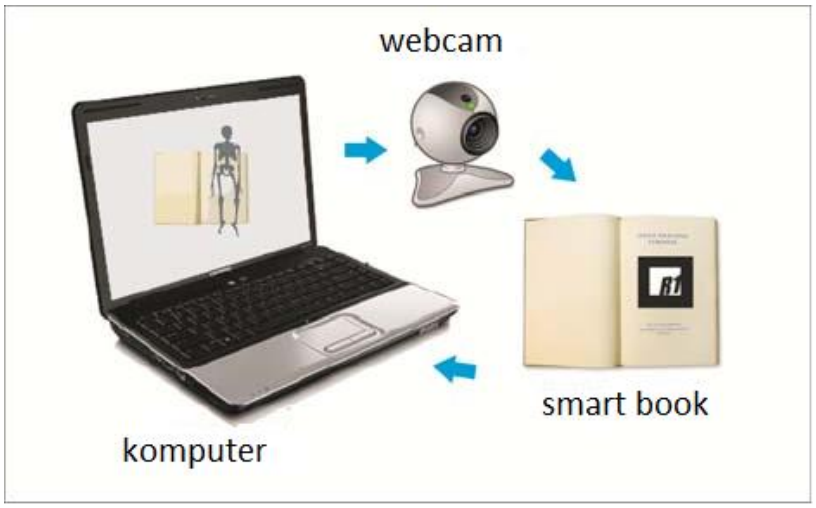
   Seorang desainer membutuhkan *Augmented Reality* untuk menampilkan hasil desain mereka secara lebih nyata terhadap klien. Dengan AR klien dapat mengetahui tentang spesifikasi yang lebih detail tentang desain.

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.

2.4 Pengertian *Smart Book*

Pengertian *Smart Book* adalah sebuah buku yang dapat menampilkan model animasi 3D dengan cara melakukan pembacaan simbol *marker* menggunakan kamera. Aplikasi buku berbasis *Augmented Reality* yang dikembangkan ini merupakan penggabungan antara media cetak dan teknologi komputer yang nantinya digunakan sebagai aplikasi *desktop* dengan menggunakan metode 2D *markerless* sebagai *marker*. Pada aplikasi ini membutuhkan alat bantu *webcam* sebagai media *input* untuk membaca *marker* (penanda khusus), dimana dari *marker* tersebut akan menampilkan model – model dan animasi 3D (tiga dimensi) pada layar monitor. Buku berbasis *Augmented Reality* ini sendiri hasil akhirnya terdiri dari dua bentuk yaitu dalam bentuk fisik (media cetak berupa buku) yang berisikan *marker* pada beberapa halamannya dan aplikasi *Augmented Reality* berbasis *desktop* dimana keduanya saling melengkapi.



Gambar 2.15 Implementasi *Smart Book* Menggunakan *Augmented Reality*

2.5 UML

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sistem dengan menggunakan diagam dan teks-teks pendukung.

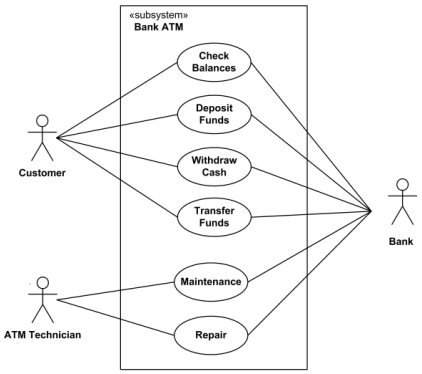
UML hanya berfungsi melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tetentu, meskipun pada kenyataanya UML paling banyak digunakan pada metodologi berorientasi objek.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain. [2]

2.5.1 *Use Case* Diagram

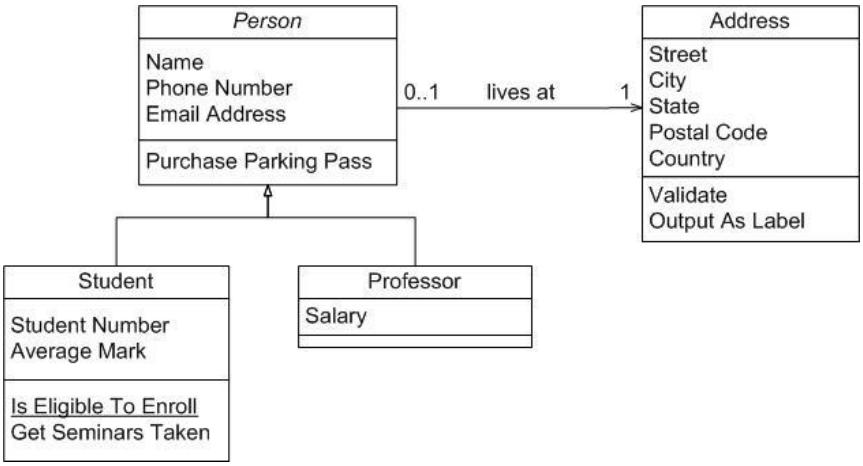
Menggambarkan sejumlah eksternal aktor dan hubungannya ke *use case* yang diberikan oleh sistem seperti pada gambar 2.16.



Gambar 2.16 *Use Case Diagram*

2.5.2 *Class Diagram*

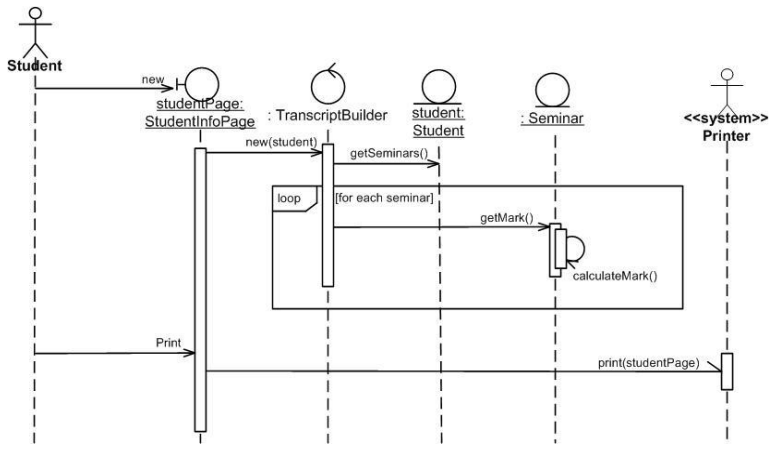
Menggambarkan struktur dan deskripsi *class*, package (paket) dan objek beserta hubungan satu sama lain seperti *containment* (penahanan), pewarisan, asosiasi dan lain-lain seperti pada gambar 2.17.



Gambar 2.17 *Class Diagram*

2.5.3 *Sequence 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 2.18.

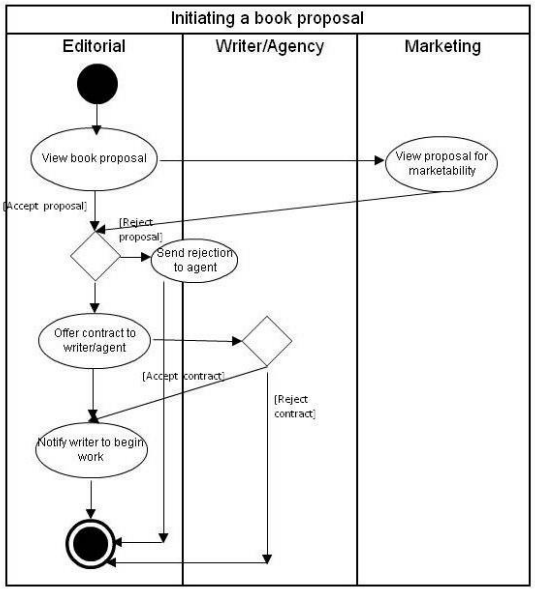


Gambar 2.18 *Sequence Diagram*

2.5.4 *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 2.19.



Gambar 2.19 *Activity Diagram*

2.5.5 Kegunaan UML

1) Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
2) Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

2.6 Bahasa Pemrograman C++

C++ adalah bahasa pemrograman tingkat tinggi yang memungkinkan seorang *programmer* perangkat lunak untuk berkomunikasi secara efisien dengan komputer. C++ adalah bahasa yang sangat fleksibel dan mudah beradaptasi. Sejak didirikan pada tahun 1980, C++ telah digunakan untuk berbagai program termasuk *firmware* untuk *mikro-controller*, sistem operasi, aplikasi, dan program grafis. C++ dengan cepat menjadi bahasa pemrograman pilihan. Ada permintaan yang luar biasa bagi orang-orang yang dapat memberitahu komputer apa yang harus dilakukan, dan C++ memungkinkan Anda melakukannya dengan cepat dan efisien. [10]

2.6.1 Sejarah

Pada tahun 1970 dua programer Brian Kernighan dan Dennis Ritchie, menciptakan bahasa pemrograman baru yang disebut C. C dirancang dengan satu tujuan didalam pikiran yaitu menulis sistem operasi. Bahasa ini sangat sederhana dan fleksibel dan dapat segera digunakan untuk berbagai jenis program. C dengan cepat menjadi salah satu bahasa pemrograman yang paling populer di dunia.

Pada tahun 1980 Bjarne Stroustrup mulai bekerja pada sebuah bahasa pemrograman baru yang disebut "*C with Classes*". Bahasa ini meningkat dari C dengan menambahkan sejumlah fitur baru yang paling penting adalah kelas . Bahasa ini diperbaiki, ditambah, dan akhirnya menjadi C++.

C++ berutang keberhasilannya pada fakta bahwa hal itu memungkinkan *programmer* untuk mengatur dan memproses informasi lebih efektif daripada kebanyakan bahasa lain. Bahkan kebanyakan program C dapat diubah menjadi C++ program dengan sedikit kesulitan. Program-program ini biasanya tidak menggunakan semua fitur baru dari C++, tetapi mereka yang melakukan. Dengan cara ini , C++ memungkinkan *programmer* untuk membangun basis yang ada kode C.

2.6.2 Penulisan

Program Bahasa C++ tidak mengenal aturan penulisan di kolom tertentu, jadi bisa dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan bahasa C++ diatur sedemikian rupa sehingga mudah dan enak dibaca. Berikut contoh penulisan Program Bahasa C++ :

```
//my first program in C++
#include <iostream.h>
int main()
{
   cout << "Hello World!" ;
   return 0;
}
```

Sisi kiri merupakan source code, yang dapat diberi nama hiworld.cpp dan sisi kanan adalah hasilnya setelah di-kompile dan di-eksekusi. Program diatas merupakan salah satu program paling sederhana dalam C++, tetapi dalam program tersebut mengandung komponen dasar yang selalu ada pada setiap pemrograman. C++. Jika dilihat satu persatu :

// my first program in C++

Baris ini adalah komentar. semua baris yang diawali dengan dua garis miring (//) akan dianggap sebagai komentar dan tidak akan berpengaruh terhadap program. Dapat digunakan oleh programmer untuk menyertakan penjelasan singkat atau observasi yang terkait dengan program tersebut.

#include <iostream.h>

Kalimat yang diawali dengan tanda (#) adalah are preprocessor directive. Bukan merupakan baris kode yang dieksekusi, tetapi indikasi untuk kompiler. Dalam kasus ini kalimat **#include <iostream.h>** memberitahukan preprocessor kompiler untuk menyertakan header file standard **iostream**. File spesifik ini juga termasuk library deklarasi standard I/O pada C++ dan file ini disertakan karena fungsi-fungsinya akan digunakan nanti dalam program.

int main ()

Baris ini mencocokan pada awal dari deklarasi fungsi **main**. Fungsi **main** merupakan titik awal dimana seluruh program C++ akan mulai dieksekusi. Diletakan diawal, ditengah atau diakhir program, isi dari fungsi **main** akan selalu dieksekusi pertama kali. Pada dasarnya, seluruh program C++ memiliki fungsi **main**. **main** diikuti oleh sepasang tanda kurung **()** karena merupakan fungsi. Pada C++, semua fungsi diikuti oleh sepasang tanda kurung **()** dimana, dapat berisi argumen didalamnya. Isi dari fungsi main selanjutnya akan mengikuti,berupa deklarasi formal dan dituliskan diantara kurung kurawal **({})**, seperti dalam contoh.

cout << "Hello World";

Intruksi ini merupakan hal yang paling penting dalam program contoh. **cout** merupakan standard output stream dalam C++ (biasanya monitor). **cout**

dideklarasikan dalam header file iostream.h, sehingga agar dapat digunakan maka file ini harus disertakan. Perhatikan setiap kalimat diakhiri dengan tanda semicolon (;). Karakter ini menandakan akhir dari instruksi dan harus disertakan pada setiap akhir instruksi pada program C++ manapun.

return 0;

Intruksi **return** menyebabkan fungsi **main()** berakhir dan mengembalikan kode yang mengikuti instruksi tersebut, dalam kasus ini **0**. Ini merupakan cara yang paling sering digunakan untuk mengakhiri program.

Tidak semua baris pada program ini melakukan aksi. Ada baris yang hanya berisi komentar (diawali //), baris yang berisi instruksi untuk preprocessor kompiler (yang diawali **#),**kemudian baris yang merupakan inisialisasi sebuah fungsi (dalam kasus ini, fungsi main) dan baris yang berisi instruksi (seperti, **cout** <<), baris yang terakhir ini disertakan dalam blok yang dibatasi oleh kurung kurawal **({})** dari fungsi **main**.

2.6.3 Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh computer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Tabel 2.1 Bentuk Tipe Data

No	Tipe Data	Ukuran	Range (Jangkauan)	Keterangan
1	*char*	1 byte	signed: -128 to 127 unsigned: 0 to 255	Character or integer 8 bits length.
2	*short*	2 byte	signed: -32768 to 32767 unsigned: 0 to 65535	Integer 16 bits length.
3	*long*	4 byte	signed:-2147483648 to 2147483647 unsigned: 0 to 4294967295	Integer 32 bits length.

4	*int*	*	See short, long	Integer. Its length traditionally depends on the length of the system's Word type, thus in MSDOS it is 16 bits long, whereas in 32 bit systems (like Windows 9x/2000/NT and systems that work under protected mode in x86 systems) it is 32 bits long (4 bytes).
5	*float*	4 byte	3.4e + / - 38 (7 digits)	floating point number.
6	*double*	8 byte	1.7e + / - 308 (15 digits)	double precision floating
7	*long double*	10 byte	1.2e + / - 4932 (19 digits)	long double precision
8	*bool*	1 byte	true or false	Boolean value. It can take one of two values: true or false NOTE: this is a type recently added by the ANSI-C++ standard. Not all compilers support it. Consult section bool type for compatibility
9	*wchar_t*	2 byte	wide characters	Wide character. It is designed as

2.7 *Visual Studio 2010*

Visual Studio 2010 (VS) adalah sebuah perangkat lunak buatan *microsoft* yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *windows*, ataupun aplikasi *web*. *Visual Studio* mencakup kompiler, SDK, *Integrated Development Environment* (*IDE*), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara

lain *Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro,* dan *Visual SourceSafe*.



Gambar 2.20 Logo *Microsoft Visual Studio* [7]

2.8 *3D Studio 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.



Gambar 2.21 Logo *3D Studio Max* [1]