

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Implementasi Sistem**

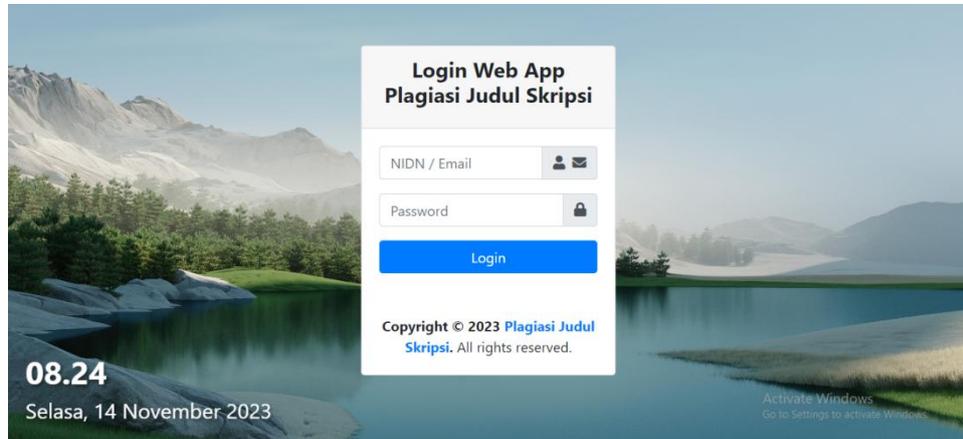
Pada bab ini menjelaskan implementasi aplikasi pendeteksi kemiripan judul skripsi yang berdasarkan analisa kebutuhan dan proses dari perancangan aplikasi tersebut. Dalam implementasi ini akan ada beberapa hal terkait dengan sistem yang akan dibangun dan juga memperhatikan ketentuan-ketentuan agar aplikasi ini berjalan dengan baik. Hasil dari penelitian ini akan berguna untuk mencari kesamaan kata yaitu kemiripan judul skripsi dengan menampilkan tingkat presentase kemiripannya. Pembahasan tentang aplikasi yang akan dimuat terdiri dari spesifikasi sistem, dan implementasi Interface.

#### **4.2 Implementasi Interface**

Pada bab ini menjelaskan tentang bagaimana implementasi antarmuka aplikasi yang mengacu pada bab sebelumnya yaitu desain antarmuka. Antarmuka ini akan menjadi salah satu patokan keberhasilan dari membangun aplikasi kemiripan judul ini. Bab ini akan menjelaskan bagaimana user interface dari sistem ini yang dibangun, menjelaskan apa saja fitur-fitur yang ada didalamnya. Sistem ini akan menampilkan antar muka admin dan user saja, karena memang focus aplikasi ini hanya untuk mencari persamaan yang hasilnya berupa presentase.

## A. Login Admin

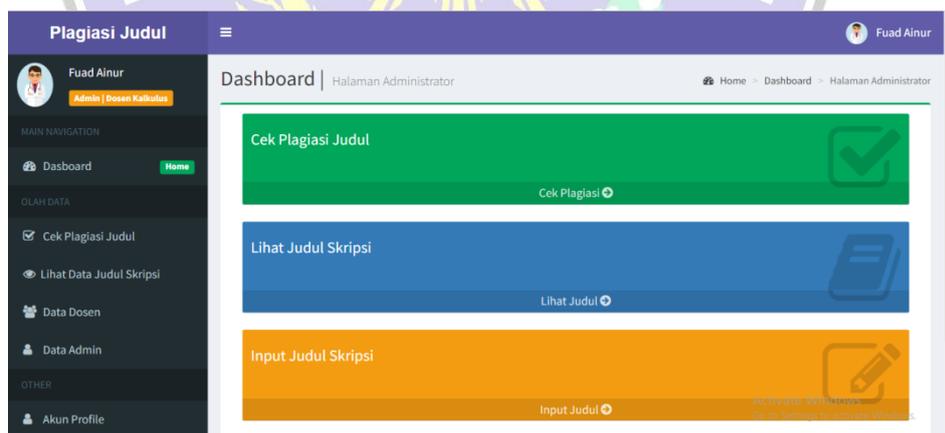
### 1. Halaman Login



Gambar 4. 1 Halaman Login

Pada gambar 4.1, halaman ini adalah tampilan awal dari sistem yang di bangun dengan melakukan sebuah login untuk admin system pendeteksi kemiripan judul skripsi dengan memasukkan data berupa username dan password.

### 2. Halaman Dashboard

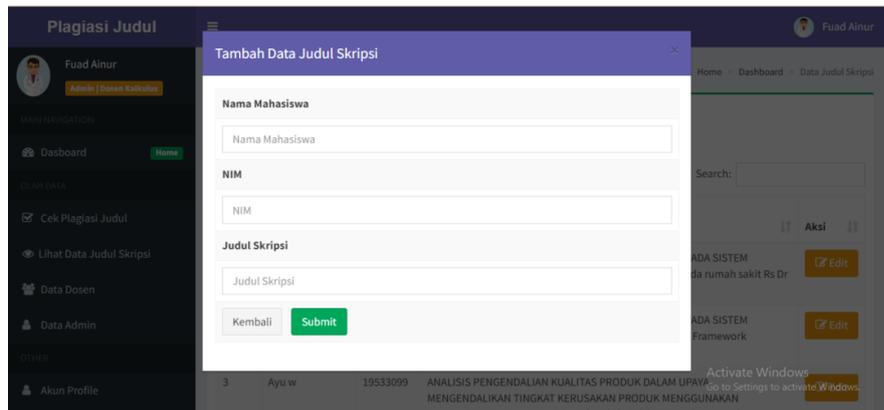


Gambar 4. 2 Dashboard

Pada halaman ini adalah halaman Dashboard dari sistem dengan memberikan beberapa menu yang dapat digunakan oleh admin, terlihat

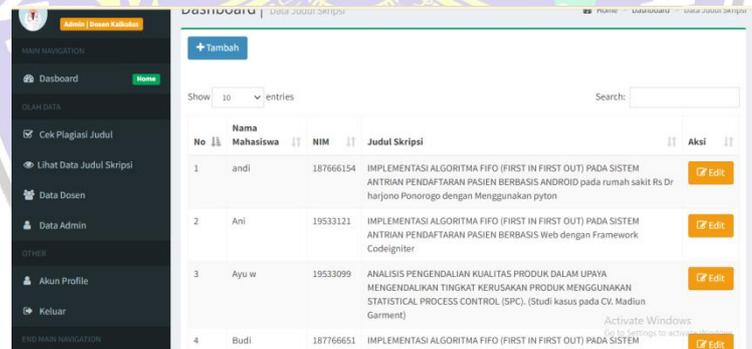
pada gambar 4.2.. Halaman dashboard berisi menu lihat data judul skripsi beserta input judul terdahulu, data dosen, data admin, dan akun profile.

### 3. Menu Lihat Data Judul Skripsi



Gambar 4. 3 input judul skripsi terdahulu

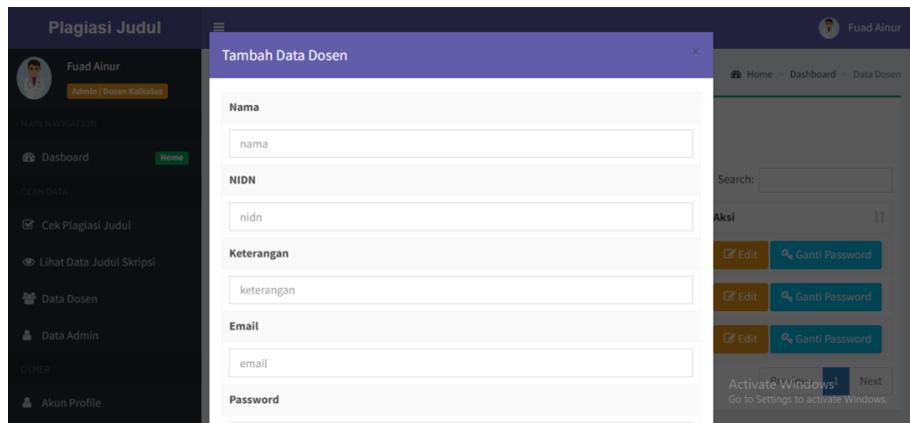
Pada gambar 4.3, admin dapat menambahkan judul skripsi angkatan sebelumnya dengan memasukkan nama mahasiswa, NIM, dan judul skripsi.



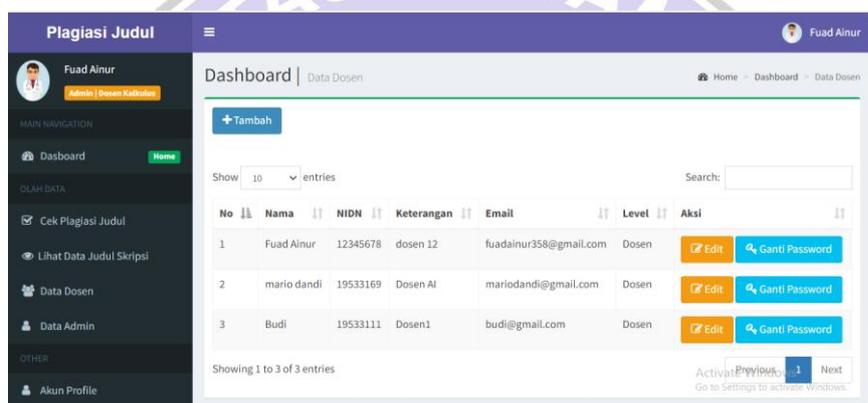
Gambar 4. 4 data judul skripsi terdahulu

Pada halaman ini berfungsi untuk mengolah judul skripsi angkatan sebelumnya, terlihat pada gambar 4.4. Setelah judul dimasukkan, judul skripsi otomatis tersimpan. Judul skripsi yang telah disimpan juga dapat di edit apabila ada kesalahan sebelumnya.

#### 4. Data dosen



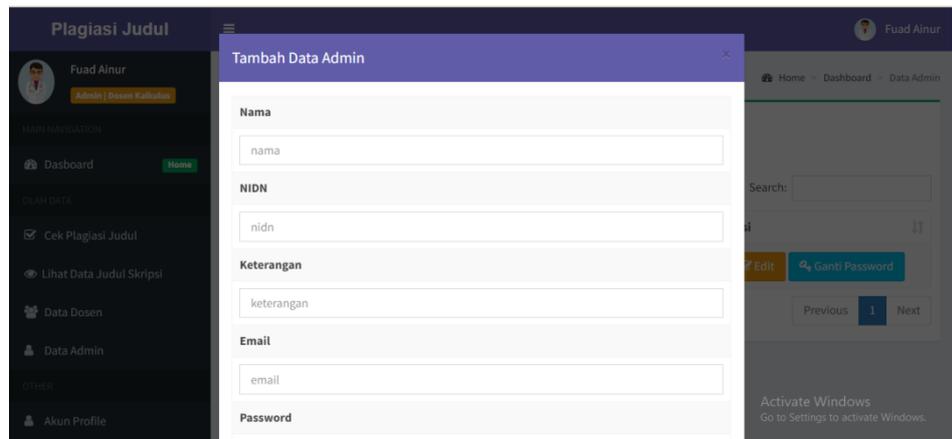
Gambar 4. 5 menambahkan data dosen



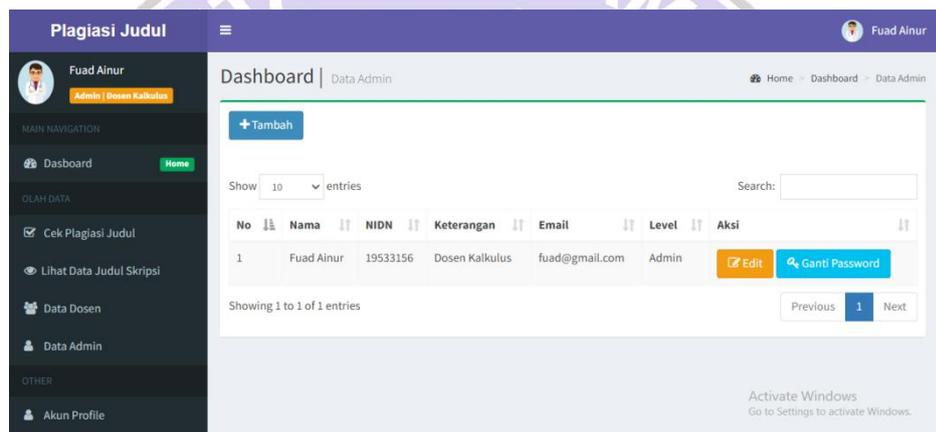
Gambar 4. 6 data dosen tersimpan

Pada halaman ini, admin dapat menambahkan user agar dapat melakukan login ke system dengan memasukkan nama dosen, nidn, keterangan, email, dan password, terlihat pada gambar 4.5. Setelah data dosen tersimpan, admin dapat mengedit data dosen tersebut, ada pada gambar 4.6.

## 5. Data admin



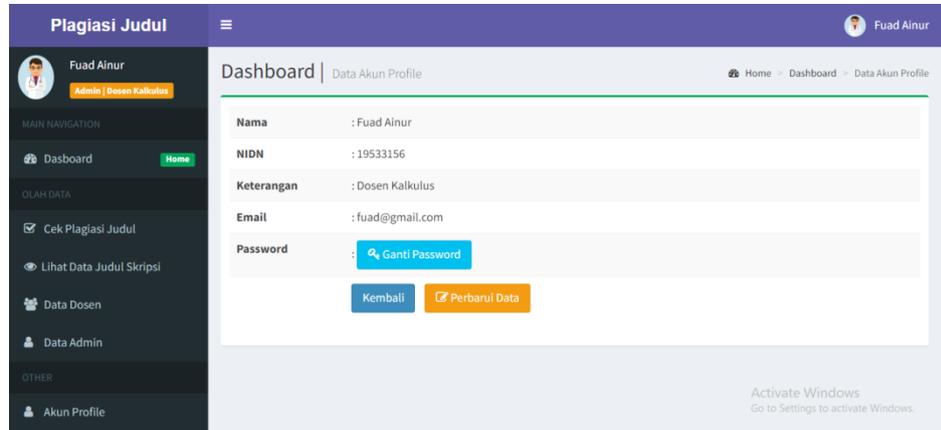
Gambar 4. 7 menambahkan admin



Gambar 4. 8 data admin tersimpan

Halaman data admin adalah untuk menambahkan admin baru dan mengolah data admin tersebut. Pada gambar 4.7, untuk menambahkan admin perlu memasukkan nama, nidn, keterangan, email, dan password. Untuk tampilan mengolah dan mengedit data admin ada pada gambar 4.8.

## 6. Akun Profile

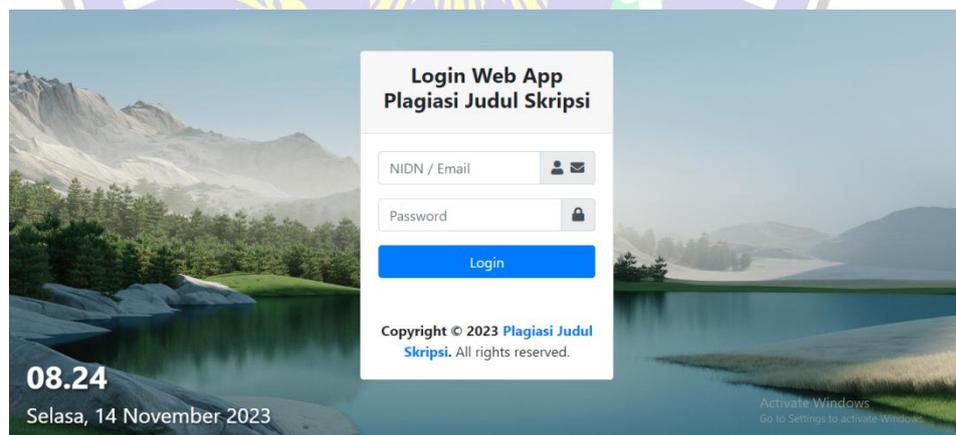


Gambar 4. 9 profil admin

Pada halaman akun profile, berisi informasi admin yang sedang login pada system. Dalam halaman ini juga dapat mengolah dan mengedit data admin, terlihat pada gambar 4.9.

### B. Login User

#### 1. Halaman login.

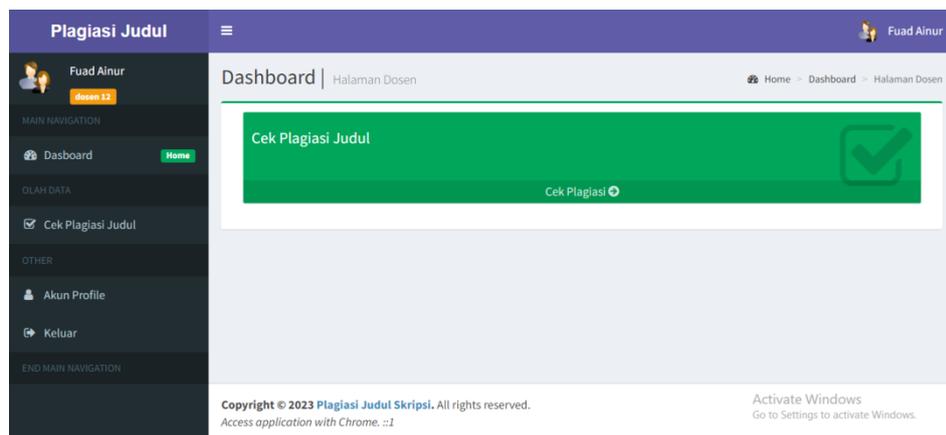


Gambar 4. 10 halaman login

Pada gambar 4.10, halaman ini adalah tampilan awal dari sistem yang di bangun dengan melakukan sebuah login untuk user dalam sistem

pendeteksi kemiripan judul skripsi dengan memasukkan data berupa username dan password.

## 2. Halaman dashboard



Gambar 4. 11 halaman dashboard

Gambar 4.11 menunjukkan halaman dashboard user. Pada dashboard user terdapat menu cek plagiasi judul skripsi dan akun profile.

## 3. Halaman cek plagiasi judul



Gambar 4. 12 input judul skripsi baru

Plagiasi Judul

Fuad Ainur

Dashboard | Hasil Cek Plagiasi

Judul skripsi anda: **implementasi algoritma fisher-yates shuffle pada permainan matching card dalam media pembelajaran bahasa indonesia**

Show 10 entries Search:

No	Nama Mahasiswa	NIM	Judul Skripsi	Persentase
21	Yulia	19533441	implementasi algoritma fisher-yates shuffle pada permainan matching card dalam media pembelajaran bahasa korea berbasis web.	75.00%
8	Gigi	19533211	implementasi algoritma decision tree pada internet of things inkubator tempe kedelai	14.29%
16	Risma	18111321	implementasi algoritma k-means clustering pada sistem informasi persebaran penyakit demam berdarah berbasis webgis	13.04%
4	Budi	18776651	implementasi algoritma fifo (first in first out) pada sistem antrian pendaftaran pasien berbasis android	12.50%

Gambar 4. 13 presentase kemiripan

Pada gambar 4.12, menunjukkan halaman cek plagiasi. user dapat melakukan cek plagiasi dengan memasukkan judul skripsi baru atau judul yang mau diajukan untuk dicek tingkat kemiripannya dengan judul skripsi angkatan sebelumnya. Lalu untuk tampilan setelah judul dicari kemiripannya terdapat pada gambar 4.13.

Judul skripsi anda: **Algoritma decision tree pada iot untuk pengelolaan suhu kandang ayam petelur**

Show 10 entries Search:

No	Nama Mahasiswa	NIM	Judul Skripsi	Persentase
24	Robot	19663374	algoritma decision tree pada iot untuk pengelolaan suhu kandang ayam petelur	100.00%
27	Udin	19663300	algoritma decision tree pada iot untuk perangkat hama ngengat tanaman bawang	97.50%
16	Gigi	19533211	implementasi algoritma decision tree pada internet of things inkubator tempe kedelai	22.22%

Gambar 4. 14 hasil presentase judul

Pada gambar 4.14, terdapat satu judul baru dan tiga judul pembanding yang ketiganya memiliki presentase kemiripan berbeda.

Dokumen 1 (A) = Algoritma decision tree pada iot untuk pengelolaan suhu kandang ayam petelur

Dokumen 2 (B) = Algoritma decision tree pada iot untuk pengelolaan suhu kandang ayam petelur

- a. Memisahkan ke dua dokumen tersebut menjadi array : (1)algoritma, (2)decision, (3)tree, (4)pada, (5)iot, (6)untuk, (7)pengelolaan, (8)suhu, (9)kandang, (10)ayam, (11)petelur

$$A = \{1,2,3,4,5,6,7,8,9,10,11\}$$

$$B = \{1,2,3,4,5,6,7,8,9,10,11\}$$

Kemudian mencari jumlah kata keseluruhan dari dua dokumen tersebut dan dari perhitungan di atas bisa dilihat jumlah kata keseluruhan yaitu 16 kata.

$$\text{Jadi } A \cup B = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

Setelah mencari jumlah seluruh kata, selanjutnya mencari kata yang sama diantara dua dokumen tersebut. Kata yang sama yaitu (1)algoritma, (2)decision, (3)tree, (4)pada, (5)iot, (6)untuk, (7)pengelolaan, (8)suhu, (9)kandang, (10)ayam, (11)petelur

$$\text{Jadi untuk jumlah } A \cap B = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

Langkah selanjutnya adalah menghitung kemiripan dari dua dokumen tersebut berdasarkan persamaan. Hasil perhitungan dikali 100 karena untuk mencari persentasenya.

$$(A, B) = \frac{11}{11} = 1 \times 100 = \mathbf{100\%}$$

Jadi nilai persamaan yang diperoleh didapat nilai kemiripan antara dua dokumen tersebut (A dan B) adalah 100%.

Dokumen 1 (A) = Algoritma decision tree pada iot untuk pengelolaan suhu kandang ayam petelur

Dokumen 2 (B) = Algoritma decision tree pada iot untuk perangkat hama ngengat tanaman bawang

- b. Memisahkan ke dua dokumen tersebut menjadi array : (1)algoritma, (2)decision, (3)tree, (4)pada, (5)iot, (6)untuk, (7)pengelolaan, (8)perangkat, (9)suhu, (10)hama, (11)kandang, (12)ngengat, (13)ayam, (14)tanaman, (15)petelur, (16)bawang.

$$A = \{1, 2, 3, 4, 5, 6, 7, 9, 11, 13, 15\}$$

$$B = \{1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16\}$$

Kemudian mencari jumlah kata keseluruhan dari dua dokumen tersebut dan dari perhitungan di atas bisa dilihat jumlah kata keseluruhan yaitu 16 kata.

$$\text{Jadi } A \cup B = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$$

Setelah mencari jumlah seluruh kata, selanjutnya mencari kata yang sama diantara dua dokumen tersebut. Kata yang sama yaitu (1)algoritma, (2)decision, (3)tree, (4)pada, (5)iot, (6)untuk

$$\text{Jadi untuk jumlah } A \cap B = \{1, 2, 3, 4, 5, 6\}$$

Langkah selanjutnya adalah menghitung kemiripan dari dua dokumen tersebut berdasarkan persamaan (2.2). Hasil perhitungan dikali 100 karena untuk mencari persentasenya.

$$(A, B) = \frac{6}{16} = 0,375 \times 100 = \mathbf{37,5\%}$$

Jadi nilai persamaan yang diperoleh didapat nilai kemiripan antara dua dokumen tersebut (A dan B) adalah 37,5%

Dokumen 1 (A) =	Algoritma decision tree pada iot untuk pengelolaan suhu kandang ayam petelur
Dokumen 2 (B) =	Implementasi algoritma decision tree pada internet of things incubator tempe kedelai

- c. Memisahkan ke dua dokumen tersebut menjadi array : (1)implementasi, (2)algoritma, (3)decision, (4)tree, (5)pada, (6)iot, (7)internet, (8)untuk, (9)pengelolaan, (10)suhu, (11)of, (12)kandang, (13)things, (14)ayam, (15)incubator, (16)petelur, (17)tempe, (18)kedelai

$$A = \{2, 3, 4, 5, 6, 8, 9, 10, 12, 14, 16\}$$

$$B = \{1, 2, 3, 4, 5, 7, 11, 13, 15, 17, 18\}$$

Kemudian mencari jumlah kata keseluruhan dari dua dokumen tersebut dan dari perhitungan di atas bisa dilihat jumlah kata keseluruhan yaitu 16 kata.

$$\text{Jadi } A \cup B = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}$$

Setelah mencari jumlah seluruh kata, selanjutnya mencari kata yang sama diantara dua dokumen tersebut. Kata yang sama yaitu (1)algoritma, (2)decision, (3)tree, (4)pada.

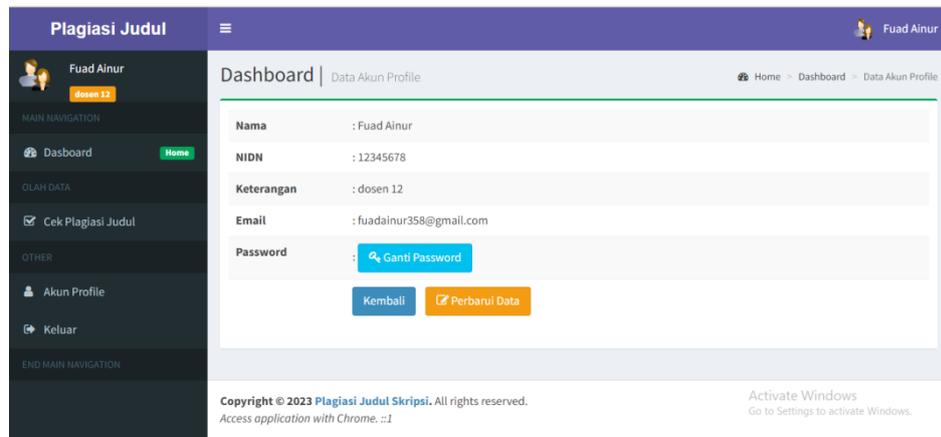
$$\text{Jadi untuk jumlah } A \cap B = \{1, 2, 3, 4\}$$

Langkah selanjutnya adalah menghitung kemiripan dari dua dokumen tersebut berdasarkan persamaan (2.2). Hasil perhitungan dikali 100 karena untuk mencari persentasenya.

$$(A, B) = \frac{4}{18} = 0,22 \times 100 = \mathbf{22\%}$$

Jadi nilai persamaan yang diperoleh didapat nilai kemiripan antara dua dokumen tersebut (A dan B) adalah 22%

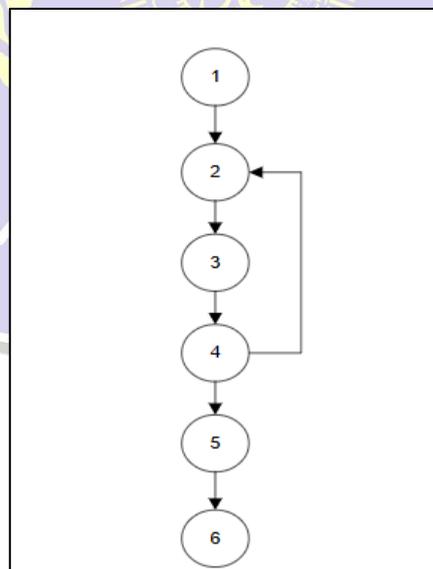
#### 4. Akun profil



Gambar 4. 15 profil user

Pada halaman akun profile, berisi informasi user yang sedang login pada sistem. Dalam halaman ini juga dapat mengolah dan mengedit data user, terlihat pada gambar 4.15

#### 4.3 Pengujian White Box testing



Gambar 4. 16 Case Login

Jalur 1 = 1-2-3-4-5-6

Jalur 2 = 1-2-3-4-2-3-5-6

Setelah diketahui jumlah jalur maka dilanjutkan dengan pengujian path yang berisikan jalur, skenario, dan hasil pengujian

Tabel 4. 1 Test Case Login

Path	1
Jalur	1-2-3-4-5-6
Skenario	<p>Start</p> <p>Masukkan username dan password</p> <p>Klik login</p> <p>Validasi data benar</p> <p>System menampilkan halaman utama</p> <p>End.</p>
Path	2
Jalur	1-2-3-4-2-3-5-6
Scenario	<p>Start</p> <p>Masukkan username dan password</p> <p>Klik login</p> <p>Validasi data salah</p> <p>Muncul pesan error. Masukkan kembali username dan password</p> <p>Klik login</p> <p>System menampilkan halaman utama</p> <p>end</p>

Halaman Pengujian	Berhasil
-------------------	----------

Tabel 4. 2 Test Case Plagiasi

Path	1
Jalur	1-2-3-4-5-6
Skenario	Start Cek plagiasi Masukkan judul skripsi Klik search System menampilkan halaman hasil presentase plagiasi End.
Halaman Pengujian	Berhasil

Setelah hasil pengujian pada tabel 4.2 berstatus berhasil, selanjutnya dilakukan Identifikasi dan kasus uji web plagiasi judul skripsi untuk proses pengujian white box.

Tabel 4. 3 identifikasi

No	Nama Class	Nama Method	Tujuan
1	Periksa	cek_plagiasi	Input judul baru untuk

---

cek  
plagiasi

---

Berikut merupakan hasil dari proses identifikasi pengujian yang telah dilakukan dan dapat dilihat pada tabel 4.3 terkait hasil atau tujuan pengujiannya

Tabel 4. 4 Tabel Uji

Nomor uji	1
Input	<pre>//Lihat Data public function input(\$value='') {     \$view = array('judul' =&gt;'Cek Plagiasi Judul Skripsi',                  'aksi' =&gt;'input',                  );      \$this-&gt;load- &gt;view('dosen/cek_plagiasi/input',\$view); }  public function calculate_similarity() {     \$userTitle = \$this-&gt;input- &gt;post('nama_judul_skripsi');      \$titles = \$this-&gt;M_judul_skripsi-&gt;view()- &gt;result();      // Threshold kemiripan yang diperbolehkan     \$threshold = 1; // Ganti dengan nilai threshold yang Anda inginkan      // Variabel untuk menyimpan data judul skripsi yang sesuai     \$matchingData = array(); // Default kosong      // Hitung kemiripan dengan judul dari database     \$similarTitles = array();     foreach (\$titles as \$dbTitle) {</pre>

---

```

        $jaccardSimilarity = $this-
>calculateJaccardSimilarity($userTitle, $dbTitle-
>nama_judul_skripsi);
        if ($jaccardSimilarity >= $threshold) {
            // Simpan data judul skripsi, nama
            mahasiswa, dan nim ke dalam array
            $matchingData[] = array(
                'nama_mahasiswa' => $dbTitle-
>nama_mahasiswa,
                'nim' => $dbTitle->nim,
                'judul_skripsi' => $dbTitle-
>nama_judul_skripsi,
                'kemiripan' =>
min($jaccardSimilarity, 100)
            );
        }
    }
}

```

---

Output

```

<?php $this->load->view('template/header'); ?>
<style>
    .highlight {
        background-color: yellow; /* Ganti warna latar belakang
sesuai preferensi Anda */
    }
</style>
<h4>Judul skripsi anda: <strong><?= $userTitle ?></strong></h4>
<div class="table-responsive">
    <table id="example1" class="table table-bordered table-
striped">
    <thead>
    <tr>
    <th>No</th>

```

---

```
<th>Nama Mahasiswa</th>
```

```
<th>NIM</th>
```

```
<th>Judul Skripsi</th>
```

```
<th>Persentase</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<?php
```

```
    // Membuat array asosiatif yang berisi judul, nama
    mahasiswa, nim, dan kemiripan
```

```
    $dataForSorting = array();
```

```
    foreach ($matchingData as $data) {
```

```
        $dataForSorting[] = array(
```

```
            'nama_mahasiswa' => $data['nama_mahasiswa'],
```

```
            'nim' => $data['nim'],
```

```
            'judul' => $data['judul_skripsi'],
```

```
            'kemiripan' => min($data['kemiripan'], 100)
```

```
        );
```

```
    }
```

```
    // Fungsi untuk mengurutkan array berdasarkan kemiripan
```

```
    // usort($dataForSorting, function ($a, $b) {
```

```
        // return $a['kemiripan'] < $b['kemiripan'];
```

---

```
// });
```

```
$no = 1;
```

```
foreach ($dataForSorting as $data) {
```

```
    $nama_mahasiswa = $data['nama_mahasiswa'];
```

```
    $nim = $data['nim'];
```

```
    $nama_judul_skripsi = $data['judul'];
```

```
    $similarity = $data['kemiripan'];
```

```
    $colorClass = "";
```

```
    if ($similarity >= 60) {
```

```
        $colorClass = 'bg-red'; // Merah untuk kemiripan di  
        atas 60% dan 100%
```

```
    } elseif ($similarity >= 45) {
```

```
        $colorClass = 'bg-yellow'; // Kuning untuk kemiripan  
        di antara 45% dan 59.9%
```

```
    } elseif ($similarity >= 26) {
```

```
        $colorClass = 'bg-blue'; // Biru untuk kemiripan di  
        antara 26% dan 44.9%
```

```
    } elseif ($similarity >= 0) {
```

```
        $colorClass = 'bg-green'; // Hijau untuk kemiripan di  
        antara 0% dan 25.9%
```

```
    }
```

---

---

```
// Membagi judul menjadi kata-kata
```

```
$userTitleWords = explode(' ', strtolower($userTitle));
```

```
$judulSkripsiWords = explode(' ',  
strtolower($nama_judul_skripsi));
```

```
// Menyimpan kata-kata yang sama
```

```
$matchingWords = array_intersect($userTitleWords,  
$judulSkripsiWords);
```

```
// Menyusun ulang judul dengan kata-kata yang sama  
diberi latar belakang warna
```

```
$highlightedTitle = "";
```

```
foreach ($judulSkripsiWords as $word) {
```

```
    if (in_array($word, $userTitleWords)) {
```

```
        $highlightedTitle .= "<span  
class='highlight'>$word</span>";
```

```
    } else {
```

```
        $highlightedTitle .= "$word ";
```

```
    }
```

```
}
```

```
?>
```

```
<tr>
```

```
    <td><?= $no++ ?></td>
```

```
    <td><?= $nama_mahasiswa ?></td>
```



---

```
<td><?= $nim ?></td>
```

```
<td><?= rtrim($highlightedTitle) ?></td>
```

```
<td>
```

```
    <span class="<?= $colorClass ?> font-weight-  
bold"><?= number_format($similarity, 2) ?> %</span>
```

```
</td>
```

```
</tr>
```

```
<?php } ?>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
<?php $this->load->view('template/footer'); ?>
```

---

Pada tabel 4.4 menjelaskan proses algoritma berfungsi pada setiap menginput data berupa judul baru. Pengujian tersebut berlaku pada setiap data yang juga akan di input pada web aplikasi. Hasil dari pengujian white box memenuhi hasil yang diharapkan.