

1. Tunjukkan bagian mana yang disebut stack dari kode di atas dalam bentuk PDF, dikumpulkan di github yang terkait?

```
void drawCar() {  
    glPushMatrix();  
    glTranslatef(-0.2f, -0.3f, 0.0f);  
  
    // Badan dan atap mobil  
    glColor3f(0.6f, 0.0f, 0.0f);  
    drawRectangle(-0.3f, 0.1f, 0.6f, 0.2f);  
    glColor3f(1.0f, 0.0f, 0.0f);  
    drawRectangle(-0.05f, 0.2f, 0.2f, 0.15f);  
  
    // Jendela  
    glColor3f(0.0f, 0.0f, 1.0f);  
    drawRectangle(0.02f, 0.17f, 0.08f, 0.05f);  
  
    // Roda  
    glColor3f(0.75f, 0.75f, 0.75f);  
    drawCircle(-0.2f, -0.1f, 0.075f);  
    drawCircle(0.2f, -0.1f, 0.075f);  
  
    glPopMatrix();  
}
```

Bagian ini menggunakan **glPushMatrix** untuk menyimpan kondisi transformasi dan **glPopMatrix** untuk mengembalikan kondisi awal setelah menggambar mobil.

2. Kenapa kita harus menggunakan Push dan Pop yang ada pada stack? Mengapa harus menggunakannya dan apa perbedaannya dengan **glLoadIdentity** yang mana sama-sama melakukan reset terhadap translasi dan rotasi?

Alasan menggunakan `glPushMatrix` dan `glPopMatrix` dalam stack adalah karena Jika kita melakukan translasi dan rotasi tanpa stack, transformasi tersebut akan mempengaruhi objek lain. Kemudian setelah menggambar objek, kondisi transformasi dikembalikan agar tidak berdampak pada objek berikutnya.

Perbedaan dengan `glLoadIdentity`:

1. `glPushMatrix` menyimpan keadaan saat ini di stack, lalu kita bisa melakukan transformasi dan mengembalikan keadaan awal dengan `glPopMatrix`.
2. `glLoadIdentity` hanya mereset transformasi tanpa menyimpan keadaan sebelumnya.

3. Apa perbedaan rotasi/translasi di dalam stack dan diluar stack?

1. Didalam stack efek transformasi hanya berlaku dalam `glPushMatrix` dan `glPopMatrix`, kalau diluar stack efek transformasi berlaku secara global dan mempengaruhi objek berikutnya.
2. Didalam stack Lebih terstruktur dan lebih mudah dikelola, kalau diluar stack Rentan menyebabkan transformasi yang tidak diinginkan