IMPLEMENTASI PRINSIP KERJA EXTREME PROGRAMMING

1. Planning

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

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

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

Tabel penambahan user stories

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

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

Tabel hasil perencanaan iterasi

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

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

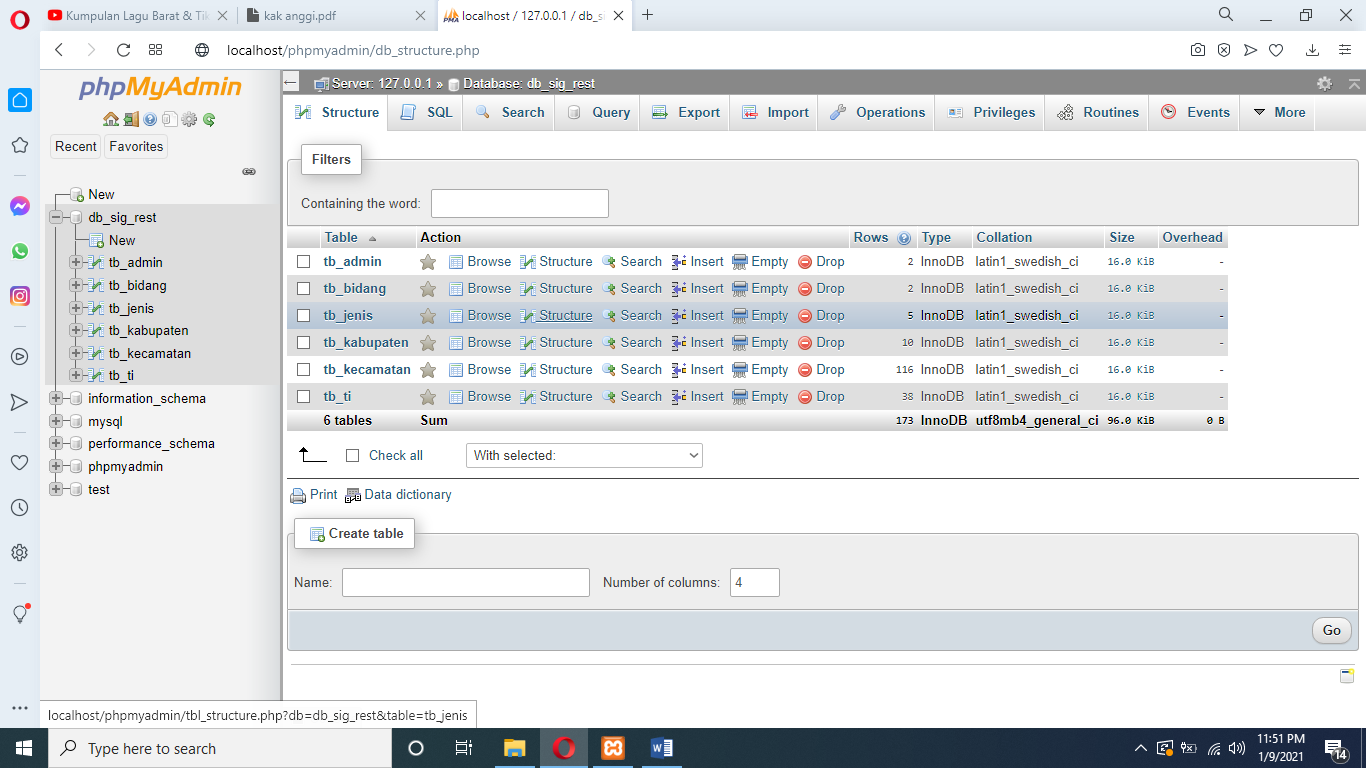
1. Design
2. Coding

Pada tahap coding ini

1. **Implementasi system**

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

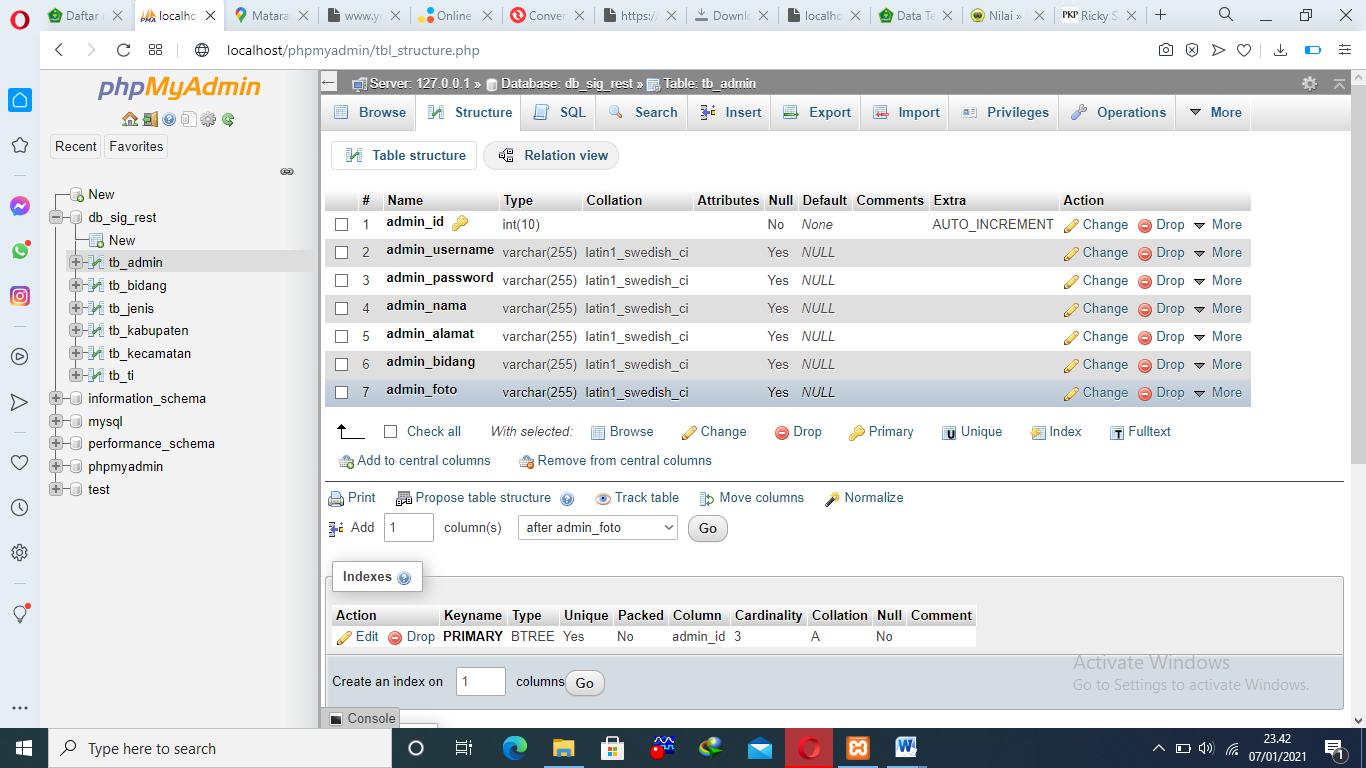
1. **Implementasi *database***



**Gambar 4.1** Implementasi *database*

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

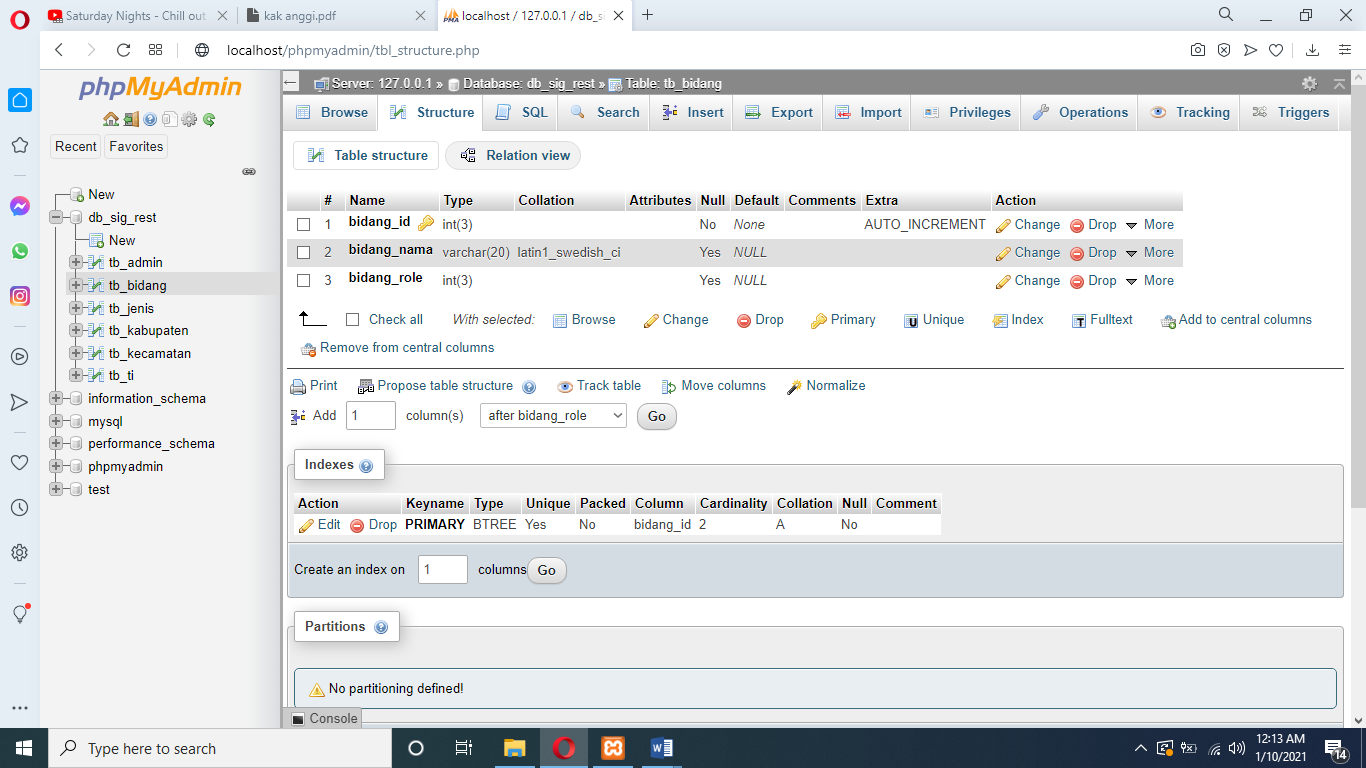
1. Tabel tb\_admin



**Gambar 4.2** Implementasi tabel tb\_admin

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

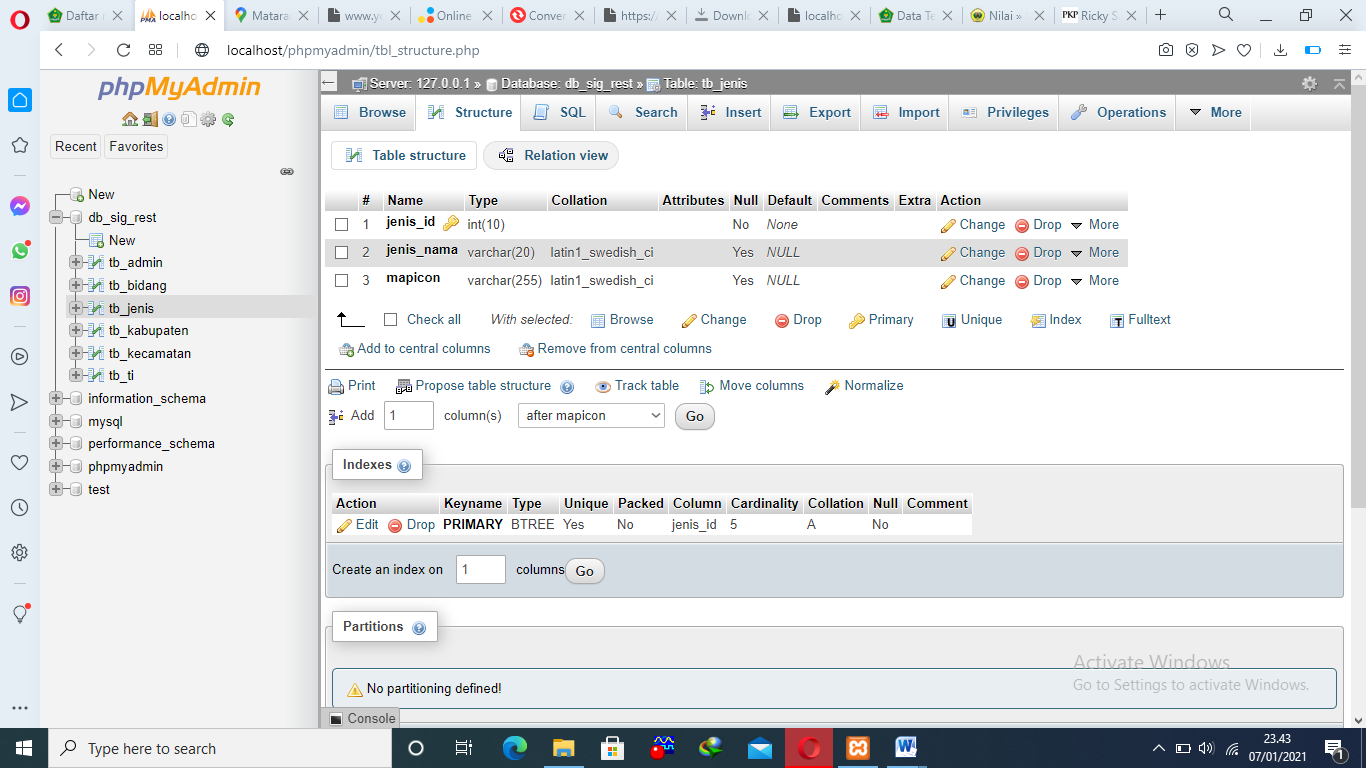
1. Tabel tb\_bidang



**Gambar 4.3** Implementasi tabel tb\_bidang

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

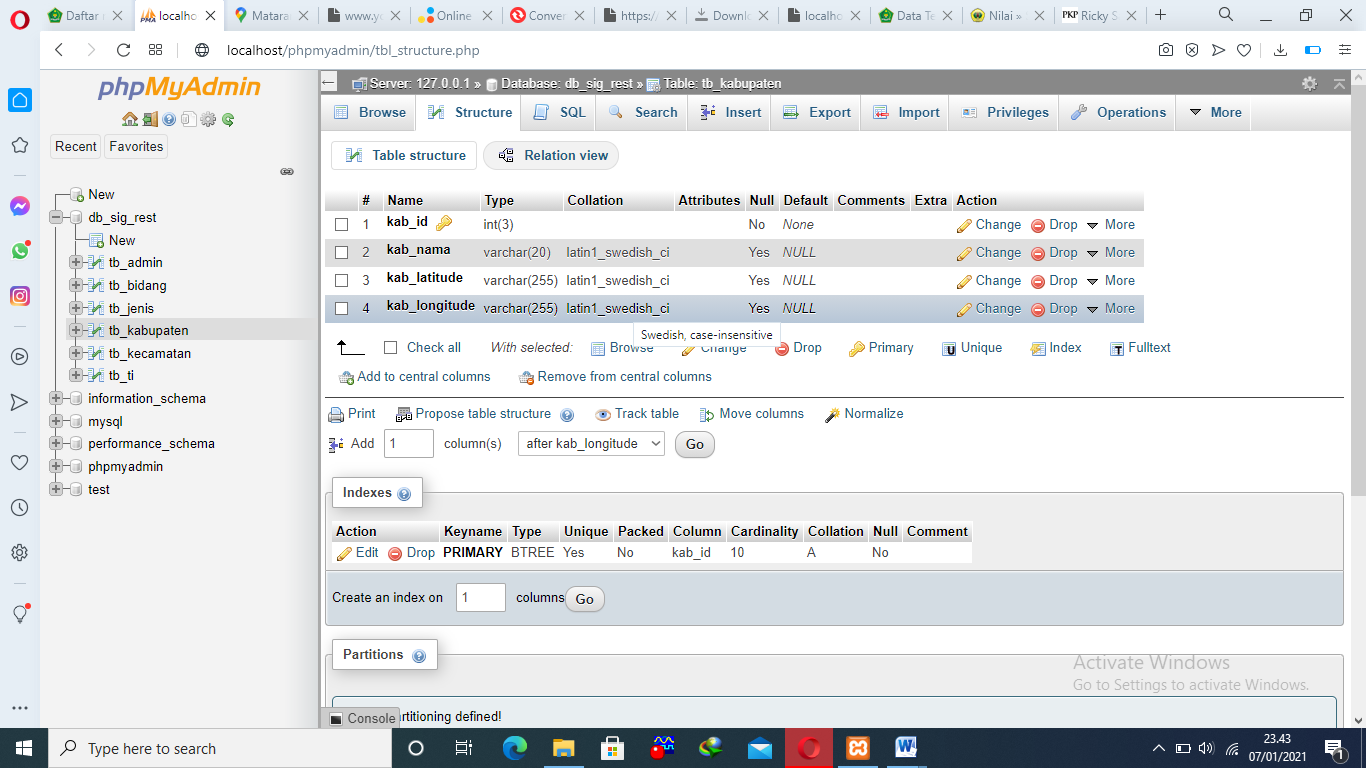
1. Tabel tb\_jenis



**Gambar 4.4** Implementasi tabel tb\_jenis

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

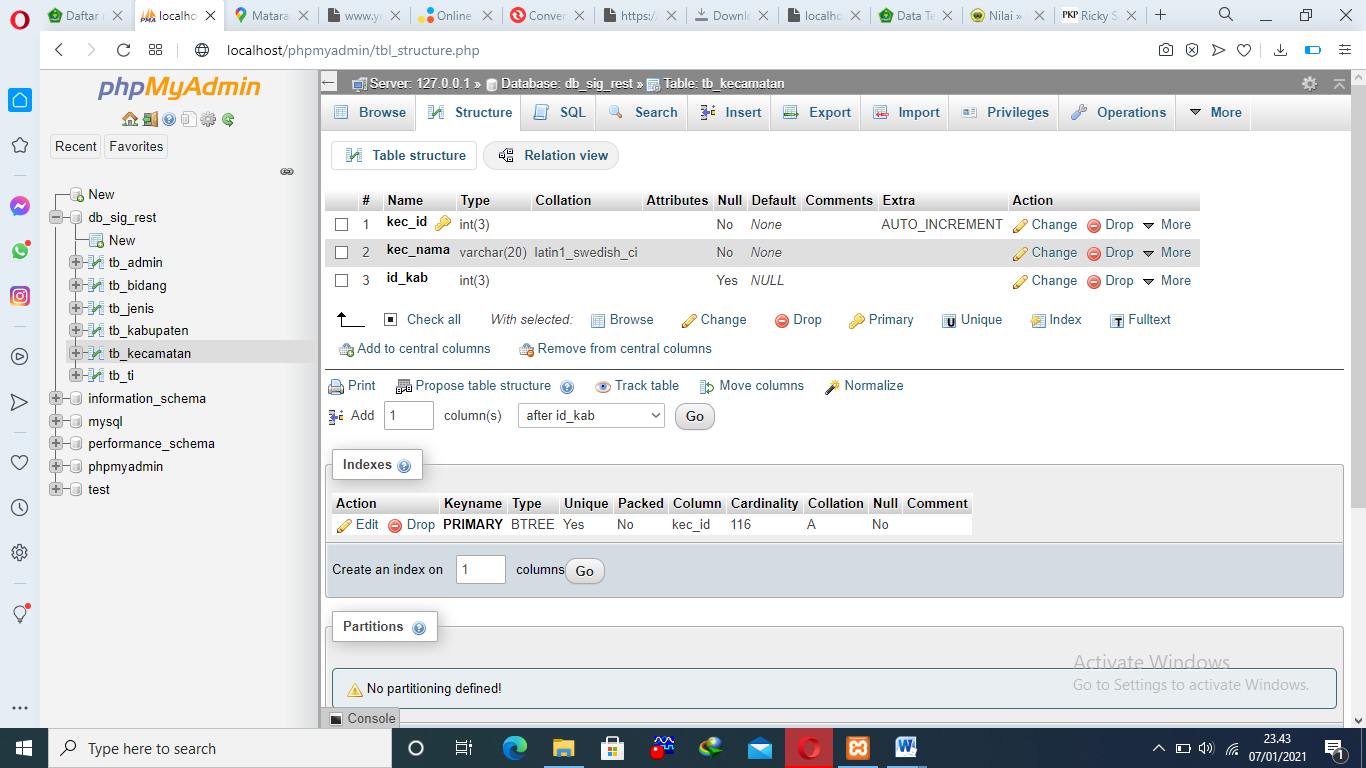
1. Tabel tb\_kabupaten



**Gambar 4.5** Implementasi tabel tb\_kabupaten

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

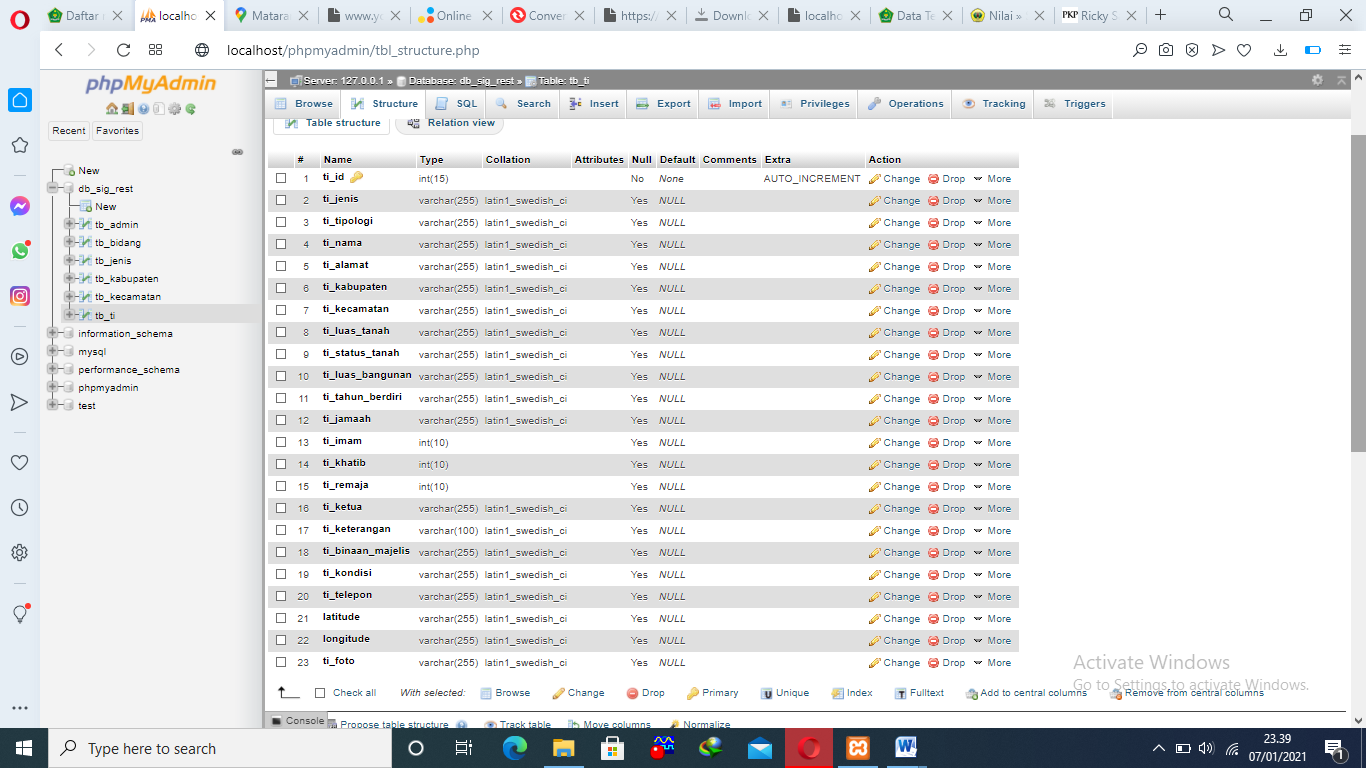
1. Tabel tb\_kecamatan



**Gambar 4.6** Implementasi tabel tb\_kecamatan

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

1. Tabel tb\_ti



**Gambar 4.7** Implementasi tabel tb\_ti

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

1. **Implementasi API**

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

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

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

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

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

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

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

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

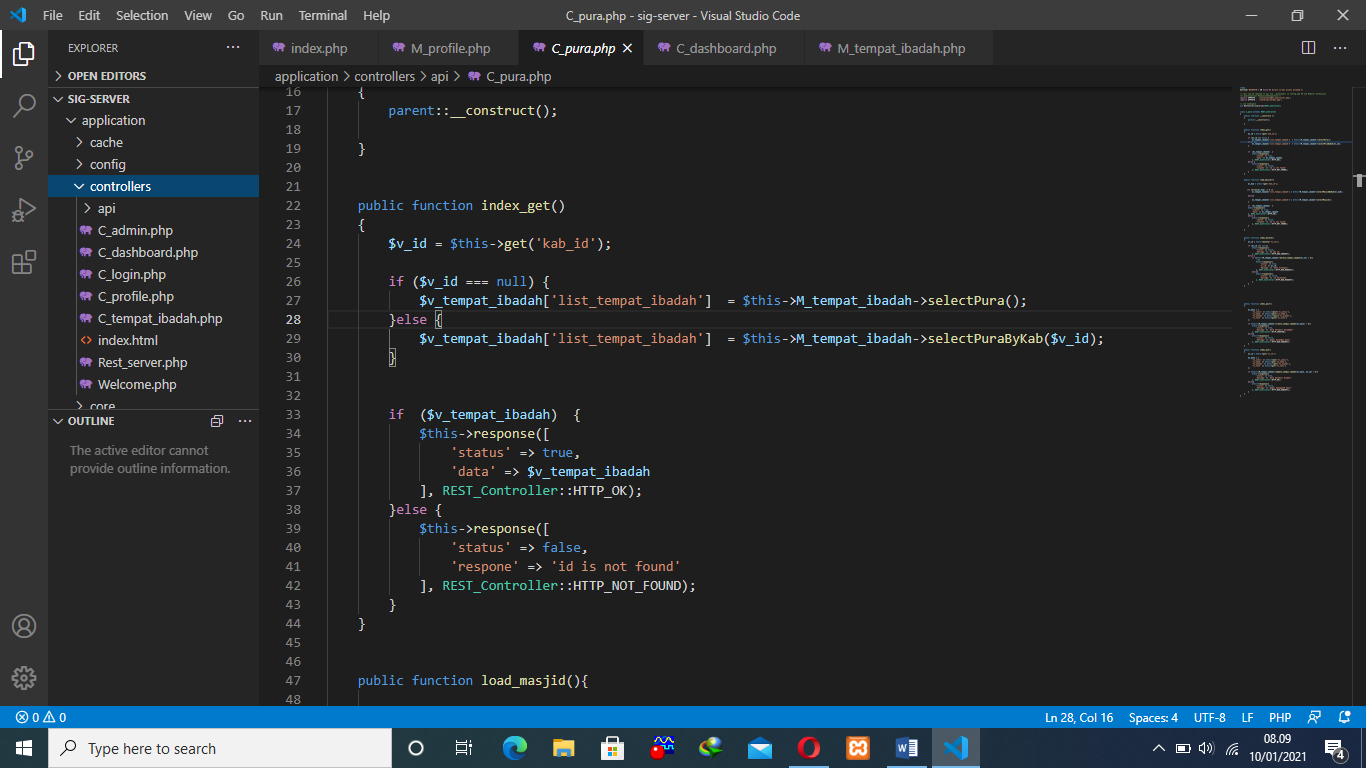
1. **Implementasi *class***

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

1. **Basis *web***

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

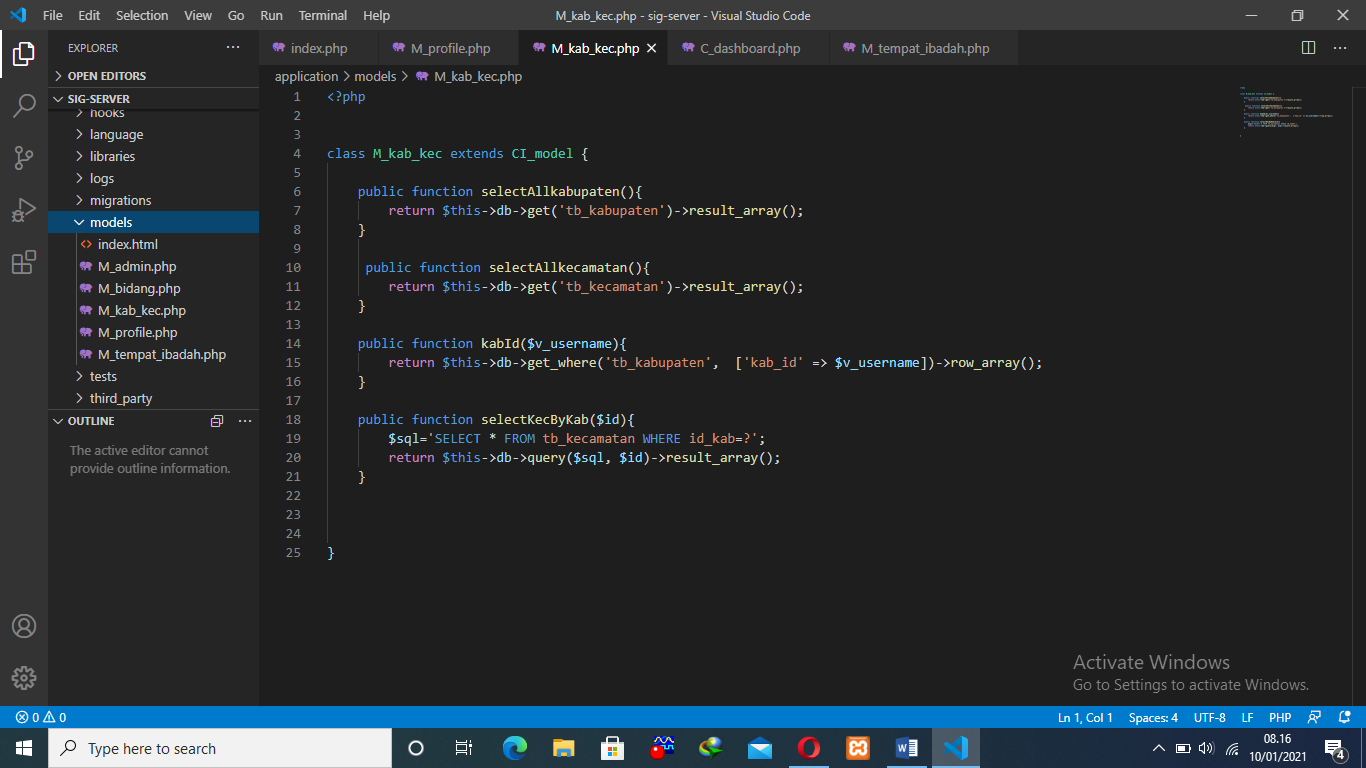
1. *Class controller*



**Gambar 4.8** Implementasi *class controller*

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

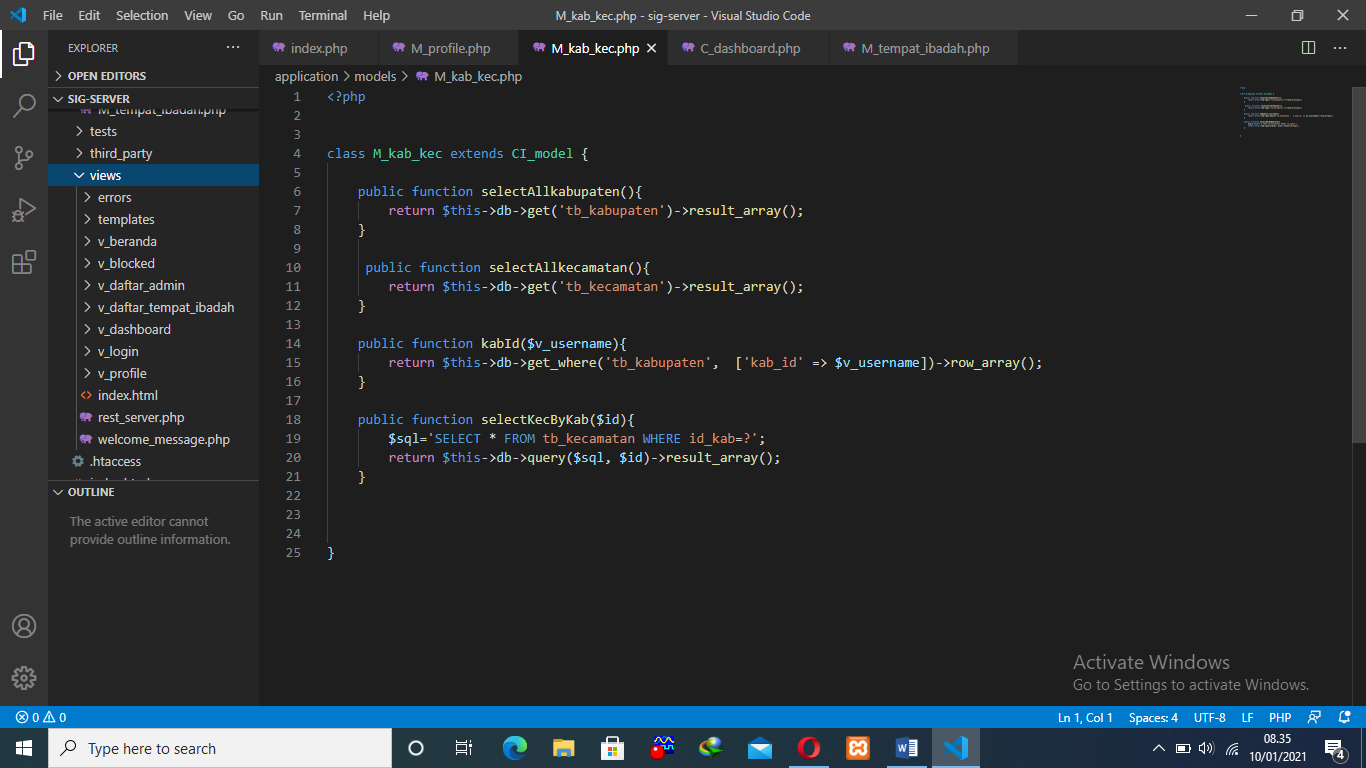
1. *Class model*



**Gambar 4.9** Implementasi *class model*

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

1. *Class view*



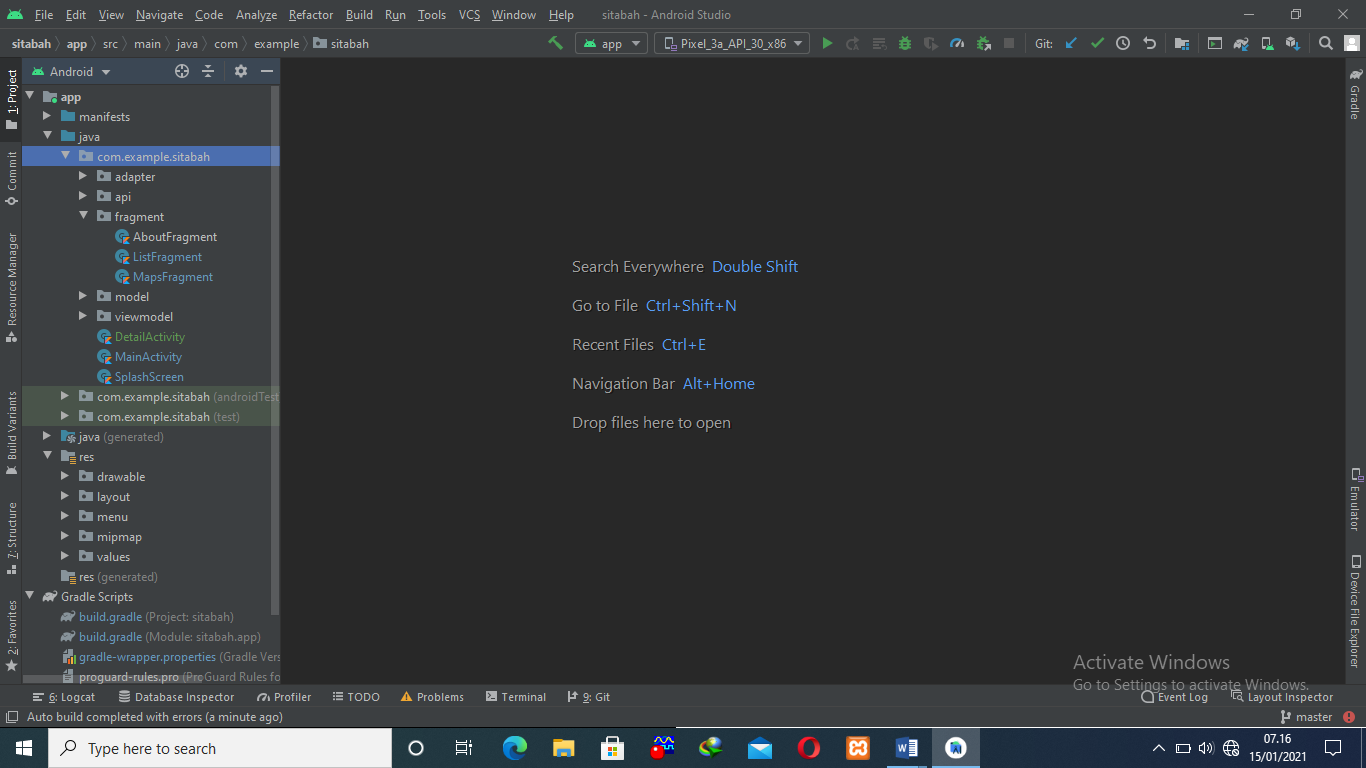
**Gambar 4.10** Implementasi *class view*

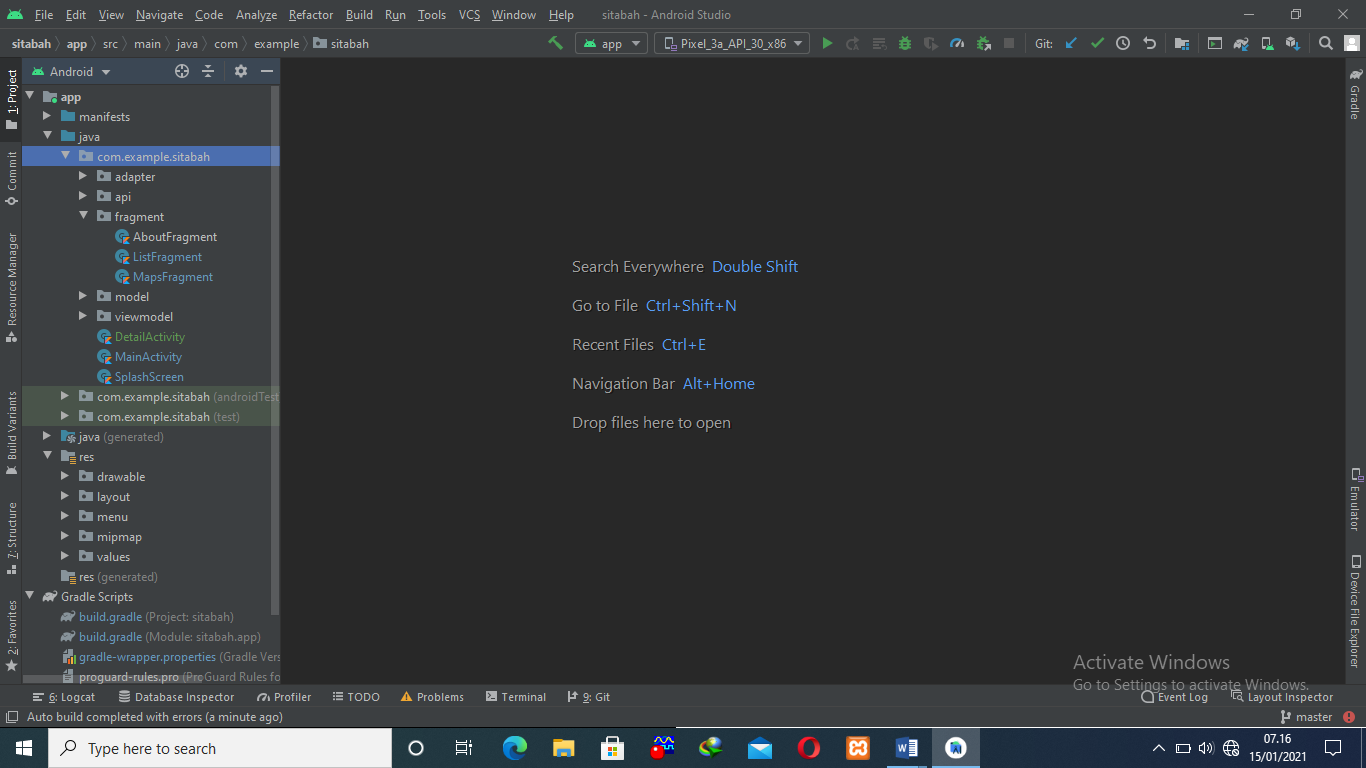
Gambar 4.10 merupakan implementasi *class* *view* yang terdiri dari V\_beranda, V\_daftar\_admin, V\_daftar\_tempat\_ibadah, V\_dashboard, V\_login, dan V\_profile. Dimana *class* ini nantinya bertugas untuk menyajikan informasi. sesuai dengan intruksi dari *controller.*

1. **Basis *android***

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

1. *Class activity*

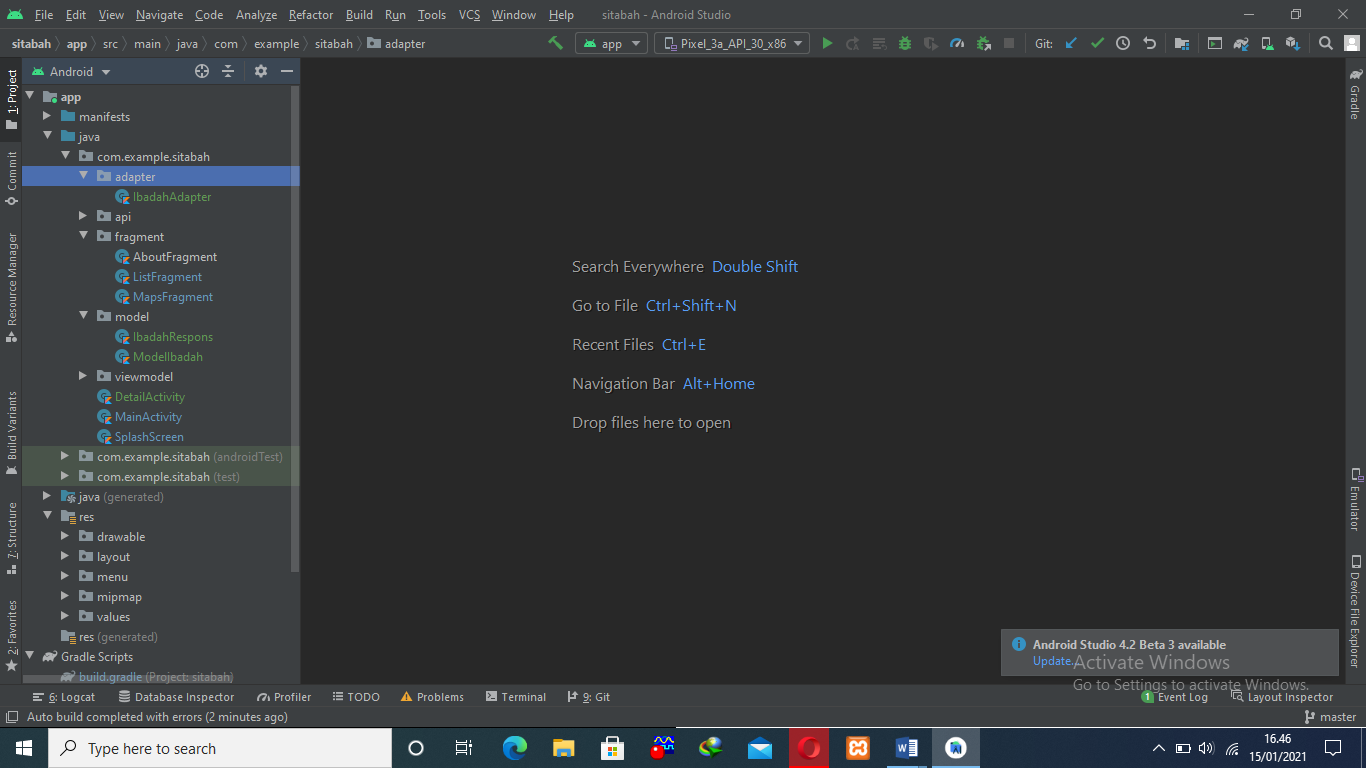




**Gambar 4.11** Implementasi *class activity*

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

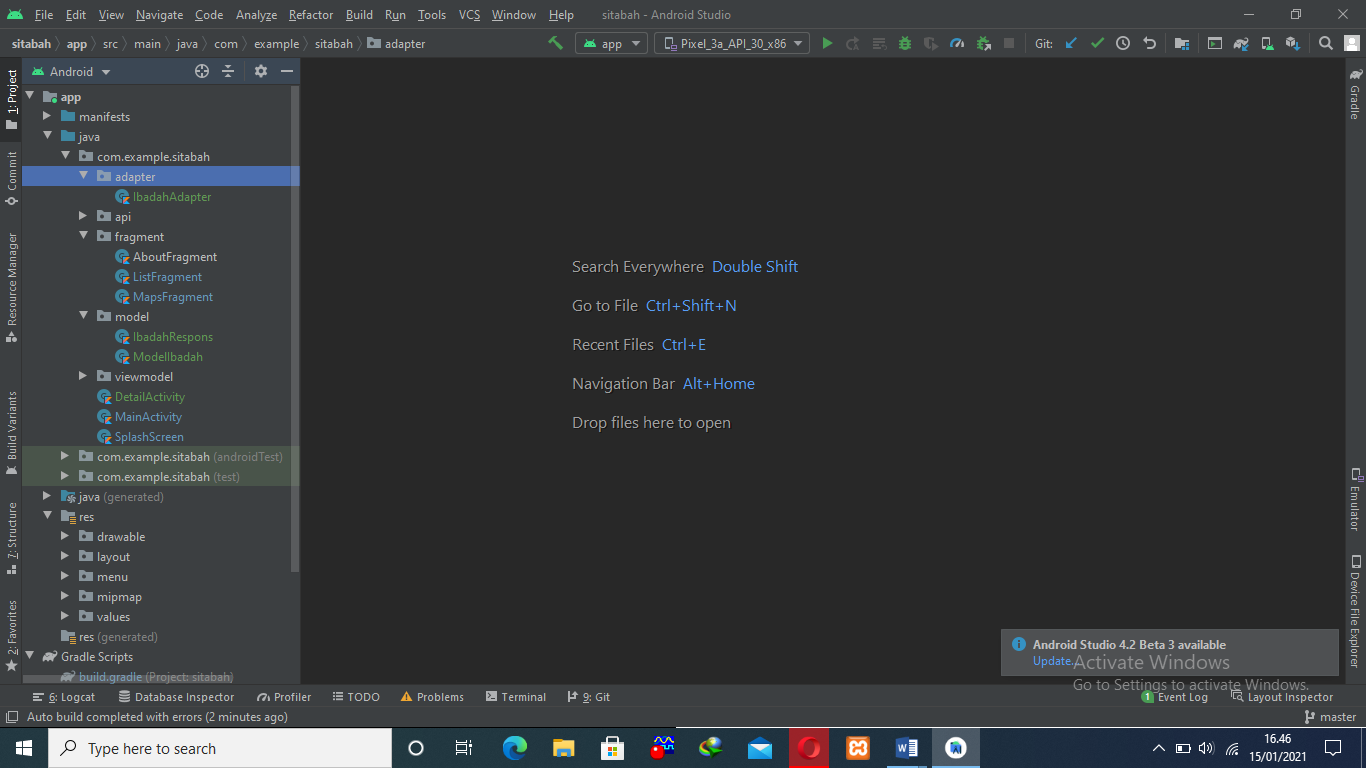
1. Adapter (Kotlin)



**Gambar 4.12** adapter

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

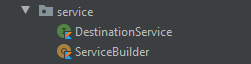
1. Model (Kotlin)



**Gambar 4.13** Implementasi *class model*

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

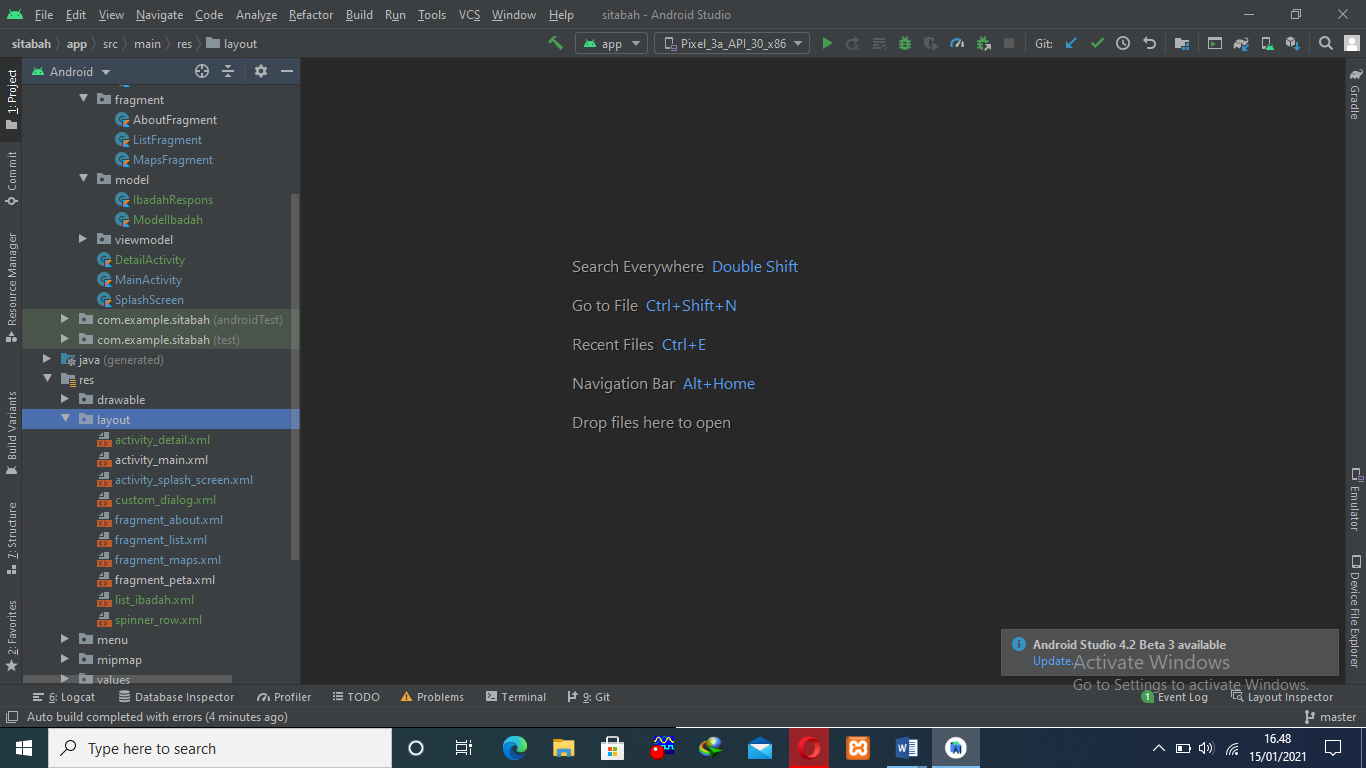
1. Service (Kotlin)



**Gambar 4.14** Implementasi *class service*

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

1. Layout (XML)



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

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

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

EVALUASI HASIL METODE EXTREME PROGRAMMING