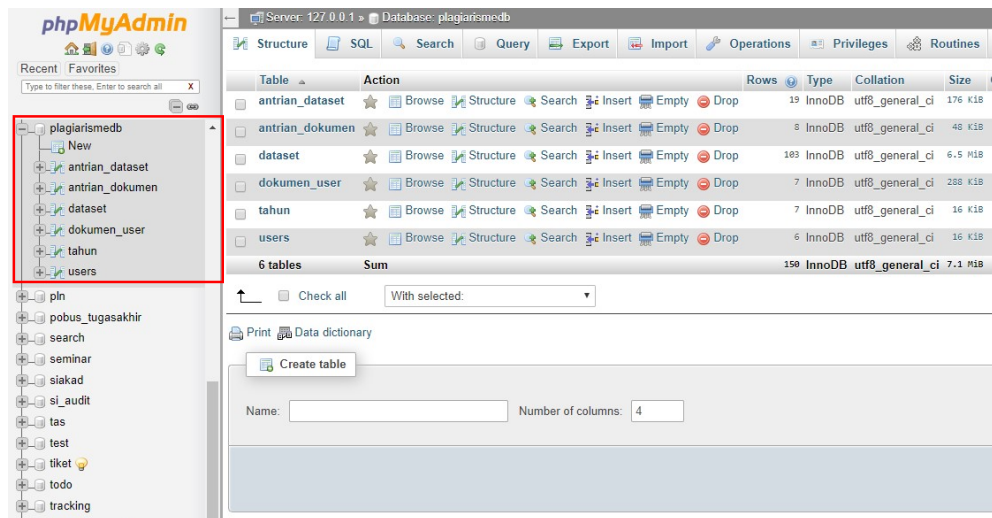


BAB V. IMPLEMENTASI DAN PENGUJIAN

Setelah dilakukan tahapan perancangan aplikasi, maka pada bab ini akan membahas mengenai proses implementasi basis data, implementasi sistem dan implementasi antar muka untuk merealisasikan perancangan menjadi nyata. Implementasi basis data dibangun menggunakan MySQL. Implementasi sistem terbagi menjadi dua, yaitu bahasa pemrograman Python digunakan sebagai fitur deteksi menggunakan LCS, dan bahasa pemrograman PHP dengan menggunakan *framework* Cogeigniter digunakan sebagai aplikasi berbasis *website* untuk pengguna mahasiswa dan panitia. Implementasi antar muka dibangun menggunakan bahasa pemrograman css dan jquery dengan menggunakan *framework* Bootstrap.

5.1 Implementasi Basis Data

Implementasi basis data merupakan hasil penerapan dari perancangan basis data yang telah dilakukan sebelumnya pada bab IV yang ditunjukkan pada gambar 4.8 yaitu skema basis data. Gambar 5.1 merupakan hasil dari penerapan asis data yang dibangun dengan menggunakan MySQL.



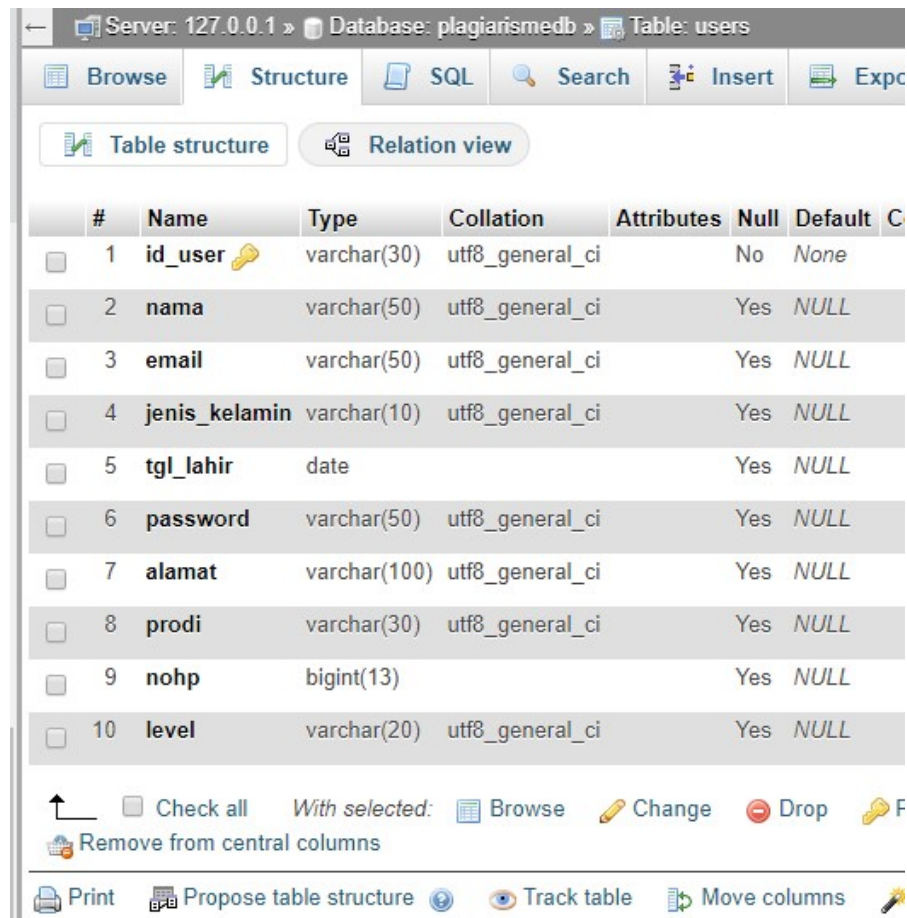
Gambar 5. 1 Basis Data Aplikasi

Basis data penelitian ini memiliki 6 tabel, yaitu tabel users, tabel tahun, tabel *dataset*, tabel dokumen_user, tabel antrian_dokumen, dan yang terakhir tabel

antrian_dataset. Implementasi dari tabel-tabel pada gambar 5.1 adalah sebagai berikut :

5.1.1 Gambar Tabel Users

Pada gambar 5.2 menunjukkan implementasi dari rancangan tabel users yang berfungsi untuk menyimpan data mahasiswa dan panitia. Tabel ini memiliki sebanyak 10 *field* dengan 1 *primary key* antara lain : id_user (*primary key*), nama, email, jenis_kelamin, tgl_lahir, password, alamat, prodi, nohp, level.



#	Name	Type	Collation	Attributes	Null	Default	C
<input type="checkbox"/> 1	id_user	varchar(30)	utf8_general_ci		No	None	
<input type="checkbox"/> 2	nama	varchar(50)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/> 3	email	varchar(50)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/> 4	jenis_kelamin	varchar(10)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/> 5	tgl_lahir	date			Yes	NULL	
<input type="checkbox"/> 6	password	varchar(50)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/> 7	alamat	varchar(100)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/> 8	prodi	varchar(30)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/> 9	nohp	bigint(13)			Yes	NULL	
<input type="checkbox"/> 10	level	varchar(20)	utf8_general_ci		Yes	NULL	

Gambar 5. 2 Tabel Users

5.1.2 Gambar Tabel Tahun

Pada gambar 5.3 menunjukkan implementasi dari rancangan tabel tahun yang berfungsi untuk menyimpan data tahun. Tabel ini memiliki sebanyak 3 *field* dengan memiliki 1 *primary key* antara lain : id_tahun (*primary key*), tahun, status.

Server: 127.0.0.1 » Database: plagiansmedb » Table: tahun

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_tahun	int(11)			No	None		AUTO_INCREMENT	Chan
2	tahun	int(4)			Yes	NULL			Chan
3	status	tinyint(1)			No	None			Chan

Check all | With selected: Browse | Change | Drop | Primary | Unique | Index

Remove from central columns

Print | Propose table structure | Track table | Move columns | Improve table structure

Gambar 5. 3 Tabel Tahun

5.1.3 Gambar Tabel Antrian_dataset

Pada gambar 5.4 menunjukkan implementasi dari rancangan tabel tahun yang berfungsi untuk menyimpan data antrian *dataset*, data antrian yang disimpan merupakan data-data yang diambil dari *dataset* yang dipilih sebagai kandidat pembanding. Tabel ini memiliki sebanyak 8 *field* dengan memiliki 1 *primary key* dan 2 *foreign key* antara lain : *id_antrian_dataset* (*primary key*), *id_antrian_dokumen* (*foreign key*), *id_data* (*foreign key*), *similarity_angka*, *similarity_text*, *status*, *created_at*, *updated_at*.

Server: 127.0.0.1 » Database: plagiansmedb » Table: antrian_dataset

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_antrian_dataset	int(11)			No	None		AUTO_INCREMENT	C
2	id_antrian_dokumen	int(11)			No	None			C
3	id_data	varchar(50)	utf8_general_ci		No	None			C
4	similarity_angka	float			Yes	NULL			C
5	similarity_text	text	utf8_general_ci		Yes	NULL			C
6	status	tinyint(1)			No	None			C
7	created_at	datetime			Yes	NULL			C
8	updated_at	datetime			Yes	NULL			C

Check all | With selected: Browse | Change | Drop | Primary | Unique | Index | Add to

Gambar 5. 4 Tabel Antrian Dataset

5.1.4 Gambar Tabel *Dataset*

Pada gambar 5.5 menunjukkan implementasi dari rancangan tabel *dataset* yang berfungsi untuk menyimpan *dataset* berupa dokumen skripsi dan tugas akhir terdahulu. Tabel ini memiliki sebanyak 11 *field* dengan memiliki 1 *primary key* dan 1 *foreign key* antara lain : *id_data* (*primary key*), *judul*, *tahun*, *keyword*, *prodi*, *ringkasan*, *penelitian*, *file*, *created_at*, *updated_at*, *id_user* (*foreign key*).

#	Name	Type	Collation	Attributes	Null	Default
1	id_data	varchar(50)	utf8_general_ci		No	None
2	judul	varchar(200)	utf8_general_ci		Yes	NULL
3	tahun	int(4)			Yes	NULL
4	keyword	varchar(200)	utf8_general_ci		Yes	NULL
5	prodi	varchar(30)	utf8_general_ci		Yes	NULL
6	ringkasan	text	utf8_general_ci		Yes	NULL
7	penelitian	text	utf8_general_ci		Yes	NULL
8	file	varchar(45)	utf8_general_ci		Yes	NULL
9	created_at	datetime			Yes	NULL
10	updated_at	datetime			Yes	NULL
11	id_user	varchar(30)	utf8_general_ci		No	None

Gambar 5. 5 Tabel *Dataset*

5.1.5 Gambar Tabel *Dokumen_user*

Pada gambar 5.6 menunjukkan implementasi dari rancangan tabel *dokumen_user* yang berfungsi untuk menyimpan data dokumen user yang akan diuji yang berupa dokumen skripsi atau tugas akhir. Tabel ini memiliki sebanyak

11 *field* dengan memiliki 1 *primary key* dan 1 *foreign key* antara lain : *id_data* (*primary key*), *judul*, *tahun*, *keyword*, *prodi*, *ringkasan*, *penelitian*, *file*, *created_at*, *updated_at*, *id_user* (*foreign key*).

Server: 127.0.0.1 » Database: plagiarismedb » Table: dokumen_user

Table structure

#	Name	Type	Collation	Attributes	Null	Defau
1	id_dokumen_user	varchar(50)	utf8_general_ci		No	None
2	judul	varchar(200)	utf8_general_ci		Yes	NULL
3	tahun	int(4)			Yes	NULL
4	keyword	varchar(200)	utf8_general_ci		Yes	NULL
5	prodi	varchar(30)	utf8_general_ci		Yes	NULL
6	ringkasan	text	utf8_general_ci		Yes	NULL
7	penelitian	text	utf8_general_ci		Yes	NULL
8	file	varchar(45)	utf8_general_ci		Yes	NULL
9	created_at	datetime			Yes	NULL
10	updated_at	datetime			Yes	NULL
11	id_user	varchar(30)	utf8_general_ci		No	None

Check all With selected: Browse Change Drop Print
Remove from central columns

Print Propose table structure Track table Move columns Im

Gambar 5. 6 Tabel Dokumen User

5.1.6 Gambar Tabel Antrian_dokumen

Pada gambar 5.7 menunjukkan implementasi dari rancangan tabel tahun yang berfungsi untuk menyimpan data antrian dokumen, data antrian yang disimpan merupakan data mahasiswa yang dipilih untuk dilakukan pengujian. Tabel ini memiliki sebanyak 6 *field* dengan memiliki 1 *primary key* dan 2 *foreign key* antara lain : *id_antrian_dokumen* (*primary key*), *id_dokumen_user* (*foreign key*), *id_user* (*foreign key*), *status*, *created_at*, *updated_at*.

Server: 127.0.0.1 » Database: plagiarismedb » Table: antrian_dokumen

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_antrian_dokumen	int(11)			No	None		AUTO_INCREMENT	Change
2	id_dokumen_user	varchar(50)	utf8_general_ci		No	None			Change
3	id_user	varchar(30)	utf8_general_ci		No	None			Change
4	status	tinyint(1)			No	None			Change
5	created_at	datetime			Yes	NULL			Change
6	updated_at	datetime			Yes	NULL			Change

Check all | With selected: Browse | Change | Drop | Primary | Unique | Index | Add to c...

Remove from central columns

Print | Propose table structure | Track table | Move columns | Improve table structure

Gambar 5. 7 Tabel Antrian Dokumen

5.2 Implementasi Sistem

Implementasi sistem merupakan proses pengubahan sistem dari perancangan diubah menjadi aplikasi nyata yang diterapkan pada program. Berikut ini akan dijelaskan bagaimana implementasi dari perancangan sistem kedalam aplikasi deteksi plagiarisme pada dokumen skripsi berdasarkan tingkat kesamaan dengan menggunakan metode *Longest Common Subsequence*.

5.2.1 Implementasi *Convert Dokumen ke Text*

Berikut ini dijelaskan mengenai implementasi *library* DocxToTextConversion untuk mengkonversi file word ke teks dan pdf2text untuk mengkonversi *file* pdf ke teks pada sistem dalam bentuk potongan-potongan listing program.

Sebagian besar kode program untuk konversi dokumen ke teks dilakukan oleh *library* PHP yang terdapat pada folder *libraries* dengan nama *file* pdf2text.php untuk konversi dari pdf ke teks dan pada folder *controller/member* dengan nama *file* Dokumen.php untuk konversi dari file doc/docx ke teks ini juga berlaku pada folder *controller/admin* dengan nama *file* Dataset.php. Pertama-tama ketika *user* melakukan *submit* terhadap *form* maka dilakukan pengecekan terhadap *form input file* jika tidak diisi dengan *file* maka proses *input* dibatalkan, jika terisi dengan *file* maka akan dilanjutkan pengambilan nama *file*.

```
//uploading File
    if(!$this->upload->do_upload('filedoc'))
    {
        $this->session->set_flashdata('error_message', $this->upload->display_errors());
        redirect('dokumen/upload', 'refresh');
    }
    else
    {
        $data['file'] = $this->upload->data()['file_name'];
```

Nama *file* digunakan untuk pengambilan kembali *file* yang telah tersimpan pada folder proyek (*assets/uploads/doc*), ketika *file* telah didapatkan dilakukan pembacaan *path info* untuk mengetahui dan mengambil *extension*. Setelah didapatkan *file extension* dilakukan pengecekan apakah *file* tersebut berupa pdf atau doc/docx, selanjutnya *file* akan diolah oleh setiap *library* yang sesuai dengan *file extension*, kemudian *library* akan mengembalikan hasil berupa *text* yang kemudian disimpan ke *database*.

```
// convert file to text
    $file = './assets/uploads/doc/'.$data['file'];

    $fileInformation = pathinfo($file);
    $extension = $fileInformation['extension'];
    //convert pdf to text
    if ($extension == 'pdf') {
        $converterpdf = new PDF2Text();
        $converterpdf->setFilename($file);
        $converterpdf->decodePDF();
        $data['penelitian'] = $converterpdf->output();
        //convert word to text
    }elseif ($extension == 'doc' || $extension == 'docx')
    {
        $converter = new DocxToTextConversion($file);
```

```

        $data['penelitian'] = $converter->convertToText();

        } else {

            return 'Invalid File Type, please use pdf, doc or docx
word document file.';

        }

        $this->session->set_flashdata('success_message',
'Tambah dokumen berhasil');

        $this->Dokumen_user_model->insert($data);

        echo "
<script
type='text/javascript'>window.location.href='http://localhost/fr
ontendskripsi/dokumen';</script>";

        // redirect('dokumen','refresh');

    }

```

5.2.2 Implementasi *Filter* Pada Kandidat Pembanding

Berikut ini dijelaskan mengenai implementasi *filter* yang dilakukan ketika pengambilan *dataset* dari *database* berdasarkan kata kunci pada dokumen uji. Kode program ini hanya diimplementasikan pada level mahasiswa, yang terletak pada folder *controller/member* dengan nama *file* Tesdokumen.php.

Kode program *filter* kandidat pembanding terdapat pada fungsi *prosesFilter()* yang terdapat pada *file* Tesdokumen.php. Pertama-tama fungsi ini akan bekerja ketika pada *form input* dilakukan aksi *submit*.

```

public function prosesFilter()
{
    if ($this->input->post('submit')) {

        $data_uji = array();

        $id = $this->input->post('id_dokumen_user');

        $result = $this->Dokumen_user_model->select_by_id($id);

        $data['dokumenuji'] = $result;

        $result_keyword = explode(",", $result[0]['keyword']);
        //ambil keyword
    }
}

```


Pada inisialisasi *array* dengan nama variabel `$data_uji` digunakan untuk menyimpan hasil dari proses *filter* sebelum disimpan kedalam *database*. Dilakukan pengambilan id dokumen *user* dari *form input*, id ini digunakan untuk mengambil data lengkap dari *database* berdasarkan id dokumen *user* yang telah didapatkan, ketika mendapatkan data lengkap dimasukkan pada variabel *array* dengan nama *dokumenuji*. Selanjutnya pengambilan kata kunci untuk dilakukan pemisahan berdasarkan karakter koma (,) dengan menggunakan fungsi *explode* sehingga akan menghasilkan data berupa *array*, seperti contoh pada tabel 5.1

Tabel 5. 1 Contoh Pemisahan Kata Kunci Berdasarkan Koma

Contoh pemisahan kata kunci berdasarkan koma (,)	
Bentuk awal	Sistem,Penelitian,Deteksi,Dokumen
Bentuk Akhir	<pre>array(4) { [0] => string(6) "Sistem" [1] => string(10) "Penelitian" [2] => string(7) "Deteksi" [3] => string(7) "Dokumen" }</pre>

Pada proses selanjutnya yaitu perulangan berdasarkan jumlah *array* kata kunci, dari setiap *array* akan diambilkan data berdasarkan isi dari setiap *array*. Karna hasil dari pengambilan *dataset* sangat mungkin untuk memiliki lebih dari satu data, maka dilakukan perulangan untuk mengambil disetiap datanya untuk dimasukkan kedalam *array* baru.

```
for ($i=0; $i < count($result_keyword) ; $i++) {

    $dataset = $this->Dataset_model->dataset_by_tags($result_keyword[$i]); //pilih dataset sesuai keyword

    // cek jika $dataset lebih dari 1
    foreach ($dataset as $key) {
        $data_uji[] = $key;
    }

    $data['datauji'] = array_unique($data_uji, SORT_REGULAR);
    //convert ke array of string
    $data['content'] = $this->load->view('member/tesdokumen/datauji', $data, TRUE);
    $data['desc'] = "Data Terpilih";
    $this->load->view('member/template', $data);
}
```

Pada *array* baru dengan nama *\$data_uji* memungkinkan untuk memiliki isi yang sama, karna ketika pengambilan berdasarkan tiap kata kunci bisa terjadi pengambilan *dataset* yang sama dengan kata kunci yang sama pula. Sehingga diperlukan pembuangan data *array* yang memiliki *value* sama.

5.2.3 Implementasi Metode LCS Pada Python

Berikut ini dijelaskan mengenai implementasi metode *Longest Common Subsequence* yang dilakukan ketika menguji dokumen untuk dibandingkan antara dokumen uji dengan kandidat pembanding dari *database*, untuk data yang diambil berdasarkan kata kunci pada dokumen uji. Kode program ini terletak pada folder *assets/python* dengan nama *file setup.py* dan *lcs.py*.

Dalam penerapannya, kode program dari metode LCS akan bekerja ketika mendapatkan *request* dari *website* untuk melakukan pengujian. Karna pada metode LCS dibangun menggunakan bahasa pemrograman *python* sehingga diperlukan pemanggilan dan kirim data secara khusus, seperti yang ditunjukkan pada kode program berikut :

```
public function executeCheker()
{
    if ($this->input->post('submit')) {
        $data1['id_dokumen_user'] = $this->input->post('id_dokumen_user');
        $data1['id_user'] = $this->session->userdata('logged_in')['id_user'];
        $data1['created_at'] = date('Y-m-d H:i:s');
        $data1['status'] = 1;

        $id_data = $this->input->post('id_data');

        $id = $this->Antrian_dokumen_model->insert($data1);
    }
}
```

Ketika tombol uji ditekan, maka akan dijalankan fungsi yang bernama *executeCheker()*, fungsi ini akan menerima data dari *form input* yang berisikan id dokumen dan id user yang akan disimpan ke *database* dengan nama tabel *antrian_dokumen* melalui model *Antrian_dokumen_model*, untuk menerima *input* berisi kandidat pembanding dari *form input* yang berisikan kumpulan id kandidat pembanding. Kumpulan dari id kandidat pembanding ini akan disimpan pada tabel *antrian_dataset*. Untuk dapat menyimpan kumpulan id *dataset* pada tabel *antrian_dataset* diperlukan penjabaran nama pada setiap id agar sesuai. Berikut ini

kode program untuk pemberian nama sesuai dengan nama kolom pada tabel *antrian_dataset*.

```
$data = array();
$path = './assets/python/setup.py';

foreach ($id_data as $value) {
    array_push($data, array(
        'id_data' => $value,
        'id_antrian_dokumen' => $id,
        'created_at' => date('Y-m-d H:i:s')
    ));
}

$this->Antrian_dataset_model->insert_batch($data);
```

Ketika pemberian nama selesai, data akan disimpan ke *database* melalui model *Antrian_dataset_model*. Sehingga dapat melanjutkan pada proses berikutnya yaitu pemanggilan kode program LCS.

```
$output = passthru("python ".$path."
".$data['id_dokumen_user']." ".$id); //array [0] = $path, array
[1] = id_dokumen_user, array [2] = $value (data yg di uji)
    redirect('tes-plagiasi');
} else {
    redirect('tes-plagiasi/uji');
}
}
```

Pemanggilan dilakukan secara khusus karna bahasa pemrograman yang digunakan menggunakan bahasa pemrograman *python*, untuk memanggil file *python* dilakukan menggunakan fungsi *passthru()*. Pada fungsi ini diwajibkan untuk menyebutkan letak dari *file python*, dan untuk dua parameter berikutnya diisi jika memang diperlukan untuk mengirim data pada *file python*. Disini penulis mengisi dua parameter yang berisi id dokumen uji dan id antrian dokumen.

File yang pertama kali menerima *request* dari *website* adalah *setup.py*, dimana pada *file* ini berisi kode program untuk mengambil data dari *database* berdasarkan id dokumen uji dan dokumen kandidat pembanding berdasarkan id antrian dokumen yang dikirim dari *website* ketika *request* dilakukan.

```
import sys
import MySQLdb
import datetime
from lcs import LCS
```

```

class Database:

    host      = "localhost"
    user      = "root"
    password  = ""
    db        = "plagiarismdb"

    def __init__(self):
        self.connection = MySQLdb.connect( host = self.host,
                                             user = self.user,
                                             password = self.password,
                                             db = self.db ,
                                             autocommit=True)

    def query(self, q):
        cursor = self.connection.cursor()
        cursor.execute(q)
        self.connection.commit()
        cursor.close()
        #self.connection.close()

    def queryall(self, q, va):
        cursor = MySQLdb.cursors.DictCursor )
        cursor.execute(q, va)
        return cursor.fetchall()

    def queryv(self, q, va):
        cursor = MySQLdb.cursors.DictCursor )
        cursor.execute(q, va)
        return cursor.fetchone()

    def __del__(self):
        self.connection.close()

```

Kode program diatas merupakan fungsi-fungsi yang digunakan untuk mengakses *database*. Untuk mengoprasikan pengambilan data serta menjalankan fungsi metode LCS diterapkan pada fungsi dibawah ini.

```

if __name__ == "__main__":
    db = Database()

    id_doc_uji = sys.argv[1] #dokumen yg di uji
    id_antrian = sys.argv[2] #ambil id antrian
    doc_uji = (id_doc_uji,) #harus ada koma agar tidak eror
    antrian = (id_antrian,)
    q = "SELECT * FROM dokumen_user WHERE id_dokumen_user = %s"
    data = db.queryv(q, doc_uji)

    q = "SELECT * FROM antrian_dataset as ad join dataset as ds
on ad.id_data = ds.id_data WHERE id_antrian_dokumen = %s"
    data1 = db.queryall(q, antrian)

```

Pertama-tama yang dilakukan adalah inisialisasi *database*, kemudian membuat variabel yang digunakan untuk menampung data yang dikirim dari website. Kemudian melakukan *query* ke *database* pada tabel dokumen_user dan antrian_dataset. Data yang didapatkan dari *query* diproses dalam perulangan berikut ini.

```
for td in data1:
    list_sentence = ""
    angka_sentence = 0

    penelitian = data['penelitian']
    while True:

        common_sentence, persent =
LCS.longest_common_sentence(penelitian, td['penelitian'])
        common_words = common_sentence.split(' ')

        #berhenti ketika hasil plagiat kurang dari 3 kata
        if(len(common_words) < 3):
            break
        else:
            list_sentence += '%s###' % common_sentence
#mengumpulkan hasil lcs ke 1 variable
            angka_sentence += float(persent)

            buang_plagiarisme =
penelitian.split(common_sentence) #membuang kalimat plagiat pada
text uji
            penelitian = ''.join(buang_plagiarisme)
#menggabungkan kembali setelah membuang kalimat plagiat

            q = "UPDATE antrian_dataset SET similarity_angka='%s',
similarity_text='%s', status=%d, updated_at='%s' WHERE
id_antrian_dataset='%s' "% (angka_sentence, list_sentence, 1,
datetime.datetime.now(), td['id_antrian_dataset'])
            db.query(q)
```

Perulangan pertama berisi dari banyaknya kandidat pembanding yang diuji, pada perulangan kedua dilakukan untuk menguji pada setiap kandidat pembanding hingga tidak ditemukan lagi kalimat yang sama dengan batas minimal tiga kata dalam satu kalimat. Selanjutnya memanggil fungsi `longest_common_sentence()` dari kelas LCS, fungsi tersebut membutuhkan dua parameter yang berisi teks dari dokumen uji dan teks dari kandidat pembanding, proses ini akan menghasilkan dua output data sekaligus yaitu kalimat yang mengandung plagiarisme dan nilai plagiarisme (dalam persen) disetiap pengujian pada masing-masing kandidat pembanding.

Kondisi berikutnya untuk menentukan jika kalimat yang mengandung plagiarisme berjumlah kurang dari tiga kata, maka perulangan akan dihentikan, namun apabila lebih dari atau sama dengan tiga kata, maka perulangan masih terus dijalankan. Setelah didapatkan hasil secara keseluruhan selanjutnya akan dikirim kembali ke database untuk disimpan kembali.

Setelah pembahasan kode program pada *file setup.py*, pembahasan selanjutnya adalah kode program LCS yang terdapat pada file *lcs.py* dengan nama fungsi `longest_common_substring()`. Disinilah metode lcs diterapkan, fungsi ini akan bekerja dan menghasilkan deretan *array* kata dalam satu kalimat panjang.

```
def longest_common_substring(s1, s2):
    m = [[0] * (1 + len(s2)) for i in range(1 + len(s1))]
    longest, x_longest = 0, 0

    for x in range(1, 1 + len(s1)): #ambil per kata dari doc
1, dan x sebagai doc terpanjang
        for y in range(1, 1 + len(s2)): #ambil per kata dari
doc 2
            if s1[x - 1] == s2[y - 1]: #cek juka kata dari doc 1
== doc 2
                m[x][y] = m[x - 1][y - 1] + 1
                if m[x][y] > longest:
                    longest = m[x][y]
                    x_longest = x
            else:
                m[x][y] = 0
    return s1[x_longest - longest: x_longest] #memilih lcs
terpanjang
```

Untuk memaksimalkan fungsi `longest_common_substring()` dalam mencari kalimat, sebelumnya diperlukan penentuan untuk mengetahui antara dokumen uji dan kandidat pembanding manakah yang memiliki kata paling Panjang. Pada fungsi dibawah ini akan dilakukan pengecekan panjang kata, kata yang paling panjang akan disimpan pada variabel `s1_arr` dan kata yang lebih pendek akan disimpan pada variabel `s2_arr`. Fungsi ini menghasilkan deretan kalimat panjang serta nilai plagiarisme disetiap satu kali fungsi ini dijalankan.

```
longest_common_sentence(s1, s2):
    s1_arr = []
    s2_arr = []
    s1_words = s1.split(' ') #memisah tiap kata menjadi
array
```



```

s2_words = s2.split(' ')

lg_s1 = len(s1_words) #menghitung panjang
lg_s2 = len(s2_words)

if lg_s1 < lg_s2: #menentukan dokumen terpanjang
    s1_arr = s2_words
    s2_arr = s1_words
else:
    s1_arr = s1_words
    s2_arr = s2_words

lcs_arr = LCS.longest_common_substring(s1_arr, s2_arr)
s1_pesent = (len(lcs_arr)*100)/len(s1_arr)

lcs = ' '.join(lcs_arr) #menggabungkan array hasil lcs
return lcs, s1_pesent

```

5.2.4 Implementasi *Output* dan *Ranking* Pada Hasil Uji Plagiarisme

Berikut ini dijelaskan mengenai implementasi *Output* hasil pengujian yang diambil dari *database* berdasarkan id dokumen yang diuji. Kode program ini terletak pada folder *controller/member* dengan nama *file* Tesdokumen.php dan pada folder *view/member/tesdokumen* dengan nama *file* detailhasil.php.

Pada *controller* Tesdokumen.php pada fungsi detail() dijelaskan pada kode program berikut ini.

```

public function detail($id)
{
    $data['detail']          =          $this->Antrian_dokumen_model->select_detail($id);
    $data['dataset']         =          $this->Antrian_dataset_model->select_detail($id);

    $data['content']         =          $this->load->view('member/tesdokumen/detailhasil', $data, TRUE);
    $data['desc'] = "Laporan Hasil Deteksi";

    $this->load->view('member/template', $data);
}

```

Data yang akan ditampilkan diambil dari model `Antrian_dokumen_model` dan `Antrian_dataset_model` kemudian dikirimkan ke `view`. Pada `view` `detailhasil.php` terdapat kode program seperti berikut ini.

```
<?php
// menghitung hasil dari dataset
$totalkata = 0; //untuk menghitung total kata
$totalangka = 0; //untuk menghitung total angka
$angka = array();
foreach ($dataset as $value) {
    array_push($angka, $value['similarity_angka']);
    $pecah_kata = explode("###", $value['similarity_text']);
    //memisahkan kalimat berdasarkan karakter ###
    $hit_total_kata = 0;
    foreach ($pecah_kata as $value_kata) {
        $hitungkata = str_word_count($value_kata, 1); //1 untuk
        membuat array dan mengatur index array, [Ketika returnformat
        diatur ke 1, itu akan mengembalikan array], [Ketika returnformat
        diatur ke 2, itu akan mengembalikan array asosiatif]
        $hit_total_kata += count($hitungkata);
    }
    $totalkata += $hit_total_kata;
    $totalangka += $value['similarity_angka'];
}

$ordered_angka = $angka;
rsort($ordered_angka); //pembuatan ranking pada dataset nilai
'similarity_angka'
$dataset_ranking = array();

foreach ($ordered_angka as $key_angka => $value_angka) {
    foreach ($dataset as $key => $value) {
        if ($value['similarity_angka'] == $value_angka) {
            //penyesuaian dataset yang akan di tampilkan sesuai ranking
            tertinggi
            array_push($dataset_ranking, $value);
        }
    }
}
}
?>
```

Pada kode program diatas terdapat bagian kode untuk menerapkan *ranking* pada kandidat pembanding. *Ranking* digunakan sebagai penentu bahwa dengan kandidat mana yang memiliki nilai plagiarisme tertinggi dan yang terendah. Selain itu ada juga kode program yang digunakan untuk menampilkan *detail* dari setiap kandidat menggunakan bahasa pemrograman javascript.

```
<script type="text/javascript">
function onDatasetView(dataset) {
    document.getElementById("judul").innerHTML = dataset.judul;
    document.getElementById("angka").innerHTML =
    Math.round(dataset.similarity_angka)+'%';
    document.getElementById("tahun").innerHTML = dataset.tahun;
```

```

document.getElementById("prodi").innerHTML = dataset.prodi;
document.getElementById("waktu").innerHTML = dataset.created_at;

var pisah_hasil_text = dataset.similarity_text.split("###");
document.getElementById("text").innerHTML = "";
pisah_hasil_text.forEach(function (value) {
    if (value === "") {
    }else {
        document.getElementById("text").innerHTML += `
        <p class="hasil_text">${value}</p>
        `
    }
})

var keyword = dataset.keyword.split(",");
document.getElementById("keyword").innerHTML = "";
keyword.forEach(function (value) {
    document.getElementById("keyword").innerHTML += `
    <span class="label label-info">${value}</span>
    `
})
$('#ModalDataset').modal('show');
}
</script>

```

5.3 Implementasi Antar Muka

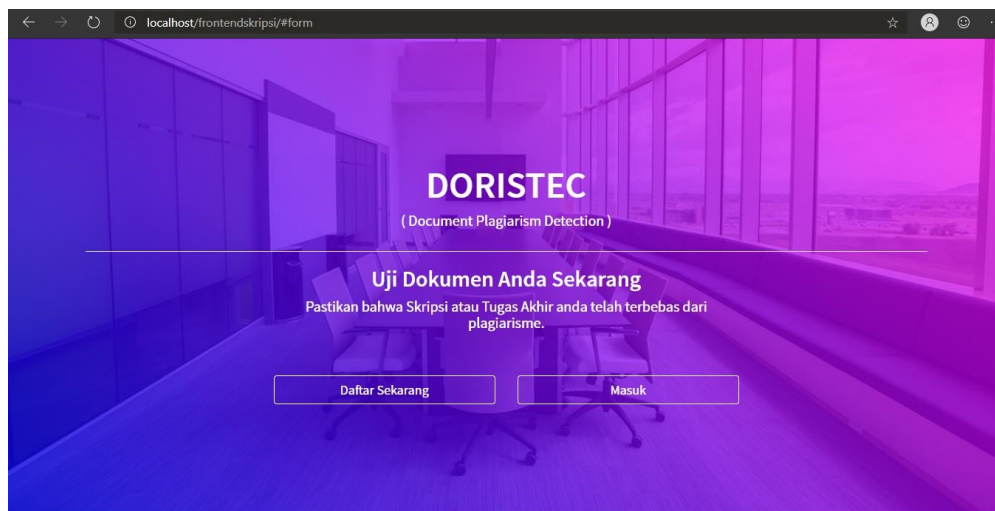
Implementasi antar muka merupakan proses pengubahan dari perancangan desain antar muka diubah menjadi tampilan aplikasi nyata yang diterapkan pada program. Berikut ini akan dijelaskan bagaimana implementasi dari perancangan desain antar muka kedalam aplikasi deteksi plagiarisme pada dokumen skripsi berdasarkan tingkat kesamaan dengan menggunakan *Longest Common Subsequence*.

5.3.1 Halaman Registrasi dan Login

Halaman Registrasi dan *Login* merupakan halaman utama pada *website* ketika pengguna mengunjungi *website* doristec ditunjukkan pada gambar 5.8, gambar 5.9, gambar 5.10, gambar 5.11, halaman ini memuat fitur-fitur yang berisikan informasi singkat seputar *website* doristec serta *form* Regidtrasi dan *form Login*. Halaman utama ini bersifat umum, sehingga bagi mahasiswa maupun panitia yang akan melakukan registrasi dan *login* melalui halaman utama ini. Untuk desain yang digunakan pada *website* ini memilih desain yang simple agar mudah digunakan. Pemilihan desain menggunakan *framework bootstrap custom*.

Pengguna yang belum memiliki akun pada *website* doristec, dihimbau untuk melakukan registrasi terlebih dahulu. Pengguna harus memasukkan nama lengkap, *email*, jenis kelamin, alamat, nomor handphone, program studi, password, hak akses. Data-data tersebut harus sesuai dengan identitas pengguna serta hak akses yang dibutuhkan.

Ketika telah mendapatkan akun, pengguna bisa melanjutkan untuk *login* kedalam sistem doristec. Untuk melakukan *login* Pengguna mahasiswa maupun panitia harus memasukkan *username* dan *password* untuk bisa mendapatkan hak akses.



Gambar 5. 8 Halaman Utama *Website*



Gambar 5. 9 Lanjutan Halaman Utama *Website*

Gambar 5. 10 Lanjutan Halaman Utama *Website*

© Imam Nawawi
 Powered by Codeigniter | Picture by Unsplash

Gambar 5. 11 Lanjutan Halaman Utama *Website*

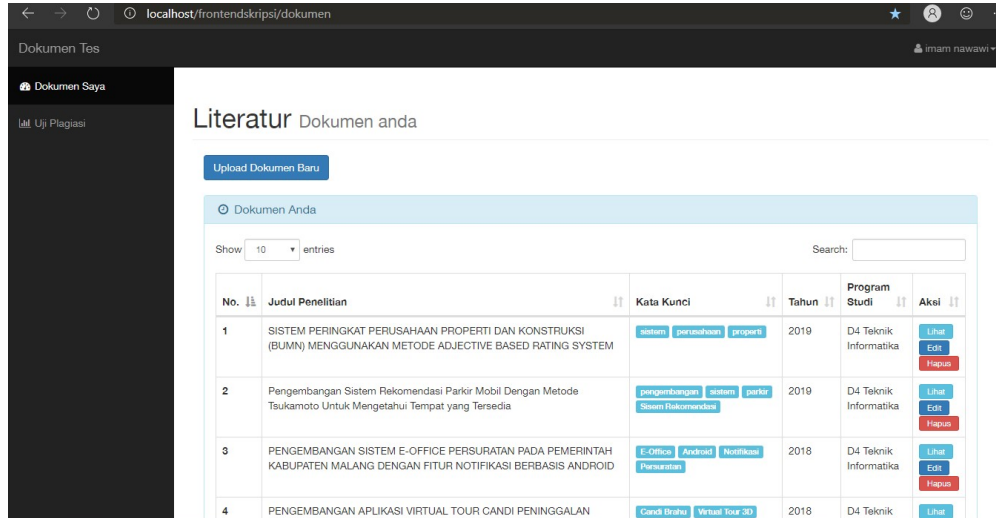
5.3.2 Halaman Mahasiswa

Halaman mahasiswa merupakan halaman admin yang ditujukan bagi pengguna mahasiswa, halaman ini memuat fitur-fitur yang berisikan daftar dokumen anda, upload dokumen baru, daftar dokumen yang pernah diuji, membuat pengujian baru, melihat hasil yang pernah diuji. Untuk desain yang digunakan pada *website* ini memilih desain yang simple agar mudah digunakan. Pemilihan desain menggunakan *theme startbootstrap-admin*.

5.3.2.1 Halaman Utama Admin Mahasiswa

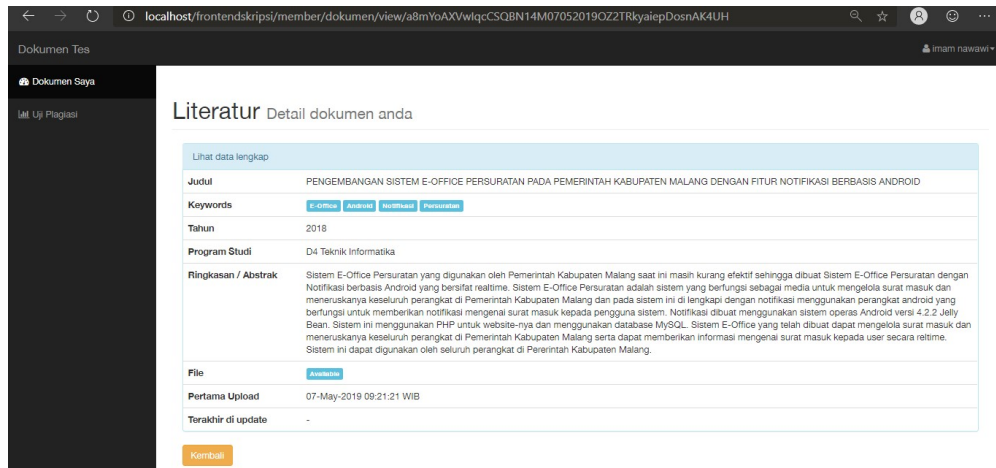
Halaman utama admin mahasiswa merupakan halaman awal yang ditampilkan dari ketika *login* berhasil dilakukan. Pada halaman ini terdapat fitur dokumen anda, *upload* dokumen baru. Pada saat halaman utama di buka mahasiswa akan diarahkan langsung menuju dokumen anda dimana fitur ini berfungsi untuk menampilkan semua data dari dokumen yang pernah di *upload* dan sudah tersimpan di *database*. Data tersebut antara lain judul penelitian, kata kunci, tahun dan program studi pada halaman ini mahasiswa mempunyai hak akses untuk mengolah data dokumen anda, pada halaman ini terdapat tiga tombol yaitu tombol lihat, edit dan hapus. Tombol lihat berfungsi untuk melihat secara keseluruhan data yang

terpilih, tombol edit berfungsi untuk mengedit data yang terpilih, sedangkan tombol hapus untuk menghapus data yang terpilih, untuk tampilan ditunjukkan pada gambar 5.12.



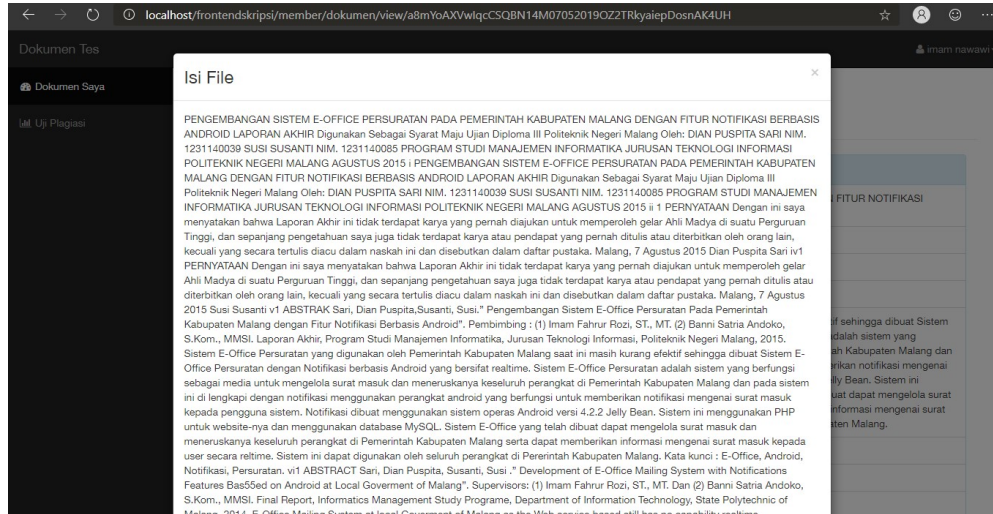
Gambar 5. 12 Halaman Utama Admin Mahasiswa

Apabila ingin melihat data secara lengkap, mahasiswa dapat menekan tombol lihat, maka akan ditampilkan data secara lengkap berdasarkan data yang dipilih seperti ditunjukkan pada gambar 5.13.



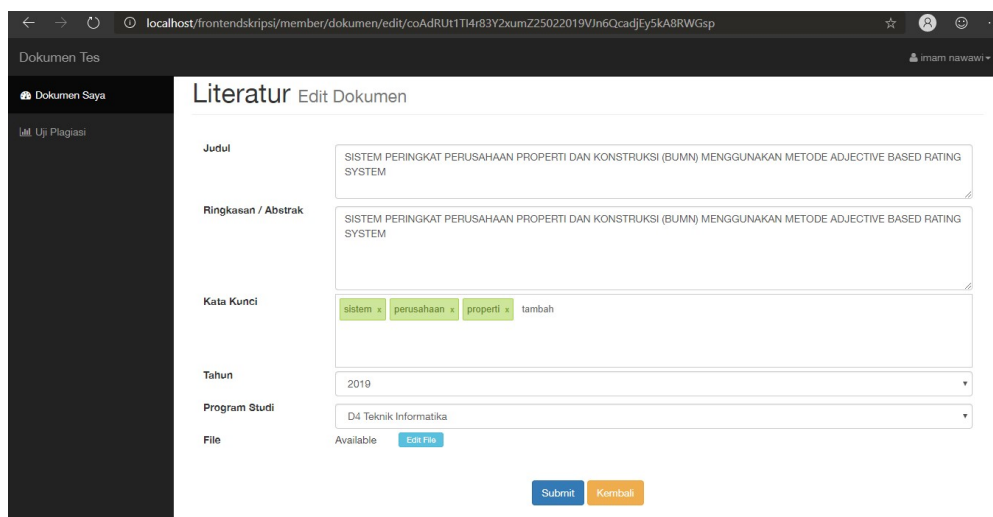
Gambar 5. 13 Halaman Lihat Data

Sedangkan untuk mengetahui isi teks secara keseluruhan, mahasiswa dapat menekan tombol *available* yang artinya teks tersedia dalam *database*, seperti yang ditunjukkan pada gambar 5.14.



Gambar 5. 14 Halaman Lihat teks keseluruhan

Ketika tombol edit ditekan, maka informasi yang akan dirubah akan tampil dihalaman input data seperti ditunjukkan gambar 5.13 namun dalam keadaan form yang sudah terisi. Mahasiswa cukup merubah data yang diinginkan saja tanpa perlu memasukkan semua informasi dari awal.



Gambar 5. 15 Halaman Edit Data

5.3.2.2 Halaman *Upload* Dokumen Baru

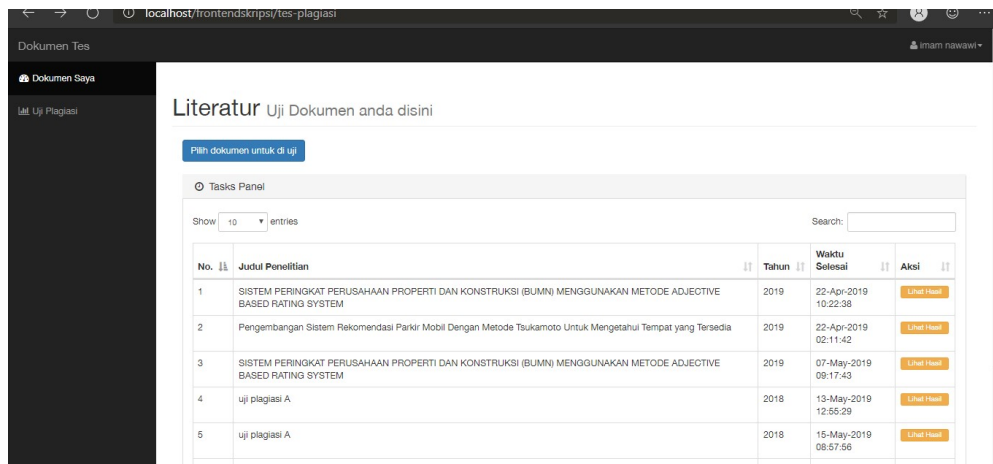
Halaman *upload* dokumen baru merupakan halaman yang berisi *form* untuk menambah dokumen baru yang ditunjukkan pada gambar 5.16. Pada halaman ini *form* upload memiliki komponen antara lain judul, abstrak, kata kunci, file yang ingin di *upload*, sedangkan untuk tahun terisi secara otomatis dari pengaturan panitia dan program studi terisi secara otomatis dari profil mahasiswa. Pada saat halaman *upload* dibuka, mahasiswa dapat mengisi *form* sesuai dengan data dokumen yang akan di upload, apabila *form* telah terisi secara keseluruhan mahasiswa dapat menekan tombol submit yang artinya data akan dikirim ke *database* untuk disimpan.

Gambar 5. 16 Halaman *Upload* Dokumen Baru

5.3.2.3 Halaman Uji Plagiasi

Halaman Uji Plagiasi merupakan halaman yang berisi data-data dokumen yang pernah dilakukan pengujian. Pada halaman ini terdapat fitur pilih dokumen untuk diuji, lihat hasil. Pada saat halaman ini di buka mahasiswa akan diarahkan langsung menuju daftar dokumen yang pernah diuji dimana fitur ini berfungsi untuk menampilkan semua data dari dokumen yang pernah diuji dan sudah tersimpan di *database*. Data tersebut antara lain judul peneliti, tahun dan waktu selesai, pada halaman ini terdapat dua tombol yaitu tombol pilih dokumen yang diuji dan lihat hasil seperti yang ditunjukkan pada gambar 5.17. Tombol pilih dokumen yang diuji berfungsi untuk melakukan tahap *filter* kandidat pembanding sebelum dilakukan

pengujian lebih lanjut, sedangkan tombol lihat berfungsi untuk melihat secara keseluruhan data yang terpilih.



No.	Judul Penelitian	Tahun	Waktu Solusi	Aksi
1	SISTEM PERINGKAT PERUSAHAAN PROPERTI DAN KONSTRUKSI (BUMIN) MENGGUNAKAN METODE ADJECTIVE BASED RATING SYSTEM	2019	22-Apr-2019 10:22:38	Lihat Hasil
2	Pengembangan Sistem Rekomendasi Parkir Mobil Dengan Metode Tsukamoto Untuk Mengetahui Tempat yang Tersedia	2019	22-Apr-2019 02:11:42	Lihat Hasil
3	SISTEM PERINGKAT PERUSAHAAN PROPERTI DAN KONSTRUKSI (BUMIN) MENGGUNAKAN METODE ADJECTIVE BASED RATING SYSTEM	2019	07-May-2019 09:17:43	Lihat Hasil
4	uji plagiasi A	2018	13-May-2019 12:55:29	Lihat Hasil
5	uji plagiasi A	2018	15-May-2019 08:57:56	Lihat Hasil

Gambar 5. 17 Halaman Uji Plagiasi

5.3.2.4 Halaman Pilih Dokumen Yang Diuji

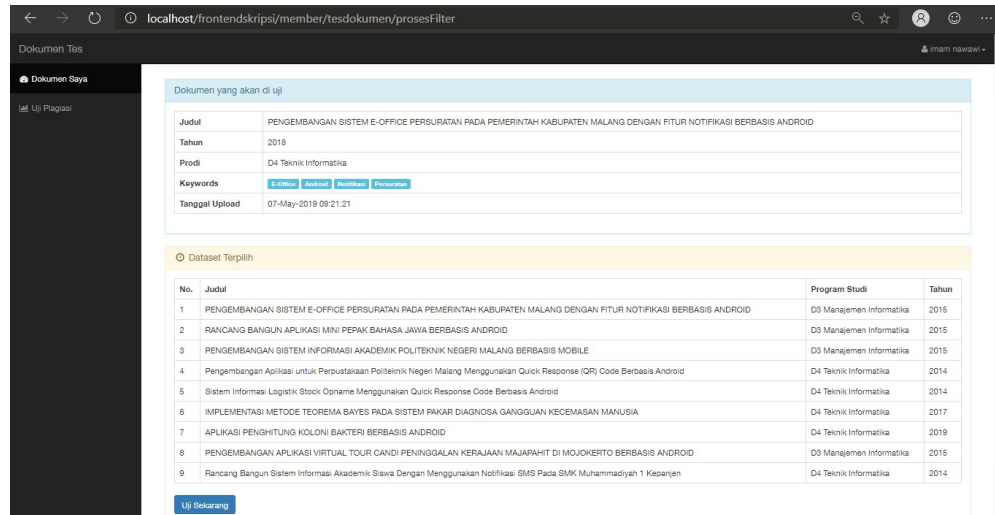
Halaman pilih dokumen yang diuji merupakan halaman yang berisi data-data dokumen mahasiswa yang tersimpan pada *database*. Pada halaman ini terdapat fitur pilihan dokumen untuk diuji, yang dapat dilihat pada gambar 5.18. ketika mahasiswa telah menentukan dokumen pilihan, kemudian menekan tombol proses untuk memulai dilakukan proses *filter* kandidat pembanding berdasarkan kata kunci pada dokumen yang diuji.



Gambar 5. 18 Halaman Pilih Dokumen Uji

5.3.2.5 Halaman Hasil *Filter* Kandidat Pembanding

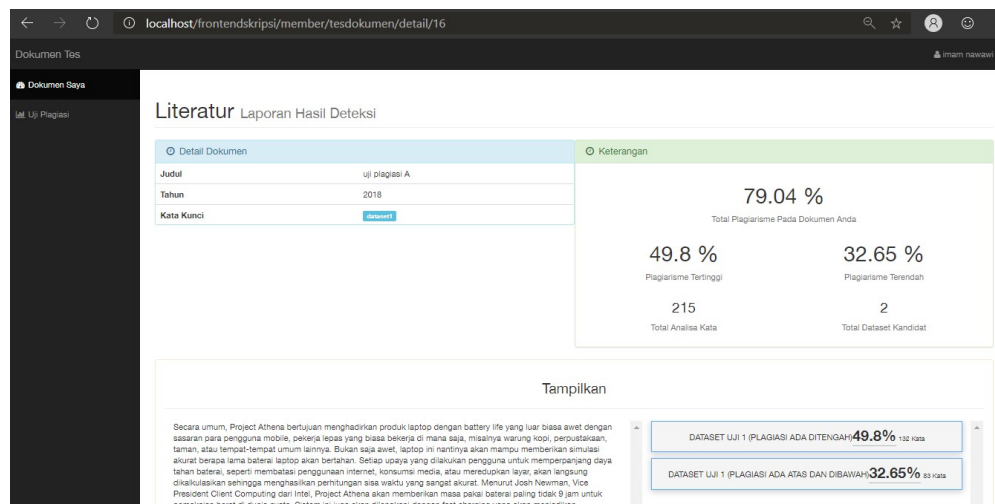
Halaman hasil *filter* kandidat pembanding merupakan halaman yang berisi data-data dari dokumen uji dan kandidat pembanding dari hasil proses *filter kandidat pembanding* yang ditunjukkan pada gambar 5.18. Pada halaman ini mahasiswa tekan tombol uji sekarang untuk memulai proses uji yang ditunjukkan pada gambar 5.19.



Gambar 5. 19 Halaman Hasil *Filter* Kandidat pembeding

5.3.2.6 Halaman Hasil Uji

Halaman hasil uji merupakan halaman yang berisi data-data dari hasil uji plagiarisme yang telah tersimpan pada *database* seperti yang ditunjukkan pada gambar 5.20. Pada halaman ini mahasiswa dapat mengetahui secara menyeluruh mengenai hasil dari dokumen yang telah diuji.



Gambar 5. 20 Halaman Hasil Uji

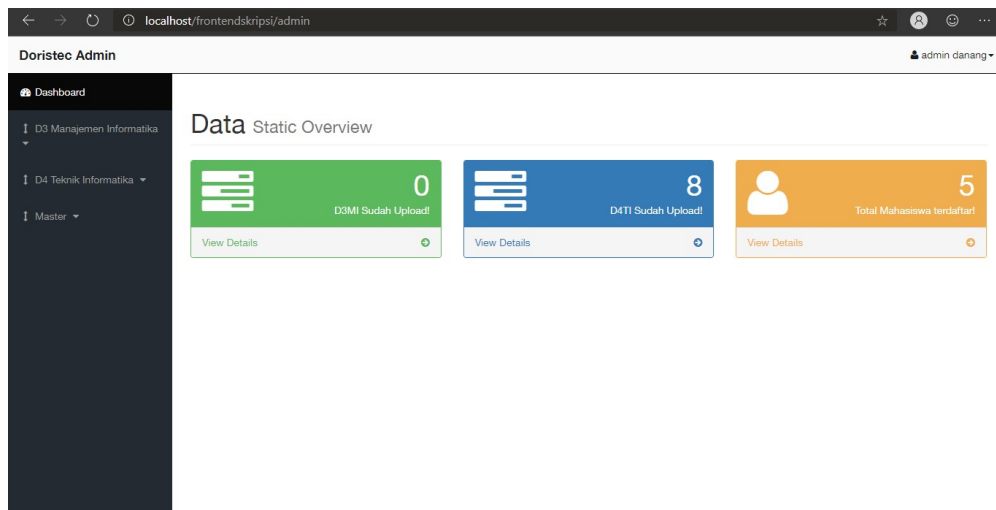
5.3.3 Halaman Panitia

Halaman panitia merupakan halaman admin yang ditujukan bagi pengguna panitia, halaman ini memuat fitur-fitur yang berisikan halaman *dashboard*, daftar mahasiswa D3 manajemen informatika, mahasiswa D3 yang sudah *upload*, daftar

mahasiswa D4 teknik informatika, mahasiswa D4 yang sudah *upload*, daftar *dataset*, dan tahun. Untuk desain yang digunakan pada *website* ini memilih desain yang simple agar mudah digunakan. Pemilihan desain menggunakan *theme startbootstrap-admin custom*.

5.3.3.1 Halaman *Dashboard*

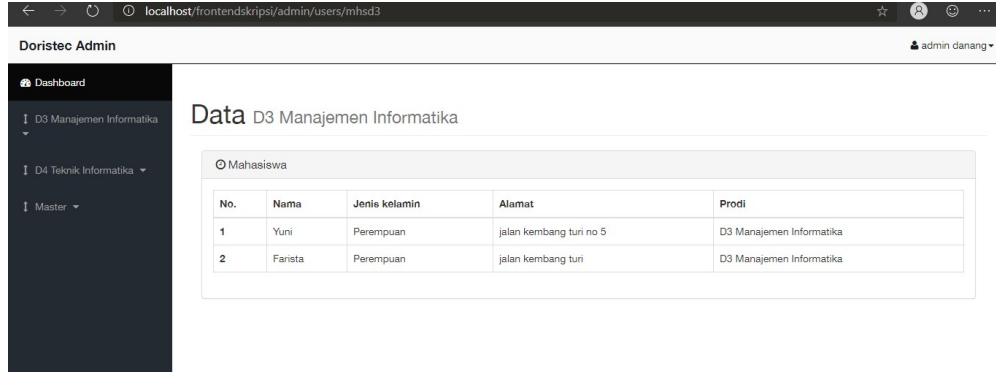
Halaman *dashboard* admin panitia merupakan halaman awal yang ditampilkan dari ketika *login* berhasil dilakukan. Pada halaman ini terdapat fitur mahasiswa D3 Manajemen Informatika, D3 yang sudah *upload* dokumen, mahasiswa D4 Teknik Informatika, D4 yang sudah *upload* dokumen, daftar *dataset*, *upload* dokumen pembanding baru dan kelola *dataset*, tahun, kelola tahun. Pada saat halaman *dashboard* dibuka panitia akan diarahkan langsung menuju halaman yang dimana fitur ini berfungsi untuk menampilkan semua data berupa grafik yang sudah tersimpan di *database* seperti ditunjukkan pada gambar 5.21.



Gambar 5. 21 Halaman *Dashboard* Panitia

5.3.3.2 Halaman Mahasiswa D3 Manajemen Informatika

Halaman mahasiswa D3 manajemen informatika merupakan halaman yang berisi data-data mahasiswa D3 yang telah mendaftar pada aplikasi doristec. Pada saat halaman ini di buka panitia akan diarahkan langsung menuju daftar mahasiswa D3 yang pernah mendaftar. Data tersebut antara lain nama, jenis kelamin, alamat, dan program studi seperti yang ditunjukkan pada gambar 5.22.



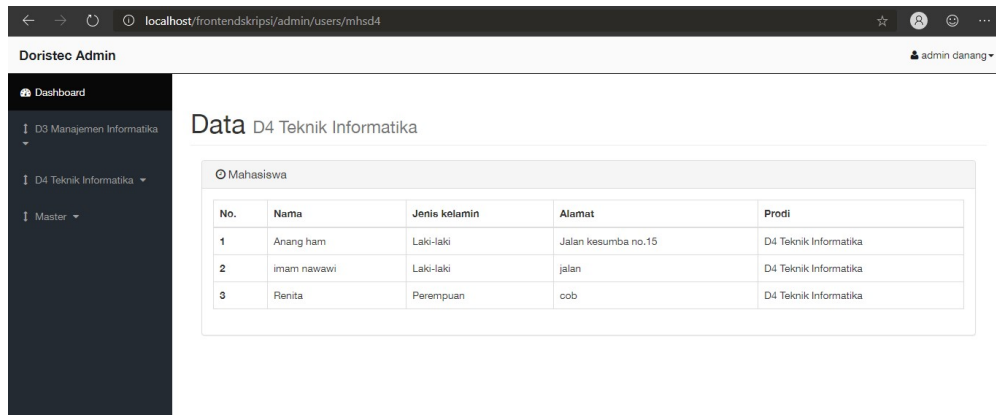
The screenshot shows a web browser at localhost/frontendskripsi/admin/users/mhsd3. The dashboard has a sidebar with 'Dashboard', 'D3 Manajemen Informatika', 'D4 Teknik Informatika', and 'Master'. The main content area is titled 'Data D3 Manajemen Informatika' and contains a table of students.

No.	Nama	Jenis kelamin	Alamat	Prodi
1	Yuni	Perempuan	jalan kembang turi no 5	D3 Manajemen Informatika
2	Farista	Perempuan	jalan kembang turi	D3 Manajemen Informatika

Gambar 5. 22 Halaman Mahasiswa D3 Manajemen Informatika

5.3.3.3 Halaman Mahasiswa D4 Teknik Informatika

Halaman mahasiswa D4 teknik informatika merupakan halaman yang berisi data-data mahasiswa D4 yang telah mendaftar pada aplikasi doristec. Pada saat halaman ini di buka panitia akan diarahkan langsung menuju daftar mahasiswa D4 yang pernah mendaftar. Data tersebut antara lain nama, jenis kelamin, alamat, dan program studi seperti yang ditunjukkan pada gambar 5.23.



The screenshot shows a web browser at localhost/frontendskripsi/admin/users/mhsd4. The dashboard has a sidebar with 'Dashboard', 'D3 Manajemen Informatika', 'D4 Teknik Informatika', and 'Master'. The main content area is titled 'Data D4 Teknik Informatika' and contains a table of students.

No.	Nama	Jenis kelamin	Alamat	Prodi
1	Anang ham	Laki-laki	Jalan kesumba no.15	D4 Teknik Informatika
2	imam nawawi	Laki-laki	jalan	D4 Teknik Informatika
3	Renita	Perempuan	cob	D4 Teknik Informatika

Gambar 5. 23 Halaman mahasiswa D4 teknik informatika

5.3.3.4 Halaman Mahasiswa D3 Yang Sudah Upload

Halaman mahasiswa D3 yang sudah *upload* merupakan halaman yang berisi data-data mahasiswa D3 yang telah *upload* dokumen tugas akhir pada aplikasi doristec. Pada saat halaman ini di buka panitia akan diarahkan langsung menuju daftar mahasiswa D3 beserta dokumen tugas akhir yang telah di *upload*. Data tersebut antara lain nama, tahun, dan judul seperti yang ditunjukkan pada gambar 5.24.



Doristec Admin

admin sedang

Dashboard

D3 Manajemen Informatika

D4 Teknik Informatika

Master

Data Telah mengumpulkan dokumen

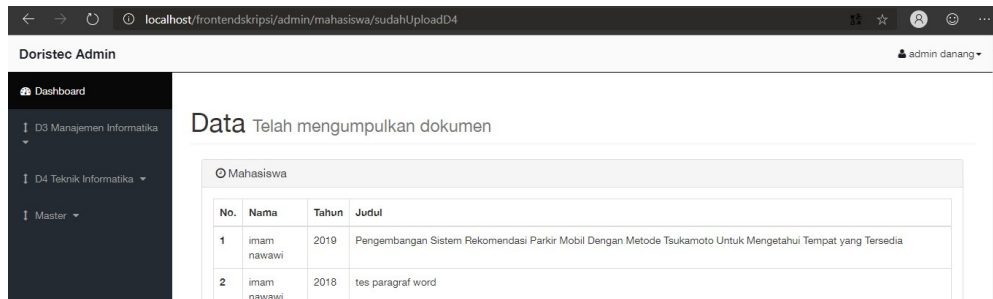
Mahasiswa

No.	Nama	Tahun	Judul
1	Farista	2018	RANCANG BANGUN GAME PERTEMPURAN 10 NOVEMBER

Gambar 5. 24 Halaman mahasiswa D3 yang sudah *upload*

5.3.3.5 Halaman Mahasiswa D4 Yang Sudah *Upload*

Halaman mahasiswa D4 yang sudah *upload* merupakan halaman yang berisi data-data mahasiswa D4 yang telah *upload* dokumen skripsi pada aplikasi doristec. Pada saat halaman ini di buka panitia akan diarahkan langsung menuju daftar mahasiswa D4 beserta dokumen skripsi yang telah di *upload*. Data tersebut antara lain nama, tahun, dan judul seperti yang ditunjukkan pada gambar 5.25.



Doristec Admin

admin sedang

Dashboard

D3 Manajemen Informatika

D4 Teknik Informatika

Master

Data Telah mengumpulkan dokumen

Mahasiswa

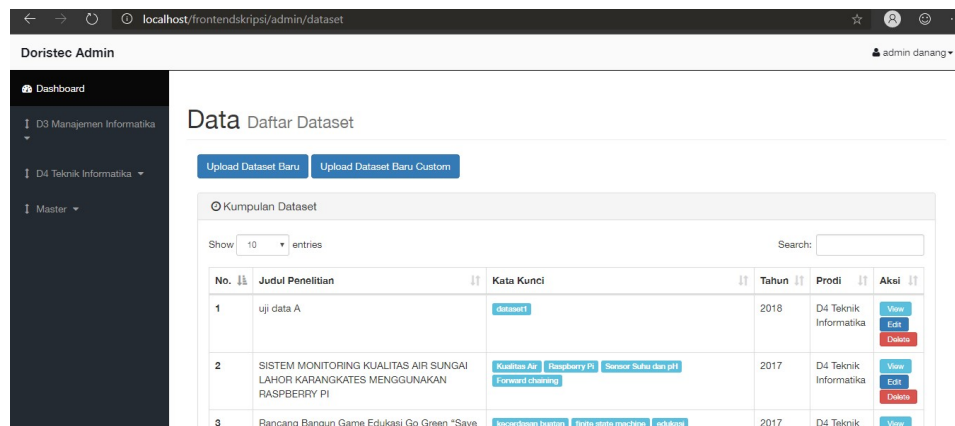
No.	Nama	Tahun	Judul
1	imam nawawi	2019	Pengembangan Sistem Rekomendasi Parkir Mobil Dengan Metode Tsukamoto Untuk Mengetahui Tempat yang Tersedia
2	imam nawawi	2018	tes paragraf word

Gambar 5. 25 Halaman mahasiswa D4 yang sudah *upload*

5.3.3.6 Halaman *Dataset*

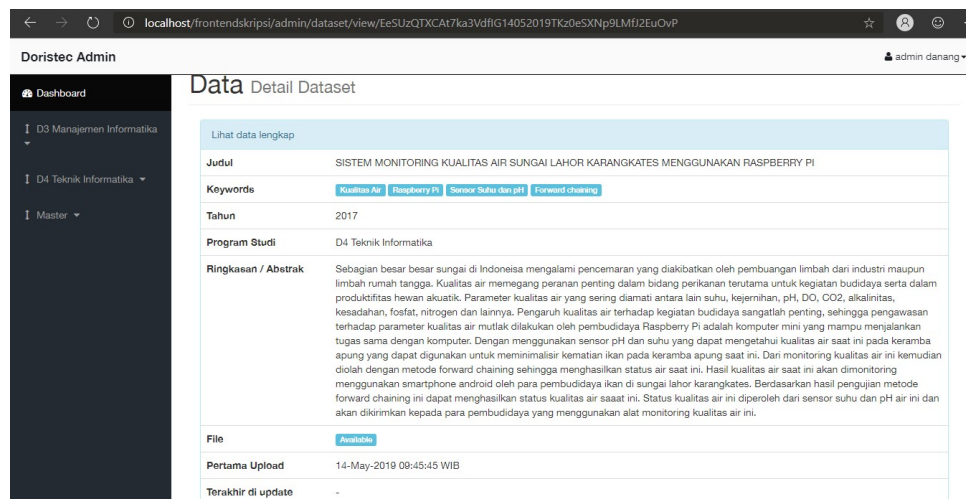
Halaman *dataset* merupakan halaman yang berisi data-data yang telah tersimpan pada *database*, data ini berupa laporan tugas akhir dan laporan skripsi yang digunakan sebagai *dataset* aplikasi ini. Fitur-fitur yang terdapat pada aplikasi ini berfungsi untuk menampilkan dokumen pembanding yang pernah di *upload* dan sudah tersimpan di *database*. Data tersebut antara lain judul penelitian, kata kunci, tahun dan program studi pada halaman ini panitia mempunyai hak akses untuk mengolah data secara keseluruhan seperti yang ditunjukkan pada gambar 5.26, pada halaman ini terdapat tiga tombol yaitu tombol lihat, edit dan hapus. Tombol lihat berfungsi untuk melihat secara keseluruhan data yang terpilih, tombol edit berfungsi untuk mengedit data yang terpilih, sedangkan tombol hapus untuk menghapus data yang terpilih.

Selain itu juga terdapat dua tombol diatas tabel yaitu tombol *upload* dokumen pembanding baru dan *upload* dokumen pembanding baru *custom*. Tombol *upload* dokumen pembanding baru berfungsi untuk menambah koleksi data pembanding baru. Dalam hal ini admin lebih memiliki fitur yang lebih dalam proses upload dokumen baru dengan batasan program studi dan tahun tidak bisa dipilih atau telah otomatis terpilih oleh sistem, sedangkan *upload* dokumen pembanding baru *custom* memiliki *form* yang serupa tetapi program studi dan tahun dapat dipilih sesuai kebutuhan.



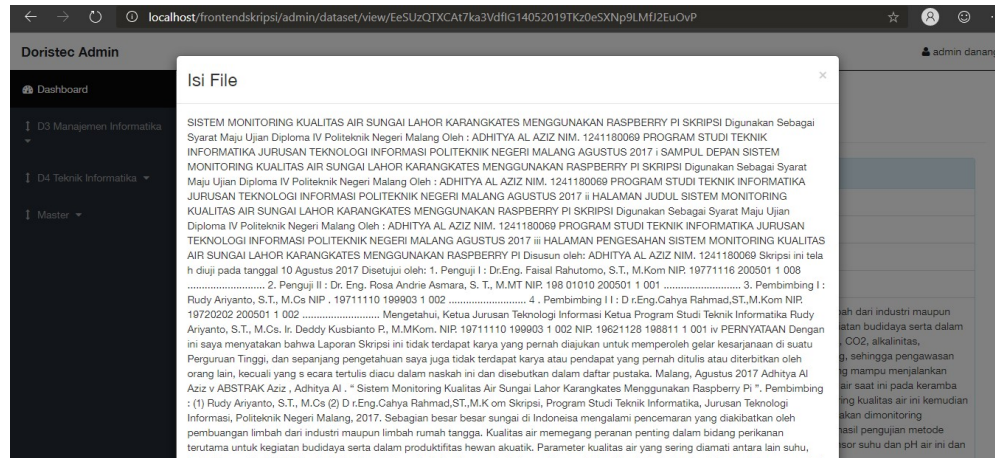
Gambar 5. 26 Halaman *dataset*

Apabila ingin melihat data secara lengkap, panitia dapat menekan tombol lihat, maka akan ditampilkan data secara lengkap berdasarkan data yang dipilih seperti ditunjukkan pada gambar 5.27.



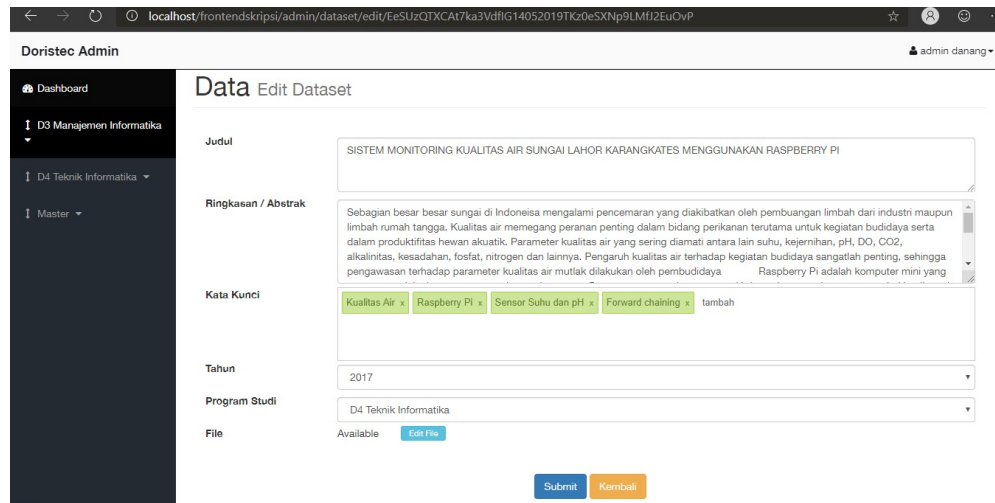
Gambar 5. 27 Halaman Lihat Data Lengkap

Sedangkan untuk mengetahui isi teks secara keseluruhan, panitia dapat menekan tombol available yang artinya teks tersedia dalam *database*, seperti yang ditunjukkan pada gambar 5.28.



Gambar 5. 28 Halaman Lihat Teks Keseluruhan

Ketika tombol edit ditekan, maka informasi yang akan dirubah akan tampil dihalaman input data seperti ditunjukkan gambar 5.29 namun dalam keadaan form yang sudah terisi. Panitia cukup merubah data yang diinginkan saja tanpa perlu memasukkan semua informasi dari awal.

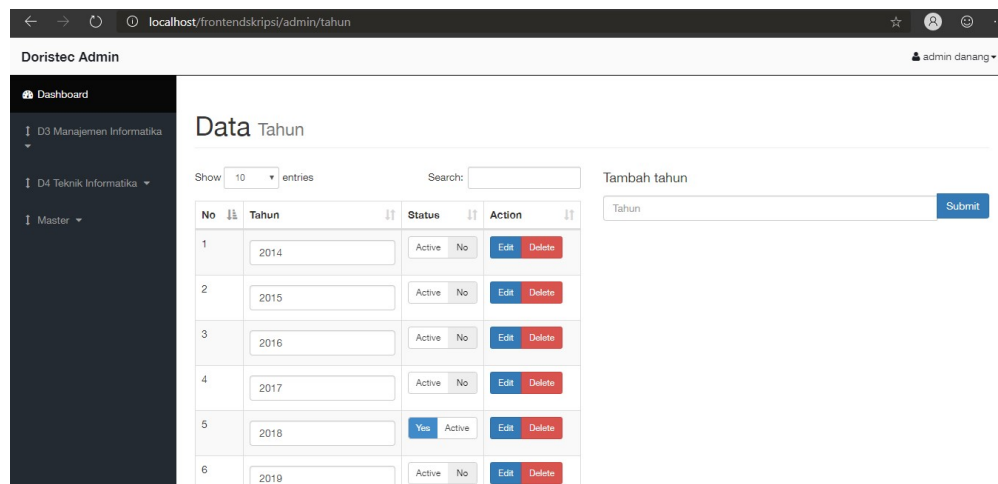


Gambar 5. 29 Halaman Edit Data

5.3.3.7 Halaman Tahun

Halaman tahun merupakan halaman yang berisi data-data yang telah tersimpan pada *database*, data ini berupa tahun yang digunakan sebagai penetapan

tahun upload dokumen mahasiswa. Fitur-fitur yang terdapat pada aplikasi ini berfungsi untuk menampilkan semua tahun yang pernah di masukkan kedalam sistem dan disimpan di *database*. Data tersebut antara lain nama tahun dan status, pada halaman ini panitia mempunyai hak akses untuk mengolah data tahun secara keseluruhan seperti yang ditunjukkan pada gambar 5.30, pada halaman ini terdapat dua tombol yaitu tombol edit dan hapus. Tombol edit berfungsi untuk mengedit data yang terpilih namun edit disini tidak perlu berganti halaman lagi melainkan dapat langsung dari tiap baris data tersebut, sedangkan tombol hapus untuk menghapus data yang terpilih. Pada bagian status merupakan opsi *switch* yang berisi kondisi aktif *Yes or No*. Selain itu juga terdapat form tambah tahun pada bagian kanan tabel.



Gambar 5. 30 Halaman Tahun

5.4 Pengujian

Pengujian merupakan proses pengujian yang dilakukan untuk menguji serta mendokumentasikan hasil pengujian. Pengujian dilakukan menggunakan metode blackbox, metode ini digunakan sebagai demo aplikasi dan mendokumentasi segala aktifitas jalannya aplikasi. Hal ini difungsikan untuk dokumentasi segala kemungkinan yang terjadi ketika aplikasi dijalankan, sehingga dapat diketahui apakah aplikasi telah berjalan dengan baik atau tidak.

5.4.1 Pengujian Fungsionalitas Sistem

Pengujian sistem merupakan pengujian yang dilakukan pada aplikasi, pengujian ini dilakukan untuk mengetahui berjalannya aplikasi dapat berfungsi

dengan baik. Pengujian pertama dilakukan pada halaman awal aplikasi yang berisi *form* registrasi dan *form* login, seperti yang ditunjukkan pada tabel 5.2.

Tabel 5. 2 Pengujian Sistem Pada Halaman Awal Aplikasi

No	Skenario	Hasil yang diharapkan	Hasil yang terjadi	Keterangan
1	Melakukan <i>input</i> pada <i>form</i> registrasi	Data pengguna dapat diinputkan	Proses berjalan dengan normal	Berhasil
2	Melakukan <i>input</i> pada <i>form</i> login	Dapat masuk pada halaman admin panel	Proses berjalan normal. Ketika tombol <i>login</i> ditekan	Berhasil

Dikarenakan aplikasi memiliki dua level yang berbeda yaitu Mahasiswa dan Panitia, maka akan dibuat tabel blackbox yang berbeda yaitu tabel 5. Digunakan untuk level mahasiswa dan tabel 5. Untuk level panitia. Berikut ini adalah metode blackbox untuk pengujian aplikasi :

Tabel 5. 3 Pengujian Sistem Pada Level Mahasiswa

No	Skenario	Hasil yang diharapkan	Hasil yang terjadi	Keterangan
1	Melakukan pengambilan data dari <i>database</i>	Dapat menampilkan semua data uji pengguna	Proses berjalan dengan normal	Berhasil
2	Menampilkan detail dokumen pengguna	Dapat menampilkan data keseluruhan dari dokumen yang dipilih	Proses berjalan dengan normal	Berhasil
3	Melakukan perubahan pada data dokumen	Dapat merubah satu atau lebih data dokumen	Proses berjalan dengan normal, ketika tombol <i>submit</i> ditekan	Berhasil
4	Menghapus dokumen	Melakukan hapus data pada <i>database</i>	Proses berjalan dengan normal	Berhasil
5	<i>Upload</i> dokumen baru	Menambah dokumen baru dan disimpan ke <i>database</i>	Proses berjalan dengan normal, ketika tombol <i>submit</i> ditekan	Berhasil
6	Menampilkan data-data yang pernah dilakukan pengujian	Dapat menampilkan data keseluruhan yang pernah diuji	Proses berjalan dengan normal	Berhasil

Lanjutan Tabel 5. 3 Pengujian Sistem Pada Level Mahasiswa

No	Skenario	Hasil yang diharapkan	Hasil yang terjadi	Keterangan
7	Menampilkan detail hasil pengujian	Dapat menampilkan data keseluruhan dari data yang dipilih	Proses berjalan dengan normal	Berhasil
8	<i>Filter</i> Kandidat Pembanding	Pengambilan kandidat pembanding berdasarkan kata kunci	Proses berjalan dengan normal, ketika tombol proses ditekan	Berhasil
9	Menampilkan data hasil <i>filter</i>	Menampilkan hasil <i>filter</i> yang digunakan selanjutnya untuk proses LCS	Proses berjalan dengan normal	Berhasil
10	LCS	Proses membandingkan dokumen uji dengan kandidat pembanding	Proses berjalan dengan normal	Berhasil
11	Menampilkan profil pengguna	Menampilkan data profil pengguna	Proses berjalan dengan normal	Berhasil

Tabel 5. 4 Pengujian Sistem Pada Level Panitia

No	Skenario	Hasil yang diharapkan	Hasil yang terjadi	Keterangan
1	Menampilkan statistik	Menampilkan ststistik pada halaman <i>dashboard</i> , yaitu jumlah D3 yang upload, jumlah D4 yang upload, mahasiswa terdaftar	Proses berjalan dengan normal	Berhasil
2	Menampilkan daftar mahasiswa D3	Menampilkan data mahasiswa D3	Proses berjalan dengan normal	Berhasil
3	Menampilkan daftar mahasiswa D3	Menampilkan data mahasiswa D3	Proses berjalan dengan normal	Berhasil
4	Menampilkan daftar mahasiswa D4	Menampilkan data mahasiswa D4	Proses berjalan dengan normal	Berhasil

Lanjutan Tabel 5. 4 Pengujian Sistem Pada Level Panitia

No	Skenario	Hasil yang diharapkan	Hasil yang terjadi	Keterangan
5	Menampilkan dokumen mahasiswa D3	Menampilkan data mahasiswa D3	Proses berjalan dengan normal	Berhasil
6	Menampilkan dokumen mahasiswa D4	Menampilkan data mahasiswa D4	Proses berjalan dengan normal	Berhasil
7	Menampilkan data tahun	Menampilkan data tahun dari <i>database</i>	Proses berjalan dengan normal	Berhasil
8	Melakukan <i>input</i> pada <i>form</i> tambah tahun	Data dapat diinputkan	Proses berjalan dengan normal	Berhasil
9	Melakukan perubahan pada data tahun	Dapat merubah satu atau lebih data tahun	Proses berjalan dengan normal, ketika tombol <i>edit</i> ditekan	Berhasil
10	Menghapus tahun	Melakukan hapus data pada <i>database</i>	Proses berjalan dengan normal	Berhasil
11	Menampilkan data semua data pada <i>dataset</i>	Menampilkan semua data pada <i>dataset</i> dari <i>database</i>	Proses berjalan dengan normal	Berhasil
12	Menampilkan detail data	Dapat menampilkan data keseluruhan dari kandidat	Proses berjalan dengan normal	Berhasil
13	Melakukan perubahan pada data	Dapat merubah satu atau lebih data	Proses berjalan dengan normal, ketika tombol <i>submit</i> ditekan	Berhasil
14	Menghapus data pada <i>dataset</i>	Melakukan hapus data pada <i>database</i>	Proses berjalan dengan normal	Berhasil
15	<i>Upload</i> dokumen pembanding baru	Menambah dokumen pembanding baru dan disimpan ke <i>database</i> , namun program studi, tahun tidak dapat dipilih	Proses berjalan dengan normal, ketika tombol <i>submit</i> ditekan	Berhasil
16	<i>Upload</i> dokumen pembanding baru <i>custom</i>	Menambah dokumen pembanding baru dengan <i>input</i> program studi, tahun dapat dipilih	Proses berjalan dengan normal, ketika tombol <i>submit</i> ditekan	Berhasil

Lanjutan Tabel 5. 4 Pengujian Sistem Pada Level Panitia

No	Skenario	Hasil yang diharapkan	Hasil yang terjadi	Keterangan
		dan disimpan ke <i>database</i>		
17	Menampilkan hasil dokumen yang telah diuji	Menampilkan hasil dokumen mahasiswa yang telah diuji	Proses berjalan dengan normal	Berhasil
18	Membuat <i>ranking</i> pada kandidat pembanding	Dapat meranking nilai plagiarisme pada kandidat pembanding, hingga ditemukan nilai tertinggi dan nilai terendah plagiarismenya	Proses berjalan dengan normal	Berhasil
19	Menampilkan detail pada setiap kandidat pembanding yang ditemukan	Dapat menampilkan nilai plagiarisme, kalimat-kalimat yang ditemukan	Proses berjalan dengan normal	Berhasil
20	Menampilkan profil pengguna	Menampilkan data profil pengguna	Proses berjalan dengan normal	Berhasil

5.4.2 Pengujian Metode

Pengujian Metode merupakan pengujian yang dilakukan pada aplikasi dengan terfokus pada metode yang diterapkan pada aplikasi, pengujian ini dilakukan untuk mengetahui berjalannya metode LCS dapat berfungsi dengan baik. Pengujian manual dengan metode LCS akan menggunakan tiga studi kasus yang diberikan. Pada setiap studi kasus memiliki beberapa bentuk adanya plagiarisme, bentuk proses perhitungan yang dilakukan dapat digambarkan pada tahapan sebagai berikut :

- a. Telah ditentukan jumlah dokumen yang diuji yaitu dua dokumen yang disebut satu dokumen uji dan satu kandidat pembanding:
 1. S1 : Dokumen terpanjang
 2. S2 : Dokumen yang lebih pendek
- b. Menentukan dokumen terpanjang dari data uji dan kandidat pembanding
 1. Dokumen terpanjang disimpan pada variabel S1

2. Dokumen yang lebih pendek disimpan pada variabel S2
- c. Melakukan pencarian kata dengan urutan kata terpanjang menggunakan rumus yang ditunjukkan pada gambar 5.31 berikut ini.

$$L(p_{1:i}, q_{1:j}) = \begin{cases} \{\epsilon\} \\ L(p_{1:i-1}, q_{1:j-1})p_i \\ \text{longest}\{L(p_{1:i}, q_{1:j}), L(p_{1:i-1}, q_{1:j-1})\} \end{cases}$$

Gambar 5. 31 Rumus LCS

- d. Simpan hasil pencarian kata terpanjang pada variabel
- e. Menghitung prosentase plagiarisme pada setiap kandidat pembanding

$$\frac{\text{Hasil pencarian kata terpanjang}}{\text{dokumen terpanjang}} \times 100$$

- f. Jika hasil pencarian kata terpanjang memiliki lebih dari tiga kata, maka ulangi langkah C. jika kurang dari tiga kata dan kandidat pembanding lebih dari nol, maka lanjutkan pencarian pada kandidat pembanding selanjutnya dan mengulangi langkah B. Apabila kandidat pembanding tidak lebih dari nol maka proses menghitung menggunakan metode LCS selesai.
- g. Hasil disimpan pada *database*.

Berikut ini pengujian menggunakan tiga studi kasus yang telah disediakan berupa tahapan sebagai berikut :

5.4.2.1 Kasus Pertama

Pengujian pada kasus pertama merupakan pengujian yang dilakukan dengan memberikan dokumen yang sama antara kandidat pembanding dan data uji, berikut ditunjukkan pada tabel 5.5.

Tabel 5. 5 Kasus Pertama Pengujian Metode

No	Isi Dokumen	Jumlah Kata	Keterangan
1	Secara umum, Project Athena bertujuan menghadirkan produk laptop dengan <i>battery life</i> yang luar biasa awet dengan sasaran para pengguna <i>mobile</i> , pekerja lepas yang biasa	84 Kata	Dokumen Uji

Lanjutan Tabel 5. 5 Kasus Pertama Pengujian Metode

No	Isi Dokumen	Jumlah Kata	Keterangan
	bekerja di mana saja, misalnya warung kopi, perpustakaan, taman, atau tempat-tempat umum lainnya. Bukan saja awet, laptop ini nantinya akan mampu memberikan simulasi akurat berapa lama baterai laptop akan bertahan. Setiap upaya yang dilakukan pengguna untuk memperpanjang daya tahan baterai, seperti membatasi penggunaan internet, konsumsi media, atau meredupkan layar, akan langsung dikalkulasikan sehingga menghasilkan perhitungan sisa waktu yang sangat akurat.		
2	Secara umum, Project Athena bertujuan menghadirkan produk laptop dengan <i>battery life</i> yang luar biasa awet dengan sasaran para pengguna <i>mobile</i> , pekerja lepas yang biasa bekerja di mana saja, misalnya warung kopi, perpustakaan, taman, atau tempat-tempat umum lainnya. Bukan saja awet, laptop ini nantinya akan mampu memberikan simulasi akurat berapa lama baterai laptop akan bertahan. Setiap upaya yang dilakukan pengguna untuk memperpanjang daya tahan baterai, seperti membatasi penggunaan internet, konsumsi media, atau meredupkan layar, akan langsung dikalkulasikan sehingga menghasilkan perhitungan sisa waktu yang sangat akurat.	84 Kata	Kandidat pembanding

Tabel 5. 6 Perhitungan Manual Pada Studi Kasus Pertama

No	Kategori Dokumen	Jumlah Kata	Jumlah ditemukan kata	Jumlah kalimat ditemukan
1	Dokumen Uji	84 Kata		
2	Kandidat pembanding	84 Kata	84 kata	1

5.4.2.2 Kasus Kedua

Pengujian pada kasus kedua merupakan pengujian yang dilakukan dengan memberikan dokumen data uji dengan memiliki beberapa kemiripan kalimat pada tiga kandidat pembanding yang diujikan, berikut ditunjukkan pada tabel 5.7.

Tabel 5. 7 Kasus Kedua Pengujian Metode

No	Isi Dokumen	Jumlah Kata	Keterangan
1	<p>Secara umum, Project Athena bertujuan menghadirkan produk laptop dengan <i>battery life</i> yang luar biasa awet dengan sasaran para pengguna <i>mobile</i>, pekerja lepas yang biasa bekerja di mana saja, misalnya warung kopi, perpustakaan, taman, atau tempat-tempat umum lainnya. Bukan saja awet, laptop ini nantinya akan mampu memberikan simulasi akurat berapa lama baterai laptop akan bertahan. Setiap upaya yang dilakukan pengguna untuk memperpanjang daya tahan baterai, seperti membatasi penggunaan internet, konsumsi media, atau meredupkan layar, akan langsung dikalkulasikan sehingga menghasilkan perhitungan sisa waktu yang sangat akurat. Menurut Josh Newman, <i>Vice President Client Computing</i> dari Intel, Project Athena akan memberikan masa pakai baterai paling tidak 9 jam untuk pemakaian berat di dunia nyata. Sistem ini juga akan dilengkapi dengan <i>fast charging</i> yang akan menjadikan pengguna <i>mobile</i> tidak akan lagi kesulitan untuk menambah daya baterainya di mana pun. Newman juga menambahkan bahwa solusi untuk mencapai hal ini bukanlah dengan memperbesar kapasitas baterai, karena ini nanti akan merusak pengalaman pengguna yang menginginkan laptop tipis premium. Intel akan mencapai ini dengan efisiensi daya yang optimal dan melakukan <i>tuning</i> yang tepat terhadap keseluruhan komponen laptop. Penyesuaian ini juga akan dibantu oleh sistem A.I, misalnya, dengan kamera yang didukung A.I, perangkat akan dapat mengenali apakah pengguna berada di depan layar monitor atau tidak, dan mengambil keputusan untuk meredupkan layar atau bahkan membawa laptop ke posisi <i>sleep</i>. Intel juga sudah mengambil langkah lebih jauh dengan bekerja bersama para pembuat komponen komputer untuk menghadirkan perangkat Project Athena generasi pertama.</p>	262 Kata	Dokumen Uji

Lanjutan Tabel 5. 7 Kasus Kedua Pengujian Metode

No	Isi Dokumen	Jumlah Kata	Keterangan
	Jika perangkat <i>prototype</i> tersedia, Intel nantinya juga akan membuka diri terhadap manufaktur PC, sehingga lebih banyak manufaktur yang dapat memanfaatkan teknologi Intel ini.		
2	Ini berarti bahwa pengguna Windows PowerShell dan PowerShell Core akan dapat menggunakan versi yang sama dari PowerShell untuk mengotomatisasi seluruh Windows, Linux, dan macOS dan pada Windows, dan pengguna PowerShell 7 akan memiliki tingkat kompatibilitas yang sangat tinggi dengan modul Windows PowerShell yang mereka andalkan. Setiap upaya yang dilakukan pengguna untuk memperpanjang daya tahan baterai, seperti membatasi penggunaan internet, konsumsi media, atau meredupkan layar, akan langsung dikalkulasikan sehingga menghasilkan perhitungan sisa waktu yang sangat akurat. Dengan kata lain, aplikasi UWP dan Win32 akan terus hidup berdampingan dengan paket fitur yang sama dan dengan Microsoft Store masih berfungsi sebagai saluran distribusi yang penting. Meskipun ini pada akhirnya bisa mendorong pengembang dan pengguna kembali ke aplikasi Win32, tentu akan menarik untuk melihat bagaimana Microsoft mengelola agar aplikasi UWP tetap relevan sebagai bagian dari pendekatan yang diperbarui.	134 Kata	Kandidat pembandingan 1
3	Dari perspektif teknis, ini berarti semua fitur akan tersedia untuk semua aplikasi, terlepas dari apakah itu aplikasi UWP atau Win32. Dengan hal ini, pada gilirannya, akan memberi pengguna pengalaman yang lebih konsisten di seluruh perangkat, yang merupakan tujuan jangka panjang Microsoft karena berencana untuk memperluas ekosistem Windows 10 dengan SKU baru dan kategori perangkat. PowerShell Core akan dapat menggunakan versi yang sama dari PowerShell untuk mengotomatisasi seluruh Windows, Linux,	197 Kata	Kandidat pembandingan 2

Lanjutan Tabel 5. 7 Kasus Kedua Pengujian Metode

No	Isi Dokumen	Jumlah Kata	Keterangan
	<p>dan macOS dan pada Windows, dan pengguna PowerShell 7 akan memiliki tingkat kompatibilitas yang sangat tinggi dengan modul Windows PowerShell yang mereka andalkan. Newman juga menambahkan bahwa solusi untuk mencapai hal ini bukanlah dengan memperbesar kapasitas baterai, karena ini nanti akan merusak pengalaman pengguna yang menginginkan laptop tipis premium. Intel akan mencapai ini dengan efisiensi daya yang optimal dan melakukan <i>tuning</i> yang tepat terhadap keseluruhan komponen laptop. Penggunaan PowerShell Core telah tumbuh secara signifikan dalam dua tahun terakhir. Secara khusus, sebagian besar pertumbuhan kami berasal dari penggunaan Linux, statistik yang menggembirakan mengingat investasi kami dalam membuat PowerShell layak untuk lintas platform. Dengan kata lain, aplikasi UWP dan Win32 akan terus hidup berdampingan dengan paket fitur yang sama dan dengan Microsoft Store masih berfungsi sebagai saluran distribusi yang penting.</p>		
4	<p>Dengan kata lain, aplikasi UWP dan Win32 akan terus hidup berdampingan dengan paket fitur yang sama dan dengan Microsoft Store masih berfungsi sebagai saluran distribusi yang penting. Meskipun ini pada akhirnya bisa mendorong pengembang dan pengguna kembali ke aplikasi Win32, tentu akan menarik untuk melihat bagaimana Microsoft mengelola agar aplikasi UWP tetap relevan sebagai bagian dari pendekatan yang diperbarui.</p> <p>Intel juga sudah mengambil langkah lebih jauh dengan bekerja bersama para pembuat komponen komputer untuk menghadirkan perangkat Project Athena generasi pertama, dengan efisiensi daya yang optimal dan melakukan <i>tuning</i> yang tepat terhadap keseluruhan komponen laptop. Penyesuaian ini juga akan dibantu oleh sistem A.I,</p>	156 Kata	Kandidat pembandingan 3

Tabel 5. 8 Perhitungan Manual Pada Studi Kasus Kedua

No	Kategori Dokumen	Jumlah Kata	Jumlah kata ditemukan	Jumlah kalimat ditemukan
1	Dokumen Uji	262 Kata		
2	Kandidat pembanding 1	134 Kata	29 kata	1
3	Kandidat pembanding 2	197 kata	44 kata	1
4	Kandidat pembanding 3	156 kata	41 kata	1

5.4.2.3 Kasus Ketiga

Pengujian pada kasus ketiga merupakan pengujian yang dilakukan dengan memberikan dokumen data uji dengan memiliki beberapa kemiripan kalimat pada dua kandidat pembanding, berikut ditunjukkan pada tabel 5.9.

Tabel 5. 9 Kasus Ketiga Pengujian Metode

No	Isi Dokumen	Jumlah Kata	Keterangan
1	Penggunaan PowerShell Core telah tumbuh secara signifikan dalam dua tahun terakhir. Secara khusus, sebagian besar pertumbuhan kami berasal dari penggunaan Linux, statistik yang menggembirakan mengingat investasi kami dalam membuat PowerShell layak untuk lintas platform. Namun, kami juga dapat dengan jelas melihat bahwa penggunaan Windows kami belum tumbuh secara signifikan, mengejutkan mengingat bahwa PowerShell dipopulerkan pada platform Windows. Ini berarti bahwa pengguna Windows PowerShell dan PowerShell Core akan dapat menggunakan versi yang sama dari PowerShell untuk mengotomatisasi seluruh Windows, Linux, dan macOS dan pada Windows, dan pengguna PowerShell 7 akan memiliki tingkat kompatibilitas yang sangat tinggi dengan modul Windows PowerShell yang mereka andalkan. Dengan kata lain, Microsoft ingin meletakkan semua platform pada versi yang	183 Kata	Dokumen Uji

Lanjutan Tabel 5. 9 Kasus Ketiga Pengujian Metode

No	Isi Dokumen	Jumlah Kata	Keterangan
	<p>sama dari PowerShell untuk kompatibilitas lintas-platform yang lebih baik. Microsoft juga membuat perubahan pada Support Life Cycle PowerShell. Microsoft juga akan merilis LTS (Long Term Servicing) dan versi PowerShell 7 non-LTS untuk menyelaraskannya dengan support life cycle .NET Core.</p> <p>Sekarang Microsoft telah meluncurkan versi preview PowerShell 7. Microsoft mengharapkan untuk meluncurkan preview baru setiap bulan. Ketersediaan Umum PowerShell akan membutuhkan waktu karena Microsoft harus menyelaraskan perilisan dengan .NET Core 3.0.</p>		
2	<p>Secara umum, Project Athena bertujuan menghadirkan produk laptop dengan <i>battery life</i> yang luar biasa awet dengan sasaran para pengguna <i>mobile</i>, pekerja lepas yang biasa bekerja di mana saja, misalnya warung kopi, perpustakaan, taman, atau tempat-tempat umum lainnya. Namun, kami juga dapat dengan jelas melihat bahwa penggunaan Windows kami belum tumbuh secara signifikan, mengejutkan mengingat bahwa PowerShell dipopulerkan pada platform Windows. Bukan saja awet, laptop ini nantinya akan mampu memberikan simulasi akurat berapa lama baterai laptop akan bertahan. Setiap upaya yang dilakukan pengguna untuk memperpanjang daya tahan baterai, seperti membatasi penggunaan internet, konsumsi media, atau meredupkan layar, akan langsung dikalkulasikan sehingga menghasilkan perhitungan sisa waktu yang sangat akurat.</p> <p>Dengan kata lain, aplikasi UWP dan Win32 akan terus hidup berdampingan dengan paket fitur yang sama dan dengan Microsoft Store masih berfungsi sebagai saluran distribusi yang penting.</p> <p>Intel juga sudah mengambil langkah lebih jauh dengan bekerja bersama para pembuat komponen komputer untuk menghadirkan perangkat Project Athena generasi pertama.</p>	198 Kata	Kandidat pembanding 1

Lanjutan Tabel 5. 9 Kasus Ketiga Pengujian Metode

No	Isi Dokumen	Jumlah Kata	Keterangan
	<p>Dengan kata lain, Microsoft ingin meletakkan semua platform pada versi yang sama dari PowerShell untuk kompatibilitas lintas-platform yang lebih baik. Jika perangkat <i>prototype</i> tersedia, Intel nantinya juga akan membuka diri terhadap manufaktur PC, sehingga lebih banyak manufaktur yang dapat memanfaatkan teknologi Intel ini.</p>		
3	<p>Sistem ini juga akan dilengkapi dengan <i>fast charging</i> yang akan menjadikan pengguna <i>mobile</i> tidak akan lagi kesulitan untuk menambah daya baterainya di mana pun. PowerShell dipopulerkan pada platform Windows Newman juga menambahkan bahwa solusi untuk mencapai hal ini bukanlah dengan memperbesar kapasitas baterai, karena ini nanti akan merusak pengalaman pengguna yang menginginkan laptop tipis premium. Microsoft ingin meletakkan semua platform pada versi yang sama intel akan mencapai ini dengan efisiensi daya yang optimal dan melakukan <i>tuning</i> yang tepat terhadap keseluruhan komponen laptop. Penyesuaian ini juga akan dibantu oleh sistem A.I, misalnya, pengguna PowerShell 7 akan memiliki tingkat kompatibilitas dengan kamera yang didukung A.I, perangkat akan dapat mengenali apakah pengguna berada di depan layar monitor atau tidak, dan mengambil keputusan untuk meredupkan layar atau bahkan membawa laptop ke posisi <i>sleep</i>.</p> <p>Intel juga sudah mengambil langkah lebih jauh dengan bekerja bersama para pembuat komponen komputer untuk menghadirkan perangkat Project Athena generasi pertama. Penggunaan PowerShell Core telah tumbuh secara signifikan. Jika perangkat <i>prototype</i> tersedia, Intel nantinya juga akan membuka diri terhadap manufaktur PC, sehingga lebih banyak manufaktur yang dapat memanfaatkan teknologi Intel ini.</p>	180 Kata	Kandidat pembanding 2

Tabel 5. 10 Perhitungan Manual Pada Studi Kasus Ketiga

No	Kategori Dokumen	Jumlah Kata	Jumlah kata disetiap LCS	Jumlah kalimat ditemukan	Total Kata
1	Dokumen Uji	183 Kata			
2	Kandidat pembanding 1	198 Kata	23 kata, 18 kata, 6 kata	3	47 kata
3	Kandidat pembanding 2	180 kata	9 kata, 6 kata, 5 kata, 4 kata	4	24 kata