

BAB IV

HASIL DAN UJI COBA

IV.1 Hasil Program

Setelah melewati tahapan perancangan sistem, pada bab ini akan menjelaskan hasil dan pengujian sistem yang telah dibuat. Bab ini membahas tentang tampilan sistem, uji coba sistem, hasil pengujian, kelebihan sistem, dan kekurangan sistem.

IV.1.1 Tampilan *Form* Utama

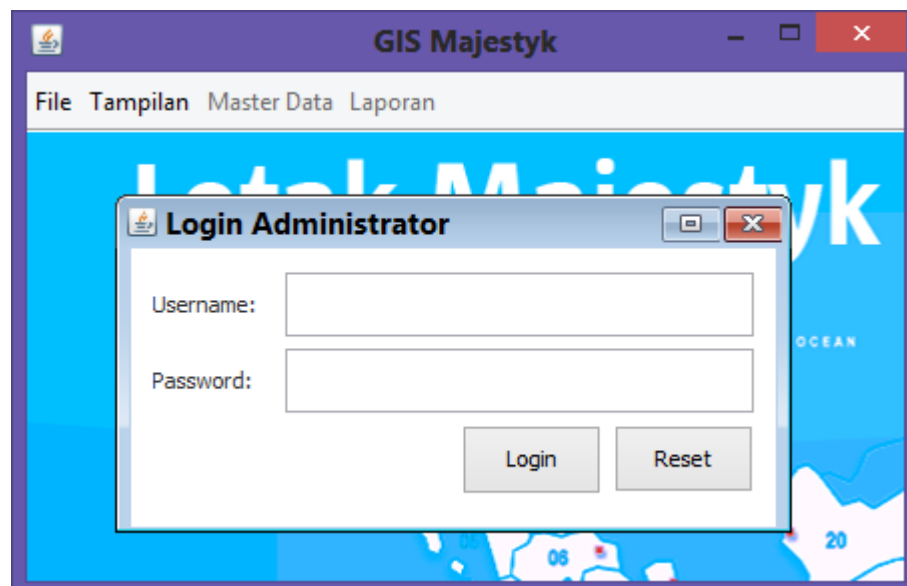
Form utama merupakan kelas aplikasi yang berada pada level *service*, dimana form utama ini menjadi *interface* atau penghubung antara pengguna sistem dengan *object* modul pengolah data yang ada pada paket sistem. Form utama ini memiliki beberapa menu yang berfungsi untuk men-*trigger* sistem dalam pemanggilan modul yang akan digunakan. Menu tersebut saat pertama kali aplikasi dijalankan masih dalam status *disable*, untuk mengaktifkannya pengguna harus menjalani fase otentikasi pengguna. Ini berguna untuk menjaga sekuritas sistem. Pada *form* utama dapat dilihat sebuah *panel* yang berisi komponen username dan komponen password. Panel tersebutlah yang berfungsi untuk mengotentikasi pengguna. Setelah proses otentikasi berhasil dilakukan dan hasilnya berupa otentikasi yang valid maka tombol-tombol menu tersebut akan aktif. Tampilan utama sistem sebelum dilakukannya otentikasi pengguna dapat dilihat pada gambar IV.1 :



Gambar IV.1 Form Utama

IV.1.2 Tampilan Form Login

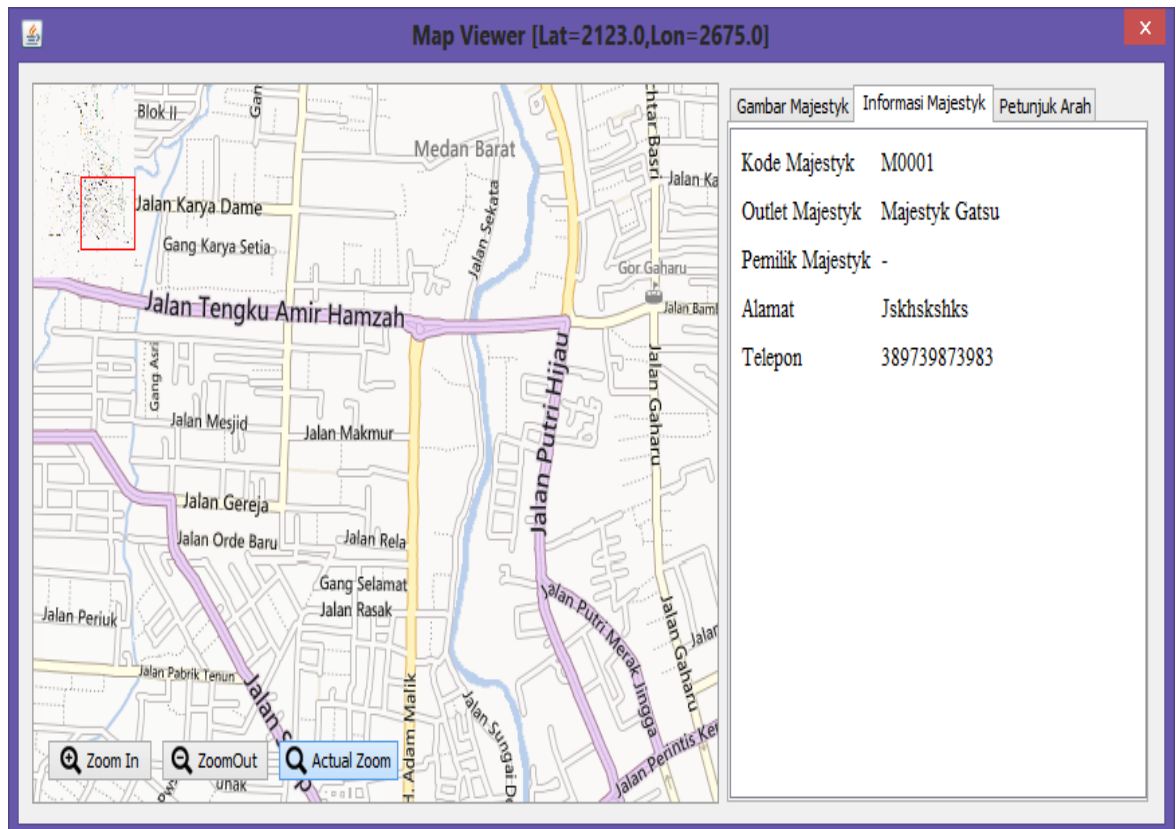
Tampilan *form* Login ini berfungsi untuk memastikan dan memvalidasi otoritas pengguna sebagai administrator yang dapat menjalankan semua modul pada sistem. Tampilan *form* tersebut dapat dilihat pada gambar IV.2 :



Gambar IV.2 Tampilan Form Login

IV.1.3 Tampilan *Form* Peta

Tampilan *form* peta ini berfungsi untuk penghubung pengguna dengan sistem dalam mengakses informasi *outlet* Majestyk yang terdapat pada peta yang disediakan oleh sistem. Tampilan *form* tersebut dapat dilihat pada gambar IV.3 :



Gambar IV.3 Tampilan *Form* Peta

IV.1.4 Tampilan *Form* Majestyk

Tampilan *form* Majestyk ini berfungsi untuk mengolah data yang berkaitan dengan Majestyk. Tampilan *form* tersebut dapat dilihat pada gambar IV.4 :

Kode Majestyk	Nama Outlet	Pemilik Franchise
M0001	Majestyk Gatsu	-

Gambar IV.4 Tampilan *Form Majestyk*

IV.1.5 Tampilan *Form Lokasi*

Tampilan *form* lokasi ini berfungsi untuk mengolah data lokasi majestyk yang akan ditampilkan pada peta. Tampilan *form* tersebut dapat dilihat pada gambar IV.5 :

Kode Lokasi	Kode Majestyk	Lat	Lon
L0001	M0001	435.833333333333	378.333333333333

Gambar IV.5 Tampilan *Form Lokasi*

IV.1.6 Tampilan *Form Arah*

Tampilan *form* arah ini berfungsi untuk mengolah data Arah, data ini akan ditampilkan saat pengguna menekan marka majestyk pada peta yang tujuannya agar pengguna mendapatkan informasi bagaimana cara menuju lokasi majestyk yang dipilihnya. Tampilan *form* tersebut dapat dilihat pada gambar IV.6 :

Kode Arah	Kode Majestyk	Petunjuk Arah
A0001	M0001	Dari jalan kapten muslim men...

Kode Arah: A0001

Kode Majestyk: M0001 Majestyk Gatsu

Petunjuk Arah: Dari jalan kapten muslim menuju pasar Sei Sikambing kemudian belok kiri menuju Plaza Medan Fair

Baru Simpan Edit Hapus Batal

Gambar IV.6 Tampilan *Form Arah*

IV.1.7 Tampilan *Form Info*

Tampilan *form* info ini berfungsi untuk mengolah data informasi majestyk, informasi yang akan ditampilkan saat pengguna memilih salah satu marka majestyk diantaranya yaitu informasi telepon, alamat dan gambar *outlet*. Tampilan *form* tersebut dapat dilihat pada gambar IV.7 :

Kode Info	Kode Majestyk	Gambar	Telepon	Alamat
I0001	M0001	C:/logo.png	389739873983	Jskhskshks

Gambar IV.7 Tampilan *Form Info*

IV.1.8 Tampilan *Form Admin*

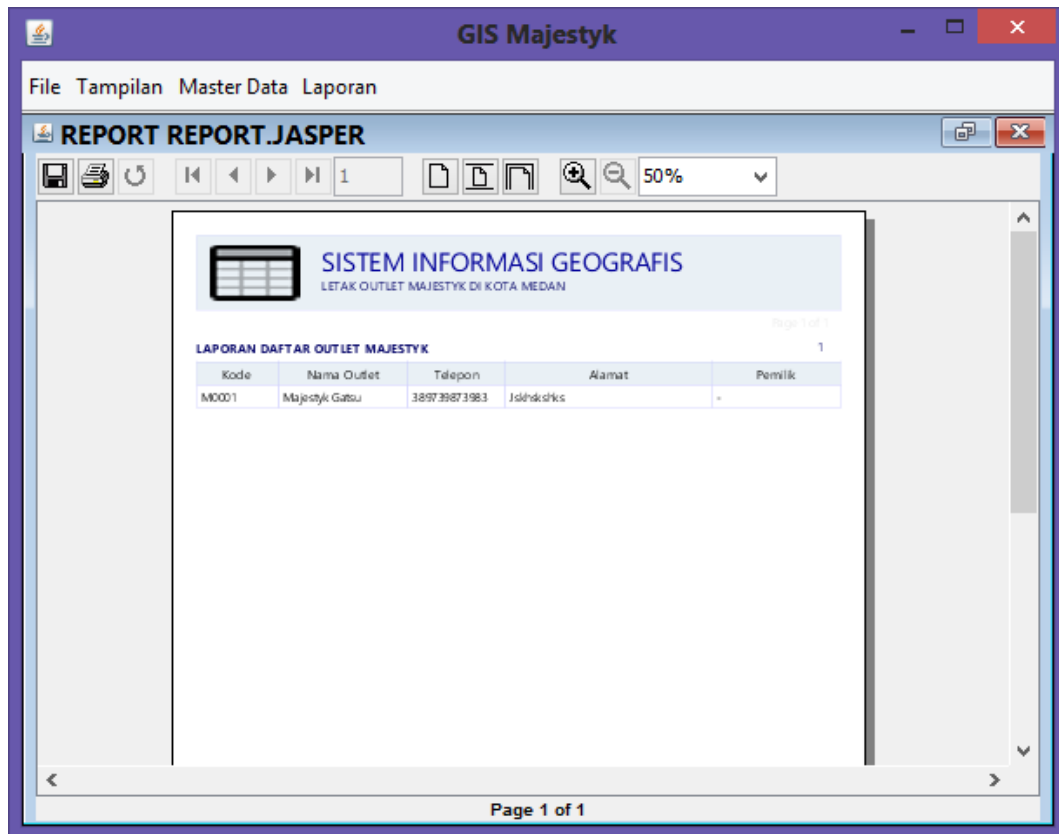
Tampilan *form* admin ini berfungsi untuk mengolah data administrator yang berhak mengakses sistem. Tampilan *form* tersebut dapat dilihat pada gambar IV.8:

Username	Password
Admin	admin
Oka	oka

Gambar IV.8 Tampilan *Form Admin*

IV.1.9 Tampilan *Form* Laporan Daftar *Outlet*

Tampilan *form* laporan dapat *outlet* ini berfungsi untuk menampilkan informasi daftar *outlet-outlet* apa saja yang terdapat di peta. Tampilan *form* tersebut dapat dilihat pada gambar IV.9 :



Gambar IV.9 Tampilan *Form* Laporan Daftar *Outlet*

IV.2 Uji Coba Program

Uji coba terhadap sistem bertujuan untuk memastikan bahwa sistem sudah berada pada kondisi siap pakai. Instrumen yang digunakan untuk melakukan pengujian ini yaitu dengan menggunakan :

1. Satu unit laptop dengan spesifikasi sebagai berikut :
 - a. Processor Intel Core I3

- b. Memory 4 Gb
 - c. Hardisk 500 Gb
2. Perangkat Lunak Netbeans dan Java Runtime Environment dengan spesifikasi sebagai berikut:
- a. Netbeans IDE versi 7.2
 - b. JRE versi 7 update 29

IV.2.1 Uji Coba Kinerja Sistem

Uji coba kinerja sistem digunakan untuk mengetahui kinerja sistem saat dijalankan. Pengujian ini menggunakan *profiling* pada aplikasi Netbeans 7.2, hasil pengujian kinerja sistem dapat dilihat pada tabel IV.1 :

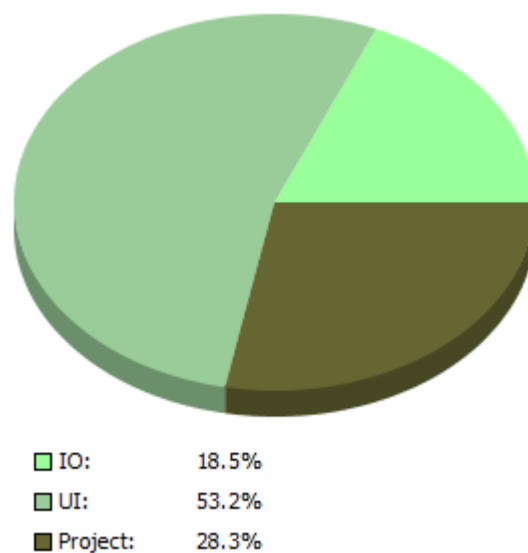
Tabel IV.1 Pengujian Hotspot Call Tree

Hot Spots – Method	%	Self time
com.geo.ok.NavigatorMap.<init>()	18.28%	809.486 ms
com.app.gui.AdminManager.initComponents()	8.78%	388.973 ms
com.app.main.MainProgram.tampilPetaActionPerformed(java.awt.event.ActionEvent)	5.27%	233.566 ms
com.app.main.MainProgram.initComponents()	2.18%	96.587 ms
com.geo.ok.MapViewer.initComponents()	2.17%	96.23 ms
com.app.gui.AdminManager.<init>()	1.55%	68.744 ms
com.app.gui.ArahManager.initComponents()	0.93%	41.14 ms
com.app.main.MainProgram.<init>()	0.86%	37.869 ms
com.app.gui.MajestykManager.initComponents()	0.85%	37.655 ms
com.app.gui.InfoManager.initComponents()	0.75%	33.22 ms
com.geo.ok.MapPicker.initComponents()	0.73%	32.227 ms
com.app.gui.LokasiManager.initComponents()	0.64%	28.399 ms
com.app.main.MainProgram.main(String[])	0.58%	25.765 ms
com.app.util.Report.<init>()	0.49%	21.728 ms
com.app.gui.LoginManager.initComponents()	0.46%	20.298 ms
com.app.gui.LokasiManager.<init>()	0.43%	19.155 ms
com.app.gui.MajestykManager.<init>()	0.41%	18.184 ms
com.geo.ok.NavigatorMap.paintComponent(java.awt.Graphics)	0.41%	18.132 ms
com.geo.ok.MapViewer.navigatorMapMouseMoved(java.awt.event.MouseEvent)	0.41%	18.103 ms
com.app.gui.ArahManager.<init>()	0.39%	17.327 ms
com.geo.ok.MarkerLoader.getMarkers()	0.36%	15.988 ms
com.app.main.MainProgram\$12.run()	0.34%	14.885 ms
com.app.gui.InfoManager.<init>()	0.26%	11.416 ms
com.app.main.MainProgram.show(javax.swing.JInternalFrame, javax.swing.JInternalFrame)	0.25%	11.031 ms
com.app.gui.LoginManager.<init>()	0.25%	10.972 ms
com.app.util.Desktop.paintComponent(java.awt.Graphics)	0.13%	5.909 ms
com.geo.ok.NavigatorMap.setMarkers(java.util.List)	0.04%	1.596 ms
com.geo.ok.NavigatorMap.panelToImageCoords(java.awt.Point)	0.03%	1.399 ms
com.geo.ok.Coords.getIntX()	0.03%	1.375 ms
com.geo.ok.NavigatorMap.addWheelZoomDevice()	0.02%	0.985 ms
com.geo.ok.Coords.<init>(double, double)	0.02%	0.846 ms
com.app.util.ImageViewer.<init>()	0.01%	0.586 ms
com.geo.ok.NavigatorMap\$2.mouseMoved(java.awt.event.MouseEvent)	0.01%	0.553 ms
com.geo.ok.NavigatorMap.imageToPanelCoords(com.geo.ok.Coords)	0.01%	0.482 ms

com.geo.ok.MapViewer.access\$000(com.geo.ok.MapViewer, java.awt.event.MouseEvent)	0.01%	0.435 ms
com.geo.ok.NavigatorMap.getScreenNavImageWidth()	0.01%	0.352 ms
com.geo.ok.NavigatorMap\$5.mouseMoved(java.awt.event.MouseEvent)	0.01%	0.306 ms
com.geo.ok.MapViewer\$1.mouseMoved(java.awt.event.MouseEvent)	0.01%	0.297 ms
com.geo.ok.NavigatorMap.isFullImageInPanel()	0.01%	0.296 ms
com.geo.ok.NavigatorMap.getScreenImageWidth()	0.01%	0.296 ms
com.geo.ok.NavigatorMap.createNavigationImage()	0.01%	0.236 ms
com.geo.ok.NavigatorMap.getScreenNavImageHeight()	0.00%	0.207 ms
com.geo.ok.NavigatorMap\$5.mouseDragged(java.awt.event.MouseEvent)	0.00%	0.194 ms
com.geo.ok.NavigatorMap.<clinit>	0.00%	0.176 ms
com.geo.ok.NavigatorMap.getScreenImageHeight()	0.00%	0.146 ms
com.app.util.ImageViewer.paintComponent(java.awt.Graphics)	0.00%	0.128 ms
com.app.main.MainProgram.menuMajestykActionPerformed(java.awt.event.ActionEvent)	0.00%	0.002 ms
com.app.main.MainProgram.menuLoginActionPerformed(java.awt.event.ActionEvent)	0.00%	0.002 ms
com.geo.ok.NavigatorMap.access\$1602(com.geo.ok.NavigatorMap, java.awt.Dimension)	0.00%	0.001 ms
com.geo.ok.NavigatorMap.removeButtonZoomDevice()	0.00%	0.001 ms

Dari pengujian profiling kinerja sistem yang ditunjukkan pada tabel IV.1 di atas, terlihat bahwa proses start time aplikasi membutuhkan ± 3.5 detik, waktu yang paling digunakan adalah saat membuat objek Main dan objek MapNavigator yang masing-masingnya membutuhkan waktu 800.288 ms dan 619.952 ms.

Secara garis besar penggunaan sumber daya perangkat yang digunakan dibagi menjadi tiga kategori, penggunaan sumber daya tersebut dapat dilihat pada grafik gambar IV.10 :



Gambar IV.10 Grafik Penggunaan Sumber Daya

IV.2.2 Uji Coba Sekuritas Sistem

Uji coba sekuritas bertujuan untuk memastikan bahwa hanya pengguna yang memiliki wewenang yang dapat mengakses modul sistem, dengan demikian informasi yang disajikan dapat dipertanggungjawabkan. Berikut ini akan dicoba beberapa teknik untuk menembus proses otentikasi sistem :

1. Menembus otentikasi dengan konsep SQL-Injection.

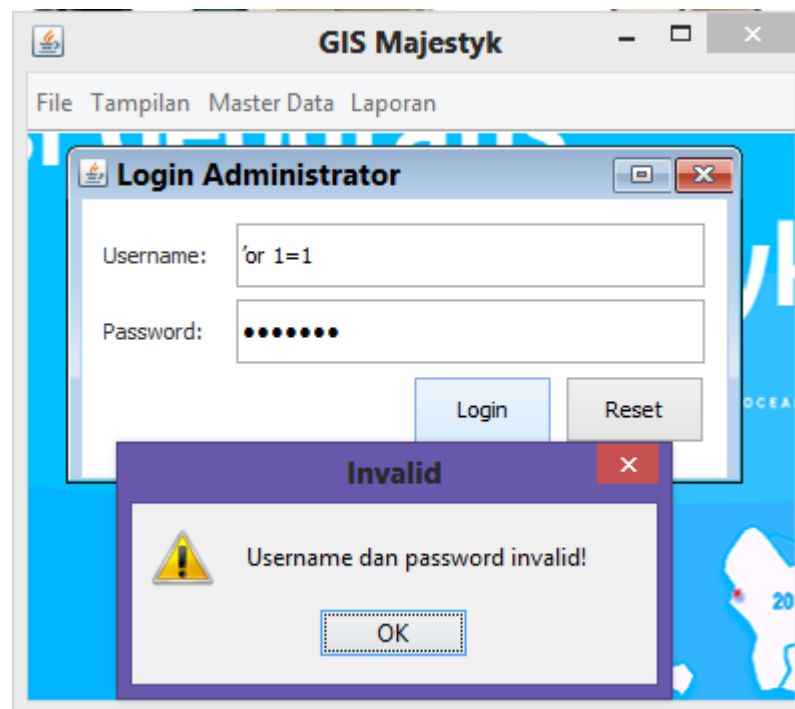
Konsep ini dilakukan dengan cara memanipulasi masukan sistem melalui parameter tertentu. Berikut ini adalah cara menembus dengan SQL-Injection:

- a. Memasukkan *username* “' or 1=1”
- b. Memasukkan parameter *password* “' or 1=1”

Dengan parameter diatas, diasumsikan untuk memeriksa ada atau tidaknya pengguna yang dimaksud yaitu dengan mengeksekusi *query* sebagai berikut:

```
SELECT * FROM admin WHERE username=' ' or 1=1 AND
password=' ' or 1=1
```

Seharusnya pada *query* diatas akan menampilkan semua *record* yang ada pada tabel admin, karena pada *query* tersebut selalu bernilai *true* dengan kondisi bahwa *field* `username=' '` atau `1=1` (satu sama dengan satu ada benar), begitu juga dengan *field* `password`. Dengan kata lain, sistem dapat ditembus tanpa harus mengetahui nama pengguna dan kata sandi yang *valid*. Teknik ini dapat dilihat pada gambar IV.11 :



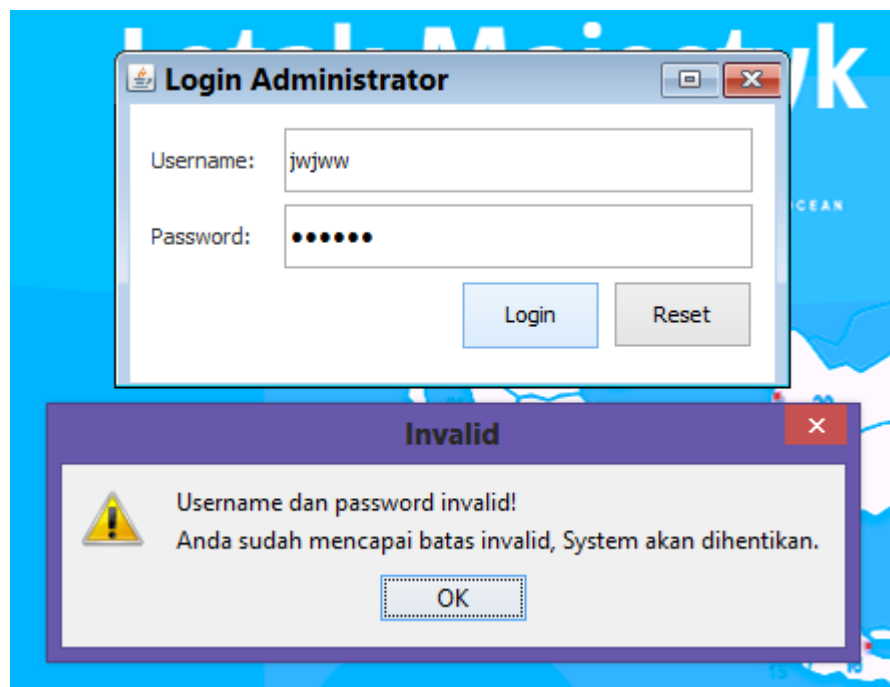
Gambar IV.11 Tampilan Konsep Penembusan Sistem Dengan SQLi

seperti yang dilihat pada gambar IV.11, ternyata teknik penembusan sistem dengan konsep SQLi tidak berhasil. Dengan demikian sistem yang telah dirancang ini aman terhadap serangan peretas yang menggunakan teknik SQLi.

2. Menembus otentikasi dengan teknik Brute Force.

Teknik Brute Force sebenarnya merupakan teknik alternatif terakhir yang dilakukan setelah teknik-teknik yang ada tidak berhasil dilakukan. Teknik ini bekerja dengan membutuhkan waktu yang sangat lama, secara umum konsepnya yaitu dengan menebak-nebak nama pengguna dan kata sandi. Teknik ini biasanya dibantu dengan aplikasi-aplikasi *hacking*, namun pada pengujian ini dilakukan secara manual, setelah 3 kali mencoba memasukkan nama pengguna dan kata sandi secara sembarang, aplikasi menampilkan

peringatan bahwa pengguna sudah mencapai batas memasukkan data akun dengan status invalid, setelah menekan tombol OK, sistem akan keluar, peringatan tersebut dapat dilihat pada gambar IV.12 :



Gambar IV.12 Tampilan Peringatan Batas Invalid Data Akun

Dengan adanya batasan tersebut, maka teknik penembusan otentikasi ini tidak akan bekerja meskipun dengan bantuan aplikasi *hacking*.

Setelah dilakukannya pengujian sekuritas diatas, maka dapat disimpulkan bahwa sistem ini memiliki keamanan yang tidak dapat ditembus dengan teknik-teknik *hacking* yang umum dilakukan.

IV.2.3 Hasil Uji Coba

Setelah melakukan uji coba terhadap sistem, maka dapat disimpulkan hasil yang didapatkan yaitu :

1. Sistem memiliki *performance* yang relatif stabil.
2. Sistem hanya membutuhkan ± 3.5 detik untuk melakukan *start-up*.

3. Sistem telah menghasilkan informasi yang *valid*.
4. Peta yang ditampilkan dapat di-*render* dengan baik dan cepat.
5. Antarmuka yang sederhana dapat mempermudah pengguna dalam mempelajari sistem ini.
6. Kebutuhan akan informasi laporan sangat cepat disajikan.
7. Keamanan sistem cukup terjamin dari resiko peretasan dengan teknik yang biasa dilakukan para peretas.

IV.3 Kelebihan dan Kekurangan Sistem

Setiap sistem memiliki kelebihan dan kekurangan, berikut ini adalah kelebihan dan kekurangan sistem yang telah dibuat.

IV.3.1 Kelebihan Sistem

Kelebihan sistem ini diantaranya yaitu:

1. Waktu yang dibutuhkan untuk proses *start-up* relatif singkat.
2. *Performance* sistem relatif stabil.
3. Sistem mampu menghasilkan informasi yang sesuai dengan yang diharapkan.
4. Sistem sangat cepat mengolah data untuk menghasilkan berbagai informasi yang dibutuhkan pengguna.
5. Sistem ini telah diproteksi dengan sistem pengamanan berbasis otentikasi.

IV.3.2 Kekurangan Sistem

Adapun kekurangan sistem yang telah dibuat diantaranya yaitu:

1. Sistem belum memiliki *splash-screen*, sehingga terkadang pengguna mengira bahwa aplikasi tidak berjalan karena menunggu beberapa detik.

2. Sistem ini belum memiliki modul pengolahan data spasial yang lengkap.
3. Sistem ini masih memiliki batas kualitas, dimana saat pengguna memperbesar peta, peta yang di-*render* akan terlihat kabur.
4. Sistem ini masih memiliki beberapa *bugs* yang terkadang muncul.
5. Antarmuka yang dibuat masih mendominasi penggunaan sumber daya komputer, yakni sebesar 52% yang seharusnya dapat diminimalisir dengan menggunakan beberapa teknik *performance tuning*.