

# Transformasi Geometri II & 3 Dimensi

Pertemuan 7

Yonathan Ferry Hendrawan  
Universitas Trunojoyo Madura

**TIU:**  
Mahasiswa mampu menghasilkan aplikasi Komputer Grafik sederhana

Transformasi dan animasi 3D

Viewing dan shading 3D

Pemrograman OpenGL API dasar dan interaksi 2 dimensi

Penggunaan aplikasi pengolah grafis 3D

Konsep-konsep mendasar sistem grafis di komputer

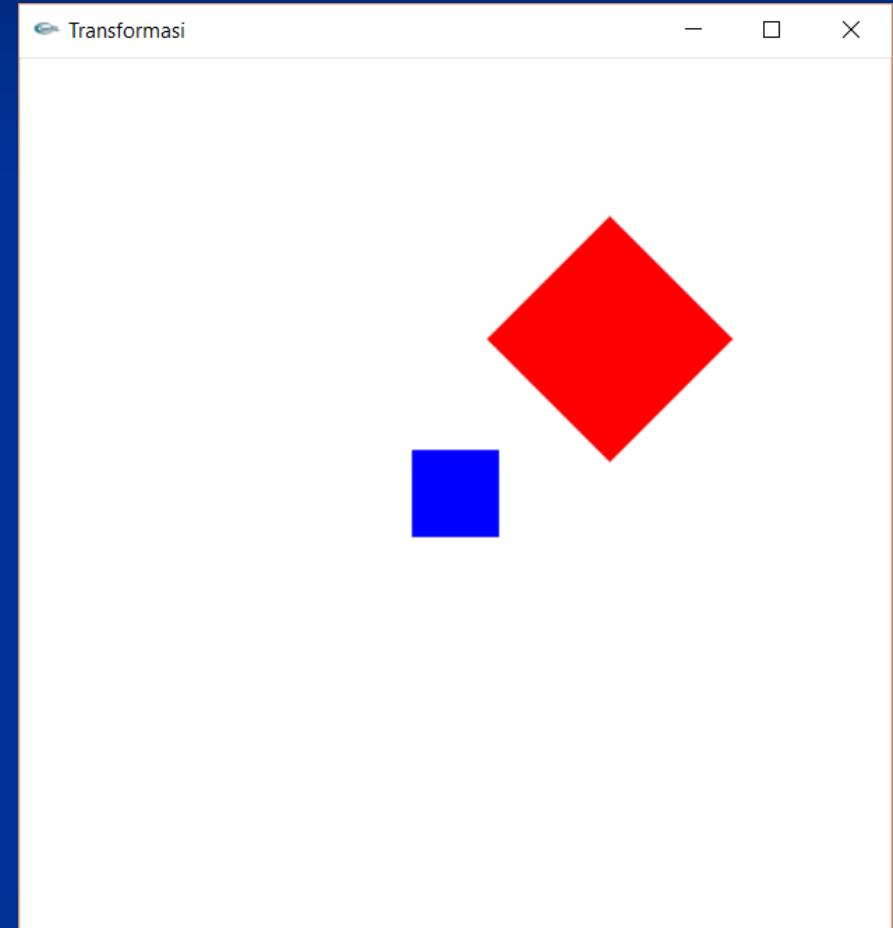
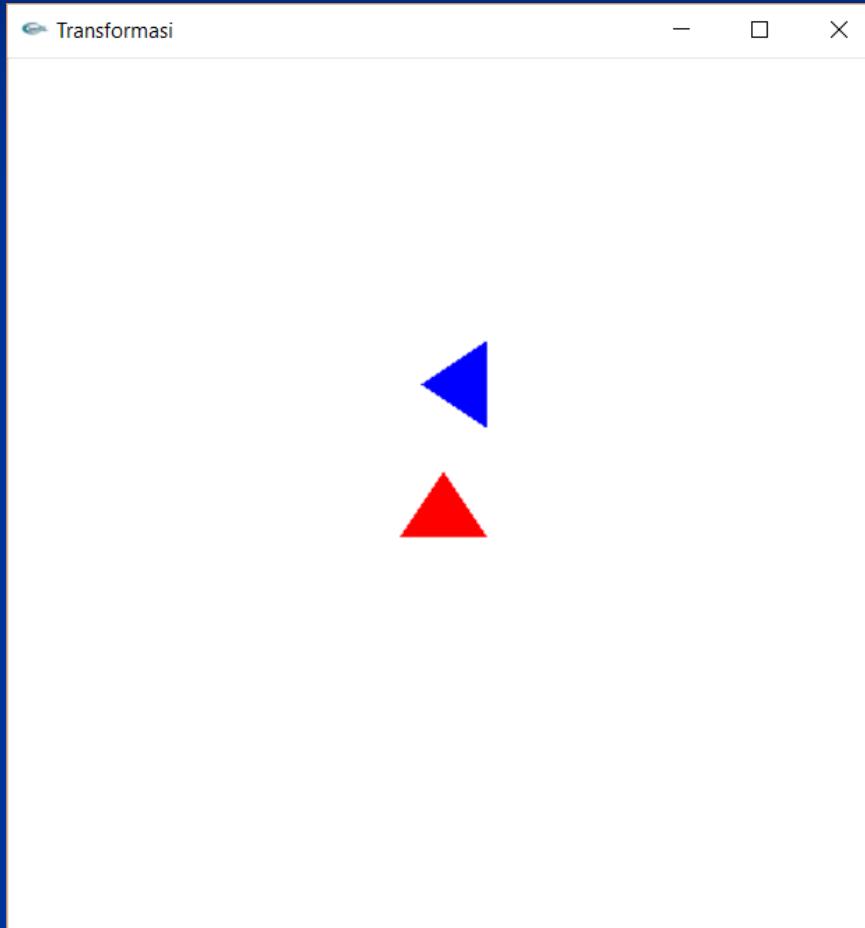
# Bahasan

- Pokok: Transformasi Geometri II & 3 dimensi
- Sub:
  - Rotasi pada sembarang titik
  - Current Transformation Matrix
  - Inverse Transformasi
  - glLoadIdentity
  - glPushMatrix & glPopMatrix
  - 3 dimensi: canvas vs viewing volume
  - 4 setting openGL untuk 3 dimensi
  - glutWire\*
  - glutSolid\*

# Bahasan

- Pokok: Transformasi Geometri II & 3 dimensi
- Sub:
  - Bangun bervolume
  - Kubus manual

# Latihan

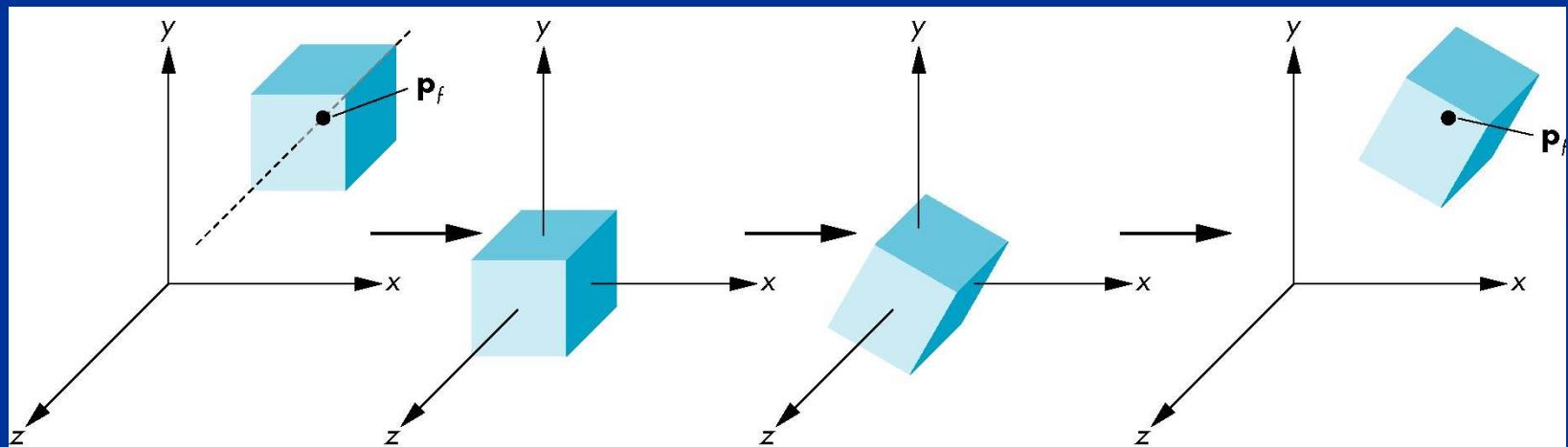


# Rotasi pada titik sembarang

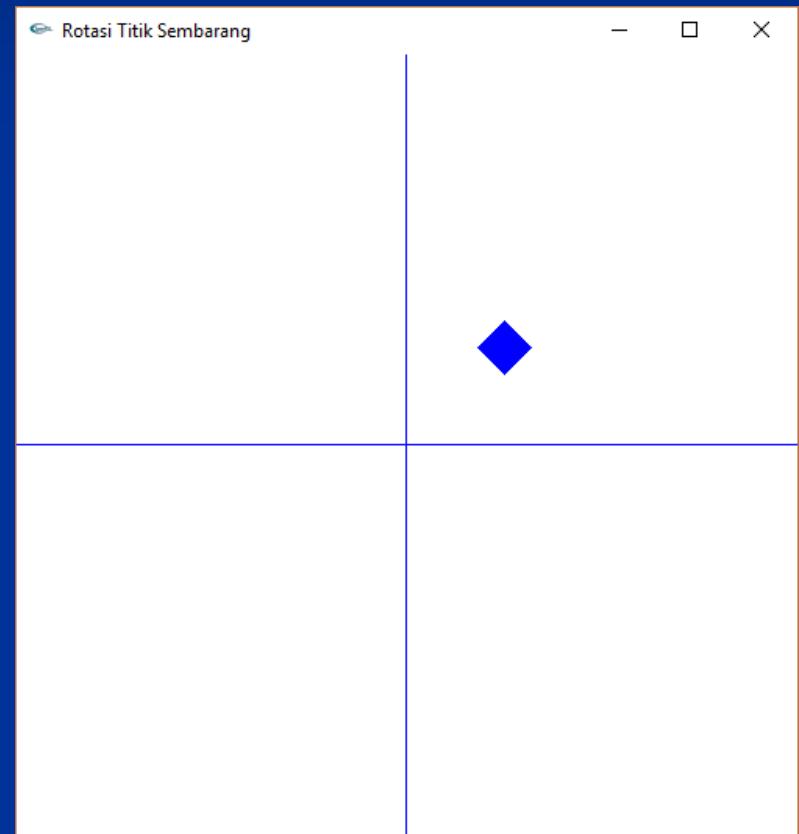
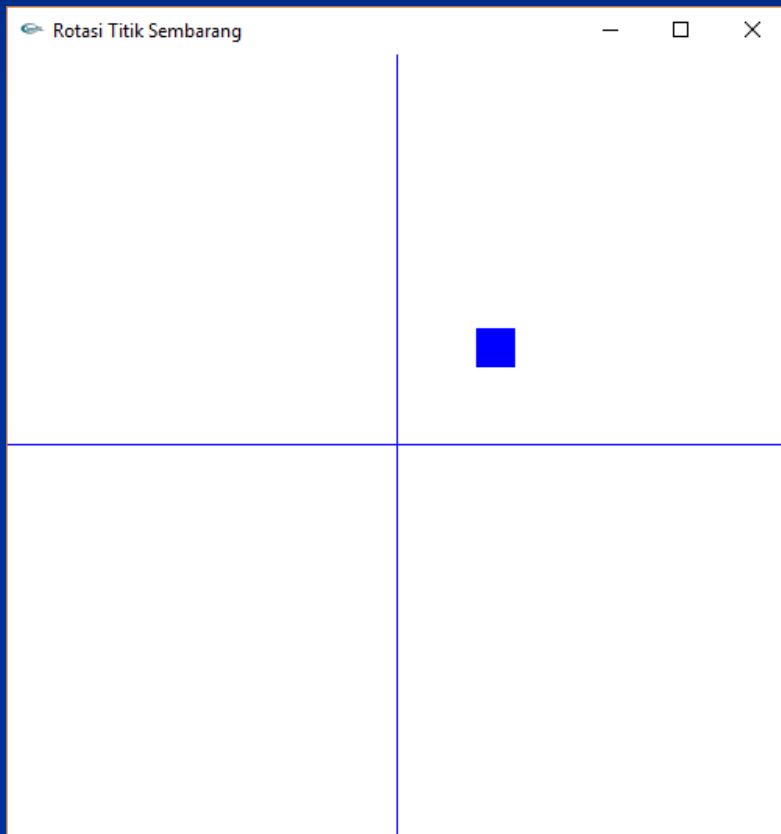
- Transformasi Rotasi dan Scale bergantung pada sumbu koordinat.
- Jika ingin merotasi atau men-scale obyek terhadap sembarang titik, maka yang perlu dilakukan adalah:
  - Mentranslasikan obyek ke sumbu koordinat
  - Melakukan rotasi/scale
  - Mentranslasikan obyek balik ke posisinya semula

# Rotasi pada titik sembarang

$$\mathbf{M} = \mathbf{T}(p_f) \mathbf{R}(\theta) \mathbf{T}(-p_f)$$



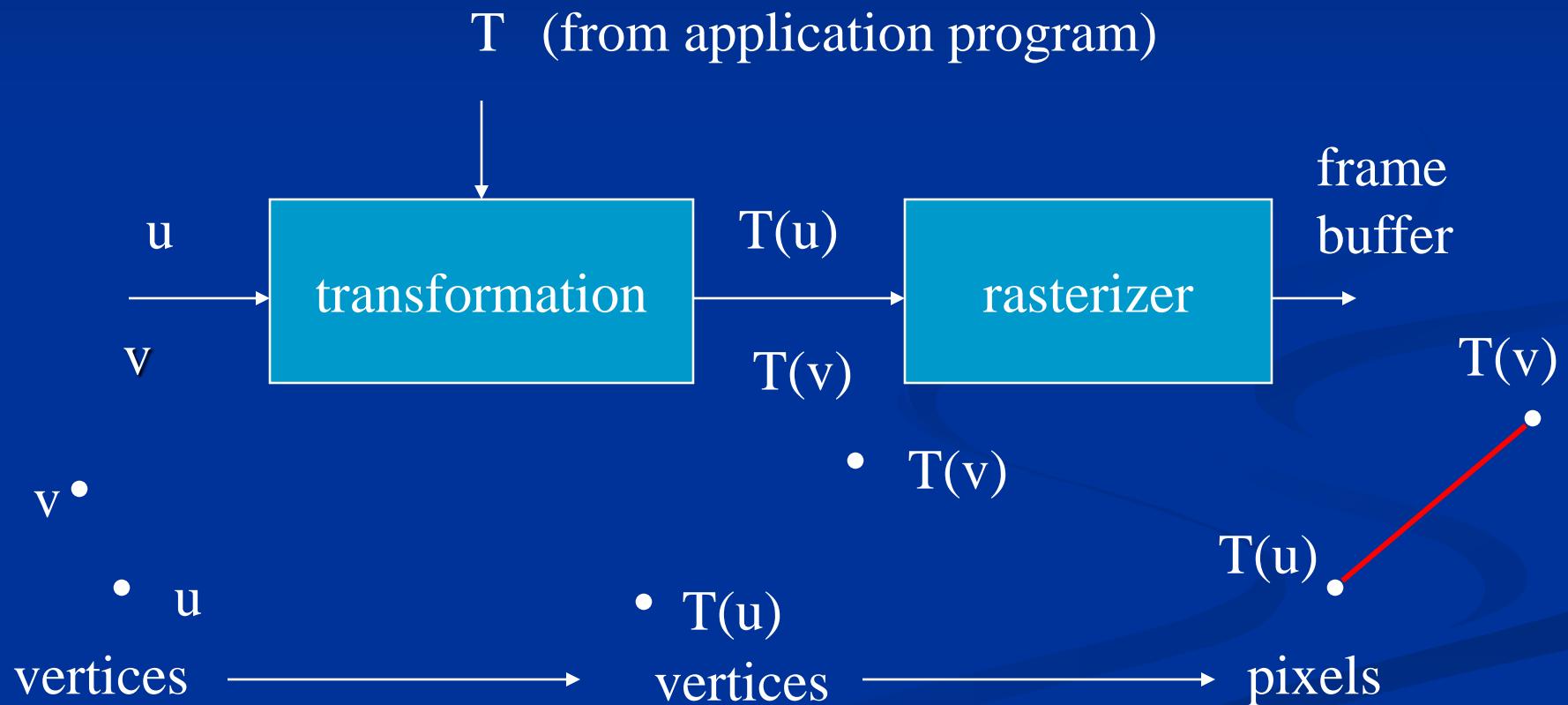
# Latihan



# Current Transformation Matrix

- Matrik yang menyimpan nilai transformasi dan diterapkan untuk semua vertex
- OpenGL punya cara untuk memanipulasi nilai CTM secara langsung
- Tapi yang diajarkan di kelas ini: untuk memanipulasi CTM, gunakan transformasi

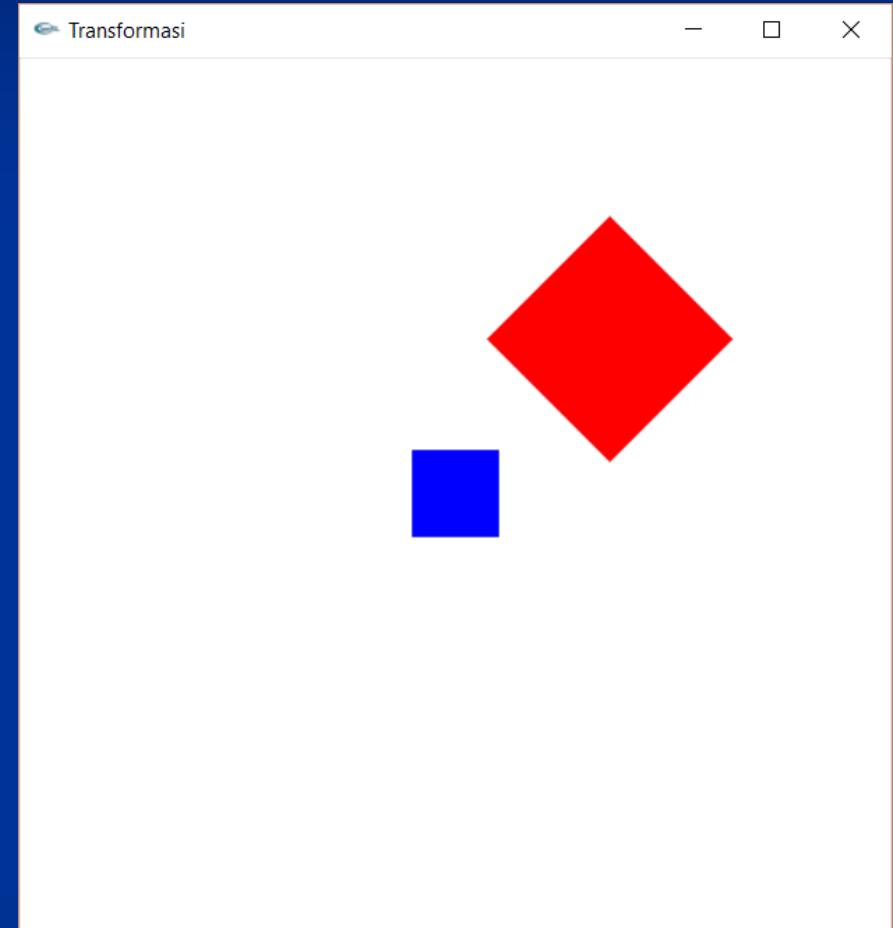
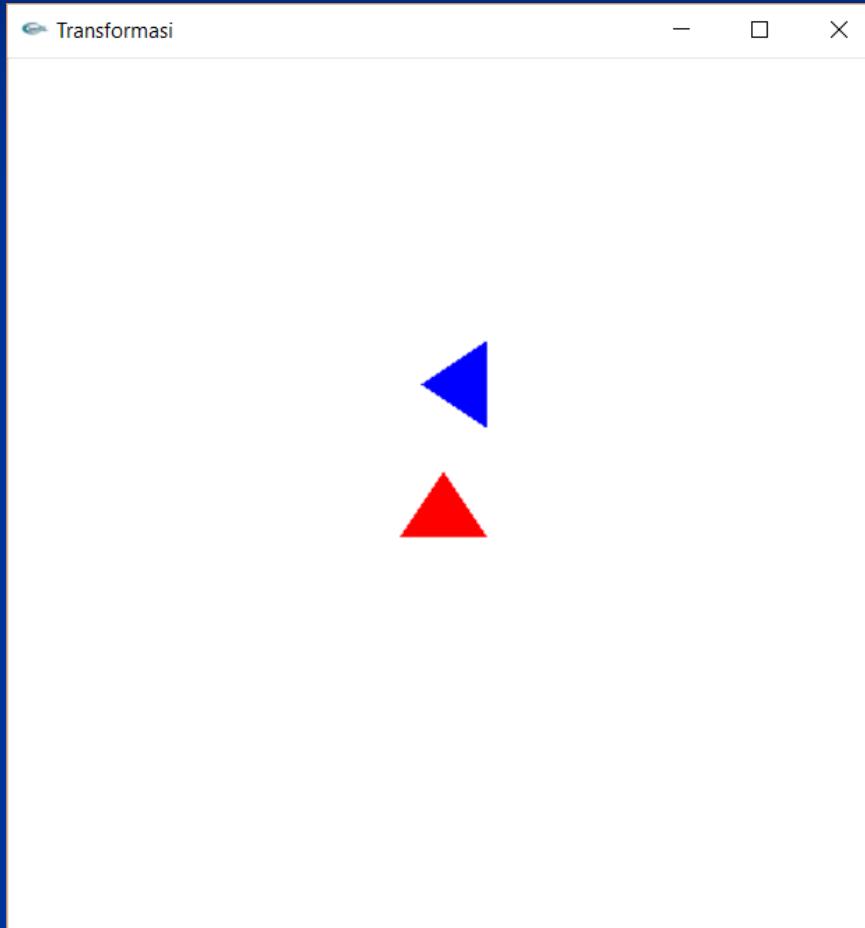
# Implementasi pada pipeline



# Inverse Transformasi

- Tiap transformasi, nilai CTM berubah
- Perubahan ini berpengaruh ke semua vertex di bawahnya
- Untuk mengembalikan nilai CTM ke nilai sebelumnya: balik transformasinya
  - Translate +  $\rightarrow$  Translate -
  - Translate -  $\rightarrow$  Translate +
  - Rotate +  $\rightarrow$  Rotate -
  - Rotate -  $\rightarrow$  Rotate +
  - Scale m  $\rightarrow$  Scale 1/m

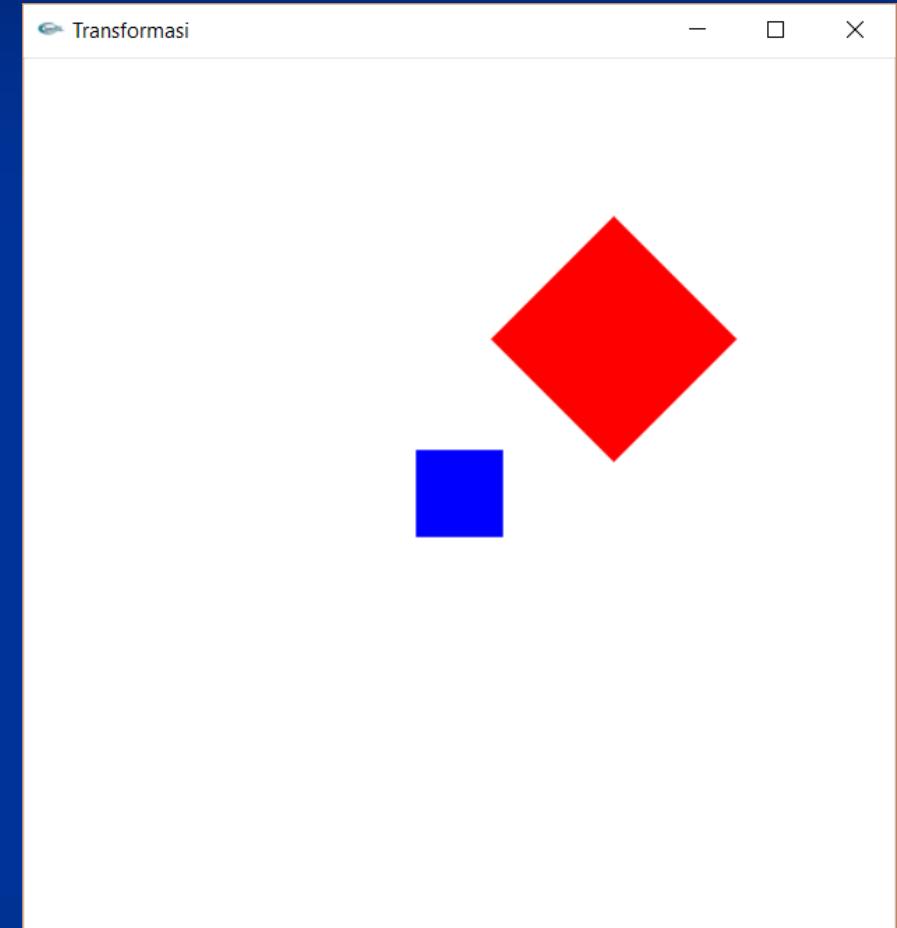
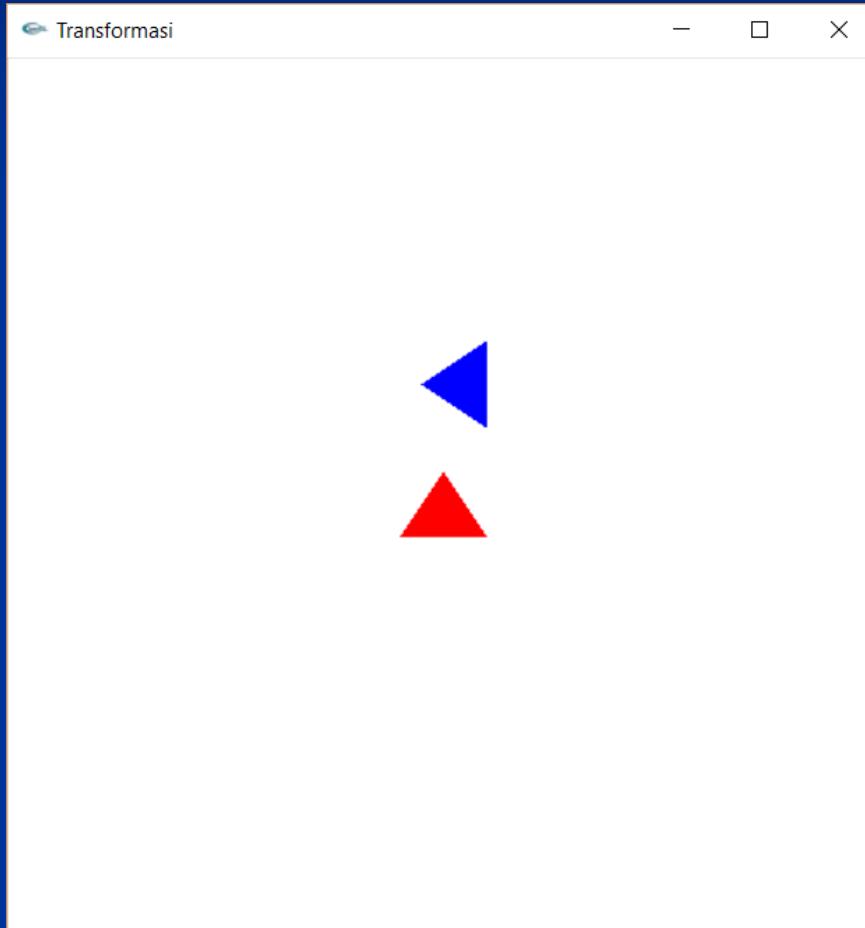
# Latihan



# glLoadIdentity

- Jika kita hendak menetralkan efek transformasi (memberi nilai identitas pada CTM), maka kita dapat menggunakan perintah glLoadIdentity

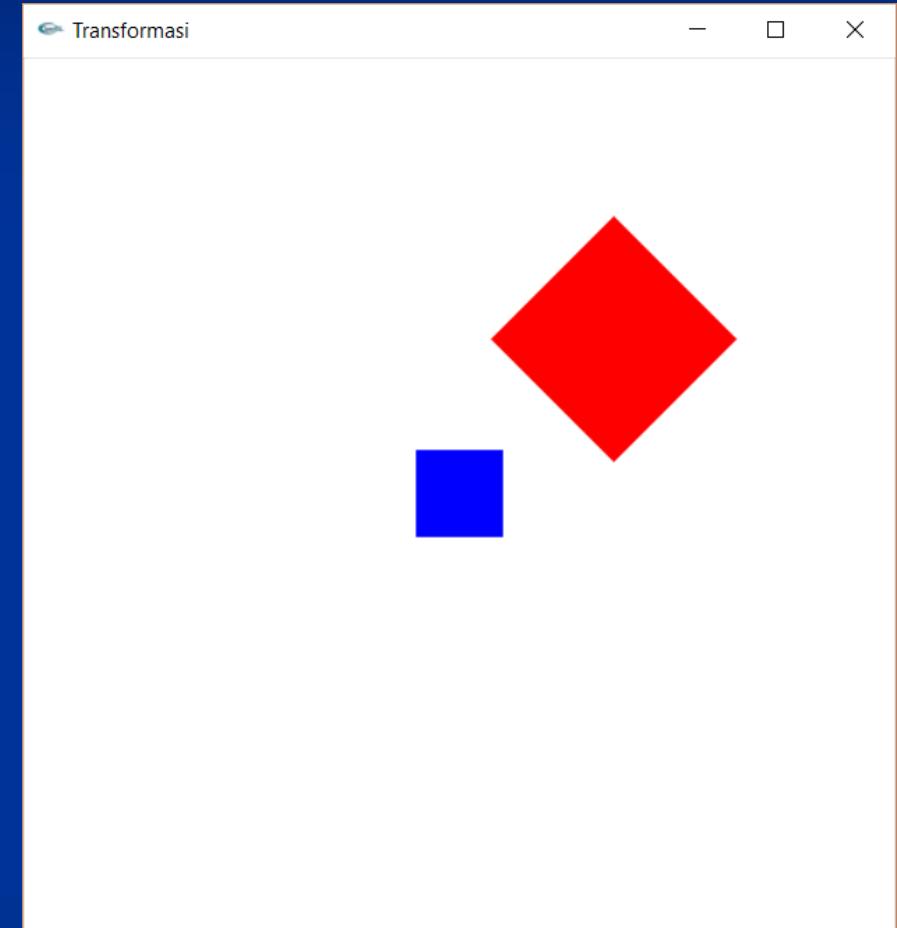
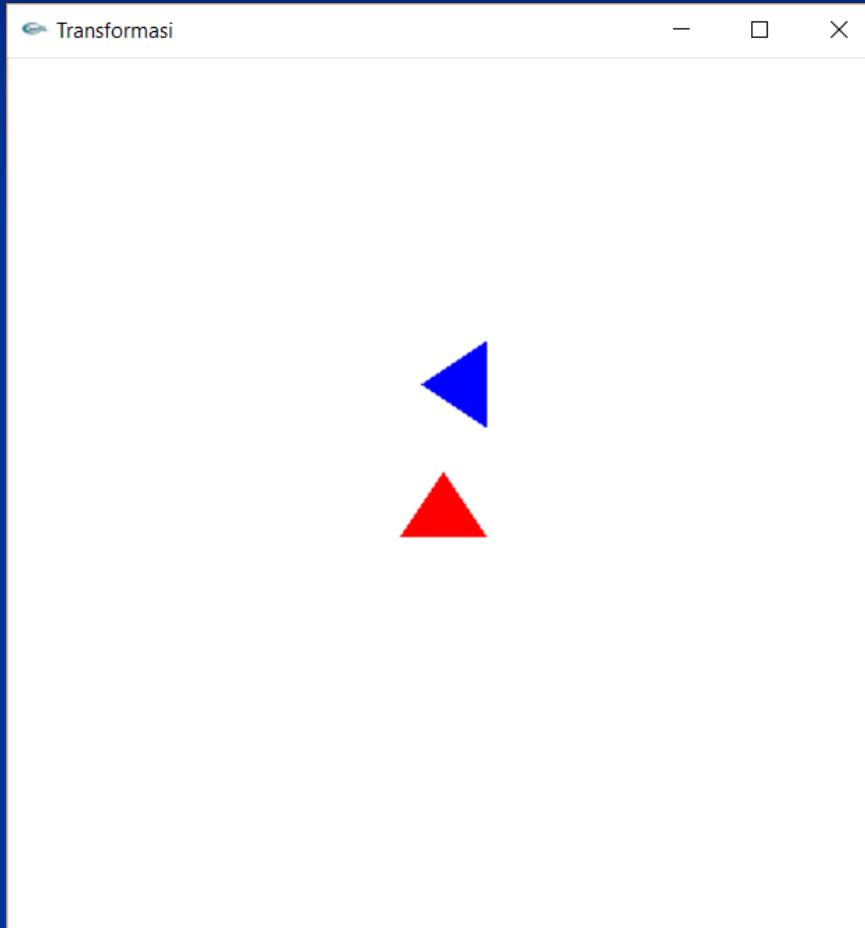
# Latihan



# CTM Stack

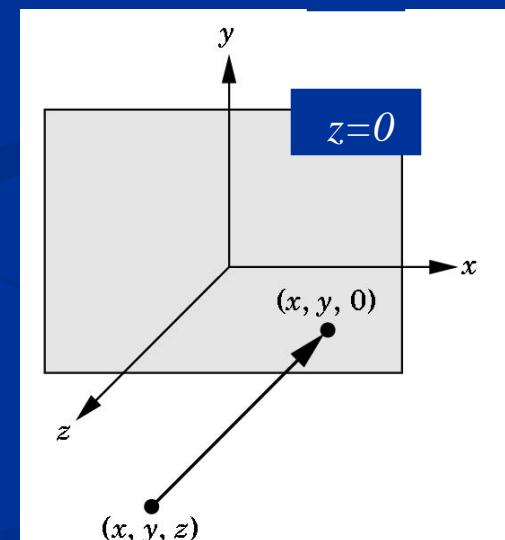
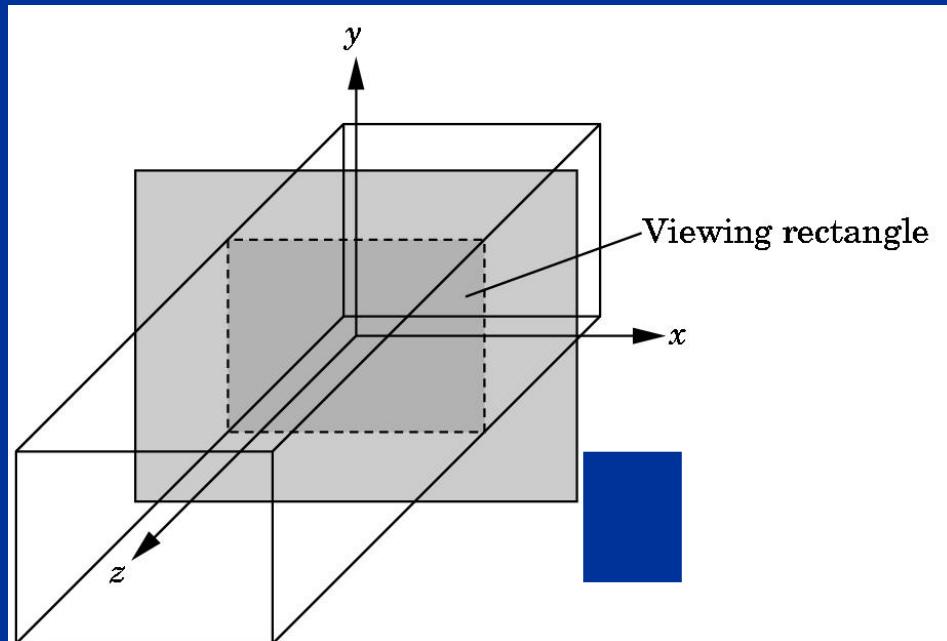
- OpenGL menyimpan CTM dalam bentuk stack
- Ukuran stack untuk GL\_MODELVIEW minimum 32 elemen, bergantung pada implementasi hardware
- glPushMatrix: untuk menduplikasi elemen teratas stack, dan meletakkannya pada top elemen
- glPopMatrix: menghapus elemen top, menjadikan elemen dibawahnya menjadi elemen top

# Latihan



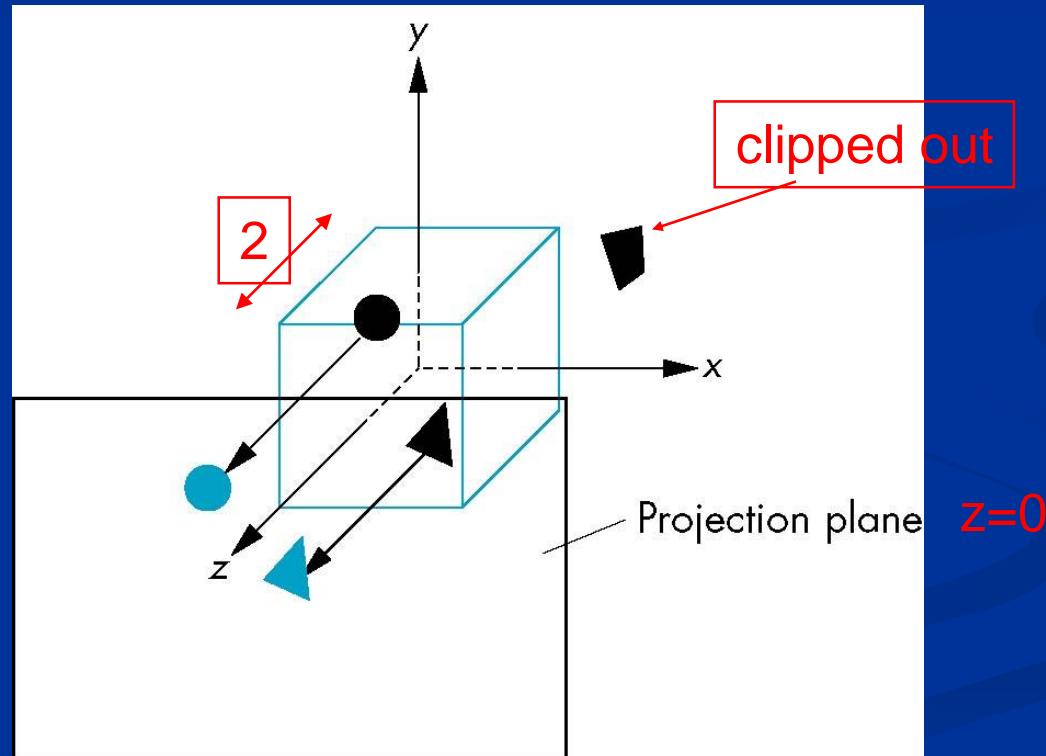
# 3 Dimensi

- Canvas/viewing rectangle vs box/viewing volume



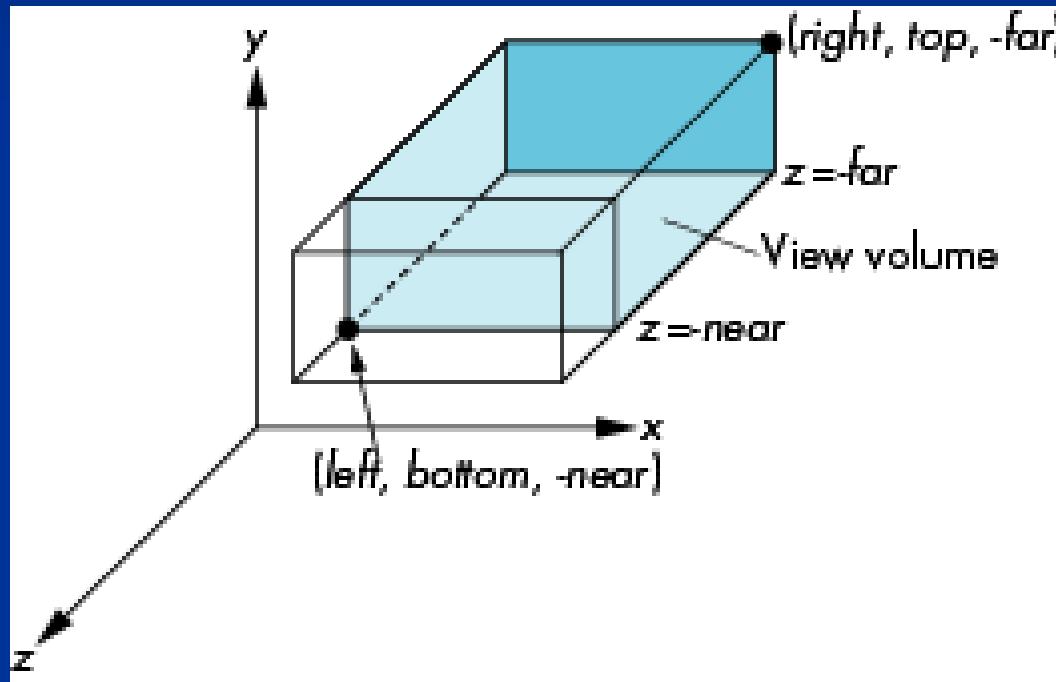
# 3 Dimensi

- Canvas/viewing rectangle vs box/viewing volume



# OpenGL Orthogonal Viewing

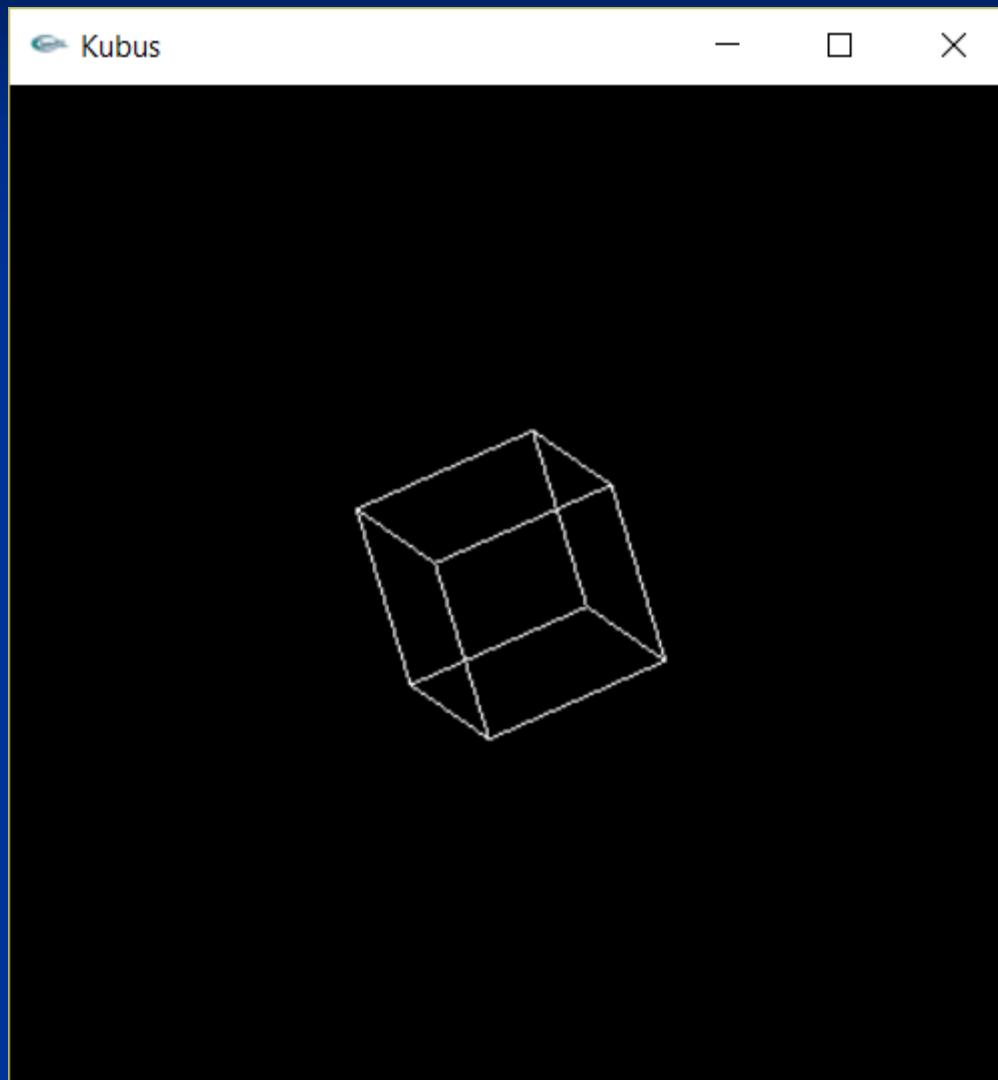
`glOrtho(left, right, bottom, top, near, far)`



# 3 Dimensi

- Hal-hal yang harus disiapkan:
  - Gunakan `glOrtho`, bukan `gluOrtho2D`
  - Gunakan `glEnable(GL_DEPTH_TEST)`
  - Gunakan `glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH) ;`
  - Gunakan `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT) ;`
  - Pada beberapa kasus diperlukan juga  
`glShadeModel(GL_FLAT) ;`

# 3 Dimensi

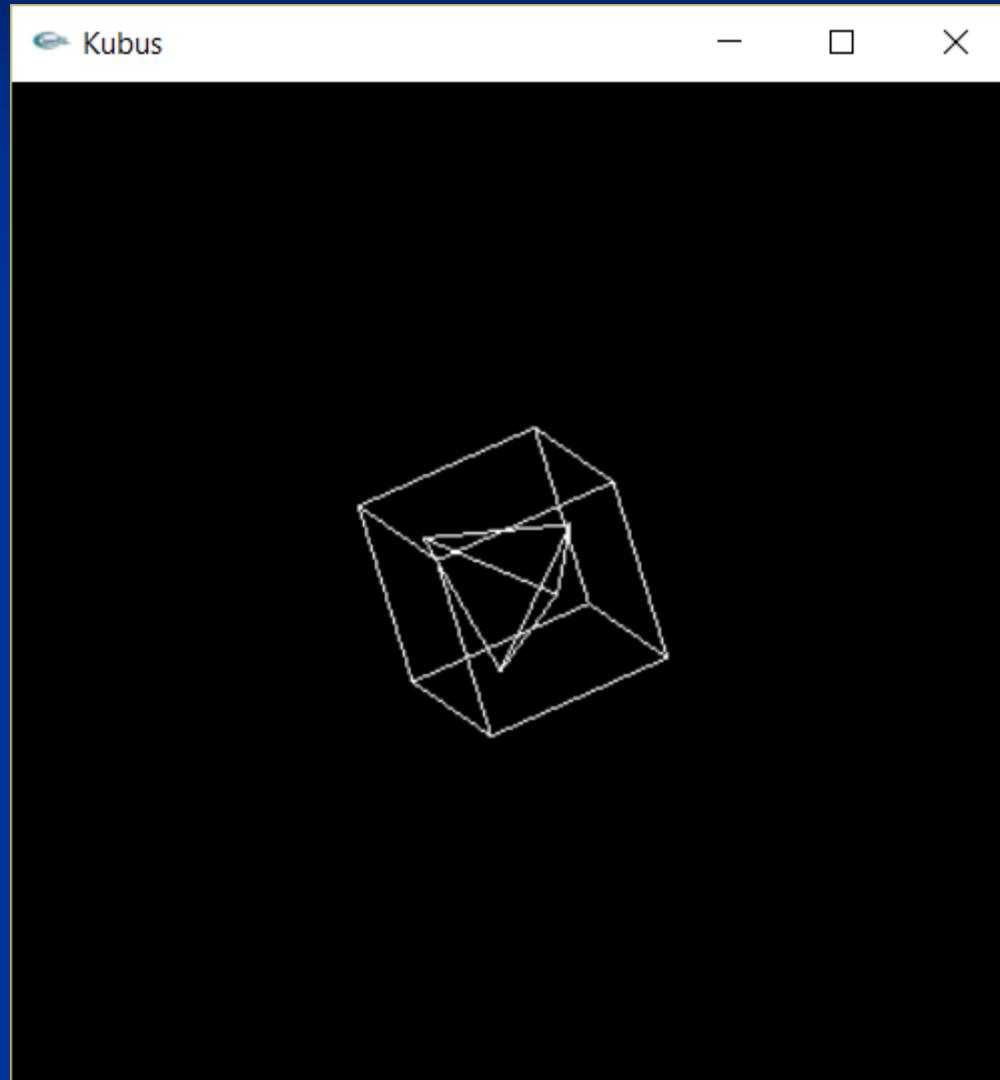


# Obyek 3 dimensi bawaan glut

## ■ Wireframe:

- glutWireTeapot(GLdouble size);
- glutWireSphere(GLdouble radius, GLint slices, GLint stacks);
- glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);
- glutWireTetrahedron(void);
- glutWireOctahedron(void);

# Obyek wireframe

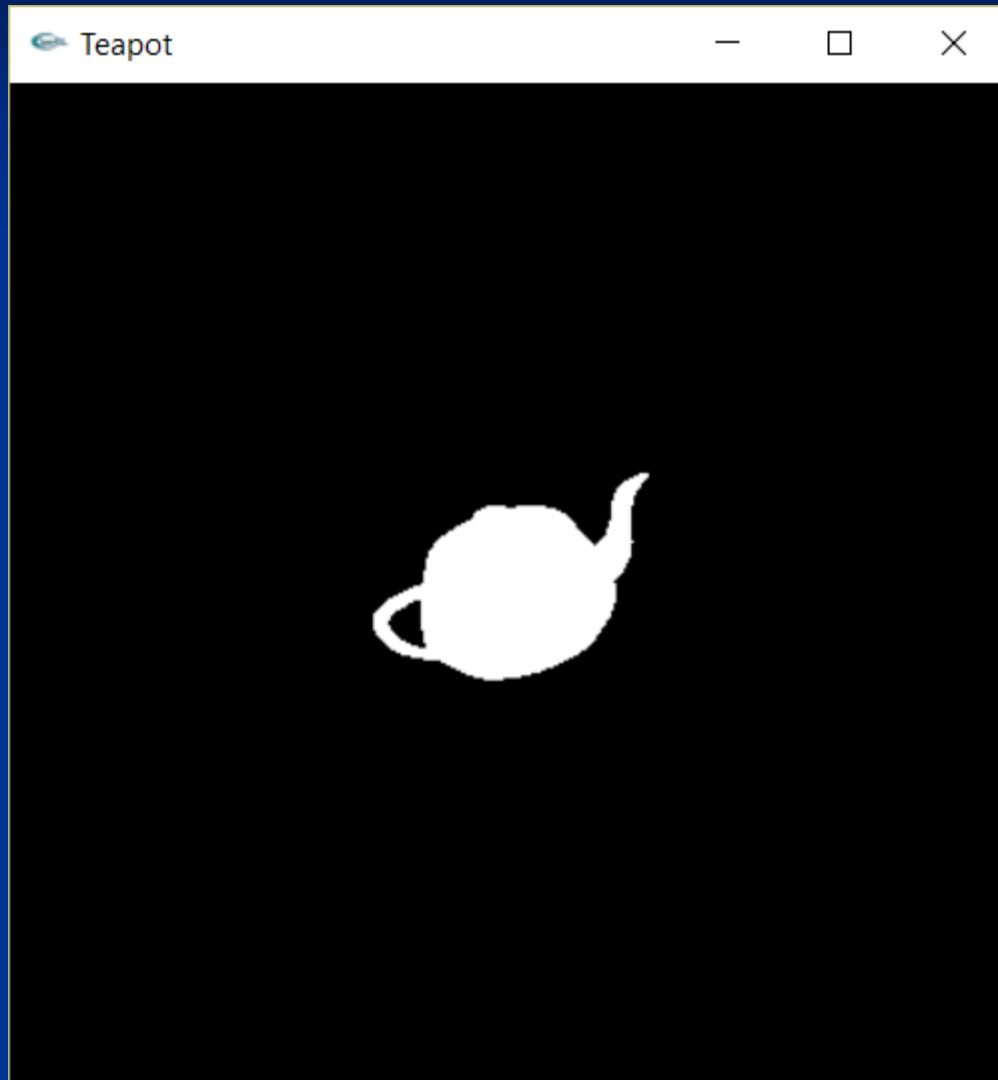


# Obyek 3 dimensi bawaan glut

## ■ Obyek solid 3 dimensi:

- glutSolidCube(GLdouble size);
- glutSolidTeapot(GLdouble size);
- glutSolidSphere(GLdouble radius, GLint slices, GLint stacks)
- glutSolidCone(GLdouble base, GLdouble height, GLint slices, GLint stacks)
- glutSolidTetrahedron(void)
- glutSolidOctahedron(void)

# Obyek solid

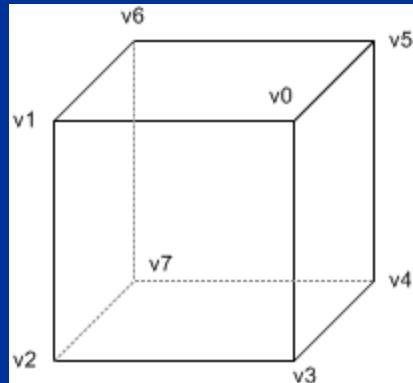


# Bangun 3 dimensi manual

- Sekalipun glut memiliki banyak obyek 3 dimensi bawaan, kadang kita perlu obyek 3 dimensi yang berbentuk khusus. Contoh: kubus warna tanpa shading
- Perlu tahu cara membuat obyek 3 dimensi secara manual

# Bangun bervolume

- Dibuat dari kumpulan bangun datar yang menjadi pembatas sisi-sisi luarnya

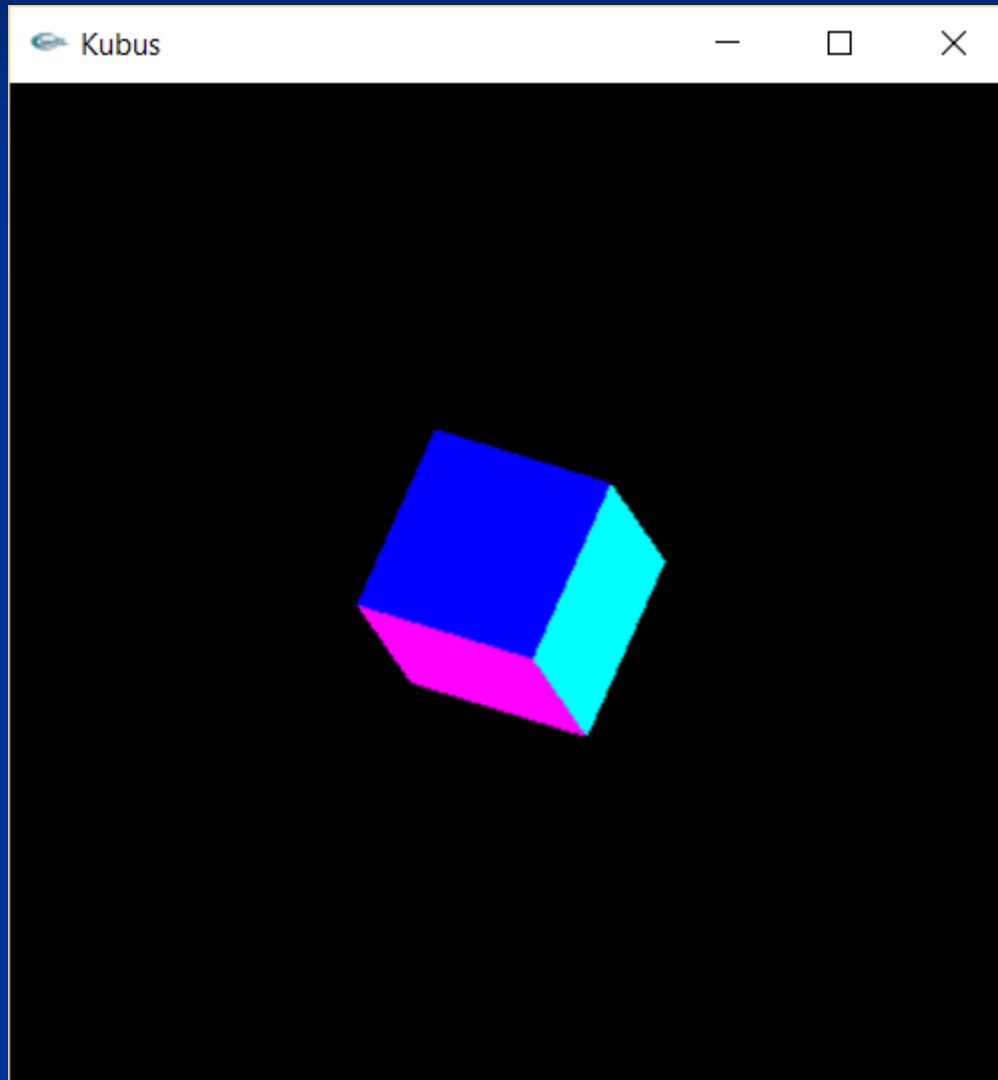


Song Ho Ahn, OpenGL Vertex Array, [http://www.songho.ca/opengl/gl\\_vertexarray.html](http://www.songho.ca/opengl/gl_vertexarray.html)

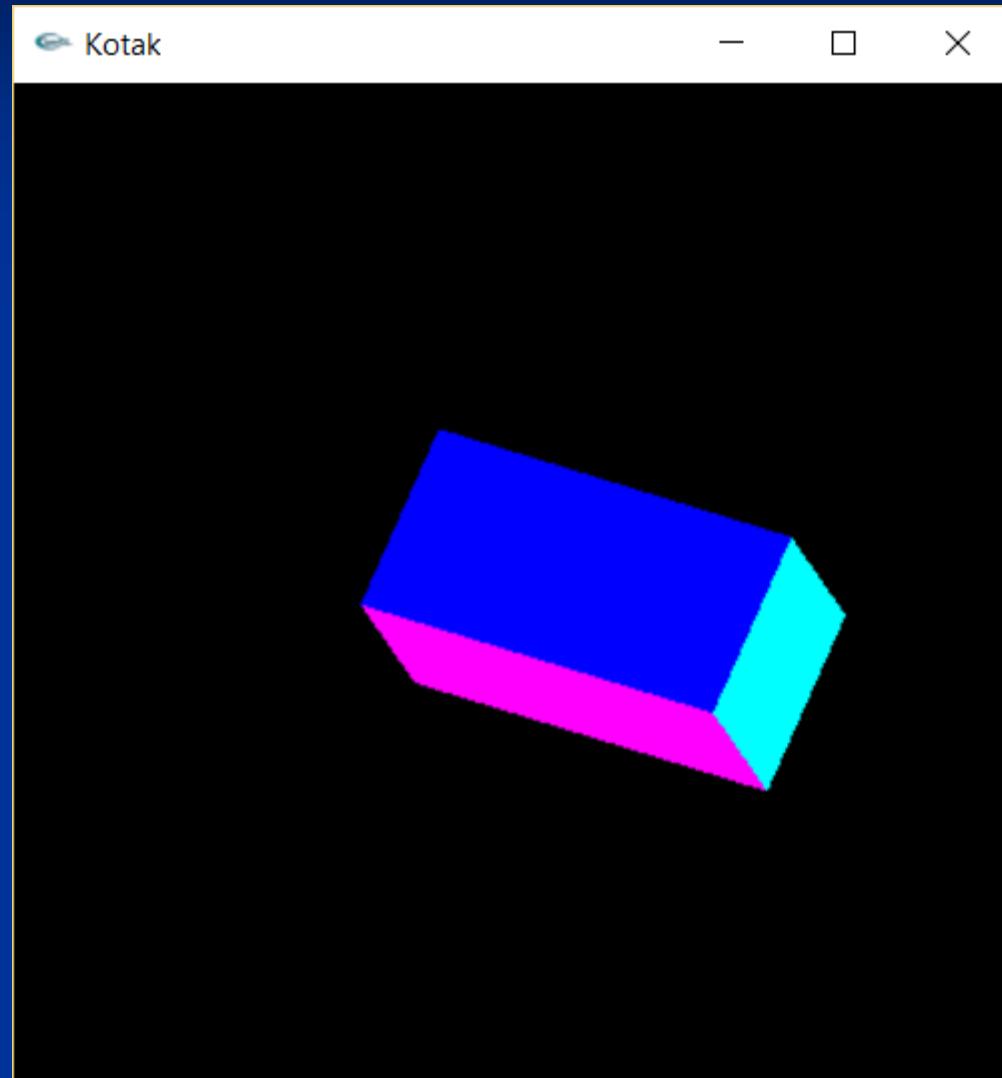
# 3 Dimensi

- **glVertex2f(x,y) → glVertex3f(x,y,z)**
- Transformasi selalu dilakukan dalam format 3 dimensi

# Kubus warna



# Latihan



# Demo 3

# Rangkuman

- Rotasi pada sembarang titik dapat dilakukan dengan bantuan transformasi jamak
- Mengembalikan nilai CTM setelah transformasi dapat dilakukan dengan inverse transformasi, glLoadIdentity, dan Push/PopMatrix
- Perpindahan ke 3 dimensi memerlukan beberapa perubahan setting program
- glut memiliki beberapa obyek 3 dimensi bawaan siap pakai

# Referensi

- Song Ho Ahn, OpenGL Vertex Array,  
[http://www.songho.ca/opengl/gl\\_vertexarray.html](http://www.songho.ca/opengl/gl_vertexarray.html)
- Edward Angel, “Interactive Computer Graphics Sixth Edition”, Pearson, 2012, ch2, p 98 – 106
- Edward Angel, “Interactive Computer Graphics Sixth Edition”, Pearson, 2012, ch 3, p 115 – 194
- F. S. Hill, Jr Jr., Stephen M. Kelley, “Computer Graphics Using OpenGL Third Edition”, Prentice Hall, 2007, ch 5, p 190 – 263