**Portofolio Project IT**

Albertus Kelvin

**Aplikasi Kompresor dan Dekompressor “miniZIPXX”**

**Tahun Pengerjaan :** 2016

**Jenis Pengerjaan :** Kelompok

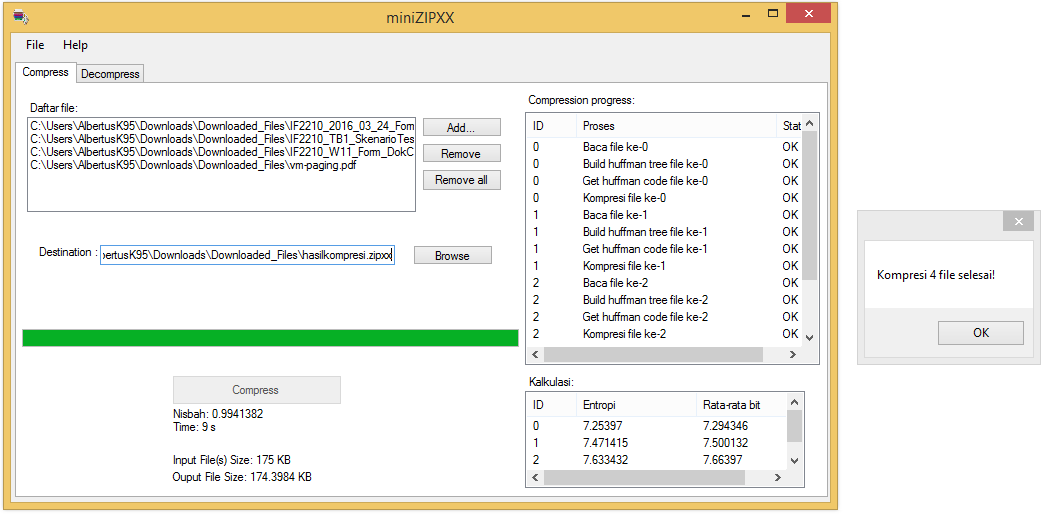
**Tujuan Project :** Membuat suatu aplikasi yang dapat melakukan kompresi terhadap suatu file agar proses pengiriman file menjadi lebih efektif dengan ukuran file yang lebih kecil

**Keterangan :** Project ini bernama “miniZIPXX” dimana file hasil kompresi akan memiliki ekstensi “zipxx”. File yang akan di kompresi dapat berupa *single file* maupun *multiple files.* Kami membuat project ini dalam bahasa pemrograman C++ dan menggunakan IDE Visual Studio.

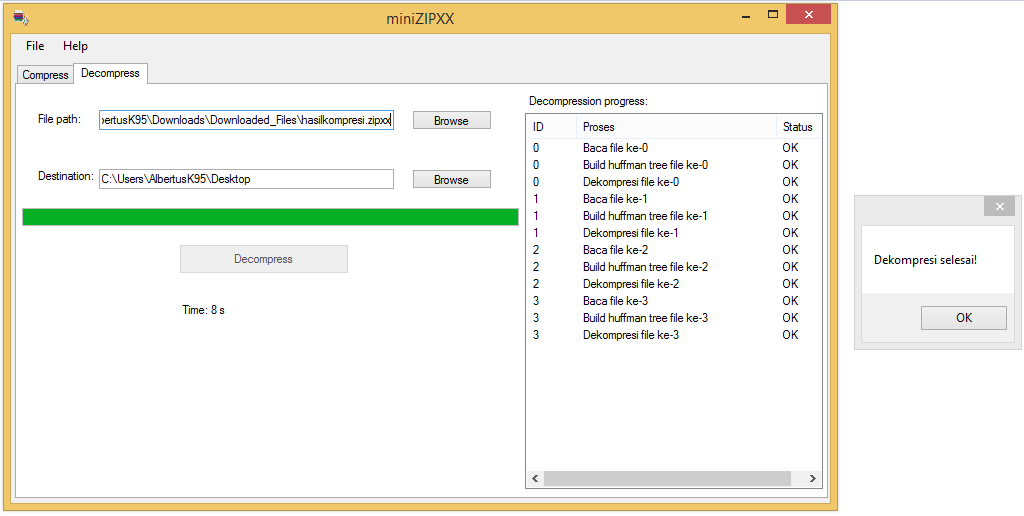
**Cara kerja aplikasi :**

1. User dapat memilih jenis aksi yang akan dilakukan terhadap suatu file, apakah *compress* (melakukan kompresi) atau *decompress* (melakukan dekompresi).
2. Jika memilih *compress*, maka user akan memilih kumpulan file (*single* atau *multiple*) dan semua file yang dipilih itu akan ditampilkan oleh aplikasi untuk konfirmasi. User juga akan memilih lokasi tujuan untuk menyimpan file hasil kompresi. Setelah itu, user dapat memulai kompresi dengan menekan tombol “Compress”. Aplikasi akan menampilkan status kompresi dalam sebuah tabel dan *progress bar*. Setelah proses kompresi selesai, aplikasi akan memunculkan informasi mengenai nilai entropi, rata-rata bit, nisbah, dan waktu kompresi yang diperlukan.
3. Jika memilih *decompress,* maka user akan memilih file hasil kompresi (ekstensi zipxx) dan lokasi tujuan untuk menyimpan file-file yang sudah di dekompresi. User dapat memulai dekompresi file dengan menekan tombol “Decompress”. Aplikasi akan menampilkan status dekompresi dalam sebuah tabel dan waktu yang diperlukan untuk melakukan dekompresi.

**Tampilan project :**

Kompresi

Dekompresi



**“SAIMemo” Aplikasi *Cloud Storage* Sederhana**

**Tahun pengerjaan :** 2016

**Jenis pengerjaan :** Individu

**Tujuan project :** Membuat sebuah *cloud storage* sederhana yang memungkinkan penggunanya untuk melakukan hal-hal dasar terhadap suatu file seperti *upload, download,* membuat folder baru, menghapus file, dll. Semua file yang disimpan pengguna ke dalam *cloud storage* ini akan dapat diakses kembali meskipun pengguna sudah melakukan aksi *log out*.

**Keterangan :** *Cloud storage* ini bernama SAIMemo. Berikut fitur yang dimiliki beserta penjelasan singkatnya.

**Informasi umum:**

***Programming Tools:***

* PHP Framework: Codeigniter
* Bootstrap
* jQuery

|  |  |  |
| --- | --- | --- |
| **Fitur** | **Detail Fitur** | **Penjelasan singkat** |
| **User authentication** | *Login* | * Terdapat layanan *Forgot password* * Pengguna diberikan *token* unik yang digunakan untuk melakukan pengubahan *password*   + *Token* memiliki batas waktu tertentu yang menyatakan validitas nya   + *Token* akan dikirimkan ke email pengguna |
| *Register* | Terdapat 2 tahap registrasi untuk pembuatan akun baru:   * *Tahap pertama*   + Pengguna memasukkan *First Name* dan *Last Name*   + Pengguna memasukkan alamat email yang sekaligus akan menjadi *username*   + User kemudian akan mendapatkan *token* unik untuk melanjutkan tahap registrasi ke-2.     - *Token* memiliki batas waktu tertentu yang menyatakan validitas nya     - *Token* akan dikirimkan ke email pengguna   + Status pengguna baru saat di tahap pertama: *pending* * *Second stage*   + Pengguna memasukkan *Password* dan *Konfirmasi Password*     - Hanya *token* yang valid (yang dikirimkan ke email) yang dapat mengakses tahap ke-2 ini   + Status pengguna baru saat di tahap ke-2: *approved*   + Status pengguna setelah berhasil membuat akun baru adalah *subscriber* |
| **Superadmin**  **&**  **Admin** | *Superadmin* | Adalah orang yang paling pertama berada dalam *database* SAIMemo.  Kemampuan:   * Memilih admin * Menambahkan dan menghapus *subscriber* * Mengubah profil pengguna (*username* dan *password*), yaitu *superadmin, admin*, dan *subscriber* * Melihat aktivitas *admin* dan *subscriber* * Menentukan lokasi penyimpanan file * Melakukan promosi (*promote*) terhadap pengguna untuk menjadi *admin* * Melakukan depromosi (*demote)* terhadap pengguna dari menjadi seorang *admin* |
|  | *Admin* | * Ditentukan oleh *superadmin* * Kemampuan:   + Menambah dan menghapus *subscriber*   + Mengubah profil (*username* dan *password*) diri sendiri, *admin* lainnya, dan *subscriber*   + Melihat aktivitas pengguna   + Menentukan lokasi penyimpanan file   + Melakukan promosi (*promote*) terhadap pengguna untuk menjadi *admin*   + Melakukan depromosi (*demote)* terhadap pengguna dari menjadi seorang *admin* |
| **Subscriber** | *Status: pending* | Tidak dapat mengakses SAIMemo jika mereka tidak menggunakan *token* yang valid untuk melakukan pembuatan *password* |
| *Status: approved* | * *Subscriber* dapat mengakses SAIMemo * Jika tidak ada aksi promosi yang dilakukan oleh *superadmin* atau *admin,* maka status nya tetap menjadi *subscriber* * Jika ada aksi promosi yang dilakukan oleh *superadmin* atau *admin,* maka status nya menjadi *admin* * Kemampuan:   + Mengubah profil diri (*username* dan *password*)   + Membuat folder baru => dapat juga membuat folder di dalam folder, dst   + *Upload* file dengan jenis ekstensi apapun   + *Download* file   + Menghapus file   + Melakukan *sorting* (pengurutan) terhadap daftar file berdasarkan kategori nama file, waktu modifikasi, dan ukuran file |
| **Session** | - | SAIMemo menggunakan *session* untuk menjaga kesinambungan proses antara bagian halaman utama, autentikasi, dan halaman pengguna |

|  |  |  |
| --- | --- | --- |
| **Log activity** | - | * Hanya dapat diakses oleh *superadmin* dan *admin* * Menampilkan *history* dari aktivitas yang dilakukan terhadap SAIMemo beserta waktu kejadiannya, misalnya:   + Aksi login dan registrasi yang dilakukan   + Modifikasi profil   + Pembentukan folder baru   + Aksi *upload, download,* dan *delete* terhadap file * Dapat dilakukan pengurutan |

**Tampilan project:**

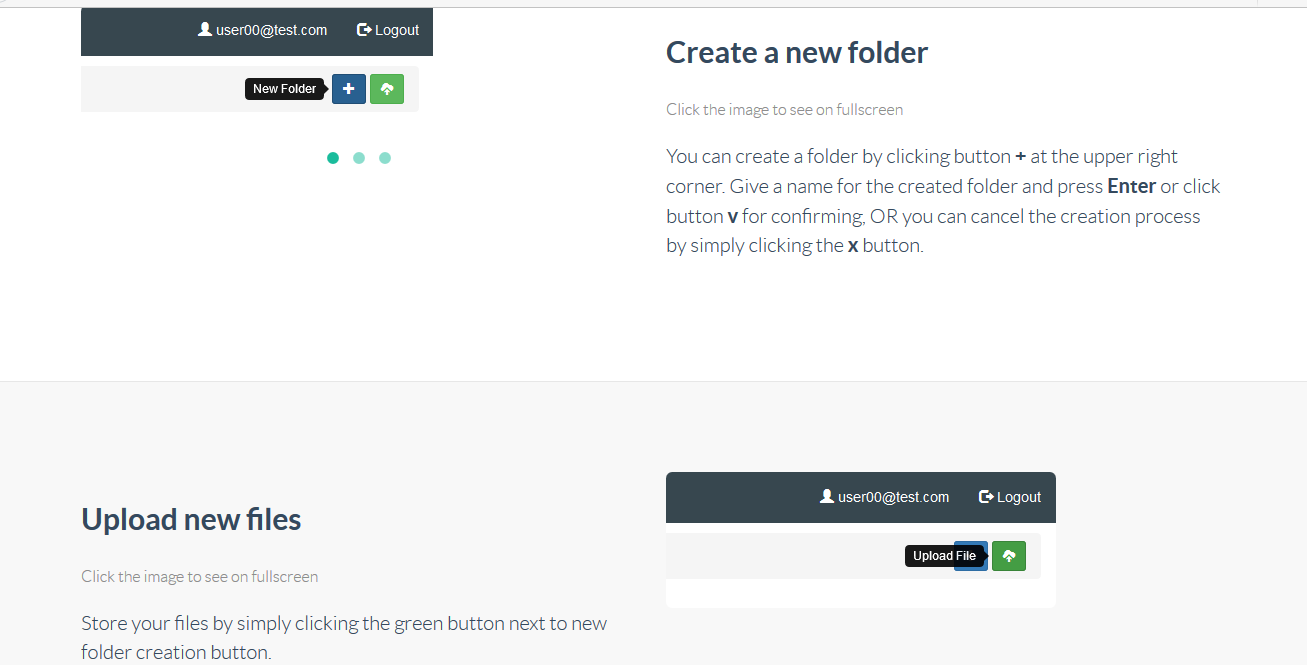
01.Tampilan awal SAIMemo. Pengguna dapat memilih tombol ‘Explore’ untuk *login* atau *registrasi* akun baru. Pengguna juga dapat memilih tombol panah ke bawah untuk melihat penjelasan mengenai apa itu SAIMemo dan bagaimana cara menggunakannya.

Gambar 1

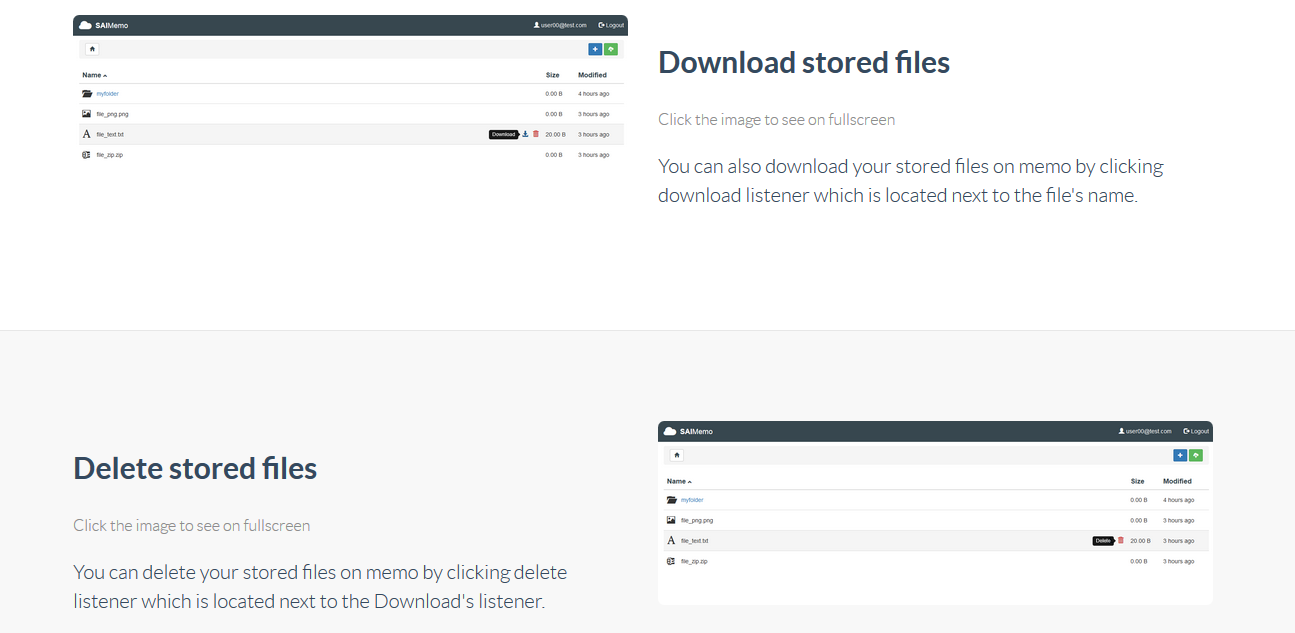
02.Tampilan yang menjelaskan apa itu SAIMemo (*What is it?*)

Gambar 2

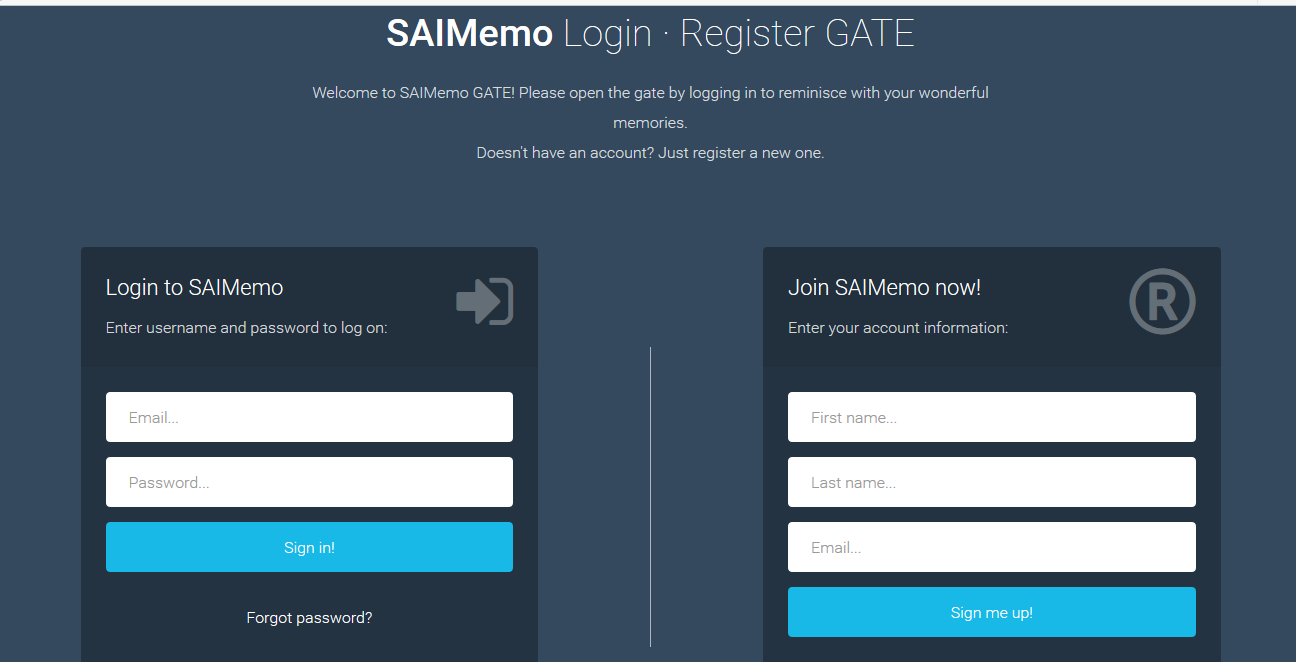
03.Tampilan yang menjelaskan bagaimana cara menggunakan SAIMemo (*How to use?*)



Gambar 3

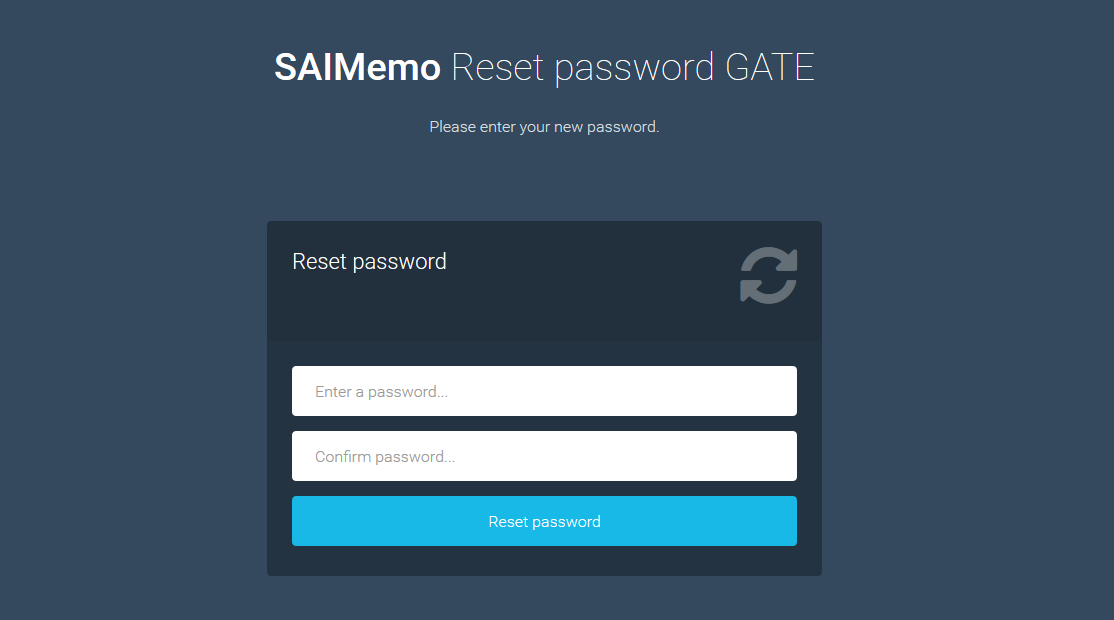


Gambar 4

04.Tampilan halaman *User authentication* => *Login* dan *Register*

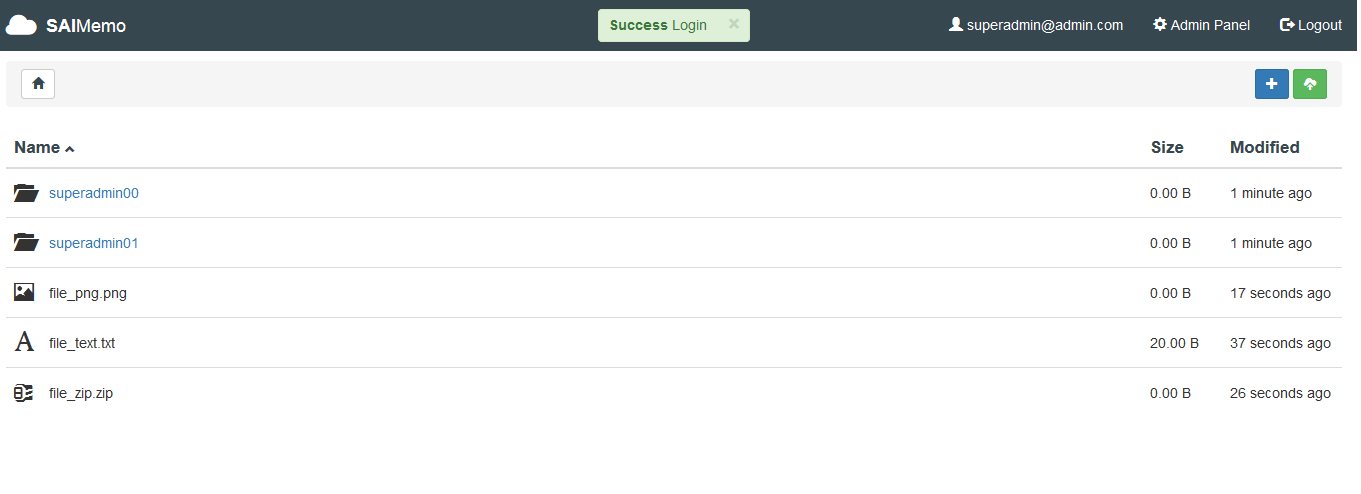
Gambar 5

05.Tampilan halaman *Reset password*



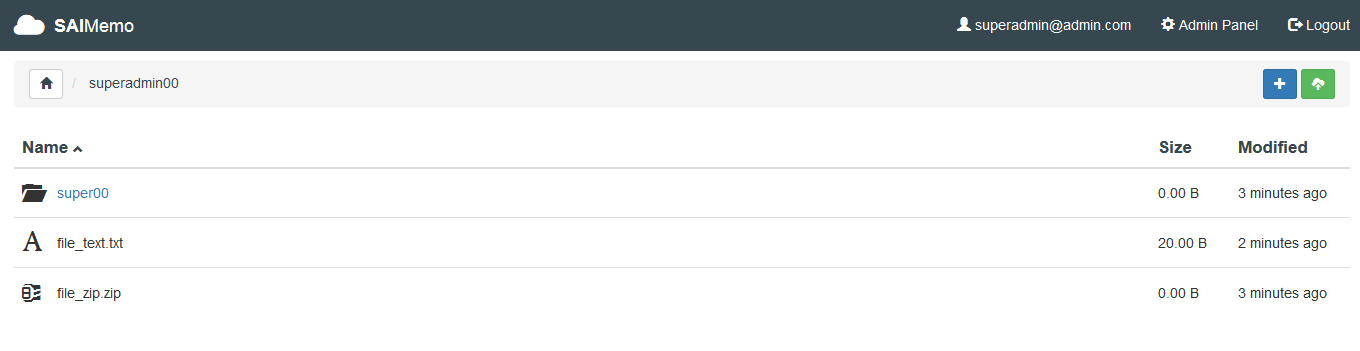
Gambar 6

06.Tampilan halaman *superadmin* dan *admin* (tampilan sama untuk *subscriber,* kecuali fitur ‘Admin Panel’)



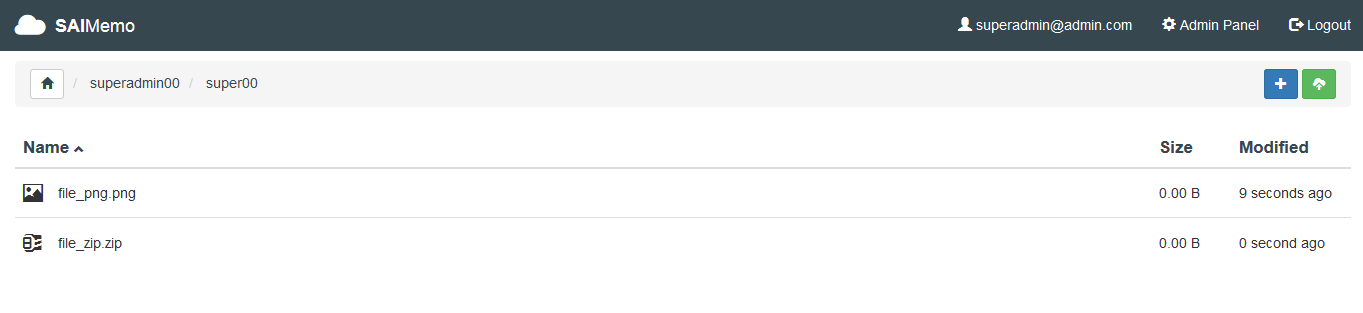
Gambar 7

Tampilan halaman di dalam folder superadmin00/



Gambar 8

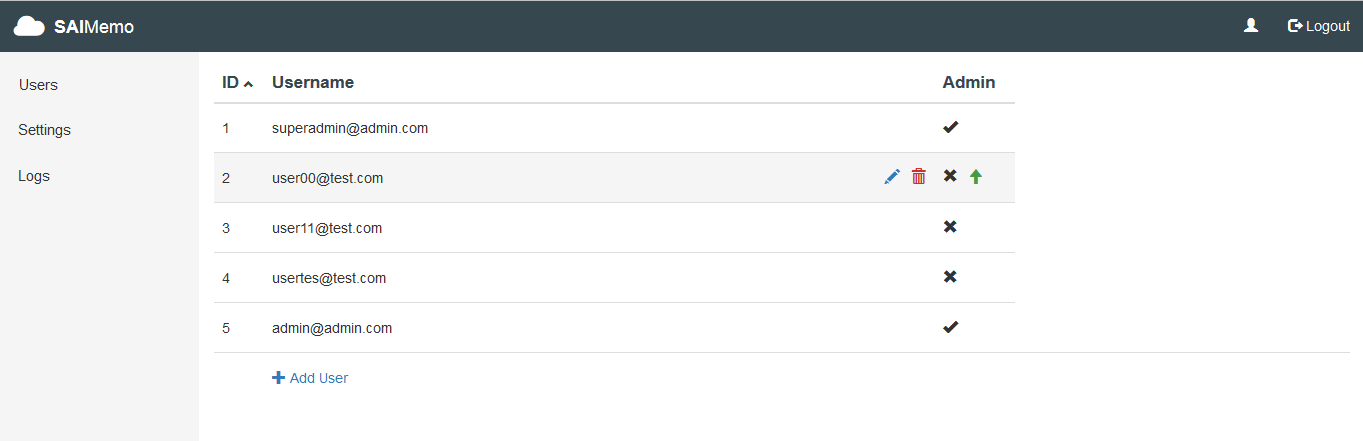
Tampilan halaman di dalam folder superadmin00/super00/



Gambar 9

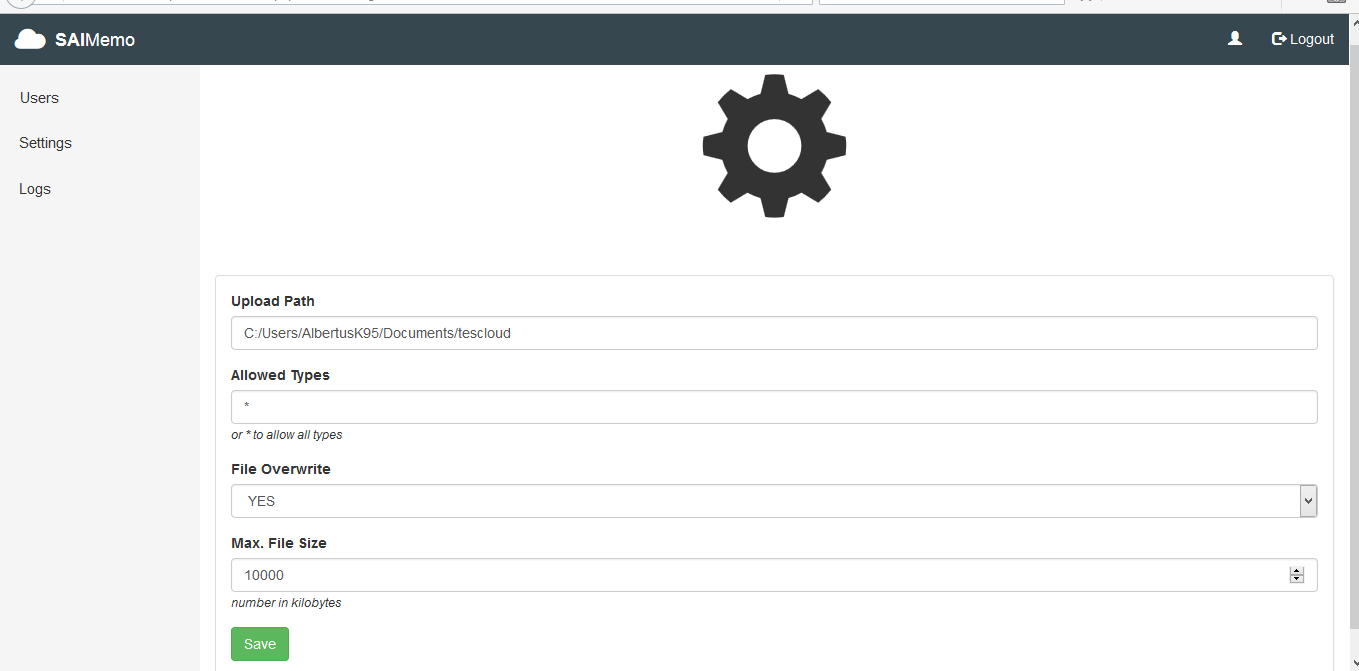
07.Tampilan halaman *Admin Panel*

07a. Isi dari menu ‘Users’ yang menampilkan daftar *username* semua pengguna yang terdaftar dalam *database* SAIMemo. Aksi *promote* dilakukan dengan memilih opsi tanda panah ke atas berwarna hijau. Aksi mengubah profil pengguna dilakukan dengan opsi gambar pensi berwarna biru. Aksi menghapus pengguna dilakukan dengan opsi gambar tempat sampah berwarna merah. Aksi menambah user dilakukan dengan opsi ‘+ Add User’ di bagian bawah.



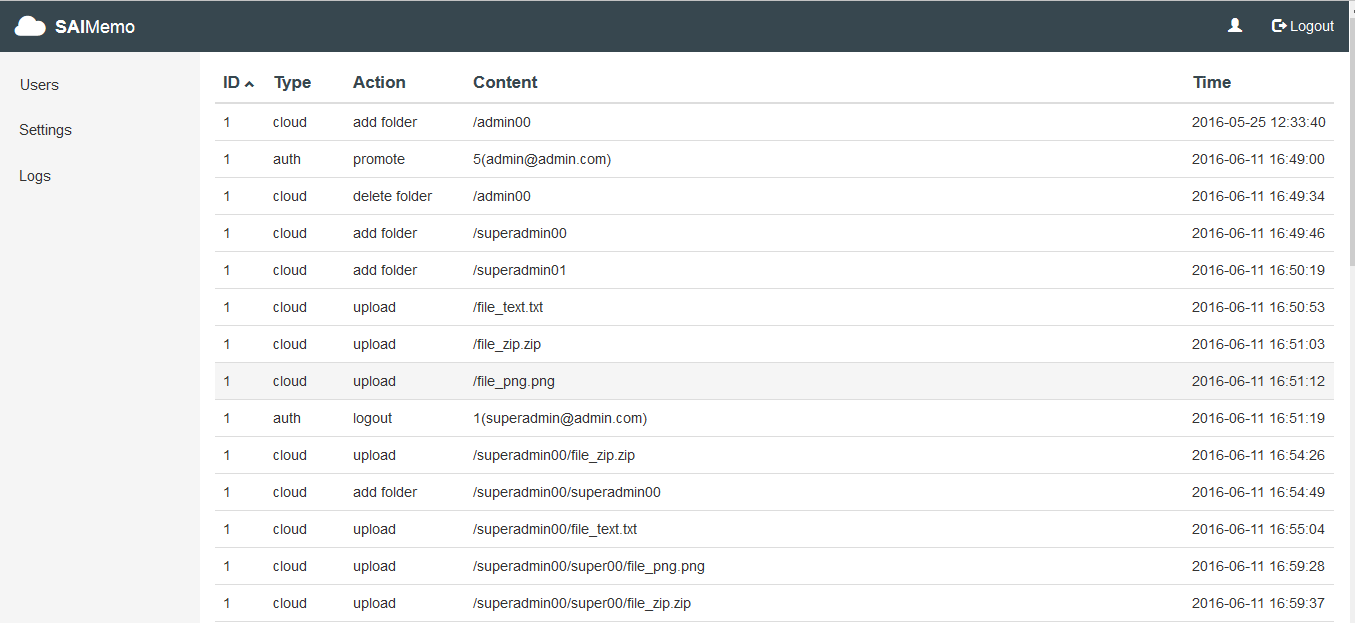
Gambar 10

07b. Isi dari menu ‘Settings’ yang menampilkan konfigurasi file, termasuk lokasi penyimpanan, jenis ekstensi, kondisi *overwrite*, dan ukuran maksimum file yang dapat di-*upload.*



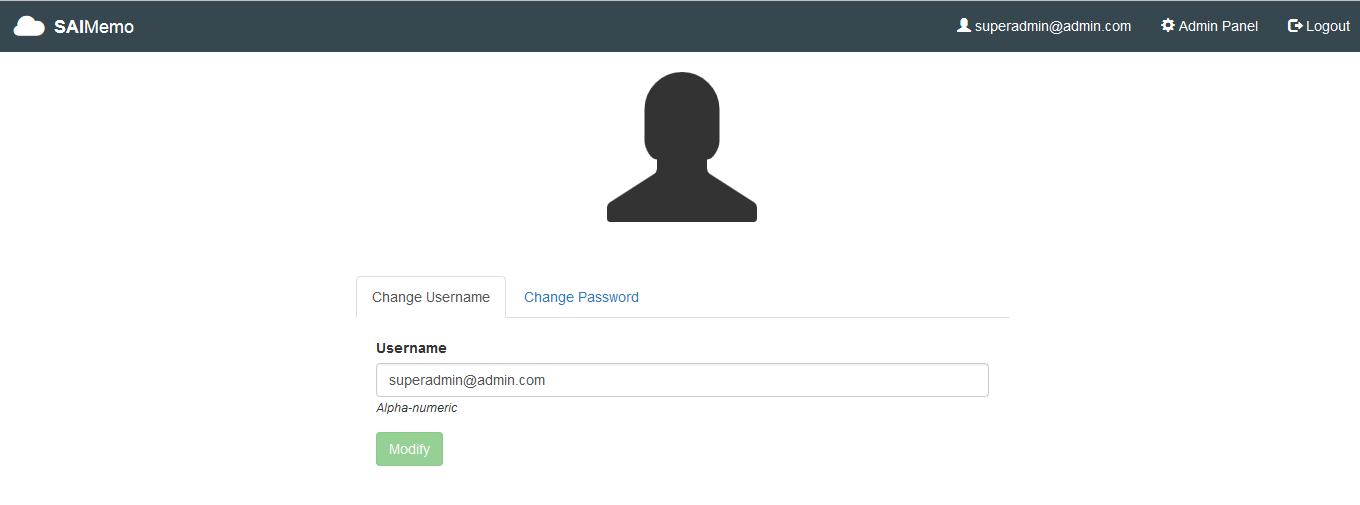
Gambar 11

07c. Isi dari menu ‘Logs’ yang menampilkan *history* dari aktivitas yang dilakukan pengguna terhadap SAIMemo. Yang ditampilkan adalah ID pengguna, tipe *session* dimana aksi itu dilakukan (cloud berarti berhubungan dengan lingkup halaman pengguna dan auth berarti berhubungan dengan lingkup *User Authentication*), jenis aksi, elemen apa yang dimodifikasi (nama folder yang dibuat, nama file yang di *upload, download,* atau *delete*, serta *username* milik user yang melakukan *log in* atau *log out* ataupun dipromosikan sebagai admin, DLL…), dan waktu dilakukannya aksi.



Gambar 12

08.Tampilan halaman untuk mengubah profil *superadmin* dan *admin*, yaitu *username* dan *password* (tampilan sama untuk *subscriber,* kecuali fitur ‘Admin Panel’)



Gambar 13

**“SPAMDetection100” Aplikasi Pendeteksi SMS SPAM**

**Tahun pengerjaan :** 2016

**Jenis pengerjaan :** Individu

**Tujuan project :** Membuat sebuah aplikasi pendeteksi SMS SPAM sederhana. Aplikasi ini nantinya akan menerima input berupa teks SMS, kemudian akan menentukan apakah teks SMS tersebut masuk ke dalam kategori SPAM atau bukan.

**Keterangan :** Aplikasi ini bernama SPAMDetection100. Berikut beberapa informasi detail nya.

***Programming Tools*** : Java + GUI

***Lingkungan pengembangan***: IDE NetBeans

***Lingkup Algoritma*** : Machine Learning

Karena aplikasi ini menggunakan Machine Learning, maka diperlukan suatu *data set* dan *data train*. Bahasa yang digunakan untuk *data train* adalah bahasa Indonesia.

**Informasi lain-lain:**

Algoritma/ Classifier: NaiveBayes

Metode evaluasi: 4-fold cross-validation

Nilai class (prediksi): SPAM atau BUKAN\_SPAM

Nama relasi dataset: my\_sms

Atribut: *spamclass* yang berisi SPAM dan BUKAN\_SPAM; *text* yang berisi data tipe String

Format data train (@data): [SPAM/BUKAN\_SPAM],[TEXT] -> tanpa []

Jumlah *data train*: 256 (spam 97; bukan-spam 159)

**Cara kerja aplikasi secara umum:**

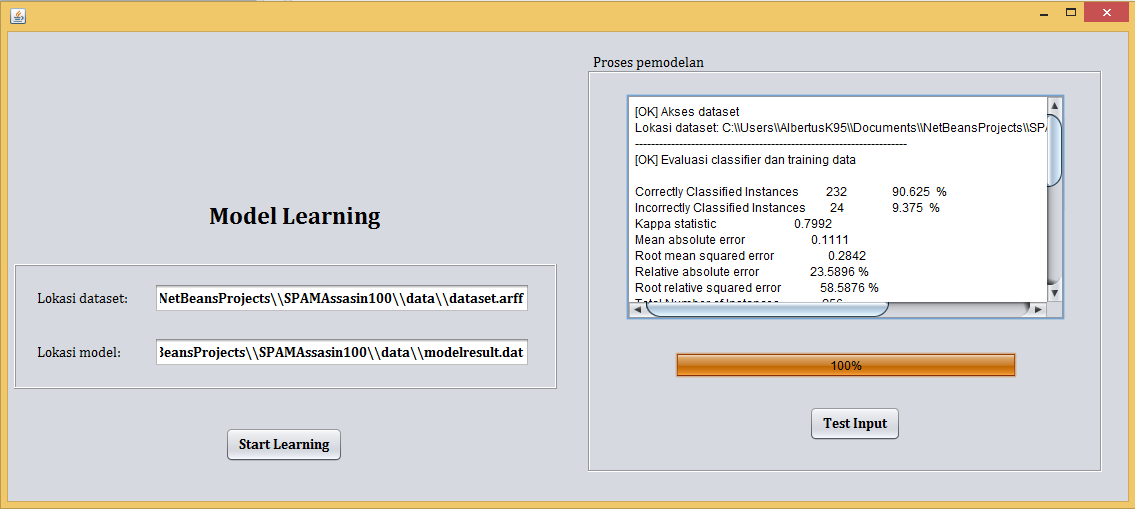
-Aplikasi terdiri dari 2 bagian, yaitu *Model Learning* dan *SPAM Detection*

-*Model Learning* bertugas untuk membentuk model dari hasil pembelajaran *data train* untuk digunakan kemudian oleh *SPAM Detection*

-*SPAM Detection* bertugas untuk menentukan kategori suatu teks input (spam atau bukan) berdasarkan model yang telah dibentuk oleh *Model Learning*

**Tampilan Project:**

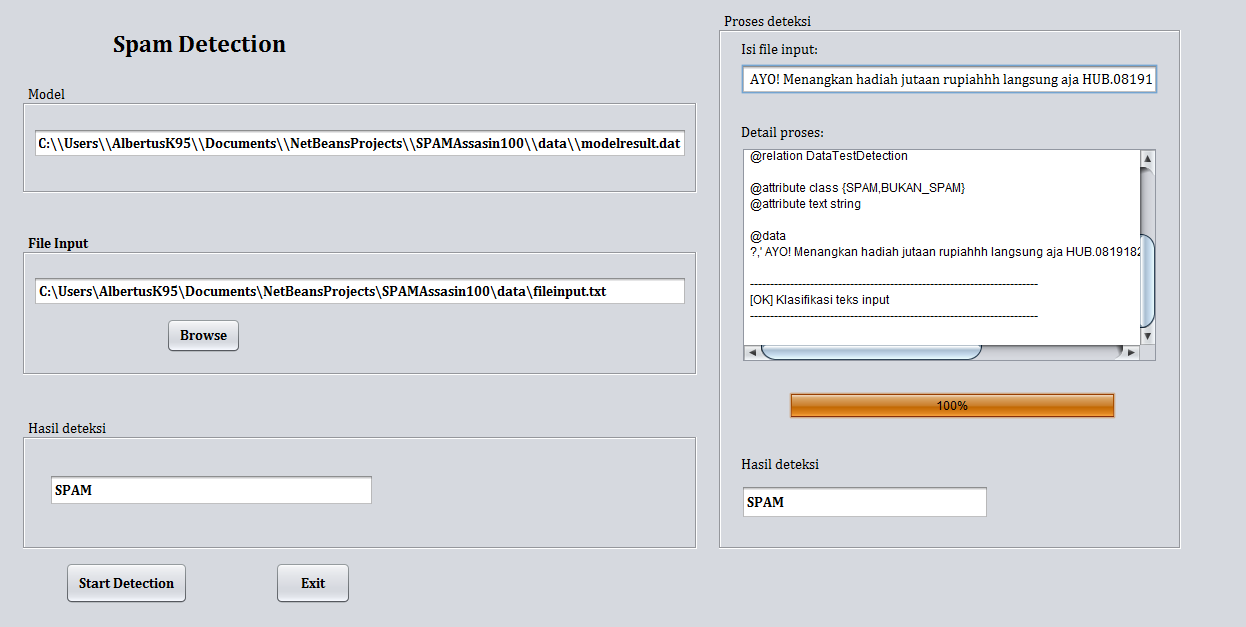
01.Proses pembelajaran dan pembentukan model oleh classifier.



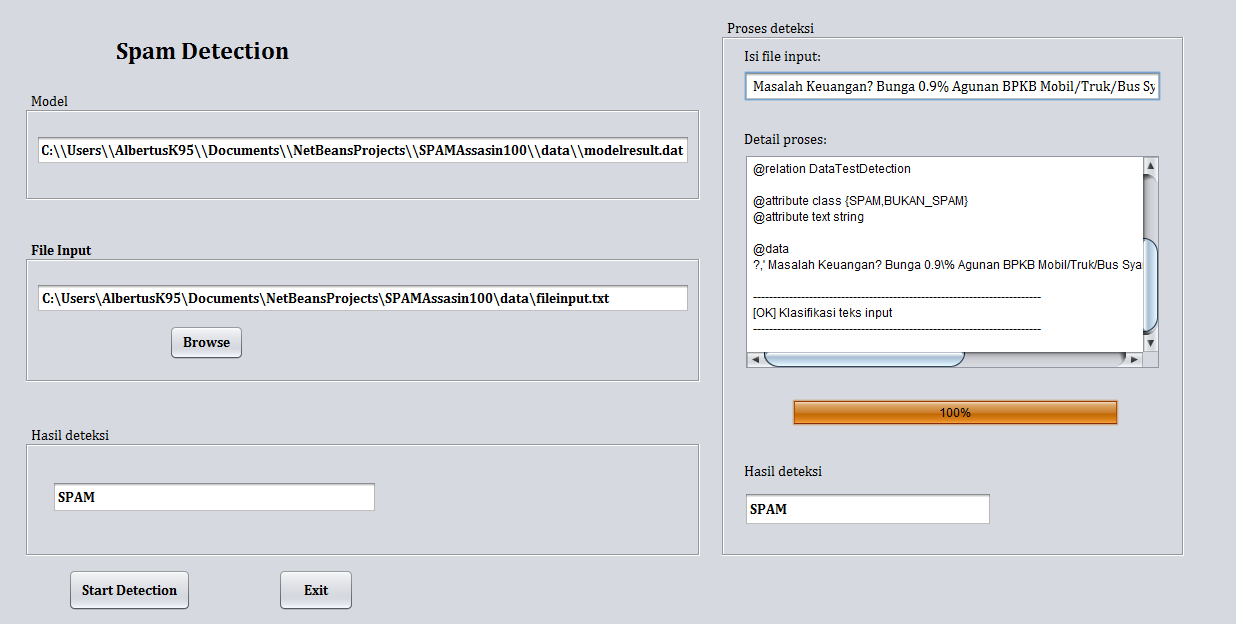
*User* akan diarahkan menuju halaman *SPAMDetection* setelah menekan tombol ‘Test Input’.

Berikut beberapa contoh *test case* untuk menguji kebenaran program.

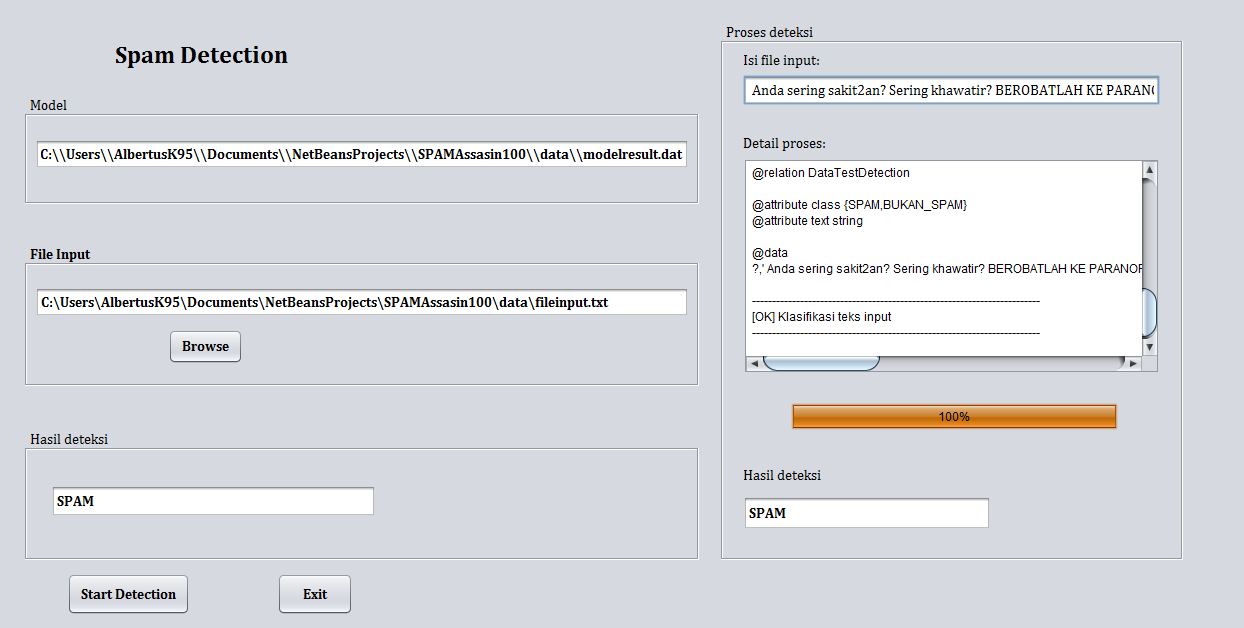
**Test 0**

**Teks input:** AYO! Menangkan hadiah jutaan rupiahhh langsung aja HUB.0819182378483 sk berlaku! DIJAMIN PUASSSSSS!!!

**Test 1**

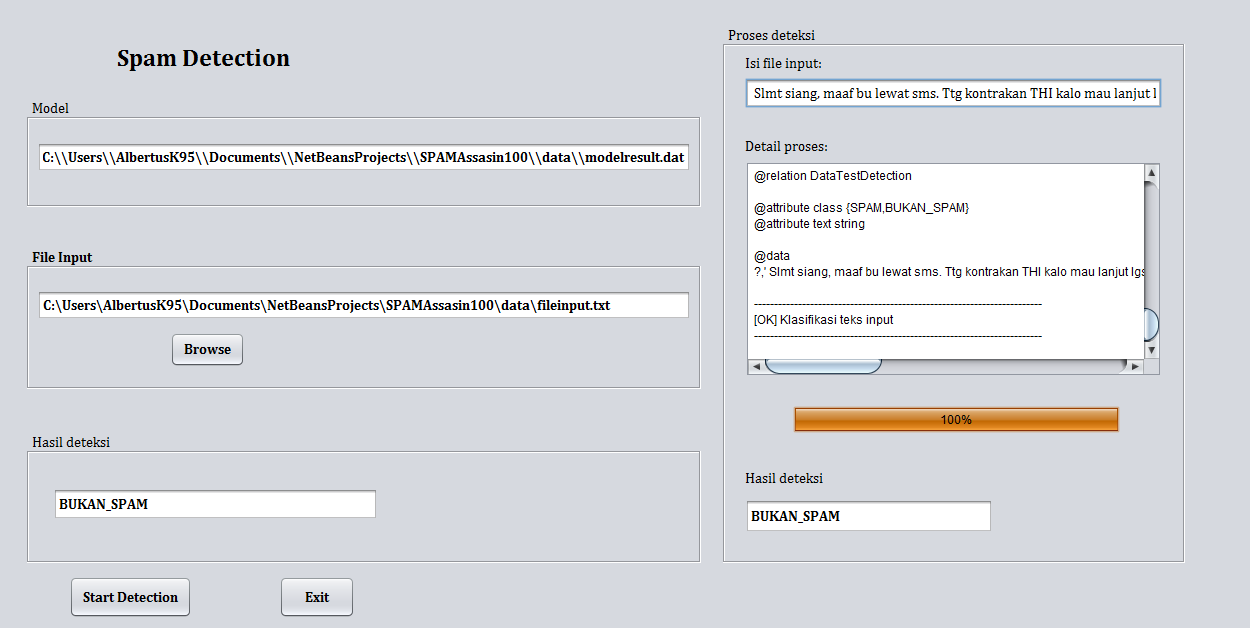
**Teks input:** Masalah Keuangan? Bunga 0.9% Agunan BPKB Mobil/Truk/Bus Syarat Ringan KTP+NPWP 1 HARI CAIR RIO 081282841114

**Test 2**

**Teks input:** Anda sering sakit2an? Sering khawatir? BEROBATLAH KE PARANORMAL KI GENDENG di daerah Gn. Sentosa. Hubungi 081283739283 utk ALAMAT JELASNYA. DIJAMIN SEMUA PENYAKIT ANDA TEMBUSSSSS 1000%

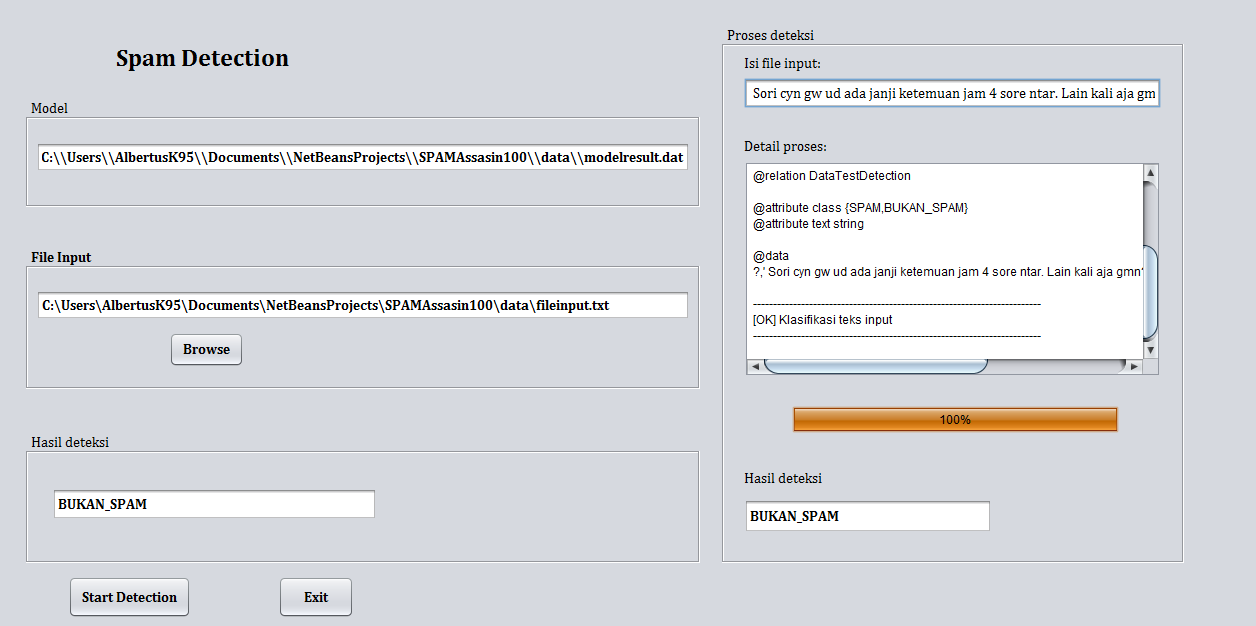
**Test 3**

**Teks input:** Slmt siang, maaf bu lewat sms. Ttg kontrakan THI kalo mau lanjut lgsg hubungi adik saya aja ya di 081739282226 krn skrg dia yg urus. Thx.



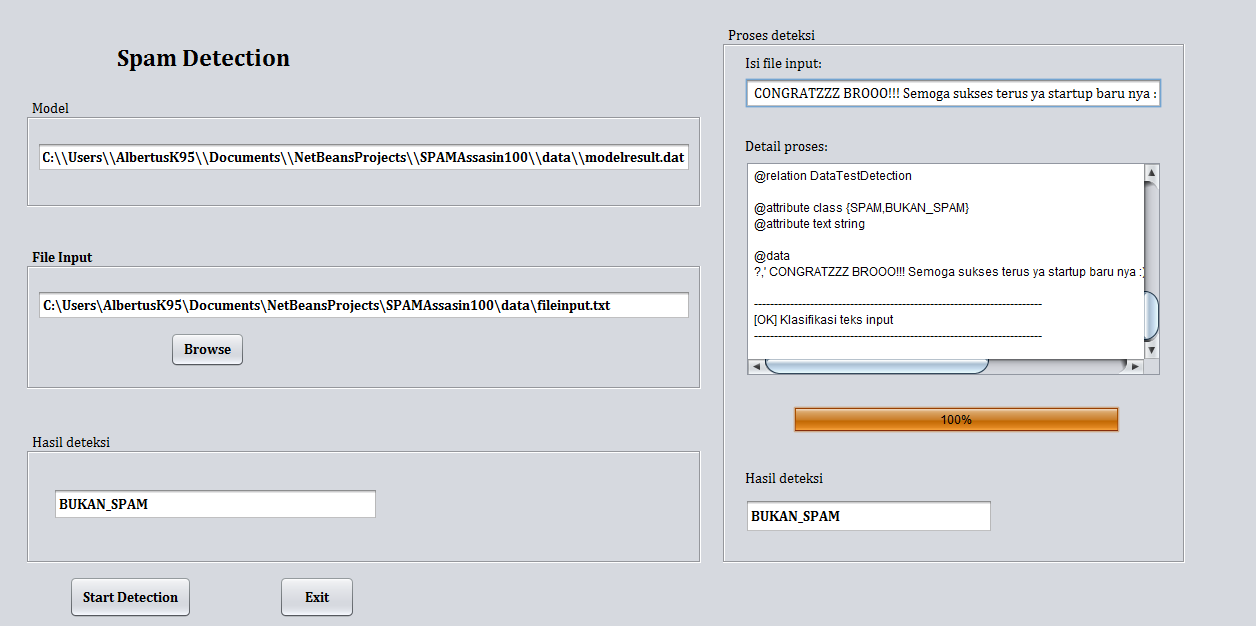
**Test 4**

**Teks input:** Sori cyn gw ud ada janji ketemuan jam 4 sore ntar. Lain kali aja gmn? Hehehe

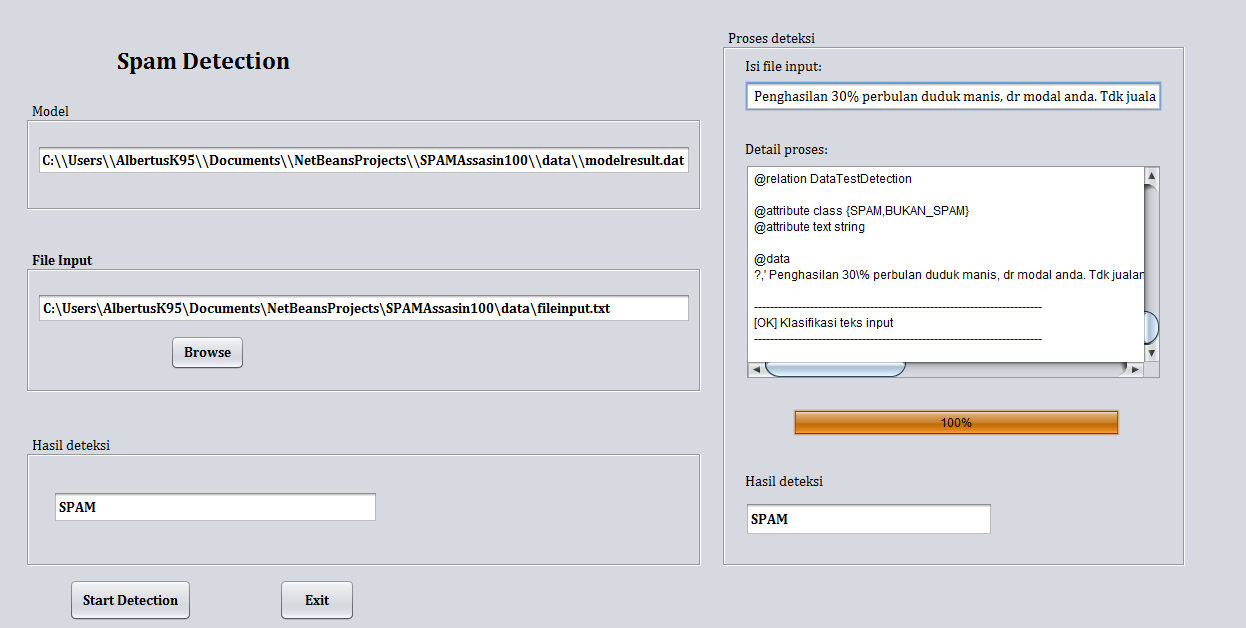
****

**Test 5**

**Teks input:** CONGRATZZZ BROOO!!! Semoga sukses terus ya startup baru nya :))) MANTAP!!

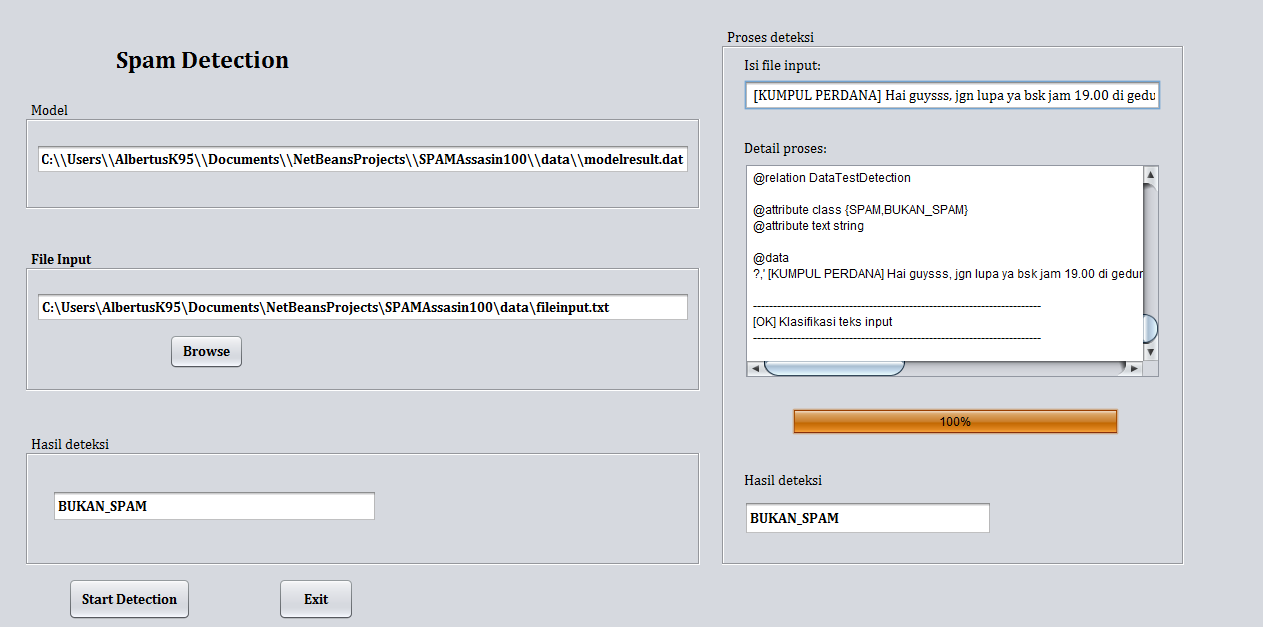
****

**Test 6**

**Teks input:** Penghasilan 30% perbulan duduk manis, dr modal anda. Tdk jualan tdk rekrut2an! 350 rb member seindonesia! REVOLUSI EKONOMI ANDA MENJADI LBH BAIK Add BB 5206B100

**Test 7**

**Teks input:** [KUMPUL PERDANA] Hai guysss, jgn lupa ya bsk jam 19.00 di gedung CAS lt.1 kita bakalan kumpul bareng pertama kali :) Jgn lupa bw SEMANGAT nya. Konfirm di sini yaaa. Thx!

****

**Aplikasi Pencarian Jalur Terpendek dengan Algoritma Branch & Bound**

**Tahun pengerjaan :** 2016

**Jenis pengerjaan :** Individu

**Tujuan project :** Membuat sebuah aplikasi yang mampu membaca input berupa matriks yang menunjukkan hubungan antar titik dalam suatu graf, kemudian memberikan solusi berupa jalur terpendek yang menghubungkan semua titik dalam graf tersebut dengan algoritma Branch and Bound.

**Keterangan :**

***Programming Tools*** : Java + GUI

**Tampilan project :**

01.Aplikasi akan menerima sebuah file input yang berisi sebuah matriks yang merepresentasikan hubungan antar titik dalam suatu graf tertentu. Hubungan antar titik itu berupa jarak / bobot antara 2 buah titik.

02.Kemudian aplikasi akan mencari solusi berupa jarak terpendek yang diperlukan untuk mengunjungi semua titik dalam graf beserta jalur solusi yang dapat ditempuh. Selain itu, aplikasi akan memberikan gambar graf tak berarah yang dibentuk dari matriks yang diberikan, dimana jalur terpendek yang merupakan solusi akan diberikan *highlight* berwarna merah.

Berikut beberapa contoh *test case* untuk menguji kebenaran program.

**Test 0**

File input:

-1 500 200 185 205

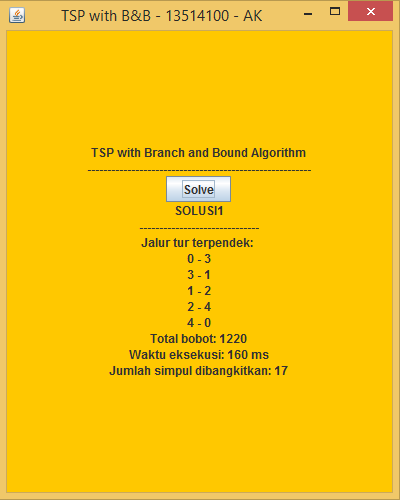
500 -1 305 360 340

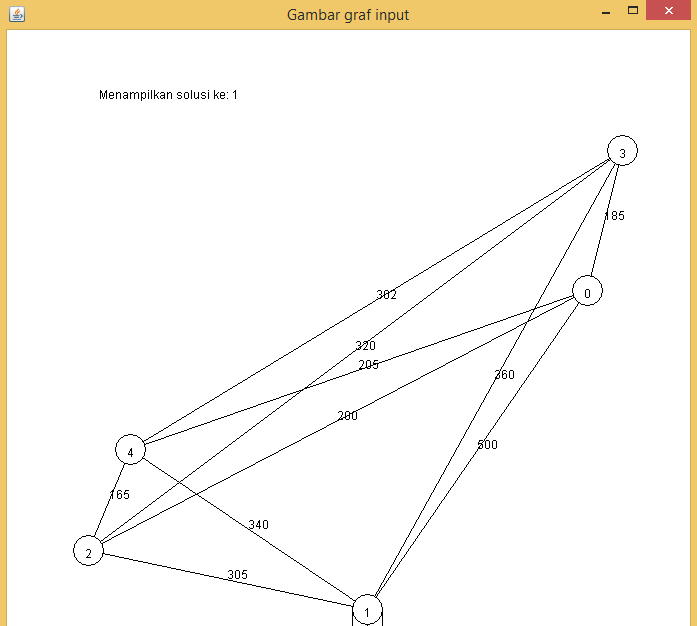
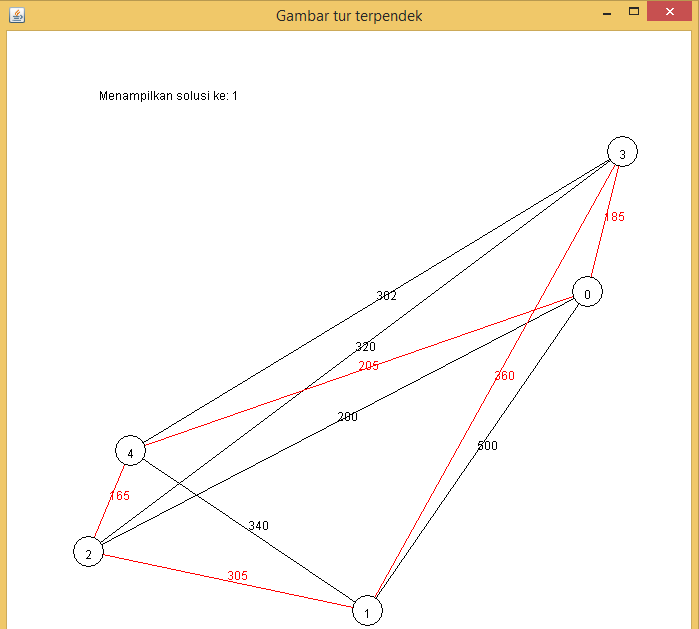
200 305 -1 320 165

185 360 320 -1 302

205 340 165 302 -1

**Aplikasi**

****



**Test 1**

File input:

-1 16 27 89 34 65 12 10

16 -1 56 78 32 16 64 32

27 56 -1 82 57 35 80 25

89 78 82 -1 34 51 19 47

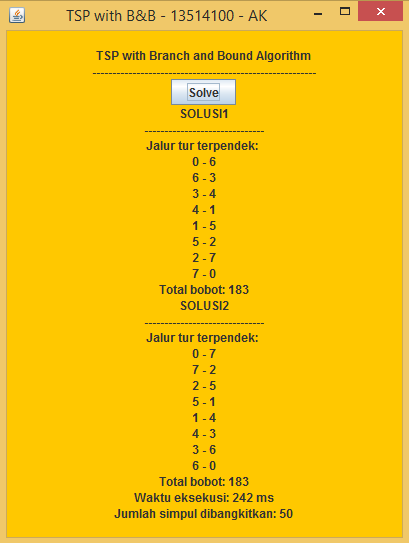
34 32 57 34 -1 42 59 41

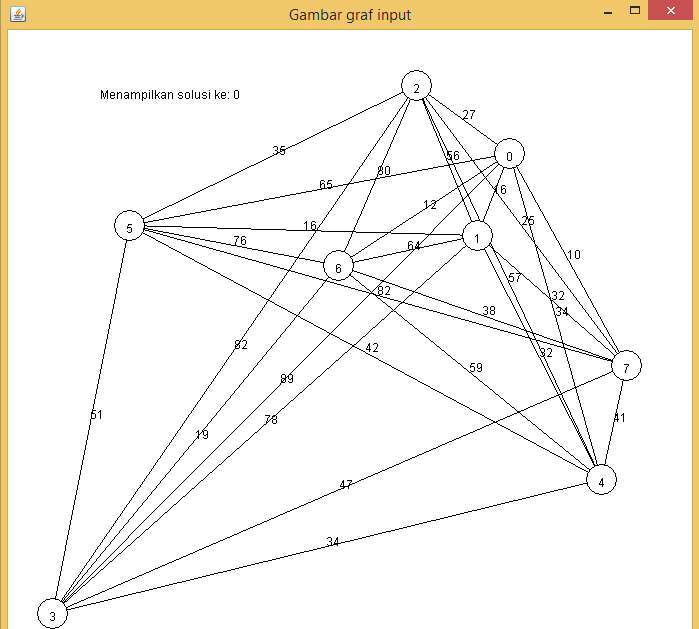
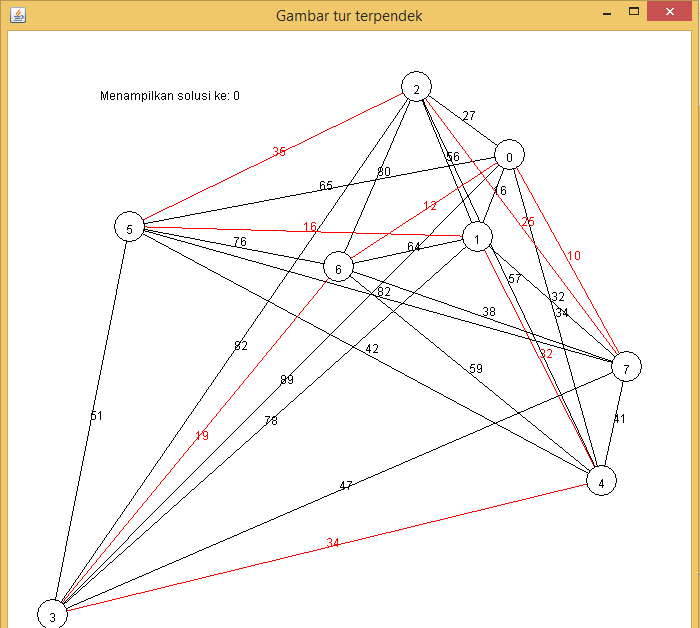
65 16 35 51 42 -1 76 82

12 64 80 19 59 76 -1 38

10 32 25 47 41 82 38 -1

**Aplikasi**

****



**Test 2**

File input:

-1 1 2 3 4 5 6 7 8 9

1 -1 11 12 13 14 15 16 17 18

2 11 -1 21 22 23 24 25 26 27

3 12 21 -1 31 32 33 34 35 36

4 13 22 31 -1 41 42 43 44 45

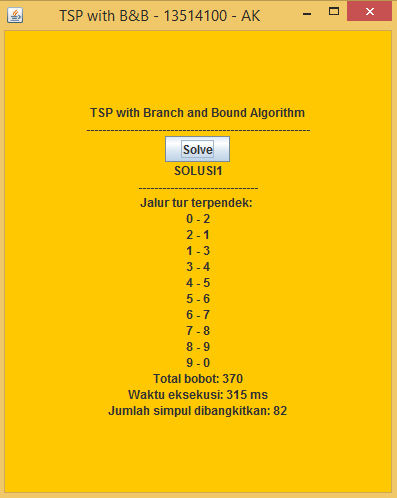
5 14 23 32 41 -1 51 52 53 54

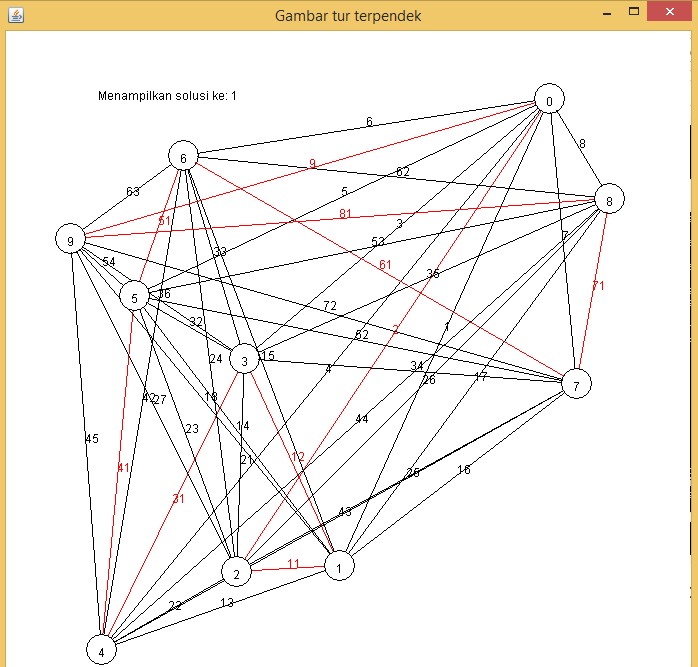
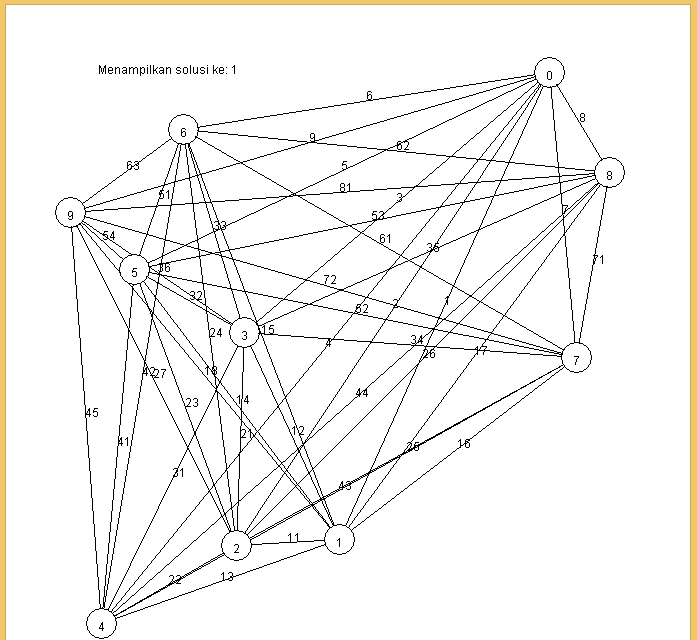
6 15 24 33 42 51 -1 61 62 63

7 16 25 34 43 52 61 -1 71 72

8 17 26 35 44 53 62 71 -1 81

9 18 27 36 45 54 63 72 81 -1

**Aplikasi**



**Aplikasi *Hyper Sudoku Solver***

**Tahun pengerjaan :** 2016

**Jenis pengerjaan :** Individu

**Tujuan project :** Membuat sebuah aplikasi yang dapat mencari solusi dari sebuah *Hyper Sudoku*

**Cara kerja program secara umum:**

- Saat program dijalankan, ia akan membaca isi file eksternal dan mengisi kotak HyperSudoku 9x9 di bawah dengan nilai yang didapat dari file eksternal tersebut

-*User* hanya perlu menekan tombol 'Solve' yang akan memberi perintah pada program untuk

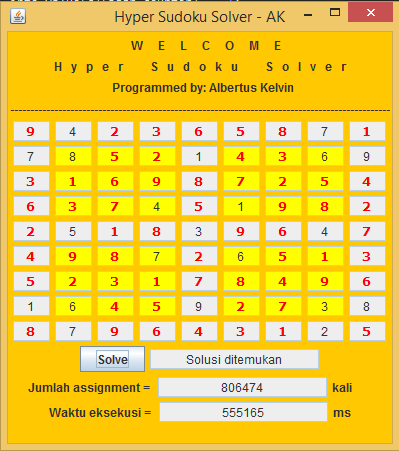
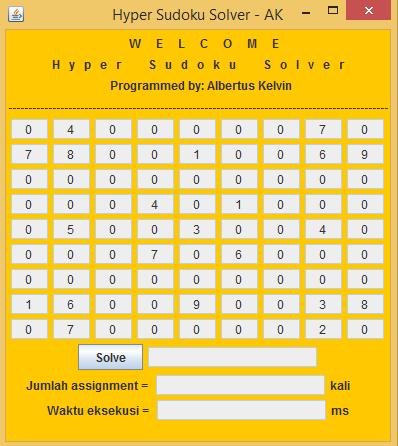
menampilkan solusi HyperSudoku

- Solusi yang didapat ditandai oleh angka berwarna merah

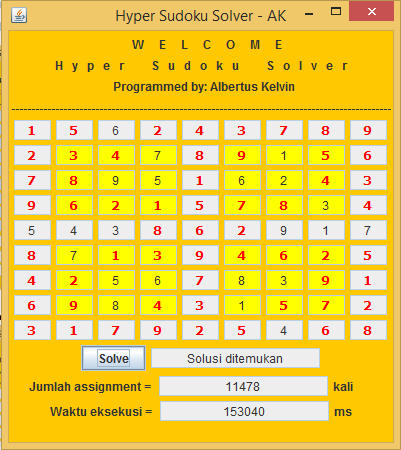
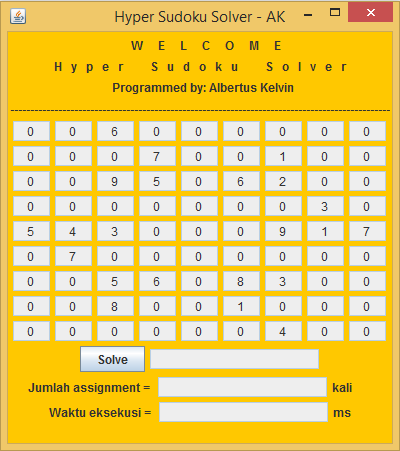
- Kotak 3x3 berjumlah 4 buah juga akan terlihat bedanya setelah program dijalankan

Berikut beberapa contoh *test case* untuk menguji kebenaran program.

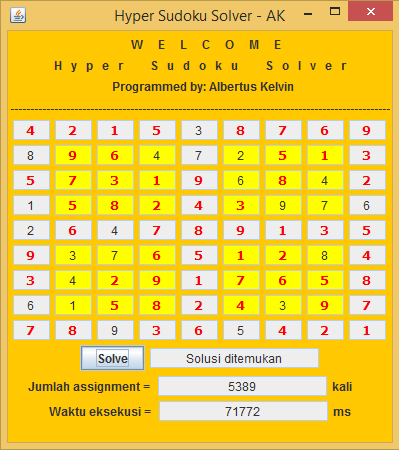
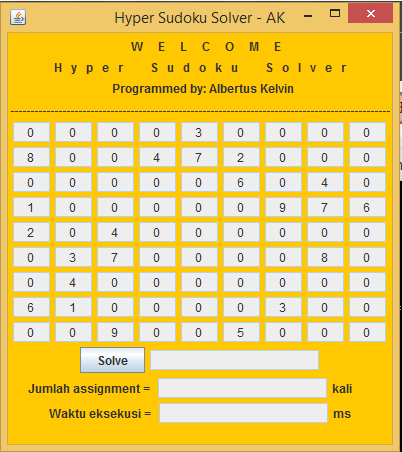
**Test 0**



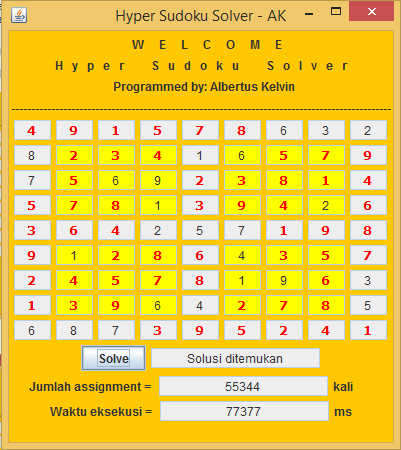
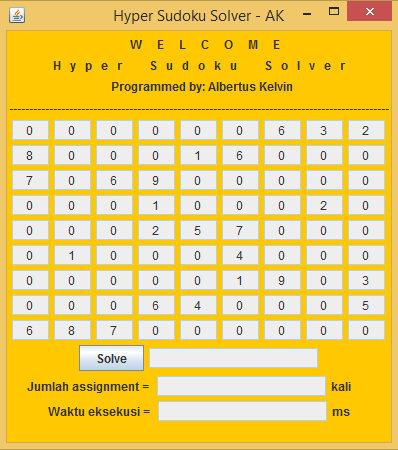
**Test 1**



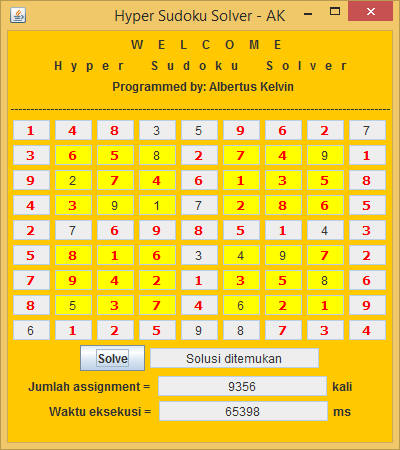
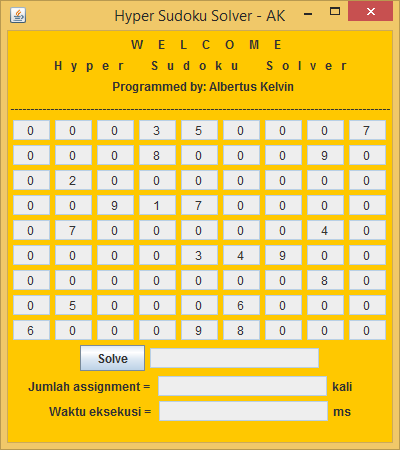
**Test 2**



**Test 3**



**Test 4**



**Aplikasi *LEGO Maze Solver with BFS and DFS***

**Tahun pengerjaan :** 2016

**Jenis pengerjaan :** Kelompok

**Tujuan project :** Membuat sebuah aplikasi menggunakan software simulasi robot LEGO Mindstorms dimana robot akan mencari jalan keluar dari sebuah *maze*. Project ini juga mengimplementasikan algoritma BFS dan DFS dalam pencarian solusi jalan keluar.

**Keterangan :**

***Programming tools* :** Bahasa C + ROBOTC for LEGO Mindstorms

**Project ini mengimplementasikan algoritma BFS dan DFS, dimana terdapat 2 buah aplikasi terpisah, yaitu aplikasi pertama yang menggunakan algoritma BFS dan aplikasi kedua yang menggunakan algoritma DFS**

**DEMO Video :**  Menggunakan algoritma BFS

[**https://www.youtube.com/watch?v=M2SjEteCZEM**](https://www.youtube.com/watch?v=M2SjEteCZEM)

Menggunakan algoritma DFS

[**https://www.youtube.com/watch?v=VWb1WbO0Pfk**](https://www.youtube.com/watch?v=VWb1WbO0Pfk)

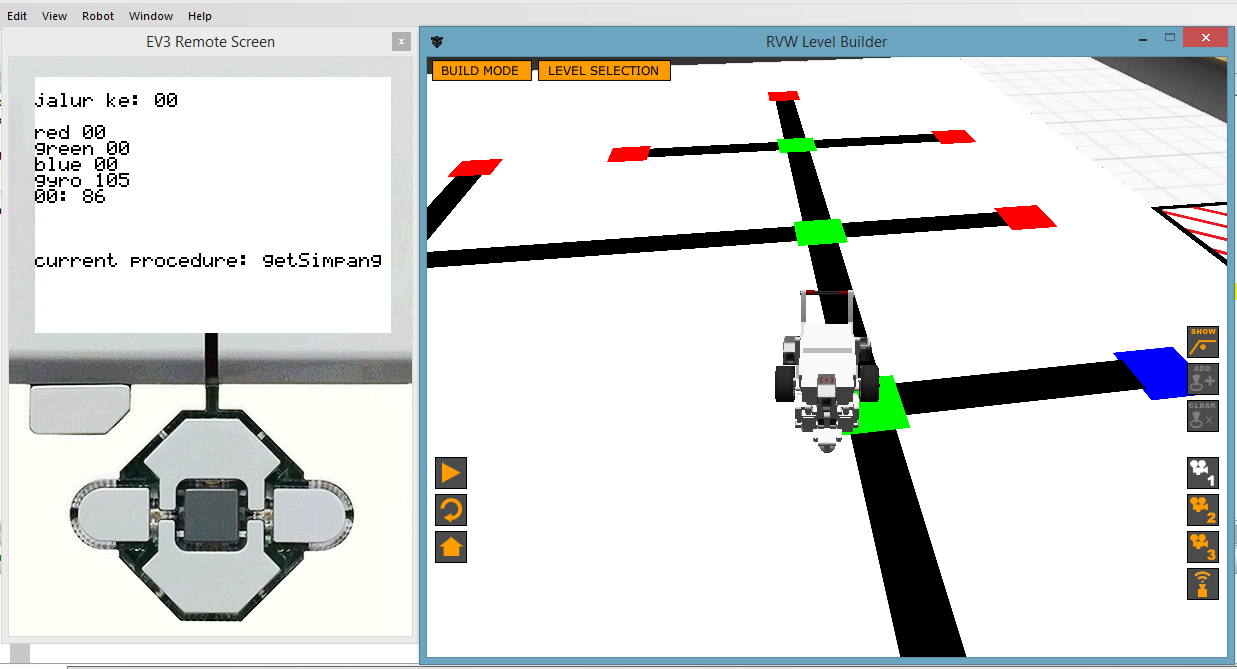
**Cara kerja program secara umum :**

* 1. *Maze* terdiri dari bagian-bagian penting seperti titik warna biru (posisi *start* dan *finish*), titik warna merah (menandakan jalan buntu), titik warna hijau (menandakan adanya percabangan), dan garis hitam (sebagai jalur gerak robot).
  2. Setelah *user* menekan tombol *start*, maka robot akan mulai mencari jalur menuju titik *finish* dengan algoritma BFS atau DFS. Selama proses pencarian jalur tersebut, robot akan mengingat semua urutan jalur yang menghubungkan titik *start* dan *finish*. Urutan jalur yang diingat itu termasuk besar sudut yang dipilih robot saat berputar untuk memilih jalur berikutnya.
  3. Setelah berhasil menemukan titik *finish*, robot akan kembali ke titik *start* dengan urutan jalur yang sudah diingat sebelumnya. Jalur yang sudah diingat tersebut merupakan solusi dari persoalan *maze* yang bersangkutan.
  4. Setelah berhasil sampai ke titik *start*, robot akan kembali lagi ke titik *finish* lagidengan solusi yang didapat. Langkah kembali ke titik *finish* ini adalah langkah terakhir dalam aplikasi *maze solver* ini.

**Tampilan project :**

1. Saat robot berada di simpangan untuk mendeteksi semua jalur percabangan yang ada.

Nilai red, green, blue merupakan besaran intensitas dari RGB yang didapat saat robot melakukan pendeteksian jalur percabangan di sekitar titik warna hijau.

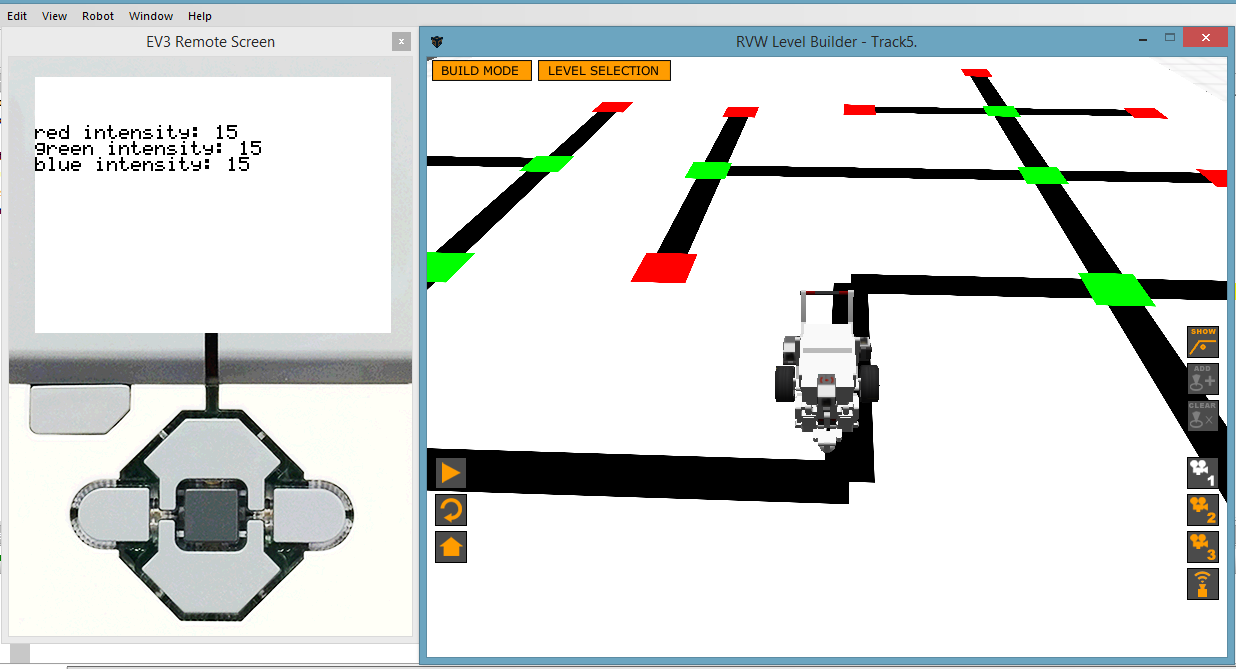


Gambar 1

1. Contoh tampilan salah satu *test case maze* secara keseluruhan.



Gambar 2

1. Saat robot sedang bergerak mengikuti garis warna hitam.

Gambar 3

**Aplikasi *OpenGLPro100***

**Tahun pengerjaan :** 2016

**Jenis pengerjaan :** Individu

**Tujuan project :** Membuat sebuah aplikasi permainan sederhana yang menggunakan OpenGL pada sistem operasi Windows

**Keterangan :**

***Programming tools* :** Bahasa pemrograman C++ dan IDE Visual Studio 2013

***Operating system* :** Windows

**DEMO Video :** [**https://www.youtube.com/watch?v=NPggBSrDQzo**](https://www.youtube.com/watch?v=NPggBSrDQzo)

**Cara kerja dan aturan teknis program secara umum :**

**Deskripsi**

*Pengenalan aplikasi*

Aplikasi bernama OpenGLPro100. Konsep dasar dari aplikasi ini adalah seperti permainan sederhana, dimana *user* akan diberikan waktu selama 12 detik untuk mencapai total poin lebih besar atau sama dengan nilai tertentu. Nilai tertentu itu didapat dari jumlah semua objek yang dikalikan dengan 3, dimana nilai 3 merupakan bobot setiap objek. Objek yang tersedia dalam aplikasi ini dibagi menjadi 2 bagian, pertama objek yang di-*generate* oleh permintaan *user*, kedua objek yang di-*generate* oleh aplikasi. Objek dari *user* berupa lingkaran (*circle*), kotak (*square*), dan segi delapan (*octagon*). Sedangkan objek dari aplikasi berupa garis (*line*) lurus dan putus-putus, persegi panjang (*rectangle*), dan kotak (*square*). Adapun objek dari aplikasi ini memiliki nama sendiri, seperti persegi panjang dinamakan kotak pemantul, garis lurus dan putus-putus dinamakan garis pembatas, dan kotak dinamakan kotak teks permainan.

*Cara kerja aplikasi secara umum*

*User* harus dapat melebihi nilai tertentu dalam waktu 12 detik untuk bisa menang. Poin awal adalah 0. Untuk menambah poin, *user* harus memantulkan objek dengan kotak pemantul dimana kecepatan gerak objek akan bertambah seiring waktu permainan berjalan. Jika pantulan berhasil dilakukan, maka poin *user* akan bertambah 3. Jika pantulan tidak berhasil terjadi (objek menyentuh bagian bawah window), maka poin *user* akan berkurang 1. Permainan berakhir dengan status YOU WIN atau YOU LOSE dengan menampilkan total poin yang berhasil didapat.

**Objek**

* *User generated*: lingkaran (*circle*), kotak (*square*), dan segi delapan (*octagon*)
* *Apps generated*: garis (*line*) lurus dan putus-putus, persegi panjang (*rectangle*), dan

kotak (*square*)

**Sistem pewarnaan objek**

Pada awal permainan, *user* akan diberi pilihan untuk memilih cara pewarnaan objek, yaitu dengan cara random (diatur oleh aplikasi) atau dengan pengaturan sendiri. Memilih cara random berarti setiap *user generated object* akan diberi warna dengan kode RGB yang random. Sebaliknya, memilih pengaturan sendiri berarti *user* akan dipersilahkan menentukan kode warna RGB sendiri. Kode RGB yang valid untuk aplikasi ini adalah dari 0 sampai dengan 1, dimana nilai R, G, dan B tidak boleh semuanya sama dengan 0 atau 1.

**Animasi**

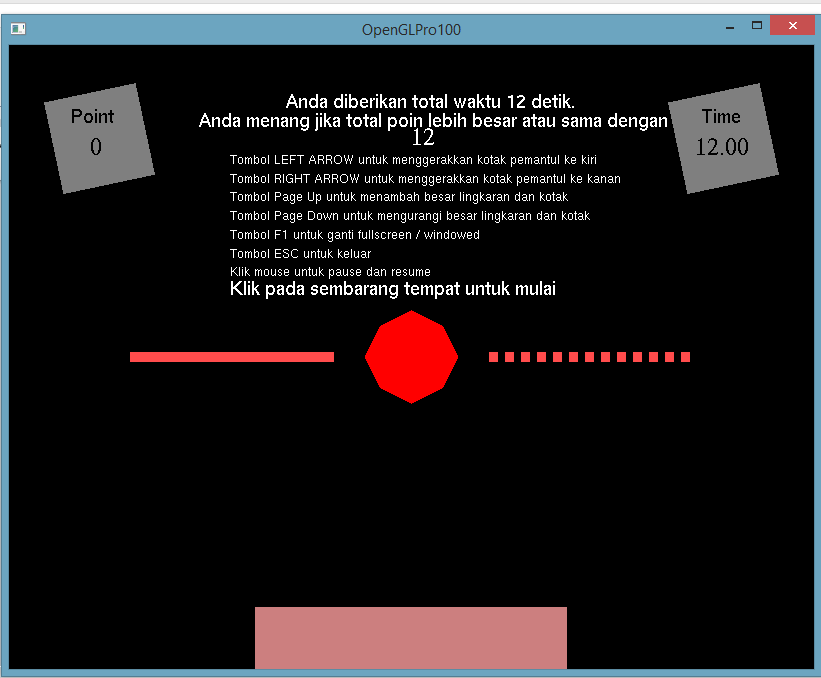
* Memantul
* Semua *user generated object* dapat memantul jika mengenai sisi kiri, kanan, dan bawah window aplikasi.
* Semua *user generated object* dapat memantul jika mengenai kotak pemantul dan garis pembatas (lurus dan putus-putus), baik pada sisi atas maupun bawah garis.
* Semua *user generated object* dapat memantul jika bertumbukkan dengan *user generated object* yang lain (*square* dan *square*, *square* dan *circle*, *octagon* dan *circle*, DLL)
* Garis pembatas dapat memantul ke arah berlainan menurut sumbu x jika bertumbukan dengan sisi kiri dan kanan window aplikasi
* Pantulan yang terjadi oleh *user generated object* akan membuat gerak objek tersebut menjadi lebih cepat
* Translasi
* Kotak pemantul dapat bergerak ke kiri dan kanan menurut sumbu x, dimana sumbu y nya tetap
* Garis pembatas dapat bergerak ke kiri dan kanan, dimana arahnya berubah jika bertumbukan dengan window aplikasi
* Rotasi
* Kotak teks permainan dapat berotasi di tempat
* Gerak bebas
* Selain dapat memantul, semua *user generated object* mampu bergerak dalam arah sumbu x dan y bersamaan
* Gerakan *user generated object* akan bertambah cepat seiring waktu permainan
* *Resize*
* *User generated object* seperti *circle* dan *square* dapat diubah ukurannya menjadi lebih besar atau kecil

**Interaksi dengan pengguna**

* *Mouse*
* *User* dapat memberi input dari *mouse* berupa sebuah “klik” untuk menghentikan dan menjalankan kembali aplikasi (*pause and resume*)
* *Keyboard*
* Tombol KEY LEFT digunakan untuk menggerakkan kotak pemantul ke arah kiri
* Tombol KEY RIGHT digunakan untuk menggerakkan kotak pemantul ke arah kanan
* Tombol PAGE UP digunakan untuk membesarkan ukuran *circle* dan *square*
* Tombol PAGE DOWN digunakan untuk mengecilkan ukuran *circle* dan *square*
* Tombol F1 untuk mengganti mode window menjadi *fullscreen* atau *windowed*
* Tombol ESC untuk keluar dari aplikasi

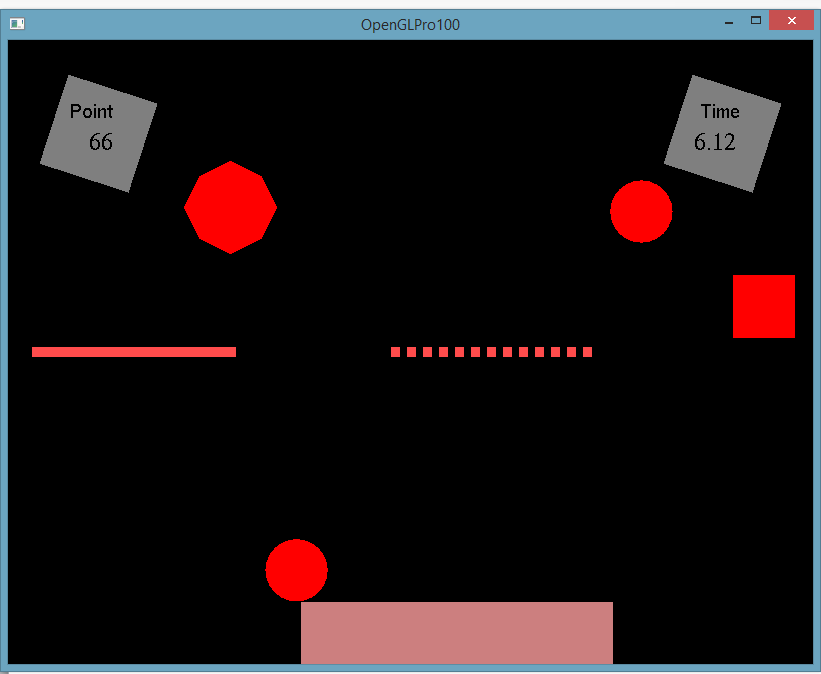
**Tampilan project :**

1. Saat aplikasi dijalankan, maka akan ditampilkan beberapa aturan teknis seperti total poin yang perlu diraih dan tombol keyboard serta klik mouse yang dapat digunakan selama permainan.

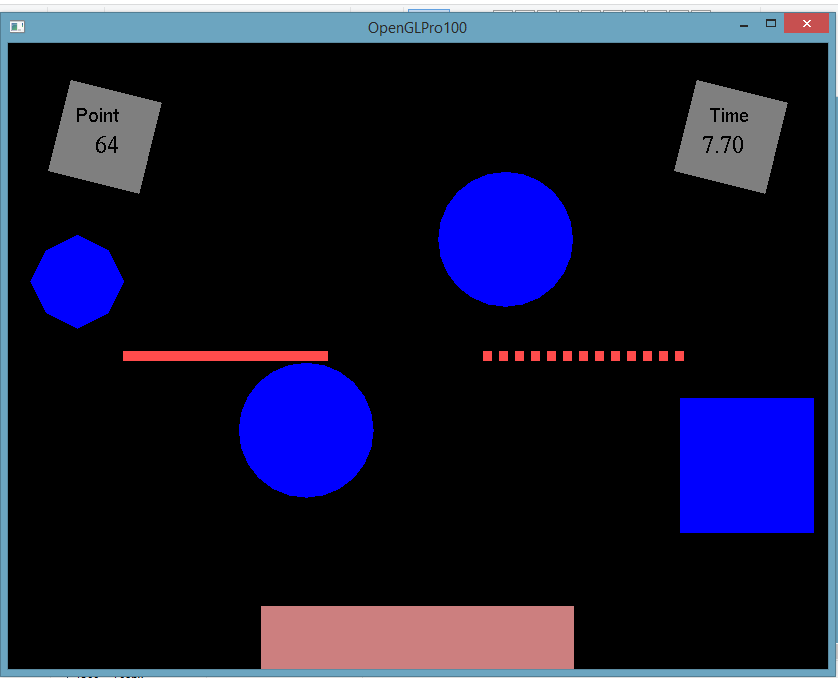


1. Saat permainan berlangsung, dimana *user* meminta untuk diberikan 1 buah *square*, 2 buah *circle*, dan 1 buah *octagon*.

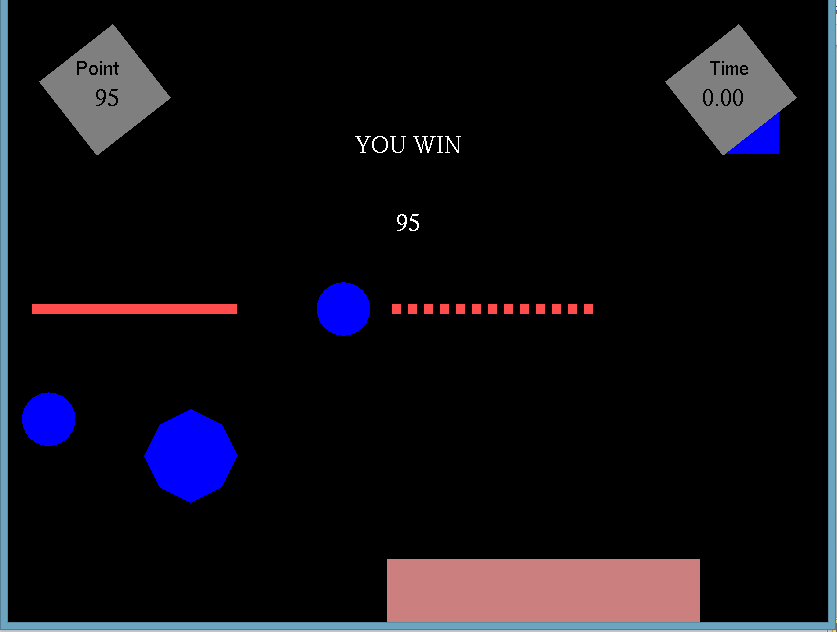
Total poin minimum yang perlu diraih adalah 12, dimana saat ini *user* sudah berhasil meraih poin 66 dan sisa waktu nya 6.12 detik.



1. Saat permainan sedang berlangsung dan *user* menekan tombol PAGE UP untuk membesarkan ukuran *circle* dan *square*.



1. Saat permainan selesai dan *user* mendapatkan total poin sebesar 95.

Status akhir permainan adalah YOU WIN.

*Untuk melihat contoh proses permainan secara lebih detil, silahkan lihat pada video demo*: <https://www.youtube.com/watch?v=NPggBSrDQzo>