APLIKASI GRAFIKA KOMPUTER UNTUK PENCAHAYAAN DAN PENGARSIRAN OBJEK 3 DIMENSI

Ike Pertiwi Windasari[1], Agung B.P.[2], Aghus Sofwan[2]
Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro

***Abstrak* - Salah satu cabang ilmu komputer yang berkembang pesat adalah grafika komputer. Bagian yang akan dibahas adalah pencahayaan dan pengarsiran pada objek 3 dimensi yang merupakan hal mendasar untuk menimbulkan kesan realistik pada objek 3 dimensi.**

Tugas akhir ini mencoba membuat aplikasi yang dapat memodelkan pencahayaan dan pengarsiran pada objek 3 dimensi. Model pencahayaan yang digunakan adalah cahaya ambient, model pencahayaan Lambert (cahaya diffuse) dan model pencahayaan Phong (cahaya specular) Metode pengarsiran yang digunakan yaitu flat shading, gouraud shading, dan phong shading. Aplikasi ini dibuat menggunakan bahasa pemograman Delphi.

Tugas akhir ini diharapkan mampu menghasilkan aplikasi yang dapat memodelkan proses pencahayaan dan pengarsiran pada objek 3 dimensi.
Kata kunci: pencahayaan, pengarsiran, *flat, gouraud, phong*

1. Pendahuluan

1.1 Latar Belakang Masalah

Penerapan grafika komputer dalam berbagai aplikasi seperti *video game*, *CAD* (*Computer Aided Design*), animasi, dan simulasi menunjukkan perkembangan yang pesat dari grafika komputer, salah satu cabang ilmu komputer. Grafika komputer sediri terbagi menjadi 2 bagian, yaitu grafika komputer pada dimensi 2 dan grafika komputer pada dimensi 3. Grafika komputer dimensi 3, terbagi lagi menjadi beberapa bagian yaitu transformasi (*transformation*), pewarnaan (*colouring*), pencahayaan dan pengarsiran (*illumination and shading*), dan *ray tracing*. Pada tugas akhir ini, bagian yang akan dibahas adalah proses pencahayaan dan pengarsiran pada objek, yang merupakan hal mendasar untuk menimbulkan kesan realistik pada objek 3 dimensi. Ada tiga jenis pengarsiran yang akan digunakan pada program ini yaitu *flat shading, gouraud shading* dan *phong shading*.

1.1 Tujuan

Tujuan pembuatan Tugas Akhir ini adalah untuk membuat aplikasi untuk memodelkan pencahayaan pada objek 3 dimensi dengan 3 teknik *shading* yaitu *flat shading, gouraud,* dan *phong shading*.

1.2 Pembatasan Masalah

Pada Tugas Akhir ini pembahasan akan dibatasi pada hal-hal berikut ini :

1. Pada Tugas akhir ini dirancang suatu aplikasi untuk memodelkan pencahayaan pada objek 3 dimensi dengan 3 teknik shading yaitu *flat shading, gouraud,* dan *phong shading.*
2. Objek 3 dimensi yang ditampilkan hanya objek 3d *convex.*
3. Aplikasi dibuat menggunakan bahasa pemrograman Delphi.

2. Pencahayaan dan Pengarsiran Objek 3 Dimensi

Ada tiga model pencahayaan lokal yaitu cahaya *ambient*, cahaya *diffuse* dan cahaya *specular*. Dalam komputer grafis ada tiga teknik pengarsiran yang umum digunakan. Teknik-teknik tersebut adalah *flat shading, gouraud shading* dan *phong shading.*

2.1 Model Pencahayaan

1. Cahaya Ambient

Walaupun sebuah benda tidak terkena cahaya secara langsung, benda tersebut masih dapat terlihat. Hal ini terjadi karena adanya cahaya yang dipantulkan secara tidak langsung oleh benda di sekitarnya. Model pencahayan sederhana ini disebut cahaya *ambient*. Benda yang hanya dikenai cahaya *ambient* akan terlihat sangat datar, seperti terlihat pada gambar 2.1.

$$I_{ambient} = K_a I_i \qquad (2.1)$$

Gambar 2.1 Cahaya Ambient

2. Cahaya Diffuse

Saat sebuah benda matt dikenai cahaya, maka intensitas cahaya akan dipantulkan rata ke semua arah. Inilah yang disebut cahaya diffuse.

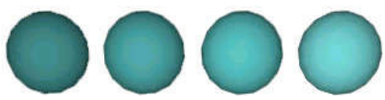
$$I_{diffuse} = K_d I_i \cos\theta \qquad (2.2)$$

di mana I_i adalah intensitas sumber cahaya dan θ adalah sudut yang dibentuk antara vektor normal dengan sumber cahaya, serta K_d adalah koefisien pantul dari poligon tersebut.

$$\cos\theta = \left(\frac{\bar{\mathbf{n}} \cdot \bar{\mathbf{v}}}{|\mathbf{n}||\mathbf{v}|}\right)$$

disederhanakan menjadi

$$\cos\theta = n \cdot v \qquad (2.3)$$



Gambar 2.2 Cahaya *Diffuse*

3. Cahaya Specular

Pada benda berkilau, terdapat sebidang daerah yang terang pada permukaan benda yang letaknya tergantung dari sudut pandang terhadap benda. Efek cahaya ini disebut pencahayaan specular.

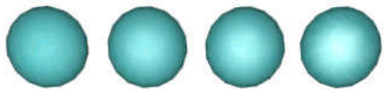
$$I_{specular} = K_s I_i (v \cdot r)^n \qquad (2.4)$$

dimana *Ks* merupakan koefisien specular, vektor *V* adalah unit vektor pengamat,dan *R* merupakan unit vektor refleksi objek.

R didapat dari rumus dibawah ini.

$$r = (2(n \cdot l))n - l \qquad (2.5)$$

dimana *n* merupakan vektor normal, dan *l* merupakan vektor cahaya.



Gambar 2.3 Cahaya Specular

4. Total Intensitas Pencahayaan

Total intensitas pencahayaan didapat dari penjumlahan intensitas cahaya ambient, cahaya diffuse, dan cahaya specular. Seperti terlihat pada rumus dibawah ini.

$$I_{total} = K_a I_i + K_d I_i (n \cdot v) + K_s I_i (v \cdot r)^n \quad (2.6)$$

2.2 Model Pengarsiran

Sebelumnya, untuk mendeteksi apakah suatu poligon perlu diarsir atau tidak digunakan metode seperti *backface culling*, *painter sort*, *z buffer*, dll. Pada tugas akhir ini, metode yang digunakan adalah *backface culling*. Cara kerja metode ini adalah dengan hanya menggambar poligon yang memiliki vektor normal keluar, dengan cara mencari dot product dari vektor normal poligon dan vektor arah pelihat seperti terlihat pada persamaan 2.8. Apabila hasilnya lebih besar dari nol maka vektor normal poligon merupakan vektor normal keluar dan poligon akan digambar.

$$n.v > 0 \qquad (2.8)$$

Kelemahan dari metode backface culling ini adalah bahwa metode ini hanya dapat digunakan pada objek *convex.*

1. *Flat Shading*

Metode pengarsiran yang paling sederhana adalah *flat shading*. Metode ini hanya sekali menghitung intensitas untuk tiap-tiap poligon pada objek. Hasil yang didapatkan tentu saja tidak memuaskan, batas-batas antar poligon terlihat jelas sehingga objek akan kelihatan tekotak-kotak.

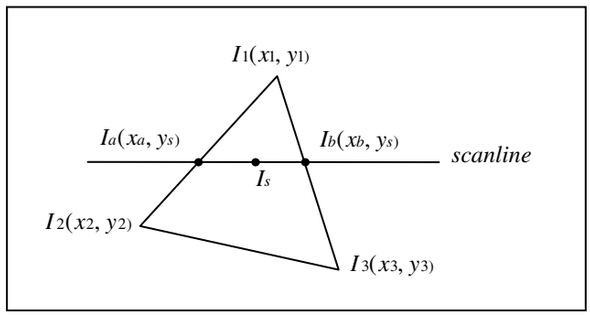
Langkah-langkah yang dilakukan untuk mengarsir poligon adalah sebagai berikut:

1. mencari vektor normal
2. mengambil sembarang titik yang terletak tepat pada poligon poligon tersebut (misalkan titik tengah poligon) sebagai titik acuan.
3. menghitung intensitas pencahayaan dari poligon tersebut.
4. Kemudian, seluruh poligon tersebut diarsir dengan intensitas yang telah dihitung.

Gambar 2.4 *Flat Shading*

2. *Gouraud Shading*

Untuk mendapatkan hasil yang lebih halus saat mengarsir poligon, digunakan metode *gouraud shading*. Perbedaan antara *gouraud shading* dengan *flat shading* adalah pada *gouraud shading*, intensitas tiap poligon dihitung pada titik-titik sudut yang membentuk poligon tersebut. Setelah semua intensitas pada tiap titik sudut poligon tersebut telah diketahui, dilakukan kalkulasi intensitas untuk tiap titik yang dibatasi oleh poligon tersebut dengan cara menginterpolasi (interpolasi = mencari nilai antara) intensitas pada sudut-sudut penyusun poligon tersebut. Berikut ini adalah gambar yang menerangkan prinsip interpolasi intensitas ini.



Gambar 2.5 Prinsip interpolasi intensitas

Karena pengarsiran dilakukan dengan cara horizontal terlebih dahulu lalu setelah itu baru vertikal, maka untuk mengarsir suatu titik pada suatu poligon yang diketahui intensitas sudut-sudut penyusunnya, intensitas pada titik-titik perpotongan antara garis horizontal proses pengarsiran, atau biasa disebut *scan line*, dengan poligon tersebut harus diketahui terlebih dahulu. Pada gambar 2.5, perpotongan antara *scan line* dengan poligon adalah titik *a(xa, ys)* dan titik *b(xb, ys)*, dan intensitasnya adalah *ia* dan *ib*. Intensitas pada kedua titik ini dapat dicari dengan menggunakan persamaan 2.8 berikut:

$$I_a = [I_1(y_s - y_2) + I_2(y_1 - y_s)] / y_1 - y_2$$
$$I_b = [I_1(y_s - y_3) + I_2(y_1 - y_s)] / y_1 - y_3 \qquad (2.8)$$

Setelah intensitas pada kedua perpotongan tersebut telah diketahui, maka intensitas pada titik yang akan diarsir dapat diketahui dengan metode yang serupa dengan metode untuk mencari intensitas pada titik perpotongan tersebut seperti persamaan 2.9 berikut.

$$I_s = [I_b(x_s - x_a) + I_a(x_b - x_s)] / x_b - x_a \qquad (2.9)$$

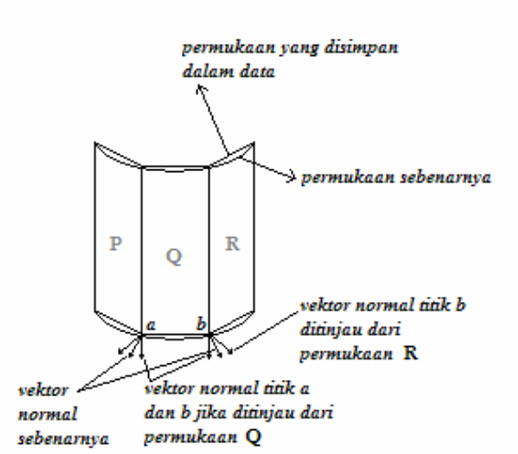
Untuk efisiensi komputasi, persamaan diterapkan kedalam perhitungan bertahap (*increment calculation*).

Intensitas suatu piksel dapat dihitung dari intensitas piksel sebelumnya dengan menambahkan step intensitas.

$$\Delta I_s = (I_b - I_a)/\, x_b - x_a$$

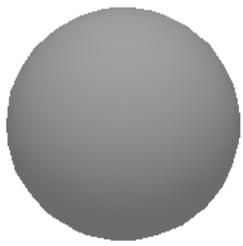
$$I_{s,n} = I_{s,n-1} + \Delta I_s \qquad (2.10)$$

Sampai saat ini, pengarsiran suatu obyek masih memberikan kesan yang terkotak-kotak seperti pada *flat shading*. Hal ini disebabkan oleh titik yang menyusun obyek tersebut, bila ditinjau dari poligon yang berbeda akan memiliki vektor normal yang berbeda, dan pada saat pengarsiran akan menghasilkan perubahan intensitas yang drastis. Gambar berikut akan lebih memperdalam pengertian akan hal ini.



Gambar 2.6 Vektor normal poligon yang berdekatan

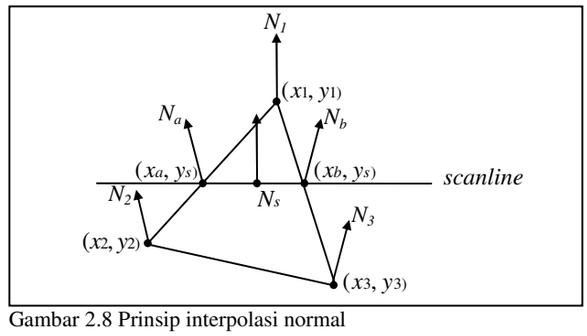
Untuk mengatasi hal ini, maka vektor normal pada titik tersebut dirata-rata untuk mendapatkan vektor normal yang sebenarnya. Hasil dari metode pengarsiran ini tampak seperti pada gambar berikut.



Gambar 2.7 *Gouraud Shading*

3. *Phong Shading*

Teknik ini mirip dengan teknik sebelumnya yaitu teknik *gouraud shading*, perbedaannya terletak pada saat melakukan interpolasi. Pada teknik sebelumnya, yang diinterpolasi adalah intensitas pada titik-titik sudut penyusun poligon yang sebelumnya telah dihitung terlebih dahulu, pada teknik ini, yang diinterpolasi adalah vektor normal (yang telah dirata-rata) dari titik-titik sudut penyusun poligon untuk mendapatkan vektor normal pada titik yang akan diarsir, dan melakukan perhitungan intensitas pada titik tersebut.



Gambar 2.8 Prinsip interpolasi normal

Oleh karena perhitungan intensitas dilakukan setiap kali akan mengarsir, maka beban komputasi dari teknik ini akan meningkat drastis daripada teknik sebelumnya. Namun demikian, hasil yang diperoleh akan lebih baik jika dibandingkan dengan teknik sebelumnya, terutama dalam perhitungan pencahayaan yang lebih rumit.

Pada gambar 2.8, perpotongan antara *scan line* dengan poligon adalah titik *a(xa, ys)* dan titik *b(xb, ys)*, dan normalnya adalah *Na* dan *Nb*. Normal pada kedua titik ini dapat dicari dengan menggunakan persamaan 2.11 berikut:

$$N_a = [N_1(y_s - y_2) + N_2(y_1 - y_s)]/\, y_1 - y_2$$

$$N_b = [N_1(y_s - y_3) + N_2(y_1 - y_s)]/\, y_1 - y_3 \qquad (2.11)$$

Setelah intensitas pada kedua perpotongan tersebut telah diketahui, maka intensitas pada titik yang akan diarsir dapat diketahui dengan metode yang serupa dengan metode untuk mencari intensitas pada titik perpotongan tersebut seperti persamaan 2.12 berikut.

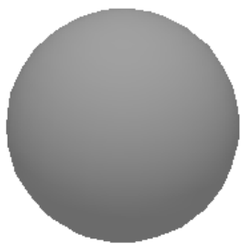
$$N_s = [N_b(x_s - x_a) + N_a(x_b - x_s)]/\, x_b - x_a \qquad (2.12)$$

Untuk efisiensi komputasi, persamaan diterapkan kedalam perhitungan bertahap (*increment calculation*).

$$\Delta N_s = (N_b - N_a)/\, x_b - x_a$$

$$N_{s,n} = N_{s,n-1} + \Delta N_s \qquad (2.13)$$

Dengan teknik ini perhitungan pencahayaan akan lebih akurat karena tiap titik yang akan diarsir memiliki vektor normal tersendiri, berbeda dengan teknik sebelumnya yang hanya menghitung intensitas pada beberapa titik dan "memperkirakan" intensitas pada titik lainnya. Berikut ini adalah hasil dari teknik *phong shading*. Hasil ini tampak serupa dengan teknik sebelumnya, karena teknik pencahayaan digunakan sederhana.

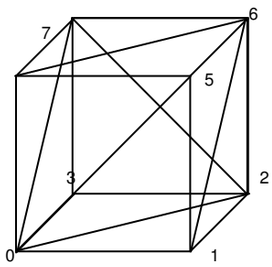


Gambar 2.9 *Phong Shading*

2.3 Objek 3 Dimensi

Sebuah objek 3 dimensi dapat disusun dengan menghubungkan titik-titik. Titik-titik penyusun objek

disebut juga dengan verteks, sedangkan kumpulan titik-titik yang dihubungkan oleh garis disebut wireframe atau kerangka. Objek 3 dimensi juga disusun oleh kumpulan permukaan. Permukaan merupakan sebuah poligon segitiga yang disusun dari verteks-verteks. Objek kubus, seperti terlihat pada gambar 2.5 dapat disusun berdasarkan data pada tabel 2.1 dan tabel 2.2.



Gambar 2.10 Kubus

Tabel 2.1 Verteks penyusun kubus

Verteks	x	y	z
0	-1	-1	-1
1	+1	-1	-1
2	+1	-1	+1
3	-1	-1	+1
4	-1	+1	-1
5	+1	+1	-1
6	+1	+1	+1
7	-1	+1	+1

Tabel 2.2 Permukaan (*face*) penyusun kubus

Face	v1	v2	v3	Keterangan
0	4	0	5	depan
1	0	1	5	depan
2	5	1	6	kanan
3	1	2	6	kanan
4	6	2	7	belakang
5	2	3	7	belakang
6	7	3	4	kiri
7	3	0	4	kiri
8	0	3	1	bawah
9	3	2	1	bawah
10	7	4	6	atas
11	4	5	6	atas

2.4 Fungsi GDI (*Graphics Device Interface*) pada Delphi

Untuk menggambar ke dalam kanvas, seorang pelukis membutuhkan palet untuk memilih warna, juga kuas, pena dan teknik menggambar. Win32 berusaha menerapkan hal ini kedalam bahasa pemrograman untuk menggambar objek yang dapat berinteraksi dengan pemakai. Fungsi-fungsi ini tersedia dalam *Graphics Device Interface* atau yang dikenal dengan GDI. Pada Delphi, fungsi-fungsi GDI dienkapsulasi dan disederhanakan dalam objek `TCanvas`. Penggunaannya cukup dengan menambahkan klausa `Graphics` pada kode program di bagian `uses`.

```
uses
Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms, Dialogs,
ComCtrls, StdCtrls, ExtCtrls, Buttons;
```

Kelas `TBitmap` berisi bitmap dan palet dari Win32. Sebelumnya, kita menentukan format piksel, juga ukuran bitmap dengan perintah dibawah ini.

```
framebuf:= tbitmap.Create;
framebuf.PixelFormat:= pf24bit;
framebuf.Width:= width;
framebuf.Height:= height;
```

Kemudian menentukan warna pena `(pen)` dan kuas `(brush)` pada `Canvas` dengan perintah berikut.

```
Canvas.Pen.Color:= clred;
Canvas.Brush.Color:= clblack;
```

Untuk menggambar ke dalam kanvas, kita memindahkan posisi `Canvas.Pen` ke posisi `(x,y)` yang diinginkan dengan fungsi `Canvas.MoveTo(x,y)` dan fungsi `Canvas.LineTo(x,y)` untuk menggambar garis dari posisi `Canvas.Pen` ke posisi `(x,y)`. Untuk menggambar poligon, GDI menyediakan fungsi `Canvas.Polygon()`.

3. Perancangan Perangkat Lunak

Pada Tugas akhir ini dirancang suatu aplikasi untuk memodelkan pencahayaan pada objek 3 dimensi dengan 3 teknik pengarsiran *(shading)* yaitu *flat shading, gouraud shading*, dan *phong shading*.

Dalam aplikasi ini, pengguna dapat melakukan tiga hal hal yaitu:

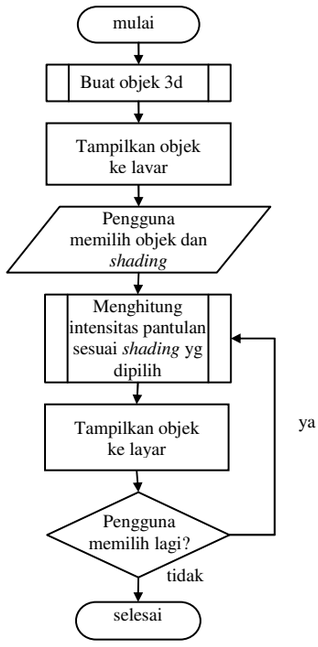
1. Pengguna dapat memilih objek yang akan ditampilkan kedalam layar, yaitu kubus, tabung, kerucut dan bola.
2. Pengguna dapat memilih jenis *shading* yang digunakan, yaitu *wireframe* (kerangka), *flat shading*, *gouraud shading*, dan *phong shading.*
3. Pengguna dapat memilih jenis pencahayaan apa yang akan diterapkan dalam *shading* tersebut, yaitu cahaya ambient, cahaya *diffuse* dan cahaya *specular*.

3.1 Algoritma Program

Berikut ini merupakan langkah-langkah dalam merancang aplikasi untuk memodelkan pencahayaan pada objek 3 dimensi dengan 3 teknik *shading* yaitu *flat shading, gouraud,* dan *phong shading*.

1. Program membuat objek 3 dimensi. Ada 4 objek yang dapat ditampilkan yaitu kubus, tabung, kerucut, bola.
2. Inisialisasi objek 3 dimensi dan jenis shading yang aktif. Saat program mulai objek 3 dimensi yang aktif adalah kubus dan jenis shading yang aktif adalah *flat shading*.
3. Program menampilkan objek 3 dimensi ke layar.
4. Pengguna memilih objek 3 dimensi dan jenis *shading*.

5. Program menghitung intensitas pantulan pada objek 3 dimensi yang aktif dengan metode *shading* yang dipilih.
6. Program menampilkan objek 3 dimensi pada layar.
7. Apabila pengguna kembali memilih objek 3 dimensi atau memilih jenis *shading* maka kembali ke langkah 5.

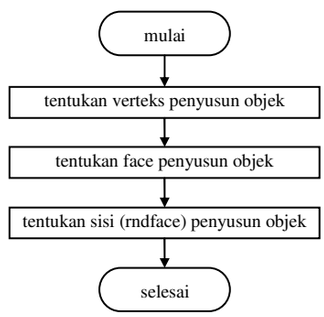


Gambar 3.1 Diagram alir program utama

3.2 Algoritma Buat Objek

Algoritma buat objek ini adalah algoritma yang digunakan untuk memasukkan data-data objek, yaitu verteks, *face*, dan sisi. Ada empat objek yang dipakai dalam tugas akhir ini, yaitu kubus, tabung, kerucut dan bola.

1. Menentukan titik penyusun objek (verteks).
2. Menentukan verteks penyusun poligon yang menyusun objek (face).
3. Menentukan poligon yang menyusun sisi objek (rndface), digunakan dalam *gouraud shading* dan *phong shading*.

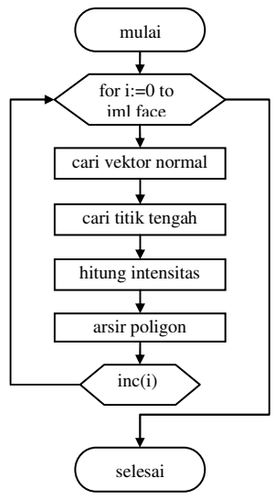


Gambar 3.2 Diagram alir prosedur buat objek

3.3 Algoritma Flat Shading

Algoritma flat shading digunakan untuk menghitung intensitas tiap-tiap poligon pada objek dan menggambarnya ke layar.

1. Lakukan langkah 2 sampai 5 untuk semua poligon dalam objek.
2. Mencari vektor normal dari poligon.
3. Mengambil sembarang titik yang terletak tepat pada poligon tersebut (misalkan titik tengah poligon) sebagai titik acuan.
4. Menghitung intensitas dari poligon tersebut.
5. Kemudian, seluruh poligon tersebut diarsir dengan nilai warna yang telah dicari.



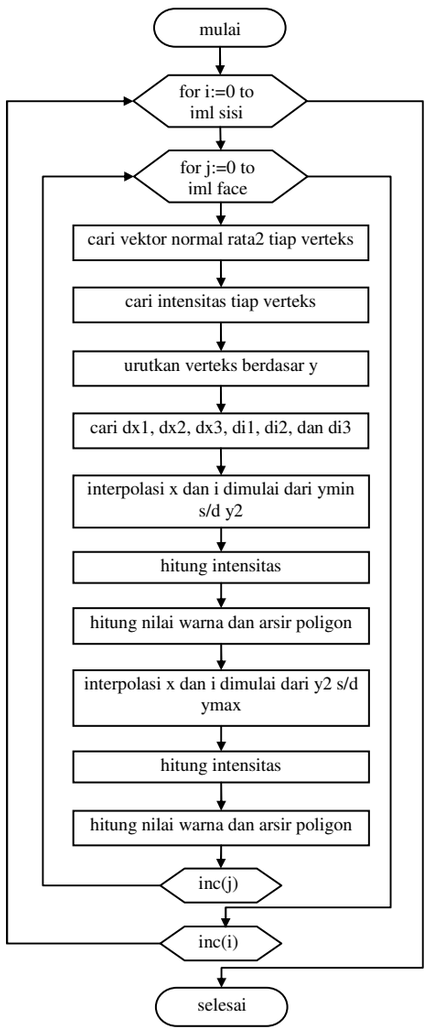
Gambar 3.3 Diagram alir prosedur flat shading

3.4 Algoritma Gouraud Shading

Algoritma *gouraud shading* digunakan untuk mengarsir poligon menggunakan teknik interpolasi. Interpolasi dilakukan pada intensitas poligon dimana intensitas dihitung pada masing-masing verteks pada poligon kemudian diinterpolasi.

1. Urutkan verteks penyusun segitiga berdasarkan y. Hitung vektor normal intensitas tiap-tiap verteks.
2. Cari dx1, dx2, dx3, di1, di2, dan di3. Yaitu, selisih x dan selisih intensitas dibagi selisih y untuk masing-masing pasangan verteks.
3. Interpolasi x dan intensitas terhadap y dimulai dari verteks yang memiliki y paling kecil.
4. Untuk setiap y, warnai titik (x,y) dengan intensitas i. x sama dengan x pada verteks 1 dan bergerak pada y dengan menambahkan dx, dan intensitas ditambah di.
5. Tambahkan y dengan 1, ulangi langkah 4 sampai y sama dengan y pada verteks kedua.
6. Untuk setiap y, warnai titik (x,y) dengan intensitas i. x sama dengan x pada verteks 2 dan bergerak pada y dengan menambahkan dx, dan intensitas ditambah di.

7. Tambahkan y dengan 1, ulangi langkah 6 sampai y sama dengan y pada verteks ketiga.

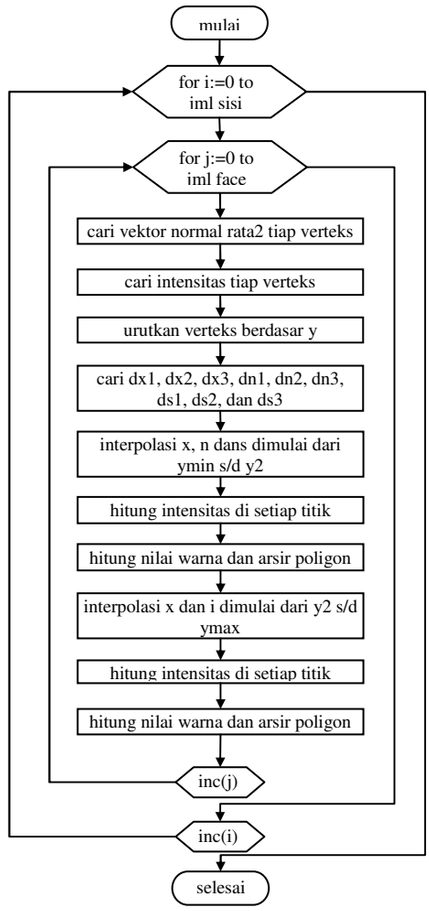


Gambar 3.4 Diagram alir prosedur gouraud shading

3.5 Algoritma Phong Shading

Pada algoritma *phong shading* yang diinterpolasi adalah vektor normalnya, sehingga intensitas dihitung pada masing-masing titik.

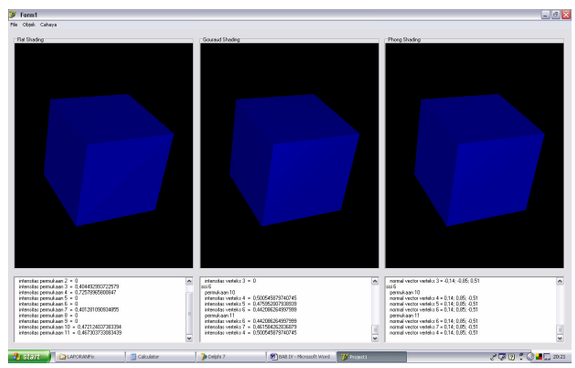
1. Mengurutkan verteks penyusun segitiga berdasarkan y.
2. Mencari dx1, dx2, dx3, dn1, dn2, dn3, ds1, ds2, dan ds3 . Yaitu, selisih x, selisih normal, dan selisih vektor 3d dibagi selisih y untuk masing-masing pasangan verteks.
3. interpolasi x dan intensitas terhadap y dimulai dari verteks yang memiliki y paling kecil.
4. untuk setiap y, warnai titik (x,y) dengan lebih dulu menghitung intensitas dengan n dan s yang diketahui. x sama dengan x pada verteks 1 dan bergerak pada y dengan menambahkan dx, dan normal ditambah dn, serta vektor 3d ditambah ds.
5. tambahkan y dengan 1, ulangi langkah 4 sampai y sama dengan y pada verteks kedua.
6. untuk setiap y, warnai titik (x,y) dengan lebih dulu menghitung intensitas dengan n dan s yang diketahui. x sama dengan x pada verteks 2 dan bergerak pada y dengan menambahkan dx, dan normal ditambah dn, serta vektor 3d ditambah ds.
7. tambahkan y dengan 1, ulangi langkah 6 sampai y sama dengan y pada verteks ketiga.



Gambar 3.3 Diagram alir prosedur phong shading

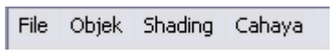
4. Implementasi

Hasil pembuatan aplikasi dapat dilihat pada gambar 4.1-4.6. Gambar 4.1 merupakan tampilan aplikasi grafika komputer untuk pencahayaan dan pengarsiran objek 3 dimensi yang terdiri dari satu form utama.



Gambar 4.1 Tampilan aplikasi grafika komputer untuk pencahayaan dan pengarsiran objek 3 dimensi.

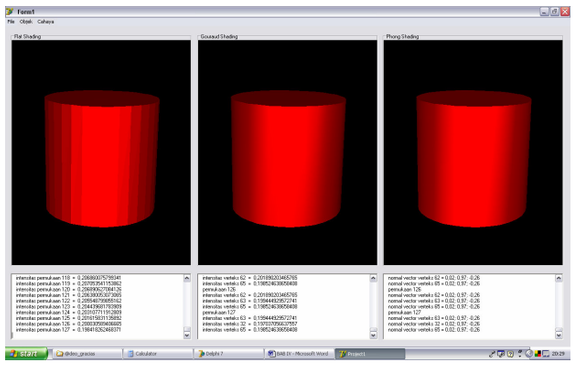
Form utama terdiri dari 2 bagian, yaitu *menubar* dan layar. Ada 4 menu yang terdapat dalam *menubar* yaitu menu File, menu Objek, menu Shading, menu Cahaya.



Gambar 4.2 *Menubar*

Pada menu File terdapat sub menu Exit untuk keluar dari program. Menu Objek merupakan menu untuk memilih objek yang akan ditampilkan. Ada 4 objek yang dapat ditampilkan, yaitu kubus, tabung, kerucut, dan bola. Menu *Shading* digunakan untuk memilih jenis *shading* yang digunakan. Ada 3 jenis *shading* yang dapat digunakan, yaitu *flat shading, gouraud shading* dan *phong shading*. Pada menu *Shading* juga terdapat terdapat pilihan untuk menampilkan objek tanpa *shading* yaitu *wireframe*.

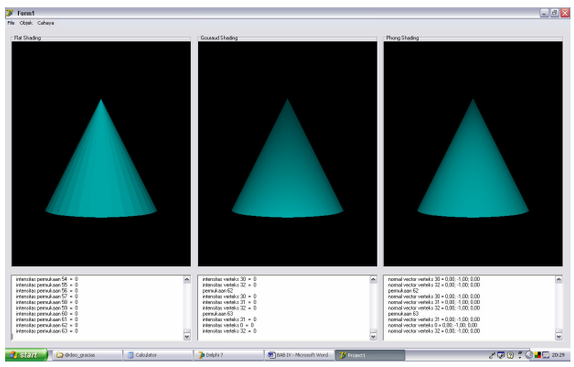
Dari hasil yang ditampilkan program untuk pencahayaan specular pengarsiran dengan menggunakan metode flat shading menghasilkan objek yang paling kasar.



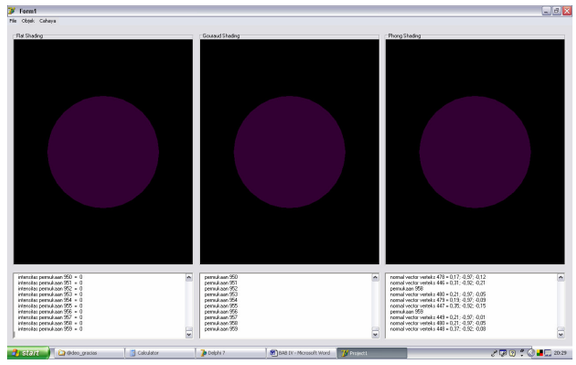
Gambar 4.3 Tampilan aplikasi grafika komputer untuk pencahayaan *specular*.

.

Untuk pengarsiran menggunakan *gouraud shading* batas verteks masih sedikit terlihat, dan hasil terbaik didapatkan saat menggunakan pengarsiran phong shading.



Gambar 4.4 Tampilan aplikasi grafika komputer untuk pencahayaan *diffuse.*



Gambar 4.5 Tampilan aplikasi grafika komputer untuk pencahayaan *ambient.*

Dari hasil yang ditampilkan program untuk pencahayaan diffuse, pengarsiran dengan menggunakan metode flat shading menghasilkan objek yang paling kasar, karena masih terlihat terkotak-kotak. Pengarsiran dengan metode gouraud shading dan *phong shading* menghasilkan hasil yang relatif sama untuk pencahayaan diffuse.

5. Kesimpulan dan Saran

5.1 Kesimpulan

1. Fleksibilitas fungsi-fungsi pengarsiran yang dibuat sendiri lebih tinggi daripada menggunakan fungsi-fungsi bawaan API seperti OpenGL karena OpenGL hanya mendukung *flat shading* dan *gouraud shading* tetapi tidak untuk *phong shading*.
2. Pengarsiran dengan menggunakan metode flat shading menghasilkan objek yang paling kasar untuk semua pencahayaan. Pengarsiran dengan metode *gouraud shading* dan *phong shading* menghasilkan hasil yang relatif sama untuk pencahayaan *diffuse*. Untuk pencahayaan *specular* pengarsiran menggunakan *gouraud shading* batas verteks masih sedikit terlihat.
3. Beban komputasi paling ringan adalah pada *flat shading*, kemudian *gouraud shading* dan yang paling berat adalah *phong shading*.

5.2 Saran

1. Aplikasi dapat dikembangkan lebih lanjut dengan menambahkan fasilitas pengarsiran untuk semua objek baik objek konveks maupun konkaf dengan metode pendeteksian permukaan terlihat yang lebih baik seperti *z-sorting* atau *z-buffer*.
2. Aplikasi dapat dikembangkan untuk pencahayaan global yaitu ray tracing dan radiocity.
3. Aplikasi *dapat* dikembangkan lebih lanjut dengan menambahkan fasilitas proyeksi dan kamera yang lebih baik.

6. Referensi

[1] Nugroho, Edi, *Teori dan Praktek Grafika Komputer Menggunakan Delphi dan OpenGL*, Penerbit Graha ilmu, 2005.

[2] Sanjaya, Dwi, *Bertualang dengan Struktur Data di* Planet *Pascal*, Penerbit ANDI, 2001.

[3] S, Budi Sutedjo, *Algoritma dan Teknik Pemograman*, Penerbit ANDI, 2004.
[4] Andrianto, Heri, *Menguasai Matriks dan Vektor,* Penerbit Rekayasa Sains, 2006.
[5] ---, *Delphi Help*, Borland Delphi Corporation, 2001.
[6] ---, http://www.geocities.com/peter_bone_uk/, Desember 2006
[7] ---, http://escience.anu.edu.au/lecture/cg/Illumination/, Desember 2006
[8] ---, http://www.pcuf.fi/~chem/, Desember 2006.
[9] ---, http://www.nitrogen.za/, Desember 2006.
[10] ---, http://www.cs.fit.edu/~wds/classes/graphics/, Desember 2006.

BIOGRAFI PENULIS

Ike Pertiwi Windasari, lahir di Semarang, 6 Desember 1984. Menempuh pendidikan di SD Mekar Indah Bekasi, SLTP Negeri 9 Bekasi, dan SMU Negeri 3 Semarang. Saat ini sedang menyelesaikan pendidikan program Strata 1 Jurusan Teknik Elektro Universitas Diponegoro dengan mengambil konsentrasi teknik informatika dan komputer. Topik Tugas Akhir yang diambil tentang pencahayaan dan pengarsiran objek 3 dimensi pada grafika komputer.

Menyetujui dan mengesahkan,

Dosen Pembimbing II

Agung B.P., S.T., M.I.T.
NIP.132 137 932
Tanggal……...

Dosen Pembimbing II

Aghus Sofwan, S.T., M.T.
NIP.132 163 757
Tanggal……...