BAB II

LANDASAN TEORI

2.1 Pengantar Grafika Komputer

Grafika Komputer merupakan sekumpulan alat yang digunakan untuk membuat gambar(to create pictures) dan berinteraksi dengan gambar dengan cara-cara seperti yang biasa digunakan(and to interact with them in natural ways).Peralatan tersebut berupa perangkat keras(hardware)ataupun piranti lunak(software).Semua peralatan tersebut memberi fasilitas kepada programmer untuk membuat program-program grafis yang baik.Secara umum program-program komputer grafis ditujukan untuk memudahkan interaksi antara manusia dengan komputer.Dengan memvisualkan kata-kata menjadi gambar maka manusia akan lebih mudah berinteraksi dengan komputer,perintah-perintah tidak lagi diberikan dengan cara menulis tetapi dengan cara menunjuk area atau gambar tertentu.

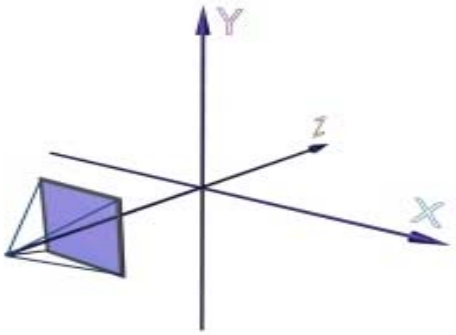
Implementasi grafika komputer yang diaplikasikan pada program simulasi berbasis visual interaktif untuk gerak benda diharapkan dapat membantu dan mempercepat pemahaman dalam mempelajari tentang gerak benda sehingga dapat meningkatkan daya minat belajar karena cara yang dipakai dalam perangkat lunak ini lebih menarik dengan fasilitas menu yang sederhana..

Grafika tiga dimensi dapat menggambarkan sebuah benda dalam dunia nyata menjadi sebuah objek yang terkesan hidup sedangkan pada grafika dua dimensi belum memimplementasikan dalam bentuk objek tiga dimensi.

2.2 Sistem Koordinat, window dan viewport

Pada koordinat cartesian arah vertikal digunakan sebagai sumbu y dengan y positif menuju keatas dan arah kebawah sebagai y negatif,arah mendatar digunakan sebagai sumbu x dengan x positif kearah kanan dan x negatif kearah kiri.Tetapi layar monitor menggunakan arah kebawah orientasi yang berbeda.Monitor menggunakan arah kebawah sebagai sumbu y positif, disamping itu monitor juga tidak mengenal nilai negatif.

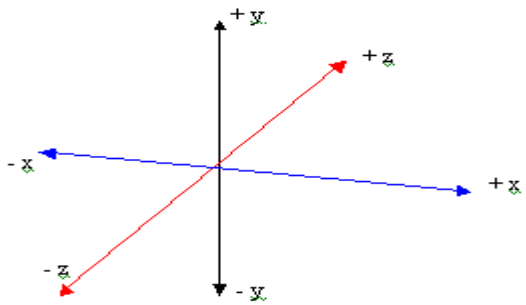
Window merupakan area didunia nyata yang menunjukkan bagian yang terlihat oleh pemirsa,sementara viewport merupakan area di layar monitor yang menunjukkan dimana window akan ditampilkan.Hubungannya pada viewport x positif berada dibawah sedangkan pada layar monitor x positif berada diatas.



2.3 Dunia Dua Dimensi Dan Tiga Dimensi

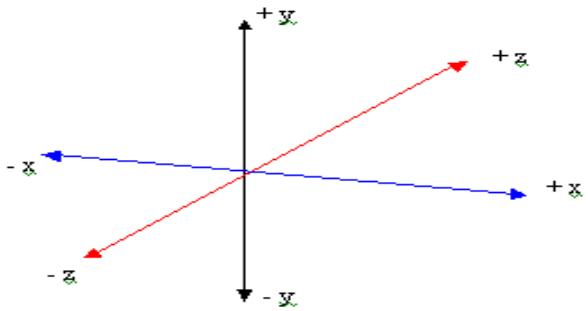
Apakah yang membedakan 2 dimensi dan 3 dimensi jawabannya adalah kedalaman yang didefenisikan sebagai jarak antara viewer terhadap benda yang dia lihat.Ini berarti berbeda dengan 2 dimensi yang hanya menggunakan 2 ukuran,yaitu panjang dan lebar,maka 3 dimensi menggunakan 3 ukuran yaitu

panjang, lebar dan kedalaman.Secara geometri ketiga ukuran tersebut disimbolkan dengan sumbu:x, y, dan z

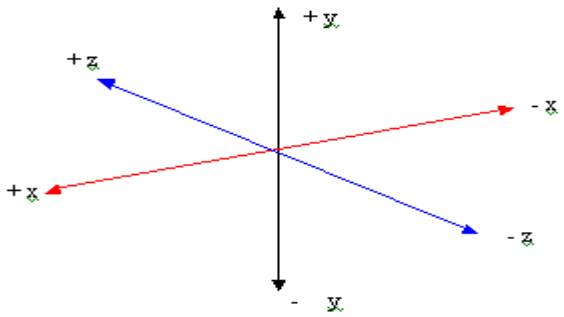


Gamabar 2.1 Sumbu Cartesian 3 Dimensi(sumber:Dunn & Pabery , 2002)

Ada dua cara menyatakan system koordinat 3 dimensi yaitu aturan tangan kiri dan aturan tangan kanan.Pada dasrnya perbedaan antara dua aturan tersebut adalah kemana sumbu x+ menunjuk.Perhatikan gambar 2.1 dan 2.2 atau aturan tangan kiri sumbu x+ ditunjuk oleh ibu jari dan mengarah kekanan sedangkan pada aturan tangan kanan sumbu x+ menuju arah kiri.Pada kedua aturan tersebut sumbu y+ maupun z+ menuju ke arah yang sama.



(a) Aturan tangan kiri



(b) Aturan tangan kanan

Gambar 2.2 Repesentasi sumbu 3 dimensi (sumber : Dunn & Pabery, 2002)

2.4 Transformasi 2 Dimensi

Tranformasi merupakan metode untuk mengubah lokasi titik, apabila tranformasi dikenakan terhadap sekumpulan titik yang membentuk sebuah benda maka benda tersebut akan mengalami perubahan.Misalkan sebuah objek dua dimensi dengan titik P ditranformasikan ke titik Q dengan menggunakan rumus-rumus tertentu sehingga Q merupakn lokasi baru dari titik P.

Rumus : (Qx , Qy) = T(Px , Py)

Atau dengan bentuk yang lebih sederhana:

Rumus : Q = TP

2.5 Transformasi 3 Dimensi

Tranformasi pada dunia 3 dimensi pada dasarnya sama dengan tranformasi pada 2 dimensi hanya pada 3 dimensi kita juga memperhitungkan sumbu z.

Sama seperti pada sumbu 2 dimensi ada 3 tranformasi dasar yang dapat dilakukan terhadap verteks yaitu:

1. Translasi
2. Penskalaan
3. Rotasi

Titik hasil tranformasi dapat diperoleh melalui rumus berikut ini

$$Q = P * M + tr$$

Keterangan:

Q = (Qx, Qy, Qz) menyatakan matriks 1x3 yang berisi titik hasil tranformasi

P = (Px, Py, Pz) menyatakan matriks 1x3 yang berisi titik yang ditranformasi.

Tr = (trx, try, trz) menyatakan matriks 1x3 yang berisi banyaknya pergeseran pada sumbu x, y, dan z.

M = Matriks tranformasi berukuran 3x3 seperti contoh dibawah ini:

$$M = \begin{pmatrix} m00 & m01 & m02 \\ m10 & m11 & m12 \\ m20 & m21 & m22 \end{pmatrix}$$

2.5.1 Translasi 3 Dimensi

Translasi adalah tranformasi yang menghasilkan lokasi baru dari sebuah objek sejauh jarak pergeseran tr =(trx, trz).Translasi dilakukan dengan menggunakan matriks seperti contoh dibawah ini:

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2.5.2 Penskalaan 3 Dimensi

Berbeda dengan translasi yang tidak mengubah bentuk objek pada penskalaan akan mengubah bentuk objek sebesar skala sx, sy, sz .Penskalaan dilakukan dengan mengisi tr =(0, 0, 0) dan matriks M adalah sebagai berikut:

$$M = \begin{pmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & Sz \end{pmatrix}$$

Tabel 2.1 niali Sx, Sy, Sz dan pengaruhnya

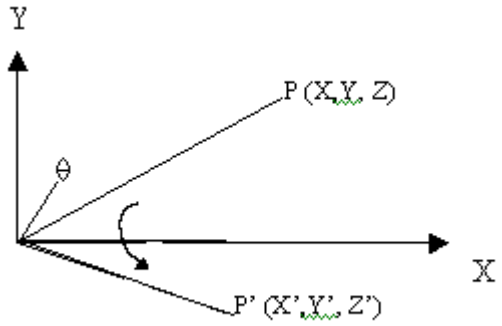
Nilai	Keterangan
(Sx, Sy, Sz) > 1	Pembesaran
(Sx, Sy, Sz) < -1	Pembesaran dengan pencerminan
-1 > (Sx , Sy , Sz) <1	Pengecilan dengan / tanpa pencerminan

2.5.3 Rotasi 3 Dimensi

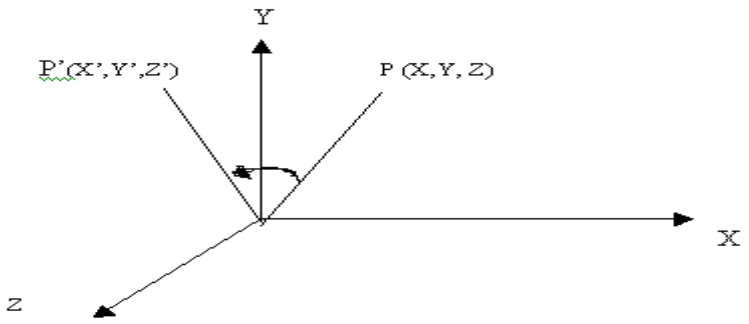
Berbeda dengan rotasi 2 dimensi yang menggunakan titik pusat (0,0) sebagai pusat perputarannya,rotasi 3 dimensi menggunakan sumbu koordinat sebagai pusat perputarannya.Dengan demikian ada 3 macam rotasi yang dapat dilakukan yaitu rotasi sumbu X, rotasi sumbu Y, rotasi sumbu Z.Mengingat ada 3

sumbu rotasi maka matriks tranformasi yang digunakan juga bergantung kepada sumbu putar.

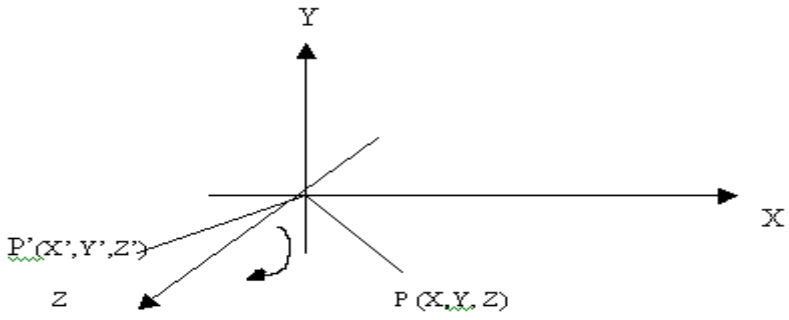
Rotasi pada sumbu x adalah sebagai berikut :



Rotasi pada sumbu Y adalah sebagai berikut :

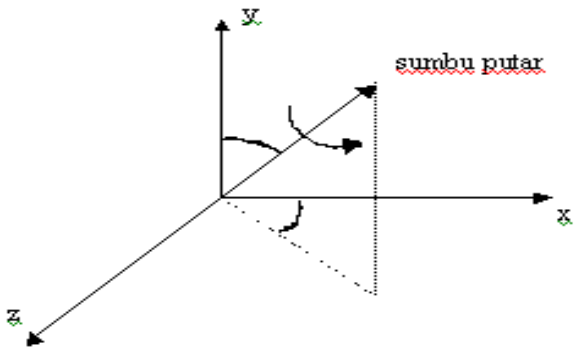


Rotasi pada sumbu Z adalah sebagai berikut :



Selain rotasi pada sumbu x , y, z ada juga rotasi gabungan dari ketiga sumbu tersebut dengan kata laian rotasi menggunakan sumbu bebas seperti dapat kita lihat pada gambar berikut ini:

Rotasi pada sumbu bebas adalah sebagai berikut :



2.6 Tranformasi Di OPENGL

OpenGL sudah menyediakan sejumlah perintah yang dapat digunakan untuk melakukan tranformasi.Perintah-perintah yang berhubungan dengan tranformasi adalah:

Procedure glMatriksMode(mode :Glenum);

Mengatur pemakaian matriks ada tiga macam matriks yang dapat digunkan yaitu :MODELVIEW,PROJECTION dan TEXTURE.MODELVIEW adalah matriks yang digunakan untuk melakukan tranformasi>PROJECTION dalah matriks yang digunkan untuk proyeksi dan TEXTURE adalah matriks yang digunakan untuk mengatur tekstur/tampilan permukaan benda.

Procedure glLoadMatriksd (m:PGLdouble);
Procedure glLoadMatriksf (m:PGLfloat);
Procedure glLoadMatriks (m:PGLdouble);
Procedure glLoadMatriks (m:PGLfloat);

Menggunakan matriks m untuk melakukan tranformasi:

glLoadIdentity

Mengatur agar matriks yang digunakan diisi dengan matriks identitasMatriks identitas adalah matriks yang elemen diagonal kekanan bawah berisi angka 1 dan elemen yang lain berisi 0.

Procedure glTranslated (x, y, z : GLdouble);
Procedure glTranslatef (x, y, z : GLfloat);
Procedure glTranslate (x, y, z : GLdouble);
Procedure glTranslate (x, y, z : PGLfloat);

Menggunakan tranformasi translasi untuk mengkalikan isi matriks yang sedang digunkan.Parameter x, y, z menyatakan banyaknya translasi pada sumbu x, y, z

Procedure glRotated (angle x, y, z : GLdouble);
Procedure glRotatef (angle x, y, z : GLfloat);
Procedure glRotate (angle x, y, z : GLdouble);
Procedure glRotate (angle x, y, z : GLfloat);

Menggunakan tranformasi rotasi untuk mengkalikan isi matriks yang sedang digunakan.Parameter x, y, z menyatakan vector yang digunakan sebagai sumbu

putar.Vektor sumbu putar dibentuk dari titik (0,0,0) menuju ke (x, y, z).Apabila salah satu parameter diisi dengan nilai 1 dan parameter yang diisi 0 maka perputaran dilakukan terhadap sumbu sesuai dengan parameter yang bernilai 1.

2.7 Karakteristik Cahaya

Cahaya dapat dilihat sebagai gelombang energi atau artikel(photon).Cahaya yang dipandang sebagai gelombang energi dapat dibagi menjadi dua bagian besar yaitu:

1. Cahaya terlihat(visible light)
2. Cahaya tidak terlihat(invisible light)

Cahaya terlihat mempunyai panjang gelombang 390 – 720 nm(namometer) dan disebut sebagai cahaya terlihat Karena sel-sel mata kita umumnya hanya peka terhadap energi dengan panjang gelombang 400-700 nm.Cahaya tidak terlihat mempunyai panjang gelombang < 390 nm atau panjang gelombang > 720 nm. Karena cahaya merupakan gelombang energi maka warna-warna yang kita lihat dapat didefinisikan sebagai fungsi dari frekuensi (f) atau fungsi dari panjang gelombang (^).Frekuensi dan panjang gelombang dari cahaya kromatik,yaitu cahaya dengan satu warna, dapat dirumuskan sebagai : c = f^
Dimana c merupakan kecepatan cahaya.Frekuensi tidak bergantung kepada materi dari objek tetapi panjang gelombang dan c bergantung pada materi objek.Pada saat cahaya menimpa permukaan benda maka sebagian cahaya akan dipantulkan dan sebagian lain diserap.Bergantung kepada frekuensi atau panjang gelombang yang dipantulkan dan diserap maka kita akan melihat warna,sebagai contoh apabila sebuah benda disorot warna putih dan benda tersebut mementulkan

sebagian besar energi dengan frekuensi rendah (600 – 700 nm)maka kita akan melihat warna merah,tetapi apabila benda tersebut memantulkan energi pada frekuensi sekitar 400 nm maka kita akan melihat warna biru.Rentang panjang gelombang yang dominant dipantulkan (dominant frequency/wavelength)disebut sebagai hue atau secara ringkas kita sebut sebagai warna.

2.8 Model Sumber Cahaya

Sumber cahaya dapat dikelompokkan menjadi dua macam yaitu:

1. **Cahaya Lingkungan**

   Didalam dunia nyata semua benda akan mementulkan cahaya meskipun sedikit,cahaya lingkungan digunaka untuk memodelkan cahaya yang berasal dari berbagai sumber tersebut.Cahaya lingkungan tidak mempunyai arah dan lokasi.Pengaruh cahaya lingkungan dirumuskan sebagai berikut : Iab = Ia ra

2. **Cahaya Titik**

   Sumber cahaya kedua adalah cahay titik(point Light),sumber cahay ini mempunyai lokasi dan arah dengan demikian jarak antara sumber cahaya terhadap benda akan berpengaruh terhadap kuat cahaya yang diterima oleh benda.Sumber cahaya directional mempunyai karakteristik energi dari sumber tersebut menyebar ke semua arah dengan kekuatan yang sama pada kenyataannya tidak ada sumber cahaya yang bersifat directional tetapi karena energi dari sumber cahaya tersebut sangat kuat dan dapat menempuh jarak

yang sangat jauh maka dianggap rectional antara lain matahari.Sumber cahaya positional mempunyai sifat dimana energi dari sumber cahaya tersebut akan melemah sebanding dengan jarak dan sudut terhadap sumber cahaya.Apabila cahaya yang keluar dari sumber cahaya positional dibatasi sudut penyebarannya maka kita akan memperoleh efek lampu sorot.

2.9 Model Pencahayaan di OPENGL

OpenGL menggunakan model pencahayaan OpenGL memerlukan informasi mengenai :

1. Karakteristik materi benda yaitu nilai koefisien ambient, diffuse, dan specular.Nilai-nilai tersebut menyatakan berapa persen cahay yang menimpa benda akan dipantulkan atau diserap.
2. Karakteristik sumber cahaya meliputi koefisien Ia, Id, Is, serta factor attenuation dan spotlight.
3. Vector normal dari vertex

Tabel 2.2 parameter glMaterialfv

Parameter	Isi	Keterangan
Face	GL_FRONT	Material hanya berpengaruh terhadap sisi depan permukaan
	GL_BACK	Material hanya berpengaruh terhadap sisi belakang permukaan.
	GL_FRONT_AND_BACK	Material berpengaruh terhadap sisi depan dan sisi belakang permukaan.
Type	GL_AMBIENT	Mengatur koefisien ambient(ra)
	GL_DIFFUSE	Mengatur koefisien diffuse (rd)
	GL_SPECULAR	Mengatur koefisien specular (rs)
Vector	Array berisi nilai R, G, B	Mengatur nilai R, G, B, A dari masing-maing type.

2.10 Render

Proses pengaplikasian *shading model* kedalam *scene* dinamakan rendering. Rendering mentransformasikan *scene* menjadi *pixel* di layar komputer karena *pixel* merupakan unit terkecil yang dapat menerima sebuah pengalamatan yang ditujukan pada layar.

Kebanyakan orang pada umumnya menggemari sesuatu yang bergerak atau hidup bukannya sesuatu yang kaku atau statik. Render merupakan tahapan untuk membuat suatu objek yang awalnya hanya berupa garis dan gambar sehingga berkesan hidup biasa dengan kata lain bisa disebut rendering.Rendering

merupakan salah satu teknik memperjelas objek tiga dimensi sehingga dapat menghasilkan suatu gambar yang realistis(nyata) diantara dengan cara penambahan beberapa efek seperti pencahayaan dan bayangan.Pada proses rendering ini gambar dapat mempengaruhi hasil dari proses rendering yang diberjalan.

2.11 Dasar-dasar OpenGL

OpenGL merupakan kepanjangan dari Open Graphic Library.OpenGL diproduksi oleh silicon Graphics dan pada awalnya ditujukan hanya untuk system komputer mereka, tetapi dalam perkembangannya OpenGL diterima menjadi salah satu bakuan dalam grafika computer dan saat ini telah diimplementasikan dalam berbagai system computer.OpenGL merupakan pustaka program yang mennyediakan sejumlah perintah yang berhubungan dengan grafika.

Antarmuka antara program yang akan memenfaatkan OpenGL disediakan melalui sejumlah Application Program Interface(API).Melalui API tesebut pemrogram dapat memerintahkan OpenGL melakukan suatu tindakan tertentu dan mengigat OpenGL telah diimplementasikan kedalam berbagai system computer maka pemrogram dapat menggunakan API yang sama dengan hasil yang sama untuk berbgai system computer.Dengan demikian pemrogram dapat berkonsentrasi pada penyelesaian masalahnya dan tidak lagi direpotkan dengan urusan-urusan implementasi teknis.Didalam system operasi Windows,OpenGL diimplementasikan ke dalam 2 buah file yaitu : opengl32.dll dan glut32.dll.Opengl32.dll merupakan pustaka program yang menyediakan perintah-

perintah utama dari OpenGL sedangkan glut32.dll menyediakan perintah-perintah tambahan(utility)dari OpenGL.

2.12.1 Menghubungkan Delphi Dengan OpenGL

Bagaimana menghubungkan Delphi dengan OpenGL..?Caranya adalah cari apakah Delphi yang anda gunakan sudah menyediakan file OpenGL.dcu di ...\lib.Jika sudah tersedia maka anda cukup menambahakan klausa OpenGL di perintah uses dari program anda seperti dicontohkan oleh listing berikut ini:

Uses

Windows,Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, OpenGL;

Apabila versi Delphi yang anda gunakan tidak menyediakan file opengl.dcu maka andaterpaksa harus mendownload sendiri file tersebut di internet dengan cara gunakan situs www.google.com untuk mencari dimana anda dapat memperoleh file opengl.dcu sesuai versi Delphi yang anda gunakan.

2.13 Menggunakan OPENGL

Mengingat OpenGL bukan merupakan bagian dari system operasi windows maka kita perlu memberitahu windows bahwa program yang kita buat

akan memanfaatkan API dari OpenGL.Informasi yang perlu disampaikan adalah informasi mengenai:

1. Format pixel yang disampaikan melalui stuktur data TpixelFormatDescriptor.Format pixel memberikan informasi mengenai jumlah warna dalam bit per pixel, kedalaman sumbu z, serta tipe pixel(pola titik).
2. Device Context yang menyatakan perangkat(device)yang akan digunkan oleh OpenGL untuk menggambar di layer.Antara perangkat yang disediakan oleh Windows dan perangkat yang disediakan oleh OpenGL harus dihubungkan sehingga apa yang digambar oleh OpenGL akan ditampilkan oleh Windows.

Tahapan untuk menggunakan OpenGL dapat dirumuskan dalam 4 langkah berikut ini :

1. Tentukan format pixel
2. Buat device context untuk me-render/menggambar
3. Aktifkan device context yang dibuat pada langkah 2.
4. Gambar pixel.

Table 2.3 Mode pada GLBegin

Nilai	Arti
GL_POINTS	Setiap verteks diperlakukan sebagai titik terpisah
GL_LINES	Dua pasang verteks diperlakukan sebagai garis.
GL_LINE_LOOP	Sama seperti GL_LINES tetapi setiap garis saling dihubungkan.
GL_LINE_LOOP	Sama seperti GL_LINE_STRIP tetapi vertek pertama dan terakhir membentuk garis pula
GL_TRIANGLES	Tiga pasang verteks dianggap sebagai bidang segitiga.
GL_TRIANGLE_STRIP	Bidang segitiga yang saling berhubungan.
GL_TRIANGLE_FAN	Mirip GL_TRIANGLE_STRIP tetapi semua bidang menggunakan satu verteks yang sama.
GL_QUADS	Empat verteks dianggap sebagai polygon empat sisi(quadrilaterals)
GL_QUAD_STRIP	Pasangan quadrilaterals
GL_POLYGON	Verteks dianggap sebagai titik sudut polygon.

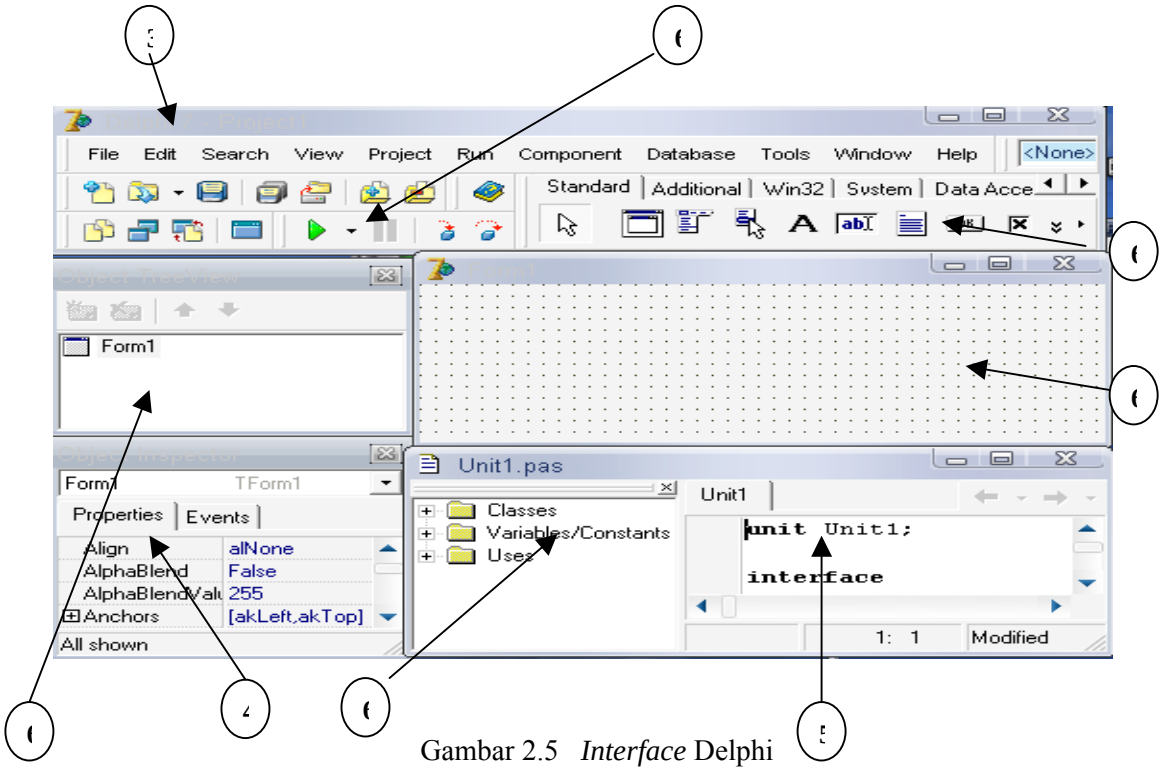
2.8 Borland Delphi 7.0

Borland Delphi merupakan suatu bahasa pemprograman yang memberikan berbagai fasilitas pembuatan aplikasi visual.Adapun keunggulan dari bahasa pemprograman ini terletak pada produktivitas,kualitas,pengembangan perangkat

lunak,kecepatan kompilasi,pola desain yang menarik serta diperkuat dengan pemprograman yang terstuktur.Keunggulan lain dari Delphi adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis windows.

Khusus untuk pemprograman database Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program.Format database yang memiliki Delphi adalah format database paradox,dBase,MS Access,ODBC,SyBASE,Oracle dan lain-lain.

Lingkungan pengembangan terpadu atau Integrated Development Environment(IDE) dalam program Delphi terbagi menjadi delapan bagian utama yaitu Main Window, ToolBar, Component Palette, Form Designer, Code Editor, Objek Inspector, Exploring, dan Objek TreeView.Untuk lebih jelasnya telihat pada gambar 2.5 Interface Delphi



Gambar 2.5 *Interface* Delphi

Keterangan :

1. Main Window yaitu jendela utama Delphi yang berisi main menu, *toolbar*, dan *component palette*.
2. Toolbar yaitu merupakan tombol yang berfunsi sebagai pengganti suatu menu perintah yang sering digunakan atau disebut juga Speedbar
3. *Component palette*, berisi *icon-icon* yang melambangkan komponen yang terdapat pada VCL(Visual Component Library).
4. Code Explorer yaitu lembar kerja baru yang terdapat dalam Delphi 7 yang tidak ada pada versi sebelumnya
5. Jendela *Object TreeView* yang berisi hirarki objek dengan *root* adalah *form* yang sedang aktif.
6. *Object Inspector* yaitu *Properti atau karakteristik dari sebuah komponen.*.
7. Code *Editor* yang digunakan untuk menulis kode program.
8. *Form* Designer yang digunakan untuk mendesain tampilan program.

2.9 Shading

Banyak sekali jenis algoritma *shading* yang dapat diterapkan dalam grafika tiga dimensi. Algoritma *shading* yang paling sederhana adalah *flat shading*. Sedangkan algoritma *shading* yang lebih kompleks adalah *diffuse shading*.

2.9.1 Flat Shading

Metode yang cepat dan mudah untuk merender permukaan objek adalah *constant intensity shading* atau *flat shading*. Dalam metode ini Semua titik pada permukaan ditampilkan dengan nilai intensitas yang sama.

```
if MoveX then                          LightPosX  ← LightPosX +
 if LightPosX < 4 then             0.1
                                       else
```

```
       MoveX  ←  False
     endif
    else
     if LightPosX > -4.1 then
        LightPosX  ← LightPosX
    endif


if MoveY then
    if LightPosY < 2.2 then
        LightPosY  ← LightPosY +
0.1
    else
       MoveY  ← False
    endif
   else
    if LightPosY > -2.6 then
       LightPosY  ← LightPosY
   endif
end
```

2.9.2 Diffuse Shading

Karena cahaya ambien hanya menghasilkan suatu bentuk permukaan yang tidak menarik, maka jarang sekali suatu gambar di render dengan hanya menggunakan cahaya ambien saja Paling tidak ada suatu sumber cahaya dalam gambar, yang pada umumnya digunakan juga sebagai titik pandang.

```
   if MoveX then
    if LightPosX < 4 then
       LightPosX  ← LightPosX +
0.1
    else
       MoveX  ←  False
    endif
   else
    if LightPosX > -4.1 then
       LightPosX  ← LightPosX -
0.1
    else
       MoveX  ← True
    endif
   endif
```

```
   if MoveY then
    if LightPosY < 2.2 then
       LightPosY  ← LightPosY +
0.1
    else
       MoveY  ← False
    endif
   else
    if LightPosY > -2.6 then
       LightPosY  ← LightPosY -
0.1
    else
       MoveY  ← True
    endif
   endif
end
```

2.9.3 Phong Shading

Model empiris untuk menghitung daerah pantulan spekular, dikembangkan oleh *Phong Bui Thong* dan disebut model pantulan spekular Phong (*Phong specular-reflection model*), atau secara singkat disebut model Phong, mengatur intensitas dari pantulan spekular sebanding dengan $\cos\theta$. sudut θ dapat bernilai 0^o sampai 90^o derajat, maka $\cos\theta$ akan bernilai antara 0 sampai 1.