Penerapan Quadtree dalam Penggambaran Data Daratan Raksasa

Rifqi Naufal Abdjul - 13520062[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13520062@itb.ac.id

***Abstract*—Quadtree merupakan jenis struktur data pohon yang biasa digunakan untuk menyimpan data secara efisien. Makalah ini membahas tentang penggunaannya dalam penggambaran data daratan raksasa yang biasa digunakan dalam program atlas online seperti Google Earth.**

***Keywords*— Quadtree, data daratan, piksel, memori**

I. PENDAHULUAN

Bumi yang kita tempati merupakan tempat yang sangat amat luas. Untuk melakukan penggambaran yang realistis terhadap bumi, maka dibutuhkan gambar dengan ukuran yang sangat besar. Oleh karena itu, untuk menyimpan dan mengakses data yang sangat amat besar dibutuhkan memori dan kemampuan komputasi yang sangat amat besar juga. Dengan perbandingan ukuran bumi yang mempunyai luas permukaan sekitar 500 juta kilometer, dengan densitas gambar 96 ppi (piksel per inci), maka gambar yang dihasilkan mempunyai 7 juta tera piksel yang harus dikomputasi secara langsung oleh komputer. Tidak hanya penyimpanan yang bisa menjadi masalah, cara pengguna mengakses atlas pun menjadi hal yang dipermasalahkan dikarenakan besarnya data yang ingin ditampilkan. Pengguna harus leluasa menggunakan atlas online seperti melakukan pergerakan dari satu titik ke titik lainnya, atau melakukan *zoom in* dan *zoom out* sesuka hati dari luar angkasa hingga seukuran denah biasa. Lalu, bagaimana program atlas online dapat melakukannya dengan sangat mudah? Atlas online dapat dijalankan dalam komputer manapun dengan cepat dan hanya berbasis browser.

Gambar 1.1 Terravision Project
Sumber: https://artcom.de/en/?project=terravision

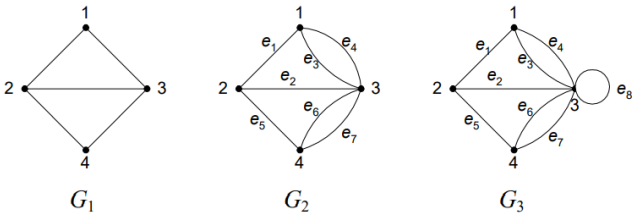
Pada tahun 1994, saat Google Earth belum dibangun atau bahkan terpikirkan dapat di bangun. Sebuah perusahaan bernama ART+COM dengan basis seni dan komunikasi mengembangkan sistem yang membuktikan bahwa untuk melakukan hal seperti menggambarkan bumi dalam komputer bukanlah hal yang tidak mungkin. Bahkan mereka membuat atlas online mereka sendiri untuk membuktikannya yang merupakan hal yang terlihat tidak mungkin pada masanya.

Rahasianya berasal dari teknik penyimpanan data gambar yang sangat besar tersebut dalam suatu data struktur berbasis pohon, yaitu quadtree. Penggunaan quadtree dalam penyimpanan data gambar tidak hanya mendukung efisien dalam sisi ukuran penyimpanan, tetapi juga mendukung dalam pengaksesan berdasarkan arsitektur komputer. Tidak hanya struktur data, tetapi banyak aspek yang mendukung penggunaan yang sangat lancar dengan kemampuan komputasi yang sangat minim. Aspek lainnya adalah, sistem koordinat yang dikembangkan khusus untuk menyimpan dan menggambarkan data daratan, sistem angka yang mendukung presisi tinggi sehingga tidak terjadi ketidakpastian pada posisi kamera atau objek.

II. TEORI DASAR

A. Graf

Graf biasanya digunakan sebagai penggambaran suatu objek diskrit dan hubungan antara objek tersebut dengan objek lainnya. Graf juga dapat didefinisikan sebagai pasangan himpunan (V,E), dengan V merupakan suatu himpunan dari simpul (*vertices*) dan E merupakan himpunan sisi (*edges*) yang menghubungkan antara kedua simpul.



Gambar 2.1. Jenis jenis graf berdasarkan ada tidaknya sisi gelang/ganda
Sumber:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis

Graf biasanya dibagi menjadi beberapa jenis berdasarkan beberapa hal, yaitu:

1. Berdasarkan ada tidaknya sisi gelang/ganda
   a. Graf Sederhana

Graf yang tidak mempunyai gelang ataupun sisi ganda

b. Graf Tidak Sederhana
Graf yang mempunyai sisi gelang atau sisi ganda

i. Graf Ganda
Graf yang mengandung sisi ganda

ii. Graf Semu
Graf yang mengandung sisi gelang

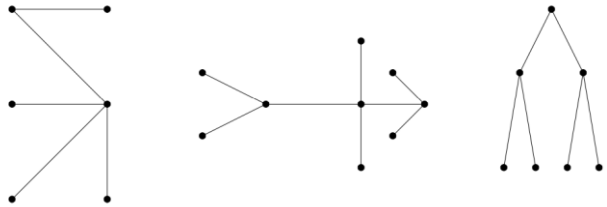
2. Berdasarkan orientasi arah pada sisi graf
   a. Graf tak-berarah
   Setiap sisi pada graf tidak mempunyai orientasi arah
   b. Graf berarah
   Graf yang setiap sisinya mempunyai orientasi arah

B. Pohon

Pohon adalah salah satu representasi dari graf terhubung yang tidak memiliki sirkuit. Pohon dapat dibentuk dari sebuah graf, jika graf tersebut memenuhi syarat berikut.

1. Setiap pasang simpul dalam pohon terhubung dengan lintasan tunggal.
2. Setiap simpul berjumlah n terhubung dan mempunyai jumlah sisi m = n – 1.
3. Pohon tidak memiliki sirkuit dan penambahan satu sisi dalam graf hanya akan membuat satu sirkuit.

Jika seluruh syarat tersebut terpenuhi, maka sebuah graf dapat disebut pohon secara definisi dan struktur.



Gambar 2.2. Hutan yang teridiri dari 3 buah pohon
Sumber:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis

Pohon yang sering digunakan adalah pohon berakar. Pohon berakar merupakan salah satu jenis graf berarah dimana terdapat satu simpul yang ditunjuk sebagai akar dan seluruh sisi pada akar pasti mengarah keluar dari akar tersebut. Berikut adalah beberapa terminologi terkait dengan pohon berakar.

1. *Parent* dari simpul a merupakan seluruh simpul b yang mempunyai sisi yang mengarah ke a.
2. Jika simpul a adalah *parent* dari simpul b, maka simpul b merupakan *children* dari simpul a.
3. Seluruh simpul yang mempunyai *parent* yang sama merupakan *siblings*.
4. *Ancestors* merupakan hubungan antar simpul yang berarti seluruh simpul yang mempunyai lintasan ke suatu simpul merupakan *ancestors* dari simpul tersebut.
5. Jika simpul a adalah *Ancestors* dari simpul b, maka simpul b merupakan *Descendants* dari simpul a.
6. *Leaves* merupakan simpul yang tidak mempunyai anak dan biasanya terletak paling bawah dalam suatu pohon berakar.
7. Simpul yang mempunyai anak disebut sebagai *internal node* atau simpul internal.
8. *Subtrees* atau upapohon merupakan pohon bagian dari suatu pohon yang memiliki *descendants* dan sisi yang sama dengan lintasan dibawah pohon awal.

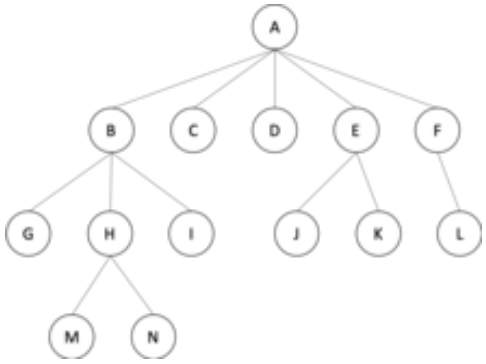
C. Pohon n-ary

Dalam teori graf, pohon n-ary merupakan pohon berakar yang setiap simpulnya mempunyai anak berjumlah tidak lebih daripada n. Terdapat beberapa jenis pohon n-ary yang spesial seperti binary tree (pohon n-ary dengan n = 2), ternary tree (pohon n-ary dengan n = 3), dan quartenary tree (pohon n-ary dengan n = 4). Berikut adalah tipe pohon n-ary berdasarkan kondisi daunnya.

a. Pohon n-ary penuh
Pohon n-ary dengan setiap simpul yang berada dalam pohon tersebut mempunyai 0 atau n anak.

b. Pohon n-ary lengkap
Pohon n-ary yang hemat ruang secara maksimal, yaitu dengan maksud setiap kedalaman terisi secara penuh selain pada kedalaman terakhir. Dan jika kedalaman terakhir tidak terisi penuh, maka seluruh daun harus diletakkan di posisi paling kiri.

c. Pohon n-ary sempurna
Pohon n-ary sempurna merupakan pohon n-ary penuh yang mempunyai kedalaman daun yang sama di seluruh cabang.



Gambar 2.3. Contoh pohon n-ary dengan n = 5
Sumber: https://en.wikipedia.org/wiki/M-ary_tree

Dikarenakan spesialnya, pohon n-ary mempunyai beberapa properti yang unik terhadap pohon n-ary sendiri, seperti

1. Untuk setiap pohon n-ary yang mempunyai ketinggian h, maka jumlah maksimal dari total daun adalah n^h.
2. Ketinggian dari pohon n-ary tidak menghitung simpul akar. Jadi, pohon n-ary yang hanya mempunyai akar mempunyai ketinggian 0.
3. Ketinggain dari pohon n-ary merupakan kedalaman maksimal dari simpul manapun di dalam pohon tersebut.

4. Jumlah simpul dalam pohon n-ary sempurna mengikuti persamaan berikut.

$$N = \Sigma_{i=0}^{h} m^i = \frac{m^{h+1} - 1}{m - 1}$$

5. Ketinggian dari pohon n-ary sempurna memenuhi persamaan berikut.

$$h \geq \lceil \log_m((m-1).N + 1) - 1 \rceil$$

6. Ketinggian dari pohon n-ary lengkap dengan jumlah simpul n memenuhi persamaan berikut.

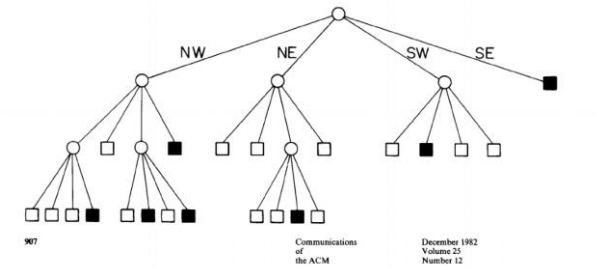
$$h = \lfloor \log_m((m-1).n) \rfloor$$

D. Quadtree

Quadtree merupakan data struktur berdasar pohon n-ary yang setiap simpul internal dari quadtree mempunyai tepat 4 anak. Quadtree biasanya digunakan untuk membagi ruang 2 dimensi secara rekursif menjadi 4 daerah atau kuadran. Ketinggian dari quadtree akan menyesuaikan dengan kedetailan informasi yang disimpan dalam simpul tersebut. Jika suatu informasi dalam suatu simpul masih dapat didetailkan, maka akan membuat turunan 4 anak dari simpul itu menyimpan detail yang lebih jelas daripada simpul awal.

Quadtree biasa diimplementasi dalam melakukan beberapa proses seperti berikut:

1. Penggambaran gambar
2. Pemrosesan gambar
3. Pengelompokan koneksi
4. Pengindeksan daerah 2 dimensi
5. Pemodelan 3 dimensi dari data daratan
6. Penyimpanan *sparse data*
7. Analisis fraktal



Gambar 2.4. Contoh penggambaran quadtree
Sumber: An effective way to represent quadtrees. Gargantini, Irene. (1982).

Quadtree merupakan struktur data yang sangat berguna dikarenakan komputer selalu menggunakan basis biner dalam melakukan segala hal. Dan quadtree merupakan bentuk paling sederhana dari penggambaran 2 dimensi oleh komputer dikarenakan jumlah anak dalam setiap simpul berjumlah 4 dapat menggambarkan 2 nilai ke dalam 2 arah/dimensi.

E. Aturan "Power of Two"

Dikarenakan angka 2 merupakan basis dari sistem numeral biner, angka yang berpangkat 2 merupakan hal yang sering muncul di dalam topik ilmu komputer. Oleh karena itu, muncullah aturan *power of two* dalam penyimpanan tekstur atau gambar dalam memori.

Aturan *power of two* secara singkat mengatakan bahwa resolusi gambar akan lebih teroptimisasi dan efisien dalam penyimpanan dibandingkan dengan resolusi *non-power of two*. Hal ini dikarenakan dari sistem penyimpanan gambar dalam VRAM (Video Random Access Memory) yang menyimpan gambar dalam bentuk bit.

Pada komputer lama, VRAM hanya bisa menerima gambar berukuran dalam bentuk *power of two*. Berikut adalah tabel berisikan beberapa contoh resolusi yang memenuhi aturan resolusi *power of two*.

Lebar (px)	Tinggi (px)	Jumlah piksel (px)	Estimasi Ukuran Gambar (KB)
128	128	16.384	48
128	256	32.768	96
256	256	65.536	192
512	512	262.144	768
512	1024	524.288	1.536
1024	1024	1.048.576	3.072
2048	2048	4.194.304	12.288

Tabel 2.1. Nilai resolusi gambar yang memenuhi aturan power of two
(Sumber : Dok penulis)

Jika suatu gambar mempunyai resolusi *non-power of two*, maka komputer biasanya akan melakukan *padding* menggunakan nilai asal terhadap gambar tersebut sehingga membentuk nilai *power of two* selanjutnya. Proses tersebut merupakan proses yang sangat lamban dikarenakan harus melakukan banyak proses yaitu, melakukan pengecekan apakah ukuran *power of two*, menambahkan *padding* pada gambar dengan resolusi *non-power of two*, menyimpan informasi data gambar asli, dan memproses kembali gambar menjadi bentuk *non-power of two*.

F. Floating Point

Dalam ilmu komputer, *floating point* merupakan cara menggambarkan suatu angka desimal dengan menukarkan kemampuan range nilai dengan presisi. *Floating point* biasa digunakan dalam komputasi dengan angka yang sangat kecil atau sangat besar dikarenakan kemampuan presisi yang dapat diatur.

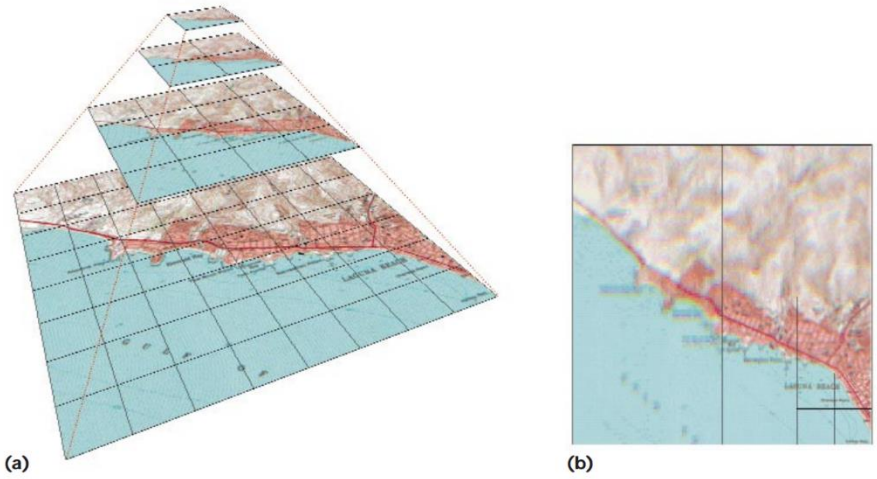
Secara umum, *floating point* direpresentasikan dengan 2 bagian yaitu, *significand* yang merupakan angka bulat yang merupakan nilai *floating point* dan bagian *exponent* yang merupakan pemangkatan suatu basis agar dapat meraih nilai yang sangat besar dan yang sangat kecil. Sebagai contoh, berikut adalah contoh representasi dari *floating point*

.

$$1.2345 = 12345 * 10^{-4}$$

Pada *floating point* diatas, 12345 merupakan bagian *significand*, 10 merupakan basisnya, dan -4 merupakan *exponent*-nya.

Dikarenakan struktur *floating point*, nilai desimal akan mempunyai presisi tertinggi saat mendekati angka 0, tetapi akan mempunyai presisi yang sangat rendah saat jauh dari angka 0.



Gambar 3.1. (a) 4 tingkatan resolusi gambar yang dipotong menjadi ukuran 128x128 (b) Penggambaran gambar dengan resolusi yang bervariasi di daerah yang berbeda
Sumber: TerraVision II: Visualizing Massive Terrain Database in VRML

III. IMPLEMENTASI

A. Struktur data quadtree

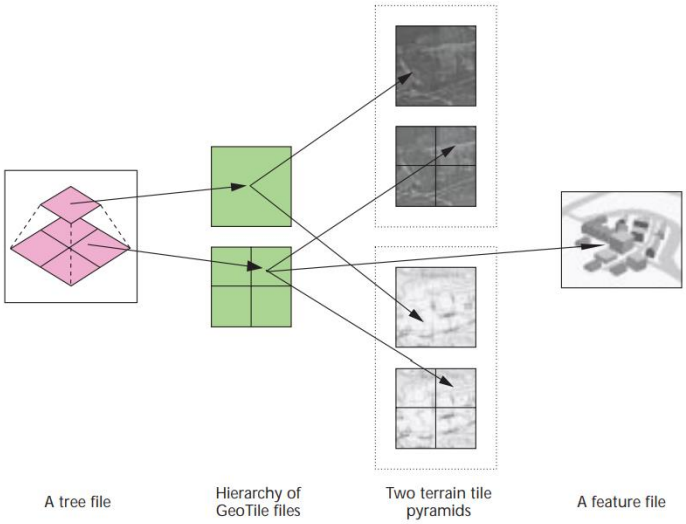
Dalam penyimpanan data, data akan diletakkan pada tiap simpul di dalam quadtree secara bertingkat. Misalnya untuk menyimpan gambar asli berukuran 1024x1024 piksel, maka quadtree akan memuat gambar asli, dan beberapa versi gambar yang telah diturunkan kualitasnya menjadi 512x512 piksel, 256x256 piksel, 128x128 piksel, dan seterusnya. Pada gambar 3.1.a. divisualisasikan cara penempatan gambar tersebut pada struktur piramida yang menggambarkan quadtree dimana setiap segmen akan dibagi menjadi 4 bagian pada kedalaman selanjutnya, sehingga setiap segmen akan berukuran tepat seperempat dari kedalaman sebelumnya. Menggunakan penggambaran ini, kita dapat memvisualisasikan kedalaman selanjutnya secara rekursif. Untuk ukuran gambar yang dimuat sebagai contoh, pada gambar 3.1.b. pada bagian kanan bawah gambar, terdapat segmen dengan resolusi tinggi dan pada bagian atas kiri merupakan segmen dengan resolusi rendah dengan kedalaman yang lebih rendah daripada segmen dengan resolusi tinggi. Sehingga ketika memuat gambar tersebut, dengan anggapan resolusi setiap segmen adalah 128x128 hanya membutuhkan memori 4.8 KB dibandingkan dengan gambar aslinya yang berukuran 3.1 MB.

Dikarenakan struktur data quadtree yang bersifat rekursif terhadap setiap segmennya, maka struktur ini sangat bisa diimplemen untuk data yang sangat besar. Menggunakan struktur data ini, program dapat mengoptimisasi jumlah data yang dikirm melewati internet, dan memori yang digunakan untuk memuat gambar tersebut.

B. Tipe file untuk menggambarkan data daratan

Data daratan biasanya tidak berbentuk rata seperti peta, tetapi mempunyai daratan yang lebih tinggi dan lebih rendah. Sebagai contoh, data yang dikumpulkan oleh survei geologi US pada daerah yang mencakup 1 derajat bumi mempunyai 1,201 x 1,201 nilai elevasi. Daripada kita menghilangkan nilai elevasi dan menganggap seluruh daratan sebagai sesuatu yang datar, kita dapat memanfaatkan nilai elevasi untuk lebih mengoptimasi dan meningkatkan pengalaman pengguna.

Untuk menghasilkan fungsionalitas maksimum, dalam penggambaran data daratan, harus dibutuhkan beberapa jenis tipe file yang digabungkan menjadi suatu gambar yang biasa dilihat pengguna. Sebagai contoh, disini akan dijelaskan suatu sistem yang menggunakan 4 tipe file untuk mendukung penggambaran data daratan. Berikut adalah gambar tentang hubungan antar tipe filenya.



Gambar 3.2. Hubungan antar tipe file yang digunakan dalam penggambaran data Daratan
Sumber:
TerraVision II: Visualizing Massive Terrain Database in VRML

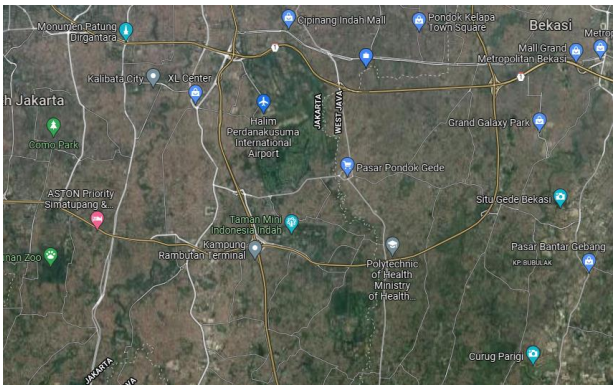
Tipe file pohon, merupakan file pohon yang menyusun tingkatan tingkatan resolusi dari resolusi paling rendah hingga resolusi tertinggi yang merupakan daun dari pohon. Dengan pohon ini kita bisa mengatur *level of detail* dari sebuah

penggambaran data daratan. Kita bisa mengatur *level of detail* sesuai dengan data yang ada atau bahkan memori yang digunakan. Dengan ini struktur data ini mempunyai sifat *scalable* (dapat diperbesar dengan mudah) dan *customizable* (dapat diatur sesuai kebutuhan). File pohon ini tidak langsung menyimpan data file lainnya, tetapi hanya menyimpan *pointer* (sebuah tipe data yang menyimpan alamat data) yang menunjuk pada tipe file selanjutnya, yaitu *geotile*.

File *geotile* menyimpan hubungan seluruh data yang penting terhadap daerah itu. File *geotile* dapat berisikan banyak alternatif data daerah tersebut. Misal, jika terdapat potret satelit, potret udara, dan penggambaran secara ilustratif. Dengan menggunakan tipe file ini, kita dapat memperbaharui data gambar yang lebih baru atau lebih baik tanpa mengganti gambar lama dengan cara menambahkan ke dalam *geotile* yang bersangkutan. Biasanya *geotile* mengandung pointer ke beberapa jenis file lainnya seperti file data daratan dan file fitur daratan.

File data daratan berisi data daratan untuk setiap daerah pada level detail tertentu. Biasanya file data daratan ini berisikan data elevasi setiap titik pada daerah tersebut dan tekstur daerah tersebut. Bentuk tekstur juga harus disesuaikan dengan bentuk daerah tersebut, karena jika terdapat perbedaan elevasi yang sangat besar, biasanya akan terjadi stretching jika hanya diberi tekstur 2 dimensi biasa.

File fitur daratan berisi objek yang terkait pada daerah daratan tersebut. Seperti contoh, dalam suatu daerah biasanya mempunyai jalan dan gedung yang berada di sekitar daerah tersebut. Selain itu juga data fitur tersebut bisa juga berupa arah angin, cuaca, dan data fisik lainnya.



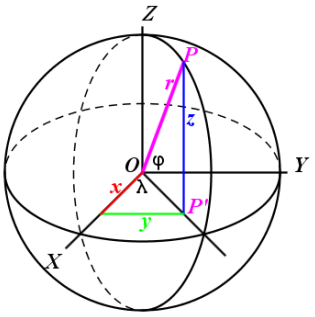
Gambar 3.3. Jenis fitur data daratan pada google maps
(Sumber: Google Maps (2021))

Menambahkan fitur ini merupakan hal yang sedikit membingungkan karena setiap fitur bisa saja memakan tempat lebih dari satu tile pada peta. Salah satu cara menyelesaikan permasalahan tersebut adalah menghubungkan seluruh geotile yang terkait pada satu file fitur daratan yang sama. Dengan ini, jika ingin memuat geotile manapun yang menunjuk file fitur daratan, maka objek tersebut pun dapat termuat.

C. Sistem Koordinat

Sistem koordinat yang digunakan untuk melakukan perjalanan dari titik 1 ke titik lainnya adalah sistem koordinat kartesian dimana suatu koordinat (x,y,z) merupakan *offset distance* (selisih jarak) dari pusat bumi. Tetapi, untuk melakukan peletakan objek di model sistem koordinat kartesian dibutuhkan sebuah pengubahan sistem koordinat projektif yang biasa digunakan objek. Hal ini harus dilakukan dikarenakan sistem koordinat projektif tersebut mempunyai banyak asumsi tentang bentuk bumi entah *ellipsoid*, bulat, dan lain-lain yang dapat mengubah lokasi absolut dari suatu objek.

Permasalahan yang dihadapi tidak hanya itu, tetapi juga dari angka yang digunakan dalam merepresentasikan koordinat tersebut juga dapat bermasalah dikarenakan nilai presisi yang dipunyai oleh suatu *floating point*. Jika hanya dilakukan perlakuan yang disebutkan diatas, maka akan terjadi *jittering* atau getaran yang tidak diinginkan karena posisi kamera akan berada pada titik ambigu yaitu diantara nilai terdekat yang dapat digambarkan *floating point* tersebut. Semakin jauh dari titik asal, maka *jittering* tersebut akan terjadi semakin parah.



Gambar 3.3. Sistem Koordinat sebuah globe
(Sumber: https://www.nosco.ch/mathematics/en/earth-coordinates.php)

Floating point standar sesuai format IEEE hanya mempunyai 23 buah bit untuk menggambarkan nilai eksponennya. Dengan perbandingan dengan ukuran diameter bumi yaitu sekitar 12,700,000m, angka berdigit 6 ($2^{23} \approx 8.36 \times 10^6$) merupakan angka yang sangat kecil. Dikarenakan itu kita hanya bisa menggambarkan secara jelas lokasi yang berjarak sekitar 100,000m dari titik asal.

Solusi dari permasalahan ini adalah melakukan *reverse imaging* yaitu memindahkan titik origin dari model tersebut dari pusat bumi menjadi posisi kamera. Dengan ini posisi kamera dan objek yang dilihatnya pada posisi dengan presisi tertinggi yang dapat digambarkan *floating point*.

D. Tech stack yang dapat digunakan

Pada masa ini, terdapat banyak sekali platform untuk membangun ini semua. Tidak perlu membangun sesuatu dari dasar dikarenakan sudah disediakan oleh platform yang dapat digunakan seperti contohnya, Unity, SketchUp, OpenGL, dan lain lain. Dan bahasa programming yang dapat digunakan untuk membangun atlas online ini bisa teridiri atas banyak bahasa seperti, python, C++, java, dan javascript untuk pembangunan websitenya.

IV. KESIMPULAN

Sesuai dengan yang telah dijelaskan pada makalah ini, melakukan penggambaran data daratan yang sangat besar dapat dilaksanakan dengan struktur yang telah dijelaskan. Dengan

memanfaatkan seluruh pengetahuan tentang arsitektur komputer, kita dapat mengoptimisasi banyak hal dengan mengganti cara pendekatan yang digunakan untuk memrogram sistem.

VII. Ucapan Terima Kasih

Penulis ingin mengucap syukur kepada Allah Swt. karena atas berkat yang diberikan-Nya sehingga penulis dapat menyelesaikan makalah ini dengan baik dan tepat waktu. Penulis juga ingin berterima kasih kepada bu Dra. Harlili, M.Sc. selaku dosen K2 yang telah membimbing penulis selama 1 semester dalam mata kuliah Matematika Diskrit IF2120. Tidak lupa juga teman yang memberikan dukungan selama proses penulisan makalah ini.

Referensi

[1] https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2021-2022/matdis21-22.htm . Diakses pada 13 Desember 2021 pukul 22.00

[2] Reddy, Martin & Leclerc, Yvan & Iverson, Lee & Bletter, Nat. (1999). TerraVision II: Visualizing massive terrain databases in VRML. Computer Graphics and Applications, IEEE. 19. 30-38. 10.1109/38.749120.

[3] Savard, John J. G. (2018) [2007], "The Decimal Floating-Point Standard", quadibloc, archived from the original on 2018-07-03, retrieved 2018-07-16

[4] Luo, F.-X & Zhong, E.-S & Liu, H.-Z & Feng, Z.-H & Cheng, J.-L. (2011). Method of solving jitter problem in global terrain rendering. 23. 506-510+515.

[5] Chris Thorne , Using a Floating Origin to Improve Fidelity and Performance of Large, Distributed Virtual Worlds. Diakses melalui https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.471.7201&rep=rep1&type=pdf. pada 12 Desember 2021, Pukul 19.00

[6] Savard, John J. G. (2018) [2007], "The Decimal Floating-Point Standard", Diakses melalui https://en.wikipedia.org/wiki/Floating-point_arithmetic pada 9 Desember 2021 pukul 13.00

[7] Gargantini, Irene. (1982). An effective way to represent quadtrees. Commun. ACM. 25. 905-910. 10.1145/358728.358741.

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, 3 Desember 2020

Rifqi Naufal Abdjul
13520062