

Grafkom



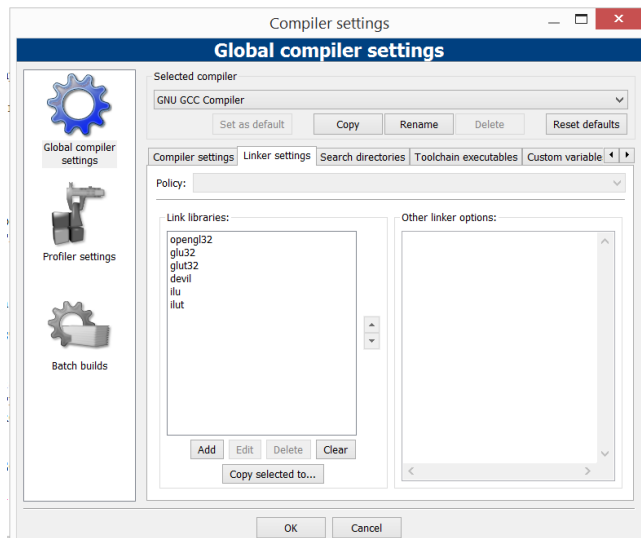
M Faris Makarim
Narendra Hanif
Brilan T Nugraha
Aditya Putra Ferza

KONFIGURASI IDE

IDE yang kami gunakan dalam pembuatan proyek ini adalah codeBlock sebelum memulai perlu melakukan beberapa pengaturan dalam code block agar dapat menjalankan dan juga mengompilasi program, sebelum itu pastikan kita memiliki devil, freeglut, glew dan juga glut, apabila belum memiliki file-file di atas dapat mengunduh file di bawah ini

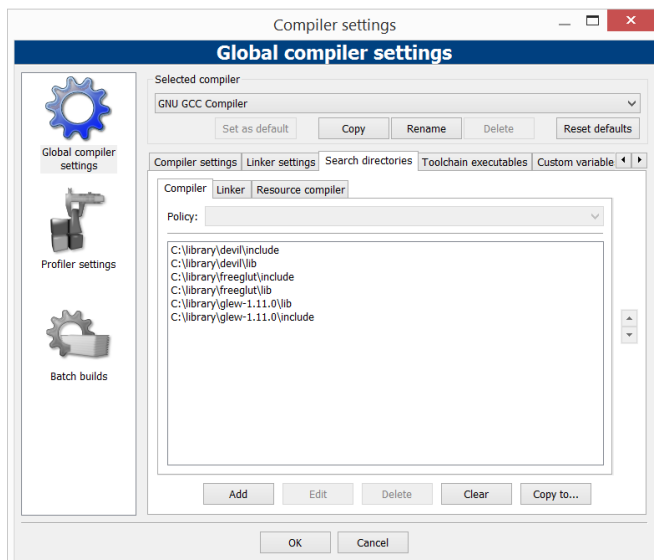
<https://drive.google.com/file/d/0Bx8Gi42N2G1EUUt0cEpfLVIEUVE/view?usp=sharing>

setelah kita buka aplikasi code block pilih setting-→ compiler→ Linker setting



Tulis link libraries seperti di atas

Setelah itu pilih search directories memberitahukan letak library yang kita gunakan



PEMBUATAN OBJEK

Pada program ini objek yang digunakan berformat obj, program yang digunakan untuk menggunakan blender dan juga sketchup

PEMBERIAN TEKSTUR

Setelah berhasil membuat objek di blender, objek yang dibuat di export ke sketch up dengan format dae untuk pemberian tekstur

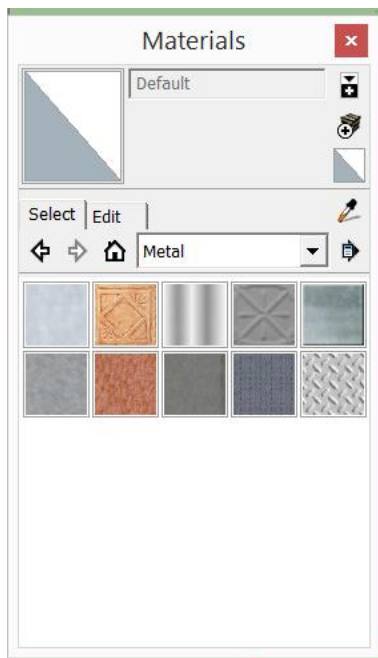
PEMBERIAN TEKSTUR DENGAN SKETCHUP

Pada sketchup pilih menu file → pilih import, kemudian kita pilih file dae yang ingin kita berikan tekstur.

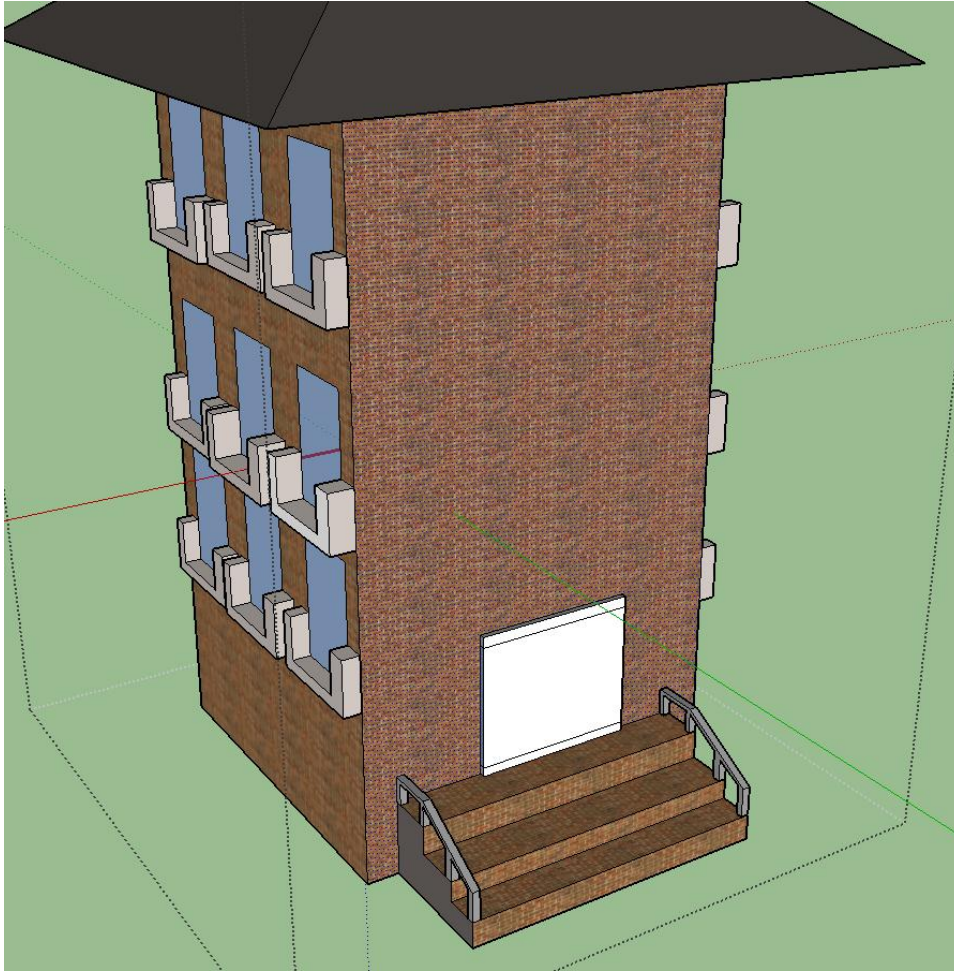
Untuk memberikan tekstur pilih pain bucket



Setelah kita memilih pain bucket akan tampil

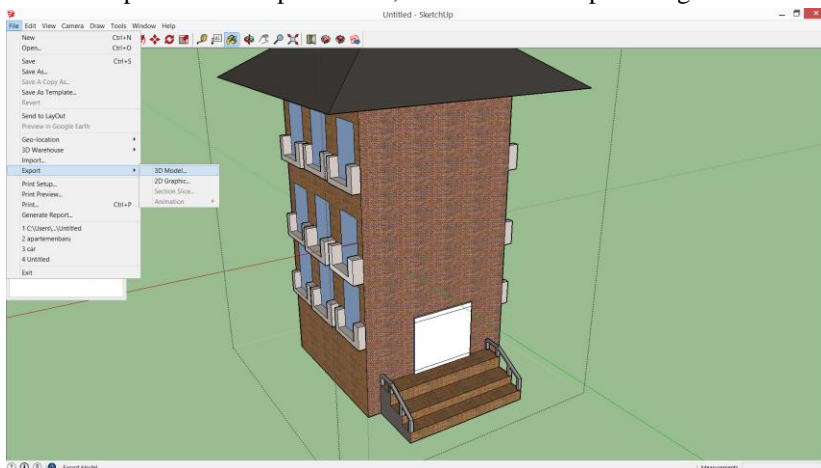


Kita pilih tekstur yang ingin kita tambahkan pada permukaan objek

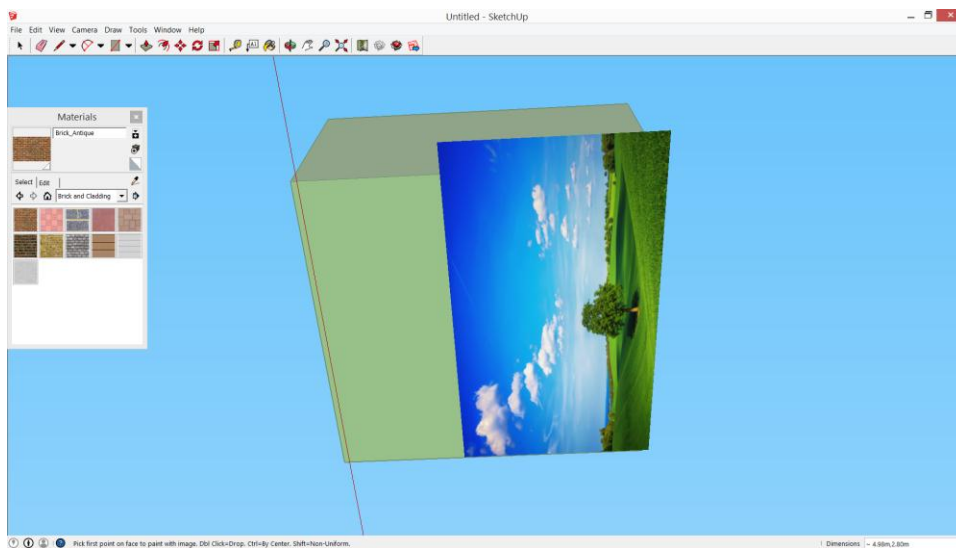
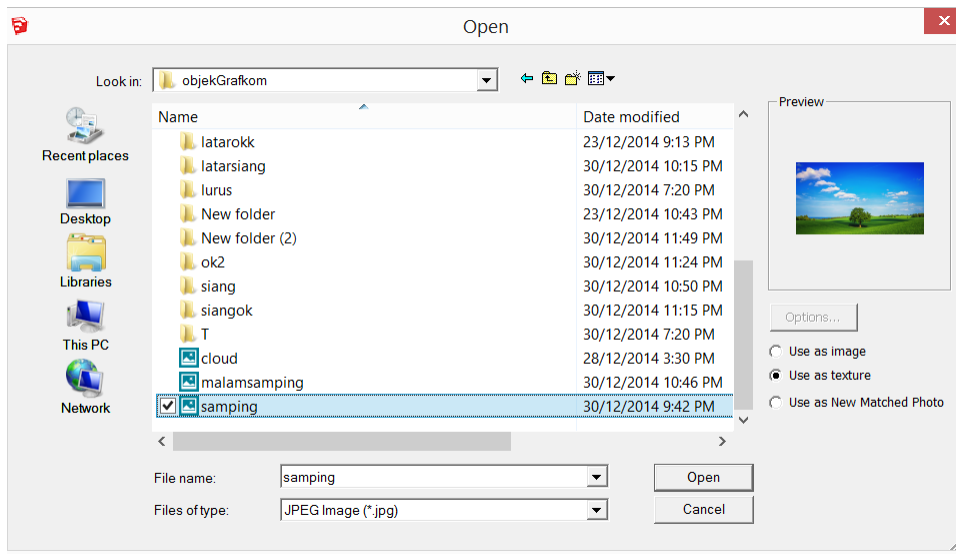


Seperti contoh diatas, memilih memilih tekstur brick

Kemudian pilih file →Export→ 3D , kemudian kita export dengan format file .obj nanti



Nanti akan terbentuk 3 file, yaitu .obj, .mtl dan juga folder yang berisi gambar



Tempelkan gambar pada permukaan objek

SOURCE CODE

VIEW LOAD DAN TEXTURE

Untuk menload objek dapat dilakukan dengan cara :

```
glPushMatrix();  
  
    if (!sample3) {  
  
        sample3 = glmReadOBJ("data/i.obj");  
  
        if (!sample3) exit(0);  
  
        glmUnitize(sample3);  
  
    }  
  
glPopMatrix();
```

sebelumnya kita perlu melakukan inisialisa sampel 3di awal program

```
#include <Windows.h>  
#include <stdio.h>  
#include <gl/GL.h>  
#include <gl/GLU.h>  
#include <gl/glut.h>  
#include "glm.h"  
  
GLMmodel *sample3, *s;
```

Bila ingin menambahkan tekstur kita tambahkan sebelum glPopMatrix()

```
glmDraw(sample3, GLM_SMOOTH | GLM_TEXTURE);
```

kita juga dapat mentranslasi objek dengan cara menambahkan kode program dibawah :

glTranslatef(0,1,-5); -> mentranslasi x = 0 , y = 1, z=-5

glScalef(30,15,20); -> objek di perbesar x sebesar 30, y 15 dan z 20

TRANSLASI DAN ROTASI

agar mobil dapat berjalan dapat kita lakukan dengan melakukan translasi objek mobil pada

```
glLoadIdentity();
```

```
glPushMatrix();
```

```
if (!sample2) {
```

```
    sample2 = glmReadOBJ("data/car.obj");
```

```
    if (!sample2) exit(0);
```

```
    glmUnitize(sample2);
```

```
}
```

```
glTranslatef(x,y,z);
```

```
glScalef(1,1,1);
```

```
glRotatef(angle,5,240,2);
```

```
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
```

```
glColor3ub(255,255,255);
```

```
glmDraw(sample2, GLM_SMOOTH | GLM_TEXTURE);
```

```
glPopMatrix();
```

setelah kita load objek kita tambahkan glTranslate dan beri nilai dengan variable x,y,z agar dapat kita ubah ubah ketika menerima inputan dari keyboard

setelah itu kita buat fungsi keyboard

```
void keyPressed(unsigned char key, int p, int q){
```

```
    if(key=='x') x+=2;
```

```
    if(key=='y') y+=2;
```

```
    if(key=='z') z+=2;
```

```
}
```

Denagn menambahkan fungsi keyboard pressed kita dapat mentranslasi objek mobil dengan menekan keyboard

CEK COLLISION

Agar mobil hanya berjalan diatas permukaan jalan dan berjalan dia atas track yang ditentukan, dapat dilakukan dengan mengecek collision :

```
void keyPressed(unsigned char key, int p, int q){  
    if(key=='w')  
    {  
        if(angle == 90 || angle == -270)  
        {  
            if((x <= 6.8 && x >= 6.2) || (x >= -7.6 && x <= -7))  
                if(z < -5.2)z += 0.1;  
        }  
        else if(angle == 180 || angle == -180)  
        {  
            if((z >= -12.7 && z < -11.9)||(z >= -8.9 && z <= -8.3)||(z >= -5.5 && z <= -4.9))  
                if(x < 6.5) x += 0.1;  
        }  
        else if(angle == 270 || angle == -90)  
        {  
            if((x <= 6.8 && x >= 6.2) || (x >= -7.6 && x <= -7))  
                if(z > -12.5 )z -=0.1;  
        }  
        else if(angle == 0)  
        {  
            if((z >= -12.7 && z < -11.9)||(z >= -8.9 && z <= -8.3)||(z >= -5.5 && z <= -4.9))  
                if(x > -7) x -= 0.1;  
        }  
    }  
}
```



```

if(key=='s')
{
    if(angle == 90 || angle == -270)
    {
        if((x <= 6.8 && x >= 6.2) || (x >= -7.6 && x <= -7))
            if(z > -12.5)z -=0.1;
    }
    else if(angle == 180 || angle == -180)
    {
        if((z >= -12.7 && z < -11.9)||(z >= -8.9 && z <= -8.3)||(z >= -5.5 && z <= -4.9))
            if(x > -7) x -= 0.1;
    }
    else if(angle == 270 || angle == -90)
    {
        if((x <= 6.8 && x >= 6.2) || (x >= -7.6 && x <= -7))
            if(z < -5.2)z +=0.1;
    }
    else if(angle == 0)
    {
        if((z >= -12.7 && z < -11.9)||(z >= -8.9 && z <= -8.3)||(z >= -5.5 && z <= -4.9))
            if(x < 7) x += 0.1;
    }
}

if(key=='a')
{
    angle += 30;

    if(angle >= 360)

```

```
    angle -= 360;

    if(angle <= -360)

        angle += 360;
}

if(key=='d')
{
    angle -= 30;

    if(angle >= 360)

        angle -= 360;

    if(angle <= -360)

        angle += 360;
}
}
```