

## BAB IV

### IMPLEMENTASI

Pada bab ini membahas tentang implementasi dari proses perancangan yang telah diuraikan pada bab sebelumnya. Implementasi sistem dilakukan untuk mengetahui keakuratan sistem yang telah dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi lingkungan (spesifikasi perangkat keras dan perangkat lunak) di mana sistem diimplementasikan, implementasi program, implementasi antarmuka sistem, dan implementasi uji coba.

#### 4.1 Spesifikasi Sistem

Spesifikasi sistem yang sesuai dibutuhkan agar proses – proses dapat diimplementasikan dan berjalan tepat. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

##### 4.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan sistem penilaian otomatis jawaban esai menggunakan algoritma *Levenshtein Distance* menggunakan computer dengan spesifikasi :

1. Processor Intel® Core™ 2 Duo CPU P7450 @ 2.13GHz
2. VGA Mobile Intel® 4 Series Express Chipset Family
3. Memory 4096MB RAM
4. Harddisk 320 GB
5. Monitor 14.1”

##### 4.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan sistem penilaian otomatis jawaban esai menggunakan algoritma *Levenshtein Distance* meliputi :

1. Sistem Operasi *Windows 7 Ultimate* 32-bit
2. XAMPP 1.7.2



## 2. Kelas *Levenshtein*

Kelas *Levenshtein* digunakan untuk proses perhitungan *string matching* setelah proses *preprocessing*, dimulai dari perhitungan jarak dari algoritma *string matching* yang digunakan dalam perhitungan kesamaan (*similarity*) antara jawaban mahasiswa dengan kunci jawaban. Terdapat fungsi `LD()` pada kelas *Levenshtein* merupakan fungsi *Levenshtein Distance* untuk menghitung jarak beda antar dua *string* masukan.

### 4.2.2 Tahapan Pemrosesan

Langkah pertama yang dilakukan setelah jawaban diinputkan adalah dilakukan tahap *preprocessing* awal yaitu *case folding* yang dapat dilihat pada kode sumber 4.1, merubah semua huruf menjadi huruf kecil semua dan menghilangkan tanda bacanya. Semua tanda baca ditampung ke dalam objek *array*.

```
public function caseFolding($string) {
    $string2      = strtolower($string); // casefolding
    $delimiter    = array('.', ',', '!', '"', '"', '"', '-', '/', '{', '}', '+',
    '\',
    '@', '#', '$', '%', '^', '&', '*', '(', ')', '?', '>', '<', '[', ']', '|', '\', '~
    ', ':', ':', '=', '\\', "\n", "\r"); //tanda baca
    $teksTemp     = str_replace($delimiter, " ", $string2); //
    //menghilangkan tanda baca diganti spasi
    $teksTemp2    = trim($teksTemp); // menghilangkan spasi kiri
    dan //kanan string
    return $teksTemp2;
}
```

Kode Sumber 4.1 Source code proses *case folding*

Tahap *preprocessing* berikutnya adalah *tokenizing* yaitu memecah jawaban menjadi kata-kata yang disimpan dalam array, baik itu kunci jawaban dosen maupun jawaban dari mahasiswa. Fungsi *tokenizingSubstring()* dapat dilihat pada kode sumber 4.2.

```
public function tokenizingSubstring ($string){
    $a=0;
    $teksTemp = $this->caseFolding($string); //memanggil
fungsi case folding
    $temp = explode(" ",trim($teksTemp)); //
tokenizing/parsing

    for($i=0;$i<count($temp);$i++){
        // Pengecekan jika ada kalimat yang tidak standar
misal: pergi.kemana
        if($temp[$i]!='')
            unset($temp[$i]);
    }

    // Pengisian nilai array
    foreach ($temp as $word):
        if($word!=''){
            $result[$a] = trim($word);
            $a++;
        }
    endforeach;

    return $result;
}
```

Kode Sumber 4.2 Source code proses tokenizing

Langkah selanjutnya adalah melakukan proses *filtering* yaitu menghilangkan kata-kata yang tergolong *stopwords*. Kemudian kata-kata yang tidak termasuk dalam *stopwords* akan dilakukan proses *stemming*.

Proses filtering pada kode sumber 4.3 yang dilakukan pertama kali adalah melakukan pemanggilan data list stopwords yang disimpan dalam database kemudian ditampung sementara ke objek array *stopword[]*. Kemudian potongan tiap kata dari kalimat dilakukan pengecekan dengan fungsi *array\_diff* untuk mendapatkan potongan kata baru yang bersih dari *stopword[]*.

```

public function filtering($string){ // Menghilangkan kata tidak
penting dari database stopwords

    $query = mysql_query("SELECT * FROM m_stopword");
        while ($row = mysql_fetch_array($query)){
            $stopword[] = trim($row['stopword']); // menampung
daftar stopwords dari database kedalam array
        }

        if ($string)
            $result = array_diff($string, $stopword); //
array diff membuang kata yang sama dari 2 array
        return $result;
    }

```

#### Kode Sumber 4.3 Source code proses filtering

Kata yang dicocokkan sama dengan kata yang ada di tabel stopwords maka kata tersebut akan dihilangkan. Sedangkan kata yang tidak ditemukan dalam tabel *stopword* akan dilakukan proses *stemming* untuk mengembalikan semua bentuk kata kedalam bentuk kata dasarnya, dan kata hasil *stemming* tersebut dijadikan sebagai *keyword*. Kumpulan kata akan dicocokkan dengan tabel stopwords yang ada di database. Kata hasil *stemming* akan disimpan dalam bentuk *array*. Setelah proses *stemming* kata-kata tersebut kemudian diurutkan secara menaik atau *ascending* berdasar huruf awal kata. Fungsi *stemming* pada kode sumber 4.4 hanyalah perintah untuk memanggil fungsi-fungsi dari kelas induknya yang menggunakan algoritma *stemming arifin*.

```

function stemming($kata,$kd){
    $sudah = false;
    $jpg_kata = strlen($kata);
    $awalan1 = array('ber','ter','per');
    $awalan2 = array('me','di','ke','pe','se','be');
    $vokal = array('a','i','u','e','o');
    $akhiran = array('i','nya','kan','an');

    $pos12 = $kata[0].$kata[1];
    $pos13 = $kata[0].$kata[1].$kata[2];
    $pos14 = $kata[0].$kata[1].$kata[2].$kata[3];

```

```
$pos15 = $kata[0].$kata[1].$kata[2].$kata[3].$kata[4];

$jpg_pos12 = strlen($pos12);
$jpg_pos13 = strlen($pos13);
$jpg_pos14 = strlen($pos14);
$jpg_pos15 = strlen($pos15);

$temp_langkah =3; // menyiapkan var jika kalimat tdk memiliki
AWII, AKI, AKII, AKIII

if(in_array($kata,$kd)){
    return $kata;
    exit;
}

// mengecek kata belakang jika KD+AK
$kt_blk13 = substr($kata,0,-3);
$kt_blk12 = substr($kata,0,-2);
$kt_blk11 = substr($kata,0,-1);

if(in_array($kt_blk13,$kd)&& in_array(substr($kata,-
3),$akhiran)){
    return $kt_blk13;
    exit;
}

if(in_array($kt_blk12,$kd)&& in_array(substr($kata,-
2),$akhiran)){
    return $kt_blk12;
    exit;
}

if(in_array($kata,$kd)){
    return $kata;
    exit;
}
```

Kode Sumber 4.4 Source code proses stemming

Dalam mengurutkan kata-kata hasil *stemming* menggunakan fungsi *asort* pada PHP. Pengurutan elemen ini sangat diperlukan untuk mengoptimalkan kerja

algoritma *Levenshtein Distance* dalam pencocokan kesamaan jawaban. Langkah selanjutnya setelah teks terurut adalah melakukan proses pengubahan teks hasil *sorting* yang dalam bentuk *array* kedalam bentuk string. Sampai dengan tahap ini maka *preprocessing* selesai dan langkah yang selanjutnya adalah tahap pencocokan *string*.

### 4.2.3 Tahapan Pencocokan String

Tahap pencocokan string adalah tahap mencocokkan string dokumen setelah dilakukan proses *preprocessing*. Pencocokan string yang dilakukan menggunakan algoritma *Levenshtein Distance*. Kode sumber 4.5 adalah proses pencocokan string menggunakan algoritma *Levenshtein Distance*.

```
function LD($str1,$str2){ // Levenshtein Distance
    $d = Array();
    $data = Array();
    $lenStr1 = strlen($str1);
    $lenStr2 = strlen($str2);
    if ($lenStr1 == 0)
        return $lenStr2; // jika string $str1 kosong
    kembalikan langsung nilai panjang string $str2
    if ($lenStr2 == 0)
        return $lenStr1; // jika string $str2 kosong
    kembalikan langsung nilai panjang string $str1
    for ($i=0; $i <= $lenStr1; $i++){
        $d[$i] = Array();
        $d[$i][0] = $i; // inisialisasi nilai awal baris
        $data['baris'][] = $d[$i][0];
    }
    for ($j=0; $j <= $lenStr2; $j++){
        $d[0][$j] = $j; // inisialisasi nilai awal kolom
        $data['kolom'][] = $d[0][$j];
    }
    for ($i=1; $i <= $lenStr1; $i++){
        for ($j=1; $j <= $lenStr2; $j++)
```

```

    {
        $cost = substr($str1, $i - 1, 1) ==
substr($str2, $j - 1, 1) ? 0 : 1;
        $d[$i][$j] = min(
            $d[$i - 1][$j] + 1,           // deletion
            $d[$i][$j - 1] + 1,         // insertion
            $d[$i - 1][$j - 1] + $cost); //
substitution
        $data['isi'][] = $d[$i][$j];
    }
}

$data['nilai'] = $d[$lenStr1][$lenStr2];
return $data;
}

```

#### Kode Sumber 4.5 Source code Levenshtein Distance

Setelah proses *string matching* langkah selanjutnya adalah proses penghitungan nilai *similarity* jawaban. Untuk menghitung *similarity* jawaban menggunakan rumus perhitungan yang sudah dijelaskan pada bab 2 skripsi ini. Implementasi untuk proses perhitungan *similarity* jawaban dapat dilihat pada kode sumber 4.6.

```

$LD = $dokumen->LD($f_stem1,$f_stem2); // hitung diff
Levenshtein

$output3 = $output3."(Diff) Kunci Jawaban$i - Jawaban$j :
".$LD['nilai']."\n";

$temp_sim = (1 - $LD['nilai'] /
max(strlen($f_stem1),strlen($f_stem2))) * 100;

$temp_sim2 = (1 - $LD['nilai'] /
max(strlen($f_stem1),strlen($f_stem2))) * $tbobot;

$sim = round($temp_sim , 2);

$output4 = "(Similarity) Kunci Jawaban$i - Jawaban$j :
".$sim." % \n\n";

```

```
$nbobot = "(Nilai) Kunci Jawaban$i - Jawaban$j :  
$temp_sim2";
```

#### Kode Sumber 4.6 Source code Penilaian *similarity* jawaban

Source code diatas digunakan untuk menghitung nilai *similarity* dari kedua teks jawaban, sehingga dari nilai *similarity* ini dapat diketahui prosentase kemiripan jawaban dan nilai hasil ujian. Nilai *similarity* jawaban dapat diketahui dari 1 dikurangi nilai *edit distance* yang didapat dengan cara *string matching levenshtein distance*. kemudian dibagi dengan perbandingan antara panjang *string* jawaban mahasiswa dengan kunci jawaban dosen diambil yang paling tinggi. Untuk mengetahui *prosentase* kemiripan jawaban nilai *similarity* akan dikalikan dengan 100 sedangkan untuk mengetahui nilai hasil ujian nilai *similarity* dapat dikalikan dengan bobot soal.

### 4.3 Implementasi Antarmuka Sistem

Bedasarkan perancangan yang telah dijelaskan pada bab 3, metodologi dan perancangan, maka pada sub bab ini akan dijelaskan implementasi dari antarmuka. Sistem ini dijalankan oleh dua user, yaitu dosen dan mahasiswa.

Ketika memilih form dosen, maka user diharuskan untuk melakukan login terlebih dahulu. Dosen, dalam hal ini berperan untuk memasukkan data soal dan kunci jawaban ke dalam database. Pada halaman masukan pertanyaan pada form dosen terdapat beberapa field yang harus diisi, diantaranya *field* memilih kategori ujian, nomor soal, tanggal ujian. Dosen juga memberikan bobot soal yang berfungsi memberikan nilai bobot dari soal yang dibuat oleh dosen tersebut. Field yang paling utama yaitu memasukan soal dan kunci jawaban kemudian disimpan dalam database. Gambar antarmuka data uji ditunjukkan pada Gambar 4.1.

SISTEM PENILAIAN ESSAY OTOMATIS MENGGUNAKAN ALGORITMA LEVENSTHEIN DISTANCE

[Home](#)    [Pertanyaan](#)    [List Soal](#)    [Hasil](#)    [Pengujian](#)    [Logout](#)

Anda Login sebagai admin

Kategori Ujian:

No Soal:

Bobot Nilai:

Soal

apa yang dimaksud dengan page?

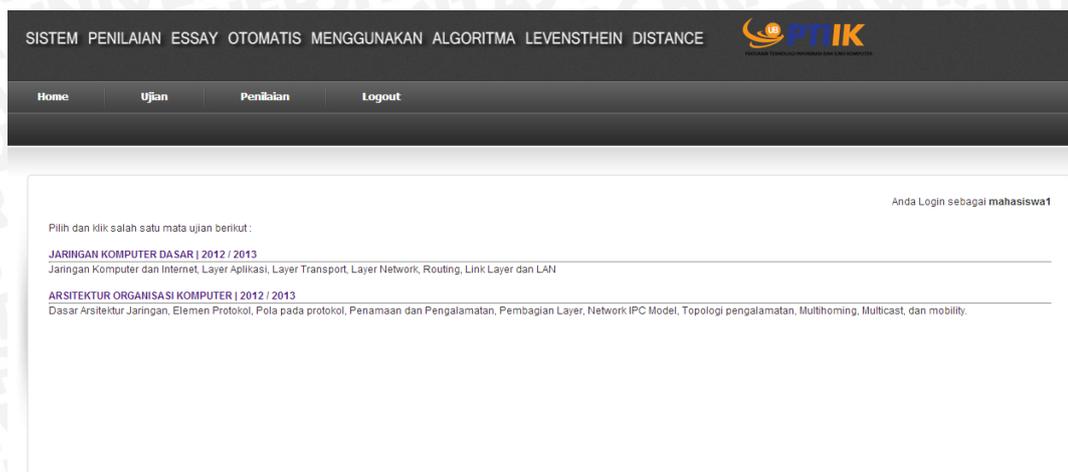
Kunci Jawaban

bagian suatu program atau potongan program

No	Kategori	Pertanyaan	Kunci Jawaban	Bobot
1	Arsitektur Organisasi Komputer	apa yang dimaksud dengan monitor program?	sebuah program yang mengamati dan mengatur serta mengendalikan atau memverifikasi operasi suatu sistem pengolahan data	20
2	Arsitektur Organisasi Komputer	apa yang dimaksud dengan scheduling?	proses yang sudah diatur dengan ketentuannya disebuah sistem operasi agar tidak terjadi suatu tabrakan dengan proses lain	20
3	Arsitektur Organisasi Komputer	apa yang dimaksud dengan swapping?	suatu proses yang dapat dialihkan sementara dari memori ke suatu tempat penyimpanan dan dipanggil kembali ke memori jika akan melanjutkan eksekusi	20
4	Arsitektur Organisasi Komputer	apa yang dimaksud dengan page?	bagian suatu program atau potongan program	20
5	Jaringan Komputer Dasar	Sebutkan 7 lapisan OSI layer?	Application, Presentation, Session, Transport, Network, Data Link, Physical	50
6	Arsitektur Organisasi Komputer	apa yang dimaksud dengan frame?	suatu bentuk bagian kerangka potongan yang tersedia di memori atau suatu area penyimpanan utama	20
7	Jaringan Komputer Dasar	apa kepanjangan dari WAN?	Wide Area Network	50

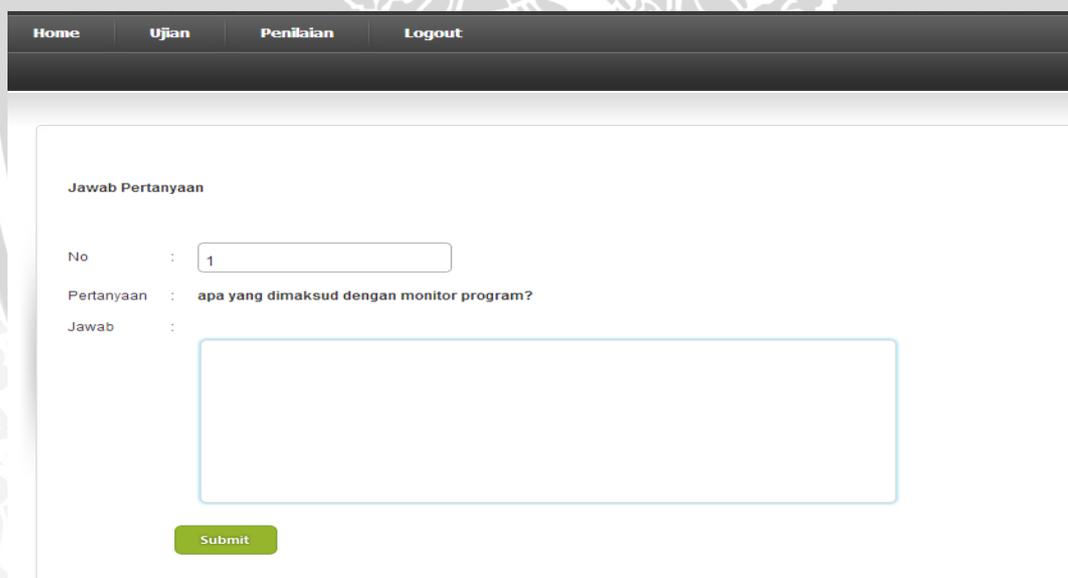
Gambar 4.1 Halaman Masukan Pertanyaan

Untuk form mahasiswa memilih jenis soal ujian dengan cara klik link soal ujian yang akan dikerjakan yang ditunjukkan pada gambar 4.2



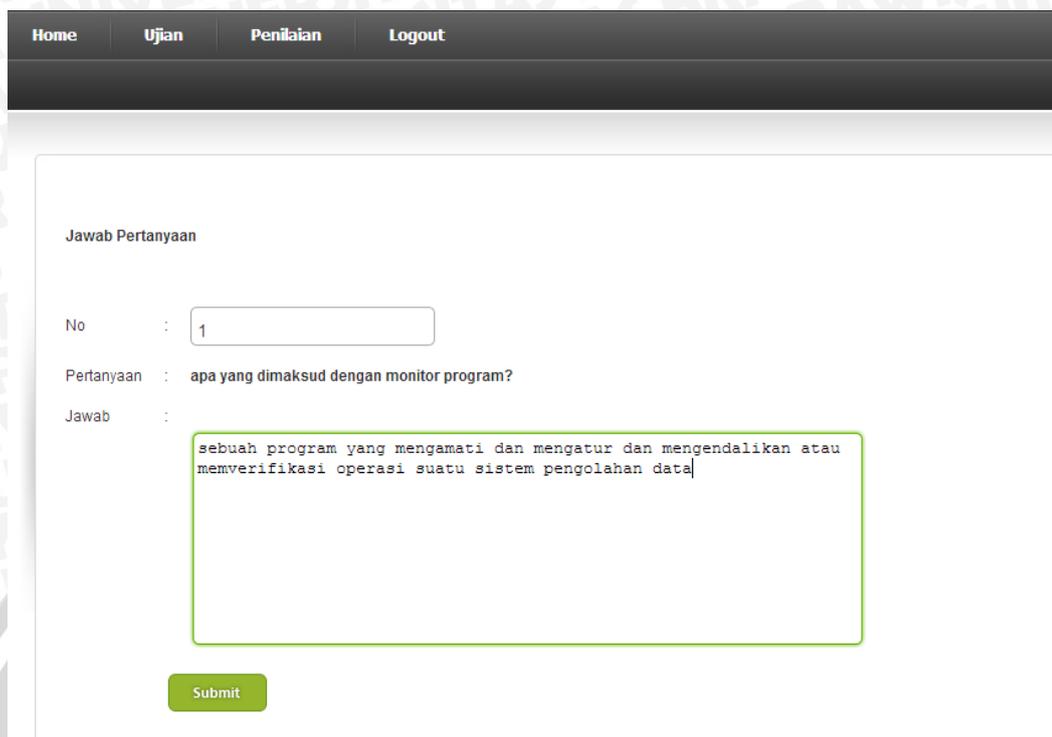
Gambar 4.2 Halaman Pilih Soal Ujian

Jika link soal tersebut di klik maka akan langsung menuju halaman kerjakan ujian yang ditunjukkan dalam gambar 4.3



Gambar 4.3 Halaman Form Jawab Ujian

Pada halaman form jawab ujian peserta ujian menjawab pertanyaan berdasarkan referensi yang sudah diberikan dosen. Pertanyaan-pertanyaan ini sebelumnya telah diinputkan dari sisi halaman dosen yang sebelumnya sudah dibahas. Jika sudah selesai menjawab pertanyaan yang ditunjukkan pada gambar 4.4 maka jawaban dimasukkan kedalam database.



Home Ujian Penilaian Logout

Jawab Pertanyaan

No : 1

Pertanyaan : apa yang dimaksud dengan monitor program?

Jawab : sebuah program yang mengamati dan mengatur dan mengendalikan atau memverifikasi operasi suatu sistem pengolahan data

Submit

Gambar 4.4 Halaman Jawab Ujian

Setelah jawaban terisi maka proses selanjutnya adalah masuk ke dalam proses review jawaban dimana saat review jawaban peserta ujian dapat melihat kembali jawaban yang telah dijawab sebelumnya dan dapat merubah jawaban tersebut. Kemudian submit jawaban maka jawaban dari peserta ujian akan diproses. Review jawaban dapat dilihat pada gambar 4.5.

Apakah Anda Yakin Dengan Jawaban Anda? :

No	Pertanyaan	Jawaban	Keterangan
1	apa yang dimaksud dengan monitor program?	sebuah program yang mengamati dan mengatur dan mengendalikan atau memverifikasi operasi suatu sistem pengolahan data	Review
2	apa yang dimaksud dengan scheduling?	proses yang sudah diatur dengan ketentuannya disebuah sistem operasi agar tidak terjadi suatu tabrakan dengan proses lain	Review
3	apa yang dimaksud dengan swapping?	suatu proses yang dapat dialihkan sementara dari memori ke suatu penyimpanan, dan dipanggil kembali ke memori jika akan melanjutkan eksekusi	Review
4	apa yang dimaksud dengan page?	halaman khusus yang sesuai untuk konten tak terbatas waktu yang gampang diakses oleh pengunjung	Review
5	apa yang dimaksud dengan frame?	satuan terkecil dalam video	Review

[Submit](#)

Gambar 4.5 Halaman *Review* Jawaban

Setelah yakin jawaban setelah review jawaban maka langkah selanjutnya adalah melakukan proses pengecekan kemiripan jawaban. Pada pengecekan kemiripan jawaban terdapat dua proses utama yaitu *preprocessing* (*tokenizing, case folding, filtering, dan stemming*), dan proses pencocokan *string*. Setelah proses diatas dilakukan semuanya maka akan didapatkan output berupa informasi jawaban yang sama, *prosentase* kemiripan jawaban (*similarity*), dan nilai. Hasil dari proses dapat dilihat pada gambar 4.6.

No	Pertanyaan	Jawaban	Nilai	Similarity
1	apa yang dimaksud dengan monitor program?	sebuah program yang mengamati dan mengatur dan mengendalikan atau memverifikasi operasi suatu sistem pengolahan data	20	100 %
2	apa yang dimaksud dengan scheduling?	proses yang sudah diatur dengan ketentuannya disebuah sistem operasi agar tidak terjadi suatu tabrakan dengan proses lain	20	100 %
3	apa yang dimaksud dengan swapping?	suatu proses yang dapat dialihkan sementara dari memori ke suatu tempat penyimpanan, dan dipanggil kembali ke memori jika akan melanjutkan eksekusi	20	100 %
4	apa yang dimaksud dengan page?	halaman khusus yang sesuai untuk konten tak terbatas waktu yang gampang diakses oleh pengunjung	3.3898305084746	16.94 %
5	apa yang dimaksud dengan frame?	satuan terkecil dalam video	2.4	12 %
			Total Nilai	65.789830508475

Gambar 4.6 Halaman Hasil Nilai Ujian