

## **BAB IV IMPLEMENTASI SISTEM**

Implementasi merupakan tahap pengembangan rancangan menjadi kode program. Pada awal bagian ini dijabarkan spesifikasi perangkat keras dan lunak yang m Implementasi dalam tahap pengembangan rancangan menjadi kode program. Pada awal bagian ini dijabarkan spesifikasi perangkat keras dan lunak yang mana program diimplementasikan. Bagian utama implementasi adalah penjabaran rancangan kelas menjadi kelas yang ditulis dalam sintaks Bahasa Pemrograman. Disamping itu disajikan juga tampilan Aplikasi Tes Akademik (Studi Kasus : Unmuh Surabaya).

### **4.1 Spesifikasi Software dan Hardware**

Pada sub bab ini menerangkan tentang *software* dan *hardware* yang digunakan untuk Aplikasi Tes Akademik.

#### 4.1.1 Pengembangan

Pada bab ini menerangkan spesifikasi *hardware* dan *software* yang digunakan pada saat dilakukannya pengembangan Aplikasi Tes Akademik.

##### a. *Hardware*

1. Notebook dengan spesifikasi antara lain :
  - a. Processor : Intel Core i3
  - b. Memori : 2 Gb
  - c. Ruang Penyimpanan : 500 GB

##### b. *Software*

Dalam menerapkan rancangan yang dibuat, dibutuhkan beberapa *software* untuk membuat program tes akademik, antara lain :

##### 1. *xampp*

Xampp menyediakan semua komponen yang diperlukan. Antara lain *Apache* Sebagai Servernya *PHP* sebagai bahasa untuk pembuatan program *PostgreSQL* sebagai databases.

##### 2. Sistem Operasi

Dapat digunakan di semua Sistem Operasi.

#### 4.1.2 Implementasi Pengguna

Pada bab ini menerangkan spesifikasi *hardware* dan *software* yang digunakan pada saat dilakukannya implementasi aplikasi oleh dosen.

a. *Hardware*

1. *Komputer*

*Komputer* digunakan untuk menjalankan program aplikasi yang telah dikembangkan. Sebagai contoh *komputer* yang digunakan adalah :

- a. Processor : Intel Pentium
- b. Memori : 1 Gb
- c. Layar : 14 Inches
- d. Ruang Penyimpanan : 320 GB

b. *Software*

Untuk melakukan implementasi aplikasi ini, *software* yang dibutuhkan antara lain :

- 4.1 Xampp 1.7
- 4.2 PostgreSQL 9.2

## 4.2 **Berkas**

Pada bab ini menerangkan berkas – berkas yang digunakan untuk Aplikasi Tes Akademik (Studi kasus : Unmuh Surabaya).

### 4.2.1 File Program Aplikasi

Pada bagian ini membahas contoh *class* dan beberapa fungsi – fungsi penting yang digunakan Aplikasi Tes Akademik. Berikut merupakan beberapa contoh *class* dan beberapa fungsi yang digunakan dalam aplikasi :

1. *Object.php*

Fungsi *class* ini adalah untuk membuat object dan menentukan sebuah attributes yang akan dibuat sebagai instance dari kelas tersebut.

2. *Build\_function.php*

Fungsi *class* ini adalah untuk pengelompokan kode program akan lebih terstruktur sesuai dengan fungsi.

3. *Menu\_Utama.php*

Fungsi *class* ini adalah untuk melihat tampilan *admin* setelah berhasil login

4. *Index\_admin.php*

Fungsi *class* ini adalah untuk melihat *login admin* saat pertama kali membuka aplikasi *admin*.

5. *Load\_kategori.php*

Fungsi *class* ini adalah untuk memanggil per kategori yang nantinya akan di panggil dalam program

6. *Load\_soal.php*

Fungsi *class* ini adalah untuk memanggil semua soal yang nantinya akan di tampilkan dalam bank soal.

7. My\_lcg.php  
Fungsi *class* ini adalah untuk membuat algoritma pengacakan soal.
8. Submit\_jawaban.php  
Fungsi *class* ini adalah untuk membuat sebuah temporary soal berdasarkan no peserta.
9. ujianResource.php  
Fungsi *class* ini adalah untuk menyimpan resource variabel jawaban hasil peserta.

### 4.3 Implementasi Algoritma *Linear Congruential Generators*

Bedasarkan analisis dan perancangan sistem pada bab tiga, algoritma acak yang digunakan dengan rumus :

$$Val_n = ((a * X_{1n}) + b) \bmod m$$

Diimplementasikan ke dalam bentuk *sourcecode* sebagai berikut :

```
<?php
class MyLCG {
    function __construct() {
    }
    public function get_lcg($m) {
        $b = ($m/2)+1;
        $a = ($b/2)+1;
        $hasil = array();
        for($i=0;$i<$m;$i++){
            if($i==0) $hasil[$i] = ($a*rand(0, $m) + $b) % $m;
            else $hasil[$i] = ($a*$hasil[$i-1] + $b) % $m;
        }
        return $hasil;
    }
}
$MyLCG = new MyLCG();
?>
```

Keterangan :

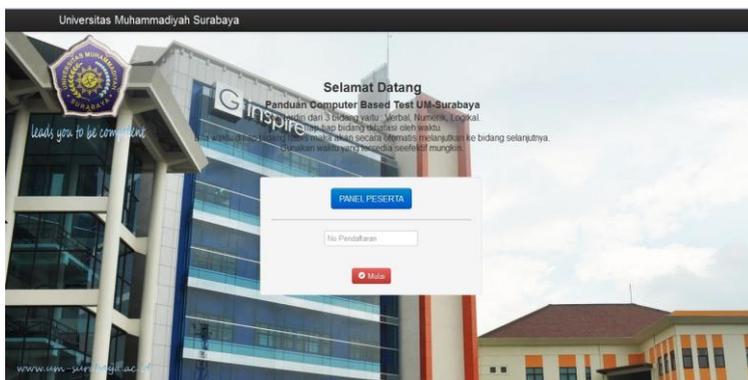
m di sini adalah bilangan *exclusive* dari *random* atau sebagai batas atas. M akan diisi oleh banyaknya soal yang akan diacak.  $X_{1n+1}$  adalah *class* yang akan menyimpan umpan(*seed*) dari nilai yang di hasilkan sebelumnya. A adalah factor pengali dan B adalah increment.  $X_{1n+1}$  akan di simpan di hasil berupa array.

## 4.4 Implementasi Rancangan Tampilan

Berdasarkan pada rancangan tampilan baik *client* maupun *server* yang ada pada bab tiga, maka disini akan ditampilkan implementasi rancangan yang sudah diterapkan dalam aplikasi *web*.

### 4.4.1 Implementasi Rancangan Tampilan Peserta

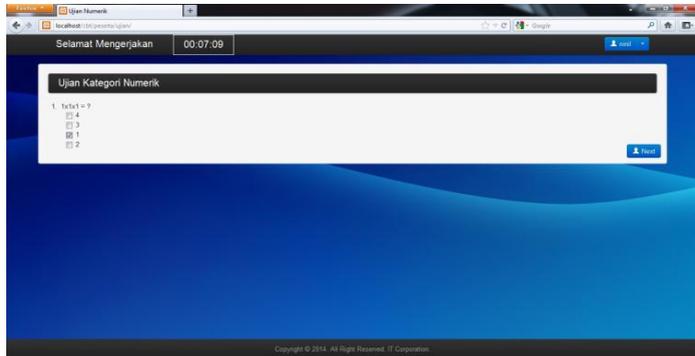
#### a) *Login* Peserta



Gambar 4.1 *Login* Peserta

Gambar 4.1 adalah implementasi dari rancangan *login* peserta. Halaman ini terdapat *field* nomor pendaftaran untuk *login*.

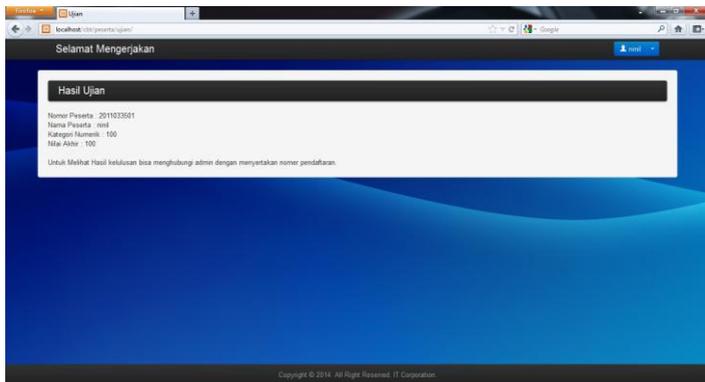
b) Halaman Pengerjaan Soal



Gambar 4.2 Halaman Pengerjaan Soal

Gambar 4.2 adalah lembar pengerjaan soal yang tampil setelah peserta berhasil *login* menggunakan nomor pendaftaran. Di atas terdapat waktu yang berjalan mundur serta terdapat tombol *next* untuk melanjutkan ke soal selanjutnya.

c) Halaman Hasil Pengerjaan

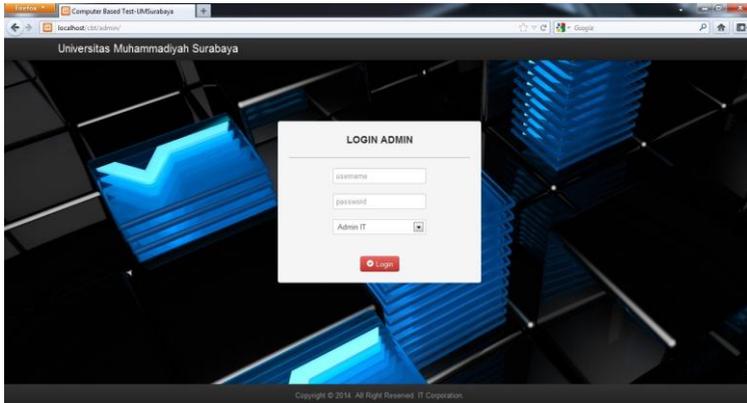


Gambar 4.3 Halaman Hasil Pengerjaan

Gambar 4.3 adalah implementasi dari rancangan tampilan hasil pengerjaan soal. Tampilan ini muncul setelah peserta menyelesaikan pengerjaan soal dan berisi tentang nomor pendaftaran, nama peserta, nilai akhir, serta nilai per kategori.

## 4.4.2 Implementasi Rancangan Tampilan *Admin*

### a) *Login Admin*



Gambar 4.4 *Login Admin*

Gambar 4.4 menunjukkan untuk *login* ke halaman *menu admin* harus memasukan *username* dan *password*. Halaman ini berisi informasi *login* ke *menu admin* dan terdapat *field username password hak akses* yang berfungsi untuk *login*.

### b) *Halaman Menu Utama*

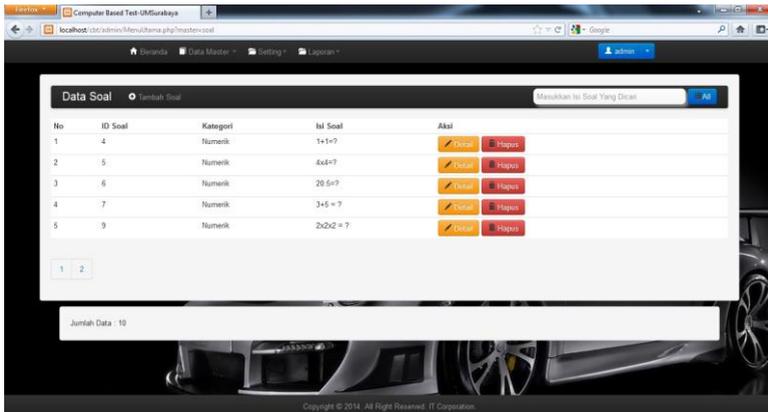


Gambar 4.5 *Halaman Menu Utama*

Gambar 4.5 adalah halaman di mana seorang *admin* sukses melakukan *login*. Di halaman ini ditampilkan beberapa menu seperti *data master*

dan *setting*. Di mana *data master* berisi *input* peserta, *admin*, soal sedangkan *setting* berisi input jurusan, kategori serta *menu* laporan yang berisi tentang hasil peserta ujian.

c) Halaman Soal



Gambar 4.6 Halaman Soal

Pada gambar 4.6 terdapat konten soal yang berisi tabel soal yang mempunyai *action edit delete* dan di bagian bawah terdapat *pagination* untuk menuju ke soal selanjutnya. serta di *navbar* terdapat tambah soal dan pencarian nama soal.

d) *Input Soal*

**Tambah Soal** [X]

Kategori:

Isi Soal

Rich text editor toolbar: B I U [bulleted list] [numbered list] [link] [font size] Font Size... Font Family. Font Format [insert image] [insert video]

Jawaban

a.  +

b.  +

c.  +

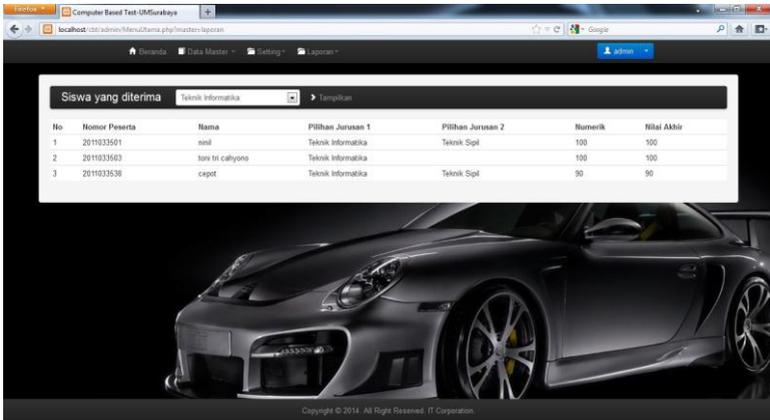
d.  +

+ Simpan Data    Batal

Gambar 4.7 *Input Soal*

Pada gambar 4.7 terdapat *combobox* kategori, *text field* soal jawaban dan juga *radiobutton* kunci. Dalam *form* ini juga sudah dilengkapi dengan *form validation* sehingga akan ada peringatan bila terdapat inputan yang dilewati.

e) Halaman Hasil Ujian



Gambar 4.8 Halaman Hasil Ujian

Pada gambar 4.8 berisi informasi hasil ujian peserta. Konten berisi nomor pendaftaran, nama, pilihan jurusan, nilai kategori, nilai akhir.

f) Halaman *Setting* Kategori



Gambar 4.9 Halaman *Setting* Kategori

Pada gambar 4.9 terdapat konten kategori yang berisi tabel kategori yang mempunyai *action insert edit delete*.

g) Halaman *input setting* kategori.



Gambar 4.10 Halaman *Input setting* kategori

Pada gambar 4.10 terdapat *text field* nama,waktu ,jumlah soal. Dalam *form* ini juga sudah dilengkapi dengan *form validation* sehingga akan ada peringatan bila terdapat inputan yang dilewati.

h) Halaman *Setting Jurusan*

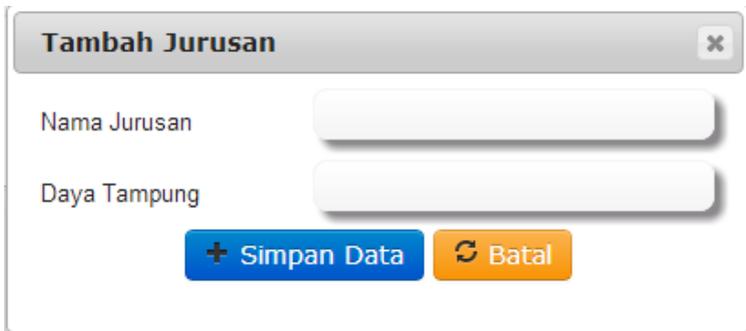


ID Jurusan	Nama Jurusan	Daya Tampung	Aksi
1	Teknik Informatika	6	 Update  Hapus
2	Teknik Sipil	10	 Update  Hapus

Gambar 4.11 Halaman *Edit Setting Jurusan*

Pada gambar 4.11 terdapat konten kategori yang berisi table jurusan, kategori, jumlah daya tampung yang mempunyai *action insert edit delete*.

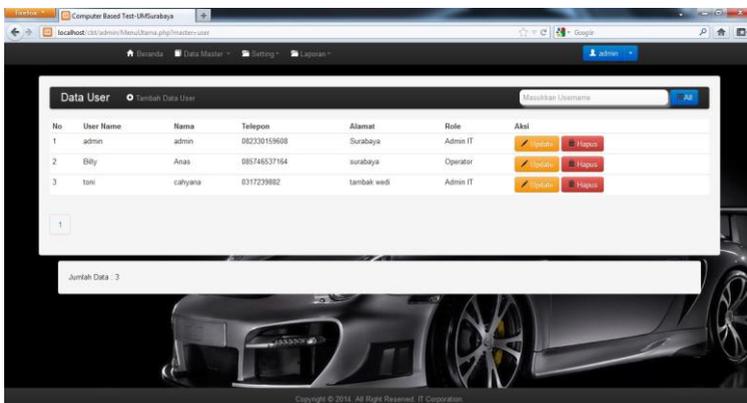
i) Halaman *Input Setting Jurusan*



Gambar 4.12 Halaman *Input Setting Jurusan*

Pada gambar 4.12 terdapat *text field* nama jurusan, daya tampung. Dalam *form* ini juga sudah dilengkapi dengan *form validation* sehingga akan ada peringatan bila terdapat inputan yang dilewati.

j) Halaman *Operator*



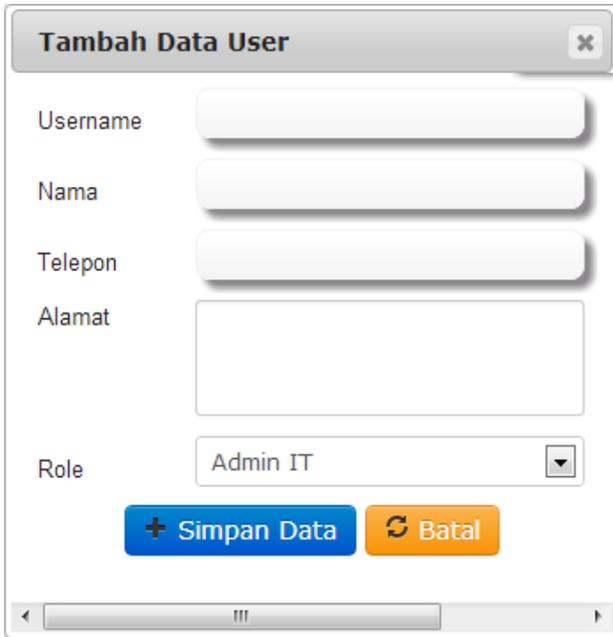
No	User Name	Nama	Telepon	Alamat	Role	Aksi
1	admin	admin	082330159608	Surabaya	Admin IT	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Billy	Anas	083746537164	surabaya	Operator	<a href="#">Edit</a> <a href="#">Hapus</a>
3	toni	cahyana	0317239882	tambak wedi	Admin IT	<a href="#">Edit</a> <a href="#">Hapus</a>

Jumlah Data : 3

Gambar 4.13 Halaman *Operator*

Pada gambar 4.13 terdapat konten operator yang berisi tabel operator yang mempunyai *action edit delete* dan di bagian bawah terdapat *pagination* untuk menuju ke soal selanjutnya. serta di *navbar* terdapat tambah soal dan pencarian nama soal.

k) *Input Operator*



The image shows a web form titled "Tambah Data User" with a close button (X) in the top right corner. The form contains the following fields and controls:

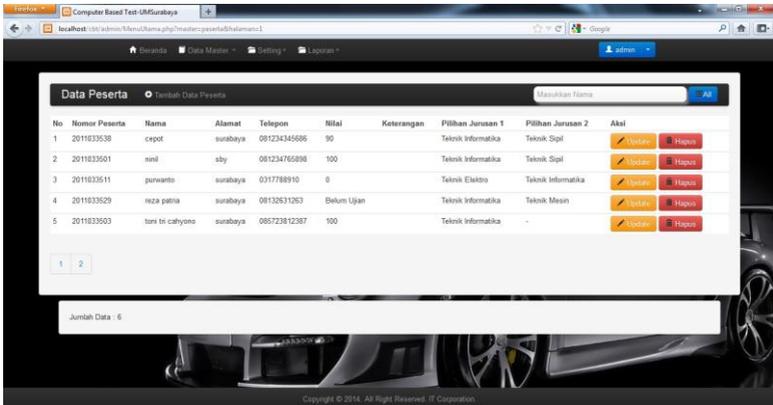
- Username:** A text input field.
- Nama:** A text input field.
- Telepon:** A text input field.
- Alamat:** A larger text input field.
- Role:** A dropdown menu currently displaying "Admin IT".
- Buttons:** A blue button labeled "+ Simpan Data" and an orange button labeled "Batal" with a refresh icon.

At the bottom of the form, there is a horizontal scrollbar.

Gambar 4.14 *Input Operator*

Pada gambar 4.14 terdapat *field username*, nama, telepon, alamat, *role* yang di dalam *form* ini juga sudah dilengkapi dengan *form validation*. sehingga akan ada peringatan bila terdapat inputan yang dilewati.

## l) Halaman *Peserta*



Gambar 4.15 Halaman *Peserta*

Pada gambar 4.15 terdapat konten peserta yang berisi tabel peserta yang mempunyai *action edit delete* dan di bagian bawah terdapat *pagination* untuk menuju ke soal selanjutnya. serta di *navbar* terdapat tambah peserta dan pencarian nama peserta.

m) *Input Peserta*

The image shows a web form titled "Tambah Data Peserta" with a close button (X) in the top right corner. The form contains the following fields:

- Nomor Peserta: A single-line text input field.
- Nama: A single-line text input field.
- Alamat: A multi-line text input field.
- Telepon: A single-line text input field.
- Keterangan: A multi-line text input field.
- Pilihan Jurusan 1: A dropdown menu with the placeholder text "---Pilih Jurusan---".
- Pilihan Jurusan 2: A dropdown menu with the placeholder text "---Pilih Jurusan---".

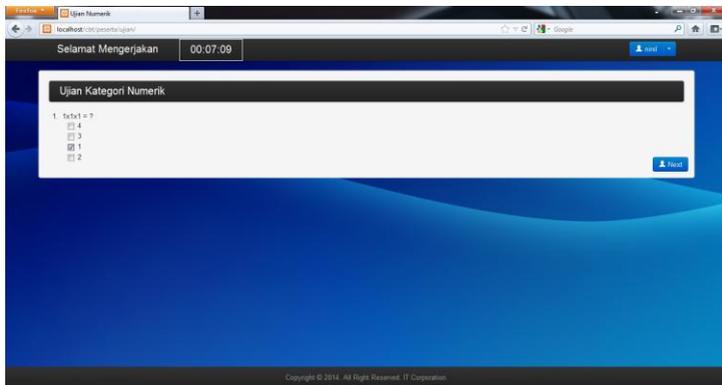
At the bottom of the form, there are two buttons: a blue button with a plus sign and the text "+ Simpan Data", and an orange button with a refresh icon and the text "Batal". Below the buttons is a horizontal scrollbar.

Gambar 4.16 *Input Peserta*

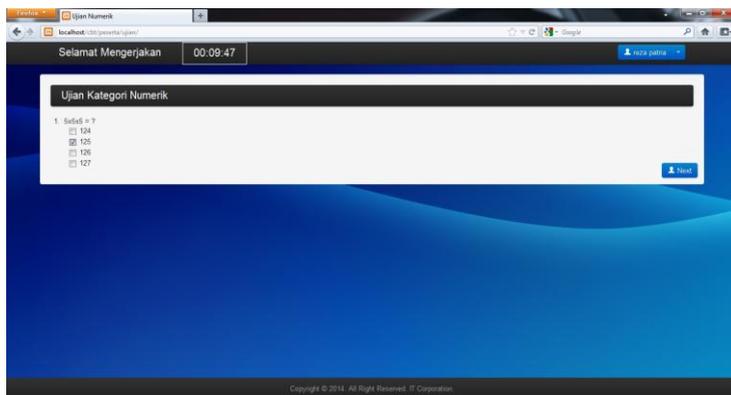
Pada gambar 4.16 terdapat *field* nomer peserta nama telepon alamat keterangan serta *combobox* pilihan jurusan 1 dan 2 .Dalam *form* ini juga sudah dilengkapi dengan *form validation* sehingga akan ada peringatan bila terdapat inputan yang dilewati.

### 4.4.3 Uji Implementasi Algoritma

Dari implementasi sistem tes potensi akademik terdapat algoritma LCG untuk pengacakan soal. Hal ini dapat dilihat dengan cara membandingkan dua soal yang muncul bersama dalam komputer yang berbeda apakah mempunyai urutan soal yang sama atau kah tidak. Ujicoba algoritma dapat dilihat pada gambar 4.17 dan gambar 4.18.



Gambar 4.17 Uji Algoritma 1



Gambar 4.18 Uji Algoritma 2

Dari gambar dapat dilihat bahwa gambar 4.17 mempunyai urutan soal yang berbeda dengan gambar 4.18.

Halaman ini sengaja di kosongkan