

**PENERAPAN METODE PROCEDURAL MODELING DAN
TRANSFORMASI GEOMETRI DALAM
PEMODELAN 3D BANGUNAN**

SKRIPSI

Oleh :
ULFATUL MUFIDA
11650088



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015**

**PENERAPAN METODE PROCEDURAL MODELING DAN
TRANSFORMASI GEOMETRI DALAM
PEMODELAN 3D BANGUNAN**

SKRIPSI

**Diajukan Kepada:
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim (UIN Maliki) Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :
ULFATUL MUFIDA
11650088**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015**

LEMBAR PENGESAHAN**PENERAPAN METODE PROCEDURAL MODELING DAN
TRANSFORMASI GEOMETRI DALAM
PEMODELAN 3D BANGUNAN****SKRIPSI****Oleh:****ULFATUL MUFIDA****NIM. 11650088**

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal, 28 Juni 2015

| Susunan Dewan Penguji | | Tanda Tangan |
|------------------------------|--|---------------------|
| Penguji Utama | : <u>Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004 | (.....) |
| Ketua Penguji | : <u>A'la Syauqi, M.Kom</u> NIP. 19771201 200801 1 007 | (.....) |
| Sekretaris Penguji | : <u>Dr. Cahyo Crysdiان</u> NIP. 19740424 200901 1 008 | (.....) |
| Anggota Penguji | : <u>Dr. M. Amin Hariyadi, M.T</u> NIP. 19670118 200501 1 001 | (.....) |

Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

MOTTO

~ KEAJAIBAN adalah...

NAMA LAIN DARI KERJA KERAS

KERJA KERAS MELAWAN PENATMU, LELAHMU DAN
MALASMU ~



HALAMAN PERSEMBAHAN

*Segala Pujibagi Allah SWT Tuhan Semesta Alam, ku persembahkan
sebuah karya untuk orang – orang yang paling kusayangi*

dan akubanggakan

kedua orang tuaku tercinta, Mama dan Bapak,

Hj. NurKhalimah dan H. Ikbal, S.Pd

Atas keikhlasannya dalam mendoakan ku

Atas dukungannya yang terusterkirim untuk menyemangati ku

Atas kasih sayangnya sejak lahir sampai sekarang

Dan atas segalanya yang tak mungkin cukup untuk ku ungkapkan

Adikku

Hidayah AinulFitroh

Yang selalumembuat tawadans semangat saat lelah

*Serta seluruh pihak yang turut andil menorehkan tinta kenangan di kisah perjuangan
Strata-1 ku.*

(^_^)

LEMBAR PERSETUJUAN

**PENERAPAN METODE PROCEDURAL MODELING DAN
TRANSFORMASI GEOMETRI DALAM
PEMODELAN 3D BANGUNAN**

SKRIPSI

Oleh:

ULFATUL MUFIDA
NIM. 11650088

Telah disetujui 16 Juni 2015

Pembimbing I

Pembimbing II

Dr. Cahyo Crysdiان
NIP. 197404242009011008

Dr. M. Amin Hariyadi, M.T
NIP. 19670118 200501 1 001

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان
NIP. 197404242009011008

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji syukur penulis panjatkan kehadirat Allah SWT, karena atas limpahan rahmat dan karunia-Nya, skripsi yang berjudul “Penerapan Metode Procedural Modeling dan Transformasi Geometri dalam Pemodelan 3D Bangunan” ini dapat penulis selesaikan dengan baik.

Dalam proses penyelesaian skripsi ini, tidak sedikit pihak yang telah memberikan bantuan baik moril maupun materil. Atas segala bantuan yang telah diberikan, penulis ingin menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Dr. Cahyo Crys dian selaku ketua jurusan Teknik Informatika dan Dosen Pembimbing I yang telah memberikan banyak waktunya untuk membimbing serta memberikan banyak ilmu, motivasi dan segala bentuk bantuan yang tidak ternilai dalam pengerjaan skripsi ini.
2. Dr. M. Amin Hariyadi, M.T selaku Dosen Pembimbing II yang telah memberikan bimbingan, ilmu, saran dan kemudahan dalam proses pengerjaan skripsi ini.
3. Dr. M. Faisal, M.T selaku Dosen Wali yang telah membimbing selama masa kuliah dan memberikan nasehat.
4. Segenap Dosen Teknik Informatika dan Staff Administrasi yang telah memberikan bimbingan keilmuan dan bantuan kepada penulis selama masa studi.

5. Teman-teman Teknik Informatika 2011 (INTEGER '11) yang terus memberikan semangat sampai akhir pengerjaan skripsi terkhusus teman sejuangan Umay, Lutfi, Bagas, Hudan. Teman sekaligus rival terbaikku Fauzan, teman miniaturku Kapid, teman kembar namaku Ulfa R. dan sahabat satu atapku Anis yang selama 3 tahun tetap betah berbagi keluh kesah.
6. Mbakkostku, mbak Rima yang waktunya selalu available untuk berbagi kisah hidup. Mas mbak ANTI 2009 - 2010 yang turut memberi banyak bantuan, motivasi dan saran. Rekan kerja otak dan fisik di pasukan orange Panitia REOC 2014 beserta bocila.k.a arek cilik O'om, Ayom, Ayat, Ilma, Irvan yang telah memberikan banyak tawar, kenangan dan "ajakan" penyemangat. Dan semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dan memberi inspirasi dalam proses pengerjaan skripsi ini. Terima kasih.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan makalah ini, oleh karena itu kritik dan saran yang sifatnya membangun sangat diterima oleh penulis. Semoga penulisan skripsi ini bermanfaat bagi pembaca sekalian.

Wassalamualaikum Wr. Wb.

Malang, 17 Juni 2014

Penulis

DAFTAR ISI

| | |
|--|----------|
| HALAMAN JUDUL | ii |
| HALAMAN PERSETUJUAN | iii |
| HALAMAN PENGESAHAN | iv |
| HALAMAN PERNYATAAN | v |
| MOTTO..... | vi |
| HALAMAN PERSEMBAHAN..... | vii |
| KATA PENGANTAR | viii |
| DAFTAR ISI | x |
| DAFTAR GAMBAR | xii |
| DAFTAR TABEL | xv |
| ABSTRAK (Bahasa Indonesia) | xvi |
| | |
| BAB I PENDAHULUAN | 1 |
| 1.1 LatarBelakang | 1 |
| 1.2 IdentifikasiMasalah | 4 |
| 1.3 TujuanPenelitian | 4 |
| 1.4 BatasanMasalah | 4 |
| 1.5 ManfaatPenelitian | 5 |
| 1.6 SistematikaPenulisanLaporan | 5 |
| | |
| BAB II TINJAUAN PUSTAKA | |
| 2.1 Pemodelan 3 Dimensi | 7 |
| 2.2 Polygonal Modeling | 11 |
| 2.3 Procedural Modeling | 16 |
| 2.4 TrasnformasiGeometri | 20 |
| 2.4.1 TransformasiTranslasi | 21 |
| 2.4.2 TransformasiRotasi | 21 |
| 2.5 Euclidean Distance..... | 22 |
| | |
| BAB III PERANCANGAN DAN IMPEMENTASI APLIKASI | |
| 3.1 LokasiPenelitian..... | 24 |
| 3.2 Perancangan Proses..... | 24 |
| 3.2.1 Pembuatan Model Permukaan Tanah..... | 27 |
| 3.2.2 Pembuatan Model 3 Dimensi | 27 |
| 3.2.2.1 Model BadanBangunan | 33 |
| 3.2.2.2 Model Atap | 38 |
| 3.2.2.3 Model Kubah..... | 41 |
| 3.2.2.4 Model Jendela | 43 |
| 3.2.2.5 Model Pintu..... | 47 |
| 3.2.3 PeletakanFiturdengan Procedural Modeling | 50 |
| 3.2.3.1 Procedural Modeling padaAtap..... | 52 |

| | | |
|---------------------------------------|---------------------------------------|-----------|
| 3.2.3.2 | Procedural Modeling padaKubah..... | 54 |
| 3.2.3.3 | Procedural Modeling padaJendela | 56 |
| 3.2.3.4 | Procedural Modeling padaPintu..... | 59 |
| 3.3 | Perancangan Interface | 62 |
| BAB IV UJI COBA DAN PEMBAHASAN | | |
| 4.1 | UjiCobaPengukuranJarak | 67 |
| 4.1.1 | Langkah – LangkahUjiCoba | 67 |
| 4.1.2 | HasilUjiCoba..... | 69 |
| 4.2 | UjiCoba Level Of Detail..... | 72 |
| 4.3 | Pembahasan..... | 79 |
| 4.4 | IntegrasidenganAkidah Islam | 80 |
| BAB V PENUTUP | | |
| 5.1 | Kesimpulan | 83 |
| 5.2 | Saran | 84 |
| DAFTAR PUSTAKA | | 85 |

DAFTAR GAMBAR

| | | |
|-------------|---|----|
| Gambar 2.1 | Bagan representasi grafis 3 dimensi (<i>Sumber : Pavitran, 2008</i>)..... | 12 |
| Gambar 2.2 | Contoh bentuk dari polygon (<i>Sumber : Autodesk, 2007</i>) | 13 |
| Gambar 2.3 | Penulisan fungsi <code>fill</code> (<i>Sumber : Matlab, 2010</i>) | 14 |
| Gambar 2.4 | Contoh penggunaan fungsi <code>fill3</code> (<i>Sumber : Matlab, 2010</i>)..... | 14 |
| Gambar 2.5 | Contoh output fungsi <code>fill3</code> (<i>Sumber : Matlab, 2010</i>)..... | 14 |
| Gambar 2.6 | Output pemodelan 3D bangunan (<i>Sumber : Crysdiان, 2011</i>) | 15 |
| Gambar 2.7 | Output pemodelan 3D bangunan (<i>Sumber : Astutik, 2014</i>)..... | 16 |
| Gambar 2.8 | Simulasi ketetangaan titik pada model (<i>Sumber : Merrel, 2011</i>) ... | 18 |
| Gambar 2.9 | Bagan rule blok bangunan kuil Romawi (<i>Sumber : Saldana, 2014</i>). .. | 20 |
| Gambar 2.10 | Ilustrasi transformasi geometri translasi (<i>Sumber : http://www.regentsprep.org/, diakses pada tanggal 30 Mei 2015</i>) | 21 |
| Gambar 2.11 | Ilustrasi transformasi geometri rotasi (<i>Sumber : http://www.regentsprep.org/, diakses pada tanggal 30 Mei 2015</i>) | 26 |
| Gambar 3.3 | Model Permukaan Tanah | 27 |
| Gambar 3.4 | Diagram alur metode polygonal modeling | 28 |
| Gambar 3.5 | Kode program metode polygonal modeling..... | 28 |
| Gambar 3.6 | Diagram alur gabungan metode polygonal modeling dan transformasi geometri | 29 |
| Gambar 3.7 | Kode program gabungan metode poligonal modeling dan transformasi geometri | 30 |
| Gambar 3.8 | Sketsa manual pemodelan menggunakan 4 titik | 31 |
| Gambar 3.9 | Sketsa analisis manual rotasi poligon | 31 |
| Gambar 3.10 | Ilustrasi letak polygon pada titik pusat..... | 32 |
| Gambar 3.11 | Contoh membuat bangunan dengan <code>pd</code> dan <code>ld</code> | 32 |
| Gambar 3.12 | Hasil output model polygon | 33 |
| Gambar 3.13 | Badan bangunan tipe 1 | 34 |
| Gambar 3.14 | Badan bangunan tipe 2 | 34 |

| | | |
|-------------|--|----|
| Gambar 3.15 | Badan bangunan tipe 3 | 34 |
| Gambar 3.16 | Badan bangunan tipe 4 | 35 |
| Gambar 3.17 | Badan bangunan tipe 5 | 35 |
| Gambar 3.18 | Diagram alur tipe bangunan | 36 |
| Gambar 3.19 | Kode program tipe badan bangunan..... | 36 |
| Gambar 3.20 | Kode program untuk menyimpan data bangunan | 37 |
| Gambar 3.21 | Atap tipe 1 | 38 |
| Gambar 3.22 | Atap tipe 2 | 38 |
| Gambar 3.23 | Atap tipe 3 | 39 |
| Gambar 3.24 | Atap tipe 4 | 39 |
| Gambar 3.25 | Diagram alur tipe atap bangunan | 40 |
| Gambar 3.26 | Kode program tipe atap bangunan | 40 |
| Gambar 3.27 | Kubah tipe 1 | 41 |
| Gambar 3.28 | Kubah tipe 2 | 42 |
| Gambar 3.29 | Diagram alur tipe kubah bangunan | 42 |
| Gambar 3.30 | Kode program tipe kubah bangunan | 43 |
| Gambar 3.31 | Jendela tipe 1 | 44 |
| Gambar 3.32 | Jendela tipe 2 | 44 |
| Gambar 3.33 | Jendela tipe 3 | 45 |
| Gambar 3.34 | Jendela tipe 4..... | 45 |
| Gambar 3.35 | Diagram alur tipe jendela bangunan..... | 46 |
| Gambar 3.36 | Kode program tipe jendela bangunan..... | 46 |
| Gambar 3.37 | Pintu tipe 1 | 48 |
| Gambar 3.38 | Pintu tipe 2 | 48 |
| Gambar 3.39 | Diagram alur tipe pintu bangunan..... | 49 |
| Gambar 3.40 | Kode program tipe pintu bangunan..... | 49 |
| Gambar 3.41 | Ilustrasi garis longitude dan latitude pada badan bangunan sudut 90 | 51 |
| Gambar 3.42 | Kode program mendapatkan titik longitude dan latitude baru.... | 51 |
| Gambar 3.43 | Ilustrasi garis longitude dan latitude pada badan bangunan sudut 0 | 51 |

| | | |
|-------------|--|----|
| Gambar 3.44 | Kode program mendapatkan titik longitude dan latitude baru | 52 |
| Gambar 3.45 | Diagram alur prosedur penempatan atap | 53 |
| Gambar 3.46 | Kode program prosedur penempatan atap | 53 |
| Gambar 3.47 | Diagram alur prosedur penempatan kubah | 54 |
| Gambar 3.48 | Kode program prosedur penempatan kubah | 55 |
| Gambar 3.49 | Kode program untuk memanggil database jendela | 56 |
| Gambar 3.50 | Kode program prosedur penempatan jendela..... | 58 |
| Gambar 3.51 | Kode program untuk memanggil database pintu..... | 59 |
| Gambar 3.52 | Kode program prosedur penempatan pintu | 61 |
| Gambar 3.53 | Halaman interface awal aplikasi | 63 |
| Gambar 3.54 | Halaman interface ketika aplikasi sudah dijalankan | 63 |
| Gambar 3.55 | Hasil model 3D bangunan pada level 0..... | 64 |
| Gambar 3.56 | Hasil model 3D bangunan pada level 1..... | 65 |
| Gambar 3.57 | Hasil model 3D bangunan pada level 2..... | 65 |
| Gambar 4.1 | Pengambilan titik (x1,y1) pada bangunan..... | 68 |
| Gambar 4.2 | Pengambilan titik (x2,y2) pada bangunan..... | 68 |
| Gambar 4.3 | Hasil pengukuran dengan euclidean distance | 69 |
| Gambar 4.4 | Tampilan bangunan fakultas ekonomi pada level 0 | 73 |
| Gambar 4.5 | Tampilan bangunan fakultas ekonomi pada level 1 | 73 |
| Gambar 4.6 | Tampilan bangunan fakultas ekonomi pada level 2..... | 74 |
| Gambar 4.7 | Tampilan bangunan fakultas saintek pada level 2..... | 74 |
| Gambar 4.8 | Model 3D bangunan pada penelitian sebelumnya (<i>Sumber : Astutik, 2014</i>) | 75 |
| Gambar 4.9 | Perbandingan Gedung C dengan LOD0, LOD1, LOD2 dan tipe 2 | 76 |
| Gambar 4.10 | Perbandingan Gedung SC dengan LOD0, LOD1, LOD2 dan tipe 3 | 76 |
| Gambar 4.11 | Perbandingan Masjid Tarbiyah dengan LOD0, LOD1, LOD2 dan tipe 4 | 77 |

DAFTAR TABEL

| | | |
|-----------|---|----|
| Tabel 4.1 | Hasil uji coba pengukuran jarak antar bangunan | 70 |
| Tabel 4.2 | Hasil uji coba panjang dan lebar tiap bangunan | 72 |
| Tabel 4.3 | Hasil uji coba perbandingan bentuk antar bangunan | 77 |



**HALAMAN PERNYATAAN
ORISINALITAS PENELITIAN**

Saya yang bertandatangan di bawah ini:

Nama : UlfatulMufida
 NIM : 11650088
 Fakultas/Jurusan : Sains Dan Teknologi / TeknikInformatika
 JudulPenelitian : **PENERAPAN METODE PROCEDURAL MODELING
 DAN TRANSFORMASI GEOMETRI DALAM
 PEMODELAN 3D BANGUNAN**

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut

Malang, 8 Juni 2015

Yang Membuat Pernyataan,

UlfatulMufida
 NIM. 11650088

ABSTRAK

Mufida, Ulfatul. 2015. **Penerapan Metode Procedural Modeling dan Transformasi Geometri dalam Pemodelan 3D Bangunan.** Pembimbing : (1) Dr. Cahyo Crysdiyan (2) Dr. M. Amin Hariyadi, M.T.

Kata Kunci : Bangunan, *Procedural Modeling*, *Transformasi Geometri*.

Model 3D kota merupakan salah satu aspek penting dalam sejumlah aplikasi yang berhubungan dengan perencanaan kota. Tersedianya model 3D kota memberikan kontribusi yang signifikan dalam pengembangan model kota seperti yang diterapkan di seluruh dunia. Hal ini menyebabkan peningkatan permintaan untuk pembuatan model 3D bangunan sebagai data yang diperlukan dalam berbagai tugas yang berhubungan dengan pengukuran dan konstruksi.

Namun banyaknya data yang harus dikelola mengakibatkan pembuatan 3D model masih terasa kompleks dan perlu prosedur pengembangan yang lebih efektif. *Procedural Modeling* merupakan salah satu metode yang berfokus untuk menciptakan model dari satu set aturan. Metode ini dirancang untuk secara otomatis atau semi-otomatis menghasilkan model yang lebih kompleks dari set model sederhana sehingga cukup untuk membuat model 3D kota yang kompleks.

Berdasarkan dari penelitian dapat diketahui bahwa *procedural modeling* mampu memberikan hasil visualisasi 3D dari bangunan yang sampai ke level LOD2 (Level Of Detail). LOD2 sendiri merupakan tingkatan detail dari sebuah bangunan yang dapat menampilkan visualisasi bangunan beserta eksteriornya yaitu atap, kubah, pintu, jendela, tiang dan teras.

ABSTRACT

Mufida, Ulfatul. 2015. Application of Procedural Modeling Method and Transformation geometry in 3D Building Modeling. Supervisor: (1) Dr. Cahyo Crysdiandian (2) Dr. M. Amin Hariyadi, M.T.

Keywords: Building, Procedural Modeling, Transformation Geometry.

3D city models is one important aspect in a number of applications related to city planning. The availability of 3D city models to contribute to the development of a model city as applied in the whole world. This leads to increased demand for the creation of 3D models of buildings as the necessary data in a variety of tasks related to the measurement and construction.

But the amount of data that must be managed resulting in the manufacture of complex 3D models is still felt and necessary development of more effective procedures. Procedural Modeling is one method that focuses on creating a model of a set of rules. This method is designed to automatically or semi - automatically generate models more complex than a simple set of models that are suitable for making complex 3D model city.

Based on the research is that procedural modeling, delivers a 3D visualization of the building up to the level LOD2 (Level Of Detail). LOD2 itself is the level of detail of a building that can display a visualization of the building and the exterior is the roof, dome, doors, windows, columns and terraces.



BAB I

PENDAHULUAN

1.1 Latar Belakang

Informasi tentang 3D kota merupakan hal yang penting untuk sejumlah aplikasi yang berhubungan dengan perencanaan kota, pengendalian lingkungan atau untuk telekomunikasi. Hal ini menyebabkan meningkatnya permintaan untuk pembuatan model 3D bangunan sebagai data yang diperlukan dalam berbagai tugas yang berhubungan dengan sejumlah pengukuran, perencanaan, konstruksi, lingkungan, transportasi, energi, manajemen properti dan penempatan pemancar (Lang et al, 1996). Sehingga ketersediaan akan model 3D bangunan tumbuh menjadi salah satu aspek kebutuhan yang harus dipenuhi di masyarakat, terlebih untuk menunjang tugas dalam bidang perencanaan kota.

Pada masa selanjutnya model 3D bangunan semakin berkembang dan menjadi alat penting dalam perencanaan kota, hal ini dikarenakan peta kota 2D yang biasa digunakan sebagai acuan perencanaan kota kurang bisa menggambarkan kondisi nyata dari lingkungan disekitarnya. Sehingga permintaan akan model 3D struktur bangunan menjadi bertambah, dan permintaan ini telah dikonfirmasi oleh survey OEEPE pada model kota 3D oleh *European Organization for Experimental Photogrammetric Research (OEEPE, cf. Fuchs et al 1998)* diketahui bahwa 95% dari peserta menyebutkan data 3D bangunan menjadi aspek yang paling diminati dalam model kota, lalu diikuti oleh informasi tentang lalu lintas jaringan (sekitar 85%) dan informasi 3D tentang vegetasi

(sekitar 75%) (Forstner, 1999). Hasil survey yang tinggi pada model 3D bangunan juga berbanding lurus dengan banyaknya peneliti yang tengah mengembangkan sebuah sistem generasi otomatis model kota 3D. Pembuatan sebuah model kota 3D yang akurat menjadi perlu untuk diproduksi dalam waktu yang singkat dan dapat tersedia secara luas di pasaran (Takase, 2003).

Kontribusi nyata dalam pengembangan model kota dengan menggunakan visualisasi 3D yang dikembangkan oleh para peneliti dibuktikan dengan dukungannya pada pengembangan kota diseluruh dunia seperti yang diterapkan di Berlin (Dollner et al, 2006), London (Batty, 2007) dan Kyoto (Takase et al, 2006). Walaupun banyak peneliti yang mengembangkannya, pembuatan 3D model masih terasa kompleks karena banyaknya informasi yang harus dikelola, sehingga perlu untuk mengembangkan prosedur pembuatan model kota 3D dengan cara yang lebih sederhana namun tetap efektif (Crysdian, 2011).

Procedural Modeling merupakan istilah umum untuk sejumlah teknik computer grafis untuk membuat model 3D dan tekstur dari set aturan. L-Systems, fractal dan pemodelan generated adalah teknik pemodelan procedural karena sama-sama menerapkan algoritma untuk menghasilkan adegan. Seperangkat aturan dapat dimasukkan ke dalam algoritma, dikonfigurasi oleh parameter atau seperangkat aturan yang terpisah dari mesin evaluasi. *Procedural Modeling* berfokus pada menciptakan model dari satu set aturan atau meningkatkan model otomatis. Metode ini sering diterapkan ketika akan membuat model 3D yang terlalu rumit jika menggunakan pemodelan 3D generik, biasanya digunakan untuk pemodelan tanaman, arsitektur atau lanskap (Ganster, 2009).

Dengan banyaknya benda dan lingkungan yang mengandung struktur yang berulang dan mempunyai kemiripan dapat dimanfaatkan untuk memudahkan dalam proses pemodelan dengan menggunakan metode *Procedural Modeling*. Metode ini dirancang untuk secara otomatis atau semi-otomatis menghasilkan model yang lebih kompleks dari set model yang sederhana (Merrel et al, 2011). Sedangkan di dalam Al-Qur'an, Allah SWT telah menyinggung masalah penciptaan suatu hal secara berulang, hal ini dijelaskan dalam firman Allah SWT yang terdapat pada surat Al-Waqi'ah ayat 62 :

وَلَقَدْ عَلِمْتُمُ النَّشْأَةَ الْأُولَىٰ فَلَوْلَا تَذَكَّرُونَ ﴿٦٢﴾

“Dan Sesungguhnya kamu telah mengetahui penciptaan yang pertama, maka mengapakah kamu tidak mengambil pelajaran (untuk penciptaan yang kedua)?”

(QS.Al-Waqi'ah:62)

Dari ayat Al-Qur'an diatas telah disebutkan bahwa untuk menciptakan atau membuat suatu hal, kita dapat mengambil pelajaran atau ilmu dari cara penciptaan yang pertama. Hal ini dikaitkan dengan penciptaan sesuatu yang banyak dan kompleks pada pembuatan model 3D bangunan yang mempunyai tingkat kerumitan yang tinggi terutama untuk mendesain bagian fitur seperti jendela, pintu, dan tiang. Semisal pada bangunan yang mempunyai fitur jendela yang jumlahnya banyak, kita dapat mengambil metode dari cara membuat fitur yang pertama sehingga cara tersebut dapat digunakan kembali pada pembuatan fitur yang kedua, ketiga dan seterusnya.

Transformasi geometri yaitu sebuah ilmu geometri yang menggunakan system koordinat dan menggunakan prinsip-prinsip ilmu aljabar dan analisis matematika (Putra, 2010). Metode ini digunakan untuk menentukan arah putar rotasi bangunan.

1.2 Identifikasi Masalah

Berdasarkan penjelasan pada latar belakang, maka identifikasi masalah dari penelitian ini adalah :

1. Apakah metode procedural modeling dan transformasi geometri dapat diterapkan dalam visualisasi 3D bangunan?
2. Seberapa baik metode procedural modeling dan transformasi geometri dalam membangun visualisasi 3D bangunan?

1.3 Tujuan Penelitian

Berdasarkan identifikasi masalah, maka tujuan dari penelitian ini adalah :

1. Membuktikan bahwa metode procedural modeling dan transformasi geometri dapat diterapkan dalam visualisasi 3D bangunan.
2. Mengukur kinerja dari hasil visualisasi 3D bangunan menggunakan metode procedural modeling dan transformasi geometri.

1.4 Batasan Masalah

Agar penelitian ini tidak menyimpang dari permasalahan maka perlu adanya batasan masalah yaitu :

1. Model bangunan yang akan di visualisasikan ke dalam bentuk 3D adalah bangunan Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Model 3D yang akan di visualisasikan berupa badan bangunan, atap, pintu, jendela, tangga, tiang dan teras.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dalam penelitian ini adalah :

1. Mempermudah pembuatan visualisasi 3D bangunan dengan banyak tipe dan fitur bangunan.
2. Membuat visualisasi 3D bangunan dengan ukuran dan jarak yang tepat sesuai bentuk nyata.

1.6 Sistematika Penelitian

Penelitian skripsi ini tersusun dalam lima bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Pendahuluan, membahas tentang latar belakang masalah, identifikasi masalah, batasan masalah, tujuan penyusunan tugas akhir, metodologi dan sistematika penyusunan tugas akhir.

BAB II Landasan Teori

Landasan teori berisikan beberapa teori yang mendasari dalam penyusunan tugas akhir ini. Adapun yang dibahas dalam bab ini adalah dasar teori yang

berkaitan dengan pembahasan tentang model 3 dimensi bangunan, Procedural Modeling dan Transformasi Geometri.

BAB III Analisa dan Perancangan

Menganalisa kebutuhan system untuk membuat aplikasi meliputi spesifikasi kebutuhan software dan langkah-langkah pembuatan Aplikasi Pemodelan 3D Bangunan menggunakan Metode Procedural Modeling dan Transformasi Geometri.

BAB IV Uji Coba dan Pembahasan

Menjelaskan tentang pengujian Aplikasi Pemodelan 3D Bangunan menggunakan metode Procedural Modeling dan Transformasi Geometri yang telah diterapkan.

BAB V Penutup

Bab ini berisikan kesimpulan dan saran dari sistem yang dibuat.

BAB II

TINJAUAN PUSTAKA

2.1 Pemodelan 3 Dimensi

Kebutuhan akan model 3D kota tumbuh dan berkembang secara pesat di berbagai bidang. Dalam pergeseran stabil dari 2D yang sederhana ke dalam bentuk 3D GIS, pembuatan sejumlah model kota 3D yang akurat menjadi perlu untuk diproduksi dalam kurun waktu yang singkat dan tersedia luas di pasaran. Para penulis telah mengembangkan sebuah system generasi otomatis model kota 3D dengan menggunakan data Laser Profile, 2D peta digital dan citra udara yang diproses oleh perangkat lunak yang baru dikembangkan. MapCube, model kota 3D yang dihasilkan oleh system sudah mencakup semua kota besar di Jepang pada tahun 2002. Namun belum ada VR (*Virtual Reality*) penampil yang tepat yang dapat menangani secara realtime, sehingga membuat Takase et al (2003) mengembangkan penampil VR baru yang disebut UrbanViewer yang dapat membuat hasil terbaik dari pembuatan model kota 3D terutama pada MapCube. Selanjutnya Takase et al juga mengembangkan jenis baru perangkat layar dengan sensor optic yang disebut NETRAX yang membuat pengguna dapat mengoperasikannya dengan panel layar sentuh, dengan proses akurasi yang lebih dan masih banyak lagi kelebihan yang lain.

Pembahasan mengenai bangunan juga sempat disinggung oleh Allah SWT dalam kitab suci Al-Qur'an tepatnya pada surat Al-'Imran ayat 96, Allah berfirman :

إِنَّ أَوَّلَ بَيْتٍ وُضِعَ لِلنَّاسِ لَلَّذِي بِبَكَّةَ مُبَارَكًا وَهُدًى لِّلْعَالَمِينَ ﴿٩٦﴾

“*Sesungguhnya rumah yang mula-mula dibangun untuk (tempat beribadat) manusia, ialah Baitullah yang di Bakkah (Mekah) yang diberkahi dan menjadi petunjuk bagi semua manusia.*” (QS.Al-‘Imran:96)

Dalam QS.Al-‘Imran ayat 96 disebutkan bahwa Baitullah merupakan bangunan tempat beribadah yang pertama dibangun sekaligus merupakan bangunan yang diberkahi dan menjadi petunjuk bagi semua manusia. Maksud dari petunjuk disini berupa petunjuk keagamaan karena merupakan arah kiblat seluruh umat dalam melaksanakan sholat. Petunjuk keduniaan karena Ka’bah merupakan titik sentral arah bumi dari seluruh dunia. (<http://saifuddinasm.com>, 2013). Adanya bangunan Ka’bah, menjadikan umat muslim dapat membangun rumah ibadah di daerahnya masing-masing dengan mengacu pada arah kiblat yang sudah ditentukan. Dari sinilah sebuah pemodelan bangunan menjadi hal yang penting karena dapat membantu pihak lain untuk bisa membangun bangunan dengan melihat konstruksi bangunan dan juga bentuk yang ada.

Salah satu hambatan utama dalam mengembangkan model virtual 3D kota adalah waktu dan biaya pembuatan model data yang tidak efisien. Pemodelan geometris manual dapat diterima dalam model kota 3D dengan skala kecil tetapi gagal jika model kota 3D digunakan untuk daerah perkotaan besar atau jika harus dikelola dalam jangka panjang. Oleh karena itu, model kota harus didasarkan pada metode akuisisi otomatis dan semi-otomatis sedapat mungkin. Namun, jika aspek semantic dan topologi diabaikan, model ini hanya dapat digunakan untuk tujuan

visualisasi tetapi tidak untuk pertanyaan tematik, analisis tugas, atau penggalian data spasial. Karena terbatasnya usabilitas dari model menghambat penggunaan yang lebih luas maka pemodelan dengan pendekatan yang lebih umum harus diambil dalam rangka untuk memenuhi kebutuhan berbagai bidang aplikasi. Sehingga Dollner et al (2006) memperkenalkan arsitektur perangkat lunak dan sistem baru untuk membuat, mengelola, mengamankan dan dengan skala besar model kota 3D yang disebut LandXplorer. Perangkat ini telah dikembangkan di Hasso Plattner- Institute sebagai platform perangkat umum untuk teknik geovisualization 3D dan system. Dengan LandXplorer, sejumlah produk untuk perencanaan dan model perkotaan telah dibuat termasuk LandXplorer Studio. Sistem ini mendukung model 3D medan, model bangunan 3D, model 3D ruang jalan, dan model 3D vegetasi disebut sebagai objek kelas satu yaitu mereka mewakili komponen utama yang berbesa dengan generasi pertama grafis berorientasi 3D kota.

Selanjutnya pemodelan 3D kota dan visualisasi perkotaan menggunakan teknologi fotogrametri adalah salah satu topic penelitian yang paling berkembang di arsitektur digital. Kobayashi (2006) memperkenalkan metode yang menciptakan model 3D dari foto udara kota menggunakan alat fotogrametri komersial dan membahas efisiensi dan efektifitas dalam segi waktu, tenaga, dan usabilitasnya. LIDAR (Light detection and ranging) dan Fotogrametri adalah teknologi yang umum digunakan dalam ekstraksi geometri 3D. Alat dari LIDAR menstransmisikan cahaya untuk target dan mengukur dengan menggunakan sinyal yang dipantulkan. Pendekatan lain adalah untuk mendapatkan data LIDAR dari

tanah dan ekstrak geometri yang rumit seperti komponen arsitektur dan struktur sipil. Satu set poin diekstrak dari LIDAR diubah menjadi polygon. Fotogrametri mempunyai solusi lain, seperti halnya LIDAR, Fotogrametri dapat digunakan baik untuk foto udara maupun tanah. Foto udara yang digunakan untuk mengekstrak bentuk abstrak bangunan dan foto tanah yang digunakan untuk mengekstrak rinciannya.

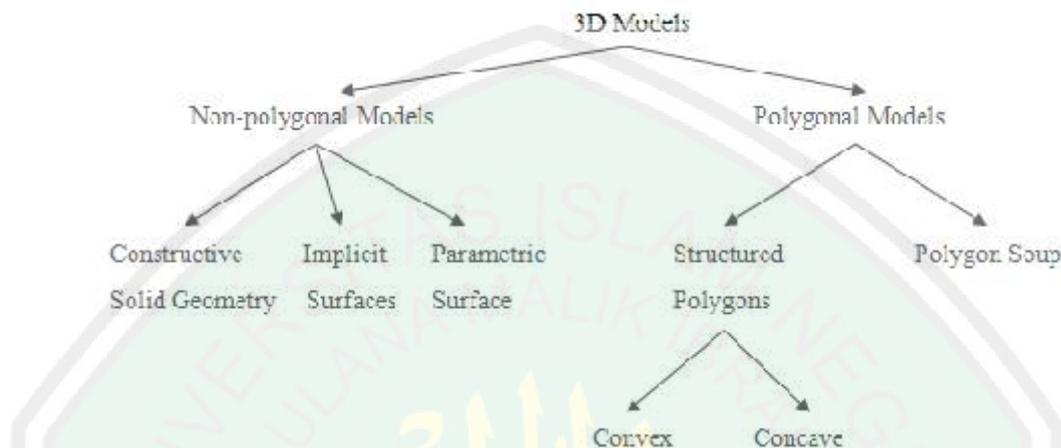
Pemodelan kota 3D mempunyai kontribusi besar dalam mendukung pengemvangan kota diseluruh dunia seperti yang diterapkan di Berlin, London dan Kyoto. Walaupun banyak peneliti yang telah bekerja untuk mengembangkan bidang ini, namun model kota 3D mempunyai kompleksitas tersendiri yaitu banyaknya informasi perkotaan yang harus dikelola, terpadu dan dikomunikasikan antara pihak yang terlibat dalam tahap implementasi. Sehingga pada tahun 2011, Crysdiان akhirnya melakukan penelitian untuk mengembangkan model kota 3D dengan biaya minimal namun tetap efisien menggunakan pendekatan pemodelan yang efisien berbasis *Polygonal Modeling*. Sebuah bagian dari kota Malang yang secara geografis terletak di 7,9939 - 7,9881 lintang selatan dan 112,6784 - 112,6836 bujur timur divisualisasikan ke dalam bentuk 3D. Untuk membuatnya, satu set model telah dikembangkan yaitu model permukaan, model bangunan, model vegetasi, dan model jalan yang merupakan batas elemen untuk menyusun model yang dikembangkan. Selanjutnya adalah mengembangkan masing-masing model individu untuk menyusun sebuah kota, fase untuk mengintegrasikan model ini dilakukan dengan menggunakan pendekatan overlay dimana model permukaan menjadi dasar untuk peletakan model lainnya.

Pemodelan 3D juga bisa digunakan pada objek rupa bumi yang diperlukan untuk mempermudah dalam mendapatkan informasi mengenai kontur tanah. Hal ini sudah dilakukan oleh Tarecha (2013) yaitu dengan memvisualisasikan 3D rupa bumi dengan menggunakan data Global Digital Elevation Model (GDEM) yang merupakan data ketinggian tanah yang membentuk kontur bumi, Dalam penelitian ini Tarecha menggunakan data GDEM ASTER yaitu data yang memiliki ketinggian 30 meter. Data GDEM ASTER tersebut diolah menggunakan metode *Euclidean distance* untuk menghitung jarak antara dua titik, metode *forward chaining* untuk menyusun empat ubin data GDEM ASTER yang terdekat dengan titik tengah, algoritma round half down dan brute force closest-pair untuk mencari pasangan titik terdekat. Algoritma round half down dan brute force closest-pair dibandingkan untuk membuktikan algoritma mana yang paling cepat mencari titik terdekat. Untuk meningkatkan ketelitian digunakan interpolasi linier sehingga ketelitian yang sebelumnya 30 meter menjadi lebih tinggi yaitu 15 meter. Aplikasi visualisasi disajikan dalam bentuk halaman web yang dapat diakses oleh pengguna melalui web browser.

2.2 Polygonal Modeling

Model 3 dimensi yang realistis sangat berguna untuk operasi seperti rotasi, zoom (perbesaran) dan dalam pembuatan bagian-bagian yang membutuhkan struktur model yang dinamis. Oleh karena itu, model 3 dimensi diharuskan menanggapi perubahan bentuk dan manipulasi yang ada dalam kenyataan. Ada

banyak jenis model representasi yang digunakan dalam grafis 3 dimensi, representasi secara luas diklasifikasikan dalam bentuk bagan pada Gambar 2.1.



Gambar 2.1 Bagan representasi grafis 3 dimensi (Sumber : Pavithran, 2008)

Polygonal Modeling adalah sebuah pendekatan untuk pemodelan objek dengan mewakili atau mendekati permukaan dengan menggunakan garis polygonal. Unit dasar yang digunakan dalam pemodelan garis simpul adalah titik dalam ruang 3 dimensi. Tiga simpul terhubung dalam urutan tertentu membentuk segitiga yang merupakan polygon sederhana dalam ruang Euclidean. Sifat datar segitiga memudahkan untuk menentukan permukaan normalnya, yang digunakan untuk menentukan pencahayaan dan pewarnaan. Sekelompok polygon yang dihubungkan bersama-sama dengan simpul disebut sebagai mesh. Kelemahan utama dengan polygon adalah bahwa ia tidak mampu mewakili permukaan yang melengkung, sehingga sejumlah besar dari mereka harus menggunakan kurva perkiraan secara manual (Pavithran, 2008).

Poligon merupakan salah satu tipe geometri yang dapat dipergunakan untuk menciptakan model 3 dimensi di Maya, Autodesk dan sebagainya. Polygon

sangat berguna untuk membangun banyak tipe dari konten 3 dimensi dan digunakan secara luas untuk pembangunan konten 3 dimensi (Autodesk, 2007). Polygon sendiri adalah bentuk sisi lurus (3 atau lebih sisi) yang didefinisikan oleh tiga poin dimensi (simpul) dan garis-garis lurus yang menghubungkan tepitepinya. Wilayah interior polygon disebut sebagai face. Simpul, tepi, dan face adalah komponen dasar dari polygon, ketika akan memodifikasikan polygon maka komponen dasar ini yang akan digunakan. Ketika memodelkan polygon biasanya akan menggunakan polygon dengan tiga sisi yang disebut segitiga atau polygon empat sisi yang diebut segiempat. Sebuah polygon individu (biasa disebut face) didefinisikan sebagai daerah yang dibatasi oleh tiga atau lebih simpul dan tepi yang terkait. Ketika banyak face yang terhubung bersama-sama mereka menciptakan jaringan face yang disebut polyset atau benda polygonal. Contoh dari bentuk polygon segi empat dan polygon segi lima ditunjukkan pada Gambar 2.2.



Gambar 2.2 Contoh bentuk dari polygon (Sumber : Autodesk, 2007)

Teknik dasar dalam pembuatan model 3 dimensi dari polygon yaitu dengan memecah objek menjadi banyak polygon. Setiap polygon yang ada dibuat dengan titik koordinat x , y dan z . Dalam IDE Matlab, terdapat beberapa cara untuk memvisualisasikan bentuk 3 dimensi, salah satu yang paling efektif adalah dengan menggunakan polygon atau metode polygonal modeling. Fungsi khusus

untuk memodelkan bentuk 3 dimensi yang terbuat dari polygon yaitu menggunakan fungsi fill, fill3, isosurface, isocaps dan patch. Berikut merupakan contoh cara penulisan dari fungsi fill yang ditunjukkan pada Gambar 2.3.

```
fill3(x-coordinates, y-coordinates, z-coordinates, colordata);
fill(x-coordinates, y-coordinates, colordata);
```

Gambar 2.3 Penulisan fungsi fill (*Sumber : Matlab, 2010*)

Keterangan Gambar 2.3 :

x-coordinates merupakan titik koordinat x yang akan dibuat

y-coordinates merupakan titik koordinat y yang akan dibuat

z-coordinates merupakan titik koordinat z yang akan dibuat

fill3 merupakan fungsi untuk membuat objek bidang 3 dimensi

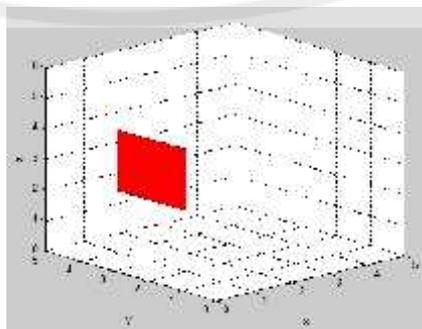
fill merupakan fungsi untuk membuat objek bidang 2 dimensi

Pada Gambar 2.4 ditunjukkan salah satu contoh penggunaan fungsi fill3 untuk membuat sebuah polygon segiempat.

```
x = [1 1 1 1];
y = [2 4 4 2];
z = [2 2 4 4];
fill3(x, y, z, 'red');
```

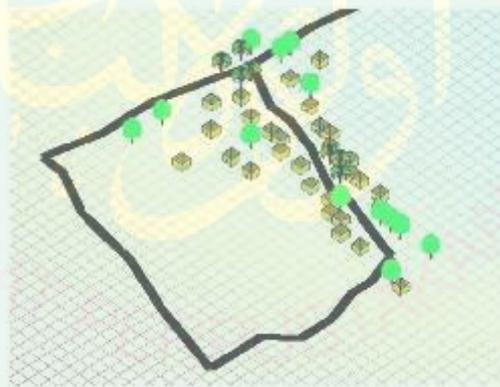
Gambar 2.4 Contoh penggunaan fungsi fill3 (*Sumber : Matlab, 2010*)

Hasil dari Gambar 2.4 dapat menampilkan sebuah bentuk polygon yang ditunjukkan pada Gambar 2.5



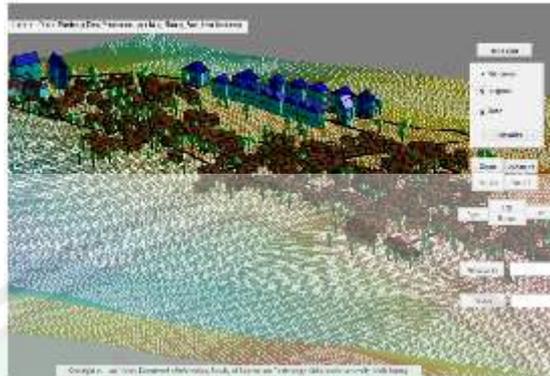
Gambar 2.5 Contoh output fungsi fill3 (*Sumber : Matlab, 2010*)

Crysdiان (2011) melakukan penelitian dengan menggunakan *Polygonal Modeling* sebagai metode untuk membuat model kota 3D di sebuah daerah di kota Malang. Pendekatan *Polygonal Modeling* merupakan metode yang sederhana namun efektif untuk membangun model kota 3D. Untuk membuat model kota 3D, satu set model telah dikembangkan yaitu model permukaan, model bangunan, model vegetasi dan model jalan yang menggunakan satu set matriks. Setelah mengembangkan masing-masing model individu, dilakukan fase integrasi dengan menggunakan pendekatan overlay dimana model permukaan menjadi dasar model lainnya berupa model bangunan, model vegetasi dan model jalan. Dengan menggunakan scenario ini, pengembangan model kota 3D dapat dilakukan dengan cara yang efisien. Hasil output dari penelitian ini ditunjukkan pada Gambar 2.6



Gambar 2.6 Output pemodelan 3D bangunan (Sumber : Crysdiان, 2011)

Astutik (2014) juga melakukan penelitian yang sama dengan menggunakan polygonal modeling untuk memvisualisasikan daerah yang lebih besar ke dalam bentuk 3 dimensi. Penelitian ini dilakukan di Dusun Plumbang, Desa Pandansari, Kecamatan Ngantang, Kabupaten Malang yang letak geografisnya adalah 7.88719 - 7.877716 Lintang Selatan dan 112.354152 - 112.361621 Bujur Timur dengan hasil output yang ditunjukkan pada Gambar 2.7.



Gambar 2.7 Output pemodelan 3D bangunan (Sumber : Astutik, 2014)

Hasil dari penelitian ini mempunyai tingkat keakuratan yang tinggi dalam memvisualisasikan bentuk 3 dimensi, namun masih perlu dikembangkan lebih lanjut dalam segi bentuk pemodelannya. Dari Gambar 2.7 dapat ditemukan bahwa bentuk 3 dimensi dari bangunan masih terbatas menjadi 3 tipe dan belum sesuai dengan bentuk aslinya membuat visualisasi tersebut menjadi kurang detail dalam pemberian informasi bangunan. Sehingga diperlukan sebuah penelitian lagi untuk mendapatkan visualisasi 3 dimensi bangunan yang lebih akurat dalam segi bentuk dan fitur yang ada dalam bangunan. Dalam hal ini peneliti bertujuan untuk membuat model visualisasi 3 dimensi yang dapat menghasilkan bentuk bangunan lebih detail dari beberapa bentuk fitur yang sederhana dengan menerapkan beberapa aturan pada setiap bangunan.

2.3 Procedural Modeling

Procedural Modeling merupakan istilah umum untuk sejumlah teknik computer grafis untuk membuat model 3D dan tekstur dari set aturan. L-Systems, fractal dan pemodelan generated adalah teknik pemodelan procedural karena sama-sama menerapkan algoritma untuk menghasilkan adegan. Seperangkat

aturan dapat dimasukkan ke dalam algoritma, dikonfigurasi oleh parameter atau seperangkat aturan yang terpisah dari mesin evaluasi. *Procedural Modeling* berfokus pada menciptakan model dari satu set aturan, atau meningkatkan model otomatis. Metode ini sering diterapkan ketika akan membuat model 3D yang terlalu rumit jika menggunakan pemodelan 3D generik, biasanya digunakan untuk pemodelan tanaman, arsitektur atau lanskap (Ganster, 2009).

Banyaknya benda dan lingkungan yang mengandung struktur berulang dan kesamaan dapat dimodelkan dengan lebih mudah menggunakan teknik pemodelan procedural. Teknik pemodelan ini dirancang untuk secara otomatis atau semi-otomatis menghasilkan model yang kompleks. Dalam hal ini, Merrel (2011) membahas masalah bagaimana menghasilkan model yang kompleks dengan menggunakan model sintesis. Algoritma model Sintesis menerima bentuk 3D yang sederhana sebagai input dan kemudian menghasilkan model yang lebih besar dengan lebih kompleks yang bentuk dan fitur lokalnya menyerupai input yang dimasukkan. Penelitian ini menyajikan algoritma model Sintesis yang memberikan kontrol pengguna yang lebih besar terhadap output yang akan dihasilkan.

Pendekatan ini diawali dengan input bentuk sederhana yang berupa set polygon, model sistesis akan menghasilkan model yang menyerupai dari bentuk inputnya. Rumus persamaan yang digunakan ditunjukkan pada Rumus 2.1 berikut :

$$M(x + \delta) = E(x' + \delta) \quad (2.1)$$

Keterangan Rumus 2.1:

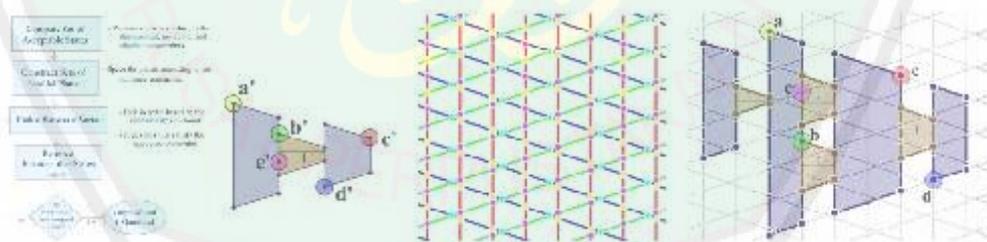
M = model dari output yang dihasilkan

E = model dari input yang dimasukkan

x = ruang dalam sebuah dimensi dengan $x \in R^d$ (ruang 3 dimensi)

δ = perubahan yang terjadi

Model input yang dinotasikan sebagai E semisal mempunyai titik a' , b' , c' , d' dan e' yang merupakan elemen x' , maka akan menghasilkan titik a , b , c , d , e dimana titik tersebut merupakan elemen x . Terdapat adjacency constraint (batasan) yang menyatakan bahwa setiap titik didalam x susunan ketetanggaannya harus sama / cocok dengan titik yang ada di dalam x' sesuai dengan persamaan tersebut. Constraint ini memastikan bahwa setiap tetangga yang ada di dalam M adalah sama persis dengan yang ada di dalam E seperti yang ditunjukkan pada Gambar 2.8.

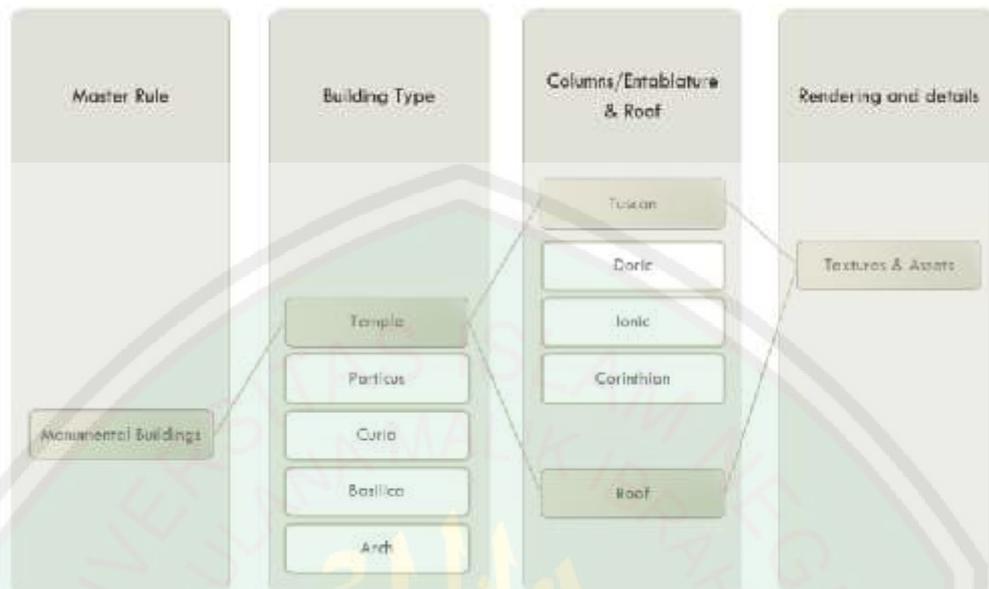


Gambar 2.8 Simulasi ketetangaan titik pada model (Sumber : Merrel, 2011)

Saldana (2014) menyajikan serangkaian aturan procedural untuk menciptakan model 3D arsitektur Romawi dan Helenistik dan lingkungan perkotaan. Istilah “aturan” dalam pemodelan procedural mengacu pada kode computer yang menghasilkan model 3D. Tidak seperti software tradisional yang menggunakan polygon untuk mensimulasikan bentuk, model procedural

memerlukan penggunaan bahasa pemrograman computer dalam deskripsi semantic tekstual bangunan yang kemudian menghasilkan model polygonal. Penelitian ini mengintegrasikan pendekatan untuk pemodelan procedural karena alur kerjanya menjelaskan tujuan pada pendekatan holistic data arsitektur seperti menggabungkan database geografis, dokumentasi, model 3D dan aturan deskriptif. Dalam penelitian Saldana ini, menggunakan sebuah kuil Romawi sebagai objek penelitian, pemodelan dimulai dengan database GIS yang dikumpulkan dan juga data survey. Karena hasilnya divisualisasikan dalam 3 dimensi, model situs kuil diproduksi dengan menggunakan garis kontur dan titik elevasi. Selanjutnya sejumlah atribut ditugaskan untuk secara otomatis direferensikan oleh aturan procedural dan akan ditampilkan dengan model akhir.

Untuk menulis aturan procedural yang efektif kita harus bekerja mundur sampai batas tertentu dan tahu untuk apa aturan itu akan digunakan. Aturan ini mengikuti struktur modular, dimana aturan utama referen ke beberapa aturan yang lebih kecil seperti jumlah kolom, serta entablature, pediment, atap dan tekstur. Dengan demikian, jumlah maksimum variasi dapat dihasilkan dengan duplikasi minimal kode. Deretan aturan bertindak seperti sebuah perpustakaan blok bangunan modular yang dapat dikombinasikan untuk membentuk tipologi yang berbeda, yang pada akhirnya dapat menghasilkan sebuah kota. Pada Gambar 2.9 ditunjukkan bagan rule blok dari bangunan kuil Romawi yang dibuat.



Gambar 2.9 Bagan rule blok bangunan kuil Romawi (Sumber : Saldana, 2014)

Bagan aturan seperti pada Gambar 2.9 tersebutlah yang juga akan digunakan dalam penelitian ini, dimana beberapa tipe bangunan akan mengacu ke pada atribut yang lebih kecil yaitu tipe fitur-fitur yang digunakan sehingga dapat menghasilkan model bangunan 3D yang kompleks dan bervariasi.

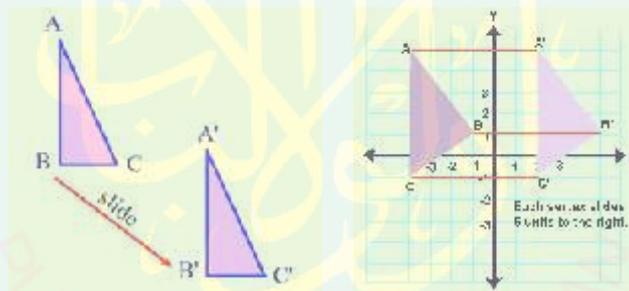
2.4 Transformasi Geometri

Transformasi geometri, juga dikenal sebagai geometri koordinasi, atau geometri Cartesian, adalah studi geometri menggunakan sistem koordinat dan prinsip-prinsip aljabar dan analisis. Transformasi geometri secara luas digunakan dalam fisika dan teknik, dan merupakan dasar dari bidang yang paling modern tentang geometri, termasuk aljabar, diferensial, diskrit, dan geometri komputasi. Biasanya sistem koordinasi Cartesian diterapkan untuk memanipulasi persamaan untuk pesawat, garis lurus, dan kotak, seringkali dalam dua dan kadang-kadang

dalam tiga dimensi. Transformasi geometri dapat dijelaskan secara lebih sederhana yaitu berkaitan dengan mendefinisikan dan mewakili bentuk geometris (Katz, 1998).

2.4.1 Transformasi Translasi

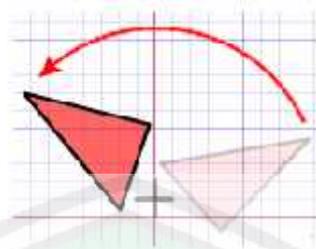
Translasi merupakan salah satu transformasi yang memindahkan objek 2D atau 3D dengan cara memindahkan setiap titik yang ada ke jarak yang sama dan arah yang sama (<http://www.mathisfun.com>, 2014). Dalam istilah lain translasi juga berarti memindah (moving) dimana objek yang ditranslasikan akan membentuk bayangan objek yang setiap titiknya sama panjang dengan objek aslinya. Gambar 2.10 merupakan contoh ilustrasi dari translasi.



Gambar 2.10 Ilustrasi transformasi geometri translasi (Sumber :<http://www.regentsprep.org/>, diakses pada tanggal 30 Mei 2015)

2.4.2 Transformasi Rotasi

Rotasi merupakan salah satu transformasi yang memindahkan objek 2D atau 3D dengan cara memutar objek tersebut (<http://www.mathisfun.com>, 2014). Objek yang dirotasi akan membentuk bayangan objek yang sama dengan bentuk yang memutar sesuai dengan besaran sudut putarnya. Gambar 2.11 merupakan contoh ilustrasi dari rotasi.



Gambar 2.11 Ilustrasi transformasi geometri rotasi (Sumber :<http://www.mathisfun.com/>, diakses pada tanggal 30 Mei 2015)

Operasi geometri yang digunakan dalam penelitian ini adalah operasi rotasi. Dalam menentukan titik (x_2, y_2) dari titik (x_1, y_1) yang diputar berdasarkan titik , maka rumus rotasi yang digunakan adalah seperti pada Rumus

2.2

$$x_2 = ((x_1 - x_0) \cos \theta) - ((y_1 - y_0) \sin \theta) + x_0 \quad (2.2)$$

$$y_2 = ((x_1 - x_0) \sin \theta) + ((y_1 - y_0) \cos \theta) + y_0$$

Keterangan Rumus 2.2:

x_1 = titik koordinat x yang pertama

x_2 = titik koordinat x yang kedua

x_0 = titik koordinat x pusat

y_1 = titik koordinat y yang pertama

y_2 = titik koordinat y yang kedua

y_0 = titik koordinat y pusat

2.5 Euclidean Distance

Euclidean distance merupakan sebuah metode untuk mengukur jarak dari dua titik (<http://mathworld.wolfram.com>, 2014). Jarak Euclidean merupakan jarak

yang umum digunakan dalam pengolahan citra, ukuran panjang diperoleh dari titik – titik yang sudah dibuat oleh pengguna. Metode ini digunakan untuk mengukur jarak antar bangunan, dengan menggunakan rumus 2.3 berikut.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.3)$$

Keterangan Rumus 2.3:

d = jarak euclidean (euclidean distance)

x_1 = titik koordinat x yang pertama

x_2 = titik koordinat x yang kedua

y_1 = titik koordinat y yang pertama

y_2 = titik koordinat y yang kedua

BAB III

PERANCANGAN DAN IMPLEMENTASI APLIKASI

3.1 Lokasi Penelitian

Lokasi dari penelitian ini terletak di Jalan Gajayana No. 50 Malang yang merupakan lokasi dari Universitas Islam Negeri Maulana Malik Ibrahim Malang. Letak geografis dari lokasi penelitian berada di 7.953785 - 7.949401 Lintang Selatan dan 112.606063 - 112.609130 Bujur Timur, seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Lokasi objek penelitian (*Sumber : Google Earth, 2013*)

3.2 Perancangan Proses

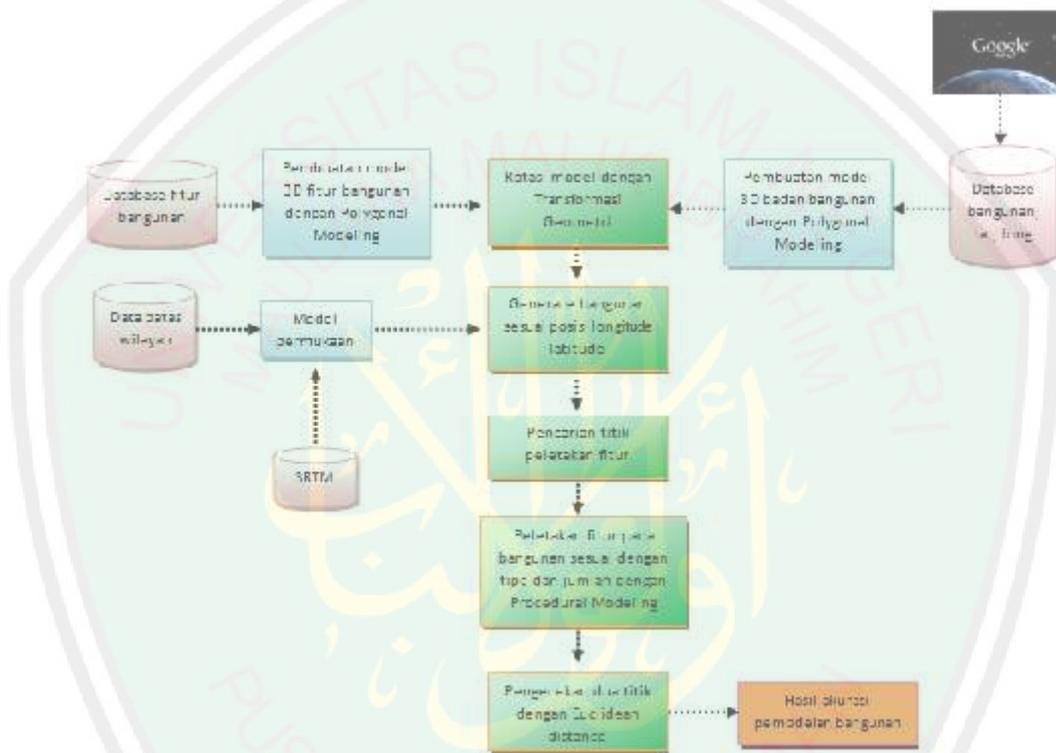
Pada penelitian ini, visualisasi 3D dikembangkan pada pemodelan bangunan yang dalam studi kasusnya mengambil tempat di kompleks lingkungan sebuah Universitas di Malang. Metode yang digunakan adalah metode procedural modeling yang diaplikasikan untuk penempatan fitur-fitur dari sebuah bangunan,

sedangkan model bangunan dan model fiturnya dibuat dengan metode polygonal modeling. Untuk mengatur arah dari bangunan digunakan metode transformasi geometri. Input data yang digunakan dalam proses pemodelan bangunan berupa citra satelit berbentuk 2D. Kemudian data batas wilayah dan letak posisi dari bangunan yang sesungguhnya diukur berdasarkan koordinat latitude dan longitude yang diambil dari Google Earth. Sedangkan untuk data ukuran fisik dari bangunan didapatkan dari Bagian Umum Universitas, sehingga pemodelan yang dihasilkan lebih valid dan sesuai bentuk aslinya.

Selanjutnya data batas wilayah digunakan dalam proses pembuatan model permukaan dari data elevasi permukaan bumi *Shuttle Radar Topography Mission* (SRTM) dengan resolusi 90 m. Kemudian, dilakukan pemodelan bangunan yang berdasarkan pada data input dimana model bangunan diletakkan sesuai dengan posisinya. Setelah itu, langkah selanjutnya adalah memodelkan fitur bangunan memilih tipe modelnya, lalu mencari titik penempatannya sesuai dengan rule yang sudah ditetapkan dan ukuran yang sudah di inputkan. Dalam proses pembuatan model sekaligus dilakukan proses rotasi untuk menentukan arah dari model menggunakan metode transformasi geometri, arah dari model 3D didapatkan dari citra satelit dari daerah yang akan divisualisasikan. Selanjutnya dilakukan tahap model integrasi antara model bangunan dengan model permukaan tanah menggunakan metode overlaying. Tahap terakhir yaitu dilakukan pengukuran antara dua titik koordinat pada model 3D bangunan menggunakan euclidean distance sehingga didapatkan hasil output pemodelan 3D bangunan yang akurat

dan detail sesuai dengan keadaan sesungguhnya dan dapat berperan sebagai bagian dari pengembangan model kota 3D.

Blok diagram dari pemodelan bangunan menggunakan metode procedural modeling dan transformasi geometri ditunjukkan pada Gambar 3.2

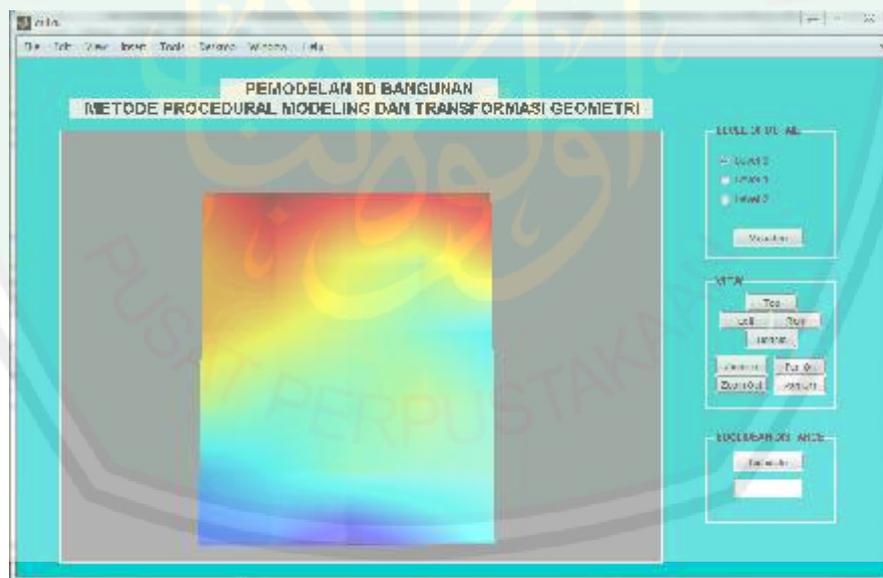


Gambar 3.2 Blok diagram pemodelan bangunan

Pada tahap pertama yaitu pengambilan input data batas wilayah dari citra daerah yang akan divisualisasikan dari Google Earth. Selanjutnya menentukan letak posisi bangunan berdasarkan dengan pengukuran koordinat longitude dan latitude, sedangkan untuk data ukuran fisik bangunan didapatkan dari database Bagian Umum Universitas. Setelah itu, dilanjutkan dengan pembuatan model permukaan dari data elevasi permukaan bumi *Shuttle Radar Topography Mission* (SRTM) dengan resolusi 90 m. Berikut merupakan penjelasan dari proses - proses yang sudah diuraikan dalam blok diagram pada Gambar 3.2

3.2.1 Pembuatan Model Permukaan Tanah

Model permukaan ini berfungsi sebagai dasar dari bangunan, fitur bangunan, jalan dan juga vegetasi yang didirikan di atasnya, pembuatan model ini mengambil data elevasi permukaan bumi *Shuttle Radar Topography Mission* (SRTM) dengan resolusi 90 m. Dari data tersebut kemudian dikalikan nilai interpolasi sebesar 1/90 sehingga menghasilkan resolusi 1 m dalam ukuran asli, sebagai catatan dalam aplikasi ini tidak menggunakan perhitungan skala melainkan nilai resolusi yang digunakan. Pada Gambar 3.3 ditunjukkan hasil dari pembuatan model permukaan tanah yang ditampilkan oleh aplikasi pemodelan 3D bangunan yang dibuat oleh peneliti.

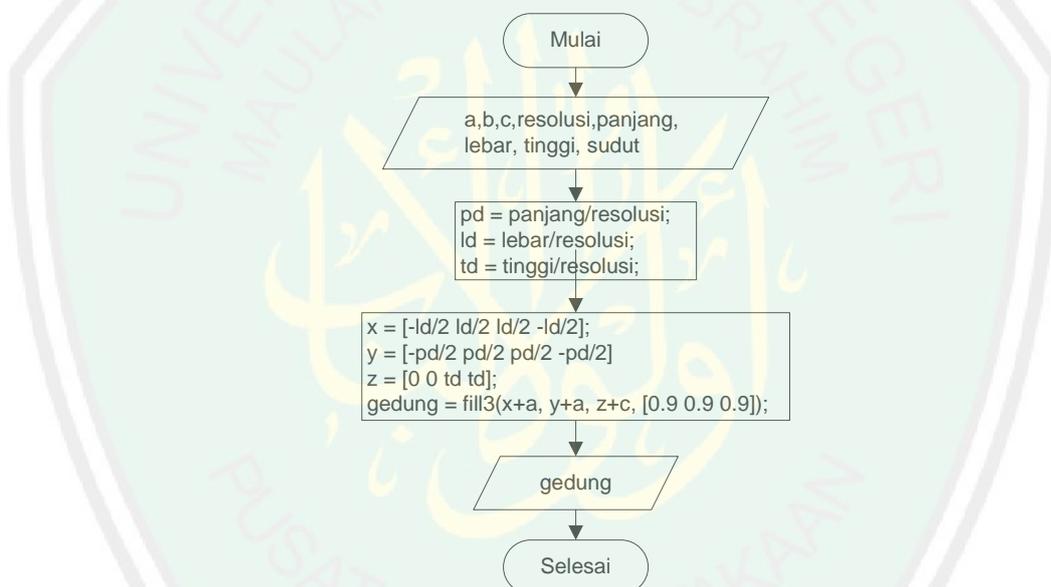


Gambar 3.3 Model Permukaan Tanah

3.2.2 Pembuatan Model 3 Dimensi

Pembuatan model bangunan dan fitur bangunan dibuat berdasarkan input data ukuran dari masing – masing bangunan dan fitur yang di ambil dari database Bagian Umum Universitas. Kemudian model bangunan dan fitur akan diletakkan

pada posisi yang sesungguhnya sesuai dengan pengukuran koordinat latitude dan longitude dari Google Earth. Sebelumnya, beberapa tipe bangunan dan fitur telah dibangun dengan menggunakan metode polygonal modeling, sehingga pada saat peletakan bangunan akan mengacu pada aturan dari tipe bangunan dan fitur yang sudah ditetapkan. Pada tahap ini juga digunakan metode transformasi geometri yang digunakan untuk menentukan arah di permukaan tanah. Berikut merupakan diagram alur metode polygonal modeling yang ditunjukkan pada Gambar 3.4



Gambar 3.4 Diagram alur metode polygonal modeling

Kode program pada pembuatan model 3D dengan metode polygonal modeling ditunjukkan pada Gambar 3.5

```

function gedung (a,b,c,resolusi,panjang,lebar,tinggi,sudut)

%poligon
ld = lebar/resolusi;
pd = panjang/resolusi;
td = tinggi/resolusi;
x = [-ld/2 ld/2 ld/2 -ld/2];
y = [-pd/2 pd/2 pd/2 -pd/2];
z = [0 0 td td];
fill3(x+a, y+b, z+c, [0.9 0.9 0.9]);
end
  
```

Gambar 3.5 Kode program metode polygonal modeling

Sedangkan untuk menentukan arahnya dengan menggunakan metode transformasi geometri rotasi. Berikut merupakan rumus perhitungan matematika dari metode transformasi geometri yang ditunjukkan pada Rumus 3.1

$$x_2 = ((x_1 - x_0) \cos \theta) - ((y_1 - y_0) \sin \theta) + x_0 \quad (3.1)$$

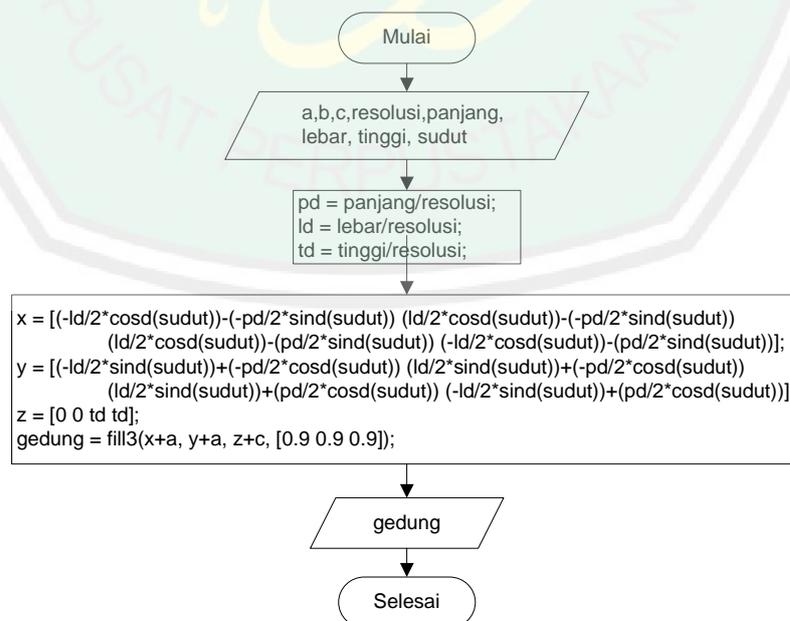
$$y_2 = ((x_1 - x_0) \sin \theta) + ((y_1 - y_0) \cos \theta) + y_0$$

Dimana (x_0, y_0) adalah koordinat titik pusat dan θ adalah sumbu putar. Sumbu putar sendiri pada umumnya memiliki arah putar searah jarum jam dengan garis horizontal, semisal untuk sumbu pusat $(0,0)$ maka perhitungan matematinya seperti yang ditunjukkan pada Rumus 3.2

$$x' = x \cos \theta - y \sin \theta \quad (3.2)$$

$$y' = x \sin \theta + y \cos \theta$$

Berikut diagram alur untuk membuat model 3D dengan menggabungkan metode polygonal modeling dan transformasi geometri yang ditunjukkan pada Gambar 3.6



Gambar 3.6 Diagram alur gabungan metode polygonal modeling dan transformasi geometri

Kode program pada pembuatan model 3D dengan metode polygonal modeling dan transformasi geometri ditunjukkan pada Gambar 3.7

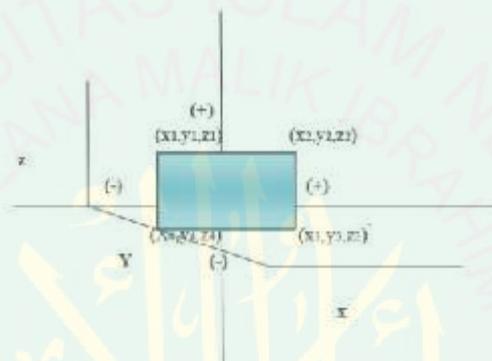
```
function gedung (a,b,c,resolusi,panjang,lebar,tinggi,sudut)
%poligon
ld = lebar/resolusi;
pd = panjang/resolusi;
td = tinggi/resolusi;
x = [(-ld/2*cosd(sudut))-(-pd/2*sind(sudut)) (ld/2*cosd(sudut))-(-pd/2*sind(sudut))
      (ld/2*cosd(sudut))-(-pd/2*sind(sudut)) (-ld/2*cosd(sudut))-(-pd/2*sind(sudut))];
y = [(-ld/2*sind(sudut))+(-pd/2*cosd(sudut)) (ld/2*sind(sudut))+(-pd/2*cosd(sudut))
      (ld/2*sind(sudut))+(-pd/2*cosd(sudut)) (-ld/2*sind(sudut))+(-pd/2*cosd(sudut))];
z = [0 0 td td];
fill3(x+a, y+b, z+c, [0.9 0.9 0.9]);
end
```

Gambar 3.7 Kode program gabungan metode poligonal modeling dan transformasi geometri

Pada Gambar 3.7 ditunjukkan pembuatan sebuah polygon yang mempunyai 4 titik, mengacu pada penelitian yang dilakukan Merrel (2011) yang menggunakan titik ketetangaan dalam pembuatan model baru, titik – titik inilah yang direpresentasikan dengan kode program pembuatan polygon tersebut. Dimana kode ini akan memastikan bahwa titik – titik yang ada pada model lama akan selalu bertetangaan dengan titik – titik yang ada pada model baru yang dikembangkan.

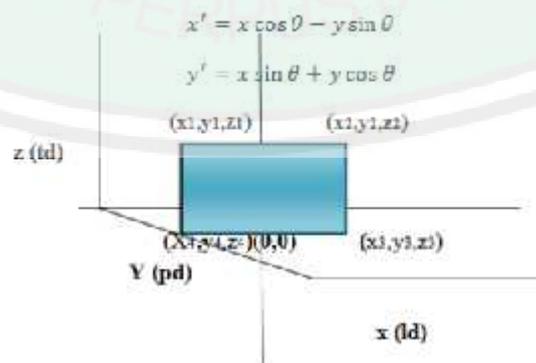
Terdapat beberapa tipe bangunan yang divisualisasikan dalam aplikasi ini, tipe - tipe bangunan yang sudah ditetapkan akan dipergunakan untuk memvisualisasikan gedung sarana perkuliahan, gedung fakultas, gedung administrasi, dan masjid. Beberapa tipe model fitur dari atap, jendela dan pintu juga divisualisasikan untuk membuat bangunan menjadi lebih detail. Pembuatan model 3D dari tipe bangunan dan fitur dibentuk dengan menggunakan kumpulan beberapa polygon yang disatukan.

Untuk membuat satu polygon membutuhkan tiga atau lebih titik yang berada pada koordinat x, y dan z. Untuk pembuatan model polygon bagian dinding yang memiliki input data latitude, longitude, panjang, lebar, tinggi, sudut dan tipe bangunan, Gambar 3.8 berikut merupakan bentuk sketsa manual pemodelan 3D yang menggunakan empat titik.



Gambar 3.8 Sketsa manual pemodelan menggunakan 4 titik

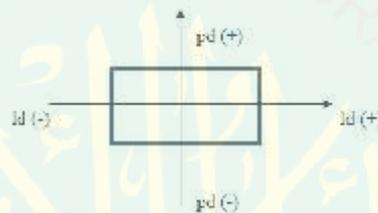
Pada Gambar 3.8, polygon yang dibuat mempunyai empat simpul yaitu simpul (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) dan (x_4, y_4, z_4) . Pada pembuatan model polygon dengan rotasi yang menggunakan metode transformasi geometri dijelaskan pada Gambar 3.9



Gambar 3.9 Sketsa analisis manual rotasi poligon

Dari Gambar 3.9 didapatkan bahwa koordinat x mewakili lebar polygon, jika titik koordinat berada dikanan garis pusat $(0,0)$ maka bernilai positif dan jika

berada dikiri garis pusat maka bernilai negative. Sedangkan koordinat y mewakili panjang polygon, jika titik koordinat berada diatas garis pusat (0,0) maka bernilai positif dan jika berada di bawah garis pusat maka bernilai negative. Koordinat z mewakili tinggi polygon, yang bernilai positif jika titik koordinat berada diatas angka 0 dan bernilai negative jika berada dibawah angka 0. Gambar 3.10 merupakan ilustrasi dari penjelasan tersebut, dimana panjang bangunan dimisalkan sebagai pd dan lebar bangunan dimisalkan sebagai ld.



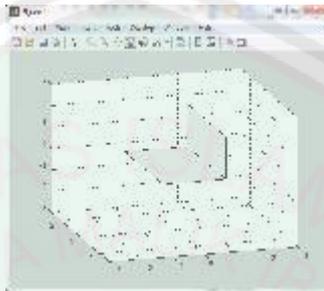
Gambar 3.10 Ilustrasi letak polygon pada titik pusat

Dari ilustrasi pada gambar Gambar 3.10, semisal dibuat sebuah alas dan dinding sebelah kanan bangunan. Daerah yang akan digunakan yaitu daerah pd(-), pd(+) ld(-) dan ld(+) untuk alas, sedangkan untuk dinding kanan menggunakan daerah pd(+), pd(-) dan ld(+). Berikut merupakan contoh penggunaannya yang ditunjukkan pada Gambar 3.11.

```
function gedung(a,b,c,resolusi,panjang,lebar,tinggi,sudut)
ld = lebar/resolusi; % lebar dinding
pd = panjang/resolusi; %panjang dinding
td = tinggi/resolusi; %tinggi dinding
%alas
x = [(-ld/2*cosd(sudut))-(-pd/2*sind(sudut)) (ld/2*cosd(sudut))-(-pd/2*sind(sudut))
      (ld/2*cosd(sudut))-(-pd/2*sind(sudut)) (-ld/2*cosd(sudut))-(-pd/2*sind(sudut))];
y = [(-ld/2*sind(sudut))+(-pd/2*cosd(sudut)) (ld/2*sind(sudut))+(-pd/2*cosd(sudut))
      (ld/2*sind(sudut))+(-pd/2*cosd(sudut)) (-ld/2*sind(sudut))+(-pd/2*cosd(sudut))];
z = [0 0 0 0];
fill3(x+a, y+b, z+c, [0.9 0.9 0.9]);
%kanan
x = [(ld/2*cosd(sudut))-(-pd/2*sind(sudut)) (ld/2*cosd(sudut))-(-pd/2*sind(sudut))
      (ld/2*cosd(sudut))-(-pd/2*sind(sudut)) (ld/2*cosd(sudut))-(-pd/2*sind(sudut))];
y = [(ld/2*sind(sudut))+(-pd/2*cosd(sudut)) (ld/2*sind(sudut))+(-pd/2*cosd(sudut))
      (ld/2*sind(sudut))+(-pd/2*cosd(sudut)) (ld/2*sind(sudut))+(-pd/2*cosd(sudut))];
z = [0 0 td td];
fill3(x+a, y+b, z+c, [0.9 0.9 0.9]);
```

Gambar 3.11 Contoh membuat bangunan dengan pd dan ld

Pada saat dijalankan diberikan input panjang 6, lebar 4, tinggi 2 dan sudut 0 maka menghasilkan output model polygon seperti yang ditunjukkan pada Gambar 3.12.



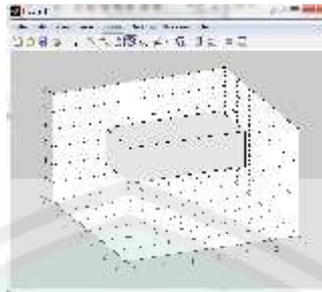
Gambar 3.12 Hasil output model polygon

Gambar 3.12 merupakan hasil dari pemodelan alas dan dinding kanan bangunan, dalam hasil tersebut terlihat bahwa separuh dari panjang dan lebar ada di daerah negative dan positif, karena peneliti menggunakan titik 0 sebagai pusat, sehingga untuk membangunnya pd dan ld dibagi 2 terlebih dahulu. Model 3D yang akan dikembangkan ada dua, yaitu model bangunan dan model fitur bangunan, dimana masing – masing mempunyai beberapa tipe yang berbeda.

3.2.2.1 Model Badan Bangunan

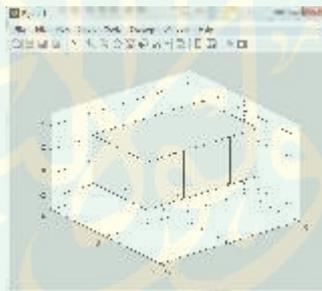
Model badan bangunan yang digunakan dalam penelitian ini dibagi menjadi lima tipe, yaitu badan bangunan tipe 1, tipe 2, tipe 3, tipe 4 dan tipe 5. Dimana setiap tipe memiliki bentuk yang berbeda – beda dan dapat diubah panjang, lebar dan tinggi sesuai dengan input di database.

Badan bangunan tipe 1 dibentuk dari sekumpulan polygon yang terdiri dari polygon alas, polygon atas, polygon dinding depan, polygon dinding belakang, polygon dinding kanan dan polygon dinding kiri. Sehingga dihasilkan tipe badan bangunan seperti Gambar 3.13.



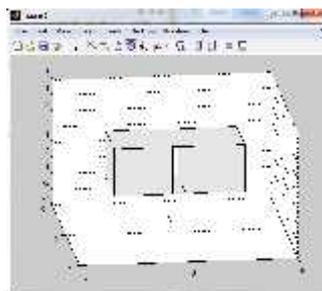
Gambar 3.13 Badan bangunan tipe 1

Badan bangunan tipe 2 dibentuk dari sekumpulan polygon yang terdiri dari polygon alas, polygon atas, polygon dinding depan, polygon dinding belakang, polygon dinding kanan, polygon dinding kiri dan memiliki lorong di bagian samping dan tengah dari bangunan. Sehingga dihasilkan tipe bangunan seperti Gambar 3.14.



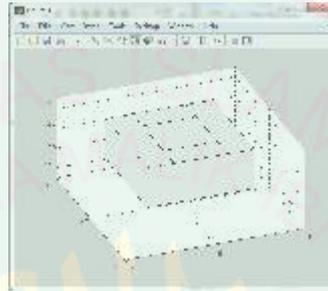
Gambar 3.14 Badan bangunan tipe 2

Bangunan tipe 3 dibentuk dari sekumpulan polygon yang terdiri dari polygon alas, polygon atas, polygon dinding depan, polygon dinding belakang, polygon dinding kanan, polygon dinding kiri dan memiliki lorong di bagian tengah dari bangunan. Sehingga dihasilkan tipe bangunan seperti Gambar 3.15.



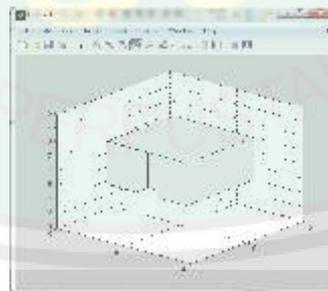
Gambar 3.15 Badan bangunan tipe 3

Badan bangunan tipe 4 dibentuk dari sekumpulan polygon yang terdiri dari polygon alas, polygon atas, polygon dinding depan, polygon dinding belakang, polygon dinding kanan, polygon dinding kiri dan memiliki ruang kosong ditengah bangunan. Sehingga dihasilkan tipe bangunan seperti Gambar 3.16.



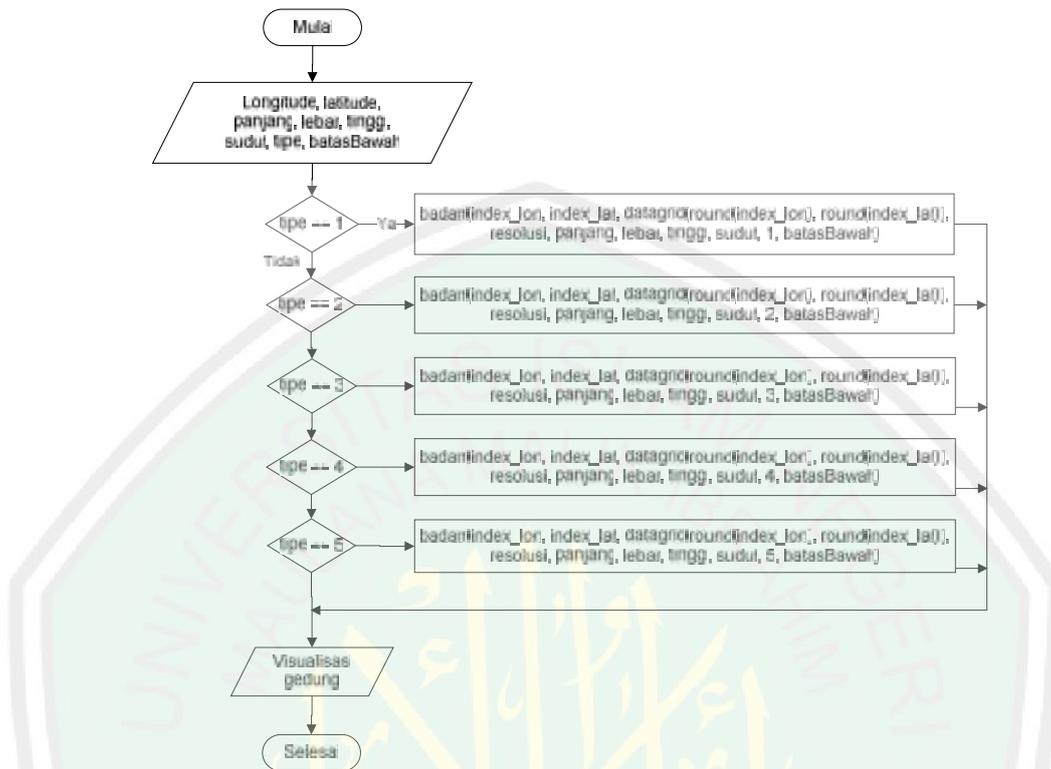
Gambar 3.16 Badan bangunan tipe 4

Bangunan tipe 5 dibentuk dari sekumpulan polygon yang terdiri dari polygon alas, polygon atas, polygon dinding depan, polygon dinding belakang, polygon dinding kanan, polygon dinding kiri dan memiliki ruang disalah satu sampingnya sebagai tempat untuk meletakkan tangga. Sehingga dihasilkan tipe bangunan seperti Gambar 3.17.



Gambar 3.17 Badan bangunan tipe 5

Kemudian tiap tipe dari model 3D badan bangunan dipanggil dengan sebuah fungsi, dimana fungsi tersebut akan menghasilkan bangunan dengan ukuran yang sesuai data input di database. Diagram alur visualisasi badan bangunan untuk memanggil tipe tipenya ditunjukkan pada Gambar 3.18.



Gambar 3.18 Diagram alur tipe bangunan

Kode program untuk tipe badan bangunan ditunjukkan pada Gambar 3.19.

```

tempat_yg_dipilih = 'D:\UinPro\Data\UIN_MALIKI\';
BADAN = xlsread([tempat_yg_dipilih 'badan.xlsx']);
for baris=1:1:size(BADAN,1)
    input_lat = BADAN(baris,1);    input_lon = BADAN(baris,2);
    id_bangunan = BADAN(baris,3);  id_badan = BADAN(baris,4);
    panjang = BADAN(baris,5);     lebar = BADAN(baris,6);
    tinggi = BADAN(baris,7);      sudut = BADAN(baris,8);
    bawah = BADAN(baris,9);
    % cari index latitude dari datalat
    error_lama = 100;
    index_lat = 1;
    for i=1:1:size(datalat,1)
        error_baru = abs(datalat(i,1)-input_lat);
        if error_baru < error_lama
            error_lama = error_baru;
            index_lat = i;
        end;
    end;
    % cari index lonfitude dari datalon
    error_lama = 100;
    index_lon = 1;
    for j=1:1:size(datalon,1)
        error_baru = abs(datalon(j,1)-input_lon);
        if error_baru < error_lama
            error_lama = error_baru;
            index_lon = j;
        end;
    end;
    badan(index_lon, index_lat, datagrid(round(index_lat),round(index_lon)),
    resolusi, panjang, lebar, tinggi, sudut, id_badan, id_bangunan, bawah);
end;
  
```

Gambar 3.19 Kode program tipe badan bangunan

Data badan bangunan yang tersimpan didalam database berisi data koordinat latitude, longitude bangunan, panjang, lebar, tinggi, sudut dan tipe model. Data koordinat dan ukuran fisik dari badan bangunan merupakan data sesungguhnya yang diambil dari Google dan Bagian Umum Universitas. Selanjutnya yaitu data sudut yang mengacu pada arah rotasi hadap dari badan bangunan saat ditempatkan di permukaan tanah. Kemudian data tipe model yang inputannya terbagi menjadi 5 tipe badan bangunan, didalam fungsi gedung terdapat percabangan untuk pemilihan tipe bangunan sehingga user hanya perlu menginputkan angka 1 sampai saja didalam database. Data batas bawah, merupakan parameter ketinggian awal badan bangunan dari permukaan tanah saat bangunan di generate.

Kemudian data badan bangunan disimpan dalam sebuah array agar data tersebut dapat digunakan kembali untuk penempatan dari fitur bangunan. Untuk menyimpan ke dalam bentuk array, ditambahkan kode program seperti pada Gambar 3.20 didalam kode program Gambar 3.19.

```

%MENYIMPAN DATA BANGUNAN
x=x+1;
arrlon(1,x) = index_lon;
arrlat(1,x) = index_lat;
arrid(1,x) = id_bangunan;
arridb(1,x) = id_badan;
arrpb(1,x) = panjang;
arrlb(1,x) = lebar;
arrtb(1,x) = tinggi;
arrsudut(1,x) = sudut;
arrlantai(1,x) = lantai;
arrbb(1,x) = bb;
save badan_all arrid arrlon arrlat arrpb arrlb arrtb arrsudut arrlantai
arrbb;

```

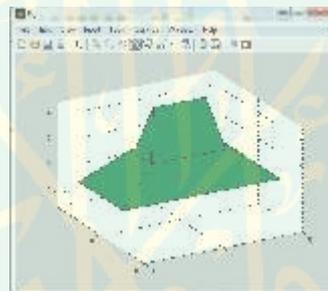
Gambar 3.20 Kode program untuk menyimpan data bangunan

Untuk mengakses data tersebut digunakan fungsi `load badan_all`, menggunakan fungsi ini akan terlihat pada kode program untuk pembuatan fitur bangunan.

3.2.2.2 Model Atap

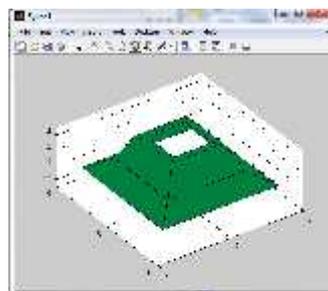
Model atap yang digunakan dalam penelitian ini dibagi menjadi 4 tipe atap, yaitu atap bangunan tipe 1, tipe 2, tipe 3 dan tipe 4. Dimana setiap tipe dapat diubah panjang, lebar dan tinggi sesuai dengan input di database.

Atap tipe 1 dibentuk dari sekumpulan polygon yang terdiri dari empat polygon bawah dan empat polygon atas, empat polygon tersebut yaitu polygon atap depan, polygon atap belakang, polygon atap kanan dan polygon atap kiri. Sehingga dihasilkan tipe atap seperti Gambar 3.21.



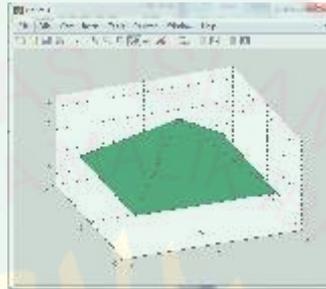
Gambar 3.21 Atap tipe 1

Atap tipe 2 dibentuk dari sekumpulan polygon yang terdiri dari empat polygon bawah, empat polygon atas dan empat polygon dalam, dimana empat polygon tersebut yaitu polygon atap depan, polygon atap belakang, polygon atap kanan dan polygon atap kiri. Tipe atap ini memiliki ruang kosong di tengah – tengahnya. Sehingga dihasilkan tipe atap seperti Gambar 3.22.



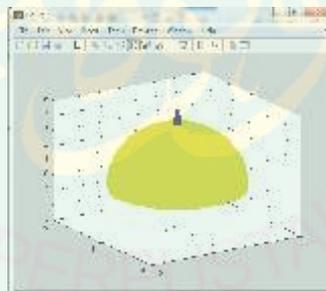
Gambar 3.22 Atap tipe 2

Atap tipe 3 dibentuk dari sekumpulan polygon yang terdiri dari empat polygon berbentuk segitiga yaitu polygon atap depan, polygon atap belakang, polygon atap kanan dan polygon atap kiri. Tipe atap ini bagian atasnya menyatu. Sehingga dihasilkan tipe atap seperti Gambar 3.23.



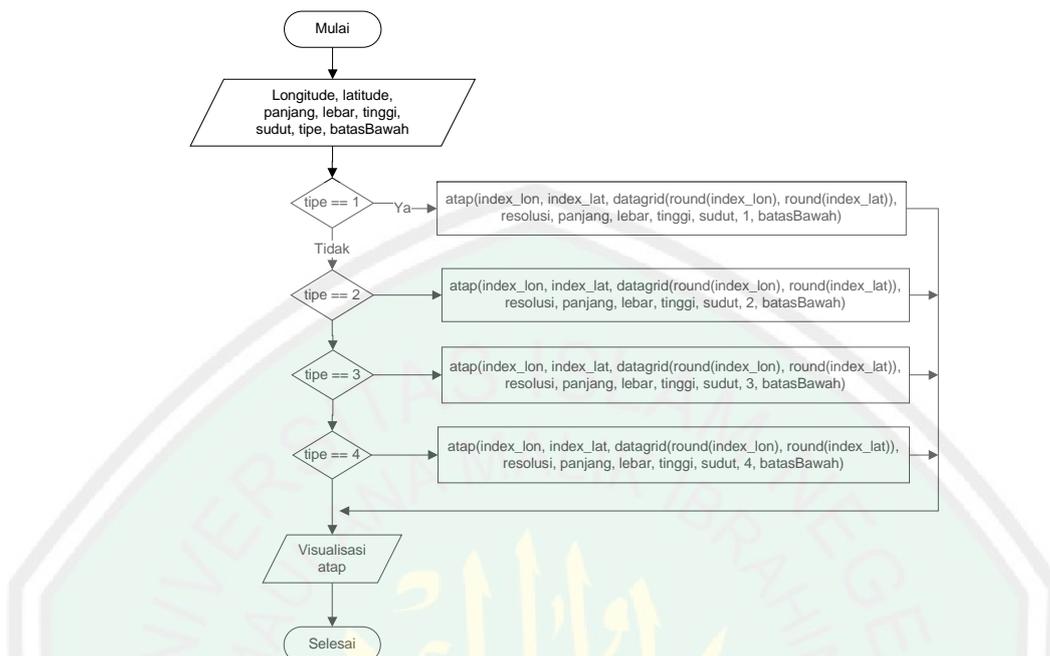
Gambar 3.23 Atap tipe 3

Tipe atap 4 dibentuk dari setengah sphere, dimana tipe ini akan digunakan sebagai kubah untuk model 3D masjid. Sehingga dihasilkan tipe atap seperti Gambar 3.24.



Gambar 3.24 Atap tipe 4

Kemudian tiap tipe dari model 3D atap bangunan dipanggil dengan sebuah fungsi, dimana fungsi tersebut akan menghasilkan atap dengan ukuran yang sesuai data input di database. Diagram alur visualisasi atap bangunan untuk memanggil tiap tipenya ditunjukkan pada Gambar 3.25.



Gambar 3.25 Diagram alur tipe atap bangunan

Kode program untuk pemodelan tipe atap bangunan ditunjukkan pada gambar 3.26

```

ATAP = xlsread([tempat_yg_dipilih 'atap.xlsx']);
for baris=1:1:size(ATAP,1)
    load badan_all
    id_bangunan_atap = ATAP(baris,1);
    id_atap = ATAP(baris,2);
    panjangA = ATAP(baris,3);
    lebarA = ATAP(baris,4);
    tinggiA = ATAP(baris,5);
    sudutA = ATAP(baris,6);
    geserlat = ATAP(baris,7);
    geserlon = ATAP(baris,8);
    bawah = ATAP(baris,9);
    for k=1:1:length(arrid)
        if arrid(1,k) == id_bangunan_atap
            atap(arrlon(1,k)+(geserlon/resolusi), arrlat(1,k)+
                (geserlat/resolusi), datagrid(round(arrlat(1,k)),
                round(arrlon(1,k))), resolusi, panjangA, lebarA, tinggiA, sudutA,
                id_atap, arrbb(1,k) + bawah);
        end
    end
end
end
  
```

Gambar 3.26 Kode program tipe atap bangunan

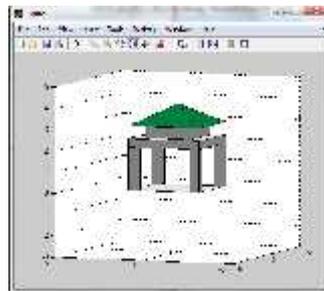
Data fitur atap yang tersimpan berisi data panjang, lebar, tinggi, sudut dan tipe model atap. Sedangkan koordinat longitude dan latitudenya didapat dari data badan bangunan yang sudah tersimpan didalam array dan dipanggil dengan fungsi load badan_all. Selanjutnya yaitu data sudut yang mengacu pada arah rotasi

hadap dari atap saat ditempatkan di badan bangunan. Kemudian data tipe model yang inputannya terbagi menjadi 4 tipe badan bangunan, didalam fungsi atap terdapat percabangan untuk pemilihan tipe atap sehingga user hanya perlu menginputkan angka 1 sampai saja didalam database. Data batas bawah, merupakan parameter ketinggian dari atap sehingga saat degenerate biasanya mempergunakan data tinggi gedung.

3.2.2.3 Model Kubah

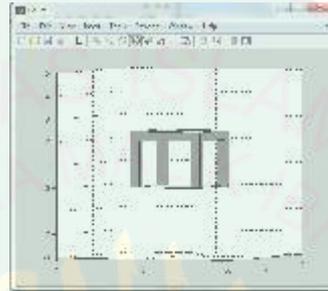
Model kubah yang digunakan dalam penelitian ini dibagi menjadi 2 tipe kubah, yaitu kubah bangunan tipe 1 dan tipe 2. Dimana setiap tipe memiliki bentuk yang berbeda – beda dan dapat diubah panjang, lebar dan tinggi sesuai dengan input di database.

Kubah tipe 1 dibentuk dari sekumpulan polygon yang membentuk tiang dan atap, dimana sebuah tiang terdiri dari polygon tiang depan, polygon tiang belakang, polygon tiang kanan dan polygon tiang kiri. Sedangkan untuk atapnya terdiri dari empat polygon berbentuk segitiga yaitu polygon atap depan, polygon atap belakang, polygon atap kanan dan polygon atap kiri. Sehingga dihasilkan tipe kubah seperti Gambar 3.27.



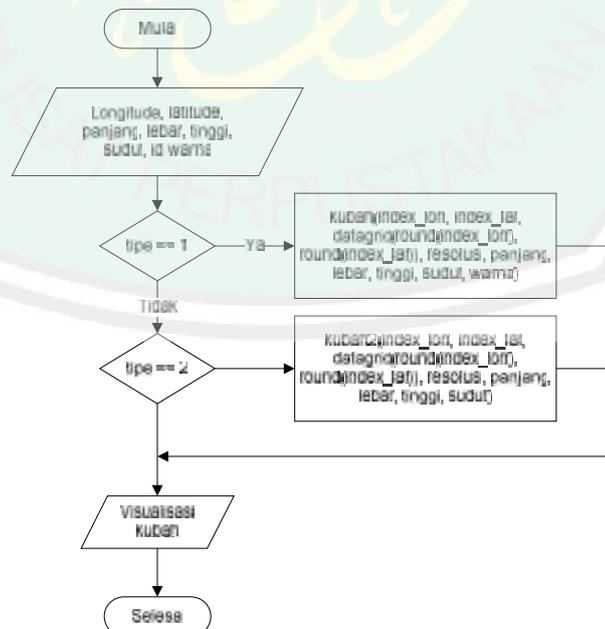
Gambar 3.27 Kubah tipe 1

Kubah tipe 2 dibentuk dari sekumpulan polygon yang terdiri dari polygon tiang depan, polygon tiang belakang, polygon tiang kanan dan polygon tiang kiri. Kubah tipe ini tidak memiliki atap di atasnya. Sehingga dihasilkan tipe kubah seperti Gambar 3.28.



Gambar 3.28 Kubah tipe 2

Kemudian tiap tipe dari model 3D kubah bangunan dipanggil dengan sebuah fungsi, dimana fungsi tersebut akan menghasilkan atap dengan ukuran yang sesuai data input di database. Diagram alur visualisasi kubah bangunan untuk memanggil tiap tipenya ditunjukkan pada Gambar 3.29.



Gambar 3.29 Diagram alur tipe kubah bangunan

Kode program untuk pemodelan tipe kubah bangunan ditunjukkan pada gambar

3.30

```

KUBAH = xlsread([tempat_yg_dipilih 'kubah.xlsx']);
for baris=1:1:size(KUBAH,1)
    load badan_all
    id_bangunan_kubah = KUBAH(baris,1);
    id_kubah = KUBAH(baris,2);
    panjang = KUBAH(baris,3);
    lebar = KUBAH(baris,4);
    tinggi = KUBAH(baris,5);
    sudut = KUBAH(baris,6);
    warna = KUBAH(baris,7);
    letak = KUBAH(baris,8);
    geserlat = KUBAH(baris,9);
    geserlon = KUBAH(baris,10);
    for k=1:1:length(arrid)
        if arrid(1,k) == id_bangunan_kubah
            kubah(arrlon(1,k)+(geserlon/resolusi),
                arrlat(1,k)+(((arrlb(1,k)/resolusi)/2)+(geserlat/resolusi)),
                datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi, panjang,
                lebar, tinggi, sudut, id_kubah, warna);
        end
    end
end
end

```

Gambar 3.30 Kode program tipe kubah bangunan

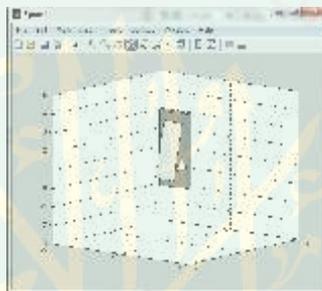
Data fitur kubah yang tersimpan berisi data panjang, lebar, tinggi, sudut dan warna. Sedangkan koordinat longitude dan latitudenya didapat dari data badan bangunan yang sudah tersimpan didalam array dan dipanggil dengan fungsi load badan_all.. Selanjutnya yaitu data sudut yang mengacu pada arah rotasi hadap dari kubah saat ditempatkan di badan bangunan. Kemudian data tipe model yang inputannya terbagi menjadi 2 tipe badan bangunan, didalam fungsi kubah terdapat percabangan untuk pemilihan tipe kubah sehingga user hanya perlu menginputkan tipe kedalam database. Data warna merupakan parameter untuk mengubah warna pada kubah yang memiliki atap, dimana parameter tersebut dapat diisi dengan nilai 1 dan 2.

3.2.2.4 Model Jendela

Model jendela yang digunakan dalam penelitian ini dibagi menjadi 4 tipe jendela, yaitu jendela bangunan tipe 1, tipe 2, tipe 3 dan tipe 4. Dimana setiap tipe

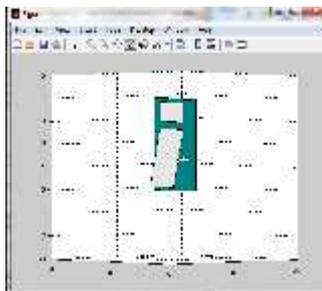
memiliki bentuk yang berbeda – beda dan dapat diubah panjang, lebar dan tinggi sesuai dengan input di database.

Jendela tipe 1 dibentuk dari sekumpulan polygon yang membentuk badan jendela, kaca dan penyangga kaca, dimana sebuah badan jendela terdiri dari polygon depan, polygon belakang, polygon kanan dan polygon kiri. Sedangkan untuk kaca dan penyangga masing – masing dibentuk dengan satu polygon. Sehingga dihasilkan tipe jendela seperti Gambar 3.31.



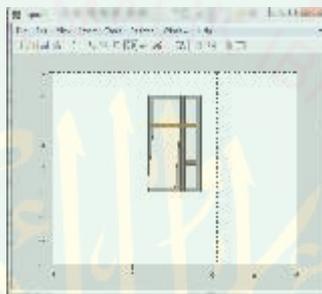
Gambar 3.31 Jendela tipe 1

Jendela tipe 2 dibentuk dari sekumpulan polygon yang membentuk badan jendela, dua kaca (atas dan bawah) dan penyangga kaca, dimana sebuah badan jendela terdiri dari polygon depan, polygon belakang, polygon kanan dan polygon kiri. Sedangkan untuk kaca atas, kaca bawa dan penyangga masing – masing dibentuk dengan satu polygon. Sehingga dihasilkan tipe jendela seperti Gambar 3.32.



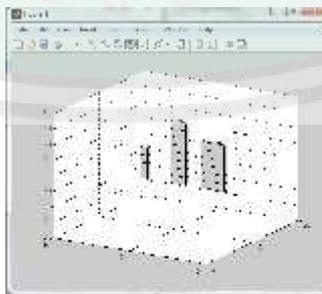
Gambar 3.32 Jendela tipe 2

Jendela tipe 3 dibentuk dari sekumpulan polygon yang membentuk badan jendela, tiga kaca (atas, bawah, samping), pembatas kaca dan penyangga kaca, dimana sebuah badan jendela terdiri dari polygon depan, polygon belakang, polygon kanan dan polygon kiri. Sedangkan untuk kaca atas, kaca bawah, kaca samping, pembatas kaca dan penyangga masing – masing dibentuk dengan satu polygon. Sehingga dihasilkan tipe jendela seperti Gambar 3.33.



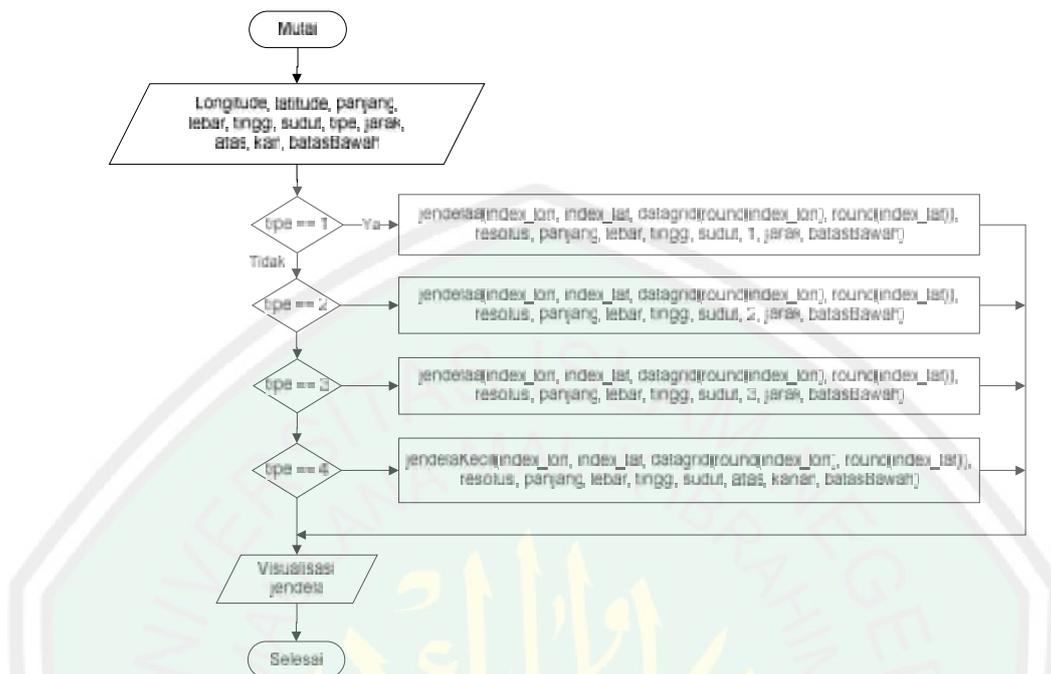
Gambar 3.33 Jendela tipe 3

Jendela tipe 4 dibentuk dari sekumpulan polygon yang terdiri dari polygon depan, polygon belakang, polygon kanan dan polygon kiri. Tipe jendela ini dapat dibangun dengan menginputkan jumlah jendela kesamping dan keatas, sehingga mempermudah dalam pemodelan jendela ukuran kecil yang berjumlah banyak. Sehingga dihasilkan tipe jendela seperti Gambar 3.34.



Gambar 3.34 Jendela tipe 4

Diagram alur visualisasi jendela bangunan untuk memanggil tiap tipenya ditunjukkan pada Gambar 3.35.



Gambar 3.35 Diagram alur tipe jendela bangunan

Kode program untuk pemodelan tipe jendela bangunan ditunjukkan pada gambar 3.36

```

JENDELA = xlsread([tempat_yg_dipilih 'jendela.xlsx']);
for baris=1:1:size(JENDELA,1)
load badan_all
id_bangunan_jendela = JENDELA(baris,1);      id_jendela = JENDELA(baris,2);
panjang = JENDELA(baris,3);                  lebar = JENDELA(baris,4);
tinggi = JENDELA(baris,5);                   atas = JENDELA(baris,6);
kanan = JENDELA(baris,7);                   sudut = JENDELA(baris,8);
lantai = JENDELA(baris,9);                  jumlah = JENDELA(baris,10);
letak = JENDELA(baris,11);                  awal = JENDELA(baris,12);
jarak1 = JENDELA(baris,13);                 jarak2 = JENDELA(baris,14);
modulo = JENDELA(baris,15);                 bawah1 = JENDELA(baris,16);
bawah2 = JENDELA(baris,17);                 spasi = JENDELA(baris,18);
tengah1 = JENDELA(baris,19);                tengah2 = JENDELA(baris,20);
jarak = JENDELA(baris,21);

for k=1:length(arrid)
if arrid(1,k) == id_bangunan_jendela
jendelaa(arrlon(1,k)-((arrpb(1,k)/resolusi)/2) +
((awal+jarak1)/resolusi), arrlat(1,k)+((arrlb(1,k)/resolusi)/2),
datagrid(round(arrlat(1,k)),round(arrlon(1,k))), resolusi,
panjang, lebar, tinggi, sudut, id_jendela, atas, kanan,bawah1);
end
end
end
end
  
```

Gambar 3.36 Kode program tipe jendela bangunan

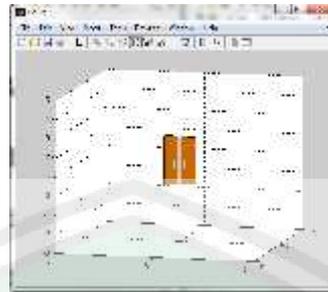
Data fitur jendela yang tersimpan berisi data panjang, lebar, tinggi, sudut, tipe, jarak dan batas bawah. Sedangkan koordinat longitude dan latitudenya

didapat dari data badan bangunan yang sudah tersimpan didalam array dan dipanggil dengan fungsi `load badan_all`. Selanjutnya yaitu data sudut yang mengacu pada arah rotasi hadap dari jendela saat ditempatkan di badan bangunan. Kemudian data tipe model yang inputannya terbagi menjadi 4 tipe badan bangunan, didalam fungsi jendela terdapat percabangan untuk pemilihan tipe kubah sehingga user hanya perlu menginputkan tipe kedalam database. Untuk jendela tipe 4 menggunakan parameter jumlah jendela atas dan kanan untuk membuat model jendela kecil. Data batas bawah merupakan parameter untuk menempatkan posisi ketinggian dari jendela, karena tiap lantai disebuah gedung memiliki jendela maka digunakanlah parameter ini untuk membantu dalam proses penempatan jendela.

3.2.2.5 Model Pintu

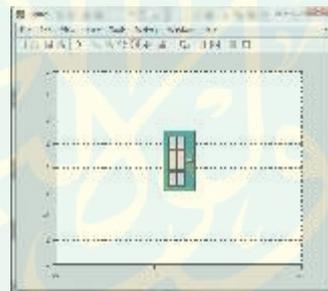
Model pintu yang digunakan dalam penelitian ini dibagi menjadi 2 tipe pintu, yaitu pintu bangunan tipe 1 dan tipe 2. Dimana setiap tipe memiliki bentuk yang berbeda – beda dan dapat diubah panjang, lebar dan tinggi sesuai dengan input di database.

Pintu tipe 1 dibentuk dari sekumpulan polygon yang membentuk badan pintu, pembatas dan dua gagang pintu, dimana sebuah badan pintu terdiri dari polygon depan, polygon belakang, polygon kanan dan polygon kiri. Sedangkan untuk pembatas dan gagang pintu masing – masing dibentuk dengan satu polygon. Sehingga dihasilkan tipe jendela seperti Gambar 3.37.



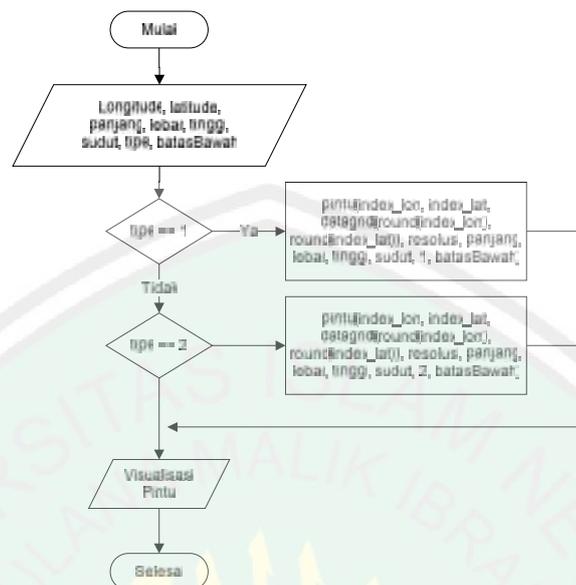
Gambar 3.37 Pintu tipe 1

Pintu tipe 2 dibentuk dari sekumpulan polygon yang membentuk badan pintu, tiga pembatas dan satu gagang pintu, dimana sebuah badan pintu terdiri dari polygon depan, polygon belakang, polygon kanan dan polygon kiri. Sedangkan untuk pembatas dan gagang pintu masing – masing dibentuk dengan satu polygon. Sehingga dihasilkan tipe jendela seperti Gambar 3.38



Gambar 3.38 Pintu tipe 2

Kemudian tiap tipe dari model 3 pintu bangunan dipanggil dengan sebuah fungsi, dimana fungsi tersebut akan menghasilkan atap dengan ukuran yang sesuai data input di database. Diagram alur visualisasi kubah bangunan untuk memanggil tiap tipenya ditunjukkan pada Gambar 3.39.



Gambar 3.39 Diagram alur tipe pintu bangunan

Kode program untuk pemodelan tipe pintu bangunan ditunjukkan pada gambar

3.40

```

PINTU = xlsread([tempat_yg_dipilih 'pintu.xlsx']);
for baris=1:1:size(PINTU,1)
    load badan_all
    id_bangunan_pintu = PINTU(baris,1);
    panjang = PINTU(baris,3);
    tinggi = PINTU(baris,5);
    lantai = PINTU(baris,7);
    letak = PINTU(baris,9);
    jarak1 = PINTU(baris,11);
    bawah2 = PINTU(baris,13);
    tengah1 = PINTU(baris,15);
    jarak = PINTU(baris,17);
    id_pintu = PINTU(baris,2);
    lebar = PINTU(baris,4);
    sudut = PINTU(baris,6);
    jumlah = PINTU(baris,8);
    awal = PINTU(baris,10);
    bawah1 = PINTU(baris,12);
    spasi = PINTU(baris,14);
    tengah2 = PINTU(baris,16);

    for k=1:1:length(arrid)
        if arrid(1,k) == id_bangunan_pintu
            pintu(arrlon(1,k)-((arrpb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                arrlat(1,k)+((arrlb(1,k)/resolusi)/2), datagrid(round(arrlat(1,k)),
                    round(arrlon(1,k))), resolusi, panjang, lebar, tinggi, sudut,id_pintu,
                    bawah1);
        end
    end
end
end
  
```

Gambar 3.40 Kode program tipe pintu bangunan

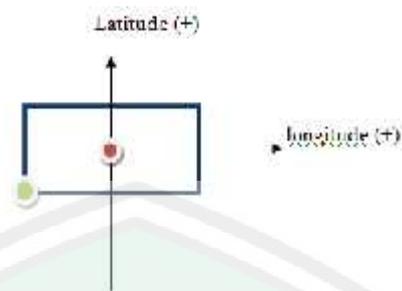
Data fitur pintu yang tersimpan berisi data panjang, lebar, tinggi, sudut, tipe dan batas bawah. Sedangkan koordinat longitude dan latitudenya didapat dari data badan bangunan yang sudah tersimpan didalam array dan dipanggil dengan fungsi load badan_all.. Selanjutnya yaitu data sudut yang mengacu pada arah

rotasi hadap dari pintu saat ditempatkan di badan bangunan. Kemudian data tipe model yang inputannya terbagi menjadi 2 tipe pintu bangunan, didalam fungsi pintu terdapat percabangan untuk pemilihan tipe pintu sehingga user hanya perlu menginputkan tipe kedalam database. Data batas bawah merupakan parameter untuk menempatkan posisi ketinggian dari pintu, karena terdapat beberapa lantai yang memiliki pintu maka digunakanlah parameter ini untuk membantu dalam proses penempatan pintu.

3.2.3 Peletakan Fitur Bangunan dengan Procedural Modeling

Pada tahap peletakan fitur bangunan, parameter yang digunakan adalah posisi longitude dan latitude dari badan bangunan kemudian beserta besarnya nilai pergeseran longitude dan latitude posisi fitur dari titik pusat badan bangunan. Fitur bangunan yang dibuat berupa fitur eksterior, maka penempatannya dapat menggunakan parameter panjang dan lebar bangunan untuk mempermudah penentuan titik fitur.

Ada dua pergeseran nilai yaitu nilai longitude dan latitude, sedangkan posisi panjang dan lebar badan bangunan tidak selalu pada garis longitude atau latitude, sehingga posisi badan bangunan pada garis longitude dan latitude dipetakan menurut sudut hadap dari bangunan itu sendiri. Pada Gambar 3.41 ditunjukkan posisi badan bangunan yang menghadap sudut 90 derajat.



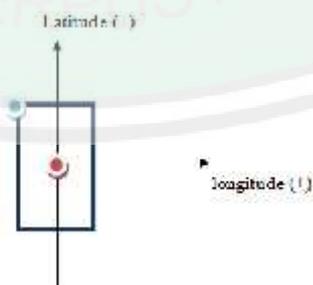
Gambar 3.41 Ilustrasi garis longitude dan latitude pada badan bangunan sudut 90

Pada Gambar 3.41 didapatkan informasi bahwa panjang badan bangunan berada di garis longitude dan lebar bangunan berada di garis latitude sedangkan garis longitude dan latitude sendiri mempunyai area positif dan negatif. Sehingga untuk mendapatkan posisi titik hijau yaitu dengan cara mengurangi longitude dengan setengah dari nilai panjang dan mengurangi latitude dengan setengah dari nilai lebar. Kode program untuk mendapatkan posisi longitude dan latitude titik hijau ditunjukkan pada Gambar 3.42.

```
index_lon = index_lon - ((panjang/resolusi)/2);
index_lat = index_lat - ((lebar/resolusi)/2);
```

Gambar 3.42 Kode program mendapatkan titik longitude dan latitude baru

Selanjutnya pada Gambar 3.43 ditunjukkan posisi badan bangunan yang diputar ke arah sudut 0 derajat.



Gambar 3.43 Ilustrasi garis longitude dan latitude pada badan bangunan sudut 0

Pada Gambar 3.43 didapatkan informasi bahwa panjang bangunan berada di garis latitude dan lebar bangunan berada di garis longitude. Sehingga untuk mendapatkan posisi titik biru yaitu dengan cara mengurangi longitude dengan

setengah dari nilai lebar dan menambah latitude dengan setengah dari nilai panjang. Kode program untuk mendapatkan posisi longitude dan latitude titik biru ditunjukkan pada Gambar 3.44.

```
index_lon = index_lon - ((lebar/resolusi)/2);
index_lat = index_lat + ((panjang/resolusi)/2);
```

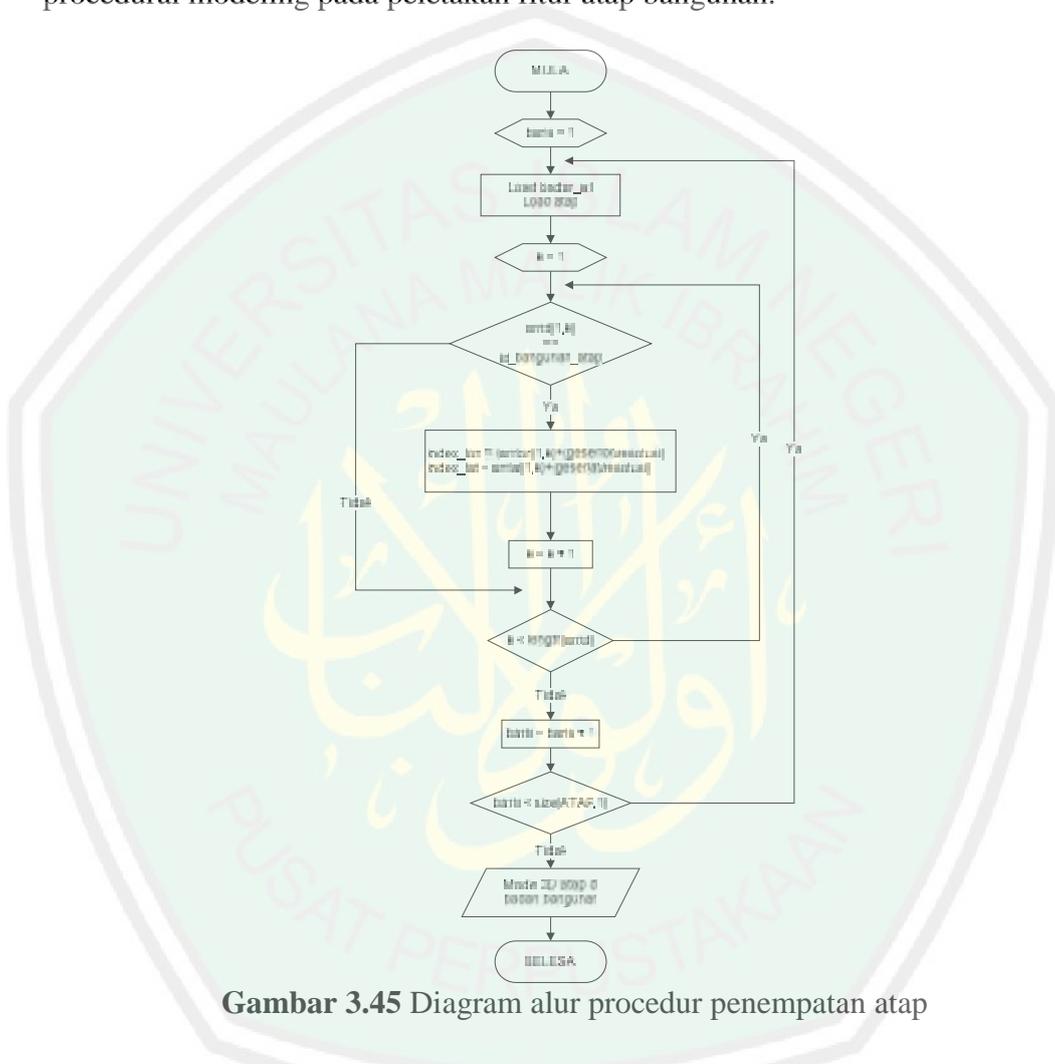
Gambar 3.44 Kode program mendapatkan titik longitude dan latitude baru

Dari penjelasan tersebut didapat sebuah prosedur untuk membuat titik baru dengan mengacu pada sudut hadap badan bangunan yang mempengaruhi letak panjang dan lebar pada garis longitude dan latitude. Kemudian untuk menentukan operasi penambahan atau pengurangan digunakan posisi fitur terhadap arah mata angin yaitu utara, timur, selatan dan barat. Dimana arah utara diwakilkan dengan nilai 1, arah timur diwakilkan dengan nilai 2, arah selatan diwakilkan dengan nilai 3 dan arah barat diwakilkan dengan nilai 4.

3.2.3.1 Procedural Modeling pada Atap

Procedural Modeling merupakan istilah umum untuk sejumlah teknik computer grafis untuk membuat model 3D dan berfokus pada menciptakan model dari satu set aturan (Ganster, 2009). Dari set aturan inilah akan didapat model baru dengan bentuk dan ukuran model yang berbeda beserta tempat penempatan model tersebut. Set aturan dalam procedural modeling untuk penetapan titik longitude dan latitude pada atap menggunakan parameter geserlon dan geserlat yang di inputkan ke dalam database. Parameter geserlon ditambahkan pada garis longitude, sedangkan parameter geserlat ditambahkan pada garis latitude. Garis longitude dan latitude mempunyai area positif dan negatif, pada saat

menempatkan fitur atap di area negatif dapat menginputkan nilai minus kedalam data geserlon dan geserlat. Pada Gambar 3.45 ditunjukkan diagram alur proses procedural modeling pada peletakan fitur atap bangunan.



Gambar 3.45 Diagram alur procedur penempatan atap

Kode program prosedur penempatan atap ditunjukkan pada Gambar 3.46

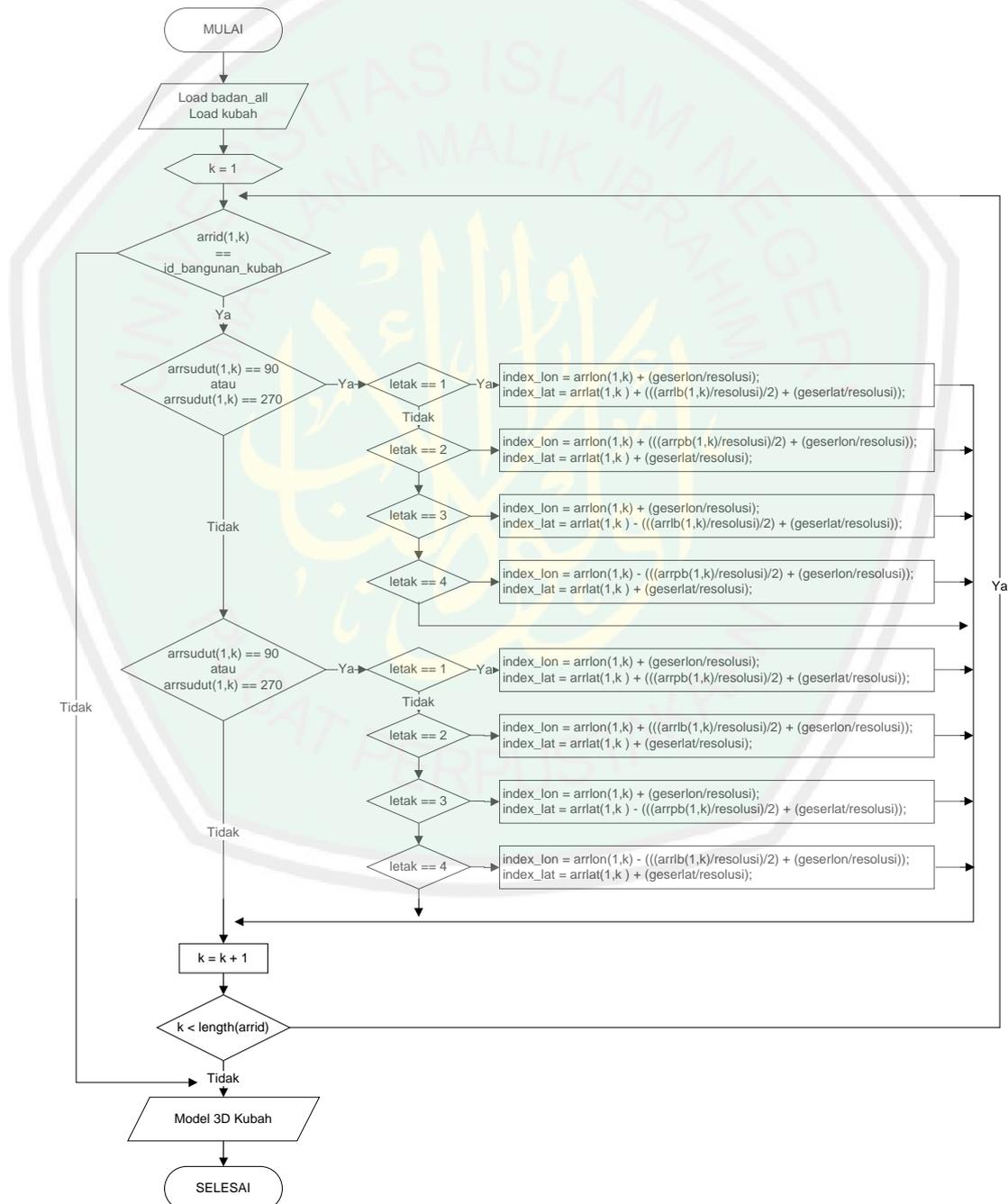
```

ATAP = xlsread([tempat_yg_dipilih 'atap.xlsx']);
for baris=1:1:size(ATAP,1)
    load badan_all
    id_bangunan_atap = ATAP(baris,1);    id_atap = ATAP(baris,2);
    panjangA = ATAP(baris,3);           lebarA = ATAP(baris,4);
    tinggiA = ATAP(baris,5);            sudutA = ATAP(baris,6);
    geserlat = ATAP(baris,7);           geserlon = ATAP(baris,8);
    bawah = ATAP(baris,9);
    for k=1:1:length(arrid)
        if arrid(1,k) == id_bangunan_atap
            atap(arrlon(1,k)+(geserlon/resolusi), arrlat(1,k)+(geserlat/resolusi),
            datagrid(round(arrlat(1,k)),round(arrlon(1,k))), resolusi, panjangA,
            lebarA, tinggiA, sudutA, id_atap, arrbb(1,k) + bawah);
        end
    end
end
end
  
```

Gambar 3.46 Kode program procedur penempatan atap

3.2.3.2 Procedural Modeling pada Kubah

Prosedur untuk penetapan titik longitude dan latitude kubah menggunakan parameter panjang, lebar, geserlon dan geserlat. Pada Gambar 3.47 ditunjukkan diagram alur prosedur penempatan kubah bangunan.



Gambar 3.47 Diagram alur prosedur penempatan kubah

Kode program prosedur penempatan kubah bangunan ditunjukkan pada Gambar

3.48

```

KUBAH = xlsread([tempat_yg_dipilih 'kubah.xlsx']);
for baris=1:1:size(KUBAH,1)
    load badan_all
    id_bangunan_kubah = KUBAH(baris,1);    id_kubah = KUBAH(baris,2);
    panjang = KUBAH(baris,3);              lebar = KUBAH(baris,4);
    tinggi = KUBAH(baris,5);               sudut = KUBAH(baris,6);
    warna = KUBAH(baris,7);               letak = KUBAH(baris,8);
    geserlat = KUBAH(baris,9);            geserlon = KUBAH(baris,10);
    for k=1:1:length(arrid)
        if arrid(1,k) == id_bangunan_kubah
            if arrsudut(1,k) == 90 || arrsudut(1,k) == 270
                if letak == 1
                    kubah(arrlon(1,k)+(geserlon/resolusi), arrlat(1,k)+
                        (((arrlb(1,k)/resolusi)/2)+(geserlat/resolusi)),
                        datagrid(round(arrlat(1,k)),round(arrlon(1,k))), resolusi,
                        panjang,lebar,tinggi,sudut,id_kubah,warna);
                elseif letak == 2
                    kubah(arrlon(1,k)+(((arrpb(1,k)/resolusi)/2)+
                        (geserlon/resolusi)),arrrlat(1,k)+(geserlat/resolusi),
                        datagrid(round(arrrlat(1,k)), round(arrlon(1,k))),resolusi,
                        panjang,lebar,tinggi,sudut,id_kubah, warna);
                elseif letak == 3
                    kubah(arrlon(1,k)+(geserlon/resolusi),arrrlat(1,k)-
                        (((arrpb(1,k)/resolusi)/2)+(geserlat/resolusi)),
                        datagrid(round(arrrlat(1,k)),round(arrlon(1,k))),resolusi,
                        panjang,lebar,tinggi,sudut,id_kubah,warna);
                elseif letak == 4
                    kubah(arrlon(1,k)-(((arrpb(1,k)/resolusi)/2)+
                        (geserlon/resolusi)),arrrlat(1,k)+(geserlat/resolusi),
                        datagrid(round(arrrlat(1,k)),round(arrlon(1,k))), resolusi,
                        panjang,lebar,tinggi,sudut,id_kubah,warna);
                end
            elseif arrsudut(1,k) == 0 || arrsudut(1,k) == 180
                if letak == 1
                    kubah(arrlon(1,k)+(geserlon/resolusi), arrrlat(1,k)+
                        (((arrpb(1,k)/resolusi)/2)+(geserlat/resolusi)),
                        datagrid(round(arrrlat(1,k)),round(arrlon(1,k))),resolusi,
                        panjang,lebar,tinggi,sudut,id_kubah,warna);
                elseif letak == 2
                    kubah(arrlon(1,k)+(((arrlb(1,k)/resolusi)/2)+
                        (geserlon/resolusi)),arrrlat(1,k)+(geserlat/resolusi),
                        datagrid(round(arrrlat(1,k)),round(arrlon(1,k))),resolusi,
                        panjang,lebar,tinggi,sudut,id_kubah,warna);
                elseif letak == 3
                    kubah(arrlon(1,k)+(geserlon/resolusi),arrrlat(1,k)-
                        (((arrpb(1,k)/resolusi)/2)+(geserlat/resolusi)),
                        datagrid(round(arrrlat(1,k)),round(arrlon(1,k))),resolusi,
                        panjang,lebar,tinggi,sudut,id_kubah,warna);
                elseif letak == 4
                    kubah(arrlon(1,k)-(((arrlb(1,k)/resolusi)/2)+
                        (geserlon/resolusi)),arrrlat(1,k)+(geserlat/resolusi),
                        datagrid(round(arrrlat(1,k)),round(arrlon(1,k))),resolusi,
                        panjang,lebar,tinggi,sudut,id_kubah,warna);
                end
            end
        end
    end
end
end
end
end

```

Gambar 3.48 Kode program procedur penempatan kubah

Prosedur penempatan kubah dibedakan menurut sudut hadap dari badan bangunan, seperti telah dijelaskan sebelumnya bahwa garis longitude dan latitude akan berbeda penempatannya jika sudut badan juga berbeda. Sudut hadap dibedakan menjadi dua yaitu sudut 90 atau 270 derajat dan sudut 0 atau 180 derajat, kemudian didalam setiap sudut terbagi menjadi 4 titik letak yaitu utara diwakili nilai 1, timur diwakili nilai 2, selatan diwakili nilai 3 dan barat diwakili nilai 4. Default dari penempatan kubah ada dibagian tengah bangunan, namun diberikan parameter tambahan berupa geserlon dan geserlat yang dapat merubah letak dari kubah.

3.2.3.3 Procedural Modeling pada Jendela

Procedur untuk penetapan titik longitude dan latitude jendela menggunakan parameter panjang, lebar, jarak pinggir jendela, jarak tengah jendela, jarak antar tiang bangunan, dan nilai tengah. Parameter selengkapnya yang tersimpan didalam database dipanggil dengan kode program seperti pada Gambar 3.49

```
JENDELA = xlsread([tempat_yg_dipilih 'jendela.xlsx']);
for baris=1:1:size(JENDELA,1)
    load badan_all
    id_bangunan_jendela = JENDELA(baris,1);    id_jendela = JENDELA(baris,2);
    panjang = JENDELA(baris,3);                lebar = JENDELA(baris,4);
    tinggi = JENDELA(baris,5);                 atas = JENDELA(baris,6);
    kanan = JENDELA(baris,7);                  sudut = JENDELA(baris,8);
    lantai = JENDELA(baris,9);                 jumlah = JENDELA(baris,10);
    letak = JENDELA(baris,11);                 awal = JENDELA(baris,12);
    jarak1 = JENDELA(baris,13);                jarak2 = JENDELA(baris,14);
    modulo = JENDELA(baris,15);                bawah1 = JENDELA(baris,16);
    bawah2 = JENDELA(baris,17);                spasi = JENDELA(baris,18);
    tengah1 = JENDELA(baris,19);               tengah2 = JENDELA(baris,20);
    jarak = JENDELA(baris,21);

    %VARIABEL PENYIMPAN NILAI LAMA
    awal_lama = awal;
    jarak_lama = jarak1;
    jumlah = jumlah + (jumlah/(modulo));
    tengah1 = tengah1 + (tengah1/(modulo));
    tengah2 = tengah2 + (tengah2/modulo);
end
```

Gambar 3.49 Kode program untuk memanggil database jendela

Data jendela yang dipanggil berupa ukuran jendela yaitu panjang, lebar, tinggi dan sudut hadap jendela. Parameter lantai menunjukkan jumlah lantai dari bangunan dan parameter jumlah menunjukkan jumlah jendela pada satu lantai. Kemudian ada parameter kanan dan atas untuk mendeklarasikan banyaknya jumlah keatas dan kekanan dari jendela tipe 4. Selanjutnya yaitu parameter letak yang menunjukkan letak dari jendela menurut mata angin, dimana utara diwakili nilai 1, timur diwakili nilai 2, selatan diwakili nilai 3 dan barat diwakili nilai 4.

Default dari letak jendela ada dibagian pojok dari bangunan, oleh karena itu diberikan parameter awal, jarak1, jarak2 dan jarak tiang untuk membantu dalam proses perulangan secara horizontal. Dimana jarak1 merupakan jarak antar pinggir jendela, jarak2 merupakan jarak tengah antar jendela yang saling berdekatan dan jarak tiang merupakan jarak antar tiang bangunan. Pada setiap jendela dibangun biasanya terjadi pola dua, tiga atau lebih jendela yang saling berdekatan, oleh karena itu diberikan parameter modulo sebagai pembagi jumlah jendela. Kemudian parameter bawah1 dan bawah2 sebagai batas bawah pertama saat jendela dilantai 1 dan batas bawah selanjutnya saat jendela dalam proses perulangan secara vertikal.

Pada saat proses pembuatan jendela dibadan bangunan biasanya terdapat tempat yang tidak terdapat jendelanya, oleh karena itu diberikan parameter spasi, tengah1 dan tengah2 yang berguna untuk melewati area yang tidak memiliki jendela. Dengan parameter yang telah disebutkan dibuat sebuah kode program untuk prosedur pembuatan jendela ditunjukkan pada Gambar 3.50

```

for k=1:1:length(arrid)
    if arrid(1,k) == id_bangunan_jendela
        for l=1:lantai
            for j=1:jumlah
                awal2 = awal;
                if mod(j,modulo+1) ~= 0
                    if arrsudut(1,k) == 90 || arrsudut(1,k) == 270
                        if letak == 1
                            jendelaa(arrlon(1,k)-((arrpb(1,k)/resolusi)/2)+
                                ((awal+jarak1)/resolusi),arrlat(1,k)+
                                (arrlb(1,k)/resolusi)/2),datagrid(round(arrlat(1,k)),
                                    round(arrlon(1,k))), resolusi,panjang,lebar, tinggi,sudut,
                                    id_jendela,atas,kanan,bawah1);
                        elseif letak == 2
                            jendelaa(arrlon(1,k)+((arrpb(1,k)/resolusi)/2),arrlat(1,k)-
                                ((arrlb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                                    datagrid(round(arrlat(1,k)),round(arrlon(1,k))),
                                    resolusi,panjang,lebar,tinggi,sudut,
                                    id_jendela,atas,kanan,bawah1);
                        elseif letak == 3
                            jendelaa(arrlon(1,k)-((arrpb(1,k)/resolusi)/2)+
                                ((awal+jarak1)/resolusi),arrlat(1,k)
                                    ((arrlb(1,k)/resolusi)/2),datagrid(round(arrlat(1,k)),
                                        round(arrlon(1,k))), resolusi,panjang,lebar,tinggi,
                                        sudut,id_jendela,atas,kanan,bawah1);
                        elseif letak == 4
                            jendelaa(arrlon(1,k)-((arrpb(1,k)/resolusi)/2),arrlat(1,k)-
                                ((arrlb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                                    datagrid(round(arrlat(1,k)),round(arrlon(1,k))),
                                    resolusi,panjang,lebar,tinggi,sudut,
                                    id_jendela,atas,kanan,bawah1);
                        end
                    elseif arrsudut(1,k) == 0 || arrsudut(1,k) == 180
                        if letak == 1
                            jendelaa(arrlon(1,k)-((arrlb(1,k)/resolusi)/2)+
                                ((awal+jarak1)/resolusi),arrlat(1,k)+
                                ((arrpb(1,k)/resolusi)/2),datagrid(round(arrlat(1,k)),
                                    round(arrlon(1,k))),resolusi,panjang,lebar,tinggi,sudut,
                                    id_jendela,atas,kanan,bawah1);
                        elseif letak == 2
                            jendelaa(arrlon(1,k)+((arrlb(1,k)/resolusi)/2),arrlat(1,k)-
                                ((arrpb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                                    datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                                    panjang,lebar,tinggi,sudut,id_jendela,atas,kanan,bawah1);
                        elseif letak == 3
                            jendelaa(arrlon(1,k)-((arrlb(1,k)/resolusi)/2)+
                                ((awal+jarak1)/resolusi),arrlat(1,k)-
                                ((arrpb(1,k)/resolusi)/2),datagrid(round(arrlat(1,k)),
                                    round(arrlon(1,k))),resolusi,panjang,lebar,tinggi,sudut,
                                    id_jendela,atas,kanan,bawah1);
                        elseif letak == 4
                            jendelaa(arrlon(1,k)-((arrlb(1,k)/resolusi)/2),arrlat(1,k)-
                                ((arrpb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                                    datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                                    panjang,lebar,tinggi,sudut,id_jendela,atas,kanan,bawah1);
                        end
                    end
                end
                if mod(j,modulo+1) ~= 0
                    jarak1 = jarak1 + jarak2;
                elseif mod(j,modulo+1) == 0
                    awal3 = awal2 + jarak;    awal = awal3;    jarak1 = jarak_lama;
                end
                if j == tengah1 || j == tengah2
                    awal3 = awal2 + (jarak*spasi);    awal = awal3;
                end
            end
            bawah1 = bawah1 + bawah2;    jarak1 = jarak_lama;    awal = awal_lama;
        end
    end
end

```

Gambar 3.50 Kode program prosedur penempatan jendela

Prosedur pembuatan jendela dimulai dengan pencocokan id pada badan bangunan dan id pada jendela, kemudian dideklarasikan perulangan sebanyak jumlah lantai yang ada di badan bangunan untuk membuat perulangan vertikal. Selanjutnya dideklarasikan perulangan sebanyak jumlah jendela yang ada pada satu lantai untuk membuat perulangan horizontal. Pola jendela yang saling berdekatan dibangun dengan menggunakan fungsi modulo, lalu terdapat percabangan untuk sudut hadap dari badan bangunan yang mempengaruhi letak garis longitude dan latitude. Percabangan letak jendela yang mengacu pada 4 arah mata angin berfungsi untuk menentukan operasi penambahan atau pengurangan nilai longitude dan latitude oleh parameter panjang dan lebar badan bangunan. Dari prosedur ini dapat dibuat banyak model dan pola peletakan jendela yang ada pada setiap badan bangunan.

3.2.3.4 Procedural Modeling pada Pintu

Procedur untuk penetapan titik longitude dan latitude pintu menggunakan parameter panjang, lebar, jarak antar pintu, jarak antar tiang bangunan, dan nilai tengah. Parameter selengkapnya dipanggil dengan kode program pada Gambar 3.51

```
PINTU = xlsread([tempat_yg_dipilih 'pintu.xlsx']);
for baris=1:1:size(PINTU,1)
    load badan_all
    id_bangunan_pintu = PINTU(baris,1);    id_pintu = PINTU(baris,2);
    panjang = PINTU(baris,3);              lebar = PINTU(baris,4);
    tinggi = PINTU(baris,5);               sudut = PINTU(baris,6);
    lantai = PINTU(baris,7);              jumlah = PINTU(baris,8);
    letak = PINTU(baris,9);               awal = PINTU(baris,10);
    jarak1 = PINTU(baris,11);             bawah1 = PINTU(baris,12);
    bawah2 = PINTU(baris,13);             spasi = PINTU(baris,14);
    tengah1 = PINTU(baris,15);           tengah2 = PINTU(baris,16);
    jarak = PINTU(baris,17);
    %VARIABEL PENYIMPAN NILAI LAMA
    awal_lama = awal;
    jarak_lama = jarak1;
end
```

Gambar 3.51 Kode program untuk memanggil database pintu

Data pintu yang dipanggil berupa ukuran pintu yaitu panjang, lebar, tinggi dan sudut hadap pintu. Parameter lantai menunjukkan jumlah lantai dari bangunan dan parameter jumlah menunjukkan jumlah pintu pada satu lantai. Selanjutnya yaitu parameter letak yang menunjukkan letak dari pintu menurut mata angin, dimana utara diwakili oleh nilai 1, timur diwakili oleh nilai 2, selatan diwakili oleh nilai 3 dan barat diwakili oleh nilai 4.

Default dari letak pintu ada dibagian pojok dari bangunan, oleh karena itu diberikan parameter awal, jarak1 dan jarak tiang untuk membantu dalam proses perulangan secara horizontal. Dimana awal merupakan jarak awal dari pojok badan bangunan saat pintu diletakkan, kemudian jarak1 merupakan jarak antar jendela saat terjadi perulangan dan jarak tiang merupakan jarak antar tiang bangunan. Kemudian parameter bawah1 dan bawah2 sebagai batas bawah pertama saat pintu dilantai 1 dan batas bawah selanjutnya saat pintu dalam proses perulangan secara vertikal.

Pada saat proses pembuatan pintu dibadan bangunan biasanya terdapat tempat yang tidak terdapat jendelanya, oleh karena itu diberikan parameter spasi, tengah1 dan tengah2 yang berguna untuk melewati area yang tidak memiliki pintu. Dimana tengah1 dan tengah2 merupakan parameter yang digunakan sebagai nilai acuan dari perulangan keberapa jendela tidak ditempatkan, dan spasi merupakan nilai kali jarak yang digunakan untuk menentukan area yang tidak ditempati jendela. Dengan parameter yang telah disebutkan dibuat sebuah kode program untuk prosedur pembuatan pintu ditunjukkan pada Gambar 3.52.

```

for k=1:1:length(arrid)
    if arrid(1,k) == id_bangunan_pintu
        for l=1:lantai
            for j=1:jumlah
                awal2 = awal;
                if arrsudut(1,k) == 90 || arrsudut(1,k) == 270
                    if letak == 1
                        pintu(arrlon(1,k)-((arrpb(1,k)/resolusi)/2)+
                            ((awal+jarak1)/resolusi),arrlat(1,k)+
                            ((arrlb(1,k)/resolusi)/2),
                            datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                            panjang,lebar,tinggi,sudut,id_pintu,bawah1);
                    elseif letak == 2
                        pintu(arrlon(1,k)+((arrpb(1,k)/resolusi)/2),arrlat(1,k)-
                            ((arrlb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                            datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                            panjang,lebar,tinggi,sudut,id_pintu,bawah1);
                    elseif letak == 3
                        pintu(arrlon(1,k)-((arrpb(1,k)/resolusi)/2)+
                            ((awal+jarak1)/resolusi),arrlat(1,k)-
                            ((arrlb(1,k)/resolusi)/2),
                            datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                            panjang,lebar,tinggi,sudut,id_pintu,bawah1);
                    elseif letak == 4
                        pintu(arrlon(1,k)-((arrpb(1,k)/resolusi)/2),arrlat(1,k)-
                            ((arrlb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                            datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                            panjang,lebar,tinggi,sudut,id_pintu,bawah1);
                    end
                elseif arrsudut(1,k) == 0 || arrsudut(1,k) == 180
                    if letak == 1
                        pintu(arrlon(1,k)-((arrlb(1,k)/resolusi)/2)+
                            ((awal+jarak1)/resolusi),arrlat(1,k)+
                            ((arrpb(1,k)/resolusi)/2),
                            datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                            panjang,lebar,tinggi,sudut,id_pintu,bawah1);
                    elseif letak == 2
                        pintu(arrlon(1,k)+((arrlb(1,k)/resolusi)/2),arrlat(1,k)-
                            ((arrpb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                            datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                            panjang,lebar,tinggi,sudut,id_pintu,bawah1);
                    elseif letak == 3
                        pintu(arrlon(1,k)-((arrlb(1,k)/resolusi)/2)+
                            ((awal+jarak1)/resolusi),arrlat(1,k)-
                            ((arrpb(1,k)/resolusi)/2),
                            datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                            panjang,lebar,tinggi,sudut,id_pintu,bawah1);
                    elseif letak == 4
                        pintu(arrlon(1,k)-((arrlb(1,k)/resolusi)/2),arrlat(1,k)-
                            ((arrpb(1,k)/resolusi)/2)+((awal+jarak1)/resolusi),
                            datagrid(round(arrlat(1,k)),round(arrlon(1,k))),resolusi,
                            panjang,lebar,tinggi,sudut,id_pintu,bawah1);
                    end
                end
                if j == tengah1 || j == tengah2
                    awal3 = awal2 + (jarak*spasi);    awal = awal3;
                else
                    awal3 = awal2 + jarak;    awal = awal3;    jarak1 = jarak_lama;
                end
                end
                bawah1 = bawah1 + bawah2;    jarak1 = jarak_lama;    awal = awal_lama;
            end
        end
    end
end
end

```

Gambar 3.52 Kode program prosedur penempatan pintu

Prosedur pembuatan pintu dimulai dengan pencocokan id bangunan pada badan bangunan dan id bangunan pada pintu, kemudian dideklarasikan perulangan sebanyak jumlah lantai yang ada di badan bangunan untuk membuat perulangan vertikal. Selanjutnya dideklarasikan perulangan sebanyak jumlah pintu yang ada pada satu lantai untuk membuat perulangan horizontal. Terdapat percabangan untuk sudut hadap dari badan bangunan yang mempengaruhi letak garis longitude dan latitude. Percabangan letak pintu yang mengacu pada 4 arah mata angin berfungsi untuk menentukan operasi penambahan atau pengurangan nilai longitude dan latitude oleh parameter panjang dan lebar badan bangunan. Dari prosedur ini dapat dibuat banyak model dan pola peletakan pintu yang ada pada setiap badan bangunan.

3.3 Perancangan Interface

Interface pada aplikasi ini dibuat dalam satu halaman interface, yang terbagi menjadi 4 (empat) panel yaitu panel Visualize, panel Level Of Detail, panel View dan panel Euclidean. Panel Visualize berfungsi sebagai panel yang menampilkan hasil dari visualisasi model 3D bangunan yang telah dibuat, kemudian panel Level Of Detail digunakan sebagai kontrol untuk menampilkan hasil dari model 3D bangunan yang terbagi menjadi 3 level detail. Didalam panel View terdapat beberapa tombol untuk control view dari penggunaan dan panel Euclidean digunakan untuk menghitung jarak pada bangunan. Berikut merupakan halaman interface aplikasi yang ditunjukkan pada Gambar 3.53.



Gambar 3.53 Halaman interface awal aplikasi

a. Panel Visualize

Pada saat aplikasi berjalan panel visualize berfungsi sebagai tempat untuk memvisualisasikan model permukaan tanah dan hasil dari pemodelan 3D bangunan. Tampilan panel saat aplikasi sudah dijalankan dan menghasilkan model permukaan serta model 3D bangunan ditunjukkan pada gambar 3.54

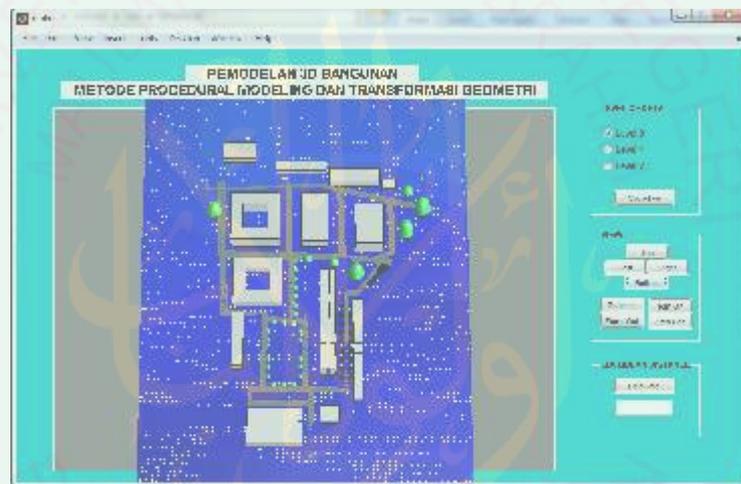


Gambar 3.54 Halaman interface ketika aplikasi sudah dijalankan

b. Panel Level Of Detail

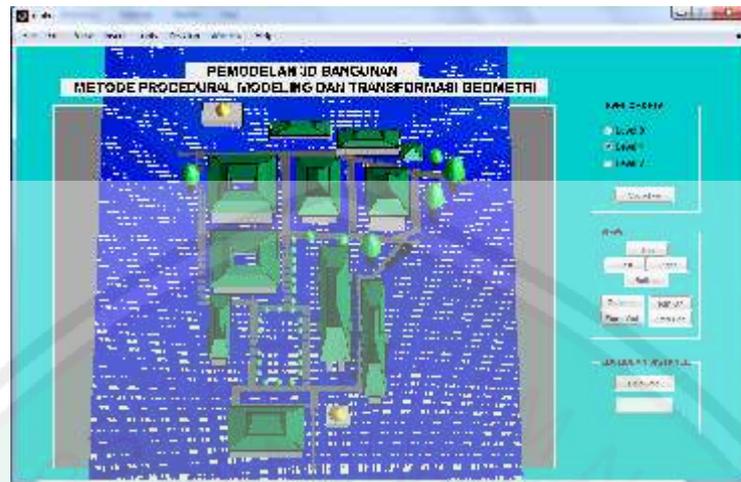
Panel level of detail merupakan panel kontrol dari visualisasi, dimana didalam panel ini terdapat tombol visualize dan 3 (tiga) radio button yaitu radio

button level 0, radio button level 1 dan radio button level 2. Radio button berfungsi mengontrol detail model yang akan divisualisasikan, level 0 akan menampilkan model badan, kemudian level 1 model badan akan dilengkapi dengan detail atap dan di level 2 model bangunan divisualisasikan lengkap dengan detail fitur jendela, pintu, tangga dan juga tiang. Pada Gambar 3.55 ditunjukkan hasil visualisasi model 3D bangunan pada level 0 yang menampilkan badan bangunan.



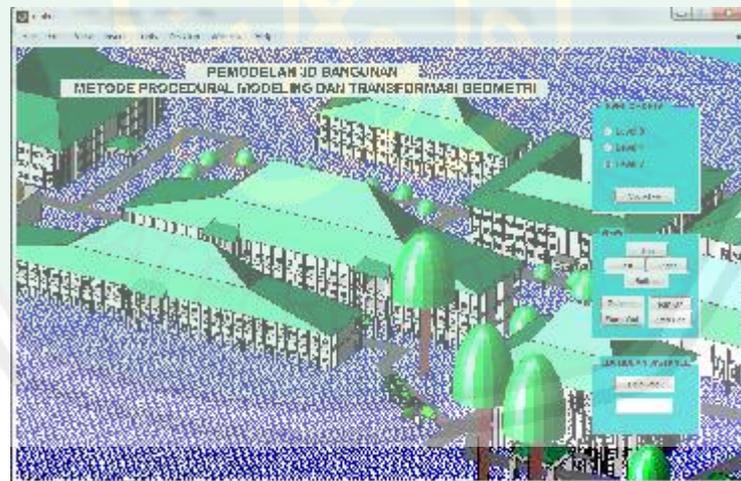
Gambar 3.55 Hasil model 3D bangunan pada level 0

Pada Gambar 3.55 ditampilkan hasil model 3D bangunan pada level 0 yang menampilkan badan bangunan saja tanpa atribut apapun di bangunannya. Kemudian pada Gambar 3.56 ditunjukkan hasil model 3D bangunan pada level 1 yang menampilkan badan bangunan dan atapnya.



Gambar 3.56 Hasil model 3D bangunan pada level 1

Pada Gambar 3.65 ditampilkan hasil model 3D bangunan pada level 1 yang menampilkan badan bangunan dan atap tanpa fitur kubah, jendela dan pintu di bangunannya. Kemudian pada Gambar 3.57 ditunjukkan hasil model 3D bangunan pada level 2 yang menampilkan badan bangunan dan seluruh fiturnya.



Gambar 3.57 Hasil model 3D bangunan pada level 2

c. Panel View

Panel view dilengkapi dengan fungsi button yang digunakan untuk control view dari user, didalam panel ini terdapat tombol top, bottom, right, left, zoom in, zoom out, pan on dan pan off. Untuk mengatur sudut pandang dari user

menggunakan 4 tombol berikut yaitu tombol top untuk membuat model 3D berotasi keatas, tombol bottom untuk membuat model 3D berotasi kebawah, tombol right untuk membuat model 3D berotasi kekanan dan tombol left untuk membuat model 3D kekiri. Tombol zoom in dan zoom out untuk memperbesar model 3D dan memperkecil model 3D, sedangkan tombol pan on dan pan off untuk mengaktifkan mode navigasi secara manual dan menon-aktifkannya.

d. Panel Euclidean

Panel Euclidean dilengkapi dengan button yang dapat mengukur panjang, lebar dan jarak suatu bangunan dengan menggunakan metode Euclidean distance. Hasil dari pengukuran ini ditampilkan di sebuah edit teks yang berada dibawah tombol calculate.

BAB IV

UJI COBA DAN PEMBAHASAN

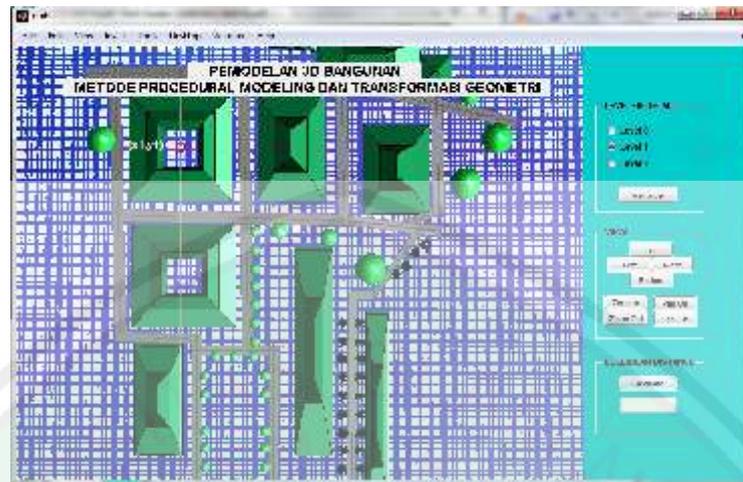
Pada bab uji coba dan pembahasan, menjelaskan mengenai langkah – langkah uji coba, hasil pengujian dan pembahasan. Uji coba dilakukan dengan dua cara, yaitu uji coba dengan mengukur jarak antar bangunan dan uji coba perbandingan dengan penelitian sebelumnya menggunakan metode yang berbeda. Uji coba pengukuran dilakukan untuk mengetahui akurasi dari jarak bangunan yang nyata dengan jarak bangunan pada aplikasi. Uji coba perbandingan dilakukan untuk mengetahui keefektifan metode yang digunakan dalam membangun model 3D bangunan dalam segi detail bangunan.

4.1 Uji Coba Pengukuran Jarak dan Ukuran Bangunan

4.1.1 Langkah – Langkah Uji Coba

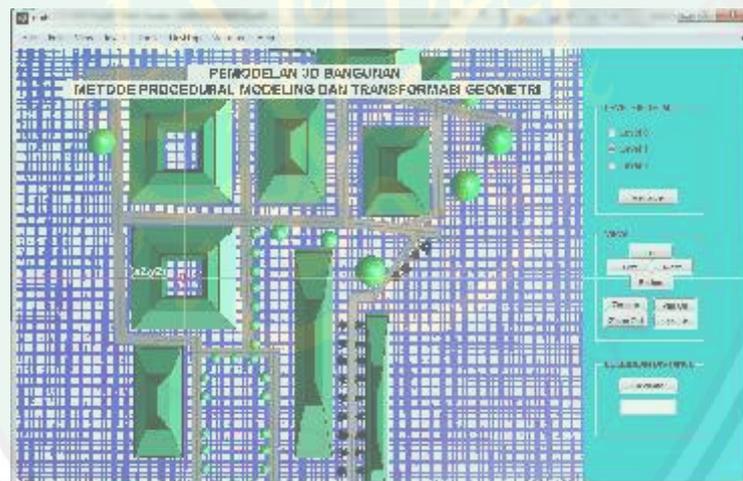
Langkah – langkah uji coba dalam penelitian ini dapat dijabarkan sebagai berikut :

- a. Pengujian awal dilakukan dengan pengukuran panjang dan lebar bangunan beserta jarak antara bangunan satu dengan bangunan lain dengan menggunakan metode euclidean distance. Uji coba aplikasi ini dilakukan dengan menandai dua titik koordinat dan menggunakan satuan panjang meter. Titik koordinat pertama (x_1, y_1) didapatkan dari sebuah bangunan yang ditunjukkan pada Gambar 4.1



Gambar 4.1 Pengambilan titik (x_1, y_1) pada bangunan

Kemudian menandai titik koordinat kedua (x_2, y_2) yang ditunjukkan pada gambar 4.2



Gambar 4.2 Pengambilan titik (x_2, y_2) pada bangunan

Setelah menandai dua titik tersebut, maka akan menghasilkan besar jarak dari kedua bangunan seperti yang ditunjukkan pada Gambar 4.3



Gambar 4.3 Hasil pengukuran dengan euclidean distance

- b. Menghitung panjang dan lebar dari tiap bangunan beserta jarak dari tiap bangunan yang ada, sehingga didapatkan sample uji coba untuk data pengukuran panjang, lebar dan jarak bangunan
- c. Pengujian dilakukan dengan cara menghitung selisih antara data pengukuran bangunan pada Google Earth dan data pengukuran yang dihasilkan oleh aplikasi.
- d. Selanjutnya menghitung rata-rata selisih dari jarak yang diukur dengan Google Earth dan jarak yang diukur dengan aplikasi. Nilai selisih dijumlahkan lalu dibagi dengan banyaknya data yang dijadikan sampel pengukuran.

4.1.2 Hasil Uji Coba

Hasil uji coba didapatkan setelah langkah – langkah pengujian dilakukan, kemudian data tersebut disajikan dalam bentuk tabel. Hasil uji coba dengan parameter pengukuran bangunan dilakukan dengan menghitung selisih dari data yang didapatkan dari Google Earth dengan data pengukuran yang dihasilkan oleh aplikasi. Pada Tabel 4.1 ditunjukkan hasil dari pengukuran antar bangunan dengan selisih jarak dari pengukuran Google Earth dan pengukuran aplikasi.

Tabel 4.1 Hasil uji coba pengukuran jarak antar bangunan

| No | Titik Awal | Titik Akhir | Pengukuran Google Earth | Pengukuran Aplikasi | Selisih (meter) | Akurasi (%) |
|----|-----------------|---------------|-------------------------|---------------------|-----------------|-------------|
| 1 | Pusat Informasi | Rektorat | 63.5 | 63.2615 | 0.2385 | 99.62441 |
| 2 | Pusat Informasi | Humaniora | 40.3 | 38.6521 | 1.6479 | 95.91092 |
| 3 | Pusat Informasi | Microteaching | 102.86 | 98.546 | 4.314 | 95.80595 |
| 4 | Pusat Informasi | Perpustakaan | 97.48 | 95.3389 | 2.1411 | 97.80355 |
| 5 | Pusat Informasi | Saintek | 157.78 | 151.423 | 6.357 | 95.97097 |
| 6 | Pusat Informasi | Gedung A | 204 | 197.709 | 6.291 | 96.91618 |
| 7 | Pusat Informasi | Gedung B | 187.6 | 183.676 | 3.924 | 97.90832 |
| 8 | Pusat Informasi | Megawati | 195.19 | 185.735 | 9.455 | 95.156 |
| 9 | Pusat Informasi | Gedung C | 257.11 | 248.326 | 8.784 | 96.58356 |
| 10 | Pusat Informasi | Sport Center | 326.18 | 312.48 | 13.7 | 95.79987 |
| 11 | Humaniora | Rektorat | 67.9 | 67.3959 | 0.5041 | 99.25758 |
| 12 | Humaniora | Microteaching | 61.23 | 60.4221 | 0.8079 | 98.68055 |
| 13 | Humaniora | Perpustakaan | 71 | 68.8417 | 2.1583 | 96.96014 |
| 14 | Humaniora | Saintek | 123.33 | 120.112 | 3.218 | 97.39074 |
| 15 | Humaniora | Gedung A | 209 | 202.367 | 6.633 | 96.82632 |
| 16 | Humaniora | Gedung B | 183.94 | 179.304 | 4.636 | 97.47961 |
| 17 | Humaniora | Megawati | 173.43 | 164.352 | 9.078 | 94.76561 |
| 18 | Humaniora | Gedung C | 239.32 | 230.411 | 8.909 | 96.27737 |
| 19 | Humaniora | Sport Center | 319.79 | 307.095 | 12.695 | 96.03021 |
| 20 | Microteaching | Rektorat | 110.21 | 108.593 | 1.617 | 98.5328 |
| 21 | Microteaching | Perpustakaan | 69.91 | 68.6845 | 1.2255 | 98.24703 |
| 22 | Microteaching | Saintek | 83.08 | 82.9834 | 0.0966 | 99.88373 |
| 23 | Microteaching | Gedung A | 231.7 | 224.857 | 6.843 | 97.04661 |
| 24 | Microteaching | Gedung B | 197 | 192.037 | 4.963 | 97.48071 |
| 25 | Microteaching | Megawati | 155.03 | 147.547 | 7.483 | 95.17319 |
| 26 | Microteaching | Gedung C | 228.09 | 219.485 | 8.605 | 96.22737 |
| 27 | Microteaching | Sport Center | 321.79 | 309.881 | 11.909 | 96.29914 |
| 28 | Rektorat | Perpustakaan | 61.69 | 61.4143 | 0.2757 | 99.55309 |
| 29 | Rektorat | Saintek | 127.53 | 124.442 | 3.088 | 97.57861 |
| 30 | Rektorat | Gedung A | 142.94 | 138.022 | 4.918 | 96.5594 |
| 31 | Rektorat | Gedung B | 125.14 | 121.216 | 3.924 | 96.86431 |
| 32 | Rektorat | Megawati | 144.04 | 134.954 | 9.086 | 93.69203 |
| 33 | Rektorat | Gedung C | 198.48 | 191.313 | 7.167 | 96.38906 |
| 34 | Rektorat | Sport Center | 262.91 | 251.644 | 11.266 | 95.71488 |
| 35 | Perpustakaan | Saintek | 65.44 | 64.3208 | 1.1192 | 98.28973 |
| 36 | Perpustakaan | Gedung A | 160.58 | 158.01 | 2.57 | 98.39955 |
| 37 | Perpustakaan | Gedung B | 126.82 | 124.972 | 1.848 | 98.54282 |
| 38 | Perpustakaan | Megawati | 101.96 | 95.9439 | 6.0161 | 94.09955 |

| | | | | | | |
|----|--------------|--------------|--------|---------|--------|----------|
| 39 | Perpustakaan | Gedung C | 170.42 | 164.542 | 5.878 | 96.55087 |
| 40 | Perpustakaan | Sport Center | 256.76 | 247.279 | 9.481 | 96.30745 |
| 41 | Saintek | Gedung A | 193.5 | 185.954 | 7.546 | 96.10026 |
| 42 | Saintek | Gedung B | 149.77 | 146.088 | 3.682 | 97.54156 |
| 43 | Saintek | Megawati | 78.76 | 75.5339 | 3.2261 | 95.90389 |
| 44 | Saintek | Gedung C | 152.45 | 145.741 | 6.709 | 95.59921 |
| 45 | Saintek | Sport Center | 256.13 | 244.725 | 11.405 | 95.54718 |
| 46 | Gedung A | Gedung B | 44.64 | 44.3575 | 0.2825 | 99.36716 |
| 47 | Gedung A | Megawati | 137.15 | 130.45 | 6.7 | 95.11484 |
| 48 | Gedung A | Gedung C | 128.07 | 124.315 | 3.755 | 97.06801 |
| 49 | Gedung A | Sport Center | 133.01 | 128.138 | 4.872 | 96.33712 |
| 50 | Gedung B | Megawati | 91.28 | 87.2742 | 4.0058 | 95.61152 |
| 51 | Gedung B | Gedung C | 93.97 | 90.561 | 3.409 | 96.37225 |
| 52 | Gedung B | Sport Center | 136.26 | 130.734 | 5.526 | 95.94452 |
| 53 | Megawati | Gedung C | 72.53 | 71.8154 | 0.7146 | 99.01475 |
| 54 | Megawati | Sport Center | 177.05 | 169.961 | 7.089 | 95.99605 |
| 55 | Gedung C | Sport Center | 110.02 | 106.671 | 3.349 | 96.95601 |

Dari Tabel 4.1 dapat diketahui rata – rata prosentase akurasi jarak dari dua pengukuran sebagai berikut :

$$\frac{\text{Total akurasi}}{\text{Data keseluruhan}} = \frac{5327,054}{55} = 96.85553 \%$$

Hasil dari pengujian pengukuran dengan menghitung rata – rata prosentase akurasi jarak antara data pengukuran yang dilakukan di Google Earth dengan data pengukuran yang dilakukan di aplikasi sebesar 96.85553 persen.

Kemudian dilakukan uji coba terhadap ukuran bangunan berupa panjang dan lebar bangunan menghitung selisih dari data yang didapatkan dari Google Earth dengan data pengukuran yang dihasilkan oleh aplikasi. Pada Tabel 4.2 ditunjukkan hasil dari pengukuran panjang dan lebar dari masing - masing bangunan dengan nilai selisih dari pengukuran Google Earth dan pengukuran aplikasi.

Tabel 4.2 Hasil uji coba pengukuran panjang dan lebar tiap bangunan

| No | Nama Gedung | Data Panjang | Hasil Aplikasi | Selisih | Akurasi (%) | Data Lebar | Hasil Aplikasi | Selisih | Akurasi (%) |
|----|-----------------|--------------|----------------|---------|-------------|------------|----------------|---------|-------------|
| 1 | Gedung B | 118.8 | 117.289 | 1.511 | 98.72811 | 24 | 24.3193 | 0.3193 | 98.66958 |
| 2 | Gedung A | 118.8 | 118.529 | 0.271 | 99.77189 | 18 | 17.4444 | 0.5556 | 96.91333 |
| 3 | Gedung C | 64 | 63.7439 | 0.2561 | 99.59984 | 16 | 16.6 | 0.6 | 96.25 |
| 4 | Megawati | 54 | 53.5449 | 0.4551 | 99.15722 | 54 | 53.5449 | 0.4551 | 99.15722 |
| 5 | Rektorat | 48 | 48.4621 | 0.4621 | 99.03729 | 32 | 32.2383 | 0.2383 | 99.25531 |
| 6 | Perpustakaan | 60 | 59.3184 | 0.6816 | 98.864 | 30 | 29.6592 | 0.3408 | 98.864 |
| 7 | Saintek | 57.6 | 58.6444 | 1.0444 | 98.18681 | 54 | 53.5449 | 0.4551 | 99.15722 |
| 8 | Pusat Informasi | 16 | 16.7639 | 0.7639 | 95.22563 | 16 | 15.2985 | 0.7015 | 95.61563 |
| 9 | Sport Center | 60 | 59.246 | 0.754 | 98.74333 | 40 | 40.7961 | 0.7961 | 98.00975 |
| 10 | Humaniora | 56 | 54.8197 | 1.1803 | 97.89232 | 16 | 15.2985 | 0.7015 | 95.61563 |
| 11 | Microteaching | 56 | 54.8197 | 1.1803 | 97.89232 | 12 | 12.7488 | 0.7488 | 93.76 |

Dari Tabel 4.2 dapat diketahui rata – rata selisih pengukuran panjang dan lebar dari tiap bangunan sebagai berikut :

$$\frac{\text{Total akurasi panjang}}{\text{Data keseluruhan}} = \frac{1083.099}{11} = 98.46352 \%$$

$$\frac{\text{Total akurasi lebar}}{\text{Data keseluruhan}} = \frac{1071.268}{11} = 97.38797 \%$$

Hasil dari pengujian pengukuran dengan menghitung rata – rata prosentase akurasi pengukuran panjang tiap bangunan sebesar 98.46352 persen dan rata – rata prosentase akurasi selisih pengukuran lebar tiap bangunan sebesar 97.38797 persen.

4.2 Uji Coba Level Of Detail

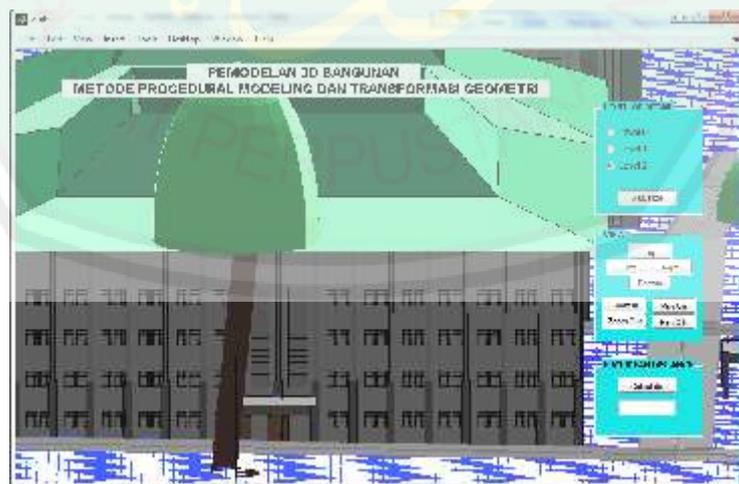
Model 3D bangunan yang menyediakan tampilan dalam beberapa level dengan tingkat ketelitian dan perincian telah dikembangkan dengan menggunakan *Terrestrial Laser Scanner* di CityGML (Akmalia et al, 2014). Pada penelitian ini dibangun aplikasi yang menyediakan tampilan model bangunan dengan 3 level of detail (LOD) yaitu level 0 (LOD0) yang menampilkan badan bangunan, level 1

Selanjutnya pada level 2 menampilkan model 3D bangunan menampilkan badan bangunan, atap bangunan beserta fitur eksterior berupa jendela, pintu dan tiang bangunan seperti pada Gambar 4.6.



Gambar 4.6 Tampilan bangunan fakultas ekonomi pada level 2

Pada bangunan yang lain dengan prosedur pembuatan bangunan yang berbeda dihasilkan model 3D bangunan pada level 2 seperti yang ditunjukkan pada Gambar 4.7

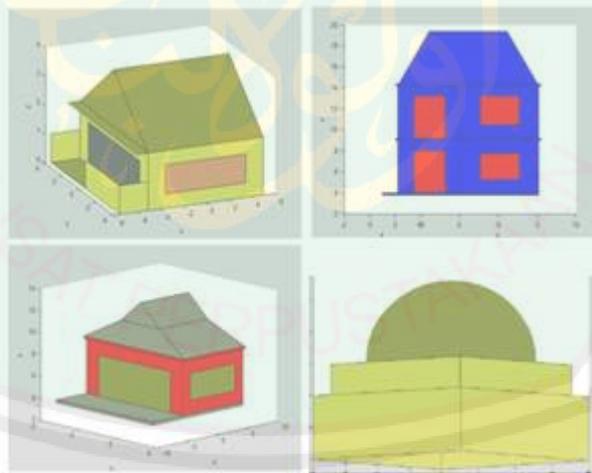


Gambar 4.7 Tampilan bangunan fakultas saintek pada level 2

Dari hasil model 3D bangunan didapatkan informasi bahwa aplikasi ini dapat menampilkan 3 level di LOD (*Level of Detail*) sampai ke LOD2 dengan

detail eksterior yang lengkap dan mirip dengan bentuk nyatanya. Dengan menggunakan kode program yang sama dapat membuat bentuk yang berbeda dengan posisi fitur dan jumlah yang berbeda pula. Uji coba komparasi dilakukan dengan membandingkan aplikasi sebelumnya yang juga membangun 3D model bangunan dengan menggunakan perangkat yang sama.

Aplikasi dalam penelitian ini diuji coba komparasi dengan aplikasi yang dihasilkan oleh penelitian sebelumnya yang dilakukan oleh Astutik (2014). Dalam penelitian sebelumnya model 3D yang dihasilkan ada 4 tipe bangunan yang menggunakan metode polygonal modeling, 4 tipe model bangunan yang dihasilkan yaitu bangunan tipe 1, bangunan tipe 2, bangunan tipe 3 dan bangunan tipe 4. Model 3D bangunan tersebut ditunjukkan pada Gambar 4.8.



Gambar 4.8 Model 3D bangunan pada penelitian sebelumnya (Sumber : Astutik, 2014)

Uji coba dilakukan dengan membandingkan model 3D yang dihasilkan oleh peneliti pada LOD0, LOD1 dan LOD2, kemudian model 3D yang dihasilkan oleh Astutik (2014) dengan menggunakan tipe bangunan yang sejenis, dan model bangunan nyata. Pada perbandingan akan diberi nilai 0 dan 1, dimana nilai 0

mewakili aspek ketidaksesuaian dan nilai 1 mewakili aspek kesesuaian, sehingga dapat diketahui bangunan mana yang memiliki tingkat kesesuaian yang tinggi dengan model bangunan nyata. Perbandingan pertama dilakukan pada bangunan gedung C dengan LOD0, LOD1, LOD2 dan bangunan tipe 2. Pada Gambar 4.9 ditunjukkan tampilan dari masing-masing gambar tersebut.



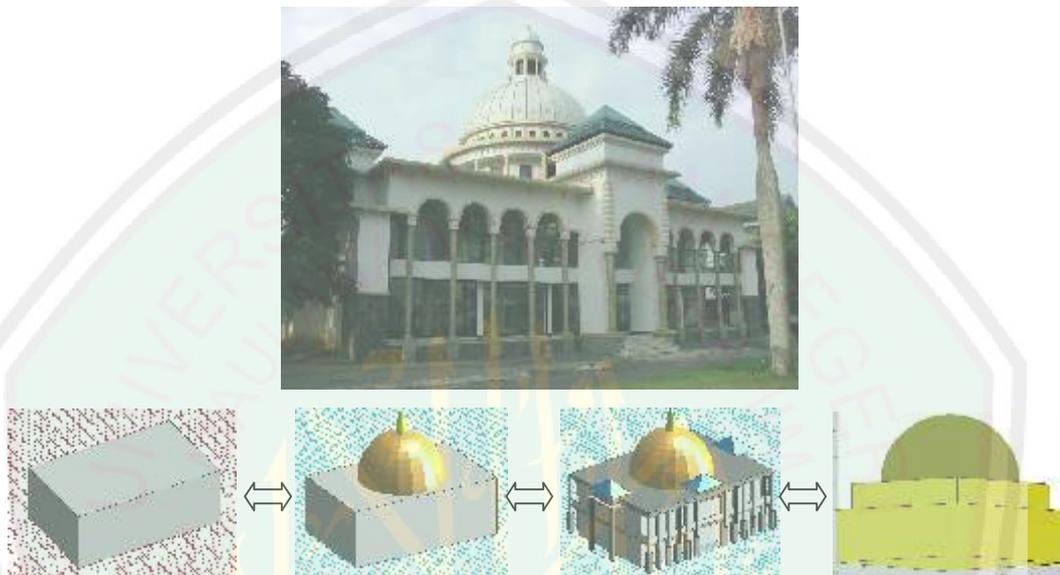
Gambar 4.9 Perbandingan Gedung C dengan LOD0, LOD1, LOD2 dan tipe 2

Perbandingan kedua dilakukan pada bangunan gedung Sport Center dengan LOD0, LOD1, LOD2 dan bangunan tipe 3. Pada Gambar 4.10 ditunjukkan tampilan dari masing-masing gambar tersebut.



Gambar 4.10 Perbandingan Gedung SC dengan LOD0, LOD1, LOD2 dan tipe 3

Perbandingan ketiga dilakukan pada bangunan Masjid Tarbiyah dengan LOD0, LOD1, LOD2 dan bangunan tipe 4. Pada Gambar 4.11 ditunjukkan tampilan dari masing-masing gambar tersebut.



Gambar 4.11 Perbandingan Masjid Tarbiyah dengan LOD0, LOD1, LOD2 dan tipe 4

Kemudian dibuat sebuah tabel yang menampilkan nilai kesesuaian bentuk dari masing – masing bangunan yang dibandingkan, dimana nilai 0 diberikan untuk mewakili ketidaksesuaian dan nilai 1 diberikan untuk mewakili kesesuaian. Aspek yang akan diuji yaitu kesesuaian bentuk dasar, bentuk atap, bentuk kubah, bentuk jendela, bentuk pintu, tiang dan juga teras. Setelah aspek ini diisi Pada Tabel 4.2 ditunjukkan hasil dari perbandingan bentuk yang sudah dilakukan.

Tabel 4.3 Hasil uji coba perbandingan bentuk antar bangunan

| No | Bangunan | Hasil | Bentuk Dasar | Bentuk Atap | Fitur | | | | | Jumlah |
|----|----------|--------|--------------|-------------|-------|---------|-------|-------|-------|--------|
| | | | | | Kubah | Jendela | Pintu | Tiang | Teras | |
| 1 | Gedung C | LOD0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | LOD1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| | | LOD2 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 6 |
| | | Tipe 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| | | | | | | | | | | |
|---|-----------------|--------|---|---|---|---|---|---|---|---|
| 2 | Gedung SC | LOD0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | LOD1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| | | LOD2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 6 |
| | | Tipe 3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 3 | Masjid Tarbiyah | LOD0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | LOD1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| | | LOD2 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 5 |
| | | Tipe 4 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |

Dari data pada Tabel 4.2 dihitung jumlah total yang diperoleh saat membandingkan LOD0, LOD1, LOD2 dan bangunan dari Astutik (2014). Setelah dihitung dapat diketahui nilai terbesar kesesuaian yang paling mirip dengan bentuk nyata dari bangunan.

$$\text{LOD0} = \text{pembanding1} + \text{pembanding2} + \text{pembanding3} = 1 + 1 + 1 = 3$$

$$\text{LOD1} = \text{pembanding1} + \text{pembanding2} + \text{pembanding3} = 2 + 2 + 2 = 6$$

$$\text{LOD2} = \text{pembanding1} + \text{pembanding2} + \text{pembanding3} = 6 + 6 + 5 = 17$$

$$\text{Astutik} = \text{pembanding1} + \text{pembanding2} + \text{pembanding3} = 1 + 2 + 2 = 5$$

Dari tiga sampel perbandingan tersebut didapatkan hasil bahwa nilai kesesuaian pada LOD2 adalah yang paling tinggi dengan akumulasi 17 poin, kemudian di urutan kedua disusul dengan bangunan LOD1 dengan 6 poin, diurutan ketiga dengan 5 poin yaitu bangunan dari Astutik (2014) lalu yang terakhir adalah bangunan LOD0 dengan 3 poin.

Bangunan LOD2 paling sesuai karena memiliki fitur bangunan yang paling lengkap dan mendekati bentuk nyata dari bangunan asli. Bangunan dari penelitian sebelumnya juga memiliki fitur yang sama yaitu jendela dan pintu seperti pada bangunan LOD2, namun pembuatan fiturnya kurang detail karena

jumlah dan bentuk dari fitur tidak menyerupai dengan bentuk jendela dan pintu yang ada. Ukuran dan penempatan fitur bangunan juga masih sebatas perkiraan dari peneliti sehingga bangunan ini bisa dikatakan belum benar - benar masuk ke dalam LOD2.

4.3 Pembahasan

Setelah dilakukan proses uji coba pada aplikasi yang dibagi menjadi dua cara yaitu uji coba pengukuran jarak dan uji coba level of detail kemudian dilakukan analisis terhadap hasil uji coba. Analisis ini berguna sebagai bahan pengembangan aplikasi kedepannya serta didapatkan pengetahuan dan ilmu baru berdasarkan hasil penelitian. Pengujian pertama yaitu pengujian pengukuran jarak antara bangunan satu dengan bangunan lain yang dilakukan dengan menggunakan metode euclidean distance yaitu metode yang sering digunakan untuk mengukur jarak dari dua titik. Berdasarkan hasil uji coba yang telah dilakukan dengan metode Euclidean distance dihasilkan akurasi yang memiliki nilai rata-rata selisih jarak yaitu 5.220762 dalam satuan meter.

Sedangkan pada pengujian kedua yaitu pengujian hasil model 3D bangunan aplikasi ini dengan aplikasi yang dikembangkan sebelumnya, didapatkan hasil yang lebih bagus dalam segi model dan lebih detail dalam segi fitur bangunannya. Model awal dari aplikasi ini berupa model 3D sederhana yang dibedakan berdasarkan badan bangunan, atap, kubah, jendela dan pintu. Dimana setiap model juga memiliki tipenya masing – masing. Dengan metode procedural modeling dibuatlah set aturan atau prosedur dalam penempatan badan bangunan

dengan fiturnya, sehingga didapatkan hasil yang lebih optimal dan bervariasi pada tiap-tiap bangunan yang dibuat. Dengan set aturan yang sudah dibuat didalam program dan nilai input yang berbeda didalam database dapat menghasilkan model bangunan yang lebih unggul dalam segi bentuk dan detail fitur dibandingkan dengan aplikasi sebelumnya.

Prosedur yang digunakan dalam pembuatan bangunan dan penempatan fitur dalam penelitian ini memiliki banyak parameter yang digunakan didalamnya, sehingga menjadikan aplikasi ini lebih dinamis dan mudah dioperasikan ketika akan membuat model bangunan yang baru dengan set model sederhana yang sudah dideklarasikan. Nilai ukuran panjang, lebar, tinggi, tipe dan sudut membuat model badan bangunan dapat dinamis berubah sesuai dengan input yang dimasukkan oleh user. Fitur bangunan juga dapat dibangun secara dinamis dengan jumlah yang berbeda dan pola yang berbeda sehingga menghasilkan model fitur yang lebih variatif dalam model kota 3D bangunan.

4.4 Integrasi Pemodelan 3D Bangunan dengan Islam

Pada latar belakang telah disebutkan bahwa model 3D kota merupakan salah satu aspek penting yang digunakan untuk sejumlah aplikasi yang berhubungan dengan perencanaan kota (Lang et al, 1996). Karena banyaknya informasi yang harus dikelola dalam pembuatan model 3D kota membuat bidang ini terasa kompleks, sehingga diperlukan prosedur pengembangan yang sederhana namun efektif dalam pembuatan model 3D kota (Crysdian, 2011). Prosedur pembuatan model 3D selayaknya dibangun dengan lebih dinamis sehingga didapatkan model bangunan kota yang lebih variatif tipenya. Untuk lebih

mengoptimalkan hasil dari prosedur tersebut, dibuatlah banyak parameter yang membuat setiap model yang dihasilkan memiliki khas tersendiri dan dapat semirip mungkin dengan bentuk nyatanya. Sehingga dihasilkan banyak model yang bervariasi hanya dengan menggunakan satu prosedur dengan banyak parameter yang dinamis.

Dalam surat An Nahl ayat 80 Allah berfirman :

وَاللَّهُ جَعَلَ لَكُمْ مِنْ بُيُوتِكُمْ سَكَنًا وَجَعَلَ لَكُمْ مِنْ جُلُودِ الْأَنْعَامِ بُيُوتًا
تَسْتَخِفُّونَهَا يَوْمَ ظَعْنِكُمْ وَيَوْمَ إِقَامَتِكُمْ وَمِنْ أَصْوَابِهَا وَأَوْبَارِهَا
وَأَشْعَارِهَا أَثْنَا وَمَتَعْنَا إِلَى حِينٍ ﴿٨٠﴾

“Dan Allah menjadikan bagimu rumah-rumahmu sebagai tempat tinggal dan Dia menjadikan bagi kamu rumah-rumah (kemah-kemah) dari kulit binatang ternak yang kamu merasa ringan (membawa)nya di waktu kamu berjalan dan waktu kamu bermukim dan (dijadikan-Nya pula) dari bulu domba, bulu onta dan bulu kambing, alat-alat rumah tangga dan perhiasan (yang kamu pakai) sampai waktu (tertentu).” (QS. An Nahl : 80)

Maksud dari ayat tersebut menurut Ibnu Katsir, bahwa Allah yang Maha Suci lagi Maha tinggi menyebutkan kesempurnaan nikmat-nikmat-Nya yang telah dianugerahkan kepada hamba-hamba-Nya, dimana Dia telah menjadikan bagi manusia rumah-rumah sebagai tempat tinggal untuk berlindung, bernaung dan memperoleh segala macam manfaat dengannya. Selain itu, Allah Ta’ala juga menjadikan bagi manusia rumah-rumah dari kulit binatang ternak yang mereka merasa ringan membawanya dalam perjalanan mereka maupun waktu mereka bermukim. Dari sini kita sebagai manusia diwajibkan untuk selalu bersyukur atas nikmat yang telah Allah berikan, dimana Allah menjadikan tempat tinggal untuk

manusia dari apa yang telah Dia ciptakan di muka bumi dan yang demikian itulah Allah menciptakannya untuk digunakan diberbagai kepentingan dan kebutuhan manusia. Manusia ketika akan membangun rumah atau gedung pastilah membutuhkan bahan seperti pasir, batu bata, semen dan lainnya yang merupakan hasil olahan yang tersedia di bumi, yang sudah disediakan oleh Allah untuk memudahkan kita dalam membuat tempat tinggal dan tempat berlindung yang kokoh dari panas atau hujan, hal seperti ini merupakan nikmat dari Allah yang dengan nikmat bisa menjadi sarana untuk lebih giat beribadah dan mentaati ajaranNya.



BAB V

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Dari hasil implementasi dan uji coba yang telah dilakukan oleh peneliti, didapatkan beberapa kesimpulan yaitu :

- a. Metode procedural modeling dapat diterapkan dalam pembuatan visualisasi model 3D bangunan yang mengacu pada penelitian yang dilakukan Merrel. Prosedur pemodelan dipergunakan untuk membuat model 3D bangunan yang banyak dan kompleks, prosedur ini juga dapat menampilkan model bangunan pada level LOD2. Beberapa set model dibangun dengan polygonal modeling dengan banyak tipe didalamnya, kemudian disatukan dengan set prosedur yang dikontrol oleh beberapa parameter dengan procedural modeling. Transformasi geometri juga berperan baik dalam menentukan sudut hadap dari model 3D bangunan dan fitur. Kombinasi antara procedural modeling dan transformasi geometri dapat membuat bentuk model 3D bangunan yang bervariasi dan detail.
- b. Hasil dari visualisasi model 3D bangunan dengan menggunakan metode procedural modeling dan transformasi geometri terbukti efektif dapat menghasilkan model 3D bangunan yang lebih bagus, detail dan sesuai dengan bentuk nyata melalui uji komparasi dengan penelitian sebelumnya. Hasil model 3D di penelitian ini lebih bervariasi dan detail fiturnya lebih

bagus dan dapat dibangun semirip mungkin dengan bangunan nyata yang ada melalui inputan parameter yang tersedia.

5.2 SARAN

Beberapa saran untuk penelitian dan pengembangan aplikasi selanjutnya adalah sebagai berikut :

- a. Aplikasi pada penelitian ini masih berbasis desktop, untuk pengembangannya dapat dibangun aplikasi yang menggunakan perangkat lain semisal web.
- b. Pemodelan 3D bangunan mencapai ke level 2 pada LOD (Level Of Detail) yang menampilkan eksterior dari bangunan, untuk pengembangan selanjutnya dapat dibangun aplikasi yang dapat menampilkan interior dari bangunan yaitu pada level LOD3.
- c. Pengembangan prosedur untuk pemodelan pada objek lain seperti vegetasi, jalan atau kendaraan.

DAFTAR PUSTAKA

- Autodesk.2007.*Polygonal Modeling*.
<http://www.arch.cuhk.edu.hk/server1/staff1/marcaurel/desc9019/tutorials/Readings/mayaDocs/PolygonalModeling.pdf> . Diunduh tanggal 17 Februari 2015 pukul 00.18 WIB.
- Crysdian, C.2011.*Development of 3D City Model using Polygonal Modeling Approach*. Yogyakarta: Conference on Information Technology and Electrical Engineering (CITEE).
- Dollner, Jurgen, Thomas H.Kolbe, Falko Liecke, Takis Sgourus and Karin Teichman. 2006.*The Virtual 3d City Model Of Berlin - Managing, Integrating And Communicating Complex Urban Information*. Aalborg: Proceeding of the 25th International Symposium on Urban Data Management UDMS.
- Forstner, Wolfgang.1999.*3D City Models Automatic and Semiautomatic Acquisition Methods*. Bonn: Bonn University.
- Ganster, Bjorn and Reinhard Klein. 2013. *Procedural Modeling*.<http://cg.cs.uni-bonn.de/en/projects/procedural-modeling>. Diakses tanggal 8 Februari 2015 pukul 19.45 WIB.
- Gulrez, Tauseff, Hassanien and Aboul Ella.2012. *Advances in Robotics and Virtual Reality*.Berlin: Springer.
- Katz, VJ.1998.*A History of Mathematics: An Introduction (second Ed)*
- Kobayashi, Y.2006.*Photogrammetry and 3D city Modeling*.USA: Arizona State University.
- Lang, Felicias and Wolfgang Forstner.1996.*3D-city modeling with a digital one-eye stereo system*. Bonn: Institute for Photogrammetri, Bonn University.
- Merrell, Paul and Dinesh Manocha.2011. *Model Synthesis: A General Procedural Modeling Algorithm*.Chapel Hill: IEEE Trancaction on Visualization and Computer Graphics.
- Muller, Pascal and Yoav I. H. Parish.2001.*Procedural Modeling of Cities*.New York: Proceedings of the 28th annual conference on Computer graphics and interactive techniques Pages 301-308.

Muller, Pascal, Peter Wonka, Simon Haegler, Andreas Ulmer and Luc Van Gool.2006.*Procedural Modeling of Building*.New York: ACM SIGGRAPH 2006 Papers Pages 614-623.

Pavithran, Rupin.2008.*Collision Detection And Penetration Depth Calculation In Virtual Surgical Simulation*. US: ProQuest LLC.

Putra, Darma.2010.*Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.

Saldana, Marie.2014.*An Integrated Approach to the Procedural Modeling of Ancient Cities and Buildings*.

Satibi, Syaifudin. 2013. *Ali-Imran:96-97 (Makkah yang Penuh Barakah)*.
<http://saifuddinasm.com/2013/02/13/ali-imran96-97-makkah-yang-penuh-barakah/>. Diakses tanggal 30 Mei 2015 pukul 19.45 WIB.

Stava, Ondrej, Bedric Benes, R. Mech, Daniel G. Aliaga and Peter Kristof.2010.*Inverse Procedural Modeling by Automatic Generation of L-systems*.USA: Computer Graphics Forum.

Takase, Y, N. Sho, A.Sone and K. Shimiya. 2003.*Automatic Generation of 3D City Models and Related Application*. Tokyo, Japan: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

Talton, Jerry O., Yu Lou, Steve Lesser, Jared Duke, Radomir Mech and Vladlen Koltun.2011.*Metropolis Procedural Modeling*.USA: ACM Transactions on Graphics

Tarecha, Agung dan Cahyo Crys dian.2013.*Visualisasi 3D Rupa Bumi Berbasis Data GDEM ASTER 30 meter*.Surabaya: Seminar Nasional Matematika

<http://www.mathisfun.com/>. Diakses pada tanggal 30 Mei 2015

<http://www.regentsprep.org/>. Diakses pada tanggal 30 Mei 2015