

Nama : Shopia Mudjahidah

NIM : 24060123120010

Lab : GKV E1

**Pertanyaan:**

1. Tunjukkan bagian mana yang disebut stack dari kode di atas!
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?
3. Apa perbedaan rotasi/translasi di dalam stack dan diluar stack?

**Jawaban:**

1. Bagian kode yang menggunakan stack adalah ketika `glPushMatrix()` dan `glPopMatrix()` digunakan. Pada kode ini, stack digunakan dalam fungsi `drawTree()`. `glPushMatrix()` digunakan untuk menyimpan status transformasi saat ini di stack. `glTranslatef(x, y, 0.0f)` digunakan untuk memindahkan objek ke posisi yang ditentukan. `glScalef(scale, scale, 1.0f)` digunakan untuk memperbesar atau memperkecil objek. `glPopMatrix()` digunakan untuk mengembalikan transformasi sebelumnya sehingga perubahan tidak mempengaruhi objek lain.
2. Menggunakan `glPushMatrix()` dan `glPopMatrix()` digunakan untuk menyimpan dan mengembalikan status matriks transformasi agar perubahan hanya berlaku pada objek tertentu. `glPushMatrix()` menyimpan status sebelum transformasi dilakukan, sedangkan `glPopMatrix()` mengembalikan status tersebut setelah objek digambar. Berbeda dengan `glLoadIdentity()` yang langsung mereset semua transformasi ke keadaan awal tanpa menyimpannya, sehingga semua perubahan sebelumnya hilang. Dengan menggunakan stack, kita dapat memastikan bahwa setiap objek memiliki transformasi yang tidak mempengaruhi objek lain dalam scene.
3. Rotasi dan translasi di dalam stack hanya berlaku sementara untuk objek tertentu karena setelah objek digambar, transformasi dapat dikembalikan ke kondisi awal menggunakan `glPopMatrix()`. Hal ini memungkinkan pengaturan transformasi lokal tanpa mempengaruhi objek lain. Sebaliknya, jika transformasi dilakukan di luar stack, efeknya akan berlaku untuk semua objek berikutnya, sehingga sulit mengontrol posisi dan orientasi masing-masing objek secara independen. Oleh karena itu, penggunaan stack sangat penting dalam mengelola transformasi agar setiap objek tetap berada di posisi dan skala yang diinginkan tanpa mengganggu elemen lain dalam scene.