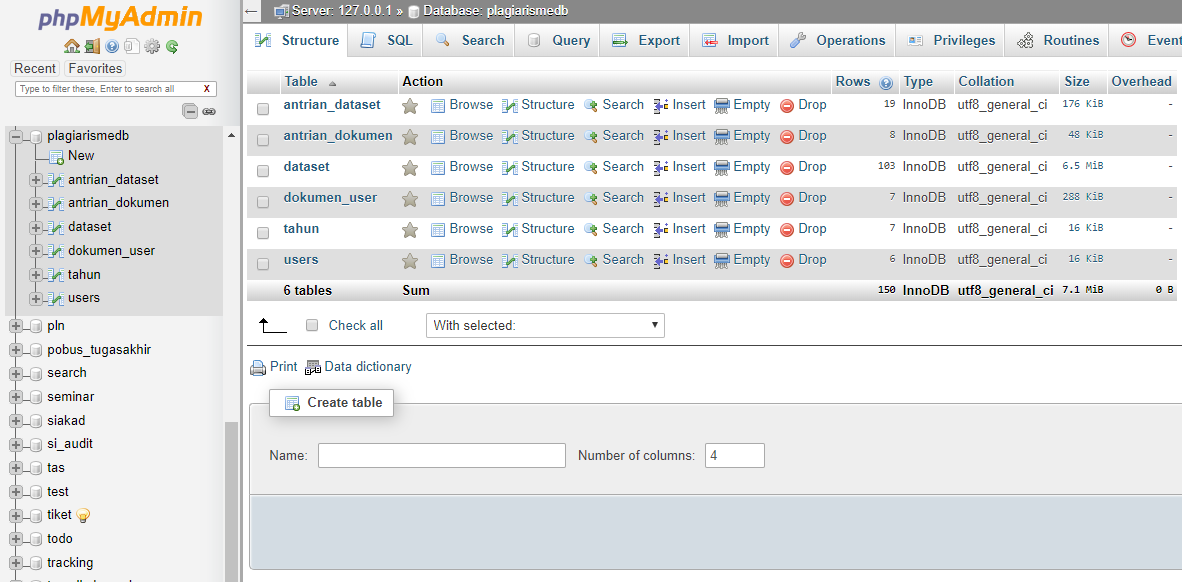
BAB V. IMPLEMENTASI DAN PENGUJIAN

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.

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 merupahan hasil dari penerapan asis data yang dibangung dengan menggunakan MySQL.



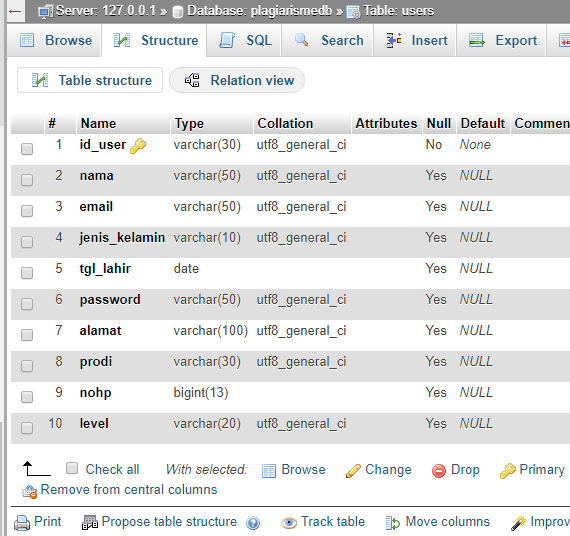
Gambar 5. 1 Basis Data Aplikasi

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 :

1. Gambar Tabel Users

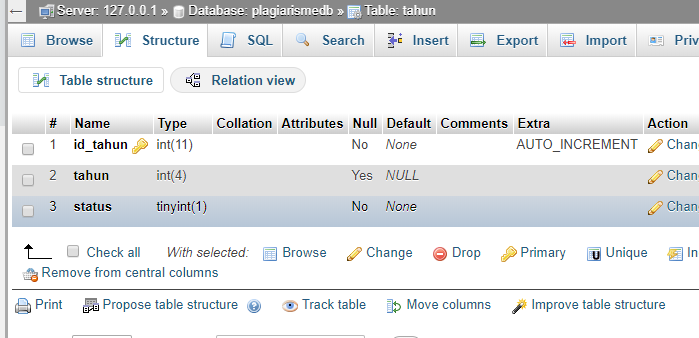
Pada gambar 5.2 menunjukkan implementasi dari rancangan tabel users yang berfungsi untuk menyimpan data mahasiswa dan panitia. Tabel ini memiliki sebanyak 10 *field* dengan 1 *primary key* antara lain : id\_user (*primary key*), nama, email, jenis\_kelamin, tgl\_lahir, *password*, alamat, prodi, nohp,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



Gambar 5. 2 Tabel Users

1. Gambar Tabel Tahun

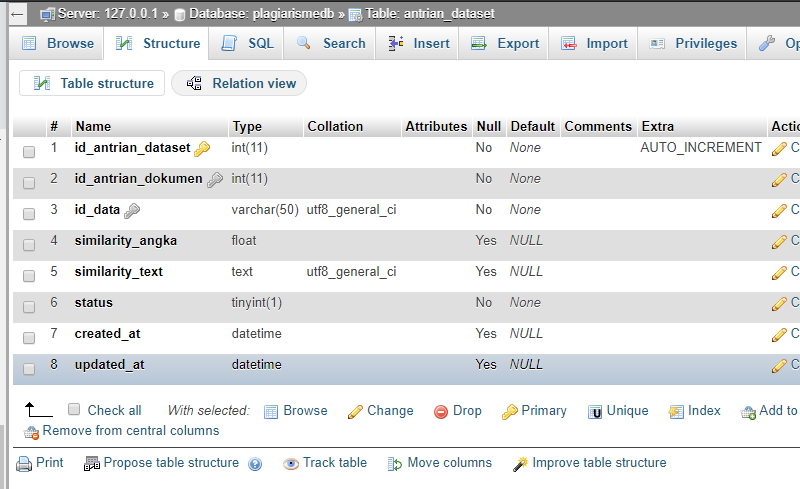
Pada gambar 5.3 menunjukkan implementasi dari rancangan tabel tahun yang berfungsi untuk menyimpan data tahun. Tabel ini memiliki sebanyak 3 *field* dengan memiliki 1 *primary key* antara lain : id\_tahun (*primary key*), tahun, status.



Gambar 5. 3 Tabel Tahun

1. Gambar Tabel Antrian\_*dataset*

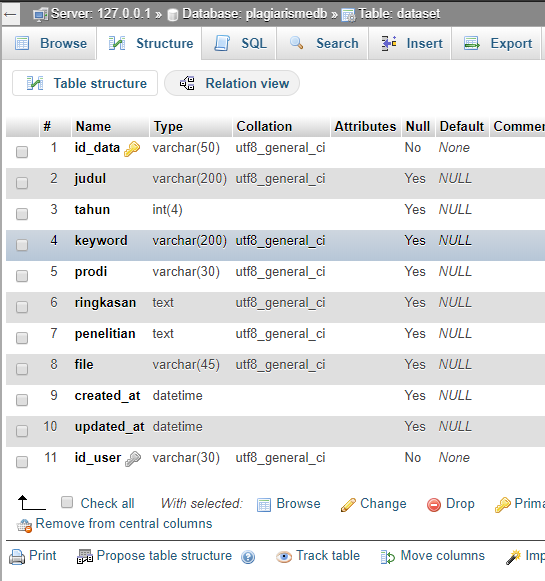
Pada gambar 5.4 menunjukkan implementasi dari rancangan tabel tahun yang berfungsi untuk menyimpan data antrian *dataset,* data antrian yang disimpan merupakan data-data yang diambil dari *dataset* yang dipilih sebagai kandidat pembanding. Tabel ini memiliki sebanyak 8 *field* dengan memiliki 1 *primary 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.



Gambar 5. 4 Tabel Antrian Dataset

1. Gambar Tabel *Dataset*

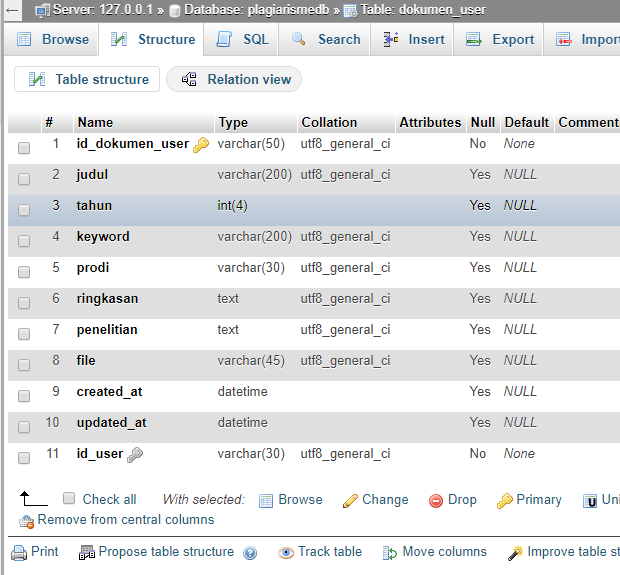
Pada gambar 5.5 menunjukkan implementasi dari rancangan tabel *dataset* yang berfungsi untuk menyimpan *dataset* berupa dokumen skripsi dan tugas akhir terdahulu. Tabel ini memiliki sebanyak 11 *field* dengan memiliki 1 *primary key* dan 1 *foreign key* antara lain : id\_data (*primary key*), judul,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



Gambar 5. 5 Tabel Dataset

1. Gambar Tabel Dokumen\_user

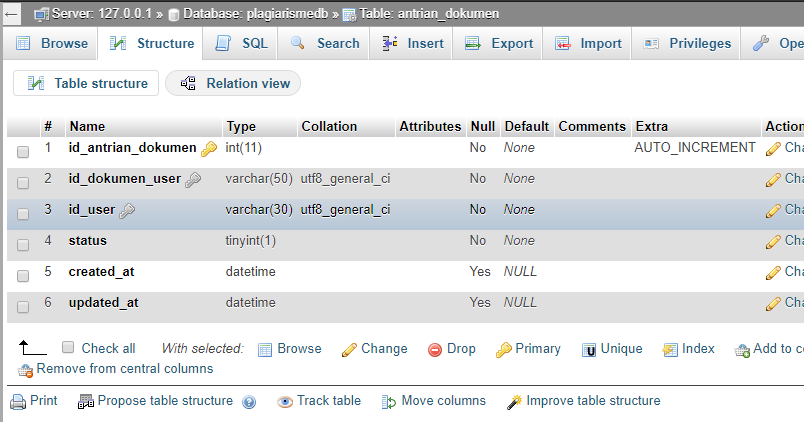
Pada gambar 5.6 menunjukkan implementasi dari rancangan tabel dokumen\_user yang berfungsi untuk menyimpan data dokumen user yang akan diuji yang berupa dokumen skripsi atau tugas akhir. Tabel ini memiliki sebanyak 11 *field* dengan memiliki 1 *primary key* dan 1 *foreign key* antara lain : id\_data (*primary key*), judul, tahun, *keyword*, prodi, ringkasan, penelitian, file, created\_at, updated\_at, id\_user (*foreign key*).



Gambar 5. 6 Tabel Dokumen User

1. 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.



Gambar 5. 7 Tabel Antrian Dokumen

1. 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.*

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 pengcekan 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 projek (*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/frontendskripsi/dokumen';</script>";  // redirect('dokumen','refresh');  } |

1. 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 keywoard |

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 | array(4) {  [0] => string(6) "Sistem"  [1] => string(10) "Penelitian"  [2] => string(7) "Deteksi"  [3] => string(7) "Dokumen"  } |

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.

1. 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 ditunjjukan 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." ".$data1['id\_dokumen\_user']." ".$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 = "plagiarismedb"  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 = self.connection.cursor( MySQLdb.cursors.DictCursor )  cursor.execute(q, va)  return cursor.fetchall()  def queryv(self, q, va):  cursor = self.connection.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 meembuang 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 padamasing-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 ngengetahui 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 |

1. 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 kandidatpembanding*.* *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 kandidatmengunakan 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> |

1. 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.*

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 hasus sesuai dengan identitas pengguna serta hak akses yang dibutuhkan.

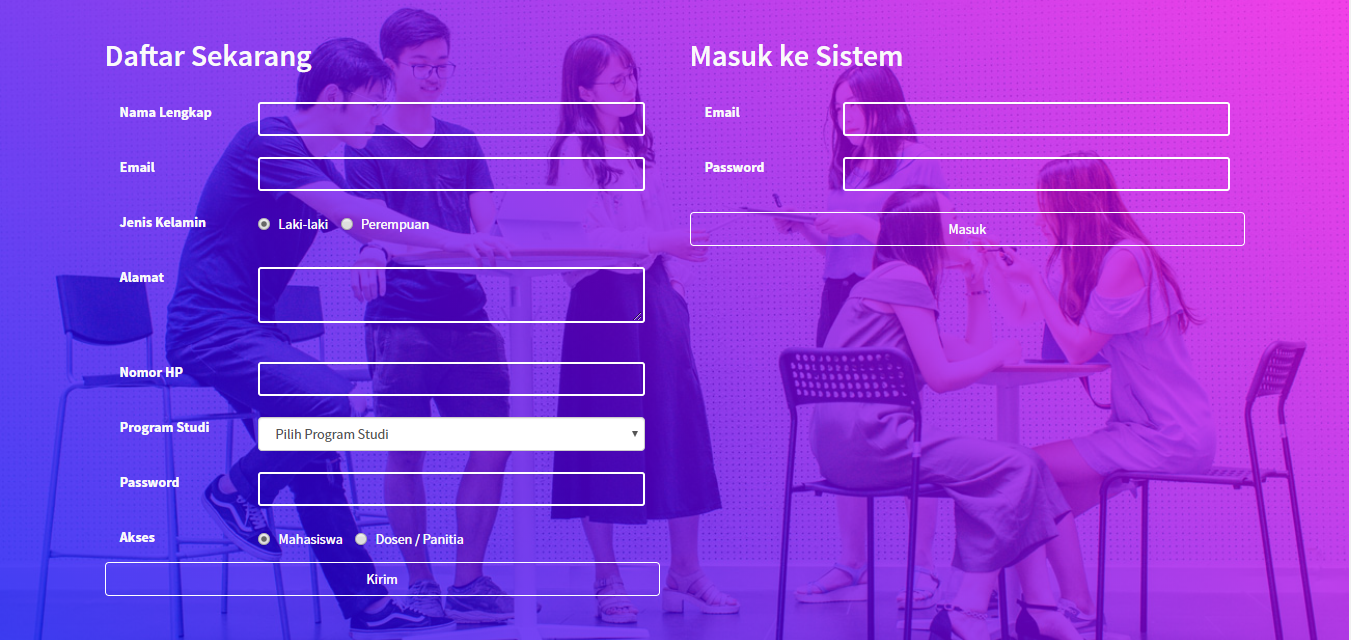
Ketika telah mendapatkan akun, pengguna bisa melanjutkan untuk *login* kedalam sistem doristec. Untuk melakukan *login* Pengguna mahasiswa maupun panitia harus memasukkan *username* dan *password* untuk bisa mendapatkan hak akses.



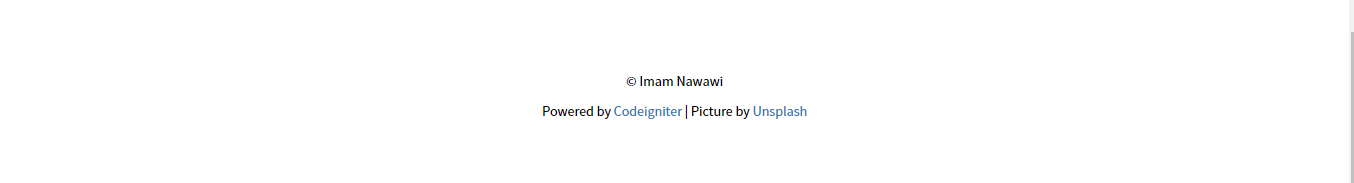
Gambar 5. 8 Halaman Utama Website



Gambar 5. 9 Lanjutan Halaman Utama Website



Gambar 5. 10 Lanjutan Halaman Utama Website



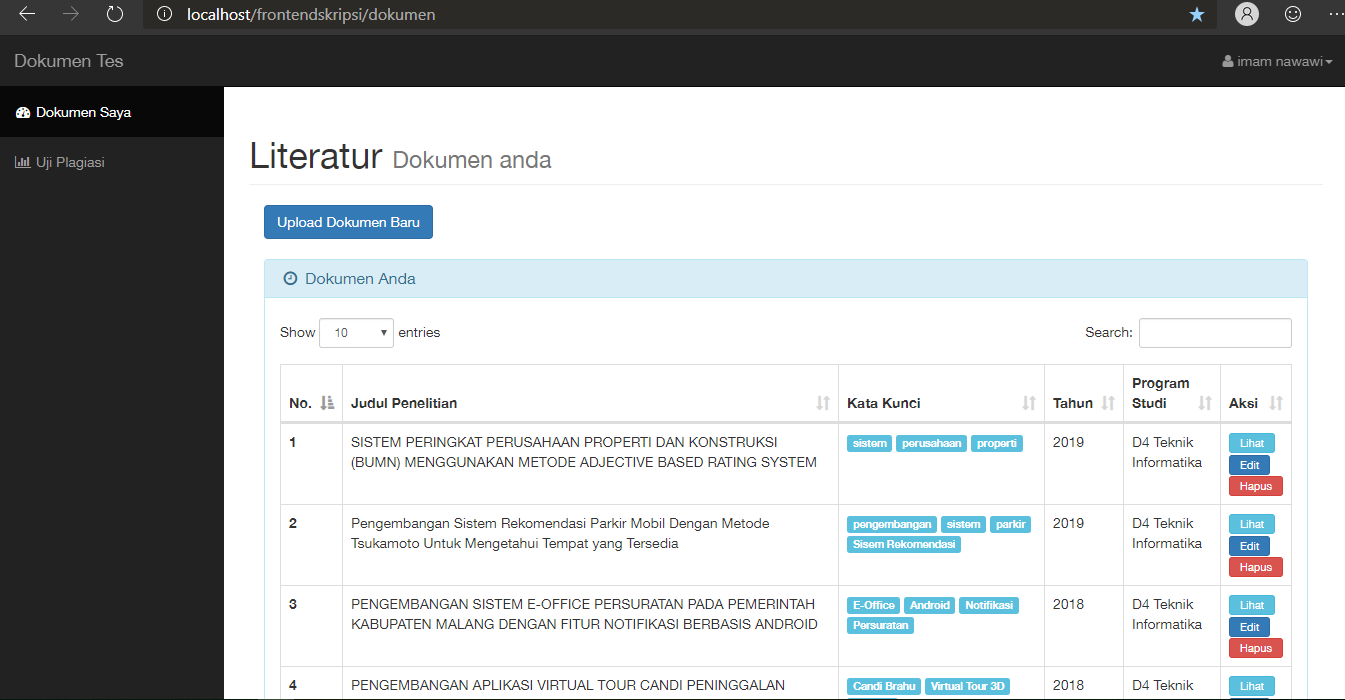
Gambar 5. 11 Lanjutan Halaman Utama Website

1. 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 startbbootstrap-admin*.

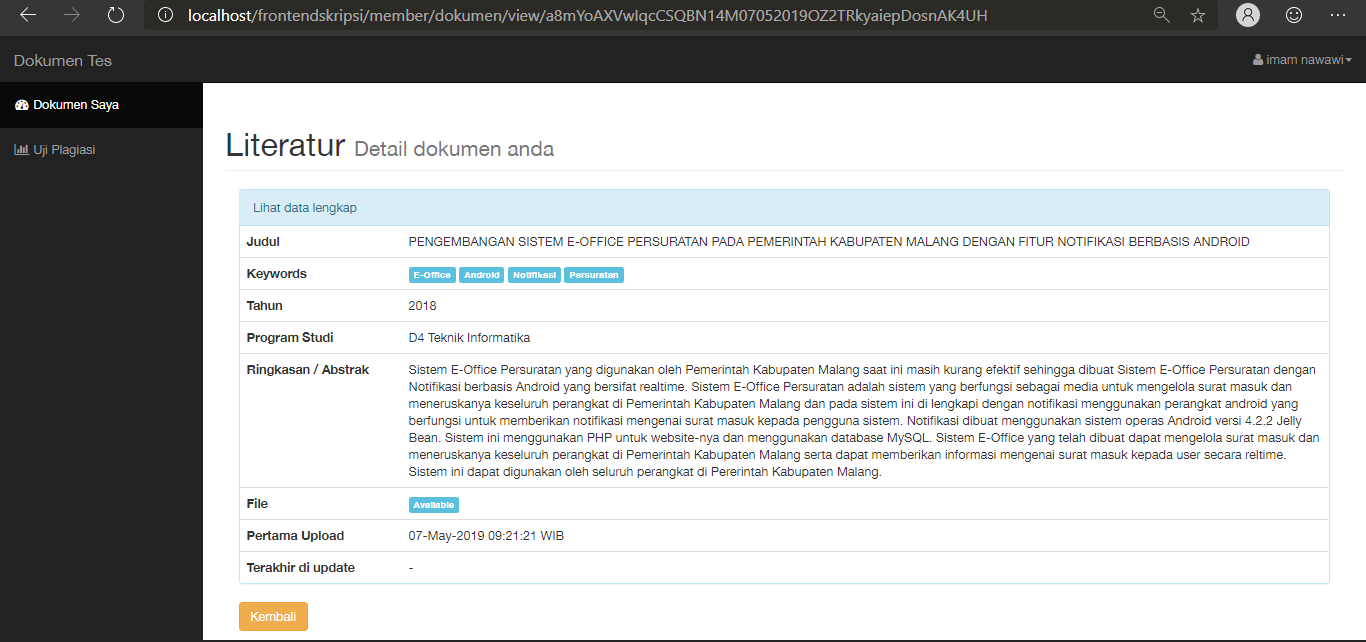
1. Halaman Utama Admin Mahasiswa

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.



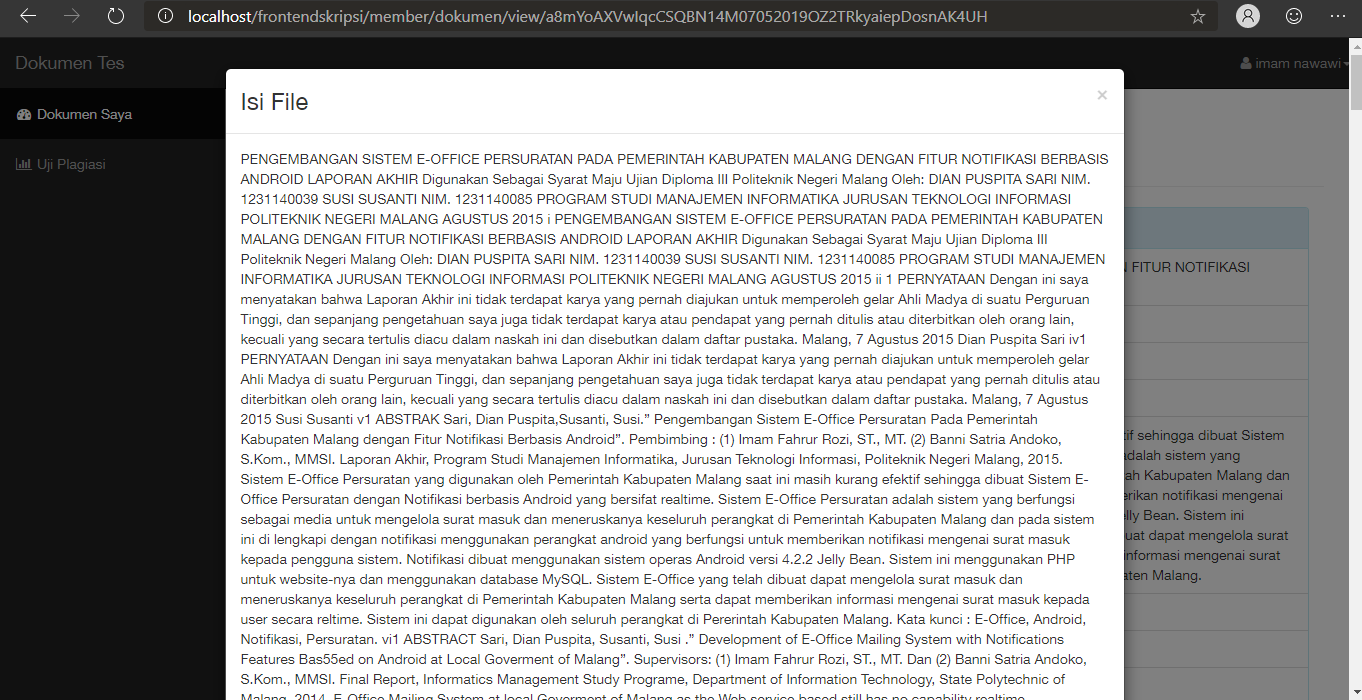
Gambar 5. 12 Halaman Utama Admin Mahasiswa

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.



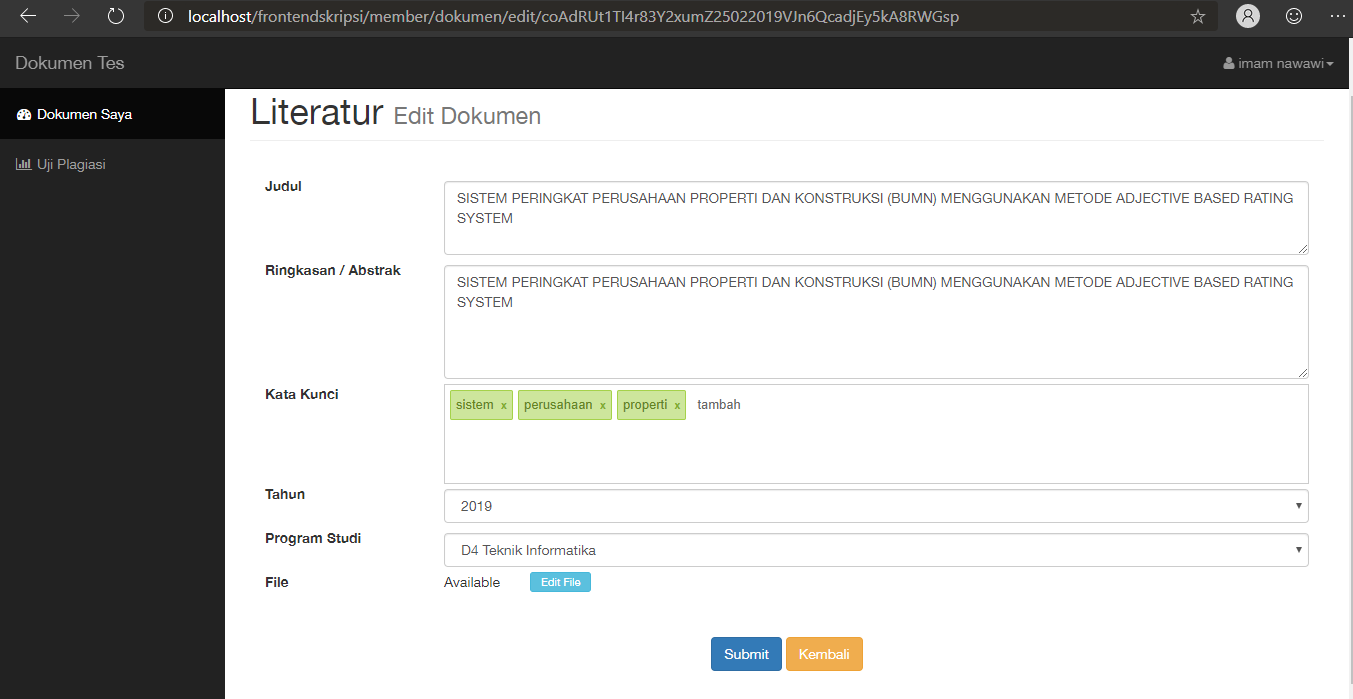
Gambar 5. 13 Halaman Lihat Data

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

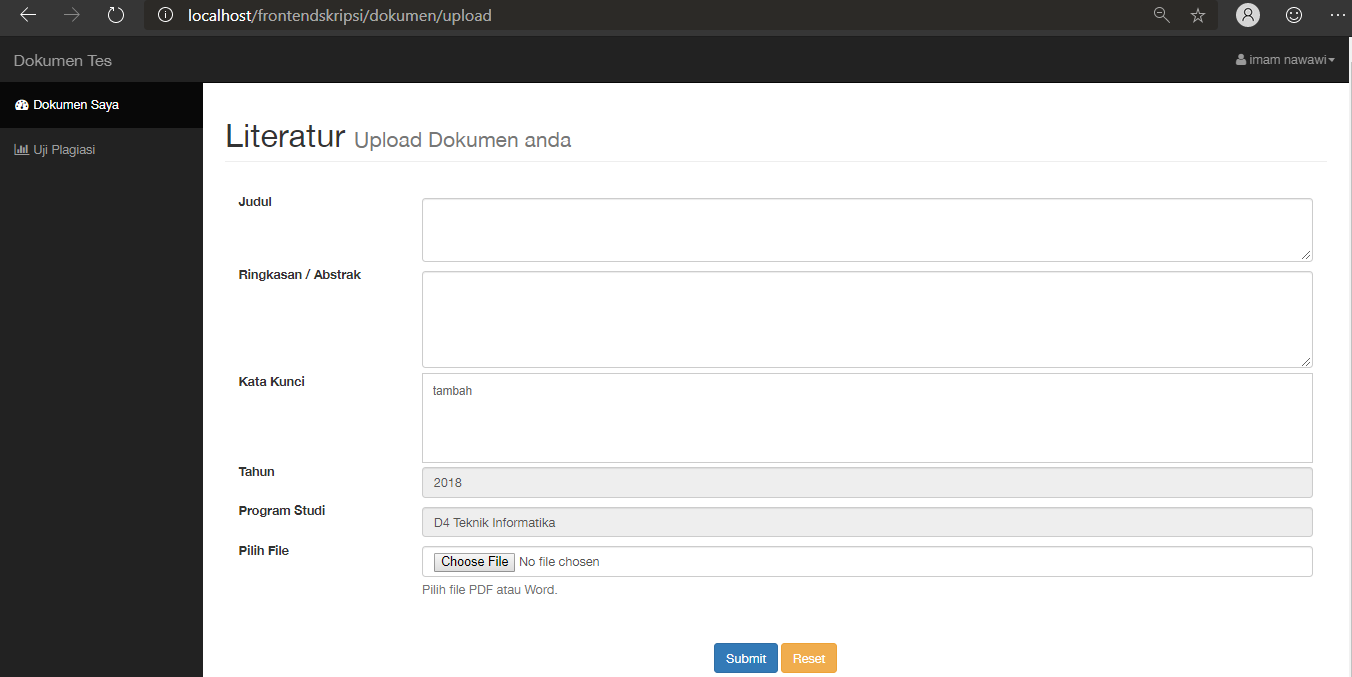
Ketika tombol edit ditekan, maka informasi yang akan dirubah akan tampil dihalaman input data seperti ditunjukkan gambar 5.13 namun dalam keadaan form yang sudah terisi. Mahasiswa cukup merubah data yang diinginkan saja tanpa perlu memasukkan semua informasi dari awal.



Gambar 5. 15 Halaman Edit Data

1. Halaman Upload Dokumen Baru

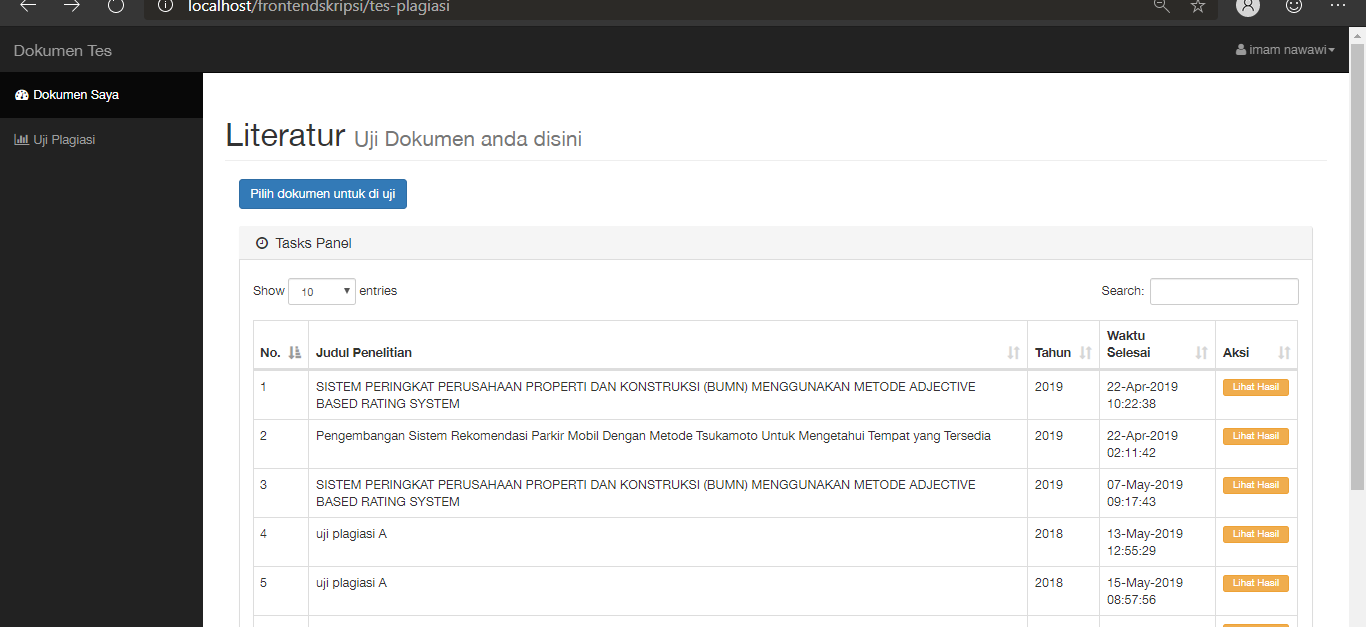
Halaman *upload* dokumen baru merupakan halaman yang berisi *form* untuk menambah dokumen baru yang ditunjukkan pada gambar 5.16. Pada halaman ini *form* upload memiliki komponen antara lain judul, abstrak, kata kunci, file yang ingin di *upload*, sedangkan untuk tahun terisi secara otomatis dari pengaturan 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

1. Halaman Uji Plagiasi

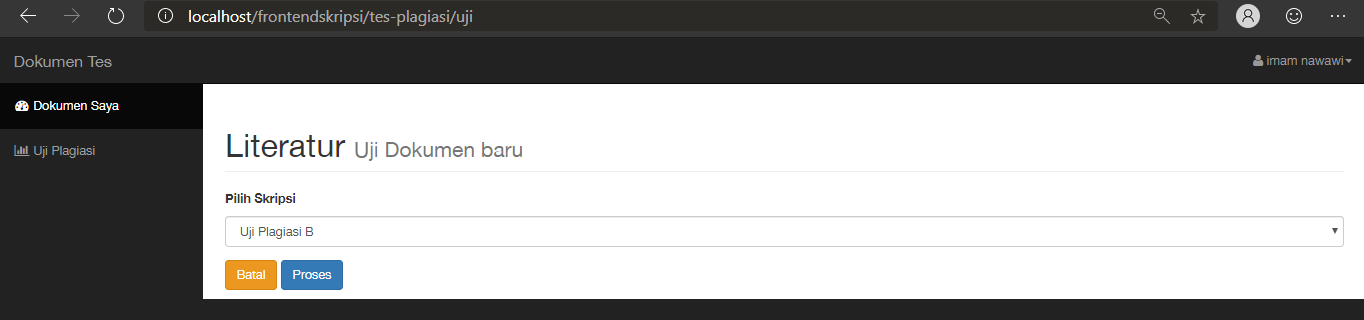
Halaman Uji Plagiasi merupakan halaman yang berisi data-data dokumen yang pernah dilakukan pengujian. Pada halaman ini terdapat fitur pilih dokumen untuk diuji, lihat hasil. Pada saat halaman ini di buka mahasiswa akan diarahkan langsung menuju daftar dokumen yang pernah diuji dimana fitur ini berfungsi untuk menampilkan semua data dari dokumen yang pernah diuji dan sudah tersimpan di *database*. Data tersebut antara lain judul penelitia, 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.



Gambar 5. 17 Halaman Uji Plagiasi

1. Halaman Pilih Dokumen Yang Diuji

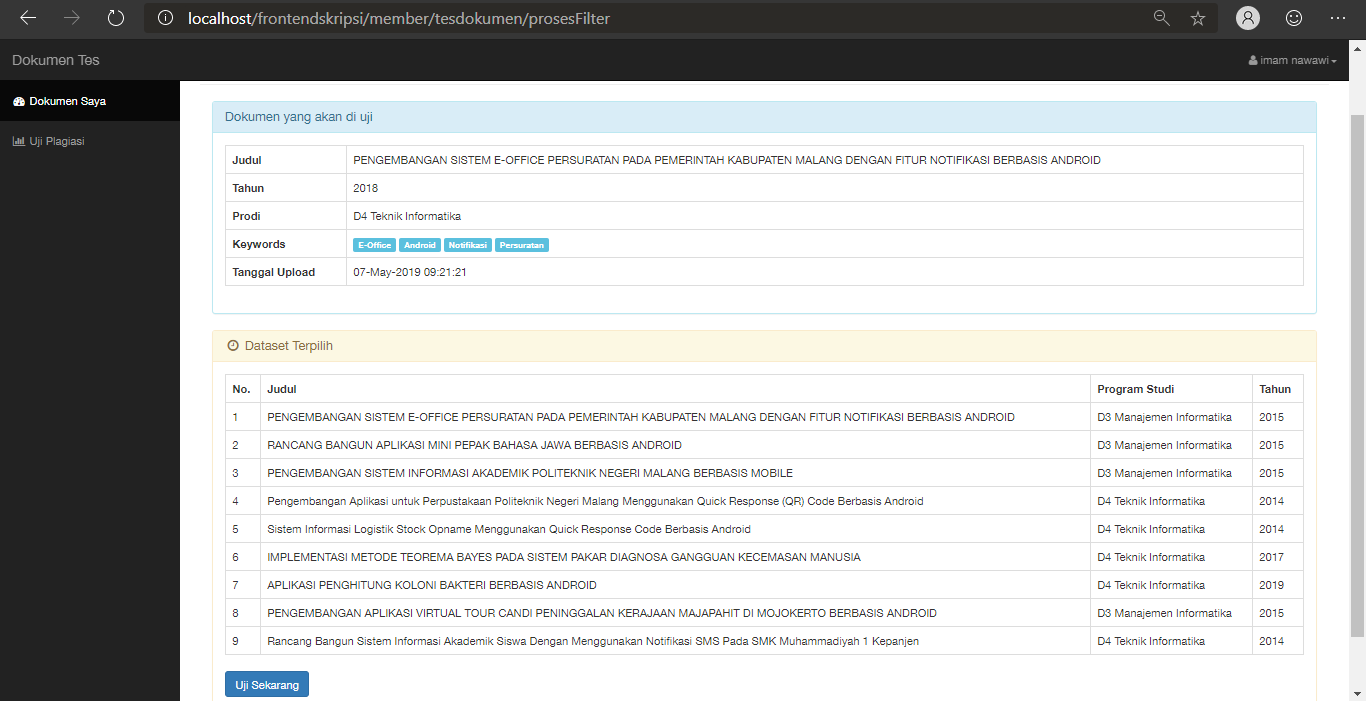
Halaman pilih dokumen yang diuji merupakan halaman yang berisi data-data dokumen mahasiswa yang tersimpan pada *database*. Pada halaman ini terdapat fitur pilihan dokumen untuk diuji, yang dapat dilihat pada gambar 5.18. ketika mahasiswa telah menentukan dokumen pilihan, kemudian menekan tombol proses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



Gambar 5. 18 Halaman Pilih Dokumen Uji

1. Halaman Hasil Filter Kandidat Pembanding

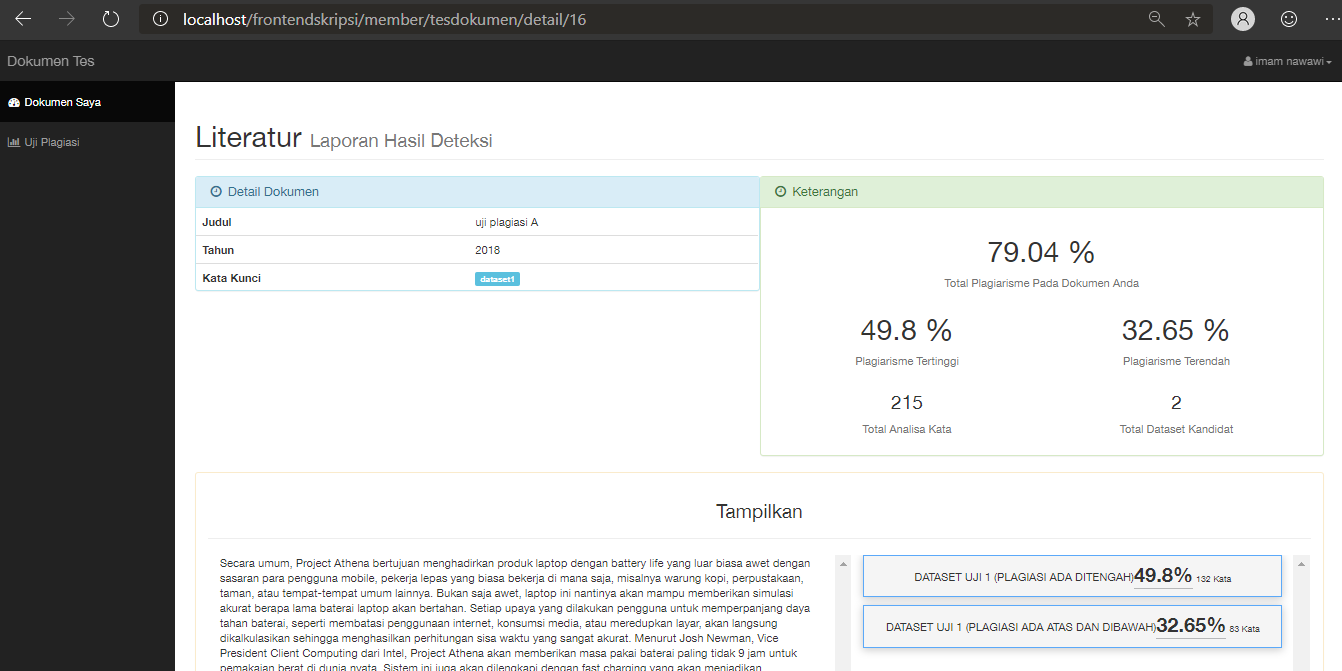
Halaman hasil *filter* kandidat pembanding merupakan halaman yang berisi data-data dari dokumen uji dan kandidat pembanding dari hasil proses *filter kandidat pembanding* yang ditunjukkan pada gambar 5.18. Pada halaman ini mahasiswa tekan tombol uji sekarang untuk memupai proses uji yang ditunjukkan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



Gambar 5. 19 Halaman Hasil Filter Kandidat pembanding

1. 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.



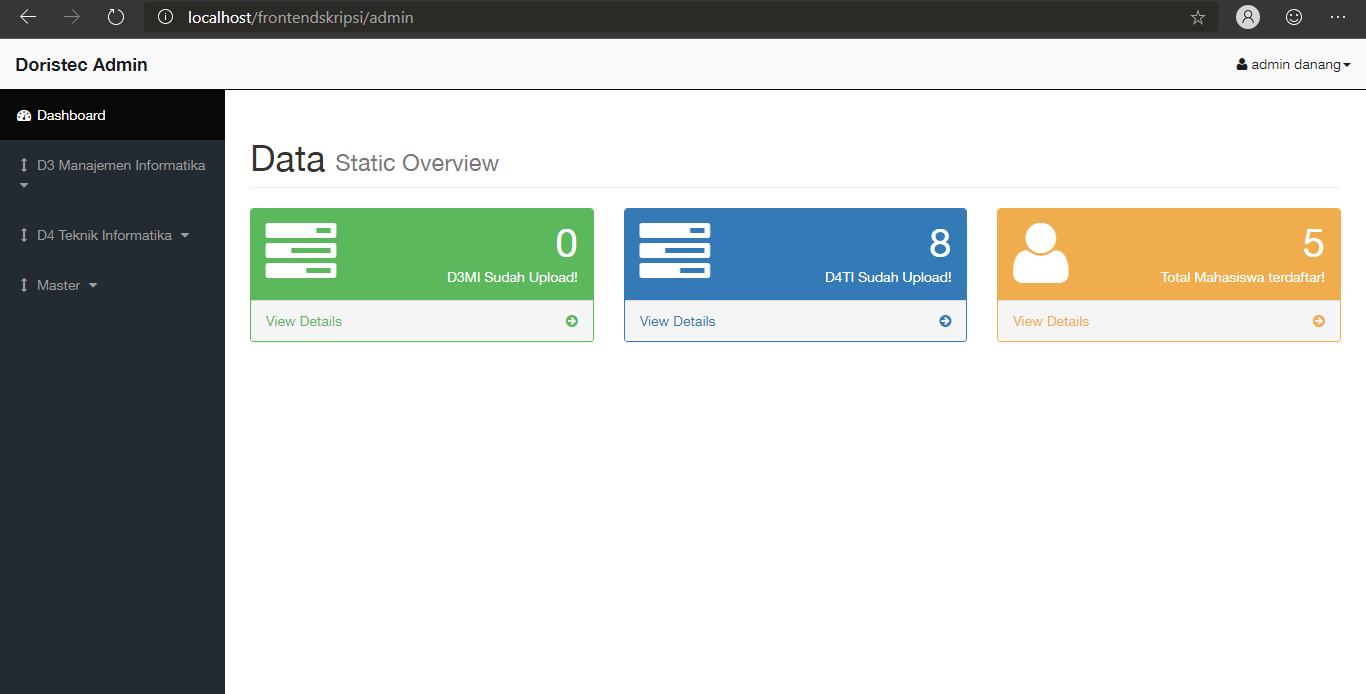
Gambar 5. 20 Halaman Hasil Uji

1. 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 startbbootstrap-admin custom.*

1. Halaman Dashboard

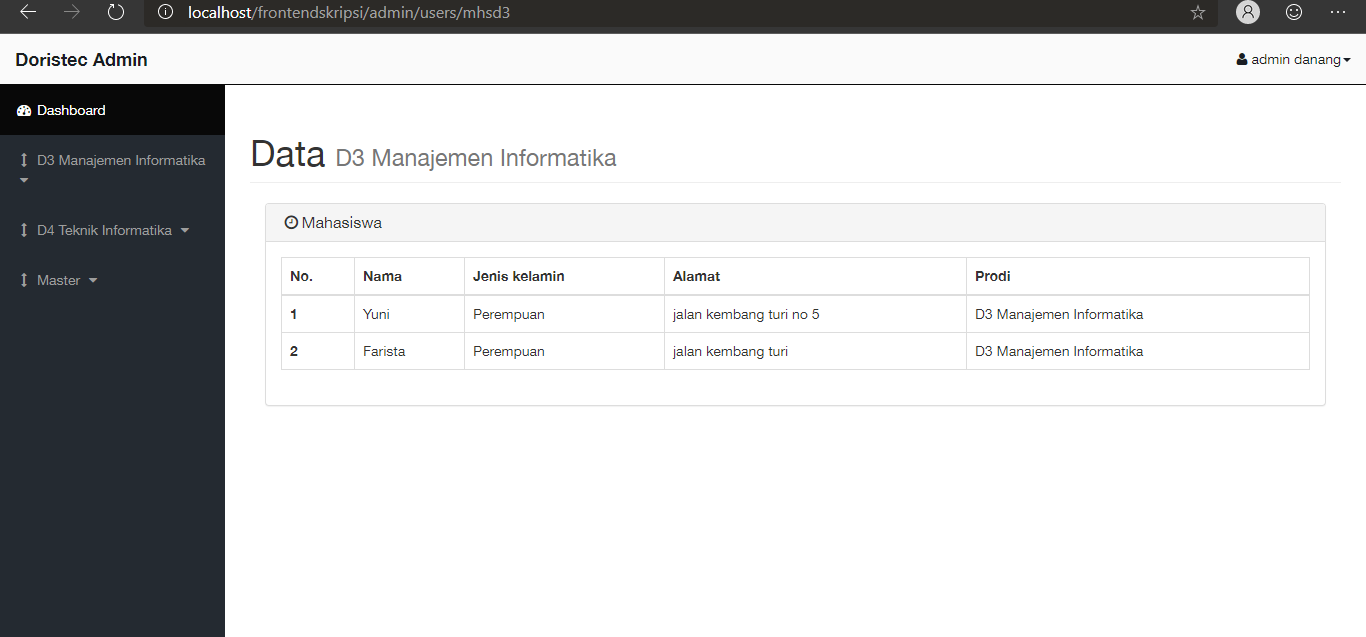
Halaman *dashboard* admin panitia merupakan halaman awal yang ditampilkan dari ketika *login* berhasil dilakukan. Pada halaman ini terdapat fitur mahasiswa D3 Manajemen Informatika, D3 yang sudah *upload* dokumen, mahasiswa D4 Teknik Informatika, D4 yang sudah *upload* dokumen, daftar *dataset*, *upload* dokumen pembandingbaru 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

1. Halaman Mahasiswa D3 Manajemen Informatika

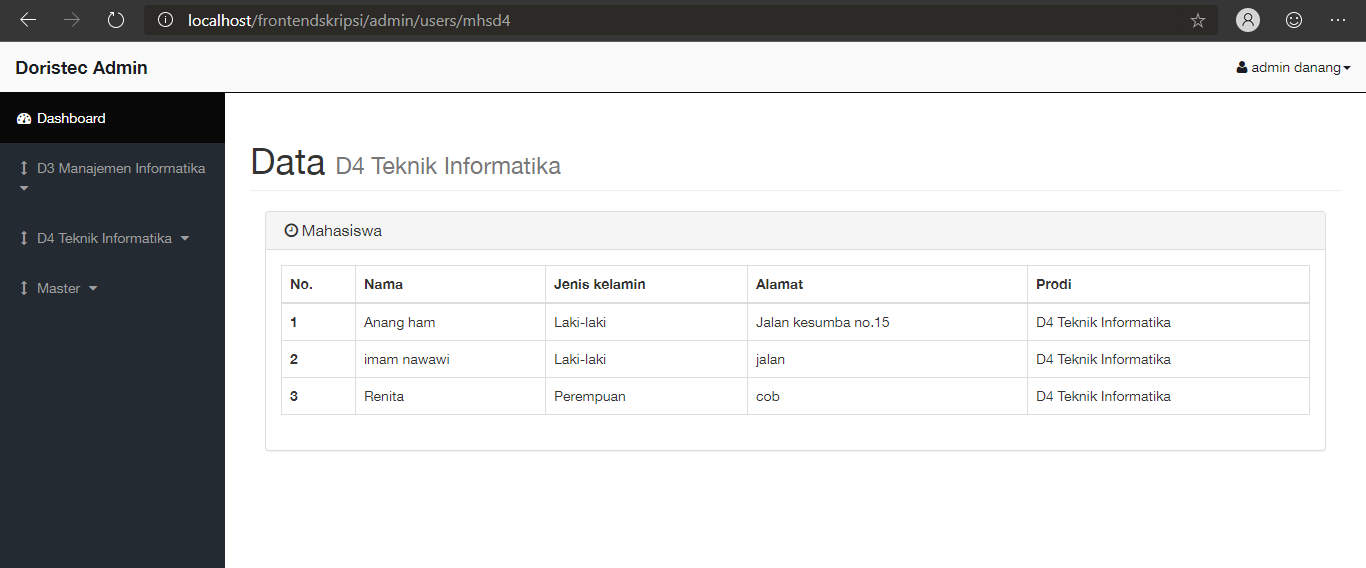
Halaman mahasiswa D3 manajemen informatika merupakan halaman yang berisi data-data mahasiswa D3 yang telah mendaftar pada aplikasi doristec. Pada saat halaman ini di buka panitia akan diarahkan langsung menuju daftar mahasiswa D3 yang pernah mendaftar. Data tersebut antara lain nama, jenis kelamain,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



Gambar 5. 22 Halaman Mahasiswa D3 Manajemen Informatika

1. Halaman Mahasiswa D4 Teknik Informatika

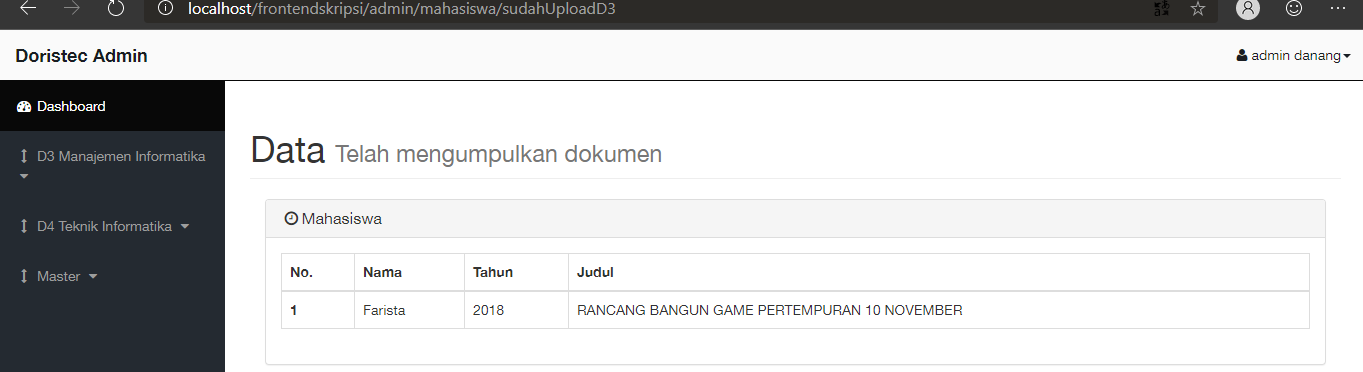
Halaman mahasiswa D4 teknik informatika merupakan halaman yang berisi data-data mahasiswa D4 yang telah mendaftar pada aplikasi doristec. Pada saat halaman ini di buka panitia akan diarahkan langsung menuju daftar mahasiswa D4 yang pernah mendaftar. Data tersebut antara lain nama, jenis kelamain, alamat,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



Gambar 5. 23 Halaman mahasiswa D4 teknik informatika

1. Halaman Mahasiswa D3 Yang Sudah Upload

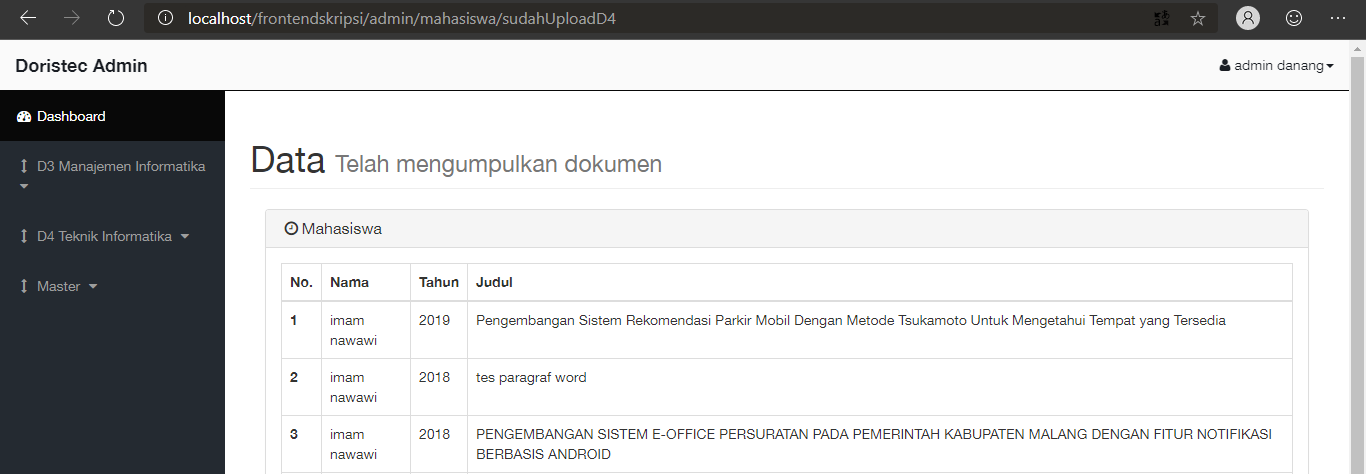
Halaman mahasiswa D3 yang sudah *upload* merupakan halaman yang berisi data-data mahasiswa D3 yang telah *upload* dokumen tugas akhir pada aplikasi doristec. Pada saat halaman ini di buka panitia akan diarahkan langsung menuju daftar mahasiswa D3 beserta dokumen tugas akhir yang telah di *upload*. Data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



Gambar 5. 24 Halaman mahasiswa D3 yang sudah upload

1. 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.

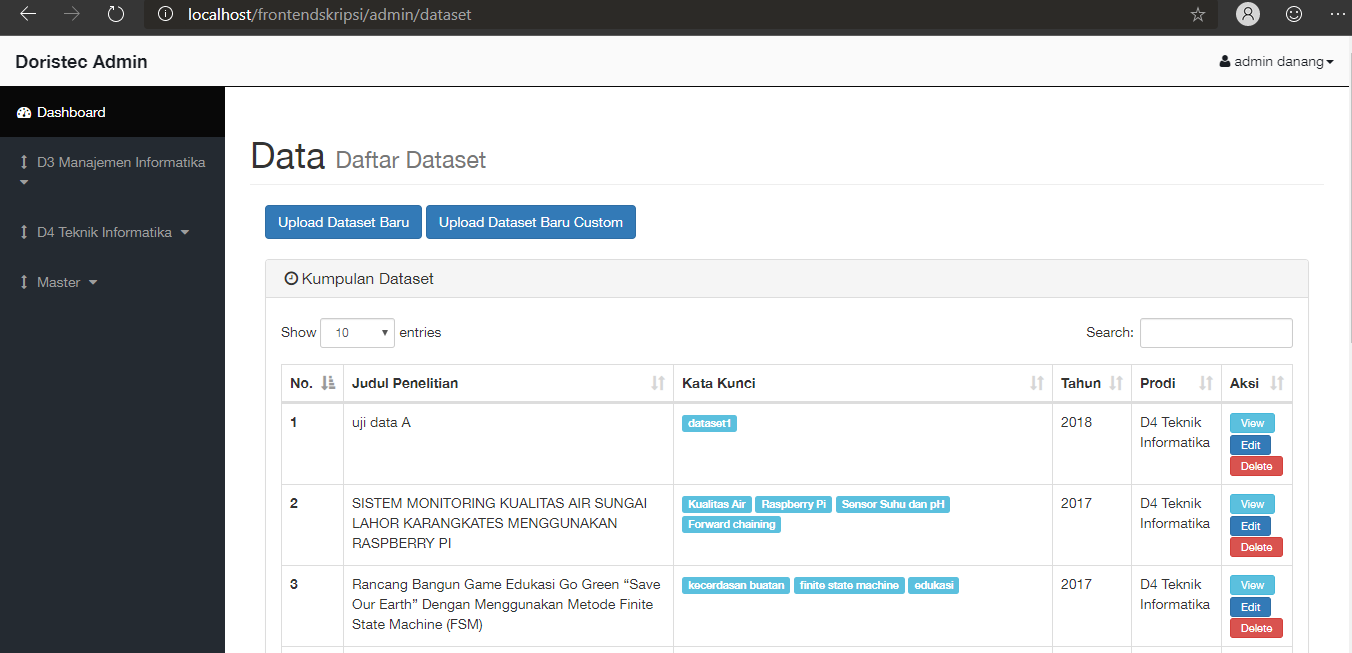


Gambar 5. 25 Halaman mahasiswa D4 yang sudah upload

1. Halaman Dataset

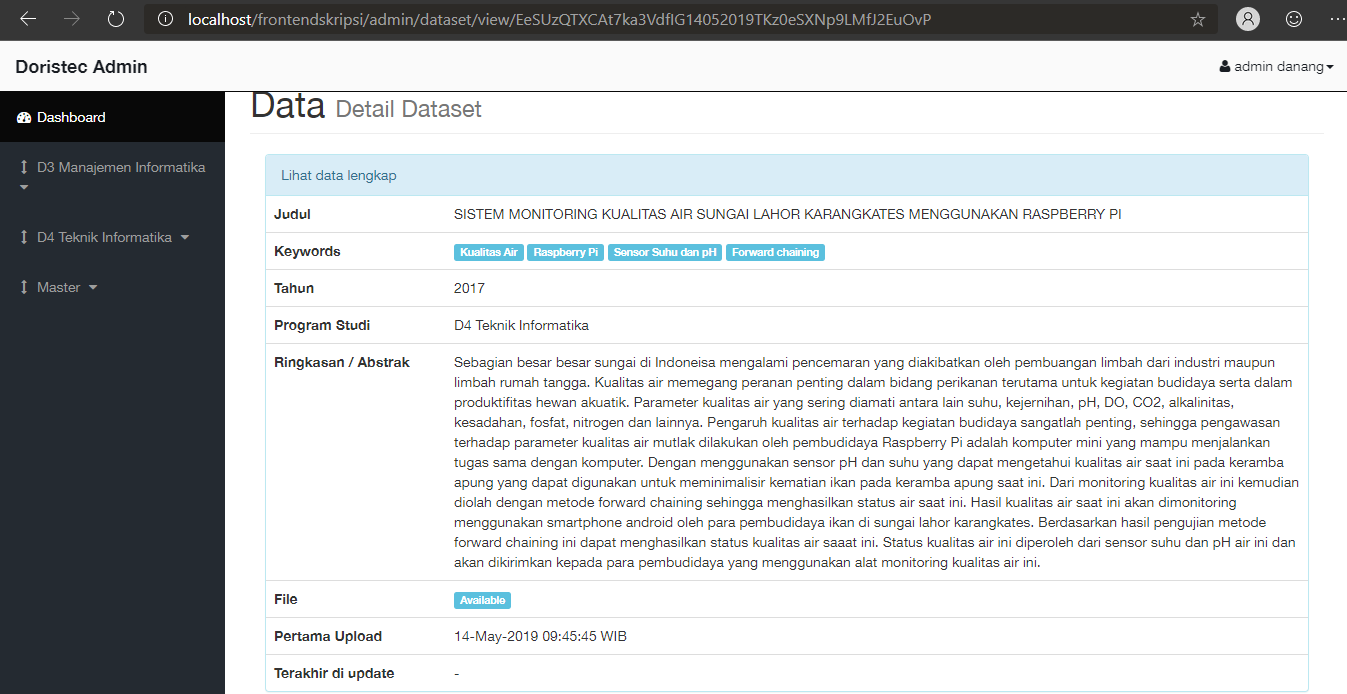
Halaman *dataset* merupakan halaman yang berisi data-data yang telah tersimpan pada *database,* data ini berupa laporan tugas akhir dan laporan skripsi yang digunakan sebagai *dataset* aplikasi ini. Fitur-fitur yang terdapat pada aplikasi ini berfungsi untuk menampilkan dokumen pembanding yang pernah 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.



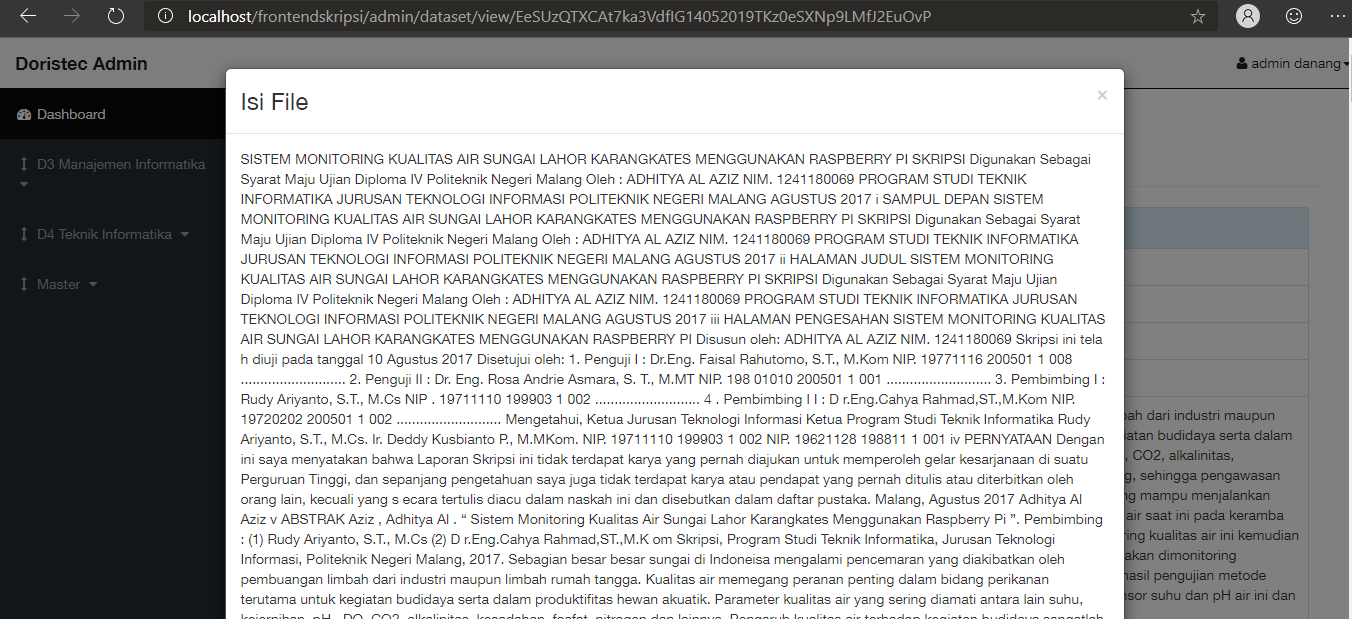
Gambar 5. 26 Halaman dataset

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.



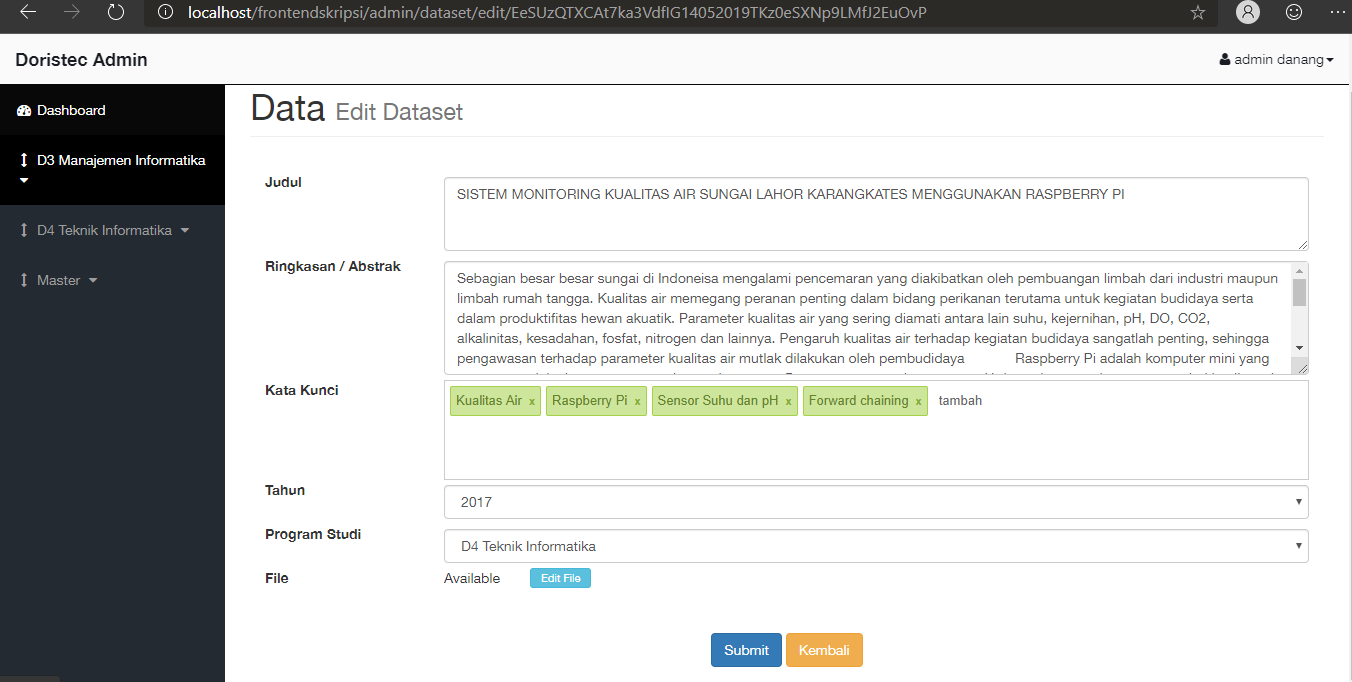
Gambar 5. 27 Halaman Lihat Data Lengkap

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

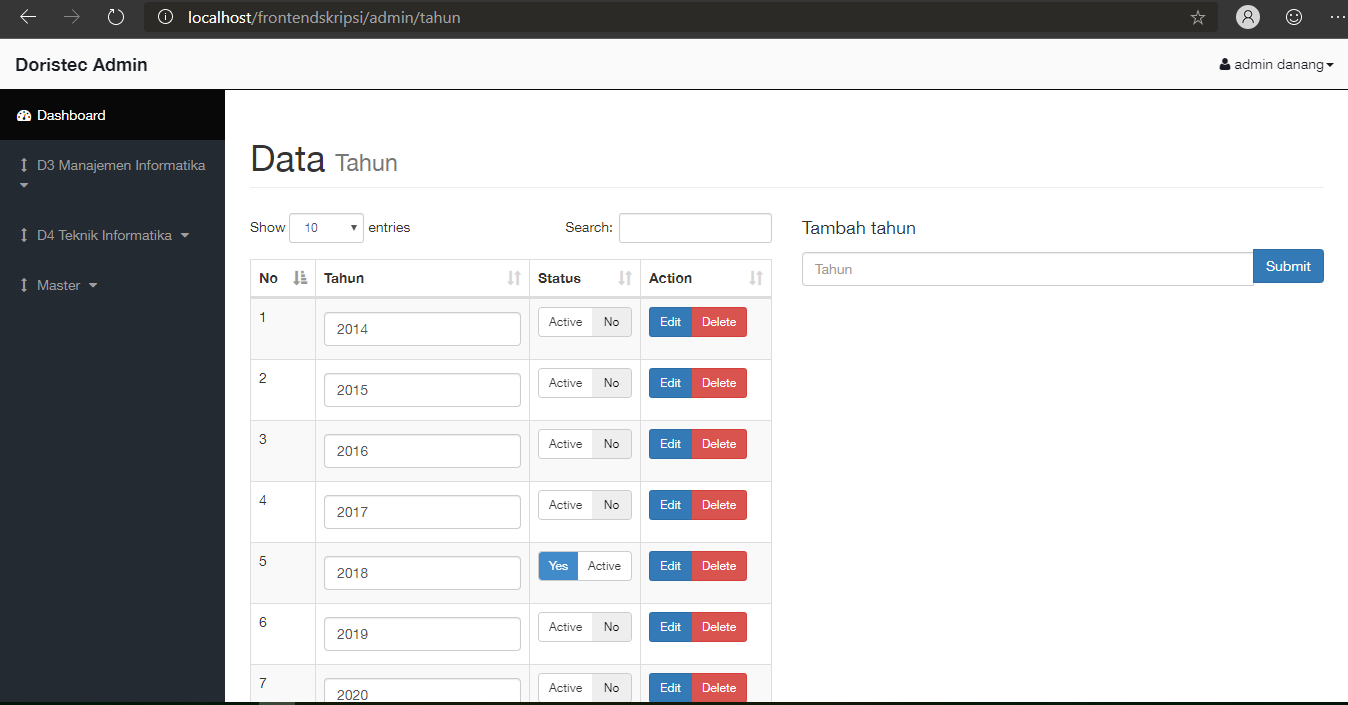
Ketika tombol edit ditekan, maka informasi yang akan dirubah akan tampil dihalaman input data seperti ditunjukkan gambar 5.29 namun dalam keadaan form yang sudah terisi. Panitia cukup merubah data yang diinginkan saja tanpa perlu memasukkan semua informasi dari awal.



Gambar 5. 29 Halaman Edit Data

1. 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 tahunsecara 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

1. 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.

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 *input* pada *form* registrasi | Data pengguna dapat diinputkan | Proses berjalan dengan normal | Berhasil |
| 2 | Melakukan *input* pada *form login* | Dapat masuk pada halaman admin panel | Proses berjalan normal. Ketika tombol *login* 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 *database* | 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 *submit* ditekan | Berhasil |
| 4 | Menghapus dokumen | Melakukan hapus data pada *database* | Proses berjalan dengan normal | Berhasil |
| 5 | *Upload* dokumen baru | Menambah dokumen baru dan disimpan ke *database* | Proses berjalan dengan normal, ketika tombol *submit* 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 | *Filter* Kandidat Pembanding | Pengambilan kandidat pembanding berdasarkan kata kunci | Proses berjalan dengan normal, ketika tombol proses ditekan | Berhasil |
| 9 | Menampilkan data hasil *filter* | Menampilkan hasil *filter* 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 *dashboard,* 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 *database* | Proses berjalan dengan normal | Berhasil |
| 8 | Melakukan *input* pada *form* 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 *edit* ditekan | Berhasil |
| 10 | Menghapus tahun | Melakukan hapus data pada *database* | Proses berjalan dengan normal | Berhasil |
| 11 | Menampilkan data semua data pada *dataset* | Menampilkan semua data pada *dataset* dari *database* | 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 *submit* ditekan | Berhasil |
| 14 | Menghapus data pada *dataset* | Melakukan hapus data pada *database* | Proses berjalan dengan normal | Berhasil |
| 15 | *Upload* dokumen pembanding baru | Menambah dokumen pembanding baru dan disimpan ke *database,* namun program studi, tahun tidak dapat dipilih | Proses berjalan dengan normal, ketika tombol *submit* ditekan | Berhasil |
| 16 | *Upload* dokumen pembanding baru *custom* | Menambah dokumen pembanding baru dengan *input* program studi, tahun dapat dipilih | Proses berjalan dengan normal, ketika tombol *submit* ditekan | Berhasil |

Lanjutan Tabel 5. 4 Pengujian Sistem Pada Level Panitia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario** | **Hasil yang diharapkan** | **Hasil yang terjadi** | **Keterangan** |
|  |  | dan disimpan ke *database* |  |  |
| 17 | Menampilkan hasil dokumen yang telah diuji | Menampilkan hasil dokumen mahasiswa yang telah diuji | Proses berjalan dengan normal | Berhasil |
| 18 | Membuat *ranking* 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 |

1. 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 :

1. 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
2. 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
3. Melakukan pencarian kata dengan urutan kata terpanjang mengunakan rumus yang ditunjukkan pada gambar 5.31 berikut ini.



Gambar 5. 31 Rumus LCS

1. Simpan hasil pencarian kata terpanjang pada variabel
2. Menghitung prosentase plagiarisme pada setiap kandidat pembanding
3. 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.
4. Hasil disimpan pada *database*.

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

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 *battery life*yang luar biasa awet dengan sasaran para pengguna *mobile*, 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-tempatumum 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 *battery life*yang luar biasa awet dengan sasaran para pengguna *mobile*, 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 membatassi 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 |

1. Kasus Kedua

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

Tabel 5. 7 Kasus Kedua Pengujian Metode

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Isi Dokumen** | **Jumlah Kata** | **Keterangan** |
| 1 | Secara umum, Project Athena bertujuan menghadirkan produk laptop dengan battery life yang luar biasa awet dengan sasaran para pengguna mobile, 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, Vice President Client Computing 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 fast charging yang akan menjadikan pengguna mobile 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 tuning 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 sleep.  Intel juga sudah mengambil langkah lebih jauh dengan bekerja bersama para pembuat komponen komputer untuk menghadirkan perangkat Project Athena generasi pertama. | 262 Kata | Dokumen Uji |

Lanjutan Tabel 5. 7 Kasus Kedua Pengujian Metode

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Isi Dokumen** | **Jumlah Kata** | **Keterangan** |
|  | Jika perangkat prototype 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 pembanding 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 pembanding 2 |

Lanjutan Tabel 5. 7 Kasus Kedua Pengujian Metode

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Isi Dokumen** | **Jumlah Kata** | **Keterangan** |
|  | 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 tuning 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. |  |  |
| 4 | 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.  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 *tuning*yang tepat terhadap keseluruhan komponen laptop. Penyesuaian ini juga akan dibantu oleh sistem A.I, | 156 Kata | Kandidat pembanding 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 |

1. 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** |
|  | 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.  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. |  |  |
| 2 | Secara umum, Project Athena bertujuan menghadirkan produk laptop dengan battery life yang luar biasa awet dengan sasaran para pengguna mobile, 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.  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.  Intel juga sudah mengambil langkah lebih jauh dengan bekerja bersama para pembuat komponen komputer untuk menghadirkan perangkat Project Athena generasi pertama. | 198 Kata | Kandidat pembanding 1 |

Lanjutan Tabel 5. 9 Kasus Ketiga Pengujian Metode

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Isi Dokumen** | **Jumlah Kata** | **Keterangan** |
|  | Dengan kata lain, Microsoft ingin meletakkan semua platform pada versi yang sama dari PowerShell untuk kompatibilitas lintas-platform yang lebih baik. Jika perangkat prototype tersedia, Intel nantinya juga akan membuka diri terhadap manufaktur PC, sehingga lebih banyak manufaktur yang dapat memanfaatkan teknologi Intel ini. |  |  |
| 3 | Sistem ini juga akan dilengkapi dengan *fast charging*yang akan menjadikan pengguna *mobile*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 *tuning*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 *sleep.*  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 *prototype* tersedia, Intel nantinya juga akan membuka diri terhadap manufaktur PC, sehingga lebih banyak manufaktur yang dapat memanfaatkan teknologi Intel ini. | 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 pembanding1 | 198 Kata | 23 kata, 18 kata, 6 kata | 3 | 47 kata |
| 3 | Kandidat pembanding2 | 180 kata | 9 kata, 6 kata, 5 kata, 4 kata | 4 | 24 kata |