

PENGGUNAAN JAVA 3D API UNTUK TRANSFORMASI DAN PENCAHAYAAN PADA OBJEK 3D

Dian Buana¹, Agung B.P.², Aghus Sofwan²

¹Mahasiswa dan ²Dosen Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, Jl. Prof. Sudharto, Tembalang, Semarang, Indonesia

Abstrak

Perkembangan dunia 3D saat ini semakin meluas dan merambah ke berbagai bidang, mulai dari perfilman, iklan, dan juga permainan. Dua hal mendasar dari penerapan 3D yaitu transformasi, dan pencahayaan. Java 3D API merupakan API yang dikembangkan oleh developer Java untuk pembuatan objek 3D menggunakan bahasa pemrograman java. Cakupan transformasi yang dibahas terdiri atas translasi, rotasi, dan skala. Sedangkan pencahayaan objek mencakup ambient light, directional light, point light, dan spot light. Dari penyinaran langsung ke objek, maka akan terjadi refleksi pencahayaan, yaitu. ambient, diffuse, emissive, dan specular. Tugas akhir ini mencoba mengimplementasikan kelas-kelas dalam Java 3D API untuk transformasi 3 dimensi dan pencahayaan dalam bentuk suatu aplikasi studio. Aplikasi dapat membuat suatu objek yang kemudian dapat ditransformasikan dan diberi efek pencahayaan sesuai dengan fungsi yang dimiliki oleh aplikasi ini. Analisis dan desain aplikasi menggunakan pendekatan berorientasi objek. Aplikasi dibuat menggunakan bahasa pemrograman Java dan Java 3D API (Application Programming Interface).

Kata kunci: transformasi, rotasi, penskalaan, translasi, pencahayaan, ambient, diffuse, emissive, specular, ambient light, directional light, point light, spot light, 3D

1. Pendahuluan

1.1 Latar Belakang Masalah

Perkembangan dunia tiga dimensi (3D) saat ini semakin meluas dan merambah ke berbagai bidang, mulai dari perfilman, iklan, dan juga permainan. Tiga dimensi atau 3D adalah bentuk dari benda yang memiliki panjang, lebar, dan tinggi^[12]. Dua hal mendasar dari penerapan 3D mencakup transformasi, dan pencahayaan.

Transformasi adalah dasar yang amat penting dari animasi. Terdiri atas translasi, rotasi, dan skala, objek 3D dapat digerakan sesuai dengan kehendak yang kita inginkan. Agar lebih terasa hidup, maka diperlukan pencahayaan atas objek-objek tersebut. Adapun macam-macam pencahayaan adalah *spot light*, *directional light*, *point light*, *ambient light*. Pemberian material pada objek juga mendukung penerangan pada objek, material itu antara lain *ambient*, *emissive*, *specular*, dan *diffuse*.

Tugas akhir ini mencoba mengimplementasikan transformasi 3 dimensi dalam bentuk suatu aplikasi studio. Transformasi yang dapat dilakukan, yaitu translasi, penskalaan, dan rotasi. Pertama-tama, aplikasi dimodelkan terlebih dahulu melalui pendekatan berorientasi objek menggunakan UML kemudian aplikasi dibuat menggunakan bahasa pemrograman Java dan Java 3D API.

1.2 Tujuan Tugas Akhir

Tujuan dalam tugas akhir ini adalah mengenalkan Java3D API sebagai media untuk membuat program yang membahas tentang transformasi dan pencahayaan pada objek 3 dimensi.

1.3 Pembatasan Masalah

1. Penerapan transformasi dengan Java 3D API yang terdiri atas translasi, rotasi, dan skala.
2. Penerapan pencahayaan dengan Java 3D API yang terdiri atas *ambient light*, *spot light*, *directional light*, dan *point light*.

3. Membahas penerapan refleksi cahaya dalam Java 3D API yang terdiri atas *ambient*, *diffuse*, *emissive*, dan *specular*.
4. Program hanya dapat menampilkan satu objek berjenis sama.
5. Bahasa pemrograman yang digunakan J2SE dan Java 3D API.
6. Tidak membahas mengenai *shading* yang dipakai dalam program ini.
7. Objek *.obj diperoleh dari hasil konversi file *.3ds dan file *.max menggunakan 3Ds Max 9.0.
8. Tidak membahas *ray tracing*.

2. Pengenalan Java 3D API Untuk Transformasi dan Pencahayaan Pada Objek 3D

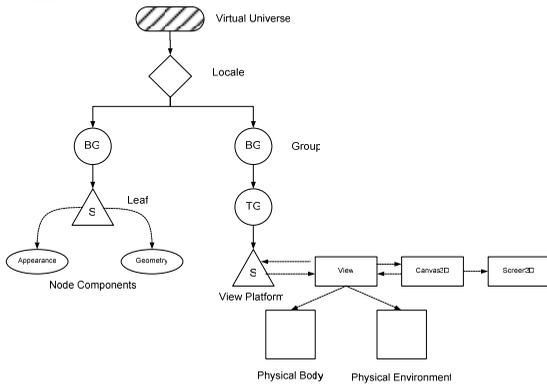
2.1 Pengenalan Java 3D API

Java 3D API adalah hirarki dari kelas-kelas Java yang menyediakan interface untuk render grafis 3D dan suara^[3]. Objek-objek 3D ini ditempatkan di suatu ruang virtual yang disebut *virtual universe*, yang nantinya akan ditampilkan pada user.

Sebuah program Java 3D menciptakan instances dari objek Java 3D dan menempatkannya ke dalam sebuah struktur data *scene graph*. *Scene graph* merupakan susunan dari objek 3D dalam bentuk pohon struktur yang secara menyeluruh menentukan isi dari *virtual universe*^[3], dan bagaimana objek tersebut ditampilkan.

Konstruksi Java 3D API menggunakan hubungan induk dan anak (*parent-child relationship*) yang ditandai dengan garis anak panah solid^[3]. Sebuah *group node* dapat memiliki satu atau lebih anak, namun hanya memiliki satu induk. Sebuah *leaf node* dapat memiliki satu induk dan tidak memiliki anak. Hubungan yang lainnya adalah *reference*, yang ditandai dengan garis anak panah putus-putus, yang menghubungkan *Node-Component object* dengan *scene graph Node*. *Node-component objects* menentukan geometri dan penampilan objek (*appearance*)

yang akan ditampilkan. Ini ditampilkan pada gambar 2.1 berikut ini.



Gambar 2.1. Parent-Child Relationship pada Java 3D API

2.2 Transformasi 3D Pada Java 3D API

Java 3D API telah menyediakan *library* untuk menerapkan transformasi 3D ke suatu objek yang diinginkan.

Jenis-jenis transformasi yang akan dibahas dalam tugas akhir ini adalah translasi, rotasi, dan skala. Berikut akan dijelaskan satu persatu.

2.2.1 Translasi

Transformasi translasi merupakan transformasi yang berfungsi memindahkan suatu objek dari posisi satu ke posisi lainnya. Matriks translasi diperlihatkan pada persamaan 2.1 berikut:

$$M_T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} \tag{2.1}$$

Pada Java 3D API, translasi menggunakan kelas-kelas seperti yang ditunjukkan pada Tabel 2.1 sebagai berikut:

Tabel 2.1 Kelas dan metode dalam Java 3D API yang digunakan dalam operasi translasi

Kelas	Metode yang penulis gunakan
TransformGroup	<ul style="list-style-type: none"> • getTransform(Transform3D t1) • setTransform(Transform3D t1)
Transform3D	<ul style="list-style-type: none"> • get(Vector3f translasi) • setTranslation(Vector3f translasi)
Vector3f	-

2.2.2 Rotasi

Rotasi merupakan transformasi yang memutar objek dengan arah dan besar sudut yang ditentukan. Matriks rotasi searah sumbu x, y, atau z ditunjukkan pada persamaan 2.2, 2.3, dan 2.4¹⁰⁾ berikut ini.

$$M_{R_x(\theta)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$$M_{R_y(\theta)} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$M_{R_z(\theta)} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Kelas-kelas dalam Java 3D API yang digunakan ditampilkan dalam Tabel 2.2 sebagai berikut:

Tabel 2.2 Tabel kelas dan metode Java 3D API yang digunakan dalam operasi rotasi

Kelas	Metode yang penulis gunakan
TransformGroup	<ul style="list-style-type: none"> • getTransform(Transform3D t1) • setTransform(Transform3D t1)
Transform3D	setRotation(AxisAngle4f rotasi)
Vector3f	-
AxisAngle4f	-

2.2.3 Skala

Penskalaan yang dibahas terdiri atas penskalaan *uniform* dan *nonuniform*.

- Uniform

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \tag{2.5}$$

- Nonuniform

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \tag{2.6}$$

Pada Java 3D implementasi skala dapat dilakukan dengan menggunakan kelas-kelas yang ditampilkan pada Tabel 2.3 sebagai berikut:

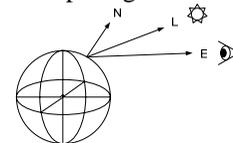
Tabel 2.3 Kelas dan metode Java 3D API untuk implementasi operasi penskalaan

Kelas	Metode yang penulis gunakan
TransformGroup	<ul style="list-style-type: none"> • getTransform(Transform3D t1) • setTransform(Transform3D t1)
Transform3D	setScale(float skalaUniform) setScale(Vector3d skalaNonUniform)
Vector3d	-

2.3 Cahaya dan Material

Dalam dunia nyata, warna yang kita ketahui adalah kombinasi dari sifat fisik dari suatu objek, karakteristik dari sumber cahaya, posisi objek terhadap sumber cahaya, dan sudut dari mana objek dilihat³⁾.

Persamaan dari model cahaya tergantung pada tiga vektor, yaitu Normal permukaan bidang (N), arah cahaya (L), dan sudut pandang mata terhadap bidang atau objek (E) seperti yang diperlihatkan pada gambar 2.4 berikut.



Gambar 2.4 Normal, arah cahaya, dan sudut pandang terhadap benda

2.3.1 Material

Jenis-jenis material yang dibahas dalam tugas akhir ini adalah ambient, diffuse, specular, dan emissive.

2.3.1.1 Ambient

Ambient merupakan refleksi cahaya dengan intensitas yang rendah. Ambient memberikan cahaya dengan

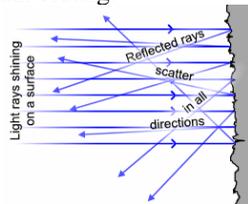
intensitas yang sama pada semua lokasi dan pada semua arah^[3].

Pada kelas Material terdapat metode untuk mengatur nilai dari ambient itu sendiri, yaitu :

```
void setAmbientColor(Color3f color)
void setAmbientColor(float r, float g, float b)
```

2.3.1.2 Diffuse

Diffuse adalah Diffuse adalah refleksi cahaya yang terjadi terhadap permukaan bidang yang kasar^[13]. Pantulan cahaya memiliki arah pantulan yang tidak menentu disepanjang permukaan bidang.



Gambar 2.5 Arah pantulan cahaya yang tidak menentu pada *diffuse reflection*

Pada kelas Material terdapat metode untuk mengatur nilai dari diffuse itu sendiri, yaitu :

```
void setDiffuseColor(Color3f color)
void setDiffuseColor(float r, float g, float b)
```

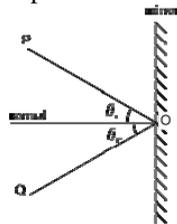
2.3.1.3 Emissive

Emissive memberikan efek pada benda seperti berpendar dalam gelap (*glow in the dark*)^[3]. Pada Material terdapat metode untuk mengatur nilai dari emissive yaitu :

```
void setEmissiveColor(Color3f color)
void setEmissiveColor(float r, float g, float b)
```

2.3.1.4 Specular

Specular adalah refleksi cahaya dengan intensitas yang tinggi dari sumber cahaya terhadap benda tertentu^[3]. Refleksi cahaya ini biasanya terjadi pada permukaan yang rata. Refleksi cahaya ini dapat bersifat seperti efek cermin.



Gambar 2.6 Pantulan cahaya pada sebuah cermin

Seperti terlihat pada gambar 2.6, refleksi specular terjadi dari satu arah datang cahaya dan dipantulkan juga satu arah dengan besar sudut pantul terhadap normal bidang yang sama dengan sudut datang cahaya.

Pada class Material pengaturan nilai dari specular ini diatur oleh metode :

```
void setSpecularColor(Color3f color)
void setSpecularColor(float r, float g, float b)
```

2.3.2 Cahaya

Pada tugas akhir ini akan dibahas macam-macam dari cahaya itu sendiri, seperti *ambient light*, *directional light*, *point light*, dan *spot light*.

2.3.2.1 Ambient Light

Ambient light akan memberikan cahaya dengan intensitas yang sama pada semua lokasi dan arah^[3]. Contoh dari ambient ini adalah bila anda melihat ke bawah meja anda, anda akan melihat permukaan bawah meja anda

meskipun tidak ada cahaya yang langsung menyinarinya.

Contoh dari ambient light terlihat pada gambar 2.7 berikut:



Gambar 2.7 *Ambient light* dengan warna merah pada kubus

Kelas-kelas dalam Java 3D yang digunakan dalam implementasi cahaya ambient diperlihatkan pada tabel 2.4 berikut ini:

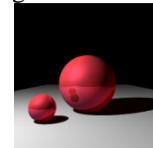
Tabel 2.4 Kelas-kelas Java 3D yang berguna dalam implementasi ambient light

Kelas	Konstruksi dasar
AmbientLight	<ul style="list-style-type: none"> AmbientLight() AmbientLight(boolean lightOn, Color3f color)
Color3f	<ul style="list-style-type: none"> Color3f() Color3f(float red, float green, float blue)

2.3.2.2 Directional Light

Directional light merupakan cahaya dengan sumber cahaya sangat jauh, mempunyai satu arah sinar cahaya, intensitas cahayanya tidak melemah. Contoh *directional light* adalah matahari.

Contoh dari penerapan *directional light* pada Java 3D API dapat dilihat pada gambar 2.8.



Gambar 2.8 Penerapan *directional light* pada kubus

Kelas-kelas dalam Java 3D yang digunakan dalam implementasi cahaya directional diperlihatkan pada tabel 2.9 berikut ini:

Tabel 2.9 Kelas-kelas Java 3D yang berguna dalam implementasi cahaya directional

Kelas	Konstruksi dasar
DirectionalLight	<ul style="list-style-type: none"> DirectionalLight() DirectionalLight(boolean lightOn, Color3f color, Vector3f direction)
Color3f	<ul style="list-style-type: none"> Color3f() Color3f(float red, float green, float blue)
Vector3f	<ul style="list-style-type: none"> Vector3f() Vector3f(float x, float y, float z)

2.3.2.3 Point Light

Point light merupakan lawan dari *directional light*. Bila intensitas sinar dari *directional light* tidak terpengaruh oleh jarak, maka intensitas sinar dari *point light* berpengaruh terhadap jarak dan lokasi (*directional light* tidak memiliki lokasi, hanya arah)^[3]. Contoh dari *point light* adalah cahaya dari lilin.

Besar intensitas cahaya dari point light ditentukan oleh persamaan 2.7^[10] berikut ini :

$$C = \frac{C_0}{k_c + k_l d + k_q d^2} \tag{2.7}$$

Dengan :

C = Intensitas cahaya

C_0 = Warna cahaya

k_c = Konstanta constant

k_l = Konstanta linear

k_q = Konstanta quadratic

d = Jarak antara sumber cahaya dengan objek

Contoh dari *point light* pada program Java 3D API diperlihatkan pada gambar 2.9 berikut ini :



Gambar 2.9 *Point light* pada kubus

Kelas-kelas dalam Java 3D yang digunakan dalam implementasi point light diperlihatkan pada tabel 2.10 berikut ini:

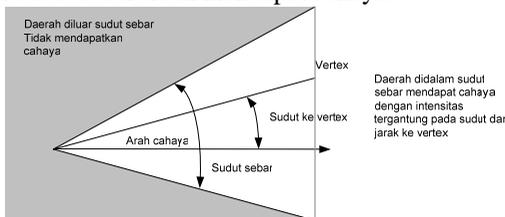
Tabel 2.10 Kelas-kelas Java 3D yang berguna dalam implementasi point light

Kelas	Konstruksi dasar
PointLight	<ul style="list-style-type: none"> PointLight() PointLight(boolean lightOn, Color3f color, Point3f position, point3f attenuation)
Color3f	<ul style="list-style-type: none"> Color3f() Color3f(float red, float green, float blue)
Point3f	<ul style="list-style-type: none"> Point3f() Point3f(float x, float y, float z)

2.3.2.4 Spot Light

Kelas *SpotLight* dalam Java 3D API adalah *subclass* dari *PointLight*^[3]. kelas *SpotLight* menambahkan arah radiasi cahaya dan konsentrasi pada parameter posisi dan attenuation dari *point light*. Contoh *spot light* adalah lampu senter.

Intensitas cahaya ditentukan dari sudut sebar dan arah cahaya itu datang. Hanya daerah yang berada didalam sudut sebar yang terdapat cahaya, sedangkan daerah diluar dari vertex sudut sebar tidak mendapat cahaya.



Gambar 2.10 Intensitas cahaya pada *spot light*

Besar intensitas cahaya dari spot light ditentukan oleh persamaan 2.8^[10] berikut ini.

$$C = \frac{\max\{-R, L, 0\}^p}{k_c + k_l d + k_q d^2} C_0 \tag{2.8}$$

$$L = \frac{P \cdot Q}{\|P - Q\|} \tag{2.9}$$

Dengan :

P = Vektor pusat cahaya

Q = Posisi objek

R = Vektor arah radiasi

p = Konstanta konsentrasi cahaya

C = Intensitas cahaya

C_0 = Warna cahaya

k_c = Konstanta constant

k_l = Konstanta linear

k_q = Konstanta quadratic

d = Jarak antara sumber cahaya dengan objek

Kelas-kelas dalam Java 3D yang digunakan dalam implementasi point light diperlihatkan pada tabel 2.11 berikut ini:

Tabel 2.11 Kelas-kelas Java 3D yang berguna dalam implementasi spot light

Kelas	Konstruksi dasar
SpotLight	<ul style="list-style-type: none"> SpotLight() SpotLight(boolean lightOn, Color3f color, Point3f position, Point3f attenuation, Vector3f direction, float sudutSebar, float concentration)
Color3f	<ul style="list-style-type: none"> Color3f() Color3f(float red, float green, float blue)
Point3f	<ul style="list-style-type: none"> Point3f() Point3f(float x, float y, float z)
Vector3f	<ul style="list-style-type: none"> Vector3f() Vector3f(float x, float y, float z)

Contoh dari *SpotLight* pada program Java 3D API



Gambar 2.11 *SpotLight* dengan warna ungu pada kubus

3. PERANCANGAN APLIKASI KOMPUTER GRAFIS

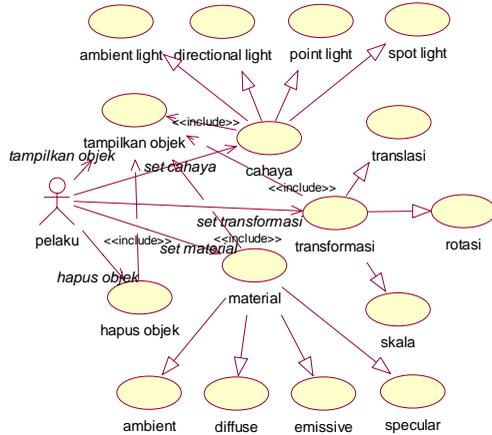
Pemodelan aplikasi grafika komputer untuk transformasi objek 3 dimensi menggunakan pendekatan berorientasi objek (*objek oriented analysis and design*). Langkah-langkah yang dilakukan seperti model yang diutarakan oleh Pressman^[6], yaitu:

1. *Use case*,
 2. Kartu Index *CRC*,
 3. Diagram Kelas,
 4. Model Hubungan Objek (*Object Relationship*), dan
 5. Model Tingkah Laku Objek (*Object Behaviour*).
- Langkah-langkah tersebut kemudian dibuat dalam bentuk diagram UML seperti berikut.

3.1. Use Case

Aplikasi yang akan dibangun paling sedikit memiliki 5

use case utama, yaitu membuat objek, menghapus objek, mengubah material, pemberian cahaya, dan mentransformasi objek 3D. Use case transformasi sendiri merupakan generalisasi dari use case translasi, penskalaan, dan rotasi. Diagramnya dapat dilihat pada gambar 3.1.



Gambar 3.1 Use case dari aplikasi komputer grafis

3.2. Kartu Index CRC

Dari use case pada subbab 3.1, aplikasi paling sedikit membutuhkan 4 buah kelas utama, yaitu kelas *ShapeGeometry*, kelas Cahaya, kelas *MaterialObj*, dan kelas transformasi. Kelas *ShapeGeometry* kemudian diturunkan sehingga terbentuk anak kelas objek kubus, kerucut, bola, dan *LoadObject*.

Tabel 3.1a. Kartu index CRC kelas *ShapeGeometry*

Kelas: <i>ShapeGeometry</i>	
Tanggung Jawab:	Kolaborator:
Atribut: <ul style="list-style-type: none"> width height depth bg Operasi: <ul style="list-style-type: none"> createGeom 	<ul style="list-style-type: none"> -

Tabel 3.1b. Kartu index CRC kelas Kubus

Kelas: Kubus	
Tanggung Jawab:	Kolaborator:
Atribut: <ul style="list-style-type: none"> mat app shininess Operasi: <ul style="list-style-type: none"> createGeom 	<ul style="list-style-type: none"> -

Tabel 3.1c. Kartu index CRC kelas Kerucut

Kelas: Kerucut	
Tanggung Jawab:	Kolaborator:
Atribut: <ul style="list-style-type: none"> radius mat app shininess Operasi: <ul style="list-style-type: none"> createGeom 	<ul style="list-style-type: none"> -

Tabel 3.1d. Kartu index CRC kelas Bola

Kelas: Bola	
Tanggung Jawab:	Kolaborator:
Atribut: <ul style="list-style-type: none"> radius mat 	<ul style="list-style-type: none"> -

<ul style="list-style-type: none"> app shininess Operasi: <ul style="list-style-type: none"> createGeom 	
---	--

Tabel 3.1e. Kartu index CRC kelas *LoadObject*

Kelas: <i>LoadObject</i>	
Tanggung Jawab:	Kolaborator:
Atribut: <ul style="list-style-type: none"> fullModelName fileName mat app shininess Operasi: <ul style="list-style-type: none"> createGeom 	<ul style="list-style-type: none"> -

Untuk kelas Transformasi, menurunkan anak kelas *TransformBox*, *TransformCone*, *TransformSphere*, dan *TransformLoad*.

Tabel 3.2a. Kartu index CRC kelas Transformasi

Kelas: Transformasi	
Tanggung Jawab:	Kolaborator:
Atribut: <ul style="list-style-type: none"> TG translation rotation scale objTranslation objNUScale objUScale objRotateAxis objRotateNAxis objRefPt objRotateAngle objRotateAxisAngle useCompoundTransform useUniformScale tmpTrans tmpAxisAngle tmpVector objScale origin yAxis Operasi: <ul style="list-style-type: none"> setObjTranslation setObjUScale setObjNUScale setObjRotation updateUsingCompoundTransform normalizeObjRotateAxis updateObjAxisAngle 	<ul style="list-style-type: none"> ShapeGeometry

Tabel 3.2b. Kartu index CRC kelas *TransformBox*

Kelas: <i>TransformBox</i>	
Tanggung Jawab:	Kolaborator:
Atribut: <ul style="list-style-type: none"> kubus Operasi: <ul style="list-style-type: none"> transformBox updateUsingCompoundTransform 	<ul style="list-style-type: none"> Kubus

Tabel 3.2c. Kartu index CRC kelas *TransformCone*

Kelas: <i>TransformCone</i>	
Tanggung Jawab:	Kolaborator:
Atribut: <ul style="list-style-type: none"> kerucut Operasi: <ul style="list-style-type: none"> transformCone updateUsingCompoundTransform 	<ul style="list-style-type: none"> Kerucut

Tabel 3.2d. Kartu index CRC kelas *TransformSphere*

Kelas: TransformSphere	
Tanggung Jawab:	Kolaborator:
Atribut:	<ul style="list-style-type: none"> • Bola
Operasi:	
<ul style="list-style-type: none"> • transformSphere • updateUsingCompoundTransform 	

Tabel 3.2e. Kartu index CRC kelas TransformLoad

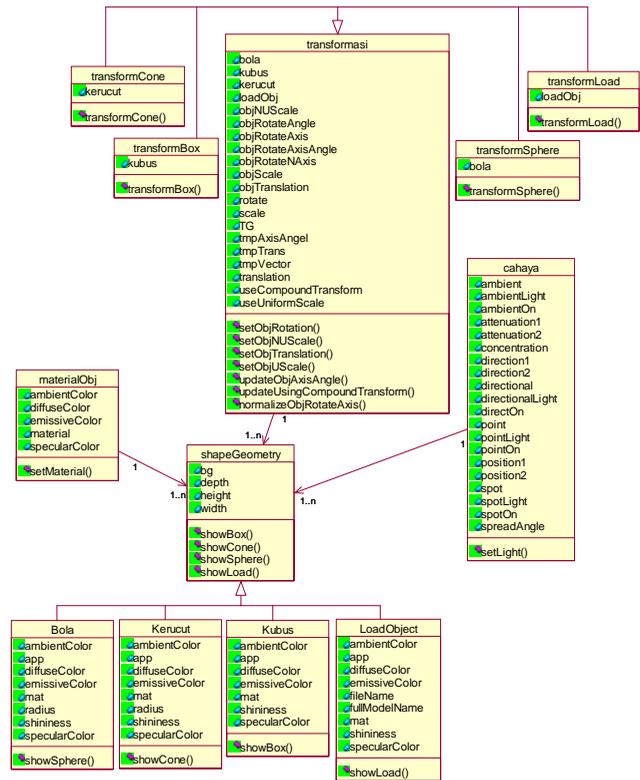
Kelas: TransformLoad	
Tanggung Jawab:	Kolaborator:
Atribut:	<ul style="list-style-type: none"> • LoadObject
Operasi:	
<ul style="list-style-type: none"> • loadObj • fullModelName • fileName • • transformLoad • updateUsingCompoundTransform 	

Tabel 3.3 Kartu index CRC KomputerGrafis untuk cahaya

Kelas: Cahaya	
Tanggung Jawab:	Kolaborator:
Atribut:	<ul style="list-style-type: none"> • KomputerGrafis
Operasi:	
<ul style="list-style-type: none"> • ambient • directional • point • spot • direction1 • direction2 • attenuation1 • attenuation2 • position1 • position2 • concentration • Angle • spreadAngle • ambientLight • directionalLight • pointLight • spotLight • setLight 	

Tabel 3.4 Kartu index CRC untuk kelas MaterialObj

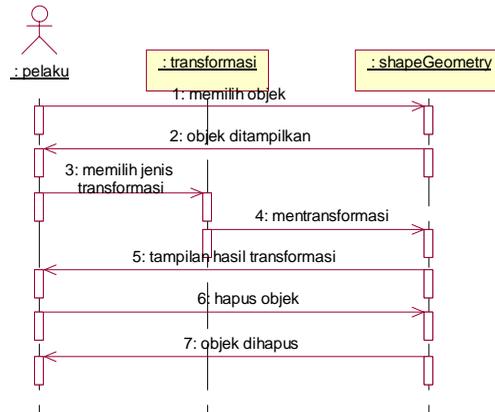
Kelas: MaterialObj	
Tanggung Jawab:	Kolaborator:
Atribut:	<ul style="list-style-type: none"> • Kubus • Kerucut • Bola • LoadObject
Operasi:	
<ul style="list-style-type: none"> • setMaterial 	



Gambar 3.2 Diagram kelas aplikasi

3.4. Model Hubungan Objek (Object Relationship)

Model hubungan objek (*object relationship model*) dapat digambarkan dengan menggunakan diagram runtun (*sequence diagram*). Untuk transformasi objek diperlihatkan pada gambar 3.3, dimana terdapat 3 skenario, yaitu menampilkan objek geometri, mentransformasi objek geometri, dan menghapus objek geometri.

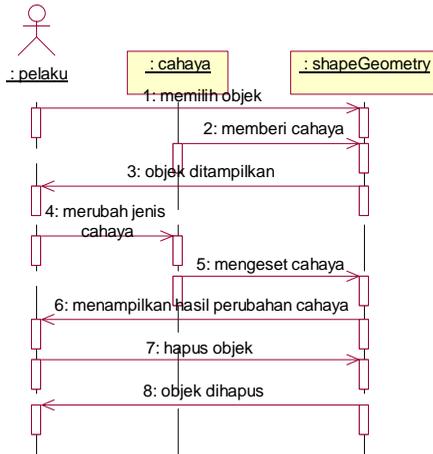


Gambar 3.3 Diagram runtun aplikasi grafika komputer untuk transformasi objek 3 dimensi

3.3. Diagram Kelas

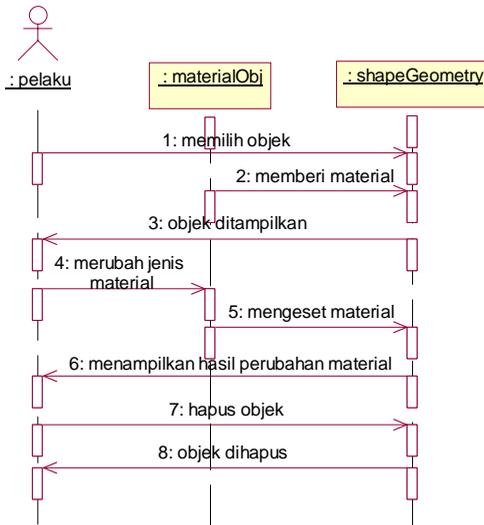
Diagram kelas (gambar 3.2) dibuat dari kartu index CRC pada subbab 3.2. Diagram kelas memberikan gambaran tentang kelas-kelas apa saja yang perlu dibuat untuk membangun aplikasi, lengkap dengan atribut dan operasinya. Pada gambar 3.2, terdapat kelas Material, kelas Cahaya, kelas ShapeGeometry dengan 4 kelas turunan, yaitu kubus, kerucut, bola, dan LoadObject. Kelas transformasi juga memiliki 4 turunan, yaitu TransformBox, TransformCone, TransformSphere, dan TransformLoad, yang masing-masing memiliki operasi-operasi untuk melakukan transformasi 3 dimensi.

Untuk pengesetan cahaya pada objek terlihat pada gambar 3.4 dimana juga terdapat 3 skenario, yaitu menampilkan objek geometri, mengeset cahaya pada objek geometri, dan menghapus objek geometri.



Gambar 3.4 Diagram runtun aplikasi grafika komputer untuk pencahayaan objek 3 dimensi

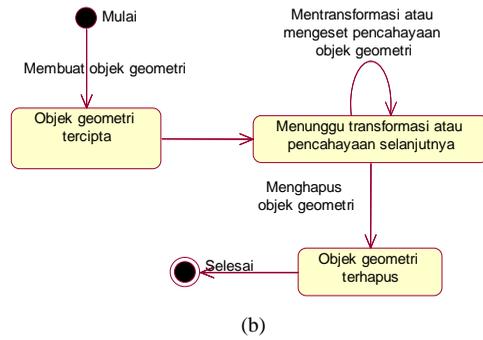
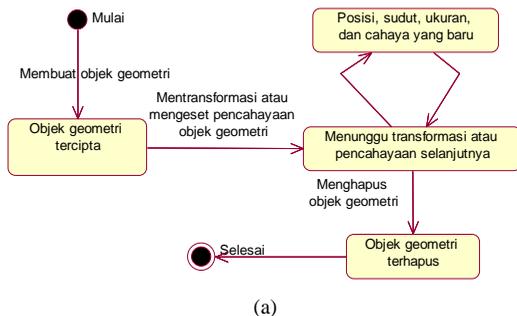
Untuk pengesetan material pada objek terlihat pada gambar 3.5 dimana juga terdapat 3 skenario, yaitu menampilkan objek geometri, mengeset material pada objek geometri, dan menghapus objek geometri.



Gambar 3.5 Diagram runtun aplikasi grafika komputer untuk material objek 3 dimensi

3.5. Model Tingkah Laku Objek (Object Behaviour)

Gambar 3.6 merupakan diagram statechart yang menggambarkan perilaku 2 objek, yaitu objek geometri (a) dan objek model (b). Model tingkah laku objek menunjukkan bagaimana sistem akan merespon kejadian atau stimulus eksternal.

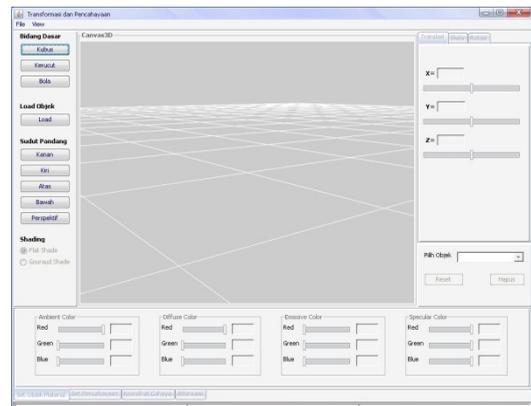


Gambar 3.6 Diagram statechart aplikasi komputer grafis
a) Diagram statechart objek geometri
b) Diagram statechart objek model

4. IMPLEMENTASI DAN PENGUJIAN APLIKASI KOMPUTER GRAFIS UNTUK TRANSFORMASI DAN PENCAHAYAAN 3 DIMENSI

4.1 Implementasi

Hasil pembuatan aplikasi dapat dilihat pada gambar 4.1-4.7. Gambar 4.1 merupakan tampilan dari aplikasi Komputer Grafis.



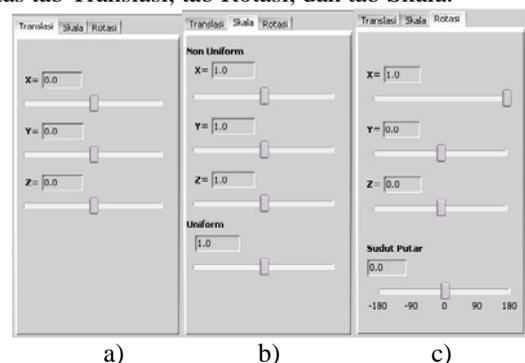
Gambar 4.1. Tampilan aplikasi Komputer Grafis

Gambar 4.2 adalah tampilan dari menubar beserta item-item-nya.



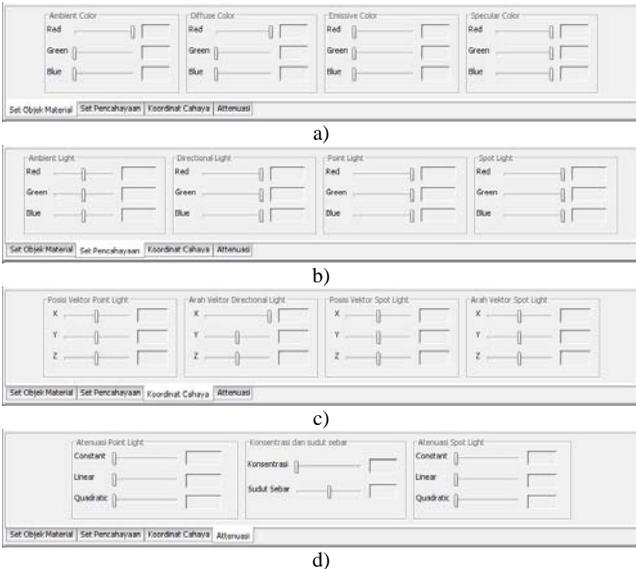
Gambar 4.2 Tampilan menubar pada aplikasi Komputer Grafis.

Tab transformasi diperlihatkan pada gambar 4.3, terletak disebelah kanan dari form aplikasi. Tab-tab tersebut terdiri atas tab Translasi, tab Rotasi, dan tab Skala.



Gambar 4.3 Tab-tab pada aplikasi komputer grafis
a) tab Translasi
b) tab Skala
c) tab Rotasi

Tab-tab pencahayaan (gambar 4.4) terletak pada sisi bawah form aplikasi. Tab-tab tersebut meliputi tab Set Objek Material, tab Set Pencahayaan, tab Koordinat Cahaya, dan tab Attenuasi.



Gambar 4.4 a) tab Set Objek Material b) tab Set Pencahayaan c) tab Koordinat Cahaya d) tab Attenuasi

4.2. Pengujian Aplikasi

4.2.1 Pengujian Transformasi

Pengujian untuk transformasi dan pencahayaan dilakukan dengan metode *black box*, dimana pengujian hanya mengetahui masukan dan melihat keluarannya apakah sudah sesuai dengan keluaran yang diharapkan atau belum.

Untuk pengujian transformasi meliputi translasi, rotasi dan skala.

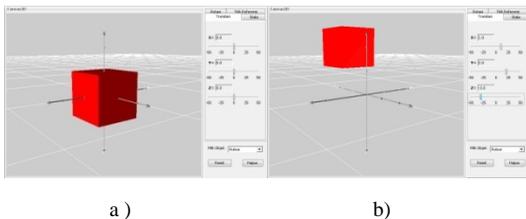
a. Translasi

Masukan berupa dari Tabel 4.1 berikut ini:

Tabel 4.1 Perubahan nilai parameter pada kelas-kelas yang dipakai dalam operasi translasi

Kelas	Nama objek kelas	Nilai awal	Nilai akhir
Vector3f	objTranslation	x=0.0, y=0.0, z=0.0	x=1.0, y=2.0, z=-3.0

Hasil pengujian dari tabel diatas diperlihatkan pada gambar 4.5 berikut ini:



Gambar 4.5 Tampilan kubus berdasarkan Tabel 4.1

- a) kubus dengan nilai awal
- b) kubus dengan nilai akhir

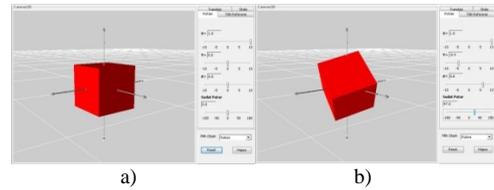
b. Rotasi

Masukan berupa dari Tabel 4.2 berikut ini:

Tabel 4.2 Perubahan nilai parameter pada kelas-kelas yang dipakai dalam operasi rotasi

Kelas/Type	Nama objek kelas	Nilai awal	Nilai akhir
Vector3f	objRotateAxis	x=0.0, y=0.0, z=0.0	x=1.0, y=-0.4, z=0.6
float	objRotateAngle	0 = 0°	47 = 47°
AxisAngle4f	objRotateAxisAngle	x=0.0, y=0.0, z=0.0	x=1.0, y=-0.4, z=0.6
	objRotateAngle	0 = 0°	47 = 47°

Hasil pengujian dari tabel diatas diperlihatkan pada gambar 4.6 berikut:



Gambar 4.6 kubus dengan macam-macam rotasi berdasarkan tabel 4.2

- a) Kubus koordinat awal (0.0, 0.0, 0.0)
- b) Kubus sumbu putar x=1.0, y=-0.4, dan z=0.6

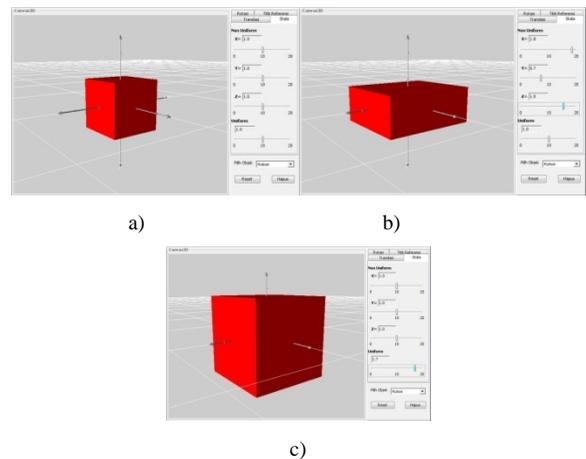
c. Skala

Masukan berupa dari Tabel 4.3 berikut ini:

Tabel 4.3 Perubahan nilai parameter pada kelas-kelas yang dipakai dalam operasi rotasi

Kelas/Type	Nama objek kelas	Nilai awal	Nilai akhir
Vector3d	objNUScale	x=0.0, y=0.0, z=0.0	x=1.8, y=0.7, z=1.5
float	objScale	s=1.0	s=1.7

Hasil pengujian dari tabel 4.3 diatas adalah sebagai berikut :



Gambar 4.7 Kubus dengan macam-macam penskalaan berdasarkan Tabel 4.3

- a) Kubus pada skala normal
- b) Kubus pada skala non-uniform x=1.8x, y=0.7x, z=1.5x
- c) Kubus pada skala uniform pembesaran sebesar 1.7X

4.2.2 Pengujian Material

Pengujian material menggunakan metode *black box*, dengan model berupa objek kerucut. Pengujian material mencakup ambient, diffuse, emissive, dan specular.

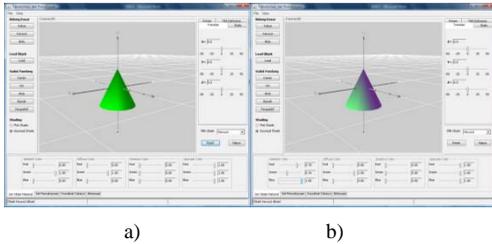
a. Ambient

Masukan berupa tabel 4.4 berikut ini :

Tabel 4.4 Perubahan nilai-nilai parameter dari kelas-kelas dalam material ambient

Kelas	Nama objek kelas	Nilai awal	Nilai akhir
Color3f	ambientColor	R=0.00, G=1.00, B=0.00	R=0.78, G=0.34, B=1.00

Hasil perubahan dari tabel diatas dapat terlihat pada gambar 4.8 berikut :



Gambar 4.8 Kerucut dengan perubahan warna ambient berdasarkan Tabel 4.4

- a) Kerucut dengan nilai awal
- b) Kerucut dengan nilai akhir

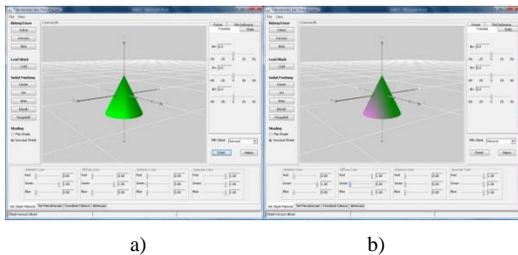
b. Diffuse

Masukan berasal dari tabel 4.5 sebagai berikut :

Tabel 4.5 Perubahan nilai-nilai parameter dari kelas-kelas dalam material diffuse

Kelas	Nama objek kelas	Nilai awal	Nilai akhir
Color3f	diffuseColor	R=0.00, G=1.00, B=0.00	R=1.00, G=0.00, B=1.00

Hasil dari pengujian terlihat pada gambar 4.11 sebagai berikut:



Gambar 4.9 Kerucut dengan perubahan warna diffuse berdasarkan Tabel 4.5

- a) Kerucut dengan nilai awal
- b) Kerucut dengan nilai akhir

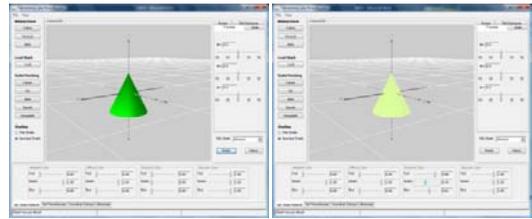
c. Emissive

Pengujian menggunakan tabel 4.6 sebagai berikut :

Tabel 4.6 Perubahan nilai-nilai parameter dari kelas-kelas dalam material emissive

Kelas	Nama objek kelas	Nilai awal	Nilai akhir
Color3f	emissiveColor	R=0.00, G=0.00, B=0.00	R=0.86, G=0.46, B=0.61

Hasil pengujian terlihat pada gambar 4.10 sebagai berikut :



Gambar 4.10 Kerucut dengan perubahan warna emissive berdasarkan Tabel 4.6

- a) Kerucut dengan nilai awal
- b) Kerucut dengan nilai akhir

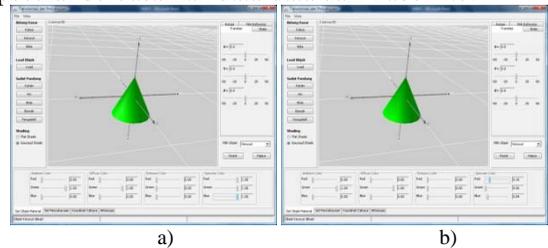
d. Specular

Pengujian didasarkan pada tabel 4.7 berikut ini :

Tabel 4.7 Perubahan nilai-nilai parameter dari kelas-kelas dalam material specular

Kelas	Nama objek kelas	Nilai awal	Nilai akhir
Color3f	specularColor	R=1.00, G=1.00, B=1.00	R=0.18, G=0.00, B=0.09

Berikut adalah hasil pengujian dari refleksi specular berdasarkan data-data dari Tabel 4.7.



Gambar 4.11 Kerucut dengan perubahan warna specular berdasarkan Tabel 4.7

- a) Kerucut dengan nilai awal
- b) Kerucut dengan nilai akhir

4.2.3 Pengujian Pencahayaan

Pengujian pencahayaan menggunakan metode *black box* dengan mengambil model berupa objek bola. Pengujian pencahayaan mencakup jenis-jenis cahaya berikut karakteristiknya, yaitu *ambient light*, *directional light*, *point light*, dan *spot light*.

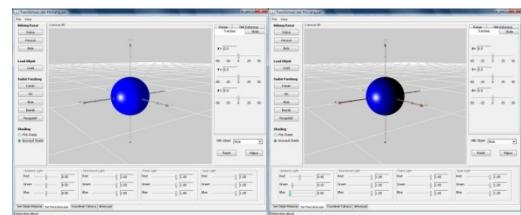
a. Ambient Light

Pengujian berdasarkan pada tabel 4.8 sebagai berikut:

Tabel 4.8 Perubahan nilai-nilai parameter dari kelas-kelas dalam cahaya ambient

Kelas	Nama objek kelas	Nilai awal	Nilai akhir
Color3f	ambient	R=0.50, G=0.50, B=0.50	R=0.40, G=0.10, B=0.00

Hasil pengujian cahaya ambient diperlihatkan pada gambar 4.12 sebagai berikut:



Gambar 4.12 pengujian warna cahaya ambient pada objek bola

- a) Warna awal dari cahaya ambient R=0.5, G=0.5, B=0.5
- b) Warna cahaya ambient setelah interaksi dari pengguna R=0.4, G=0.1, B=0.0

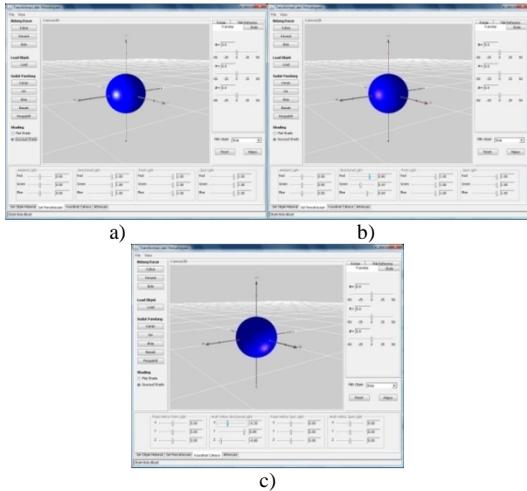
b. Directional Light

Pengujian berdasarkan pada tabel 4.9 sebagai berikut:

Tabel 4.9 Perubahan nilai-nilai parameter dari kelas-kelas dalam cahaya directional

Kelas	Nama objek kelas	Nilai awal	Nilai akhir
Color3f	directional	R=1.00, G=1.00, B=1.00	R=0.40, G=0.10, B=0.00
Vector3f	direction1	x=1.00, y=0.00, z=0.00	x=-0.30, y=0.89, z=-0.80

Hasil dari pengujian terlihat pada gambar 4.13 sebagai berikut :



Gambar 4.13 Perubahan karakteristik *directional light* pada objek bola

- a) *Directional light* awal
- b) Perubahan pada warna dengan R = 0.80, G = 0.42, B = 0.69
- c) Perubahan arah cahaya dengan x = -0.30, y = 0.89, dan z = -0.80

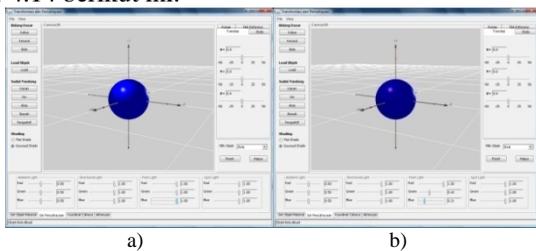
c. Point Light

Masukan berupa tabel 4.10 sebagai berikut :

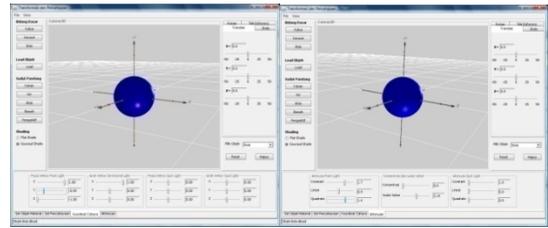
Tabel 4.10 Perubahan nilai-nilai parameter dari kelas-kelas dalam point light

Kelas	Nama objek kelas	Nilai awal	Nilai akhir
Color3f	point	R=1.00, G=1.00, B=1.00	R=1.00, G=0.42, B=0.21
Pointf	position1	x=0.00, y=0.00, z=0.00	x=1.00, y=-0.54, z=-1.00
	attenuation1	c=1.0, l=0.0, q=0.0	c=1.7, l=0.9, q=1.4

Hasil pengujian point light pada bola terlihat pada gambar 4.14 berikut ini.



a) b)



c) d)

Gambar 4.14 bola dengan point light

- a) Point light pada keadaan awal
- b) Perubahan pada warna dengan R = 1.00, G = 0.42, B = 0.21
- c) Perubahan pada posisi dengan x = 1.00, y = -0.54, dan z = -1.00
- d) Perubahan pada koefisien atenuasi, constant = 1.7, linear = 0.9, dan quadratic = 1.4

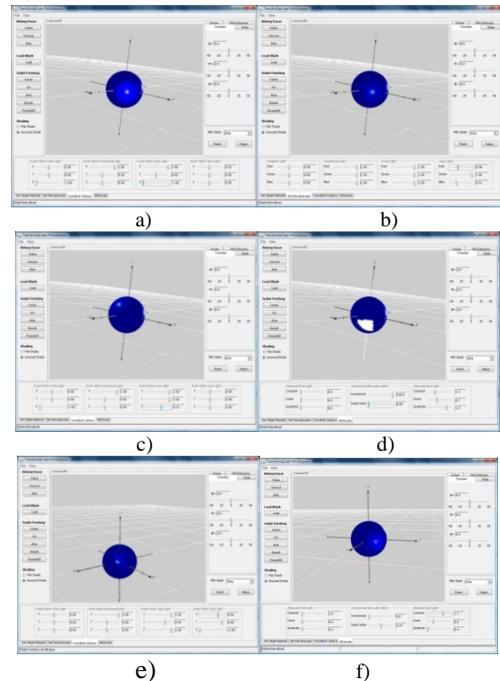
d. Spot Light

Pengujian dilakukan dengan mengacu pada tabel 4.11 sebagai berikut :

Tabel 4.11 Perubahan nilai-nilai parameter dari kelas-kelas dalam spot light

Kelas/Tipe	Nama objek kelas	Nilai awal	Nilai akhir
Color3f	spot	R=1.00, G=1.00, B=1.00	R=0.38, G=1.00, B=0.31
Pointf	position2	x=0.00, y=0.00, z=0.00	x=1.00, y=1.00, z=0.33
	attenuation2	c=1.0, l=0.0, q=0.0	c=1.7, l=0.5, q=0.0
Vector3f	direction2	x=-9.13, y= 2.61, z=-1.74	x=-9.13, y= 2.61, z=-16.96
float	concentration	c=0.0	c=128.0
	spreadAngle	0 rad	0 rad

Hasil pengujian dapat dilihat pada gambar 4.15 berikut ini :



e) f)

Gambar 4.15. Spot light pada bola dengan ragam perubahan parameter

- a) Spot light keadaan awal
- b) Spot light dengan warna R=0.38, G=1.00, dan B=0.30
- c) Spot light dengan posisi x=1.00, y=1.00, dan z=0.33
- d) Spot light dengan konsentrasi = 128.0, dan sudut sebar = 0.00

- e) Spot light dengan arah $x=0.50$, $y=0.80$, dan $z=-1.00$
 f) Spot light dengan attenuasi constant = 1.7, linear = 0.5, dan quadratic = 0.0

5. Kesimpulan dan Saran

Kesimpulan

1. Java 3D API memberikan efektifitas dalam perancangan dan pembuatan program 3 dimensi. Fungsi-fungsi transformasi dan pencahayaan yang tersedia sangat lengkap dan mudah untuk diterapkan, sehingga dapat menyingkat waktu pembuatan program.
2. Translasi pada objek tiga dimensi memiliki tiga parameter, yaitu x, y, dan z. Translasi terhadap sumbu x mengakibatkan objek berpindah posisi searah sumbu x. Translasi terhadap sumbu y mengakibatkan objek berpindah posisi searah sumbu y, dan translasi terhadap sumbu z mengakibatkan objek berpindah tempat searah sumbu z.
3. Skala secara *uniform* mengakibatkan semua parameter dari objek 3D berubah dengan besar perubahan yang sama. Sedangkan skala *nonuniform* mengakibatkan objek berubah ukuran dengan besar perubahan parameter dapat sama ataupun berbeda nilainya.
4. Warna yang dilihat oleh mata merupakan jumlah spektrum warna cahaya yang dipantulkan oleh objek.

Saran

1. Aplikasi dapat dikembangkan lebih lanjut dengan menambahkan fasilitas *texture mapping* dan efek bayangan sehingga objek memiliki tekstur yang membuat tampilannya menjadi lebih realistis.
2. Aplikasi dapat dikembangkan lebih lanjut dengan menambahkan fasilitas untuk memuat objek geometri dari file objek dengan tipe 3 dimensi yang telah ada, misalnya .3ds, .lws, .wrl, dll sehingga objek yang dapat ditampilkan lebih bervariasi. Adapun salah satu caranya adalah dengan mengimport *loader-loader* tambahan yang mendukung tipe-tipe file tersebut. *Loader-loader* tersebut dapat didownload dari internet.
3. Aplikasi dapat dikembangkan agar dapat menampilkan lebih dari satu objek bertipe sama, misalkan dapat menampilkan lebih dari satu objek bola pada layar.

6. Referensi

- [1] Adjie, B, *Modeling dan Animasi dengan 3D Studio Max 7.x*, Elex Media Komputindo, Jakarta, 2005.
- [2] Boughen, N, *3ds Max Lighting*, Wordware Publishing, 2005.
- [3] Bouvier, D. J, *Getting Started with the Java 3D API*, Sun microsystems, 1999.
- [4] Chandra, H, *Polygonal dan NURBS Modeling 3ds max 6 & 7*, Maxicom, Palembang, 2005.
- [5] Clingman, D, K. Shawn, S. Mesdaghi, *Practical Java Game Programming*, Charles River Media, 2004.
- [6] Daniel, S, *Java 3D Programming*, Manning Publishing, 2003.

- [7] Deitel, H, *Java How To Program*, Prentice-Hall, 1999.
- [8] Djalle, Z. G, *3D Animation Movies Using 3DStudioMax*, Informatika, Bandung, 2006.
- [9] Eric, L, *Mathematics for 3D Game Programming and Computer Graphics 2nd edition*, Charles River Media, Massachusetts, 2004.
- [10] Paul, S, *Modelling a Character in 3ds max*, Wordware Publishing, 2005.
- [11] Pressman, R.S., *Rekayasa Perangkat Lunak Pendekatan Praktisi*, Buku Satu dan Dua, McGraw-Hil Book Co., Penerbit ANDI, 2003.
- [12] Wijono, M. S, *Java 2 SE dengan JBuilder*, Penerbit Andi, Yogyakarta, 2005.
- [13] ---, *3D graphics*, http://www.webopedia.com/TERM/3/3D_graphics.html, Juni 2007
- [14] ---, *Cahaya*, <http://en.wikipedia.org/wiki/Light>, Mei 2007
- [15] ---, *Light type*, <http://msdn2.microsoft.com/en-us/library/bb174694.aspx>, Maret 2007
- [16] ---, *Tiga dimensi*, http://id.wikipedia.org/wiki/3_dimensi, Mei 2007
- [17] ---, *Transform*, <http://msdn2.microsoft.com/en-us/library/bb206269.aspx>, Maret 2007

BIOGRAFI PENULIS



Dian Buana Setyo Putro, lahir di Yogyakarta, 1 Januari 1984. Menempuh pendidikan di SDN Duren Seribu IV Sawangan, SLTP Negeri 4 Bogor, dan SMU Negeri 2 Bogor. 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 transformasi dan pencahayaan objek 3 dimensi pada komputer grafis.

Menyetujui dan mengesahkan,

Dosen Pembimbing I

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.....