PERTEMUAN 02
PENGANTAR OPENGL

ADHI PRAHARA

Teknik Informatika. Fakultas Teknologi Industri

Universitas Ahmad Dahlan

CAPAIAN PEMBELAJARAN

- Menjelaskan tentang konsep OpenGL
- Menjelaskan fungsi-fungsi dalam OpenGL
- Menjelaskan kegunaan OpenGL
- Membuat program sederhana dengan OpenGL API (Praktikum)

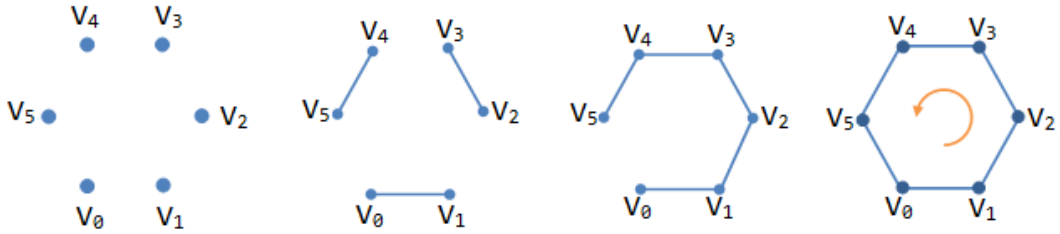
APA ITU OPENGL?

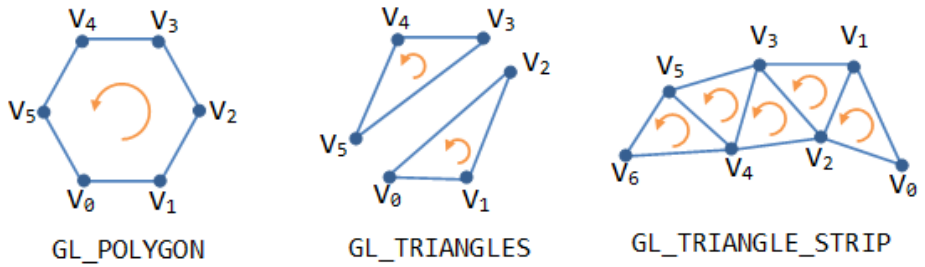
- Software library untuk mengakses fitur di hardware grafis
- Dikembangkan di Silicon Graphics Computer Systems – Juli 1994
- Memuat lebih dari 500 perintah untuk aplikasi interaktif 3D
- Sifat :
  - Hardware-independent
  - Software-independent
- Situs -> https://www.opengl.org/

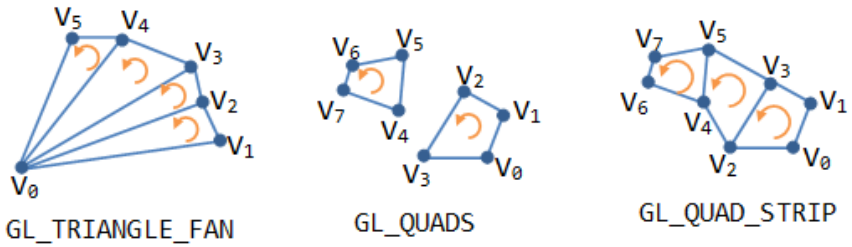
KONSEP OPENGL

- Obyek 3D dibangun dari obyek primitif geometri
- Obyek primitif :
  - Titik (points / vertex)
  - Garis (lines)
  - Bidang (patches)
- Citra primitif
  - Data piksel citra dan bitmap

OPENGL OBYEK PRIMITIF



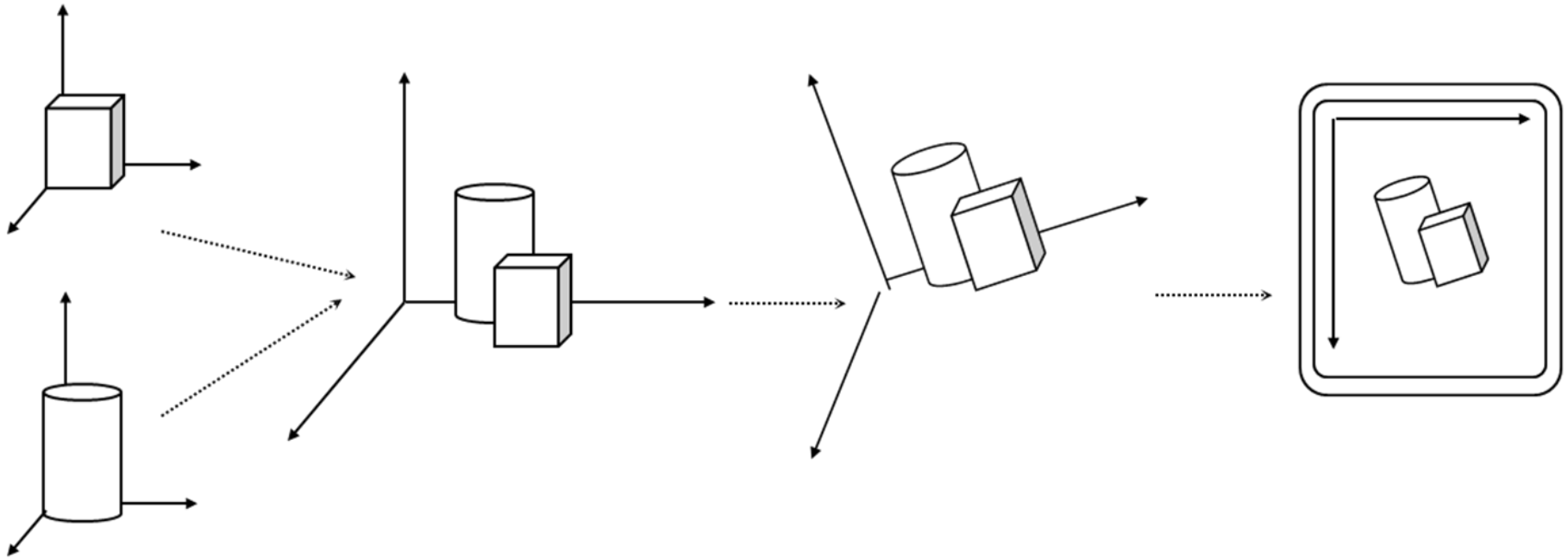




KONSEP KOORDINAT

- **Koordinat model** (modelling coordinates) : sistem koordinat yang digunakan dalam pembentukan model
- **Koordinat dunia/nyata** (world coordinates) : sistem koordinat yang digunakan pada lingkungan tempat obyek diletakkan
- **Koordinat pandang** (viewing coordinates) : sistem koordinat yang digunakan dalam memproyeksikan kenampakan obyek berdasarkan posisi sudut pandang (orientation)
- **Koordinat layar** (device coordinates) : sistem koordinat yang digunakan pada peralatan display

CONTOH



Model coordinates World coordinates Viewing Coordinates Device Coordinates

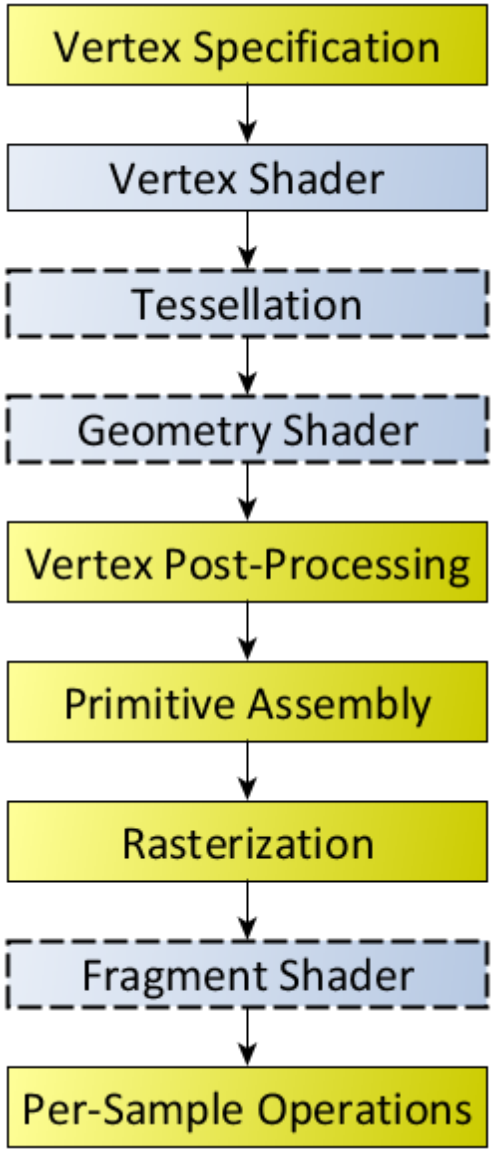
ISTILAH KOMPUTER GRAFIS

- **Rendering** : membuat citra dari obyek 2D atau 3D
- **Rasterisasi** : metode membuat citra dari obyek 2D / 3D
- **Fragment** : data untuk membuat piksel pada gambar primitif
- **Shaders** : kode dalam bahasa pemrograman grafis (shading language) untuk eksekusi pada prosesor pemrograman grafis
- **Piksel** : elemen terkecil pada display
- **Framebuffer** : blok memori yang digunakan untuk rendering

OPENGL RENDERING PIPELINE

1. Siapkan array dari data vertex kemudian direnderer
2. Proses vertex:
   a. Setiap vertex diproses oleh vertex shader
   b. Hasil dari langkah **1.a** masuk ke tessellation shader (optional)
   c. Hasil dari langkah **1.b** masuk ke geometry shader (optional).

   Keluaran dari langkah **2** adalah obyek-obyek primitif
3. Pos-proses vertex:
   a. Hasil dari langkah **2** di transformasi posisinya
   b. Diterapkan clipping dan transformasi viewport
4. Setiap obyek primitif dikumpulkan
5. Konversi scan dan interpolasi parameter obyek primitif yang menghasilkan sejumlah fragmen
6. Setiap fragmen diproses oleh fragment shader menghasilkan sejumlah keluaran
7. Operasi per keluaran/sampel:
   a. Scissor Test
   b. Stencil Test
   c. Depth Test
   d. Blending
   e. Logical Operation
   f. Write Mask



OPENGL SHADER

- **Vertex shader** : menerima informasi data vertex untuk diolah warna, transformasinya, dll
- **Tesselation control shader** : set parameter untuk interpolasi dan banyaknya tessellation yang dilakukan
- **Tesselation evaluation shader** : menentukan komputasi interpolasi yang digunakan pada geometry
- **Geometry shader** : mengubah / menambah geometri asli dengan mengembangkan vertices baru
- **Fragment shader** : menerima informasi fragment untuk komputasi warna piksel, posisi pada layar

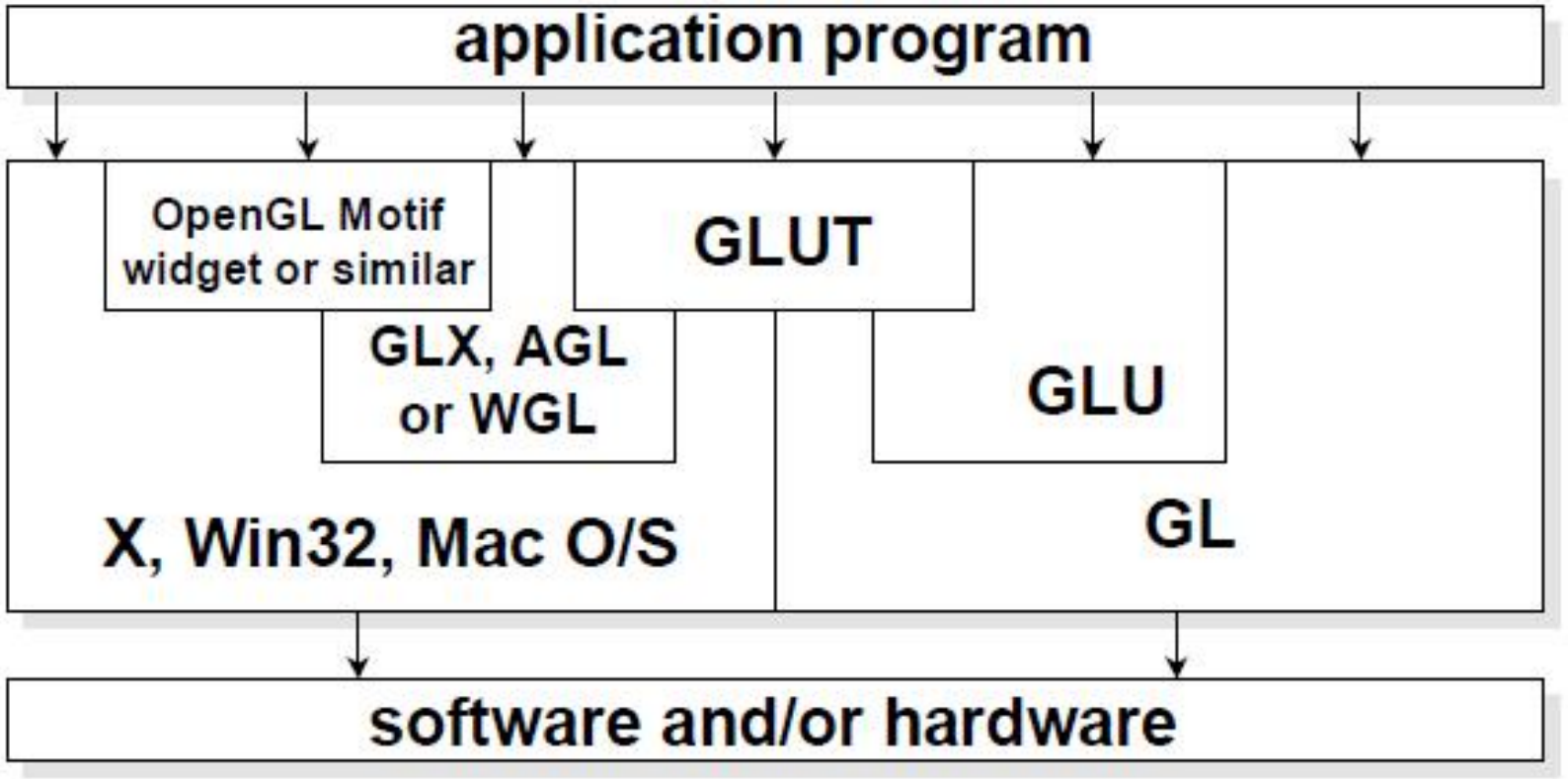
OPENGL KE SISTEM OPERASI

- OpenGL : software independent
- Diperlukan perantara untuk menghubungkan OpenGL ke Sistem Operasi
  - GLX -> X Windows -> Unix
  - AGL -> Apple Macintosh
  - WGL -> Microsoft Windows
- Support GUI :
  - Motif
  - Win32 API

LIBRARY TAMBAHAN OPENGL

- GLU, memudahkan rendering :
  - Quadric surfaces (bola, kerucut, silinder, dll)
  - NURBS (Non-Uniform Rational Basis Spline) dan kurva
- GLUT :
  - Memudahkan pemograman OpenGL : membuat window
  - Menghandle events
  - Animasi

OpenGL dan API



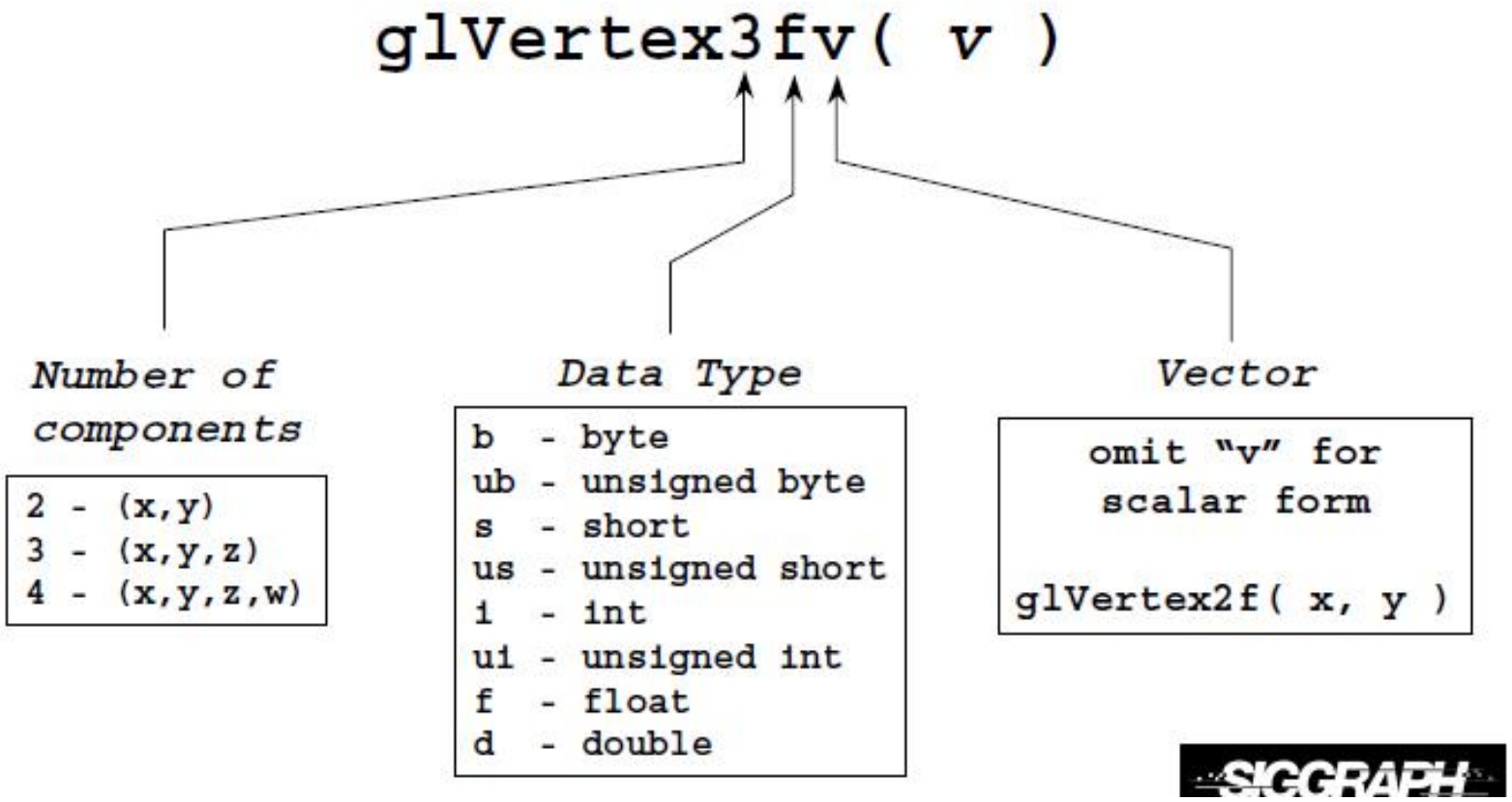
PROGRAM OPENGL

- OpenGL header -> gl.h
- GLU header -> glu.h
- GLUT header -> glut.h, dll
- Unix library -> libGL.so
- Windows library -> opengl32.lib, glu32.lib, glut32.lib

TIPE DATA OPENGL

Suffix	Data Type	Typical Corresponding C-Language Type	OpenGL Type Definition
b	8-bit integer	signed char	GLbyte
s	16-bit integer	signed short	GLshort
i	32-bit integer	int	GLint, GLsizei
f	32-bit floating-point	float	GLfloat, GLclampf
d	64-bit floating-point	double	GLdouble, GLclampd
ub	8-bit unsigned integer	unsigned char	GLubyte
us	16-bit unsigned integer	unsigned short	GLushort
ui	32-bit unsigned integer	unsigned int	GLuint, GLenum, GLbitfield

FORMAT INSTRUKSI OPENGL



MODEL WARNA OPENGL

- Menggunakan model warna RGBA / TrueColor dan Color Index
- Fungsi untuk mode warna RGBA -> glColor*(red, green, blue, alpha)
- Fungsi untuk mode color index -> glIndex*( )
- Nilai piksel warna antara 0.0f – 1.0f
- Komponen A -> alpha -> transparansi
- Dalam inisialisasi mode display :
  - GLUT_RGBA
  - GLUT_INDEX
- Contoh : Set warna hijau

R G B

```
glColor3f(0.0f, 1.0f, 0.0f);
```

MODEL KOORDINAT OPENGL

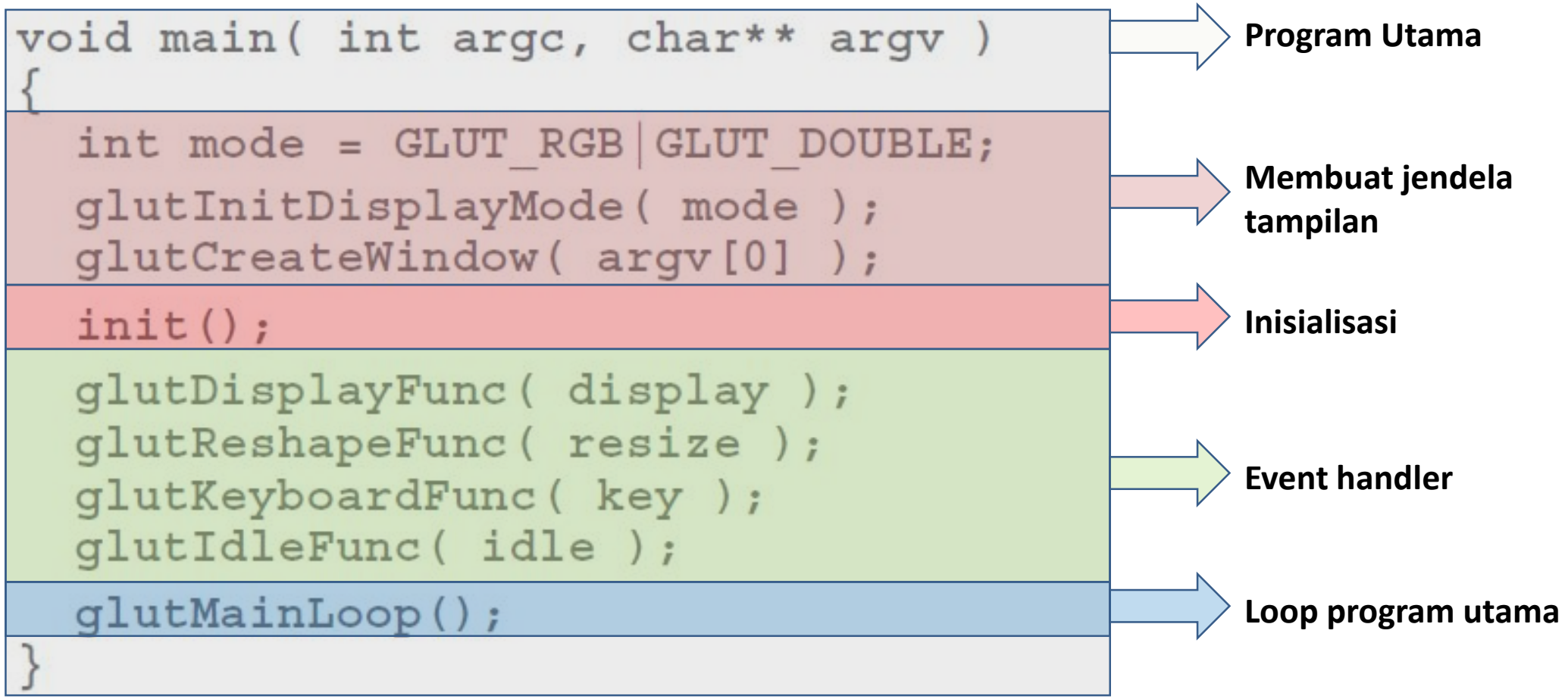
- Menggunakan format (x, y, z, w)
- X -> sumbu X, Y -> sumbu Y, Z -> sumbu Z
- W -> koordinat homogen
- Koordinat homogen : (x/w, y/w, z/w, w)
- Bila didefinisikan x, y saja maka z = 0.0f dan w = 1.0f
- Contoh : Set koordinat x=1, y=1, z=0 dan w=1

```
                    X     Y     Z
glVertex3fv(1.0f, 1.0f, 0.0f);
```

GLUT : PROGRAM DASAR

1. Memilih tipe jendela (window) dan membuat jendela tampilan
2. Inisialisasi variabel pada OpenGL :
    - Set warna latar belakang
    - Inisialisasi sumber cahaya
    - Insialisasi proyeksi
3. Handle event :
    - Display
    - Mouse, keyboard
    - Resize scene
4. Jalankan loop untuk looping event utama

CONTOH PROGRAM



CONTOH PROGRAM (1)
DEKLARASI HEADER DAN VARIABEL

```
// deklarasikan semua header disini
#include <windows.h>
#include <glut.h>

// inisialisasi variabel untuk transformasi seperti translasi, rotasi
atau scaling
float angle = 0.0f;// sudut transformasi kamera
float posX = 0.0f, rotX =  0.0f;// posisi kamera di sumbu X
float posY = 0.0f, rotY =  0.0f;// posisi kamera di sumbu Y
float posZ = 5.0f, rotZ = -1.0f;// posisi kamera di sumbu Z

float objectAngleX = 0.0f;// sudut tranformasi obyek di sumbu X
float objectAngleY = 0.0f;// sudut tranformasi obyek di sumbu Y
float objectAngleZ = 0.0f;// sudut tranformasi obyek di sumbu Z
```

CONTOH PROGRAM (2)
BUAT FUNGSI MENGGAMBAR OBYEK

```
// fungsi untuk menggambar obyek kubus
void drawObject()
{
	// obyek bisa dimasukkan diantara glPushMatrix() dan glPopMatrix()
	// fungsinya agar obyek tidak terpengaruh atau mempengaruhi obyek lain saat diwarnai, ditransformasi dan sebagainya
	glPushMatrix();

	// operasi transformasi rotasi obyek ke arah kanan-kiri
	glRotatef(objectAngleY, 0.0f, 1.0f, 0.0f);

	glPushMatrix();

	// operasi transformasi rotasi obyek ke arah atas-bawah
	glRotatef(objectAngleX, 1.0f, 0.0f, 0.0f);

	// set warna obyek ke warna hijau (0.0f, 1.0f, 0.0f)
	glColor3f(0.0f, 1.0f, 0.0f);

	// bila menggambar obyek harus diawali glBegin(tipe obyek) dan diakhiri dengan glEnd()
	// kecuali menggunakan fungsi yang sudah ada di GLUT-OpenGL seperti dibawah ini
	glutSolidCube(1.0f); // menggambar obyek kubus

	glPopMatrix();

	glPopMatrix();
}
```

CONTOH PROGRAM (3)
BUAT FUNGSI MENAMPILKAN OBYEK

```
// taruh semua obyek yang akan digambar di fungsi display()
void display()
{
     // bersihkan dan reset layar dan buffer
     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
     glLoadIdentity();

     // posisikan kamera pandang
     // dalam hal ini sumbu Y ada diatas dan posisi kamera pandang di (posX, posY, posZ)
     gluLookAt(posX, posY, posZ, posX + rotX, posY + rotY, posZ + rotZ, 0.0f, 1.0f, 0.0f);

     // panggil fungsi untuk menggambar obyek
     drawObject();

     // tampilkan obyek ke layar
     // gunakan glFlush() bila memakai single buffer
     // gunakan glutSwapBuffers() bila memakai double buffer
     glutSwapBuffers();
}
```

CONTOH PROGRAM (4)
BUAT FUNGSI INISIALISASI

```
// inisialisasikan pencahayaan, tekstur dan pandangan kamera di fungsi init()
void init(void)
{
    // inisialisasi warna latar belakang layar dalam hal ini warna putih (1.0, 1.0, 1.0, 0.0)
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glEnable(GL_DEPTH_TEST);// mengaktifkan depth buffer
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0, 1.0, 1.0, 100.0);// set proyeksi ke perspektif
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    // inisialisasi kamera pandang
    gluLookAt(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
}
```

CONTOH PROGRAM (5)
BUAT FUNGSI RESIZE WINDOW

```
// fungsi ini digunakan bila layar akan diresize (default)
void reshape(int w, int h)
{
    glViewport(0, 0, (GLsizei)w, (GLsizei)h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45, (GLfloat)w / (GLfloat)h, 1.0, 100.0);
    glMatrixMode(GL_MODELVIEW);
}
```

CONTOH PROGRAM (6)
BUAT FUNGSI MASUKAN KEYBOARD

```
// fungsi untuk mengatur masukan dari keyboard untuk arah kiri, kanan, atas, bawah, PgUp, dan PgDn
void keyboard(int key, int x, int y)
{
	float fraction = 0.1f;
	switch (key)
	{
		// masukkan perintah disini bila tombol kiri ditekan dalam hal ini perintah rotasi obyek ke kiri sebanyak 1 derajat
		case GLUT_KEY_LEFT:
		objectAngleY -= 1.0f;
		glutPostRedisplay();// update obyek
		break;
		// masukkan perintah disini bila tombol kanan ditekan dalam hal ini perintah rotasi obyek ke kanan sebanyak 1 derajat
		case GLUT_KEY_RIGHT:
		objectAngleY += 1.0f;
		glutPostRedisplay();// update obyek
		break;
		// masukkan perintah disini bila tombol atas ditekan dalam hal ini perintah rotasi obyek ke atas sebanyak 1 derajat
		case GLUT_KEY_UP:
		objectAngleX -= 1.0f;
		glutPostRedisplay();// update obyek
		break;
		// masukkan perintah disini bila tombol bawah ditekan dalam hal ini perintah rotasi obyek ke bawah sebanyak 1 derajat
		case GLUT_KEY_DOWN:
		objectAngleX += 1.0f;
		glutPostRedisplay();// update obyek
		break;
		// zoom in
		case GLUT_KEY_PAGE_UP:
		posX += rotX * fraction;
		posZ += rotZ * fraction;
		glutPostRedisplay();// update obyek
		break;
		// zoom out
		case GLUT_KEY_PAGE_DOWN:
		posX -= rotX * fraction;
		posZ -= rotZ * fraction;
		glutPostRedisplay();// update obyek
		break;
	}
}
```

CONTOH PROGRAM (7)
BUAT FUNGSI TIMER

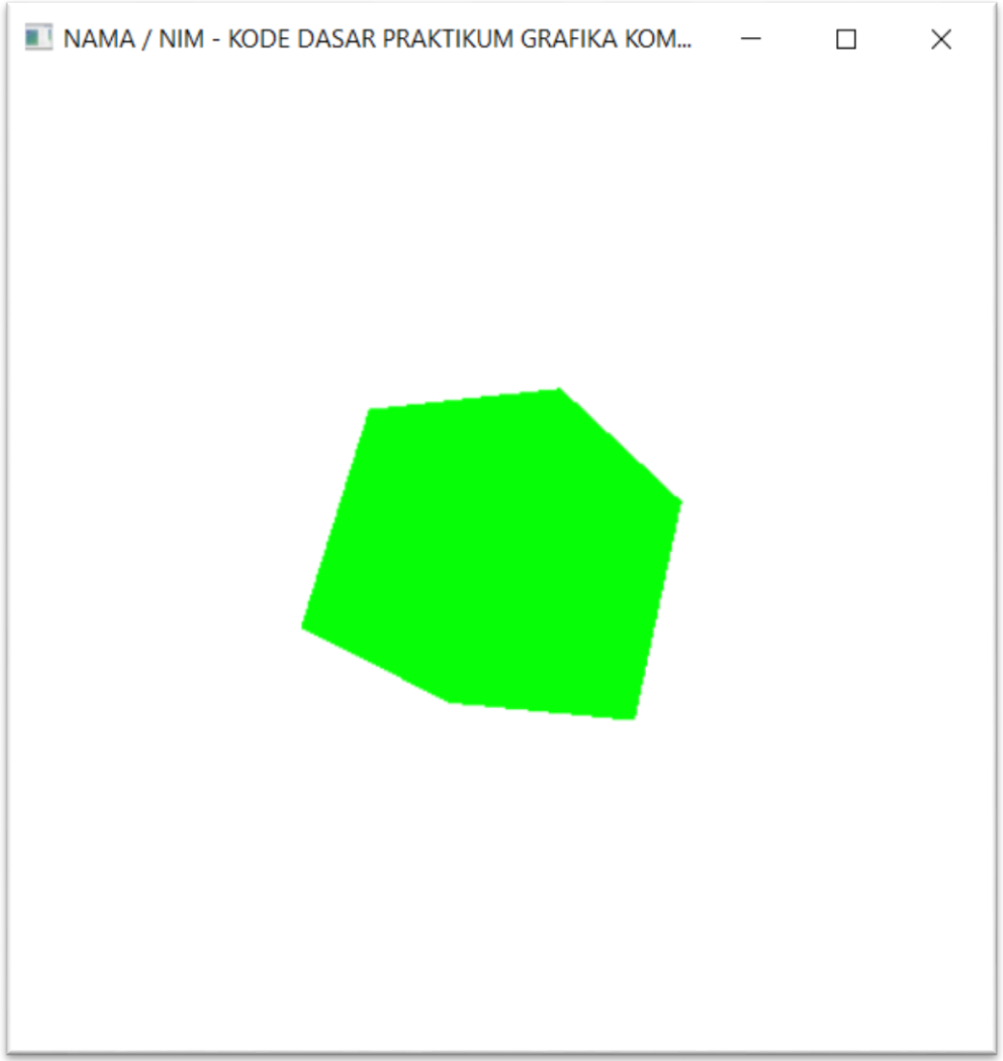
```
// timer untuk animasi (gunakan bila perlu)
void timer(int value)
{
    glutPostRedisplay();
    glutTimerFunc(55, timer, 0);
}
```

CONTOH PROGRAM (8) BUAT FUNGSI UTAMA

```
// program utama
int main(int argc, char** argv)
{
	// inisialisasi jendela OpenGL
	// GLUT_SINGLE berarti memakai single buffer
	// GLUT_DOUBLE berarti memakai double buffer
	// GLUT_RGB berarti mode tampilan yang dipakai RGB
	// GLUT_RGBA berarti mode tampilan yang dipakai RGBA
	// GLUT_DEPTH berarti memakai depth buffer
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);

	// set ukuran jendela tampilan
	glutInitWindowSize(480, 480);// besarnya jendela dalam piksel dalam hal ini 300x300
	glutInitWindowPosition(100, 100);// posisi jendela dilayar komputer dalam piksel
	// judul jendela (wajib diubah dengan informasi NAMA / NIM - JUDUL PRAKTIKUM masing-masing)
	glutCreateWindow("NAMA / NIM - KODE DASAR PRAKTIKUM GRAFIKA KOMPUTER");

	// panggil fungsi init untuk inisialisasi awal
	init();

	// event handler untuk display, reshape dan keyboard
	glutDisplayFunc(display);   // display
	glutReshapeFunc(reshape);   // reshape
	glutSpecialFunc(keyboard);  // keyboard
	//glutTimerFunc(0, timer, 0); // aktifkan timer bila perlu

	// looping
	glutMainLoop();

	return 0;
}
```

CONTOH PROGRAM (9) HASIL PROGRAM

- Jalankan program dengan menekan tombol run atau f5 pada visual studio
- Gerakkan dan zoom obyek dengan keyboard

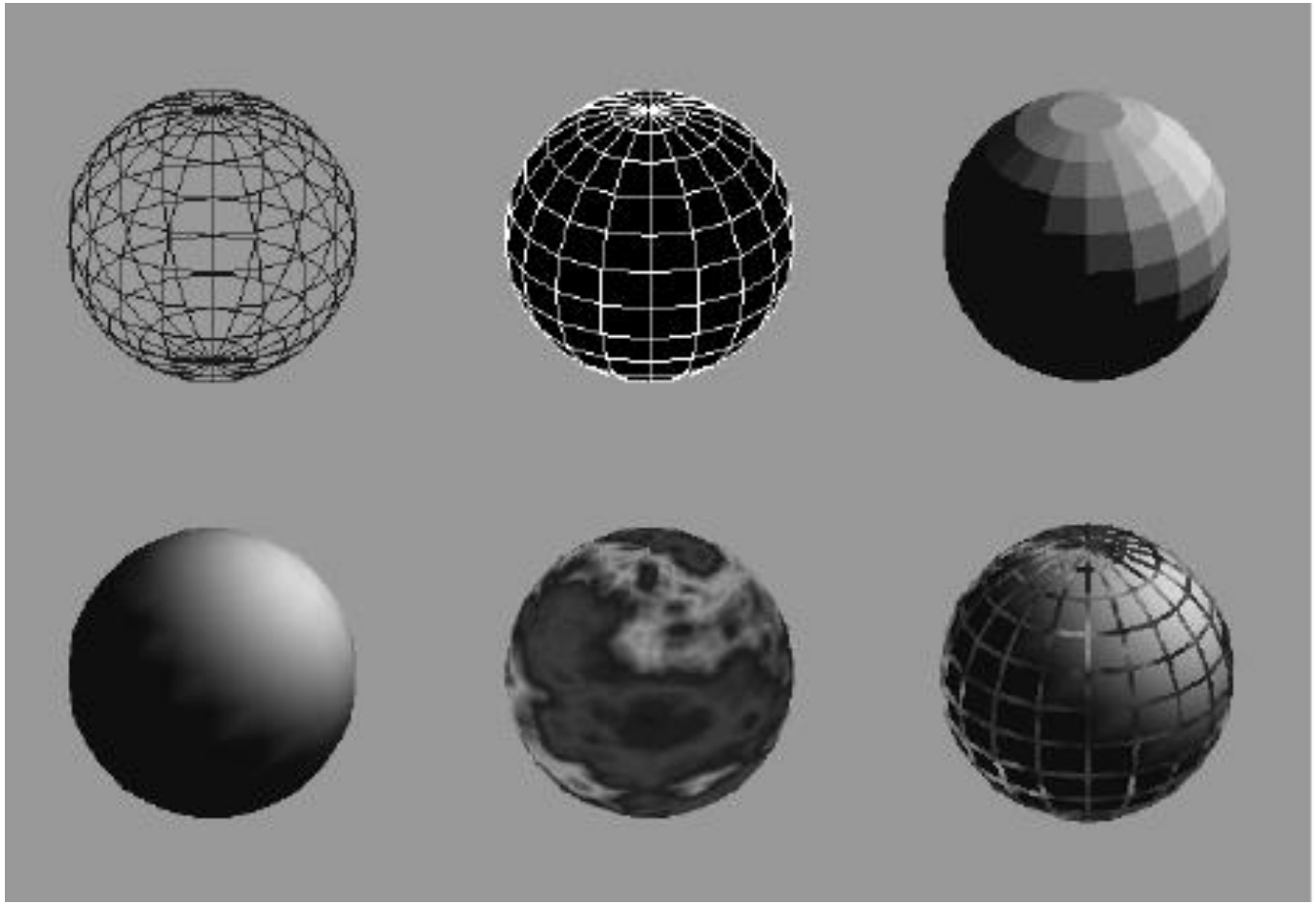


CONTOH LAIN: MENGGAMBAR OBYEK

```
void drawObject()
{
        glClear( GL_COLOR_BUFFER_BIT );     // bersihkan layar
        glBegin( GL_TRIANGLE );             // render bidang segitiga
        glVertex3fv( 1.0f, 1.0f, 0.0f );    // titik pertama
        glVertex3fv( 3.0f, 1.0f, 0.0f );    // titik kedua
        glVertex3fv( 2.0f, 3.0f, 0.0f );    // titik ketiga
        glEnd();                            // akhir dari menggambar bidang
        glutSwapBuffers();                  // smooth animasi transisi
}
```

CONTOH OPENGL PRIMITIF

RENDERING OPENGL



OBYEK PADA OPENGL