IMPLEMENTASI PRINSIP KERJA EXTREME PROGRAMMING

1. Planning

Pada tahap planning ini sebelumnya pengembang telah melakukan diskusi dengan *client* yang dimana pada kasus penelitian tugas akhir ini pihak kemenag NTB merupakan *client.* Didapatkan total 11 user stories dari *client* yang dapat pada tabel 3.1. dimana guna dari user stories ini yaitu untuk mendapatkan gambaran tentang system yang dibuat.

Pada tahap ini juga pengembang dan *client* telah melakukan diskusi terhadap penentuan iterasi yang terdiri dari user stories yang akan dikerjakan terlebih dahulu. Didapatkan 3 iterasi dan tiap iterasi dikerjakan selama 10 hari, sehingga secara keseluruhan rencana system akan dikerjakan selama 30 hari. Rincian penentuan iterasi tersebut dapat dilihat pada tabel 3.2., dimana guna penentuan iterasi ini yaitu untuk mengetahui prioritas user stories dan estimasi waktu pengerjaan.

Selama proses iterasi berlangsung terdapat beberapa permintaan penambahan user stories dari *client* seperti yang dirincikan pada tabel … berikut:.

Tabel penambahan user stories

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Kode US | Judul | Deskripsi | *Acceptance Criteria* | Estimasi (hari) |
|  | US-12 | Profile | Sebagai admin, dapat melihat informasi dan mengubah data diri | Terdapat halaman profile admin yang telah login ke dalam system serta dapat mengubah profile tersebut. | 5 |
|  | US-13 | Beranda | Sebagai admin, dapat melihat halaman beranda dengan total termpat ibadah | Terdapat halaman beranda untuk admin yang dapat melihat total dari masing – masing jenis tempat ibadah |  |
|  | US-14 | Import data excel tempat ibadah | Sebagai admin, dapat menambahkan data tempat ibadah dengan cara memasukkan file excel yang berisi data tempat ibadah, agar mempermudah penginputan data dalam jumlah yang banyak | Terdapat form untuk menginputkan data dalam bentuk file excel, dan pemilihan kabupaten. | 5 |
|  | US-15 | Unduh template format excel import data | Sebagai admin, dapat mendownload format excel untuk import data tempat ibadah, sehingga dapat menyesuaikan kolom – kolom masukan data yang diinginkan sistem | Terdapat tombol untuk mengunduh format data tempat ibadah dalam bentuk file excel | 2 |
|  | US-16 | Export data excel tempat ibadah | Sebagai admin, dapat melakukan export excel, sehingga dapat melakukan pembacaan data tempat ibadah yang ada dalam bentuk *hardcopy* | Terdapat tombol untuk mengexport data tempat ibadah dalam bentuk file excel. | 3 |
|  | US-17 | Melihat peta lokasi tempat ibadah pada basis *web* | Sebagai masyarakat, dapat melihat peta lokasi tempat ibadah pada basis web, namun terdapat filter kabupaten yang ada di provinsi NTB. Sehingga masyarakat juga mengetahui peta lokasi yang ada di kabupaten selain di NTB | Terdapat peta lokasi tempat ibadah pada basis web, yang tidak hanya dapat menampilkan di kabupaten Kota Mataram, melainkan di kabupaten yang ada pada provinsi NTB juga. | 7 |
|  | US-18 | Melihat rute lokasi tempat ibadah pada basis *web* | Sebagai masyarakat, dalam melihat rute lokasi menuju tempat ibadah pada basis web | Terdapat tombol rute menuju lokasi tempat ibadah | 3 |

Penambahan user tories pada tabel 3 tersebut menyebabkan pengembang bersama dengan *client* melakukan perencanaan ulang terhadap penyesuaian pengerjaan tiap itetari. hasil perencanaan iterasi dapat dilihat pada Tabel 4 berikut.

Tabel hasil perencanaan iterasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Kode US | Deskripsi | Estimasi (hari) |
| *Iteration* – 1 | | | |
|  | US-01 | Login ke dalam system dan mengakhiri sistem | 3 |
|  | US-12 | Profile admin *login* | 4 |
|  | US-02 | Melihat data admin | 3 |
| *Velocity* | | | 10 |
| *Iteration* – 2 | | | |
|  | US-03 | Menambah data admin | 2 |
|  | US-04 | Mengubah data admin | 2 |
|  | US-05 | Menghapus data admin | 2 |
|  | US-07 | Menambah data tempat ibadah | 4 |
| *Velocity* | | | 10 |
| *Iteration* – 3 | | | |
|  | US-14 | Import data excel tempat ibadah | 5 |
|  | US-15 | Unduh template format excel import data | 2 |
|  | US-16 | Export data excel tempat ibadah | 3 |
| *Velocity* | | | 10 |
| *Iteration* – 4 | | | |
|  | US-06 | Melihat daftar tempat ibadah | 6 |
|  | US-13 | Beranda halaman admin | 2 |
|  | US-08 | Mengubah data tempat ibadah | 2 |
| *Velocity* | | | 10 |
| *Iteration* – 5 | | | |
|  | US-09 | Menghapus data tempat ibadah | 2 |
|  | US-10 | Melihat peta lokasi tempat ibadah basis *android* | 7 |
|  | US-11 | Melihat rute lokasi tempat ibadah android | 1 |
| *Velocity* | | | 10 |
| *Iteration* – 6 | | | |
|  | US-17 | Melihat peta lokasi tempat ibadah pada basis *web* | 8 |
|  | US-18 | Melihat rute menuju lokasi tempat ibadah pada basis web | 2 |
| *Velocity* | | | 10 |

Karena ada penambahan user stories lagi dari klient, dilakukannya perencanaan ulang terhadap user stories baru, dan menyebabkan penambahan iterasi seperti yang terlampir pada tabel. Terdapat 3 iterasi tambahan dan dimana tiap iterasi dikerjakan selama 10 hari, sehingga rencana waktu pengerjaan system bertambah menjadi 60 hari.

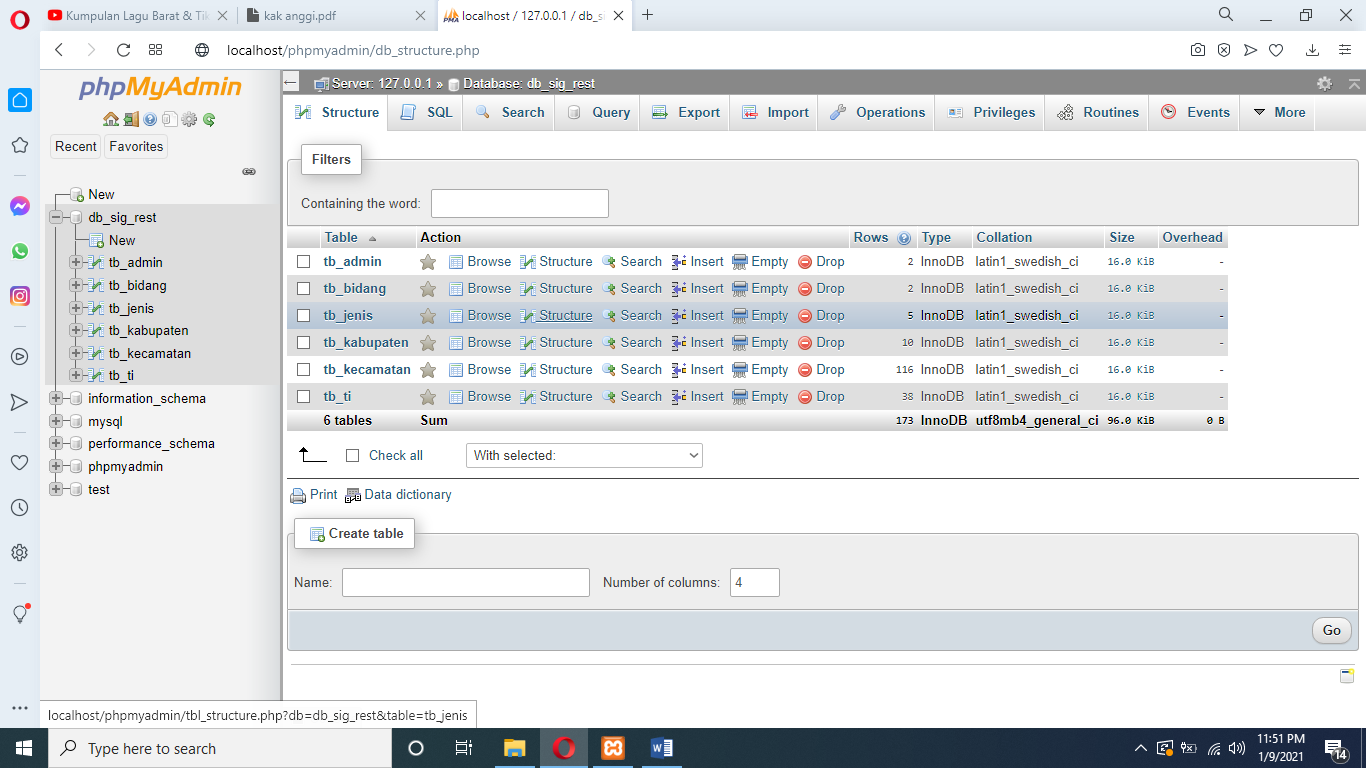
1. Design
2. Coding

Pada tahap coding ini

1. **Implementasi system**

Pada sub-bab ini, akan dibahas hasil implementasi (*coding*) SIG Pemetaan Lokasi Tempat Ibadah. Implementasi yang dilakukan telah dibuat sesuai dengan perancangan yang dijabarkan pada Bab III. Pembahasan yang akan dijelaskan meliputi implementasi *database*, implementasi API, implementasi *class*, dan implementasi *interface.*

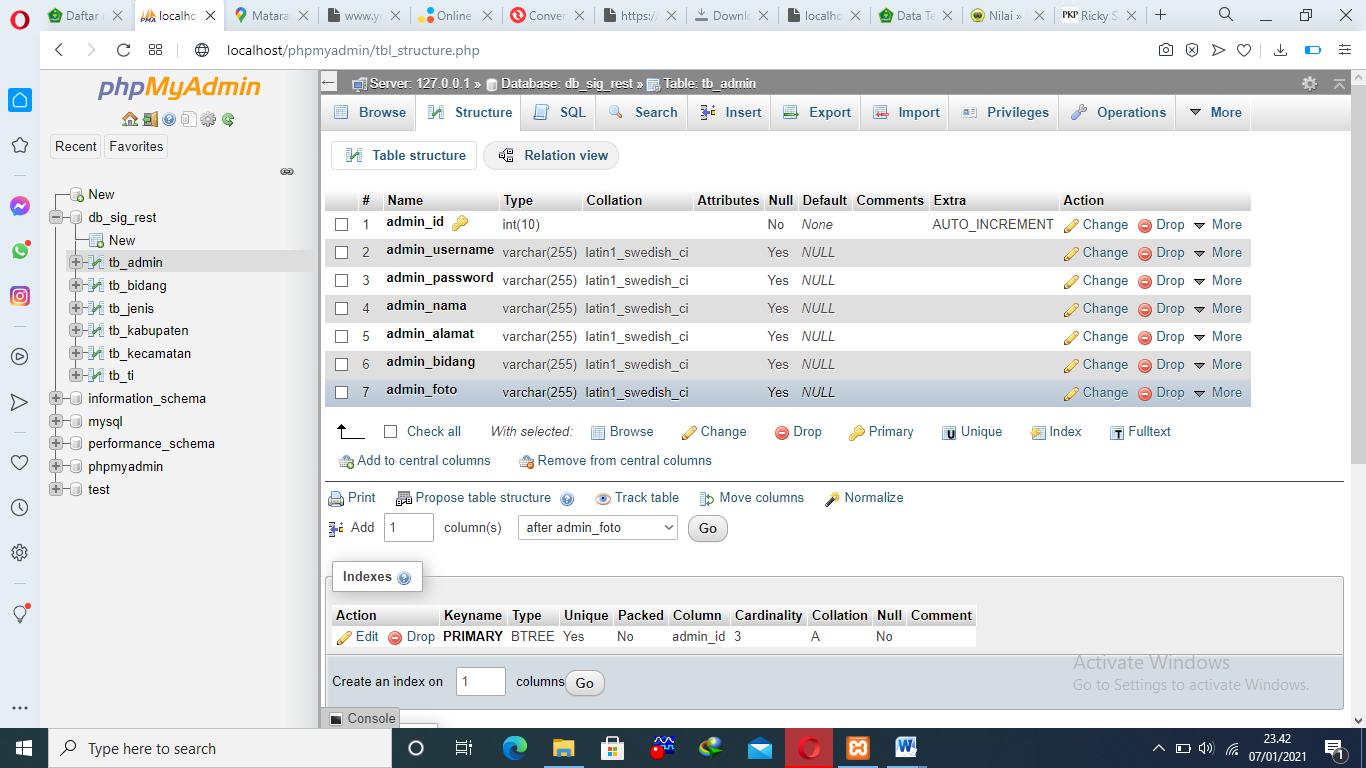
1. **Implementasi *database***



**Gambar 4.1** Implementasi *database*

Gambar 4.1 merupakan implementasi database yang bernama “db\_sig\_rest”, terdapat 6(enam) tabel terdiri tb\_admin, tb\_bidang, tb\_jenis, tb\_kabupaten, tb\_kecamatan, dan tb\_ti. Berikut penjelasan dari masing masing tabel:

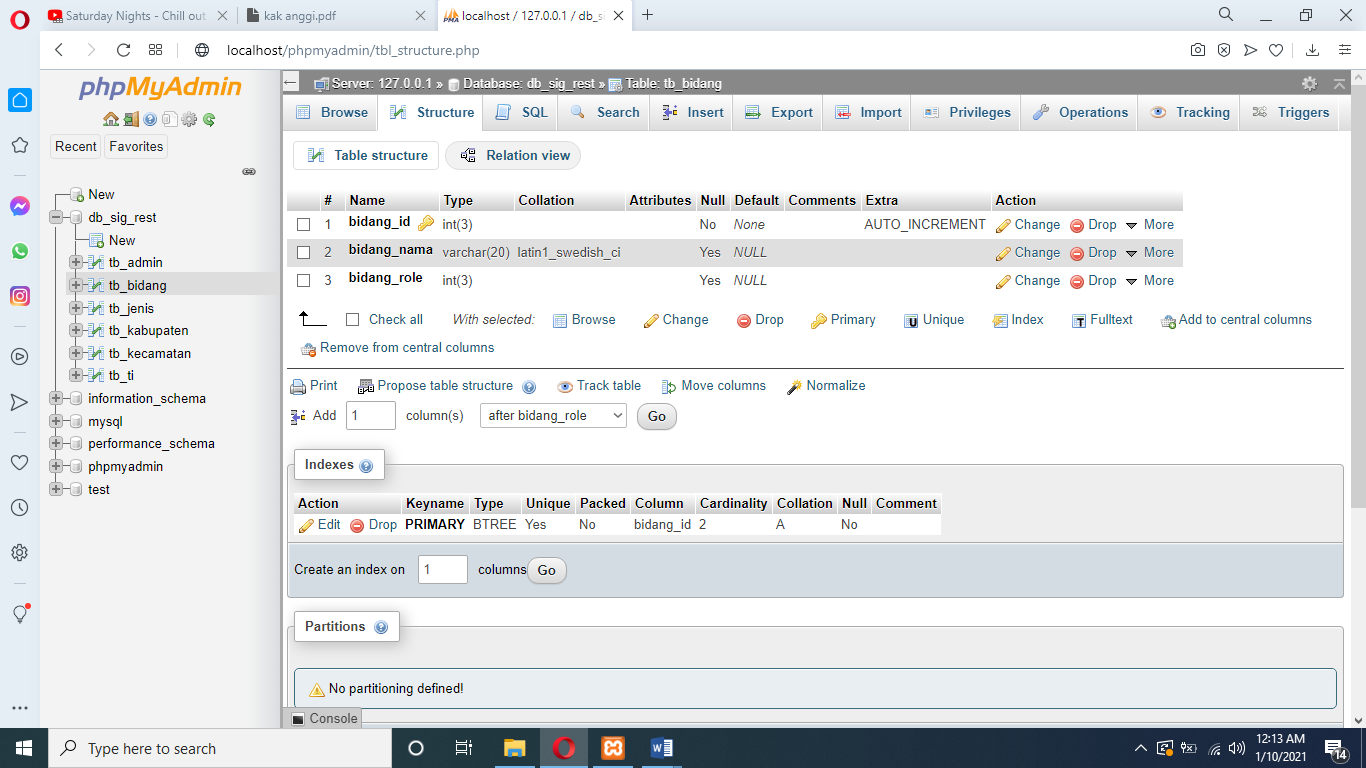
1. Tabel tb\_admin



**Gambar 4.2** Implementasi tabel tb\_admin

Gambar 4.2 merupakan implementasi tabel tb\_admin. Terdapat field admin\_id, admin\_username, admin\_password, admin\_nama, admin\_alamat, admin\_bidang, dan admin\_foto.

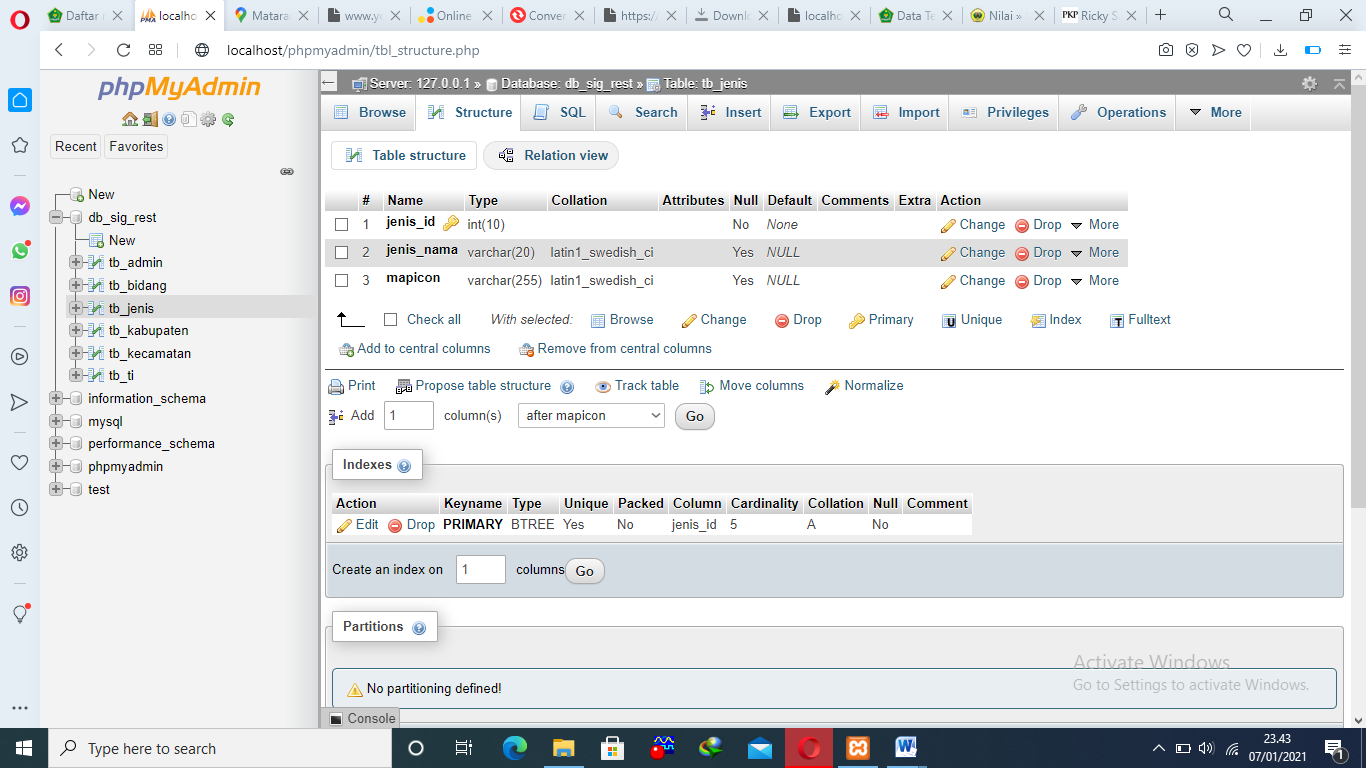
1. Tabel tb\_bidang



**Gambar 4.3** Implementasi tabel tb\_bidang

Gambar 4.3 merupakan implementasi tabel tb\_bidang. Terdapat field bidang\_id, bidang\_nama, dan bidang\_role.

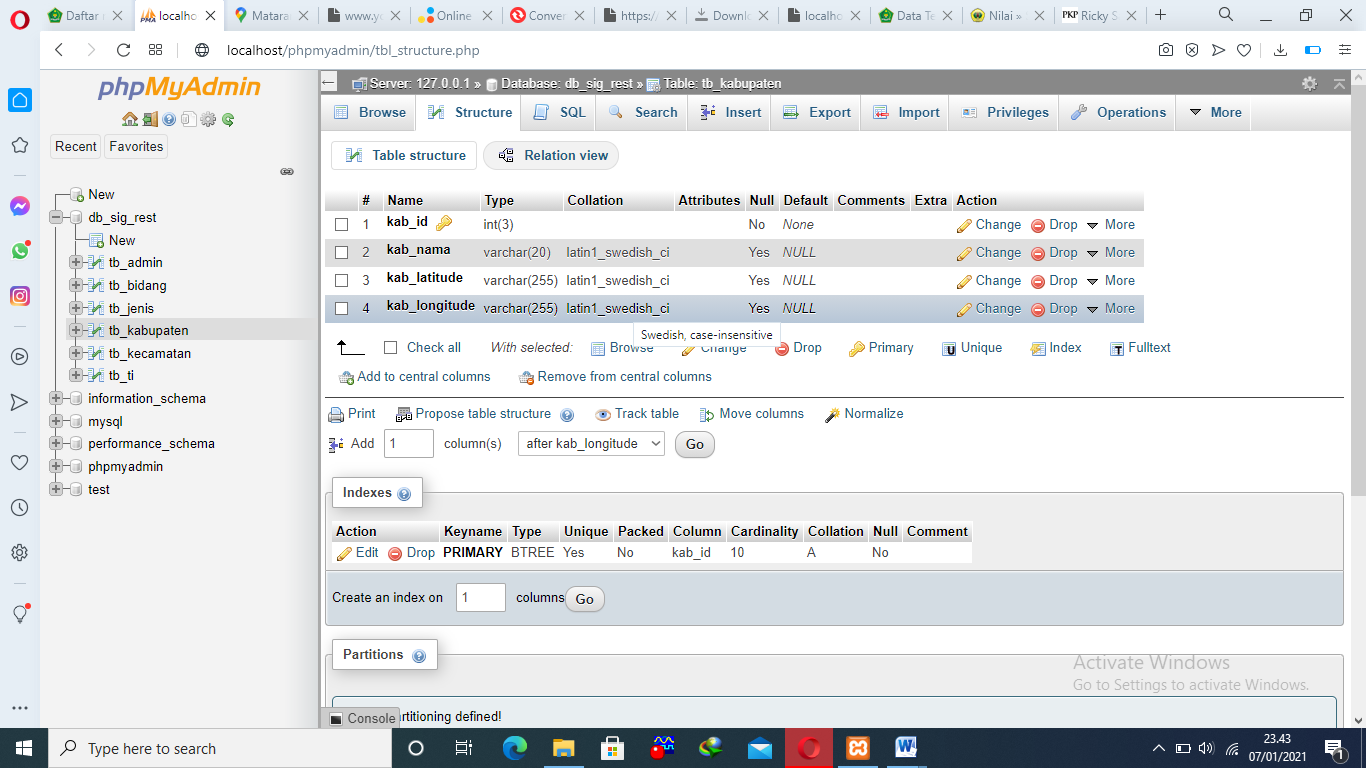
1. Tabel tb\_jenis



**Gambar 4.4** Implementasi tabel tb\_jenis

Gambar 4.4 merupakan implementasi tabel tb\_jenis. Terdapat field bidang\_id, bidang\_nama, dan bidang\_role.

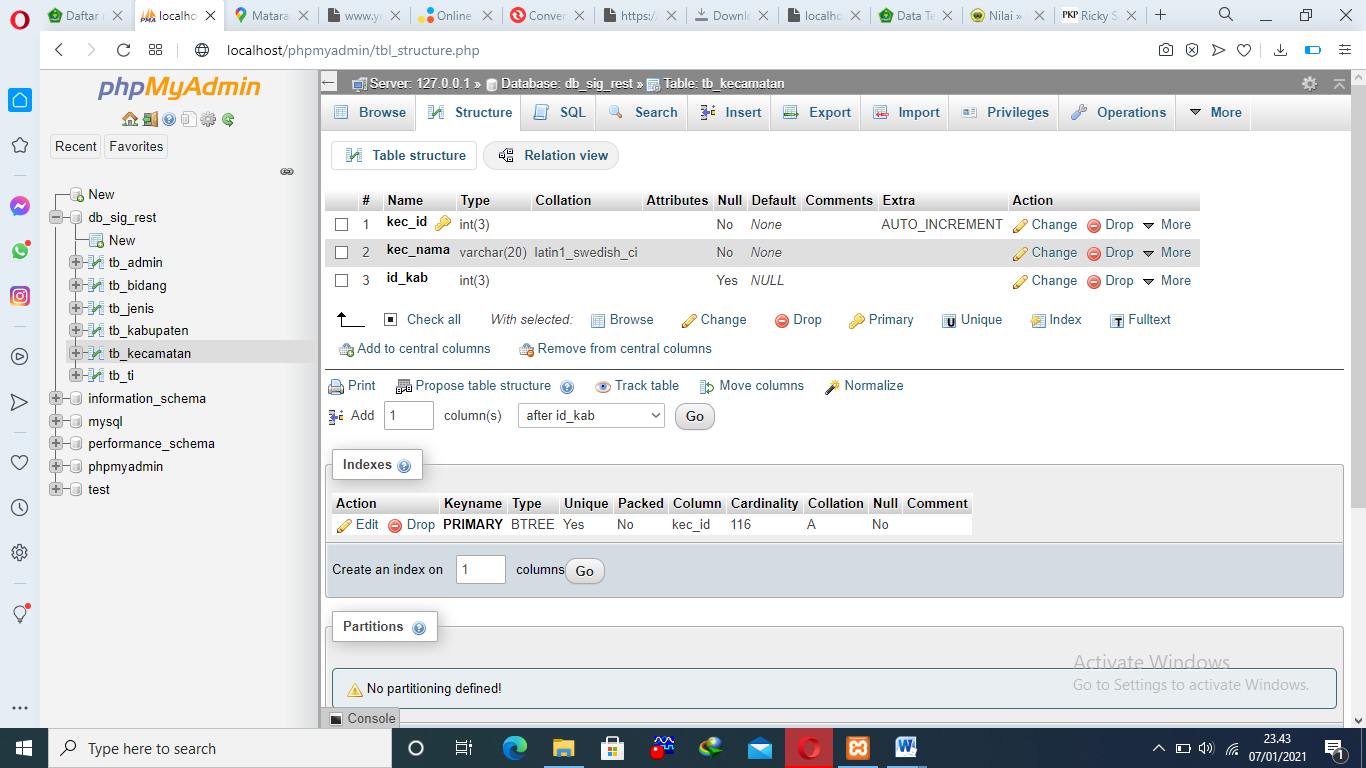
1. Tabel tb\_kabupaten



**Gambar 4.5** Implementasi tabel tb\_kabupaten

Gambar 4.5 merupakan implementasi tabel tb\_kabupaten. Terdapat field kab\_id, kab\_nama, kab\_latitude dan kab\_longitude.

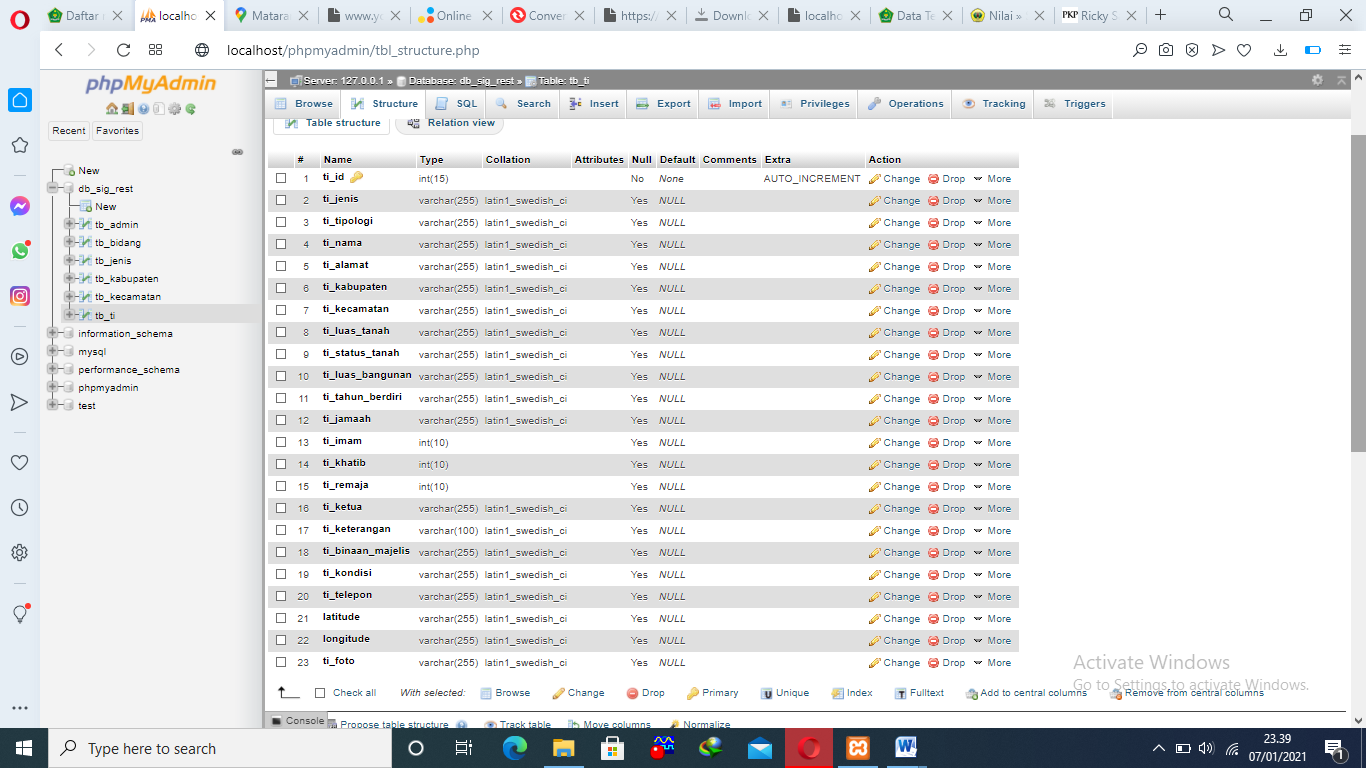
1. Tabel tb\_kecamatan



**Gambar 4.6** Implementasi tabel tb\_kecamatan

Gambar 4.6 merupakan implementasi tabel tb\_kecamatan. Terdapat field kec\_id, kec\_nama, dan id\_kab.

1. Tabel tb\_ti



**Gambar 4.7** Implementasi tabel tb\_ti

Gambar 4.7 merupakan implementasi tabel tb\_ti. Terdapat field ti\_id, ti\_jenis, ti\_tipologi, ti\_nama, ti\_alamat dan lainnya dapat dilihat pada Gambar 4.7.

1. **Implementasi API**

Berikut merupakan implementasi API digunakan controller dan model dalam pembuatannya. API ini bertujuan agar sistem pada basis *android* dapat terintegrasi atau terhubung dengan *database* yang dibuat.

Berikut merupakan fungsi get pada *class controller* yang digunakan untuk *interface* agar terhubung dengan *class* model APInya.

1. public function index\_get()
2. {
3. $v\_id\_kab = $this->get('kab\_id');
4. $v\_id\_jenis = $this->get('jenis\_id');
6. if (($v\_id\_kab != null) && ($v\_id\_jenis != null)) {
7. $v\_tempat\_ibadah  = $this->M\_tempat\_ibadah->selectByKabJenis($v\_id\_kab,$v\_id\_jenis);
8. }elseif (($v\_id\_kab != null) && ($v\_id\_jenis === null)) {
9. $v\_tempat\_ibadah  = $this->M\_tempat\_ibadah->selectByKabupaten($v\_id\_kab);
10. }elseif (($v\_id\_kab === null) && ($v\_id\_jenis != null)) {
11. $v\_tempat\_ibadah  = $this->M\_tempat\_ibadah->selectByJenis($v\_id\_jenis);
12. }else{
13. $v\_tempat\_ibadah  = $this->M\_tempat\_ibadah->selectAlltempatIbadah();
14. }

17. if  ($v\_tempat\_ibadah)  {
18. $this->response([
19. 'status' => true,
20. 'data' => $v\_tempat\_ibadah
21. ], REST\_Controller::HTTP\_OK);
22. }else {
23. $this->response([
24. 'status' => false,
25. 'respone' => 'id is not found'
26. ], REST\_Controller::HTTP\_NOT\_FOUND);
27. }
28. }

*Source code* 4.1 implementasi fungsi get pada *controller*.

Berikut merupakan fungsi yang terdapat pada *class model* yang digunakan untuk mengambil data dalam *database.*

1. public function selectByJenis($jenis){
2. $sql='SELECT \* FROM tb\_ti LEFT JOIN tb\_kabupaten ON tb\_ti.ti\_kabupaten = tb\_kabupaten.kab\_id LEFT JOIN tb\_kecamatan ON tb\_kecamatan.id\_kab = tb\_kabupaten.kab\_id AND tb\_kecamatan.kec\_id = tb\_ti.ti\_kecamatan JOIN tb\_jenis ON tb\_ti.ti\_jenis = tb\_jenis.jenis\_id WHERE ti\_jenis =?';
3. return $this->db->query($sql,$jenis)->result\_array();
4. }
5. public function selectAlltempatIbadah(){
6. $sql='SELECT \* FROM tb\_ti LEFT JOIN tb\_kabupaten ON tb\_ti.ti\_kabupaten = tb\_kabupaten.kab\_id LEFT JOIN tb\_kecamatan ON tb\_kecamatan.id\_kab = tb\_kabupaten.kab\_id AND tb\_kecamatan.kec\_id = tb\_ti.ti\_kecamatan JOIN tb\_jenis ON tb\_ti.ti\_jenis = tb\_jenis.jenis\_id';
7. return $query=$this->db->query($sql)->result\_array();
8. }

*Source code* 4.2 implementasi fungsi get pada *model*.

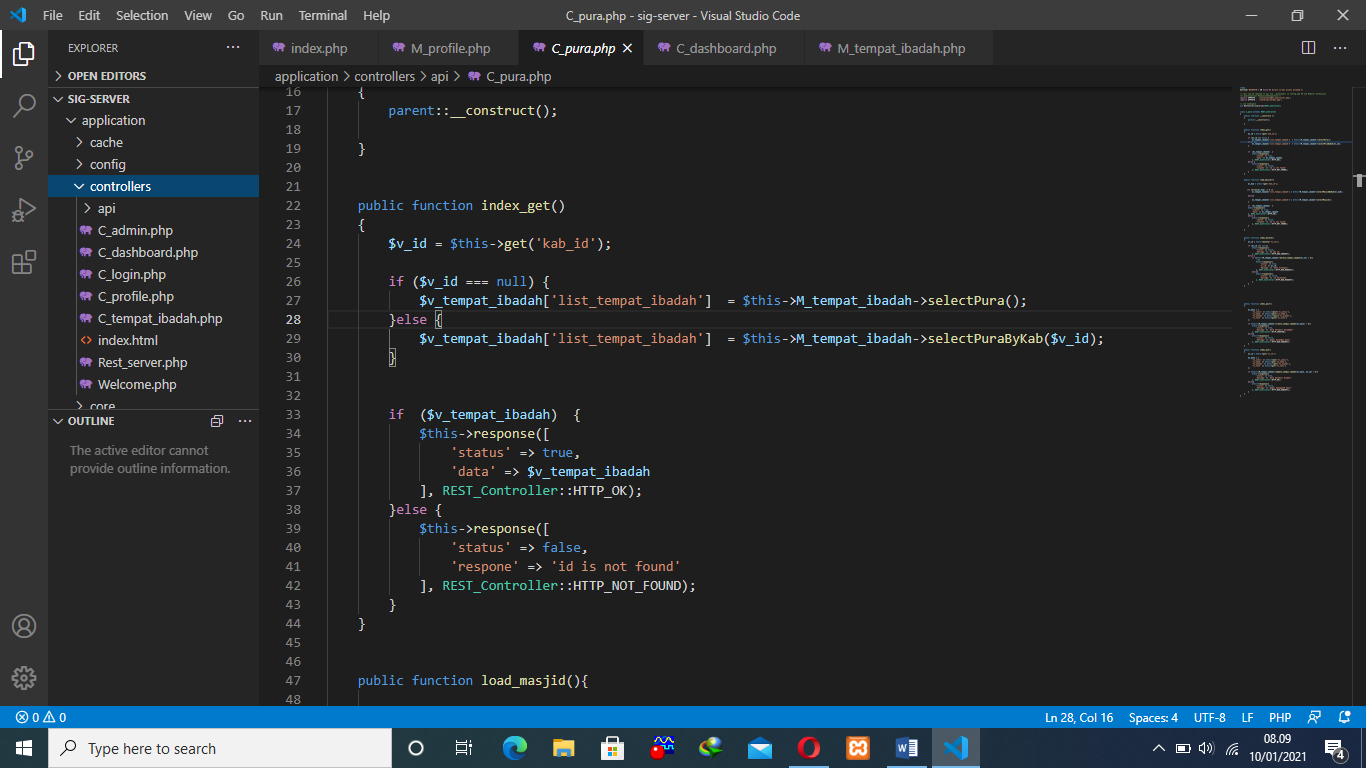
1. **Implementasi *class***

Berdasarkan perancangan class diagram yang telah dilakukan sebelumnya, implementasi *class* yang dilakukan dirasa sudah sesuai dengan perancangan tersebut. *Class - class* ini merupakan *class* yang berisi bahasa pemrograman PHP yang dibuat untuk mengimplementasikan sistem basis *web* sesuai dengan perancangan yang dilakukan berupa *sourcecode* (*coding*) dan bahasa pemrograman Kotlin untuk mengimplementasikan system basis *android*. Hasil dari proses *coding* tersebut akan menghasilkan *interface* yang akan berinteraksi langsung dengan pengguna sistem. Berikut merupakan implementasi *class* yang dilakukan dalam pengembangan “Sistem Informasi Geografis Untuk Pemetaan Lokasi Tempat Ibadah di Kota Mataram menggunakan Metode *Extreme Programming*”.

1. **Basis *web***

Berikut merupakan rincian *class – class* yang digunakan dalam membuat system basis *web* pada “Sistem Informasi Geografis untuk Pemetaan Lokasi Tempat Ibadah di Kota Mataram menggunakan Metode *Extreme Programming*”.

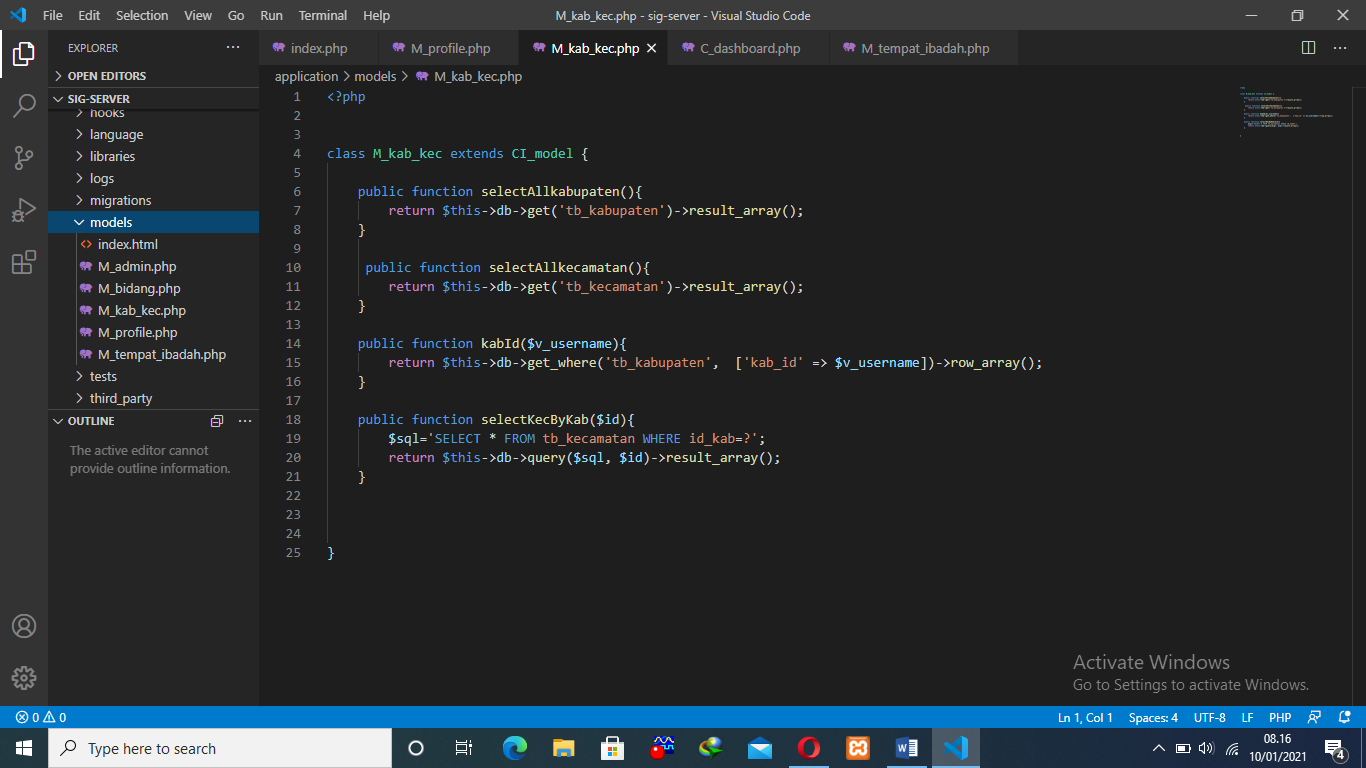
1. *Class controller*



**Gambar 4.8** Implementasi *class controller*

Gambar 4.8 merupakan implementasi *class* turunan dari *class* *controller* yang terdiri dari C\_admin, C\_dashboard, C\_login, C\_profile C\_tempat\_ibadah, dan Rest\_server. *Class – class* ini merupkan *class* utama yang menjadi pengatur atau penghubung pertukaran data antara *class model* dan *view.*

1. *Class model*



**Gambar 4.9** Implementasi *class model*

Gambar 4.9 merupakan implementasi *class* turunan dari *class* *model* yang terdiri dari M\_admin, M\_bidang, M\_kab\_kec, M\_profile, dan M\_tempat\_ibadah. Pada *class* ini nantinya berguna mengatur atau memanipulasi data dari *database* sesuai dengan intruksi dari *controller.*

1. *Class view*



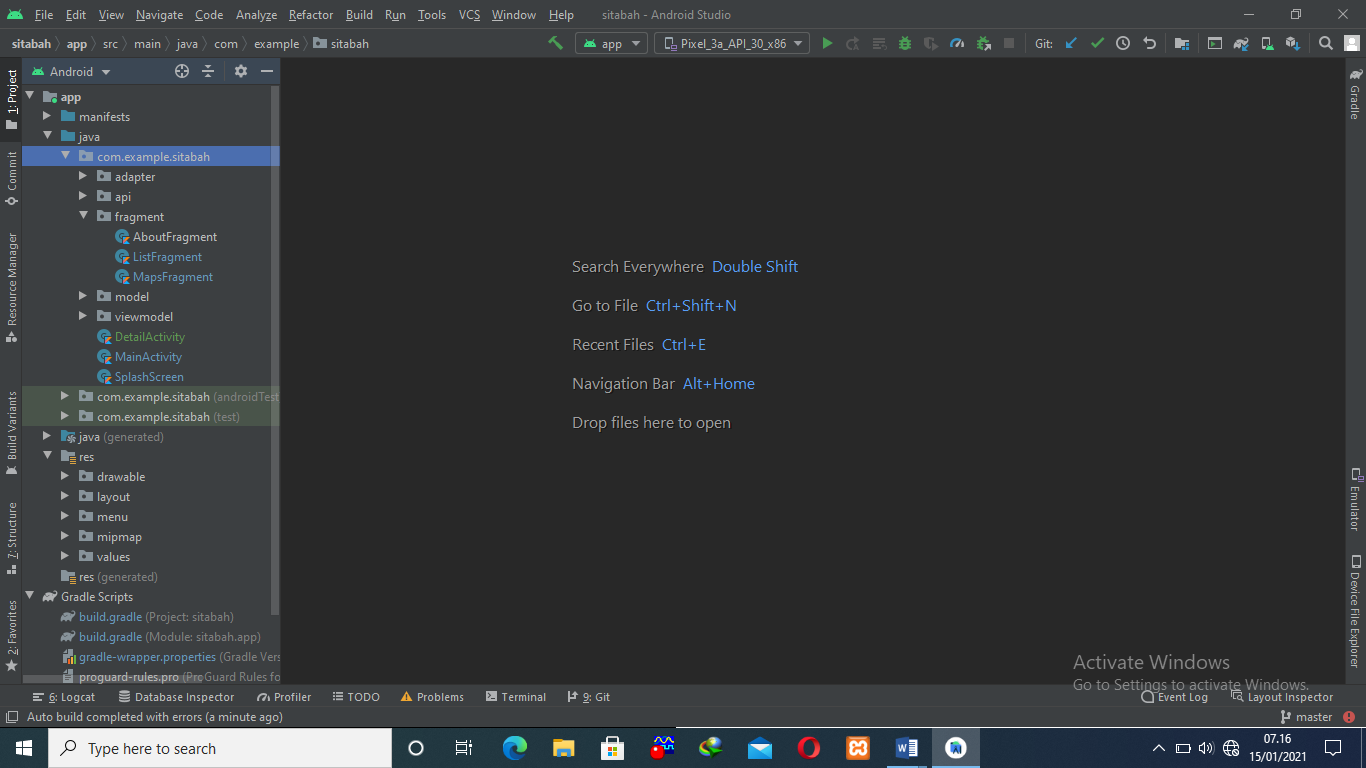
**Gambar 4.10** Implementasi *class view*

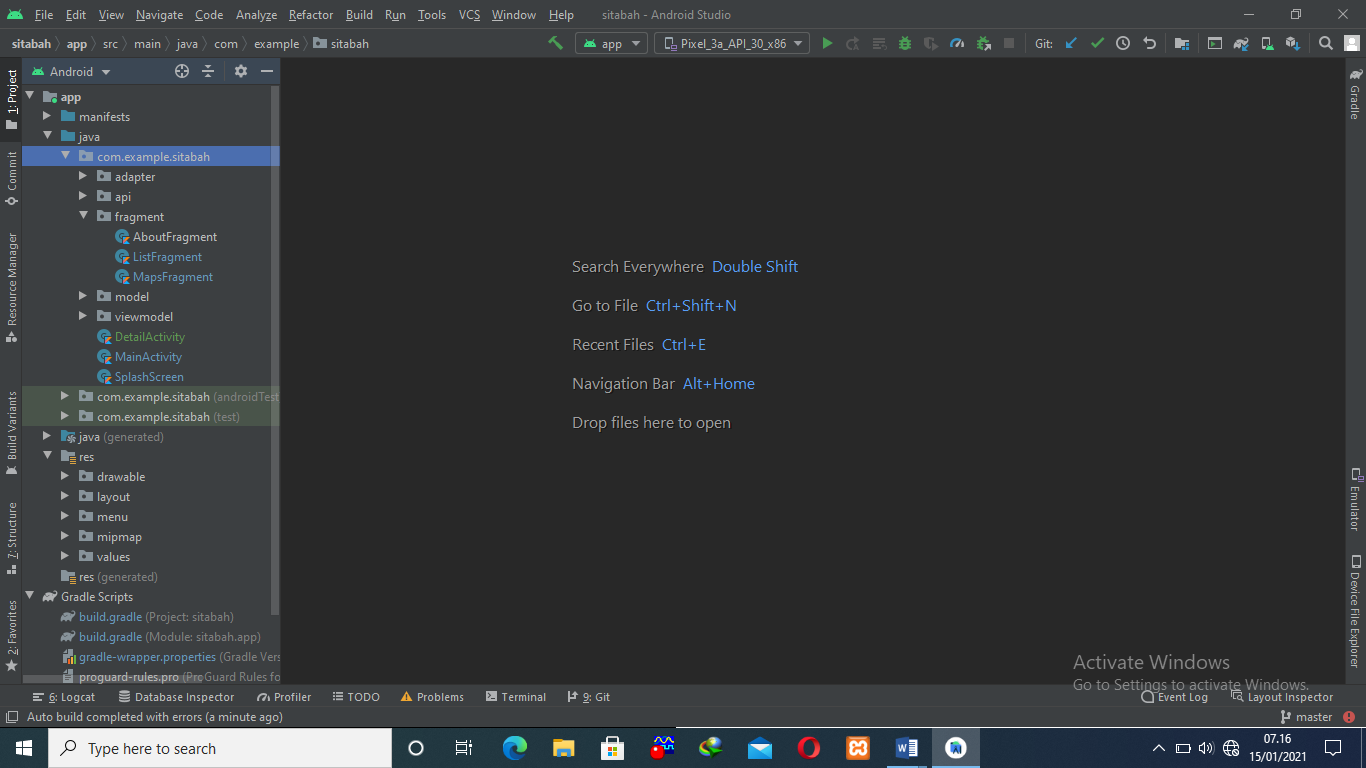
Gambar 4.10 merupakan implementasi *class* *view* yang terdiri dari V\_beranda, V\_daftar\_admin, V\_daftar\_tempat\_ibadah, V\_dashboard, V\_login, dan V\_profile. Dimana *class* ini nantinya bertugas untuk menyajikan informasi. sesuai dengan intruksi dari *controller.*

1. **Basis *android***

Berikut merupakan *class – class* yang digunakan dalam membuat system basis *android* pada “Sistem Informasi Geografis untuk Pemetaan Lokasi Tempat Ibadah di Kota Mataram menggunakan Metode *Extreme Programming*”.

1. *Class activity*

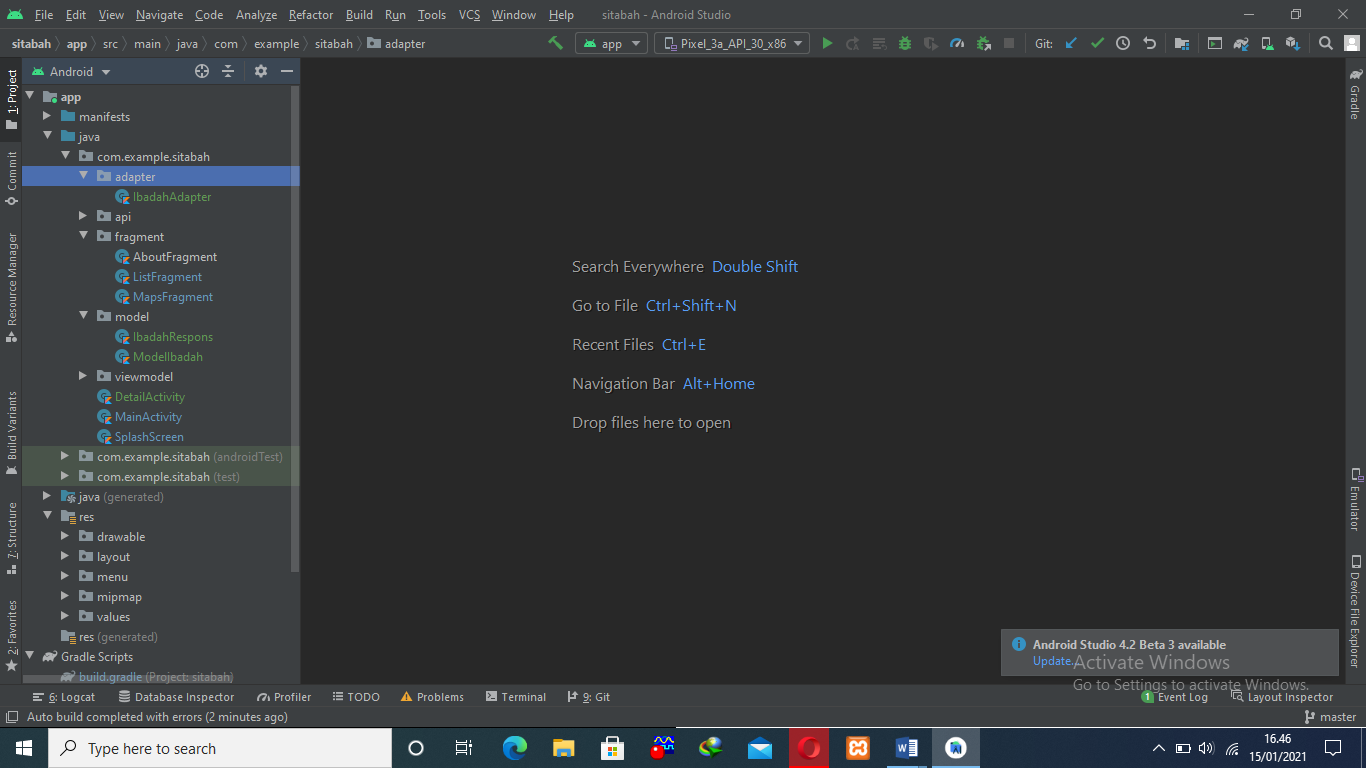




**Gambar 4.11** Implementasi *class activity*

Gambar 4.11 merupakan implementasi *Package Activity* yang terdiri dari DetailActivity, MainActivity, SplashScreen, AboutFragment, ListFragment, dan MapFragment. *Package Activity* ini merupakan sebuah kelas yang nantinya terhubung ke Layout (XML) atau tampilan dari aplikasi. *Activity* ini nantinya tempat dilakukan koding untuk membuat fungsi-fungsi yang akan ditampilkan pada sebuah halaman.

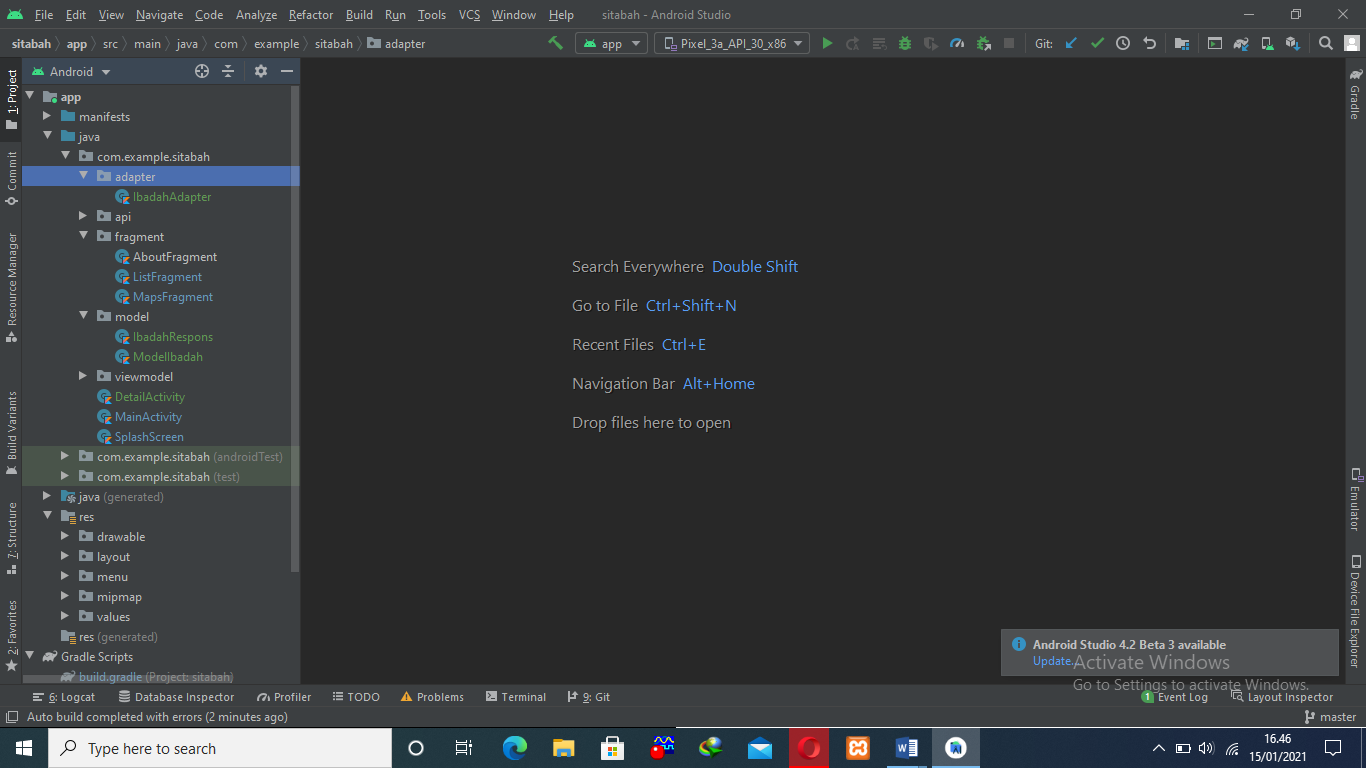
1. Adapter (Kotlin)



**Gambar 4.12** adapter

Berbeda dengan *package Activity*, package adapter pada Gambar 4.12 ini merupakan kelas yang menghubungkan antara data yang ada pada *database* ke dalam sebuah *listview* pada aplikasi.

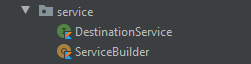
1. Model (Kotlin)



**Gambar 4.13** Implementasi *class model*

Gambar 4.13 merupakan implementasi *Package Model* yang terdiri dari IbadahRespons dan ModelIbadah. Package model ini merupakan pengimplementasian pada sebuah tabel di *database*. Dimana dapat dilihat nama setiap kelas yang dibuat merupakan nama yang ada pada tabel di *database*.

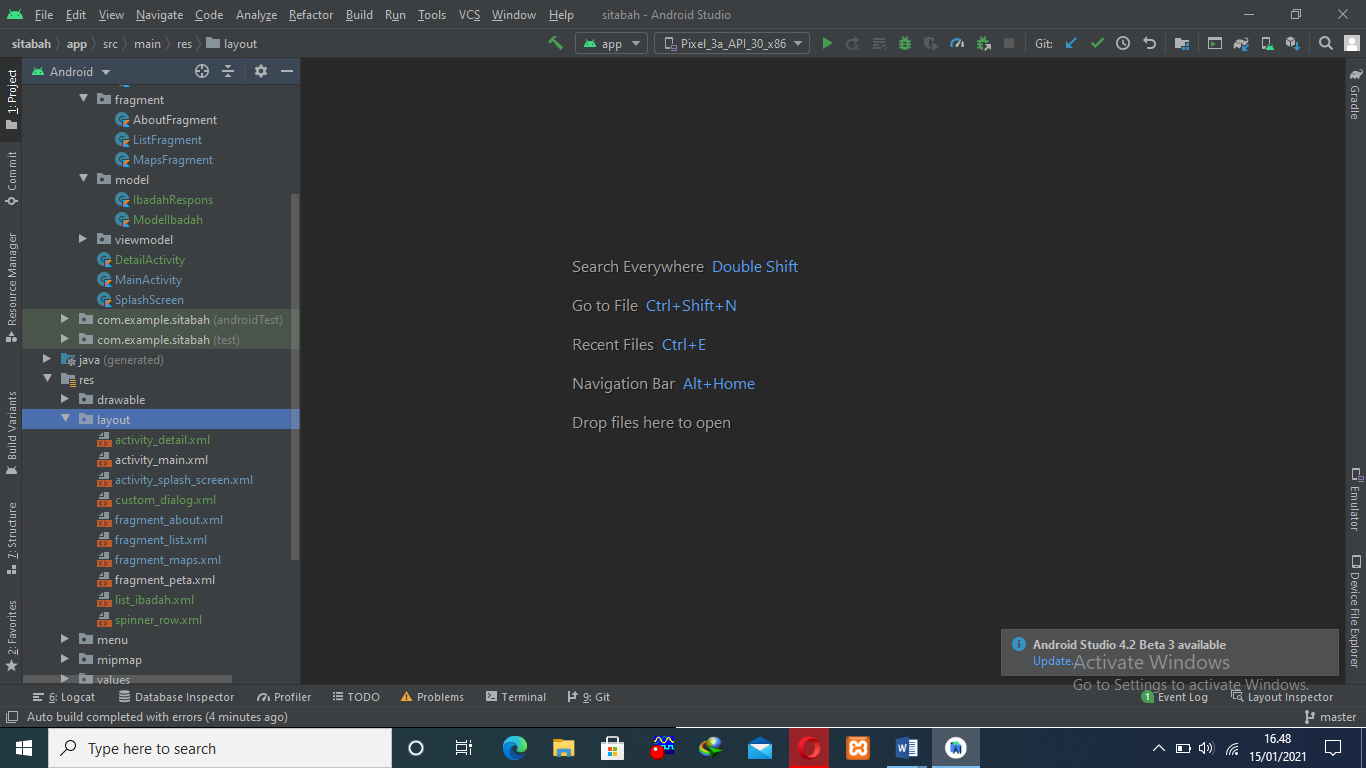
1. Service (Kotlin)



**Gambar 4.14** Implementasi *class service*

Gambar 4.14 merupakan implementasi *Package Service* yang terdiri dari DestinationService dan ServiceBuilder. *Package Service* merupakan cara yang digunakan untuk terhubung dengan API server sistem. Dimana ServiceBuilder ini merupakan kelas untuk menghubungkan aplikasi dengan Base url API sistem, dan kelas ini juga terhubung dengan kelas interface DestinationService. Sedangkan kelas DestinationService digunakan untuk mengimplementasi *endpoint* dari *base url* yang telah dibuat pada API.

1. Layout (XML)



**Gambar 4.15** *Layout* (XML)

Gambar 4.15 merupakan implementasi Layout yang berguna untuk tempat membuat sebuah tampilan yang nantinya akan dilihat oleh pengguna.

1. **Implementasi *interface***
2. Testing

EVALUASI HASIL METODE EXTREME PROGRAMMING

Pada kasus kode *user stories* US-1 ini admin dapat melakukan login untuk dapat mengakses system dan dapat mengakhiri system (*logout*). Berikut rincian analisa, desain, dan *code* system, tampilan serta hasil pengujian systemnya.

1. Analisa sistem

Dilakukan analisa terhadap kebutuhan sistem berupa *task – task* yang akan diselesaikan untuk kasus kode *user stories* US-1 dirincikan pada tabel 3 berikut.

Tabel 4.1 kebutuhan system kode *user stories* US-1

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Tampilan halaman muka | Menampilkan *form* *login* (*input username* dan *password*)Menampilkan pilihan sub bagian (*level* admin) pada *form login*Menampilkan tombol *login* |
| Tampilan *logout* | Menampilkan *popup* *logout*Menampilkan tombol *logout*Menampilkan tombol batalmenghapus *session* |

1. Desain sistem

Setelah analisa, dilakukannya desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-1 dapat dilihat pada Tabel 5.

Tabel 4.2 desain system kode *user stories* US-1

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-01 | *Login* | Gambar 3.21 |

Pada CRC *card* Login terdapat class C\_login pada *controller*, M\_admin pada *model* dan V\_login pada *view*.

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-1 yang telah dibuat berdasarkan CRC *card* Login yang telah dipaparkan pada tabel 4.2 dan *code* yang dibuat dirasa merupakan *code* sederhana (hasil *refactor*). Hasil *code* tersebut dapat dilihat pada *source code* berikut:

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3. class C\_login extends CI\_Controller {
5. public function \_\_construct(){
6. parent::\_\_construct();
7. }
8. function validasi\_bidang($v\_bidang)
9. {
10. if($v\_bidang == "none"){
11. $this->form\_validation->set\_message('validasi\_bidang', 'Pilih Subbag!');
12. return false;
13. } else{
14. return true;
15. }
16. }
17. public function index(){
18. if ($this->session->userdata('id\_username')) {
19. redirect('c\_profile');
20. }
21. $this->form\_validation->set\_rules('bidang','Bidang','required|callback\_validasi\_bidang');
22. $this->form\_validation->set\_rules('username', 'Username', 'required|trim', [
23. 'required' => 'Form tidak boleh kosong!'
24. ]);
25. $this->form\_validation->set\_rules('password', 'Password', 'required|trim', [
26. 'required' => 'Form tidak boleh kosong!'
27. ]);
29. $v\_data['tittle'] = 'Login page!';
30. $v\_data['list\_bidang'] = $this->M\_bidang->selectAllbidang();
31. if($this->form\_validation->run() == false){
32. $this->load->view('templates/header\_login', $v\_data);
33. $this->load->view('v\_login/index', $v\_data);
34. $this->load->view('templates/footer\_login');
35. }else{
36. $v\_username = $this->input->post('username');
37. $v\_password = $this->input->post('password');
38. $v\_idbidang = $this->input->post('bidang');
39. $v\_admin = $this->M\_admin->auth($v\_username, $v\_password, $v\_idbidang);
41. if ($v\_admin){
42. $bidang\_role = $this->M\_bidang->selectBidangBy($v\_admin['admin\_bidang']);
43. $v\_id\_admin = $this->M\_admin->selectIdAdmin($v\_username);
44. $v\_data = [
45. 'id\_username' => $v\_admin['admin\_id'],
46. 'role' => $bidang\_role['bidang\_role']
47. ];
48. $this->session->set\_userdata($v\_data);
49. redirect('c\_profile');
50. }else {
51. $this->session->set\_flashdata('pesan', '<div class="alert alert-danger" role="alert">Username dan password salah!</div>');
52. redirect('c\_login');
53. }
54. }
55. }

*Source code* 4.3 *class* C\_login US-01

Pada *Source code* 4.3 merupakan *source code class* C\_login yang ada pada *controller.*

1. public function auth($v\_username, $v\_password, $v\_idbidang){
2. $sql='SELECT \* FROM tb\_admin where admin\_username=? AND admin\_password=? AND admin\_bidang=?';
3. return $this->db->query($sql, array($v\_username,$v\_password,$v\_idbidang))->row\_array();
4. }

*Source code* 4.4 *class* M\_admin US-01

Pada *Source code* 4.4 merupakan *source code class* M\_admin pada *model* yang digunakan untuk memvalidasi kesesuaian data yang dimasukkan dengan yang ada di *database.*

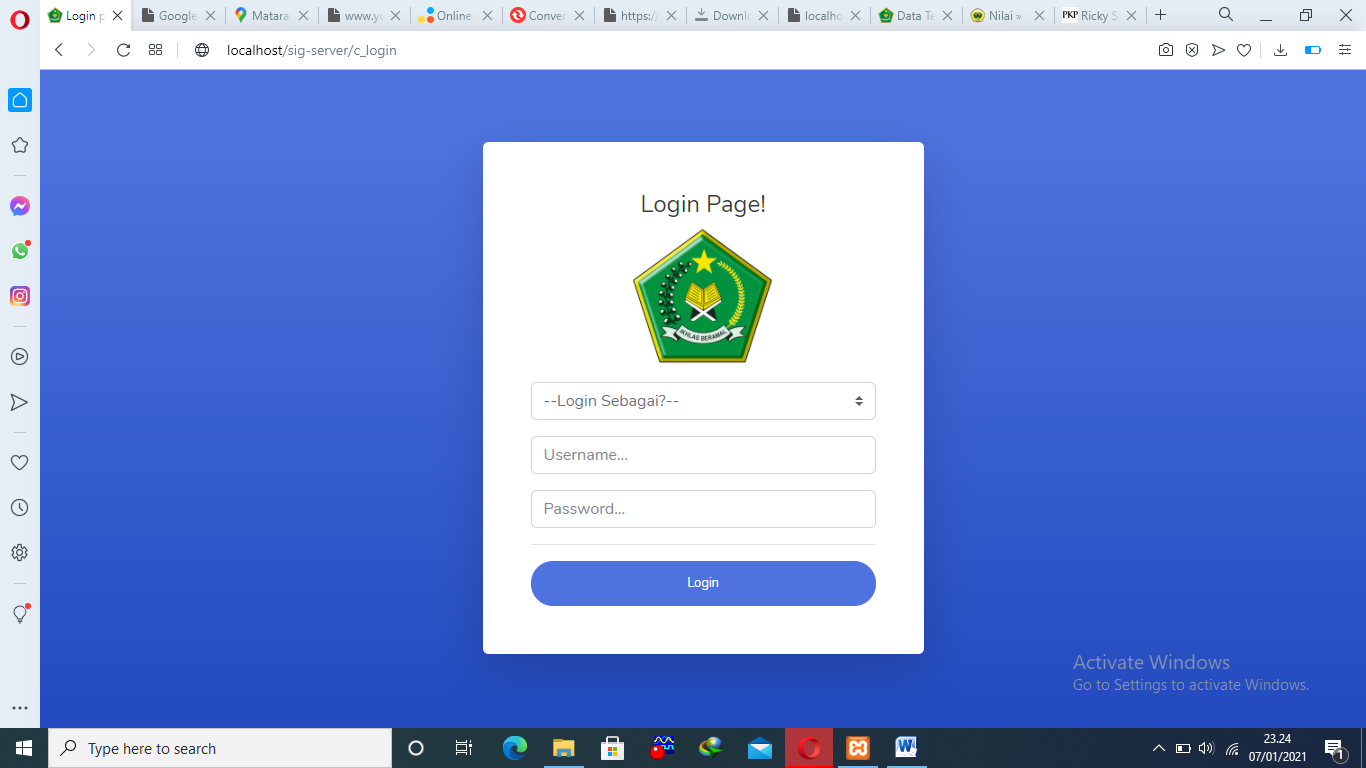
1. <div class="container">
2. <div class="row justify-content-center">
3. <div class="col-lg-5 mt-4">
4. <div class="card o-hidden border-0 shadow-lg my-5" >
5. <div class="card-body p-0">
6. <div class="row">
7. <div class="col-lg">
8. <div class="p-5">
9. <div class="text-center">
10. <h1 class="h4 text-gray-900">Login Page!</h1>
11. <img src="<?= base\_url('assets/foto/tittle.png'); ?>" width="" class="card-img mb-3" style="width: 9rem;">
12. <?= $this->session->flashdata('pesan'); ?>
13. </div>
14. <form class="user" method="post" action="<?= base\_url('c\_login'); ?>">
16. <div class="form-group mb-3" >
17. <div class="input-group-prepend">
18. </div>
19. <select class="custom-select " id="inputGroupSelect01" name="bidang">
20. <option value="none" selected="selected">--Login Sebagai?--</option>
21. <?php foreach($list\_bidang as $bidang) : ?>
22. <option value="<?= $bidang['bidang\_id']; ?>"><?= $bidang['bidang\_nama']; ?></option>
23. <?php endforeach; ?>
24. </select>
25. <?= form\_error('bidang', '<small class="text-danger">', '</small>'); ?>
26. </div>
27. <div class="form-group">
28. <input type="text" class="form-control " id="exampleInputEmail" aria-describedby="emailHelp" placeholder="Username..." name="username" value="<?= set\_value('username'); ?>">
29. <?= form\_error('username', '<small class="text-danger">', '</small>'); ?>
30. </div>
31. <div class="form-group">
32. <input type="password" class="form-control " id="exampleInputPassword" placeholder="Password..." name="password">
33. <?= form\_error('password', '<small class="text-danger">', '</small>'); ?>
34. </div>
35. <div class="form-group">
36. </div><hr>
37. <button type="submit" class="btn btn-primary btn-user btn-block">Login</button>
38. </form>
39. </div>
40. </div>
41. </div>
42. </div>
43. </div>
44. </div>
45. </div>
46. </div>
47. </div>

*Source code* 4.5 *class* V\_login US-01

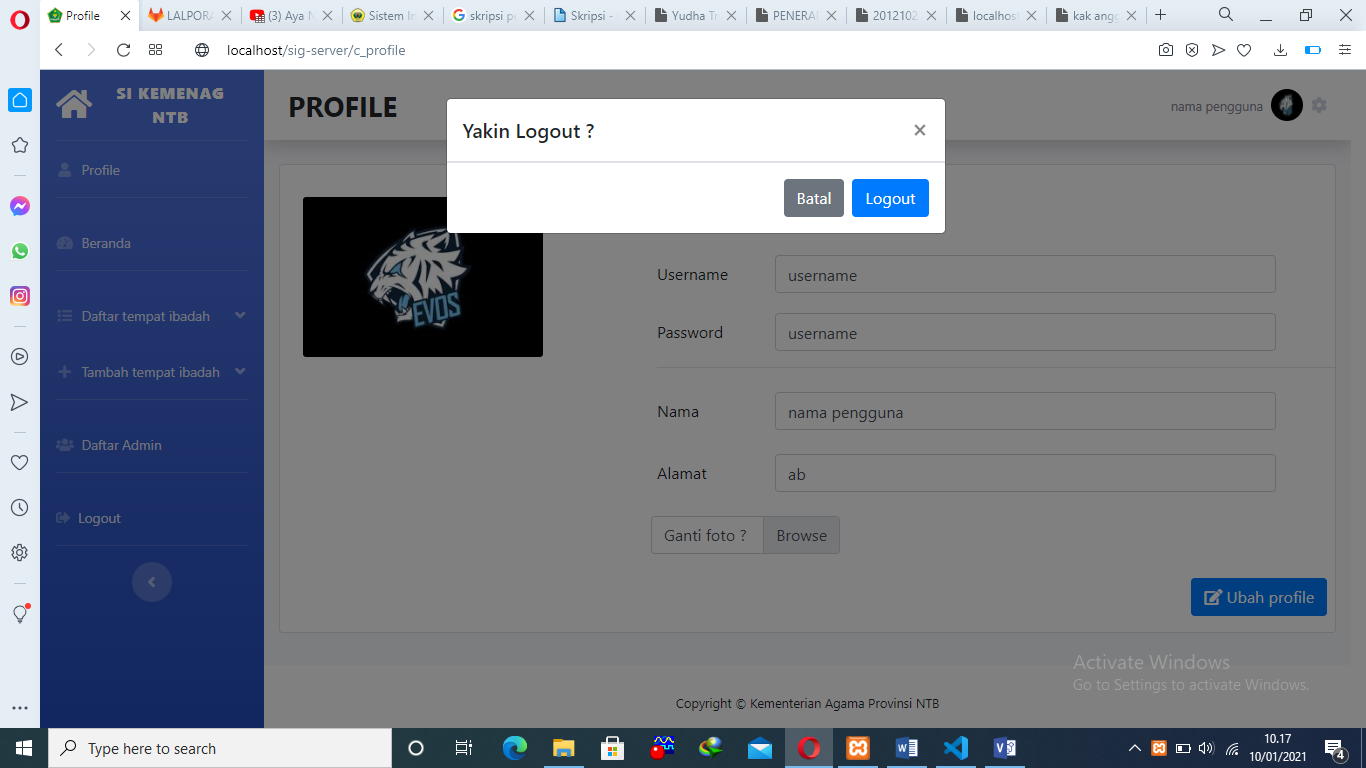
Pada *Source code* 4.5 merupakan *source code* V\_login yang ada pada *view* yang berguna untuk menampilkan halaman login*.*

1. Tampilan dan pengujian sistem

Berikut merupakan tampilan sistem dari kasus kode *user stories* US-1 yang telah dibuat dapat dilihat pada gambar berikut:



**Gambar 4.15** tampilan halaman *login*



**Gambar 4.16** tampilan *logout*

Pengujian sistem pada kasus kode *user stories* US-1 ini dilakukan untuk menguji apakah fungsi *login* yang terdapat pada proses tersebut sudah berjalan dengan baik dan benar. Pengujian tersebut dijelaskan pada Tabel 4.3.

Tabel 4.3 pengujian login system

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Bidang, *Username,* dan *password* tidak diisi kemudian klik tombol login | Sistem akan menolak akses dan akan dirujuk kembali ke halaman login | Sesuai | Valid |
| Bidang tidak dipilih, Mengetik *username* dan *password* tidak diisi atau kosong kemudian klik tombol login | Sistem akan menolak akses dan akan dirujuk kembali ke halaman login | Sesuai | Valid |
| Bidang dipilih, *Username* tidak diisi dan *password* diisi kemudian klik tombol login | Sistem akan menolak akses dan akan dirujuk kembali ke halaman login | Sesuai | Valid |
| Bidang, *Username* dan *password* diisi tidak benar kemudian klik tombol login | Sistem akan menolak akses dan akan dirujuk kembali ke halaman login | Sesuai | Valid |
| Bidang, *Username* dan *password* diisi dengan benar kemudian klik tombol login | Sistem menerima akses *login* dan kemudian langsung menampilkan halaman admin sesuai *role* bidang admin | Sesuai | Valid |

Sistem hanya akan menerima akses jika bidang, *username* dan *password* yang diisi sesuai dengan yang ada di *database*, selain itu sistem akan menolak akses.

Tabel 4.4 pengujian *logout* system

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Menekan tombol *batal* | Proses *logout* tidak akan terjadi dan dirujuk kembali ke halaman admin | Sesuai | Valid |
| Menekan tombol *logout* | Proses *logout* terjadi dan diarahkan langsung ke halaman *login* | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.3 dan table 4.4, dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. **Kode *user stories* US-06**

Pada kasus kode *user stories* US-06 ini pengguna dapat melihat daftar tempat ibadah yang tersusun dalam bentuk table ataupun *list*. Berikut penjelasan analisa, desain, code system dan tampilan serta hasil pengujian systemnya.

1. Analisa sistem

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-06 digambarkan pada tabel 4.6 berikut.

Tabel 4.6 kebutuhan system kode *user stories* US-6

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Tampilan halaman daftar tempat ibadah pada sisi masyarakat basis *web* | Menu tab : data, map, contact us, loginMenampilkan tabel data tempat ibadahTerdapat tombol detail data tempat ibadahFungsi filter berdasarkan kabupaten dan kecamatanFungsi pencarian data tempat ibadah |
| Tampilan halaman daftar tempat ibadah pada sisi admin basis *web* | Menu tab : profile, beranda, daftar tempat ibadah, tambah data tempat ibadah, daftar admin, logout.Menampilkan poto dan nama profile admin yang telah *login*Menampilkan tabel data tempat ibadahTerdapat tombol detail, edit, dan hapus data tempat ibadahFungsi filter berdasarkan kabupaten dan kecamatanFungsi pencarian data tempat ibadah |
| Tampilan halaman daftar tempat ibadah pada basis *android* | Menu tab : list, map, about us.Menampilkan data tempat ibadah dalam bentuk *list*Terdapat tombol detail data tempat ibadahFungsi filter berdasarkan jenis tempat ibadahFungsi pencarian data tempat ibadah |
| Tampilan halaman detail data tempat ibadah pada basis *web* dan *android* | Menampilkan detail data tempat ibadah |

1. Desain sistem

Setelah analisa, seperti biasa dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-06 dapat dilihat pada Tabel 4.7.

Tabel 4.7 kebutuhan system kode *user stories* US-06

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-06 | Dashboard | Gambar 3.20 |
| Tempat ibadah | Gambar 3.23 |

1. *Code* sistem

Berikut merupakan *code* sistem dari kasus kode *user stories* US-06 yang telah dibuat dapat dilihat pada listing code berikut:

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
4. class C\_dashboard extends CI\_Controller {
6. public function \_\_construct(){
7. parent::\_\_construct();
8. }
10. public function index(){
11. $v\_data['list\_jenis'] = $this->M\_tempat\_ibadah->selectAlljenis();
12. $v\_data['list\_kabupaten'] = $this->M\_kab\_kec->selectAllkabupaten();
13. $v\_data['list'] = $this->M\_tempat\_ibadah->selectAlltempatIbadah();
14. $this->load->view('v\_dashboard/index', $v\_data);
15. }
17. public function load\_jenis(){
18. echo json\_encode($this->M\_tempat\_ibadah->selectAlljenis());
19. }
21. public function load\_kabupaten(){
22. echo json\_encode($this->M\_kab\_kec->selectAllkabupaten());
23. }
25. public function load\_kab\_by(){
26. echo json\_encode($this->M\_kab\_kec->kabId($this->input->post('id\_kab')));
27. }
29. public function load\_data\_to\_tabel(){
30. $v\_jenis = $this->input->post('id\_jenis');
31. $v\_kab = $this->input->post('id\_kab');
32. if ( (strlen($v\_kab) != 0) && (strlen($v\_jenis)  != 0) ) {
33. $data = $this->M\_tempat\_ibadah->selectByKabJenis($v\_kab,$v\_jenis);
34. }elseif((strlen($v\_kab) == 0) && (strlen($v\_jenis) != 0)){
35. $data = $this->M\_tempat\_ibadah->selectByJenis($v\_jenis);
36. }elseif((strlen($v\_kab) != 0) && (strlen($v\_jenis) == 0)){
37. $data = $this->M\_tempat\_ibadah->selectByKabupaten($v\_kab);
38. }else{
39. $data = $this->M\_tempat\_ibadah->selectAlltempatIbadah();
40. }
41. echo json\_encode($data);
42. }
43. }

*Source code* 4.6 *class* C\_tempat\_ibadah US-06

1. <?php
2. class M\_tempat\_ibadah extends CI\_model {
3. public function selectAlltempatIbadah(){
4. $sql='SELECT \* FROM tb\_ti LEFT JOIN tb\_kabupaten ON tb\_ti.ti\_kabupaten = tb\_kabupaten.kab\_id LEFT JOIN tb\_kecamatan ON tb\_kecamatan.id\_kab = tb\_kabupaten.kab\_id AND tb\_kecamatan.kec\_id = tb\_ti.ti\_kecamatan JOIN tb\_jenis ON tb\_ti.ti\_jenis = tb\_jenis.jenis\_id';
5. return $query=$this->db->query($sql)->result\_array();
6. }
7. public function selectByKabupaten($kab){
8. $sql='SELECT \* FROM tb\_ti LEFT JOIN tb\_kabupaten ON tb\_ti.ti\_kabupaten = tb\_kabupaten.kab\_id LEFT JOIN tb\_kecamatan ON tb\_kecamatan.id\_kab = tb\_kabupaten.kab\_id AND tb\_kecamatan.kec\_id = tb\_ti.ti\_kecamatan JOIN tb\_jenis ON tb\_ti.ti\_jenis = tb\_jenis.jenis\_id WHERE ti\_kabupaten =?' ;
9. return $this->db->query($sql,$kab)->result\_array();    }
11. public function selectByJenis($jenis){
12. $sql='SELECT \* FROM tb\_ti LEFT JOIN tb\_kabupaten ON tb\_ti.ti\_kabupaten = tb\_kabupaten.kab\_id LEFT JOIN tb\_kecamatan ON tb\_kecamatan.id\_kab = tb\_kabupaten.kab\_id AND tb\_kecamatan.kec\_id = tb\_ti.ti\_kecamatan JOIN tb\_jenis ON tb\_ti.ti\_jenis = tb\_jenis.jenis\_id WHERE ti\_jenis =?' ;
13. return $this->db->query($sql,$jenis)->result\_array();
14. }
15. public function selectByKabJenis($kab, $jenis){
16. $sql='SELECT \* FROM tb\_ti LEFT JOIN tb\_kabupaten ON tb\_ti.ti\_kabupaten = tb\_kabupaten.kab\_id LEFT JOIN tb\_kecamatan ON tb\_kecamatan.id\_kab = tb\_kabupaten.kab\_id AND tb\_kecamatan.kec\_id = tb\_ti.ti\_kecamatan JOIN tb\_jenis ON tb\_ti.ti\_jenis = tb\_jenis.jenis\_id WHERE ti\_kabupaten =? AND ti\_jenis=?' ;
17. return $this->db->query($sql, [$kab, $jenis])->result\_array();
18. }

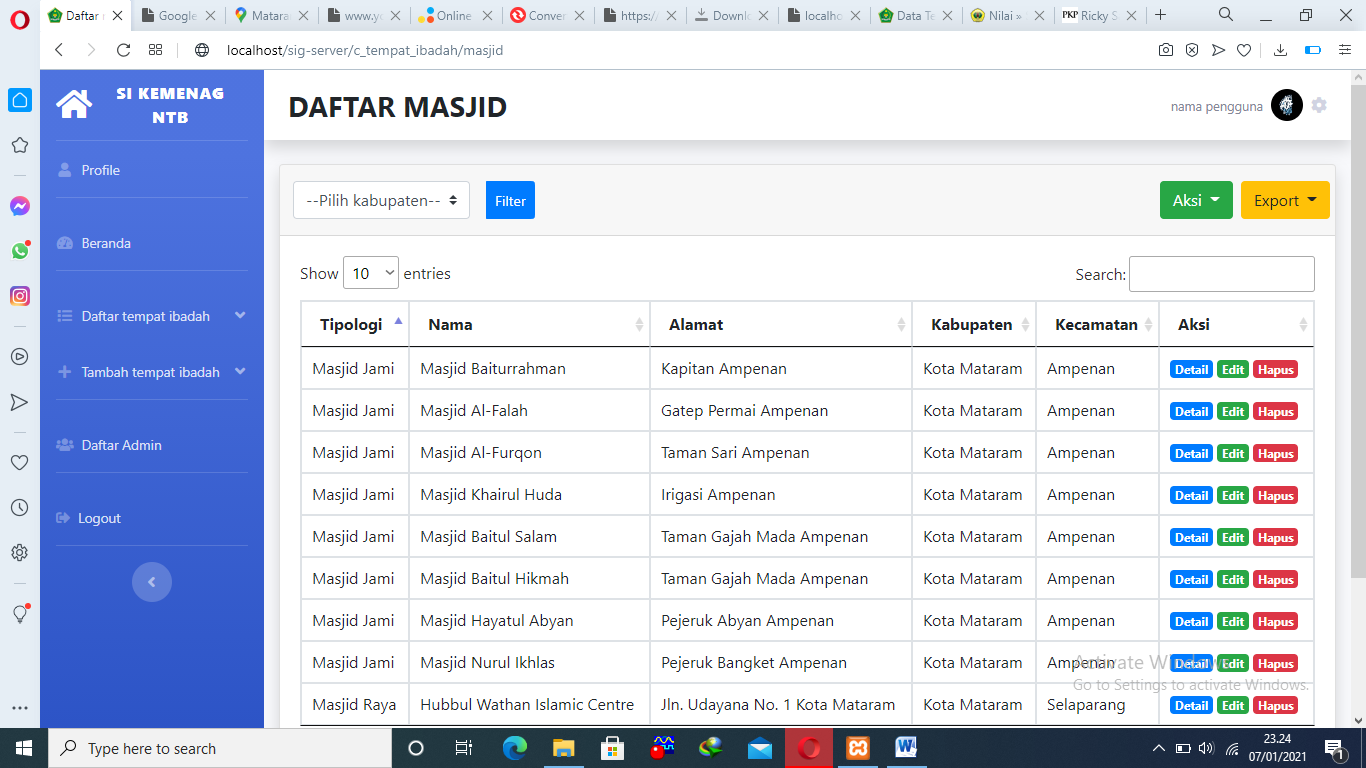
*Source code* 4.7 *class* M\_tempat\_ibadah US-06

1. <!doctype html>
2. <html lang="en" id="home">
3. <head>
4. <meta charset="utf-8">
5. <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7. <link href="<?= base\_url('assets/'); ?>vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
8. <link href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">
9. <link href="<?= base\_url('assets/'); ?>css/sb-admin-2.min.css" rel="stylesheet">
11. <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">
13. <link rel="stylesheet" type="text/css" href="<?= base\_url('assets/dashboard/'); ?>style.css">
15. <link rel="shortcut icon" type="image/x-icon" href="<?= base\_url('assets/foto/'); ?>tittle.png">
17. <!-- fonts  -->
18. <link rel="preconnect" href="https://fonts.gstatic.com">
19. <link href="https://fonts.googleapis.com/css2?family=Viga&display=swap" rel="stylesheet">
21. <link rel="stylesheet" type="text/css" href="//cdn.datatables.net/1.10.22/css/jquery.dataTables.min.css">
23. <title>Data Tempat Ibadah</title>
24. </head>
25. <body>
26. <!-- navbar -->
27. <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
28. <div class="container">
29. <a class="navbar-brand font-weight-bolder page-scroll" href="#home"> Kemenag NTB</a>
30. <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
31. <span class="navbar-toggler-icon"></span>
32. </button>
33. <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
34. <div class="navbar-nav ml-auto" >
35. <a class="nav-link page-scroll" href="#data\_tempat\_ibadah" >Data</a>
36. <a class="nav-link page-scroll" href="#peta\_tempat\_ibadah" >Map</a>
37. <a class="nav-link page-scroll" href="#contact\_us" >Contact Us</a>
38. <a class="nav-link page-scroll" href="<?= base\_url(); ?>c\_login">Log in</a>
39. </div>
40. </div>
41. </div>
42. </nav>
44. <div class="jumbotron jumbotron-fluid">
45. <div class="container">
46. <h1 class="display-4">
47. <img src="<?= base\_url('assets/foto/'); ?>kemenag\_ntb.png" class="img-fluid" alt="Responsive image">
48. </h1>
49. <p class="lead"></p>
50. </div>
51. </div>
52. <div class="container-fluid" id="data\_tempat\_ibadah">
53. <h4 class="ml-2 mt-2 font-weight-bold ml-5"> Daftar Tempat Ibadah</h4>
54. <div class="card shadow mb-4 mr-5 ml-5">
55. <div class="card-header py-3">
57. <select class="custom-select col-3 ml-auto" id="jenis\_ibadah" name="scrolljenis" required="">
58. <option value="">--Pilih jenis tempat ibadah--</option>
59. </select>
61. <div class="btn-group">
63. <select class="custom-select ml-2 mr-2" id="kabupaten" name="scrollkab" required="">
64. <option value="">--Pilih kabupaten--</option>
65. </select>
67. <button type="button" id="filter" class="btn btn-primary btn-sm ml-2 mr-2">Filter</button>
69. </div>
70. </div>
71. <div class="card-body">
72. <div class="table-responsive">
73. <table class="table table-bordered " id="data\_tabel" width="100%" cellspacing="0">
74. <thead>
75. <tr>
76. <th>Jenis</th>
77. <th>Nama</th>
78. <th>Alamat</th>
79. <th>Kabupaten</th>
80. <th>Kecamatan</th>
81. <th>Aksi</th>
82. </tr>
83. </thead>
84. </table>
85. </div>
86. </div>
87. </div>
88. </div>
89. </div>

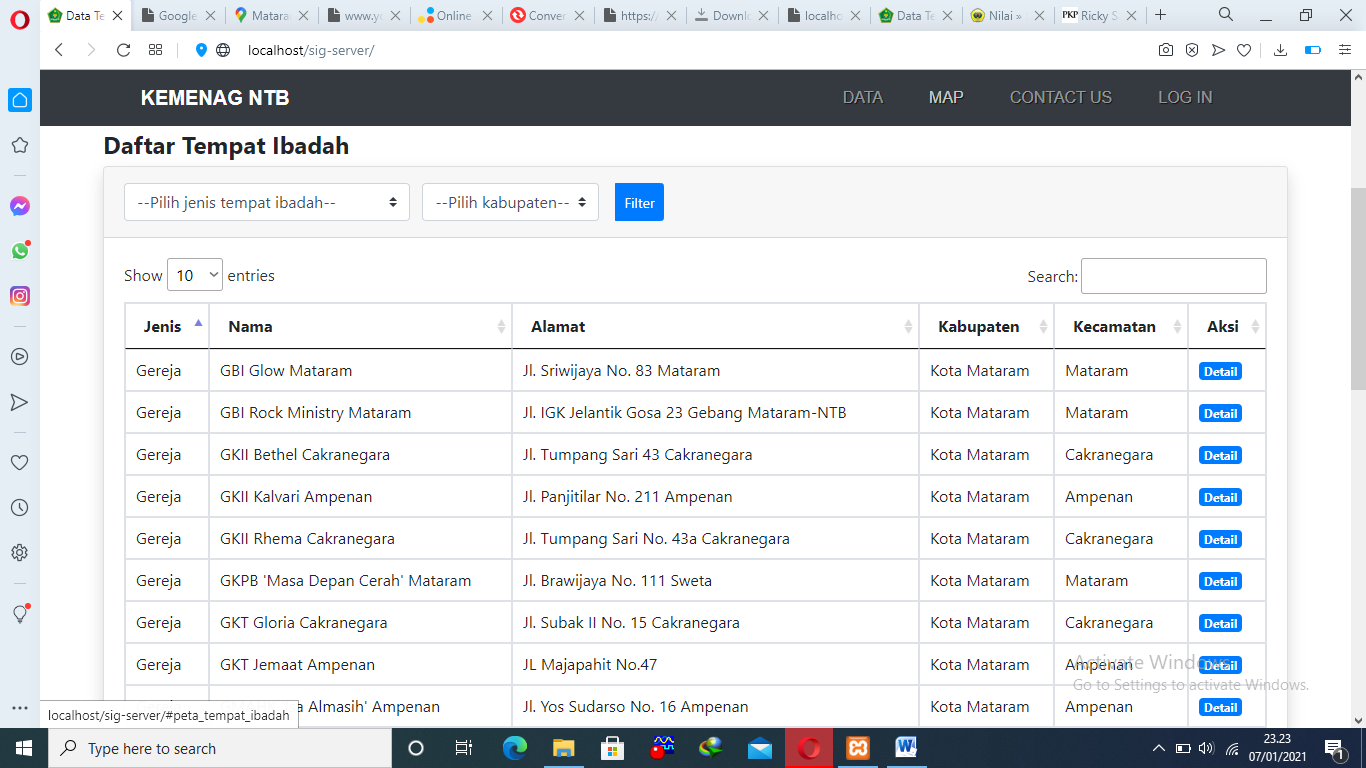
*Source code* 4.8 *class* V\_tempat\_ibadah US-06

1. Tampilan dan pengujian sistem

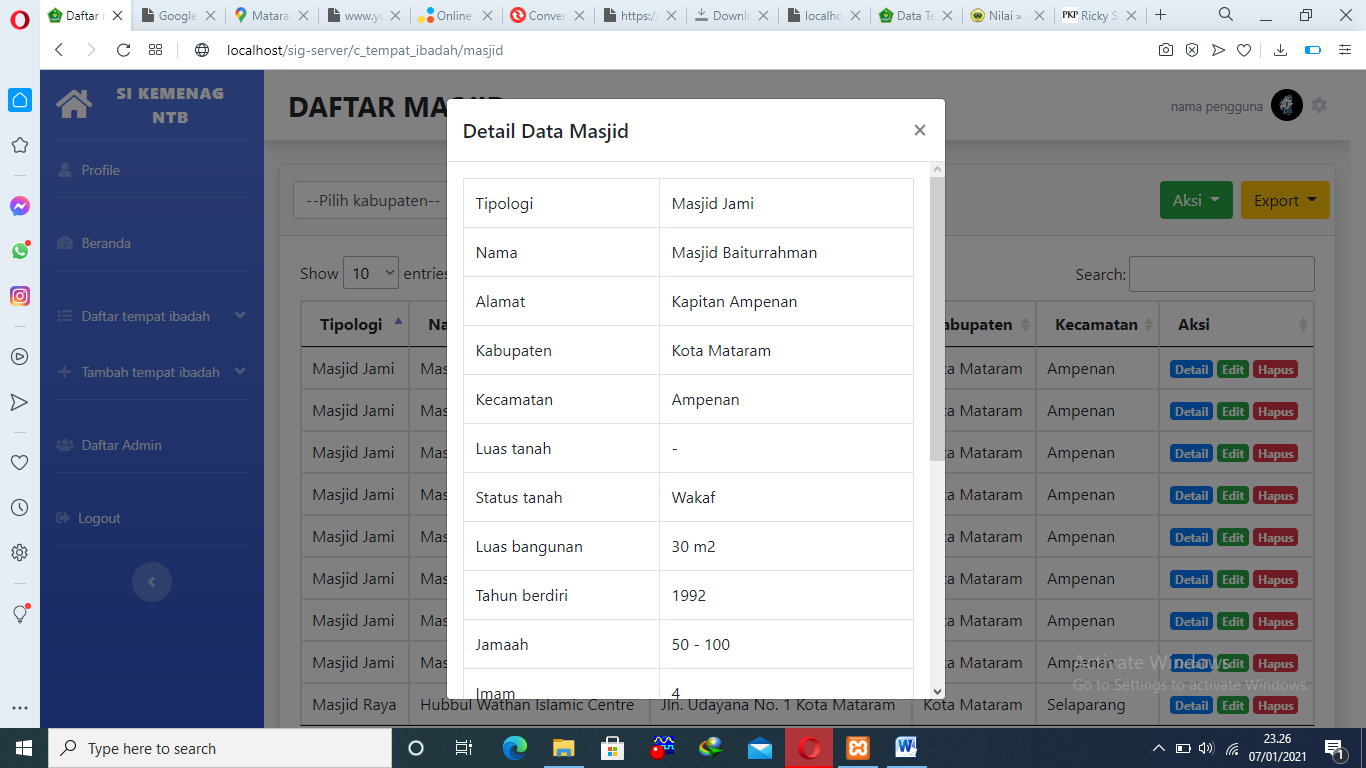
Berikut merupakan tampilan sistem dari kasus kode *user stories* US-06 yang telah dibuat dapat dilihat pada gambar berikut:



**Gambar 4.17** tampilan daftar tempat ibadah basis web sisi admin



**Gambar 4.18** tampilan daftar tempat ibadah basis *web* sisi masyarakat



**Gambar 4.19** tampilan detail data tempat ibadah basis *web*

GAMBAR ANDROID

**Gambar 4.20** tampilan daftar tempat ibadah basis *android*

GAMBAR ANDROID

**Gambar 4.21** tampilan detail tempat ibadah basis *android*

Pengujian fungsi pada kasus kode *user stories* US-06 ini dilakukan untuk menguji apakah fungsi untuk menampilkan daftar dan detail data tempat ibadah sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.8

Tabel 4.8 pengujian menampilkan data tempat ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol halaman data tempat ibadah (*web* atau *android*) | Data tempat ibadah berhasil ditampilkan | Sesuai | Valid |
| Menekan tombol detail data tempat ibadah yang dipilih (*web* atau *android*) | Detail data tempat ibadah yang dipilih berhasil ditampilkan | Sesuai | Valid |
| Memasukkan *keyword* pada fitur pencarian terhadap data tempat ibadah (*web* atau *android*) | Menampilkan hasil *keyword* pencarian | Sesuai | Valid |
| Menggunakan filter kabupaten dan kecamatan (*web*) | Menampilkan hasil tempat ibadah berdasarkan filter yang diiginkan | Sesuai | Valid |
| Menggunakan filter jenis tempat ibadah (*android*) | Menampilkan hasil tempat ibadah berdasarkan filter yang diiginkan | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.8 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. **Iteration 2**

Pada *iteration* 2 dilakukan pembuatan system pada kode *user stories*  US-2 yaitu Melihat data admin, US-03 yaitu Menambah data admin, US-04 yaitu Mengubah data admin, US-05 yaitu Menghapus data admin, dan US-07 yaitu Menambah data tempat ibadah.

1. **Kode *user stories* US-02**

Pada kasus kode *user stories* US-02 ini bidang inmas (super admin) dapat melihat daftar admin yang tersusun dalam bentuk tabel. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-02 digambarkan pada tabel 3 berikut.

Tabel 4.9 kebutuhan system kode *user stories* US-02

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Halaman muka | Menu tab : profile, beranda, daftar tempat ibadah, tambah data tempat ibadah, daftar admin, logout.Menampilkan poto dan nama profile admin yang telah *login*Menampilkan tabel data adminTerdapat tombol detail, edit, dan hapus serta tambah data adminFungsi pencarian data tempat ibadah |
| Tampilan detail data admin | Menampilkan detail data admin |

1. Desain system

Dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-02 dapat dilihat pada Tabel 4.10.

Tabel 4.10 kebutuhan system kode *user stories* US-02

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-02 | Admin | Gambar 3.25 |

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-02 yang telah dibuat dapat dilihat pada listing code berikut:

*Listing* 4.3 *Source code* pada *class model*

|  |
| --- |
| <?php  class M\_admin extends CI\_model {      public function selectAlladmin($id){          $sql='SELECT \* FROM tb\_admin JOIN tb\_bidang ON tb\_admin.admin\_bidang = tb\_bidang.bidang\_id WHERE admin\_id!=?';          return $this->db->query($sql, $id)->result\_array();      }  } |

*Listing* 4.4 *Source code* pada *class controller*

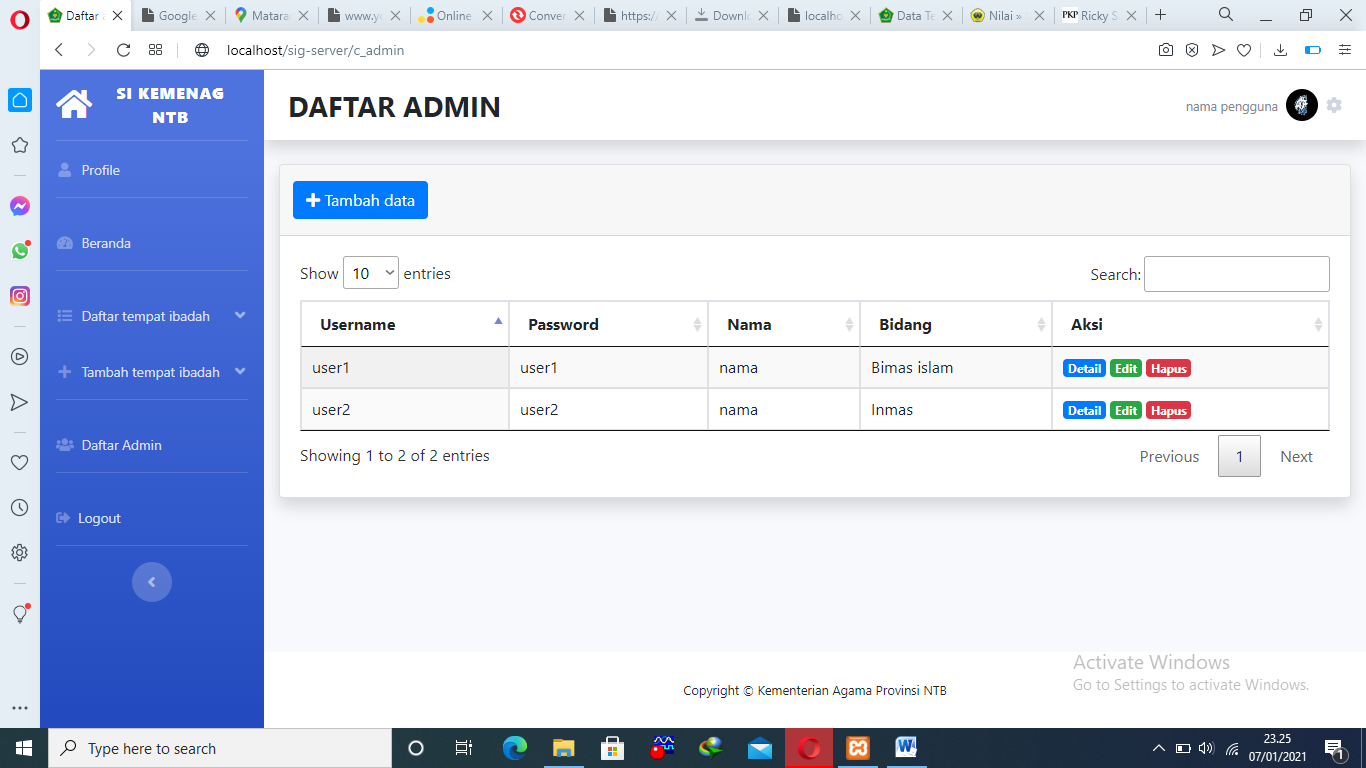
|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class C\_admin extends CI\_Controller {      public $var;      public function \_\_construct(){          parent::\_\_construct();          cek\_login();      }      public function load\_admin(){            echo json\_encode($this->M\_admin->selectAlladmin($this->session->userdata('id\_username')));      }      public function index(){          $v\_data['judul'] = 'DAFTAR ADMIN';          $v\_role = $this->session->userdata('role');          if ($v\_role == 1) {              $v\_data['hilangkan'] = '';          }else{              $v\_data['hilangkan'] = 'd-none';          }          $v\_id\_username = $this->session->userdata('id\_username');          $v\_data['data\_pengguna'] = $this->M\_admin->get\_pengguna($v\_id\_username);          $v\_data['list\_admin'] = $this->M\_admin->selectAlladmin($this->session->userdata('id\_username'));          $v\_data['tittle'] = 'Daftar admin';          $this->load->view('templates/header', $v\_data);          $this->load->view('templates/sidebar', $v\_data);          $this->load->view('templates/topbar', $v\_data);          $this->load->view('v\_daftar\_admin/index', $v\_data);          $this->load->view('templates/footer');      }  } |

Listing 4.5 *Source code* pada *class view*

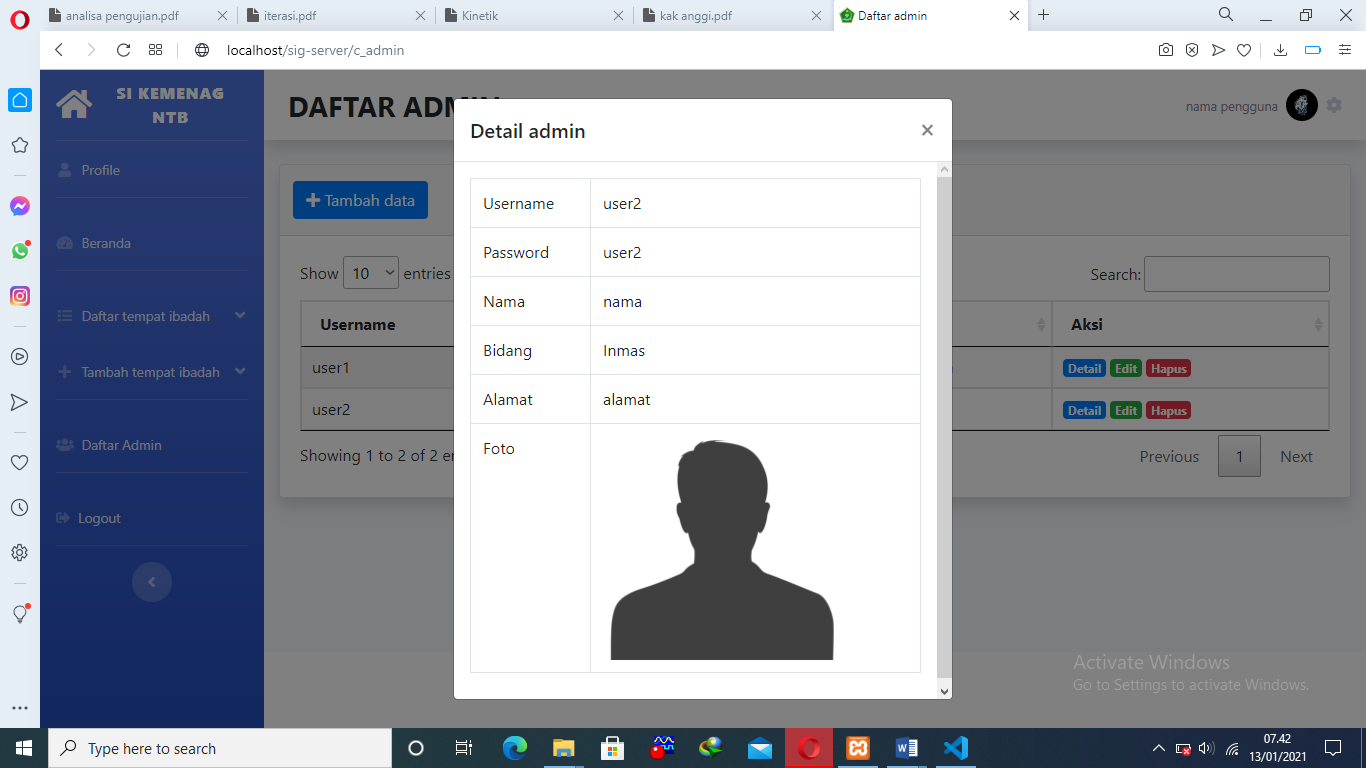
|  |
| --- |
| <div class="container-fluid">  <?= $this->session->flashdata('pesan'); ?>  <div class="card shadow mb-4">  <div class="card-header py-3">                <div class="row">                      <div class="d-sm-block"></div>                    <ul class="navbar-nav ml-2">                      <a href="<?= base\_url('c\_admin/tambah'); ?>" class="btn btn-primary mr-auto mr-lg-0"><i class="fa fa-plus"></i> Tambah data</a>                    </ul>                </div>              </div>              <div class="card-body">                <div class="table-responsive">                  <table class="table table-bordered display" id="data\_tabel" width="100%" cellspacing="0">                    <thead>                      <tr>                        <th>Username</th>                        <th>Password</th>                        <th>Nama </th>                        <th>Bidang</th>                        <th>Aksi</th>                      </tr>                    </thead>                  </table>                </div>              </div>            </div>          </div>          <!-- /.container-fluid -->  <?php foreach($list\_admin as $admin ) : ?>    <div class="modal fade" id="modal\_detail<?= $admin['admin\_id']; ?>" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">    <div class="modal-dialog modal-dialog-scrollable">        <div class="modal-content">          <div class="modal-header">            <h5 class="modal-title" id="exampleModalLabel">Detail admin</h5>            <button type="button" class="close" data-dismiss="modal" aria-label="Close">              <span aria-hidden="true">&times;</span>            </button>          </div>          <div class="modal-body">                    <table class="table table-bordered">                      <tbody>                        <tr>                          <td>Username </td>                          <td><?= $admin['admin\_username']; ?></td>                        </tr>                        <tr>                          <td>Password</td>                          <td><?= $admin['admin\_password']; ?></td>                        </tr>                        <tr>                          <td>Nama</td>                          <td><?= $admin['admin\_nama']; ?></td>                        </tr>                        <tr>                          <td>Bidang</td>                          <td><?= $admin['bidang\_nama']; ?></td>                        </tr>                        <tr>                          <td>Alamat</td>                          <td><?= $admin['admin\_alamat']; ?></td>                        </tr>                        <tr>                          <td>Foto</td>                          <td><img src="<?= base\_url('assets/foto/admin/').$admin['admin\_foto']; ?>" width="100" class="card-img" style="width: 15rem;"> </td>                        </tr>                      </tbody>                    </table>          </div>        </div>      </div>    </div>  <?php endforeach; ?>        </div>        <!-- End of Main Content -->  <script src="https://code.jquery.com/jquery-3.5.1.min.js" integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous"></script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>  <script src="//cdn.datatables.net/1.10.22/js/jquery.dataTables.min.js"></script>  <script type="text/javascript">    function getData(kab){        $.ajax({              url: "<?php echo base\_url(); ?>c\_admin/load\_admin",              type: "post",              data: {                  id\_kab: kab              },              dataType: "json",              success: function(data) {                  // console.log(data);                   $('#data\_tabel').DataTable( {                      "data": data,                      "columns": [                          {                              "data": "admin\_username"                          },                          {                              "data": "admin\_password"                          },                          {                              "data": "admin\_nama",                          },                          {                              "data": "bidang\_nama"                          },                          {                              "data": "admin\_id",                              "render": function(data, type, row, meta) {                              return `                                <a href="${row.admin\_id}" class="badge badge-primary" data-toggle="modal" data-target="#modal\_detail${row.admin\_id}" >Detail</a> <a href='<?= base\_url('c\_admin/edit/')?>${row.admin\_id}' class="badge badge-success">Edit</a> <a href='<?= base\_url('c\_admin/hapus/')?>${row.admin\_id}' class="badge badge-danger" onclick="return confirm('Yakin dihapus?');">Hapus</a>                              `;                              }                          }                ]            } );              }          });      }      getData();  </script> |

1. Tampilan dan pengujian system

Berikut merupakan tampilan sistem dari kasus kode *user stories* US-02 yang telah dibuat dapat dilihat pada gambar berikut:



**Gambar 4.10** tampilan daftar data admin



**Gambar 4.10** tampilan detail data admin

Pengujian fungsi pada kasus kode *user stories* US-02 ini dilakukan untuk menguji apakah fungsi untuk menampilkan daftar dan detail data admin sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.11

Tabel 4.11 pengujian menampilkan data admin

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol halaman daftar admin | Daftar data admin berhasil ditampilkan | Sesuai | Valid |
| Menekan tombol detail data admin yang dipilih | Detail data admin yang dipilih berhasil ditampilkan | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.11 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. **Kode *user stories* US-03**

Pada kasus kode *user stories* US-03 ini admin bidang inmas dapat menambahkan data admin. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-03 digambarkan pada tabel 4.12 berikut.

Tabel 4.12 kebutuhan system kode *user stories* US-03

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Halaman muka | Menu tab : profile, beranda, daftar tempat ibadah, tambah data tempat ibadah, daftar admin, logout.Menampilkan poto dan nama profile admin yang telah *login*Menampilkan *form* tambah data adminTerdapat tombol untuk menambahkan data adminTerdapat tombol kembali ke halaman daftar data admin |

1. Desain system

Dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-03 dapat dilihat pada Tabel 4.13.

Tabel 4.13 kebutuhan system kode *user stories* US-03

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-03 | Admin | Gambar 3.29 |

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-03 yang telah dibuat dapat dilihat pada listing code berikut:

*Listing* 4.3 *Source code* pada *class model*

|  |
| --- |
| <?php  class M\_admin extends CI\_model {  public function create\_admin ($v\_data)      {          $this->db->insert('tb\_admin', $v\_data);          return $this->db->affected\_rows();      } |

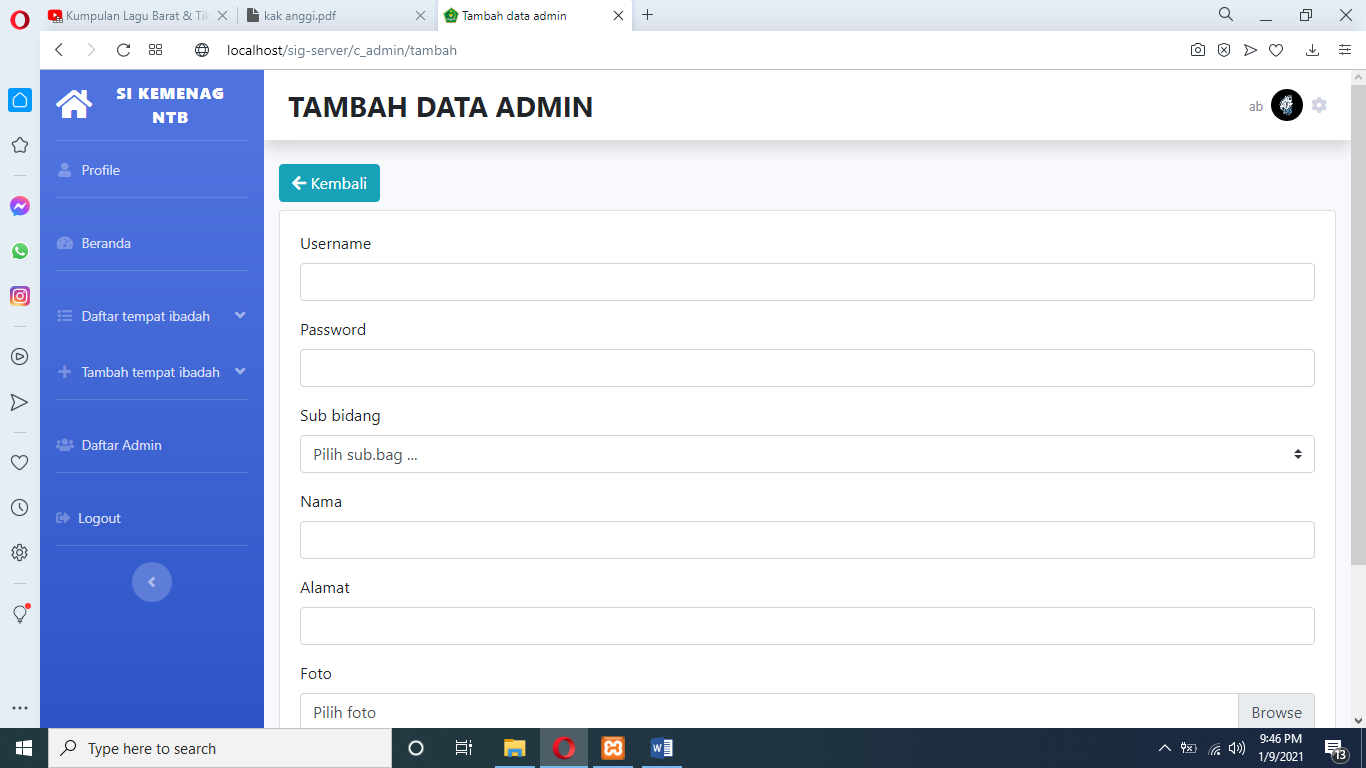
*Listing* 4.4 *Source code* pada *class controller*

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class C\_admin extends CI\_Controller {      public $var;      public function \_\_construct(){          parent::\_\_construct();          cek\_login();      }      public function tambah()      {          $v\_data['judul'] = 'TAMBAH DATA ADMIN';          $v\_role = $this->session->userdata('role');          if ($v\_role == 1) {              $v\_data['hilangkan'] = '';          }else{              $v\_data['hilangkan'] = 'd-none';          }          $v\_id\_username = $this->session->userdata('id\_username');          $v\_data['data\_pengguna'] = $this->M\_admin->get\_pengguna($v\_id\_username);          $this->form\_validation->set\_rules('bidang','Bidang','required|callback\_validasi\_bidang');            $this->form\_validation->set\_rules('username', 'Username', 'required|trim|is\_unique[tb\_admin.admin\_username]', [              'required' => 'Form tidak boleh kosong!',              'is\_unique' => 'Username telah terdaftar!'          ]);          $this->form\_validation->set\_rules('password', 'Password', 'required|trim|min\_length[3]', [              'required' => 'Form tidak boleh kosong!',              'min\_length' => 'Password terlalu pendek!'          ]);          $this->form\_validation->set\_rules('nama', 'Nama', 'required|trim', [              'required' => 'Form tidak boleh kosong!',          ]);          $v\_data['list\_bidang'] = $this->M\_bidang->selectAllbidang();          if($this->form\_validation->run() == false){              $v\_data['tittle'] = 'Tambah data admin';              $this->load->view('templates/header', $v\_data);              $this->load->view('templates/sidebar', $v\_data);              $this->load->view('templates/topbar', $v\_data);              $this->load->view('v\_daftar\_admin/tambah\_data', $v\_data);              $this->load->view('templates/footer');          }else{              $v\_idbidang = $this->input->post('bidang');              $v\_username = htmlspecialchars($this->input->post('username', true));              $v\_password = $this->input->post('password');              $v\_nama     = htmlspecialchars($this->input->post('nama', true));              $v\_alamat   = $this->input->post('alamat');              $upload\_foto = $\_FILES['foto']['name'];              if($upload\_foto){                    $config['allowed\_types'] = 'gif|jpg|png|jpeg';                  $config['max\_size']     = '5000';                  $config['upload\_path'] = './assets/foto/admin/';                    $this->load->library('upload', $config);                  if ($this->upload->do\_upload('foto')){                      $v\_nama\_foto = $this->upload->data('file\_name');                        $v\_data = [                          'admin\_username' => $v\_username,                          'admin\_password' => $v\_password,                          'admin\_nama' => $v\_nama,                          'admin\_alamat' => $v\_alamat,                          'admin\_bidang' => $v\_idbidang,                          'admin\_foto' => $v\_nama\_foto                      ];                  }                  else                  {                      echo $this->upload->display\_errors();                  }              }else{                  $v\_data = [                      'admin\_username' => $v\_username,                      'admin\_password' => $v\_password,                      'admin\_nama' => $v\_nama,                      'admin\_alamat' => $v\_alamat,                      'admin\_bidang' => $v\_idbidang,                      'admin\_foto' => 'foto.png'                  ];              }              $this->M\_admin->create\_admin($v\_data);              $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Data berhasil ditambah!</div>');              redirect('c\_admin');          }      } |

Listing 4.5 *Source code* pada *class view*

|  |
| --- |
| <div class="container-fluid">  <a href="<?= base\_url('c\_admin'); ?>" class="btn btn-info mb-2"><i class="fa fa-arrow-left"></i> Kembali</a>  <div class="card mb-3">              <div class="card-body">            <?= form\_open\_multipart('c\_admin/tambah'); ?>              <label for="basic-url">Username</label>            <div class="form-group mb-3">              <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="username" value="<?= set\_value('username'); ?>">              <?= form\_error('username', '<small class="text-danger">', '</small>'); ?>            </div>              <label for="basic-url">Password</label>            <div class="form-group mb-3">              <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="password" value="<?= set\_value('password'); ?>">              <?= form\_error('password', '<small class="text-danger">', '</small>'); ?>            </div>            <label for="basic-url">Sub bidang</label>             <div class="form-group mb-3">                        <select class="custom-select " id="inputGroupSelect01" name="bidang">                        <option value="none" selected="selected">Pilih sub.bag ...</option>  <?php foreach($list\_bidang as $bidang ) : ?>                          <option value="<?= $bidang['bidang\_id']; ?>"><?= $bidang['bidang\_nama']; ?></option>                        <?php endforeach; ?>                      </select>                      <?= form\_error('bidang', '<small class="text-danger">', '</small>'); ?>            </div>            <label for="basic-url">Nama</label>            <div class="form-group mb-3">              <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="nama" value="<?= set\_value('nama'); ?>">              <?= form\_error('nama', '<small class="text-danger">', '</small>'); ?>            </div>              <label for="basic-url">Alamat</label>            <div class="form-group mb-3">              <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="alamat" value="<?= set\_value('nama'); ?>">            </div>            <label for="basic-url">Foto</label>            <div class="custom-file">              <input type="file" accept="image/\*" class="custom-file-input" name="foto">              <label class="custom-file-label" for="customFile">Pilih foto</label>            </div>          <hr>          <div class="row float-right mr-1">              <button type="submit" class="btn btn-primary">Simpan</button>          </div>          <?= form\_close(); ?>        </div>  </div>          </div>        </div> |

1. Tampilan dan pengujian system



**Gambar 4.10** tampilan tambah data admin

Pengujian fungsi pada kasus kode *user stories* US-03 ini dilakukan untuk menguji apakah fungsi untuk menambahkan data admin sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.14

Tabel 4.14 pengujian tambah data admin

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Sub bidang tidak dipilih, serta *Username* dan *password* tidak diisi | Data tidak berhasil ditambah | Sesuai | Valid |
| Sub bidang dipilih, *Username* dan *password* diisi. Namun *username* telah terdaftar dalam sistem dan *password* kurang 3 kalimat | Data tidak berhasil ditambah | Sesuai | Valid |
| Sub bidang dipilih, serta *password* diisi dengan benar dan *username* belum tedaftar dalam *database* | Sistem akan melakukan proses dan data berhasil ditambahkan | Sesuai | Valid |

Sistem hanya akan berhasil menambahkan data jika sub bidang terpilih, dan *username* yang diinputkan belum terdaftar dalam *database* serta *password* yang diinputkan lebih dari 3 kata, selain itu sistem akan menolak akses untuk menambahkan data. Sehingga berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.14 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. **Kode *user stories* US-04**

Pada kasus kode *user stories* US-04 ini admin bidang inmas dapat mengubah data admin. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-04 digambarkan pada tabel 4.15 berikut.

Tabel 4.15 kebutuhan system kode *user stories* US-04

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Halaman muka | Menu tab : profile, beranda, daftar tempat ibadah, tambah data tempat ibadah, daftar admin, logout.Menampilkan poto dan nama profile admin yang telah *login*Menampilkan *form* edit data adminTerdapat tombol untuk mengubah data adminTerdapat tombol kembali ke halaman daftar data admin |

1. Desain system

Dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-04 dapat dilihat pada Tabel 4.15.

Tabel 4.15 kebutuhan system kode *user stories* US-04

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-04 | Admin | Gambar 3.30 |

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-04 yang telah dibuat dapat dilihat pada listing code berikut:

*Listing* 4.3 *Source code* pada *class model*

|  |
| --- |
| <?php  class M\_admin extends CI\_model {  public function edit\_admin($id, $data){          $this->db->update('tb\_admin', $data, array('admin\_id' => $id));      } |

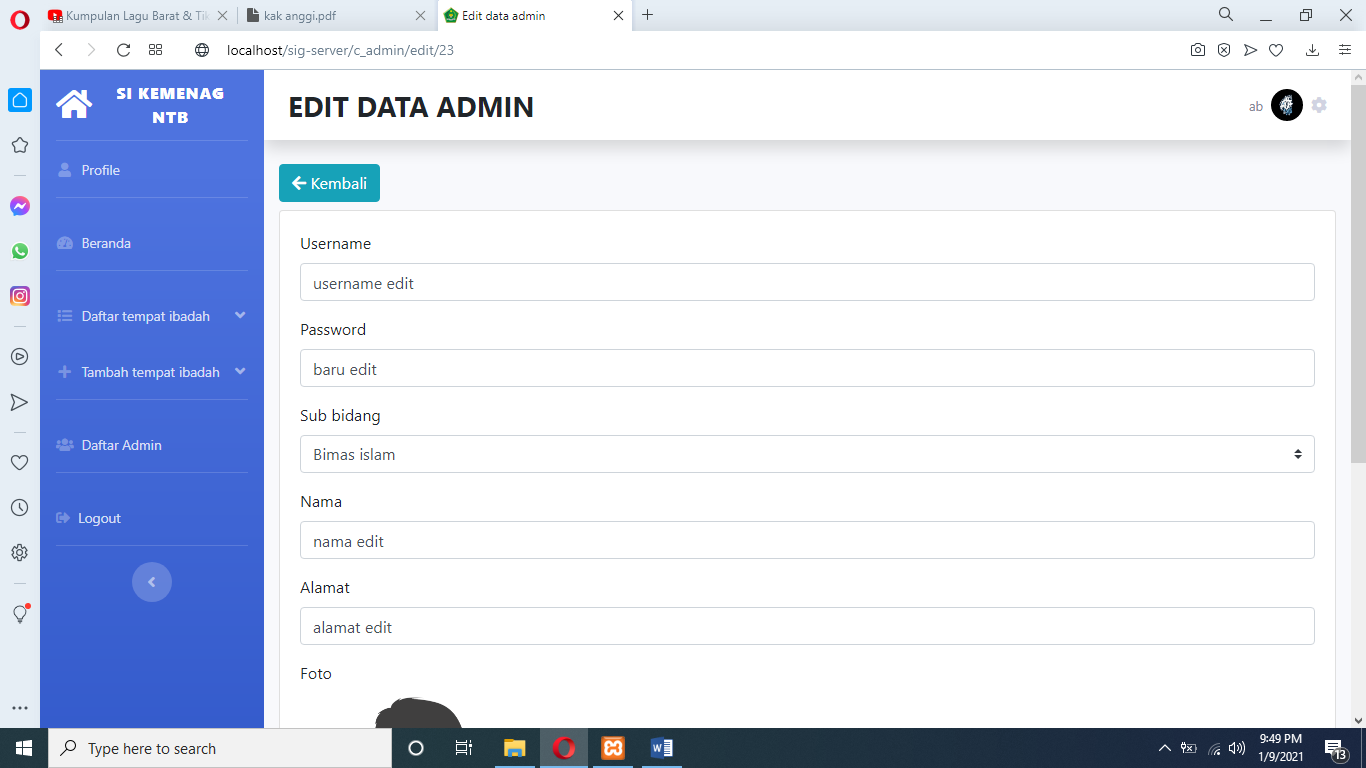
*Listing* 4.4 *Source code* pada *class controller*

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class C\_admin extends CI\_Controller {      public $var;      public function \_\_construct(){          parent::\_\_construct();          cek\_login();      }      public function edit($v\_id) {          $v\_data['judul'] = 'EDIT DATA ADMIN';          $v\_role = $this->session->userdata('role');          if ($v\_role == 1) {              $v\_data['hilangkan'] = '';          }else{              $v\_data['hilangkan'] = 'd-none';          }          $v\_id\_username = $this->session->userdata('id\_username');          $v\_data['data\_pengguna'] = $this->M\_admin->get\_pengguna($v\_id\_username);          $v\_data['list\_bidang'] = $this->M\_bidang->selectAllbidang();          $v\_data['tittle'] = 'Edit data admin';          $v\_data['admin'] = $this->M\_admin->get\_pengguna($v\_id);          $this->form\_validation->set\_rules('bidang','Bidang','required|callback\_validasi\_bidang');          $this->form\_validation->set\_rules('nama', 'Nama', 'required|trim', [              'required' => 'Form tidak boleh kosong!',          ]);          $this->form\_validation->set\_rules('password', 'Password', 'required|trim|min\_length[3]', [              'required' => 'Form tidak boleh kosong!',              'min\_length' => 'Password terlalu pendek!'          ]);          $this->form\_validation->set\_rules('username', 'Username', 'required|trim', [              'required' => 'Form tidak boleh kosong!',          ]);          if($this->form\_validation->run() == false){              $this->load->view('templates/header', $v\_data);              $this->load->view('templates/sidebar', $v\_data);              $this->load->view('templates/topbar', $v\_data);              $this->load->view('v\_daftar\_admin/edit\_data', $v\_data);              $this->load->view('templates/footer');          }else{              $id\_admin = $this->input->post('id');              $v\_idbidang = $this->input->post('bidang');              $username\_input = htmlspecialchars($this->input->post('username', true));              $v\_nama = htmlspecialchars($this->input->post('nama', true));              $v\_password = $this->input->post('password');              $v\_alamat = htmlspecialchars($this->input->post('alamat', true));              $upload\_foto = $\_FILES['foto']['name'];              if($this->M\_admin->cek\_username($username\_input,$id\_admin)){                  $this->session->set\_flashdata('pesan', '<div class="alert alert-danger" role="alert">Gagal mengubah! Username telah terdaftar!</div>');                  redirect('c\_admin/edit/'.$id\_admin);              }else{                    if($upload\_foto){                        $config['allowed\_types'] = 'gif|jpg|png|jpeg';                      $config['max\_size']     = '5000';                      $config['upload\_path'] = './assets/foto/admin/';                        $this->load->library('upload', $config);                        if ($this->upload->do\_upload('foto')){                          $v\_nama\_foto = $this->upload->data('file\_name');                            $v\_foto\_lama = $v\_data['admin']['admin\_foto'];                            if($v\_foto\_lama != 'foto.png'){                              unlink(FCPATH . 'assets/foto/admin/' . $v\_foto\_lama);                          }                            $v\_data = [                              'admin\_password' => $v\_password,                              'admin\_nama' => $v\_nama,                              'admin\_alamat' => $v\_alamat,                              'admin\_foto' => $v\_nama\_foto,                              'admin\_username' => $username\_input,                              'admin\_bidang' => $v\_idbidang                          ];                      }                      else                      {                          echo $this->upload->display\_errors();                      }                    }else{                      $v\_data = [                          'admin\_password' => $v\_password,                          'admin\_nama' => $v\_nama,                          'admin\_alamat' => $v\_alamat,                          'admin\_username' => $username\_input,                          'admin\_bidang' => $v\_idbidang                      ];                  }                    $this->M\_admin->edit\_admin($id\_admin, $v\_data);                  $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Data berhasil diubah!</div>');                  redirect('c\_admin');              }          }      } |

Listing 4.5 *Source code* pada *class view*

|  |
| --- |
| <div class="container-fluid">            <a href="<?= base\_url('c\_admin'); ?>" class="btn btn-info mb-2"><i class="fa fa-arrow-left"></i> Kembali</a>              <?= $this->session->flashdata('pesan'); ?>              <div class="card mb-3">              <div class="card-body">            <?= form\_open\_multipart(); ?>          <input type="hidden" id="id" name="id" value="<?= $admin['admin\_id']; ?>">            <label for="basic-url">Username</label>            <div class="form-group mb-3">              <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="username" value="<?= $admin['admin\_username']; ?>">              <?= form\_error('username', '<small class="text-danger">', '</small>'); ?>            </div>            <label for="basic-url">Password</label>            <div class="form-group mb-3">              <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="password" value="<?= $admin['admin\_password']; ?>">              <?= form\_error('password', '<small class="text-danger">', '</small>'); ?>            </div>            <label for="basic-url">Sub bidang</label>             <div class="form-group mb-3">                        <select class="custom-select " id="inputGroupSelect01" name="bidang">                        <option value="none" selected="selected">Pilih sub.bag ...</option>                        <?php foreach($list\_bidang as $bidang ) : ?>                          <?php if($bidang['bidang\_id'] == $admin['admin\_bidang']) : ?>                            <option value="<?= $bidang['bidang\_id']; ?>" selected><?= $bidang['bidang\_nama']; ?></option>                          <?php else : ?>                            <option value="<?= $bidang['bidang\_id']; ?>"><?= $bidang['bidang\_nama']; ?></option>                          <?php endif; ?>                        <?php endforeach; ?>                      </select>                      <?= form\_error('bidang', '<small class="text-danger">', '</small>'); ?>            </div>            <label for="basic-url">Nama</label>            <div class="form-group mb-3">              <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="nama" value="<?= $admin['admin\_nama']; ?>">              <?= form\_error('nama', '<small class="text-danger">', '</small>'); ?>            </div>              <label for="basic-url">Alamat</label>            <div class="form-group mb-3">              <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="alamat" value="<?= $admin['admin\_alamat']; ?>">            </div>            <label for="basic-url">Foto</label>            <div class="mb-2">            <img src="<?= base\_url('assets/foto/admin/').$admin['admin\_foto']; ?>" width="100" class="card-img" style="width: 15rem;">            </div>            <div class="custom-file">                <input type="file" accept="image/\*" class="custom-file-input" name="foto">              <label class="custom-file-label" for="customFile">Ganti foto</label>            </div>          <hr>          <div class="row float-right mr-1">              <button type="submit" class="btn btn-primary">Simpan</button>          </div>          <?= form\_close(); ?>        </div>  </div>          </div>        </div> |

1. Tampilan dan pengujian system



**Gambar 4.10** tampilan edit data admin

Pengujian fungsi pada kasus kode *user stories* US-04 ini dilakukan untuk menguji apakah fungsi untuk mengubah data admin sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.16

Tabel 3.16 pengujian mengubah data admin

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Sub bidang tidak dipilih, serta *Username* atau *password* tidak diisi | Data tidak berhasil diubah | Sesuai | Valid |
| Sub bidang dipilih, *Username* dan *password* diisi. Namun *username* telah terdaftar dalam sistem dan *password* kurang 3 kalimat | Data tidak berhasil diubah | Sesuai | Valid |
| Sub bidang dipilih, serta *password* diisi dengan benar dan *username* belum tedaftar dalam *database* | Sistem akan melakukan proses dan data berhasil diubah | Sesuai | Valid |

Sistem hanya akan berhasil mengubah data admin jika sub bidang terpilih, dan *username* yang diinputkan belum terdaftar dalam *database* serta *password* yang diinputkan lebih dari 3 kata, selain itu sistem akan menolak akses untuk mengubah data admin. Sehingga berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.17 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. **Kode *user stories* US-05**

Pada kasus kode *user stories* US-05 ini admin bidang inmas dapat menghapus data admin. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-05 digambarkan pada tabel 4.18 berikut.

Tabel 4.18 kebutuhan system kode *user stories* US-05

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Tampilan sebelum menghapus data | Terdapat tombol batal dan tombol hapus untuk menghapus data admin |

1. Desain system

Dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-05 dapat dilihat pada Tabel 4.19.

Tabel 4.19 kebutuhan system kode *user stories* US-05

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-05 | Admin | Gambar 3.31 |

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-04 yang telah dibuat dapat dilihat pada listing code berikut:

*Listing* 4.3 *Source code* pada *class model*

|  |
| --- |
| <?php  class M\_admin extends CI\_model {  public function hapus\_admin($v\_id) {          $this->db->where('admin\_id', $v\_id);          $this->db->delete('tb\_admin');      } |

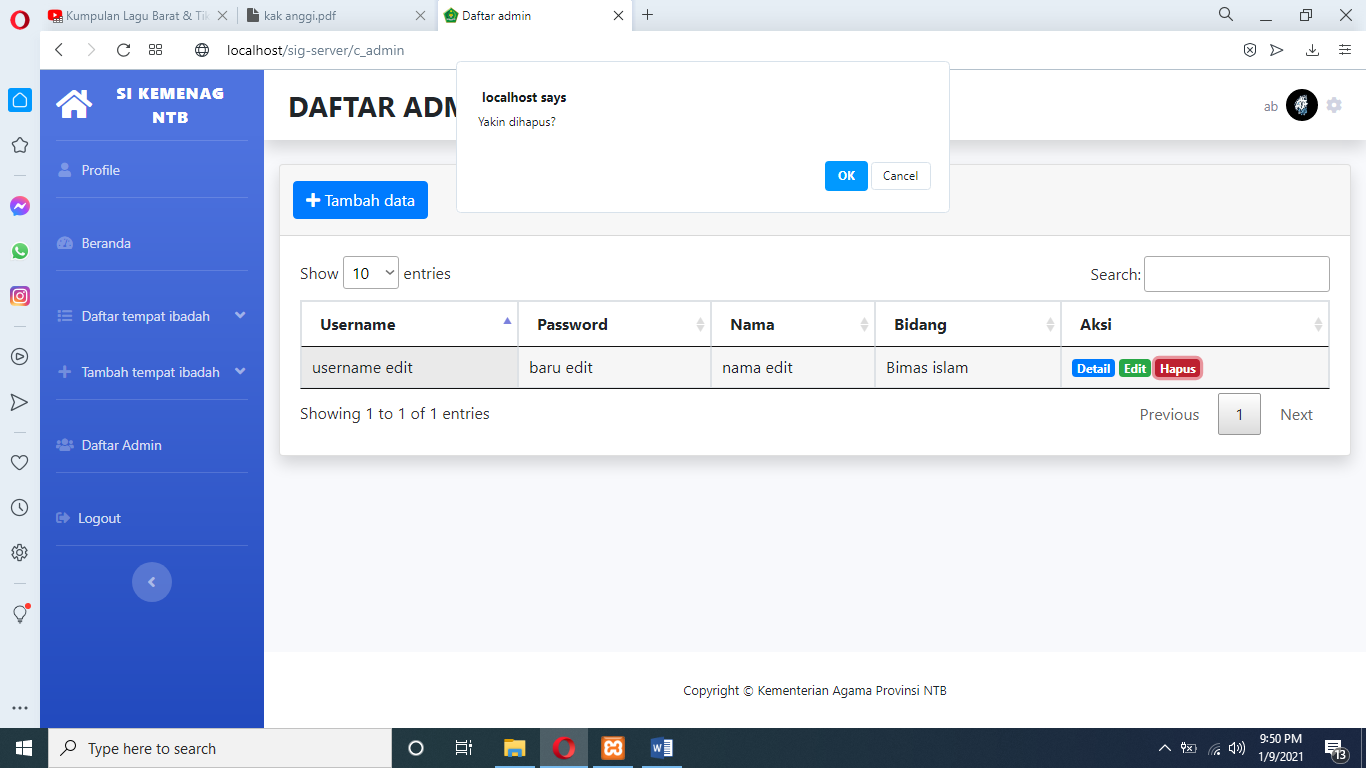
*Listing* 4.4 *Source code* pada *class controller*

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class C\_admin extends CI\_Controller {      public $var;      public function \_\_construct(){          parent::\_\_construct();          cek\_login();      }      public function hapus($v\_id){          $hapus\_foto = $this->M\_admin->get\_pengguna($v\_id);          if($hapus\_foto['ti\_foto'] != 'klenteng.png'){              unlink(FCPATH . 'assets/foto/admin/' . $hapus\_foto['admin\_foto']);          }          $this->M\_admin->hapus\_admin($v\_id);          $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Data berhasil dihapus!</div>');          redirect('c\_admin');      } |

Listing 4.5 *Source code* pada *class view*

|  |
| --- |
| <a href='<?= base\_url('c\_admin/hapus/')?>${row.admin\_id}' class="badge badge-danger" onclick="return confirm('Yakin dihapus?');">Hapus</a> |

1. Tampilan dan pengujian system



**Gambar 4.10** tampilan hapus data admin

Pengujian fungsi pada kasus kode *user stories* US-05 ini dilakukan untuk menguji apakah fungsi untuk mengubah data admin sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.20

Tabel 4.20 pengujian mengahpus data admin

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol “*Cancel*” | Data tidak terhapus | Sesuai | Valid |
| Menekan tombol “*Ok*” | Data terhapus dalam sistem | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.20 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. **Kode *user stories* US-07**

Pada kasus kode *user stories* US-07 ini admin dapat menambah data tempat ibadah. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-07 digambarkan pada tabel 3 berikut.

Tabel 4.21 kebutuhan system kode *user stories* US-07

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Halaman muka | Menu tab : profile, beranda, daftar tempat ibadah, tambah data tempat ibadah, daftar admin, logout.Menampilkan poto dan nama profile admin yang telah *login*Menampilkan *form* tambah data tempat ibadahMenampilkan peta dan *marker* untuk menentukan titik lokasi tempat ibadahTerdapat tombol untuk mengambil lokasi terkiniTerdapat tombol untuk menambahkan data tempat ibadahTerdapat tombol kembali ke halaman daftar data tempat ibadah |

1. Desain system

Dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-07 dapat dilihat pada Tabel 4.22.

Tabel 4.23 kebutuhan system kode *user stories* US-07

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-07 | Tempat ibadah | Gambar 3.24 |

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-07 yang telah dibuat dapat dilihat pada listing code berikut:

*Listing* 4.3 *Source code* pada *class model*

|  |
| --- |
| <?php  class M\_tempat\_ibadah extends CI\_model {  public function create\_ti($v\_data)      {          $this->db->insert('tb\_ti', $v\_data);          return $this->db->affected\_rows();      } |

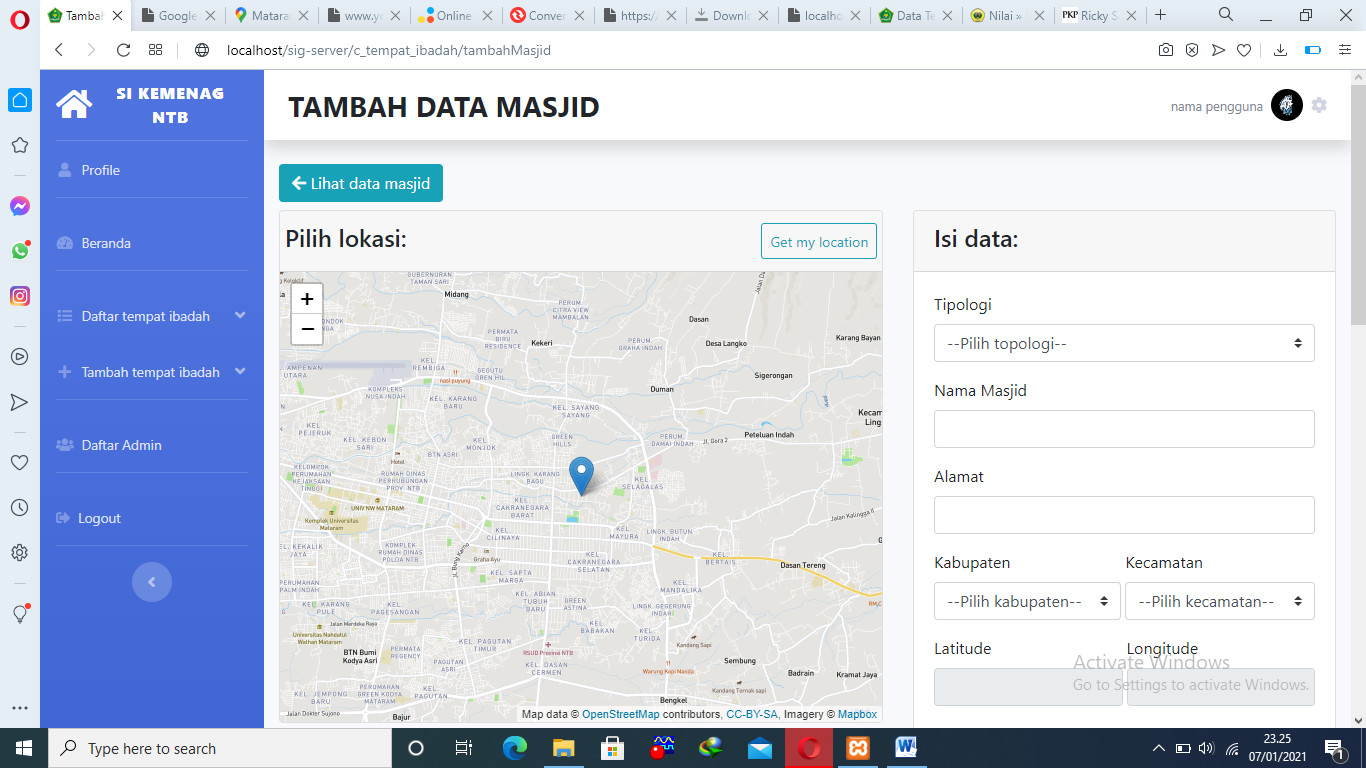
*Listing* 4.4 *Source code* pada *class controller*

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class C\_tempat\_ibadah extends CI\_Controller {      public $var;      public function \_\_construct(){          parent::\_\_construct();          cek\_login();      }      public function tambahMasjid(){          $v\_data['judul'] = 'TAMBAH DATA MASJID';          $v\_role = $this->session->userdata('role');          if ($v\_role == 1) {              $v\_data['hilangkan'] = '';          }else{              $v\_data['hilangkan'] = 'd-none';          }          $v\_id\_username = $this->session->userdata('id\_username');          $v\_data['data\_pengguna'] = $this->M\_admin->get\_pengguna($v\_id\_username);          $v\_data['list\_kabupaten'] = $this->M\_kab\_kec->selectAllkabupaten();          $v\_data['tittle'] = 'Tambah data masjid';          $this->form\_validation->set\_rules('scrollkab','Scrollkab','required|callback\_validasi\_kabupaten');          $this->form\_validation->set\_rules('scrollkec','Scrollkec','required|callback\_validasi\_kecamatan');          $this->form\_validation->set\_rules('scrolltip','Scrolltip','required|callback\_validasi\_tipologi');          $this->form\_validation->set\_rules('nama\_masjid', 'Nama\_masjid', 'required|trim', [              'required' => 'Form tidak boleh kosong!',          ]);          $this->form\_validation->set\_rules('alamat', 'Alamat', 'required|trim', [              'required' => 'Form tidak boleh kosong!',          ]);          if($this->form\_validation->run() == false){              $this->load->view('templates/header', $v\_data);              $this->load->view('templates/sidebar',$v\_data);              $this->load->view('templates/topbar', $v\_data);              $this->load->view('v\_daftar\_tempat\_ibadah/masjid/tambah\_masjid', $v\_data);              $this->load->view('templates/footer');          }else{              $v\_kabupaten = $this->input->post('scrollkab');              $v\_kecamatan = $this->input->post('scrollkec');              $v\_tipologi = $this->input->post('scrolltip');              $v\_nama     = $this->input->post('nama\_masjid');              $v\_alamat = $this->input->post('alamat');              $v\_luas\_tanah = $this->input->post('luas\_tanah');              $v\_status\_tanah = $this->input->post('status\_tanah');              $v\_luas\_bangunan = $this->input->post('luas\_bangunan');              $v\_tahun\_berdiri = $this->input->post('tahun\_berdiri');              $v\_jamaah = $this->input->post('jamaah');              $v\_imam = $this->input->post('imam');              $v\_khatib = $this->input->post('khatib');              $v\_remaja = $this->input->post('remaja');              $v\_telepon = $this->input->post('telepon');              $v\_longitude = $this->input->post('longitude');              $v\_latitude = $this->input->post('latitude');              $upload\_foto = $\_FILES['foto']['name'];              if($upload\_foto){                    $config['allowed\_types'] = 'gif|jpg|png|jpeg';                  $config['max\_size']     = '5000';                  $config['upload\_path'] = './assets/foto/tempat\_ibadah/';                    $this->load->library('upload', $config);                  if ($this->upload->do\_upload('foto')){                      $v\_nama\_foto = $this->upload->data('file\_name');                        $v\_data = [                          'ti\_jenis' => '1',                          'ti\_tipologi' => $v\_tipologi,                          'ti\_nama' => $v\_nama,                          'ti\_alamat' => $v\_alamat,                          'ti\_kabupaten' => $v\_kabupaten,                          'ti\_kecamatan' => $v\_kecamatan,                          'ti\_luas\_tanah' => $v\_luas\_tanah,                          'ti\_status\_tanah' => $v\_status\_tanah,                          'ti\_luas\_bangunan' => $v\_luas\_bangunan,                          'ti\_tahun\_berdiri' => $v\_tahun\_berdiri,                          'ti\_jamaah' => $v\_jamaah,                          'ti\_imam' => $v\_imam,                          'ti\_khatib' => $v\_khatib,                          'ti\_remaja' => $v\_remaja,                          'ti\_telepon' => $v\_telepon,                          'longitude' => $v\_longitude,                          'latitude' => $v\_latitude,                          'ti\_foto' => $v\_nama\_foto                      ];                  }                  else                  {                      echo $this->upload->display\_errors();                  }              }else{                  $v\_data = [                          'ti\_jenis' => '1',                          'ti\_tipologi' => $v\_tipologi,                          'ti\_nama' => $v\_nama,                          'ti\_alamat' => $v\_alamat,                          'ti\_kabupaten' => $v\_kabupaten,                          'ti\_kecamatan' => $v\_kecamatan,                          'ti\_luas\_tanah' => $v\_luas\_tanah,                          'ti\_status\_tanah' => $v\_status\_tanah,                          'ti\_luas\_bangunan' => $v\_luas\_bangunan,                          'ti\_tahun\_berdiri' => $v\_tahun\_berdiri,                          'ti\_jamaah' => $v\_jamaah,                          'ti\_imam' => $v\_imam,                          'ti\_khatib' => $v\_khatib,                          'ti\_remaja' => $v\_remaja,                          'ti\_telepon' => $v\_telepon,                          'longitude' => $v\_longitude,                          'latitude' => $v\_latitude,                          'ti\_foto' => 'masjid.png'                      ];              }              $this->M\_tempat\_ibadah->create\_ti($v\_data);              $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Data berhasil ditambah!</div>');              redirect('c\_tempat\_ibadah/masjid');          }      } |

Listing 4.5 *Source code* pada *class view*

|  |
| --- |
| <div class="container-fluid">  <a href="<?= base\_url('c\_tempat\_ibadah/masjid'); ?>" class="btn btn-info mb-2"><i class="fa fa-arrow-left"></i> Lihat data masjid</a>  <div class="row">    <div class="col-sm-7">      <div class="card">              <div class="card-header">                <div class="row ">                  <h4 class="ml-2">Pilih lokasi:</h4>                  <button type="button" class="btn btn-outline-info btn-sm ml-auto mr-2" onclick="getLocation()">Get my location</button>              </div>                </div>                  <div id="mapid" style="height: 450px;"></div>      </div>    </div>    <div class="col-sm-5">      <div class="card">              <div class="card-header">                  <h4>Isi data:</h4>              </div>                <div class="card-body">                  <?= form\_open\_multipart('c\_tempat\_ibadah/tambahMasjid'); ?>                      <label for="basic-url">Tipologi</label>                    <div class="form-group ">                              <select class="custom-select " id="tipologi" name="scrolltip">                                <option value="">--Pilih topologi--</option>                                  <option value="Masjid Jami"> Masjid Jami</option>                                  <option value="Masjid Raya"> Masjid Raya</option>                                </select>                              <?= form\_error('scrolltip', '<small class="text-danger">', '</small>'); ?>                    </div>                    <label for="basic-url">Nama Masjid</label>                    <div class="form-group ">                      <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="nama\_masjid" value="<?= set\_value('nama\_masjid'); ?>">                      <?= form\_error('nama\_masjid', '<small class="text-danger">', '</small>'); ?>                    </div>                    <label for="basic-url">Alamat</label>                    <div class="form-group ">                      <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="alamat" value="<?= set\_value('alamat'); ?>">                      <?= form\_error('alamat', '<small class="text-danger">', '</small>'); ?>                    </div>                    <div class="btn-group">                        <div class="form-group mr-1">                          <label for="basic-url">Kabupaten</label>                          <select class="custom-select " id="kabupaten" name="scrollkab" onchange="get\_peta\_by\_kab()">                            <option value="">--Pilih kabupaten--</option>                            <?php foreach($list\_kabupaten as $kabupaten ) : ?>                              <option value="<?= $kabupaten['kab\_id']; ?>"><?= $kabupaten['kab\_nama']; ?></option>                            <?php endforeach; ?>                          </select>                          <?= form\_error('scrollkab', '<small class="text-danger">', '</small>'); ?>                      </div>                      <div class="form-group ">                        <label for="basic-url">Kecamatan</label>                          <select class="custom-select " id="kecamatan" name="scrollkec">                            <option value="" >--Pilih kecamatan--</option>                          </select>                          <?= form\_error('scrollkec', '<small class="text-danger">', '</small>'); ?>                      </div>                    </div>                    <div class="btn-group">                      <div class="form-group mr-1">                          <label for="basic-url">Latitude</label>                          <input type="text" class="form-control" id="Latitude" aria-describedby="basic-addon3" name="latitude" readonly="">                      </div>                      <div class="form-group ">                         <label for="basic-url">Longitude</label>                         <input type="text" class="form-control" id="Longitude" aria-describedby="basic-addon3" name="longitude" readonly="">                      </div>                    </div>                  </div>      </div>    </div>  </div>  <div class="card">      <div class="card-body">                    <label for="basic-url">Luas tanah</label>                    <div class="form-group ">                      <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="luas\_tanah" value="<?= set\_value('luas\_tanah'); ?>">                    </div>                    <label for="basic-url">Status tanah</label>                    <div class="form-group ">                      <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="status\_tanah" value="<?= set\_value('status\_tanah'); ?>">                    </div>                    <label for="basic-url">Luas bangunan</label>                    <div class="form-group ">                      <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="luas\_bangunan" value="<?= set\_value('luas\_bangunan'); ?>">                    </div>                    <label for="basic-url">Tahun berdiri</label>                    <div class="form-group ">                      <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="tahun\_berdiri" value="<?= set\_value('tahun\_berdiri'); ?>">                    </div>                    <label for="basic-url">Jamaah</label>                    <div class="form-group ">                      <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="jamaah" value="<?= set\_value('luas\_bangunan'); ?>">                    </div>                    <label for="basic-url">Imam</label>                    <div class="form-group ">                      <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="imam" value="<?= set\_value('imam'); ?>">                    </div>                    <label for="basic-url">Khatib</label>                    <div class="form-group ">                      <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="khatib" value="<?= set\_value('khatib'); ?>">                    </div>                    <label for="basic-url">Remaja</label>                    <div class="form-group ">                      <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="remaja" value="<?= set\_value('remaja'); ?>">                    </div>                    <label for="basic-url">No. Telepon</label>                    <div class="form-group ">                      <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="telepon" value="<?= set\_value('telepon'); ?>">                    </div>                    <label for="basic-url">Foto</label>                    <div class="custom-file">                      <input type="file" accept="image/\*" class="custom-file-input" name="foto">                      <label class="custom-file-label" for="customFile">Pilih foto</label>                    </div>                    <hr>                    <div class="row float-right mr-1">                        <button type="submit" class="btn btn-primary">Simpan</button>                    </div>                  <?= form\_close(); ?>                </div>  </div>          </div>        </div> |

1. Tampilan dan pengujian system



**Gambar 4.10** tampilan halaman tambah data tempat ibadah

Gambar 4.27 merupakan implementasi halaman tambah data tempat ibadah yang akan ditampilkan dalam sistem basis *web*. Berikut merupakan table pengujian system terhadap kasus kode *user stories* US-07 dapat dilihat pada table 4.23.

Tabel 4.23 pengujian penambahan fungsi fitur system

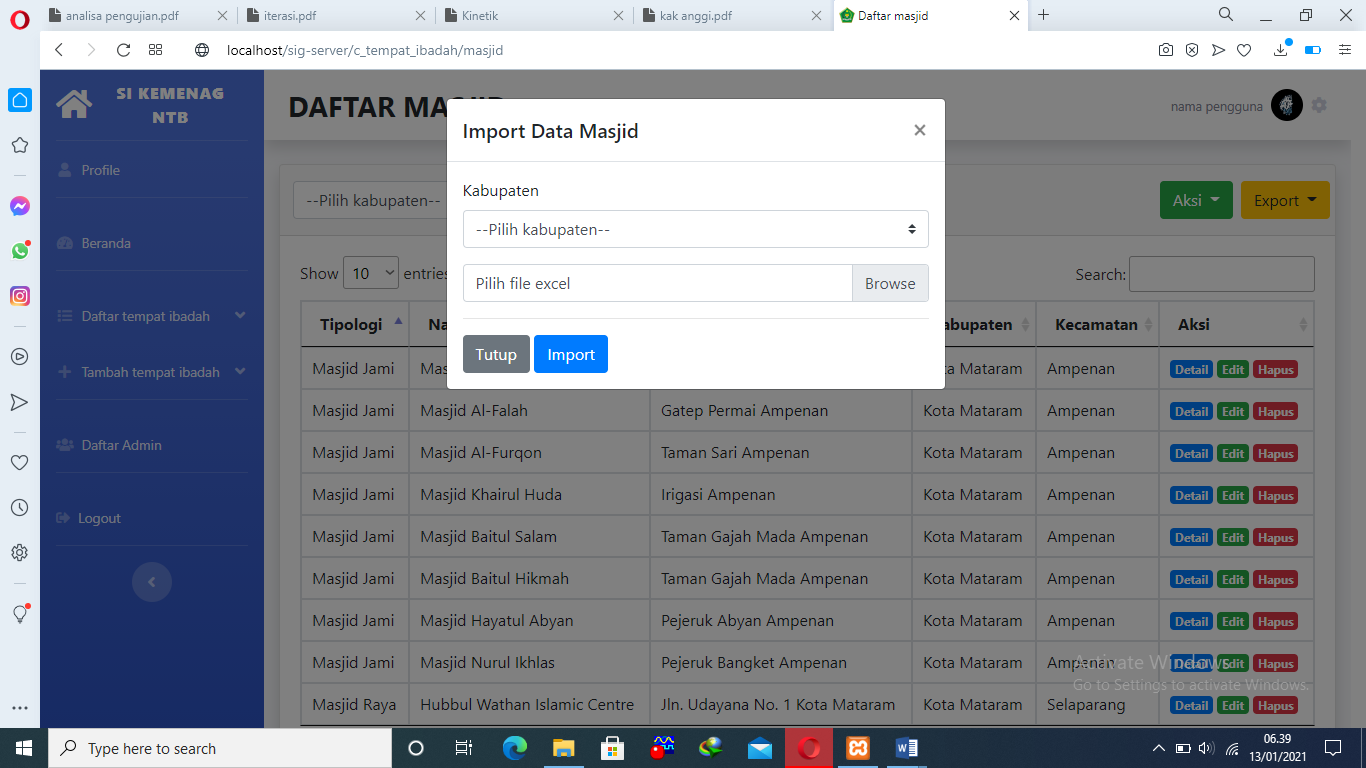
|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Nama atau alamat tidak diisi, serta kabupaten atau kecamatan tidak dipilih | Sistem akan melakukan proses dan data tidak berhasil ditambahkan | Sesuai | Valid |
| Nama dan alamat diisi, serta kabupaten dan kecamatan dipilih | Sistem akan melakukan proses dan data berhasil ditambahkan | Sesuai | Valid |

Sistem hanya akan berhasil menambahkan data jika *form* nama dan alamat tempat ibadah terisi serta kabuputen dan kecamatan terpilih, selain itu sistem akan menolak akses untuk menambahkan data tempat ibadah.

Selama proses pengembangan system kode *user stories* US-07, terdapat beberapa permintaan penambahan fungsi fitur systemdari *client*. Fungsi fitur tersebut bertujuan untuk mempermudah admin dalam menginputkan banyak data. Berikut merupakan analisa terhadap kebutuhan sistem untuk penambahan fungsi fitur yang diinginkan *client* dapat dilihat pada tabel 4.24 berikut.

Tabel 4. 24 penambahan fungsi fitur

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Halaman *import* data tempat ibadah | Terdapat tombol pilih kabupaten yang akan di *upload* datanyaTerdapat tombol “pilih *file* excel” data tempat ibadahTerdapat tombol tutup dan “*import*” untuk memasukkan data |
| Tombol *export* data tempat ibadah | Untuk mengunduh semua data tempat ibadah berupa file excel. |
| Tombol *download* *template* *import* data tempat ibadah | Untuk mengunduh *template* *import* data tempat ibadah berupa file excel. |



**Gambar 4.10** tampilan halaman *import* data tempat ibadah

Gambar 4.27 merupakan implementasi halaman *import* data. Terdapat tombol import data untuk mengimport data excel. Tabel 4.25 merupakan pengujian system terhadap penambahan fungsi fitur system.

Tabel 4.25 pengujian fungsi fitur system yang ditambahkan

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol *“import data”* | Menampilkan *form* untuk memasukkan file excel data tempat ibadah | Sesuai | Valid |
| Menekan tombol “pilih file excel” | Menampilkan jendela baru untuk memasukkan file *excel* | Sesuai | Valid |
| Menekan tombol “*import*” | Memproses dan memasukkan data ke dalam *database* | Sesuai | Valid |
| Menekan tombol “*unduh format excel*” | Format excel data tempat ibadah terunduh | Sesuai | Valid |
| Menekan tombol “*export*” | Data tempat ibadah terunduh dalam bentuk excel | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 24 dan pengujian terhadap penambahan fitur yang diinginkan pada table 4.25 dapat dikatakan system yang dibuat berhasil dan sesuai dengan kebutuhan *client.*

1. **Iteration 3**

Pada *iteration* 3 dilakukan pembuatan system pada kode *user stories*  US-8 yaitu Mengubah data tempat ibadah, US-09 yaitu Menghapus data tempat ibadah, US-10 yaitu Melihat Peta, dan US-11 yaitu Melihat Rute Lokasi Tempat Ibadah

1. **Kode *user stories* US-08**

Pada kasus kode *user stories* US-08 ini admin dapat mengubah data tempat ibadah. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-08 digambarkan pada tabel 4.26 berikut.

Tabel 4.26 kebutuhan system kode *user stories* US-08

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Halaman muka | Menu tab : profile, beranda, daftar tempat ibadah, tambah data tempat ibadah, daftar admin, logout.Menampilkan poto dan nama profile admin yang telah *login*Menampilkan *form* ubah data tempat ibadahMenampilkan peta dan *marker* untuk menentukan titik lokasi tempat ibadahTerdapat tombol untuk mengambil lokasi terkiniTerdapat tombol untuk mengubah data tempat ibadahTerdapat tombol kembali ke halaman daftar data tempat ibadah |

1. Desain system

Dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-08 dapat dilihat pada Tabel 4.27.

Tabel 4.27 kebutuhan system kode *user stories* US-08

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-08 | Tempat ibadah | Gambar 3.25 |

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-08 yang telah dibuat dapat dilihat pada listing code berikut:

*Listing* 4.3 *Source code* pada *class model*

|  |
| --- |
| <?php  class M\_tempat\_ibadah extends CI\_model {  public function edit\_ti($id, $data){          $this->db->update('tb\_ti', $data, array('ti\_id' => $id));      } |

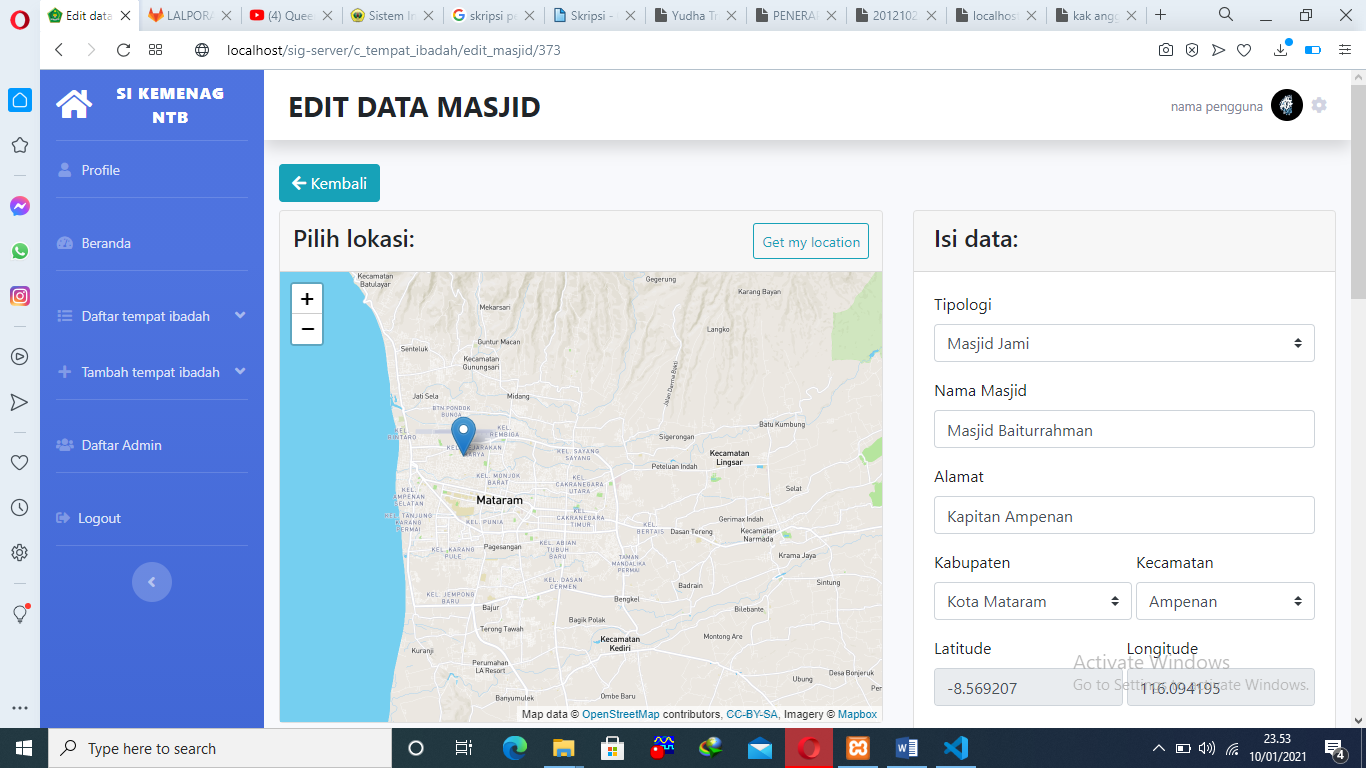
*Listing* 4.4 *Source code* pada *class controller*

|  |
| --- |
| public function edit\_masjid($v\_id)      {          $v\_data['judul'] = 'EDIT DATA MASJID';          $v\_role = $this->session->userdata('role');          if ($v\_role == 1) {              $v\_data['hilangkan'] = '';          }else{              $v\_data['hilangkan'] = 'd-none';          }          $v\_id\_username = $this->session->userdata('id\_username');          $v\_data['data\_pengguna'] = $this->M\_admin->get\_pengguna($v\_id\_username);          $v\_data['tempat\_ibadah'] = $this->M\_tempat\_ibadah->selectById($v\_id);          $v\_data['list\_tipologi'] = ['Masjid Jami', 'Masjid Raya'];          $v\_data['list\_kabupaten'] = $this->M\_kab\_kec->selectAllkabupaten();          $v\_data['list\_kecamatan'] = $this->M\_tempat\_ibadah->selectKecamatan($v\_data['tempat\_ibadah']['ti\_kabupaten']);            $v\_data['tittle'] = 'Edit data tempat ibadah';          $this->form\_validation->set\_rules('scrollkec','Scrollkec','required|callback\_validasi\_kecamatan');          $this->form\_validation->set\_rules('nama\_masjid', 'Nama\_masjid', 'required|trim', [              'required' => 'Form tidak boleh kosong!',          ]);          $this->form\_validation->set\_rules('alamat', 'Alamat', 'required|trim', [              'required' => 'Form tidak boleh kosong!',          ]);          if($this->form\_validation->run() == false){              $this->load->view('templates/header', $v\_data);              $this->load->view('templates/sidebar', $v\_data);              $this->load->view('templates/topbar', $v\_data);              $this->load->view('v\_daftar\_tempat\_ibadah/masjid/edit\_masjid', $v\_data);              $this->load->view('templates/footer');          }else{              $v\_id = $this->input->post('id\_tempat\_ibadah');              $v\_kabupaten = $this->input->post('scrollkab');              $v\_kecamatan = $this->input->post('scrollkec');              $v\_tipologi = $this->input->post('scrolltip');              $v\_nama     = $this->input->post('nama\_masjid');              $v\_alamat = $this->input->post('alamat');              $v\_luas\_tanah = $this->input->post('luas\_tanah');              $v\_status\_tanah = $this->input->post('status\_tanah');              $v\_luas\_bangunan = $this->input->post('luas\_bangunan');              $v\_tahun\_berdiri = $this->input->post('tahun\_berdiri');              $v\_jamaah = $this->input->post('jamaah');              $v\_imam = $this->input->post('imam');              $v\_khatib = $this->input->post('khatib');              $v\_remaja = $this->input->post('remaja');              $v\_telepon = $this->input->post('telepon');              $v\_longitude = $this->input->post('longitude');              $v\_latitude = $this->input->post('latitude');              $upload\_foto = $\_FILES['foto']['name'];              if($upload\_foto){                    $config['allowed\_types'] = 'gif|jpg|png|jpeg';                  $config['max\_size']     = '5000';                  $config['upload\_path'] = './assets/foto/tempat\_ibadah/';                    $this->load->library('upload', $config);                  if ($this->upload->do\_upload('foto')){                      $v\_nama\_foto = $this->upload->data('file\_name');                        $v\_foto\_lama = $v\_data['tempat\_ibadah']['ti\_foto'];                      if($v\_foto\_lama != 'masjid.png'){                              unlink(FCPATH . 'assets/foto/tempat\_ibadah/' . $v\_foto\_lama);                      }                      $v\_data = [                          'ti\_tipologi' => $v\_tipologi,                          'ti\_nama' => $v\_nama,                          'ti\_alamat' => $v\_alamat,                          'ti\_kabupaten' => $v\_kabupaten,                          'ti\_kecamatan' => $v\_kecamatan,                          'ti\_luas\_tanah' => $v\_luas\_tanah,                          'ti\_status\_tanah' => $v\_status\_tanah,                          'ti\_luas\_bangunan' => $v\_luas\_bangunan,                          'ti\_tahun\_berdiri' => $v\_tahun\_berdiri,                          'ti\_jamaah' => $v\_jamaah,                          'ti\_imam' => $v\_imam,                          'ti\_khatib' => $v\_khatib,                          'ti\_remaja' => $v\_remaja,                          'ti\_telepon' => $v\_telepon,                          'longitude' => $v\_longitude,                          'latitude' => $v\_latitude,                          'ti\_foto' => $v\_nama\_foto                      ];                  }                  else                  {                      echo $this->upload->display\_errors();                  }              }else{                  $v\_data = [                          'ti\_tipologi' => $v\_tipologi,                          'ti\_nama' => $v\_nama,                          'ti\_alamat' => $v\_alamat,                          'ti\_kabupaten' => $v\_kabupaten,                          'ti\_kecamatan' => $v\_kecamatan,                          'ti\_luas\_tanah' => $v\_luas\_tanah,                          'ti\_status\_tanah' => $v\_status\_tanah,                          'ti\_luas\_bangunan' => $v\_luas\_bangunan,                          'ti\_tahun\_berdiri' => $v\_tahun\_berdiri,                          'ti\_jamaah' => $v\_jamaah,                          'ti\_imam' => $v\_imam,                          'ti\_khatib' => $v\_khatib,                          'ti\_remaja' => $v\_remaja,                          'ti\_telepon' => $v\_telepon,                          'longitude' => $v\_longitude,                          'latitude' => $v\_latitude                      ];              }              $this->M\_tempat\_ibadah->edit\_ti($v\_id, $v\_data);              $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Data berhasil diubah!</div>');              redirect('c\_tempat\_ibadah/masjid');          }      } |

Listing 4.5 *Source code* pada *class view*

|  |
| --- |
| <div class="container-fluid">            <a href="<?= base\_url('c\_tempat\_ibadah/masjid'); ?>" class="btn btn-info mb-2"><i class="fa fa-arrow-left"></i> Kembali</a>  <div class="row">    <div class="col-sm-7">      <div class="card">              <div class="card-header">                <div class="row ">                  <h4 class="ml-2">Pilih lokasi:</h4>                  <button type="button" class="btn btn-outline-info btn-sm ml-auto mr-2" onclick="getLocation()">Get my location</button>                </div>              </div>                  <div id="mapid" style="height: 450px;"></div>      </div>    </div>    <div class="col-sm-5">      <div class="card">              <div class="card-header">                    <h4>Isi data:</h4>                </div>              <div class="card-body">                <?= form\_open\_multipart(); ?>              <input type="hidden" name="id\_tempat\_ibadah" value=" <?= $tempat\_ibadah['ti\_id']; ?>">                  <label for="basic-url">Tipologi</label>                <div class="form-group">                          <select class="custom-select " id="tipologi" name="scrolltip">                            <?php foreach($list\_tipologi as $t ) : ?>                              <?php if($t == $tempat\_ibadah['ti\_tipologi']) : ?>                                <option value="<?= $t; ?>" selected><?= $t; ?></option>                              <?php else : ?>                                <option value="<?= $t; ?>"><?= $t; ?></option>                              <?php endif; ?>                            <?php endforeach; ?>                          </select>                </div>                <label for="basic-url">Nama Masjid</label>                <div class="form-group">                  <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="nama\_masjid" value="<?= $tempat\_ibadah['ti\_nama']; ?>">                  <?= form\_error('nama\_masjid', '<small class="text-danger">', '</small>'); ?>                </div>                <label for="basic-url">Alamat</label>                <div class="form-group">                  <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="alamat" value="<?= $tempat\_ibadah['ti\_alamat']; ?>">                  <?= form\_error('alamat', '<small class="text-danger">', '</small>'); ?>                </div>                <div class="btn-group">                  <div class="form-group mr-1">                    <label for="basic-url">Kabupaten</label>                    <select class="custom-select " id="kabupaten" name="scrollkab" onchange="get\_peta\_by\_kab()">                     <?php foreach($list\_kabupaten as $kab) : ?>                        <?php if($kab['kab\_id'] == $tempat\_ibadah['ti\_kabupaten']) : ?>                          <option value="<?= $kab['kab\_id']; ?>" selected><?= $kab['kab\_nama']; ?></option>                        <?php else : ?>                          <option value="<?= $kab['kab\_id']; ?>"><?= $kab['kab\_nama']; ?></option>                        <?php endif; ?>                      <?php endforeach; ?>                    </select>                  </div>                  <div class="form-group">                  <label for="basic-url">Kecamatan</label>                    <select class="custom-select " id="kecamatan" name="scrollkec">                      <?php foreach($list\_kecamatan as $kec) : ?>                        <?php if($kec['kec\_id'] == $tempat\_ibadah['ti\_kecamatan']) : ?>                          <option value="<?= $kec['kec\_id']; ?>" selected><?= $kec['kec\_nama']; ?></option>                        <?php else : ?>                          <option value="<?= $kec['kec\_id']; ?>"><?= $kec['kec\_nama']; ?></option>                        <?php endif; ?>                      <?php endforeach; ?>                    </select>                  </div>                </div>                  <div class="btn-group">                  <div class="form-group mr-1">                    <label for="basic-url">Latitude</label>                    <input type="text" class="form-control" id="Latitude" aria-describedby="basic-addon3" name="latitude" value="<?= $tempat\_ibadah['latitude']; ?>" readonly>                  </div>                  <div class="form-group">                    <label for="basic-url">Longitude</label>                    <input type="text" class="form-control" id="Longitude" aria-describedby="basic-addon3" name="longitude" value="<?= $tempat\_ibadah['longitude']; ?>" readonly>                  </div>                </div>              </div>      </div>    </div>  </div>  <div class="card">      <div class="card-body">                <label for="basic-url">Luas tanah</label>                <div class="form-group">                  <input type="tesxt" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="luas\_tanah" value="<?= $tempat\_ibadah['ti\_luas\_tanah']; ?>">                </div>                  <label for="basic-url">Status tanah</label>                <div class="form-group">                  <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="status\_tanah" value="<?= $tempat\_ibadah['ti\_status\_tanah']; ?>">                </div>                <label for="basic-url">Luas bangunan</label>                <div class="form-group">                  <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="luas\_bangunan" value="<?= $tempat\_ibadah['ti\_luas\_bangunan']; ?>">                </div>                <label for="basic-url">Tahun berdiri</label>                <div class="form-group">                  <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="tahun\_berdiri" value="<?= $tempat\_ibadah['ti\_tahun\_berdiri']; ?>">                </div>                <label for="basic-url">Jamaah</label>                <div class="form-group">                  <input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="jamaah" value="<?= $tempat\_ibadah['ti\_jamaah']; ?>">                </div>                <label for="basic-url">Imam</label>                <div class="form-group">                  <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="imam" value="<?= $tempat\_ibadah['ti\_imam']; ?>">                </div>                <label for="basic-url">Khatib</label>                <div class="form-group">                  <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="khatib" value="<?= $tempat\_ibadah['ti\_khatib']; ?>">                </div>                <label for="basic-url">Remaja</label>                <div class="form-group">                  <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="remaja" value="<?= $tempat\_ibadah['ti\_remaja']; ?>">                </div>                <label for="basic-url">No. Telepon</label>                <div class="form-group">                  <input type="number" class="form-control" id="basic-url" aria-describedby="basic-addon3" name="telepon" value="<?= $tempat\_ibadah['ti\_telepon']; ?>">                </div>                <label for="basic-url">Foto</label>                <div class="mb-2">                <img src="<?= base\_url('assets/foto/tempat\_ibadah/').$tempat\_ibadah['ti\_foto']; ?>" width="100" class="card-img" style="width: 10rem;">                </div>                <div class="custom-file">                  <input type="file" accept="image/\*" class="custom-file-input" name="foto">                  <label class="custom-file-label" for="customFile">Pilih foto</label>                </div>                <hr>                <div class="row float-right mr-1">                    <button type="submit" class="btn btn-primary">Simpan</button>                </div>              <?= form\_close(); ?>            </div>      </div>    </div>  </div>          </div>        </div> |

1. Tampilan dan pengujian system



**Gambar 4.10** tampilan halaman edit data tempat ibadah

Gambar 4.27 merupakan implementasi halaman edit data tempat ibadah yang akan ditampilkan dalam sistem basis *web*. Berikut merupakan table pengujian system terhadap kasus kode *user stories* US-07 dapat dilihat pada table 4.28.

Tabel 4.28 pengujian ubah data tempat ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Nama atau alamat tidak diisi, serta kabupaten atau kecamatan tidak dipilih | Sistem akan melakukan proses dan data tidak berhasil ditambahkan | Sesuai | Valid |
| Nama dan alamat diisi, serta kabupaten dan kecamatan dipilih | Sistem akan melakukan proses dan data berhasil ditambahkan | Sesuai | Valid |

Sama seperti halnya pengujian menambahkan data, sistem hanya akan berhasil mengubah data jika *form* nama dan alamat tempat ibadah terisi serta kabuputen dan kecamatan terpilih, selain itu sistem akan menolak akses untuk mengubah data tempat ibadah. Berdasarkan hasil pengujian dan kesimpulan yang didapat pada table 4.28, dapat dikatakan bahwa sistem yang dibuat benar dan sesuai dengan keinginan *client.*

1. **Kode *user stories* US-09**

Pada kasus kode *user stories* US-09 ini admin dapat menghapus data tempat ibadah. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-09 digambarkan pada tabel 4.29 berikut.

Tabel 4.29 kebutuhan system kode *user stories* US-09

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Tampilan sebelum menghapus data | Terdapat tombol batal dan tombol hapus untuk menghapus data tempat ibadah |

1. Desain system

Dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-09 dapat dilihat pada Tabel 4.30.

Tabel 4.30 kebutuhan system kode *user stories* US-09

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-09 | Tempat ibadah | Gambar 3.26 |

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-09 yang telah dibuat dapat dilihat pada listing code berikut:

*Listing* 4.3 *Source code* pada *class model*

|  |
| --- |
| <?php  class M\_tempat\_ibadah extends CI\_model {  public function hapus\_ti($v\_id) {          $this->db->where('ti\_id', $v\_id);          $this->db->delete('tb\_ti');      } |

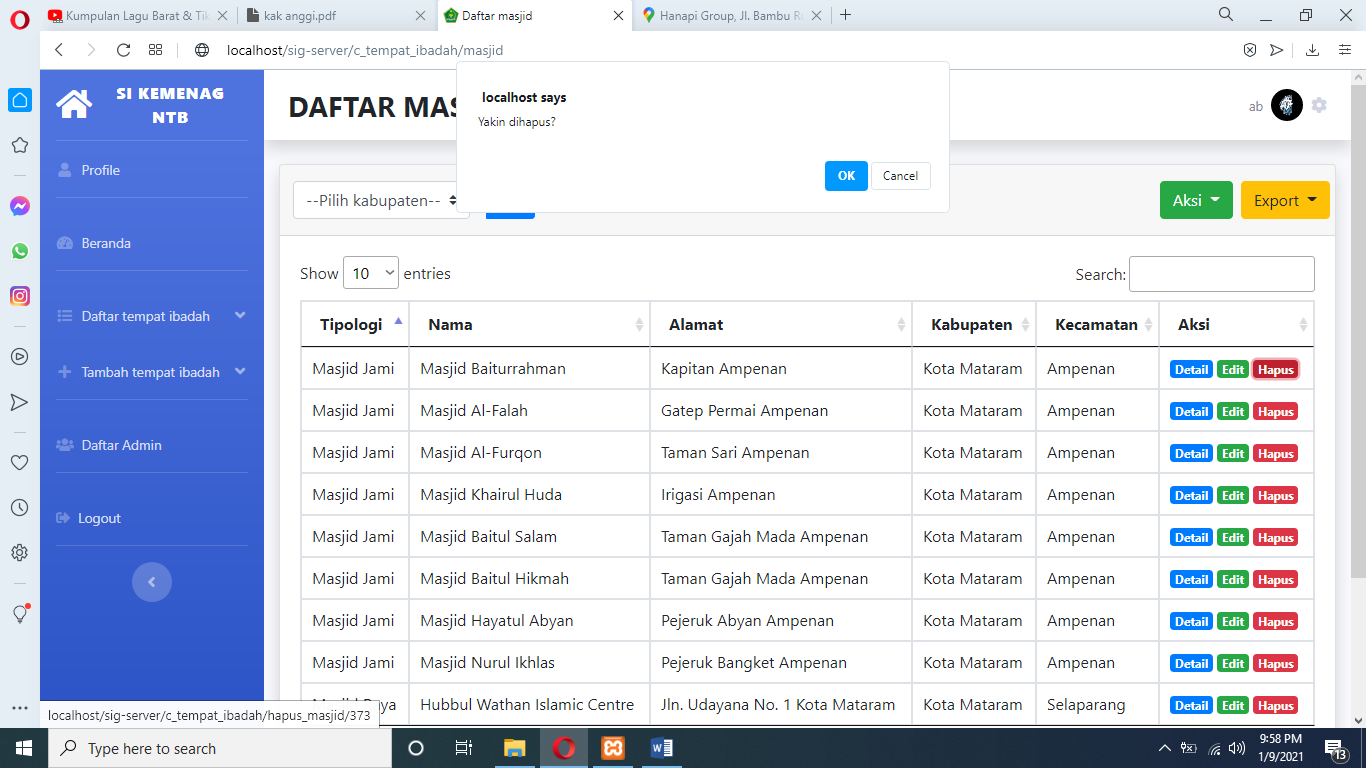
*Listing* 4.4 *Source code* pada *class controller*

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class C\_tempat\_ibadah extends CI\_Controller {      public $var;      public function \_\_construct(){          parent::\_\_construct();          cek\_login();      }      public function hapus\_masjid($v\_id){          $hapus\_foto = $this->M\_tempat\_ibadah->selectById($v\_id);            if($hapus\_foto['ti\_foto'] != 'masjid.png'){              unlink(FCPATH . 'assets/foto/tempat\_ibadah/' . $hapus\_foto['ti\_foto']);            }          $this->M\_tempat\_ibadah->hapus\_ti($v\_id);          unlink(FCPATH . 'assets/foto/tempat\_ibadah/' . $v\_foto\_lama);          $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Data berhasil dihapus!</div>');          redirect('c\_tempat\_ibadah/masjid');      } |

Listing 4.5 *Source code* pada *class view*

|  |
| --- |
| <a href='<?= base\_url('c\_tempat\_ibadah/hapus\_masjid/')?>${row.ti\_id}' class="badge badge-danger" onclick="return confirm('Yakin dihapus?');">Hapus</a> |

1. Tampilan dan pengujian system



**Gambar 4.10** tampilan hapus data tempat ibadah

Pengujian fungsi pada kasus kode *user stories* US-09 ini dilakukan untuk menguji apakah fungsi untuk mengahapus data tempat ibadah sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.31

Tabel 4.31 pengujian mengahpus data tempat ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol “*Cancel*” | Data tidak terhapus | Sesuai | Valid |
| Menekan tombol “*Ok*” | Data terhapus dalam sistem | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.31 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. **Kode *user stories* US-10**

Pada kasus kode *user stories* US-10 ini pengguna dapat melihat lokasi tempat ibadah dalam peta. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-10 digambarkan pada tabel 4.32 berikut.

Tabel 4.32 kebutuhan system kode *user stories* US-10

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Tampilan peta basis *android* | Menu tab : list, map, dan about usMenampilkan peta lokasi tempat ibadahTerdapat tombol detail data tempat ibadahTerdapat tombol rute dan detail data tempat ibadahTerdapat tombol lokasi terkini penggunaFungsi filter berdasarkan jenis tempat ibadahFungsi pencarian lokasi tempat ibadah |

1. Desain system

Dilakukan desain system menggunakan CRC *Cards* seperti yang telah dipaparkan pada tabel 3.3 dan *spike solution prototype* untuk skema desain *prototype*-nya. *Design* dari kasus kode *user stories* US-10 dapat dilihat pada Tabel 4.33.

Tabel 4.33 kebutuhan system kode *user stories* US-10

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-10 | Tempat ibadah | Gambar 3.29 |

1. *Code* system

Berikut merupakan *code* sistem dari kasus kode *user stories* US-10 yang telah dibuat dapat dilihat pada listing code berikut:

1. class MapsFragment : Fragment(), OnMapReadyCallback, PermissionListener {
2. lateinit var mainViewModel : IbadahViewModel
3. private lateinit var destination : ArrayList<ModelIbadah>
4. var markerOptions = MarkerOptions()
5. var ceck = 0
6. var mMap : GoogleMap? = null
7. private lateinit var fusedLocationProviderClient: FusedLocationProviderClient
8. override fun onCreateView(
9. inflater: LayoutInflater,
10. container: ViewGroup?,
11. savedInstanceState: Bundle?
12. ): View? {
13. return inflater.inflate(R.layout.fragment\_maps, container, false)
14. }
16. override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
17. super.onViewCreated(view, savedInstanceState)
18. val mapFragment = childFragmentManager.findFragmentById(R.id.map) as SupportMapFragment?
19. mapFragment!!.getMapAsync(this)
20. fusedLocationProviderClient = FusedLocationProviderClient(requireActivity())
21. }
23. @SuppressLint("ResourceType")
24. fun loadMarker(){
25. clean.setOnClickListener {
26. search.setText("")
27. }
29. val kategori = resources.getStringArray(R.array.kategori)
31. if (listItem != null) {
32. val adapter = ArrayAdapter(
33. requireContext(),
34. android.R.layout.simple\_spinner\_dropdown\_item, kategori
35. )
36. listItem.adapter = adapter
38. listItem.onItemSelectedListener = object :
39. AdapterView.OnItemSelectedListener {
40. override fun onItemSelected(
41. parent: AdapterView<\*>,
42. view: View, position: Int, id: Long
43. ) {
44. val id = listItem.selectedItemPosition
46. if(id == 0){
47. cari("")
48. }else if(id ==1){
49. cari("1")
50. }
51. else if(id ==2){
52. cari("2")
53. }
54. else if(id ==3){
55. cari("3")
56. }
57. else if(id ==4){
58. cari("4")
59. }
60. else if(id ==5){
61. cari("5")
62. }
63. Toast.makeText(
64. requireContext(),
65. "${adapter.getItem(id)}", Toast.LENGTH\_SHORT
66. ).show()
67. }
69. override fun onNothingSelected(parent: AdapterView<\*>) {
70. // write code to perform some action
71. }
72. }
73. }
75. }

78. override fun onResume() {
79. super.onResume()
80. activity!!.requestedOrientation = ActivityInfo.SCREEN\_ORIENTATION\_PORTRAIT
81. loadMarker()
82. cari("")
84. }
86. fun cari(jenis\_id: String){
88. val klentengBitmap = BitmapFactory.decodeResource(resources, R.drawable.klenteng)
89. val klentengMarker = Bitmap.createScaledBitmap(klentengBitmap, 90, 90, false)
90. val masjidBitmap = BitmapFactory.decodeResource(resources, R.drawable.masjid)
91. val masjidMarker = Bitmap.createScaledBitmap(masjidBitmap, 90, 90, false)
92. val puraBitmap = BitmapFactory.decodeResource(resources, R.drawable.pura)
93. val puraMarker = Bitmap.createScaledBitmap(puraBitmap, 90, 90, false)
94. val viharaBitmap = BitmapFactory.decodeResource(resources, R.drawable.vihara)
95. val viharaMarker = Bitmap.createScaledBitmap(viharaBitmap, 90, 90, false)
96. val gerejaBitmap = BitmapFactory.decodeResource(resources, R.drawable.gereja)
97. val gerejaMarker = Bitmap.createScaledBitmap(gerejaBitmap, 90, 90, false)
99. mainViewModel = ViewModelProvider(this, ViewModelProvider.NewInstanceFactory()).get(
100. IbadahViewModel::class.java
101. )
102. mainViewModel.setJenis(jenis\_id)
104. mainViewModel.getDestination().observe(this, Observer { destination ->
105. mMap!!.clear()
106. if (destination != null) {
108. destination.forEach {
110. if (it.latitude == null || it.longitude == null) {
112. } else {
114. val lat = it.latitude!!.toDouble()
115. val lng = it.longitude!!.toDouble()
117. val latlng = LatLng(lat, lng)
119. val dataMarker = markerOptions.position(latlng)
120. .title(it.ti\_nama)
121. .snippet(it.ti\_alamat)
123. if (it.ti\_jenis == "1") {
124. dataMarker.icon(BitmapDescriptorFactory.fromBitmap(masjidMarker))
125. } else if (it.ti\_jenis == "2") {
126. dataMarker.icon(BitmapDescriptorFactory.fromBitmap(puraMarker))
127. } else if (it.ti\_jenis == "3") {
128. dataMarker.icon(BitmapDescriptorFactory.fromBitmap(gerejaMarker))
129. } else if (it.ti\_jenis == "4") {
130. dataMarker.icon(BitmapDescriptorFactory.fromBitmap(viharaMarker))
131. } else if (it.ti\_jenis == "5") {
132. dataMarker.icon(BitmapDescriptorFactory.fromBitmap(klentengMarker))
133. }

136. val modelIbadah = it
137. val m = mMap!!.addMarker(markerOptions)
138. m.tag = modelIbadah
140. Log.d("Tes", "${m.tag}")
141. mMap!!.setOnInfoWindowClickListener { p0 ->
142. val modelIbadah = p0.tag as ModelIbadah

145. val intent = Intent(context, DetailActivity::class.java)
146. intent.putExtra(DetailActivity.EXTRA\_DETAIl, modelIbadah)
147. startActivity(intent)
149. }
151. if (search.text.equals("")) {
152. clean.visibility = View.GONE
153. search.isCursorVisible = false
154. } else {
155. search.isCursorVisible = true
156. clean.visibility = View.VISIBLE
158. val var\_adapter = ArrayAdapter(
159. requireContext(),
160. android.R.layout.simple\_spinner\_dropdown\_item,
161. destination
162. )
164. search.threshold = 0
165. search.setAdapter(var\_adapter)
167. search.setOnItemClickListener { parent, view, position, id ->
169. var position = search.adapter.getItem(position) as ModelIbadah
171. if (position.latitude == null || position == null) {
172. Toast.makeText(
173. requireContext(),
174. "Maaf, garis bujur dan garis lintang dari loksai ini belum ditemukan",
175. Toast.LENGTH\_LONG
176. ).show()
177. } else {
178. var latitude = position.latitude!!.toDouble()
179. var longitude = position.longitude!!.toDouble()
180. val kamera\_posisi = CameraPosition.builder()
181. .target(LatLng(latitude, longitude))
182. .zoom(15f)
183. .build()
184. mMap!!.animateCamera(
185. CameraUpdateFactory.newCameraPosition(
186. kamera\_posisi
187. )
188. )
189. }
190. }
191. }
193. }
195. }
197. } else {
198. mMap!!.clear()
199. }
200. })
201. }
203. override fun onMapReady(googleMap: GoogleMap?) {
204. mMap = googleMap!!
206. // val cameraPosition = CameraPosition.builder()
207. // .target(LatLng(-8.577669, 116.100659))
208. // .zoom(14f)
209. // .build()
210. // googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition))
211. //
212. //
213. // googleMap.setPadding(0,300,0,0)
214. // googleMap.isMyLocationEnabled = true
215. // googleMap.isTrafficEnabled = true
216. // googleMap.uiSettings.isMyLocationButtonEnabled = true
217. // googleMap.uiSettings.isZoomControlsEnabled = true
219. if (isPermissionGiven()){
220. val cameraPosition = CameraPosition.Builder()
221. .target(LatLng(-8.577669, 116.100659))
222. .zoom(14f)
223. .build()
225. googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition))
227. imgMyLocation.setOnClickListener {
228. getCurrentLocation()
229. }
231. if (ActivityCompat.checkSelfPermission(
232. requireContext(),
233. Manifest.permission.ACCESS\_FINE\_LOCATION
234. ) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(
235. requireContext(),
236. Manifest.permission.ACCESS\_COARSE\_LOCATION
237. ) != PackageManager.PERMISSION\_GRANTED
238. ) {
239. // TODO: Consider calling
240. // ActivityCompat#requestPermissions
241. // here to request the missing permissions, and then overriding
242. // public void onRequestPermissionsResult(int requestCode, String[] permissions,
243. // int[] grantResults)
244. // to handle the case where the user grants the permission. See the documentation
245. // for ActivityCompat#requestPermissions for more details.
246. return
247. }
248. googleMap.isMyLocationEnabled = true
249. googleMap.isTrafficEnabled = true
250. googleMap.uiSettings.isZoomControlsEnabled = true
251. getCurrentLocation()
252. } else {
253. givePermission()
254. }
256. }
258. private fun isPermissionGiven(): Boolean{
259. return ActivityCompat.checkSelfPermission(requireContext(), android.Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED
260. }
262. private fun givePermission() {
263. Dexter.withActivity(activity)
264. .withPermission(android.Manifest.permission.ACCESS\_FINE\_LOCATION)
265. .withListener(this)
266. .check()
267. }
269. override fun onPermissionGranted(response: PermissionGrantedResponse?) {
270. getCurrentLocation()
271. }
273. override fun onPermissionRationaleShouldBeShown(
274. permission: PermissionRequest?,
275. token: PermissionToken?
276. ) {
277. token!!.continuePermissionRequest()
278. }
280. override fun onPermissionDenied(response: PermissionDeniedResponse?) {
281. Toast.makeText(requireContext(), "Permission required for showing location", Toast.LENGTH\_LONG).show()
282. }
284. private fun getCurrentLocation() {
286. val locationRequest = LocationRequest()
287. locationRequest.priority = LocationRequest.PRIORITY\_HIGH\_ACCURACY
288. locationRequest.interval = (10 \* 1000).toLong()
289. locationRequest.fastestInterval = 2000
291. val builder = LocationSettingsRequest.Builder()
292. builder.addLocationRequest(locationRequest)
293. val locationSettingsRequest = builder.build()
295. val result = LocationServices.getSettingsClient(requireActivity()).checkLocationSettings(locationSettingsRequest)
296. result.addOnCompleteListener { task ->
297. try {
298. val response = task.getResult(ApiException::class.java)
299. if (response!!.locationSettingsStates.isLocationPresent){
300. getLastLocation()
301. }
302. } catch (exception: ApiException) {
303. when (exception.statusCode) {
304. LocationSettingsStatusCodes.RESOLUTION\_REQUIRED -> try {
305. val resolvable = exception as ResolvableApiException
306. resolvable.startResolutionForResult(activity,
307. MainActivity.REQUEST\_CHECK\_SETTINGS
308. )
309. } catch (e: IntentSender.SendIntentException) {
310. } catch (e: ClassCastException) {
311. }
313. LocationSettingsStatusCodes.SETTINGS\_CHANGE\_UNAVAILABLE -> { }
314. }
315. }
316. }
317. }
319. private fun getLastLocation() {
320. if (ActivityCompat.checkSelfPermission(
321. requireContext(),
322. Manifest.permission.ACCESS\_FINE\_LOCATION
323. ) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(
324. requireContext(),
325. Manifest.permission.ACCESS\_COARSE\_LOCATION
326. ) != PackageManager.PERMISSION\_GRANTED
327. ) {
328. // TODO: Consider calling
329. // ActivityCompat#requestPermissions
330. // here to request the missing permissions, and then overriding
331. // public void onRequestPermissionsResult(int requestCode, String[] permissions,
332. // int[] grantResults)
333. // to handle the case where the user grants the permission. See the documentation
334. // for ActivityCompat#requestPermissions for more details.
335. return
336. }
337. fusedLocationProviderClient.lastLocation
338. .addOnCompleteListener(requireActivity()) { task ->
339. if (task.isSuccessful && task.result != null) {
340. val mLastLocation = task.result
342. var address = "Alamat tidak ditemukan"
344. val gcd = Geocoder(context, Locale.getDefault())
345. val addresses: List<Address>
346. try {
347. addresses = gcd.getFromLocation(mLastLocation!!.latitude, mLastLocation.longitude, 1)
348. if (addresses.isNotEmpty()) {
349. address = addresses[0].getAddressLine(0)
350. }
351. } catch (e: IOException) {
352. e.printStackTrace()
353. }

356. val cameraPosition = CameraPosition.Builder()
357. .target(LatLng(mLastLocation!!.latitude, mLastLocation!!.longitude))
358. .zoom(14f)
359. .build()
360. mMap!!.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition))
361. } else {
362. Toast.makeText(requireContext(), "Lokasi terkini tidak ditemukan", Toast.LENGTH\_LONG).show()
363. }
364. }
365. }
367. override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
369. when (requestCode) {
370. MainActivity.REQUEST\_CHECK\_SETTINGS -> {
371. if (resultCode == Activity.RESULT\_OK) {
372. getCurrentLocation()
373. }
374. }
375. }
376. super.onActivityResult(requestCode, resultCode, data)
378. }
379. }

*Source code* 4.3 pada *class MapsFragment*

1. Tampilan dan pengujian system

**Gambar 4.10** tampilan peta lokasi tempat ibadah basis *android*

Gambar 4.27 merupakan implementasi halaman peta lokasi tempat ibadah yang akan ditampilkan dalam sistem basis *android*. Berikut merupakan table pengujian system terhadap kasus kode *user stories* US-10 dapat dilihat pada table 4.34.

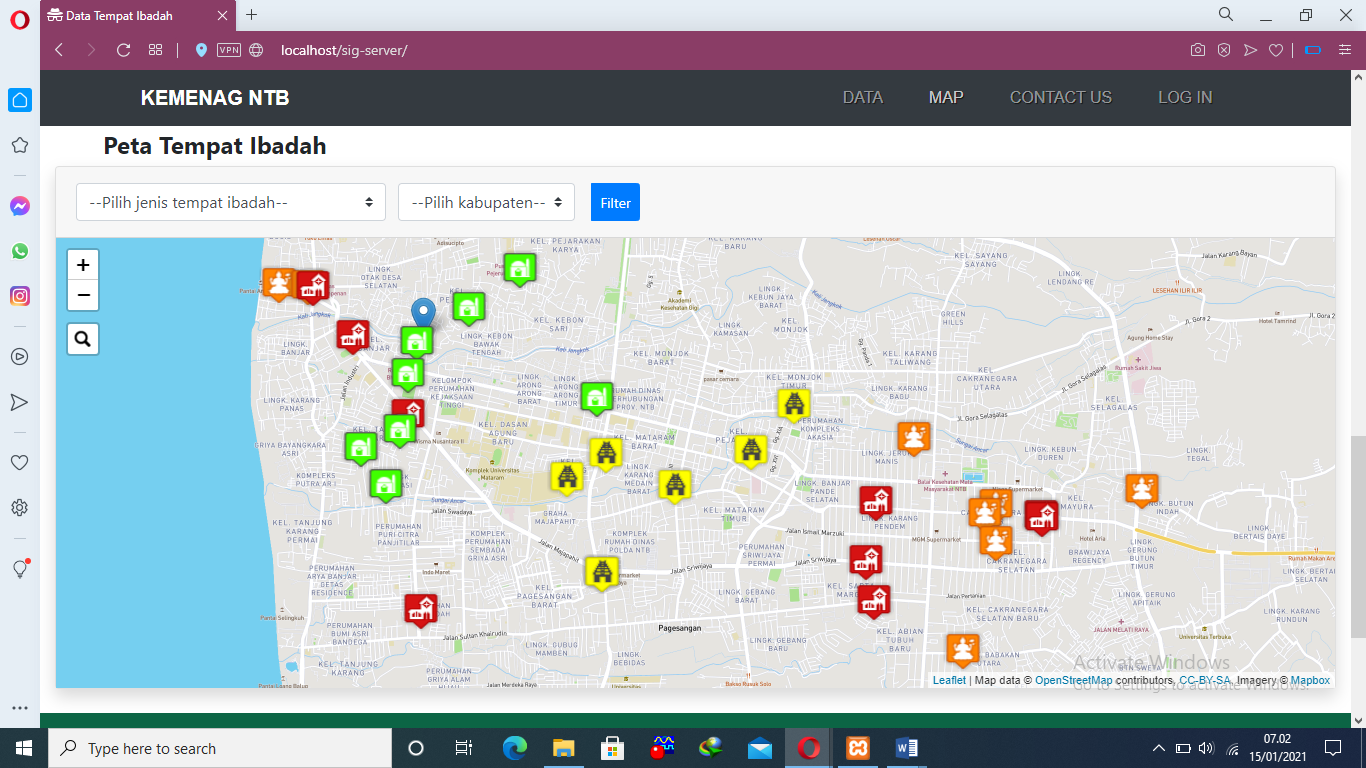
Tabel 4.34 pengujian peta lokasi tempat ibadah basis *android*

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol halaman peta lokasi tempat ibadah | Peta lokasi tempat ibadah berhasil ditampilkan | Sesuai | Valid |
| Memasukkan nama tempat ibadah atau *keyword* ke dalam fitur pencarian | Menampilkan hasil pencarian *keyword* yang dimasukkan | Sesuai | Valid |
| Memilih jenis tempat ibadah yang diinginkan berdasarkan filter jenis tempat ibadah | Peta lokasi tempat ibadah berhasil ditampilkan berdasarkan jenis tempat ibadah yang dipilih | Sesuai | Valid |
| Menekan tombol lokasi terkini pengguna | Lokasi terkini pengguna berhasil ditampilkan | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.34 dapat dikatakan system berhasil dan sesuai*,* namunselama proses pengembangan system kode *user stories* US-10, terdapat beberapa permintaan penambahan fungsi fitur system. Fungsi fitur tersebut yaitu terdapat peta lokasi tempat ibadah pada basis *web* yang menampilkan peta lokasi ibadah berdasarkan jenis dan daerah kabupaten yang ada di provinsi NTB. Penambahan fungsi fitur ini bertujuan untuk memberitahukan pihak kemenag NTB maupun masyarakat sebagai pengguna sistem agar mengetahui persebaran lokasi tempat ibadah yang ada di provinsi NTB. Berikut merupakan analisa terhadap kebutuhan sistem untuk penambahan fungsi fitur yang diinginkan *client* dapat dilihat pada tabel 4.35 berikut.

Tabel 4.35 penambahan fungsi fitur peta lokasi tempat ibadahbasis *web*

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Tampilan halaman muka peta basis *web* | Menu tab : data, map, contact us, loginMenampilkan peta lokasi tempat ibadah dan lokasi terkini penggunaTerdapat tombol rute dan detail data tempat ibadahFungsi filter berdasarkan kabupaten dan jenis tempat ibadahFungsi pencarian lokasi tempat ibadah |



**Gambar 4.10** tampilan halaman peta lokasi tempat ibadah basis *web*

Gambar 4.27 merupakan implementasi halaman peta lokasi tempat ibadah basis *web*. Berikut merupakan tabel pengujian system terhadap penambahan fitur peta lokasi tempat ibadah basis *web* dapat dilihat pada table 4.36.

Tabel 4.36 pengujian peta lokasi tempat ibadah basis *web*

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol halaman lokasi tempat ibadah | Menampilkan data lokasi tempat ibadah dalam bentuk map | Sesuai | Valid |
| Menekan salah satu lokasi tempat ibadah | Menampilkan deskripsi singkat dan terdapat tombol detail dan rute tempat ibadah | Sesuai | Valid |
| Menekan tombol detail tempat ibadah | Data detail tempat ibadah berhasil ditampilkan | Sesuai | Valid |
| Memilih kabupaten dan jenis tempat ibadah dan kemudian menekan tombol filter | Data tempat ibadah yang ditampilkan berdasarkan filter kabupaten dan jenis tempat ibadah | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.35 dan pengujian terhadap penambahan fitur yang diinginkan pada table 4.36 dapat dikatakan system yang dibuat berhasil dan sesuai dengan kebutuhan *client.*

1. **Kode *user stories* US-11**

Pada kasus kode *user stories* US-11 ini pengguna dapat melihat rute lokasi tempat ibadah. Berikut penjelasan analisa, desain, code dan tampilan serta hasil pengujian systemnya.

1. Analisa system

Dilakukan analisa terhadap kebutuhan sistem untuk kasus kode *user stories* US-11 digambarkan pada tabel 4.37 berikut.

Tabel 4.37 kebutuhan system kode *user stories* US-10

|  |  |
| --- | --- |
| *Task* | Fungsi – fungsi yang disediakan |
| Tombol rute tempat ibadah | Menampilkan hasil rute menuju lokasi tempat ibadah |

1. Desain system

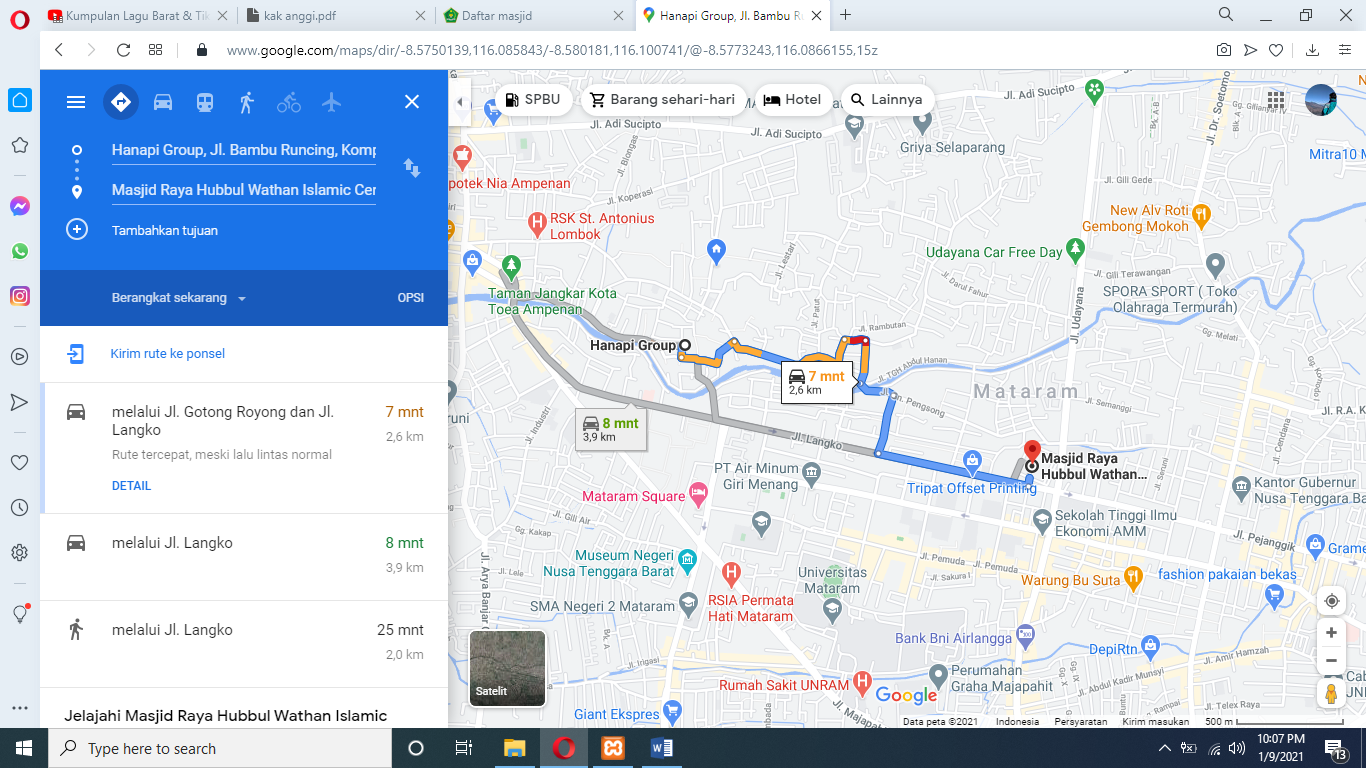
Tabel 4.33 kebutuhan system kode *user stories* US-10

|  |  |  |
| --- | --- | --- |
| Kode US | CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| US-10 | Dashboard | Gambar 3.29 |

1. *Code* system

<a href='https://www.google.com/maps/dir/?api=1&origin="+location.coords.latitude+","+location.coords.longitude+"&destination="+data[i].latitude+","+data[i].longitude+"' class='btn btn-sm btn-outline-success' target='\_blank'>Rute</a></h6></div>

1. Tampilan dan pengujian system



**Gambar 4.10** tampilan rute menuju lokasi tempat ibadah

Gambar 4.27 merupakan implementasi halaman peta rute menuju lokasi tempat ibada. Berikut merupakan tabel pengujian system terhadap kasus kode *user stories* US-11 dapat dilihat pada table 4.38.

Tabel 4.38 pengujian peta rute menuju lokasi tempat ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol rute menuju lokasi tempat ibadah | Rute menuju lokasi tempat ibadah berhasil ditampilkan | Sesuai | Valid |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.38, dapat dikatakan system yang dibuat berhasil dan sesuai dengan kebutuhan *client.*

1. **Pengujian system**

Walaupun pengujian system telah dilakukan sebelumnya, namun system harus tetap diuji lagi bermaksud untuk memastikan agar system yang dibuat tidak terjadi *error* atau *bug* pada system dan sesuai dengan keinginan *client*.

1. **Hasil pengujian *black box***

Metode pengujian *black box* merupakan metode pengujian yang menguji fungsi - fungsi di dalam sistem untuk menentukan apakah fungsi-fungsi tersebut sudah berjalan sesuai harapan atau tidak. Berikut merupakan Hasil pengujian sistem yang telah dibuat dapat dilihat pada Tabel 4.39.

Tabel 4.39 hasil pengujian *black box*

|  |  |  |
| --- | --- | --- |
| Fitur | Ditunjukan tabel | Hasil uji |
| Login ke dalam system | Tabel 4.3 | Diterima |
| Mengakhiri system (*logout*) | Tabel 4.4 | Diterima |
| Melihat daftar dan detail data admin | Tabel 4.11 | Diterima |
| Menambah data admin | Tabel 4.14 | Diterima |
| Mengubah data admin | Tabel 4.16 | Diterima |
| Menghapus data admin | Tabel 4.20 | Diterima |
| Melihat daftar dan detail data tempat ibadah | Tabel 4.8 | Diterima |
| Menambah data tempat ibadah | Tabel 4.23 | Diterima |
| Mengimport dan export data tempat ibadah | Tabel 4.25 | Diterima |
| Mengubah data tempat ibadah | Tabel 4.28 | Diterima |
| Menghapus data tempat ibadah | Tabel 4.31 | Diterima |
| Melihat peta lokasi tempat ibadah *android* | Tabel 4.34 | Diterima |
| Melihat peta lokasi tempat ibadah *web* | Tabel 4.36 | Diterima |
| Melihat rute lokasi tempat ibadah | Tabel 4.38 | Diterima |