**BAB V**

**HASIL DAN PENGUJIAN**

**5.1 Pengujian *blackbox***

Teknik pengujian *blackbox* berfokus pada domain informasi dari perangkat lunak, dengan melakukan *test case* dengan mempartisi domain *input* dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Metode ujicoba ini juga memfokuskan pada keperluan fungsional dari software, karena itu ujicoba *blackbox* memungkinkan pengembang software untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program.

Adapun rencana pengujian yang akan di uji dengan teknik pengujian *black box* adalah sebagai berikut :

Tabel 5. 1 rencana pengujian

| No | Komponen Sistem yang di uji | Butir uji | Jenis Pengujian |
| --- | --- | --- | --- |
| 1 | Halaman *Login* | *Input Email* dan *Password* yang benar | *Black box* |
| *Input Email* dan *password* yang salah | *Black box* |
| 2 | Menu Utama admin | Tambah Komisi | *Black box* |
| Tambah data training Komisi | *Black box* |
| Menambah *User* Komisi | *Black box* |
| 3 | Menu Utama *User* Komisi | Melihat daftar aspirasi | *Black box* |
| Melihat detail aspirasi | *Black box* |
| Menanggapi aspirasi | *Black box* |
| 4 | Edit Profile | *Form Update* *Profile* | *Black box* |
| 5 | Ganti *password* | Mengisi *Form update* *password* yang benar | *Black box* |
| Mengisi *form update password* yang salah | *Black box* |
| 6 | Masyarakat Register | Mengisi *Form Register* Masyarakat yang benar | *Black box* |
| Mengisi *form register* yang salah | *Black box* |
| 7 | Masyarakat *input* aspirasi | Proses *Text mining* dan *cosine similarity* | *Black box* |
| 8 | Masyarakat membalas tanggapan | Mengisi *form* balas tanggapan dan mengirim ke komisi penanggap | *Black box* |

**5.2 Pengujian Fungsional**

Pada bagian ini di bahas mengenai pengujian program, proses- proses pada masing- masing modul program berdasarkan dari rencana pengujian yang di bahas sebelumnya:

Tabel 5. 2 Pengujian Sistem Web Admin

|  |  |  |  |
| --- | --- | --- | --- |
| Pengujian | Hasil | | Keterangan |
| Halaman Login Admin | ✓ | | Berhasil |
| Screen Shoot | | | |
| User Login dengan memasukkan password yang benar akan menuju ke menu berikutnya | | Menampilkan pesan sukses jika username dan password benar | |
| Untuk login kami menggunakan javascript dan api :  loadFileJs(baseUrl + "node\_modules/sweetalert2/dist/sweetalert2.min.js", "url");  addCss(baseUrl + "node\_modules/sweetalert2/sweetalert2.min.css", "url");  addCss("layout/user/css/style.css");  let formLogin = document.querySelector("#form-login");  formLogin.addEventListener("submit", async function(e) {  e.preventDefault();  let email = formLogin.querySelector("[name='email'").value;  let password = formLogin.querySelector("[name='password'").value;  try {  let formdata = new FormData();  formdata.append("email", email);  formdata.append("password", password);  Swal.showLoading();  let requestOptions = {  method: "POST",  body: formdata  };  let login = await getData("api/user/login", requestOptions);  await Swal.fire({  position: "top-center",  icon: "success",  title: "Login Success",  showConfirmButton: false,  timer: 1500  });  location.replace(baseUrl + "admin/permition/auth?id=" + login.id\_user);  } catch (error) {  Swal.fire({  icon: "error",  title: "Oops...",  // onBeforeOpen: () => {  // Swal.showLoading();  // },  html: error  });  }  }); | | | |
| Mengolah komisi | ✓ | | Berhasil |
| Screen Shoot | | | |
| Menambah Komisi | | Pessan sukses berhasil menambahkan komisi | |
| $komisi->setPrimaryKey($komisi->getLastId());  $komisi->setFieldTable($this->post);  $komisi->add();  $ks = hasilCUD("Komisi Ditambahkan");  $type = $ks->status ? "success" : "danger";  $this->session->set\_flashdata("message","<div class='alert alert-$type' role='alert'> $ks->message</div>");  redirect(base\_url('admin/komisi/home')); | | | |
| Menambah Data training | | Data yang di input berhasil di tambahkan | |
| $this->post['komisi\_id'] = $id;  /\*\*  \* proses menghilankan kata awal dan akhir  \*/  $stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();  $stemmer = $stemmerFactory->createStemmer();  $sentence = $this->input->post('label');  $outputstemmer = $stemmer->stem($sentence);  /\*\*  \* proses menghilankan kata yang tidak penting  \*/  $stopWordRemoverFactory = new \Sastrawi\StopWordRemover\StopWordRemoverFactory();  $stopword = $stopWordRemoverFactory->createStopWordRemover();  $outputstopword = $stopword->remove($outputstemmer);  $array = preg\_split('/[^[:alnum:]]+/', strtolower($outputstopword));  $arr\_textmining = [];  foreach ($array as $item) {  if (array\_key\_exists($item, $arr\_textmining)) {  $arr\_textmining[$item]++;  } else {  if (strlen($item) > 2) {  $tbl = initTable("web\_komisi\_label", "lbl");  $data = [  $tbl["key"] => $tbl['field'][$tbl["key"]],  "komisi\_id" => $this->post['komisi\_id'],  "label" => $item  ];  $this->db->insert($tbl['name'], $data);  $arr\_textmining[$item] = 1;  $respon = hasilCUD("Sukses Menambahkan Label");  }  }  }  redirect('admin/komisi/label/' . $id); | | | |
| Memilih menu daftar aspirasi | | Menampilkan daftar aspirasi | |
| $().ready(function() {  $.ajax({  url: baseUrl + "api/aspirasi",  type: "get",  dataType: "json",  beforeSend: () => {  $("thead.load").after("<div class='mdlProgresBarr'></div>");  },  success: respon => {  if (respon.status) {  let Urut = 1;  respon.data.forEach(e => {  let tr = `  <tr id="${e.id\_aspirasi}">  <th scope="row">${Urut}</th>  <td name='username'>${e.username}</td>  <td name='message'>${e.message}</td>  <td name='komisi\_id'>${e.komisi}</td>  <td>  <a href="#" class="badge badge-primary mdl btn" data-komisi="${e.komisi\_id}" data-message="${e.message}" data-id="${e.id\_aspirasi}" data-toggle="modal" data-target=".modal#edit" id="edit">Edit</a>  <a href="" class="mdl badge badge-danger" id="delete" data-id="${e.id\_aspirasi}" data-toggle="modal" data-url="${baseUrl}api/aspirasi?id\_aspirasi=${e.id\_aspirasi}" data-target=".modal#logout">delete</a>  </td>  </tr>  `;  dftAspirasi.append(tr);  Urut++;  });  } else {  dftAspirasi.html(`<tr><td col=5>${respon.message}</td></tr>`);  }  },  error: e => {  console.log(e);  },  complete: () => {  $(".mdlProgresBarr").remove();  }  });  }); | | | |
| Menambah User pada komisi | | Menampilkan pesan user berhasil ditambahkan | |
| $post = $this->input->post();  $image = "default.png";  if (isset($\_FILES['image'])) {  $config['allowed\_types'] = 'gif|jpg|jpeg|png';  $config['max\_size'] = '3048';  $config['upload\_path'] = './assets/img/profile/';  $this->load->library('upload', $config);  if ($this->upload->do\_upload('image')) {  $image = $this->upload->data('file\_name');  /\*\*  \* Helper CI  \* Create thumb image with  \* \_img\_create\_thumbs({nama file image}, {Folder file gambar});  \*/  \_img\_create\_thumbs($image, "profile");  }  }  $tblUser = initTable("tbl\_user", "user");  $tblUser['lastkey'] = $tblUser['field'][$tblUser['key']];  $data = [  $tblUser['key'] => $tblUser['lastkey'],  "name" => $post['name'],  "email" => $post['email'],  "no\_hp" => $post['no\_hp'],  "alamat" => $post['alamat'],  "role\_id" => $this->input->post('role\_id') ? $this->input->post('role\_id') : 3,  "is\_active" => $this->input->post('is\_active') ? $post['is\_active'] : 0,  "image" => $image,  "password" => password\_hash($post['password'], PASSWORD\_DEFAULT)  ];  $this->db->insert($tblUser['name'], $data);  $tbl = initTable("web\_komisi\_user", "kom\_u");  $this->db->insert($tbl['name'], [  $tbl['key'] => $tbl['field'][$tbl['key']],  "komisi\_id" => $this->input->post("komisi\_id"),  "dapil\_id" => $this->input->post("dapil\_id"),  "user\_id" => $data["id\_user"],  "jabatan" => $this->input->post("jabatan"),  ]);  $respon = hasilCUD("Data User Ditambahkan");  $type = $respon->status ? "success" : "danger";  $this->session->set\_flashdata('message', '<div class="alert alert-' . $type . '" role="alert">' . $respon->message . '!</div>'); | | | |

Tabel 5. 3 Sistem web komisi

|  |  |  |  |
| --- | --- | --- | --- |
| Pengujian | Hasil | | Keterangan |
| Halaman Utama user komisi | ✓ | | Berhasil |
| Screen Shoot | | | |
| Memilih daftar aspirasi | | Menampilkan daftar aspirasi | |
| $this->data['page']['title'] = 'Daftar Aspirasi Yang Masuk';  $this->data['page']['description'] = 'Silahkan edit menu Atau Tambahkan Komisi.!';  // $this->data['page']['before'] = ['url' => base\_url('admin/admin/role'), "title" => "Role Access"];  $this->data['page']['submenu'] = 'Menu Komisi';  $this->db->select("web\_komisi\_user.\*,web\_komisi.name");  $this->db->from("web\_komisi\_user");  $this->db->join("web\_komisi", "web\_komisi.id\_komisi=web\_komisi\_user.komisi\_id");  $this->db->where("web\_komisi\_user.user\_id", $this->data['user']['id\_user']);  $eks = $this->db->get()->row\_array();  $this->data['komisi'] = $eks;  $this->template->load('admin', 'aspirasi/komisi', $this->data); | | | |
| Mengklik tombol lihat | | Menampilkan detail aspirasi | |
| let elAdd = document.querySelector("[data-name='evAdd']");  elAdd.addEventListener("click", e => {  dataModal.title = "Tambah User";  elMdlUser  .querySelector("form")  .setAttribute("action", baseUrl + "admin/komisi/user/add");  elMdlUser.querySelector("[data-name='title'").innerHTML = dataModal.title;  e.preventDefault();  dataModal.urlU = "api/user";  dataModal.methodU = {  method: "POST",  body: new FormData()  };  dataModal.urlK = "api/komisi/user";  dataModal.methodK = {  method: "POST",  body: new FormData()  };  let input = Array.from(elMdlUser.querySelectorAll("input"));  input.forEach(res => {  if (res.getAttribute("type") != "submit") {  if (res.getAttribute("name") == "is\_active") res.value = 1;  let paren = res.closest(".form-group");  if (paren.classList.contains("is-filled")) {  paren.classList.remove("is-filled");  res.value = "";  }  }  });  let select = Array.from(elMdlUser.querySelectorAll("select"));  select.forEach(res => {  res.querySelector("option").selected = "selected";  });  }); | | | |
| Mengisi form tanggapan dan kirim tanggapan | | Muncul pesan sukses menanggapi | |
| $("#formInput").on("submit", function(e) {  e.preventDefault();  var myForm = $(this);  let url = myForm.attr("action");  var data = $(this).serializeArray(); //serialize form inputs and pass them to php  $.ajax({  url: url,  type: "PUT",  data: data,  dataType: "json",  beforeSend: () => {  myForm.after("<div class='mdlProgresBarr'></div>");  $(".fullpage").addClass("tampil");  },  success: dt => {  let userId = $('input[name="user\_id"]').val();  let nameKomisi = $("select[name='komisi\_id'] option:selected").text();  if (dt.status) {  \_aspirasi  .children("td[name=komisi\_id]")  .text($("#komisi\_id option:selected").text());  \_aspirasi  .children("td[name=message]")  .text($("textarea[name=message]").val());  }  status = dt.status;  message = dt.message;  },  complete: function() {  $("head").append("<style>.mdlProgresBarr::before{width:100%;}</style>");  $(".mdlProgresBarr").remove();  $(".modal").modal("hide");  let css = status ? "success" : "danger";  let notif = `<div class="alert alert-${css}">  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <i class="material-icons">close</i>  </button>  <span>${message}</span>  </div>`;  $(".fullpage").hide("slow", function() {  $(".fullpage").remove();  });  $(".notif#ajax").html(notif);  $(".main-panel").animate(  {  scrollTop: $(".notif#ajax").offset().top  },  "slow"  );  }  });  }); | | | |
| Mengedit profile | | Data profil berhasil berubah | |
| $status = false;  $upload\_image = $\_FILES['image']['name'];  $post = $this->input->post();  /\*\*  \* Change format Time  \*/  list($m, $d, $y) = explode("/", $post['tgl\_lahir']);  $date = "{$d}-{$m}-{$y}";  $timestamp = strtotime($date);  $post['tgl\_lahir'] = $timestamp;  $fileName = $this->data['user']['image'];  if ($upload\_image) {  $config['allowed\_types'] = 'gif|jpg|jpeg|png';  $config['max\_size'] = '3048';  $config['upload\_path'] = './assets/img/profile/';  $this->load->library('upload', $config);  if ($this->upload->do\_upload('image')) {  // cretate thumbnail image  if ($fileName != "default.png") //file default  {  if (is\_file(FCPATH . 'assets/img/profile/' . $fileName))  unlink(FCPATH . 'assets/img/profile/' . $fileName);  if (is\_file(FCPATH . 'assets/img/thumbnail/profile\_' . $fileName))  unlink(FCPATH . 'assets/img/thumbnail/profile\_' . $fileName);  }  $fileName = $this->upload->data('file\_name');  \_img\_create\_thumbs($fileName, "profile");  $msg[] = "Foto Profil Updated";  } else {  $msg[] = $this->upload->display\_errors();  }  }  $post['image'] = $fileName;  $this->db->update("tbl\_user", $post, ['id\_user' => $this->data['user']['id\_user']]);  $eks = hasilCUD("Sukses Update");  $msg[] = $eks->message;  setNotif($eks->status, $msg);  header("Refresh:0"); | | | |
| Ganti password | | Pesan sukse berhasil ganti password | |
| $this->form\_validation->set\_rules('newPassword', 'Password', 'trim|required|min\_length[3]');  $this->form\_validation->set\_rules('password', 'Repeat Password', 'trim|required|min\_length[3]|matches[newPassword]');  if ($this->form\_validation->run() == false) {  $this->data['page']['title'] = 'Mengganti Password';  $this->data['page']['description'] = 'Silahkan ubah password lama anda, gunakan karakter yang susah di tebak.!';  // $this->data['page']['before'] = ['url' => base\_url('admin/menu'), "title" => "Menu Access"];  $this->data['page']['submenu'] = 'Ganti password';  $this->template->load('admin', 'user/profile/changepassword', $this->data);  } else {  $oldPassword = $this->input->post('oldPassword');  // cek kebenran password lama  $user = $this->db->get\_where("tbl\_user", ["id\_user" => $this->data['user']['id\_user']])->row\_array();  if (password\_verify($oldPassword, $user['password'])) {  $password = password\_hash($this->input->post('password'), PASSWORD\_DEFAULT);  $email = $user['email'];  $this->db->set('password', $password);  $this->db->where('email', $email);  $this->db->update('tbl\_user');  if ($this->db->affected\_rows() > 0) {  $this->session->unset\_userdata('email');  $this->session->set\_flashdata('message', '<div class="alert alert-success" role="alert">Password has been changed! Please login.</div>');  redirect('admin/auth');  } else {  $this->session->set\_flashdata('message', '<div class="alert alert-danger" role="alert">Gagal.!</div>');  }  } else {  $this->session->set\_flashdata('message', '<div class="alert alert-danger" role="alert">Password lama tidak cocok.!</div>');  redirect('admin/user/changepassword');  }  } | | | |

Tabel 5. 4 Sistem Android Masyarakat

|  |  |  |  |
| --- | --- | --- | --- |
| Pengujian | Hasil | | Keterangan |
| Masyarakt register | ✓ | | Berhasil |
| Screen Shoot | | | |
| Mengisi form register  C:\Users\TOMMY\Downloads\Screenshot_2020-02-24-13-38-03-405_com.example.dprdbottom.jpg | | Pesan sukses register  C:\Users\TOMMY\Downloads\Screenshot_2020-02-24-13-24-37-051_com.example.dprdbottom.jpg | |
| Mengisi form register yang salah | | Pesan gagal inputan | |
| Masyarakt Input Aspirasi | ✓ | | Berhasil |
| Screen Shoot | | | |
| Mengisi form input aspirasi | | Pesan sukses mengirim  C:\Users\TOMMY\Downloads\Screenshot_2020-02-24-13-32-11-081_com.example.dprdbottom.jpg | |
| Masyarakt Membalas tanggapan | ✓ | | Berhasil |
| Screen Shoot | | | |
| Mengisi form tanggapan  C:\Users\TOMMY\Downloads\Screenshot_2020-02-24-13-28-20-890_com.example.dprdbottom.jpg | | Pesan sukses menanggapi  C:\Users\TOMMY\Downloads\Screenshot_2020-02-24-13-28-35-372_com.example.dprdbottom.jpg | |

**5.2 Pengujian *White Box***

Pada bab ini akan dibahas yang berkaitan dengan penerapan metode *white box* pada sistem implementasi algoritma *text minin*  dan *cosine similarity.*

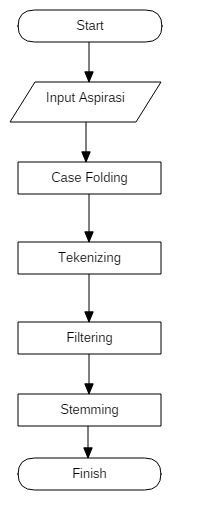
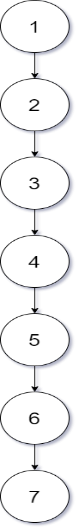
Adapun rencana pengujian yang akan dilakukan dengan metode *white box*  adalah proses penerapan algoritma *text mining* dan *cosine similarity*  yang penulis terapkan dalam penelitian ini.

**5.3.1 *Text mining***

Algoritma dalam melakukan pengimputan aspirasi dengan menggunakan *text mining,* :

1. Input aspirasi pada aplikasi *mobile*
2. *query* inputan di lakukan *case folding* untuk menghilankan karakter yang tidak penting dan mengubah semua huruf menjadi huruf kecil
3. setelah dilakukan *case folding* selanjutnya masuk ke tahap *tokenizing* untuk memotong *string query* setiap kata penyusunnya menjadi *array*
4. tahap selanjutnya adalah *filtering* yaitu untuk mengambil kata kata penting dan membuang kata kata umum yang sering di gunakan
5. tahap selanjutnya ialah tahap *stemming* untuk mencari *root* tiap pada query.

Berdasarkan algoritma diatas dibuat dalam bentuk flowchart sebagai berikut.

 **

Dari gambar diatas maka dapat ditentukan *Cyclomatic Complexity* sebagai berikut :

Path = 1-2-3-4-5-6-7

V(G) = E-N + 2

= 6 – 7 + 2

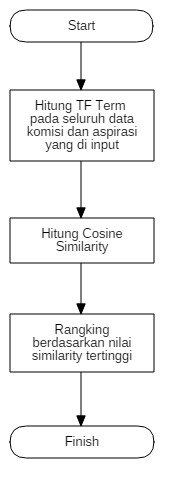
= 1

* + 1. ***Cosine Similarity***

Adapun algoritma pada tahap *cosine similarity* yaitu sebagai berikut :

1. Menghitung TF term pada seluruh data komisi dan query aspirasi yang sudah melalui tahap *text mining*
2. Kemudian dilakukan perhitungan *cosine similarity*
3. Dari hasil perhitungan *cosine similarity*  maka dilakukan peringkinan berdasarka nilai *similarity* tertinggi.

Adapun *flowhchart* dan *flowgrap* dari *cosine similarity* sebagai berikut :

** 

Dari gambar diatas maka dapat ditentukan Cyclomatic Complexity sebagai berikut :

Path = 1-2-3-4-5

V(G) = E-N + 2

= 5 – 4 + 2

= 1

**5.4 Kesimpulan hasil pengujian aplikasi**

Setelah melakukan pengujian implementasi text mining dan cosine similarity sebagai sarana aspirasi publik berbasis mobile (Studi Kasus DPRD makasar). Kesimpulan dari hasil pengujian tersebut adalah baik, dimana semua fungsi aplikasi telah berjalan dengan baik.