

# VISUALISASI 3D RUPA BUMI BERBASIS DATA GDEM ASTER 30 METER

Mochamad Agung Tarecha<sup>1)</sup>, Cahyo Crysdian<sup>2)</sup>

<sup>1)2)</sup>Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, UIN Maulana Malik Ibrahim  
Jalan Gajayana 50, Malang

<sup>1)</sup>agung.tarecha@gmail.com

<sup>2)</sup>crysdian@yahoo.com

**Abstract**— Penggunaan informasi spasial sangat vital untuk mendukung kegiatan perencanaan pembangunan, pemanfaatan sumber daya alam, militer, strategi penanggulangan bencana dan lain sebagainya. Informasi spasial umumnya divisualisasikan dalam bentuk peta 2D, pendekatan ini sulit untuk memperoleh detail kontur suatu wilayah. Oleh karena itu, visualisasi 3D rupa bumi dewasa ini diperlukan guna mempermudah dalam mendapatkan informasi mengenai kontur tanah. Visualisasi 3D rupa bumi membutuhkan data *Global Digital Elevation Model (GDEM)* yang merupakan data ketinggian tanah yang membentuk kontur bumi. Salah satu data *GDEM* tersebut adalah *ASTER* yang memiliki ketelitian 30 meter. Data *GDEM ASTER* tersebut diolah menggunakan beberapa metode dan algoritma antara lain 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. Hasil akhirnya sebuah aplikasi visualisasi yang dapat digunakan pada berbagai perangkat.

**Keywords**— visualisasi 3D, rupa bumi, *GDEM ASTER*.

## I. PENDAHULUAN

Saat ini informasi mengenai data spasial (referensi ruang kebumihan / georeference) banyak dibutuhkan di berbagai bidang. Sebagai contoh informasi data spasial dibutuhkan dalam perencanaan pembangunan, pemanfaatan sumber daya alam, militer, dan strategi penanggulangan bencana. Salah satu contoh adalah strategi penanggulangan bencana pesawat Sukhoi Super Jet 100 yang menabrak tebing Gunung Salak pada tanggal 9 Mei 2012, informasi mengenai kontur Gunung Salak diperlukan oleh Tim SAR guna

perencanaan evakuasi korban pesawat Sukhoi Super Jet 100 di tebing gunung yang terjal.

Informasi data spasial yang tersaji diharapkan dapat dicerna dengan mudah oleh pengguna, informasi tersebut biasanya ditampilkan dalam peta 2D, untuk menyajikan data berupa ketinggian tanah digunakan peta topografi yang memiliki garis kontur yang menyatakan ketinggian tanah pada garis yang sama. Namun sejalan dengan perkembangan teknologi, visual 3D dapat memberikan informasi yang lebih mudah dicerna oleh pengguna. Informasi rupa bumi haruslah sesuai atau paling tidak mendekati bentuk rupa bumi yang sebenarnya, oleh karena itu dibutuhkan sebuah informasi yang konsisten untuk mendukung berbagai kebutuhan yang telah dipaparkan.

## II. TINJAUAN PUSTAKA

### A. Digital Elevation Model

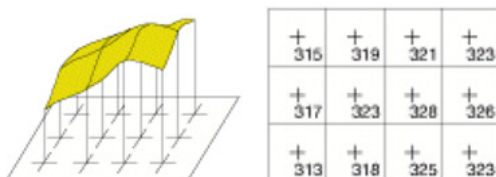
Sebuah model elevasi digital adalah model digital atau representasi 3D dari permukaan bumi yang dibuat dari data elevasi tanah. Menurut Jacobsen (2003) *Digital Elevation Model (DEM)* berdasarkan jumlah titik dengan x, y, dan z koordinat yang menggambarkan tanah gundul. DEM mungkin disusun dalam bentuk raster atau bentuk acak. Poros x dan y merupakan poros koordinat bumi, sedangkan poros z merupakan data ketinggian. DEM diperlukan untuk beberapa keperluan seperti generasi ortoimages, perencanaan penanggulangan banjir, pengendalian erosi, pertanian dan lain sebagainya.

### B. Model Data Raster

Dalam bentuk paling sederhana, *raster* terdiri dari matriks sel / piksel yang diatur dalam baris dan kolom (atau grid), di mana setiap sel berisi nilai yang mewakili informasi seperti suhu, ketinggian, dan lain sebagainya. Data *raster* dapat diperoleh melalui foto udara, citra satelit, gambar digital, remote sensing atau peta yang dipindai (ESRI, 2009).

Dalam set data raster setiap sel (yang juga dikenal sebagai piksel) memiliki nilai. Nilai-nilai sel merupakan fenomena digambarkan oleh data raster ditetapkan seperti kategori, magnitude, ketinggian, atau nilai spektral. Nilai data elevasi dapat disimpan dalam model data raster, dimana

ketinggian tiap titik direpresentasikan dalam bentuk nilai numerik. Untuk beberapa jenis data, nilai sel mewakili nilai yang diukur pada titik tengah sel. Sel tersebut memiliki posisi dan dikonversikan menjadi posisi baris dan kolom. Sel – sel tersebut disusun dalam suatu urutan yang membentuk data raster. Dalam penelitian ini tiap sel tersebut menyimpan data DEM. Contohnya adalah raster elevasi yang diilustrasikan melalui Gambar 1 berikut.



Gambar 1. Contoh raster elevasi. Sumber : Understanding Raster Data, ESRI.

Pada Gambar 1 sebuah data raster yang menyimpan nilai elevasi tanah. Posisi sel direpresentasikan dalam baris dan kolom, nilai sel merupakan data elevasi. Kumpulan sel tersebut membentuk matriks elevasi, dimana longitude direpresentasikan menjadi kolom matriks, dan latitude direpresentasikan menjadi baris matriks.

### C. Data GDEM ASTER

Salah satu data elevasi tanah dengan model data raster adalah data GDEM ASTER. Untuk membangun aplikasi visualisasi rupa bumi dibutuhkan data DEM, data GDEM ASTER merupakan data *Global Digital Elevation Model* (GDEM) berbasis model data raster. *The Ministry of Economy, Trade, and Industry (METI) of Japan* dan *the United States National Aeronautics and Space Administration (NASA)* bersama-sama mengumumkan rilis dari *Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) Global Digital Elevation Model (GDEM)* versi 2 pada 17 Oktober 2011 (LP DAAC, 2011). Cakupan GDEM ASTER membentang dari 83° lintang utara hingga 83° lintang selatan, meliputi 99 persen dari daratan bumi (Melebihi cakupan SRTM dari 60° lintang utara hingga 56° lintang selatan). Karakteristik data GDEM ASTER ditunjukkan pada tabel 1 berikut.

TABEL 1. KARAKTERISTIK GDEM ASTER

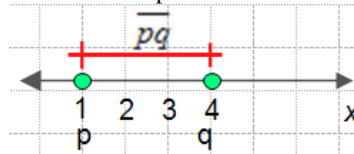
|                            |  |
|----------------------------|--|
| Ukuran ubin                | 3601 x 3601 ( 1° x 1°)   |
| Ukuran piksel              | 1 arc-second   |
| Sistem Koordinat Geografis | Geografis lintang dan bujur  |
| Format Output DEM          | GeoTIFF, signed 16-bit, vertikal dalam satuan meter<br>Direferensikan ke the WGS84/EGM96 geoid |

|  |  |
|--|--|
| Nilai spesial<br><i>Digital Number</i> | -9999 untuk piksel kosong dan 0 untuk permukaan laut |
| Cakupan                                | Utara 83° sampai selatan 83°, 22.702 ubin            |

(Sumber : ASTER GDEM 2 README)

### D. Data GDEM ASTER

*Euclidean distance* adalah sebuah metode untuk mengukur panjang antara dua titik, *euclidean distance* antara  $p$  dan  $q$  adalah panjang ruas garis yang menghubungkan mereka ( $pq$ ) (Wikipedia). Perhitungan *euclidean distance* satu dimensi diilustrasikan pada Gambar 2 berikut.



Gambar 2. Ilustrasi *euclidean distance* satu dimensi.

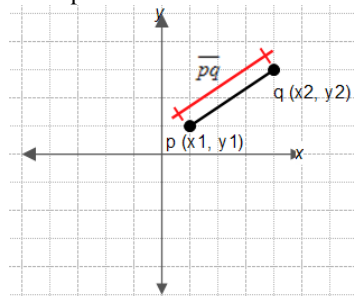
Perhitungan jarak titik menghitung jarak tiap poros  $x$  dan  $y$  satu persatu maka menggunakan rumus *euclidean distance* satu dimensi yang ditunjukkan pada Equation 1 berikut.

$$\begin{aligned} dx &= |x_1 - x_2| \\ dy &= |y_1 - y_2| \end{aligned} \quad (1)$$

dimana:

$dx$  = distance / jarak  $x_1$  ke  $x_2$   
 $dy$  = distance / jarak  $y_1$  ke  $y_2$   
 $x_1$  = titik koordinat  $p$  pada poros  $x$   
 $x_2$  = titik koordinat  $q$  pada poros  $x$   
 $y_1$  = titik koordinat  $p$  pada poros  $y$   
 $y_2$  = titik koordinat  $q$  pada poros  $y$

Perhitungan *euclidean distance* dua dimensi diilustrasikan pada Gambar 3 berikut.



Gambar 3. Ilustrasi *euclidean distance* dua dimensi.

Untuk perhitungan jarak titik dua dimensi dimana menghitung kedua titik pada poros  $x$  dan  $y$  secara bersamaan maka menggunakan rumus *euclidean distance* yang ditunjukkan pada Equation 2 berikut.

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

dimana :

$d(p, q)$  = distance / jarak

$x1$  = titik koordinat  $p$  pada poros  $x$   
 $x2$  = titik koordinat  $q$  pada poros  $x$   
 $y1$  = titik koordinat  $p$  pada poros  $y$   
 $y2$  = titik koordinat  $q$  pada poros  $y$

#### E. Algoritma Rounding

Menurut Maxfield (2006) bahwa fakta paling mendasar terkait dengan pembulatan adalah mengubah beberapa kuantitas dari presisi yang lebih tinggi ke presisi yang lebih rendah. Misal pengukuran panjang pensil menggunakan jangka sorong mendapat hasil 10,967 cm bisa dibulatkan menjadi 11 cm. Ada beberapa jenis algoritma pembulatan ini, dapat dipilih yang sesuai untuk meminimalkan efek kehilangan presisi. Algoritma tersebut antara lain *round-up*, *round-down*, *round-toward-nearest*, *arithmetic rounding*, *round-half-up*, *round-half-down*, *round-half-even*, *round-half-odd*, *round-toward-zero*, *round-away-from-zero*, *round-ceiling*, *round-floor*, *truncation*, *round-alternate*, dan *round-random*. Dalam penelitian ini hanya menggunakan 3 algoritma pembulatan yaitu *round-ceiling*, *round-floor*, dan *round-half-down*. Untuk penggunaan algoritma *round-half-down* menggunakan *euclidean distance* satu dimensi dalam proses penentuan titik terdekatnya.

#### F. Algoritma Brute Force Closest-Pair

Menurut Levitin (2012) algoritma *brute force closest-pair* adalah menghitung jarak antara setiap pasang titik yang berbeda dan menemukan sepasang dengan jarak terkecil, dan tidak menghitung pasangan yang sama dua kali, untuk menghindarinya hanya pasangan titik  $(p_i, p_j)$  yang  $i < j$ . Diasumsikan bahwa titik-titik tersebut ditetapkan secara standar oleh titik  $(x, y)$  pada koordinat kartesius dan bahwa jarak antara dua titik  $p_i (x_i, y_i)$  dan  $p_j (x_j, y_j)$  adalah *euclidean distance* dua dimensi.

*Pseudocode* dari algoritma *brute force closest-pair* adalah ditunjukkan pada Gambar 4 berikut.

```

algorithm BruteForceClosestPair(P)
//Finds distance between two closest
points in the plane by brute force
//Input: A list P of n (n ≥ 2) points
p1(x1, y1), . . . , pn(xn, yn)
//Output: The distance between the closest
pair of points
d ← ∞
for i ← 1 to n-1 do
  for j ← i + 1 to n do
    d ← min(d, sqrt((xi - xj)2 + (yi - yj)2))
return d

```

Gambar 4. Pseudocode algoritma *brute force closest-pair*.

#### G. Forward Chaining

Menurut Russell (2003) *forward chaining* adalah salah satu metode penalaran saat menggunakan *inference rule* dan dapat dijelaskan secara logis sebagai aplikasi modus ponens. *Forward chaining* dimulai dengan data yang tersedia dan menggunakan aturan inferensi untuk mengekstrak lebih banyak data sampai tujuan

tercapai. Sebuah mesin inferensi yang menggunakan *forward chaining* diterapkan untuk orde pertama klausa dan dapat diimplementasikan secara efisien. Klausa seperti situasi  $\rightarrow$  respon (jika situasi a maka respon b) berguna untuk sistem yang membuat kesimpulan untuk menanggapi informasi yang diberikan. Inferensi dimulai dari keadaan awal yaitu informasi atau fakta yang ada kemudian dicoba untuk dicocokkan dengan basis pengetahuan kemudian diperoleh tujuan yang diharapkan. *Forward Chaining* digunakan untuk menyusun empat ubin data yang berdekatan dengan titik tengah visualisasi.

#### H. Koreksi Tampilan

Menurut Crysdian (2009) ketika sebuah data DEM divisualisasikan terdapat masalah yaitu satuan pada poros  $x$  dan  $y$  berbeda dengan skala pada poros  $z$ . Korelasi tidak seimbang antara koordinat horizontal yang direpresentasikan oleh  $x$ - $y$  ke vertikal koordinat yang direpresentasikan oleh  $z$ . Tampak bahwa nilai vertikal mendominasi nilai horizontal karena sumbu poros tersebut tidak sama satuannya. Pada data *GDEM ASTER* interval tiap titik tanpa interpolasi pada poros  $x$  dan  $y$  adalah 30 meter, sedangkan interval tiap titik pada poros  $z$  adalah 1 meter. Matriks data tersebut dikonversikan baris menjadi poros  $x$ , kolom menjadi poros  $y$ , isi tiap sel matriks menjadi poros  $z$ .

Formula dari penyamaan satuan interval ditunjukkan pada Equation 3 berikut.

$$z_{\max} = \begin{cases} \text{interval} * (x_{\max} - x_{\min}) - 1 & \leftrightarrow x_{\max} \geq y_{\max} \\ \text{interval} * (y_{\max} - y_{\min}) - 1 & \leftrightarrow x_{\max} < y_{\max} \end{cases}$$

dimana:

$x_{\min}$  = 1 (index pertama matriks)  
 $y_{\min}$  = 1 (index pertama matriks)  
 $x_{\max}$  = panjang baris matriks  
 $y_{\max}$  = panjang kolom matriks  
 $-1$  = karena visualisasi dimulai pada ketinggian 0 pada poros  $z$ .

#### I. Interpolasi Linier

Menurut Pujiyanta (2007) interpolasi digunakan untuk memperkirakan nilai (tengah) di antara titik-titik dari satu set nilai yang sudah diketahui. Dalam arti yang lebih luas, interpolasi merupakan upaya mendefinisikan suatu fungsi dekat suatu fungsi analitik yang tidak diketahui atau pengganti fungsi rumit yang tak mungkin diperoleh persamaan analitiknya. Dalam penelitian ini interpolasi digunakan untuk memperhalus kontur tanah karena data yang sebelumnya memiliki interval antar titik 30 meter menjadi 15 meter dengan penambahan titik elevasi ditengah-tengah titik elevasi yang ada sehingga intervalnya menjadi semakin pendek mengakibatkan tingkat kepresisiannya meningkat. Misalnya

menempatkan objek berupa pohon pada area visualisasi, untuk mencari titik terdekat posisi pohon akan lebih presisi pada interval 15 meter daripada 30 meter. Interpolasi dapat ditingkatkan lagi menjadi 7.5 meter, 3.75 meter dan seterusnya namun dengan konsekuensi data yang akan diproses semakin besar dan memerlukan sumber daya komputer yang lebih besar sehingga mengakibatkan waktu untuk memproses data lebih lama, oleh karena itu pada penelitian ini dibatasi interpolasi hanya sampai 15 meter mengingat sumber daya komputer yang terbatas.

### J. Polygonal Modelling

Menurut Crysdian (2011) pendekatan *polygonal modeling* adalah metode sederhana namun efektif untuk membangun model kota menggunakan visualisasi 3D. Dengan menggunakan pendekatan ini berbagai fitur kota seperti bangunan, vegetasi, dan jalan dapat dikembangkan dengan cara yang efisien berdasarkan bentuk tepi objek. Informasi spasial yang kompleks biasanya meliputi perkembangan model 3D kota dapat dikelola secara sederhana. Penelitian ini akan mengimplementasikan *polygonal modeling* yang telah diteliti sebelumnya, objek yang akan divisualisasikan adalah pohon. Tujuan implementasi visualisasi pohon ini adalah membuktikan bagaimana *interpolasi* memberikan dampak visualisasi menjadi semakin detail dan akurat. Perbedaan akan terlihat ketika objek pohon divisualisasikan sebelum dan setelah interpolasi. Objek pohon ditunjukkan pada Gambar 5 berikut.



Gambar 5. Objek pohon. Sumber: *Development of 3D City Model using Polygonal Modeling Approach*, Crysdian

## III. METODE PENELITIAN

### A. Design Sistem

Aplikasi yang dibangun adalah aplikasi visualisasi 3D rupa bumi berbasis data *GDEM ASTER* yang merupakan produk dari METI dan NASA. Untuk memudahkan akses oleh pengguna maka visualisasi diwujudkan dalam bentuk aplikasi web sehingga bisa diakses kapan saja dan di mana saja selama memiliki koneksi dengan *server* dalam hal ini bisa menggunakan *Local Area Network (LAN)* atau jika ingin lebih luas cakupan penggunaannya dapat menggunakan *internet*, hal ini dapat dipilih sesuai dengan kebutuhan. Dengan menggunakan antar muka halaman web maka pengguna tidak perlu menginstall aplikasi visualisasi di komputer,

cukup menggunakan *web browser* yang umumnya sudah terinstal di komputer. Pengguna mengakses alamat *web server* untuk mengakses visualisasi, dalam *web server* terdapat aplikasi web berbasis ASP .Net dan C# .Net sebagai *front-end layer* dan *MATLAB Compiler Runtime* sebagai *back-end layer*.

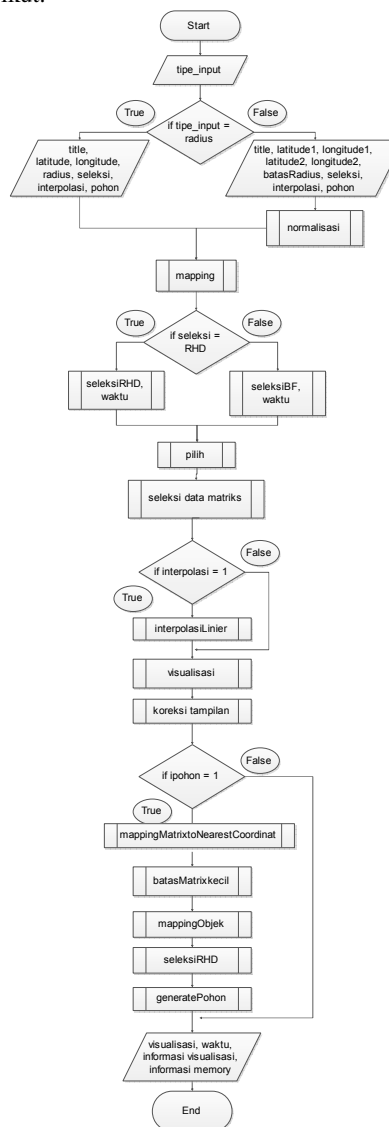
Desain dari perancangan sistem dapat dilihat pada Gambar 6 berikut.



Gambar 6. Design sistem

### B. Design Algoritma

Berikut adalah diagram alir dari design algoritma yang digunakan untuk membangun sebuah aplikasi visualisasi 3D rupa bumi. Diagram alir sistem secara garis besar ditunjukkan Gambar 7 berikut.



Gambar 7. Diagram alir sistem secara garis besar.

Penjelasan diagram alir sistem sebagai berikut

1) *Tipe input*: Untuk melakukan visualisasi pengguna dapat memilih menggunakan input data koordinat bumi satu titik tengah dengan radius visualisasi atau dengan dua titik tepi, pemilihan ini disimbolkan dengan *variable* tipe\_input. Bila menggunakan titik tepi kemudian dinormalisasikan terlebih dahulu yaitu mengubah dua titik tepi menjadi sebuah titik tengah dengan radius.

2) *Mapping*: Data input berupa *latitude-longitude* titik tengah visualisasi tersebut kemudian dipetakan menjadi baris kolom matriks.

3) *Seleksi*: Setelah mendapatkan baris kolom matriks dari proses *mapping* karena nilai baris kolom tersebut masih mengandung nilai pecahan dimana untuk seleksi data membutuhkan referensi titik tengah berupa nilai bulat (baris kolom matriks bernilai *integer*) maka diseleksi dahulu untuk mencari pasangan titik terdekatnya, proses ini membandingkan pencarian pasangan titik terdekat antara algoritma *round-half-down* dan *brute force closest-pair* mana yang lebih cepat dan tepat.

4) *Pilih*: Untuk memvisualisasikan wilayah yang terbagi dalam ubin data yang berbeda maka fungsi pilih mengimplementasikan *forward chaining* untuk menyusun empat ubin data yang berdekatan dengan titik tengah visualisasi.

5) *Seleksi data matriks*: Seleksi data matriks dari keempat ubin data yang telah disusun diperlukan untuk memvisualisasikan wilayah yang didefinisikan oleh data titik tengah atau titik tepi yang telah diinputkan.

6) *Interpolasi*: Interpolasi linier digunakan untuk menambahkan titik data ketinggian diantara titik ketinggian yang telah diketahui guna meningkatkan ketelitian visualisasi.

7) *Visualisasi*: Data yang telah diseleksi kemudian divisualisasikan menggunakan fungsi dasar MATLAB yaitu *mesh*.

8) *Koreksi tampilan*: Koreksi tampilan digunakan agar visual yang dihasilkan proporsional antara poros *x-y-z*.

9) *Generate pohon*: Visualisasi pohon digunakan untuk membuktikan bahwa dengan interpolasi ketelitian visualisasi meningkat. Visual yang dihasilkan akan berbeda sebelum dan setelah interpolasi.

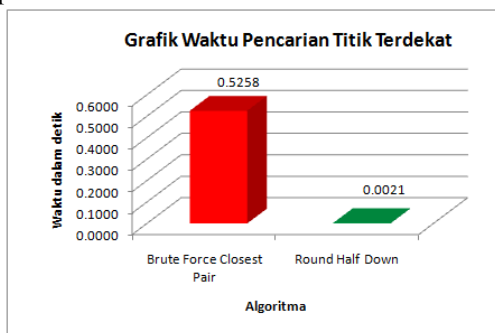
#### IV. HASIL DAN PEMBAHASAN

Setelah dilakukan pengujian terhadap hasil penelitian ini didapatkan hasil sebagai berikut.

##### A. Pengujian Ketepatan dan Kecepatan Pencarian Titik Tengah

Pengujian ini membandingkan implementasi antara algoritma *round-half-down* dan algoritma *brute force closest-pair* untuk mencari titik *latitude-longitude* dengan pasangan titik terdekat baris kolom matriks yang menjadi titik tengah wilayah visualisasi. Parameter yang dibandingkan adalah ketepatan dan kecepatan.

Pengujian dilakukan dengan melakukan visualisasi 5 buah titik lokasi dengan tiap titik menggunakan dua algoritma yaitu *round-half-down* (*rhd*) dan *brute force closest-pair* (*bf*). Nilai baris-kolom asli akan dibandingkan dengan nilai baris-kolom terdekat pada masing-masing algoritma diuji apakah tepat mencari titik terdekatnya. Setelah itu diuji berapa lama waktu yang dihabiskan untuk proses pencarian titik terdekat tersebut menggunakan kedua algoritma tersebut. Hasilnya adalah antara algoritma *round-half-down* dan algoritma *brute force closest-pair* menghasilkan titik yang tepat sama. Artinya menggunakan kedua algoritma tersebut akan menghasilkan gambar visual yang sama karena titik tengahnya sama. Sedangkan yang berbeda adalah waktu pencarian titik tengah terdekat tersebut. Perbedaan waktu tersebut ditunjukkan pada Gambar 8 berikut.



Gambar 8. Grafik waktu pencarian titik terdekat.

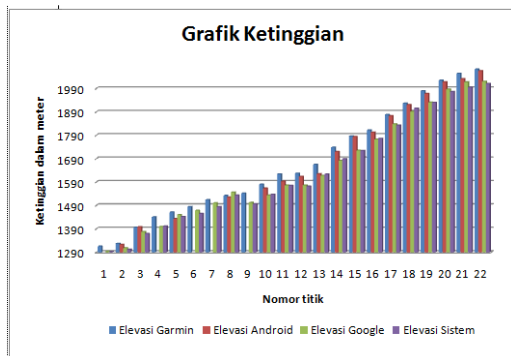
Seperti yang terlihat pada Gambar 8 Grafik waktu pencarian titik terdekat diatas bahwa bila dibandingkan maka perbandingan kedua algoritma tersebut adalah  $\frac{\text{waktu BF}}{\text{waktu RHD}} = \frac{0.5258}{0.0021} = 250.38$ , dapat disimpulkan bahwa algoritma *round-half-down* lebih cepat **250 x** daripada algoritma *brute force closest-pair*. Hal ini disebabkan karena algoritma *round-half-down* hanya menyeleksi titik yang sudah diprediksi sebagai pasangan terdekatnya yaitu batas bawah dan batas atas, sedangkan algoritma *brute force closest-pair* menyeleksi seluruh titik yang ada dan menghitung *euclidean distance* satu persatu. Kelemahan algoritma *round-half-down* dalam mencari titik terdekat adalah interval tiap titik tersebut harus sama dengan 1, algoritma tersebut tidak dapat digunakan untuk mencari titik terdekat bila interval antar titik tidak beraturan, sedangkan



algoritma *brute force closest-pair* menawarkan fleksibilitas dalam pencarian pasangan titik terdekat karena membandingkan *euclidean distance* terhadap seluruh titik yang akan dicari pasangan terdekatnya tanpa terpengaruh berapapun interval antar titiknya.

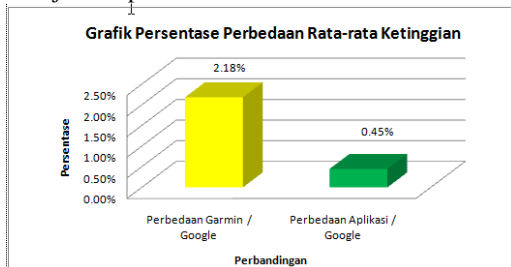
### B. Pengujian Ketelitian Titik Tengah

Pengujian ini membandingkan data ketinggian pada titik tengah wilayah visualisasi. Titik tengah ini memiliki nilai ketinggian yang akan dibandingkan dengan ketinggian GPS Garmin 80 Mil dan Google Earth pada titik yang sama, sedangkan GPS HP Android hanya digunakan sebagai data referensi saja untuk membuktikan bahwa pada lokasi yang sama sekalipun hasil pengukuran ketinggian menggunakan GPS memiliki hasil yang berbeda tergantung kualitas dan tingkat ketelitian masing-masing GPS. Setelah data tersebut diperoleh maka diperoleh nilai ketinggian dari masing-masing sumber tersebut kemudian dibandingkan satu sama lain dan persentase perbedaan ketinggian absolut tersebut dihitung kemudian dirata-rata, akan terlihat berapa persen tingkat ketelitian ketinggian. Dengan adanya pengujian ini maka dapat diketahui berapa persentase *error* ketelitian aplikasi visualisasi yang dibuat. Akuisisi data dilakukan di Gunung Panderman Batu pada tanggal 10 Pebruari 2013, hasil akuisisi data ditunjukkan pada Gambar 9 berikut.



Gambar 9. Grafik ketinggian.

Dari grafik ketinggian pada Gambar 10 diatas akan dibandingkan perbedaan absolut dari ketinggian masing-masing sumber data. Grafik persentase perbedaan ketinggian rata-rata ditunjukkan pada Gambar 10 berikut.

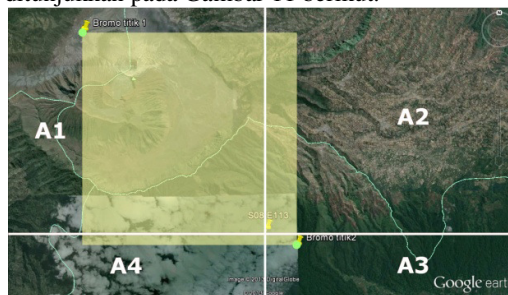


Gambar 10. Grafik persentase perbedaan rata-rata ketinggian.

Ketinggian yang dihasilkan pada titik yang sama antara aplikasi visualisasi dan Google Earth memiliki tingkat *error* yang kecil yaitu  $< 0,5\%$  bila dibandingkan dengan antara GPS Garmin dan Google Earth sebesar  $2,18\%$ . Hal ini membuktikan bahwa data satelit dalam hal ini *GDEM ASTER* yang digunakan pada aplikasi visualisasi dan *GDEM SRTM* yang digunakan pada Google Earth memiliki perbedaan yang relatif kecil bila dibandingkan menggunakan GPS Garmin. Keuntungan menggunakan data *GDEM ASTER* dibandingkan *GDEM SRTM* adalah resolusi *GDEM ASTER* lebih detail dan cakupan data lebih luas bila dibandingkan dengan *GDEM SRTM*. Beda antara GPS Garmin dan Google Earth cukup signifikan hal ini disebabkan oleh ketelitian GPS Garmin tersebut, jangkauan dengan Google Earth dengan GPS Android pun memiliki perbedaan ketinggian meski pada titik yang sama, hal ini dipengaruhi kualitas perangkat tersebut namun hal ini masih bisa dianggap wajar karena tiap perangkat pasti memiliki toleransi dalam pengukuran.

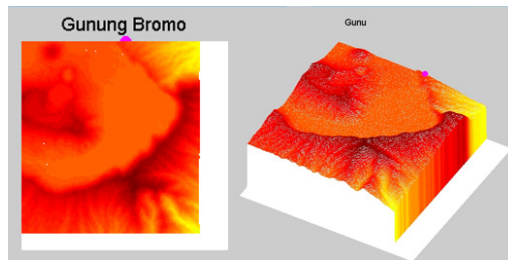
### C. Pengujian Ketepatan Penyusunan Empat Ubin Data Terdekat

Pengujian ketepatan penyusunan empat ubin data terdekat menguji apakah penggunaan metode *forward chaining* berhasil untuk menyusun empat ubin terdekat tersebut kemudian divisualisasikan. Cara mengujinya dengan memvisualisasikan titik tepi dimana masing-masing titik tersebut berada pada ubin data yang berbeda seperti yang ditunjukkan pada Gambar 11 berikut.



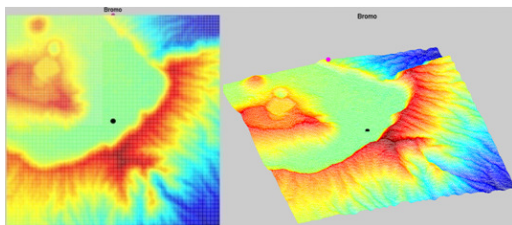
Gambar 11. Ilustrasi titik visualisasi di Gunung Bromo. Sumber: Google Earth.

Hasil pengujian tanpa penyusunan empat ubin data ditunjukkan pada Gambar 12 berikut.



Gambar 12. Visualisasi tanpa penyusunan empat ubin data.

Hasil pengujian tanpa penyusunan empat ubin data ditunjukkan pada Gambar 13 berikut.



Gambar 13. Visualisasi dengan penyusunan empat ubin data.

Jika tidak menggunakan *forward chaining* untuk menyusun empat ubin data yang berdekatan maka hasil visualisasi akan terpotong seperti yang terlihat pada Gambar 12. Jika menggunakan *forward chaining* untuk menyusun empat ubin data yang berdekatan, hasil visualisasi terlihat seperti yang ditunjukkan pada Gambar 13, gambar yang dihasilkan sesuai dengan kontur tanah yang semestinya. Artinya pada wilayah perpotongan data dari keempat ubin data tidak terdapat keganjilan kontur tanah, sehingga dapat disimpulkan bahwa metode *forward chaining* berhasil menyusun keempat ubin data terdekat.

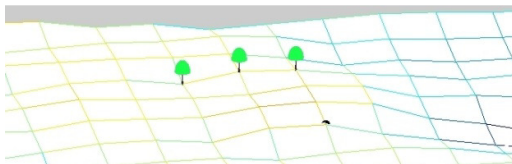
#### D. Pengujian Interpolasi Linier

Untuk menguji apakah interpolasi linier memberikan dampak meningkatnya ketelitian visualisasi maka dilakukan pengujian visualisasi pohon tanpa interpolasi dan dengan interpolasi berdasarkan koordinat *latitude-longitude* sebenarnya, akan terlihat bagaimana ketelitian meningkat dengan interpolasi. Pohon yang akan divisualisasikan ditunjukkan Gambar 14 berikut.



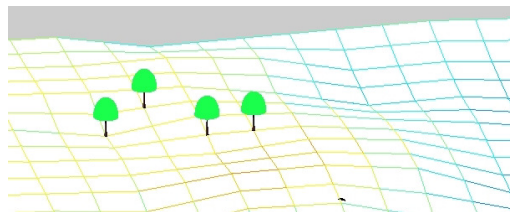
Gambar 14. Pohon yang akan divisualisasikan.

Hasil dari visualisasi tanpa interpolasi linier terlihat seperti pada Gambar 15 berikut.



Gambar 15. Visualisasi pohon tanpa interpolasi.

Hasil dari visualisasi dengan interpolasi linier terlihat seperti pada Gambar 16 berikut.

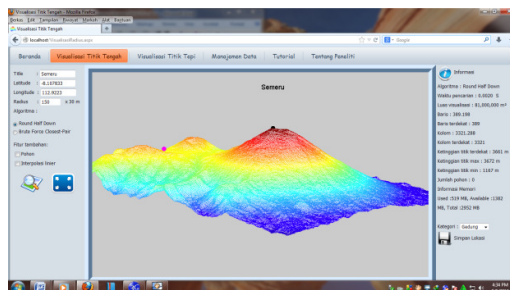


Gambar 16. Visualisasi pohon dengan interpolasi

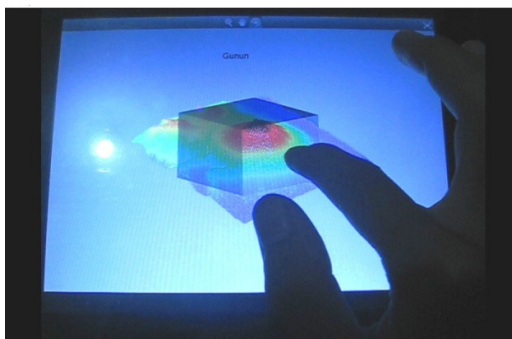
Jumlah pohon yang divisualisasikan diatas kontur tanah sebanyak empat buah namun terdapat perbedaan gambar yang dihasilkan, Gambar 15 visualisasi pohon tanpa interpolasi terlihat hanya tiga buah pohon saja, hal ini disebabkan karena jarak antar kedua pohon terlalu dekat sehingga ketika dicari titik elevasi terdekat dengan pohon menunjukkan titik yang sama dan gambar pohon tersebut tertumpuk pada titik tersebut. Jumlah pohon yang terlihat pada Gambar 16 visualisasi pohon dengan interpolasi terlihat sebanyak empat buah dan letak pohon tidak segaris lurus tapi letak tersebut lebih mirip dengan letak yang sebenarnya seperti yang terlihat pada Gambar 14. Dari perbedaan gambar yang dihasilkan tersebut dapat disimpulkan bahwa dengan interpolasi ketelitian visualisasi meningkat. Interpolasi linier terbukti mampu untuk meningkatkan ketelitian visualisasi, semakin banyak titik hasil interpolasi maka ketelitian penempatan objek pohon akan semakin sesuai dengan lokasi aslinya namun data yang akan diproses semakin besar, hal ini dapat menyebabkan proses untuk membangkitkan visual semakin lama. Nilai titik baru hasil interpolasi adalah hasil perhitungan matematis, bukan pengukuran sebenarnya dari ketinggian tanah, oleh karena itu interpolasi dapat digunakan seperlunya dengan mempertimbangan besarnya data yang akan diproses dan sumber daya yang tersedia.

#### E. Pengujian Kompatibilitas pada Perangkat

Pengujian kompatibilitas pada perangkat menguji apakah aplikasi visualisasi dapat diakses oleh banyak perangkat yang berbeda.



Gambar 17. Pengujian pada Windows 7.



Gambar 18. Pengujian pada Tablet Android.

## V. KESIMPULAN DAN SARAN

Kesimpulan yang diperoleh dari penelitian visualisasi 3D rupa bumi berbasis data *GDEM ASTER* adalah sebagai berikut:

1. Algoritma *round-half-down* dan *brute force closest-pair* dapat menemukan pasangan titik terdekat dengan tepat dan algoritma *round-half-down* lebih cepat daripada *brute force closest-pair*.
2. Metode *euclidean distance* dan *forward chaining* dapat digunakan untuk mendukung visualisasi dan memberikan hasil yang sesuai.
3. Perbedaan ketelitian elevasi tanah antara Google Earth yang menggunakan data *GDEM SRTM* 90 meter dan aplikasi visualisasi yang menggunakan data *GDEM ASTER* 30 meter memiliki perbedaan yang sangat kecil yaitu  $< 0,5 \%$  sehingga dapat disimpulkan bahwa aplikasi visualisasi lebih akurat bila dibandingkan dengan GPS Garmin 80 Mil sebesar 2.18 %.
4. Interpolasi terbukti mampu meningkatkan ketelitian visualisasi dibuktikan dengan peletakan objek berupa pohon.

Dalam penelitian ini masih terdapat beberapa kekurangan. Untuk pengembangan lebih lanjut terdapat saran-saran sebagai berikut ini.

1. Metode interpolasi dapat diperbanyak lagi seperti interpolasi *Nearest Neighbor*, interpolasi *Cubic Spline*, interpolasi kuadrat, dan lain sebagainya.
2. Visualisasi objek dapat diperbanyak lagi seperti visualisasi gedung, jalan, sungai, dan lain sebagainya.
3. Penelitian selanjutnya dapat ditambahkan simulasi bencana yang berkaitan dengan rupa bumi seperti gunung meletus, wilayah kebakaran hutan, dan lain sebagainya.

## DAFTAR PUSTAKA

- Anonim. 2009. *Understanding Raster Data*. New York: ESRI.
- Anonim. 2011. *ASTER GDEM 2 Readme*. United States of America: LP DAAC.
- Crysdian, Cahyo. 2009. *3D Visualization of Spatial Objects using Elevation Model*. Surabaya: seminar Intelligent Technology and It's Applications.
- Crysdian, Cahyo. 2011. *Development of 3D City Model using Polygonal Modeling Approach*. Yogyakarta : Conference on Information and Electrical Engineering (CITEE).
- Jacobsen, K. 2003. *DEM generation from satellite data*. Germany: University of Hannover.
- Levitin, Anany. 2012. *Introduction to The Design & Analysis of Algorithms 3<sup>rd</sup> Edition*. New Jersey: Pearson.
- Maxfield, Clive. 2006. *An introduction to different rounding algorithms*. (<http://www.eetimes.com/design/signal-processing-dsp/4004643/An-introduction-to-differentrounding-algorithms>, diunduh pada tanggal 28 Februari 2013)
- Pujiyanta, Ardi. 2007. *Komputasi Numerik dengan Matlab*. Yogyakarta: Graha Ilmu.
- Russell, Stuart J. dan Peter Norvig. 2003. *Artificial Intelligence A Modern Approach Second Edition*. New Jersey: Pearson.
- [http://en.wikipedia.org/wiki/Color\\_mapping](http://en.wikipedia.org/wiki/Color_mapping) (diunduh pada tanggal 21 Januari 2013).
- <http://www.islamquest.net/id/archive/question/fa4232> (diunduh pada tanggal 9 Juni 2013).
- <http://arduino.cc/en/Reference/map> (diunduh pada tanggal 4 Juni 2013).
- [http://en.wikipedia.org/wiki/Wire-frame\\_model](http://en.wikipedia.org/wiki/Wire-frame_model) (diunduh pada tanggal 20 April 2013)
- [http://en.wikipedia.org/wiki/Advanced\\_Spaceborne\\_Thermal\\_Emission\\_and\\_Reflection\\_Radiometer](http://en.wikipedia.org/wiki/Advanced_Spaceborne_Thermal_Emission_and_Reflection_Radiometer) (diunduh pada tanggal 21 Januari 2013).
- [http://en.wikipedia.org/wiki/Shuttle\\_Radar\\_Topography\\_Mission](http://en.wikipedia.org/wiki/Shuttle_Radar_Topography_Mission) (diunduh pada tanggal 13 Januari 2013).
- [https://en.wikipedia.org/wiki/Google\\_Earth](https://en.wikipedia.org/wiki/Google_Earth) (diunduh pada tanggal 13 Januari 2013).
- <http://www.beritabogor.com/2012/05/sukhoi-superjet-100-jatuh-di-gunung.html> (diunduh pada tanggal 12 Agustus 2012).