**USULAN TUGAS AKHIR**

**RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS UNTUK PEMETAAN LOKASI TEMPAT IBADAH DI KOTA MATARAM MENGGUNAKAN METODE EXTREME PROGRAMMING**



**Oleh :**

**RICKY SATRIA**

**F1D 016 077**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**2020**

**PROPOSAL TUGAS AKHIR**

**RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS UNTUK PEMETAAN LOKASI TEMPAT IBADAH DI KOTA MATARAM MENGGUNAKAN METODE EXTREME PROGRAMMING**

Oleh :

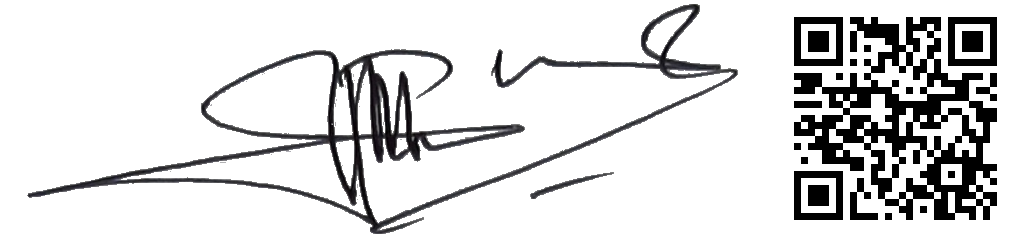
**RICKY SATRIA**

F1D016077

Telah diperiksa oleh Tim Pembimbing:

|  |  |  |
| --- | --- | --- |
| 1. | Pembimbing Utama ttd  **Royana Afwani, ST., MT. NIP. 198507072014042001** | Tanggal: 07/10/2020 |
| 2. | Pembimbing Pendamping ttd  **Ir. Sri Endang Anjarwani, M.Kom NIP. 196604032006042001** | Tanggal: 16/10/2020 |

Mengetahui,  
Ketua Program Studi Teknik Informatika  
Fakultas Teknik  
Universitas Mataram



**Prof. Dr. Eng. I Gede Pasek Suta Wijaya, ST., MT.  
NIP. 197311302000031001**

**PROPOSAL TUGAS AKHIR**

**RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS UNTUK PEMETAAN LOKASI TEMPAT IBADAH DI KOTA MATARAM MENGGUNAKAN METODE EXTREME PROGRAMMING**

Oleh :

**RICKY SATRIA**

F1D016077

Telah diujikan di depan penguji  
Pada tanggal 31 Agustus 2020

Susunan Tim Penguji :

|  |  |  |
| --- | --- | --- |
| 1. | Penguji 1 ttd  **Nadiyasari Agitha, S.Kom.,M.MT NIP. 198608132018032001** | Tanggal: 16/09/2020 |
| 2. | Penguji 2 ttd  **Moh. Ali Albar, ST., M.Eng. NIP. 198311252015041002** | Tanggal: 07/10/2020 |
| 3. | Penguji 3 ttd  **Ari Hernawan, S.Kom., M.Sc. NIP. 199001142019031018** | Tanggal: 07/10/2020 |

Mengetahui,  
Ketua Program Studi Teknik Informatika  
Fakultas Teknik  
Universitas Mataram



**Prof. Dr. Eng. I Gede Pasek Suta Wijaya, ST., MT.  
NIP. 197311302000031001**

**ABSTRAK**

Rumah ibadah merupakan suatu tempat bertemunya para umat beragama untuk beribadah menurut ajaran agama atau kepercayaan mereka masing-masing. Rumah ibadah sangatlah penting bagi seseorang yang hendak ingin meningkatkan hubungan spritual dengan Tuhan. Mengetahui informasi lokasi tempat ibadah sangatlah penting untuk memenuhi kewajiban peribadatan. Di Kota Mataram, pencarian tempat ibadah terbilang masih cukup sulit karena tidak mengetahui posisi/jalan di sekitar mereka berada. Dari sisi kemenag juga masih belum memiliki system pengelolaan data tempat ibadah, saat ini pengelolaan data tempat ibadah masih dilakukan menggunakan Ms.Excel yang membuat pendataan menjadi tidak efektif dan membutuhkan waktu yang lama dalam melakukan pencarian terhadap data-data yang ada, media informasi dan pemetaan tempat ibadah di Kota Mataram juga belum ada. Pada penelitian Tugas Akhir ini, dirancang dan dibangun sebuah Sistem Informasi Geografis Pemetaan Lokasi Tempat Ibadah di Kota Mataram guna untuk memberikan informasi dan tata letak mengenai tempat ibadah yang ada di Kota Mataram karena saat dilakukan pencarian pencarian tempat ibadah berarti letak geografisnya yang lebih utama dibutuhkan, dan dari sisi kemenag juga dapat dilakukan pendataan dari system yang membuat pengelolaan menjadi lebih efektif dan efisien. Dalam penelitian Tugas Akhir ini digunakan metode *Extreme Programming* atau XP. Pada XP terdapat model Personal *Extreme Programming* atau PXP yang proses pengembangannya dapat dilakukan oleh *programmer* tunggal, dimana dalam kasus penelitian Tugas Akhir ini dikerjakan oleh penulis sendiri. Dalam XP menekankan konsep lebih berinteraksi dengan *client* dalam proses pengambangan sistem dan berfokus pada sistem yang dikembangkan, sehingga meminimalisir terjadinya kesalahan pada sistem dan sesuai dengan kebutuhan *client,* dalam kasus penelitian Tugas Akhir ini pihak Kementerian Agama prov. NTB sebagai *client.*

**Kata Kunci** **:** SIG, Kota Mataram, tempat ibadah, *Extreme Programming*

**BAB 1**

**PENDAHULUAN**

## Latar Belakang

Agama merupakan suatu sistem yang mampu mengatur tata keimanan dan kepercayaan serta ibadah pada Tuhan Yang Maha Kuasa disertai dengan tata kaidah yang berkaitan langsung dengan ciri pergaulan manusia dengan manusia lainnya ataupun manusia dengan lingkungan sekitarnya[1]. Tempat ibadah merupakan suatu tempat bertemunya para umat beragama untuk beribadah menurut ajaran agama atau kepercayaan mereka masing-masing. Tempat peribadatan setiap harinya atau waktu tertentu selalu banyak dikunjungi oleh umatnya. Kunjungan ke tempat ibadah tersebut akan meningkat ketika ada kegiatan khusus atau perayaan hari besar keagamaan sehingga banyak interaksi terjadi[2]. Sarana peribadatan seperti rumah ibadah merupakan bangunan yang memiliki ciri – ciri tertentu yang khusus dipergunakan untuk beribadat bagi para pemeluk agama masing – masing[3].

Berdasarkan kuesioner yang telah dilakukan terhadap 61 responden, 55,7% data responden menyatakan bahwa fungsi rumah ibadah yaitu digunakan bagi seseorang yang hendak ingin meningkatkan hubungan spritual dengan Tuhan-Nya dan 34,4% data responden menyatakan rumah ibadah digunakan bagi orang muslim yakni untuk mendapatkan lebih banyak pahala daripada sholat dirumah. 62,3% data responden menyatakan cukup sulit dalam melakukan pencarian tempat ibadah di Kota Mataram, yang disebabkan karena 68,8% data responden menunjukkan bahwa mereka tidak mengetahui posisi/jalan di sekitar mereka berada. Adapun berbagai macam solusi yang dilakukan masyarakat saat ini dalam mengatasi masalah tersebut yaitu 34,4% data responden memilih bertanya kepada penduduk sekitar, 32.8% memilih untuk menelusuri jalanan dan 29.5% memilih untuk menggunakan *google maps*. 75,4% data respoden berpendapat bahwa mengetahui informasi lokasi tempat ibadah sangatlah penting untuk memenuhi kewajiban peribadatan. Untuk mengetahui penting atau tidaknya dibangun Sistem Informasi Geografis untuk Pemetaan Lokasi Tempat ibadah di Kota Mataram, didapatkan hasil data responden 54.1% berpendapat sangat penting 44.3% berpendapat penting, dan 1.6% berpendapat kurang penting. Dilihat dari hasil kuesioner tersebut dapat dikatakan bahwa masyarakat setuju untuk dibangunnya sistem informasi pemetaan lokasi tempat ibadah di Kota Mataram tersebut. Dan 75,4% data responden memilih untuk dibuatkan sistem berbasis *android*, karena lebih efektif dan efisien dalam penggunaannya.

Berdasarkan hasil wawancara yang telah dilakukan terhadap salah satu *staff* sub. bagian Inmas Kementerian Agama provinsi NTB yaitu bapak H. Toni, menyebutkan bahwa dalam pengelolaan data informasi mengenai tempat ibadah yang ada saat ini masih menggunakan cara konvensional yaitu menggunakan Ms. Excel, sehingga pendataan sering kali tidak efektif serta membutuhkan waktu yang cukup lama ketika melakukan pencarian terhadap data-data yang ada. Berdasarkan observasi yang penulis lakukan juga tidak adanya media informasi dan pemetaan tempat ibadah di Kota Mataram, sistem informasi geografis untuk lokasi tempat ibadah di Kota Mataram juga masih belum ada.

Sistem Informasi Geografis (SIG) merupakan suatu sistem informasi yang dirancang untuk bekerja dengan data yang bereferensi spasial atau berkordinat geografis. SIG berarti sifat dari informasinya yaitu mengenai obyek-obyek atau hal-hal yang ada atau terjadi atau diperkirakan terjadi di muka bumi, tepatnya disuatu lokasi entah itu wilayah yang luas atau kecil kecil, bisa rumah, kampung, desa, kota, hutan, sawah, negara, bahkan dunia, tergantung dari maksud. Geografis atau informasi geografis bisa juga ditandai dengan data-data seperti koordinat [4].Dalam penelitian Tugas Akhir ini, SIG sangat cocok untuk mengatasi masalah yang telah dipaparkan sebelumnya, guna untuk memberikan informasi dan tata letak mengenai tempat ibadah yang ada di Kota Mataram karena saat dilakukan pencarian tempat ibadah berarti letak geografisnya yang lebih utama dibutuhkan.

Menurut salah satu *staff* sub. bagian Inmas Kementerian Agama provinsi NTB yaitu Hj. Diah Purnawati, sistem yang sudah ada di kantor Kementerian Agama provinsi NTB saat ini yaitu sistem informasi berbasis *website* yang dapat diakses melalui link berikut: <https://ntb.kemenag.go.id>. Berdasarkan hal tersebut sistem informasi yang akan penulis buat dapat menjadi salah satu fitur atau menu dalam sistem informasi yang sudah ada, dimana guna dari system yang akan dibuat yaitu untuk memberikan informasi kepada masyarakat umum dan dapat dikelola oleh pihak Kemenag prov. NTB mengenai data tempat ibadah yang ada.

Dalam penelitian Tugas Akhir ini digunakan metode *Extreme Programming* atau XP. Pada XP terdapat model Personal *Extreme Programming* atau PXP yang proses pengembangannya dapat dilakukan oleh *programmer* tunggal, dimana dalam kasus penelitian Tugas Akhir ini dikerjakan oleh penulis sendiri. Dalam XP menekankan konsep lebih berinteraksi dengan *client* dalam proses pengambangan sistem dan berfokus pada sistem yang dikembangkan, sehingga meminimalisir terjadinya kesalahan pada sistem dan sesuai dengan kebutuhan *client,* dalam kasus penelitian Tugas Akhir ini pihak Kementerian Agama prov. NTB sebagai *client*.

Berdasarkan hal tersebut, maka penulis akan merancang dan membangun suatu sistem informasi geografis yang berjudul “Rancang Bangun Sistem Informasi Geografis Untuk Pemetaan Lokasi Tempat Ibadah Di Kota Matram Menggunakan Metode *Extreme Programming*”. Diharapkan setelah terciptanya sistem ini nantinya dapat membantu pihak Kemenag prov. NTB dalam mengelola data informasi tempat ibadah yang ada serta membantu masyarakat umum dalam mengetahui informasi dan lokasi mengenai tempat ibadah yang ada di Kota Mataram dengan mudah.

## Rumusan Masalah

## Berdasarkan uraian latar belakang yang ada maka dirumuskan masalah dalam tugas akhir ini adalah sebagai berikut:

## Bagaimana melakukan rancang bangun sistem informasi berbasis *website* dan *android* yang dapat menampilkan infromasi dan lokasi tempat ibadah di Kota Mataram dalam bentuk peta digital yang dapat diakses secara *online* dan memberikan rute/jalur perjalanan dari posisi *user* berada menuju tempat ibadah yang diinginkan?

## Bagaimana melakukan rancang bangun sistem informasi berbasis *web* yang dapat digunakan sebagai media informasi dan mempermudah Kemenag prov. NTB dalam melakukan pengelolaan terhadap data tempat ibadah yang ada?

## Bagaimana mengimplementasikan metode *Extreme Programming* dalam merancang dan membangun “Sistem Informasi Geografis Untuk Pemetaan Lokasi Tempat Ibadah Di Kota Mataram”?

## Batasan Masalah

## Penelitian ini memiliki batasan-batasan masalah untuk memberikan lingkup penelitian agar lebih terfokus ketika pengerjaan. Adapun batasan masalah yang diberikan adalah sebagai berikut:

## Sistem informasi tempat ibadah ini menyediakan informasi berbasis *web* dan *android* yang hanya dapat diakses secara *online*.

## Pada sistem informasi berbasis *web* digunakan oleh masyarakat umum untuk mengetahui informasi mengenai tempat ibadah dan digunakan oleh pihak kemenag prov. NTB untuk mengelola data tempat ibadah yang sudah ada.

## Pada sistem informasi berbasis *android* digunakan oleh masyarakat umum untuk mengetahui informasi dan lokasi mengenai tempat ibadah dalam bentuk peta digital dengan rute/jalur dari posisi *user* berada menuju ke tempat peribadatan yang diinginkan.

## Sistem informasi ini diperuntukan hanya lokasi tempat ibadah yang ada di Kota Mataram.

## Pembuatan peta tempat ibadah ini menggunakan peta dari *Google maps.*

## Sistem informasi ini hanya menampilkan lokasi tempat ibadah, tidak dengan yang lain.

## Pengguna tidak dapat menambahkan lokasi pada sistem.

## Tujuan

## Adapun tujuan dari penulisan tugas akhir ini adalah sebagai berikut:

## Membantu masyarakat dalam memberikan informasi mengenai tempat ibadah di Kota Mataram dan memberikan jalur perjalanan menuju tempat ibadah yang dibutuhkan.

## Membantu mempermudah Kementerian Agama Provinsi NTB dalam melakukan pengelolaan terhadap data tempat ibadah yang ada.

## Mengimplementasikan metode *Extreme Programming* dalam proses perancangan dan pembangunan “Sistem Informasi Geografis Untuk Pemetaan Lokasi Tempat Ibadah Di Kota Mataram”

## Manfaat

## Adapun manfaat dari penulisan tugas akhir ini adalah sebagai berikut:

## Memudahkan Kemenag Prov. NTB dalam mengelola data tempat ibadah yang ada.

## Memudahkan masyarakat dalam mencari tau informasi dan lokasi mengenai tempat ibadah serta memudahkan dalam menggapai tempat ibadah yang ada di Kota Mataram.

## Sistematika Penulisan

## Sistematika penulisan tugas akhir adalah sebagai berikut:

## Bab 1 Pendahuluan

## Bab ini mengemukakan latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan yang digunakan untuk menyusun laporan tugas akhir.

## Bab 2 Tinjauan Pustaka dan Dasar Teori

## Bab ini berisi dasar teori yang digunakan dalam melakukan analisis, perencangan dan implementasi tugas akhir yang dilakukan pada bab bab selanjutnya.

## Bab 3 Metodologi Perancangan

## Bab ini berisi tentang pelaksanaan, alat, bahan, jalannya perencanaan, dan hasil yang diharapkan.

## Bab 4 Hasil dan Pembahasan

## Bab ini memuat tentang analisa dan pembahasan hasil penelitian yang telah dilakukan.

## Bab 5 Penutup

## Bab ini berisi kesimpulan dan saran-saran berdasarkan hasil pembahasan yang diproleh.

**BAB II**

**TINJAUAN PUSTAKA DAN LANDASAN TEORI**

## Tinjauan Pustaka

## Pembuatan sistem informasi mengenai tempat ibadah telah banyak dilakukan oleh peneliti terdahulu. Adapun sistem informasi yang telah dibuat oleh peneliti terdahulu diantaranya adalah sebagai berikut:

## Penelitian yang berjudul “Implementasi *Geographic Information System* (GIS) Penentuan Tampat Ibadah Masjid Di Kecamatan Kempas Kabupaten Indragiri Hilir Provinsi Riau” yang bertujuan untuk dapat membantu masyarakat terutama yang berasal dari luar kota Kecamatan Kempas maupun dari daerah-daerah lain untuk menemukan suatu tempat ibadah di daerah Kecamatan Kempas. Aplikasi WebGis ini dibuat dengan menggunakan GIS, *Dreamuwever CS 5*, *MySql*, dan *Web Browser* sebagai media untuk menyajikan informasinya. Dimana informasi yang disajikan hanya informasi mengenai lokasi atau letak tempat ibadah masjid saja yang ada di Kecamatan Kempas. Untuk metode pengembangan sistem yang digunakan yaitu SDLC (*System Development Life Cycle*). SDLC merupakan metodologi klasik yang digunakan oleh analis sistem dan programmer dalam mengembangkan, memilihara, dan menggunakaan sistem informasi [8].

## Pada penelitian yang berjudul “Sistem Informasi Geografis Persebaran Tempat Ibadah Dan Kapasitasnya Berdasarkan Jumlah Penduduk Di Kota Banjarmasin” merupakan suatu pembangunan sistem informasi geografis tempat ibadah berbasis web dengan menggunakan data-data yang didapatkan dari Kementrian Agama Kota Banjarmasin, serta peta administrasi Kota Banjarmasin. Aplikasi WebGis ini dibangun menggunakan Quantum GIS dan Google Maps. Dengan adanya aplikasi ini akan dapat dijadikan pertimbangan untuk pengambilan keputusan dalam menentukan layak atau tidak untuk mendirikan tempat ibadah baru di Kota Banjarmasin [9].

## Dalam penelitian yang berjudul “Sistem Informasi Masjid Di Surakarta” dibangun untuk menampilkan informasi mengenai masjid yang ada di Surakarta. Dimana tujuannya tidak hanya untuk memberikan kemudahan kepada masyarakat untuk mencari informasi seputar masjid di Surakarta, tetapi juga digunakan pihak IT di Kementrian Agama Kantor Kota Surakarta untuk memantau segala sesuatu yang ada di dalam *website* serta dapat menyebarluaskan informasi menjadi lebih efektif dan efisien. Aplikasi berbasis *website* ini dibangun menggunakan bahasa pemrograman PHP dan MySQL untuk penyimpanan *database*-nya, serta untuk *user interface*-nya dibangun menggunakan bahasa pemrograman HTML dan CSS. *Website* pada penelitian ini dilengkapi fitur *searching*, *maps*, dan *newsfeed* [10].

## Pada penelitian yang berjudul “Membangun Aplikasi WebGis Untuk Tempat Ibadah Di Provinsi Lampung” yang membahas tentang pembuatan aplikasi *WebGis* yang dapat digunakan untuk mencari tempat ibadah yang ada di Lampung secara online. Aplikasi pada penelitian ini dibangun menggunakan bahasa pemrograman HTML dan PHP dengan *database* MySql. Untuk metode pengembangan sistem yang digunakan yaitu *Extreme Programming. Extreme Programming* adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel [5].

## Dalam penelitian yang berjudul “Perancangan Sistem Informasi Tempat Ibadah Beragam Agama Di Kota Solo” yang membahas tentang pembuatan sistem informasi berbasis *website* yang berguna untuk menampilkan dan memuat informasi tempat ibadah beragam agama di Kota Solo, yaitu: Masjid, Gereja, Pura, Vihara, dan Kelenteng. Sistem informasi berbasis website ini Dibangun menggunakan bahasa pemrograman PHP dan *database* MySql. Untuk metode pengembangan sistem yang digunakan yaitu metode *waterfall* [5]*.*

## Penelitian yang berjudul “Sistem Informasi Geografis Pemetaan Masjid Di Samarinda Berbasis Web” merupakan suatu pembangunan sistem informasi geografis pemetaan masjid berbasis web yang bertujuan untuk memberikan informasi kepada masyrakat yang ada di Samarinda tentang tempat ibadah masjid beserta kegiatan yang ada didalamnya dalam visualisasi peta menggunakan Google Maps API dan terdapat fitur rute yang menggambarkan rute arah jalan dari posisi *user* berada menuju lokasi masjid yang diinginkan. Aplikasi pada penelitian ini dibuat menggunakan bahasa pemrograman PHP dan untuk *database*-nya menggunakan MySQL. Untuk metode pengembangan sistem yang digunakan yaitu *waterfall. Waterfall* memiliki beberapa tahapan dalam pengembangan sistem yang dilakukan, yakni: tahap perencanaan, tahap analsis, tahap perancangan, tahap penerapan, tahap pengujian dan tahap pemeliharaan[11].

## Berdasarkan tinjauan pustaka yang telah dipaparkan, penulis akan merancang dan membangun Sistem Informasi Geografis Untuk Pemetaan Lokasi Tempat Ibadah di Kota Mataram menggunakan metode *Extreme Programming*. Studi kasus ini diambil karena belum ada penelitian tentang SIG pemetaan lokasi tempat ibadah di Kota Mataram sehingga penulis mengusulkan tugas akhir dengan judul tersebut. Tujuan dari penelitian ini yaitu untuk mempermudah masyarakat umum dalam mencari mencari tau informasi dan lokasi tempat ibadah yang diinginkan serta memudahkan pihak Kemenag prov. NTB dalam mengelola data tempat ibadah yang ada.

## Pada Gambar 2.1 berikut merupakan *fishbone* diagram dari tinjauan 6 pustaka dan usulan sistem yang akan dikembangkan:

|  |
| --- |
|  |

**Gambar 2.1** *Fishbone* diagram

## Dasar Teori

## Berikut merupakan teori-teori dasar atau umum yang digunakan dalam penelitian tugas akhir ini.

## *Google Maps API*

## Svennerberg telah mencatat bahwa, Google Maps API adalah API yang paling popular di internet. Hasil pencatatan pada bulan Mei tahun 2010 menyatakan bahwa 43% mashup (aplikasi dan situs web yang menggabungkan dua atau lebih sumber data) menggunakan Google Maps API. Beberapa tujunnya dari penggunaan Google aMaps API adalah melihat lokasi, serta mencari alamat dan lain sebagainya [12].

## Google maps API adalah fungsi-fungsi pemrograman yang disediakan oleh Google maps agar Google maps bisa di integrasikan kedalam Web atau aplikasi yang sedang buat. Contoh sederhanya misalkan penulisingin membuat Sistem informasi Geografis kampus di jogja, dengan memanfaatkan Google Maps API penulis bisa membuat GIS tanpa perlu memikirkan Peta Jogja, penulis tinggal pake Google maps dan memanggil fungsi fungsi yang dibutuhkan seperti menampilkan peta, menempatkan *marker* dan lain sabagainya [13].

## GPS

## *Global Positioning System* (GPS) berfungsi untuk mengetahui letak tempat yang akan dituju dan mengetahui dimana pengguna berada, memberikan informasi yang tepat dan akurat mengenai posisi, kecepatan, arah, dan waktu dengan bantuan sinyal satelit [12].

## *Website*

## *Website* merupakan sebuah kumpulan halaman-halaman *web* beserta *file-file* pendukungnya, seperti *file* gambar, *video*, dan *file* digital lainnya yang disimpan pada sebuah *web server* yang umumnya dapat diakses melalui internet. Atau dengan kata lain, *website* adalah sekumpulan *folder* dan *file* yang mengandung banyak perintah dan fungsi fungsi tertentu, seperti fungsi tampilan, fungsi menangani penyimpanan data, dsb[14].

## Sistem Informasi

## Sistem merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan dalam usaha mencapai suatu tujuan. Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem tersebut menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan dan dibutuhkan oleh orang untuk menambah pemahamannya terhadap fakta-fakta yang ada. Sistem Informasi juga dapat didefinisiskan sebagai perangkat elemen yang bekerja mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengawasan, analisis, dan visualisasi dalam organisasi [15].

## Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang menudukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan [16].

## Sistem Informasi Geografis

Sistem Informasi Geografis merupakan suatu sistem informasi yang dirancang untuk bekerja dengan data yang bereferensi spasial atau berkordinat geografis atau dengan kata lain SIG adalah suatu sistem basis data dengan kemampuan khusus untuk menangani data yang bereferensi keruangan (spasial) bersamaan dengan seperangkat operasi kerja.

## Sistem informasi geografis memiliki tiga unsur yakni sebagai berikut.

## Sistem, dapat diartikan sebagai berbagai hal yang saling berkaitan atau saling mempengaruhi dalam mengerjakan proses untuk satu tujuan dalam SIG.

## Informasi, sesuai dengan karakter SIG, informasi disini tentu saja adalah informasi tentang bumi (geografis) dengan apa yang ada di bumi. SIG merupakan media untuk menggambarkan apa yang ada di bumi dengan segala yang ada sesuai dengan tenpat atau lokasi dia berada. Informasi inilah yang menjadi obyek kerja SIG.

## Geografis, geografis dalam SIG berarti sifat dari informasinya yaitu mengenai obyek-obyek atau hal-hal yang ada atau terjadi atau diperkirakan terjadi di muka bumi, tepatnya disuatu lokasi entah itu wilayah yang luas atau kecil kecil, bisa rumah, kampung, desa, kota, hutan, sawah, negara, bahkan dunia, tergantung dari maksud. Geografis atau informasi geografis bisa juga ditandai dengan data-data seperti koordinat [4].

## *Framework CodeIgniter*

## *Codeigniter* adalah sebuah *framework* PHP yang dibuat berdasarkan *design pattern* *model view controller* atau biasa disingkat MVC. *Design Pattern* adalah kumpulan penjelasan mengenai metode-metode bagaimana cara menyelesaikan suatu masalah yang umum ditemui dalam proses perancangan perangkat lunak (*Software Design*) [14].

## PHP

## PHP Merupakan singkatan *recursive* dari PHP : *Hypertext Prepocessor*. Pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994 [17]. PHP merupakan bahasa pemograman untuk membuat web. PHP dapat digunakan untuk membuat halaman *web* yang dinamis. PHP dapat digunakan pada sistem operasi Windows, Mac OS, Linux, dan sistem operasi yang lainnya.

## XAMPP

## XAMPP adalah perangkat yang menggabungkan tiga aplikasi kedalam satu paket yaitu Apache, MySQL,dan PhpMyAdmin, Dengan Xampp pekerjaan menjadi sangat dimudahkan karena dapat menginstalasi dan mengkonfigurasi ketiga aplikasi tersebut dengan sekaligus dan otomatis. Xampp telah mengalami perkembangan dari waktu ke waktu.versi yang terbaru adalah revisi dari yang terdahulu, sehingga lebih baik dan lebih lengkap. Aplikasi utama dalam paket Xampp yakni terdiri atas *web server* Apache, MySQL, PHP, dan PhpMyAdmin.

* + 1. **Apache**

Apache adalah sebuah *web server open source*, jadi semua orang dapat menggunakannya secara gratis, bahkan anda bisa mengedit kode programnya. fungsi utama dari *apache* yakni menghasilkan halaman *web* yang benar sesuai dengan yang dibuat oleh seorang *web programmer*, dengan menggunakan kode PHP [18].

* + 1. ***Web server***

*Web Server* merupakan sebuah perangkat lunak dalam *server* yang berfungsi menerima permintaan (*request*) berupa halaman web melalui HTTP atau HTTPS dari *klien* yang dikenal dengan *browser web* dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML [17].

* + 1. **PHPMyAdmin**

PHPMyAdmin merupakan *front-end* MySQL berbasis *web*. PHPMyAdmin dibuat dengan menggunakan PHP. Saat ini, PHPMyAdmin banyak digunakan dalam hampir semua penyedia hosting yang ada di internet. PHPMyAdmin mendukung berbagai fitur administrasi MySQL termasuk manipulasi database, tabel, index dan juga dapat mengeksport data ke dalam berbagai format data [17].

* + 1. **MySQL**

## MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia[17]. MySQL adalah sistem manajemen *database* yang sering digunakan bersama PHP. PHP juga mendukung pada Microsoft Access, Database Oracle, dBase, dan sistem manajemen database lainnya. SQL (*Structured Query Language*) adalah bahasa terstruktur yang digunakan secara khusus untuk mengolah database, dan MySQL merupakan sebuah sistem manajemen *database* [18].

## *Visual Studio Code*

## *Visual Studio Code* menyediakan pengembang dengan pilihan baru alat pengembang yang menggabungkan kesederhanaan dan pengalaman ramping dari *code editor* yang terbaik dari apa yang pengembang butuhkan untuk siklus kode-builddebug inti mereka. *Visual Studio Code* adalah editor kode pertama, dan alat pengembangan lintas *platform* pertama - mendukung OS X, Linux, dan Windows [19].

## *Black Box Testing*

## *Black Box* adalah teknik pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi masukan dan melakukan pengetesan pada spesifikasi fungsional. Tujuan *Black Box Testing* untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang telah diharapkan dan apakah informasi yang disimpan serta eksternal selalu dijaga kemutakhirannya. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih sebagai pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* mengidentifikasi jenis kesalahan dalam beberapa kategori antara lain fungsi yang tidak benar atau tidak ada, kesalahan antarmuka (*interface errors*), kesalahan pada struktur data dan akses basis data, kesalahan performansi (*performance errors*) dan kesalahan inisialisasi maupun terminasi[20].

## *Bootstrap*

## *Bootstrap* merupakan *framework* HTML, CSS, *Javascript* populer untuk membangun situs web yang *responsive*. RWD (*Responsive Web Design*) adalah desain situs yang otomatis akan menyesuaikan diri agar tampil baik di semua perangkat dari ponsel sampai dekstop. Bootstrap terdiri dari satu set *file* css, *javascript*, dan *jquery*[21].

## *Postman*

## *Postman* adalah satu-satunya *platform* pengembangan API yang lengkap. Alat bawaan *Postman* mendukung setiap tahap siklus hidup API. Anda dapat merancang dan men-*debug*, secara otomatis menguji, mendokumentasikan, memantau, dan menerbitkan API.

## *Postman* tersedia untuk MacOS, Windows, dan Linux. *Platform* pengembangan API lengkap *Postman* memungkinkan kita untuk membuat permintaan, membangun dan mengelola koleksi, dan mengatur siklus hidup API yang kita miliki[22].

## *Android*

## *Android* adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia[23].

## Android Studio

## Android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio mengantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.

## Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Android studio memiliki fitur : a. Projek berbasis pada Gradle Build b. Refactory dan pembenahan bug yang cepat c. Tools baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat. d. Mendukung Proguard And App-signing untuk keamanan. e. Memiliki GUI aplikasi android lebih mudah f. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan [24].

## *Java Development Kit* (JDK)

## Java Development Kit (JDK) adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis Java, sedangkan JRE adalah sebuah implementasi dari Java Virtual Machine yang benarbenar digunakan untuk menjalankan program java. Baisanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber compiler java, bundling, debuggers, development libraries dan lain sebagainya[24].

## Metode pengembangan sistem *Extreme Programming*

## *Extreme Programming* (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan–perubahan requirement yang sangat cepat [6]. XP ditujukan untuk tim kecil dan menengah, ukuran tim dibatasi antara tiga dan maksimal dua puluh anggota proyek, dan *pair programming* (dua *programmer* yang melakukan pengkodean di satu computer) merupakan karakteristik dari XP. Dibandingkan dengan *scrum* yang beroperasi dalam tim yang terdiri dari lima hingga sembilan anggota [25]. Pada Gambar 2.2 berikut merupakan kerangka kerja metode Extreme Programming (XP) in meliputi: *Planning*, *Design*, *Coding* dan *Testing* :

## 

## Gambar 2.2 Siklus *Extreme Programming*

## Siklus *Extreme Programming* membagi sebuah proyek menjadi beberapa fase yaitu:

## *Planning* : Pembuatan perencanaan untuk mendapatkan gambaran fitur dan fungsi dari perangkat lunak yang akan dibangun. Aktivitas *planning* dimulai dengan membuat kumpulan gambaran atau cerita (*user stories*) yang telah diberikan oleh *client* kemudian dari hasil *user stories* tersebut maka akan dilakukan perencanaan kebutuhan-kebutuhan mana yang akan dikerjakan proses ini dinamakan *iteration planning*.

## *Design* : Metode ini menekankan desain aplikasi yang sederhana (*simple design*). Pada fase ini digunakan *Class Responsibilities Collaboration Card* (CRC-Card) untuk merancang sebuah sistem dan untuk mengetahui interaksi atau hubungan antar objek yang ada pada sistem. CRC-*Card* merupakan sebuah kartu indeks yang terbagi menjadi tiga bagian yaitu nama kelas, *responsibility*, dan *collaborator* yang digunakan untuk mendeklarasikan objek-objek yang ada pada kelas dan kelas yang saling berhubungan. Digunakannya CRC-*Card* pada metode XP adalah karena mengacu pada nilai kesederhanaan yang ada pada metode XP itu sendiri.

## *Coding* : XP menerapkan konsep *Pair Programming* dimana proses *coding* dikembangkan oleh dua orang *programmer*, namun XP juga dapat dilakukan oleh *programmer* tunggal yang disebut sebagai model Personal *Extreme Programmer* (PXP). Model PXP ini memiliki prinsip, nilai, dan langkah kerja yang sama dengan XP[7]. Dalam membangun sebuah aplikasi pada metode XP ini terlebih dahulu dilakukan pengujian terhadap kode program yang akan diimplementasikan. Pengujian kode program pada fase ini menggunakan teknik *Test Driven Development* (TDD) dimana melakukan pengembangan perangkat lunak berdasarkan hasil pengujian yang telah dilakukan, hal ini bermaksud untuk dapat mengurangi waktu pengembangan dan dapat minimalisir adanya *bug* atau *error* pada aplikasi[26].

## *Testing* : Dalam XP juga terdapat *acceptance test* atau biasa disebut *customer test*. Pengujian ini dilakukan oleh *customer* yang berfokus kepada fitur dan fungsi dari sistem secara keseluruhan. *Acceptance test* ini berasal dari *user stories* yang telah diimplementasikan [27].

## Setelah semua pengujian selesai dilakukan, apabila terdapat *error* atau kesalahan pada aplikasi maka akan segera dilakukan perbaikan pada struktur kode atau dilakukan *refactoring* pada *error* tersebut. Dan apabila terdapat penambahan permintaan kebutuhan (*software increment*) dari pengguna maka akan dicatat dan masuk ke dalam *iteration planning*. Jika tidak ada *error* ataupun permintaan kebutuhan baru oleh pengguna makan aplikasi akan diluncurkan atau *release*[26].

## *Personal Extreme Programming*

## Personal Extreme Programming atau PXP merupakan metode pengembangan perangkat lunak yang termasuk kedalam *agile software development.* PXP merupakan metode pengembangan dari *Extreme Programming* yang disesuaikan agar dapat dikerjakan oleh pengembang tunggal[28]. Pada Tabel 2.1 menunjukkan pemetaan antara XP dan PXP dalam hal dua belas praktik inti[29].

## Tabel 2.1 *Twelve Core Practices Compared*

|  |  |  |  |
| --- | --- | --- | --- |
| No | *Core Practices* | *Extreme Programming* (XP) | *Personal Extreme Programming* (PXP) |
|  | *The Planning Game* | Kegiatan utama dalam *planning game* adalah menulis, estimasi, prioritas atau *user stories* yang dilakukan pengembang *client*. *Client* dan pengembangan berkumpul untuk memutuskan fitur apa dari sistem yang diperlukan yang akan menjadi nilai maksimum bagi bisnis. | Jika pengembang bekerja untuk dirinya sendiri atau cukup tahu untuk menggantikan pelanggan, maka seseorang hanya perlu berganti peran (bermain peran) selama perencanaan. Merupakan praktik yang baik untuk menulis *user stories*, memperkirakan, memprioritaskan, dan melacak saat mengembangkan sendiri. Menulis dan menggunakan cerita adalah cara yang baik untuk memecah proyek besar menjadi bagian-bagian kecil yang lebih mudah ditangani dan dikelola. |
|  | *Small Releases* | Sistem sederhana yang berisi serangkaian fitur yang berguna dimasukkan ke dalam produksi lebih awal dan sering diperbarui dalam siklus yang singkat. | Rilis awal dan sering kali dapat diterapkan dengan mudah dengan pemrogram tunggal seperti dengan tim pengembang. |
|  | *Metaphor* | Setiap proyek memiliki "nama sistem" dan deskripsi yang membantu memandu proses pengembangan dan komunikasi antara semua pihak. | Tim satu orang dapat menggunakan dan menyempurnakan metafora apa pun yang terbukti paling baik. Selalu pilih metafora yang membantu Anda memahami bagian-bagian sistem yang Anda bicarakan |
|  | *Simple Design* | Desain paling sederhana selalu digunakan untuk membangun aplikasi selama memenuhi persyaratan bisnis saat ini. Jangan khawatir tentang persyaratan masa depan karena persyaratan berubah seiring waktu. | Sebagai pengembang tunggal, mudah untuk mempertahankan desain yang sederhana. Solusinya adalah menggunakan desain tingkat tinggi sebagai tujuan untuk dikerjakan, tetapi mengembangkan desain alternatif yang lebih sederhana dan lebih baik tetapi tetap memberikan semua fungsionalitas. |
|  | *Testing* | *Test-driven development* adalah salah satu kekuatan utama XP. Perangkat lunak yang dikembangkan dengan XP divalidasi setiap saat. Sebelum fitur baru ditambahkan, tes ditulis untuk memverifikasi perangkat lunak. Perangkat lunak tersebut kemudian dikembangkan untuk lulus tes ini. | Saat menulis modul baru, programmer harus menulis antarmuka terlebih dahulu, kemudian unit test, dan baru kemudian melanjutkan untuk mengimplementasikan modul. Dalam satu orang, setelan tes tim dapat dengan mudah ditulis dan digunakan. Pengembang harus menguji dan mengkompilasi kodenya sebelum memasukkannya ke dalam baseline refactor. Sampai saat itu, kode tetap berada di baseline pengembangan |
|  | *Refactoring* | Merupaan teknik untuk meningkatkan desain basis kode yang sudah ada. Esensinya adalah menerapkan serangkaian transformasi kecil yang melestarikan perilaku yang meningkatkan struktur kode. Melakukannya dalam langkah-langkah kecil mengurangi risiko terjadinya kesalahan | Sebagai seorang pemrogram tunggal, seseorang dapat dan harus mempraktikkan pemfaktoran ulang ke hulu kecuali dalam situasi di mana Anda tidak memiliki izin untuk mengubah kode yang tidak Anda miliki. Kode yang difaktorkan ulang harus diintegrasikan dengan baseline produksi dan diuji. |
|  | *Pair Programming* | Pemrogram yang menggunakan XP dipasangkan dan menulis semua kode produksi menggunakan satu mesin per pasang. Ini membantu kode untuk terus ditinjau saat sedang ditulis. Pemrograman berpasangan telah terbukti menghasilkan kode berkualitas tinggi dengan sedikit atau tanpa penurunan produktivitas. | Saat bekerja sebagai programmer mandiri, manfaat dari pemrograman berpasangan akan hilang. Anda bisa meminta rekan kerja untuk sering memberi Anda pengingat. Panduan informal dan uji filosofi pertama dapat diterapkan. |
|  | *Collective Code Ownership* | Semua kode adalah milik setiap anggota tim, tidak ada satu pun anggota tim yang memiliki sepotong kode dan siapa pun dapat membuat perubahan pada basis kode kapan saja. Ini mendorong semua orang untuk menyumbangkan ide-ide baru ke semua segmen proyek. | Jika Anda adalah satu-satunya pengembang dalam proyek ini maka tidak ada masalah karena Anda memiliki semua kode sumber. Namun, Anda tidak akan mendapatkan keuntungan dari masukan pemrogram lain. Pengembang dapat mempertahankan versi kode dalam repositori dan teknik kontrol versi dapat diterapkan. |
|  | *Continuous Integration* | Tujuan dari integrasi berkelanjutan adalah untuk mencegah atau mengurangi penyebaran kode dari basis kode utama; semakin sering kode diintegrasikan ke dalam basis kode utama, semakin kecil kemungkinan terjadinya pengalihan. Sistem perangkat lunak dibangun dan diintegrasikan beberapa kali sehari; setidaknya semua perubahan diintegrasikan ke dalam basis kode utama setidaknya sekali sehari. Setiap build diuji menggunakan kasus uji yang disiapkan. | Jika pengembang bekerja sendiri dengan tidak ada orang lain yang membuat perubahan pada basis kode utama, maka tidak ada masalah karena basis kode berfungsi sebagai catatan linier dari pekerjaan seseorang. Integrasi tidak akan menyebabkan konflik apa pun dan bebas masalah tetapi masih cukup mudah untuk menyimpang dari basis kode utama semakin lama Anda bekerja tanpa mengintegrasikan. Oleh karena itu, integrasi berkelanjutan tetap diperlukan. Cara terbaik untuk mengembangkannya adalah dengan mengerjakan tugas, mengintegrasikan, dan kemudian pindah ke tugas berikutnya. Dengan cara ini divergensi dijaga seminimal mungkin |
|  | 40*-Hour Week* | Programmer dalam proyek XP biasanya mematuhi 40 jam kerja seminggu untuk menjaga produktivitas dan menghindari kelelahan | Semua yang diperlukan untuk melakukan ini adalah mematuhi 40 jam seminggu dan berhenti bekerja pada hari yang Anda butuhkan. Artinya berhenti bekerja saat tidak produktif lagi, stres atau lelah untuk mengurangi rasa lelah dan membuat Anda dalam kondisi prima. |
|  | *On-site Customer* | Satu atau lebih pelanggan yang akan menggunakan sistem yang dibangun dialokasikan ke tim pengembangan. Pelanggan membantu memandu pengembangan dan diberdayakan untuk memprioritaskan dan menyatakan persyaratan atau menjawab pertanyaan apa pun yang mungkin dimiliki pengembang. Hal ini memastikan bahwa ada komunikasi yang efektif dengan pelanggan dan akibatnya akan lebih sedikit dokumentasi yang diperlukan | Jika Anda adalah pelanggan Anda sendiri (setidaknya pada awalnya) maka bergumam pada diri sendiri adalah hal yang wajar. Jika Anda memiliki pelanggan lain maka komunikasi melalui email atau telepon mungkin akan meredakan masalah ini selama pelanggan terbuka untuk komunikasi. |
|  | *Coding Standards* | Setiap orang menggunakan standar pengkodean yang sama yang memudahkan untuk bekerja berpasangan dan berbagi kepemilikan semua kode. Seseorang seharusnya tidak dapat mengetahui siapa yang mengerjakan kode apa dalam proyek XP. | Sebagai tim satu orang, cara Anda memilih untuk membuat kode adalah standar pengkodean Anda. Tetapi beberapa standar pengkodean harus diikuti untuk menjaga konsistensi selama pengembangan |

## BAB III

## METODE PENELITIAN

### Alat dan Bahan

### Dalam penelitian Tugas Akhir ini memerlukan beberapa alat yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*softaware*) serta bahan yang berupa data-data dan informasi yang diperlukan.

* + 1. **Alat**

Adapun alat-alat yang digunakan dalam penelitian Tugas Akhir ini dapat dirincikan sebagai berikut.

1. Laptop ACER Aspire 4739 dengan *processor* Core i3 2.4GbHz *hardisk* 500 Gb menggunakan OS (*operating system*) Windows 10 Pro.
2. *CodeIgniter,* merupakan *framework* bahasa pemrograman PHP yang digunakan untuk membangun sistem berbasis *web*.
3. XAMPP, sebagai peneydia *localhost web server.*
4. MySQL, sebagai *server database.*
5. Visual Studio Code*,* sebagai *text* editor pada saat melakukan proses *coding.*
6. Microsoft Word2016*,* digunakan untuk menyusun laporan.
7. Microsoft Visio2016*,* digunakan untuk membuat rancangan diagram.
8. Mendeley, digunakan untuk membuat sitasi dan daftar pustaka.
9. Postman, digunakan sebagai *tester* terhadap penyedia layanan API.
10. Android Studio, digunakan untuk membangun sistem berbasis *android.*
    * 1. **Bahan**

Adapun sumber data sebagai bahan yang digunakan dalam penelitian Tugas Akhir ini adalah sebagai berikut.

1. Data informasi umum mengenai Kementerian Agama (KEMENAG) Provinsi Nusa Tenggara Barat.
2. Data setiap tempat ibadah yang ada di Kota Mataram beserta deskripsinya.yang didapat dari kemenag NTB.
3. Data *latitude* dan *longitude* tempat ibadah di Kota Mataram.

### Diagram alir penelitian

### Pada penelitian Tugas Akhir ini menggunakan metodologi pengembangan sistem *Extreme* *Programming* yang terdiri dari 4 tahapan yaitu : *planning, design, coding,* dan *testing.* Tahap-tahap pengembangan sistem tersebut diilustrasikan pada diagram alir penelitian pada Gambar 3.1 berikut :

|  |
| --- |
|  |

### Gambar 3.1 Diagram alir penelitian

### Studi Literatur

### Tahap pertama dalam penelitian Tugas Akhir ini adalah studi literatur. Studi literatur dilakukan dengan mengumpulkan data dan memahami teori-teori dari berbagai buku, skripsi, jurnal dan penelitian-penelitian sebelumnya yang berkaitan dengan pembuatan dan perancangan aplikasi pemetaan tempat ibadah pada penelitian Tugas Akhir ini.

### *Planning*

### Pada tahap ini merupakan analisa terhadap kebutuhan dalam pengembangan sistem yang akan dibuat berupa data-data informasi meliputi lokasi tempat ibadah, deskripsi tempat ibadah, garis bujur dan garis lintang tempat ibadah, hingga foto sesungguhnya dari tempat ibadah yang ada di Kota Mataram dan perencanaan untuk mendapatkan gambaran fitur dan fungsi dari perangkat lunak yang akan dibangun. Sistem ini ditujukan untuk masyarakat umum yang ingin mendatangi tempat ibadah yang diinginkan dan pihak Kemenag prov. NTB sebagai admin yang mengelolanya. Sistem pada aplikasi ini dapat memberikan informasi terkait dengan destinasi tempat ibadah seperti lokasi pada peta, rute dari lokasi pengguna ke tempat ibadah. Pada penelitian Tugas Akhir ini, penulis menggunakan beberapa metode dalam pengumpulan kebutuhan yaitu melalui kuesioner, wawancara, mengambil titik koordinat dari *google maps* dan observasi secara langsung.

### Kuesioner

### Pada penelitian Tugas Akhir ini, kuesioner ditujukan kepada masyarakat umum guna untuk mendapatkan respon dari masyarakat terkait dengan penting atau tidak dibangunnya sistem pada penelitian Tugas Akhir ini. Kuesioner ini dilakukan dengan cara memberikan pertanyaan-pertanyaan terkait dengan apa kegunaan tempat ibadah, seberapa pentingnya tempat ibadah untuk memenuhi kewajiban peribadatan, kesulitan apa saja yang dialami masyarakat dalam mencari informasi dan lokasi mengenai tempat ibadah, solusi apa saja yang telah dilakukan masyarakat untuk mengatasi masalah tersebut dan penting atau tidak kah masyarakat jika dibuatnya sistem informasi geografis, serta fitur yang diharapkan oleh *user* terhadap sistem yang akan dibuat.

### Berdasarkan hasil-hasil kuesioner terhadap 61 responden yang telah dipaparkan sebelumnya, dapat disimpulkan bahwa masyarakat setuju untuk dibangunnya sistem pemetaan lokasi tempat ibadah ini, serta fitur yang diharapkan oleh *user* pada sistem yang akan dibuat.

### Wawancara

### Pada wawancara yang dilakukan di kantor Kementerian Agama Prov. NTB dengan Bapak H. Toni selaku *staff* sub. bagian Inmas Kementerian Agama Prov. NTB*,* dijelaskan tentang proses penyimpanan data tempat ibadah yang ada saat ini masih dilakukan secara konvensional yaitu menggunakan Ms. Excel. Dengan cara seperti itu untuk melakukan pengelolaan ataupun pencarian terhadap data tempat ibadah yang ada menjadikan proses nya tidak efektif dan efisien. Penulis menawarkan untuk melakukan penelitian pengembangan sebuah sistem informasi yang dapat membantu pihak kemenag dalam mengelola data tempat ibadah yang ada, tidak hanya itu penulis juga menawarkan untuk dibuatkan sistem pemetaan lokasi tempat ibadah yang ada di Kota Mataram yang dapat dijangkau oleh masyarakat umum dalam memenuhi kewajiban peribadatan. Adapun hasil lain dari wawancara yang telah dilakukan yaitu sebagai berikut :

### Terdapat 5 jenis tempat ibadah yang ada di Kota Mataram, yaitu masjid, pura, gereja, vihara, dan klenteng.

### Mendapatkan data informasi mengenai tempat ibadah yang ada di Kota Mataram seperti nama tempat ibadah, alamat, kecamatan, tahun berdiri dll.

### Pihak kemenag hanya mengelola data mengenai masjid, pura, gereja, vihara, dan klenteng, tidak dengan musholla.

### Mendapatkan *user stories* untuk kebutuhan sistem yang diinginkan oleh pengguna seperti pada tabel 3.1.

### Mengambil datadari *Google maps*

### Pengambilan data dari *google maps* pada penelitian Tugas Akhir ini tidak hanya bertujuan untuk mendapatkan banyaknya tempat ibadah yang ada di Kota Mataram, tetapi juga bertujuan untuk mendapatkan data *latitude* dan *longitude* atau titik koordinat terhadap lokasi masing-masing tempat ibadah. Adapun guna dari pada titik koordinat ini yakni untuk pembuatan *marker* padapeta sistem yang akan dibuat.

### Observasi Langsung

### Karena tidak semua tempat ibadah yang ada di Kota Mataram terdaftar dalam *google maps*, sehingga perlu dilakukan observasi langsung untuk mendapat lebih banyak data mengenai adanya tempat ibadah di Kota Mataram. Observasi dilakukan dengan cara menelusuri atau mencari langsung keberadaan tempat ibadah yang ada di Kota Mataram. Hasil yang didapat dari observasi yang dilakukan yakni terdapat beberapa tempat ibadah salah satu contohnya GKII Kalvari Ampenan yang masih berada dalam wilayah Kota Mataram dan terdapat dalam data kemenag, namun tidak terdaftar dalam *google maps*. Oleh karena itu guna dari pada observasi langsung yang penulis lakukan adalah untuk mendaftarkan tempat ibadah yang tidak terdaftar pada *google maps* ke dalam peta sistem yang akan dibuat, sehingga nantinya dapat menjadi acuan sebagai sistem informasi rujukan untuk mengetahui lokasi tempat ibadah yang valid posisinya.

### *User stories*

### Pada tahap ini pengguna menentukan requirement-nya dan ditulis dalam bentuk *user story*. *User story card* yang baik menerapkan model INVEST yaitu *Independent, Negotiable, Valuable, Estimable,* dan *Small*. *Independent* berarti tidak adanya ketergantungan antar *user story*, *Negotiable* berarti seluruh detil diperoleh melalui kolaborasi, *Valuable* berarti memiliki nilai bagi pengguna, *Estimable* berarti dapat diperkirakan baik dalam segi waktu dan biaya, *Testable* berarti dapat ditentukan *acceptance* *criteria*-nya. Sebuah *user story* dapat dikatakan “selesai” jika *acceptance criteria user story* tersebut telah dipenuhi[30].

### Pembahasan penelitian Tugas Akhir ini dimulai dengan merangkum *user story* dalam bentuk tabel (Tabel 3.1). Terdapat 4 poin dasar *user story* yaitu, nama *user story,* deskripsi tujuan *user story* berupa paragraf pendek, estimasi waktu berapa lama *user story* akan diimplementasikan, dan tingkat kepentingan *user story* (*seperti must have, should have*)[31]. *User story* pada Tabel 3.1 dikelompokkan berdasarkan pengguna aplikasi baik secara umum maupun lebih detail berdasarkan kebutuhan. Pengguna aplikasi adalah masyarakat umum yang tidak perlu login untuk dapat mengakses sistem dan pihak Kemenag prov. NTB admin yang harus login untuk dapat mengakses dan mengelola sistem. Berikut merupakan rangkuman dari *user story* pada sistem yang akan dibuat dapat dilihat pada Tabel 3.1.

### Tabel 3.1 *User story cards*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Kode US | Judul | Deskripsi | *Acceptance Criteria* | Estimasi (hari) |
|  | US-01 | Login | Terdapat 2 jenis sub. Bagian yaitu Bimas Islam yang berperan sebagai admin dan Inmas yang berperan sebagai super admin yang dapat masuk ke dalam sistem. Dan admin dapat *logout* dari system | Terdapat tampilan *login* dimana super admin dan admin diharuskan memasukkan *username* dan *password* yang telah terdaftar dalam sistem untuk dapat mengakses sistem. Sedangkan masyarakat tidak perlu login untuk dapat mengakses sistem. | 4 |
|  | US-02 | Melihat data admin | Sebagai Super Admin, saya dapat melihat daftar admin berupa *list* admin serta deskripsinya. | Setelah super admin melakukan login, terdapat tampilan daftar admin pada menu navbar dan terdapat *list* admin beserta deskripsinya. | 2 |
|  | US-03 | Menambah data admin | Sebagai Super Admin, saya dapat menambah data admin baru beserta deskripsinya. | Setelah super admin melakukan login, pada bagian menu navbar daftar admin, super admin dapat menambah data admin baru beserta deskripsinya dengan cara mengisi *form* untuk tambah data admin baru. | 2 |
|  | US-04 | Mengubah data admin | Sebagai Super Admin, saya dapat mengubah data admin. | Pada bagian *list* admin, super admin dapat mengubah data admin yang dinginkan dengan memasukkan data admin yang baru yang terdapat dalam *form* ubah data admin. | 2 |
|  | US-05 | Menghapus data admin | Sebagai Super Admin, saya dapat menghapus data admin. | Pada bagian *list* admin, super admin dapat menghapus data admin yang dinginkan dengan memilih data admin yang akan dihapus dan kemudian menekan tombol hapus data admin. | 2 |
|  | US-06 | Melihat daftar tempat ibadah | Sebagai Super Admin, Admin, dan masyarakat saya dapat melihat daftar tempat meliputi nama tempat ibadah dan deskripsinya | Setelah super admin dan admin melakukan login, terdapat tampilan daftar tempat ibadah pada menu navbar dan terdapat *list* tempat ibadah serta deskripsinya, dan masyarakat langsung dapat melihat *list* tempat ibadah tanpa perlu login. | 6 |
|  | US-07 | Menambah data tempat ibadah | Sebagai Super Admin, dan Admin, saya dapat menambah data tempat ibadah baru beserta deskripsinya. | Pada bagian *list* tempat ibadah, super admin dan admin dapat mengubah data tempat ibadah yang dinginkan dengan memasukkan data tempat ibadah yang baru yang terdapat dalam *form* ubah data tempat ibadah. | 2 |
|  | US-08 | Mengubah data tempat ibadah | Sebagai Super Admin, dan Admin, saya dapat mengubah data tempat ibadah. | Setelah super admin dan admin melakukan login, pada bagian menu navbar daftar tempat ibadah, super admin dan admin dapat menambah data tempat ibadah baru beserta deskripsinya dengan cara mengisi *form* untuk tambah data tempat ibadah baru. | 2 |
|  | US-09 | Menghapus data tempat ibadah | Sebagai Super Admin, dan Admin, saya dapat menghapus data tempat ibadah. | Setelah super admin dan admin melakukan login, pada bagian menu navbar daftar tempat ibadah, super admin dan admin dapat menghapus data tempat ibadah dengan cara menekan tombol untuk hapus data tempat ibadah. | 2 |
|  | US-10 | Melihat Peta Lokasi tempat ibadah | Masyarakat sebagai pengguna, saya dapat mengakses lokasi terkini | Terdapat tampilan peta dan tombol untuk menentukan lokasi terkini pengguna | 3 |
|  | US-11 | Melihat Rute Lokasi Tempat Ibadah | Masyarakat sebagai pengguna, saya dapat mengakses rute / jalur menuju tempat ibadah | Terdapat tampilan peta dan pilihan untuk menuju lokasi tujuan serta terdapat tombol untuk dapat mengetahui rute dari lokas pengguna menuju lokasi tempat ibadah. | 3 |
| Total *story points* | | | | | 30 |

### Pada penelitian Tugas Akhir ini, direncanakan implementasi program dapat terselesaikan dalam waktu 30 hari aktif kerja, hari aktif kerja disini mencakup hari *weekday*. *Velocity* merupakan waktu yang diperlukan untuk menyelesaikan tiap iterasi. nilai velocity bisa ditentukan dengan cara memperkirakan nilai velocity itu sendiri. Nilai velocity ditentukan sendiri oleh pihak pengembang[32]. Nilai *velocity* yang digunakan pada penelitian Tugas Akhir ini adalah 10.

### *Iteration plan*

### *Release plan* merupakan proses menentukan *user stories* mana yang akan diimplementasikan pada tiap iterasi. *Relase plan* meliputi perencanaan *estimasi story* dan pembagian iterasi untuk setiap *story.* Dalam mencari jumlah iterasi digunakan dengan membagi *total story* dengan *velocity*[31]*.* Penelitian Tugas Akhir ini memiliki *story points* yang berjumlah 30 dan nilai *velocity* adalah 10, maka dapat diketahui jumlah iterasi adalah 30/10 = 3 iterasi. Tiap iterasi pada pembuatan sistem pada penelitian Tugas Akhir ini akan diselesaikan dalam kurun waktu 10 hari. Pembagian iterasi dan alokasi *story* dapat dilihat pada tabel 3.2.

### Tabel 3.2 Pembagian iterasi *story*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Kode US | Deskripsi | Estimasi (hari) |
| *Iteration* – 1 | | | |
|  | US-01 | Login ke dalam system dan mengakhiri sistem | 4 |
|  | US-06 | Melihat daftar dan detail data tempat ibadah | 6 |
| *Velocity* | | | 10 |
| *Iteration* – 2 | | | |
|  | US-02 | Melihat daftar dan detail data admin | 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 | 2 |
| *Velocity* | | | 10 |
| *Iteration* – 3 | | | |
|  | US-08 | Mengubah data tempat ibadah | 2 |
|  | US-09 | Menghapus data tempat ibadah | 2 |
|  | US-10 | Melihat peta lokasi tempat ibadah | 3 |
|  | US-11 | Melihat rute lokasi tempat ibadah | 3 |
| *Velocity* | | | 10 |

### *Design*

### Desain sistem merupakan tahapan perancangan atau penggambaran tentang bagaimana sistem yang dikembangkan akan bekerja. Pada penelitian Tugas Akhir ini, sistem dirancang dengan konsep berorientasi objek menggunakan UML dan *Class Responsibilities Collaboration Card* (CRC-Card) untuk mengetahui interaksi atau hubungan antar objek yang ada pada sistem.

### *Usecase* Diagram

### Berikut merupakan *usecase diagram* dari Sistem Informasi Geografis Pemetaan Tempat Ibadah di Kota Mataram dapat dilihat pada Gambar 3.2:

|  |
| --- |
|  |

### Gambar 3.2 *Usecase* Diagram

### Gambar 3.2 merupakan *usecase* dengan aktor super admin, admin dan masyarakat umum. Admin merupakan pihak kemenag yg berperan sebagai admin yang dapat melakukan aksi pada sistem, terlebih dahulu admin harus melakukan *login* menggunakan akun yang sudah diberikan oleh super admin. Berikut yaitu aksi-aksi yang dapat dilakukan oleh admin:

### Melakukan *login.*

### Melihat daftar tempat ibadah.

### Mencari tempat ibadah.

### Melihat informasi hasil pencarian.

### Melihat informasi lengkap tempat ibadah.

### Mengelola data tempat ibadah.

### Mengakhiri sistem.

### Kemudian masyarakatmerupakan pengguna yang dapat mengakses sistem ini. berikut merupakan aksi-aksi yang dapat dilakukan oleh masyarakat:

### Melihat daftar tempat ibadah

### Mencari tempat ibadah.

### Melihat informasi hasil pencarian

### Melihat informasi lengkap tempat ibadah

### Kemudian pihak kemenag juga merupakan Super adminyang dapat mengakses sistem ini, dimana aksi yang dilakukan sama halnya dengan admin dan *user*, namun super admin dapat mengelola data admin.

### *Class Responsibilities Collaboration Card* (CRC-*Card*)

### CRC-*Card* didapat dari penjabaran hubungan *responsibility* antar *class* di sistem sesuai dengan yang telah digambarkan dalam *usecase* diagram. CRC-*Card* merupakan sebuah kartu indeks yang terbagi menjadi tiga bagian yaitu nama kelas, *responsibility*, dan *collaborator* yang digunakan untuk mendeklarasikan objek-objek yang ada pada kelas dan kelas yang saling berhubungan. Digunakannya CRC-*Card* pada metode XP adalah karena mengacu pada nilai kesederhanaan yang ada pada metode XP itu sendiri[26]. *Spike solution prototype* merupakan skema desain *prototype* berdasarkan CRC *Cards* dari *user stories*. Penjelasan CRC-*Card* pada penelitian Tugas Akhir ini dapat dilihat pada Tabel 3.3 :

### Tabel 3.3 Penjelasan CRC-*Card*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama | *Class* | *Responsibilities* | *Collaborator* |
|  | Tempat ibadah basis web | C Tempat Ibadah | Mengatur jalannya data tempat ibadah yang akan dikelola maupun yang akan ditampilkan pada sistem. | M tempat ibadah,V daftar tempat ibadah |
| M tempat ibadah | Digunakan untuk mengambil dan mengelola data tempat ibadah seperti menambah, mengubah serta menghapus data tempat ibadah dalam *database*. | C tempat ibadah |
| V daftar tempat ibadah | Menampilkan data tempat ibadah yang tersusun dalam bentuk tabel. | C tempat ibadah, M Tempat ibadah |
| M kab kec | Untuk mengambil data kabupaten dan kecamatan | C tempat ibadha |
|  | Admin | C admin | Mengatur jalannya data admin yang akan dikelola maupun yang akan ditampilkan pada sistem. | M admin, V daftar admin |
| M admin | Digunakan untuk mengambil dan mengelola data admin seperti menambah, mengubah serta menghapus data admin dalam *database*. | C admin |
| V daftar admin | Menampilkan data admin yang tersusun dalam bentuk tabel | C admin, M admin |
|  | *Login* | C login | Mengatur jalannya validasi pengguna yang akan mengakses system dan untuk mengakhiri system (*logout*) | M admin,V login |
| M admin | Digunakan untuk validasi kesesuaian *username* dan *password* pengguna yang ada pada *database*. | C login, |
| V login | Menampilkan *form* untuk login | C login |
|  | *Dashboard* | C dashboard | Untuk mengatur jalannya data yang akan ditampilkan pada halaman *dashboard* | M tempat ibadah, V dashboard |
| V dashboard | Menampilkan daftar tempat ibadah dan peta lokasi serta rute tempat ibadah. | C dashboard |
| M tempat ibadah | Digunakan untuk mengambil data tempat dalam *database*. | C dashboard |
| M kab kec | Untuk mengambil data kabupaten dan kecamatan | C tempat ibadha |
|  | *Profile* | C profile | Untuk menampilkan dan mengatur jalannya data *profile* admin yang telah login | M admin,V profile |
| V profile | Untuk menampilkan data admin yang telah login | C profile |
| M profile | Untuk mengambil data admin yang telah *login* | C profile |
|  | Tempat ibadah basis android | C Tempat Ibadah | Berguna sebagai interface yang ada pada API agar dapat terhubung dengan *database* | M tempat ibadah |
| M tempat ibadah | Digunakan untuk mengambil data tempat ibadah dari *database* | C tempat ibadah |
| ListFragment | Menampilkan daftar tempat ibadah dalam bentuk *listview* | XML, adapter, C tempat ibadah |
| MapFragment | Menampilkan peta lokasi tempat ibadah dalam | XML, adapter, C tempat ibadah |

### *Class* diagram

### Berikut merupakan *class diagram* dari Sistem Informasi Geografis Pemetaan Tempat Ibadah di Kota Mataram dapat dilihat pada Gambar 3.3:

|  |
| --- |
|  |

### Gambar 3.3 *Class* diagram pada web

### Pada Gambar 3.3 merupakan *class* diagram pada sistem informasi pemetaan tempat ibadah pada web. Sistem ini menggunakan metode analisa dan perancangan berorientasi objek dan menggunakan pola arsitektur *Model* *View* *Controller* (MVC) di setiap modul, sehingga dilakukan perancangan kelas – kelas yang ada pada sistem yang terdiri dari *view*, *controller* dan *model.* Pada *Controller* terdapat kelas “C\_tempat\_ibadah”, “C\_admin”, dan “C\_login”. Kemudian pada *Model* terdapat beberapa kelas yaitu “M\_tempat\_ibadah”, “M\_admin”, dan “M\_kab\_kec”. Sedangkan *View* ini bertugas untuk menerima dan menampilkan data kepada *user.*

|  |
| --- |
|  |

### Gambar 3.4 *Class* diagram pada *android*

### Pada Gambar 3.4 merupakan class diagram pada aplikasi *android* dimana masyarakat langsung dapat mengakses halaman *dashboard* yang memiliki *class* tempat ibadah untuk melihat daftar tempat ibadah yang dimana tempat ibadah memiliki peta / koordinatnya masing-masing.

### *Activity* Diagram

### Berikut merupakan *usecase diagram* dari Sistem Informasi Geografis Pemetaan Tempat Ibadah di Kota Mataram:

### Proses *login* admin

### Pada sistem pemetaan tempat ibadah ini, admin dapat melakukan aktivitas *login* dengan menggunakan *username* dan *password* yang sudah diberikan. Terlebih dahulu admin mengisi form *login* yang ditampilkan sistem dengan *username* dan *password* yang sudah diberikan tersebut. Kemudian sistem melakukan validasi apakah *username* dan *password* sesuai dengan yang terdapat dalam *database*. Jika sesuai maka admin dapat mengakses halaman admin, namun jika tidak sesuai maka admin harus mengisi form *login* dengan *username* dan *password* yang sesuai. *Activity* diagram proses login admin dapat dilihat pada gambar 3.5 sebagai berikut:

### 

### Gambar 3.5 *Activity* diagram proses *login* admin

### Proses menambah data tempat ibadah

### Pada sistem pemetaan tempat ibadah ini, admin dapat melakukan aktivitas menambah data tempat ibadah. Dimana terlebih dahulu sistem akan menampilkan *list* data tempat ibadah, kemudian admin dapat menambah data dengan mengisi *form* tambah tempat ibadah yang ditampilkan sistem. Sistem melakukan validasi apakah data yang diisi sudah sesuai, jika tidak admin akan diarahkan kembali mengisi form, jika sudah sesuai maka data tempat ibadah tersebut akan masuk ke dalam *list* tempat ibadah dan tersimpan dalam *database*. *Activity* diagram proses menambah data tempat ibadah dapat dilihat pada gambar 3.6 sebagai berikut:

### 

### Gambar 3.6 *Activity* diagram proses menambahdata tempat ibadah

### Proses mengubah data tempat ibadah

### Pada sistem pemetaan tempat ibadah ini, admin dapat melakukan aktivitas mengubah data tempat ibadah. Terlebih dahulu sistem akan menampilkan *list* tempat ibadah, kemudian admin dapat memilih tempat ibadah mana yang datanya ingin diubah, setelah itu sistem akan menampilkan *form* ubah data tempat ibadah. Dalam *form* tersebut admin akan mengisi data tempat ibadah yang baru, setelah dilakukan validasi kesesuaian data maka data tempat ibadah yang baru tersebut dapat disimpan dalam *database*. *Activity* diagram proses mengubah data tempat ibadah dapat dilihat pada gambar 3.7 sebagai berikut:

### 

### Gambar 3.7 *Activity* diagram proses mengubahdata tempat ibadah

### Proses menghapus data tempat ibadah

### Pada sistem pemetaan tempat ibadah ini, admin dapat melakukan aktivitas menghapus data tempat ibadah. Terlebih dahulu sistem akan menampilkan list tempat ibadah, kemudian admin dapat memilih tempat ibadah mana yang datanya ingin dihapus, setelah itu sistem akan menghapus data tempat ibadah tersebut dari dalam *database*. *Activity* diagram proses menghapus data tempat ibadah dapat dilihat pada gambar 3.8 sebagai berikut:

### 

### Gambar 3.8 *Activity* diagram proses menghapusdata tempat ibadah

### Proses melihat daftar tempat ibadah

### Pada sistem pemetaan tempat ibadah ini, *user* dan admin dapat melakukan aktivitas melihat tempat ibadah. Terlebih dahulu sistem akan menampilkan daftar tempat ibadah, kemudian pengguna dapat melihat daftar tempat ibadah serta detail dari tempat ibadah yang ditampilkan oleh sistem. *Activity* diagram proses melihat daftar tempat ibadah dapat dilihat pada gambar 3.9 sebagai berikut:

### 

### Gambar 3.9 *Activity* diagram proses melihat daftar tempat ibadah

### Proses mencari lokasi tempat ibadah

### Pada sistem pemetaan tempat ibadah ini, *user* dapat melakukan aktivitas mencari lokasi tempat ibadah. Terlebih dahulu sistem akan menampilkan peta, kemudian *user* dapat memasukkan *keyword* yang ditampilkan sistem, kemudian sistem melakukan validasi apakah *keyword* sesuai dengan yang terdapat dalam *database*. Jika sesuai maka sistem akan menampilkan hasil pencarian, namun jika tidak sesuai maka user harus mengisi *keyword* yang sesuai. *Activity* diagram proses mencari lokasi tempat ibadah dapat dilihat pada gambar 3.10 sebagai berikut:

### 

### Gambar 3.10 *Activity* diagram proses mencari lokasi tempat ibadah

### Proses melihat jarak / rute lokasi tempat ibadah

### Pada sistem pemetaan tempat ibadah ini, *user* dan admin dapat melakukan aktivitas melihat jarak atau rute lokasi tempat ibadah. Terlebih dahulu sistem akan menampilkan peta, kemudian penggunadapat memilih salah satu lokasi tempat ibadah yang ditampilkan sistem, kemudian sistem akan menampilkan jarak atau rute menuju lokasi tempat ibadah. *Activity* diagram proses melihat jarak/rute lokasi tempat ibadah dapat dilihat pada gambar 3.11 sebagai berikut:

### 

### Gambar 3.11 *Activity* diagram proses melihat jarak lokasi tempat ibadah

### *Sequence* Diagram

### Berikut merupakan *sequence* diagram dari Sistem Informasi Geografis Pemetaan Tempat Ibadah di Kota Mataram:

### Proses *login* admin

|  |
| --- |
|  |

### Gambar 3.12 *Sequence* diagram proses *login* admin

### Pada Gambar 3.12 merupakan *sequence* diagram proses login admin, proses *login* dimulai ketika *admin* memasukkan *username* dan *password* halaman “V\_form\_login”*.* Kemudian hasil *input form* tersebut akan dikirim dengan mengakses fungsi “login()” pada *class* “C\_auth” yang ada di *Controller* untuk diperiksa dan dilakukan validasi apakah *username* dan *password* yang dimasukkan sesuai dengan yang ada pada *database* atau tidak dengan menjalankan fungsi “auth()” yang ada di *Model*. Jika ada dan sesuai dengan data yang ada pada *database*, maka akan diarahkan ke *View* atau halaman “V\_halaman\_admin”, jika tidak sesuai maka akan diarahkan ke *View* atau halaman “V\_form\_login”.

### Proses menambah data tempat ibadah

|  |
| --- |
|  |

### Gambar 3.13 *Sequence* diagram proses menambah data tempat ibadah

### Pada Gambar 3.13 merupakan *sequence* diagram proses menambah data tempat ibadah, proses tambah data tempat ibadahdimulai ketika *admin* memasukkan datatempat ibadahpada *View* atau halaman “V\_form\_data”*.* Kemudian hasil *input form* tersebut akan dikirim dengan mengakses fungsi “tambah()” pada *class* “C\_tempat\_ibadah” yang ada di *Controller*, kemudian dari *class* “C\_tempat\_ibadah” menjalankan fungsi “create()” yang ada di *class* “M\_tempat\_ibadah” pada *Model* yang berfungsi untuk menyimpan data tempat ibadah ke dalam *database*. Setelah data tersimpan admin akan langsung diarahkan ke halaman “V\_tempat\_ibadah”.

### Proses mengubah data tempat ibadah

|  |
| --- |
|  |

### Gambar 3.14 *Sequence* diagram proses mengubah data tempat ibadah

### Pada Gambar 3.14 merupakan *sequence* diagram proses mengubah data tempat ibadah, proses mengubah data tempat ibadahdimulai ketika *admin* memasukkan datatempat ibadahyangbarupada *View* atau halaman “V\_form\_data”*.* Kemudian hasil *input form* tersebut akan dikirim dengan mengakses fungsi “ubah()” pada *class* “C\_tempat\_ibadah” yang ada di *Controller*, kemudian dari *class* “C\_tempat\_ibadah” menjalankan fungsi “update()” yang ada di *class* “M\_tempat\_ibadah” pada *Model* yang berfungsi untuk menyimpan data tempat ibadah yang baru ke dalam *database* dan setelah tersimpan admin akan diarahkan ke halaman “V\_ tempat\_ibadah”.

### Proses menghapus data tempat ibadah

|  |
| --- |
|  |

### Gambar 3.15 *Sequence* diagram proses menghapus data tempat ibadah

### Pada Gambar 3.15 merupakan *sequence* diagram proses menghapus data tempat ibadah, proses menghapus data tempat ibadahdimulai ketika *admin* mengakses fungsi “hapus()” pada *class* “C\_tempat\_ibadah” yang ada di *Controller,* kemudian data akan terhapus dari dalam *database* melalui fungsi “deleted()” pada *class* “M\_tempat\_ibadah” yang ada di *Model.* Setelahdata terhapus, admin akan diarahkan ke halaman “V\_ tempat\_ibadah”.

### Proses melihat lokasi tempat ibadah

|  |
| --- |
|  |

### Gambar 3.16 *Sequence* diagram proses melihat lokasi tempat ibadah

### Pada Gambar 3.16 merupakan *sequence* diagram proses melihat lokasi tempat ibadah, proses melihat lokasi tempat ibadahdimulai ketika *user* pertama kali masuk ke aplikasi setelah menekan tombol “map” dan memanggil fungsi “getPeta()” yang ada di *class* Peta dan “getList()” yang ada di *class* M\_tempat\_ibadah untuk mengambil data detail tempat ibadah beserta *longitude* dan *latitude*, kemudian sistem akan menampilkan peta digital beserta dengan marker atau lokasi tempat ibadah yang ada di Kota Mataram.

### Proses melihat jarak/rute lokasi tempat ibadah

|  |
| --- |
|  |

### Gambar 3.17 *Sequence* diagram proses melihat jarak/rute lokasi tempat ibadah

### Pada Gambar 3.17 merupakan *sequence* diagram proses melihat jarak/rute lokasi tempat ibadah, proses melihat jarak lokasi tempat ibadahdimulai ketika *user* masuk ke aplikasi setelah menekan tombol “map” dan memilih salah satu lokasi tempat ibadah yang ada, sistem akan memanggil fungsi “getRute()” yang ada di *class* Peta dan “getList(id)” yang ada di *class* M\_tempat\_ibadah yang berfungsi untuk mengambil *longitude* dan *latitude* dan sistem akan menampilkan rute menuju lokasi tempat ibadah yang dipilih.

### Proses mencari lokasi tempat ibadah

|  |
| --- |
|  |

### Gambar 3.18 *Sequence* diagram proses mencari lokasi tempat ibadah

### Pada Gambar 3.18 merupakan *sequence* diagram proses mencari lokasi tempat ibadah, proses mencari lokasi tempat ibadahdimulai ketika *user* memasukkan *keyword* dan menekan tombol “search”, kemudian sistem akan memanggil fungsi “getLocataion(keyword)” yang ada di *class* Peta dan “getList(keyword)” yang ada di *class* M\_tempat\_ibadah. Jika terdapat kesesuaian *keyword* yang dimasukkan dengan yang ada di *database* maka sistem akan menampilkan hasil pencarian, jika tidak *user* diarahkan untuk memasukkan *keyword* yang sesuai.

### *Entity Relationship* Diagram

### Berikut merupakan *Entity Relationship* Diagram (ERD) dari Sistem Informasi Geografis Pemetaan Tempat Ibadah di Kota Mataram dapat dilihat pada Gambar 3.19:

|  |
| --- |
|  |

### Gambar 3.19 *Entity Relationship* Diagramsistem

### Berdasarkan ERD pada Gambar 3.19 maka didapatkan tabel-tabel *database* sebagai berikut:

Tabel 3.4 Tabel Admin

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_admin | Int(5) | *Primary Key* |
| 2. | *Username* | Varchar(10) |  |
| 3. | *Password* | Varchar(10) |  |
| 4. | Nama\_admin | Varchar(30) |  |
| 5 | Alamat | Varchar(30) |  |
| 6. | Id\_bidang | int(5) | *Foreign key* |
| 7 | Foto | Varchar(30) |  |

Tabel 3.5 Tabel Bidang

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_bidang | Int(5) | *Primary Key* |
| 2. | Nama\_bidang | Varchar(10) |  |
| 3. | Role | Int(3) |  |

Tabel 3.6 Tabel Tempat Ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_tempat\_ibadah | Int(5) | *Primary Key* |
| 2. | Id\_Jenis | Varchar(5) | *Foreign Key* |
| 3. | Nama\_tempat\_ibadah | Varchar(30) |  |
| 4. | Alamat\_tempat\_ibadah | Varchar(100) |  |
| 5. | Id\_Kecamatan | Varchar(5) | *Foreign Key* |
| 6. | Foto\_tempat\_ibadah | Varchar(100) |  |
| 7. | *Latitude* | Varchar(25) |  |
| 8. | *Longitude* | Varchar(25) |  |
| 9. | Telepon | Varchar(15) |  |
| 10. | Id\_kabupaten | Int(3) | *Foreign Key* |
| 11 | Luas tanah | Varchar(15) |  |
| 12 | Status tanah | Varchar(15) |  |
| 13 | Luas bangunan | Varchar(15) |  |
| 14 | Tahun berdiri | Varchar(15) |  |
| 15 | jamaah | Varchar(15) |  |
| 16 | Imam | Varchar(15) |  |
| 17 | Khatib | Varchar(15) |  |
| 18 | Remaja | Varchar(15) |  |
| 19 | Ketua | Varchar(15) |  |
| 20 | Keterangan | Varchar(15) |  |
| 21 | Binaan majelis | Varchar(15) |  |
| 22 | Kondisi | Varchar(15) |  |

Tabel 3.7 Tabel Kecamatan

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_kecamatan | int(5) | *Primary Key* |
| 2. | Nama\_kecamatan | Varchar(10) |  |
| 3. | Id\_kebupaten | int(5) | *Foreign Key* |

Tabel 3.8 Tabel Jenis Tempat Ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_jenis | int(5) | *Primary Key* |
| 2. | Nama\_jenis | Varchar(15) |  |
| 3. | Mapicon | Varchar(25) |  |

Tabel 3.9 Tabel Kabupaten

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Type | Keterangan |
| 1. | Id\_kabupaten | int(5) | *Primary Key* |
| 2. | Nama\_kabupaten | Varchar(10) |  |
| 3. | Latitude | Varchar(25) |  |
| 4. | Longitude | Varchar(25) |  |

### Desain *interface*

### Setelah melakukan analisis terhadap kebutuhan dan perancangan sistem, tahapan selanjutnya dari penelitian Tugas Akhir ini yaitu merancangan *interface* dari sistemyang akan dibuat. Berikut ini adalah rancangan *interface* dari sistem yang akan dibuat:

### Halaman utama sistem pada web

### 

### Gambar 3.20 Rancangan tampilan halaman utama pada web

### Pada Gambar 3.20 merupakan rancangan halaman utama dalam web. Pada halaman basis web ini pengguna langsung dapat melihat daftar dan detail tempat ibadah yang tersusun dalam bentuk tabel. Dan terdapat menu login untuk admin login dan menu untuk melihat *website* kemenag NTB.

### Halaman *login* admin

### 

### Gambar 3.21 Rancangan tampilan halaman *login*

### Pada Gambar 3.21 merupakan rancangan untuk halaman *login* yang akan ditampilkan pada sistem basis web. Pada halaman ini, jika admin ingin mengakses sistem maka harus mengisi *username* dan *password* yang sesuai dengan yang terdapat dalam *database*.

### Halaman beranda admin

### 

### Gambar 3.22 Rancangan tampilan halaman beranda admin

### Pada Gambar 3.22 merupakan rancangan untuk halaman berandayang akan ditampilkan pada sistem basis web pada saat super admin dan admin berhasil melakukan *login*. Pada halaman ini admin dapat melihat jumlah tempat ibadah yang ada berdasarkan kabupaten.

### Halaman kelola data tempat ibadah

### 

### Gambar 3.23 Rancangan tampilan halaman kelola data tempat ibadah

### Pada Gambar 3.23 merupakan rancangan untuk halaman kelola data tempat ibadah yang akan ditampilkan pada sistem basis web. Pada halaman ini, super admin dan admin dapat melihat data tempat ibadah yang tersusun dalam bentuk tabel, serta dapat mengelola data tersebut seperti menambahkan data baru, mengubah data, dan menghapus data tempat ibadah yang ada, termasuk mengatur titik koordinat / *longitude* dan *latitude* lokasi tempat ibadah yang ada.

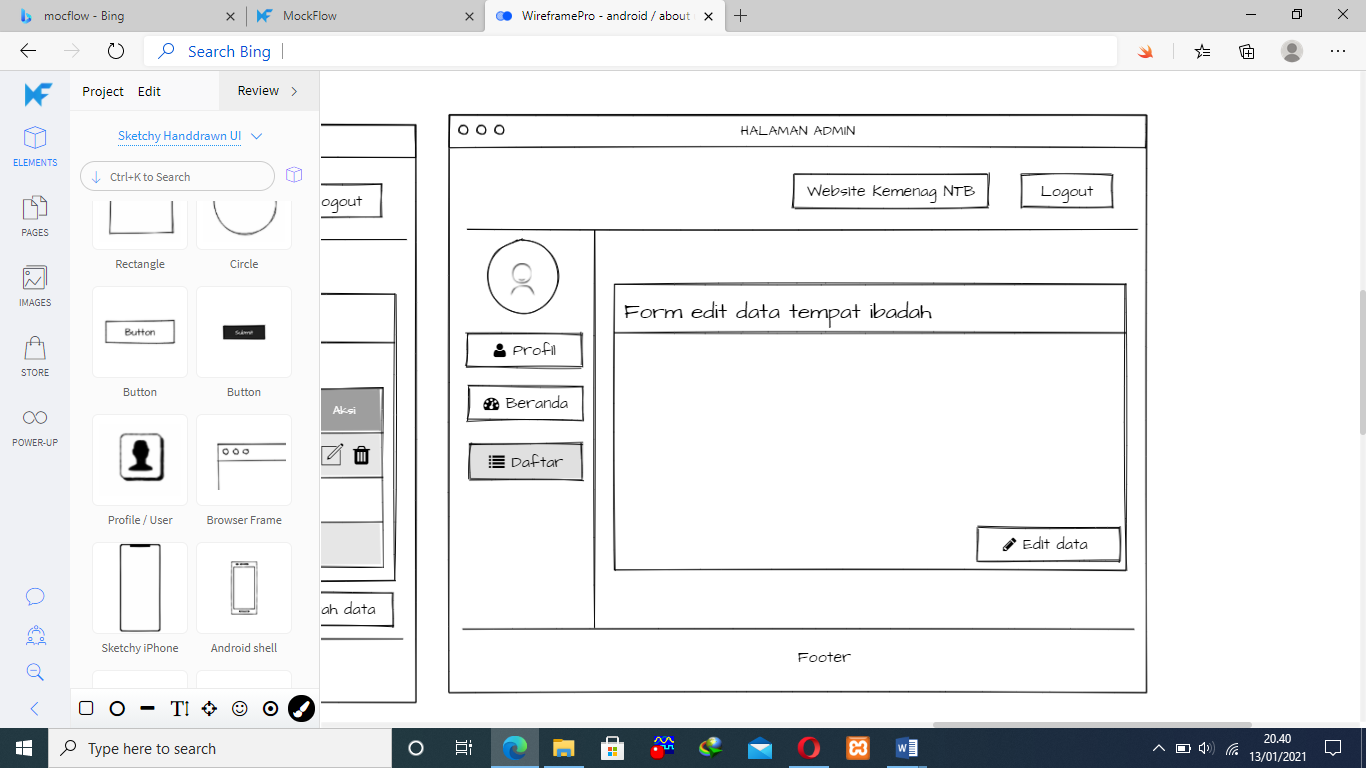
### Halaman tambah data tempat ibadah

### 

### Gambar 3.24 Rancangan tampilan halaman tambah data tempat ibadah

### Pada Gambar 3.24 merupakan rancangan untuk halaman tambah data tempat ibadah yang akan ditampilkan pada sistem basis web. Pada halaman ini, super admin dan admin dapat menambahkan data tempat ibadah ke dalam system.

### Halaman edit data tempat ibadah



### Gambar 3.25 Rancangan tampilan halaman edit data tempat ibadah

Pada Gambar 3.25 merupakan rancangan untuk halaman edit data tempat ibadah yang akan ditampilkan pada sistem basis web. Pada halaman ini, super admin dan admin dapat mengubah data tempat ibadah yang ada di dalam sistem.

### Tampilan hapus data tempat ibadah

### 

### Gambar 3.26 Rancangan tampilan hapus data tempat ibadah

Pada Gambar 3.26 merupakan rancangan untuk tampilan hapus data tempat ibadah yang akan ditampilkan pada sistem basis web. Pada halaman ini, super admin dan admin dapat menghapus data tempat ibadah yang ada di dalam sistem.

### Halaman profil admin

### 

### Gambar 3.27 Rancangan tampilan profil admin

### Pada Gambar 3.27 merupakan rancangan untuk halaman profil admin yang akan ditampilkan pada sistem basis web. Pada halaman ini admin dapat melihat serta dapat mengubah data profil admin itu sendiri.

### Halaman kelola data admin

### 

### Gambar 3.28 Rancangan tampilan halaman kelola data admin

### Pada Gambar 3.28 merupakan rancangan untuk halaman kelola data admin yang akan ditampilkan pada sistem basis web. Pada halaman ini, super admin dapat melihat data admin yang tersusun dalam bentuk tabel, serta dapat mengelola data admin tersebut seperti menambahkan data baru, mengubah data, dan menghapus data admin yang ada.

### Halaman tambah data admin

### 

### Gambar 3.29 Rancangan tampilan halaman tambah data admin

### Pada Gambar 3.29 merupakan rancangan untuk halaman tambah data admin yang akan ditampilkan pada sistem basis web. Pada halaman ini, hanya super admin yang dapat menambahkan data admin ke dalam system.

### Halaman edit data admin

### 

### Gambar 3.30 Rancangan tampilan halaman edit data admin

### Pada Gambar 3.30 merupakan rancangan untuk halaman edit data admin yang akan ditampilkan pada sistem basis web. Pada halaman ini, super admin dapat mengubah data admin ke dalam system.

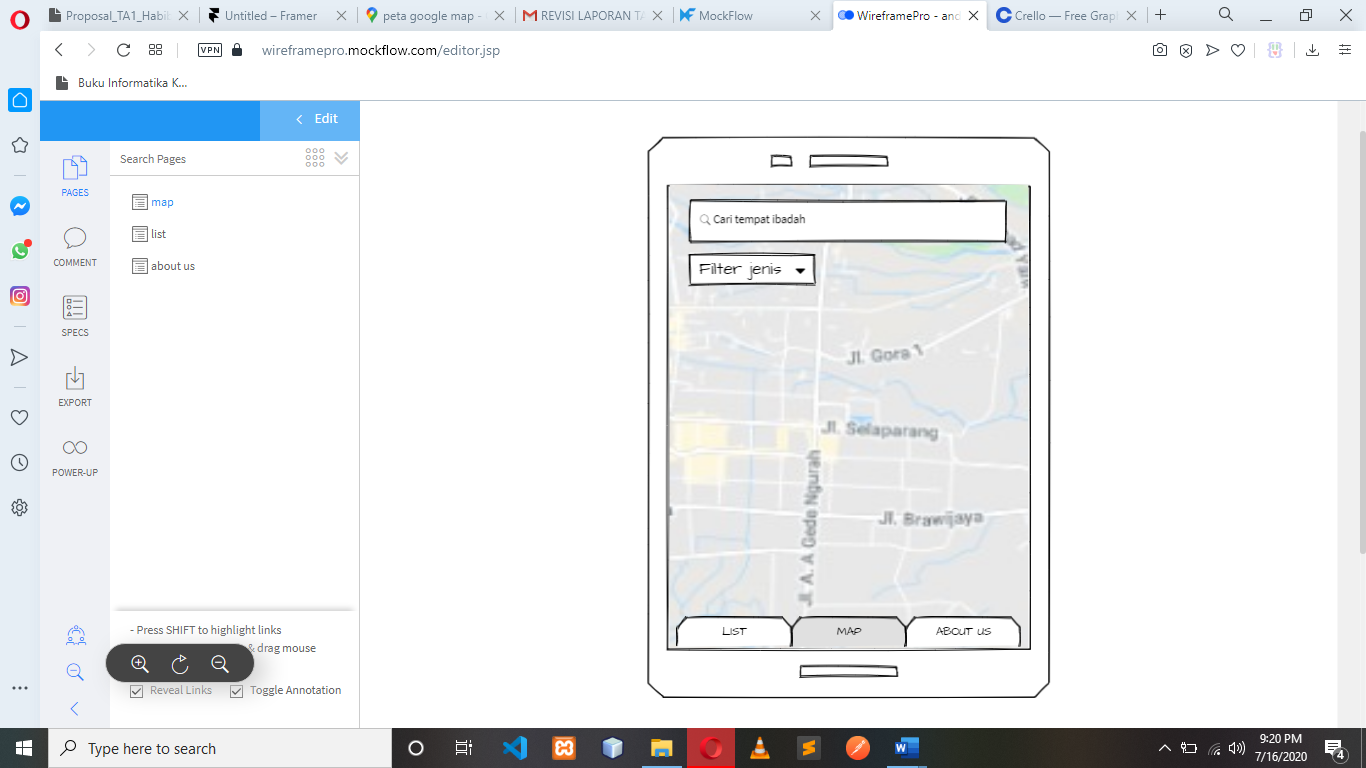
### Tampilan hapus data admin

### 

### Gambar 3.31 Rancangan tampilan hapus data admin

### Pada Gambar 3.31 merupakan rancangan untuk hapus data admin yang akan ditampilkan pada sistem basis web. Pada halaman ini, super admin dapat menghapus data admin yang ada di dalam system.

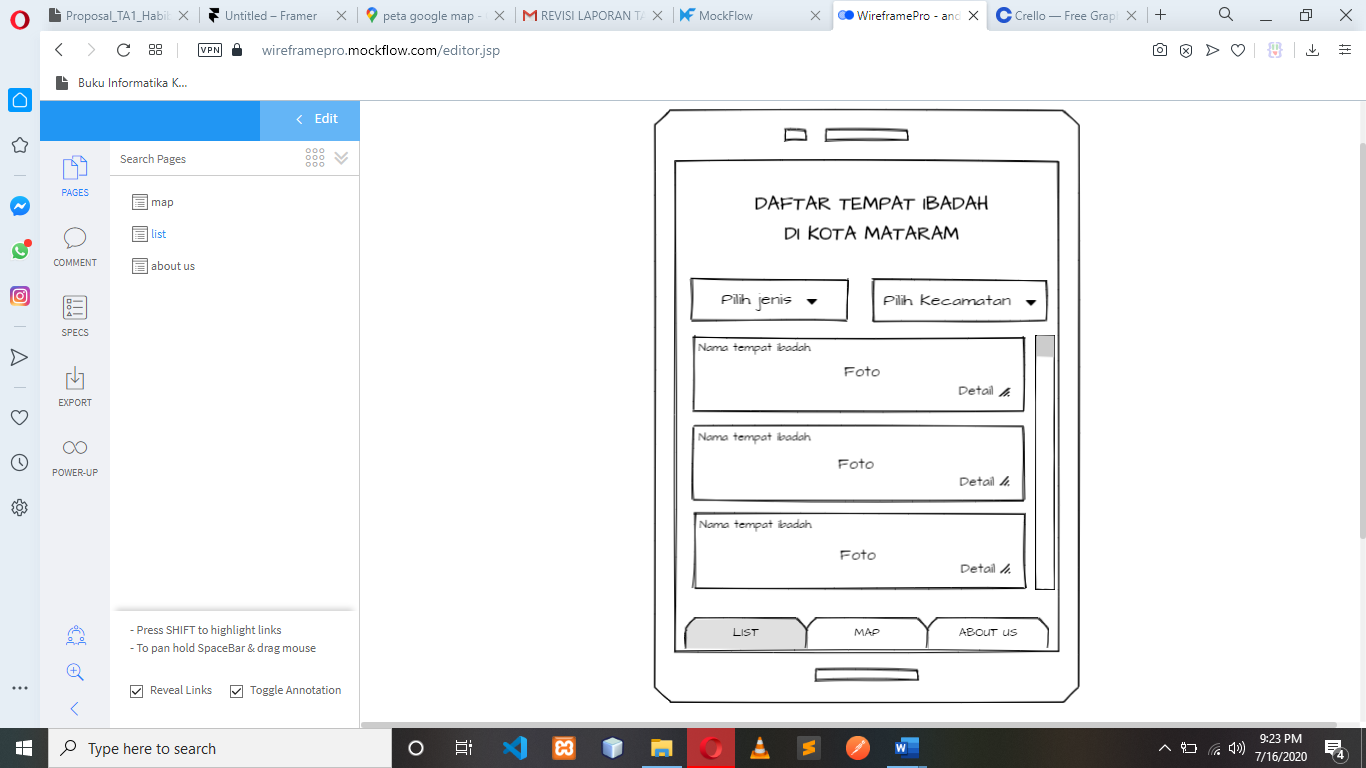
### Rancangan *map android*



**Gambar 3.32** Rancangan tampilan halaman peta tempat ibadah

Pada Gambar 3.32 merupakan rancangan untuk mencari lokasi tempat ibadah yang akan ditampilkan pada sistem informasi geografis pemetaan tempat ibadah. Pada halaman ini, *user* dapat melihat daftar lokasi-lokasi tempat ibadah yang divisualisasikan ke dalam bentuk peta kota Mataram. Pada halaman ini, *user* dapat mencari lokasi tempat ibadah berdasarkan jenis tempat ibadah yang diinginkan serta dapat melihat jarak dan gambar rute atau penunjuk arah jalan dari posisi *user* menuju lokasi tempat ibadah yang diinginkan.

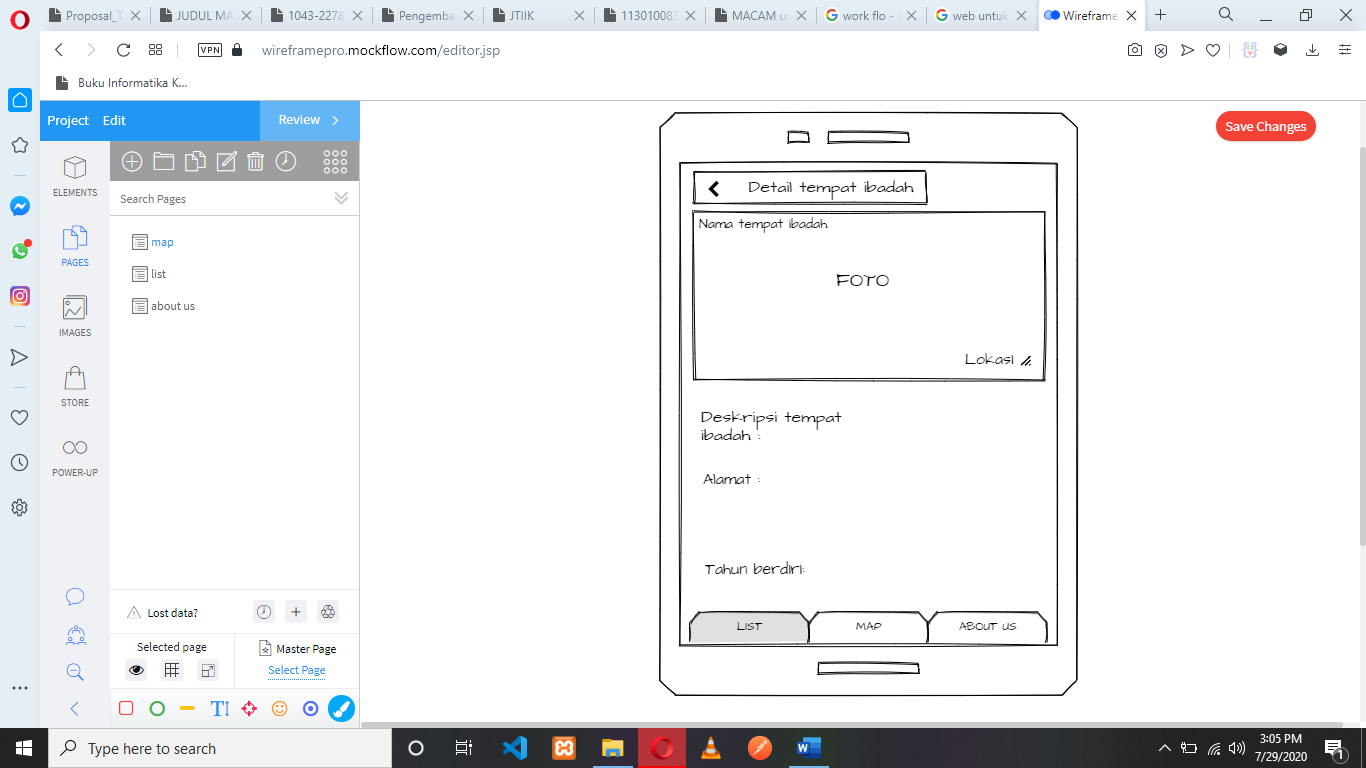
1. Rancangan *list* tempat ibadah *android*



**Gambar 3.33** Rancangan tampilan *list* tempat ibadah *android*

Pada Gambar 3.33 merupakan rancangan untuk menampilkan daftar tempat ibadah yang akan ditampilkan pada sistem informasi geografis pemetaan tempat ibadah. Pada halaman ini, *user* dapat melihat daftar-daftar tempat ibadah yang ada di Kota Mataram berdasarkan filter jenis tempat ibadah dan daerah kecamatan diinginkan.

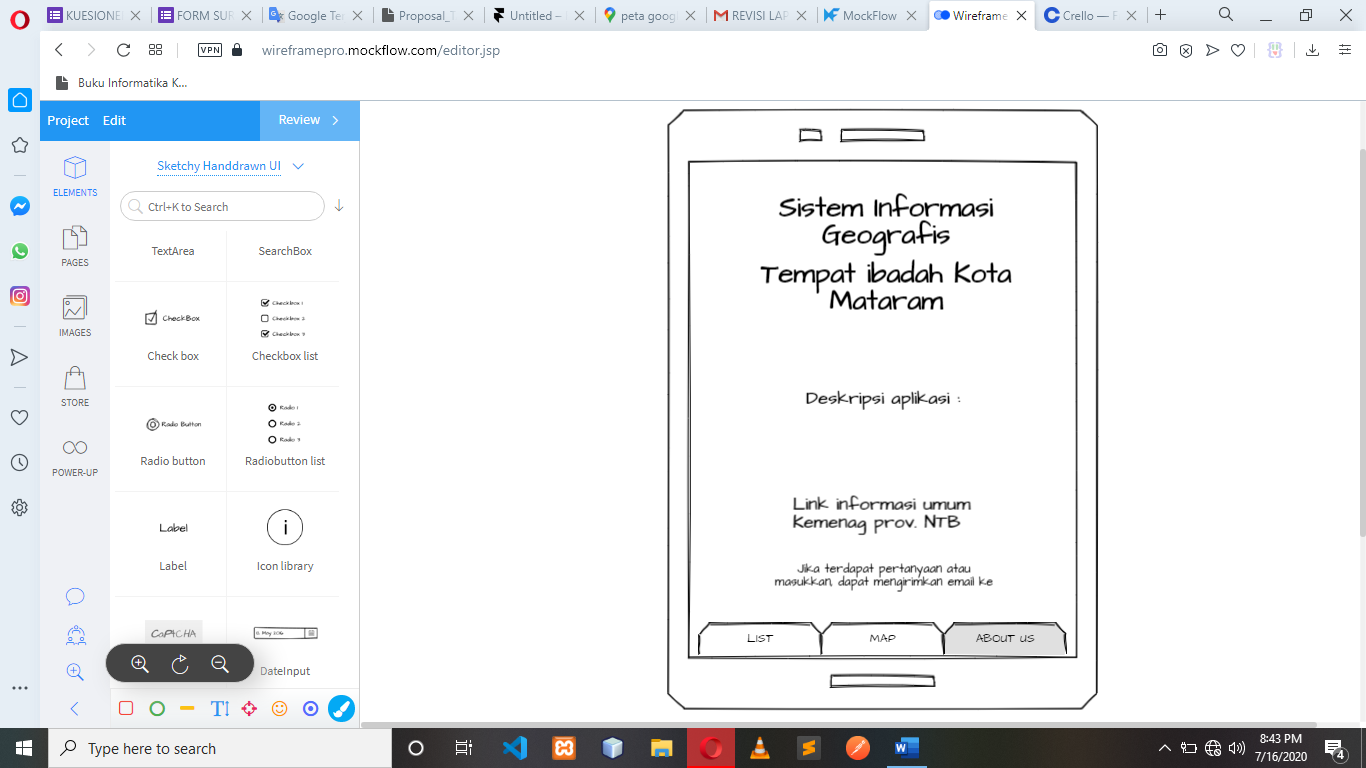
1. Rancangan informasi detail tempat ibadah *android*



**Gambar 3.34** Rancangan tampilan informasi detail tempat ibadah *android*

Pada Gambar 3.34 merupakan rancangan untuk menampilkan informasi detail tempat ibadah yang akan ditampilkan pada sistem informasi geografis pemetaan tempat ibadah. Pada halaman ini, *user* dapat melihat informasi detail dari tempat ibadah seperti nama, foto, alamat, luas tanah dsb.

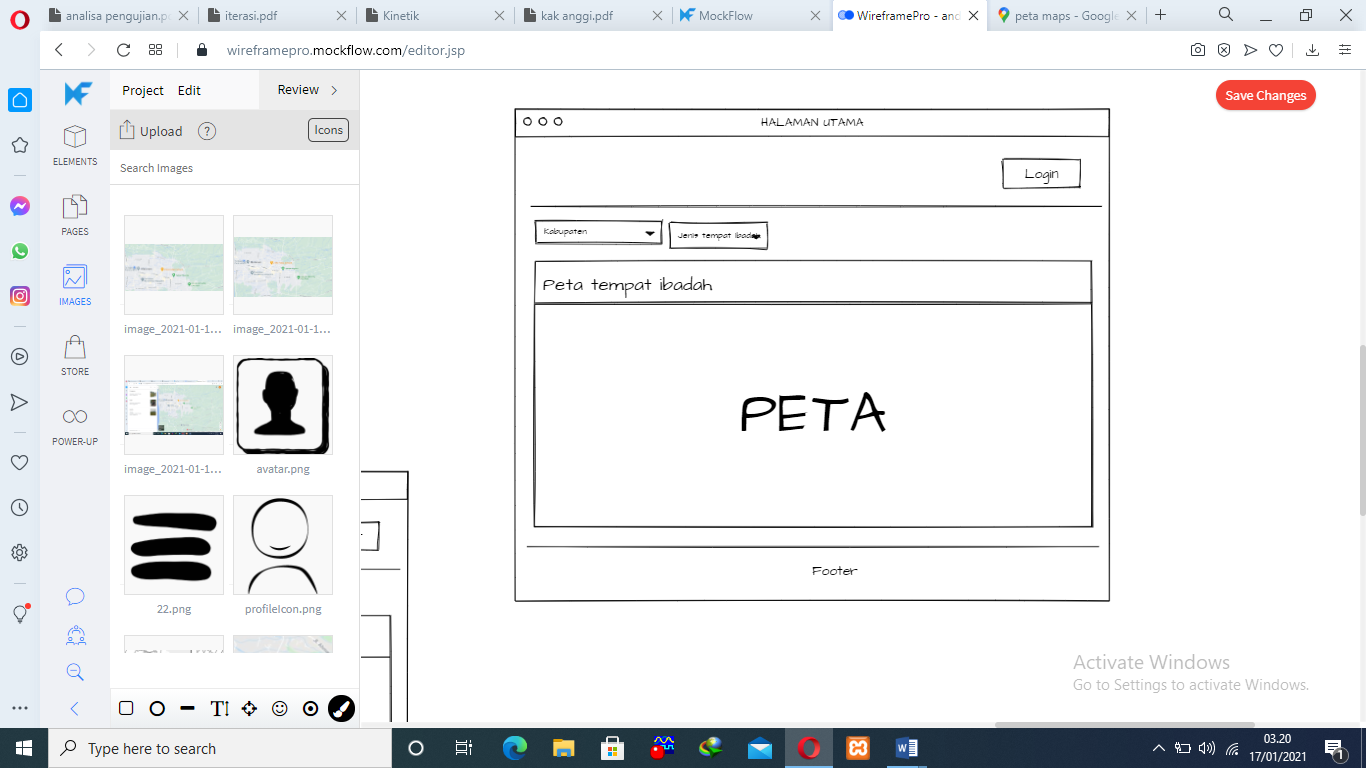
1. Rancangan *about us*



**Gambar 3.35** Rancangan tampilan *about us* *android*

Pada Gambar 3.35 merupakan rancangan untuk menampilkan informasi *about us* yang akan ditampilkan pada sistem informasi geografis pemetaan tempat ibadah. Pada halaman ini, *user* dapat melihat informasi terkait deskripsi dari aplikasi dan dapat menghubungi email jika terdapat pertanyaan atau masukkan terkait dengan sistem, sehingga admin dan *user* dapat saling berinteraksi.

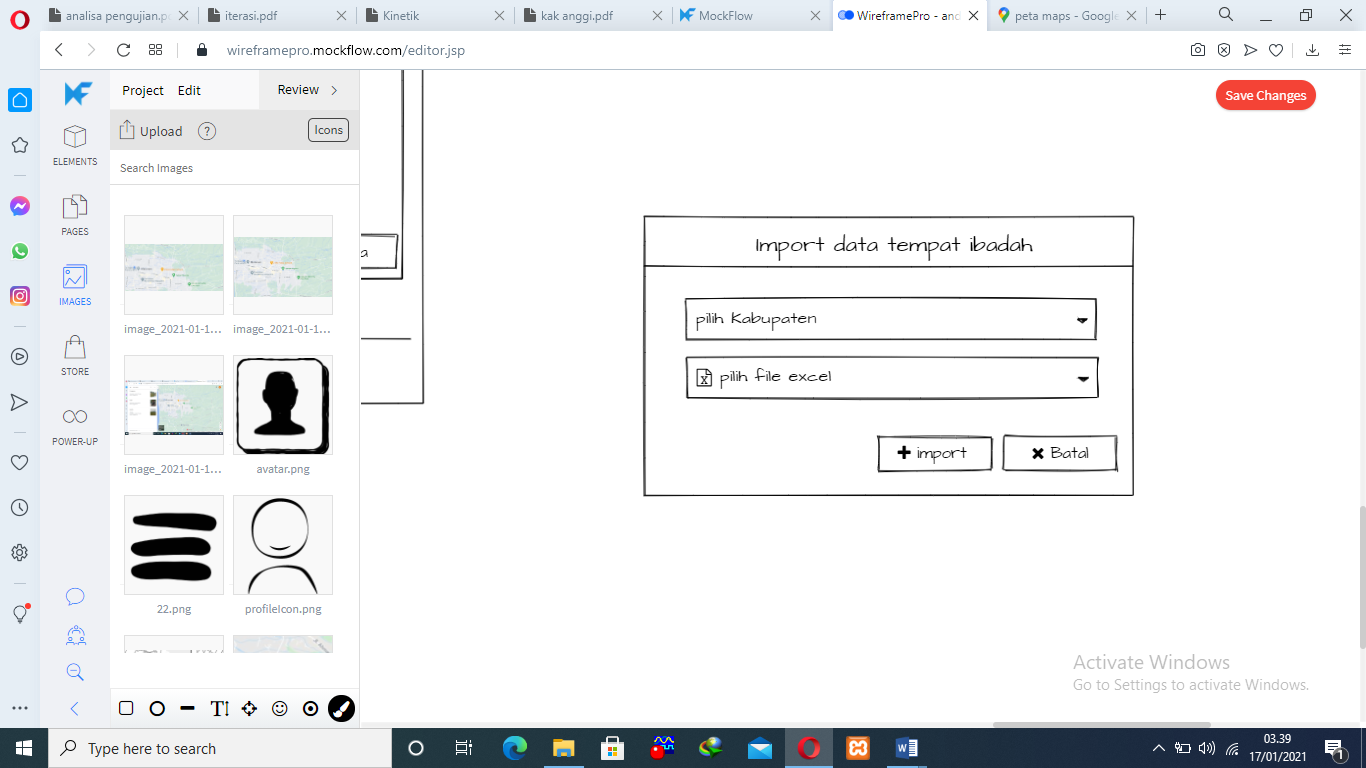
1. Rancangan petabasis *web*



**Gambar 3.36** Rancangan tampilan halaman peta tempat ibadah pada basis web

Pada Gambar 3.36 merupakan rancangan untuk mencari lokasi tempat ibadah yang akan ditampilkan pada sistem informasi geografis pemetaan tempat ibadah basis web. Pada halaman ini, *user* dapat melihat daftar lokasi-lokasi tempat ibadah yang divisualisasikan ke dalam bentuk peta kota Mataram. Pada halaman ini, *user* dapat mencari lokasi tempat ibadah berdasarkan jenis tempat ibadah dan kabupaten yang diinginkan serta dapat melihat jarak dan gambar rute atau penunjuk arah jalan dari posisi *user* menuju lokasi tempat ibadah yang diinginkan.

1. Rancangan *import* data tempat ibadah



**Gambar 3.37** Rancangan tampilan import data tempat ibadah pada basis web

Pada Gambar 3.37 merupakan rancangan untuk *import* data tempat ibadah yang akan ditampilkan pada system basis web. Pada halaman ini, admin dapat memilih kabupaten dan memasukkan file berupa excel yang untuk dapat mengimportnya data ke dalam *database*.

### Desain arsitektur sistem

### Berikut merupakan desain arsitektur sistemdari Sistem Informasi Geografis Pemetaan Tempat Ibadah di Kota Mataram dapat dilihat pada Gambar 3.38:

|  |
| --- |
|  |

### Gambar 3.38 arsitektur sistem

### Pada Gambar 3.38 merupakan desain arsitektur sistem yang akan dikembangkan. Pada basis *android* dapat diakses melalui *multi*-*device,* dimana *device* yang mengakses sistem mengirimkan *request* atau permintaan melalui jaringan internet dan langsung terhubung ke server kemudian akan mengakses database dan mengembalikan *response* berupa html.

### Pada basis *application* hanya dapat diakses melalui *android*, dimana mengirimkan *request* melalui jaringan internet terhadap layanan API yang tersedia, setelah terhubung maka server akan mengakses database dan mengembalikan *response.*

### *Coding*

### Setelah mengetahui gambaran fungsi-fungsi kebutuhan dari sistem yang akan dibuat, tahapan selanjutnya adalah proses implementasi rancangan sistem yang sudah dilakukan ke dalam bentuk bahasa pemrograman (*coding*). Proses pengkodingan pada pembuatan sistem penelitian Tugas Akhir ini dilakukan oleh penulis sendiri karena mengacu pada karakteristik dari Personal Extreme Programming (PXP) yang dimana proses pengkodingan hanya dilakukan oleh programmer tunggal[7]. Pembuatan untuk basis *web* pada sistem penelitian Tugas Akhir ini dibangun dengan bahasa pemrograman PHP menggunakan *framework* *Code Igniter* sebagai *backend* dan *template* *bootsrap* untuk *frontend*. Sedangkan untuk basis *android* dibangun dengan bahasa pemrograman Kotlin. Adapun teknologi tambahan yang digunakan yaitu *google maps* API digunakan sebagai pembuatan peta, MySQL digunakan untuk manajemen *database,* dan Postman digunakan sebagai *tester* terhadap penyedia layanan API.

### Pada fase *coding* dalam XP terdapat pengujian kode program menggunakan teknik *Test Driven Development* (TDD) dimana melakukan pengembangan perangkat lunak berdasarkan hasil pengujian yang telah dilakukan, hal ini bermaksud untuk dapat mengurangi waktu pengembangan dan dapat minimalisir adanya *bug* atau *error* pada aplikasi[26]. TTD memiliki alur kerja seperti Gambar 3.39 sebagai berikut:

|  |
| --- |
|  |

### Gambar 3.39 Alur kerja TDD

### Pengembang menuliskan *unit test* terlebih dahulu, sesuai spesifikasi dan bisnis proses dari sistem yang diinginkan oleh pengguna (*user stories*).

### Setelah membuat *unit test*, pengembang menuliskan kode program untuk menyelesaikan *unit test* sampai *pass* tidak ada *error*.

### Pengembang melakukan *refactoring* kode program yaitu mengubah struktur program agar lebih mudah dipahami dan dimodifikasi, tanpa harus mengubah dari *behaviour* program[33].

### Berikut merupakan contoh dari penggunaan TDD:

### Menentukan *unit test*

### Misal terdapat sebuah contoh kasus seperti login sistem tidak dapat diakses melalui *link*. Dari contoh kasus tersebut kita mendapatkan spesifikasi gambaran tentang nama *function* atau *method*-nya, fungsinya, inputnya dan output yang akan dibuat.

Nama *function* : auth()

Fungsi : validasi untuk dapat mengakses sistem

*Input* : *username* dan *password*

*Output* : Hasil proses

Didapatkan *unit test* seperti inputnya *username* : macan dan *password* : macan hasilnya berhasil login, selain itu termasuk mengakses login sistem melalui *link* maka akan menolak akses.

### Mengimplementasikan *unit test* / *coding*

### Implementasi *code* atau biasa disebut melakukan pengkodingan terdahap *unit test*. Misal berikut implementasi *code* dari *unit test* pada *point* 1

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3. class C\_login extends CI\_Controller {
4. public function \_\_construct(){
5. parent::\_\_construct();
7. }
8. public function auth(){
9. {
10. $ci = get\_instance();
12. $v\_username = $ci->session->userdata('username');
13. if (!$v\_username) {
14. redirect('c\_login');
15. }else {
16. $role\_bidang = $ci->session->userdata('role');
17. $menu = $ci->uri->segment(1);
18. if ($role\_bidang == 2 && ($menu == 'c\_admin' || $menu == 'c\_bidang')) {
19. redirect('c\_login/blocked'); }
20. }
21. }

### *Source code* 3.1 Contoh *source code* *point* 1

### Pada *Source code* 3.1 merupakan *source code* contoh dari penerapan pada *point* 1 yang diasumsikan tidak ada *error.* Jika di tes dengan mengakses login sistem melalui *link*, maka sistem akan menolak akses.

### Pada tahap ini diasumsikan bahwa *code* yang dibuat telah benar dan tidak ada *error / bug*, serta telah memenuhi prasyarat dan siap dilakukan *refactoring*.

### Sebab jika terdapat *error*/*bug* pada code tersebut, maka tidak dapat dilakukann *refactoring*, karena *refactoring* hanya dapat dilakukan jika tidak ada *error* pada *code* [33].

### *Refactoring*

### Pada *Source code* 3.2 dan 3.3 merupakan contoh bentuk *refactoring* dari pada *source code* 3.1

1. <?php
2. function auth(){
3. $ci = get\_instance();
4. $v\_username = $ci->session->userdata('username');
5. if (!$v\_username) {
6. redirect('c\_login');
7. }else {
8. $role\_bidang = $ci->session->userdata('role');
9. $menu = $ci->uri->segment(1);
10. if ($role\_bidang == 2 && ($menu == 'c\_admin' || $menu == 'c\_bidang')) {
11. redirect('c\_login/blocked');
12. }}}

### *Source code* 3.2 yang ada di *file helper*

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3. class C\_tempat\_ibadah extends CI\_Controller {
4. public function \_\_construct(){
5. parent::\_\_construct();
6. auth();}

### *Source code* 3.3 *code* yang dieksekusi

### Diasumsikan bahwa hasil *refactoring* berjalan dengan benar dan hasilnya outputnya sesuai dengan yangdiinginkan yaitu login sistem melalui akan ditolak atau pengguna harus memasukkan *username* dan *password* yang sesuai. Sehingga jika terdapat modul baru, tidak perlu menuliskan ulang *code*, cukup memanggil nama fungsinya saja. *Refactoring* hanya bertujuan untuk merubah struktur *code* supaya lebih singkat dan menghemat memori serta mudah dipahami dan tidak menambahkan fungsionalitas atau tetap sesuai dengan spesifikasi dan tidak merusak fungsi program sebelumnya.

### *Testing*

### Pada tahap ini dilakukan pengujian (*testing*) untuk mengetahui sistem telah berjalan sesuai dengan rancangan atau tidak. Pengujian merupakan kegiatan dimana suatu sistem atau komponen dieksekusi dibawah kondisi tertentu, hasilnya diamati atau dicatat untuk kemudian dievaluasi berdasarkan aspek sistem atau komponen[34]. Dalam tahap *testing* pada XP digunakan *acceptance testing* yang berguna untuk mengevaluasi kesesuaian sistem dengan kebutuhan bisnis dan menilai apakah itu dapat diterima. Dalam *acceptance testing* terdapat beberapa pengujian diantaranya yaitu *Alpha* dan *Beta testing*[35]. Pengujian *alpha* merupakan pengujian yang bertujuan untuk memastikan bahwa aplikasi yang diuji dapat berjalan dengan lancar tanpa ganguan *error* atau *bug*. Pengujian *Beta* merupakan pengujian yang bersifat langsung di lingkungan yang sebenarnya dengan penyebaran kuesioner yang akan dihitung untuk dapat diambil kesimpulan terhadap penilaian aplikasi yang dibangun.

### Penekanan pada tahap ini adalah tentang pengujian, pengujian yang dilakukan menggunakan *alpha testing* dengan metode *black box* dan *beta testing* dengan penyebaran kuesioner, dimana pengguna akan menilai sejauh mana aplikasi berjalan sesuai dengan tujuan[34]. Digunakan pengujian *alpha* dan *beta* karena dirasa cocok atau sesuai dengan tujuan untuk mengetahui sistem yang dikembangkan telah sesuai atau tidak dengan kebutuhan *client*.

### Pada Gambar 3.21 merupakan rancangan halamanlogin sistem dimana terdapat beberapa rencana pengujian. Pada rencana pengujian *username* dan *password*, data akan valid jika *username* atau *password* diisi dengan benar dan tidak boleh kosong. Sebaliknya, data tidak akan valid jika diisi dengan tidak benar atau kosong. Berikut merupakan rencana pengujian pada login:

### Tabel 3.10 Rencana pengujian pada login

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Skenario pengujian | *Test case* | Hasil yang diharapkan |
| L01 | *Username* dan *password* tidak diisi kemudian klik tombol login | Username :(kosong)Password:(kosong) | Sistem akan menolak akses dan menampilkan “Form *username* dan *password* harus diisi!”. |
| L02 | Mengetik *username* dan *password* tidak diisi atau kosong kemudian klik tombol login | Username :(macan)Password:(kosong) | Sistem akan menolak akses dan menampilkan “Form *password* harus diisi!”. |
| L03 | *Username* tidak diisi dan *password* diisi kemudian klik tombol login | Username :(kosong)Password:(123) | Sistem akan menolak akses dan menampilkan “Form *username* harus diisi!”. |
| L04 | *Username* dan *password* diisi tidak benar kemudian klik tombol login | Username :(salah)Password:(salah) | Sistem akan menolak akses dan menampilkan “*Username* dan *password* salah!”. |
| L05 | *Username* dan *password* diisi dengan benar kemudian klik tombol login | Username :(benar)Password:(benar) | Sistem menerima akses login dan kemudian langsung menampilkan halaman admin |

### Pada Gambar 3.28 merupakan rancangan halaman kelola data admin sistem dimana terdapat beberapa rencana pengujian menambah data admin. Pada rencana pengujian *username* atau *password*, data akan valid jika *username* atau *password* diisi atau tidak boleh kosong. Sebaliknya, data tidak valid jika data kosong. Pada rencangan pengujian menambah *username* baru juga data akan berhasil ditambahkan jika *username* belum terdaftar dalam sistem, jika sudah terdaftar tidak berhasil ditambahkan. Rencana pengujian ini juga berlaku terhadap mengubah data admin. Berikut merupakan rencana pengujian pada penambahan data admin:

### Tabel 3.11 Rencana pengujian tambah data admin

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Skenario pengujian | *Test case* | Hasil yang diharapkan |
| T01 | *Username* dan *password* tidak diisi kemudian klik tombol tambah | Username :(kosong)Password:(kosong) | Sistem gagal menambahkan data admin dan menampilkan “Form *username* dan *password* harus diisi!”. |
| T02 | Mengisi *username* dan *password* tidak diisi atau kosong kemudian klik tombol tambah | Username :(macan)Password:(kosong) | Sistem gagal menambahkan data admin dan menampilkan “Form *password* harus diisi!”. |
| T03 | *Username* tidak diisi dan *password* diisi kemudian klik tombol tambah | Username :(kosong)Password:(123) | Sistem gagal menambahkan data admin dan menampilkan “Form *username* harus diisi!”. |
| T04 | *Username* yang diisi sudah terdaftar dalam *database* | Username :(terisi = database) | Sistem gagal menambahkan data admin dan menampilkan “*Username* telah terdaftar!”. |
| T05 | *Username* dan *password* diisi dengan benar kemudian klik tombol tambah, dan *username* belum terdaftar dalam *database* | Username :(terisi)Password:(terisi) | Sistem berhasil menambahkan data admin dan kemudian langsung menampilkan daftar admin yang ada. |

### Digunakan 30 responden dalam melakukan kuesioner pada penelitian Tugas Akhir ini, sebagaimana dikemukakan oleh Cohen, semakin besar *sample* dari besarya populasi yang ada adalah semakin baik, akan tetapi ada jumlah batas minimal yang harus diambil oleh peneliti yaitu sebanyak 30 *sample* [36]. Penulis membagi responden menjadi masyarakat 28 orang dan 2 orang pegawai Kemenag bidang Inmas dan Bimas Islam yang berperan sebagai admin sistem yang akan menggunakannya. Tujuan dari kuesioner ini adalah menilai apakah aplikasi berjalan sesuai dengan tujuan.

### Berikut merupakan tabel skor penilaian dapat dilihat pada Tabel 3.12 berikut:

### Tabel 3.12 Skor penilaian

|  |  |
| --- | --- |
| Tingkat kepuasan | Skala |
| Sangat setuju | 5 |
| Setuju | 4 |
| Cukup | 3 |
| Tidak setuju | 2 |
| Sangat tidak setuju | 1 |

### Untuk melakukan perhitungan kuesioner digunakan rumus:

### Persentase = J / N \*100 %

### Keterangan :

J : Total nilai jawaban yang diberikan pengguna pada setiap nomor.

### N : Jumlah pengguna

Selanjutnya dilakukan proses perhitungan persentase rata-rata dari setiap poin jawaban, perhitungan yang digunakan adalah menggunakan rumus sebagai berikut:

|  |
| --- |
| Rata-rata = (Persentase ke-1 + ….. + Persentase ke-6) / 6 |

### *Release*

Langkah terakhir yang dilakukan setelah dilakukannya pengujian sistem yaitu penerapan sistem dalam kehidupan sehari-hari (*release*). Sistem dapat dikatakan *release* jika sistem sudah berjalan atau sesuai dengan kebutuhan pengguna berdasarkan dari penjabaran *user stories* yang didapat pada tabel (Tabel 3.1). Namun, aplikasi tetap harus ada dalam pendampingan jika suatu saat terdapat *bug* atau pembaharuan terhadap sistem.

## BAB IV

## HASIL DAN PEMBAHASAN

1. **Implementasi siklus metode *Extreme Programming***

Dalam proses pengembangan sistem informasi geografis untuk pemetaan lokasi tempat ibadah ini digunakan siklus metode *extreme programming* seperti yang telah dipaparkan sebelumnya pada diagram alir penelitian pada gambar 3.1, guna untuk mengetahui pengembangan sistem mencapai hasil yang diinginkan oleh *client* atau tidak. Pembahasan yang akan dijelaskan pada sub bab ini sama halnya dengan siklus metode *extreme programming* yaitu meliputi *planning, design, coding* dan *testing.*

1. ***Planning***

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

Pada tahap ini juga pengembang dan *client* telah melakukan diskusi terhadap penentuan iterasi yang terdiri dari *user stories* yang akan dikerjakan terlebih dahulu. Dimana guna penentuan iterasi ini yaitu untuk mengetahui prioritas *user stories* dan estimasi waktu pengerjaan. 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.

Selama proses iterasi berlangsung terdapat beberapa permintaan penambahan user stories dari *client* seperti yang dirincikan pada tabel 4.1 berikut:.

Tabel 4.1 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. | 4 |
|  | 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 | 2 |
|  | 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. | 8 |
|  | 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 | 2 |

Penambahan *user tories* pada tabel 4.1 tersebut menyebabkan pengembang bersama dengan *client* melakukan perencanaan ulang terhadap penyesuaian pengerjaan tiap itetari. Hasil perencanaan iterasi dapat dilihat pada Tabel 4.2 sebagai berikut.

Tabel 4.2 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 |

Terdapat 3 iterasi tambahan dan dimana tiap iterasi dikerjakan selama 10 hari, sehingga rencana waktu pengerjaan system bertambah menjadi 60 hari.

1. ***Design***

Setelah dilakukannya perencanaan terhadap *user stories* yang ada, selanjutnya pada tahap ini akan dibahas mengenai hasil *design* atau perancangan terdahap *user stories* yang didapat. Rancangan yang dibuat pengembangan merupakan CRC *card* seperti yang telah dijabarkan pada tabel 3.3 dan *Spike Prototype Solution* merupakan skemadesain *prototype* berdasarkan CRC *card.* Berikut merupakan hasil pembuatan *design* dapat dilihat pada tabel 4.3

Tabel 4.3 Hasil design CRC *card* dan *Spike Solution Prototype*

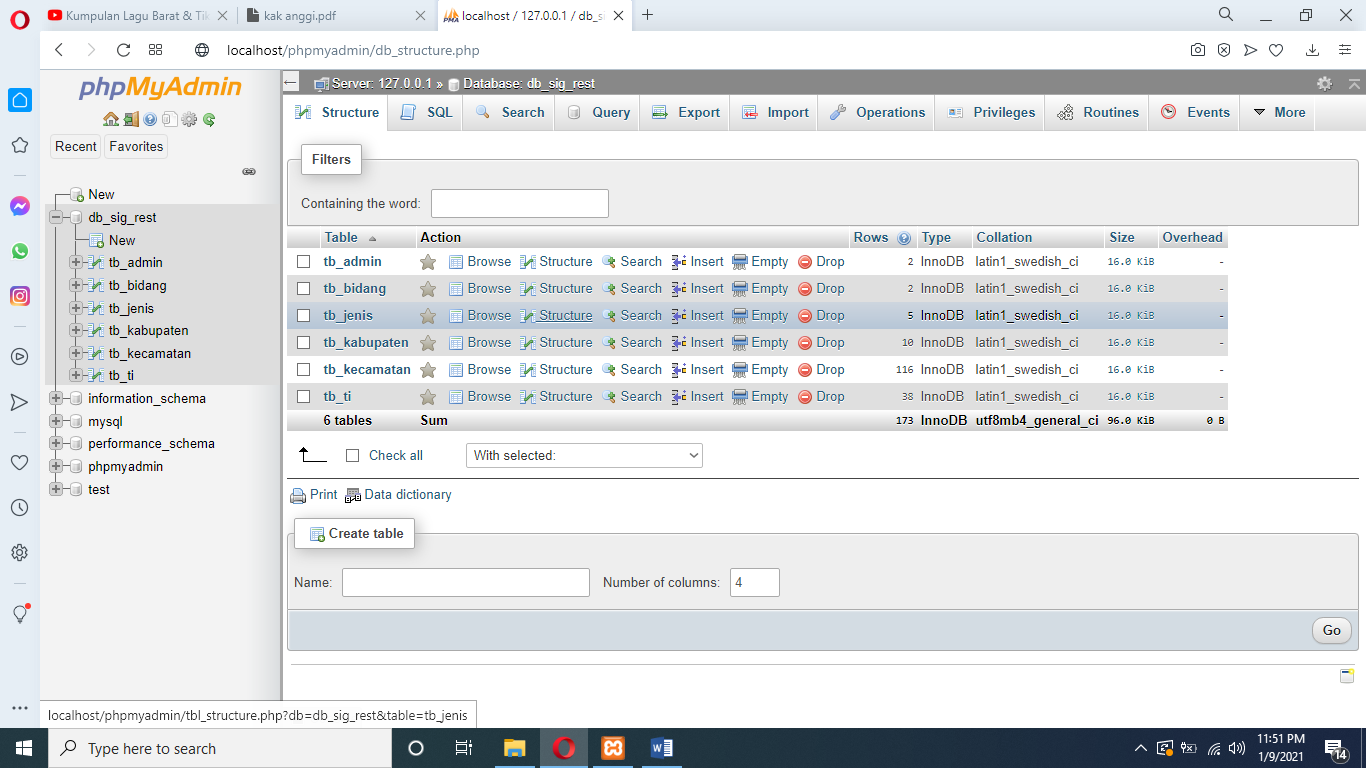
|  |  |  |  |
| --- | --- | --- | --- |
| *Iteration* | Kode US | Nama CRC *Card* | *Spike Solution Prototype*Ditujukan gambar |
| 1 | US-01 | Login | Gambar 3.21 |
| US-12 | Profile | Gambar 3.27 |
| US-02 | Admin | Gambar 3.28 |
| 2 | US-03 | Admin | Gambar 3.29 |
| US-04 | Admin | Gambar 3.30 |
| US-05 | Admin | Gambar 3.31 |
| US-07 | Tempat ibadah basis web | Gambar 3.24 |
| 3 | US-14 | Tempat ibadah basis web | Gambar 3.37 |
| US-15 | Tempat ibadah basis web | - |
| US-16 | Tempat ibadah basis web | - |
| 4 | US-06 | Dashboar, tempat ibadah basis *web* dan *android* | Gambar 3.20, 3.23, dan 3.33 |
| US-13 | Tempat ibadah basis web | Gambar 3.22 |
| US-08 | Tempat ibadah basis web | Gambar 3.25 |
| 5 | US-09 | Tempat ibadah basis *web* | Gambar 3.26 |
| US-10 | Tempat ibadah basis *android* | Gambar 3.32 |
| US-11 | Tempat ibadah basis *android* | Gambar 3.32 |
| 6 | US-17 | Dashboard | Gambar 3.36 |
| US-18 | Dashboard | Gambar 3.36 |

1. ***Coding***

Pada sub-bab ini, akan dibahas mengenai hasil implementasi (*coding*) SIG Pemetaan Lokasi Tempat Ibadah. Implementasi yang dilakukan telah dibuat sesuai dengan perancangan yang dijabarkan pada tabel DESIGN*.* Selain itu, pada bab ini juga akan dibahas mengenai hasil pengujian dari sistem yang telah dibangun berdasarkan dengan *user stories* yang didapat. Pembahasan yang akan dijelaskan pada sub bab ini meliputi implementasi *database*, implementasi API, implementasi *class*, dan implementasi *interface*

1. **Impelementasi *database***

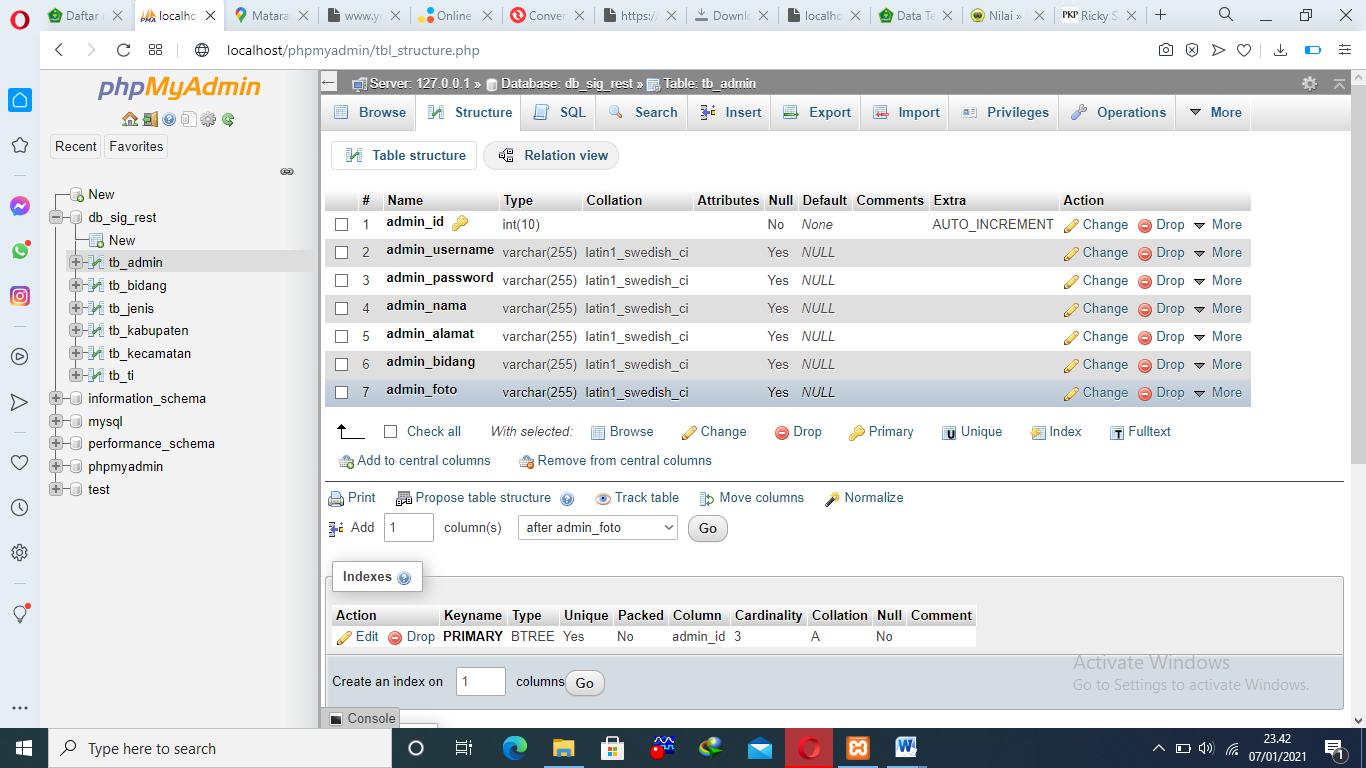
Berikut merupakan implementasi *database* dari Sistem Informasi Geografis Untuk Pemetaan Lokasi Tempat Ibadah dapat dilihat pada gambar 4.1



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 merupakan struktur dari masing masing tabel:

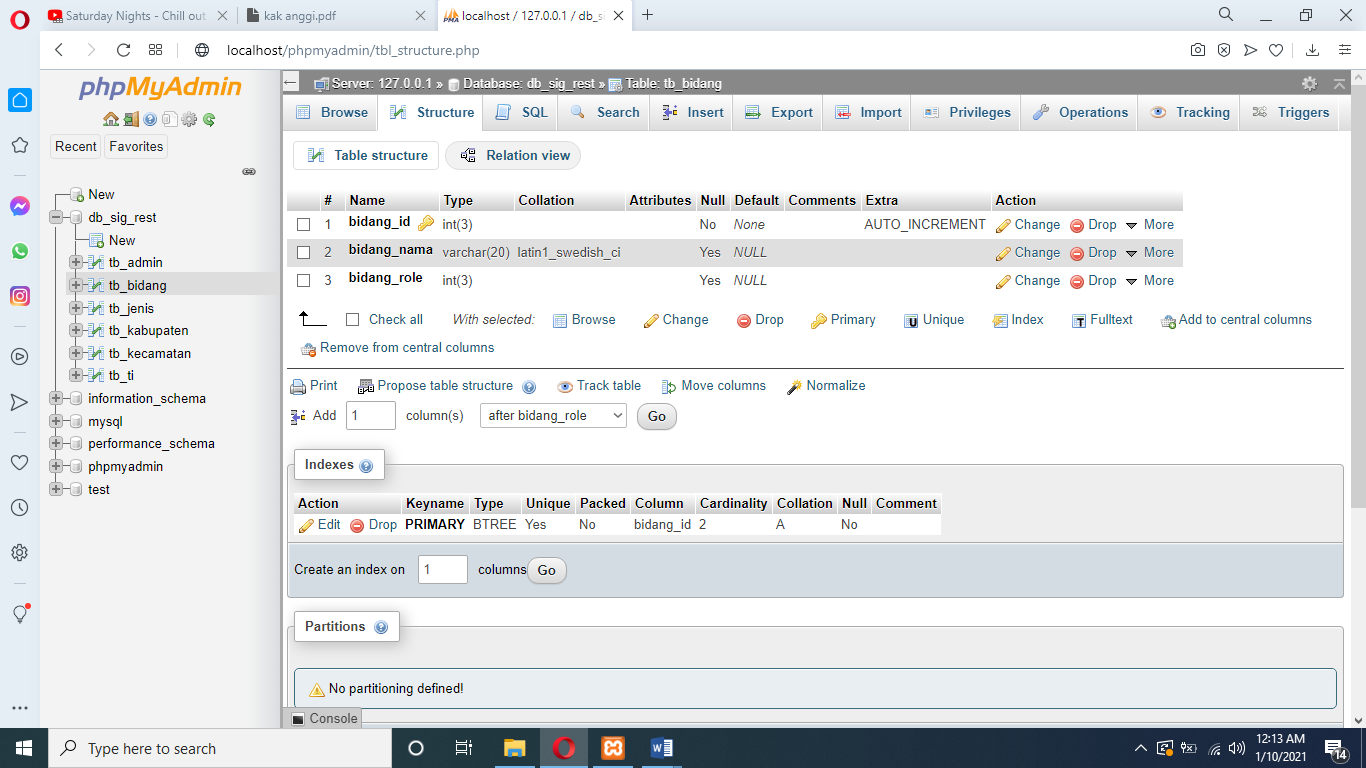
1. Tabel tb\_admin



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.

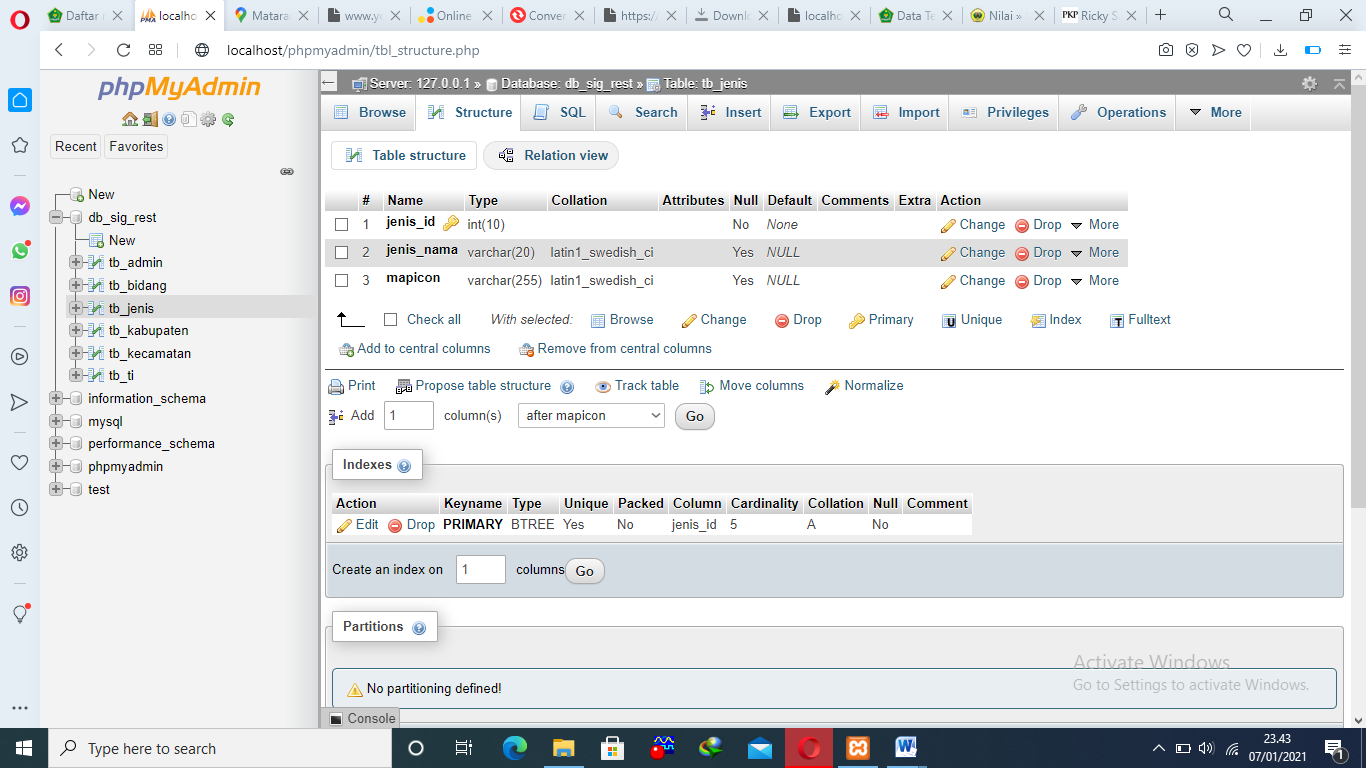
1. Tabel tb\_bidang



Gambar 4.3 Implementasi tabel tb\_bidang

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

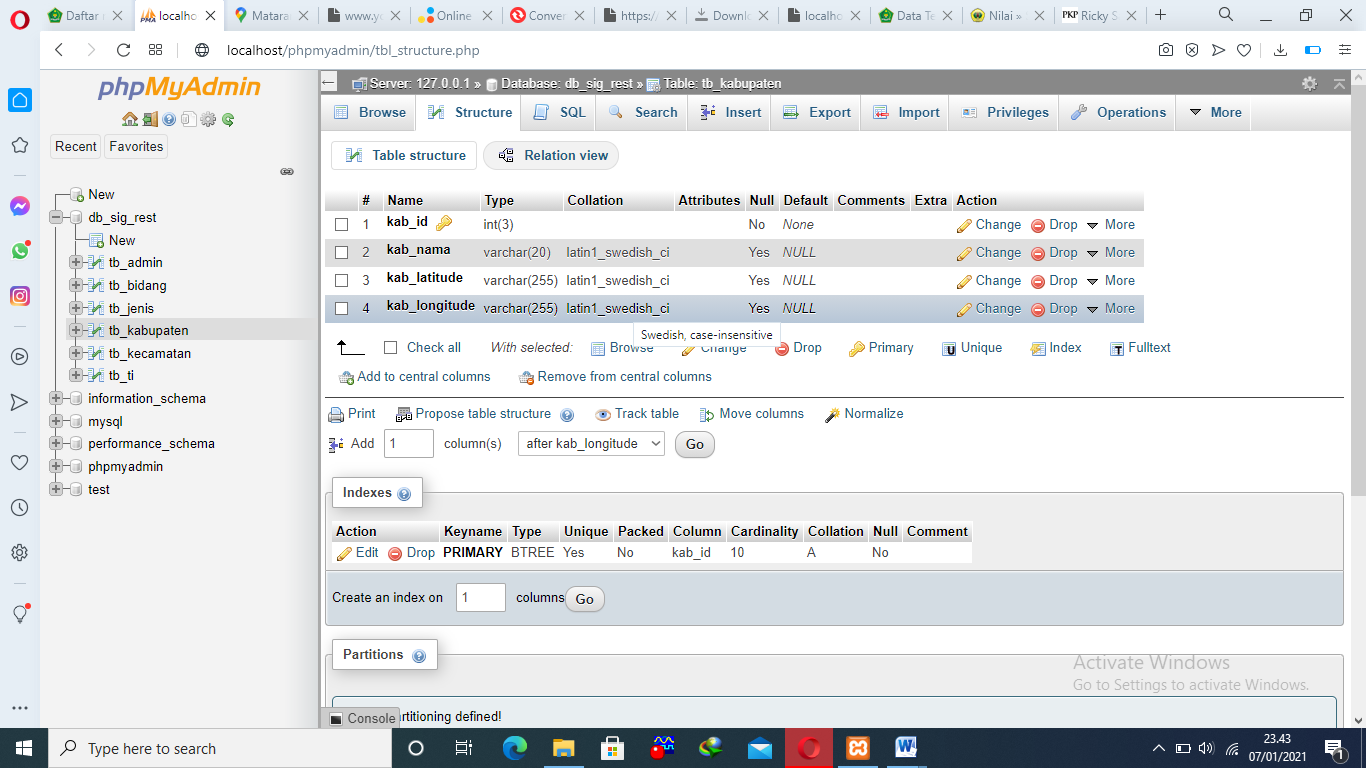
1. Tabel tb\_jenis



Gambar 4.4 Implementasi tabel tb\_jenis

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

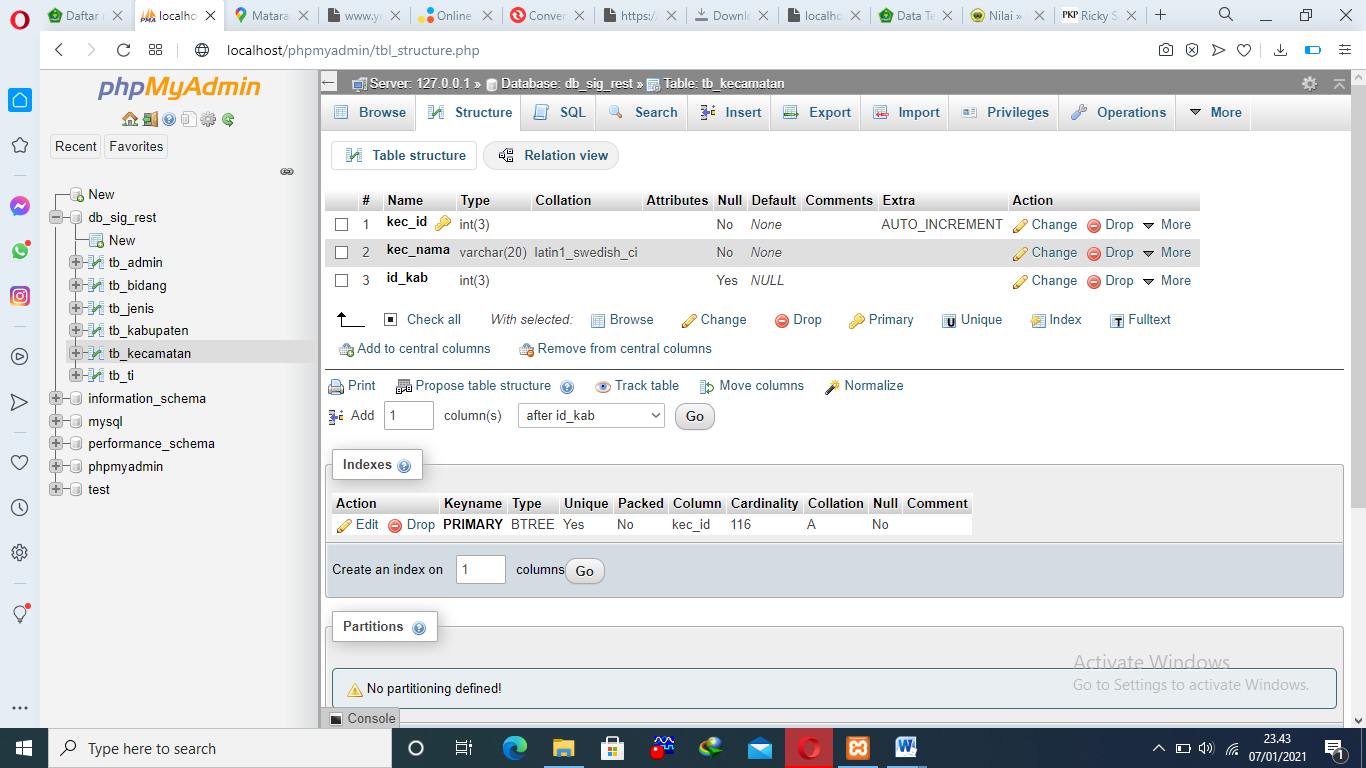
1. Tabel tb\_kabupaten



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.

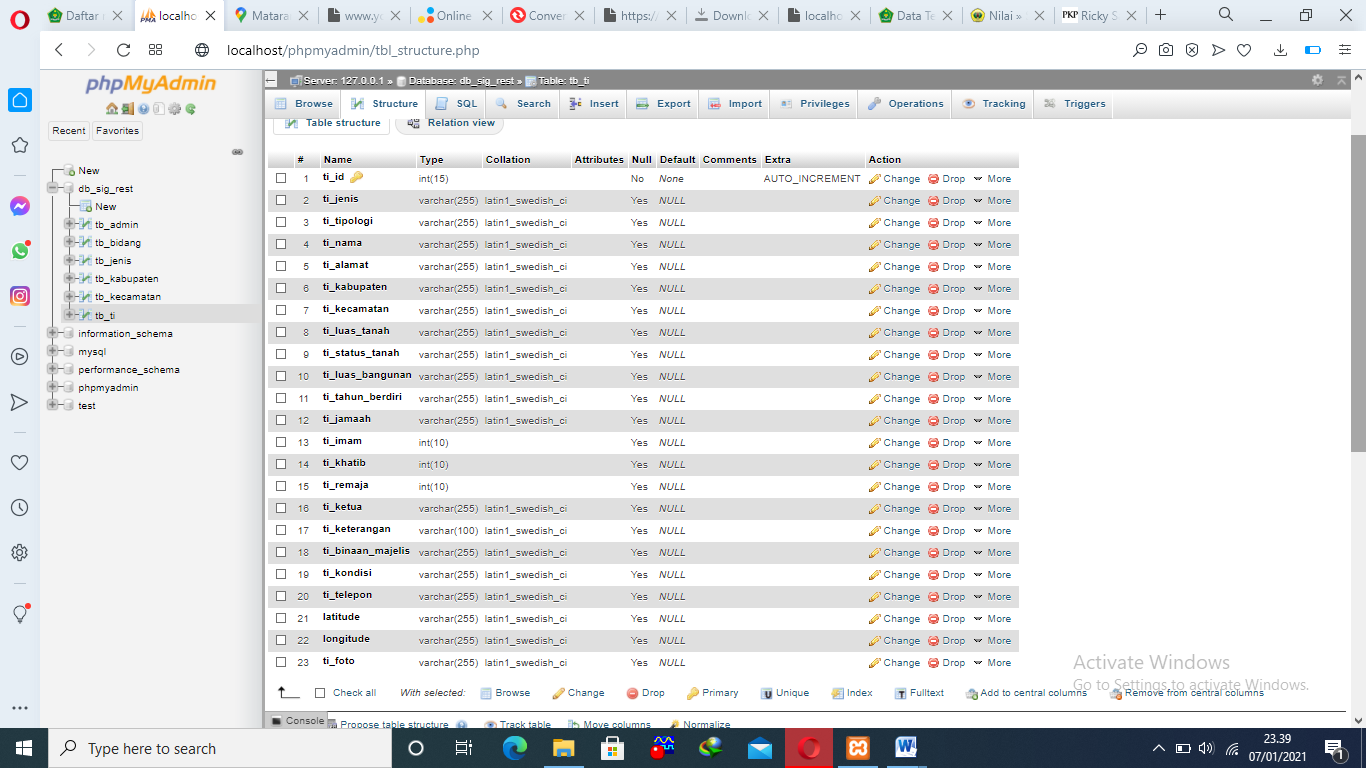
1. Tabel tb\_kecamatan



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  = $this->M\_tempat\_ibadah->selectAlltempatIbadah();
14. }
15. if  ($v\_tempat\_ibadah)  {
16. $this->response([
17. 'status' => true,
18. 'data' => $v\_tempat\_ibadah
19. ], REST\_Controller::HTTP\_OK);
20. }else {
21. $this->response([
22. 'status' => false,
23. 'respone' => 'id is not found'
24. ], REST\_Controller::HTTP\_NOT\_FOUND);
25. }
26. }

*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*

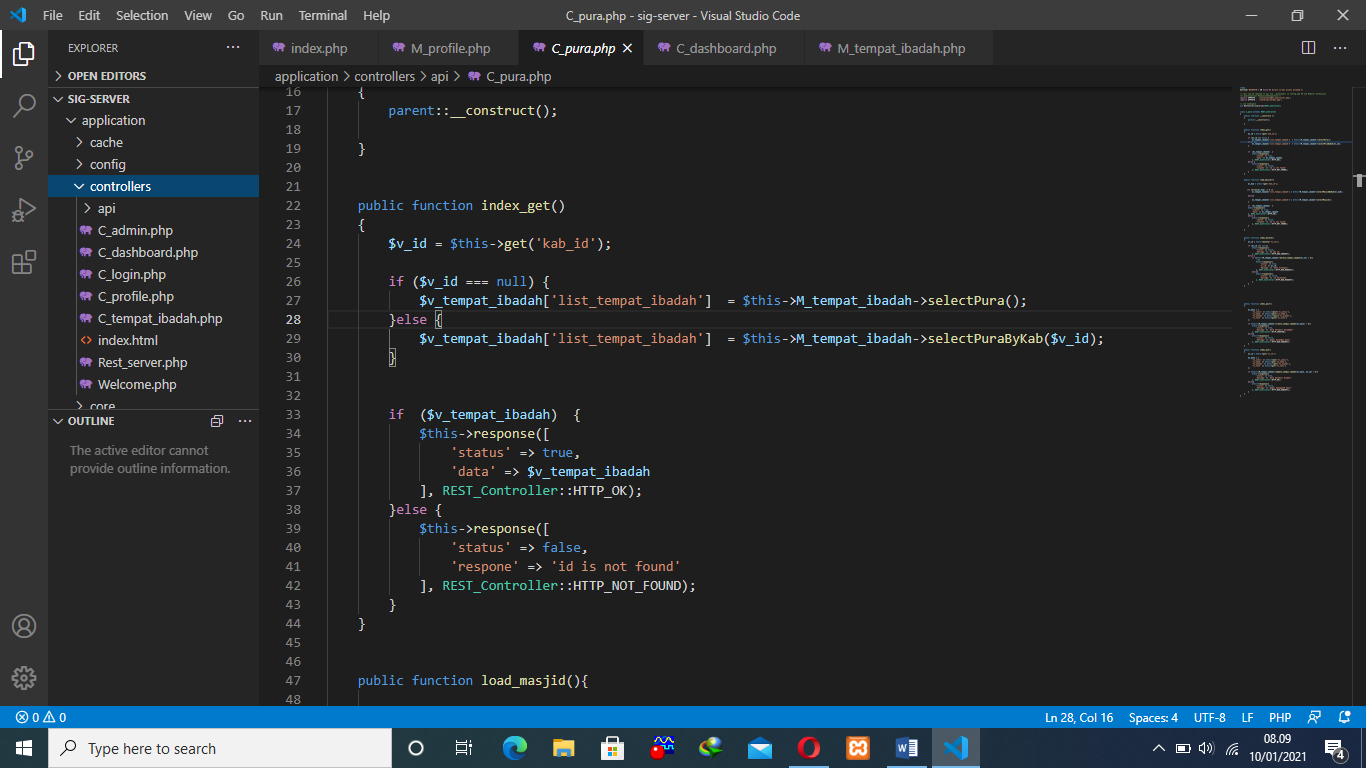
1. **Implementasi *class***

Berdasarkan perancangan *class* diagram yang telah dirancang pada bab 3, 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.

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*”.

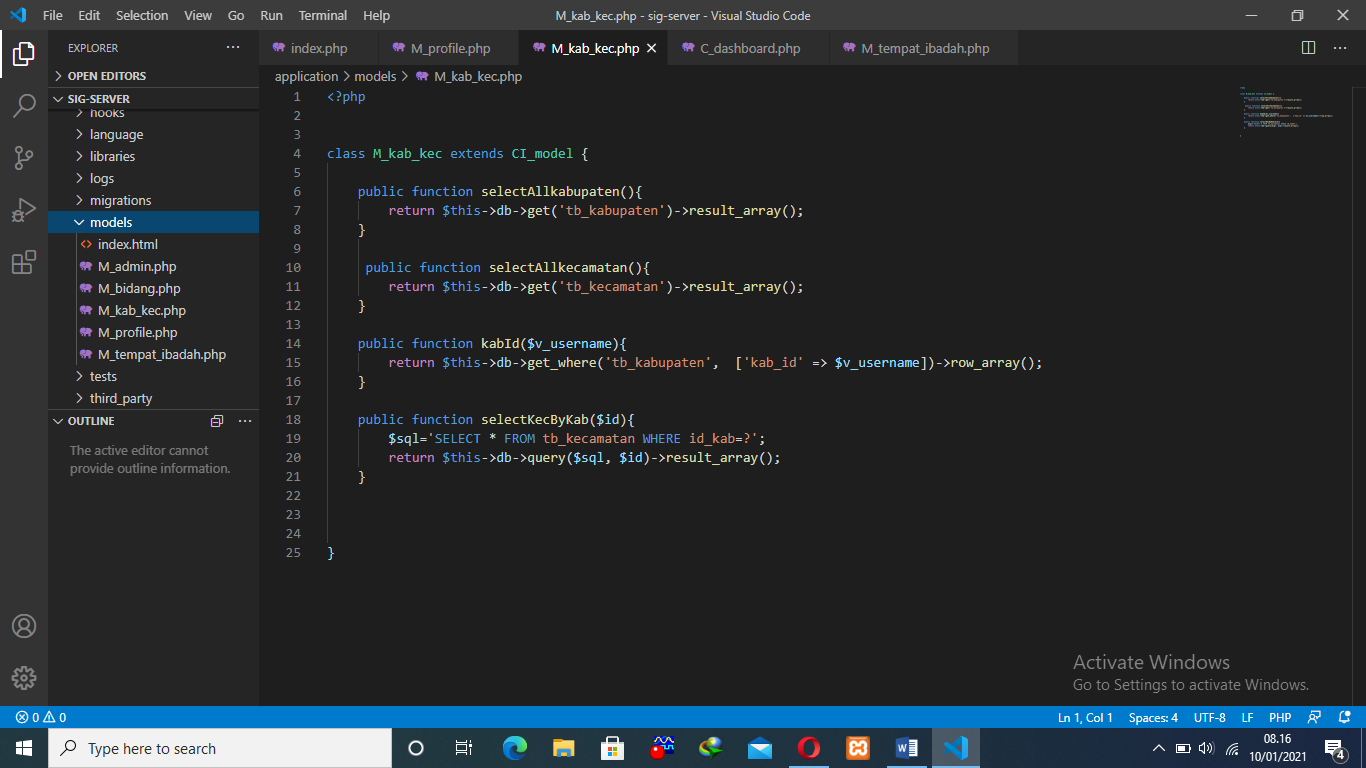
1. *Class controller*



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*



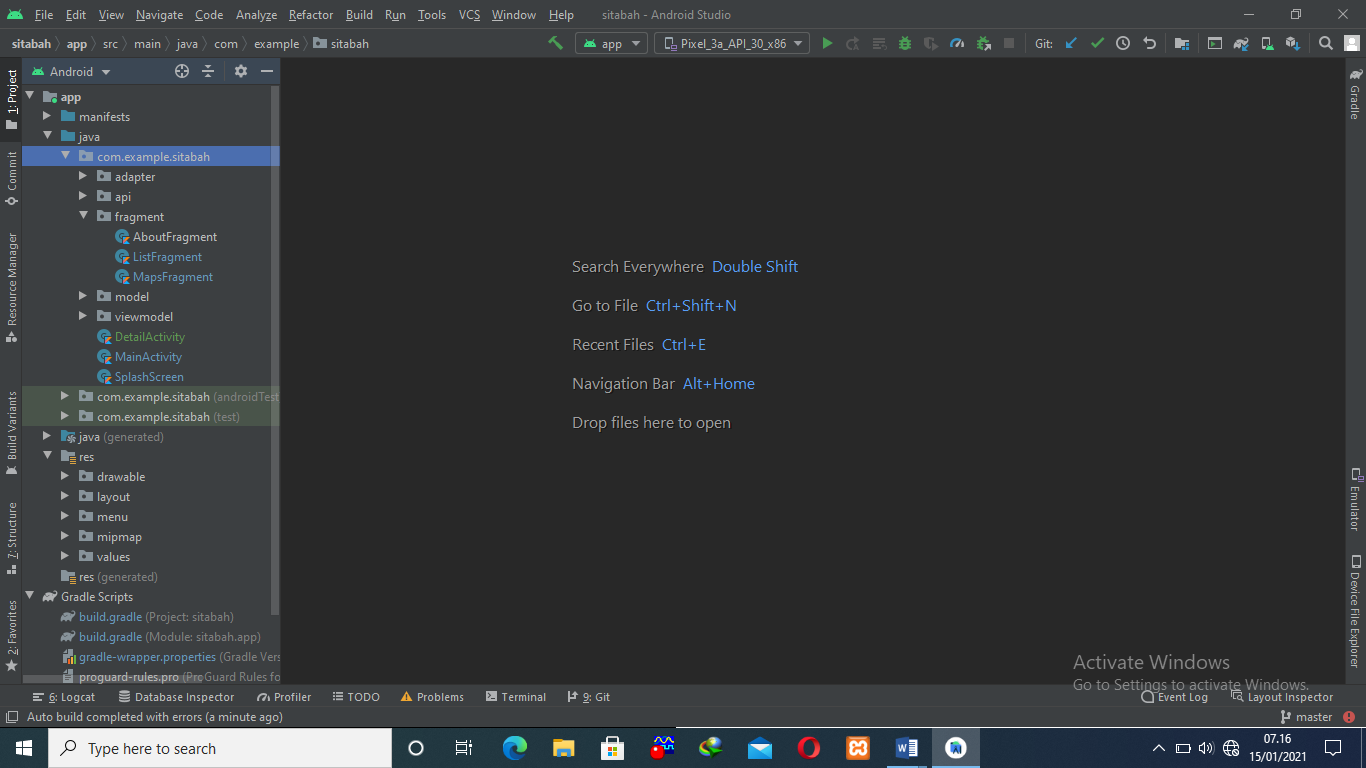
Gambar 4.10 Implementasi *class view*

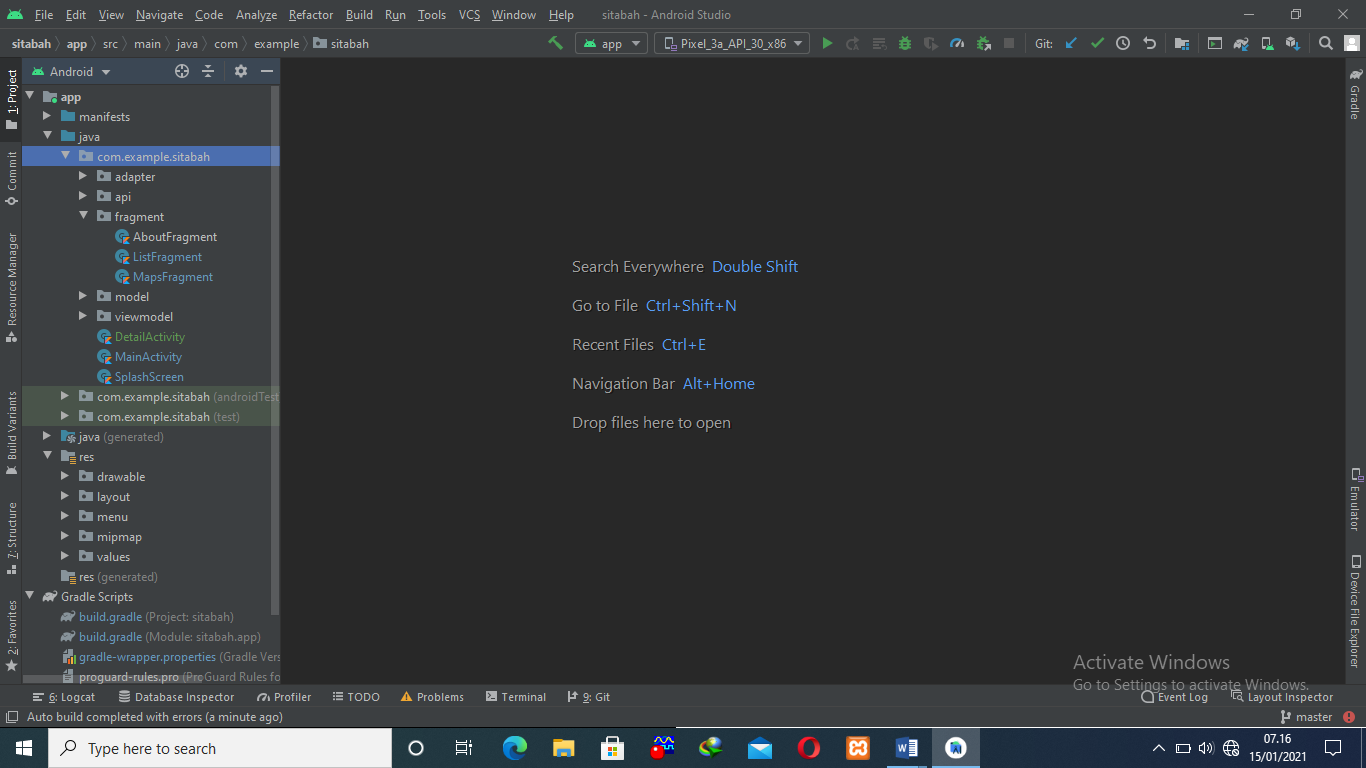
Gambar 4.10 merupakan implementasi *class* *view* yang terdiri dari V\_beranda, V\_daftar\_admin, V\_daftar\_tempat\_ibadah, V\_dashboard, V\_login, dan V\_profile. Dimana *class* ini nantinya bertugas untuk menyajikan informasi. sesuai dengan intruksi dari *controller.*

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*

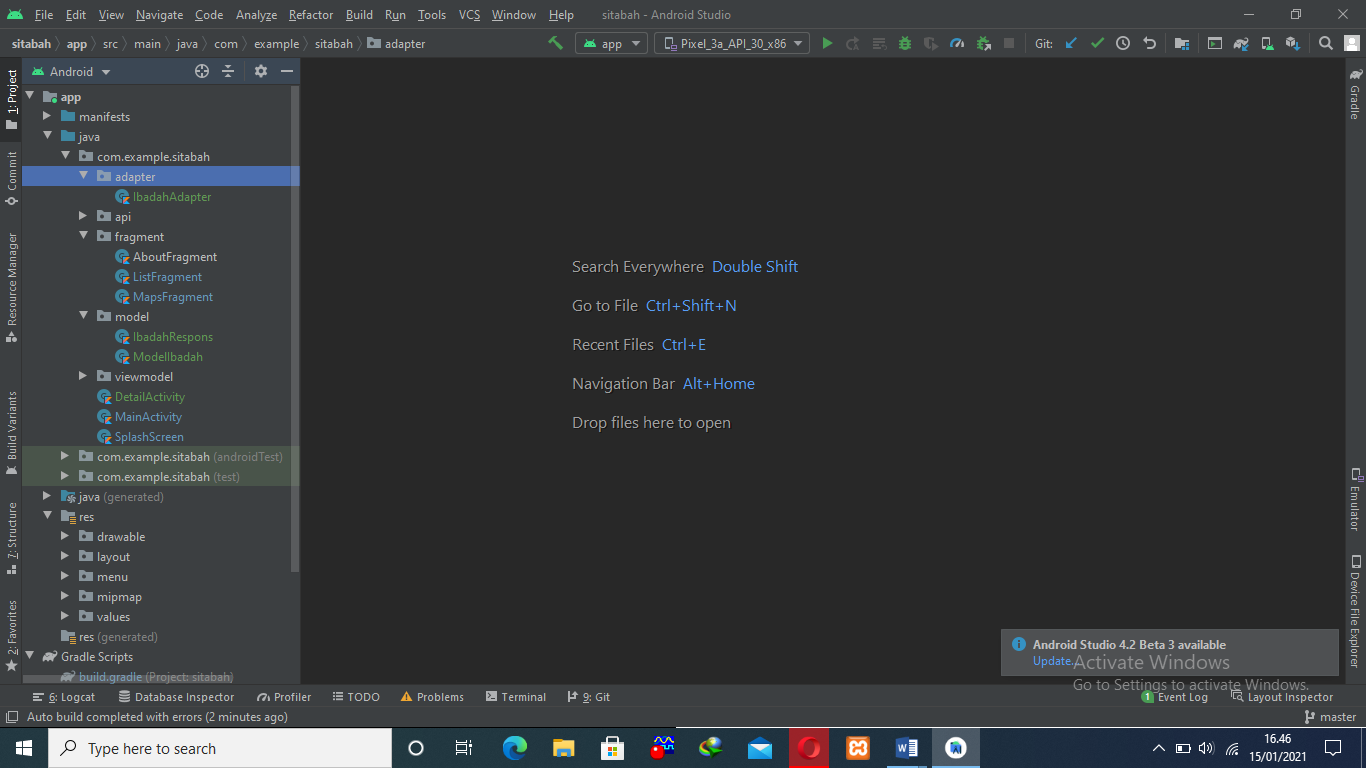




Gambar 4.11 Implementasi *class activity*

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.

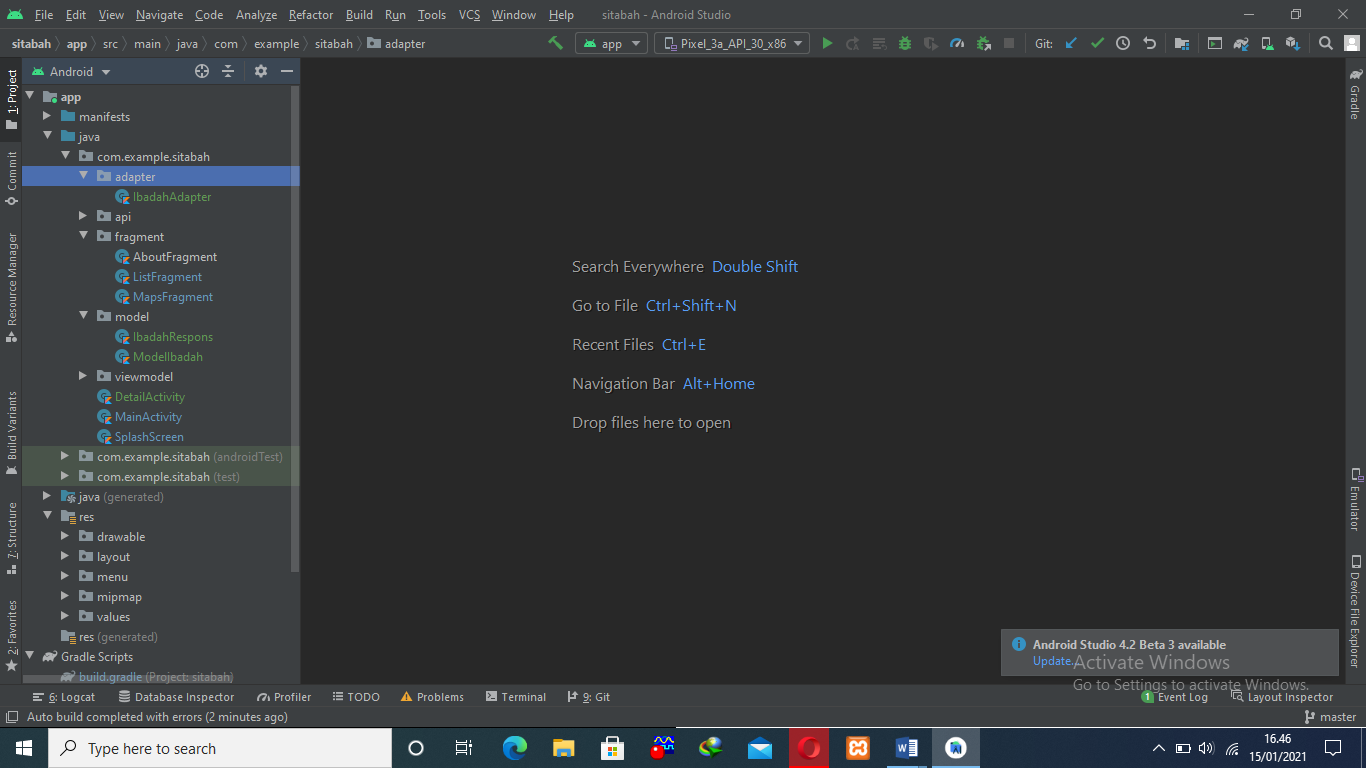
1. Adapter (Kotlin)



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.

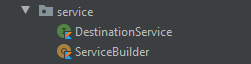
1. Model (Kotlin)



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*.

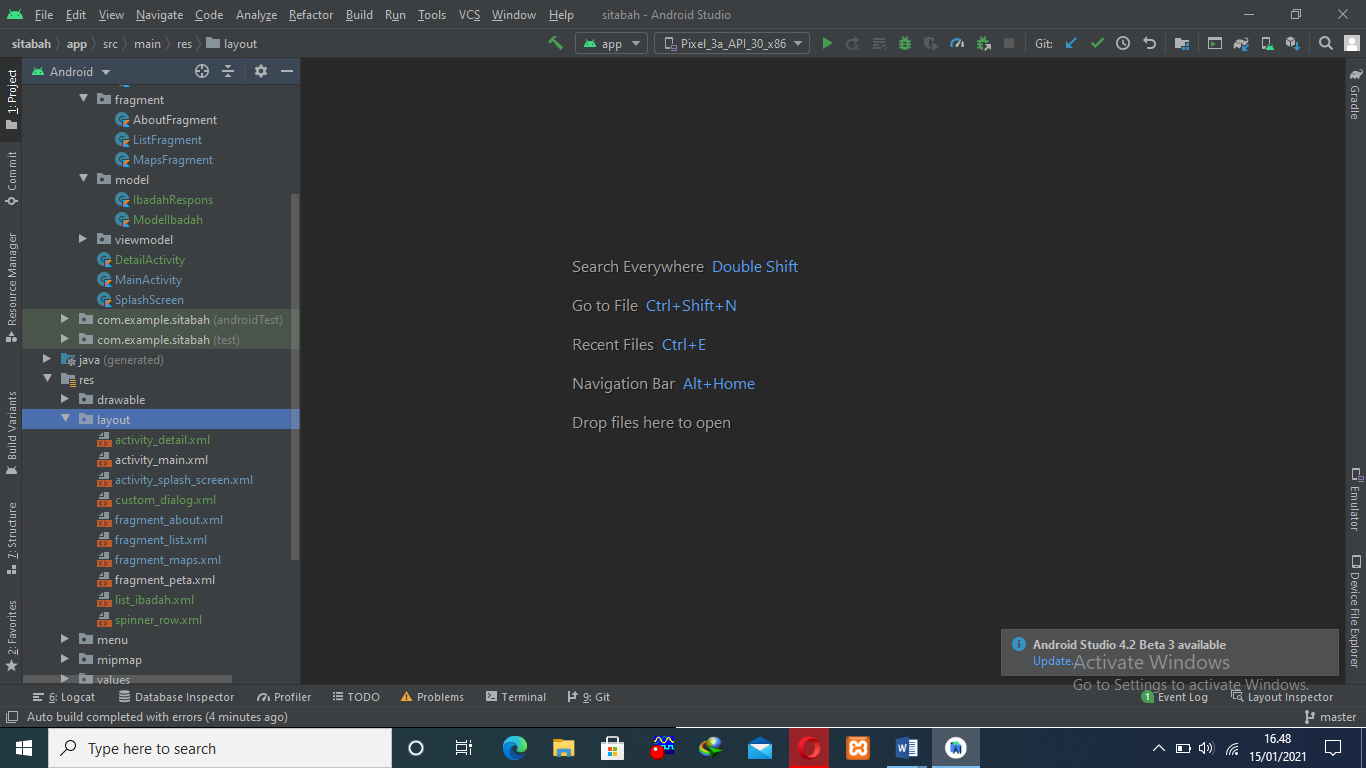
1. Service (Kotlin)



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)

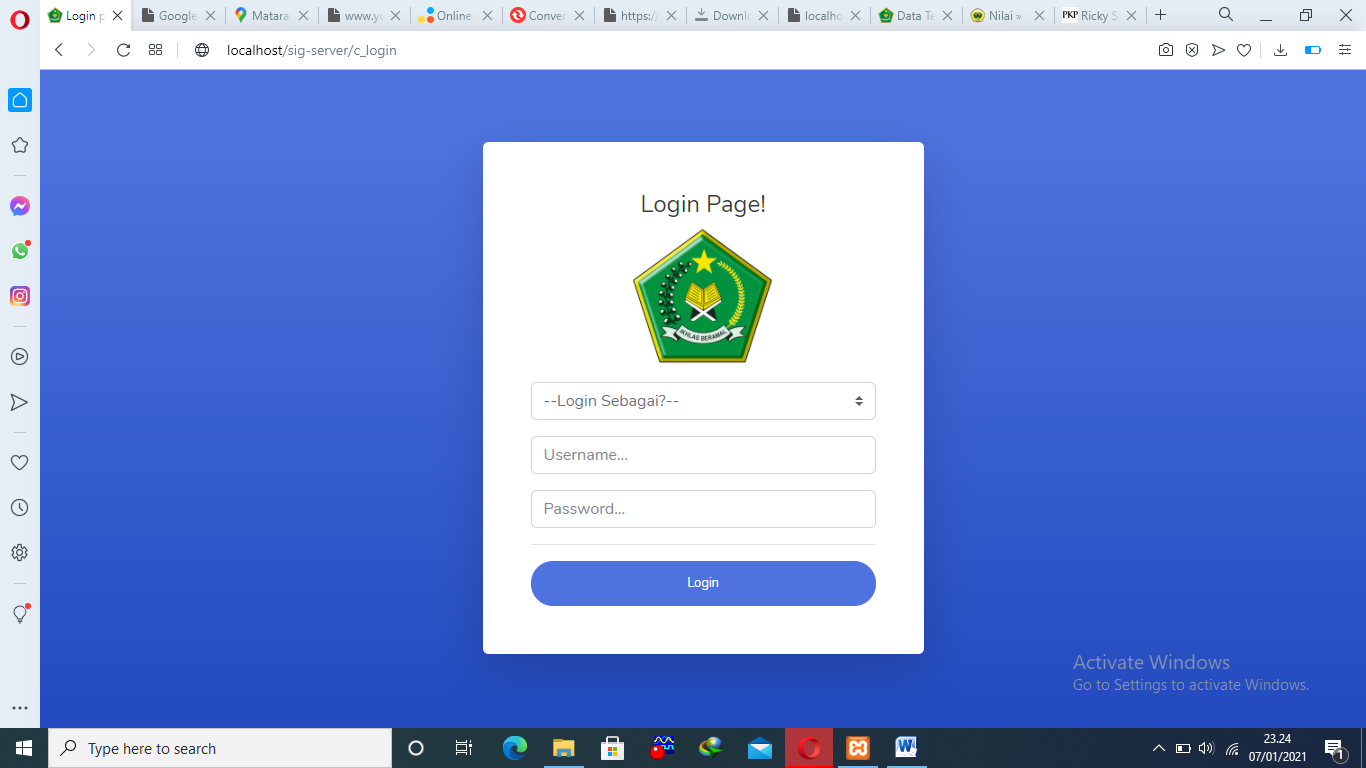
Gambar 4.15 merupakan implementasi Layout yang berguna untuk tempat membuat sebuah tampilan yang nantinya akan dilihat oleh pengguna.

1. **Implementasi *interface***

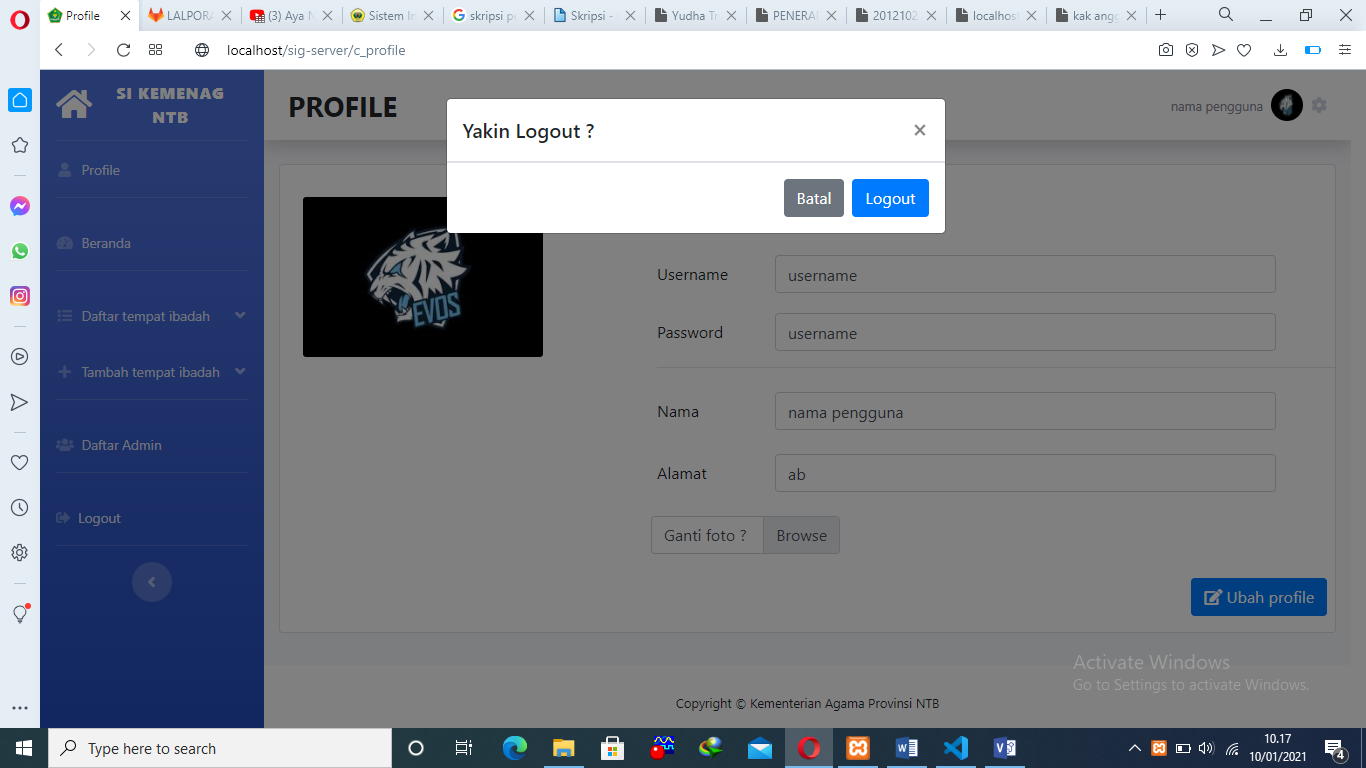
Berikut merupakan implementasi *interface* sistem yang telah dibuat beserta dengan hasil pengujian yang telah uji oleh *client* berdasarkan perencanaan dari *user stories* yang dijabarkan.

1. US-01

Berikut merupakan tampilan sistem dari kasus kode *user stories* US-01 yang telah dibuat dapat dilihat pada gambar 4.16 berikut:



Gambar 4.16 tampilan halaman *login*



Gambar 4.17 tampilan *logout*

Pengujian sistem pada kasus kode *user stories* US-01 ini dilakukan untuk menguji apakah fungsi *login* dan *logout* sistem sudah berjalan dengan baik dan benar. Pengujian tersebut dijelaskan pada Tabel 4.4 dan tabel 4.5.

Tabel 4.4 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 | Diterima |
| 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.

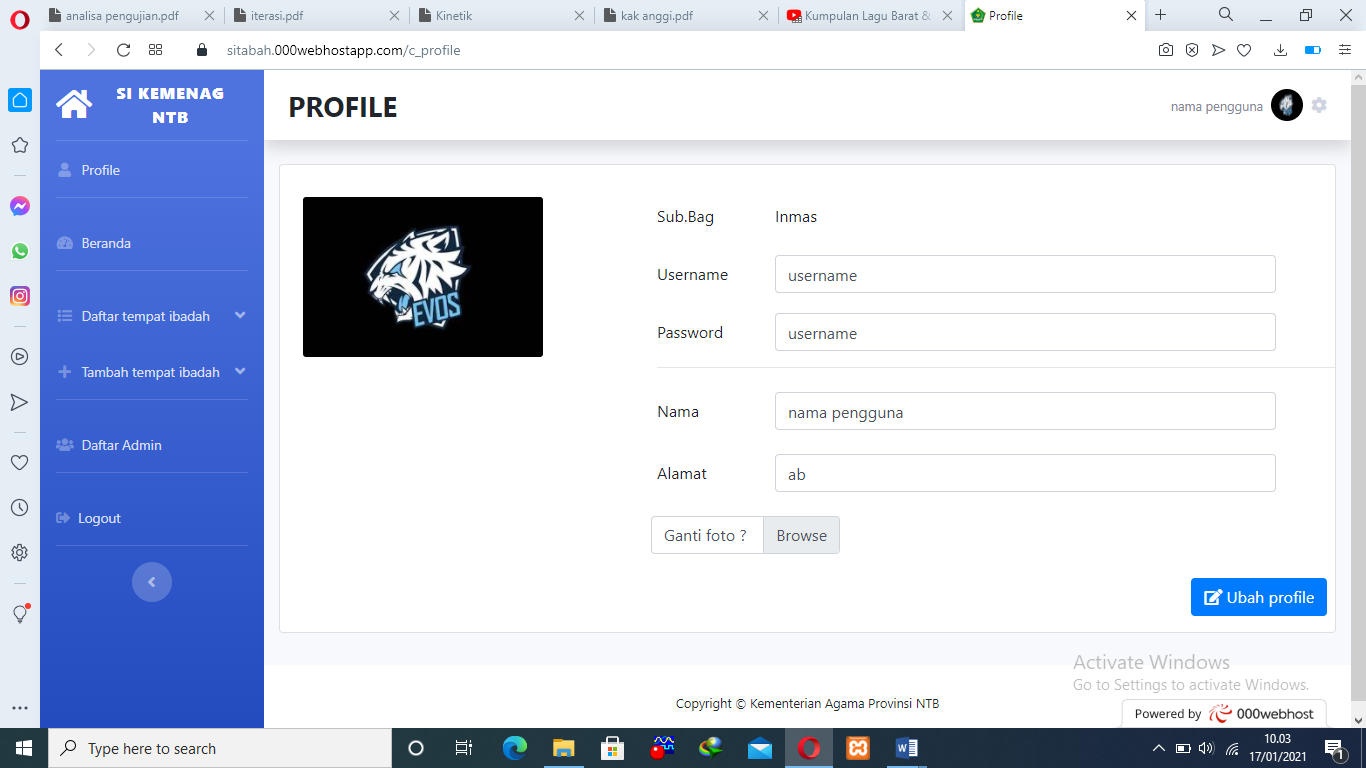
Tabel 4.5 pengujian *logout* system

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

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.4 dan table 4.5, dapat dikatakan system berhasil dan sesuai keinginan *client.*

1. US-12

Berikut merupakan tampilan halaman *profile* admin sistem dari kasus kode *user stories* US-12 yang telah dibuat dapat dilihat pada gambar 4.18 berikut:



Gambar 4.17 tampilan halaman *profile* admin

Pengujian sistem pada kasus kode *user stories* US-12 ini dilakukan untuk menguji apakah fungsi melihatdan mengubah *profile* admin pada sistem sudah berjalan dengan baik dan benar. Pengujian tersebut dijelaskan pada Tabel 4.5

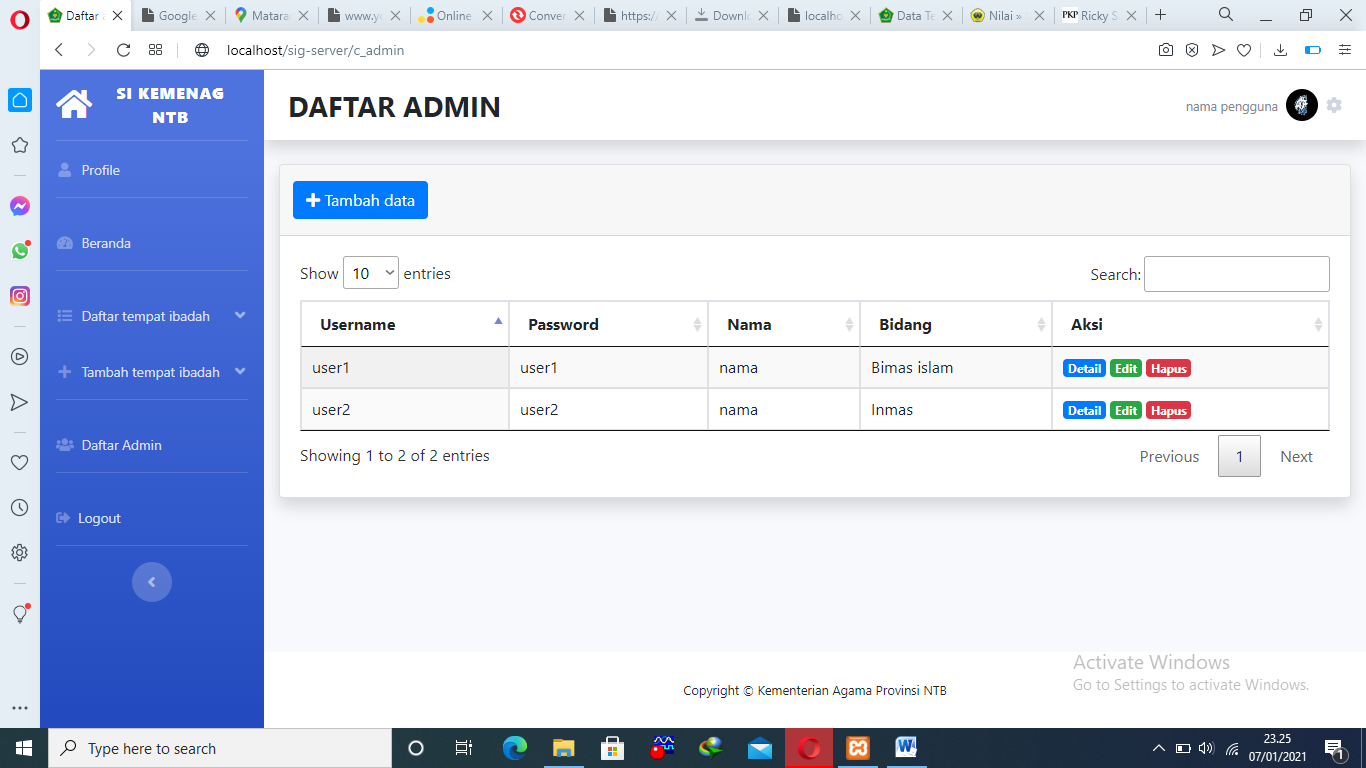
Tabel 4.4 pengujian halaman *profile* admin

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario pengujian | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| Menekan tombol halaman profile admin | Data informasi profile admin berhasil ditampilkan | Sesuai | Diterima |
| Mengisi dan mengubah data profile baru kemudian menekan tombol edit profile | Data informasi profile yang baru berhasil diubah | Sesuai | Diterima |

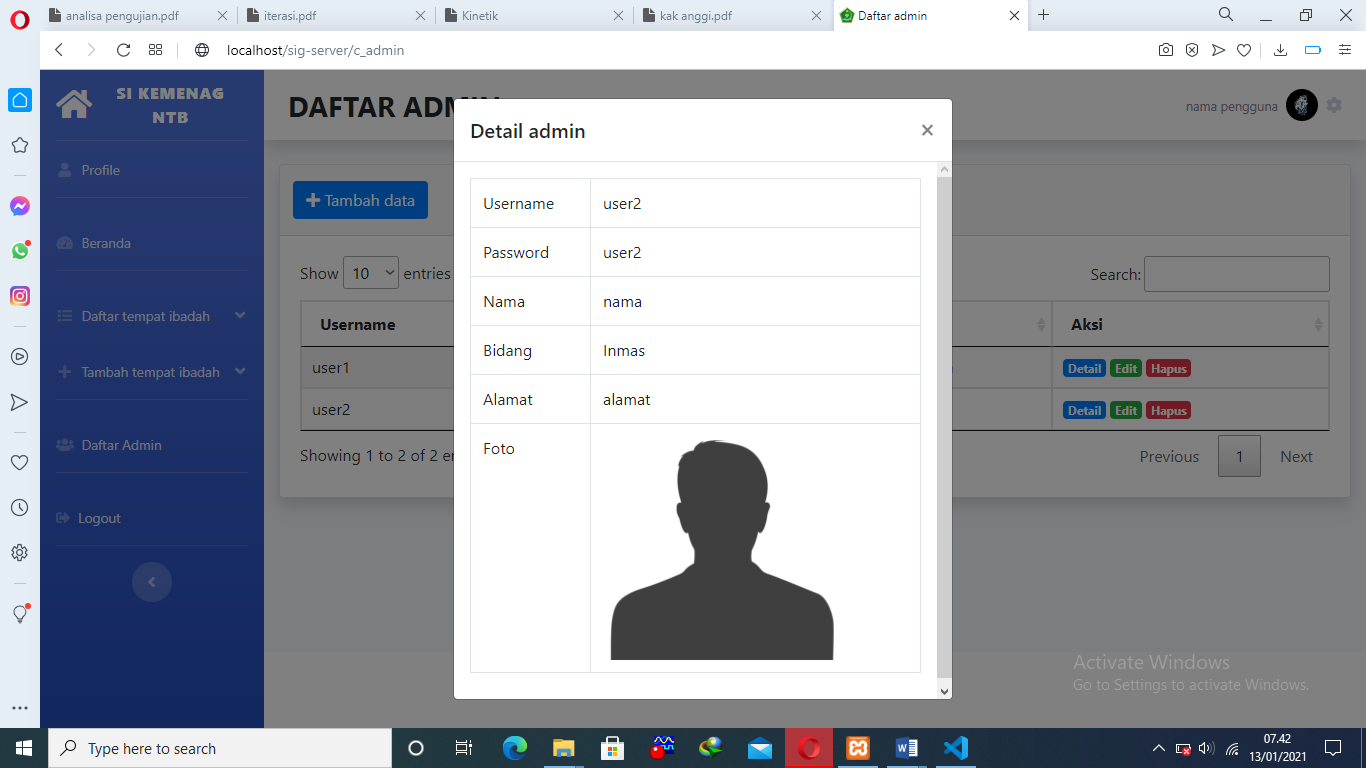
Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.4 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. US-02

Berikut merupakan tampilan sistem dari kasus kode *user stories* US-02 yang telah dibuat dapat dilihat pada gambar 4.18 dan 4.19 berikut:



Gambar 4.18 tampilan halaman daftaradmin



Gambar 4.19 tampilan detail data admin

Dilakukan pengujian untuk mengetahui apakah fungsi untuk menampilkan daftar dan detail data admin sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.5

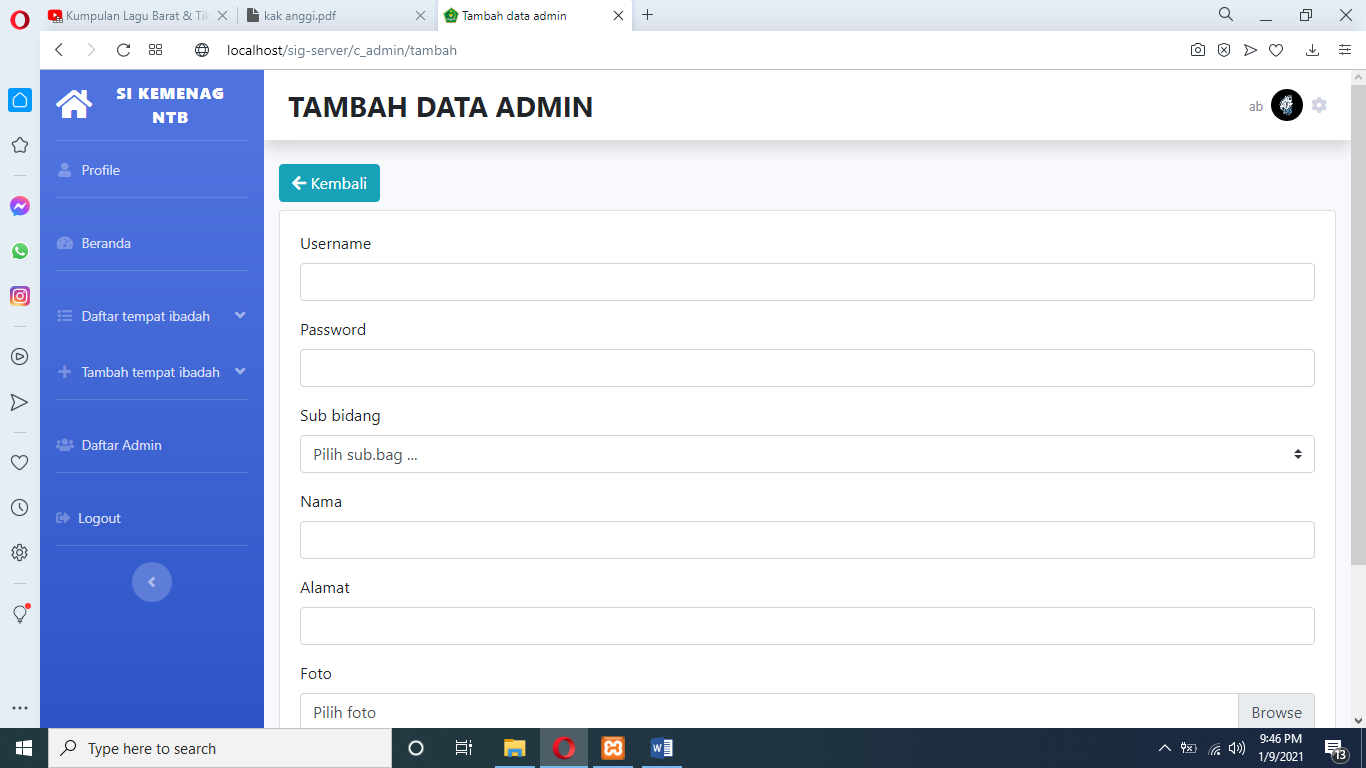
Tabel 4.5 pengujian menampilkan data admin

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

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.5 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. US-03

Berikut merupakan tampilan halaman tambah data admin dari kasus kode *user stories* US-03 yang telah dibuat dapat dilihat pada gambar 4.20 berikut:



Gambar 4.20 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 4.6

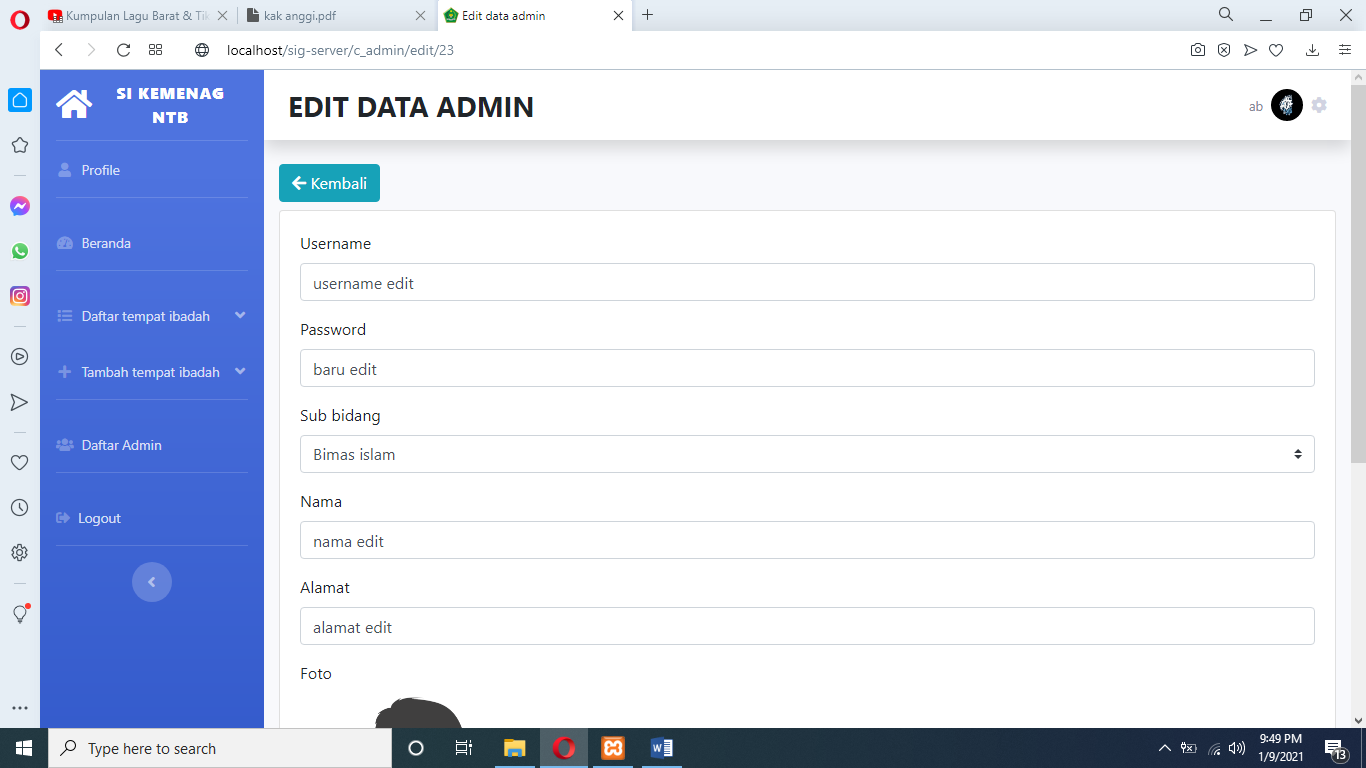
Tabel 4.6 pengujian tambah data admin

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

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 admin. Sehingga berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.6 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. US-04

Berikut merupakan tampilan halaman edit data admin dari kasus kode *user stories* US-04 yang telah dibuat dapat dilihat pada gambar 4.21 berikut:



Gambar 4.21 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 4.7

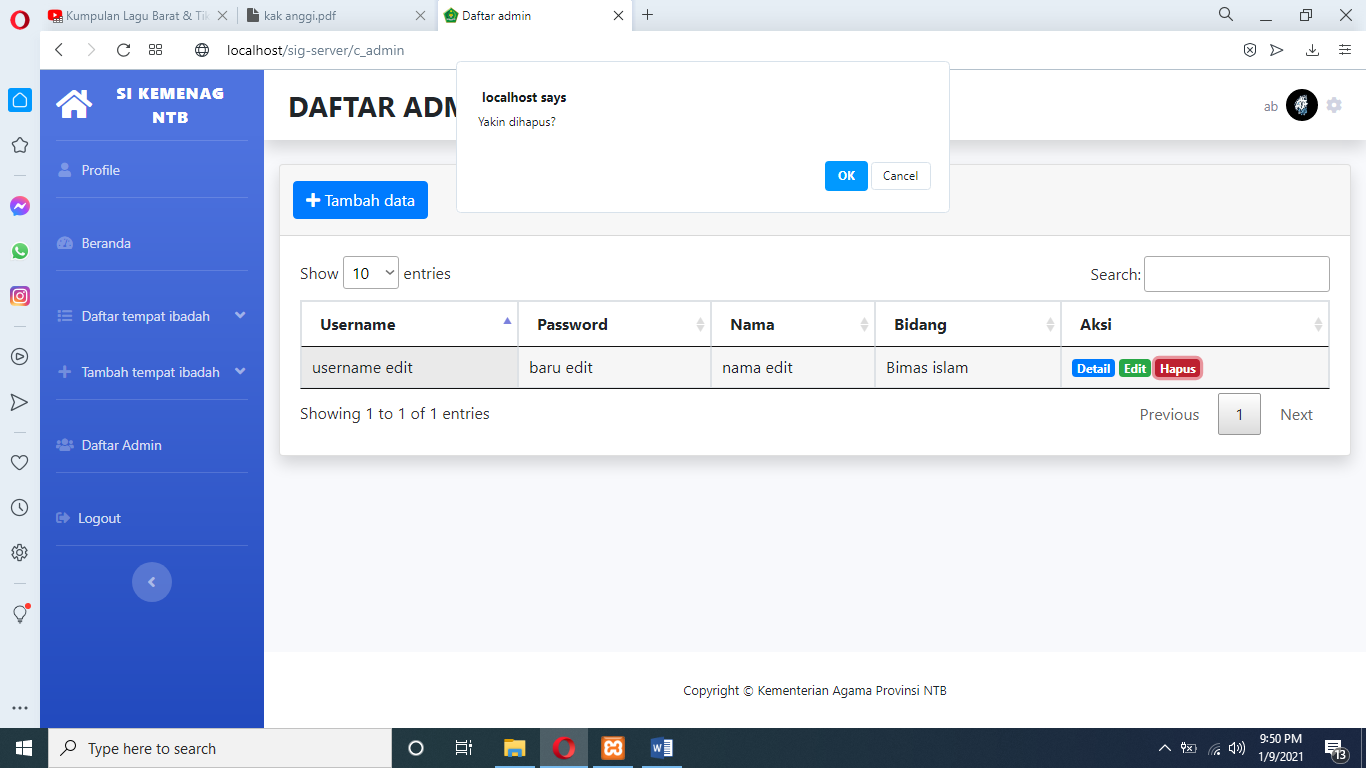
Tabel 4.7 pengujian mengubah data admin

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

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 4.7 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. US-05

Berikut merupakan tampilan hapus data admin dari kasus kode *user stories* US-05 yang telah dibuat dapat dilihat pada gambar 4.22 berikut:



Gambar 4.22 tampilan hapus data admin

Pengujian fungsi pada kasus kode *user stories* US-05 ini dilakukan untuk menguji apakah fungsi untuk menghapus data admin sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.8

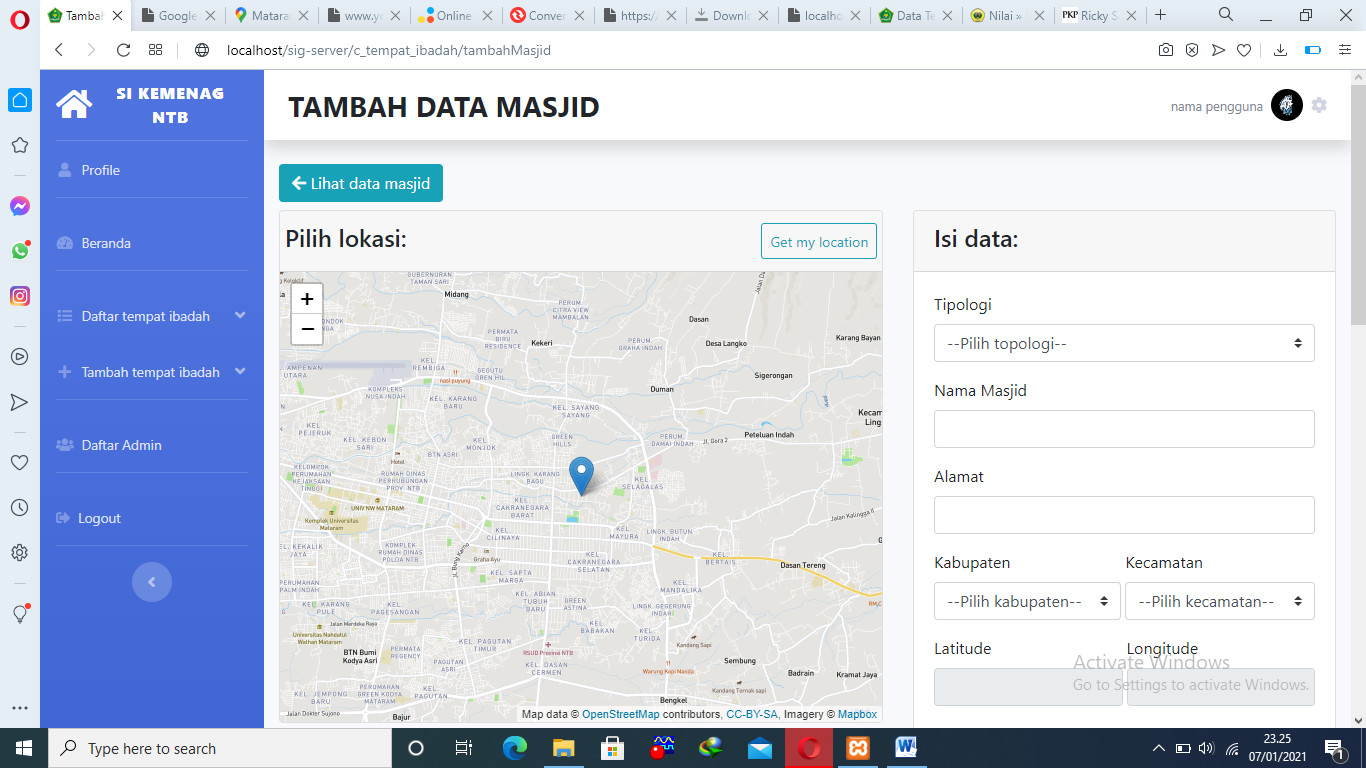
Tabel 4.8 pengujian mengahpus data admin

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

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

1. US-07

Berikut merupakan tampilan halaman tambah data tempat ibadah dari kasus kode *user stories* US-07 yang telah dibuat dapat dilihat pada gambar 4.23 berikut:



Gambar 4.23 tampilan halaman tambah data tempat ibadah

Gambar 4.23 merupakan implementasi halaman tambah data tempat ibadah yang akan ditampilkan dalam sistem basis *web*. Berikut merupakan tabel pengujian system terhadap kasus kode *user stories* US-07 dapat dilihat pada table 4.9.

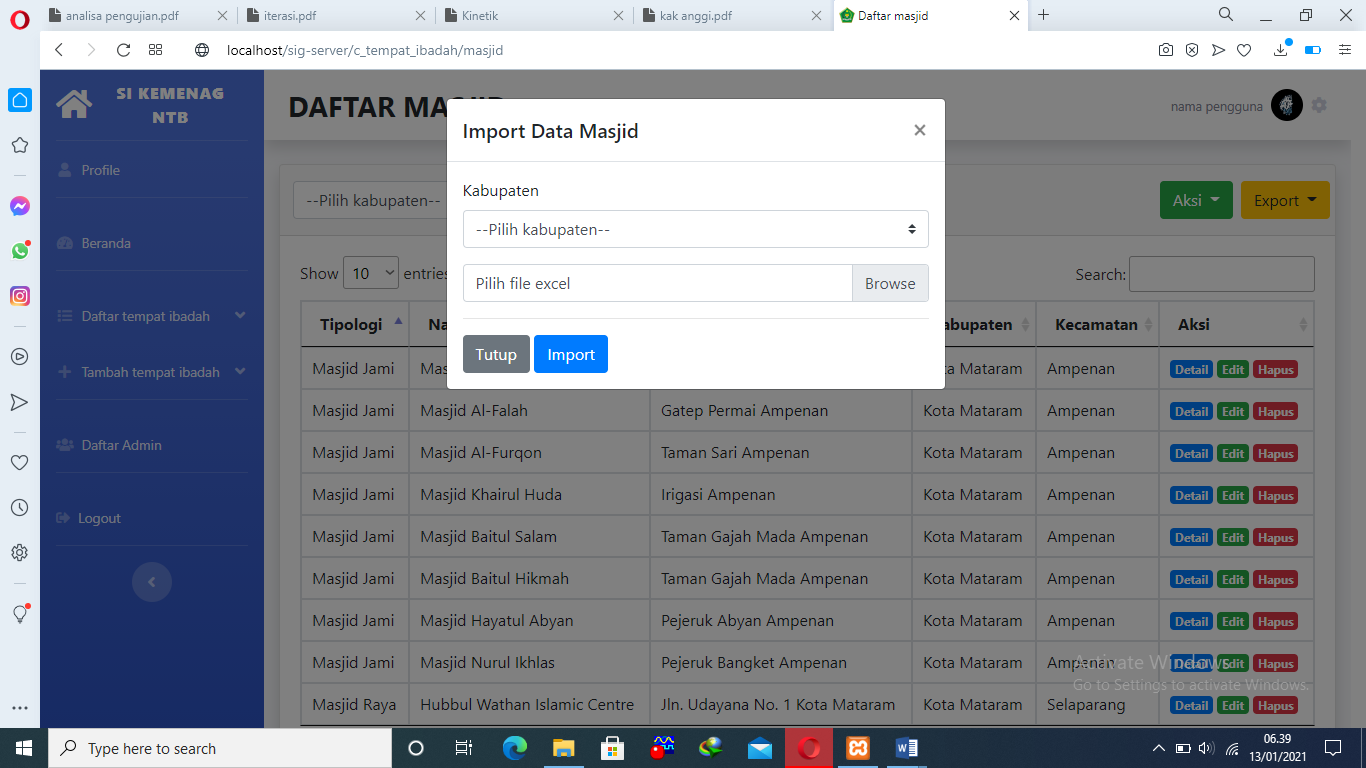
Tabel 4.9 pengujian tambah data tempat ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Nama atau alamat tempat ibadah tidak diisi, serta kabupaten atau kecamatan tidak dipilih | Sistem akan melakukan proses dan data tidak berhasil ditambahkan ke dalam *database* | Sesuai | Diterima |
| Nama dan alamat tempat ibadah diisi, serta kabupaten dan kecamatan dipilih | Sistem akan melakukan proses dan data berhasil ditambahkan ke dalam *database* | Sesuai | Diterima |

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. Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.9 dapat dikatakan system yang dibuat berhasil dan sesuai dengan kebutuhan *client.*

1. US-14

Berikut merupakan tampilan halaman tambah data tempat ibadah dari kasus kode *user stories* US-14 yang telah dibuat dapat dilihat pada gambar 4.24 berikut:



Gambar 4.24 tampilan import data tempat ibadah

Gambar 4.24 merupakan implementasi halaman *import* data. Terdapat tombol *import* data untuk meng-*import* data excel. Berikut merupakan tabel pengujian system dapat dilihat pada table 4.10.

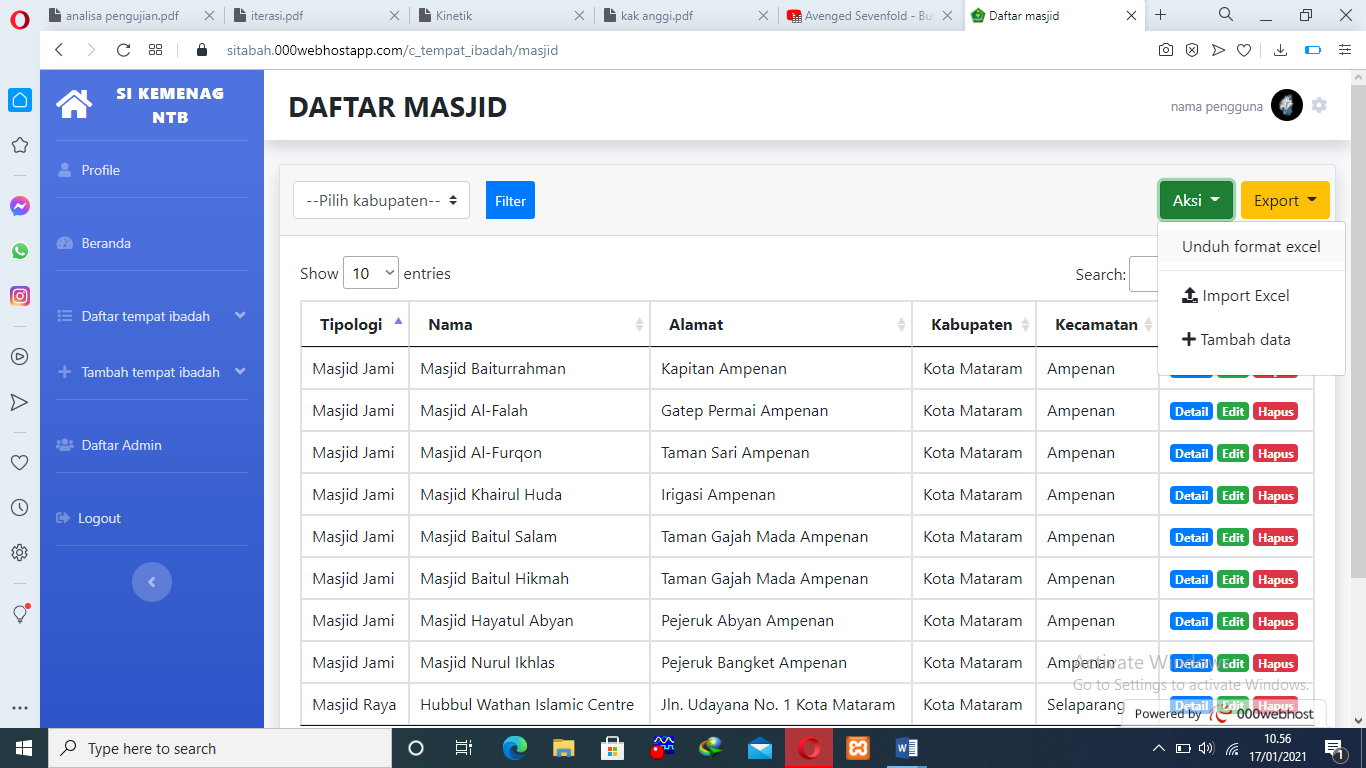
Tabel 4.10 pengujian *import* data tempat ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol *“import data”* | Menampilkan *form* untuk memasukkan file excel data tempat ibadah | Sesuai | Diterima |
| Menekan tombol “pilih file excel” | Menampilkan jendela baru untuk memasukkan file *excel* | Sesuai | Diterima |
| Menekan tombol “*import*” | Memproses dan memasukkan data ke dalam *database* | Sesuai | Diterima |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.10 dapat dikatakan system yang dibuat berhasil dan sesuai dengan kebutuhan *client.*

1. US-15

Berikut merupakan tampilan untuk mengunduh format excel import data tempat ibadah pada sistem dari kasus kode *user stories* US-15 yang telah dibuat dapat dilihat pada gambar 4.25 berikut:



Gambar 4.25 tampilan import data tempat ibadah

Gambar 4.24 merupakan implementasi tampilan tombol untuk mengunduh format excel data tempat ibadah. Berikut merupakan tabel pengujian system dapat dilihat pada table 4.11.

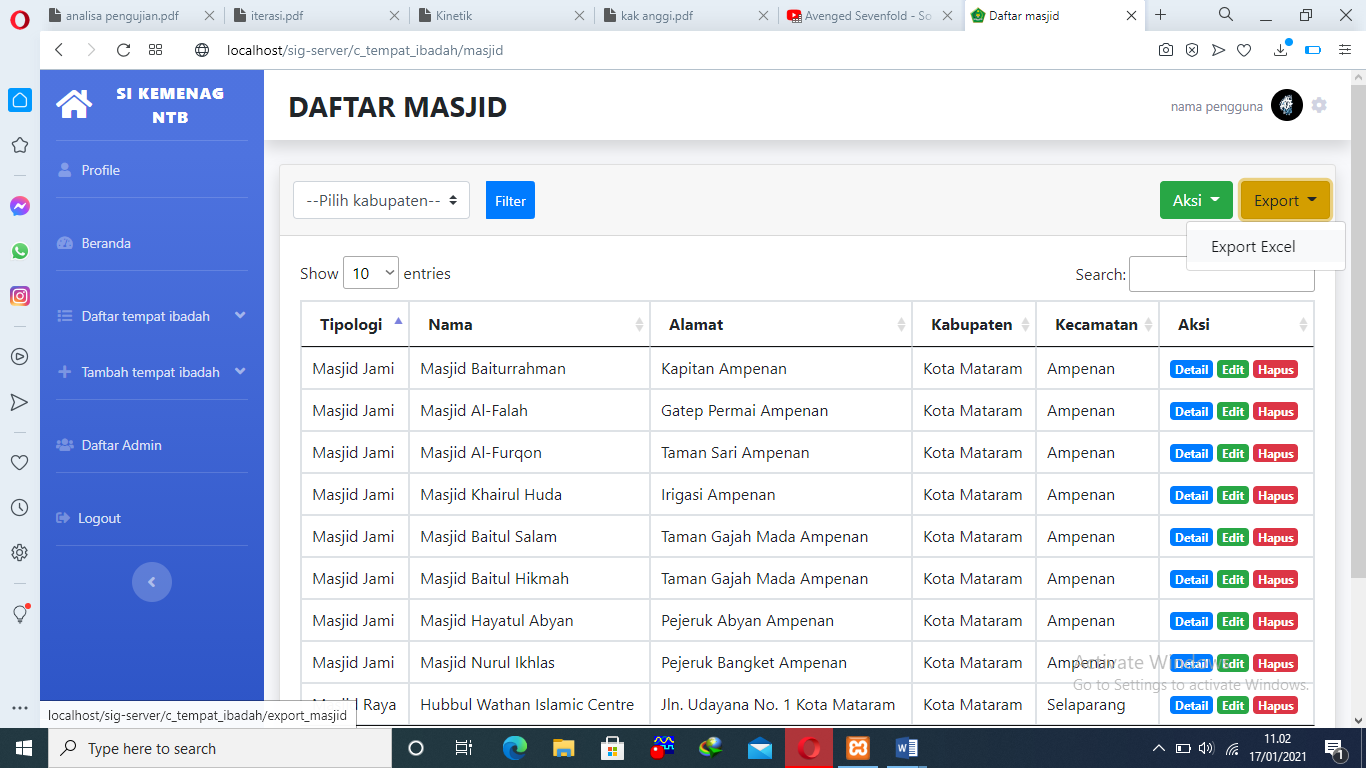
Tabel 4.11 pengujian unduh template format excel import data

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol “*unduh format excel*” | Format excel data tempat ibadah terunduh | Sesuai | Diterima |

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

1. US-16

Berikut merupakan tampilan tombol untuk *export* data tempat ibadah pada sistem dari kasus kode *user stories* US-16 yang telah dibuat dapat dilihat pada gambar 4.26 berikut:



Gambar 4.26 tampilan *export* data tempat ibadah

Gambar 4.26 merupakan implementasi tampilan tombol untuk export data tempat ibadah dalam bentuk *file excel*. Berikut merupakan tabel pengujian system dapat dilihat pada table 4.12.

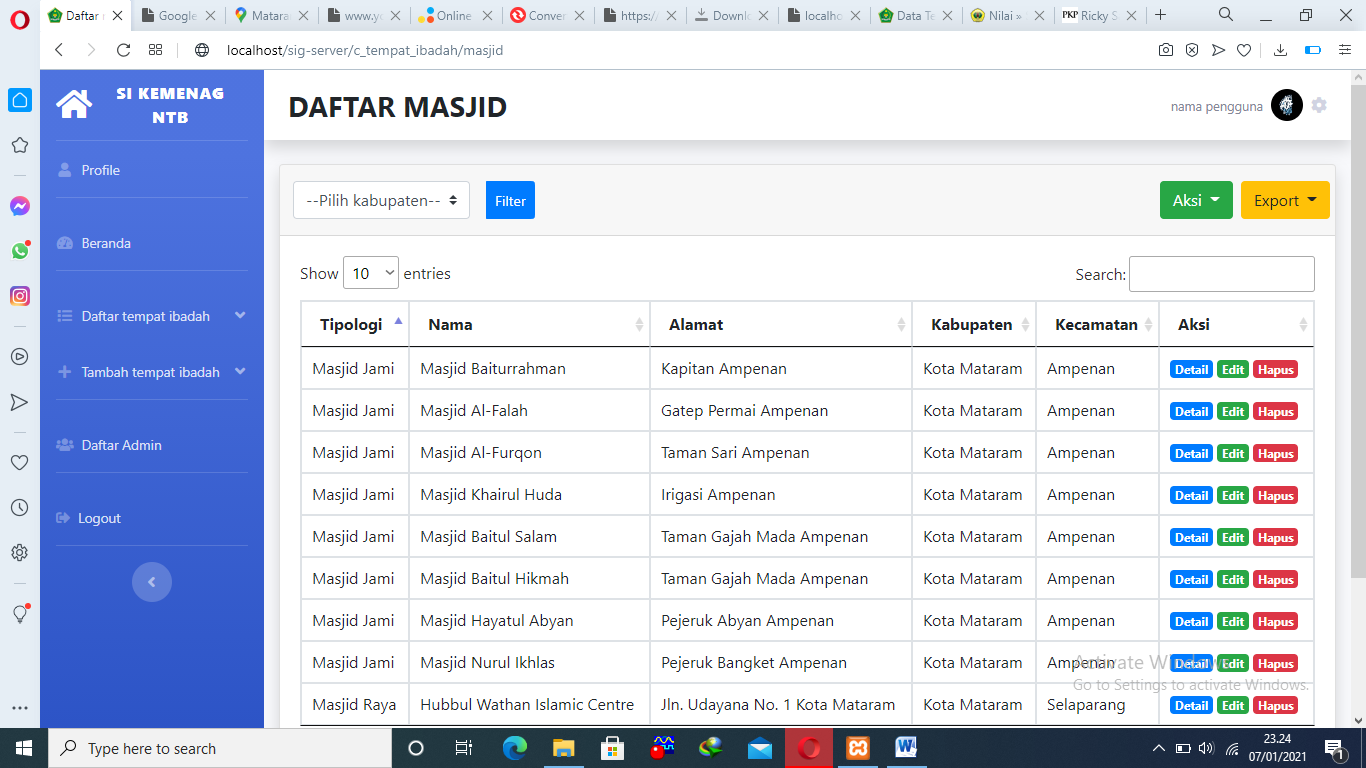
Tabel 4.12 pengujian *export* data *excel* tempat ibadah

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol “*export*” | Data tempat ibadah terunduh dalam bentuk excel | Sesuai | Diterima |

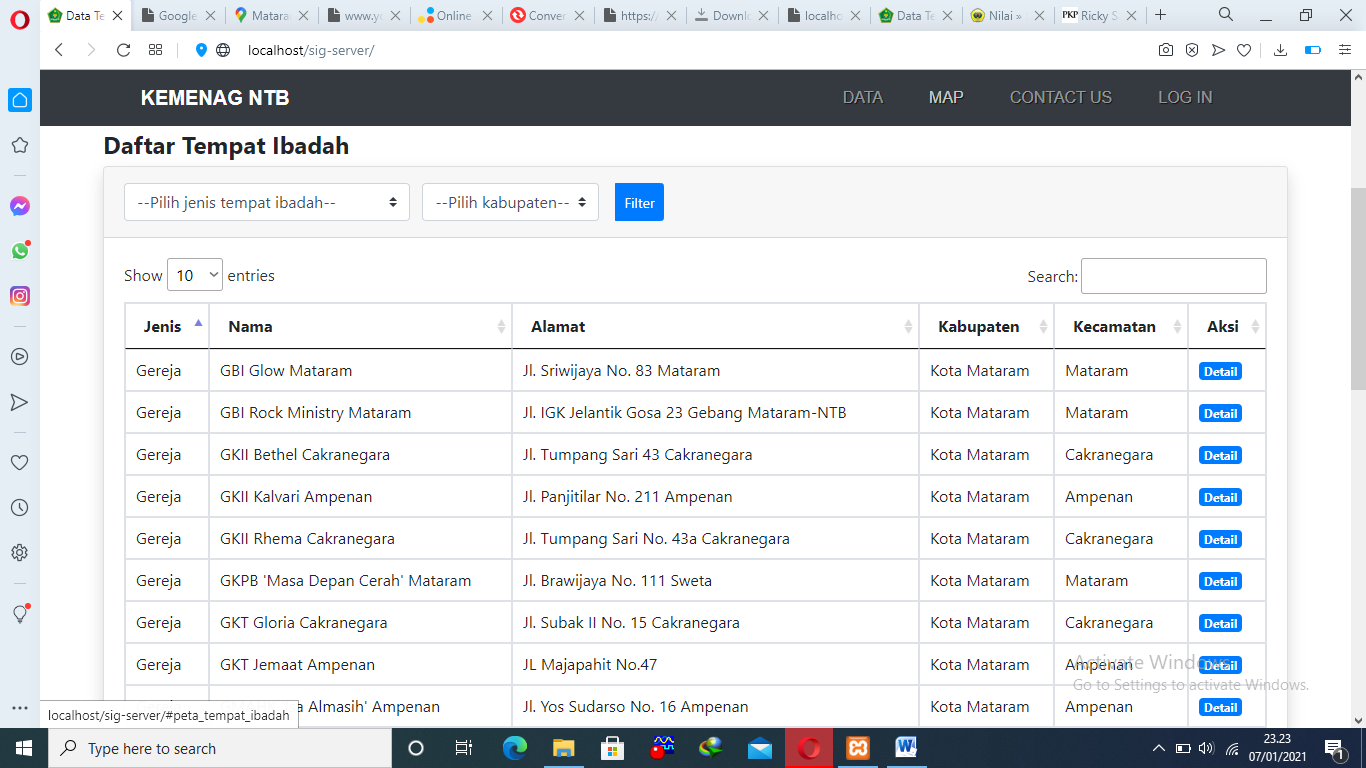
Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.12 dapat dikatakan system yang dibuat berhasil dan sesuai dengan kebutuhan *client.*

1. US-06

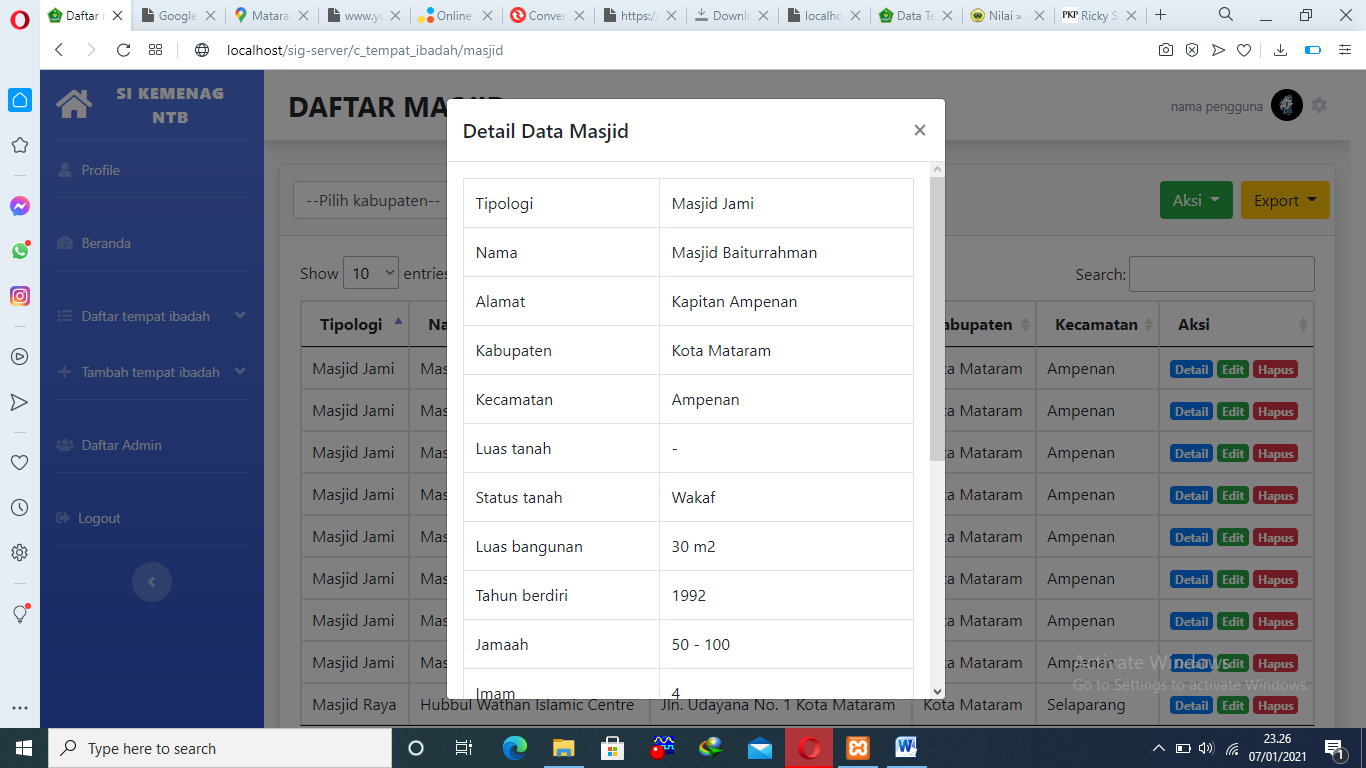
Berikut merupakan tampilan halaman daftar tempat ibadah yang ada pada basis *web* dan *android* dari kasus kode *user stories* US-06 yang telah dibuat dapat dilihat pada gambar 4.27, 4.28, 4.29, 4.30 dan 4.31berikut:



Gambar 4.27 tampilan daftar tempat ibadah basis web sisi admin



Gambar 4.28 tampilan daftar tempat ibadah basis *web* sisi masyarakat



Gambar 4.29 tampilan detail data tempat ibadah basis *web*

GAMBAR ANDROID

Gambar 4.30 tampilan daftar tempat ibadah basis *android*

GAMBAR ANDROID

Gambar 4.31 tampilan detail tempat ibadah basis *android*

Dilakukan pengujian sistem untuk mengetahui apakah fungsi untuk menampilkan daftar dan detail data tempat ibadah sudah berjalan dengan baik dan benar. Pengujian fungsi tersebut dapat dilihat pada Tabel 4.13

Tabel 4.13 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 | Diterima |
| Menekan tombol detail data tempat ibadah yang dipilih (*web* atau *android*) | Detail data tempat ibadah yang dipilih berhasil ditampilkan | Sesuai | Diterima |
| Memasukkan *keyword* pada fitur pencarian terhadap data tempat ibadah (*web* atau *android*) | Menampilkan hasil *keyword* pencarian | Sesuai | Diterima |
| Menggunakan filter kabupaten dan jenis tempat ibadah(*web*) | Menampilkan hasil tempat ibadah berdasarkan filter yang diiginkan | Sesuai | Diterima |
| Menggunakan filter jenis tempat ibadah (*android*) | Menampilkan hasil tempat ibadah berdasarkan filter yang diiginkan | Sesuai | Diterima |

Berdasarkan hasil pengujian dan kesimpulan dari pengujian yang telah dilakukan pada tabel 4.13 dapat dikatakan system berhasil dan sesuai kebutuhan *client.*

1. US-13
2. US-08
3. US-09
4. US-10
5. US-11
6. US-17
7. US-18
8. ***Testing***

Walaupun pengujian system telah dilakukan pengujian 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*. Pengujian pada tahap ini juga dilakukan tidak hanya untuk mengetahui adanya *error* atau tidak namun juga untuk mengetahui kualitas sistem dari sisi pengguna.

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 |

Berdasarkan table hasil pengujian black box yang telah dipaparkan pada table … dapat dikatakan bahwa sistem yang dibuat telah berhasil berjalan dengan baik dan benar serta telah sesuai dengan keinginan *client*

1. **Hasil pengujian *Mean Opinion Score* (MOS)**

Pengujian ini bertujuan untuk mengetahui bagaimana kualitas sistem dilihat dari sisi pengguna. Pada pengujian ini menggunakan kuesioner dengan mencari responden untuk mencoba menjalankan sistem, lalu memberikan pernyataan berupa kuesioner.

Pengujian ini dilakukan oleh 30 responden yang terdiri dari 2 orang pihak kemenag NTB sub bagian inmas dan bimas islam sebagai admin, dan sisanya dipilih secara acak yang terdiri dari 28 orang sebagai pengguna aplikasi. Hasil dari jawaban responden nantinya akan dihitung dan ditarik kesimpulan mengenai hasil pengujian sistem. Berikut daftar pertanyaan yang diajukan kepada responden:

1. Admin
2. Apakah tampilan aplikasi mudah dipahami?
3. Apakah aplikasi mudah digunakan?
4. Apakah aplikasi ini dapat membantu pihak dalam mengelola data tempat ibadah yang ada di aplikasi?
5. Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?
6. Apakah aplikasi berjalan dengan baik?
7. Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?
8. Masyarakat
9. Apakah tampilan aplikasi mudah dipahami?
10. Apakah aplikasi mudah digunakan?
11. Apakah aplikasi ini dapat membantu masyarakat dalam mencari lokasi tempat ibadah yang diinginkan?
12. Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?
13. Apakah aplikasi berjalan dengan baik?
14. Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?

Responden diminta untuk menjawab kuesioner tersebut dengan pilihan jawaban sebagai berikut:

1 = Tidak Setuju

2 = Kurang Setuju

3 = Cukup

4 = Setuju

5 = Sangat Setuju

Berdasarkan jawaban tersebut, dilakukan perhitungan persentase dari setiap jawaban yang diberikan oleh pengguna di setiap pertanyaannya. Rumus untuk menghitung persentase jawaban adalah sebagai berikut:

Persentase = J / N \* 100%

J : Total nilai jawaban yang diberikan pengguna pada setiap nomor.

N : Jumlah pengguna.

Selanjutnya dilakukan proses perhitungan persentase rata-rata dari setiap poin jawaban, perhitungan yang digunakan adalah menggunakan rumus sebagai berikut:

|  |
| --- |
| Rata-rata = (Persentase ke-1 + ….. + Persentase ke-6) / 6 |

Persentase jawaban responden terhadap kuesioner yaitu, tidak setuju, kurang setuju, cukup, setuju dan sangat setuju dapat dihitung secara keseluruhan dan didefinisikan sebagai nilai rata-rata. Dengan menggunakan rumus perhitungan diatas, perhitungan untuk hasil jawaban pengguna pada setiap pertanyaan pada kuesioner adalah sebagai berikut:

1. Admin

Pada Tabel 4.40 merupakan hasil pengujian kuisioner yang telah dilakukan terhadap 2 pegawai kemenag NTB. Sehingga didapatkan hasil bahwa rata-rata yang menjawab sangat setuju adalah 0%, setuju 0%, Cukup 0%, Kurang setuju 0% dan Tidak Setuju adalah 0%

Keterangan :

Pertanyaan 1 = Apakah tampilan sistem informasi mudah dipahami?

Pertanyaan 2 = Apakah sistem mudah digunakan?

Pertanyaan 3 = Apakah aplikasi ini dapat membantu pihak kemenag NTB dalam mengelola data tempat ibadah yang ada di aplikasi?

Pertanyaan 4 = Apakah aplikasi sudah sesuai dengan kebutuhan anda?

Pertanyaan 5 = Apakah aplikasi berjalan dengan baik?

Pertanyaan 6 = Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?

SS = Sangat setuju, S = Setuju, C = Cukup, KS = Kurang Setuju, Ts = Tidak Setuju

Tabel 4.40 Hasil Pengujian Kuisoner Dengan Pegawai KEMENAG NTB

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pertanyaan | SS | | S | | C | | KS | | TS | |
| total | % | total | % | total | % | total | % | total | % |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Rata-rata | 0% | | 0% | | 0 | | 0 | | 0 | |

1. Masyarakat

Pada Tabel 4.41 merupakan hasil pengujian kuisioner yang telah dilakukan terhadap 28 orang. Sehingga didapatkan hasil bahwa rata-rata yang menjawab sangat setuju adalah 0%, setuju 0%, Cukup 0%, Kurang setuju 0% dan Tidak Setuju adalah 0%

Keterangan :

Pertanyaan 1 = Apakah tampilan sistem informasi mudah dipahami?

Pertanyaan 2 = Apakah sistem mudah digunakan?

Pertanyaan 3 = Apakah aplikasi ini dapat membantu masyarakat menemukan informasi tempat ibadah yang diinginkan dengan mudah?

Pertanyaan 4 = Apakah aplikasi sudah sesuai dengan kebutuhan anda?

Pertanyaan 5 = Apakah aplikasi berjalan dengan baik?

Pertanyaan 6 = Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?

SS = Sangat setuju, S = Setuju, C = Cukup, KS = Kurang Setuju, Ts = Tidak Setuju

Tabel 4.2. Hasil Pengujian Kuisioner dengan masyarakat

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pertanyaan | SS | | S | | C | | KS | | | TS | |
| total | % | total | % | total | % | total | % | | total | % |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |
| Rata-rata | 0% | | 0% | | 0% | | 0% | | 0% | | |

Dapat dilihat pada Gambar 4.47 yang merupakan grafik persentase rata-rata responden yang telah menggunakan Aplikasi SITABAH, dapat disimpulkan bahwa 00% setuju, 00% sangat setuju terkait dengan kemudahan dalam mengakses aplikasi, tampilan mudah di fahami, aplikasi berjalan dengan baik, membantu pengguna dalam menemukan informasi mengenai tempat ibadah dengan mudah.

Gambar 4. 1. Grafik Perentase rata-rata responden pengguna aplikasi SITABAH

## BAB V

## PENUTUP

1. **Kesimpulan**

Berdasarkan penelitian dan pembangunan sistem yang telah dilakukan, dapat ditarik beberapa kesimpulan diantaranya:

1. Sistem yang dibuat dapat memberikan informasi tentang data tempat ibadah di Kota Mataram berbasis *android* dan data tempat ibadah se-NTB berbasis *web* yang dapat memberikan jalur perjalanan menuju tempat ibadah yang diiginkan dan dapat mengetahui persebaran lokasi tempat ibadah yang ada di provinsi NTB.
2. Dapat membantu mempermudah Kementerian Agama Provinsi NTB dalam melakukan pengelolaan terhadap data tempat ibadah yang ada.
3. Metode *Extreme Programming* melibatkan *client* langsung dalam proses perancangan dan pembangunan sistem sehingga berdasarkan hasil pengujian yang telah dilakukan, dapat dikatakan sistem yang dibuat telah sesuai dengan kebutuhan *client.*
4. Hasil pengujian *black box* menunjukkan bahwa seluruh fungsi bekerja dengan baik pada system dan system yang dibuat telah berjalan sesuai dengan fungsi yang dirancang.
5. **Berdasarkan hasil pengujian dengan menggunakan kuesioner, menunjukkan bahwa rata-rata penilaian responden pengguna terhadap sistem menyatakan setuju dan sangat setuju masing – masing bernilai 45.60% dan 22,83%**
6. **Saran**

Dikarenakan masih banyak kekurangan dari penulis dalam mengembangkan sistem ini, terdapat beberapa saran untuk dapat membuat sistem ini menjadi lebih baik pada waktu yang akan datang, diantaranya:

1. Sebaiknya aplikasi pada basis *android* tidak hanya diterapkan di Kota Mataram saja, melainkan dapat diterapkan di provinsi dan di kota-kota lainnya.
2. *Device* yang digunakan dalam membuat sistem harus diperhatikan terlebih dahulu, karena dalam XP jika terjadi kerusakan *device* pada saat proses pembuatan sistem berlangsung, maka akan mengganggu estimasi atau membuat target *release* system menjadi lebih lama dari yang telah direncanakan.
3. Sebaiknya sistem yang dibuatdisertai dengan fitur dua Bahasa, yaitu Bahasa Inggris dan Bahasa Indonesia untuk membantu pengguna sistem yang tidak berasal dari Indonesia
4. Pada menu map, sebaiknya ditambahkan *polygon* atau *polyline* pada setiap kabupaten atau kecamatan menjadikan desain *interface* yang lebih optimal dan menjadi sistem informasi yang lebih efektif dalam pembacaan wilayahnya.

**DAFTAR PUSTAKA**

[1] *Kamus Besar Bahasa Indonesia*. Jakarta: PUSAT BAHASA DEPARTEMEN PENDIDIKAN NASIONAL, 2008.

[2] Komite Teknis Perumusan Standar Pelayanan Masyarakat pada Kementerian Lingkungan Hidup dan Kehutanan, “Standar Pelayanan Masyarakat pada Fasilitas Publik Standar.” 2016.

[3] M. Agama and D. Negeri, “Pedoman Pelaksanaan Tugas Kepala Daerah/Wakil Kepala Daerah Dalam Pemeliharaan Kerukunan Umat Beragama, Pemberdayaan Forum Kerukunan Umat Beragama, Dan Pendirian Rumah Ibadat,” 2006.

[4] G. S. Perdana, “Sistem informasi geografis tempat olahraga di provinsi daerah istimewa yogyakarta berbasis web,” 2017.

[5] Sukatmi and F. A. Ani, “Membangun Aplikasi Webgis Untuk Tempat Ibadah Di Provinsi Lampung,” *Cendikia*, vol. XVI, pp. 115–119, 2018.

[6] A. Supriyatna, “Metode Extreme Programming Pada Pembangunan Web Aplikasi Seleksi Peserta Pelatihan Kerja,” *J. Tek. Inform.*, vol. 11, no. 1, pp. 1–18, 2018, doi: 10.15408/jti.v11i1.6628.

[7] R. Pamungkas, *Implementasi Model Personal Extreme Programming (Pxp) Dalam Pengembangan Sistem Informasi Geografis Pemetaan Flora Dan Fauna Di Taman Nasional Meru Betiri*. 2018.

[8] D. Y. Prasetyo and Apriyanto, “Implementasi Geographic Information System (GIS) Penentuan Tampat Ibadah Masjid Di Kecamatan Kempas Kabupaten Indragiri Hilir Provinsi Riau,” *SISTEMASI*, vol. 8, pp. 10–18, 2019.

[9] N. Alamsyah and Wagino, “Sistem Informasi Geografis Persebaran Tempat Ibadah Dan Kapasitasnya Berdasarkan Jumlah Penduduk Di Kota Banjarmasin,” *Technologia*, vol. 9, no. 2, pp. 133–138, 2018.

[10] C. PARAMITA, “Sistem informasi masjid di surakarta,” 2016.

[11] S. Maharani, D. Apriani, A. H. Kridalaksana, P. Studi, I. Komputer, and U. Mulawarman, “Sistem informasi geografis pemetaan masjid di samarinda berbasis web,” vol. 11, pp. 9–20, 2017.

[12] W. N. NUGROHO, “Aplikasi Pencarian Masjid Terdekat Di Kota Bandar Lampung Berbasis Mobile Menggunakan Algoritma Dijkstra,” 2017.

[13] R. S. HAMSYAH, “Rancang Bangun Aplikasi Go-Ban Untuk Mencari Dan Memanggil Teknisi Tambal Ban Menggunakan Google Maps Api,” 2018.

[14] Suhartini, M. Sadali, and Y. K. Putra, “Sistem Informasi Berbasis Web Sma Al- Mukhtariyah Mamben Lauk Berbasis Php Dan Mysql Dengan Framework Codeigniter,” *infotek*, vol. 3, no. 1, pp. 79–84, 2020.

[15] M. Arifin and R. H. H. Hs, “Perancangan Sistem Informasi Pusat Karir Sebagai Upaya Meningkatkan Relevansi Antara Lulusan Dengan Dunia Kerja Menggunakan UML,” *IC-Tech*, vol. XII, no. 2, pp. 42–49, 2017.

[16] T. Sutabri, *KONSEP SISTEM INFORMASI*. Yogyakarta, 2012.

[17] A. Solichin, “Pemrograman Web dengan PHP dan MySQL,” pp. 1–122.

[18] B. Utama, “Aplikasi Peminjaman Dan Pengembalian Barang Menggunakan Rfid Dan Barcode Scanner Berbasis Web,” 2019.

[19] T. Kahlert and K. Giza, “Visual Studio Code - Code Editing. Redefined,” *Mikrosoft*, vol. 1, no. March, pp. 1–26, 2016.

[20] A. Krismadi, A. F. Lestari, A. Pitriyah, I. W. P. A. Mardangga, M. Astuti, and A. Saifudin, “Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 155, 2019, doi: 10.32493/jtsi.v2i4.3771.

[21] W. Widyantoro, “Buku Panduan Bootstrap,” 2015.

[22] M. Lenz, “Postman,” *Python Contin. Integr. Deliv.*, pp. 1–12, 2019, doi: 10.1007/978-1-4842-4281-0\_1.

[23] M. Ichwan and F. Hakiky, “Pengukuran Kinerja Goodreads Application Programming Interface (API) Pada Aplikasi Mobile Android (Studi Kasus Untuk Pencarian Data Buku),” vol. 2, no. 2, pp. 13–21, 2011.

[24] J. Andi, “Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A-GPS ) Dengan Platform Android,” *J. Ilm. Komput. dan Inform.*, vol. 1, no. 1, pp. 1–8, 2015.

[25] P. Abrahamson, Outi Salo, Jussi Ronkainen, and Juhani Warsta, “Agile software development methods: Review and analysis,” *VTT Publ.*, p. 112, 2002.

[26] A. Prasetya, A. H. Brata, and M. T. Ananta, “Pengembangan Aplikasi Pemesanan Lapangan Futsal Di Kota Malang Berbasis Android Menggunakan Metode Pengembangan Extreme Programming (Studi Kasus Champion Tidar, Zona SM Futsal, dan Viva Futsal),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 7293–7301, 2018.

[27] A. Lisna, “Extreme Programming,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2016, doi: 10.1017/CBO9781107415324.004.

[28] M. Ulfi, G. I. Marthasari, and N. Ilyas, “Implementasi Metode Personal Extreme Programming dalam Pengembangan Sistem Manajemen Transaksi Perusahaan (Studi Kasus: CV. Todjoe Sinar Group),” vol. 2, no. 3, pp. 261–268, 2020.

[29] R. Agarwa and D. Umphress, “Extreme programming for a single person team,” *Proc. 46th Annu. Southeast Reg. Conf. XX, ACM-SE 46*, no. August, pp. 82–87, 2008, doi: 10.1145/1593105.1593127.

[30] R. A. Azdy and A. Rini, “Penerapan Extreme Programming Dalam Membangun Aplikasi Pengaduan Layanan Pelanggan (Palapa) Pada Perguruan Tinggi,” vol. 5, no. 2, pp. 197–206, 2018, doi: 10.25126/jtiik.201852658.

[31] M. A. F. Z and A. Wibowo, “Aplikasi gameplay edukasi pencegahan obesitas dengan menggunakan algoritma astar dan greedy pada pencarian jalur makanan,” vol. 10, no. 2016.

[32] A. Mu, “BAB III, metodologi penelitian,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019, doi: 10.1017/CBO9781107415324.004.

[33] A. N. A. Thohari and A. E. Amalia, “Implementasi Test Driven Development Dalam Pengembangan Aplikasi Berbasis Web,” *SITECH J. Sist. Inf. dan Teknol.*, vol. 1, no. 1, pp. 1–10, 2018, doi: 10.24176/sitech.v1i1.2255.

[34] S. Masripah and L. Ramayanti, “Penerapan Pengujian Alpha Dan Beta Pada Aplikasi Penerimaan Siswa Baru,” *J. Swabumi*, vol. 8, no. 1, pp. 100–105, 2020.

[35] A. B. Mutiara, U. Gunadarma, A. Muslim, U. Gunadarma, T. Oswari, and U. Gunadarma, “Testing Implementasi Website Rekam Medis Elektronik Opeltgunasys Dengan Metode Acceptance Testing,” no. February 2016, 2014, doi: 10.13140/RG.2.1.1159.5286.

[36] R. A. Lestari, “Pengaruh Kepemimpinan Partisipatif D an Komitmen Organisasi Terhadap Efektifitas Implementasi Rencana Stratejik Pada Madrasah Aliyah D i Kabupaten Sukabumi,” pp. 1–25, 2014.

[37] A. Suandi, F. N. Khasanah, and E. Retnoningsih, “Pengujian Sistem Informasi E-commerce Usaha Gudang Cokelat Menggunakan Uji Alpha dan Beta,” *Inf. Syst. Educ. Prof.*, vol. 2, no. 1, pp. 61–70, 2017.

**LAMPIRAN**

Surat keterangan telah melakukan wawancara :

