BAB II

LANDASAN TEORI

2.1. Grafika Komputer

Grafika komputer adalah salah satu cabang ilmu komputer, perkembangan serta pemanfaatanya grafika komputer belakangan ini semakin pesat seiring banyaknya ditemukannya teknologi-teknologi baru. Grafika komputer dapat diartikan sebagai seperangkat alat yang terdiri hardware dan software untuk membuat grafik, gambar atau citra realisitis untuk yang dapat di terapkan pada bidang seni, game atau permainan komputer, foto atau film animasi. Grafika komputer secara umum dapat dibagi menjadi dua bagian, yaitu grafik dua dimensi dan tiga dimensi. Grafika tiga dimensi memiliki keunggulan yaitu dapat menggambarkan sebuah benda yang dapat dilihat dari berbagai sudut pandang.

2.2. Permainan atau game

Permainan merupakan sebuah aktivitas rekreasi dengan tujuan bersenang-senang, mengisi waktu luang, atau berolahraga ringan. Permainan biasanya dilakukan sendiri atau bersama-sama.

a. Permainan anak-anak

Di lingkungan yang masih terlihat keakraban antar anggota masyarakat, banyak permainan yang dilakukan oleh anak-anak secara beramai-ramai dengan teman-teman mereka di halaman atau di teras rumah. Mereka berkelompok, berlarian, atau duduk melingkar memainkan salah satu

permainan dan tercipta keakraban. Beberapa permainan ini karena tercipta di masa yang lama berlalu disebut dengan *permainan tradisional*, sedangkan di sisi lain beberapa permainan yang lebih akhir (dan biasanya menggunakan peralatan yang canggih) disebut *permainan modern*.

1. Bentengan
2. Boneka
3. Bulu Ayam
4. Congkak
5. Dor Tap
6. Galasin
7. Gansingan
8. Gatrik
9. Goncang Kaleng
10. Kasti
11. Kuda Panjang
12. Layangan(layang-layang)
13. Lego
14. Main Hadang
15. Pam Semambu
16. Petak Umpet
17. Tam-tambuku
18. Yoyo

a. Permainan dewasa

1. Kartu Remi atau Bridge
2. Surakarta
3. Catur
4. Domino
5. Kuis
6. Sudoku

b. Permainan komputer

1. Permainan komputer
2. Permainan video
3. Permainan online

Mendengar kata ”game”, tentu kita langsung teringat pada konsol *Playstation*, atau mungkin komputer pribadi. Game memang menjadi salah satu hiburan yang mengasyikkan, tidak hanya bagi anak-anak dan remaja, terkadang, orang dewasa pun masih menikmati game di waktu senggang mereka.

2.2.1. Sejarah Game

1. Generasi Awal

Video game pertama yang tercatat diciptakan oleh *William Higinbotham*, seorang ilmuan pada tahun 1958. Game itu dinamakan *“Tennis For Two”*, game ini tidak pernah di release secara komersil karena bukan merupakan bentuk video game yang sebenarnya. Karena merupakan eksperimen elektronik. Banyak yang berspekulasi bahwa ide William Higinbotham dicuri oleh pencipta *Pong* pada tahun 1971, karena penemuan William Higinbotham tidak terlalu dikenal, maka masyarakat dibuat takjub saat game Pong keluar pada tahun 1975.

2. Generasi I

Lalu pada tahun 1972, saat itu masih belum banyak yang mengenal apa itu konsol, apa itu game komputer, yang ada hanya istilah ”video game”, yaitu sebuah permainan elektronik yang menampilkan gambar bergerak (video). Pada tahun 1972 ini, sebuah perusahaan bernama *Magnavox* meluncurkan video game pertama, yaitu *Odyssey*.



Gambar 2.1 Magnavox Odyssey, konsol game pertama di dunia mengoperasikan **Pong**.
(sumber : http://www.jagogame.com)

Odyssey bukanlah sebuah mesin yang sukses, penjualannya tidak menunjukkan antusias konsumen. Tidak lama berselang, sebuah game *arcade* legendaris *Atari* bertitel *Pong* muncul. *Pong* merupakan sebuah game sederhana yang mengambil konsep permainan tenis, satu bola dan 2 papan di kiri dan kanan, pemain sebisa mungkin harus berusaha mengembalikan bola ke daerah lawan. Atari merilis *Pong* dalam bentuk sebuah mesin ding dong bernama *Sears*. Akhirnya, pada tahun 1975, *Magnavox* menyerah dan menghentikan produksi *Odyssey*. Sebagai gantinya, mereka mengikuti jejak *Atari*, memproduksi mesin ding dong bernama *Odyssey 100*, yang khusus menyajikan game *Pong*.

3. Generasi II

Pada tahun 1976, *Fairchild* mencoba menghidupkan kembali dunia video game dengan menciptakan *VES (Video Entertainment System)*. *VES* adalah mesin pertama yang disebut ”konsol”. Konsol ini menggunakan kaset magnetik yang disebut *cartridge*. Konsep ini kemudian diikuti oleh beberapa produsen lain, termasuk *Atari*, *Magnavox*, dan *RCA*, ketiga perusahaan tersebut juga merilis konsol serupa.



Gambar 2.2 Fairchild VES, pertama di dunia yang menggunakan media *cartridge.*
(sumber : http://www.jagogame.com)

Pada tahun 1977, dunia konsol menjadi tidak populer, game-game yang ada tidak berhasil menarik minat. *Fairchild* dan RCA mengalami kebangkrutan. Praktis, hanya ada *Atari* dan *Magnavox* yang masih bertahan di dunia video game. Sekitar tahun 1978, *Magnavox* meluncurkan *Odyssey 2*, seperti halnya *Odyssey* pertama, konsol ini pun gagal menjadi hit. Tak lama berselang, Atari meluncurkan konsol legendaris, *Atari 2600*, yang terkenal

dengan game *Space Invaders*-nya. Dan mulai tahun 1980, berbagai produsen konsol muncul, dan mereka mengambil *Atari 2600* sebagai konsep dasar, perkembangan dunia game pun semakin pesat.



Gambar 2.3 Atari 2600, sempat hit tahun 80-an. Memiliki "adik" bernama Atari 2600 Jr.
(sumber : http://www.jagogame.com)

Tiga tahun berselang, tepatnya tahun 1983, dunia video game kembali ambruk. Game-game yang kurang kreatif membuat konsol kembali mendapat sambutan dingin, apalagi, PC saat itu menjadi semakin canggih. Orang lebih memilih membeli PC ketimbang konsol video game, selain untuk bermain, PC juga diproduksi untuk membantu pekerjaan.

Kampanye seperti ini lah yang terjadi di pasar dan membuat hampir seluruh perusahaan konsol mengalami kebangkrutan. Dan tahun 1983 ini menjadi titik di mana game-game komputer (PC Game) semakin berkembang pesat, hingga saat ini. Pelopor PC ber-game saat itu adalah Commodore 64, konsol sekaligus *personal computer* yang menyediakan tampilan grafis 16-warna dan

memiliki kapasitas memori jauh lebih baik dari konsol videogame model apa pun.

4. Generasi III

Hancurnya dunia konsol, menggugah perusahaan bernama Famicom untuk mencoba industri videogame kembali. Perusahaan Jepang ini menciptakan gebrakan baru, sebuah konsol bernama *Famicom/Nintendo Entertainment System* (NES) dirilis di akhir 1983. Konsol ini menampilkan gambar dan animasi resolusi tinggi untuk pertama kalinya. Setelah mendapat sambutan hangat di Jepang, Famicom memperluas pemasarannya ke Amerika, yang dikenal dengan NES (Nintendo Entertainment System). Nintendo memiliki *chip* pengaman pada *cartridge* game mereka, dengan demikian seluruh game yang akan dirilis haruslah seijin developer Nintendo. Dan akhirnya, muncul sebuah game legendaris, *Super Mario Brothers*, yang dibintangi karakter fenomenal yang tetap eksis hingga kini, *Mario*.



Gambar 2.4 Famicom dari Nintendo, berhasil merajai pasar videogame di era generasi ketiga.
(sumber : http://www.jagogame.com)

5. Generasi IV

NES mendapat sambutan hangat di seluruh dunia, dan sebuah perusahaan bernama *Sega* mencoba menyaingi *Nintendo*. Pada tahun 1988, *Sega* merilis konsol *next-generation* mereka, *Sega Mega Drive* (yang juga dikenal dengan *Sega Genesis*). Konsol ini menyajikan gambar yang lebih tajam dan animasi yang lebih halus dibanding *NES*. Konsol ini cukup berhasil memberi tekanan, tetapi *NES* tetap bertahan dengan angka penjualan tinggi. Dua tahun berselang, pada 1990, Nintendo kembali menggebrak dengan konsol *next-gen* mereka, *SNES (Super Nintendo Entertainment System).* Selama 4 tahun, *Nintendo* dan *Sega* menjadi bebuyutan, meskipun ada beberapa produsen seperti *SNK* dengan *NeoGeo*-nya, *NEC* dengan *TurboGrafx*-16 dan *Phillips CD-i,* tapi kedua konsol mereka begitu handal dan populer.



Gambar 2.5 Rival yang legendaris, *Super NES* dan *Mario Brothers* sebagai ikonnya melawan *SEGA Mega Drive* dan *Sonic the Hedgehog* sebagai ikonnya. (sumber : http://www.jagogame.com)

6. Generasi V

Mulai dari 1990 sampai 1994, Sega dan Nintendo tetap bersaing. Berbagai game fenomenal dirilis. SNES menyertakan *chip* Super FX pada *cartridge* mereka, dan *Sega* menggunakan *Sega Virtua Processor*, keduanya bertujuan untuk meningkatkan kualitas grafis dari game. Alhasil, *SNES* dan *Sega* saling beradu dengan game-game sukses seperti *Donky Kong Country* (*SNES*) dan *Vectorman* (*Sega*). Awal generasi kelima dimulai ketika tahun 1993, sebuah perusahaan ternama, Panasonic, merilis konsolnya yang bernama Panasonic *3DO*. Ini adalah konsol pertama yang menggunakan CD sebagai pengganti *cartridge*. Harganya yang sangat mahal membuat konsol ini tidak populer, *3DO* tidak bertahan lama dan harus segera menghentikan produksinya.



Gambar 2.6 *Panasonic 3DO,* konsol game pertama yang menggunakan media CD.
(sumber : http://www.jagogame.com)

Selanjutnya, tahun 1994, Atari kembali meluncurkan konsol baru untuk menandingi *Nintendo* dan *Sega*. *Atari Jaguar* jelas jauh lebih canggih ketimbang NES maupun Mega Drive, tetapi

penggunaannya yang sulit menjadi batu sandungan, belum lagi, pada tahun yang sama, Sony merilis konsol super legendaris, PlayStation. Atari bangkrut dan akhirnya melakukan merger.

Konsol basis CD yang pertama kali menuai sukses adalah *Sony PlayStation*. Konsol Jepang ini segera mendapat sambutan hangat, dan hingga saat ini, *PlayStation* sudah terjual ratusan juta unit. *PlayStation* yang juga disebut PS-One merupakan konsol terlaris sepanjang masa. *Sega* dan *Nintendo* tampaknya menyadari ketertinggalan mereka dari *Sony*. *Sega* kemudian merilis *Sega Saturn*, dan *Nintendo* mengeluarkan *Nintendo 64*.



Gambar 2.7 *Sony Playstation* Generasi pertama
(sumber : http://www.jagogame.com)

7. Generasi VI

Setelah jatuhnya *Nintendo* dan *Sega*, kini dunia konsol jadi milik *Sony*. *PlayStation* menjadi raja dan bisa dibilang tidak memiliki pesaing. *Sega* mencoba meluncurkan *Sega Dreamcast* pada 1998 untuk mematahkan dominasi *Sony*, tetapi kembali gagal,

akhirnya pada tahun itu juga, *Sega* mengundurkan diri dari dunia produsen konsol.

Sony semakin ’merajalela’ ketika mereka berhasil merilis konsol barunya, *PlayStation 2*, yang sudah berbasis DVD pada tahun 2000. *Nintendo* mencoba bertahan di dunia konsol dengan merilis *GameCube*. Konsol ini tidak menggunakan DVD 12 cm biasa, melainkan DVD yang berukuran lebih kecil, yaitu 8 cm. Ukuran keping medianya yang membuat *GameCube* kurang populer.



Gambar 2.8 Dari kiri ke kanan: *Nintendo GameCube, Microsoft Xbox, Sony Playstation 2*
(sumber : http://www.jagogame.com)

Satu-satunya pesaing serius *PlayStation 2* adalah *Xbox*. Sebuah konsol keluaran *Microsoft* ini menggebrak dengan tampilan visual yang sangat tajam dan berkualitas yang kala itu lebih menarik dibanding dengan *PlayStation 2*. Sayangnya game-game *Xbox* ternyata tidak sepopuler *PlayStation 2*. Satu game *Xbox* yang menjadi hit dan cukup fenomenal yaitu Halo. Karena game ini udah memanfaatkan fasilitas ‘unggul’ dari *Microsoft*, yaitu *Xbox Live*.

8. Generasi VII

Xbox agak terlambat meluncur ke pasaran dibanding *PlayStation2*, dan *support* game-game tenar juga sangat minim. Tetapi, *Microsoft* seolah belajar dari kesalahannya. Pada saat *Sony* masih melakukan riset untuk konsol *PlayStation 3* yang menggunakan *Blue-Ray*, Microsoft kali ini telah mengambil seribu langkah lebih cepat. Xbox 360, konsol generasi terkini yang memanfaatkan media *HD-DVD* (nantinya) meluncur pada November 2005 silam. *Xbox 360* hadir dengan segudang fitur istimewa, mulai dari grafis, hingga titel-titel game terkenal. Di antaranya *Best Game of The Year 2006* versi beberapa situs game terkemuka, *Gears of War*. Apalagi, Xbox Live semakin disempurnakan, dan mendapat sambutan luar biasa dari para gamer. Kali ini, giliran *Sony* yang terlambat. *PlayStation 3* dirilis pada November 2006, selang seminggu sebelum *Nintendo* meluncurkan terobosannya, yaitu *Nintendo Wii*. Posisi *PlayStation 3* kurang menguntungkan, selain karena *Xbox 360* sudah keburu tenar duluan, Wii juga menawarkan inovasi pada stik kontrol mereka yang *'motion sensitive'*. Apalagi, harga konsol terbaru *Sony* itu merupakan yang paling mahal dibanding dua pesaingnya. Alhasil, penjualan *PlayStation 3* menjadi yang terendah di bawah *Xbox 360* dan *Wii* hingga artikel ini ditulis.



Gambar 2.9 *Xbox 360, Wii, Playstation 3,*menjadi pesaing tetap dari generasi sebelumnya
(sumber : http://www.jagogame.com)

9. Generasi Handle

Merebaknya popularitas game membuat berbagai perusahaan elektronik berusaha membuat terobosan baru. Di antaranya adalah membuat sebuah mesin game berukuran kecil, yang bisa dibawa ke mana pun. Belakangan, konsol pun dibuat mini, serupa dengan *handheld*, tentu saja, ini merupakan sebagian terobosan besar yang tidak boleh dilupakan dalam sejarah game.

Sejarah video game saku ini bermula pada tahun 1976, beberapa piranti dari Mattel dirilis ke pasaran, tetapi tidak begitu populer. Demikian pula dengan *handheld* buatan Milton Bradley yang dilempar ke pasaran tahun 1979 silam. Perusahaan-perusahaan Jepang mulai merambah pasar *handheld* pada awal

1980-an, tetapi tetap sama saja hasilnya. Hal ini terus berlanjut hingga 1984. Pada waktu itu, sebuah nama yang tentu tidak asing sampe sekarang, Game Boy, muncul. *Handheld* buatan Nintendo ini begitu diminati dan dinobatkan sebagai *handheld* pertama di dunia yang angka penjualannya boleh dikatakan sukses.



Gambar 2.10 *Mattel Electronics*, video game saku pertama di dunia.
(sumber : http://www.jagogame.com)

Lima tahun kemudian, pada 1979, Atari mengakhiri era *handheld* hitam putih. Produk andalannya, Atari Lynx, membawa dimensi baru. Ini *handheld* pertama yang mampu menampilkan warna, sekaligus animasi 3D yang sederhana. Tahun 1990, dunia *handheld* semakin menggila, *NEC*, perusahaan elektronik terkemuka di Jepang membuat *handheld* yang mampu me*render* animasi 3D lebih kompleks, karena menerapkan konsep grafis 3D untuk *P*ersonal *Computer* (PC).



Gambar 2.11 Handheld beda generasi: 1. *Sony PSP*, 2. *GamePark XGP*, 3. *GamePark GP32*, 4. *Atari Lynx*, dan 5. *NEC TurboExpress*.
(sumber : http://www.jagogame.com)

Semenjak tahun tersebut, produsen game semakin gencar melakukan riset untuk *handheld*. *Sega* merilis *Game Gear* dan setahun berselang, *Nintendo* memperbarui produknya dengan *Super Game Boy*. Bahkan, *Sega* memproduksi *handheld* tanpa layar, *Mega Jet*, untuk diimplementasikan di pesawat terbang guna menghibur penumpangnya. *Nintendo Virtual Boy* menyusul pada tahun 1994, lengkap dengan kacamata 3D, yang sekarang banyak ditiru untuk pelengkap berbagai paket produk grafis 3D.

Mulai tahun 1995, ada ide untuk mengecilkan ukuran konsol, dimulai dari *Sega Nomad*. Konsol ini membutuhkan *cartridges Sega Mega Drive*, tetapi ukurannya kecil, maka dari itu tergolong *handheld*. Pada tahun 1996, muncul *Neo Geo Pocket*, disusul oleh beberapa variasi *Game Boy Pocket* dan *Game Boy Color*, yang terus berinovasi tiap tahunnya.

Sony merilis *PocketStation* pada 1998 dan memberikan kejutan besar di dunia konsol. *Handheld* ini memiliki kualitas visual yang jauh lebih baik dibanding *handheld* lain yang ada di pasaran. Salah satu pentolan *Nintendo, Gumpei Yokoi*, memutuskan untuk keluar dan bergabung dengan *Bandai*, kemudian merilis *WonderSwan* dan *WonderSwan Color* pada 1999 dan 2000.



Gambar 2.12 *Nokia NGage QD,* usaha pabrikan ponsel Nokia merambah industri game.
(sumber : http://www.jagogame.com)

Pada 2001, *Game Park GP32* muncul. *Handheld* buatan Korea ini sangat unik, selain fitur multimedia, pemiliknya bisa mendesain aplikasi dan game sendiri untuk *GP32*. *Nintendo* juga merilis *Game Boy Advance* pada tahun yang sama. Bahkan, *Nokia*, merilis *handheld Nokia N-Gage* di tahun 2003. Ini merupakan ponsel sekaligus piranti game yang lengkap dengan fitur-fitur *multimedia* dan *interkonektivitas*, seperti *Bluetooth*. Dan juga pada tahun ini, dirilis *Game Boy Advance SP* dengan model yang menarik, solid, dan padat. Akhir 2004, Sony merilis *handheld* pertama yang menggunakan cakram bernama PSP dan dibarengi

dengan hadirnya Nintendo DS, yang menggunakan konsep *dual screen* (layar ganda). Disusul oleh *Game Boy Micro* dan *Game Park XGP* yang rilis sepanjang 2005. dan akhirnya tahun 2006 lalu, Nintendo DS Lite dan Pelican *VG Pocket Caplet* menjadi *handheld* terbaru yang dilempar ke pasaran.



Gambar 2.13 Meski kalah jumawa dari *Nintendo DS, Sony PSP* telah menetapkan standar baru teknologi *handheld* di masa depan.
(sumber : http://www.jagogame.com)

2.2.2. Perkembangan Game Komputer

Permainan komputer adalah program komputer yang terdiri dari dunia maya yang dikendalikan oleh sebuah komputer di mana pemainnya bisa berinteraksi untuk mencapai sejumlah tujuan (*goal*).

Permainan komputer juga dapat digolongkan ke dalam beberapa aliran/genre:

a. Laga (Action)
b. Petualangan (Adventure)
c. Bermain Peran (Role-Playing (RPG))
d. Simulasi (Simulation)
e. Olahraga (Sports)

Akhir-akhir ini berkembang pemikiran bahwa permainan komputer juga merupakan suatu bentuk kesenian seperti halnya film, hanya medianya lebih bersifat interaktif.

Perkembangan industri video dan komputer game dimulai dari sebuah hobi pada tahun 1970-an ketika Komputer Personal (PC) baru saja tersedia luas, seperti yang telah dipaparkan pada generasi ke dua di sejarah game. Industri ini terus berkembang sesuai dengan kemajuan teknologi komputer, yang terkadang menjadi sebuah teknologi itu sendiri.

Saat ini Industri game PC menjadi sebuah bisnis yang mengguntungkan, yang sebelumnya membuat industri game sebagai industri yang mahal dengan pengeluaran biaya yang cukup besar dengan hasil tidak berimbang sekarang sebuah game bisa dibuat oleh seorang programmer yang disebut game programmer atau oleh sebuah team kecil yang terdiri dari beberapa programmer dan beberapa seniman dapat menjual ribuan kopi untuk tiap game yang elah dibuat. Karena pembuatan sebuah game bisa diselesaikan dalam hitungan bulan, jadi dapat memproduksi beberapa judul tiap tahunnya. Serta royalti yang didapat membuat industri ini terus banyak diminati dan terus berkembang.

2.3. Game Engine

Game Engine, game engine disebut juga komponen inti dari sebuah game atau aplikasi interaktif lainnya yang disajikan secara real time. Game engine memberikan teknologi dasar, mempermudah pengembangan sebuah game dan

bahkan memberikan teknologi yang dapat dijalankan dengan platform yang berdeda-beda, seperti game console, Operasi sistem berbasis desktop Linux, Max OS, dan Microsoft Windows. Fungsi-fungsi utama yang biasanya disediakan oleh game engine termasuk didalamnya rendering engine (renderer) untuk grafik 2D atau 3D, physic engine atau deteksi benturan (collation detection), suara, scripting, animation, AI (Artificial Intelligence), networking dan susunan adegan/kejadian/film pendek (scene graph).

2.3.1. Perkembangan Game Engine

Istilah game engine berkembang pada era tahun 1980-an, yang secara khusus sering dihubungkan dengan Game 3D seperti First Person Shoteers (FPS). Generasi awal FPS Grafik Engine, kebutuhan dan sifatnya sangat bermacam-macam , tetapi maksudnya mengarah ke proses pe-render-an secara sudut pandang orang pertama (first-person perspective) dan dengan bertujuan untuk menembak sesuatu, kebanyakan dibuat melalui engine berbasis grafik vector (Vector graphics engines), contoh-contoh gamenya adalah :

a. Maze War (1973)
b. Battlezone (1980)
c. MIDI Maze (1987)

2.3.1.1 Generasi I (Dari WireFrame ke Dunia 3D dan Texture)

Dunia yang rata (*Planar Worlds*) (jaringan persegi ; rectangular grid pada game *Wolfenstein 3D*, bidang beringakat berbasis sektor (sector-based plane levels) pada game *Doom).*

Kebutuhan rata-ratanya hardware video-nya dapat pemprosessan yang dilakukan pada sisi software (powered software rendering). Enginenya dibentuk dari beberapa *sprites* (gambar) untuk semua visualisasinya, tetapi bisa menerapkan tingkatan geometri 3D (3D-level geometry). Contohnya :

1. Driller (1987) Freescape engine
2. Castle Master (1990)
3. Hovertank 3D (1990) (PC)
4. Catacomb 3D (1991) (PC)
5. Ultima Underworld (1992)
6. Ultima Underworld II (1993)
7. System Shock (1994)
8. Wolfenstein 3D (1992)
9. Rise of the Triad (1994)
10. Doom (1993) Doom engine
11. Doom II (1994)
12. Marathon (1994) (Mac)
13. Star Wars: Dark Forces (1995) (PC)
14. Duke Nukem 3D (1996) Build engine (PC)
15. Blood (1997)
16. Shadow Warrior (1997)

Awal generasi pertama dimulai dari Wolfenstein 3D (1992).

2.3.1.2 Generasi IIa (Bangkitnya pemodelan 3D)

Untuk pertama kali, game engine dibuat kembali dengan dunai 3D yang sebenarnya dengan tingkatan geometri yang beragam. Yang dikenal sebagai generasi ke dua. Bahkan *sprites* pada enginenya menggunakan objek poligon ter-texture (*textured polygon objects*) yang sederhana (*single pass texturing,* tanpa detail cahaya). Kebutuhan rata-ratanya hardware video-nya dengan tiga demensi accelerator pertama (digunakan pada VGA Voodoo, Voodoo 2) , game Quake menggunakan animasi sprites yang lebih sedikit, mengikuti trend tiga demensi dibandingkan dengan object dua dimensi.

1. Quake (1996) Quake engine (PC)
    a. Half-Life (1998) GoldSrc (PC)
2. GoldenEye 007 (1997) (N64)
    a. Perfect Dark (2000)
3. Quake II (1997) Quake II engine
    a. Sin (1998)
    b. Anachronox (2001)
    c. Soldier of Fortune (2000) (PC)
4. Shogo: Mobile Armor Division (1998) Lithtech 1.0 (PC)
5. No One Lives Forever (2000) Lithtech 1.5: Talon (PC)
    a. Alien vs. Predator 2 (2001) (PC)

6. Unreal (1998) Unreal Engine (PC)
    a. Unreal Tournament (1999) (PC)
    b. Deus Ex (2000)
    c. Clive Barker's Undying (2001)
7. Thief: The Dark Project (1998) Dark Engine
    a. System Shock 2 (1999)
    b. Thief II: The Metal Age (2000)
8. Quake III Arena (1999) Quake III engine
    a. Return to Castle Wolfenstein (2001)
    b. Call of Duty (2003) (PC)
    c. Star Wars Jedi Knight II: Jedi Outcast (2002)
    d. Star Wars Jedi Knight: Jedi Academy (2003)

2.3.1.3 Generasi IIb (Kemampuan baru dan peningkatan detail)

Perangakat keras yang baru menyediakan kemampuan yang baru pula, menyediakan game engine yang baru untuk menambah variasi effect yang baru, seperti particle effects, fog, coloured lighting dan juga menambah detail dari sebuah texture dan poligon. Banyak game yang memerikan lingkungan luar (out door) yang luas, kendaraan, rag-doll physics. Kebutuhan rata-ratanya hardware video-nya ada pada *GeForce 2* atau sejenisnya.

1. Codename Eagle (2000) Refractor engine (PC)
2. Battlefield 1942 (2002) Refractor 2 engine (PC)
3. Battlefield Vietnam (2004) (PC)
4. Max Payne (2001) Remedy MaxFX
    a. Max Payne 2 (2003)
5. Grand Theft Auto III (2001) RenderWare engine
    a. Grand Theft Auto: Vice City (2002)
    b. Grand Theft Auto: San Andreas (2004)
6. Halo: Combat Evolved (2001) (XBOX/PC)
    a. Halo 2 (2004) (XBOX)
    b. Stubbs the Zombie in Rebel Without a Pulse (2005)
7. Mafia (2002) LS3D engine
    a. Vietcong (2003)
    b. Hidden & Dangerous 2 (2004)
8. Unreal Tournament 2003 (2002) Unreal Engine 2.0
    a. Unreal II: The Awakening (2002)
    b. Tom Clancy's Rainbow Six 3: Raven Shield (2003)
9. No One Lives Forever 2 (2002) Lithtech 2.0: Jupiter
    a. Tron 2.0 (2003)
10. Global Operations (2002) Lithtech 2.5
11. Metroid Prime (2002)
    a. Metroid Prime 2: Echoes (2004)
12. Cube: Game and engine (2002-2005)

2.3.1.4 Generasi III (Pendekatan Foto Realistis)

Pada Generasi ke tiga pengembangan era dari 3D engine lebih mengarah ke peningkatan kualitas foto realistis. Semua peta dapat memberikan lingkungan luar dan dalam yang tidak berlapis (menjadi satu kesatuan/tidak terpisah). Beberapa atau semua telah berbasis pixel shader-based texture,bump mapping, vertex sheader untuk animasinya, teknologi pencahayaan dan bayangan secara umum. Kebutuhan rata-ratanya hardware video-nya ada pada *GeForce 3* atau sejenisnya yang menyediakan shader effect.

1. AMP2 Tech Demo (2003) AMP2 Game Engine
2. Tribes Vengeance (2004) Unreal Engine 2.5
3. Painkiller (2004) PAIN engine
4. Far Cry (2004) CryENGINE
5. Doom 3 (2004) Doom 3 engine
    a. Quake 4 (2005)
6. Half-Life 2 (2004) Source engine
    a. Vampire: The Masquerade - Bloodlines (2004)
    b. Dark Messiah of Might and Magic (2006)
    c. Sin Episodes - Emergence (2006)
7. Battlefield 2 (2005) (PC)
8. F.E.A.R. (2005) Lithtech: Jupiter EX
9. Call of Duty 2 (2005)
10. Call of Duty 3 (2006)

11. Red Steel (2006)
12. Paragraf-78 (2006*) Dagor Engine 3
13. Metroid Prime 3: Corruption (2007*)
14. Crysis (2007*)
15. Halo 3 (2007*)
16. S.T.A.L.K.E.R.: Shadow of Chernobyl (2007*) X-Ray engine
17. Operation Flashpoint 2 (2007*)

Tanda bintang * menandakan belum di release. Ada pun tanggal releasenya masih perkiraan.

2.3.1.5 Generasi IV

Generasi ke empat belum ada saat ini, tapi kunci utamanya adalah sudah bisa menggunakan teknik ray tracing, dan teknik rendernya berhungungan dengan resterization, yang sedang digunakan saat ini.

2.3.2. Komponen Game Engine

Dalam perancangan sebuah game engine dibutuhkan komponen dasar yang wajib di implemtasikan. Kunci utama dalam pembuatan game adalah *game loop*, yang harus melanjutkan ke operasi tanpa menghiraukan inputan dari user. Game loop menyediakan proses ini. Contoh sederhana dari game loop secara *pseudocode* bisa seperti berikut :

```
while( user doesn't exit )
    check for user input
    run AI
    move enemies
    resolve collisions
    draw graphics
    play sounds
end while
```

Prosess Game loop bisa dijalankan secara multi thread, jadi semua proses dapat beralan masing-masing tanpa terganggu oleh proses lainnya sehingga kerugian akibat lambatnya proses dapat ditekan. Sebuah game dapat berjalan lebih halus lagi dan tentunya effisien jika menggunakan teknologi hyperthreading atau multicore processor, dengan demikian lebih banyak proses yang dikerjakan.Console seperti *Xbox 360* dan *Playstation 3* telah menerapkan teknologi multicore dan tentunya multi thread.

Disamping itu juga proses yang harus dimplementasikan agar sebuah game dapat berjalan atau bekerja.

1. Inisialisasi Device
2. Rendering
    a. Model Representation
    b. Camera
    c. Rendering
3. Drawing
    a. Vertex/Index Buffer
    b. Render State
    c. Persiapan Drawing

   d. Drawing dengan vertex/index buffer
   e. Object Geometri
4. Coloring
   a. Gambaran umum coloring
   b. Vertex Color
   c. Shading
5. Lighting
   a. Lighting Component
   b. Material
   c. Vertex Normals
   d. Light Sources
6. Texturing
   a. Texture coordinates
   b. Creating and Enabling texture
   c. Filter
   d. Mipmaps
   e. Address Modes
7. Blending
   a. Perumusan Bleding
   b. Blend Factor
   c. Transparancy
8. Stenciling
   a. Stencil Buffer

Semua proses diatas adalah komponen dari sebuah grafik engine, saat ini sebuah game terdiri dari beberapa engine terpisah untuk digabungkan menjadi satu kesatuan game utuh, yang diantaranya adalah :

1. Input Engine

Sebuah program yang menangani semua masukan yang didapat dari user, dapat berupa keyboard, mouse, joystick dan alat input lainnya. Program ini juga harus bisa menangani inputan yang hilang/terlepas ketika program sedang berjalan atau adanya inputan baru yang telah diinstall pada komputer user.

Input engine juga harus mampu menganggapi respon yang diberikan oleh user secara cepat dan akurat.

2. Graphic Engine

Sebuah program yang mengangani tampilan yang akan diimplentasikan pada sebuah alat keluaran yang biasanya Monitor atau bahkan pada sebuah Virtual Reality (VR) device. Graphic Engine memegang peranan yang paling penting dalam sebuah game karena dengan Graphic Engine kualitas, optimasi, interface sampai dengan special effect dapat di implementasikan pada Graphic Engine, yang tentunya dengan sebuah Graphic Engine yang bagus dan optimal dapat menambah daya tarik pada semua game. Sekarang produsen dan game devolepment berlomba-lomba memberikan kualitas yang baik dengan menggunakan teknologi terbaru pada design grafisnya sehingga user merasa di manjakan

dengan kualitas grafis yang “cantik” yang sebisa mungkin mendekati realistis.



Gambar 2.14 Contoh hasil render pada Directx 9
(sumber : http://www.winmatrix.com/forums/index.php?showtopic=9550)



Gambar 2.15 Contoh hasil render pada Directx 10
(sumber : http://www.winmatrix.com/forums/index.php?showtopic=9550)

Pada gambar 2.14 dan gambar 2.15 bisa dilihat perbedaan hasil render pada directx 9 (gambar 2.14) dengan directx 10 (gambar 2.15) . Perkembangan teknologi pixel dan vertex shader yang

cukup maju pada directx 10 membuat hasil render in game (bukan still image) , nyaris mendekati seperti aslinya.



Gambar 2.16 Directx 9
(sumber : http://www.winmatrix.com/forums/index.php?showtopic=9550)



Gambar 2.17 Directx 10
(sumber : http://www.winmatrix.com/forums/index.php?showtopic=9550)

Dan dilanjutkan dengan perbedaan yang cukup jauh pada gambar 2.14 dengan gambar 2.15 yang menampilkan detail yang sangat tinggi dengan kualitas yang nyaris sempurna. Dengan

perkembangan texture, lighting dan shader saat ini yang nyaris sempurna. Untuk bisa direnderer seperti pada gambar 2.14 dan gambar 2.15 dalam hal ini directx 10 maka dari sisi hardware dan teknik optimasi merupakan hal yang paling penting, tetapi nilai positifnya game masa datang akan lebih realistis dengan kehidupan nyata.

3. Sound Engine

Sebuah program yang menangani semua keluaran berupa suara dari suara suara latar sampai sound effect yang menambah tingkat realistis pada game. Game engine sekarang telah mendukung teknologi suara seperti THX, Doubly Surround dan lainya yang dipakai pada film-film sekelas hollywood. Beberapa artis dunia telah menjadikan game sebagai salah satu media promosi album mereka dengan menjadikan lagu-lagu mereka sebagai lagu latar (theme song) pada game-game tertentu. Dukungan terhadap format-format kompresi suara juga dapat diemplementasikan contohnya : wav, mp3, ogg, midi dan lainnya.

4. Network Engine

Sebuah program yang menangani semua jalur komunikasi diluar komputer user. Multiuser Game, Massive Multiplayer Game, LAN Game, dan Online Game adalah implentasi dari network engine. Hal security jaringan juga merupakan hal yang paling penting pada network engine.

Network Engine dibuat agar beberapa user dapat berinteraksi satu sama lain secara real time yang dapat menjadi nilai tambah pada semua game. Di Indonesia trend game online juga sedang menjamur karena dalam sebuah game online di Indonesia memenuhi sifat sosial. User bisa chating, bertualang bersama, treading (jual beli) dengan network engine diharapkan bisa implentasikan pada game yang akan kita buat nantinya.

5. Physics Engine

Physics Engine merupakan hal yang cukup baru pada teknolgi game saat ini, dimana Physics Engine lebih menambah lagi realistis sebuah game karena mampu menerapkan ilmu Fisika pada effect ledakan, reruntuhan yang berterbangan, cairan yang mengalir percikan air dan pergerakan objek yang tertiup angin. Hal ini berarti tidak ada lagi animasi dan interaksi yang kaku. [10] Karena Physics Engine menjadi hal yang sangat penting dalam pembuatan game maka khusus buat Physics Engine sekarang telah diciptakan grafik card khusus Physics yang digabungkan kinerjanya dengan Grafik Card.

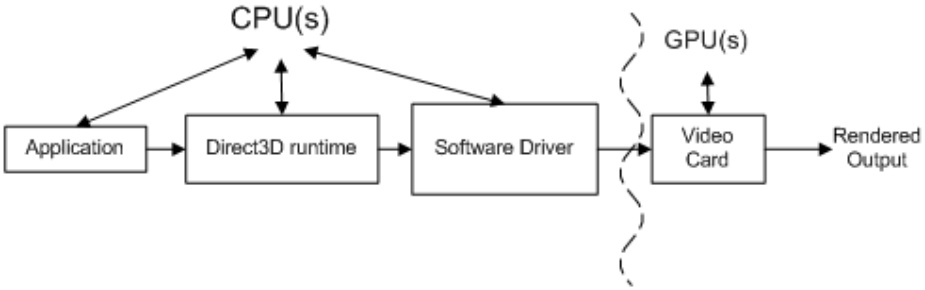
Game engine atau 3D engine atau rendering system biasanya dibangun melalui sebuah Application Programming Interfaces (API) grafik seperti Direct3D atau OpenGL yang menyediakan sebuah abtraksi dari sebuah Graphic Prosessing Unit (GPU) atau Video Card. Kepustakaan tingkat

rendah (low-level library) seperti DirectX, SDL (Simple DirectMedia Layer) dan OpenAL juga biasa digunakan pada pembuatan game yang menyediakan akses hardware secara independent ke perangkat komputer lainnya seperti alat masukan (input device) (Mouse, Keyboard dan joystik), kartu jaringan (network card), dan kartu suara (sound card).



Gambar 2.18 diagram konseptual komponen yang harus dilalui untuk melakukan proses render.

Sebelum sebuah perangkat meningkatkan kinerja grafis (hardware-accelerated), proses render secara hardware telah digunakan terlebih dahulu, proses render secara hardware masih digunakan pada beberapa perangkat pemodelan (modeling tools) atau untuk digunakan pada perendaran gambar diam (still-rendered image) dimana akurasi tampilan yang dihasilkan melebihi nilai hasil secara real time (frame per second;FPS) atau pada saat perangakat keras tidak memenuhi kebutuhan seperti dukungan terhadap teknologi shader.

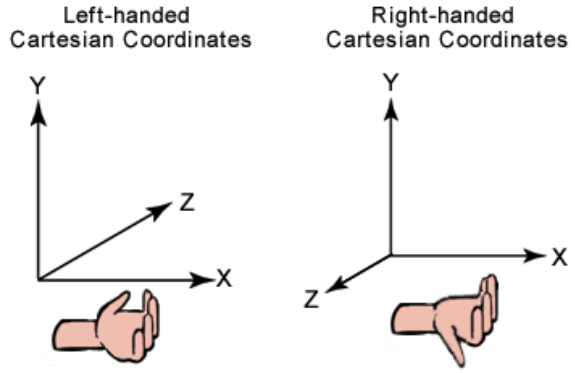


Gambar 2.19 Dua jenis pengolahan central processing unit (CPU) dan graphic processing unit (GPU).

Metode yang gunakan dalam perancangan sebuah game engine menggunakan metode perhitungan Geometry dan perhitungan Aljabar, yang digunakan dalam menghitung dan memproses Transformasi Grafik 3D.

1. Vektor

Biasanya penerapan Grafik 3D menggunakan dua jenis system koordinat *Cartesian*, left-handed dan right-handed. Pada kedua jenis sistem tersebut sumbu x dan y adalah sama, sumbu x berada di sebelah kanan dan sumbu y berada di atas, dengan mengingat posisi jari pada kedua tangan, posisi jempol sebagai sumbu z-nya.



Gambar 2.20 left-handed dan right-handed pada sistem koordinat Cartesian.

Secara geometri, dua vektor dikatakan sama apabila kedua titiknya mempunyai arah yang sama dan mempunyai panjang yang sama. Secara aljabar, dua vektor dikatakan sama apabila mempunyai dimensi yang sama dan komponen yang sesuai.

$$(u_x,u_y,u_z) = (v_x,v_y,v_z)$$

dimana

$$u_x = v_x \text{ , } u_y = v_y \text{ dan } u_z = v_z$$

a. Menghitung besaran vektor

Secara geometri, besaran dari sebuah vektor adalah panjang dari arah segment garis, dan secara aljabar untuk menghitung besaran vektor adalah:

$$\|u\| = \sqrt{u_x^2 + u_y^2 + u_z^2}$$

b. Normalisasi vektor

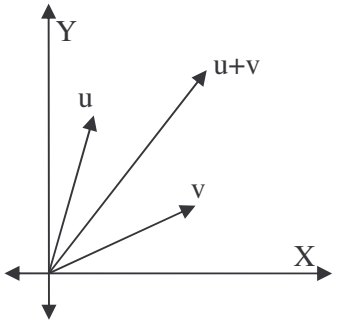
Dengan menormalisasikan sebuah vektor membuat basaran vektor menjadi 0, dengan membagi tiap komponennya dengan besaran vektornya.

$$\hat{u} = \frac{u}{\|u\|} = \left(\frac{u_x}{\|u\|}, \frac{u_y}{\|u\|}, \frac{u_z}{\|u\|} \right)$$

c. Penjumlahan Vektor

Penjumlahan vektor dapat dilakukan dengan syarat dimensinya harus sama.

$$u + v = (u_x + v_x \text{ , } u_y + v_y \text{ , } u_z + v_z)$$

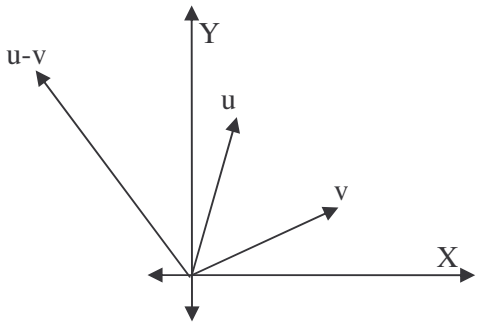


Gambar 2.21 Penjumlahan vektor

d. Pengurangan Vektor

Pengurangan vektor dapat dilakukan dengan syarat dimensinya harus sama.

$$u - v = u + (-v) = (u_x - v_x , u_y - v_y , u_z - v_z)$$



Gambar 2.22 Pengurangan vektor

e. Scalar Multiplication

Vektor dapat dikalikan dengan bentukan scalar, operasi ini tidak menghasilkan arah vektor baru, kecuali dengan merubah skala vektor dengan bilangan negatif yang menghasilkan arah terbalik.

$$ku = (ku_x , ku_y , ku_z)$$

contohnya :

D3DVECTOR3 u(1.0f , 1.0f , -1.0f);

D3DVECTOR3 scaledVec = u * 10.0f ;

Maka, variabel scaledVec akan menghasilkan nilai

u(10.0f , 10.0f , -10.0f);

f. Dot Product

Dot product, hasil dari perkalian (dot ; notasinya ".") vektor adalah salah satu product (hasil) hari perhitungan aljabar.

$$u \cdot v = (u_x v_x , u_y v_y , u_z v_z) = s$$

rumusan diatas belum dapat digunakan untuk merepresentasikan kedalam geometri sebenarnya. Dengan menggunakan hukum cosinus, didapat relasinya.

$$u \cdot v = \|u\| \|v\| \cos \theta$$

dan,

I. jika u . v = 0 maka u tegal lurus terhadap v.

II. jika u . v > 0 maka ɵ berada diantara dua vektor dengan sudut kurang dari 90º.

III. jika u . v < 0 maka ɵ berada diantara dua vektor dengan sudut lebih dari 90º.

g. Cross Product

Dengan melakukan perhitungan cross product dari dua buah vektor maka dihasilkan sebuah vektor baru (p) yang tegak lurus dengan vektor u dan v

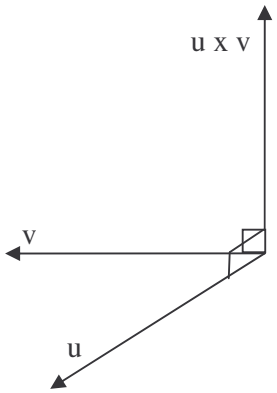
$p = u \times v = [(u_y v_z - u_z v_y), (u_z v_x - u_x v_z), (u_x v_y - u_y v_z)]$

dimana,

$p_x = (u_y v_z - u_z v_y)$

$p_y = (u_z v_x - u_x v_z)$

$p_z = (u_x v_y - u_y v_z)$



Gambar 2.23 Cross Product

2. Matriks

Perhitungan matriks dalam transformasi grafik 3D menggunakan matriks 4x4, lalu digunakan matriks 1x4 untuk meletakan koordinat sebuah titik atau komponen dari sebuah vektor (v). Hasil dari vX adalah hasil vektor baru yang telah ditransformasikan (v'), contoh : Jika X direpresentasikan 10 unit translasi pada sumbu x dan v = [2,6,-

3,1] maka hasil vX = v' = [12,6,-3,1], dimana [x,y,z,w]. Kenapa menggunakan matriks 4x4 ? sekilas sekilas matriks 3x3 cocok, tetapi dengan menggunakan matriks 3x3 tidak semua perhitungan tipe transformasi yang diimplementasikan, contohnya translasi, proyeksi perspective (*perspective projection*) dan refleksi, serta karena perhitungannya menggunakan product vector-matriks maka terdapat batasan aturan dalam perkalian matriks untuk melakukan transformasi. Lalu kenapa menggunakan 1x4 untuk meletakan koordinat sebuah titik atau komponen dari sebuah vektor? karena hasil dari 1x3 (cross) pada baris vecktor dan hasil dari 4x4 pada baris matriks hasilnya tidak dapat ditentukan. Maka dari itu dipakailah komponen tambahan yaitu "w" yang merepresentasikan titik (*points*) dimana w=0 untuk vektor dan w=1 untuk points, karena vektor tidak mempunyai arah maka translasi dari sebuah vektor tidak dapat ditentukan. Untuk mencegahnya maka w pada vektor harus di ubah menjadi w=0. Jadi, P = (P_1,P_2,P_3) dapat diletakan pada sebuah matriks menjadi P = [P_1,P_2,P_3,1] dan untuk vector V=(V_1,V_2,V_3) menjadi V=[V_1,V_2,V_3,0]. Terkadang dari sebuah transformasi matriks dapat dihasilkan perubahan nilai w dari sebuah vektor dimana $w \neq 0$ dan $w \neq 1$. Yang berdasarkan dari :

$$P = [P_1,P_2,P_3,1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = [P_1,P_2,P_3, P_3] = P'$$

untuk $P_3 \neq 0$ dan $P_3 \neq 1$

Didapat w = P_3, dimana w ≠ 0 dan w ≠ 1, yang dapat disebut vektor pada jarak yang semua sama (homogeneous space) yang berlawanan dengan vektor 3-space (berjarak 3). Homogeneous space dapat menghasilkan vektor tiga demensi dengan membagi tiap komponennya dengan w yang biasa digunakan untuk menghitung proyeksi perspective.

$$\left(\frac{x}{w},\frac{y}{w},\frac{z}{w},\frac{w}{w}\right)=\left(\frac{x}{w},\frac{y}{w},\frac{z}{w},1\right)=\left(\frac{x}{w},\frac{y}{w},\frac{z}{w}\right)=x$$

Matriks Transformasi dapat dirumuskan sebagai berikut :

$$[\mathrm{x'},\ \mathrm{y'},\ \mathrm{z'},1]=[x,y,z,1]\begin{bmatrix}M_{11} & M_{12} & M_{13} & M_{14}\\ M_{21} & M_{22} & M_{23} & M_{24}\\ M_{31} & M_{32} & M_{33} & M_{34}\\ M_{41} & M_{42} & M_{43} & M_{44}\end{bmatrix}$$

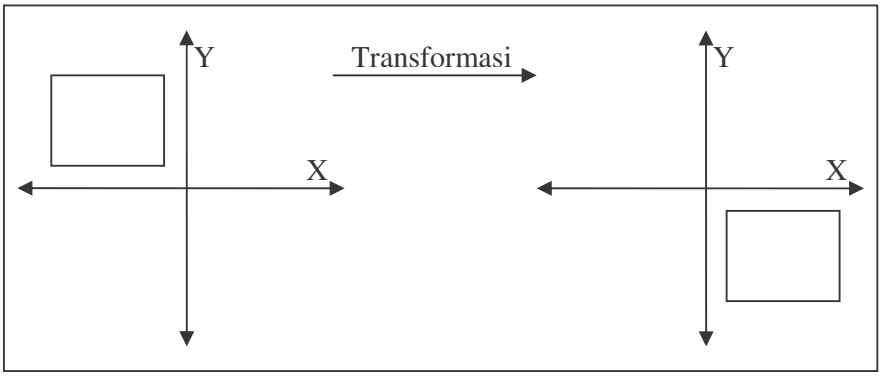
Dimana operasinya adalah:

$$\mathrm{x'}=(\mathrm{x}\times M_{11})+(\mathrm{y}\times M_{21})+(\mathrm{z}\times M_{31})+(1\times M_{41})$$
$$\mathrm{y'}=(\mathrm{x}\times M_{12})+(\mathrm{y}\times M_{22})+(\mathrm{z}\times M_{32})+(1\times M_{42})$$
$$\mathrm{z'}=(\mathrm{x}\times M_{13})+(\mathrm{y}\times M_{23})+(\mathrm{z}\times M_{33})+(1\times M_{43})$$

Transformasi pada Grafik 3D meliputi :

a. Matriks Translasi

Dibawah ini adalah Matrik Translasi dari (x,y,z) ke (x', y', z')

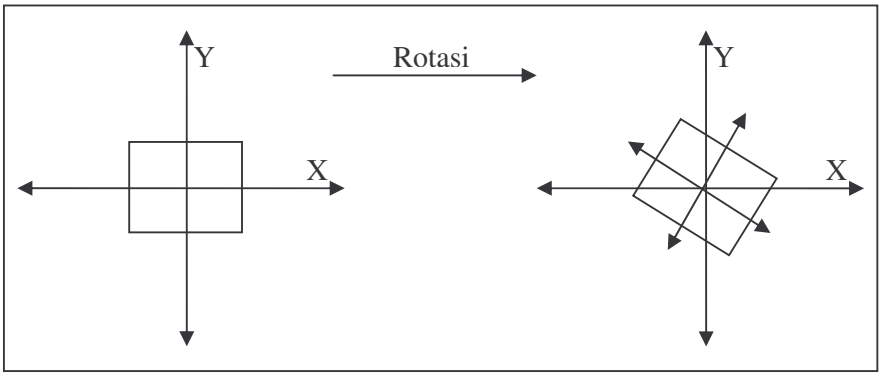


Gambar 2.24 Transformasi

$$T(P) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ P_x & P_y & P_z & 1 \end{bmatrix}$$

b. Matriks Rotasi

Dibawah ini adalah Matrik Rotasi dari (x,y,z) ke (x', y', z')



Gambar 2.25 Rotasi

Rotasi pada sumbu x

$$X(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotasi pada sumbu y

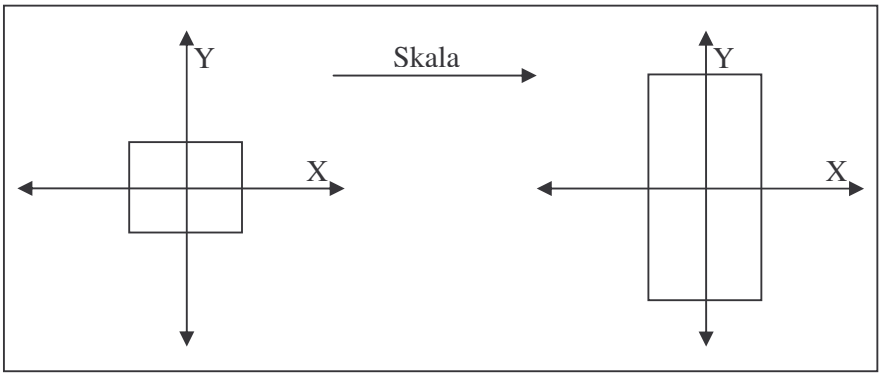
$$Y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotasi pada sumbu z

$$Z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

c. Matriks Skala

Dibawah ini adalah Matrik Skala dari (x,y,z) ke (x', y', z')



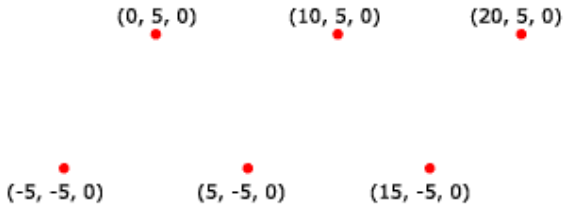
Gambar 2.26 Skala

$$S(q) = \begin{bmatrix} q & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & q & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Primitives

a. Point Lists

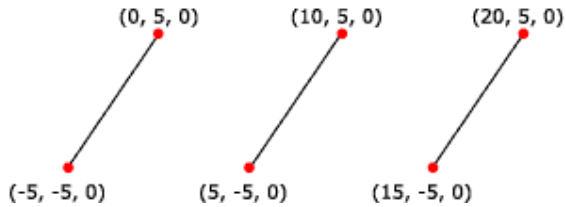
Point lists adalah kumpulan dari titik-titik yang direnderer menjadi satu kesatuan.



Gambar 2.27 Point lists

b. Line Lists

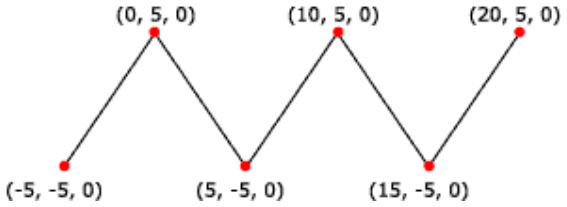
Line list adalah kumpulan titik-titik yang direnderer menjadi sebuah garis, yang minimal terdiri dari dua titik.



Gambar 2.28 Line lists

c. Line Strips

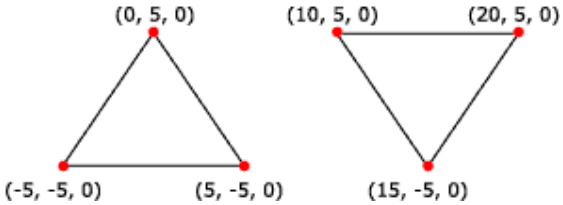
Line Strips adalah kumpulan titik-titik yang direnderer menjadi garis yang terhubung menjadi tadi sebuah segment berbentu garis, biasa digunakan untuk membuat poligon yang tidak tertutup.



Gambar 2.29 Line Strips

d. Triangle Lists

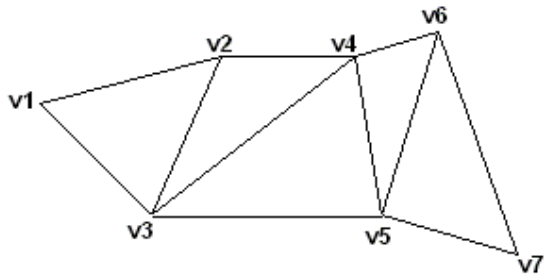
Triangle lists adalah kumpulan titik-titik yang direnderer menjadi sebuah setiga yang bisa berdekatan atau tidak dengan segitiga lainnya. Triangle lists minimal membutuhkan tiga titik dan totalnya harus bisa dibagi tiga.



Gambar 2.30 Triangle lists

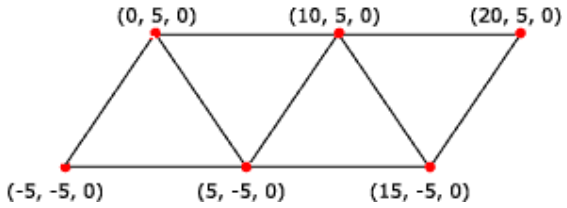
e. Triangle Strips

Triangle strips adalah kumpulan dari setiga yang direnderer menjadi satu kesatuan, kerena semua segitiga akan terhubung maka tidak perlu menyebutkan satu-satu titik pendukunganya. Contoh untuk membuat sebuah Triangle strips dibawah ini cukup dibutuhkan tujuh titik.



Gambar 2.31 Contoh pengguaan Triangle strips

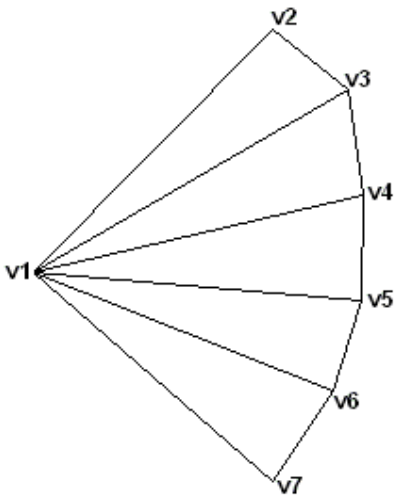
Titik yang digunakan adalah : segitiga pertama terdiri dari v1,v2,v3 untuk segitiga yang kedua terdiri dari v2,v3,v4 untuk yang ketiga terdiri dari v3,v4,v5 dan seterusnya.



Gambar 2.32 Triangle strips

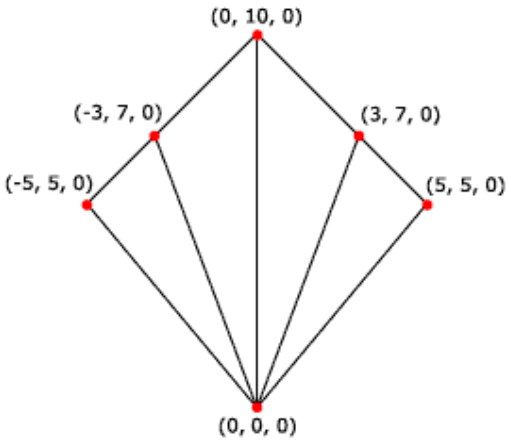
f. Triangle Fans

Triangle fans sama dengan Triangle strips, kecuali semua titik bersumber pada satu titik.



Gambar 2.33 Contoh pengguaan Triangle fans

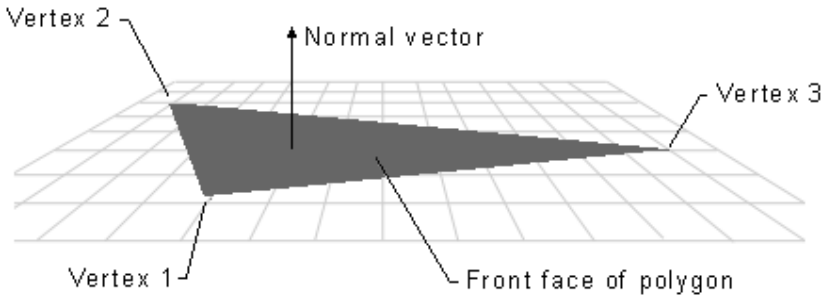
Titik yang digunakan adalah : segitiga pertama terdiri dari v2,v3,v1 untuk segitiga yang kedua terdiri dari v3,v4,v1 untuk yang ketiga terdiri dari v4,v5,v1 dan seterusnya.



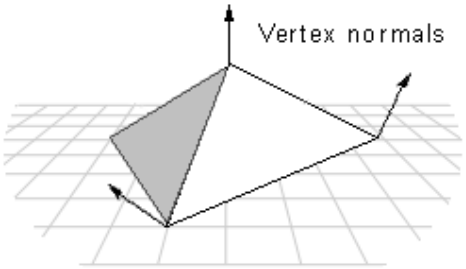
Gambar 2.34 Triangle fans

4. Face dan vertex normal vektor

Setiap face pada setiap mesh/object tegak lurus dengan normal vektornya.



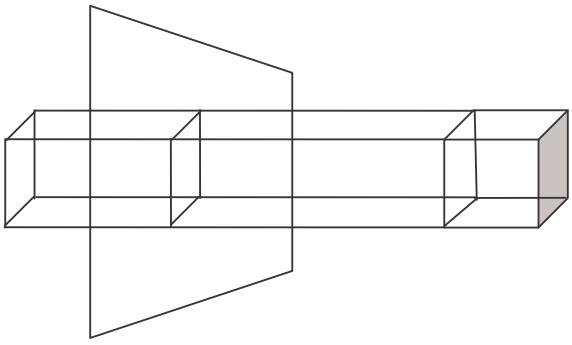
Gambar 2.35 Face dan vertex normal vektor
(sumber : DirectX SDK,november 2006, Accurately Profiling Direct3D API Calls)



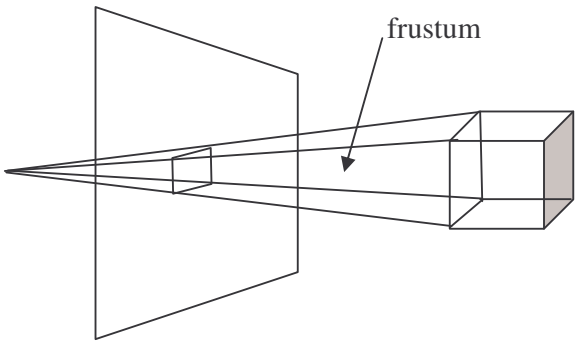
Gambar 2.36 vertex normals
(sumber : DirectX SDK,november 2006, Accurately Profiling Direct3D API Calls)

5. Projection

Proyeksi pada sistem tiga dimensi terdiri dari dua jenis, yaitu orthogonal projection dan perspective projection.



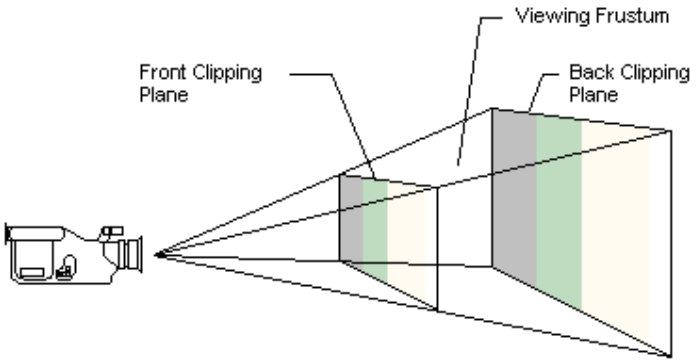
Gambar 2.37 orthogonal projection



Gambar 2.38 perspective projection

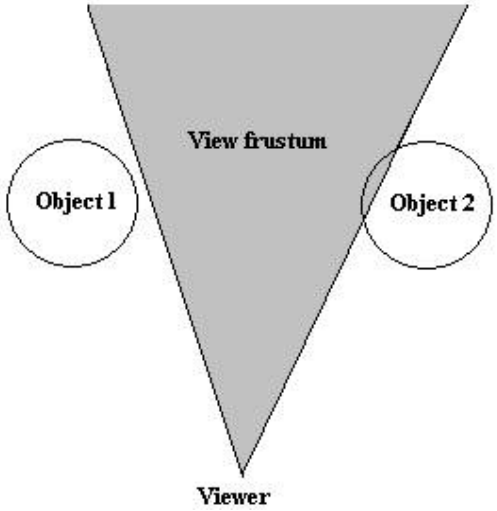
6. Clipping

Clipping terdiri dari dua jenis, yaitu front clipping dan back clipping, ruang diantaranya disebut frustum. Hanya object yang berada di area frustum akan ditampilkan



Gambar 2.39 Clipping

Adalah culling digunakan untuk menghemat FPS (Frame per second) yang bila terlalu rendah akan membuat kinerja game melambat, dan untuk menghapus (culling) poligon-poligon tertentu yang terdapat diluar area kamera (frustum).



Gambar 2.40 Frustum Culling

2.4. JAVA

Java merupakan *development tools* yang fleksibel dan *powerfull*. Salah satu keunggulannya adalah *platform independence* dimana artinya program yang ditulis tidak tergantung pada sistem operasi pada saat program dibuat. Java dibangun dengan kekuatan C++ dengan mengambil fitur-fitur terbaiknya dan menghilangkan permasalahan yang ada. Java memiliki fitur *garbage collection* (manajemen memory secara otomatis), *multithreading* dan keamanan yang baik.

2.4.1. Sejarah Java

Bahasa pemrograman Java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan *Oak*. Proyek ini dimotori oleh *Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy*, beserta sembilan pemrogram lainnya dari *Sun Microsystems*.

Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh *Joe Palrang*.

Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road* di *Menlo Park*. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (touch screen), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (Star Seven).

Setelah era Star Seven selesai, sebuah anak perusahaan TV kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di *100 Hamilton Avenue, Palo Alto*.

Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Mereka menjadikan perambah (browser) *Mosaic* sebagai landasan awal untuk membuat perambah Java pertama yang dinamai *Web Runner,* terinsipirasi dari film 1980-an, *Blade Runner*. Pada perkembangan rilis pertama, *Web Runner* berganti nama menjadi *Hot Java.*

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk

pemeritaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995.

Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel *Sheraton Palace*. Tiga dari pimpinan utama proyek, *Eric Schmidt* dan *George Paolini* dari *Sun Microsystems* bersama *Marc Andreessen*, membentuk *Netscape*.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "bapak java", *James Gosling*. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling.

Versi awal Java ditahun 1996 sudah merupakan versi release sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

1. java.lang: Peruntukan kelas elemen-elemen dasar.
2. java.io: Peruntukan kelas input dan output, termasuk penggunaan file.
3. java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
4. java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.

5. java.awt: Kelas dasar untuk aplikasi antar muka dengan pengguna (GUI)
6. java.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada web browser.

2.4.2. Kelebihan dan kekurangan Java

Kelebihan Java

a. **Multiplatform**. Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip write once, run every where. Dengan kelebihan ini pemrogram cukup menulis sebuah program java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris.

b. **OOP**, singkatan dari *Object Oriented Programming* yang artinya semua aspek yang terdapat di Java adalah objek. Java merupakan salah satu bahasa pemrograman berbasis object murni. Semua tipe data diturunkan dari kelas dasar Object. Hal

ini sangat memudahkan programmer untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan java sebagai salah satu bahasa pemograman termudah, bahkan untuk fungsi fungsi yang advance seperti komunikasi antara komputer sekalipun.

c. **Library Kelas Sangat Lengkap**, Java terkenal dengan kelengkapan library (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para programmer untuk membangun aplikasinya. Kelengkapan library ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat library-library baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

d. **Bergaya C++,** memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak programmer C++ untuk pindah ke Java. Saat ini pengguna java sangat banyak, sebagian besar adalah programmer C++ yang pindah ke Java.

e. **Garbage Collection,** memiliki fasilitas pengaturan penggunaan memori sehingga programmer tidak perlu melakukan pengaturan memori secara langsung.

Kekurangan Java

a. **Write once, debug everywhere** - Ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

b. **Mudah didekompilasi**. Dekompilasi adalah proses membalikkan dari executable code menjadi source code. Ini dimungkinkan karena executable Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, method, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan.

c. **Heavy memory usage**. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus mamakai mesin desktop berumur lebih dari 4 tahun.

Untuk mengembangkan aplikasi berbasis java, maka diperlukan Java Develompent Kit (JDK) yang terbagi menjadi tiga ketegori, yaitu :

a. Java 2 Standard Edition (J2SE)

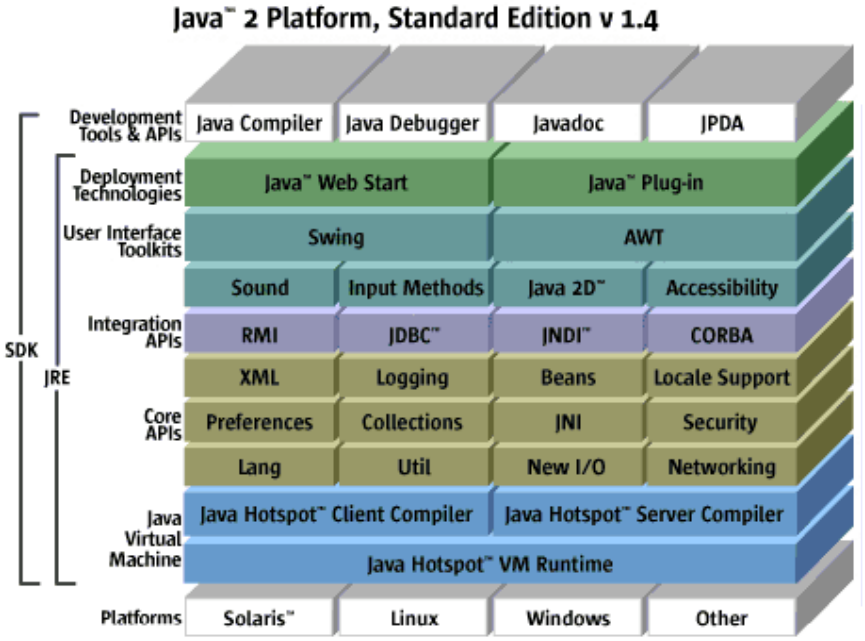
   Digunakan untuk membangun applikasi java berbasis PC.

b. Java 2 Enterprise Edition (J2EE)

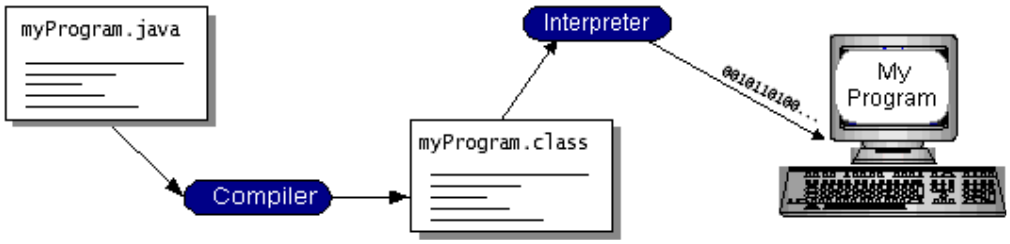
   Digunakan untuk membangun applikasi java pada lingkungan enterprise, dengan fungsi-fungsi seperti *Enterprise Java Bean* (EJB), COBRA, Servlet dan Java Server Page (JSP).

c. Java 2 Micro Edition (J2ME)

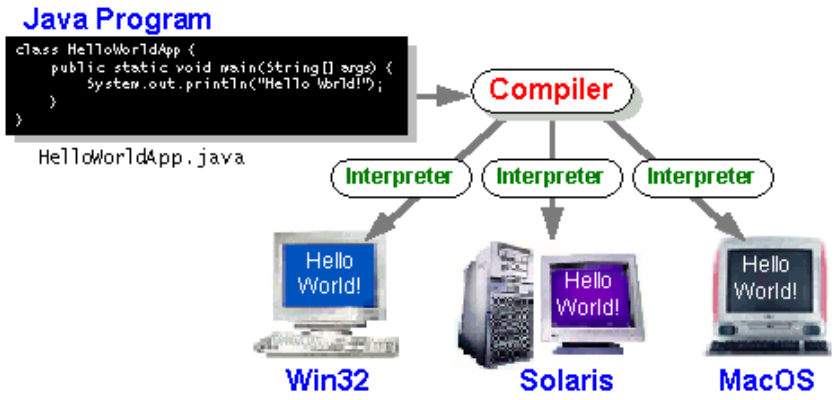
   Digunakan untuk membangun applikasi java berbasis *handhled device*. Seperti PDA, Handphone dan Pocket PC.



Gambar 2.41 J2SE platform
(sumber : Franck Allimant, "The Java Tutorial")



Gambar 2.42 Ilutrasi Java
(sumber : Franck Allimant, "The Java Tutorial")



Gambar 2.43 Kompabilitas Java
(sumber : Franck Allimant, "The Java Tutorial")

Java menggunakan metode *Object Oriented Programming*, yang dimaksud *Object Oriented Programming* adalah sebuah applikasi yang di rancang berdasarkan karakteristik modelnya baik secara abstrak atau dengan objek sebenarnya yang menggunakan kelas-kelas dan object-object.

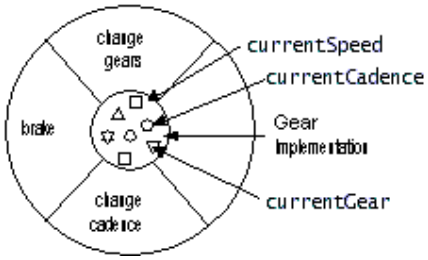
'Model Data Berorientasi Objek' atau object-oriented programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dikemas dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

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

1. **Class**

   Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya. Class atau kelas adalah cetak biru, prototipe,

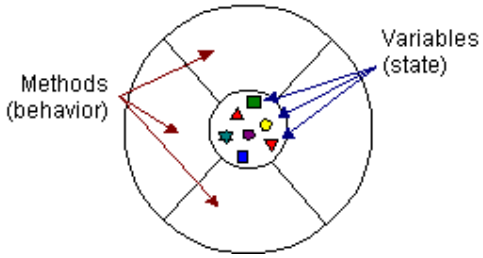
yang mendefinisikan variable-variable dan metode-metode umum dari beberapa jenis objek .



Gambar 2.44 Conoh pengguaan Class.
(sumber : Franck Allimant, "The Java Tutorial")

2. Object

Objek - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek. Terdiri dari *state* dan *behavior*, contoh : Kucing mempunyai state (nama, warna, ras dan lainnya) dan mempunyai *behavior* (mengeong, mendengkur, suka bermain dan lainnya)



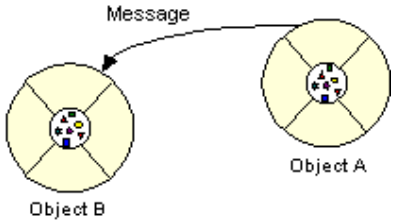
Gambar 2.45 Ilustrasi gambaran umum dari sebuah applikasi berorientasi object.
(sumber : Franck Allimant, "The Java Tutorial")

3. Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4. Message

Aplikasi yang bersifat objek dapat saling berinteraksi dengan mengirimkan sebuah pesan (Message).



Gambar 2.46 Ilustrasi peririman pesan
(sumber : Franck Allimant, "The Java Tutorial")

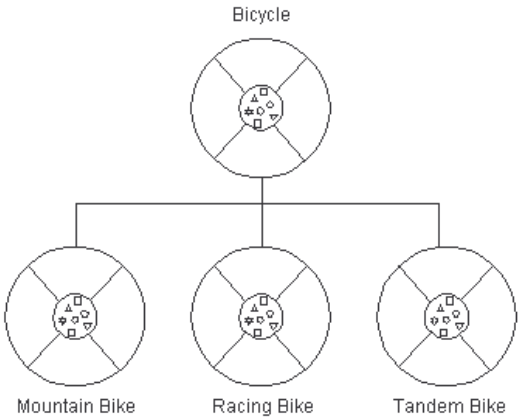
5. Polimorfisme

Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan

"gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

6. Inheritance

Inheritance atau sifat bawaan, adalah turunan dari super class atau kelas utama yang sama sifat-sifatnya dan menciptakan sebuah object yang baru. Inheritance mengatur polimorfisme dan enskapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada - objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas.)



Gambar 2.47 Contoh Inheritance
(sumber : Franck Allimant, "The Java Tutorial")

7. Enkapsulasi

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

8. Interface

Interface atau tampilan adalah entitas yang tidak mempunyai hubungan satu dengan yang lainnya yang digunakan untuk melakukan interaksi dan biasanya mempunyai aturan-aturan tertentu.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah

tersebut(terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bagian administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bagian administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah cukup dengan mengolaborasi antar objek-objek yang ada karena setiap objek memiliki perannya masing-masing.

2.5 JAVA 3D

Java 3D adalah Application Programming Interface (API) yang berdasarkan scene graph dari platform java. Java 3D berjalan diatas OpenGL atau Direct3D. sejak versi 1.2 Java 3D dikembangkan dibawah Java Community Process. JSR 926 ditetapkan sebagai Java 3D 1.4, sejak tahun 2007 versi terbarunya adalah versi 1.5 (yang dikeluarkan pada Desember 2006) dan versi 1.5.1 sedang dalam tahap pengembangan yang rencananya akan di luncurkan pada bulan Mei 2007.

Bila di banding-banding kan dengan metode lainnya, Java 3D tidak hanya meliputi grafik API saja, tetapi juga mengapsulitasi antarmuka grafika programming menggunakan konsep objek orientasi sebenarnya. Terdapat sebuah scene yang di bangun melalui scene graph yang mereprensenasikan sebuah objek

yang akan ditampilkan. Scene graph terdiri dari sebuah hirarki yang terdiri dari beberapa elemen yang dibutuhkan untuk menampilkan objek-objek tertentu. Dan juga sebagai tambahan Java 3D juga menyediakan dukungan suara.

Intel, Silicon Graphics, Apple, dan Sun semuanya telah memakai scene graph API yang sedang dikembangkan pada tahun 1996. Karena mereka menginginkan untuk membuat versi Java, maka dimulailah proyek Java 3D tersebut. Pekerjaan pengembangannya telah dimulai pada tahun 1997, lalu versi beta umumnya dirilis pada bulan maret 1998. Dan versi pertamanya pada bulan desember 1998. Dari pertengahan tahun 2003 sampai dengan musim panas 2004 pengambangannya tidak dilanjatkan. Lalu Java 3D di rilis sebagai proyek komunitas sumber dan Sun serta para sukarelawan melanjutkan pengembangannya. Version 1.4.0 di rilis pada tanggal 24 Februari 2006. Version 1.5.0 di rilis pada tanggal 13 Desember 2006.

Fasilitas-fasilitas yang terdapat pada Java 3D antar lain :

a. Multithreaded scene graph structure .
b. Platform independent .
c. Generic Real-time API, bergunan untuk visualisasi dan game.
d. Bisa digunakan untuk retained, compiled-retained, dan immediate mode rendering.
e. Termasuk hardware-accelerated JOGL, OpenGL and Direct3D renderers (tergantung platform-nya) .

f. Sophisticated virtual-reality-based model dapat dilihat dengan dukungan stereoscopic rendering dan konfigurasi complex multi-display.
g. CAVE (multiple screen projectors) .
h. 3D spatial sound.
i. Programmable shaders, mendukung GLSL dan CG
j. Stencil buffer.
k. Mendukung fasilitas import terhadap hampir semua format populer, seperti 3DS, OBJ, VRML, X3D, NWN, dan FLT.