

Penggunaan Quaternion pada SimCity 5

Fitra Rahmamuliani 13513095
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13513095@std.stei.itb.ac.id

Abstrak— *Simulation Games* adalah sebuah kategori permainan yang mencoba untuk secara akurat menggambarkan situasi dunia nyata, bentuk, dan peristiwa seakurat mungkin. *Simulation Game* dapat dilihat sebagai cara untuk menurunkan kemungkinan hasil dari suatu tindakan (atau proses). *Simulation games* mengikuti filsafat “*Learning by Doing*”. *Simulation Game* memiliki segudang aplikasi dalam pendidikan yang dapat digunakan secara efektif untuk mengajarkan berbagai mata pelajaran dan topik. Ada beberapa jenis permainan simulasi, tapi mungkin hanya tiga jenis yang sangat terkenal, yaitu *Racing Simulator*, *Simulator Flight*, dan jenis permainan ‘Sim’ yang meniru variabel lingkungan ke dalam permainan. Salah satu permainan berkategori *simulation games* ini adalah SimCity5 dengan jenis permainan ‘Sim’. Grafis tiga dimensi adalah grafis yang menggunakan representasi tiga dimensi data geometris (yang biasa digunakan adalah Chartistian) yang disimpan dalam komputer dengan tujuan melakukan perhitungan dan *rendering* gambar 2D. Pada SimCity5 ini memiliki tampilan grafis tiga dimensi yang diterapkan melalui prinsip quaternion. Maka dari itu, makalah ini menjelaskan tentang penggunaan quaternion pada permainan SimCity5.

Kata Kunci—Quaternion, SimCity5, Simulation, Game, 3D, Grafis.

I. PENDAHULUAN

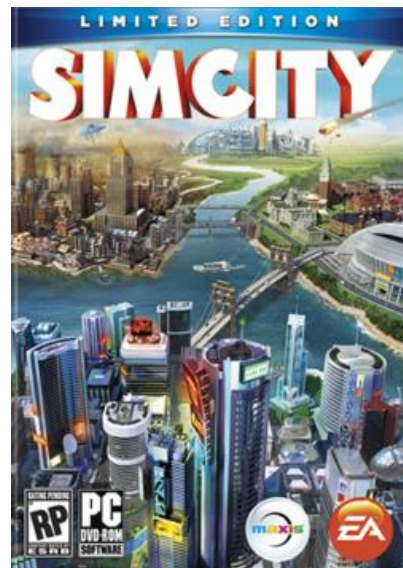
Simulation Games adalah sebuah kategori permainan yang mencoba untuk secara akurat menggambarkan situasi dunia nyata, bentuk, dan peristiwa seakurat mungkin. Kategori permainan ini dapat dilihat sebagai cara untuk menurunkan kemungkinan hasil dari suatu tindakan (atau proses). *Simulation games* ini mengikuti filsafat “*Learning by Doing*” yang memiliki segudang aplikasi dalam pendidikan yang dapat digunakan secara efektif untuk mengajarkan berbagai mata pelajaran dan topik.

Mengingat sejarah *video game*, mungkin bentuk paling awal dan paling banyak diterima dari *simulation game* adalah simulator kendaraan. Namun saat ini, segala macam simulasi telah berputar dengan banyak orang yang telah menunjukkan minat dalam simulasi berbagai aspek kehidupan nyata dalam dunia digital. Sebagai permainan yang dikembangkan selama bertahun-tahun, sub-genre baru dari simulator telah muncul. Salah satu genre adalah konstruksi dan manajemen. Dengan keberhasilan awal seperti Will Wriugh SimCity (awalnya berjudul City Planner 1.0), konsep memiliki kontrol penuh atas pembangunan dan pengelolaan kota, taman, dan bahkan seluruh ekosistem ditemukan menjadi formula ajaib untuk

menarik penggemar selama bertahun-tahun.

SimCity5 adalah pengembangan dari Will Wriugh SimCity sebuah pembangunan kota dan simulasi perencanaan kota secara besar-besaran, memiliki tipe *multiplayer online game* yang dikembangkan oleh Maxis, anak perusahaan dari Electronic Arts dan diterbitkan ke seluruh dunia sejak tanggal 5 Maret 2013. Pada permainan ini, pemain dapat membuat pemukiman yang bisa tumbuh menjadi sebuah kota dengan menjadikannya sebagai daerah komersial, industri, pengembang perumahan (membuat fasilitas pelayanan masyarakat dan merawatnya), transportasi, dan keperluan lainnya.

SimCity5 mengendalikan wilayah yang memberikan skala lebih dari satu kota dan dapat bermain untuk satu kota atau sampai dengan enam belas kota sekaligus. Kota-kota tersebut masing-masing dengan spesialisasi yang berbeda-beda. *Multiplayer* menambahkan aspek baru dalam permainan ini sebagai keputusan akan memiliki efek untuk kota atau wilayah dan menciptakan cara-cara baru untuk bermain, entah dengan kolaborasi atau pun bersaing untuk mendapatkan medali.



Gambar 1 Cover depan SimCity

Sumber : http://2.bp.blogspot.com/-n1k4mifqWrs/UVA5kqWjL1U/AAAAAAAAAADU/9vuJTGT72I/s1600/250px-SimCity_2013_Limited_Edition_cover.png
(diakses pada 15 Desember 2015)

SimCity 5 menggunakan teknologi *engine GlassBox*,

sehingga memungkinkannya untuk memberikan simulasi yang lebih detail daripada permainan sebelumnya. Saat pengembangannya, SimCity mendapat kritik untuk *engine* dan *gameplay* barunya. Permainan ini memerlukan koneksi internet untuk memainkannya. Koneksi tersebut digunakan untuk menyimpan data permainan ke server SimCity dan memungkinkan pemain untuk saling berbagi sumber daya yang dimiliki.

Pada awal permainan, pemain dapat memilih "Create Game" untuk memulai permainan dari awal atau "Resume Game" untuk melanjutkan permainan yang sudah ada sebelumnya. Pada permainan ini, pemain bisa menyimpan lebih dari satu permainan. Saat melihat penampilan awalnya, dapat dilihat bahwa SimCity 5 memiliki grafis yang sangat detail mulai dari bentuk bangunan, penduduk, lalu lintas kendaraan hingga bayangan bangunan.

Langkah selanjutnya adalah memilih "Region" sebagai lahan untuk dijadikan kota. Pemain harus memberikan nama kota yang ingin dibuat. Setiap lahan memiliki kelebihan dan tantangan tersendiri, jadi pemain harus menentukan lokasi kota sesuai dengan rencana pembangunannya.

Pembangunan kota diawali dengan membuat jalan raya, sebagai akses bagi para penduduk menuju lokasi pemukiman. Selanjutnya pemain dapat membangun perumahan dan akan melihat para penduduk yang berdatangan dengan mobilnya setelah lokasi perumahan ditentukan. Pekerjaan selanjutnya adalah membangun fasilitas penunjang kehidupan penduduk seperti pengairan, listrik, dll. Pekerjaan membangun jalan, pemukiman, dan fasilitas merupakan pekerjaan yang dilakukan oleh pemain secara terus menerus.

Jumlah penduduk yang kian ramai membuat kebutuhan dan masalah yang muncul jauh semakin kompleks. Mereka menuntut sanitasi, perlindungan keamanan terhadap berbagai masalah sosial seperti kejahatan, api, dan kesehatan. Semakin luas wilayah, semakin banyak pula kantor polisi, pemadam kebakaran, dan rumah sakit yang harus disediakan. Berbagai peristiwa pun akan bermunculan seiring berkembangnya kota yang dibangun oleh pemain. Oleh karena itu, sering kali permainan ini disebut sebagai permainan yang tiada akhirnya. Walaupun sebenarnya akhir dari permainan ini adalah ketika pengguna tidak dapat memainkannya dengan baik dan menjadikannya kota mati.

II. DASAR TEORI

Quaternion adalah bagian dari aljabar *noncommutative division* pertama kali ditemukan oleh William Rowan Hamilton pada hari Senin, 16 Oktober 1843 di Dublin, Ireland. Ide untuk quaternion muncul ketika dia sedang berjalan di sepanjang Canal Kerajaan dalam perjalanan ke pertemuan Akademi Irlandia, dan Hamilton sangat senang dengan penemuan bahwa ia tergores rumus dasar aljabar quaternion,

$$i^2 = j^2 = k^2 = ijk = -1 \quad (1)$$

ke dalam batu jembatan Broughman (Mishchenko dan Solovoyov 2000). Himpunan quaternion dinotasikan sebagai \mathbb{H} , H , atau Q_8 , dan quaternion adalah salah satu contoh dari kelas yang lebih umum dari nomor *hypercomplex* ditemukan oleh Hamilton. Sementara quaternion tidak komutatif, mereka asosiatif, dan mereka membentuk kelompok yang dikenal sebagai kelompok quaternion.

Analogi dengan bilangan kompleks yang merepresentasikan sebagai jumlah dari bagian real dan imajiner, $a + bi$, sebuah quaternion dapat ditulis juga sebagai kombinasi lanjar,

$$H = a \cdot 1 + bi + cj + dk. \quad (2)$$

Quaternion dapat direpresentasikan menggunakan matriks kompleks 2×2 .

$$H = \begin{bmatrix} z & w \\ -\bar{w} & z \end{bmatrix} = \begin{bmatrix} a+ib & c+id \\ -c+id & a-ib \end{bmatrix}, \quad (3)$$

dengan z dan w adalah angka kompleks, a, b, c , dan d adalah riil dan \bar{z} adalah konjugasi yang kompleks dari z .

Quaternion dapat dituliskan dengan bentuk

$$a = a_1 + a_2 i + a_3 j + a_4 k. \quad (4)$$

Konjugasi quaternion dapat dituliskan sebagai

$$\bar{a} = a_1 - a_2 i - a_3 j - a_4 k. \quad (5)$$

Penjumlahan dua buah quaternion dapat dituliskan sebagai

$$a + b = (a_1 + b_1) + (a_2 + b_2)i + (a_3 + b_3)j + (a_4 + b_4)k, \quad (6)$$

Hasil kali dari dua buah quaternion adalah

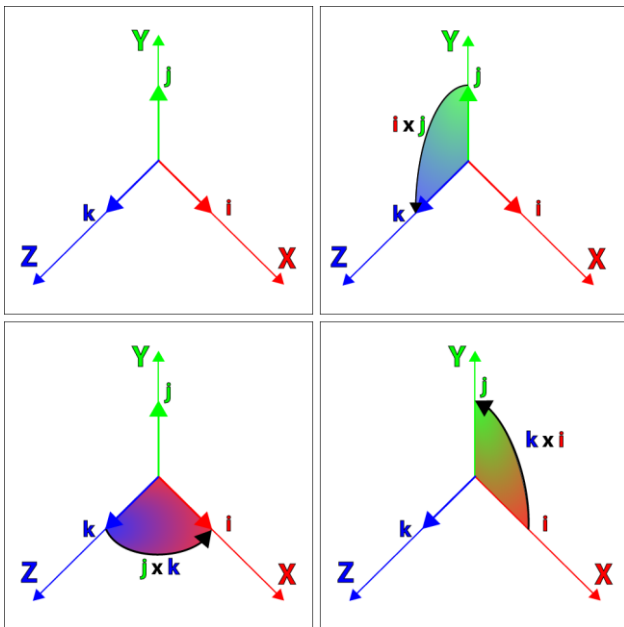
$$ab = (a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4) + (a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3)i + (a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2)j + (a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1)k \quad (7)$$

Norma dari quaternion didefinisikan sebagai

$$n(a) = \sqrt{\bar{a}a} = \sqrt{a_1^2 + a_2^2 + a_3^2 + a_4^2}. \quad (8)$$

Rumus yang terkenal yang muncul dari ide Hamilton (rumus (1)) dapat diperjelas menjadi

$$\begin{aligned} ij &= k & jk &= i & ki &= j \\ ji &= -k & kj &= -i & ik &= -j \end{aligned} \quad (9)$$



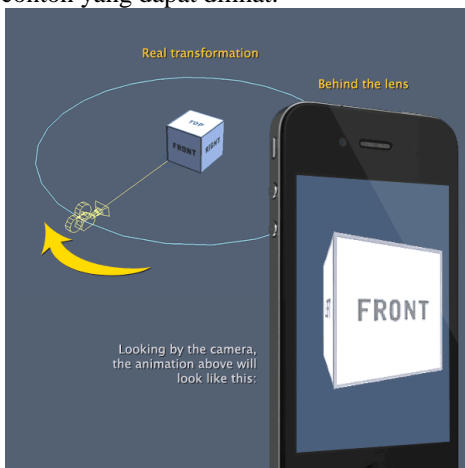
Gambar 2 Visualisasi dari ij, jk, dan ki

Sumber: <http://3dgep.com/wp-content/uploads/2012/06/Visualizing-the-properties-of-ij-jk-ki.png> diakses pada 15 Desember 2015

III. KAMERA PADA DUNIA 3D

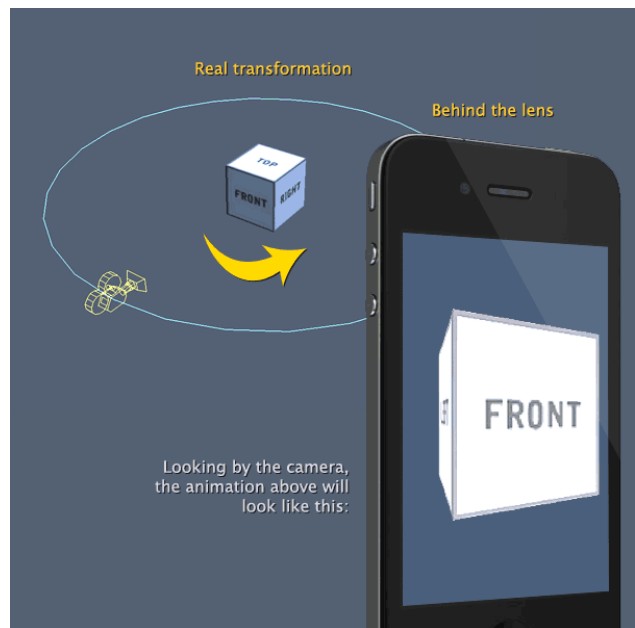
Hal pertama yang perlu dipahami adalah tentang bagaimana transformasi terjadi di dunia 3D. Setelah mendefinisikan struktur objek 3D, strukturnya akan tetap utuh (struktur adalah simpul, koordinat tekstur, dan normal). Hal yang akan berubah di *frame* demi *frame* hanyalah beberapa matriks (sering kali hanya satu matriks saja). Matriks tersebut akan menghasilkan perubahan sementara berdasarkan struktur asli. Jadi struktur asli dari objek tidak akan pernah berubah.

Dengan demikian, ketika memutar objek pada layar, apa yang pengguna lakukan adalah menciptakan matriks yang berisi informasi untuk membuat faktor rotasi tersebut terjadi. Ketika kita mengalikan matriks dengan simpul dari objek pada *shaders*, objek tampaknya berputar. Hal yang sama berlaku untuk setiap elemen 3D lainnya, seperti lampu atau kamera. Tetapi objek kamera memiliki perilaku khusus, yaitu semua transformasi harus dibalik. Berikut adalah contoh yang dapat dilihat.



Gambar 3 Contoh 1 perilaku kamera

Sumber: http://db-in.com/images/camera_behavior_1_example.gif diakses pada 15 Desember 2015.



Gambar 4 Contoh 2 perilaku kamera

Sumber: http://db-in.com/images/camera_behavior_2_example.gif diakses pada 15 Desember 2015.

Perhatikan gambar di atas bahwa gambar yang dihasilkan pada layar perangkat sama dalam kedua kasus. Perilaku ini memberikan ide bahwa setiap gerakan kamera terbalik dibandingkan dengan ruang objek. Sebagai contoh, jika kamera masuk ke sumbu Z positif akan menghasilkan efek yang sama seperti mengirim objek 3D dengan sumbu Z negatif. Memutar kamera di sumbu Y positif akan memiliki efek yang sama seperti memutar objek 3D di sumbu Y negatif. Dengan ide ini, setiap transformasi pada kamera akan terbalik dan terus digunakan.

Konsep berikutnya tentang kamera adalah bagaimana cara ruang lokal berinteraksi dengan ruang dunia. Pada contoh di atas, jika kita berpikir dalam memutar objek ke sumbu Y negatif pada lokal akan menghasilkan hasil yang sama seperti memutar kamera di sumbu Y positif dan bergerak pada sumbu X dan Y di sekitar objek, dengan asumsi kamera sebagai poros. Operasi dengan matriks adalah salah satu cara yang ampuh untuk menghadapi urusan tersebut. Untuk mengubah rotasi ruang lokal menjadi rotasi ruang global, kita butuh mengubah urutan matriks yang dapat diselesaikan dengan operasi perkalian ($A \times B = \text{lokal}$, $B \times A = \text{global}$, mengingat perkalian matriks tidak bersifat komutatif). Jadi kita harus kalikan matriks kamera dengan matriks objek dalam urutan matriks.

IV. IMPLEMENTASI QUATERNION PADA SIMCITY 5

Seperti konsep yang telah dijelaskan mengenai kamera pada dunia 3D, dapat disimpulkan bahwa simcity 5 membutuhkan minimal 2 kelas untuk implementasi quaternionsnya, yaitu kelas building yang merepresentasikan bangunan dan kelas camera untuk merepresentasikan kamera tiga dimensinya. Kelas building maupun kamera ini membutuhkan transformasi vektor berupa rotasi. Hal ini dikarenakan ketika pengguna melakukan rotasi (melalui kelas kamera), maka bangunan tersebut akan terlihat berbeda dari sebelumnya, tergantung perspektifnya di mana. Oleh karena itu, kelas building dapat kita tuliskan kodenya sebagai berikut.

```
public class building
{
    public Model model;
    public Matrix world =
Matrix.Identity;
    public Vector3 position;
    public Vector3 direction;
    public Quaternion
rotation =
Quaternion.Identity;
    public BoundingSphere
collision;
    public float speed = 2f;
    public float
moveForwards;
    public float moveLeft;
    public float forward;
    Matrix[] transform;

    public building(Model m,
Vector3 iPos)
    {
        model = m;
        position = iPos;
        direction = new
Vector3(0, 0, 1);
        transform = new
Matrix[m.Bones.Count];

m.CopyAbsoluteBoneTransformsT
o(transform);

        foreach (ModelMesh
mesh in m.Meshes)
        {
            collision =
BoundingSphere.CreateMerged(c
ollision,
mesh.BoundingSphere);
        }
    }
}
```

```
public building()
{
}

public BoundingSphere
getSphere()
{
    return
collision.Transform(world);
}

public void
sampleInputMovement (GamePadSt
ate building)
{
    moveForwards =
(building.ThumbSticks.Left.Y
/ 5);
    moveLeft =
(building.ThumbSticks.Left.X
/ 32) * -1;
}

public void update()
{
    rotation *=
Quaternion.CreateFromAxisAngl
e(new Vector3(0, 0, -1),
moveLeft) *
Quaternion.CreateFromAxisAngl
e(new Vector3(1, 0, 0),
moveForwards);

    Vector3 motion =
Vector3.Transform(direction,
rotation);
    position += motion *
forward;
    world =
Matrix.CreateFromQuaternion(r
otation) *
Matrix.CreateTranslation(posi
tion);
}

public void Draw(Matrix
projection, Matrix view)
{
    Matrix[] transforms =
new
Matrix[model.Bones.Count];

model.CopyAbsoluteBoneTransfo
rmsTo(transforms);
}
```

```

foreach (ModelMesh mesh in
model.Meshes)
    {
        foreach (BasicEffect
effect in mesh.Effects)
            {
effect.EnableDefaultLighting();
                effect.Projection =
projection;
                effect.View = view;
                effect.World =
transforms[mesh.ParentBone.Index] *
world;
            }
            mesh.Draw();
        }
    }
}

```

Berikut adalah *pseudo code* dari kelas Camera.

```

public class Camera
{
    Vector3 position;
    Vector3 newposition;
    Quaternion cameraRotation;
    Vector3 targetOffset = new
Vector3(0, 1f, -6f);

    public Matrix view =
Matrix.Identity;
    public Matrix projection =
Matrix.Identity;

    public Camera(Game game, Vector3
xPostion)
    {
        position = xPostion;

        projection =
Matrix.CreatePerspectiveFieldOfView(M
athHelper.PiOver4,
game.GraphicsDevice.Viewport.AspectRa
tio, 1.0f, 1000000f);
    }
    public void update(Matrix
targetWorld)
    {
        Quaternion rotation;
        Vector3 target;
        Vector3 scale;
        targetWorld.Decompose(out
scale, out rotation, out target);
        cameraRotation =
Quaternion.Lerp(cameraRotation,
rotation, 0.15f);
        newposition =
Vector3.Transform(targetOffset,
targetWorld);
    }
}

```

```

position =
Vector3.SmoothStep(position,
newposition, 0.8f);

//I want to locate my
camera 25 units behind the
player and 5 units above him
Vector3 cameraPosition =
targetWorld.Translation +
(targetWorld.Backwards * 25) +
(targetWorld.Up * 5);

//I want my cameras to
look at a point 10 units
directly in front of my player
Vector3 cameraTarget =
targetWorld.Translation +
(targetWorld.Forward * 10);

//build a view matrix
that represents this
view =
Matrix.CreateLookAt(cameraPositi
on, cameraTarget, Vector3.Up);
}
}

```

Dibandingkan dengan menggunakan matriks saja, quaternion membuat segala aturan untuk rotasi kamera menjadi semakin lebih mudah. Pada Simcity 5, quaternion tersebut digunakan sebagai kamera yang berperan sebagai *third person camera*. Jika quaternion digunakan sebagai *first person camera* pada simcity, sangatlah aneh karena tidak ada peran manusia, tidak dapat dengan gampang melihat seluruh isi kota, dan akan membutuhkan waktu yang lama untuk mendefinisikan gedung-gedung yang tinggi.

Tidak hanya simcity, hampir semua permainan yang memiliki genre *simulation games* dan telah menerapkan konsep 3D menggunakan konsep quaternion ini karena lebih mudah digunakan dibandingkan mendefinisikan masing-masing matriks satu persatu. Selain itu, quaternion juga dapat dengan mudahnya dikonversi menjadi matriks. Akan tetapi hal tersebut tidak dapat dibalik yang berarti tidak semua matriks bisa dikonversi menjadi quaternion.

V. KESIMPULAN

Penggunaan quaternion sangat berguna dan mampu menyelesaikan masalah yang dihadapi oleh matriks pada Simcity 5. Hal ini dikarenakan mudahnya penggunaan serta memiliki aspek-aspek yang berguna umumnya pada *game* dengan grafis tiga dimensi. Tidak hanya pada Simcity 5, akan tetapi hampir semua *game* dengan grafis tiga dimensi menggunakan quaternion untuk penglihatan kamera, entah itu sebagai *first person* atau sebagai *third person*.

VI. UCAPAN TERIMA KASIH

Penulis bersyukur kepada Allah SWT oleh karena rahmat dan hidayahnya, penulis dapat menyelesaikan makalah berjudul “Penggunaan Quaternion pada Sim City 5” ini berhasil diselesaikan tepat waktu. Penulis juga berterima kasih kepada Dr. Ir. Rinaldi Munir dan Drs. Judhi Santoso, M.Sc. untuk arahan dan ajaran mereka pada mata kuliah Aljabar Geometri sehingga penulis mendapatkan pengetahuan yang sangat berguna dalam menyelesaikan makalah ini dan masih butuh belajar lagi. Selain itu, penulis juga tidak lupa berterima kasih kepada kedua orang tua, Raka Nurul Fikri, Andra Wahyu Purnomo, keluarga, dan seluruh teman yang telah memberikan masukan, dukungan, semangat, dan doa mereka.

REFERENCES

- [1] <http://www.zwsoft.com/CAD-Support/3d-computer-graphics-software.html> diakses pada 15 Desember 2015.
- [2] <http://amitadeshpande.blogspot.co.id/2008/01/simulation-games-1-definition.html> diakses pada 15 Desember 2015
- [3] <https://www.youtube.com/playlist?list=PLYhuQUmCiA-AZbhfZ3aJGB8YLaj2hylaLf> diakses pada 15 Desember 2015.
- [4] <http://www.metacritic.com/game/pc/simcity> diakses pada 15 Desember 2015.
- [5] <http://www.alteredgamer.com/pc-gaming/58403-from-planes-to-pets-to-people-the-growth-and-breadth-of-simulation-games/> diakses pada 15 Desember 2015.
- [6] http://www.kotakgame.com/review/detail_review/3502/477/SimCity-2013 diakses pada 15 Desember 2015.
- [7] <http://jagatplay.com/2013/05/pc-2/review-simcity-kota-yang-tak-lagi-dapat-berdiri-mandiri/> diakses pada 15 Desember 2015.
- [8] <http://mathworld.wolfram.com/Quaternion.html> diakses pada 15 Desember 2015.
- [9] <http://www.3dgep.com/understanding-quaternions/> diakses pada 15 Desember 2015.
- [10] <http://blog.db-in.com/cameras-on-opengl-es-2-x/> diakses pada 15 Desember 2015

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Desember 2015



Fitra Rahmamuliani 13513095