Berikut adalah jawaban dari pertanyaan yang diberikan:

1. **Fungsi glColor3f(1.0f, 0.0f, 0.0f)**  
   Fungsi ini digunakan untuk mengatur warna dalam OpenGL dengan format RGB (Red, Green, Blue).
   * 1.0f pada parameter pertama berarti merah penuh.
   * 0.0f pada parameter kedua berarti tidak ada hijau.
   * 0.0f pada parameter ketiga berarti tidak ada biru.  
     Jadi, fungsi ini akan menghasilkan warna **merah**.  
     Huruf f di belakang nilai menunjukkan bahwa angka tersebut bertipe **float**.

Untuk menghasilkan warna **hijau**, kita perlu mengatur parameter merah menjadi 0.0f, hijau menjadi 1.0f, dan biru tetap 0.0f:

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

1. **Apakah ada fungsi glColor4f dalam OpenGL?**  
   Ya, dalam OpenGL terdapat fungsi glColor4f(r, g, b, a), yang mirip dengan glColor3f, tetapi dengan tambahan parameter **alpha** untuk transparansi.
   * r, g, b adalah warna merah, hijau, dan biru seperti pada glColor3f.
   * a adalah **alpha**, yang menentukan transparansi (1.0 = tidak transparan, 0.0 = transparan sepenuhnya).  
     Contoh:
2. glColor4f(0.0f, 1.0f, 0.0f, 0.5f); // Warna hijau dengan transparansi 50%
3. **Fungsi glVertex3f(-0.10, -0.10, 0.00)**  
   Fungsi ini digunakan untuk menentukan koordinat titik dalam ruang 3D dengan format (x, y, z).
   * -0.10 adalah koordinat **x** (horizontal).
   * -0.10 adalah koordinat **y** (vertikal).
   * 0.00 adalah koordinat **z** (kedalaman dalam ruang 3D).  
     Fungsi ini sering digunakan dalam **pembuatan objek primitif**, seperti segitiga atau kotak.
4. **Fungsi glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGBA)**  
   Fungsi ini digunakan untuk mengatur mode tampilan dalam OpenGL menggunakan GLUT.
   * GLUT\_SINGLE berarti menggunakan **single buffering**, yang berarti gambar akan langsung ditampilkan tanpa buffering tambahan.
   * GLUT\_RGBA berarti menggunakan mode warna **RGBA**, di mana setiap warna direpresentasikan dalam bentuk **Red, Green, Blue, dan Alpha (transparansi)**.

Jika kita ingin menggunakan double buffering untuk tampilan yang lebih halus, kita bisa menggunakan GLUT\_DOUBLE sebagai gantinya.