

BAB IV

IMPLEMENTASI DAN EVALUASI

1.1 Kebutuhan Sistem

Untuk dapat menjalankan sistem ini maka diperlukan perangkat keras dan perangkat lunak yang dengan kondisi dan persyaratan tertentu agar sistem dapat berjalan dengan baik. Adapun kebutuhan perangkat keras dan perangkat lunak adalah sebagai berikut.

1.1.1 Kebutuhan Perangkat Keras

Sistem ini nantinya akan digunakan pada laptop maupun desktop milik pegawai maupun staff IT Dinas Pertanian Propinsi Jawa Timur. Adapun syarat kebutuhan dari hardware yang harus dipenuhi adalah:

1. *Processor* Intel Pentium IV, sejenis atau di atasnya
2. *Memory* 768 Mb atau lebih
3. *Harddisk* 80 Gb atau lebih
4. *VGA Card* dengan RAM 512Mb *support OpenGL*
5. *Printer, Mouse dan Keyboard*

1.1.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah:

1. *Operating System* Windows 7.
2. Basis data untuk pengolahan data menggunakan *Sqlite*.
3. Qt Sebagai IDE dalam membuat aplikasi.

4. Untuk perancangan sistem menggunakan *Power Designer* 15.0
5. *GDAL Source development kit* untuk memproses *file* DTED dan SHP.
6. *OpenCV Source development kit* untuk melakukan *Image Processing*.

1.2 Pembuatan Program

Program atau aplikasi ini dibuat dengan menggunakan bahasa pemrograman C++, XML, dengan menggunakan *Framework* Qt. Alasan penggunaan C++ sebagai bahasa pemrograman adalah karena aplikasi yang dibangun dengan C++ memiliki performa yang sangat cepat. Hal itu karena hasil kompilasi dari C++ adalah *Native Application* yaitu aplikasi yang berjalan tanpa memerlukan *intepreter* seperti aplikasi dengan menggunakan VB. Dalam menampilkan grafis aplikasi ini menggunakan OpenGL sebagai *library* untuk menampilkan grafis, karena OpenGL dapat menampilkan grafis dalam bentuk 2D dan 3D dengan memanfaatkan *video card* sehingga hasil yang ditampilkan akan sangat jelas dan cepat. Aplikasi ini mengolah *file* dengan format DTED dan SHP dimana *file-file* tersebut adalah format untuk GIS. Oleh karena itu aplikasi ini menggunakan *library* dari GDAL, yaitu sebuah *library* yang khusus menangani *file-file* dari GIS. Selain itu *Library* ini bersifat *Open Source* sehingga tidak diperlukan biaya lisensi untuk menggunakannya. Sementara untuk *Image Processing* aplikasi ini menggunakan OpenCV, karena di dalam OpenCV sudah terdapat banyak fitur *Image Processing* yang dapat dimanfaatkan untuk membantu jalannya aplikasi ini.

Dengan menggunakan bahasa C++ nantinya aplikasi ini akan bersifat *native application*. Artinya *program* dapat langsung berjalan tanpa diperlukan program tambahan seperti *intepreter*. Aplikasi ini juga didesain *portable* karena

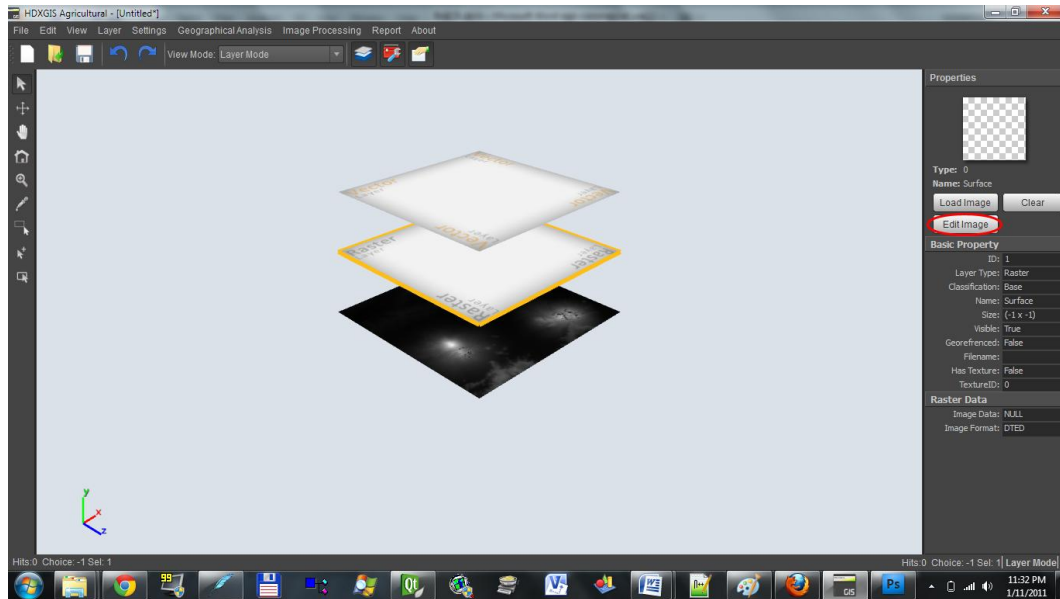
semua *resource* dibuat dalam satu lokasi, sehingga tidak menyulitkan *user* dalam proses instalasi dan penggunaannya. Selain itu *native application* berjalan lebih cepat dari aplikasi *interpreter*.

1.3 Implementasi Sistem

Setelah kebutuhan perangkat keras dan perangkat lunak telah terpenuhi, maka tahap selanjutnya adalah melakukan implementasi sistem yang telah dibuat. Aplikasi ini terdiri dari 4 *view* yang berbeda, dimana tiap *view* memiliki fungsi dan kegunaan yang berbeda-beda. Aplikasi ini juga menerapkan *Single Document Interface* yang artinya hanya satu buah *project* atau dokumen yang dapat dioalah dalam satu waktu. Hal ini karena alasan *resource* yang digunakan untuk mengolah satu dokumen cukup besar sehingga akan sangat memakan banyak *resource* apabila diterapkan *Multiple Document Interface*.

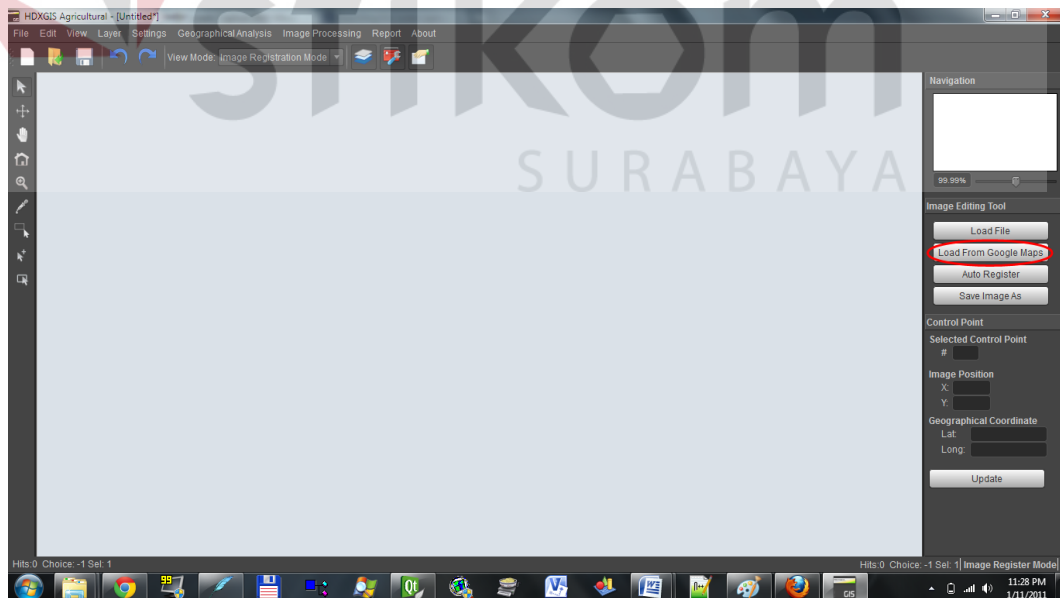
1.3.1 Implementasi Metode Penggunaan *GoogleMaps*

Penggunaan *GoogleMaps* mewajibkan menggunakan internet untuk melakukan proses *download* gambar dari website milik google. Fitur *GoogleMaps* terdapat pada *view image register mode*, sehingga hanya *layer* dengan tipe *raster* yang dapat menggunakan fitur ini. Untuk masuk ke dalam *view image register*, *user* pertama-tama harus menekan tombol *edit image* pada *property window* pada *layer list view* seperti yang dapat dilihat pada Gambar 4.1. Kemudian untuk melakukan proses *download* gambar dari *google maps*, *user* harus memilih tombol *Load from google maps*. Tombol untuk melakukan proses *download* dari *google maps* dapat dilihat pada Gambar 4.2.



Gambar 4.1 Tampilan *Layer Editor*

Setelah bagian tersebut dipilih maka akan muncul tampilan dari *image register editor* yang dapat dilihat pada Gambar 4.2. Untuk mengaktifkan *GoogleMaps Mode*, user harus menekan tombol *Load From Google Maps*.



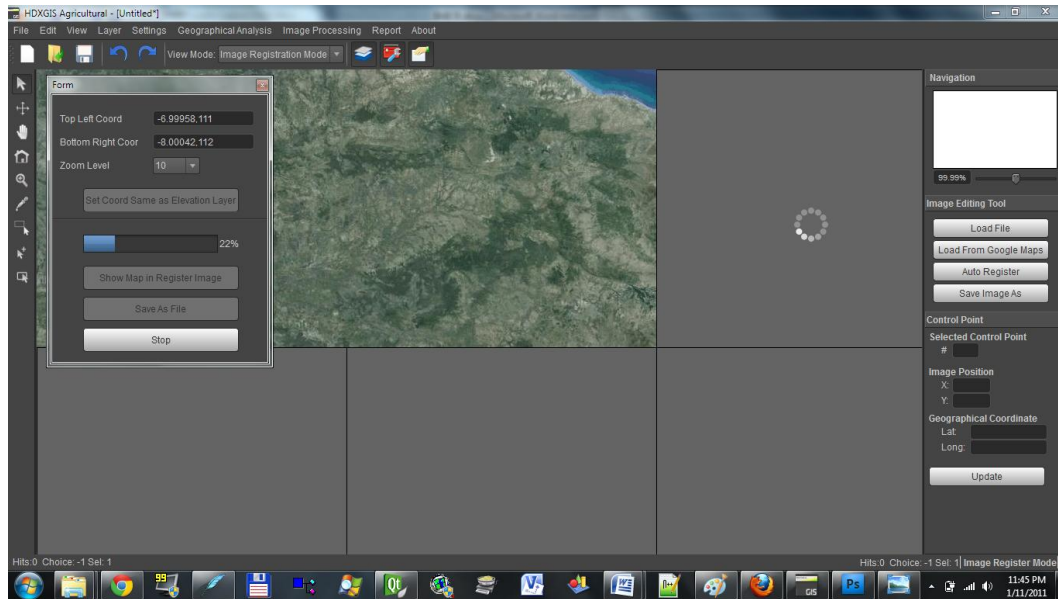
Gambar 4.2 Tampilan Dari Tombol *Load Google Maps*

Apabila *View Image Register* telah masuk ke dalam mode GoogleMaps akan muncul sebuah *Form* untuk melakukan input data yang diperlukan untuk mendownload data dari GoogleMaps yaitu titik koordinat. Apabila *user* ingin mendownload data dengan daerah yang sama dengan data ketinggian, maka *user* dapat membuat data yang akan didownload sama dengan data elevasi dengan cara menekan tombol *Set Coord Same as elevation*. Seperti yang dapat dilihat pada Gambar 4.3.



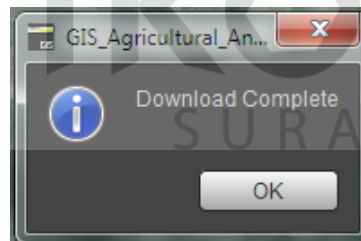
Gambar 4.3 Tampilan Tombol *Set Coord as Same as Elevation*

Penerapan *block-block matrix* ditampilkan dalam bentuk kotak seperti yang dapat dilihat pada proses melakukan *download*. Animasi ditunjukkan dengan gambar berputar menandakan bahwa daerah tersebut sedang dalam proses *download*. Apabila *block* sudah selesai di *download* maka yang ditampilkan adalah hasil *download* yaitu berupa gambar peta. Penerapan blok untuk *download GoogleMaps* dapat dilihat pada Gambar 4.4. di halaman 103.



Gambar 4.4 Tampilan Blok Proses *Download GoogleMaps*

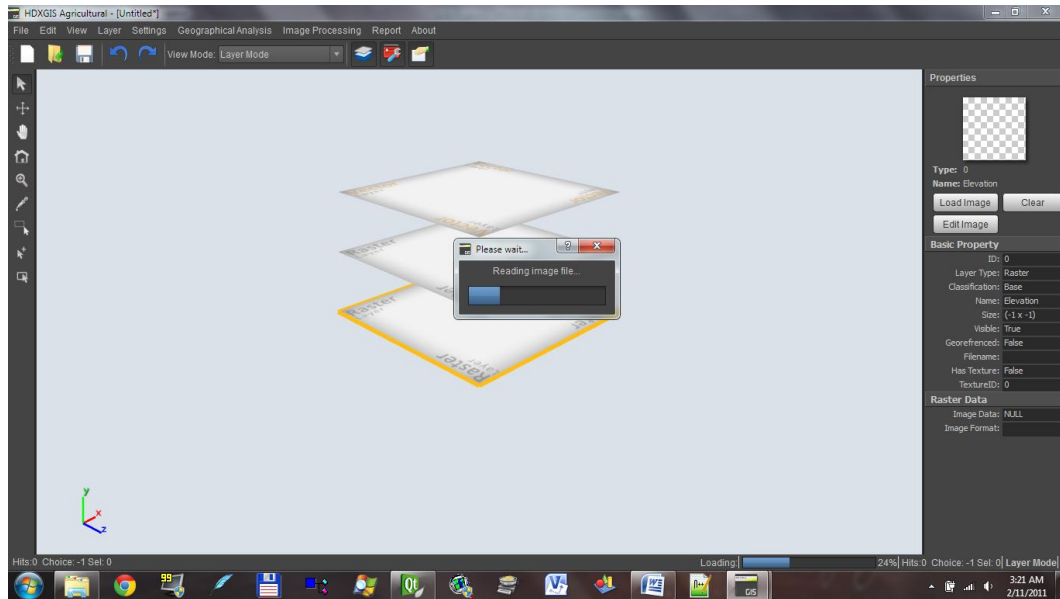
Apabila proses download dari Google maps telah selesai yang artinya semua blok yang kita inginkan telah selesai men-*download* data maka akan muncul pesan informasi yang memberitahu bahwa proses *download* telah selesai. Tampilan dari *messagebox* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Pesan *Download* Telah Berhasil

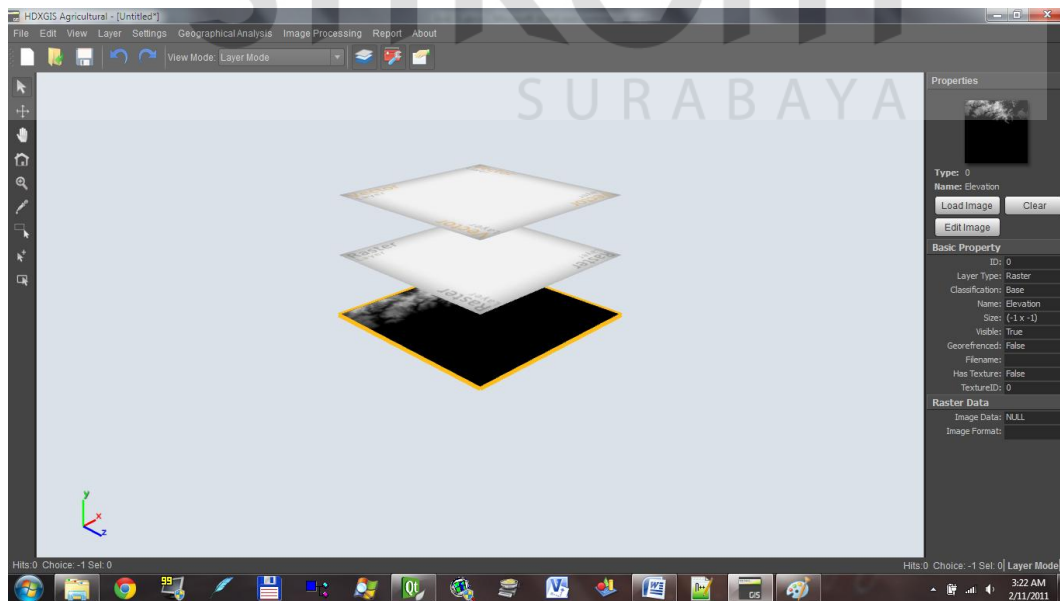
1.3.2 Implementasi Metode *Generate Terrain* Tiga Dimensi

Implementasi *Generate Terrain* tiga dimensi diawali dengan menambah atau memberi data pada data *elevation*. Proses penambahan data elevasi dapat dilihat pada Gambar 4.6. di halaman 104.



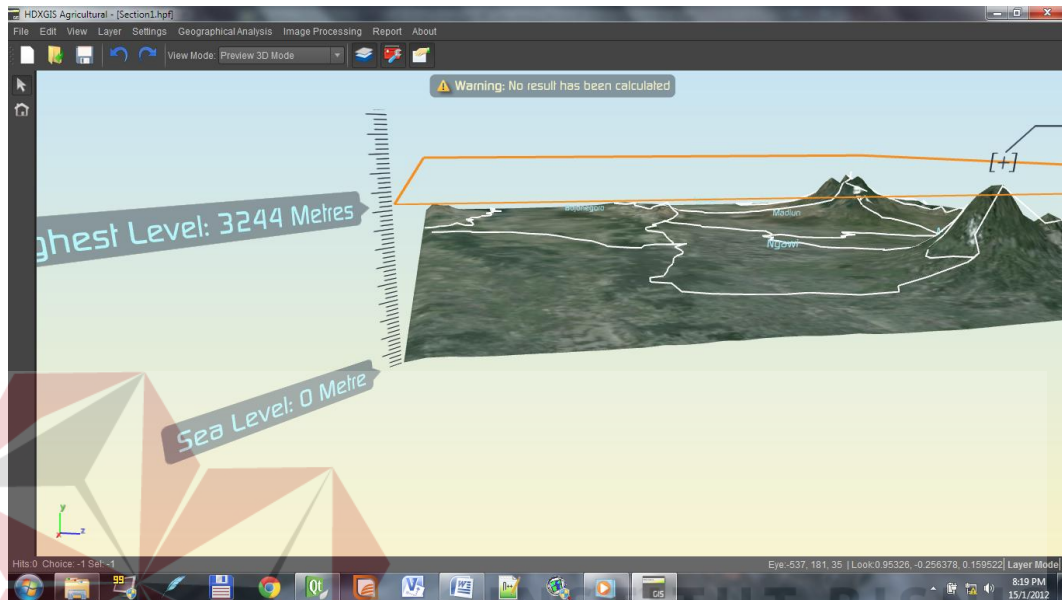
Gambar 4.6 Tampilan Proses *Loading* Dari *Project*

Pada saat proses penambahan, sistem secara otomatis akan membuat sebuah *terrain* 3D yang ditandai dengan adanya progress dialog. Ketika proses untuk membaca dan merubah data menjadi *terrain* selesai maka *form* loading akan hilang dan *layer* akan memiliki gambar dari Layer, seperti pada Gambar 4.7.



Gambar 4.7 Tampilan *Layer* Setelah Selesai *Loading*

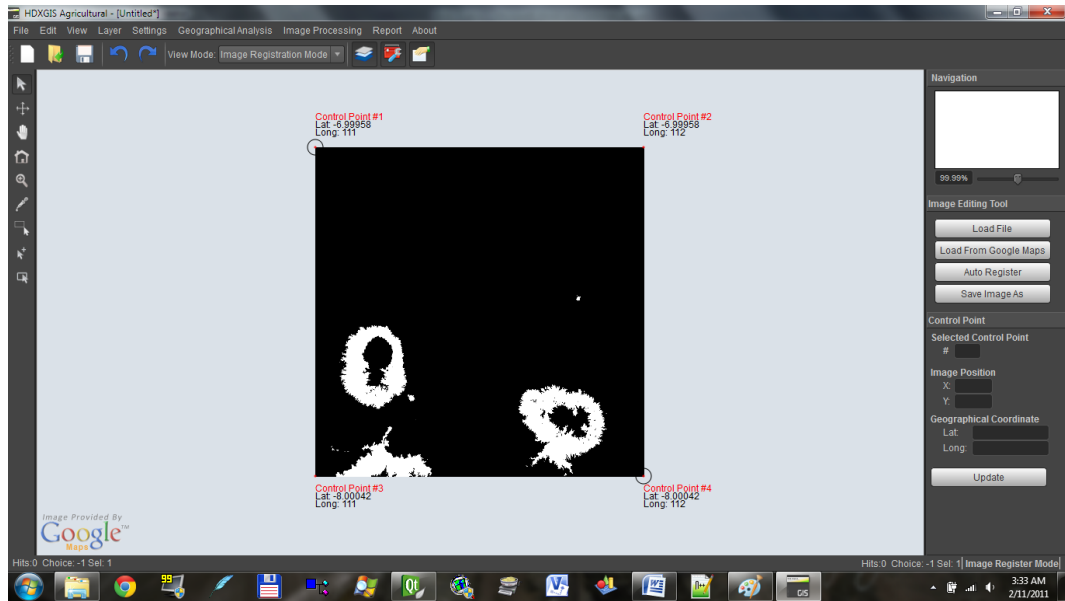
Untuk dapat melihat hasil dari proses tiga dimensi, *user* harus mengganti menu menjadi *preview result*. Akan muncul sebuah terrain 3D yang dibuat berdasarkan data ketinggian. Tampilan aplikasi dapat dilihat pada Gambar 4.8



Gambar 4.8 Tampilan *Terrain* Tiga Dimensi

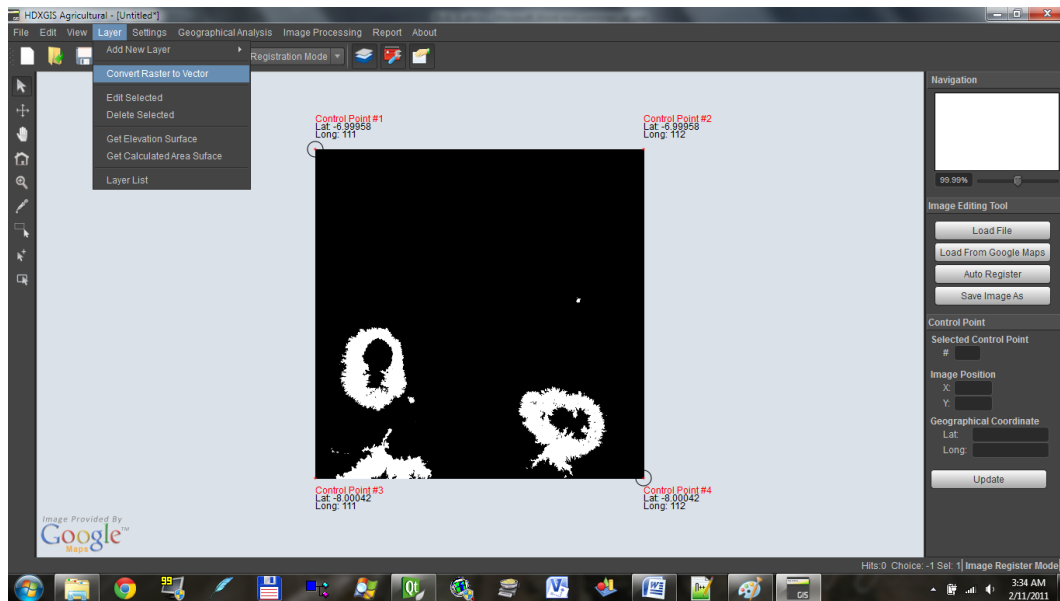
1.3.3 Implementasi Metode Perubahan Data *Raster* Menjadi Data *Vector*

Fitur untuk merubah data *raster* menjadi *vector* sebenarnya sudah ada pada saat *user* ingin melakukan analisis. Namun *user* dapat melakukan proses ini sebagai fitur yang terpisah, sehingga dapat mencoba melakukan proses ini pada sebuah data raster sebagai data gambar. Untuk menggunakan fitur ini *user* terlebih dahulu harus masuk ke dalam *image register mode* seperti yang dapat dilihat pada Gambar 4.9. di halaman 106.



Gambar 4.9 Tampilan Dari *View Image Register Mode*

Fitur merubah data dari *raster* menjadi data *vector* dapat dilihat pada menu *layer* kemudian *convert data to vector* seperti yang dapat dilihat pada Gambar 4.10. Menu ini hanya akan berpengaruh pada *user* yang sedang aktif di *mode image register*. Apabila *user* memilih menu tersebut maka proses perubahan data akan dijalankan. Setelah proses selesai maka akan muncul sebuah *layer* dengan tipe *vector* yang merupakan hasil *vector* dari data *raster* yang dirubah seperti yang dapat dilihat pada Gambar 4.11.

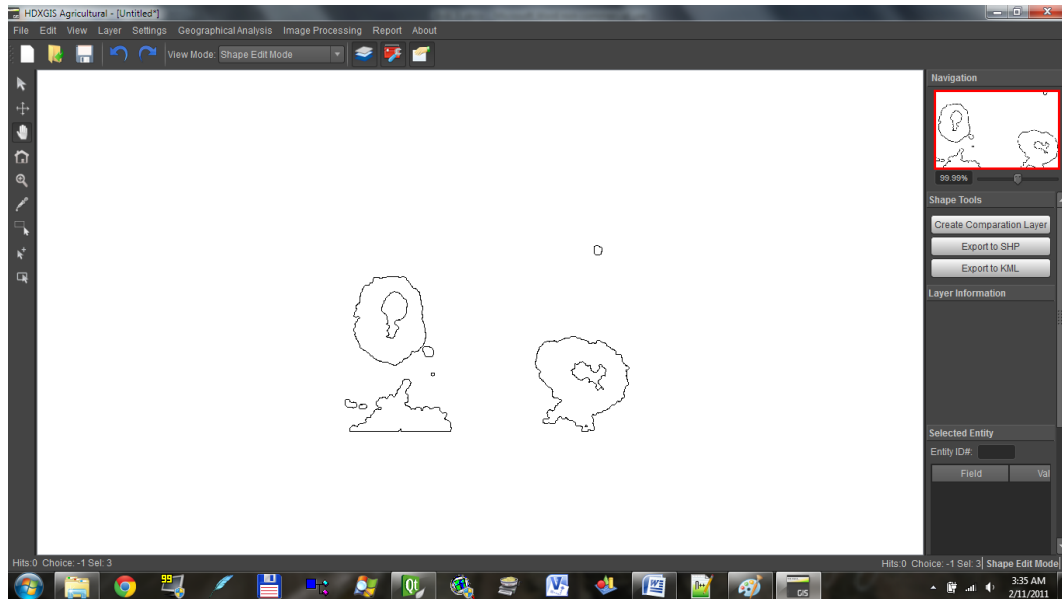


Gambar 4.10 Tampilan *Menu Convert Raster to Vector*



Gambar 4.11 Tampilan Hasil *Vector Hasil Convert*

Apabila *user* ingin melihat hasil dari *layer vector* tersebut *user* dapat memilih *mode edit* pada *property window* kemudian akan sistem akan berpindah ke *mode shape vector editor* dimana akan ditampilkan data *raster* yang telah dipilih seperti pada Gambar 4.12. di halaman 108.



Gambar 4.12 Tampilan *Shape Vector Editor*

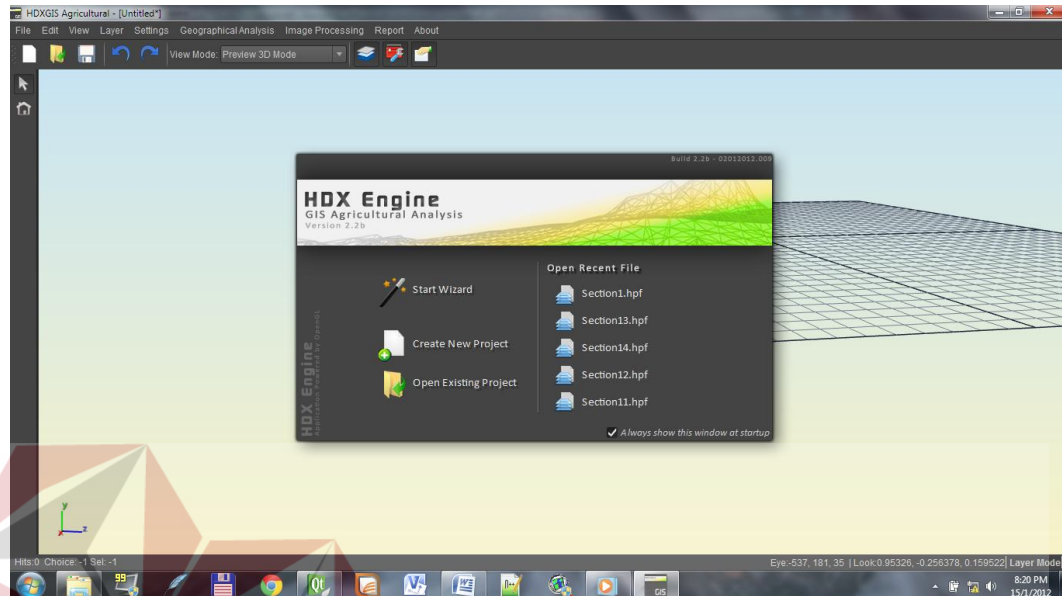
1.3.4 Implementasi Splash Screen

Splash Screen adalah *form* yang pertama kali muncul pada saat aplikasi pertama kali dijalankan. Pada *splash screen* terdapat *MRU File list* di sisi kanan yang berisi list *project* yang telah dibuka sebelumnya.

Form splash screen merupakan *form* yang akan muncul pada saat awal program dijalankan. *Form* memiliki dua sisi sisi kanan di sisi kiri. Disisi kiri terdapat pilihan untuk membuat proyek baru atau membuka proyek sebelumnya yang telah dibuat. Sedangkan pada sisi kanan terdapat *MRU file* yang berarti list dari proyek-proyek yang terakhir dibuka oleh *user*. *MRU list* ini akan memudahkan *user* untuk membuka proyek yang telah digunakan sebelumnya.

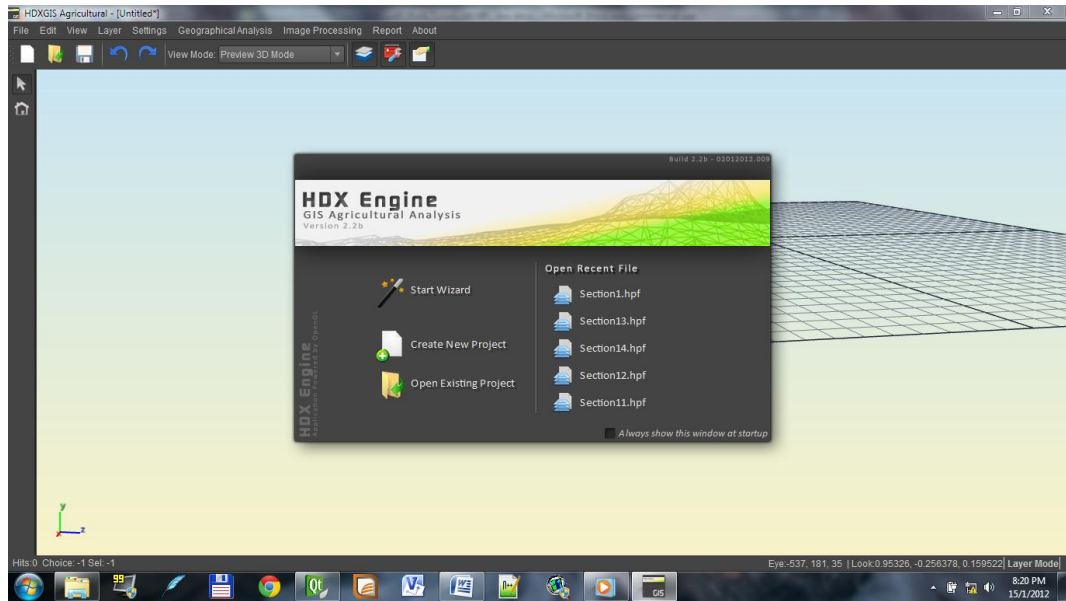
Hasil tampilan dari aplikasi dapat dilihat bahwa pada sisi kanan terdapat list dari *MRU* yang sesuai dengan data pada tabel *configuration*. Data yang tersimpan pada tabel adalah alamat dari *file* secara lengkap, namun pada *splash*

screen hanya ditampilkan nama *filenya* saja. Tampilan dari aplikasi dapat dilihat pada Gambar 4.13.

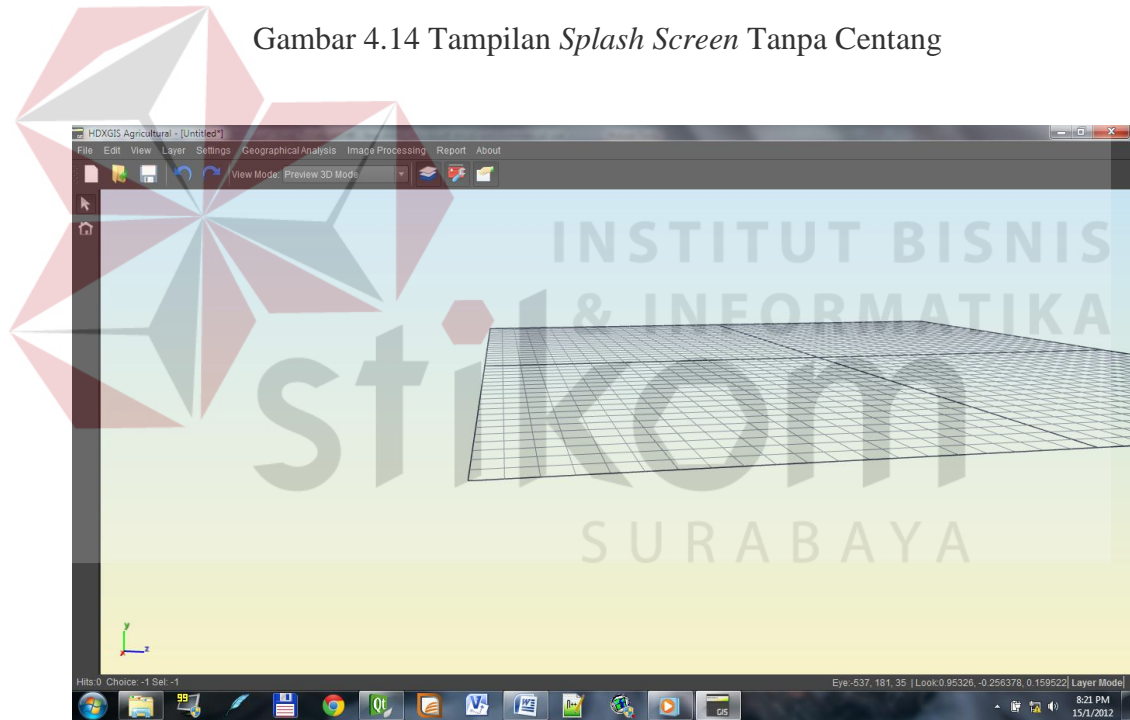


Gambar 4.13 Tampilan *Splash Screen*

Pada tampilan *Splashscreen* yang dapat dilihat Gambar 4.13 terdapat sebuah checkbox dengan keterangan *Always show this window at startup*. Kondisi pada saat aplikasi dijalankan adalah dalam keadaan tercentang. Ketika *user* menghilangkan tanda centang tersebut seperti yang dapat dilihat pada Gambar 4.14 di halaman 110 maka pada saat aplikasi dibuka kembali maka *form* splash akan hilang dan langsung menampilkan *view layer editor* seperti yang dapat dilihat pada Gambar 4.15. di halaman 110.



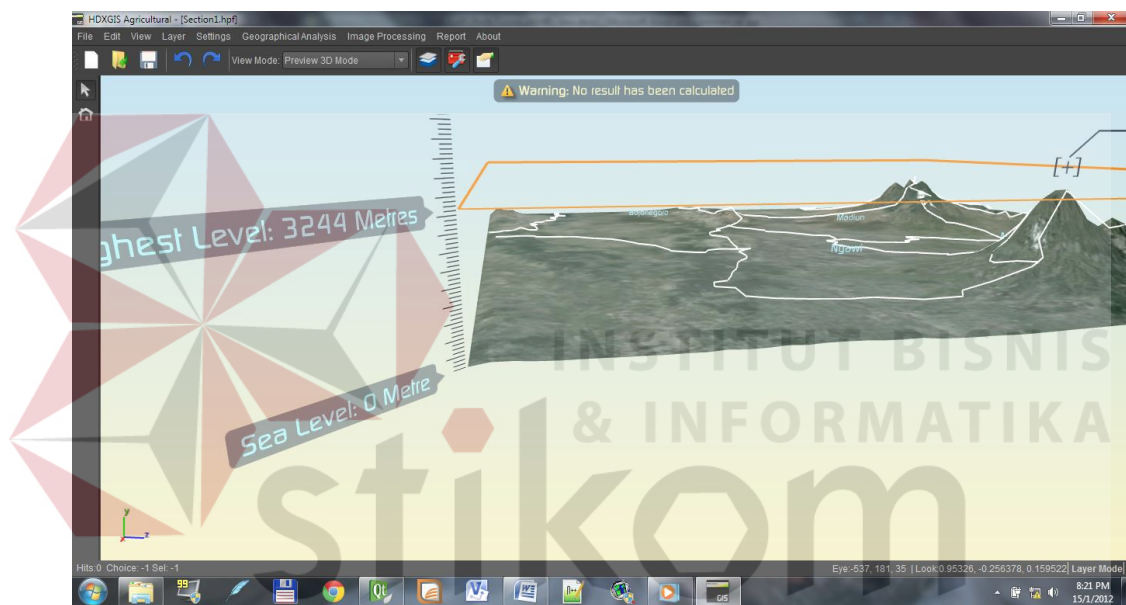
Gambar 4.14 Tampilan *Splash Screen* Tanpa Centang



Gambar 4.15 Tampilan Setelah Menekan *Create New Project*

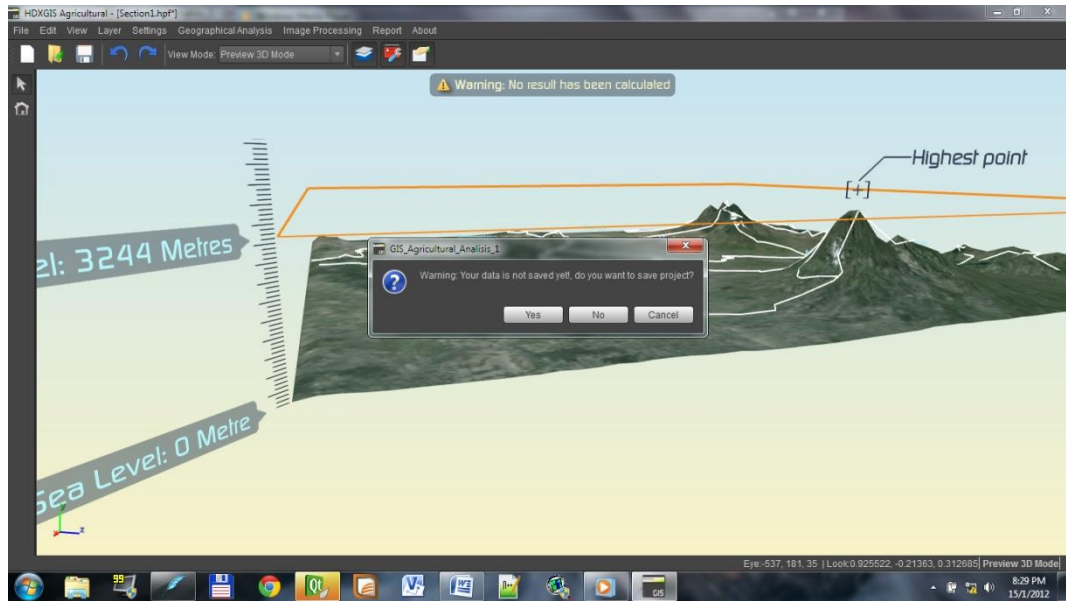
1.3.5 Implementasi Buat *Project* Baru

Dalam aplikasi untuk dapat membuat *project* baru dapat dilakukan dengan beberapa cara salah satunya adalah dengan menggunakan menu *file* kemudian *new project* atau dapat juga dengan menekan tombol *keyboard* Ctrl+N. Tampilan dari aplikasi dimana sebuah *project* sudah terbuka kemudian *user* ingin membuat *project* baru dengan menggunakan menu *new project* pada menu *file* dapat dilihat pada Gambar 4.16.



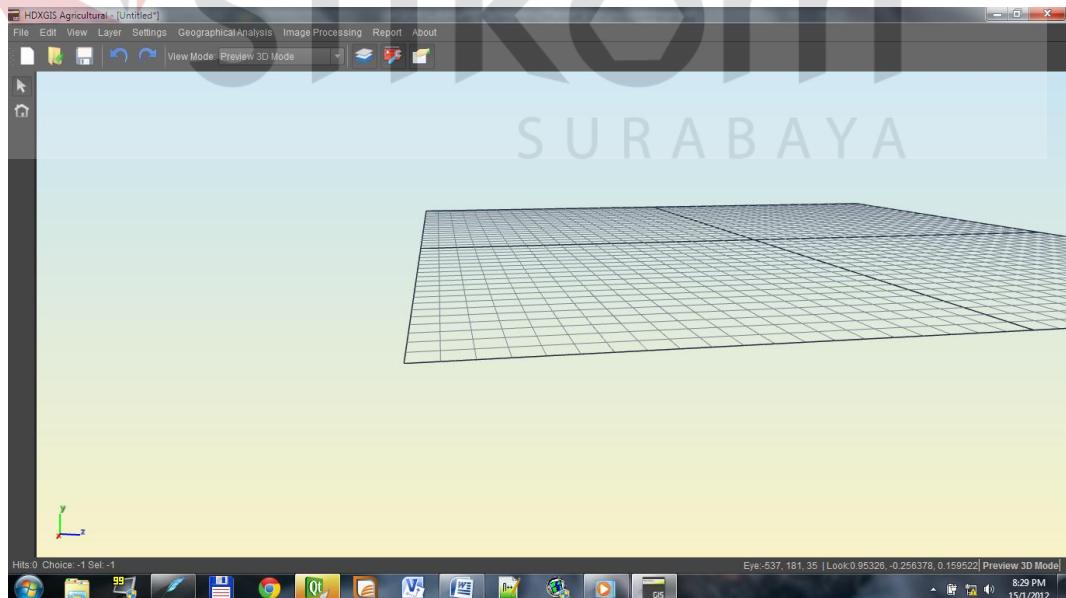
Gambar 4.16 Tampilan *Layer Editor* Setelah Proses *Load Project*

Karena sudah ada proyek baru yang akan dibuka. Maka sistem akan mengeluarkan pesan untuk menyimpan proyek sebelumnya seperti yang dapat dilihat pada Gambar 4.17. di halaman 112.



Gambar 4.17 Pesan Apabila Ada *Project* Yang Sedang Terbuka

Ketika *user* telah memilih *yes* maka sistem akan menutup proyek yang lama kemudian akan membuat proyek yang baru dan ditampilkan pada *mode layer edit view* seperti pada Gambar 4.23.

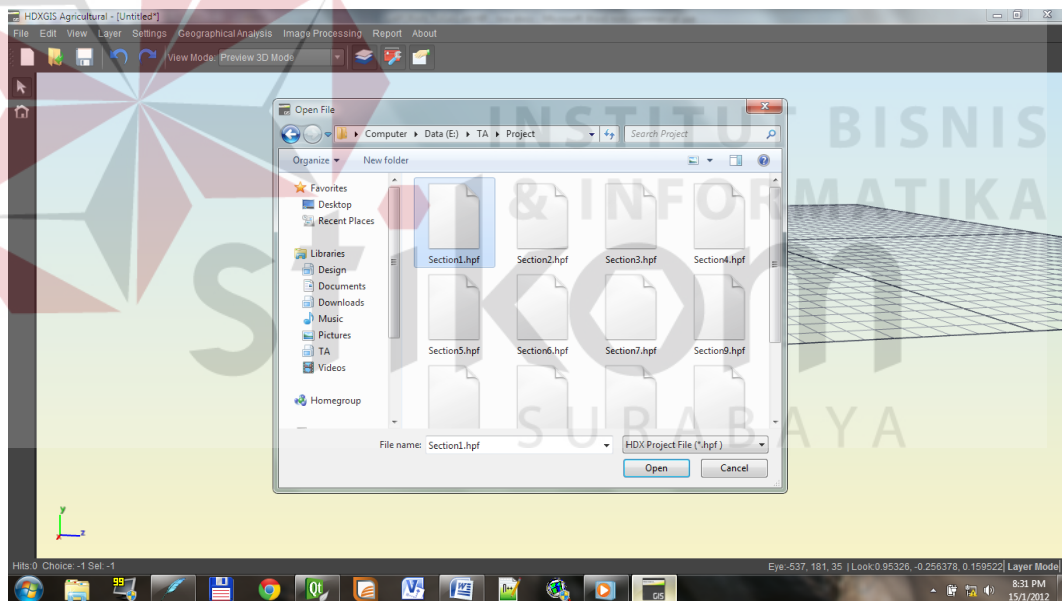


Gambar 4.18 Tampilan *New Project*

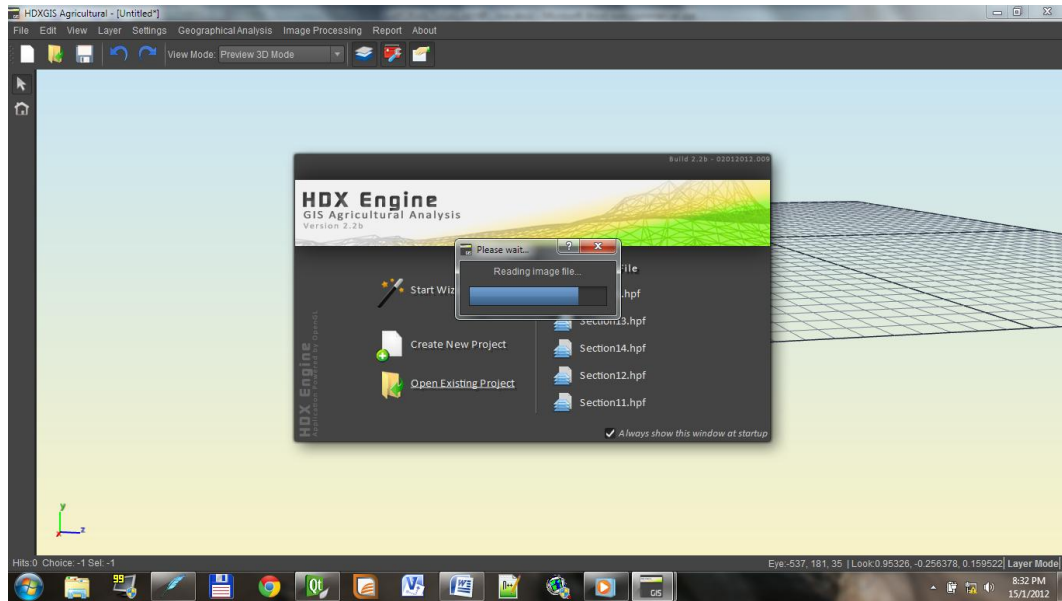
1.3.6 Implementasi Load Project

Membuka *project* pada aplikasi ini dapat dilakukan dengan beberapa cara, bisa melalui *menu*, menggunakan *shortcut* pada keyboard atau memilih *create project* pada menu *splash screen*.

Ketika *user* ingin membuka *project* maka akan muncul sebuah kotak dialog yang meminta *user* untuk memilih *file project* yang akan dibuka. Tampilan aplikasi pada saat ingin membuka *project* dapat dilihat pada Gambar 4.19. Apabila *user* telah memilih *file* maka proses akan membuka proyek tersebut dengan menampilkan *progress bar* seperti yang dapat dilihat pada Gambar 4.20 di halaman 114.

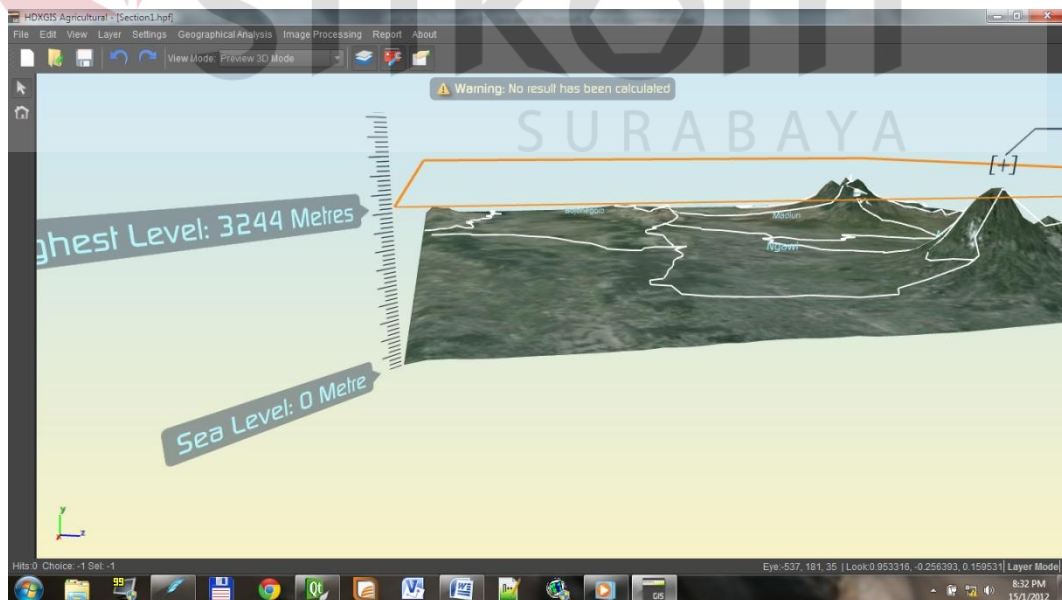


Gambar 4.19 Tampilan Kotak Dialog *Load Project*



Gambar 4.20 Tampilan *Load Project*

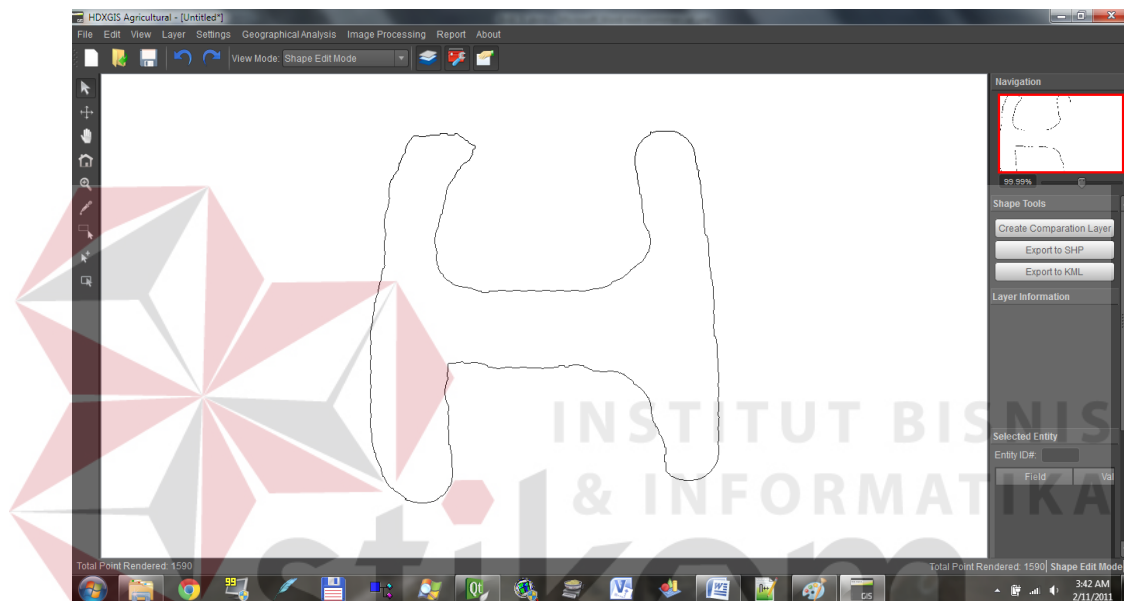
Setelah proses *load project* selesai maka *layer* yang sedang aktif saat ini akan dihapus dan akan ditambah dengan *layer* baru yang merupakan data dari *project* yang telah dibuka. Tampilan dari aplikasi yang telah berhasil melakukan *load project* dapat dilihat pada Gambar 4.21.



Gambar 4.21 Tampilan Setelah *Load Project*

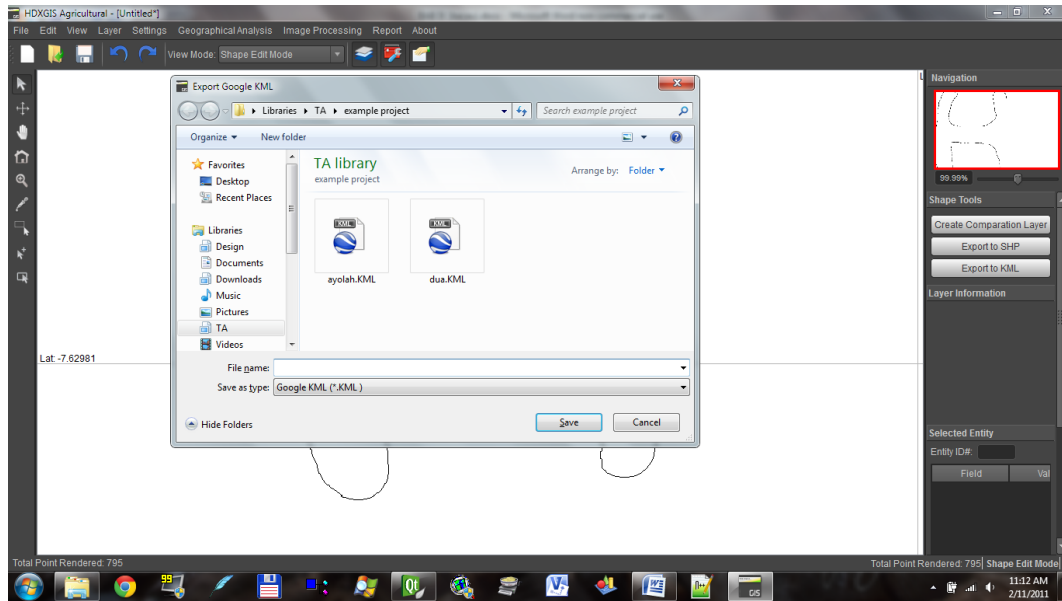
1.3.7 Implementasi Export Google KML

Sistem informasi geografis ini dapat menyimpan format ke dalam bentuk Google KML. *Layer* dengan tipe data *vector* dapat di *export* ke dalam format ini. Berikut adalah tampilan ketika *user* ingin melakukan *export* menjadi format Google KML. *User* dapat memilih jenis *export* yang bisa dipilih pada *property panel*. Tampilan aplikasi dapat dilihat pada Gambar 4.22.



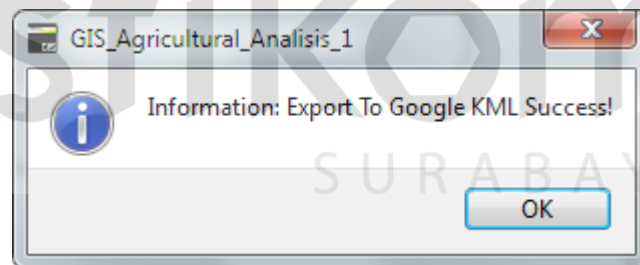
Gambar 4.22 Tampilan *Vector* Yang Akan Dilakukan *Export*

Tampilan ketika tombol *export* as Google KML akan muncul sebuah kotak dialog *save as*. Kotak dialog ini akan memilih dimana lokasi penyimpanan *file* KML yang merupakan hasil *export* dari Sistem Informasi Geografis ini. *Screenshot* aplikasi dari *kotak dialog* dapat dilihat pada Gambar 4.23 di halaman 116.



Gambar 4.23 Tampilan *Export Google KML*

Dan ketika *user* telah memilih *file* yang akan di *save* kemudian proses *export* selesai maka akan muncul pesan *export* telah berhasil seperti pada Gambar 4.24.

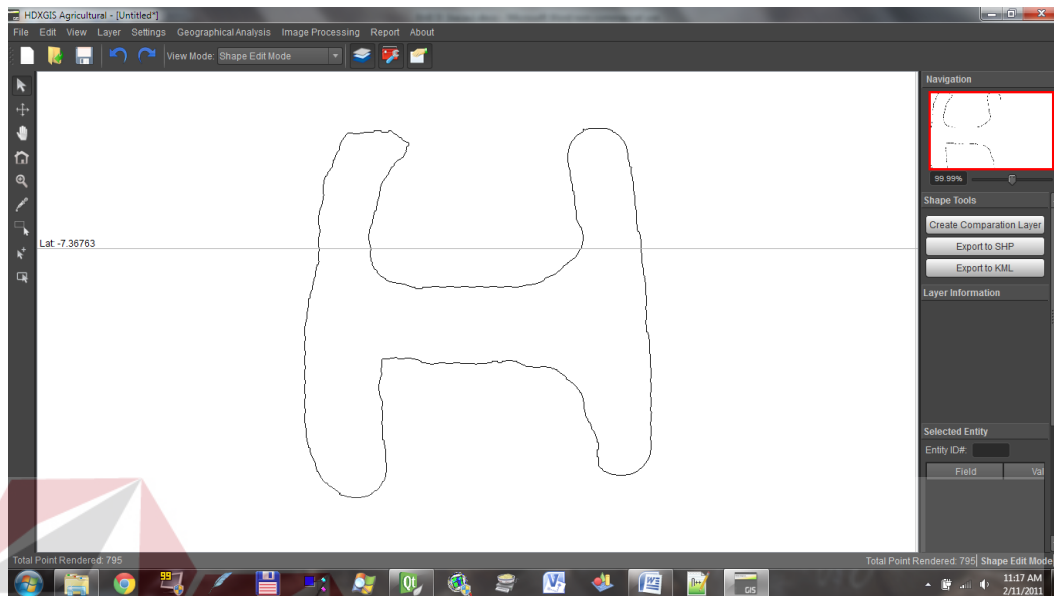


Gambar 4.24 Pesan Proses *Export* Selesai

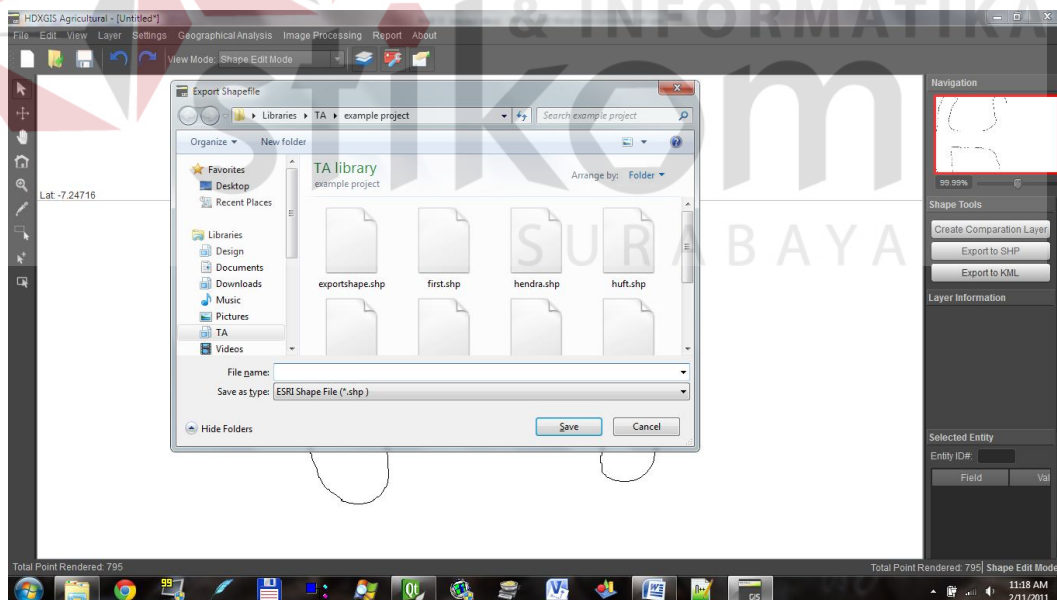
1.3.8 Implementasi Export Shapefile

Sistem informasi geografis ini dapat menyimpan format ke dalam bentuk *Google KML*. Layer dengan tipe data *vector* dapat di *export* ke dalam format ini. Berikut adalah tampilan ketika *user* ingin melakukan *export* menjadi format

Google KML. User dapat memilih jenis *export* yang bisa dipilih pada *property panel*. Tampilan aplikasi dapat dilihat pada Gambar 4.25.

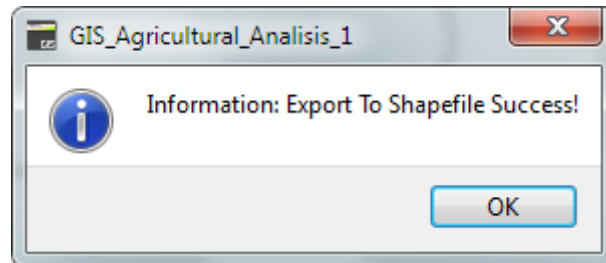


Gambar 4.25 Tampilan Data Vector Yang Akan Dilakukan *Export*
Akan muncul *form export shapefile* seperti pada Gambar 4.26.



Gambar 4.26 Tampilan *Export Shapefile*

Dan ketika *user* telah memilih *file* yang akan di *save* kemudian proses *export* selesai maka akan muncul pesan *export* telah berhasil seperti pada Gambar 4.27.



Gambar 4.27 Pesan Proses *Export* Selesai

1.3.9 Implementasi *Form* Master Tanaman

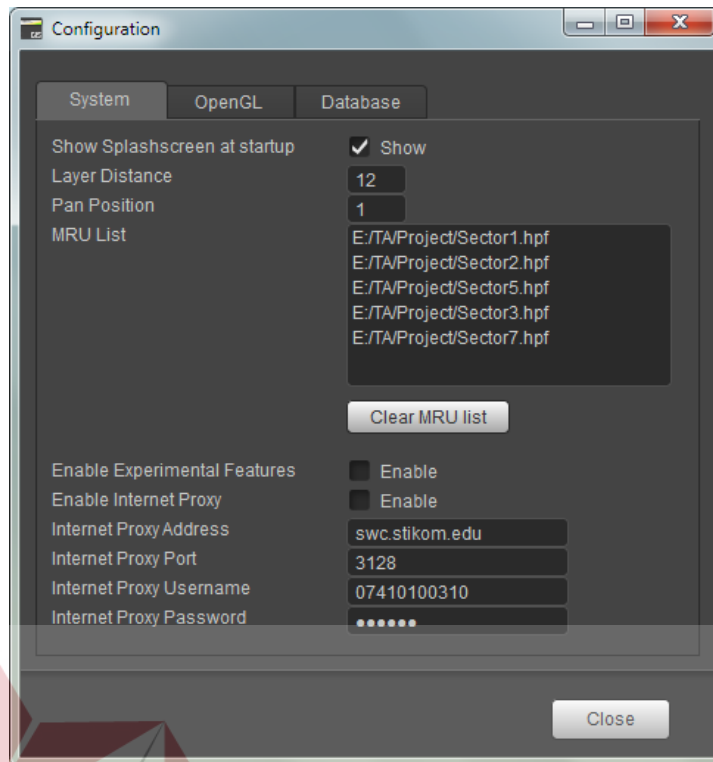
Master tanaman merupakan *form* untuk mengatur data dari *database* Tanaman. Dengan *form* ini *user* dapat menambah, merubah atau menghapus data dari tabel tanaman. Master tanaman digunakan *user* untuk memasukkan data-data tanaman pertama kali. *Field master* tanaman terdiri dari tiga yaitu: ID tanaman, nama tanaman dan kategori tanaman. Tampilan dari *form* master tanaman dapat dilihat pada Gambar 4.28 di halaman 119.

ID_Tanaman	Nama	Desc
0001	coklat	
0002	kopi	
0003	karet	
0004	tembakau	
0005	kelapa	
0006	pinang	
0007	teh	
0008	kina	

Gambar 4.28 Tampilan *Form Master* Tanaman

1.3.10 Implementasi *Form Configuration*

Form System configuration digunakan untuk mengatur *setting* dari *database configuration*. *Form* ini akan merubah *setting* yang digunakan pada saat aplikasi running. *Form configuration* adalah *form* yang mengatur *setting* dari aplikasi yang dijalankan. *Configuration* mengatur beberapa *setting library* seperti OpenGL dan GDAL. Dalam *form configuration* terdapat tiga pilihan *tab* yaitu *System*, OpenGL dan *database*. Tiap *tab* mewakili konfigurasi masing-masing *configuration* dari *library* tersebut. *System* berarti *configuration* secara keseluruhan. OpenGL berarti kumpulan *setting* yang berhubungan dengan visual 3D yang dilakukan oleh *library* openGL. Tampilan dari *Form Configuration* dapat dilihat pada Gambar 4.29. di halaman 120.



Gambar 4.29 Tampilan *Form Configuration*

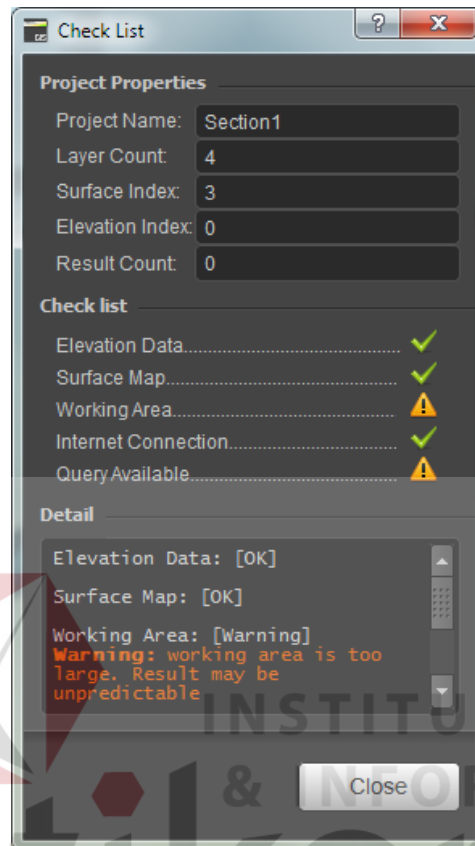
1.3.11 Implementasi *Form Checklist*

Form prepare analysis berfungsi untuk menyiapkan *user* sebelum melakukan *proses analysis*. Terdapat beberapa syarat untuk dapat melakukan proses analisis.

Form ini bertujuan memberikan informasi kepada *user* kriteria-kriteria yang harus dipenuhi sebelum melakukan proses analisis. Apabila ada kriteria yang belum terpenuhi maka *form* akan memberikan keterangan kriteria apa saja yang belum terpenuhi dan apa yang harus dilakukan oleh *user* agar dapat memenuhi kriteria yang belum terpenuhi.

Tampilan dari *form prepare analysis* dapat dilihat pada Gambar 4.30. di halaman 121. Dimana status *checkllist* digambarkan dengan *icon* cek berwarna

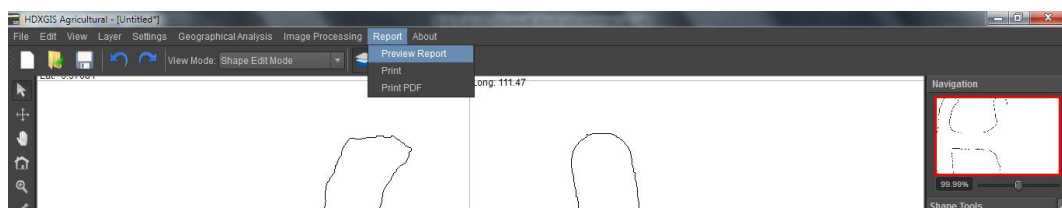
hijau. Apabila kriteria belum terpenuhi maka akan digambarkan dengan tanda seru atau tanda silang.



Gambar 4.30 Tampilan *Form Checklist*

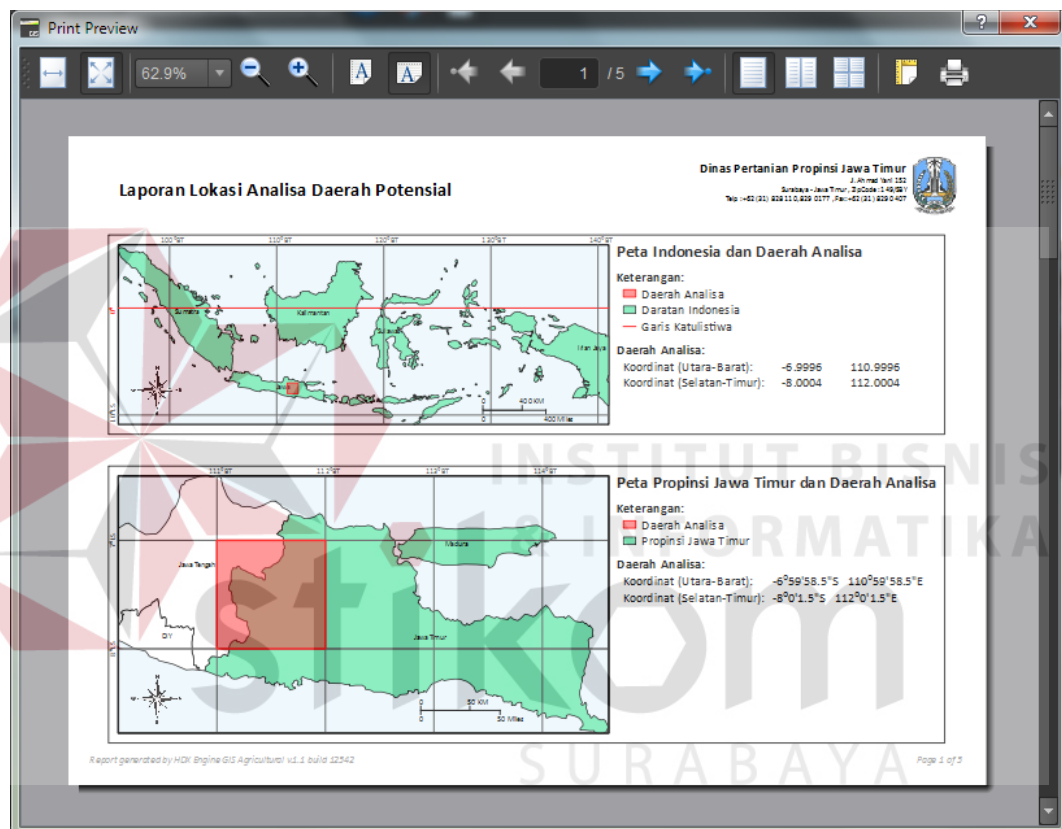
1.3.12 Implementasi *Form Report*

Proses pembuatan laporan cukup dengan memilih menu *report*. *User* dapat memilih beberapa pilihan *menu*, diantaranya *user* dapat melakukan *preview* terlebih dahulu sebelum mencetak laporan. Untuk mengakses fitur *print* dapat dilakukan pada *menu print* seperti pada Gambar 4.31.



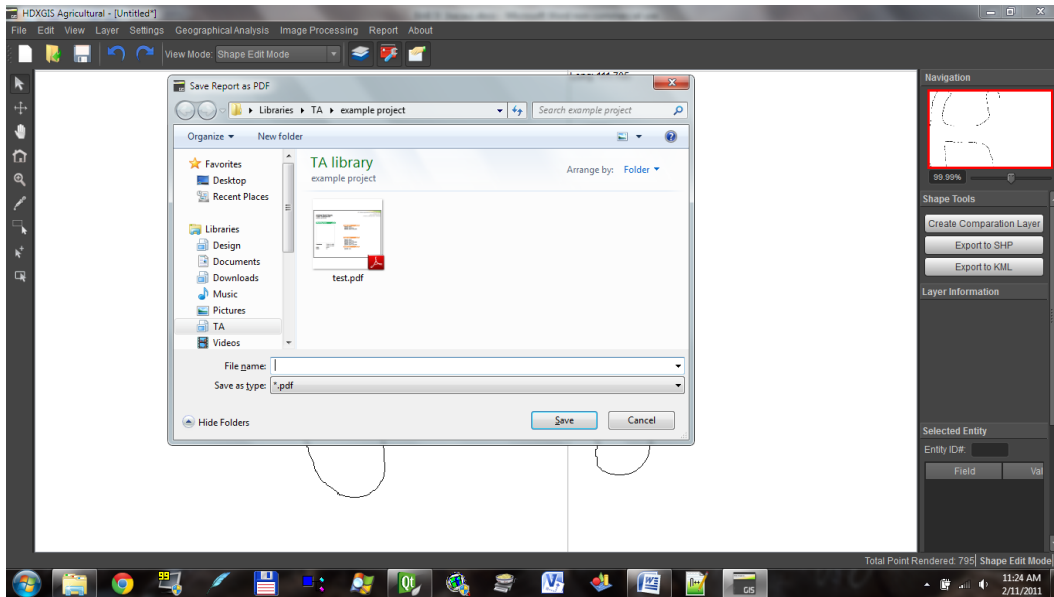
Gambar 4.31 Tampilan *Menu Report*

Tampilan dari *view report* dapat dilihat pada Gambar 4.32. *Report* akan terdiri dari 3 atau lebih halaman. Apabila belum ditentukan tanaman yang ingin ditentukan lokasi daerah potensialnya maka *report* hanya akan berjumlah 2 (dua) halaman. Sedangkan apabila terdapat *query* tambahan, maka jumlah halaman akan disesuaikan dengan jumlah tanaman pada *query*.

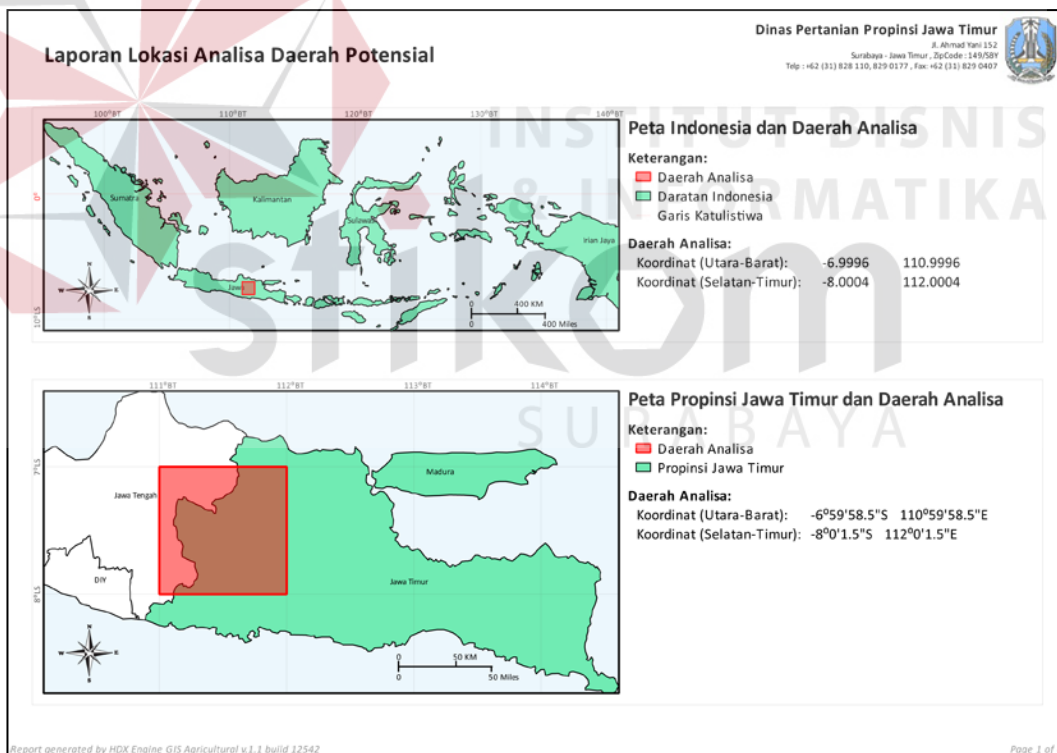


Gambar 4.32 Tampilan *View Report*

Ketika *user* ingin menyimpan laporan dalam *form* PDF maka *user* dapat memilih *print pdf* sehingga akan muncul dialog *save PDF* seperti yang dapat dilihat tampilan aplikasi pada Gambar 4.33. di halaman 123.



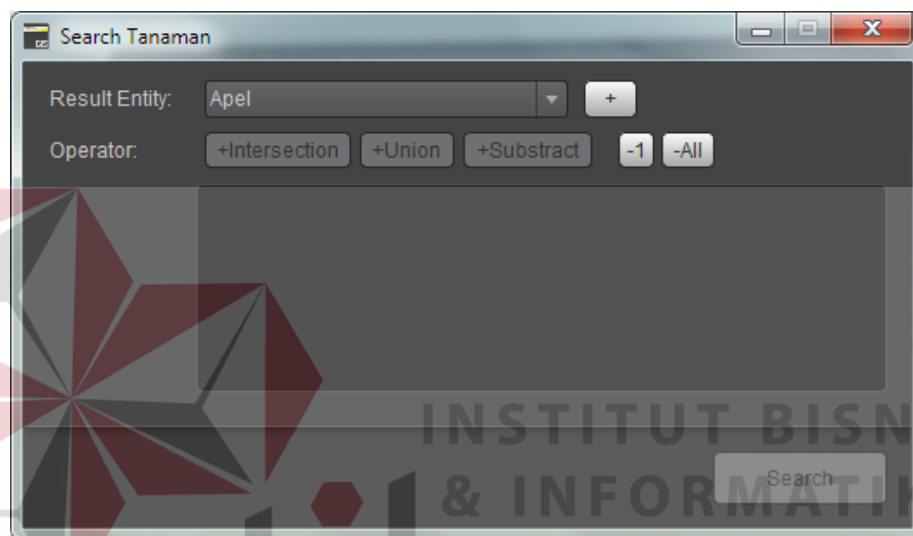
Gambar 4.33 Tampilan *Print To PDF*



Gambar 4.34 Tampilan *Report*

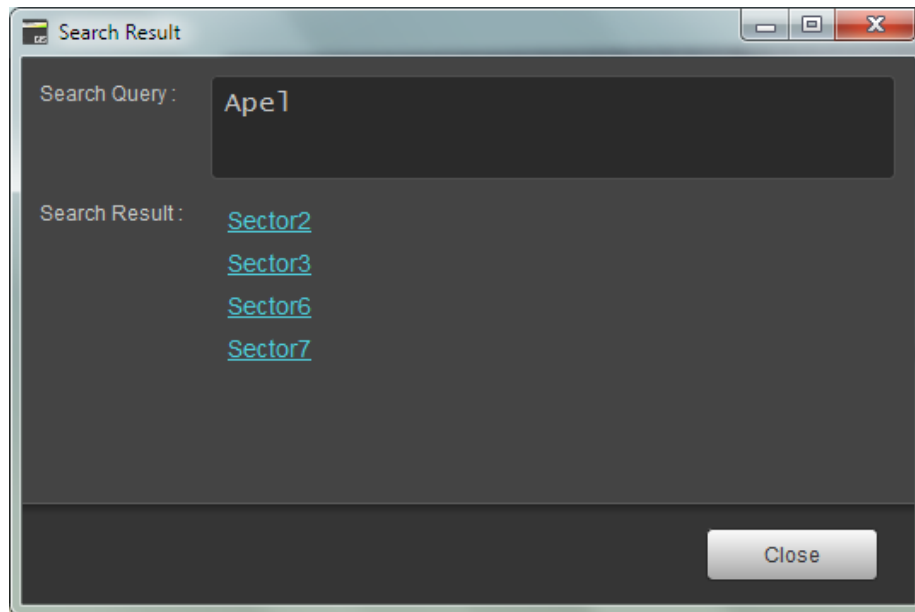
1.4 Implementasi Pencarian Daerah Potensial Berdasarkan Tanaman

Untuk memudahkan *user* dalam mencari lokasi yang cocok ditanami jenis tanaman tertentu, dalam Sistem Informasi ini terdapat fitur untuk mencari lokasi daerah potensial berdasarkan tanaman yang diinputkan oleh *user*. Tampilan *Form* dari fitur pencarian daerah potensial berdasarkan tanaman dapat dilihat pada Gambar 4.35.



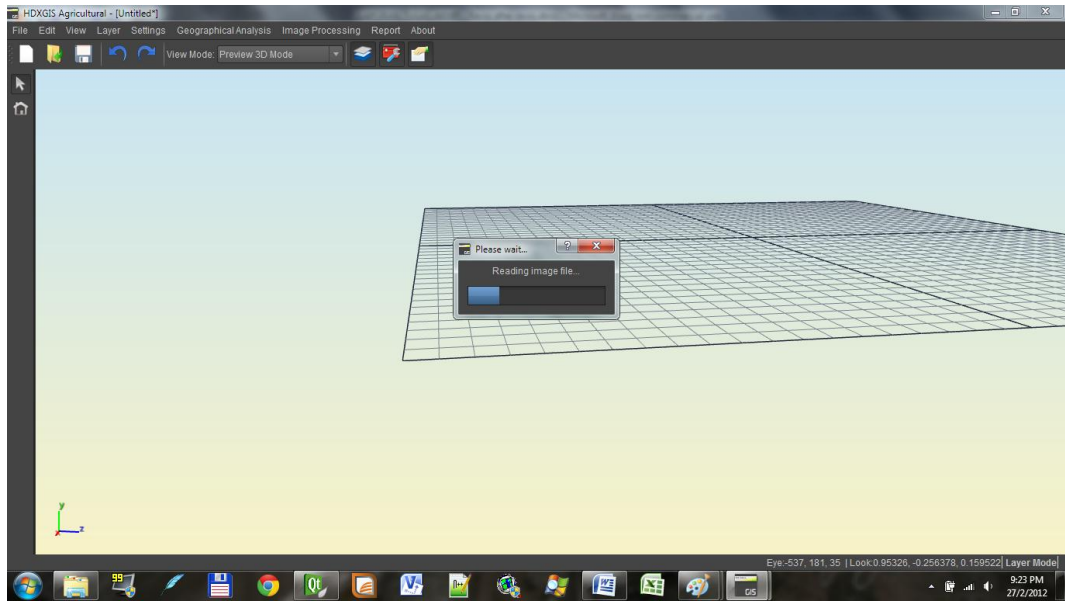
Gambar 4.35 Tampilan *Search Tanaman*

Apabila *user* ingin mencari tanaman yang akan dicarinya *user* dapat menginputkan jenis tanaman dengan memilih jenis tanaman terlebih dahulu pada *combobox* kemudian menekan tombol dengan tanda +(plus). Apabila *user* ingin mencari beberapa jenis tanaman di daerah yang sama, *user* juga dapat memasukkan *query* pada menu ini. Jika jenis tanaman sudah dimasukkan lalu kemudian *user* menekan tombol *search* maka akan muncul *form search result* yang dapat dilihat pada Gambar 4.36 di halaman 125.

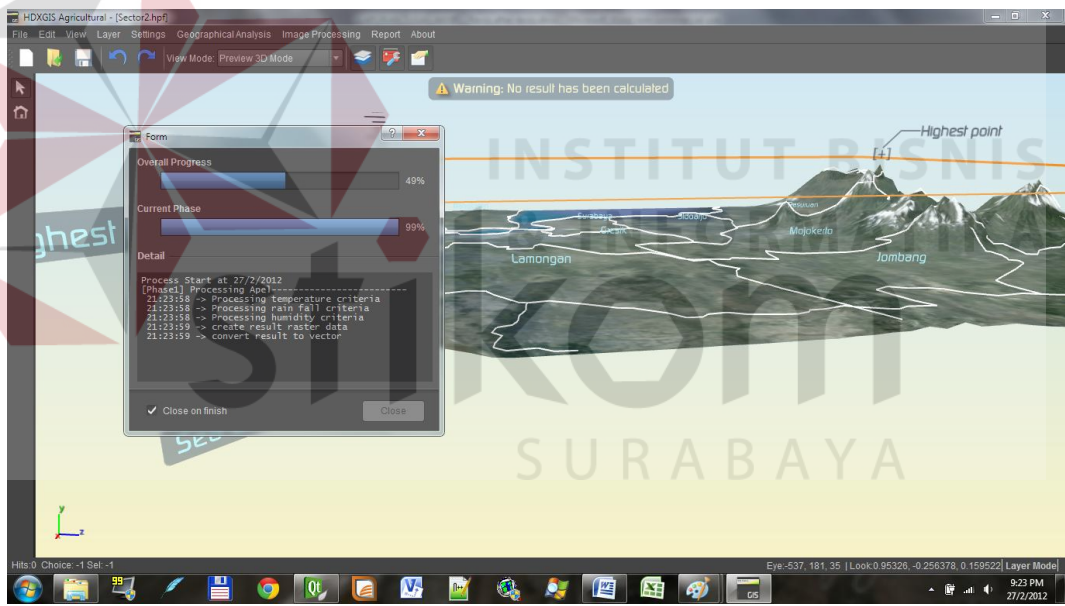


Gambar 4.36 Tampilan *Search Result*

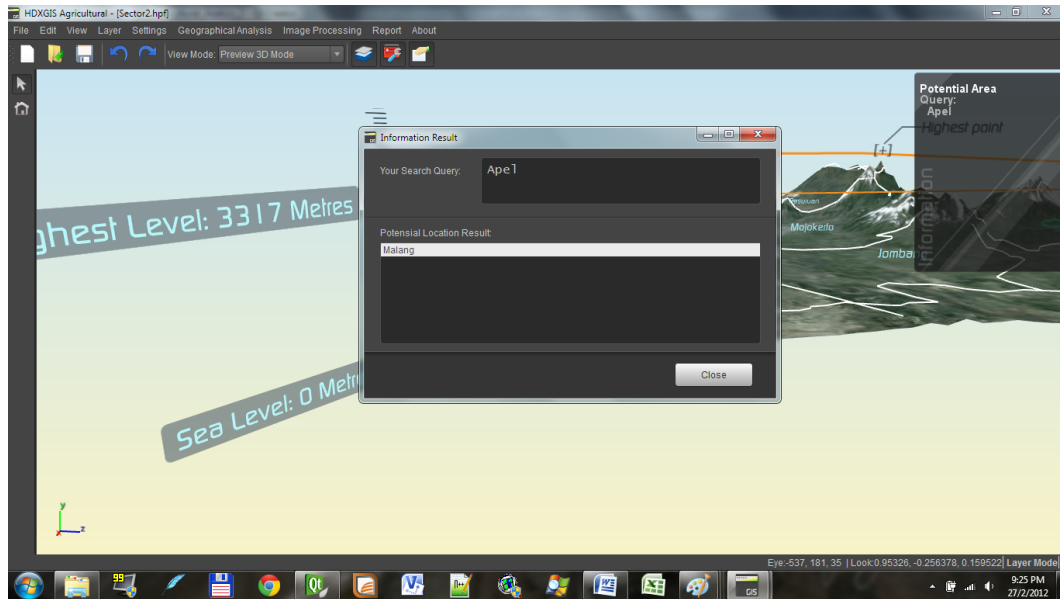
Pada *form* search result terdapat beberapa link yang merupakan daerah pencarian dari *query* tanaman yang telah diinputkan sebelumnya. Sector2 berarti tanaman apel berada di sector2. Ketika *user* meng-klik salah satu dari *link* hasil pencarian, pertama-tama sistem akan membuka *project* sesuai dengan *sector* yang bersangkutan, setelah selesai proses dilanjutkan dengan proses analisis potensial area dengan *query* jenis tanaman yang sama dengan *query* pada *search*. Setelah selesai proses analisa, sistem akan menampilkan pesan bahwa proses telah selesai dan menunjukkan daerah yang potensial. Proses pertama saat membuka *project* dapat dilihat pada Gambar 4.37 di halaman 126. Proses analisa dapat dilihat pada Gambar 4.38 di halaman 126. Dan ketika proses analisa selesai dan menampilkan daerah yang dicari dapat dilihat pada Gambar 4.39 di halaman 127.



Gambar 4.37 Tampilan Proses *Load Project*



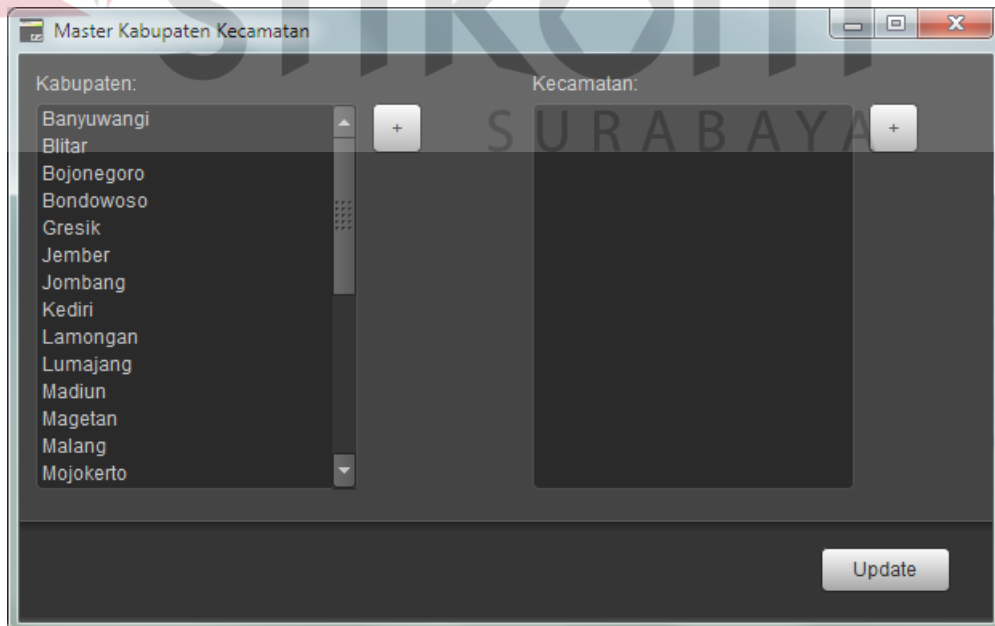
Gambar 4.38 Tampilan Proses Analisis



Gambar 4.39 Tampilan Hasil Analisis

1.5 Implementasi Master Data Kabupaten dan Kecamatan

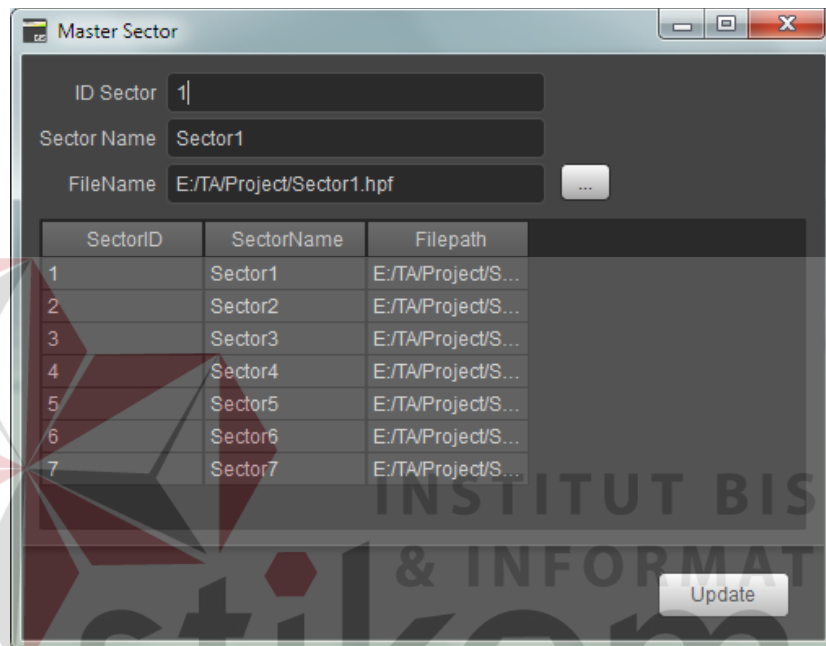
Sistem Informasi ini mencakup seluruh kabupaten dan beberapa kecamatan di Jawa Timur. Untuk itu diperlukan fitur master kabupaten dan kecamatan. Tampilan program dapat dilihat pada Gambar 4.40.



Gambar 4.40 Tampilan Form Master Kabupaten Kecamatan

1.5.1 Implementasi Master Sector

Untuk memudahkan *user* dan menghemat waktu. Aplikasi ini memiliki fitur untuk menyimpan *sector* atau area yang berupa *project file*, sehingga data-data mengenai *project file* telah tersusun rapi. Tampilan aplikasi untuk *master sector* dapat dilihat pada Gambar 4.41.



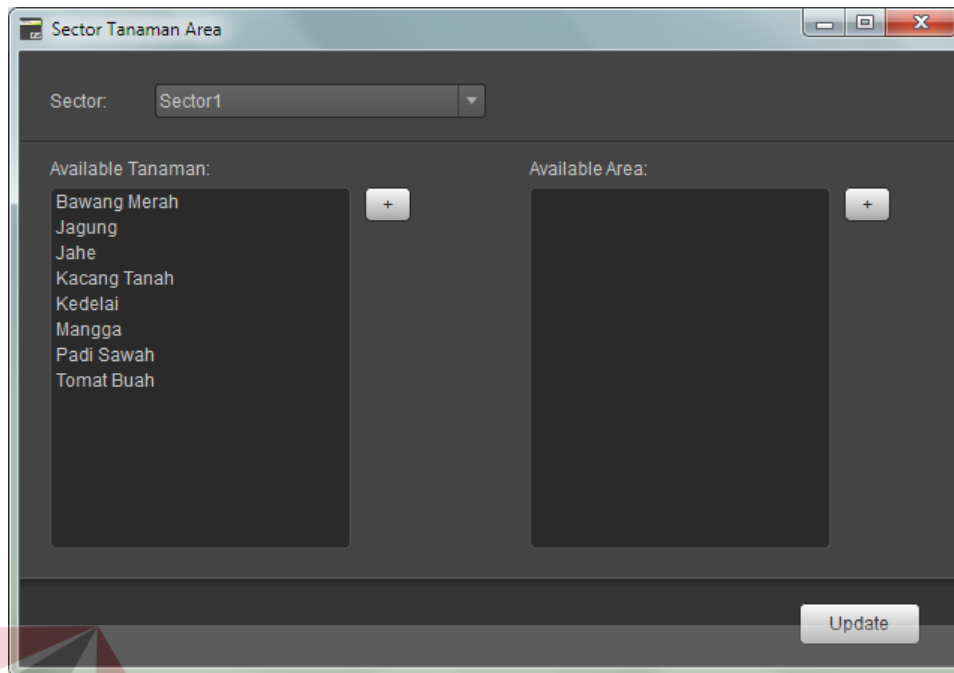
The screenshot shows a window titled "Master Sector" with a dark theme. It contains a form with three input fields: "ID Sector" with the value "1", "Sector Name" with the value "Sector1", and "FileName" with the value "E:/TA/Project/Sector1.hpf". Below the form is a table with three columns: "SectorID", "SectorName", and "Filepath". The table contains seven rows of data. An "Update" button is located at the bottom right of the window.

SectorID	SectorName	Filepath
1	Sector1	E:/TA/Project/S...
2	Sector2	E:/TA/Project/S...
3	Sector3	E:/TA/Project/S...
4	Sector4	E:/TA/Project/S...
5	Sector5	E:/TA/Project/S...
6	Sector6	E:/TA/Project/S...
7	Sector7	E:/TA/Project/S...

Gambar 4.41 Tampilan *Form Master Sector*

1.5.2 Implementasi Sector Tanaman

Aplikasi ini mampu melakukan pencarian daerah potensial berdasarkan tanaman. Agar dapat melakukannya dibutuhkan *database* yang menyimpan setiap kali proses analisis dilakukan. Ketika proses analisis selesai dilakukan. Data daerah dan data tanaman disimpan kedalam *database sector* tanaman, sehingga data *sector*, daerah dan tanaman dikaitkan agar dapat dilakukan proses pencarian. Tampilan aplikasi dapat dilihat pada Gambar 4.42 di halaman 129.

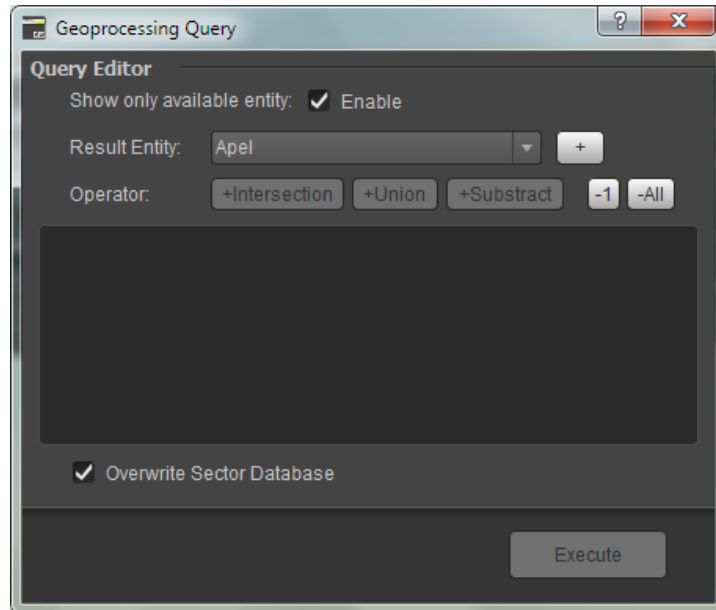


Gambar 4.42 Tampilan *Form Master SectorTanaman*

1.5.3 Implementasi Geoprocessing Query

Fitur *geoprocessing query* adalah proses analisa yang digabungkan dengan *query*. Apabila *user* ingin mencari lokasi daerah potensial untuk tanaman tertentu fitur ini dapat digunakan. Jenis tanaman yang dicari bisa digabungkan dengan jenis tanaman lain dengan menggunakan *geoprocessing*. Adapun beberapa *geoprocessing* yang didukung dalam sistem ini adalah *Intersection*, *Union* dan *Substract*.

Dalam fitur ini juga terdapat pilihan untuk menampilkan jenis tanaman yang memiliki daerah potensial, sehingga untuk tanaman yang tidak memiliki kemungkinan maka jenis tanamannya tidak akan ditampilkan dalam *combobox*. Tampilan aplikasi dari fitur ini dapat dilihat pada Gambar 4.43 di halaman 130.

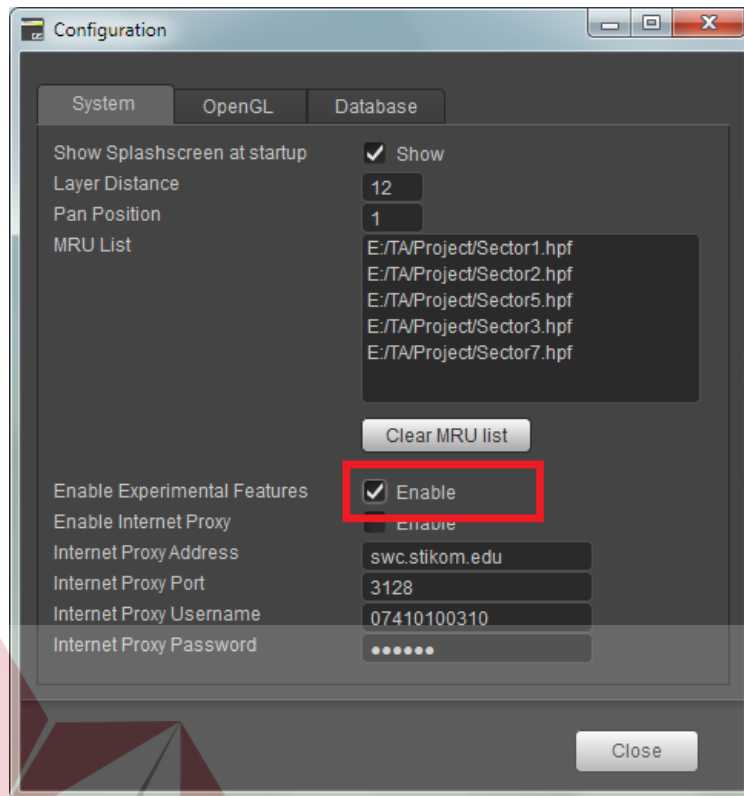


Gambar 4.43 Tampilan *Geoprocessing Query*

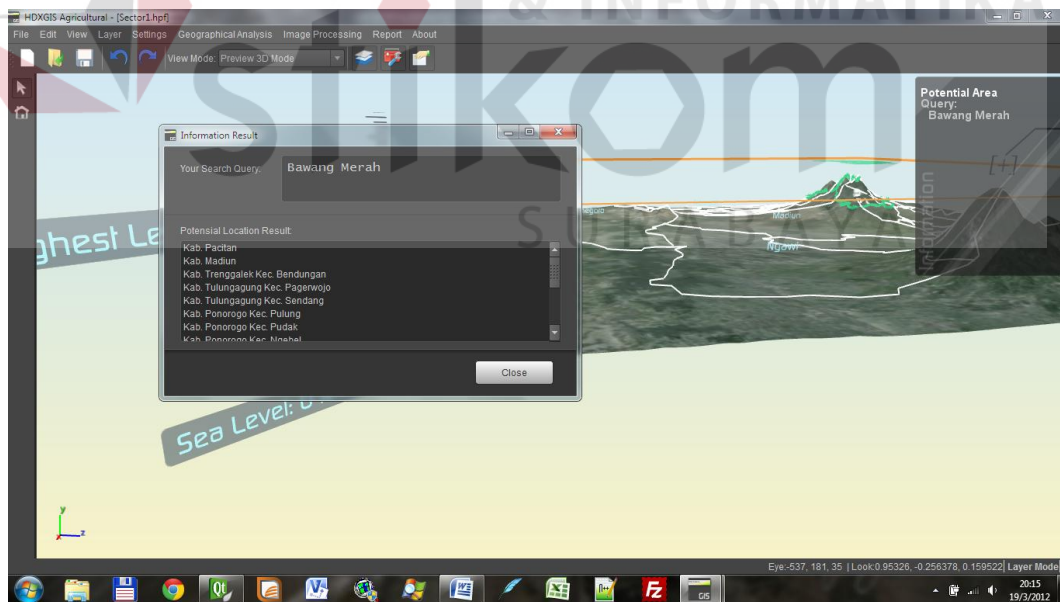
1.5.4 Implementasi *Experimental Feature*

Untuk dapat menggunakan fitur ini user pertama-tama harus mengaktifkan fitur ini pada *form setting*. *Form setting* dapat dibuka melalui *menu setting* lalu *submenu system configuration*. Pada *form setting* pilih *tab system* dan disana terdapat *checkbox* yang berisi tulisan *Enable Experimental Feature*. Beri tanda centang pada *checkbox* tersebut. Untuk lebih jelasnya gambar *form System Configuration* dapat dilihat pada Gambar 4.44 di halaman 131.

Perubahan juga terjadi pada hasil analisis ketika proses analisis selesai *form* akan menunjukkan kecamatan seperti pada Gambar 4.45 di halaman 131. Selain itu *report* juga menampilkan gambar jenis tanah pada kolom kelima. Gambar *report* saat *experimental feature* dapat dilihat pada Gambar 4.46. di halaman 131.



Gambar 4.44 Tampilan Form setting

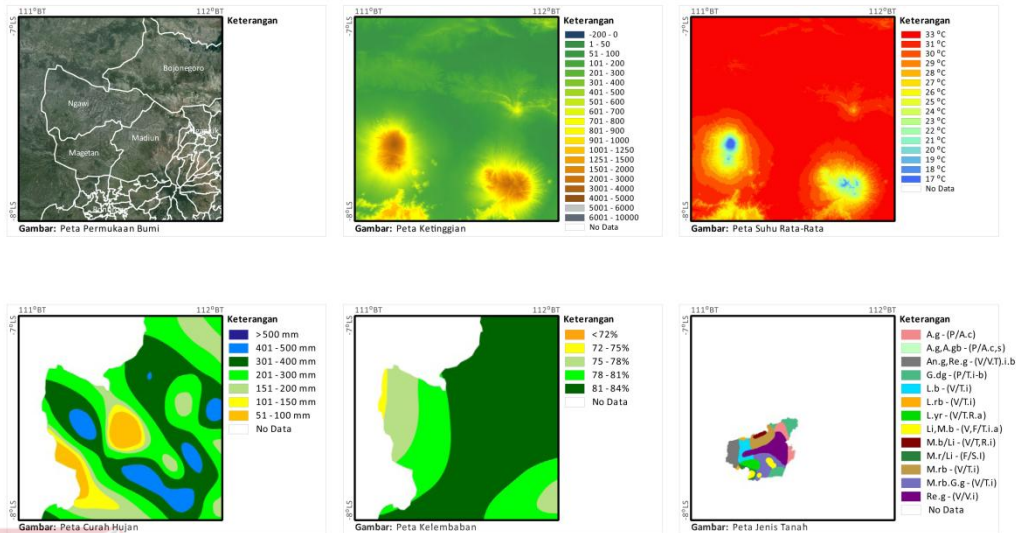


Gambar 4.45 Tampilan Setelah Proses Analisis Selesai

Laporan Detail Daerah Analisa

Dinas Pertanian Propinsi Jawa Timur

Jl. Ahmad Yani 152
Surabaya - Jawa Timur, ZipCode: 60189
Telp: +62 (31) 828 110, 829 0177, Fax: +62 (31) 829 0407

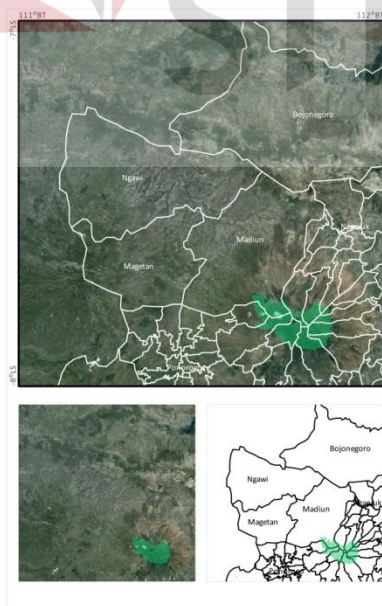


Gambar 4.46 Tampilan Report Experimental

Detail Laporan Daerah Potensial

Dinas Pertanian Propinsi Jawa Timur

Jl. Ahmad Yani 152
Surabaya - Jawa Timur, ZipCode: 60189
Telp: +62 (31) 828 110, 829 0177, Fax: +62 (31) 829 0407



Hasil Analisa Daerah Potensial

Keterangan:
Daerah Potensial: Tanaman -> Bawang Merah

Kategori Tanaman:
Sayur-sayuran

Karakter Lahan Untuk Tanaman:
Temperatur Rata-Rata : 20-25 °C
Curah Hujan : 350-600 mm
Kelembaban : >10 %

Kabupaten Yang Berpotensi:

- Kab. Pacitan
- Kab. Madiun
- Kab. Trenggalek Kec. Bendungan
- Kab. Tulungagung Kec. Pagerwojo
- Kab. Tulungagung Kec. Sendang
- Kab. Ponorogo Kec. Pulung
- Kab. Ponorogo Kec. Pudak
- Kab. Ponorogo Kec. Ngebel
- Kab. Kediri Kec. Grogol
- Kab. Kediri Kec. Banyakan
- Kab. Kediri Kec. Semen
- Kab. Kediri Kec. Mojo
- Kab. Nganjuk Kec. Ngetos
- Kab. Nganjuk Kec. Sawahan
- Kab. Nganjuk Kec. Loceret

Gambar 4.47 Tampilan Report

1.6 Evaluasi Hasil Uji Coba Sistem

Evaluasi hasil uji coba dilakukan untuk menguji kembali semua tahapan yang sudah dilakukan selama pengujian berlangsung dan analisa hasil uji coba sistem bertujuan untuk menarik kesimpulan terhadap hasil-hasil uji coba yang dilakukan terhadap sistem. Untuk memastikan bahwa sistem telah dibuat sesuai dengan kebutuhan atau tujuan yang diharapkan maka dilakukan beberapa uji coba. Uji coba meliputi pengujian terhadap fitur dasar aplikasi dan uji coba validasi pengguna terhadap aplikasi dengan menggunakan *blackbox testing*.

Uji coba dilakukan dengan 3 (tiga) tahap. Tahap pertama adalah memastikan bahwa sistem memberikan hasil yang sesuai dengan diharapkan berdasarkan input tertentu. Apabila *output* yang dikeluarkan sesuai dengan apa yang telah diperhitungkan sebelumnya maka aplikasi ini lulus. Tahap kedua adalah uji visualisasi tiga dimensi (3D). Uji dilakukan dengan membuka sebuah *project* yang telah dibuat sebelumnya. Apabila sistem berhasil membuka *project* dan menampilkan dalam visualisasi 3D maka sistem dinyatakan lulus. Tahap ketiga adalah uji tingkat keakuratan.

Dalam uji coba ini digunakan data dari BPS dan Dinas Pertanian berupa data hasil produksi pertanian suatu daerah. Data akan dibandingkan dengan hasil output dari sistem dan akan ditentukan berapa persentase kecocokan antara hasil daerah potensial yang di hasilkan oleh sistem dengan data hasil produksi daerah yang berasal dari BPS dan Dinas Pertanian.

A. Hasil Evaluasi Output

Evaluasi ini dilakukan dari awal ketika input pertama kali dimasukkan hingga Sistem menghasilkan *output*. Pertama Sistem Informasi Geografis ini akan menampilkan *Splash Screen* seperti yang dapat dilihat pada Gambar 4.44

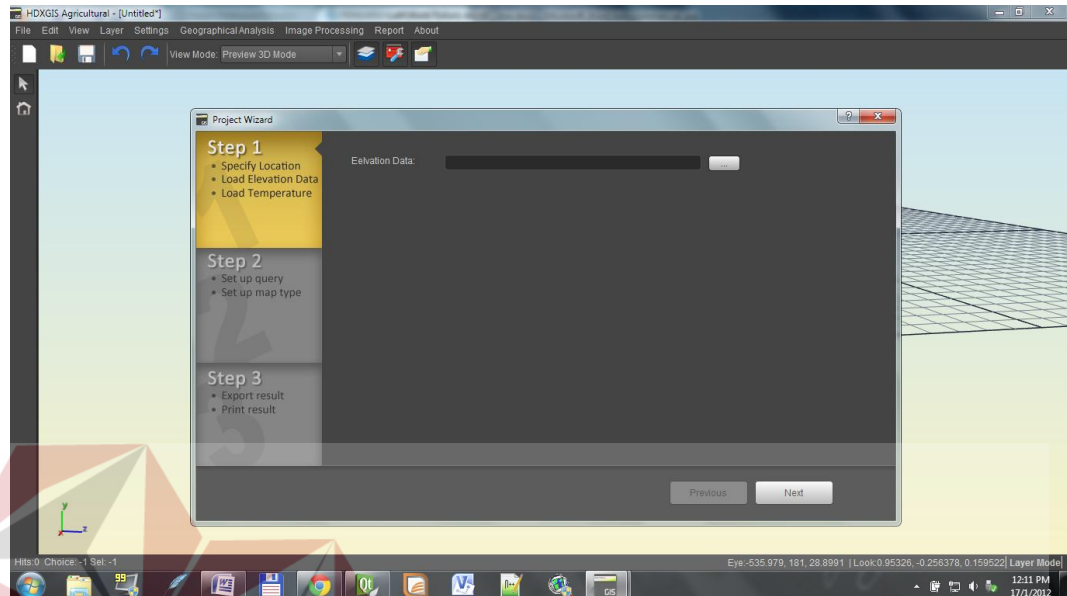
Sistem Informasi Geografis ini memiliki fitur *wizard* yang memudahkan *user* dalam menginputkan data sehingga lebih mudah dan cepat. Fitur *Wizard* akan muncul ketika *user* memilih menekan menu *Start Wizard* pada *Splash Screen*. Tampilan *Wizard page* yang dapat dilihat pada Gambar 4.45 pada halaman 135.



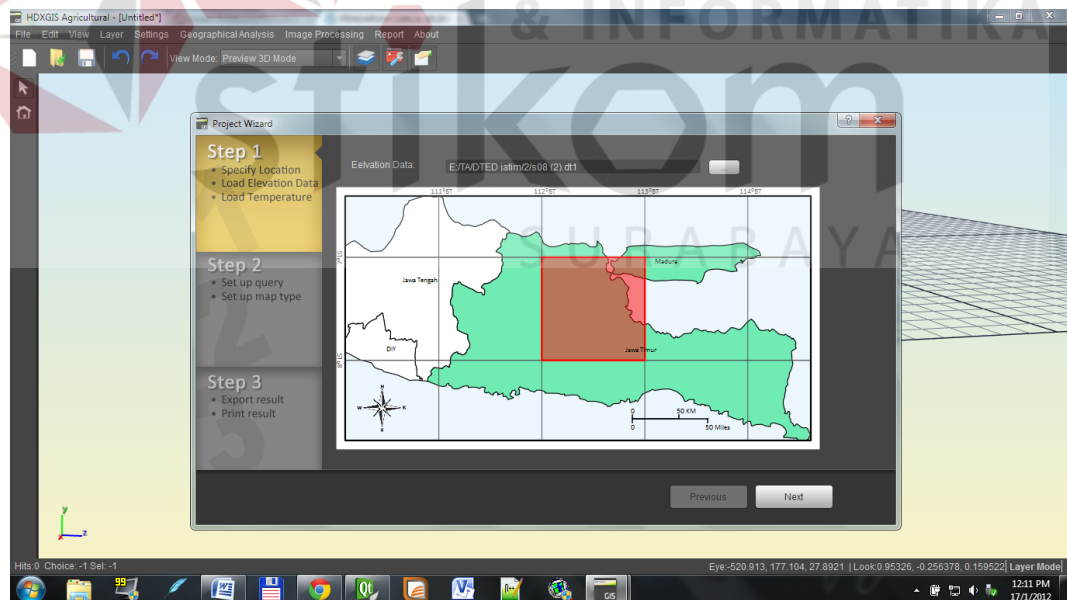
Gambar 4.44 Tampilan *Splash Screen*

Halaman *Wizard* terdiri dari 3 *page*. *Page* pertama berfungsi untuk me-load *DTED file* yang merupakan *file* dari elevasi. *Page* kedua adalah *page* yang menampilkan proses *load* dari semua data yang dibutuhkan, termasuk *download* gambar dari Google Maps. *Page* ketiga yang merupakan *page* terakhir adalah *page* yang menampilkan *query* yang akan menampilkan hasil *geoprocessing*

dari *query*. Halaman pertama *wizard page* dapat dilihat pada Gambar 4.45. Ketika sebuah *file* telah dibuka maka akan tampak seperti Gambar 4.46.



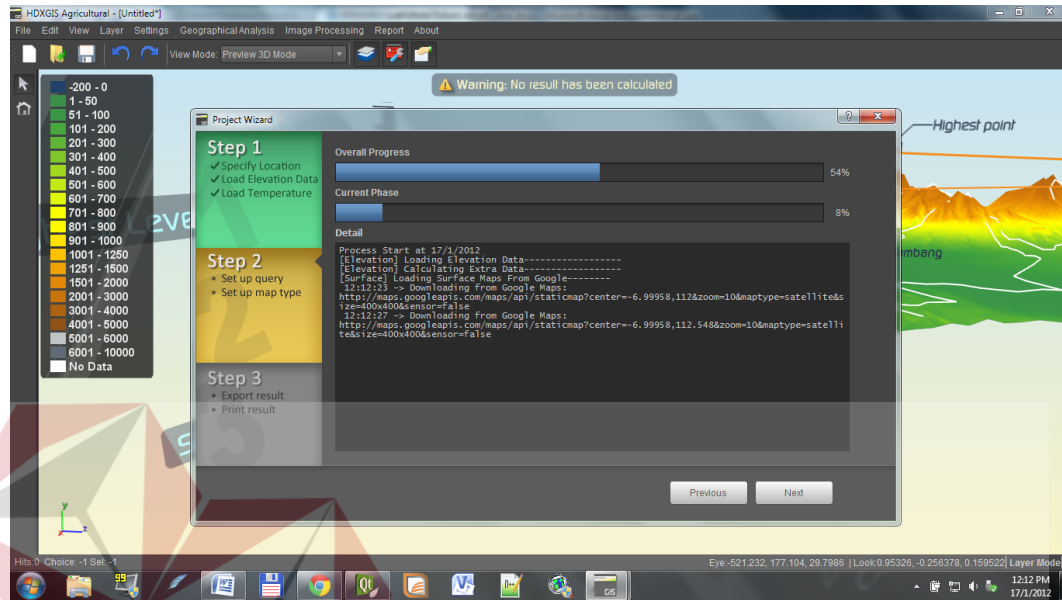
Gambar 4.45 Tampilan Wizard Page



Gambar 4.46 Tampilan Wizard Page

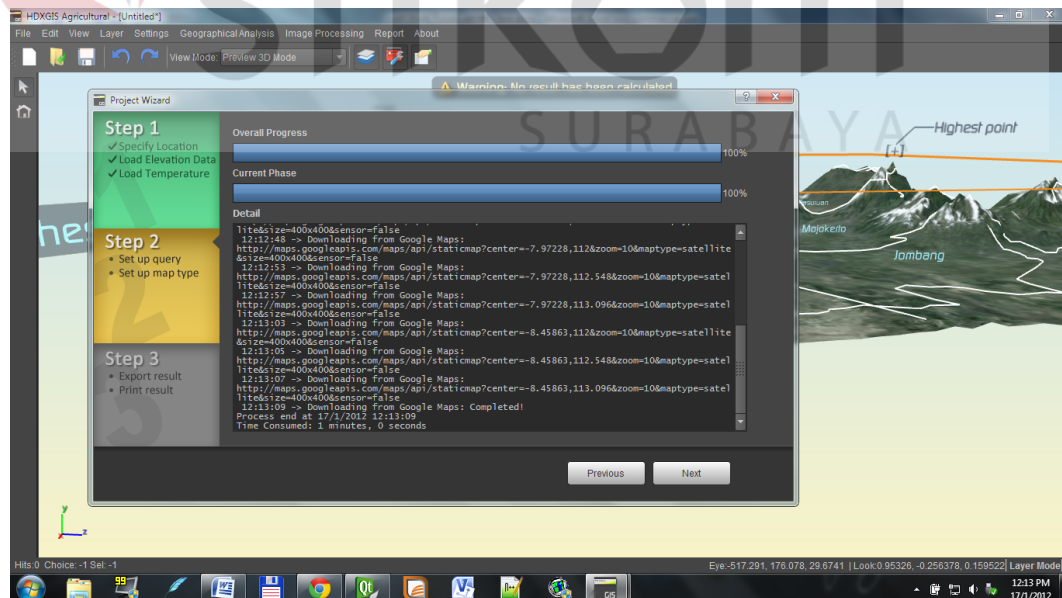
Ketika *user* menekan tombol *Next* maka halaman akan berpindah ke *page* selanjutnya yaitu ke *page 2*. Di *page* dua ini *user* hanya dapat melihat proses yang

sedang berlangsung. *User* akan diberikan informasi berupa proses yang sedang berlangsung, prosentasenya dan berapa lama proses yang dibutuhkan untuk memproses semua data. Gambar *page 2* dapat dilihat pada Gambar 4.47.



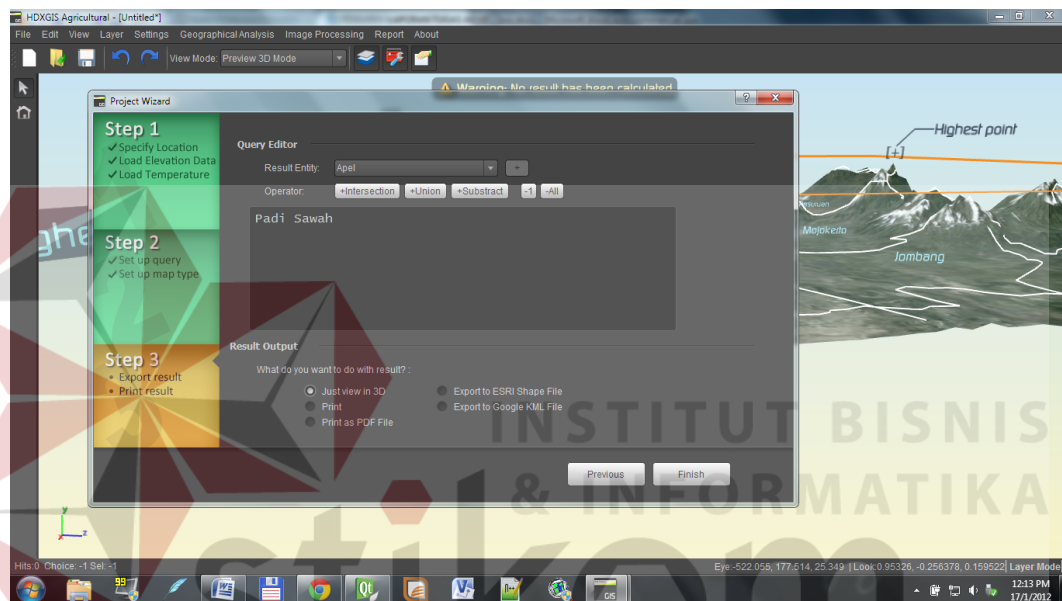
Gambar 4.47 Tampilan Wizard Page 2

Dan ketika proses *loading* selesai akan tampak seperti Gambar 4.48.



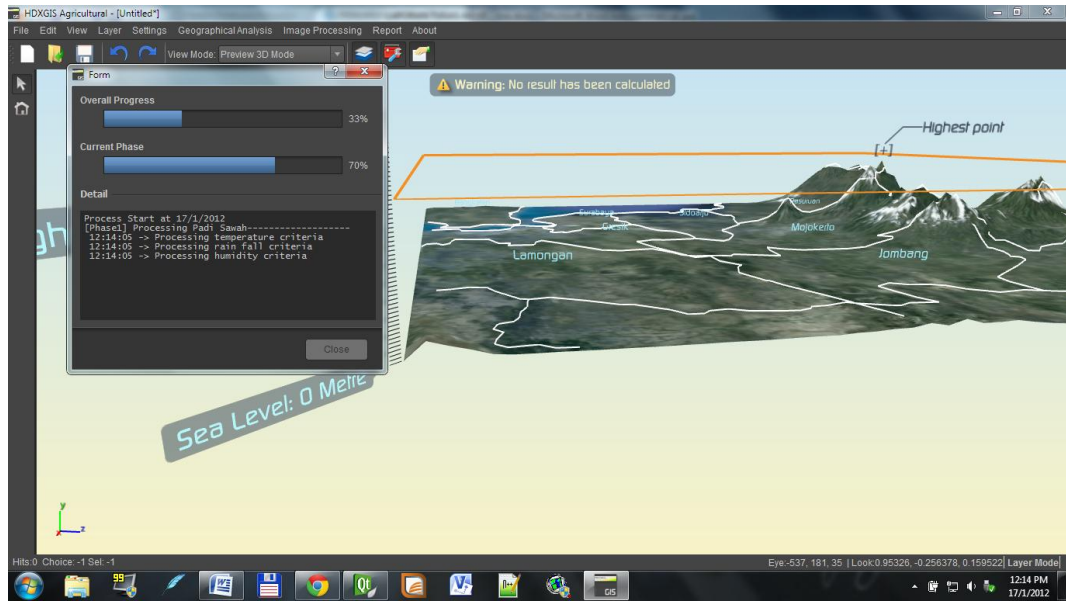
Gambar 4.48 Tampilan Wizard Page 2

Page terakhir adalah page 3 pada Gambar 4.49. pada *page* ini *user* akan diminta ingin menampilkan data yang ingin diproses. *User* diberi beberapa *geoprocessing* diantaranya adalah *Intersection*, *union* dan *subtract*. Setelah *query* diinputkan *user* dapat memilih hasil dari *query* dan akan dilakukan proses selanjutnya. Akan dicetak, dilakukan proses *export* atau dilakukan proses *view* dalam bentuk 3D.



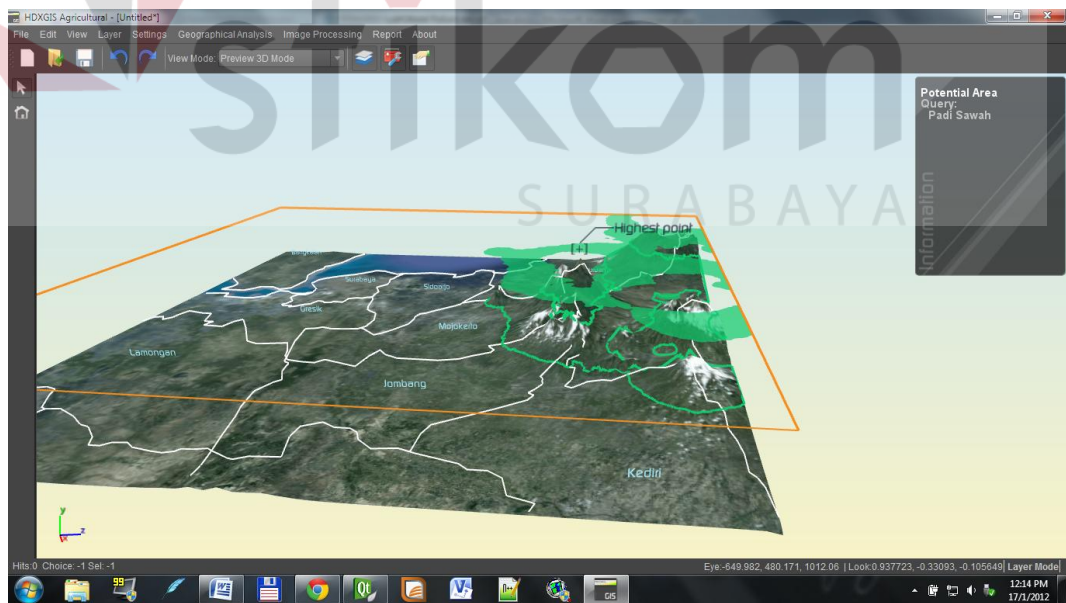
Gambar 4.49 Tampilan *Query Page* Pada Wizard

Ketika *user* menekan tombol *finish wizard* *page* akan menutup dengan sendirinya dan proses *geoprocessing* dari *query* yang sebelumnya telah diinputkan oleh *user* akan dijalankan, ini dapat dilihat dari *window* yang akan muncul. Yaitu *form* Analysis Result. *Form* ini menampilkan proses dari *geoprocessing* yang telah diinputkan sebelumnya. Tampilan dari *Analisis Result* dapat dilihat pada Gambar 4.50 di halaman 138.



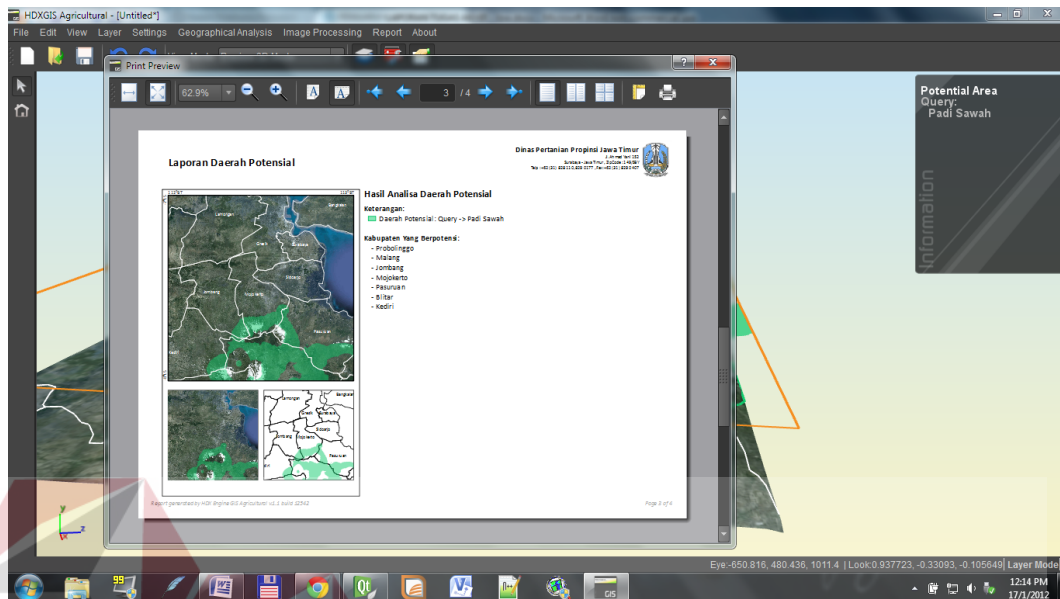
Gambar 4.50 Tampilan *Analysis Process*

Ketika proses telah selesai maka hasil akan dapat dilihat pada Gambar 4.51 dimana daerah potensial ditampilkan dalam bentuk tiga dimensi.



Gambar 4.51 Tampilan Hasil Daerah Analisa

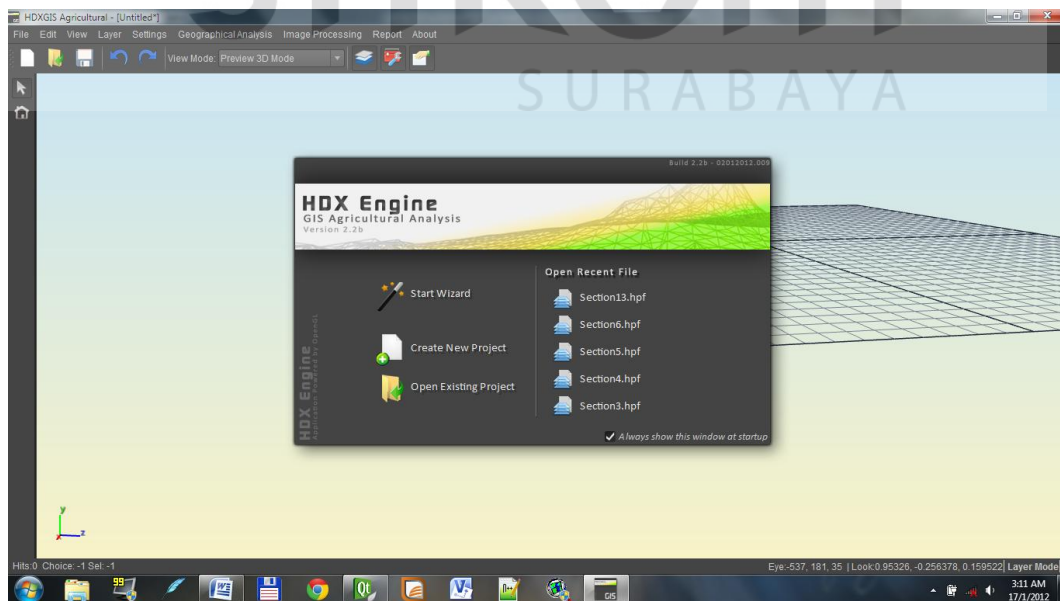
Hasil dari *print preview* daerah analisa dapat dilihat pada *page print preview*. *Form print preview* dapat dilihat pada Gambar. 4.52.



Gambar 4.52 Tampilan Hasil *Print Preview*

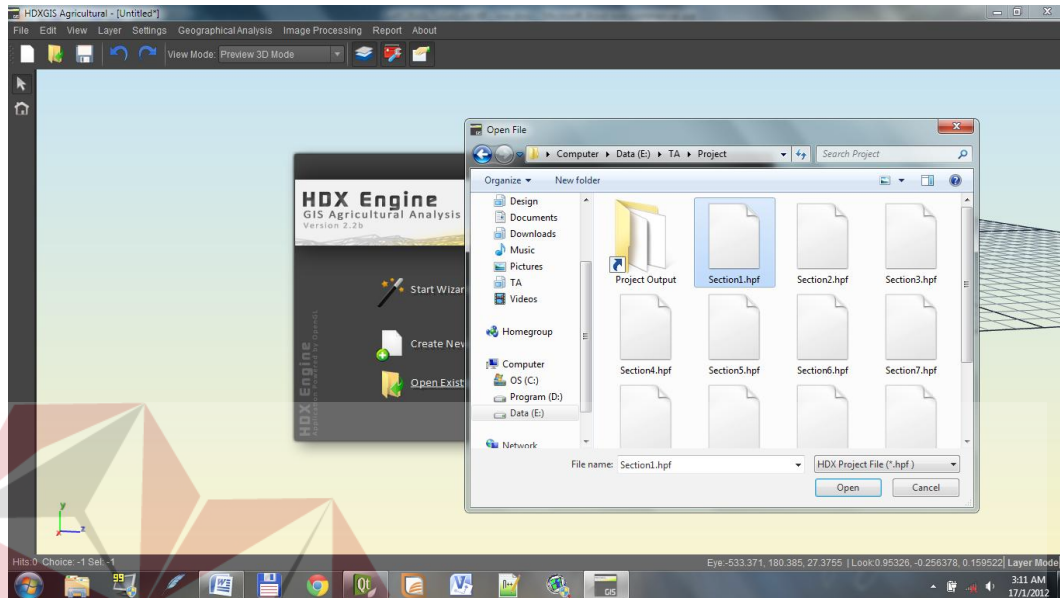
B. Hasil Evaluasi Visualisasi 3D

Tampilan *Splashscreen* pada saat aplikasi saat pertama kali dijalankan dapat dilihat pada Gambar 4.53.



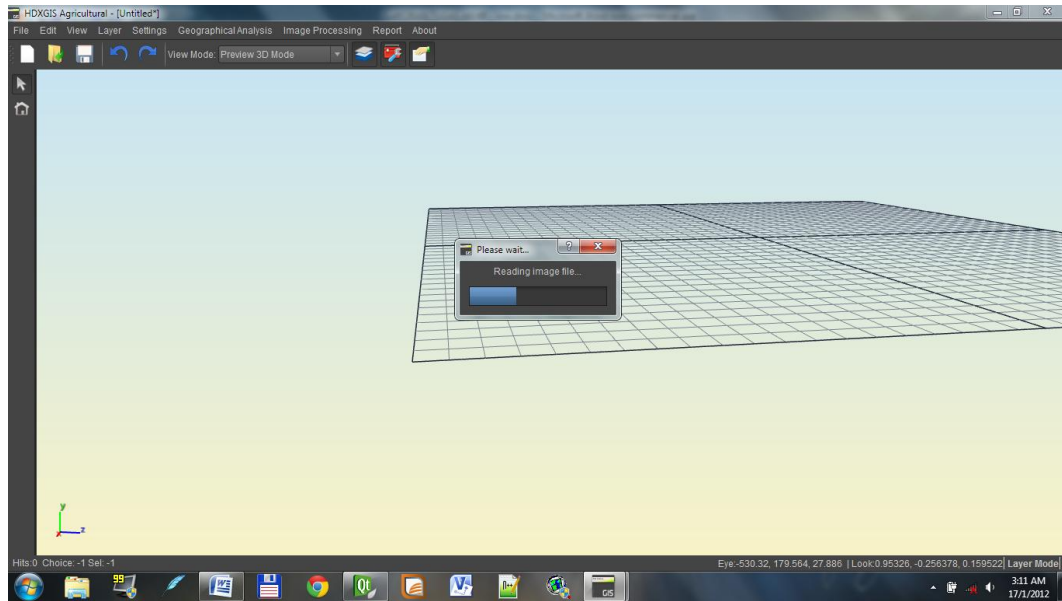
Gambar 4.53 Tampilan Aplikasi Saat Pertama Kali Dibuka.

Setelah *splashscreen* muncul kemudia *user* akan memilih *open project* dan akan muncul dialog *open project* seperti pada Gambar 4.54.



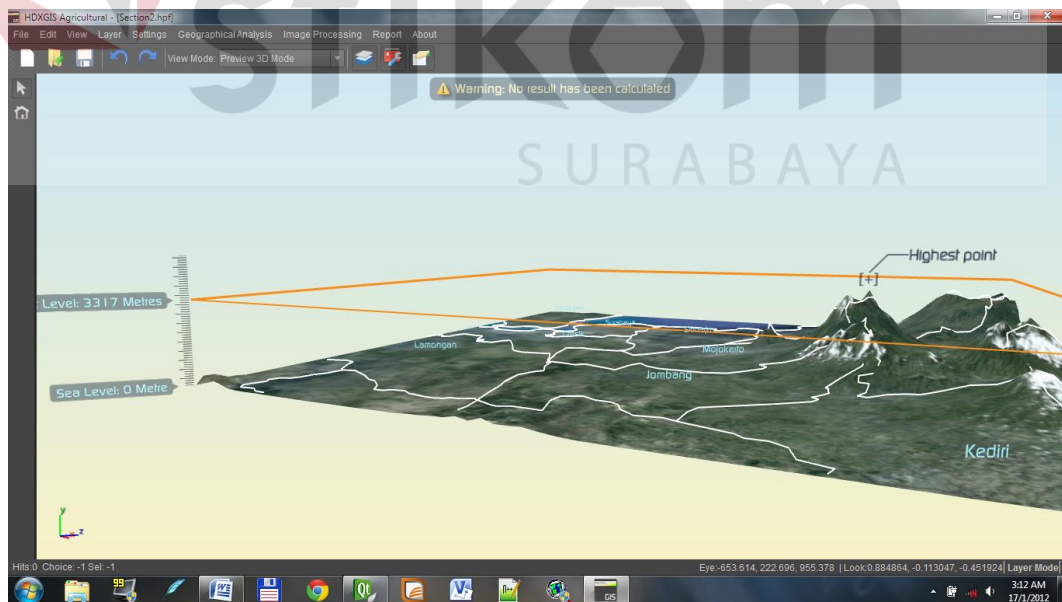
Gambar 4.54 Tampilan Aplikasi Saat Membuka *Project*

Setelah *project* yang ingin dibuka dipilih kemudian *user* menekan tombol *open* aplikasi akan menampilkan *progress dialog* yang menampilkan proses *loading data* seperti pada Gambar 4.55.



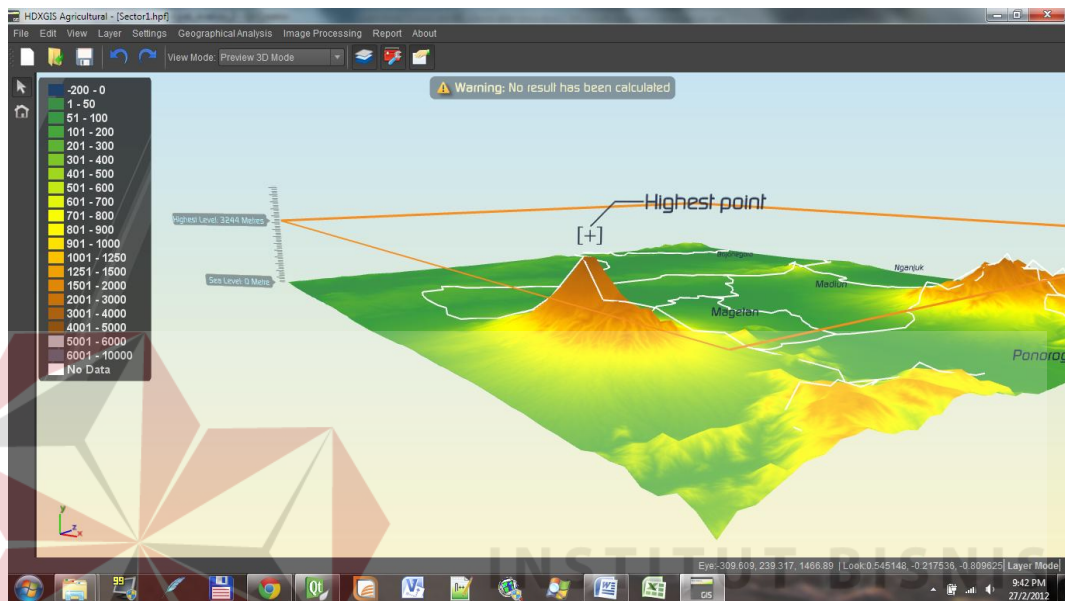
Gambar 4.55 Tampilan Aplikasi Saat Proses Berlangsung

Tampilan setelah proses selesai sistem akan menampilkan peta ketinggian dalam bentuk tiga dimensi. Tampilan tiga dimensi dapat dilihat pada Gambar 4.56.

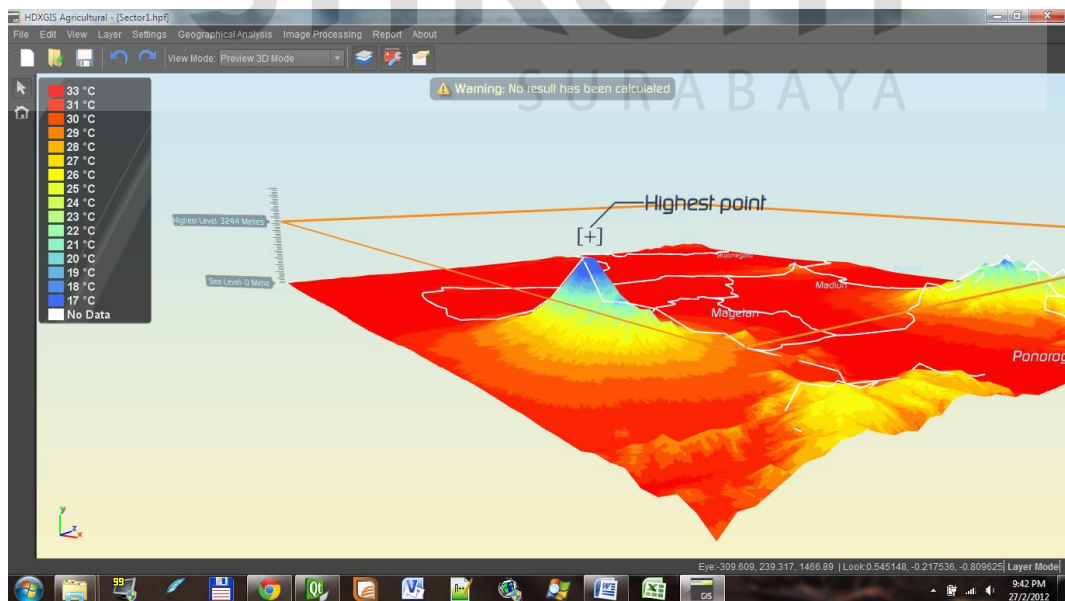


Gambar 4.56 Tampilan Aplikasi Saat Menampilkan Visualisasi 3D

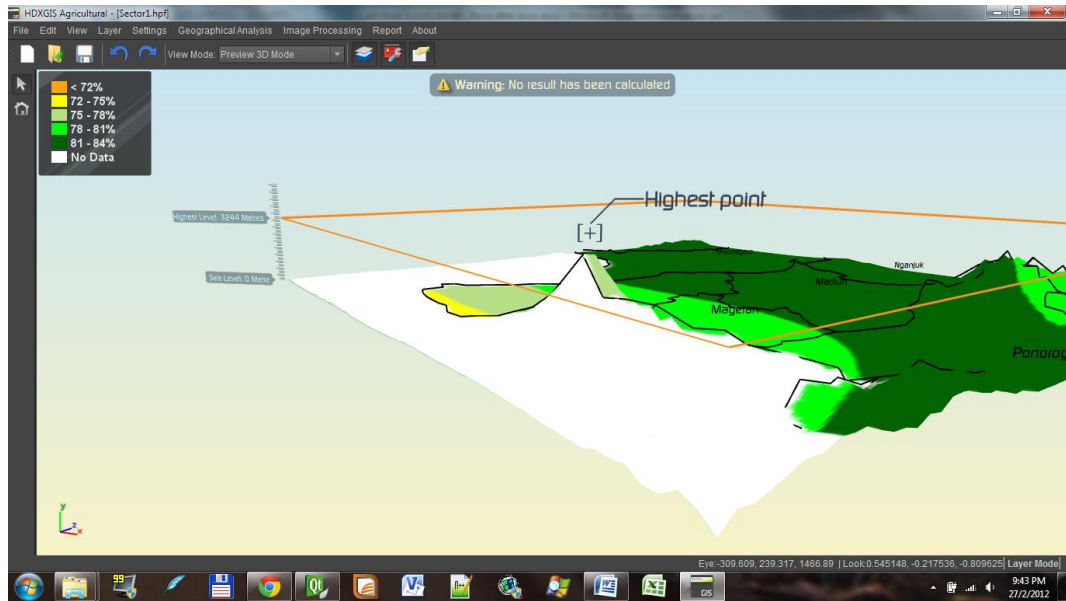
Visualisasi ditampilkan dalam beberapa permukaan peta yaitu tampilan permukaan ketinggian yang dapat dilihat pada Gambar 4.57. Tampilan permukaan suhu pada Gambar 4.58. tampilan kelembaban pada Gambar 4.59 di halaman 143 dan tampilan permukaan curah hujan pada Gambar 4.60 di halaman 143.



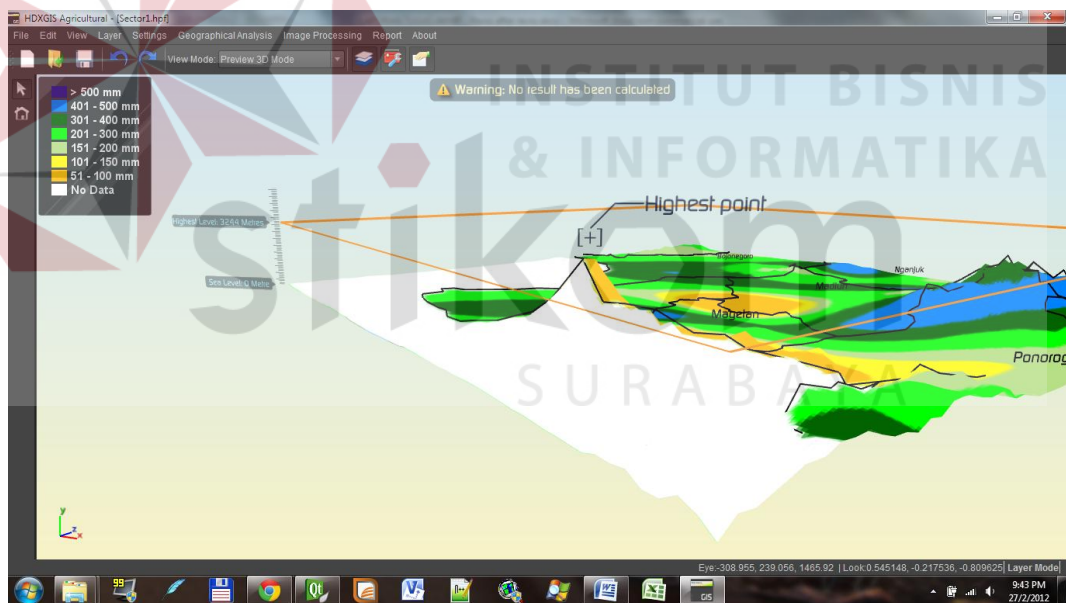
Gambar 4.57 Tampilan Visualisasi 3D Menampilkan Ketinggian



Gambar 4.58 Tampilan Visualisasi 3D Menampilkan Suhu



Gambar 4.59 Tampilan Visualisasi 3D Menampilkan Kelembaban

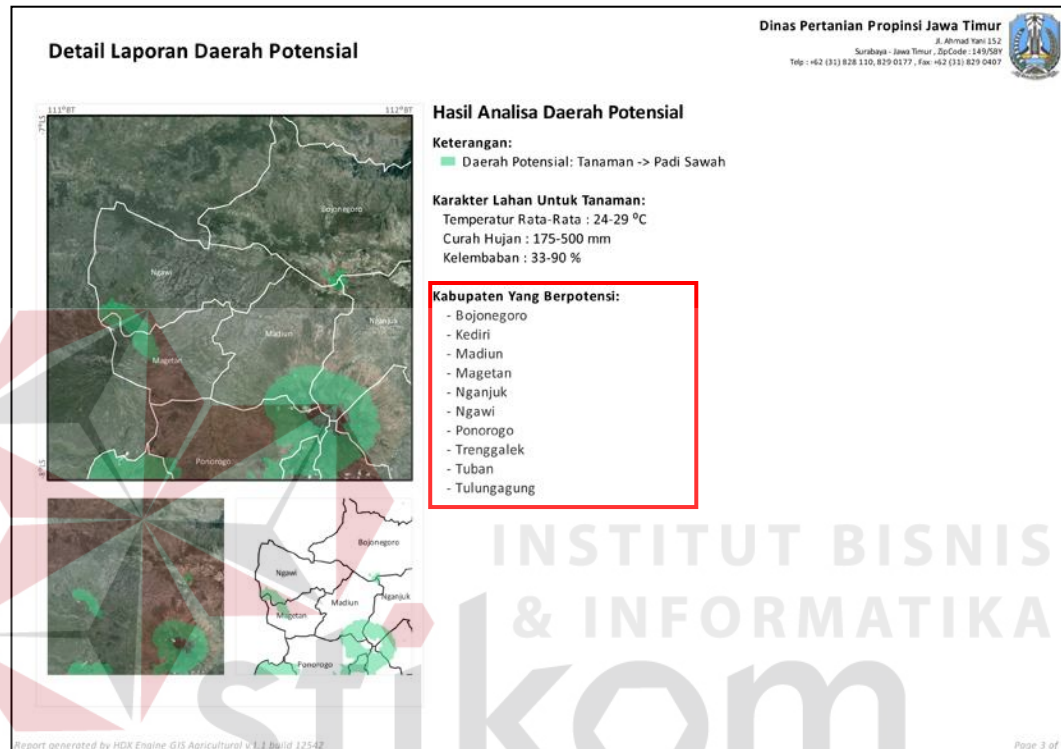


Gambar 4.60 Tampilan Visualisasi 3D Menampilkan Curah Hujan

Dari hasil visualisasi 3D yang dapat dilihat pada Gambar 4.56 sampai Gambar 4.60 Sistem berhasil menampilkan visualisasi 3D dari *project* yang telah dibuka. maka sistem ini lulus pada uji coba visualisasi 3D.

C. Hasil Evaluasi Tingkat Keakuratan Penentuan Lahan

Uji coba ini dilakukan untuk membandingkan hasil output dari sistem dengan data yang ada pada Dinas Pertanian Propinsi Jawa Timur. Hasil dari sistem dapat dilihat dari laporan yang dihasilkan. Tampilan dari laporan sistem.



Gambar 4.62 Hasil Output Sistem Kabupaten

Berikut ini adalah salah satu contoh uji coba untuk tanaman Bawang Merah pada setiap kabupaten di Jawa Timur. Hasil sistem diperoleh dari proses analisa 7 lokasi yang merupakan wilayah Jawa Timur. Hasil perbandingan dapat dilihat pada Tabel 4.1

Tabel 4.1 Kabupaten Daerah Potensial Untuk Bawang Merah

Data Dinas Pertanian	Poin	Kriteria Tanaman S1			Kriteria Tanaman S3		
		Hasil Sistem	Poin	Hasil	Hasil Sistem	Poin	Hasil
Banyuwangi	1	Banyuwangi	1	1	Banyuwangi	1	1
Blitar	0	Blitar	1	0	Blitar	1	0
Bojonegoro	1		0	0	Bojonegoro	1	1
Bondowoso	1		0	0	Bondowoso	1	1

Tabel 4.1(Lanjutan)

Data Dinas Pertanian	Poin	Kriteria Tanaman S1			Kriteria Tanaman S3		
		Hasil Sistem	Poin	Hasil	Hasil Sistem	Poin	Hasil
Gresik	1		0	0	Gresik	1	1
Jember	1	Jember	1	1	Jember	1	0
Jombang	1	Jombang	1	1	Jombang	1	1
Kediri	1	Kediri	1	1	Kediri	1	1
Lamongan	1		0	0	Lamongan	1	1
Lumajang	1	Lumajang	1	1	Lumajang	1	1
Madiun	1	Madiun	1	1	Madiun	1	1
Magetan	1		0	0	Magetan	1	1
Malang	1	Malang	1	1	Malang	1	1
Mojokerto	1	Mojokerto	1	1	Mojokerto	1	1
Nganjuk	1	Nganjuk	1	1	Nganjuk	1	1
Ngawi	1		0	0	Ngawi	1	1
Pamekasan	1		0	0	Pamekasan	1	1
Pasuruan	1	Pasuruan	1	1	Pasuruan	1	1
Ponorogo	1	Ponorogo	1	1	Ponorogo	1	1
Probolinggo	1	Probolinggo	1	1	Probolinggo	1	1
Sampang	1		0	0	Sampang	1	1
Sidoarjo	1		0	0	Sidoarjo	1	1
Situbondo	1	Situbondo	1	1	Situbondo	1	1
Sumenep	1		0	0	Sumenep	1	1
Surabaya	0		0	1	Surabaya	1	1
Trenggalek	1	Trenggalek	1	1	Trenggalek	1	1
Tuban	1		0	0	Tuban	1	1
Tulungagung	1	Tulungagung	1	1	Tulungagung	1	1
TOTAL	28			16			0
Prosentase				57%			93%

Dari hasil ujicoba untuk tanaman Bawang Merah pada Tabel 4.1 di halaman 145 dapat diambil kesimpulan bahwa Sistem informasi ini mampu menentukan daerah potensial untuk tanaman bawang merah dengan tingkat keakuratan sebesar **57%** untuk kriteria tanaman S1 dan **93%** untuk kriteria tanaman S3.

Proses uji coba dilakukan hingga tanaman yang berjumlah 10 selesai diuji coba dan diambil kesimpulan akhir. Untuk data hasil pengujian ke sepuluh

tanaman dapat dilihat pada lampiran 22 – 31 di halaman 172 – 181. Setelah kesepuluh tanaman selesai dilakukan uji coba maka diambil kesimpulan akhir dari aplikasi ini yaitu berupa tingkat keakuratan daerah potensial yang dihasilkan oleh sistem.

Tabel 4.2 Kesimpulan Akhir Uji Tingkat Keakuratan

Nama Tanaman	Kategori	Tingkat Keakuratan	
		Kriteria S1	Kriteria S3
Apel	Buah-buahan	96%	96%
Bawang Merah	Sayuran	57%	93%
Jagung	Serelia	64%	93%
Jahe	Tanaman Obat	61%	100%
Kacang Tanah	Kacang-kacangan	57%	89%
Kedelai	Kacang-kacangan	57%	93%
Mangga	Buah-buahan	64%	100%
Mawar	Bunga	57%	57%
Padi Sawah	Serelia	75%	100%
Tomat	Sayuran	57%	89%
Rata - Rata		65%	91%

Dari hasil perhitungan Tabel 4.2 kesimpulan akhir, maka dapat diambil kesimpulan Rata-rata tingkat keakuratan Sistem Informasi Geografis penentuan lahan potensial menggunakan *Image Processing* dalam menentukan daerah atau wilayah potensial produk pertanian di Propinsi Jawa Timur sebesar **65%** dengan kriteria tanaman S1 dan **91%** untuk wilayah kriteria tanaman S3. Dapat dilihat pada tabel, hasil ujicoba untuk kriteria S3 lebih besar dibandingkan dengan uji coba untuk kriteria S1, hal ini dikarenakan kriteria S1 lebih presisi dan lebih spesifik. Dengan demikian Sistem Informasi Geografis ini dianggap benar karena hasil output untuk kriteria S3 lebih besar dari S1.

Hasil tidak mencapai 100% hal ini dikarenakan dalam tugas akhir ini kriteria yang digunakan hanya 3 (tiga) buah sedangkan untuk menentukan suatu

daerah potensial dibutuhkan jumlah kriteria sebanyak 25 (dua puluh lima). Kriteria karakter dapat dilihat pada Lampiran 2-11 di halaman 152-161. Untuk pengembangan aplikasi ini nanti dapat menggunakan seluruh kriteria yang berjumlah 25 (dua puluh lima) sehingga akan menghasilkan keakuratan hasil data yang mendekati 100%.

Dengan menggunakan tiga kriteria ini masih ada hasil daerah yang belum tepat. Sehingga dapat diambil kesimpulan bahwa hanya dengan tiga kriteria saja belum dapat sepenuhnya digunakan untuk melakukan penentuan daerah potensial. Kriteria yang harus digunakan sebanyak 25 buah yaitu: Temperatur udara, Curah hujan, Lamanya masa kering, Kelembaban udara, Drainase, Tekstur, Bahan kasar, Kedalaman tanah, Ketebalan gambut, Kematangan gambut, KTK liat, Kejenuhan basa, Reaksi tanah (pH), C-organik, Salinitas, Alkalinitas, Kedalaman bahan sulfidik, Lereng, Bahaya erosi, Genangan, Batuan di permukaan, Singkapan batuan, Sumber air tawar, Amplitudo pasang-surut, dan Oksigen.

