Kata **Gereja** berasal dari kata dalam bahasa Yunani “Ekklesia” yang didefinisikan sebagai “perkumpulan” atau “orang-orang yang dipanggil keluar.” Akar kata ”Gereja” tidak berhubungan dengan gedung, tetapi dengan orang. Roma 16:5 berkata, “Salam juga kepada jemaat di rumah mereka...” Paulus merujuk pada Gereja di rumah mereka, bukan pada gedung gereja, namun kumpulan orang-orang percaya. Gereja itu tubuh Kristus. Efesus 1:22-23 mengatakan, “Dan segala sesuatu telah diletakkan-Nya di bawah kaki Kristus dan Dia telah diberikan-Nya kepada jemaat sebagai Kepala dari segala yang ada. Jemaat yang adalah tubuh-Nya, yaitu kepenuhan Dia, yang memenuhi semua dan segala sesuatu.”. Gereja tidak hanya berbicara mengenai sebuah bangunan akan tetapi perkumpulan orang-orang, semakin besarnya sebuah gereja diperlukannya membuat **penyebaran informasi** akan semakin sulit sehingga diperlukannya sebuah **sistem informasi** yang berguna membantu penyebaran informasi kepada jemaat gereja secara lebih maksimal. Adapun sistem informasi ini akan dibangun berbasis **web application** sehingga informasi yang diberikan tidak hanya secara statis akan tetapi dinamis.

**Web Application** adalah sebuah aplikasi berbasis website yang berbasis perangkat lunak dan berjalan pada browser internet atau melalui halaman web. Web application ini akan dikodekan menggunakan bahasa pemrograman yang mendukung perangkat lunak berbasis web seperti **PHP** dengan bantuan framework **Laravel**, **Javascript** dengan bantuan library **jQuery, HTML2Canvas, SweetAlert2, FontAwesome, TinyMCE** dan konsep **AJAX**, dan **CSS** dengan bantuan framework **Boostrrap** beserta **HTML**.

Adapun aplikasi yang digunakan untuk membangun web application ini adalah

* Visual Studio Code
* XAMPP (PHP > 7.4)
* Git bash
* Composer
* Google Chrome
* Draw.io
* Figma

Tahapan pembuatan web application ini :

* Analisis kelayakan dan kebutuhan website
* Analisis dan tentukan keperluan website
* Desain (Proses Bisnis dan Pemrograman)
* Pengembangan
* Uji coba
* Pemeliharaan

**Analisis Kelayakan dan kebutuhan Website**

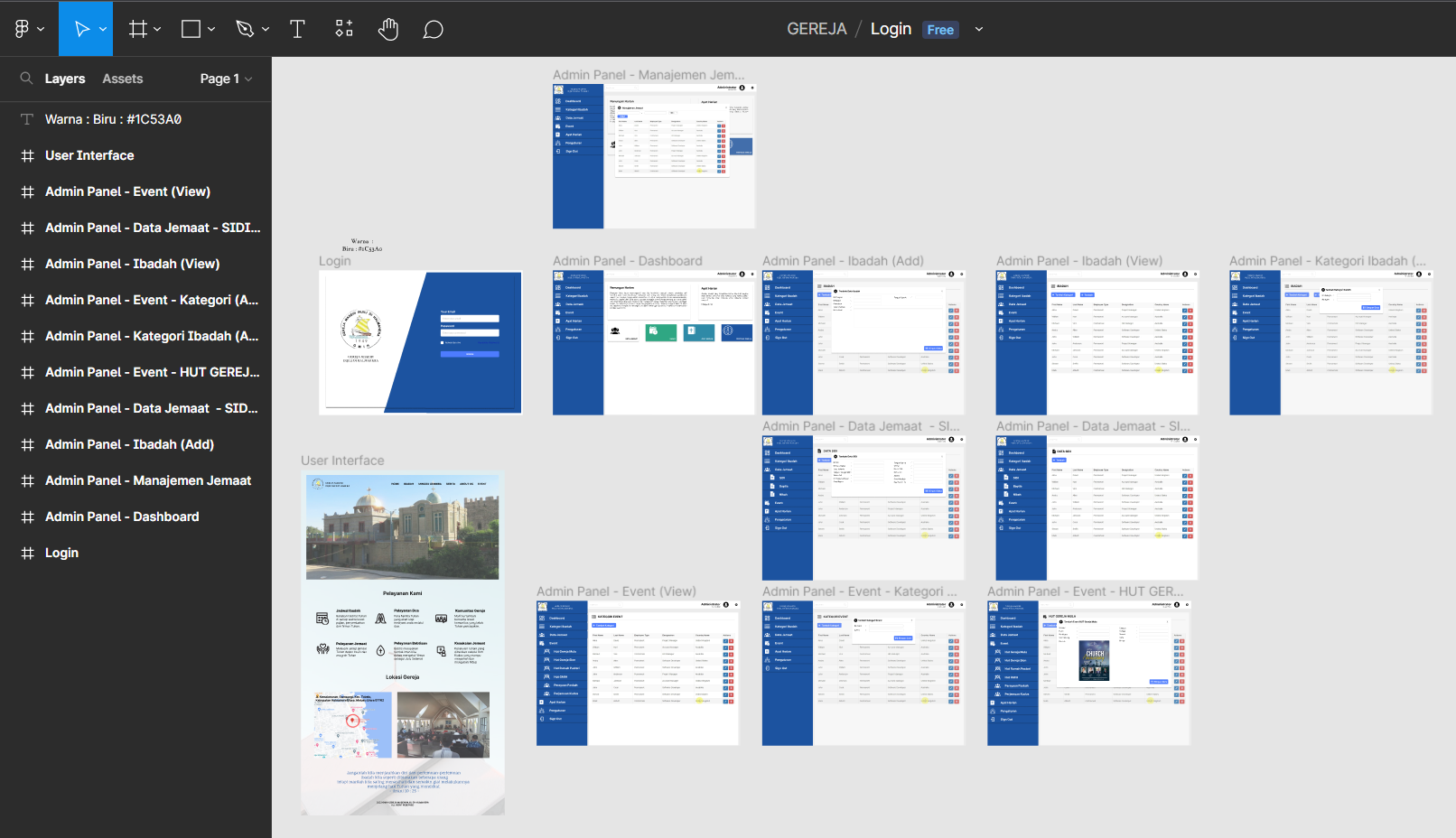
Pada langkah ini, akan dilakukan analisis kelayakan dan kebutuhan *website* seperti fungsi *website*, biaya *website*, arah kegunaan *website*, tujuan *website*, dan sasaran dari *website* sehingga pada tahap ini akan menjadi batu pijakan awal dalam pembuatan sebuah *web application*. Pemahaman mengenai alur aplikasi, aturan , fungsionalitas sangatlah diperlukan pada tahapan ini.

**Analisis dan tentukan keperluan website**

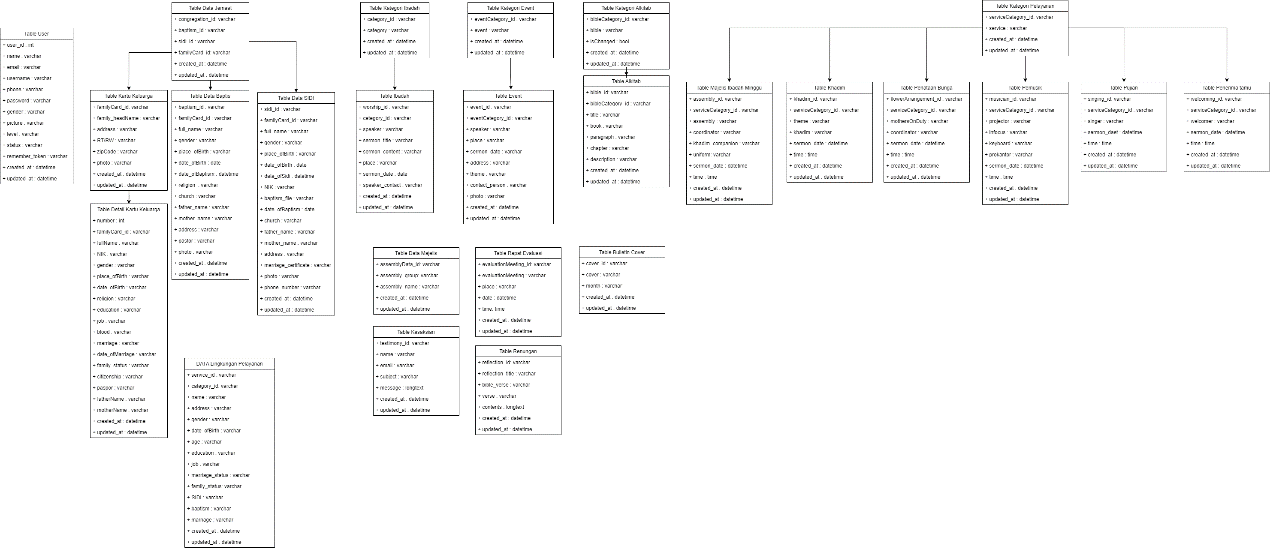
Pengembangan sebuah *website* tak luput dan adanya persiapan keperluan dalam pembangunan sebuah *website*, infrastruktur *website* akan dirancang pada tahapan ini sehingga alur kerja dan arah pergerakan *website* akan terdeteksi pada langkah ini. Perangkat yang digunakan, lingkup kerja keinginan *user*, keperluan *user* hingga hal lainnya akan dimulai pada tahapan ini.

**Desain**

* Perencangan **tampilan** website dilakukan dengan menggunakan bantuan aplikasi ***figma*** yang akan menghasilkan sebuah **prototipe** kepada user sehingga tampilan desain yang diberikan dapat diklik sehingga user dapat melihat response dari web application yang akan dibangun. Perancangan tampilan ini akan menjadi penentu dalam model web application yang akan dibangun mulai dari warna, tata letak, gambar, model hingga jenis tulisan yang digunakan.



* Perancangan **database** pada web app ini menggunakan bantuan aplikasi **Draw.io** yang berfungsi dalam melakukan sebuah database mulai dalam menentukan nama table, kolom pada table hingga relasi antar satu table dengan table lainnya.



**Pengembangan**

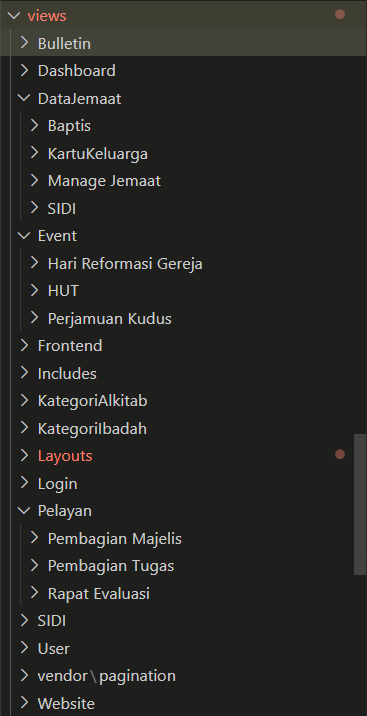
Tahapan pengembangan menjadi tahapan yang penting dimana sebuah web application akan mulai dibangun menggunakan bahasa pemrograman. Tahapan ini akan mulai dikerjakan dengan menulis program ke komputer menggunakan bahasa yang sudah disepakati diawal yakni PHP sebagai bahasa *server side.* Pengerjaan sebuah program akan dikerjakan secara Back End side dan Front End side dimana Back End bertugas dalam merancang atau mengembangkan software di sisi server yang berkaitan dengan logika serta database. Sedangkan Front End side bertugas dalam komposisi tampilan sebuah web application mulai dari isi konten, warna, jenis tulisan, gambar, tombol dan lain sebagainya.

Berikut adalah tahapan pengerjaan Back End Side :

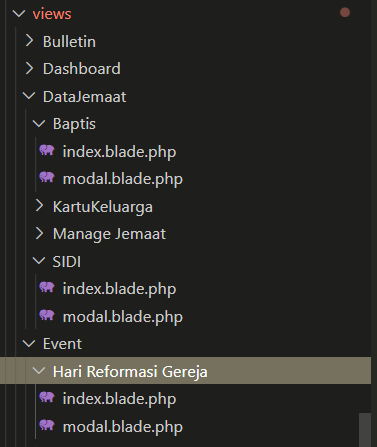
* Menginstall git bash
* Menginstall composer
* Menginstall PHP yang hendak digunakan menggunakan aplikasi XAMPP sebagai server
* Menginstall Laravel menggunakan git bash

Langkah – langkah pembuatan web application :

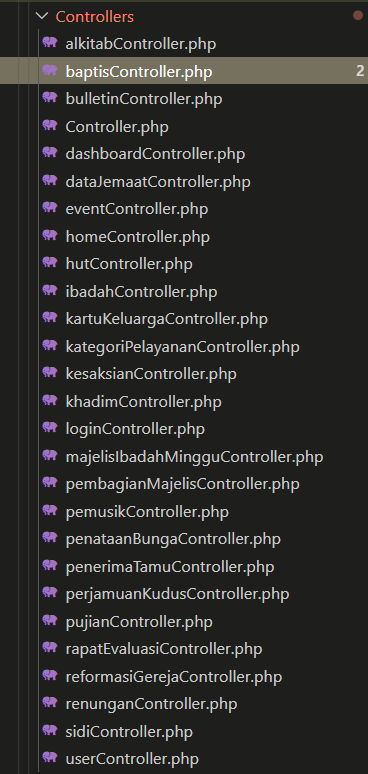
* Membuat folder dan subfolder berdasarkan nama tampilan pada web application



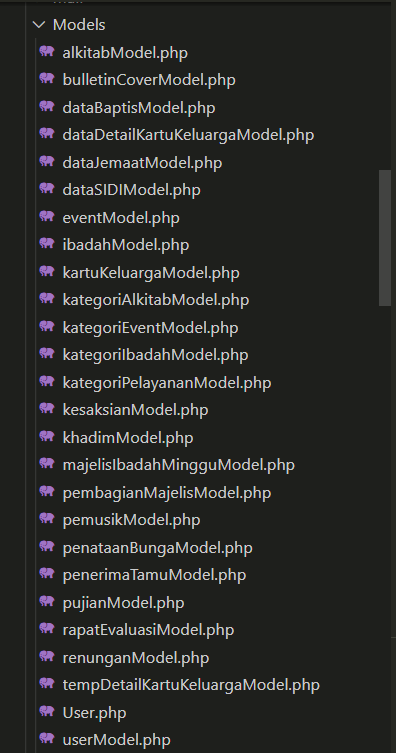
* Menentukan penamaan file pada subfolder yang ada guna memudahkan dalam pengerjaan



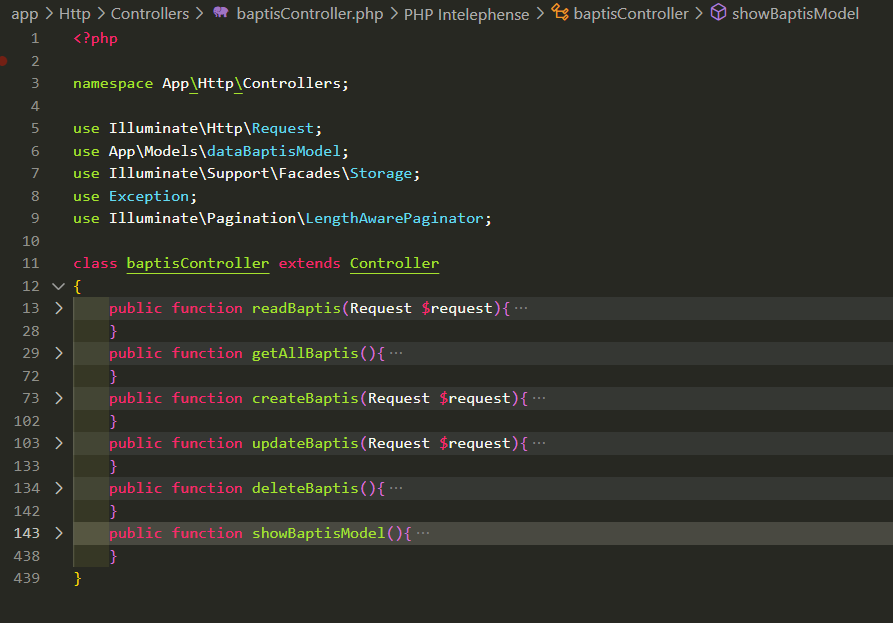
* Menentukan penamaan controller yang mudah sesuai dengan tampilan pada web application



* Menentukan penamaan model yang mudah sesuai dengan nama table yang sudah dibuat



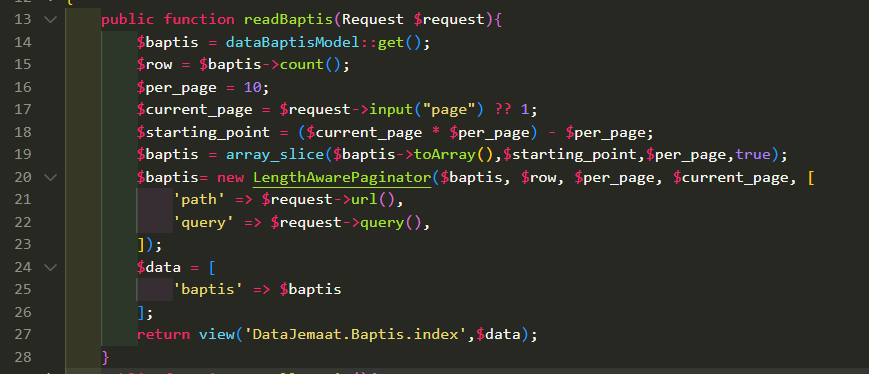
* Menentukaan penamaan fungsi di controller yang akan sejalan dengan penamaan routing



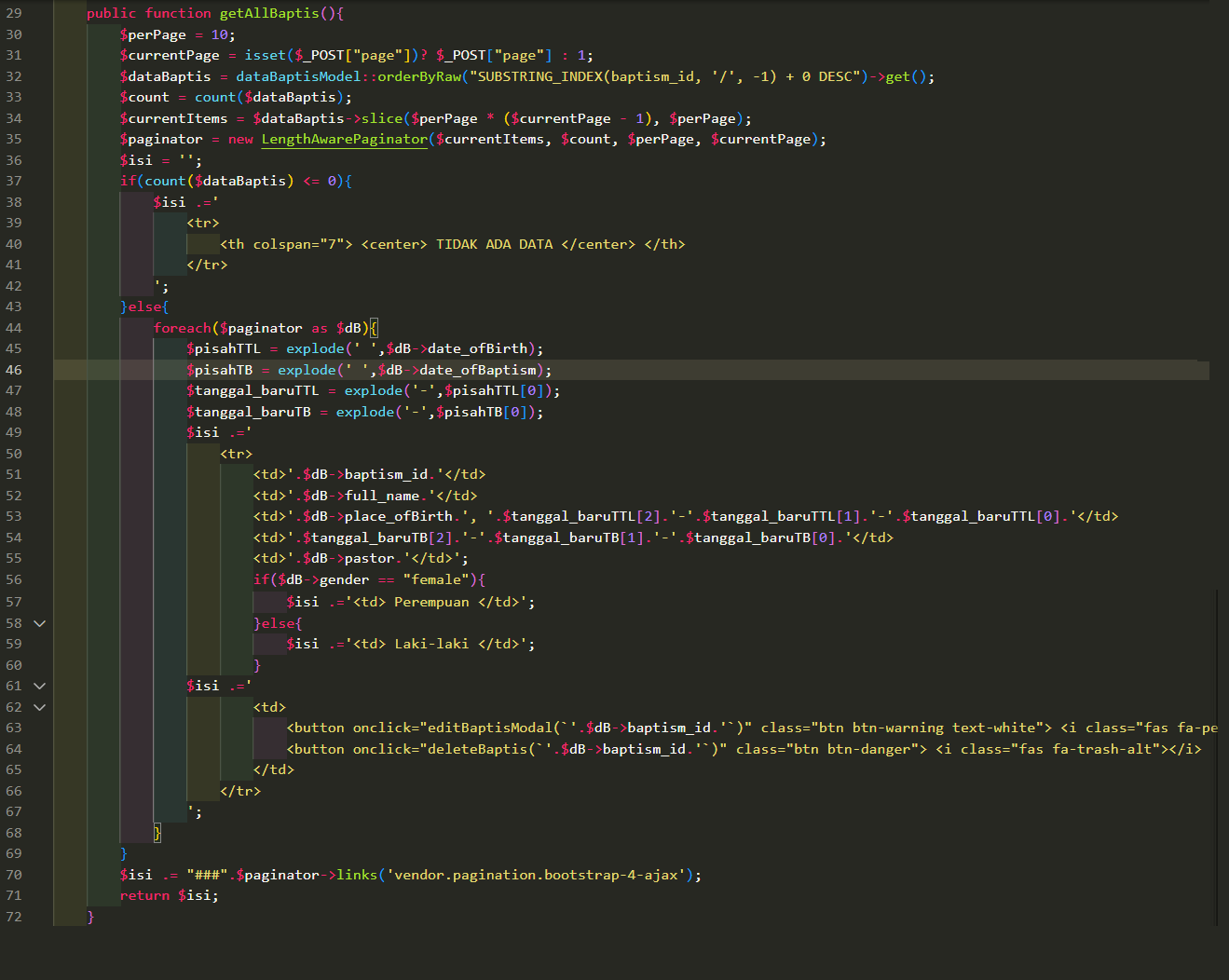
* Menentukan penamaan routing yang mudah dan sederhana sehingga tidak membuat ambigu dalam pengerjaan dengan membagi berdasarkan method yang digunakan mulai dari POST, PUT, GET dan DELETE



Contoh Penjelasan Coding :



|  |  |
| --- | --- |
| Line | Penjelasan |
| 13 | Membuat sebuah fungsi dengan tipe public dengan nama readBaptis yang menampung sebuah data dengan nama request pada paramaternya |
| 14 | Membuat variable dengan nama baptis yang berisikan pemanggilan semua data yang ada pada table baptis (SELECT \*FROM table\_baptis) |
| 15 | Membuat variable dengan nama row yang berisikan jumlah kolom pada table baptis |
| 16 | Membuat variable dengan nama per\_page yang berisikan nilai 10 |
| 17 | Membuat variable dengan nama current\_page yang berisikan nilai dari parameter dengan data yang diambil adalah data page dan dilakukan percabangan untuk mengecek apakah data page memiliki nilai atau tidak |
| 18 | Membuat variable dengan nama starting\_point yang beisikan rumus dari (variable current\_page \* variable per\_page) – variable per\_page |
| 19 | Memanggil variabel dengan nama baptis yang akan berisikan dengan fungsi array\_slice yang berfungsi untuk memotong data dari data variable baptis sebelumnya, nilai pertama untuk dikembalikan valuenya, panjang dari array variable baptis, menentukan array index pada return value |
| 20 | Memanggil variable dengan nama baptis yang akan berisikan dengan fungsi LengthAwarePaginator (sebuah fitur pada framework Laravel yang berfungsi untuk melakukan pagination) |
| 21 | Membuat data dengan nama path yang terkandung didalam variable baptis |
| 22 | Membuat data dengan nama query yang terkandung didalam variable baptis |
| 23 | Tutup kurung dari line 20 yang merupakan penutup data dari variable baptis |
| 24 | Membuat variable dengan nama data yang merupakan sebuah array |
| 25 | Membuat data dengan nama baptis yang berisikan isi dari variable baptis |
| 26 | Tutup kurung dari line 24 yang merupakan penutup dari variable data |
| 27 | Mengembalikan data kedalam bentuk view ke Folder DataJemaat, subfolder Baptis dan file Index , dimana data tersebut akan dikirimkan dalam sebuah object |
| 28 | Tutup kurung dari fungsi readBaptis |



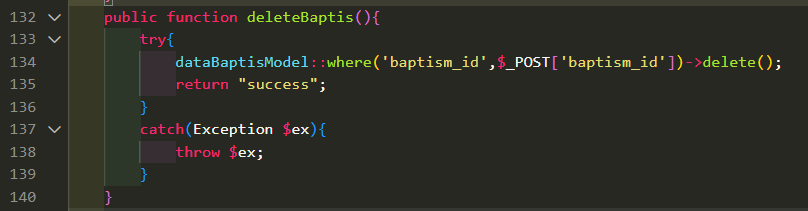
|  |  |
| --- | --- |
| Line | Penjelasan |
| 29 | Membuat sebuah fungsi dengan tipe public dengan nama getAllBaptis |
| 30 | Membuat variable dengan nama perPage yang memiliki nilai 10 |
| 31 | Membuat variable dengan nama currentPage yang memiliki nilai dari data yang dikirimkan melalui method POST dengan nama variable POST tersebut adalah page serta dilakukan perbandingan apabila variable page tersebut memiliki nilai atau tidak, jika memiliki nilai maka akan mengembalikan nilai itu sendiri dan jika tidak maka akan diberikan nilai 1 |
| 32 | Membuat variable dengan nama dataBaptis yang berisikan pemanggilan data baptis dengan melakukan orderBy melalui baptism\_id serta dilakukan secara DESC (SELECT \*FROM table\_baptis ORDER BY SUBSTRING\_INDEX(baptism\_id, '/', -1) + 0 DESC) |
| 33 | Membuat variable dengan nama count yang berisikan jumlah kolom pada table baptis |
| 34 | Membuat variable dengan nama currentItems yang berisikan data variable dataBaptis pada line 32 akan dislice (menyalin element array ke element array yang baru melalui penentuan dari array yang akan diambil) |
| 35 | Memanggil variable dengan nama baptis yang akan berisikan dengan fungsi LengthAwarePaginator (sebuah fitur pada framework Laravel yang berfungsi untuk melakukan pagination) |
| 36 | Membuat variable dengan nama isi yang berisikan string kosong |
| 37 | Melakukan percabangan dengan menghitung jumlah kolom pada table baptis (line 33) dengan membandingkan apakah isi tersebut <=0 (JIKA) |
| 38 | Memanggil variable isi dan dilakukan penambahan value (append) pada variable isi tersebut |
| 39 | Mengisi variable isi dengan value <tr> |
| 40 | Mengisi variable isi dengan value <th colspan="7"> <center> TIDAK ADA DATA </center> </th> |
| 41 | Mengisi variable isi dengan value </tr> |
| 42 | Menutup variable isi (yang dibuka pada line 38) |
| 43 | Melakukan percabangan apabila percabangan pada line 37 tidak berhasil (MAKA) |
| 44 | Melakukan perulangan foreach dengan melakukan perulangan melalui variable paginator dan diset ke dalam variable baru yang bernama dB |
| 45 | Membuat variable dengan nama pisahTTL dengan isi adalah fungsi pemisah (explode) dari data hasil perulangan variable DB dengan value date\_ofBirth yang dipisahkan melalui sebuah [spasi] yang akan berbentuk array |
| 46 | Membuat variable dengan nama pisahTB dengan isi adalah fungsi pemisah (explode) dari data hasil perulangan variable DB dengan value date\_ofBaptism yang dipisahkan melalui sebuah [spasi] yang akan berbentuk array |
| 47 | Membuat variable dengan nama tanggal\_baruTTL dengan isi adalah fungsi pemisah (explode) dari data variable pisahTTL index ke 0 yang dipisahkan melalui sebuah [spasi] yang akan berbentuk array |
| 48 | Membuat variable dengan nama tanggal\_baruTB dengan isi adalah fungsi pemisah (explode) dari data variable pisahTB index ke 0 yang dipisahkan melalui sebuah [spasi] yang akan berbentuk array |
| 49 | Memanggil variable isi dan dilakukan penambahan value (append) pada variable isi tersebut |
| 50 | Mengisi variable isi dengan value <tr> |
| 51 | Mengisi variable isi dengan value <td>'.$dB->baptism\_id.'</td> |
| 52 | Mengisi variable isi dengan value <td>'.$dB->full\_name.'</td> |
| 53 | Mengisi variable isi dengan value <td>'.$dB->place\_ofBirth.', '.$tanggal\_baruTTL[2].'-'.$tanggal\_baruTTL[1].'-'.$tanggal\_baruTTL[0].'</td> yang merupakan hasil dari variable array pada line 47 |
| 54 | Mengisi variable isi dengan value <td>'.$tanggal\_baruTB[2].'-'.$tanggal\_baruTB[1].'-'.$tanggal\_baruTB[0].'</td> yang merupakan hasil dari variable array pada line 48 |
| 55 | Mengisi variable isi dengan value <td>'.$dB->pastor.'</td>'; |
| 56 | Melakukan percabangan dari value hasil loopingan foreach dengan data gender, apabila gender bernilai “female” maka akan disisi value Perempuan, jika tidak akan diisi oleh value Laki-laki |
| 57 | Mengisi variable isi '<td> Perempuan </td>'; |
| 58 | Fungsi percabangan apabila statement di line 56 tidak terpenuhi |
| 59 | Mengisi variable isi '<td> Laki-laki </td>'; |
| 60 | Menutup fungsi percabangan |
| 61 | Memanggil variable isi dan dilakukan penambahan value (append) pada variable isi tersebut |
| 62 | Mengisi variable isi dengan value <td> |
| 63 | Mengisi variable isi dengan value <button onclick="editBaptisModal(`'.$dB->baptism\_id.'`)" class="btn btn-warning text-white"> <i class="fas fa-pencil-alt"></i> </button> yang merupakan sebuah button yang bernama editBaptisModal serta diberikan sebuah parameter yakni value dari hasil loopingan dengan data baptism\_id serta diberikan class yang dipanggil dari framework bootstrap serta memanggil tag HTML dari librarby FontAwesome untuk icon pencil |
| 64 | Mengisi variable isi dengan value <button onclick="deleteBaptis(`'.$dB->baptism\_id.'`)" class="btn btn-danger"> <i class="fas fa-trash-alt"></i> </button> yang merupakan sebuah button yang bernama deleteBaptis serta diberikan sebuah parameter yakni value dari hasil loopingan dengan data baptism\_id serta diberikan class yang dipanggil dari framework bootstrap serta memanggil tag HTML dari librarby FontAwesome untuk icon tempat sampah |
| 65 | Mengisi variable isi dengan value </td> |
| 66 | Mengisi variable isi dengan value </tr> |
| 67 | Memanggil variable isi dan dilakukan penambahan value (append) pada variable isi tersebut |
| 68 | Menutup loopingan foreach yang dimulai pada line 44 |
| 69 | Menutup fungsi percabangan pada statement yang tidak terpenuhi pada line 43 |
| 70 | Mengisi variable isi dengan value “###” serta di (append) value dari variable paginator yang dihubungkan pada page vendor.pagination.bootstrap-4-ajax. Fungsi ### disini adalah untuk menjadi separator |
| 71 | Mengembalikan semua nilai dari variabel isi |
| 72 | Menutup fungsi getAllBaptis |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 73 | Membuat sebuah fungsi dengan tipe public dengan nama createBaptis yang menampung sebuah data dengan nama request pada paramaternya |
| 74 | Membuat variable dengan nama date yang memiliki value pemanggilan fungsi date PHP dengan format month/Year |
| 75 | Memmbuat variable dengan nama split yang memiliki value sebuah fungsi split (pemisah) dari variable dengan nama date dengan menggunakan “/” sebagai seperatornya sehingga value variable split akan menjadi sebuah array |
| 76 | Membuat variable dengan nama row yang memiliki value sebuah fungsi split (pemisah) dari variable dari data parameter dengan data baptism\_id dengan menggunakan “/” sebagai separator sehingga value dari variable row adalah sebuah array |
| 77 | Melakukan percabangan dari data parameter dengan data bernama photo, serta dilakukan pembandingan apakah data bernama foto bernilai true atau false |
| 78 | Membuat variable dengan nama photoName yang berisikan hasil array dari variable row digabunggkan dengan “.” [titik] serta ditambahkan kembali value hasil array dari variable split serta digabgungkan dengan “.” [titik] serta ditambahkan value dari hasil array variable split serta digabungkan dengan “.” [titik] |
| 79 | Membuat variable dengan nama extPhotoFile yang berisikan pengambilan extensi file dari data parameter dengan nama photo |
| 80 | Membuat variable dengan nama urlPhotoFile yang berisikan pemanggilan URL storage dari BAPTIS, serta dibuatkan sebuah folder dengan nama ‘Baptis’ yang akan berisikan dari file foto yang didapat dari data parameter dengan nama photo, serta memiliki nama file yang diambil dari variable photoName dengan extensi yang diambil dari variable extPhotoFile |
| 81 | Fungsi percabangan apabila statemen pada line 77 tidak terpenuhi |
| 82 | Mengisi variable urlPhotoFile dengan value string kosong “” |
| 83 | Menutup fungsi percabangan yang dimulai pada line 77 |
| 84 | Enter code |
| 85 | Memanggil dataBaptisModel yang akan di insert value |
| 86 | Mengisi table baptis pada kolom baptism\_id dengan data dari parameter yang bernama baptism\_id |
| 87 | Mengisi table baptis pada kolom familyCard\_id dengan data dari parameter yang bernama familyCard\_id |
| 88 | Mengisi table baptis pada kolom full\_name dengan data dari parameter yang bernama full\_name |
| 89 | Mengisi table baptis pada kolom gender dengan data dari parameter yang bernama gender |
| 90 | Mengisi table baptis pada kolom place\_ofBirth dengan data dari parameter yang bernama place\_ofBirth |
| 91 | Mengisi table baptis pada kolom date\_ofBirth dengan data dari parameter yang bernama date\_ofBirth |
| 92 | Mengisi table baptis pada kolom date\_ofBaptism dengan data dari parameter yang bernama date\_ofBaptism |
| 93 | Mengisi table baptis pada kolom church dengan value SION WKO |
| 94 | Mengisi table baptis pada religion dengan value Christian |
| 95 | Mengisi table baptis pada kolom father\_name dengan data dari parameter yang bernama father\_name |
| 96 | Mengisi table baptis pada kolom mother\_name dengan data dari parameter yang bernama mother\_name |
| 97 | Mengisi table baptis pada kolom address dengan data dari parameter yang bernama address |
| 98 | Mengisi table baptis pada kolom pastor dengan data dari parameter yang bernama pastor |
| 99 | Mengisi table baptis pada kolom photo dengan data dari variable urlPhotoFile |
| 100 | Menutup dataBaptisModel yang dibuka pada line 85 |
| 101 | Mengembalikan hasil fungsi ke tampilan sebelumnya yakni adm/dataJemaat/baptis serta diberikan sebuah notifikasi |
| 102 | Menutup fungsi createBaptis |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 103 | Membuat sebuah fungsi dengan tipe public dengan nama updateBaptis yang menampung sebuah data dengan nama request pada paramaternya |
| 104 | Membuat variable dengan nama dataBaptisById yang berisikan pemanggilan dataBaptisModel dengan value dari table baptis kolom baptism\_id adalah sama dengan data pada parameter dengan nama baptism\_id yang diambil value pertamanya (SELECT TOP 1\*FROM table\_baptis where baptism\_id = baptism\_id |
| 105 | Membuat variable dengan nama data yang berisikan fungsi date dari PHP dengan format month/Year |
| 106 | Memmbuat variable dengan nama split yang memiliki value sebuah fungsi split (pemisah) dari variable dengan nama date dengan menggunakan “/” sebagai seperatornya sehingga value variable split akan menjadi sebuah array |
| 107 | Membuat variable dengan nama row yang memiliki value sebuah fungsi split (pemisah) dari variable dari data parameter dengan data baptism\_id dengan menggunakan “/” sebagai separator sehingga value dari variable row adalah sebuah array |
| 108 | Melakukan percabangan dari data parameter dengan data bernama photo, serta dilakukan pembandingan apakah data bernama foto bernilai true atau false |
| 109 | Membuat variable dengan nama photoName yang berisikan hasil array dari variable row digabunggkan dengan “.” [titik] serta ditambahkan kembali value hasil array dari variable split serta digabgungkan dengan “.” [titik] serta ditambahkan value dari hasil array variable split serta digabungkan dengan “.” [titik] |
| 110 | Membuat variable dengan nama extPhotoFile yang berisikan pengambilan extensi file dari data parameter dengan nama photo |
| 111 | Membuat variable dengan nama urlPhotoFile yang berisikan pemanggilan URL storage dari BAPTIS, serta dibuatkan sebuah folder dengan nama ‘Baptis’ yang akan berisikan dari file foto yang didapat dari data parameter dengan nama photo, serta memiliki nama file yang diambil dari variable photoName dengan extensi yang diambil dari variable extPhotoFile |
| 112 | Fungsi percabangan apabila statemen pada line 108 tidak terpenuhi |
| 113 | Mengisi variable urlPhotoFile dengan value dari data pada variable dataBaptisByID dengan kolom photo |
| 114 | Menutup fungsi percabangan pada line 108 |
| 115 | Memanggil dataBaptisModel yang akan di update value berdasarkan baptism\_id yang dicocokan dengan data parameter dengan nama baptism\_id |
| 116 | Mengupdate table baptis pada kolom familyCard\_id dengan data dari parameter yang bernama familyCard\_id |
| 117 | Mengupdate table baptis pada kolom full\_name dengan data dari parameter yang bernama full\_name |
| 118 | Mengupdate table baptis pada kolom gender dengan data dari parameter yang bernama gender |
| 119 | Mengupdate table baptis pada kolom place\_ofBirth dengan data dari parameter yang bernama place\_ofBirth |
| 120 | Mengupdate table baptis pada kolom date\_ofBirth dengan data dari parameter yang bernama date\_ofBirth |
| 121 | Mengupdate table baptis pada kolom date\_ofBaptism dengan data dari parameter yang bernama date\_ofBaptism |
| 122 | Mengupdate table baptis pada kolom church dengan value SION WKO |
| 123 | Mengupdate table baptis pada religion dengan value Christian |
| 124 | Mengupdate table baptis pada kolom father\_name dengan data dari parameter yang bernama father\_name |
| 125 | Mengupdate table baptis pada kolom mother\_name dengan data dari parameter yang bernama mother\_name |
| 126 | Mengupdate table baptis pada kolom address dengan data dari parameter yang bernama address |
| 127 | Mengupdate table baptis pada kolom pastor dengan data dari parameter yang bernama pastor |
| 128 | Mengupdate table baptis pada kolom photo dengan data dari variable urlPhotoFile |
| 129 | Menutup dataBaptisModel yang dibuka pada line 115 |
| 130 | Mengembalikan hasil fungsi ke tampilan sebelumnya yakni adm/dataJemaat/baptis serta diberikan sebuah notifikasi |
| 131 | Menutup fungsi createBaptis |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 132 | Membuat sebuah fungsi dengan tipe public dengan nama deleteBaptis |
| 133 | Membuat sebuah fungsi try catch yang berfungsi untuk melakukan pengujian apabila selama pengujian error maka akan masuk ke catch |
| 134 | Memanggil dataBaptisModel yang akan dilakukan delete data berdasarkan kolom baptism\_id dari data yang dikirimkan melalui method POST dengan nama baptism\_id (DELETE FROM table\_baptis WHERE baptism\_id = baptism\_id) |
| 135 | Mengembalikan nilai “success” apabila try berhasil dilakukan |
| 136 | Menutup fungsi try yang dibuka pada line 133 |
| 137 | Membuat fungsi catch serta diberikan parameter sebuah Exception yang bernama ex |
| 138 | Mengembalikan nilai dari variable ex apabila terdapat kesalahan selama pengujian try berlangsung |
| 139 | Menutup fungsi catch yang dibuka pada line 137 |
| 140 | Menutup fungsi deleteBaptis |

modal.blade.php



index.blade.php

