

LAPORAN TUGAS RANCANG GRAFIKA KOMPUTER

Perancangan dan Pembuatan Bangunan “Rumah dinas walikota salatiga” Berbasis Vektor dengan OpenGL



Disusun oleh :

1. Rizky Satya Pambudi (672020148)
2. Andrian Roland K Pakpahan (672020150)
3. Yohanes Albryan Simanjuntak (672020151)

**Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
2022**

1. PENDAHULUAN

Jika kita berjalan – jalan di kota Salatiga , Provinsi Jawa Tengah, kita pasti akan melihat beberapa bangunan yang ada di kota tersebut. Nah, disini kami tertarik dengan salah satu bangunan di kota tersebut yaitu Rumah Dinas Wali Kota Salatiga adalah bangunan yang terinventarisasi untuk ditetapkan sebagai cagar budaya, yang terletak di Jalan Diponegoro No.1, Kelurahan Salatiga, Kecamatan Sidorejo, Kota Salatiga, Provinsi Jawa Tengah. Secara umum, bangunan tersebut memperlihatkan konsep ruang sebuah kota modern karena berada di tengah-tengah kota, seperti halnya bangunan lain di Jawa yang menjadi identitas wilayahnya. Bangunan ini dibagi menjadi dua bagian berdasarkan coraknya, yaitu bangunan pertama dan kedua. Bangunan pertama rumah dinas itu awalnya merupakan tempat mukim anggota Majelis Gereja Protestan Indonesia Bagian Barat (GPIB) dan diperkirakan dibangun sekitar tahun 1825. Menurut catatan *Binnenlandsch Bestuur* (kepegawaian kolonial), bangunan induk itu disebutkan mulai ditempati oleh asisten residen sejak tahun 1903. Arsitektur bangunan ini dilihat dari bentuknya menunjukkan adanya perpaduan antara kebudayaan Jawa dan Eropa. Adapun bangunan kedua diperkirakan didirikan tahun 1830 dengan konstruksi arsitektur Eropa. Penyair Prancis, Jean Nicolas Arthur Rimbaud, pernah singgah di rumah dinas ini pada 2–15 Agustus 1876, sedangkan Soekarno bertemu dengan istri keempatnya, Hartini Soewondo, ketika melakukan kunjungan kerja di tempat ini pada 1952.

Laporan yang kami buat ini berisi tentang panduan sederhana tentang penulisan laporan Tugas Rancang kami dengan object salah satu Bangunan yang ada di Salatiga yaitu “Rumah Dinas Wali Kota Salatiga ” Berbasis Vektor dengan OpenGL . Tugas Rancang yang kami buat adalah sebuah miniatur atau duplikat bangunan tersebut dalam bentuk aplikasi program untuk menampilkan vektor – vektor dengan OpenGL dari object yang kami pilih, jadi disini kami membuat suatu gambar 3Dimensi dengan menggunakan bahasa pemrograman C++. Laporan ini kami susun sebagai pemenuhan Tugas Akhir (Tugas Rancang) mata kuliah Grafika Komputer (Grafkom) di semester antara tahun ajaran 2021 / 2022 yang diampu oleh dosen kami Bapak Evangs Mailoa. Yang telah mengajarkan kami cara pembuatan bangunan 3D atau bidang 2D melalui bahasa pemrograman C++ berbasis Vektor dengan Open GL. Dalam perancangan dan pembuatan ini bisa menggunakan banyak aplikasi seperti Dev C++, Visual Studio, dan lain-lain. Disini kami menggunakan aplikasi CodeBlocks.

Metode penilitan yang digunakan adalah dengan melihat gambar melalui google serta dan aplikasi lainnya, untuk mengambil beberapa dokumentasi dari bergai sudut agar untuk membantu penelitian kami supaya dalam pembuatan nanti kurang lebih mirip dengan object aslinya.

Disini kami akan membahas secara singkat tentang, bagaimana cara kami membuat suatu bangunan dengan menggunakan bahasa pemrograman C++. Dalam pembahasan ini menganut konsep vektor dan bidang, dengan software bantu OpenGL.

2. TINJAUAN PUSTAKA

Pada artikel lain seperti pada laporan kelompok lain mungkin juga membahas pembuatan suatu bangunan tetapi pada dasarnya cara membuatnya sama-sama memakai bahasa pemrograman C++ dan OpenGL. Dan perlu diketahui OpenGL merupakan aplikasi software berisi package (paket) tentang kode programan yang digunakan untuk mendesain dan menentukan titik-titik vektor. Seperti untuk menentukan bidang segi empat itu diperlukan empat titik, nah titik-titik tersebut mempunyai syntax program yang sama tetapi letak tiap titik tersebut berbeda, contoh kode program:

`glBegin(GL_POLYGON);` → memulai vektor dengan polygon yaitu ada isi (warna) bidang.

`glColor3f(0.0,0.0,0.0);` → menentukan warna dengan urutan Red, Green, Blue.

`glVertex3f(1.0,1.0,0.0);` → titik pertama dengan koordinat x,y,z (1,1,0);

`glVertex3f(1.0,2.0,0.0);` → titik kedua dengan koordinat x,y,z (1,2,0);

`glVertex3f(2.0,2.0,0.0);` → titik ketiga dengan koordinat x,y,z (2,2,0);

`glVertex3f(2.0,1.0,0.0);` → titik keempat dengan koordinat x,y,z (2,1,0);

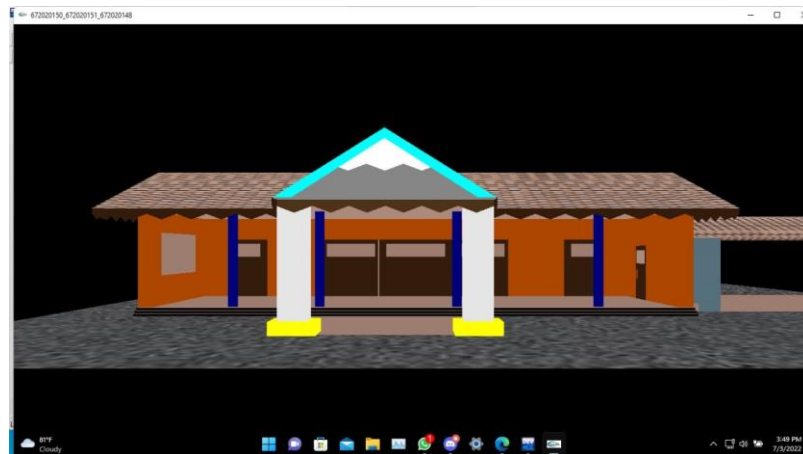
`glEnd();` → akhir vektor.

Nah contoh program diatas adalah untuk menentukan bidang segi empat. `glVertex3f` merupakan kode program untuk menentukan vektor atau titik – titik yang kita inginkan dalam 3f (3 koordinat x,y,z dengan tipe data float (f)). Dalam pembuatan vektor bidang atau bangunan mempunyai banyak titik – titik, dan untuk mengakalinya agar tidak ribet kita harus bisa logika bahasa pemrograman seperti dalam pengetahuan konsep Dasar-Dasar Pemrograman , Pemrograman Berorientasi Object ,dll. Nah setiap orang pastinya mempunyai logikanya masing -masing yang berbeda-beda dengan orang lain. Untuk itu kami tidak memberikan contoh artikel dari orang lain yang juga membahas tentang ini, karena kita tahu pasti bahasa yang mereka gunakan adalah Bahasa C++, tetapi perbedaannya ada di aplikasi software C++ yang mereka gunakan dan juga logika yang mereka terapkan. Tetapi konsep dasarnya sama yaitu menentukan titik – titik, pewarnaan, dan lain sebagainya, yang diperlukan untuk project kita.

3. PEMBAHASAN

Untuk pembahasan, diatas tadi sudah disebutkan bahwa kami membuat duplikat bangunan Rumas Dinas Walikota Salatiga dalam bentuk vektor dengan menggunakan OpenGL bahasa Pemrograman C++. Nah kami akan menjelaskan sedikit tentang fungsi –fungsi program yang kami gunakan dalam pembuatan tersebut.

Sebelumnya Ini Hasil Project kami saat Program kami dijalankan:



Penjelasan :

Sebelumnya aplikasi Code Bloks yang kami gunakan telah terinstal “glut”. Glut disini merupakan software bantu yang berisi paket-paket untuk membuat suatu vektor beserta fungsi-fungsinya:

```
#include<gl/glut.h> // ini untuk menginisialisasi/ memanggil library grafik  
#include "imgloader.h" // ini untuk memasukan gambar pada tekstur.
```

Selanjutnya, dibawah ini merupakan nama fungsi method – method yang digunakan :

1. Display ()

Disini intinya kita membuat suatu bidang 3D sesuai dengan bangunan yang kita buat, adapun beberapa fungsi pemrograman yang kita gunakan.

Contoh coding:

```
// display, yang kami beri nama void tampil, digunakan untuk menampung method method  
yang berisikan object bangunan.
```

```
void display(void)
```

```
{
```

```
// disini kondisi untuk mengatur kedalaman, dimana jika variable is_depth berlaku maka  
akan melakukan pembersihan layar. Jika else maka akan menampilkan warna yang berisikan  
gambar bangunan.
```

```

    if (is_depth) {
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    }
    else {
        glClear(GL_COLOR_BUFFER_BIT);
        // disini untuk memanggil beberapa method yang berisi tentang gambar yang kita buat.
    }

    glLoadIdentity();
    gluLookAt(0.0f, 0.0f, 3.0f, 0.0f, 0.0f, 0.0f, 1.0f, 0.0f); // mengatur pandangan
    mata
    glRotatef(xrot, 1.0f, 0.0f, 0.0f); //mengatur rotasi x
    glRotatef(yrot, 0.0f, 1.0f, 0.0f); //mengatur rotasi y
    glRotatef(zrot, 0.0f, 0.0f, 1.0f); //mengatur rotasi z

    glScalef(sca + 1, sca + 1, sca + 1); //untuk penskalaan
    glTranslatef(xmov, ymov, zmov); // menghasilkan matriks terjemahan (x, y, z)

    glPushMatrix(); //mendorong matriks diatas

    glBegin(GL_POLYGON);
    glColor3f(0.5, 0.8, 1.0);
    glVertex3f(60.0, -20.0, -99);
    glVertex3f(60.0, -20.0, -125);
    glVertex3f(90.0, -20.0, -125);
    glVertex3f(90.0, -20.0, -99);
    glEnd();

    glBegin(GL_POLYGON);
    glColor3f(0.5, 1.5, 0.5);
    glVertex3f(-80.0, -20.1, 100);
    glVertex3f(-80.0, -20.1, -200);
    glVertex3f(130.0, -20.1, -200);
    glVertex3f(130.0, -20.1, 100);
    glEnd();

    glPopMatrix(); //Memunculkan matriks
    glutSwapBuffers();
}

```

→ Contoh bidang yang tidak ada teksturnya

// ini contoh coding tanpa tekstur, untuk membuat salah satu dinding kanan kiri berada di void display, yang telah kami inisialisasikan diatas.

//Dinding kiri

```
glBegin(GL_POLYGON);  
glColor3f(1.0, 0.5, 0.0);  
glVertex3f(-59.9, 20.0, 17);  
glVertex3f(-59.9, -17.0, 17);  
glVertex3f(-59.9, -17.0, -99);  
glVertex3f(-59.9, 20.0, -99);  
glEnd();
```

ini contoh outputnya tembok :



→ Contoh bidang yang ada teksturnya

// ini contoh coding atap yang ada teksturnya.

// ini untuk mengaktifkan tekstur yang ada di class tekstur.

// atap depan

```
glEnable(GL_TEXTURE_2D);  
glBindTexture(GL_TEXTURE_2D, _textureId2);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);  
glBegin(GL_POLYGON);  
glColor3f(1.0,1.0,1.0);  
glTexCoord2f(1.0f, 0.0f); // ini untuk memposisikan tekstore  
glVertex3f(-65.0, 40.0, -15);  
glTexCoord2f(1.0f, 1.0f);  
glVertex3f(-65.0, 20.0, 30);  
glTexCoord2f(0.0f, 1.0f);  
glVertex3f(65.0, 20.0, 30);  
glTexCoord2f(0.0f, 0.0f); // sampai sini  
glVertex3f(65.0, 40.0, -15);  
glEnd();
```

// coding diatas untuk membuat tekstur atap.

Berikut contoh output dari atap:



Contoh koding diatas merupakan sebagian dari pembuatan kita, dan yang lainnya disesuaikan dengan apa yang kita buat, tetapi caranya tetap sama tinggal menentukan bidangnya saja.

2. Init ()

Berisi metode untuk menginisialisasi OpenGL.

Contoh coding :

// fungsi init dengan nama method lumia bertipe Void, untuk menginisialisasi OpenGL

```
void init(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0); //mengatur warna background

    // ini untuk memposisikan tepat ditengah dan mengatur warna.
    glShadeModel(GL_SMOOTH);
    glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_LINE_SMOOTH);
    glEnable(GL_NORMALIZE);

    is_depth = 1; // memberi nilai 1 pada variabel is_depth
    glMatrixMode(GL_MODELVIEW); // Transformasi object
    glEnable(GL_DEPTH_TEST); // untuk mengaktifkan memory Depth Buffer
    Image* image1 = loadBMP("aspal.bmp"); //mengambil foto untuk tekstur
    //ganti nama file foto anda formatx harus .bmp_textureId = loadTexture(image1);

    delete image1; // ini untuk menghapus gambar tekstur kita setelah dipakai, dan disimpan
    di coding, agar tidak boros memori.

    Image* image2 = loadBMP("atap.bmp"); //mengambil foto untuk tekstur
    //ganti nama file foto anda formatx harus .bmp_textureId2 = loadTexture(image2);

    delete image2; // ini untuk menghapus gambar tekstur kita setelah dipakai, dan disimpan
    di coding, agar tidak boros memori.
}
```

3. Keyboard ()

Membuat fungsi kontrol pada keyboard agar bisa melihat object dari berbagai sudut yang kita inginkan.

Contoh coding:

```
void keyboard(unsigned char key, int x, int y)
{
    // ini untuk tombol keyboard, berfungsi untuk memindahkan object berdasarkan
    // kordinat x,y,z.
    switch (key)
    {
        case 'w':
        case 'W':
            zmov += 3.0; // untuk maju kedepan
            break;
        case 'd':
        case 'D':
            xmov += 3.0; // untuk begeser kesamping kanan
            break;
        case 's':
        case 'S':
            zmov -= 3.0; // untuk mundur kedepan
            break;
        case 'a':
        case 'A':
            xmov -= 3.0; // untuk begeser kesamping kiri
            break;
        case 't':
        case 'T':
            xrot += 3.0; // untuk rotasi/memutar kedepan
            break;
        case 'g':
        case 'G':
            xrot -= 3.0; // untuk rotasi/memutar kebelakang
            break;
        case 'y':
        case 'Y':
            yrot += 3.0; // untuk rotasi/memutar kesamping kanan
            break;
        case 'h':
        case 'H':
            yrot -= 3.0; // untuk rotasi/memutar kesamping kiri
            break;
        case 'u':
        case 'U':
            zrot += 3.0; // untuk rotasi/memutar disumbu z ke kanan
```

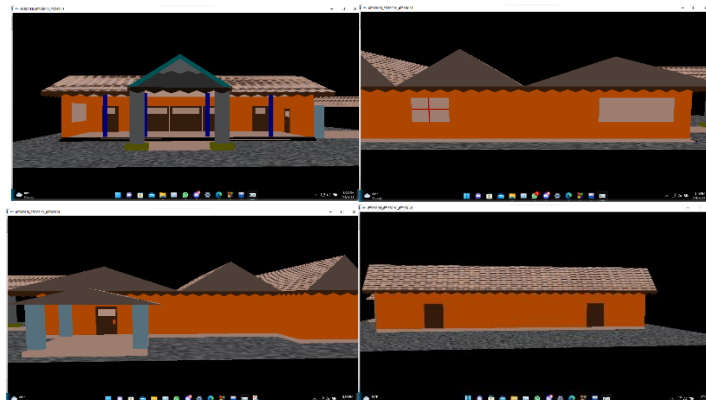


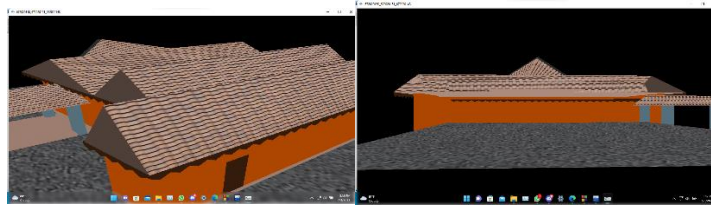
```

        break;
    case 'j':
    case 'J':
        zrot -= 3.0; // untuk rotasi/memutar disumbu z ke kiri
        break;
    case 'i':
    case 'I':
        sca += 0.1;
        break;
    case 'k':
    case 'K':
        sca -= 0.1;
        break;
    case '1':
        ymov += 3.0; // untuk naik keatas pada sumbu y
        break;
    case '2':
        ymov -= 3.0; // untuk turun kebawah pada sumbu y
        break;
    case '3': // untuk transparan
        if (is_depth)
        {
            is_depth = 0;
            glDisable(GL_DEPTH_TEST);
        }
        else
        {
            is_depth = 1;
            glEnable(GL_DEPTH_TEST);
        }
    }
    display(); // ini untuk memanggil void display kita.
}

```

Contoh gambar object dilihat beberapa sisi, dan transparent:





4. Mouse ()

Membuat fungsi kontrol pada keyboard agar bisa melihat object dari berbagai sudut yang kita inginkan.

Contoh coding:

```
void idle() {
    if (!mouseDown) {
        xrot += 0.3f; // untuk memutar menggunakan mouse pada sumbu x ke kanan.
        yrot += 0.4f; // untuk memutar menggunakan mouse pada sumbu y ke atas.
    }
    glutPostRedisplay();
}

void mouse(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // update state tombol, jadi kalo di klik kiri mouse baru diaktifkan rotasi

        mouseDown = true;
        // update posisi mouse, supaya gampang tracking/atau membaca pergerakan mouse untuk rotasi
        xdiff = x - yrot;          /* Posisi mouse */
        ydiff = -y + xrot;
    }
    else
        mouseDown = false;
}

void mouseMotion(int x, int y) {
    if (mouseDown) {
        /* Kalo tombol kiri mouse diklik, lakukan rotasi jika mouse digerakan*/

        yrot = x - xdiff;
        xrot = y + ydiff;

        glutPostRedisplay();
    }
}
```

5. Ukuran ()

Disini berfungsi untuk mengubah kembali ukuran windows. Fungsi Reshape untuk mengatur pemanggilan kembali fungsi glut reshape. Pada fungsi ini, user dapat mengubah rasio dari window yang mempunyai lebar window dan tinggi window.

Contoh koding :

```
// resize dengan nama method ukuran dan mempunyai paramater integer lebar (
lebar window ) dan integer tinggi ( tinggi widow ).
```

```
void ukuran(int lebar, int tinggi)
{
    // ini merupakan suatu kondisi dimana, jika tinggi bernilai 0 maka tinggi bernilai 1
    if (tinggi == 0) tinggi = 1;
    // Ini untuk memposisikan object yang kita buat tepat ditengah, dalam artian koordinat
    x,y,z : (0,0,0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    // Ini untuk mengatur letak object
    gluPerspective(50.0, lebar / tinggi, 5.0, 1000.0);
    glTranslatef(0.0, -5.0, -200.0);
    glMatrixMode(GL_MODELVIEW);
}
```

6. Main ()

Ini merupakan fungsi utama dari dari program yang kami buat, yang berisikan beberapa fungsi untuk mengatur ukuran windows aplikasi, memberi nama aplikasi yang kita buat, menentukan letak windows aplikasi di layar monior kita, memanggil fungsi – fungsi method yang kita buat.

Contoh koding :

```
int main(int argc, char** argv)
{
    // Untuk inisialisasi glut
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    // Untuk mengatur ukuran Windows, disini kami mengaturnya dengan ukuran 800 x 600
    pixel
    glutInitWindowSize(800, 600);
    // Untuk mengatur letak posisi windows , disini ditentukan titik pertama windows kami
    menyetingnya dengan titik pertama kooordinat x,y ( 250,80 ).
    glutInitWindowPosition(250, 80);
    // Disini untuk memberi nama windows, kami memberi nama Nim kami
    glutCreateWindow("672020150_672020151_672020148");
}
```

```

init(); // Memanggil init
glutDisplayFunc(display); //Untuk menampilkan frame yang diambil dari void display() ,
jadi disini untuk menampilkan apa yang telah kita buat pada void display ();
glutKeyboardFunc(keyboard); // Untuk menjalankan fungsi keyboard, yang diambil dari
void Keyboard() → fungsi keyboard biasa.
glutMouseFunc(mouse); // Untuk menjalankan fungsi mouse, yang diambil dari
void mouse()
glutMotionFunc(mouseMotion); // Untuk melakukan fungsi rotasi pada mouse, yang
diambil dari void mouseMotion()
glutReshapeFunc(ukuran); // Untuk mengubah kembali ukuran windows, diambil dari
void ukuran()
glutMainLoop();// Untuk meLooping atau mengulang fungsi main.
return 0;
}

```

Itu tadi merupakan penjelasan singkat dari kami mengenai fungsi – fungsi program, Syntax – syntax program yang kita buat dan akhirnya bisa membuat aplikasi untuk menampilkan sebuah Object berbasis vektor dengan OpenGL. Kita sebagai manusia biasa yang tidak sempurna, pastinya memiliki kesalahan ataupun kelemahan. Maka dari itu dalam pembuatan Tugas Rancang ini kami mengalami beberapa kesulitan – kesulitan yang kami hadapi, tetapi kami coba mengatasi permasalahan tersebut dengan kemampuan yang kami punya, seperti :

1. Object tidak bisa kita bawa kemanapun.

Object yang kita buat merupakan sebuah bangunan besar dan sulit untuk kita melihatnya dari atas. Dan juga tidak bisa kita bawa pulang untuk kita lihat. Untuk itu solusi yang kita gunakan adalah dengan melihat object melalui google.

2. Keditailan object yang rumit.

Seperti yang kita tahu bahwa object yang kita teliti memiliki tingkat kerumitan yang menurut kami cukup sulit karena keditailan dari bangunan tersebut, seperti ukir-ukurnya, detail teksturnya, dan juga kita tidak bisa melihat detailnya dari atas. Untuk itu kami hanya bisa membuat sebisanya agar terlihat mirip seperti aslinya.

3. Tekstur object.

Kami mengalami kesulitan dalam membuat tekstur dari object yang kita buat seperti, tekstur tanah, aspal dan lain lain, agar mirip sekali dengan aslinya. Untuk itu kami mencari tekstur yang terlihat mirip dengan aslinya.

4. Object interior

Kami mengalami kesulitan untuk membuat object interior, dikarenakan keterbatasan dokumentasi objek interior, kami hanya mengandalkan google serta aplikasi lainnya untuk melihat keditailan objek yang kami buat .

4. KESIMPULAN

Kami dapat menyimpulkan, Dari penjelasan diatas bahwasannya, Bahasa C++ dapat digunakan untuk mendesain suatu bangunan 3D maupun 2D seperti yang kami buat. Selain itu dalam bahasa C++ terutama software OpenGL juga bisa untuk membuat suatu Game, animasi gerak, dan lain sebagainya. Serta didalam program tersebut banyak sekali macam-macam fungsi yang dapat digunakan seperti yang telah kami uraikan diatas, seperti glVertex, glColor, dan lain sebagainya. Sekian laporan dari kami kiranya pembaca sekalian dapat memahami apa yang telah kami sampaikan dalam laporan ini. Dan apabila terdapat kesalahan dalam penulisan maupun penjelasan yang sengaja maupun tidak sengaja kami mohon maaf sebesar-besarnya. Terima kasih.

5. SUMBER PUSTAKA

Grafika Komputer, Evangs Mailoa (n.d.), diambil 26 November 2013, dari <http://evangsmailoa.wordpress.com/materi-kuliah/grafika/>