

2. LANDASAN TEORI

2.1 Tujuan Pustaka

Untuk menggambar grafik pada komputer, biasanya dimulai dengan *pixel* (Angel, 2005). Gambar sebuah titik dimanapun dalam layar dan pilih warna. Dari konsep tersebut, dikembangkan menjadi garis, poligon, lingkaran, dan bentuk-bentuk lain. Namun dengan OpenGL, menggambar di komputer secara fundamental berbeda. Hal yang perlu diperhatikan adalah posisi koordinat. OpenGL akan mendapatkan titik-titik yang ditentukan, kemudian memproyeksikan titik-titik tersebut dalam implementasinya sendiri (Angel, 2005). Penggunaan fungsi “glVertex” digunakan untuk menentukan sebuah titik dalam ruang 3D di OpenGL. Dalam pemrosesan vertex, setiap vertex akan diproses secara independen (Angel & Shreiner, 2012). Pemrosesan vertex digunakan untuk transformasi koordinat dan komputasi warna tiap vertex.

Seperti obyek 2 dimensi, obyek geometris 3 dimensi dapat didefinisikan menggunakan seperangkat entitas sederhana yang terbatas (Angel & Shreiner, 2012). Obyek geometris dasar ini dan hubungan di antaranya dapat dijelaskan menggunakan tiga tipe dasar: skalar, titik, dan vektor. Dengan mengatur skalar, titik, dan vektor tersebut, dapat dihasilkan obyek geometris yang beragam. Permukaan kuadrat dijelaskan oleh persamaan aljabar implisit dimana setiap suku adalah polynomial berbentuk $x^i y^j z^k$, dengan $i + j + k \leq 2$. Quadrik apapun dapat dituliskan dalam formula: $q(x, y, z) = a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + a_{33}z^2 + 2a_{23}yz + 2a_{13}xz + b_1x + b_2y + b_3z + c = 0$ (Angel & Shreiner, 2012).

2.2 Game-based Learning

Sebuah *game* dapat didefinisikan sebagai seluruh permainan tersusun yang terdiri dari peraturan, tujuan dan tantangan yang dilakukan untuk hiburan (Cheng et al., 2015). *Game-based learning* merujuk ke pencapaian hasil belajar yang ditetapkan melalui konten *game* dan meningkatkan pembelajaran dengan melibatkan kemampuan pemecahan masalah dan tantangan yang memberi peserta didik, juga sebagai pengguna, rasa pencapaian dan kebanggaan. (Qian & Clark, 2016). *Game-based learning* juga merupakan bagian dari *serious games* (Abt, 1970). *Serious games* mendalami masalah intelektual dan sosial di berbagai macam bidang.

Game-based learning memiliki beberapa keuntungan dibandingkan metode belajar yang lain. Salah satunya, peningkatan aktivasi di area otak yang terkait dengan emosi dan pemrosesan penghargaan (Greipl et al., 2021). Dengan penggunaan *game-based learning*, ditemukan bahwa peserta didik lebih terlibat secara emosional. Selain itu, telah ditemukan efek positif yang signifikan dari *gamification* pada hasil belajar kognitif, motivasi dan perilaku (Sailer & Homner, 2020, Zapata-Caceres & Martin-Barroso, 2021, Lopez-Fernandez et al., 2021, Lopez-Gazpio, 2021).

2.3 Grafika Komputer Obyek 3 Dimensi

Grafika komputer yang akan dibahas dalam skripsi ini adalah mengenai cara menggambarkan obyek 3 dimensi dalam komputer. Obyek 3 dimensi dalam grafika komputer sebenarnya adalah gambar 2 dimensi dalam layer monitor komputer yang memberikan sebuah ilusi *depth*, atau dimensi ketiga (Wright et al., 2007). Proses penggambaran suatu deskripsi geometric dari obyek 3 dimensi menjadi gambar di layer monitor disebut *render*. Dalam *rendering* ditambahkan beberapa efek agar obyek di layer tersebut memiliki *depth*. Efek-efek tersebut yaitu *perspective*, warna, *shading*, cahaya dan bayangan, *texture mapping*, *fog*, *blending and transparency*, serta *antialiasing*.

2.3.1 Obyek Mesh

Mirip dengan cara menggambar obyek 2 dimensi, menggambar obyek 3 dimensi juga dimulai dari sebuah titik. Namun, perbedaannya apabila pada obyek 2 dimensi yang diperhatikan adalah koordinat fisik pada layar dan pixel, pada obyek 3 dimensi yang diperhatikan adalah posisi koordinat dalam volume penglihatan (Wright et al., 2007). Sebuah titik dalam ruang tersebut atau yang disebut "*lowest common denominator*" adalah *vertex*.

Setelah titik, selanjutnya adalah garis. Menggambar garis pada obyek 3 dimensi memerlukan 2 *vertices* agar dapat menggambarkan garis diantara *vertices* tersebut (Angel, 2005, Wright et al., 2007). Garis tersebut memiliki istilah lain yaitu *edge*. Setiap *edge* terdiri dari 2 *vertices*, namun sebuah *vertex* dapat memiliki lebih dari 1 *edge*. Maka sebuah kubus dapat dikatakan terdiri dari 8 titik *vertices* dan 6 garis *edges*.

Vertex dan *edge*, titik dan garis dapat membentuk bermacam-macam obyek (Wright et al., 2007). Namun, obyek yang terbentuk masih belum memiliki warna dan hanya membentuk kerangka obyek bukan suatu obyek yang padat. Untuk menggambar suatu obyek padat,

diperlukan *polygons*. Sebuah *polygon* adalah sebuah bentuk tertutup yang merupakan basis dari semua komposisi obyek padat. *Polygon* yang paling sederhana adalah segitiga dengan 3 sisinya. Apabila menggunakan contoh kubus, maka suatu obyek kubus sederhana terdiri dari 12 *polygons* segitiga dengan 2 *polygonsi* pada tiap permukaannya. Permukaan dari kubus tersebut juga dapat disebut dengan istilah *face*.

2.3.2 Obyek Quadrik

Dengan menggunakan *vertex*, *edge*, dan *face*, berbagai macam jenis obyek 3 dimensi dapat digambar. Namun, seluruh obyek tersebut terbatas hanya menggunakan bidang datar. Apabila ingin menggambar obyek yang memiliki bidang lengkung seperti bola, tabung, atau kerucut, perlu menggunakan garis lengkung atau kurva. Maka digunakanlah kurva Bezier yang dapat merepresentasikan kurva dan permukaan dengan menspesifikasikan sekumpulan kecil titik control (Wright et al., 2007).

Sebuah kurva memiliki satu titik awal, panjang, dan satu titik akhir (Wright et al., 2007). Kurva adalah sebuah garis yang berlekuk-lekuk. Pada kurva Bezier, titik awal dan akhir dari kurva adalah bagian dari kurva, sedangkan titik-titik diantara titik awal dan akhir berfungsi sebagai magnet yang menarik kurva ke arah titik tersebut.

2.3.3 Transformasi Obyek

Sebelumnya telah dibahas bahwa obyek 3 dimensi dalam grafika komputer sebenarnya adalah gambar 2 dimensi di layar monitor dengan ilusi bahwa benda tersebut adalah 3 dimensi. Proses pengubahan data 3 dimensi ke data 2 dimensi disebut *projection* (Wright et al., 2007). Bentuk *projection* yang paling sederhana adalah *orthographic projection* (Angel, 2005). Cara kerja *orthographic projection* yaitu dengan spesifikasi sebuah kotak atau persegi volume pengelihatan. Apapun yang diluar volume pengelihatan ini tidak digambar. Obyek yang berukuran sama juga terlihat sama besar walaupun berbeda jarak. *Perpective projection* adalah jenis *projection* kedua. *Perspective projection* menambahkan efek jarak sehingga obyek yang dekat terlihat lebih besar dan obyek yang jauh terlihat lebih kecil.

Ketika mentransformasi sebuah obyek, pengguna akan selalu mereferensikan *projection* sehingga dapat melihat perubahan yang dilakukan (Wright et al., 2007). Transformasi membuat pengguna dapat memutar (rotasi), menggerakkan (translasi), membesarkan dan mengecilkan (skala) obyek.

2.3.3.1 Translasi

Translasi memindahkan suatu obyek di ruang 3 dimensi. Pengoperasian translasi adalah dengan menambahkan pemindahan pada setiap *vertex* dari obyek (Angel, 2005). Maka, apabila $(x, y, z, 1)$ adalah representasi koordinat dari *vertex* pada (x, y, z) , sebuah translasi akan menggerakkan *vertex* ini ke $(x + d_x, y + d_y, z + d_z, 1)$. Kemudian, proses yang sama akan dilakukan pada *vertices* lain yang dimiliki obyek.

2.3.3.2 Rotasi

Rotasi lebih kompleks dibandingkan translasi. Untuk merotasikan suatu obyek, diperlukan sebuah posisi titik pusat rotasi, kemudian sudut rotasi dari obyek (Angel, 2005, Wright et al., 2007). Dalam titik pusat rotasi, titik tersebut tidak mengalami rotasi. Rotasi dilakukan terhadap titik tersebut sesuai dengan sudut yang dimasukkan.

2.3.3.3 Skala

Transformasi skala mengubah ukuran suatu obyek menjadi lebih besar atau kecil dengan cara memperluas atau menyusut seluruh *vertices* obyek searah tiga axis sesuai faktor yang dispesifikasikan (Angel, 2005, Wright et al., 2007). Skala tidak harus seragam, dan bisa digunakan untuk meregangkan dan menekan obyek searah direksi yang berbeda. Contohnya, sebuah operasi skala $(2.0f, 2.0f, 1.0f)$ dilakukan pada sebuah kubus. Maka, kubus akan berubah menjadi balok.