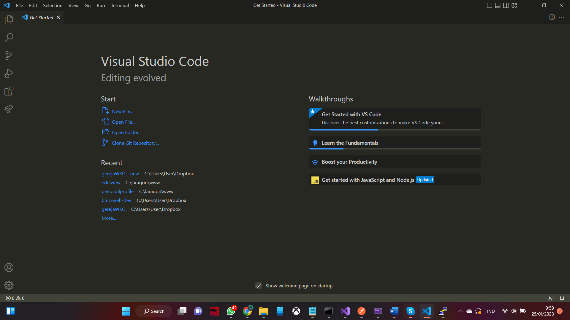
Kata **Gereja** berasal dari kata dalam bahasa Yunani “Ekklesia” yang didefinisikan sebagai “perkumpulan” atau “orang-orang yang dipanggil keluar.” Akar kata ”Gereja” tidak berhubungan dengan gedung, tetapi dengan orang. Roma 16:5 berkata, “Salam juga kepada jemaat di rumah mereka...” Paulus merujuk 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

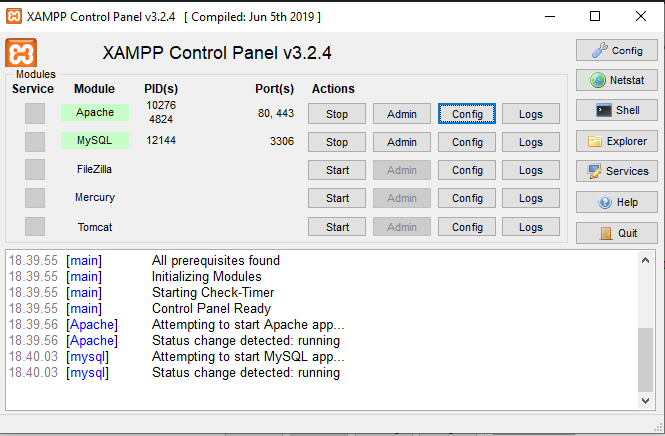
* Visual Studio Code
* XAMPP (PHP > 7.4)
* Git bash
* Composer
* Google Chrome
* Draw.io
* Figma
* Visual Studio Code

Sebuah code editor yang bisa dijalankan di perangkat desktop berbasis Windows, Linux, dan MacOS yang dapat digunakan untuk membuat dan mengedit source code berbagai bahasa pemgrograman Misalnya, seperti JavaScript, TypeScript, dan Node.js. Bahkan, Visual Code Studio juga kompatibel dengan bahasa dan runtime environment lain, seperti PHP, Pyhton , Java , dan .NET. Hal ini berkat ekosistemnya yang luas dan ketersediaan extension yang melimpah.

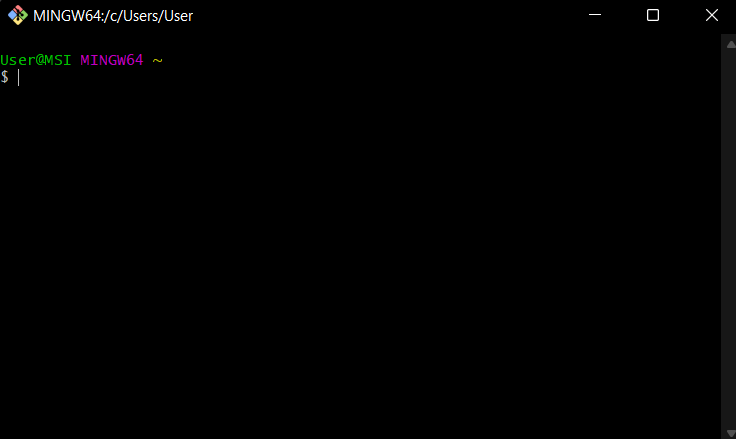
.

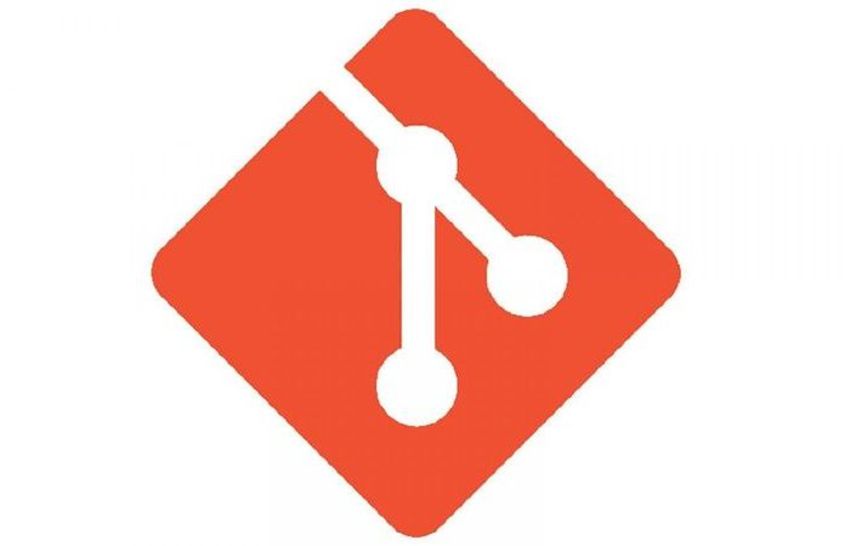
* XAMPP (PHP > 7.4)

XAMPP adalah software atau aplikasi komputer yang banyak digunakan dalam dunia web developer yang juga bisa dipelajari untuk membuat website. XAMPP adalah perangkat lunak berbasis web server yang bersifat open source (bebas) serta mendukung di berbagai sistem operasi seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. XAMPP adalah singkatan dari X (cross platform), A (Apache), M (MySQL/MariaDB), P (PHP), dan P (Perl) yang adalah program-program yang tersedia di software ini.



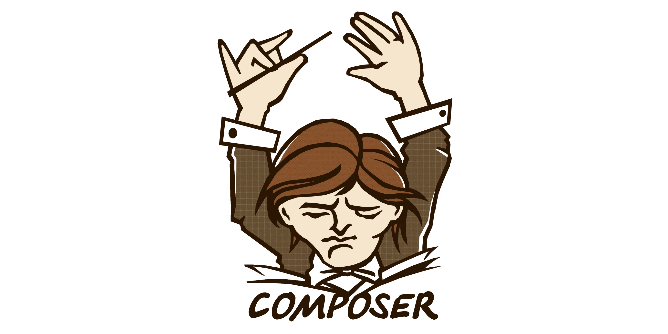
* Git Bash

Git Bash adalah aplikasi untuk lingkungan Microsoft Windows yang menyediakan lapisan emulasi untuk pengalaman Git command line. Bash adalah akronim untuk Bourne Again Shell. Shell adalah aplikasi terminal yang digunakan untuk berinteraksi dengan sistem operasi melalui perintah tertulis. Bash adalah shell default populer di Linux dan macOS. Git Bash adalah paket yang menginstal Bash, beberapa utilitas bash umum, dan Git pada sistem operasi Windows.



* Composer

Composer adalah alat dependency manager untuk bahasa pemrograman PHP. Dengan kata lain, Composer adalah aplikasi yang diinstal ke perangkat untuk memfasilitasi developer menggunakan library open source milik orang lain ke dalam project yang sedang dibangun.



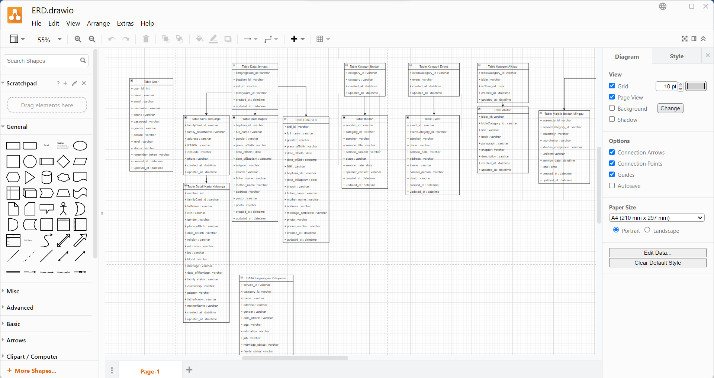
* Google Chrome

**Google Chrome merupakan peramban web/ browser lintas platform yang dikembangkan oleh perusahaan Google Inc. Google Chrome** juga menjadi salah satu media yang sangat dibutuhkan untuk pengembangan suatu web oleh para programmer maupun developer. Selain teks editor, para programmer juga membutuhkan Google Chrome untuk melihat hasil run/ compile code, baik offline ataupun online. **Programmer juga biasa menggunakan fitur Inspect Element untuk melihat kode-kode yang ditulisnya.**



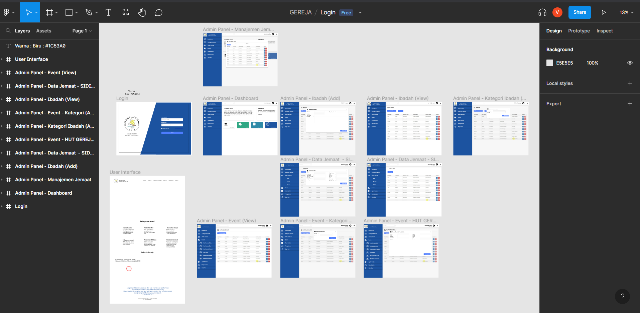
* Draw.io

Draw.io adalah sebuah aplikasi yang didesain khusus untuk menggambarkan diagram. Semua fitur yang ada pada situs ini bisa kalian nikmati hanya dengan bermodalkan browser yang mendukung HTML 5. Draw.io menawarkan kemudahan untuk membuat diagram tanpa batasan jumlah diagram yang dibuat.



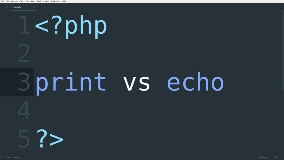
* Figma

Figma adalah editor grafis vektor dan alat prototyping dengan berbasis web serta fitur offline tambahan yang diaktifkan oleh aplikasi desktop untuk Mac OS dan Windows. Aplikasi pendamping Figma Mirror untuk Android dan iOS memungkinkan untuk melihat prototype Figma pada perangkat seluler. Rangkaian fitur Figma berfokus pada penggunaan dalam antarmuka pengguna dan desain pengalaman pengguna dengan penekanan pada kolaborasi waktu nyata (real- time).



Penjelasan mengenai bahasa pemgrograman beserta library yang digunakan :

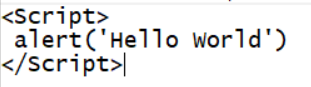
* PHP

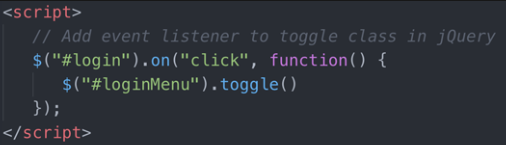
**PHP (PHP: Hypertext Preprocessor)** adalah sebuah bahasa pemrograman **server side scripting** yang bersifat **open source**. PHP merupakan bahasa pemrograman server-side, maka script dari PHP nantinya akan diproses di server. Jenis server yang sering digunakan bersama dengan PHP antara lain **Apache**, **Nginx**, dan **LiteSpeed**.

* + Laravel

Laravel adalah *framework*berbasis bahasa pemgrograman PHP yang bisa digunakan untuk membantu proses pengembangan sebuah website agar lebih maksimal. Dengan menggunakan Laravel, website yang dihasilkan akan lebih dinamis. *Framework*Laravel menggunakan struktur MVC (*Model View Controller)*. MVC merupakan model aplikasi yang memisahkan antara data dan tampilan berdasarkan komponen aplikasi.

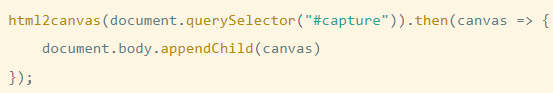
* JavaScript

JavaScript adalah bahasa skrip yang digunakan untuk membuat konten halaman web dinamis, berfungsi untuk membuat elemen yang mampu meningkatkan interaksi pengunjung seperti menu drop-down, animasi, dan warna background dinamis serta membangun web server dan aplikasi server, membuat website yang interaktif, serta game development.

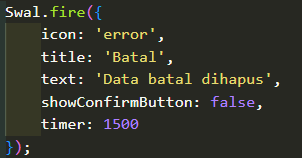
* + jQuery

jQuery adalah library JavaScript yang cukup andal, ringkas, dan mempunyai fitur yang cukup lengkap. Library ini membuat pemrosesan di HTML seperti perubahan dan manipulasi dokumen, event handling, animasi, dan AJAX JavaScript menjadi lebih sederhana. Hal ini didukung dengan API yang mudah digunakan dan dapat bekerja di berbagai macam browser. Menggunakan kombinasi versatility dan extensibillty.

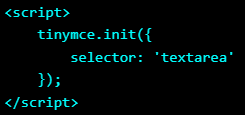
* + HTML2Canvas

html2canvas memungkinkan kita untuk mengambil tangkapan layar pada suatu halaman web langsung dari browser pengguna. Tangkapan layar dilakukan berdasarkan informasi yang tersedia di halaman berdasarkan DOM.

* + SweetAlert2

SweetAlert merupakan library JavaScript yang menyediakan berbagai pilihan bagi kita untuk membuat berbagai jenis pop-up alert seperti alert biasa, alert error, alert success, alert konfirmasi, dan sebagainya.

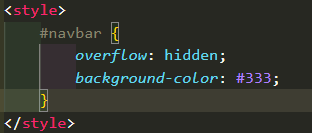
* + TinyMCE

****TinyMCE merupakan salah 1 WYSIWYG (What You See Is What You Get) Editor yang dapat memberikan fitur manajemen data yang dimasukkan ke dalam sebuah form yang terintegrasi dengan tag html "textarea".

* + AJAX

AJAX atau Asynchronous JavaScript and XML adalah teknik yang digunakan untuk membuat website yang dinamis. Artinya website mampu mengupdate dan menampilkan data baru dari server tanpa perlu melakukan reload.

* CSS

CSS adalah bahasa Cascading Style Sheet dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs.

* + Bootsrap

Bootstrap adalah untuk membuat website responsive dan mobile-first. Jadi, semua elemen antarmuka website dipastikan bisa bekerja secara optimal di semua ukuran layar, baik desktop maupun perangkat seluler.

* + FontAwesome5

Font Awesome merupakan font yang berisi icon-icon yang telah dimuat menjadi framework css yang bisa digunakan untuk mempermudah menampilkan icon-icon standar dalam front-end development

* HTML

Hypertext Markup Language atau HTML adalah bahasa markup standar yang digunakan untuk membuat halaman website dan aplikasi web. Kode HTML tersebut memastikan format teks dan gambar yang tepat untuk browser Internet. Tanpa HTML, browser tidak akan tahu bagaimana menampilkan teks sebagai elemen atau memuat gambar atau elemen lainnya.

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**

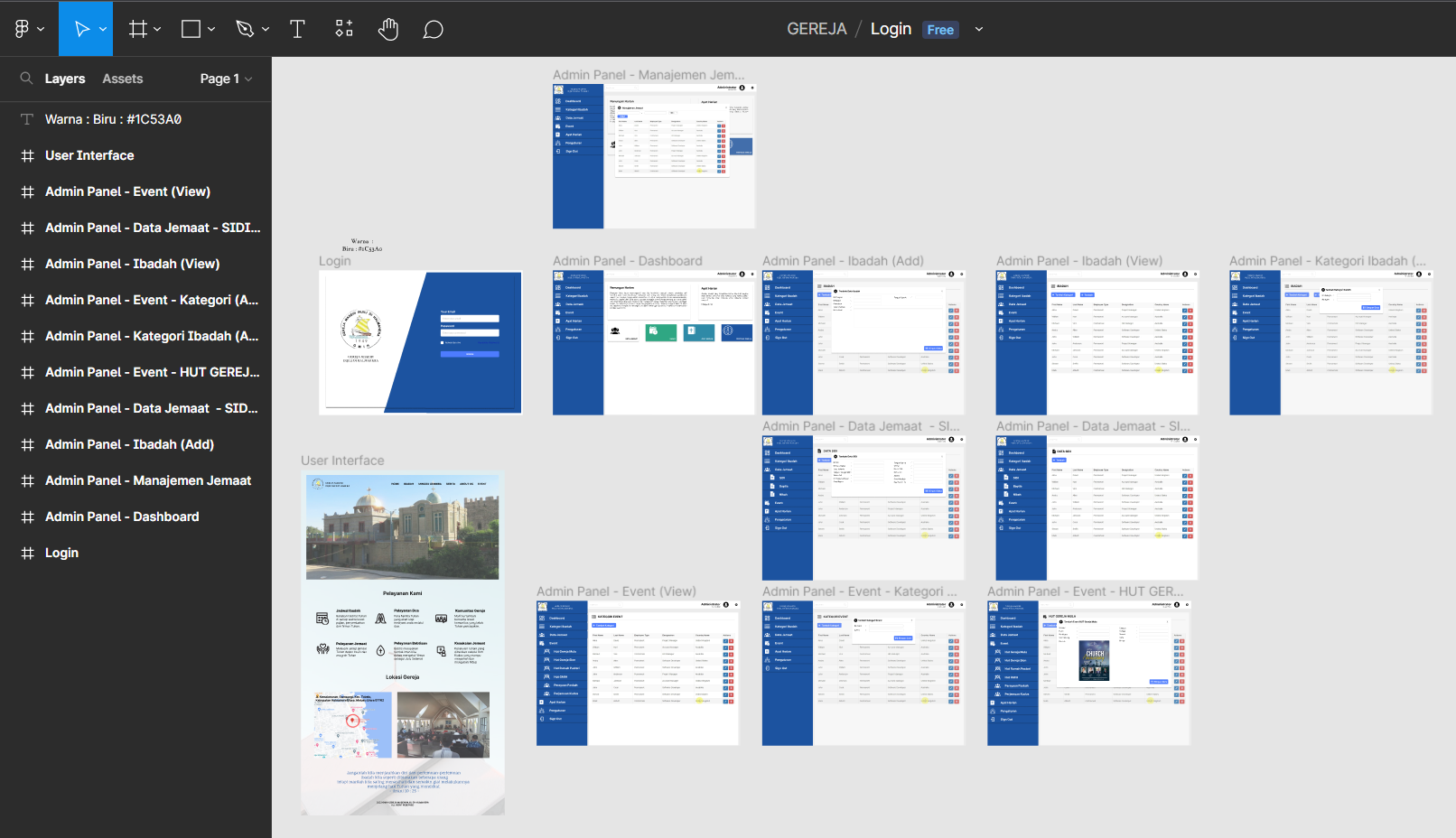
Pada langkah ini, akan dilakukan analisis kelayakan dan kebutuhan *website* seperti fungsi *website*, biaya *website*, arah kegunaan *website*, tujuan *website*, dan sasaran dari *website* sehingga pada tahap ini akan menjadi batu pijakan awal dalam pembuatan sebuah *web application*. Pemahaman mengenai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

**Analisis dan tentukan keperluan website**

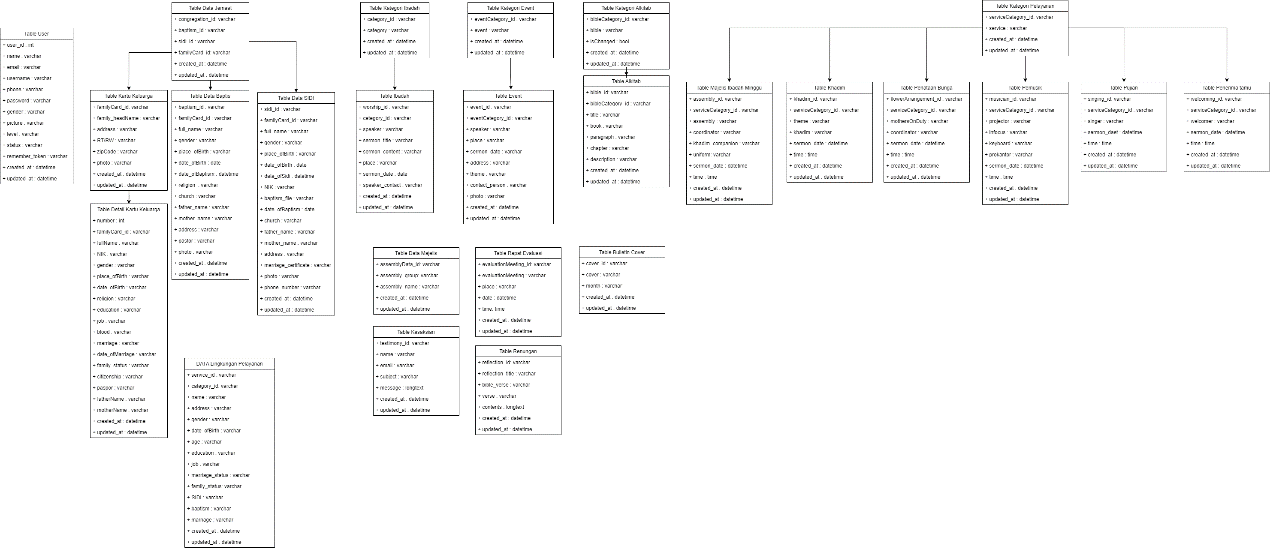
Pengembangan sebuah *website* tak luput dan adanya persiapan keperluan dalam pembangunan sebuah *website*, infrastruktur *website* akan dirancang pada tahapan ini sehingga alur kerja dan arah pergerakan *website* akan terdeteksi pada langkah ini. Perangkat yang digunakan, lingkup kerja keinginan *user*,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

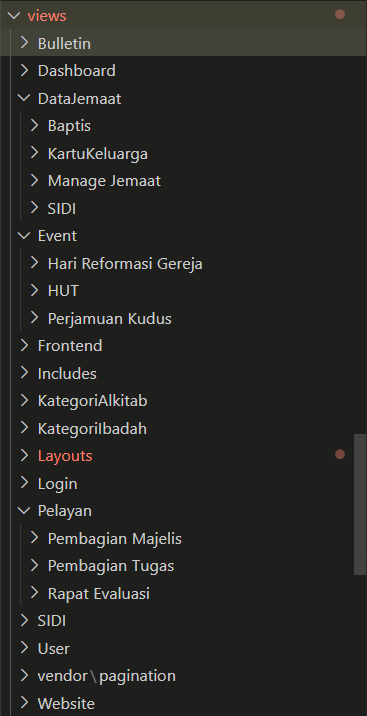
Tahapan pengembangan menjadi tahapan yang penting dimana sebuah web application akan mulai dibangun menggunakan bahasa pemrograman. Tahapan ini akan mulai dikerjakan dengan menulis program ke komputer menggunakan bahasa yang sudah disepakati diawal yakni PHP sebagai bahasa *server side.* Pengerjaan sebuah 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 :

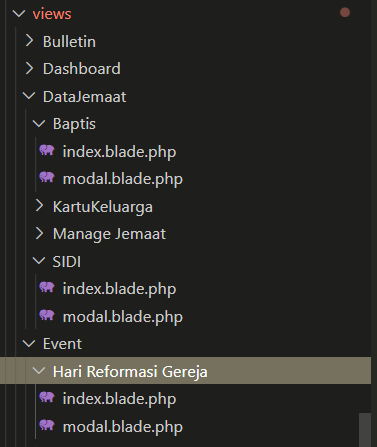
* Menginstall git bash
* Menginstall composer
* Menginstall PHP yang hendak digunakan menggunakan aplikasi XAMPP sebagai server
* Menginstall Laravel menggunakan git bash

Langkah – langkah pembuatan web application :

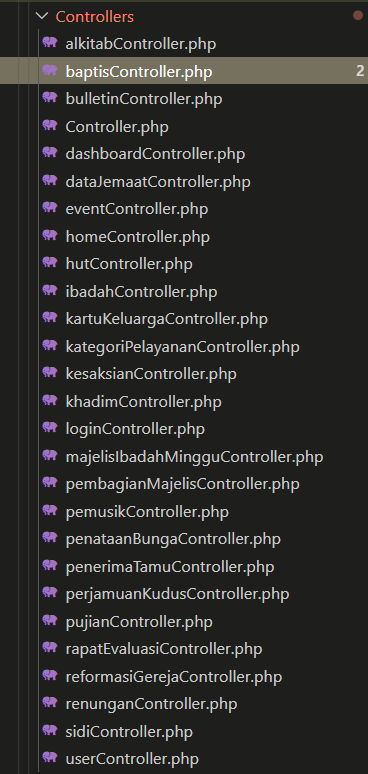
* Membuat folder dan subfolder berdasarkan nama tampilan pada web application



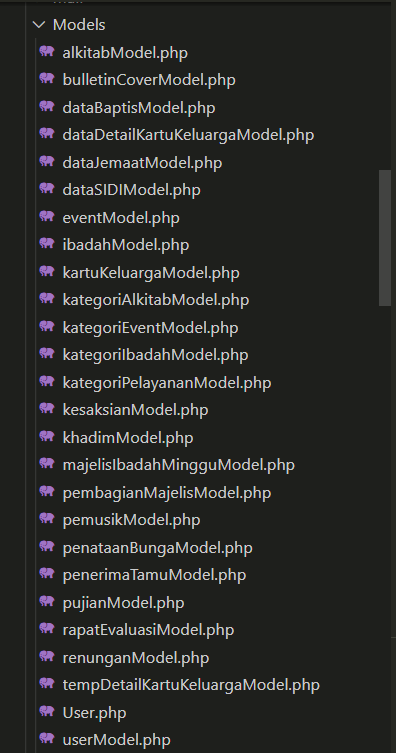
* Menentukan penamaan file pada subfolder yang ada guna memudahkan dalam pengerjaan



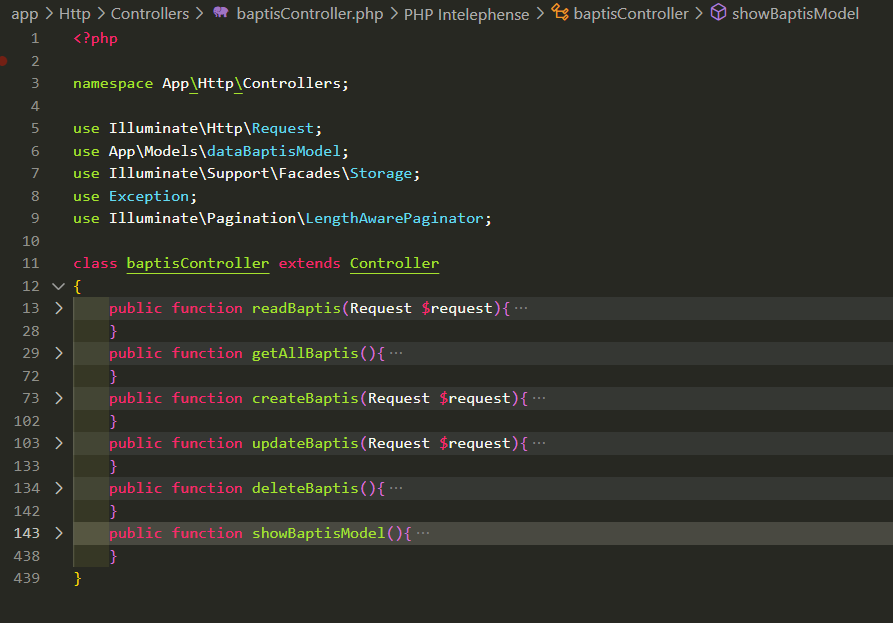
* Menentukan penamaan controller yang mudah sesuai dengan tampilan pada web application



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



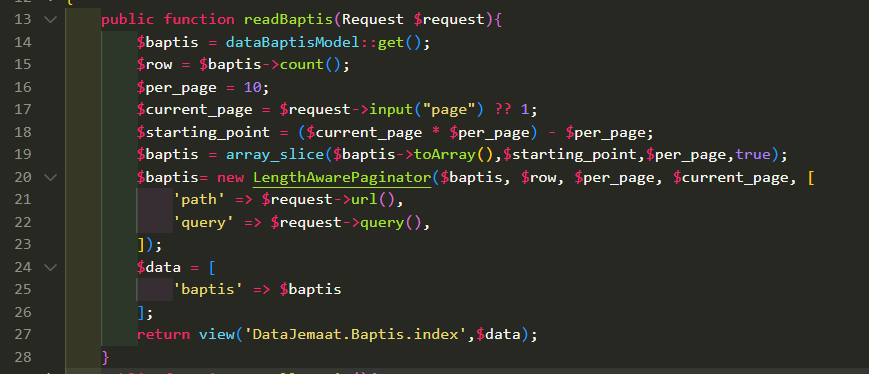
* Menentukaan penamaan fungsi di controller yang akan sejalan dengan penamaan routing



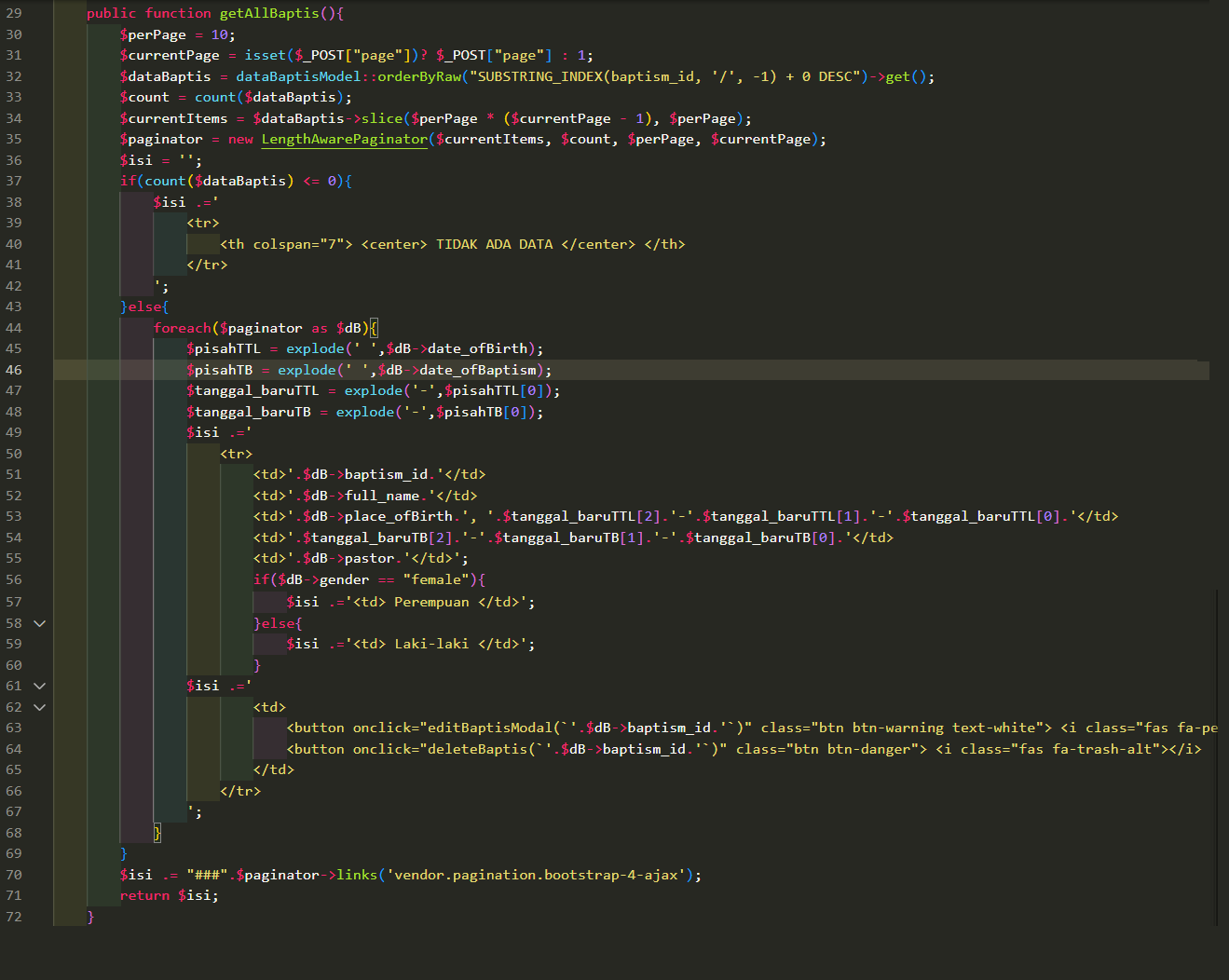
* Menentukan penamaan routing yang mudah dan sederhana sehingga tidak membuat ambigu dalam pengerjaan dengan membagi berdasarkan method yang digunakan mulai dari POST, PUT, GET dan DELETE



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 |



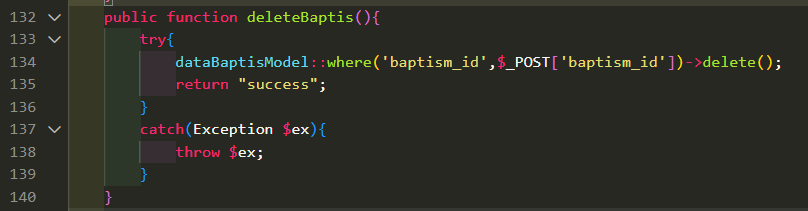
|  |  |
| --- | --- |
| Line | Penjelasan |
| 29 | Membuat sebuah fungsi dengan tipe public dengan nama getAllBaptis |
| 30 | Membuat variable dengan nama perPage yang memiliki nilai 10 |
| 31 | Membuat variable dengan nama currentPage yang memiliki nilai dari data yang dikirimkan melalui method POST 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 |



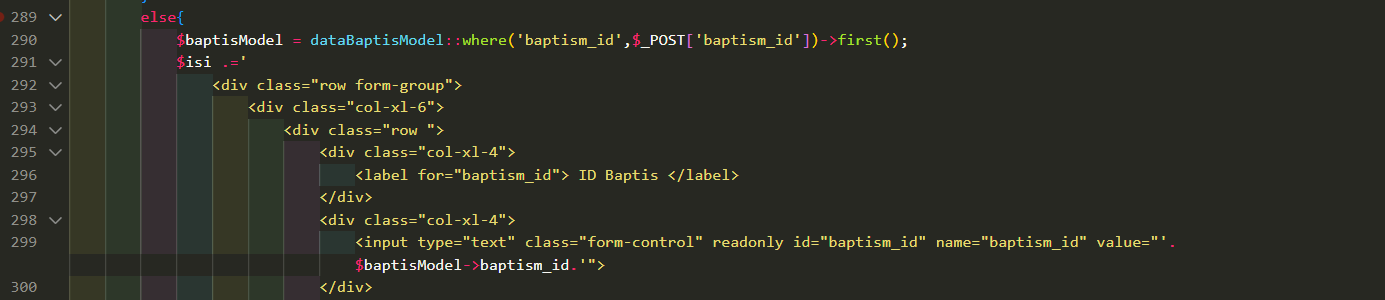
|  |  |
| --- | --- |
| Line | Penjelasan |
| 103 | Membuat sebuah fungsi dengan tipe public dengan nama updateBaptis yang menampung sebuah data dengan nama request pada paramaternya |
| 104 | Membuat variable dengan nama dataBaptisById yang berisikan pemanggilan dataBaptisModel dengan value dari table 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 |



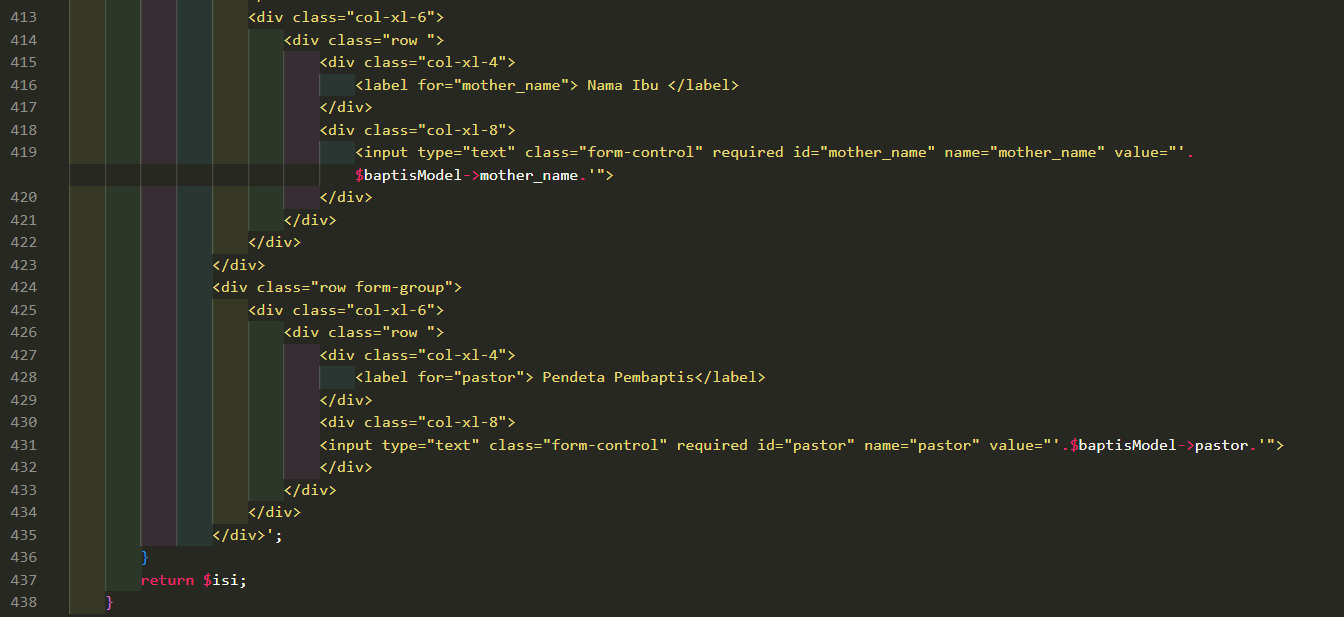
|  |  |
| --- | --- |
| Line | Penjelasan |
| 132 | Membuat sebuah fungsi dengan tipe public dengan nama deleteBaptis |
| 133 | Membuat sebuah fungsi try catch yang berfungsi untuk melakukan pengujian apabila selama pengujian error maka akan masuk ke catch |
| 134 | Memanggil dataBaptisModel yang akan 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 |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 143 | Membuat fungsi yang bertipe public dengan nama “showBaptisModel” |
| 144 | Membuat variabel dengan nama dataBaptis dan diberikan value dengan memanggil table baptis serta diberikan orderBy berdasarkan baptism\_id yang diambil index terakhir dari id tersebut serta dibuat ASC |
| 145 | Membuat variable dengan nama row yang berisikan jumlah kolom pada table baptis |
| 146 | Membuat percabangan dari variabel row apakah bernilai 0 atau tidak |
| 147 | Jika bernilai 0, maka variabel row akan diisikan nilai 1 |
| 148 | Percabangan apabila statement pada line 146 tidak terpenuhi |
| 149 | Membuat variabel dengan nama id\_baptis yang berisikan value mengambil data dari dataBaptis berdasarkan index dari variabel row yang diambil datanya adalah baptism\_id |
| 150 | Membuat variabel pisah yang akan diberikan value fungsi split dari PHP , dengan seperatornya berupa tanda “/” dimana value yang displit diambil dari variabel id\_baptis yang akan berbentuk array |
| 151 | Memanggil variabel dengan nama row yang berisiakan hasil split dari variabel pisah yang diambil index ke 3 ditambah 1 |
| 152 | Menutup hasil percabangan pada line 148 |
| 153 | Membuat variabel date dengan nilai fungsi PHP date dengan format (‘m/Y’) |
| 154 | Membuat variabel split yang akan diberikan value fungsi split dari PHP , dengan seperatornya berupa tanda “/” dimana value yang displit diambil dari variabel date yang akan berbentuk array |
| 155 | Membuat variabel dengan nama bulan yang berisikan hasil dari variabel split pada index 0 |
| 156 | Membuat variabel dengan nama tahun yang berisikan hasil dari variabel split pada index 1 |
| 157 | //ENTER |
| 158 | Membuat variabel dengan nama isi dengan value char kosong |
| 159 | //ENTER |
| 160 | Membuat percabangan dari data yang dikirimkan dengan method POST yakni data cmd serta dibandingkan isinya apakah bernilai “add” atau tidak |
| 161-288 | Mengisi variabel isi dengan tag HTML |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 289 | Melakukan fungsi percabangan apabil statement pada line 160 tidak terpenuhi |
| 290 | Membuat variabel dengan nama baptisModel dengan isi memanggil data pada table baptis dimana baptism\_id sama nilainya dengan data yang dikirimkan melalui method POST yakni data baptism\_id |
| 291-436 | Mengisi variabel isi dengan tag HTML |



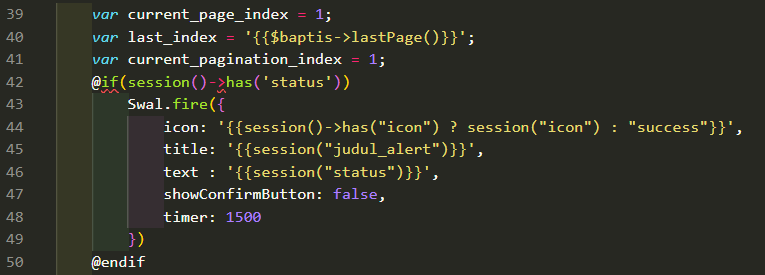
|  |  |
| --- | --- |
| Line | Penjelasan |
| 437 | Mengembalikan nilai dari variabel isi |
| 438 | Menutup fungsi showBaptisModel |

modal.blade.php

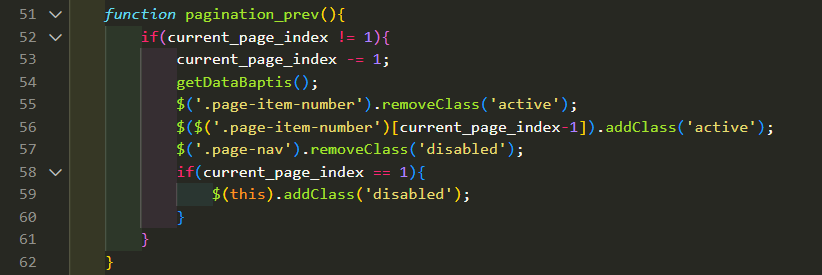


|  |  |
| --- | --- |
| Line | Penjelasan |
| 1 | Memberikan comment sebagai penanda dengan nama Baptis Model |
| 2 | Membuat sebuah div yang diberikan sebuah class dengan nama "modal" yang dipanggil berdasarkan class name pada framework boostrap serta diberikan id dengan nama "addBaptis" pada div tersebut sebagai penanda |
| 3 | Membuat sebuah div yang diberikan sebuah class dengan nama "modal-dialog modal-xl" serta diberikan role document yang dipanggil berdasarkan class name pada framework bootstrap |
| 4 | Membuat sebuah div yang diberikan sebuah class dengan nama "modal-content" yang dipanggil berdasarkan class name pada framework bootstrap |
| 5 | Membuat sebuah div yang diberikan sebuah class dengan nama "modal-header" yang dipanggil berdasarkan class name pada framework bootstrap |
| 6 | Membuat sebuah tag H5 yang akan bertuliskan DISINI JUDUL serta diberikan sebuah class dengan nama "modal-title" yang dipanggil berdasarkan class name pada framework boostrap |
| 7 | Membuat sebuah button dengan type button serta diberikan attribte data-disiss ,aria-label serta sebuah class dengan nama close yang dipanggil berdasarkan class name pada framework bootstrap |
| 8 | Membuat sebuah span yang diberikan attribute aria-hidden serta disisipkan specialcharacters &times; yang akan menjadi sebuah X dari sisi user |
| 9 | Menutup tag button yang dibuka pada line 7 |
| 10 | Menutup tag div yang dibuka pada line 5 |
| 11 | Membuat tag form dengan attribute action , method , id dan enctype dimana id tersebut bernama "tampilBaptisModal", action tersebut memiliki value # serta method adalah POST |
| 12 | Membuat tag div yang diberikan sebuah attribute class dengan nama "modal-body" yang dipanggil berdasarkan class name pada framework bootstrap |
| 13 | Memanggil framework template dari laravel yang berguna untuk memberikan value token untuk transfer data dengan method POST |
| 14 | Membuat tag div dengan attribute id yang bernama "isiBaptisModal" yang berfungi untuk menampung hasil dari nilai balikan dari sebuah AJAX |
| 15 | Menutup tag div yang dibuka pada line 12 |
| 16 | Membuat tag div yang diberikan sebuah attribute class dengan nama "modal-footer" yang dipanggil berdasarkan class name pada framework bootstrap |
| 17 | Membuat tag button dengan attribute class, id serta type dimana class tersebut bernama "btn btn-primary" yang dipanggil berdasarkan class name pada framework boostrtap, id dengan nama "cmd" yang akan menjadi penanda sebuah button , serta type dengan nama submit untuk memberikan penekanan bahwa fungi button tersebut sebagai submit yang akan berguna untuk mengeksekusi dari sebuah form dengan method POST serta disisipkan tulisan DISINI FOOTER yang akan berubah-ubah mengkuti data AJAX yang dikirim |
| 18 | Menutup tag div yang dibuka pada line 16 |
| 18 | Menutup tag form yang dibuka pada line 11 |
| 20 | Menutup tag div yang dibuka pada line 4 |
| 21 | Menutup tag div yang dibuka pada line 3 |
| 22 | Menutup tag div yang dibuka pada line 2 |

index.blade.php



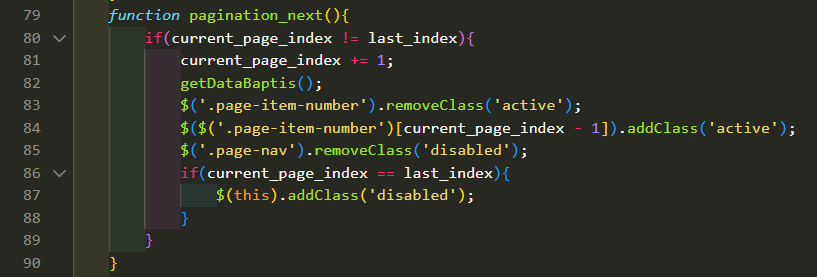
|  |  |
| --- | --- |
| Line | Penjelasan |
| 39 | Membuat variabel current\_page\_index dengan value 1 yang dapat diakses secara public |
| 40 | Membuat variabel last\_index dengan value yang didapatkan dari data yang dikirim oleh baptisController dengan variabel bernama baptis dengan data lastPage yang dapat diakses secara public |
| 41 | Membuat variabel dengan current\_pagination\_index dengan value 1 |
| 42 | Membuat percabangan dengan memanfaatkan template dari laravel yakni fungsi "@", dengan melakukan perbandingan dari sebuah session dengan nama status yang dikirimkan dari baptisController |
| 43 | Memanggil library sweetalert2 yakni fungsi Swal.fire |
| 44 | Memanggil variabel icon dari library sweetalert2 dengan isi dari session yang dikirimkan dari baptisController |
| 45 | Memanggil variabel title dari library sweetalert2 dengan isi dari session yang dikirimkan dari baptisController |
| 46 | Memanggil variabel text dari library sweetalert2 dengan isi dari session yang dikirimkan dari baptisController |
| 47 | Memanggil variable showConfirmButton dari library sweetalert2 dengan isi false |
| 48 | Memanggil variable timer dari library sweetalert2 dengan isi 1500 |
| 49 | Menutup library sweetaler2 yang dipanggil pada line 43 |
| 50 | Menutup fungsi percabangan dari template laravel |



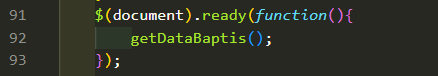
|  |  |
| --- | --- |
| Line | Penjelasan |
| 51 | Membuat sebuah fungsi pada javascript dengan nama pagination\_prev |
| 52 | Membuat percabangan dengan memanggil variable current\_page\_index serta dilakukan pengecekan dari value variable tersebut tidak boleh bernilai 1 |
| 53 | Mengisi variable current\_page\_index dengan mengurangi nilainya dengan 1 |
| 54 | Memanggil fungsi getDataBaptis() |
| 55 | Memanggil class name “page-item-number” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “active” (document.getElementByClassName(‘page-item-number’)) |
| 56 | Memanggil class name “page-item-number+(value dari variable current\_page\_index dikurangi 1)” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “active” |
| 57 | Memanggil class name “page-nav” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “disabled” |
| 58 | Membuat percabangan dengan membandingkan isi dari variable current\_page\_index apakah nilainya bernilai 1 |
| 59 | Memanggil diri sendiri (memanggil pagination\_prev()) berdasarkan function yang menekannya, serta diambil classname ataupun idnamenya serta diberikan attribute tambahan yakni penamabahan classname “disabled” |
| 60 | Menutup percabangan yang dibuka pada line 58 |
| 61 | Menutup percabangan yang dibuka pada line 52 |
| 62 | Menutup fungsi pagination\_prev yang dibuka pada line 51 |



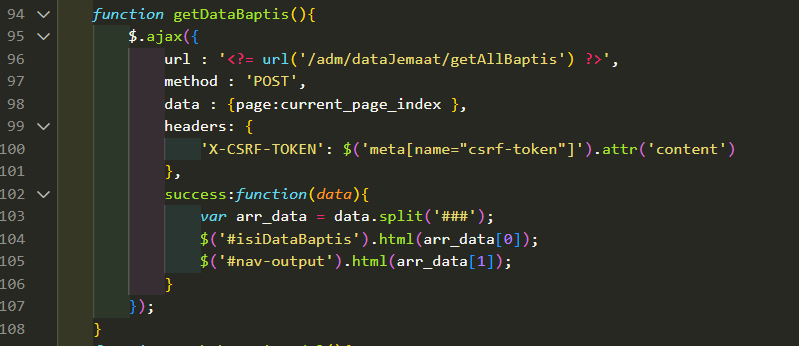
|  |  |
| --- | --- |
| Line | Penjelasan |
| 63 | Membuat sebuah fungsi dengan nama pagination\_item di JavaScript serta diberikan sebuah parameter dengan nama page |
| 64 | Memanggil variabel current\_page\_index serta diisikan dengan nilai baru yang diambil dari parameter yang dikirimkan yakni dengan nama page |
| 65 | Memanggil fungsi getDataBaptis |
| 66 | Mebuat sebuah percabangan dengan membandingkan isi variabel current\_page\_index apakah bernilai 1 atau tidak |
| 67 | Memanggil id dengan name “prev-page” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “disabled” |
| 68 | Melakukan percabangan kondisi kedua apabila kondisi pertama pada line 66 tidak terpenuhi |
| 69 | Memanggil id dengan name “prev-page” dengan memanfaatkan library jQuery serta diberikan attribute yakni penambahan classname “disabled” |
| 70 | Menutup percabangan yang dibuka pada line 66 |
| 71 | Membuat percabangan dengan kondisi memanggil variabel current\_page\_index serta dibandingkan isinya dengan variabel last\_index apapakah isinya sama atau tidak |
| 72 | Memanggil id dengan name “next-page” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “disabled” |
| 73 | Melakukan percabangan kondisi kedua apabila kondisi pertama pada line 71 tidak terpenuhi |
| 74 | Memanggil id dengan name “next-page” dengan memanfaatkan library jQuery serta diberikan attribute yakni penambahan classname “disabled” |
| 75 | Menutup percabangan yang dibuka pada line 71 |
| 76 | Memanggil class name “page-item-number” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “active” |
| 77 | Memanggil diri sendiri (memanggil pagination\_item()) berdasarkan function yang menekannya, serta diambil classname ataupun idnamenya serta diberikan attribute tambahan yakni penamabahan classname “active” |
| 78 | Menutup fungsi pagionation\_item yang dibuka pada line 63 |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 79 | Membuat sebuah fungsi pada javascript dengan nama pagination\_next |
| 80 | Membuat percabangan dengan memanggil variable current\_page\_index serta dibandingkan dengan variable last\_index apakah memiliki nilai yang sama atau tidak |
| 81 | Mengisi variable current\_page\_index dengan menambahni nilainya dengan 1 |
| 82 | Memanggil fungsi getDataBaptis() |
| 83 | Memanggil class name “page-item-number” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “active” (document.getElementByClassName(‘page-item-number’)) |
| 84 | Memanggil class name “page-item-number+(value dari variable current\_page\_index dikurangi 1)” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “active” |
| 85 | Memanggil class name “page-nav” dengan memanfaatkan library jQuery serta diberikan attribute yakni menghapus classname “disabled” |
| 96 | Membuat percabangan dengan membandingkan isi dari variable current\_page\_index dengan variable last\_index apakah memiliki nilai yang sama atau tidak |
| 87 | Memanggil diri sendiri (memanggil pagination\_prev()) berdasarkan function yang menekannya, serta diambil classname ataupun idnamenya serta diberikan attribute tambahan yakni penamabahan classname “disabled” |
| 88 | Menutup percabangan yang dibuka pada line 86 |
| 89 | Menutup percabangan yang dibuka pada line 80 |
| 90 | Menutup fungsi pagination\_prev yang dibuka pada line 79 |



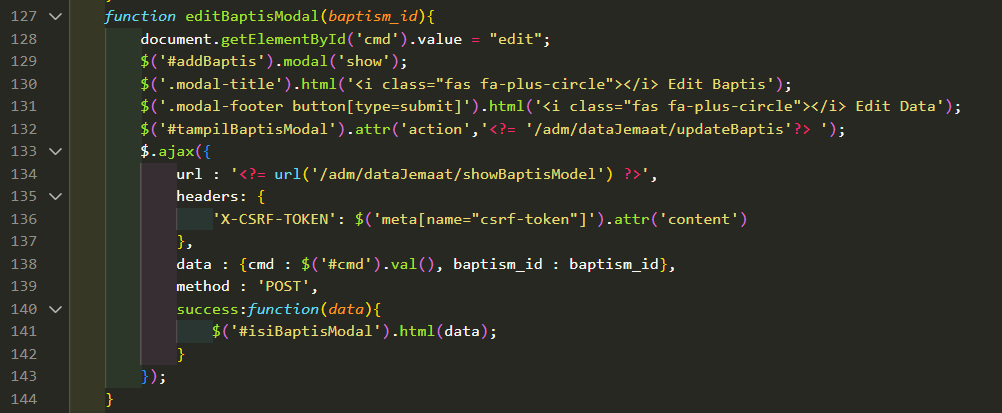
|  |  |
| --- | --- |
| Line | Penjelasan |
| 91 | Membuat sebuah statement bahwa setiap code yang diprogramkan didalamnya akan bersifat ready, sehingga setiap page direload maka fungsi ini akan otomatis ikut terjalankan juga |
| 92 | Memanggil fungsi getDataBaptis |
| 93 | Menutup statement yang dibuka pada line 91 |



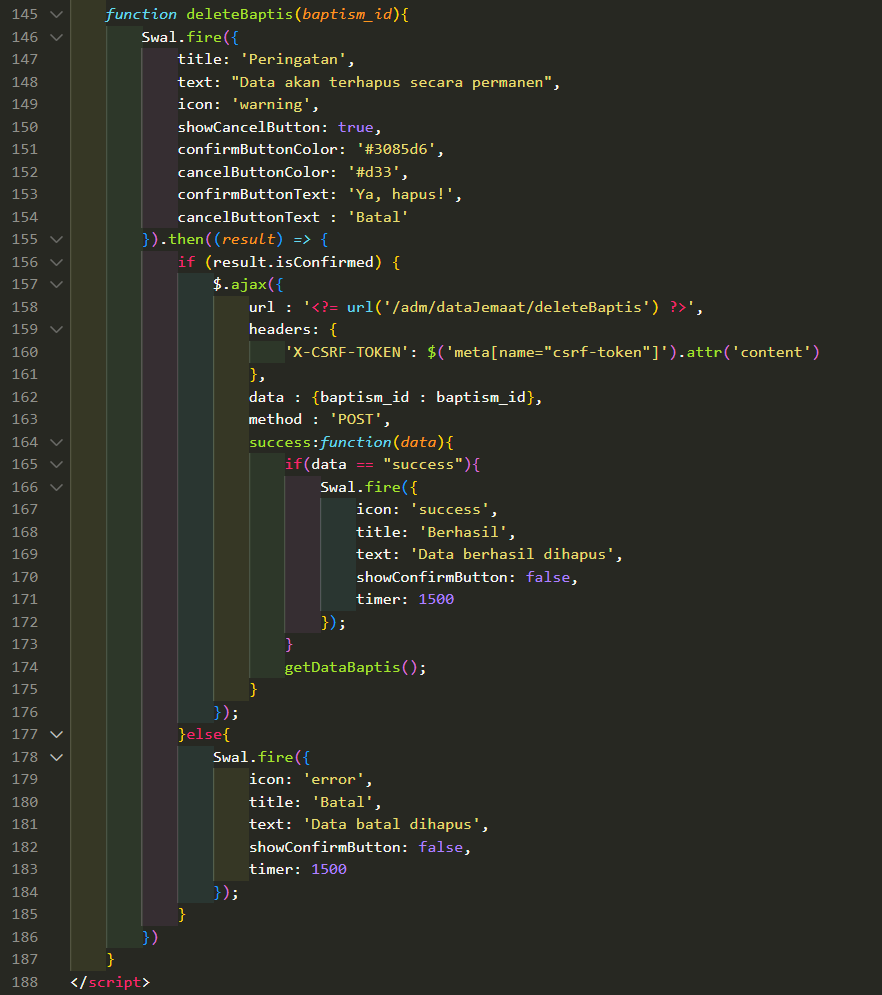
|  |  |
| --- | --- |
| Line | Penjelasan |
| 94 | Membuat sebuah fungsi di JavaScript dengan nama getDataBaptis |
| 95 | Memanggil fungsi ajax dengan menggunakan library jQuery |
| 96 | Memanggil variabel url yang ada didalam fungsi ajax jQuery yang diisikan value dari routingan controller tujuan dengan memanfaatkan fungsi plugin dari framework laravel yakni "url" |
| 97 | Memanggil variabel method yang ada didalam fungsi ajax jQuery yang disikan oleh method yang digunakan yakni "POST" |
| 98 | Memanggil variabel data yang ada didalam fungsi ajax jQuery yang disikan value dari variabel global "current\_page\_index" yang disimpan didalam sebuah varibel baru dengan nama "page" |
| 99 | Memanggil variabel headers yang ada di dalam fungsi ajax jQuery yang disikan value |
| 100 | Mengisi value dari variabel headers dengan sebuah token dikarenakan method yang digunakan untuk pengiriman data adalah "POST" sehingga adanya penjagaan bahwa data hanya bisa dikirimkan 1x oleh sebab itu diberikan sebuah token |
| 101 | Menutup pemanggilan variabel headers yang dibuka pada line 100 |
| 102 | Membuat response apakah pengiriman data berhasil dilakukan atau tidak, pada baris ini akan menampilkan response apabila data berhasil dikirim dengan response yang dikirim akan ditampung oleh sebuah parameter dengan nama "data" |
| 103 | Membuat sebuah variabel dengan nama "arr\_data" yang akan memiliki value hasil return apabila transfer data sukses serta dilakukan split dengan seperatornya yakni tanda "###" |
| 104 | Memanggil sebuah document yang memiliki id "isiDataBaptis" dimana document tersebut akan diisikan sebuah data HTML didalamnya melalui sebuah variabel "arr\_data" yang sudah berhasil displit sehingga menjadi array |
| 105 | Memanggil sebuah document yang memiliki id "nav-output" dimana document tersebut akan disikan oleh sebuah data HTML didalamnya melalui sebuah variabel "arr\_data" yang sudah berhasil displit sehingga menjadi array |
| 106 | Menutup hasil response sukses yang dibuka pada line 102 |
| 107 | Menutup pemanggilan fungsi ajax yang dibuka pada line 95 |
| 108 | Menutup fungsi getDataBaptis yang dibuka pada line 94 |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 109 | Membuat sebuah fungsi di JavaScript dengan nama “tambahBaptisModal” |
| 110 | Memanggil sebuah document dengan id “cmd” yang akan di isikan value “add” |
| 111 | Memanggil sebuah document dengan id “addBaptis” serta memanggil fungsi modal yang akan dipanggil “show” |
| 112 | Memanggil sebuah class dengan nama “modal-title” serta disikan HTML dengan value yang diambil dari library fontawesome5 yakni icon serta diberikan tulisan “Tambah Baptis” |
| 113 | Memanggil sebuah class dengan nama “modal-footer” dan “button yang bertipe submit" yang artinya memanggil class name dengan nama modal footer yang berHTML button serta bertipe submit yang akan di isikan sebuah value HTML yang diambil dari library fontawesome5 yakni sebuah icon serta diberikan tulisan “Tambah Data” |
| 114 | Memanggil sebuah document dengan id “tampilBaptisModal” yang akan diberikan sebuah attribute action serta diset kemana data ini akan dikirimkan |
| 115 | Memanggil sebuah fungsi ajax dari library jQuery |
| 116 | Memanggil variabel url yang ada didalam fungsi ajax jQuery yang diisikan value dari routingan controller tujuan dengan memanfaatkan fungsi plugin dari framework laravel yakni "url" |
| 117 | Memanggil variabel headers yang ada di dalam fungsi ajax jQuery yang disikan value |
| 118 | Mengisi value dari variabel headers dengan sebuah token dikarenakan method yang digunakan untuk pengiriman data adalah "POST" sehingga adanya penjagaan bahwa data hanya bisa dikirimkan 1x oleh sebab itu diberikan sebuah token |
| 119 | Menutup pemanggilan variabel headers yang dibuka pada line 116 |
| 120 | Memanggil variabel data yang ada didalam fungsi ajax jQuery yang disikan value dari document dengan id “cmd” yakni valuenya adalah “add” yang disimpan di dalam sebuah varibel baru dengan nama "cmd" |
| 121 | Memanggil variabel method yang ada didalam fungsi ajax jQuery yang disikan oleh method yang digunakan yakni "POST" |
| 122 | Membuat response apakah pengiriman data berhasil dilakukan atau tidak, pada baris ini akan menampilkan response apabila data berhasil dikirim dengan response yang dikirim akan ditampung oleh sebuah parameter dengan nama "data" |
| 123 | Memanggil sebuah document yang memiliki id "isiBaptisModal" dimana document tersebut akan diisikan sebuah data HTML di dalamnya |
| 124 | Menutup hasil response sukses yang dibuka pada line 122 |
| 125 | Menutup pemanggilan fungsi ajax yang dibuka pada line 115 |
| 126 | Menutup fungsi tambahBaptisModal yang dibuka pada line 109 |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 127 | Membuat sebuah fungsi pada JavaScript dengan nama “editBaptisModal” yang berisikan sebuah parameter dengan data yang ditampung disimpan oleh variabel “baptism\_id” |
| 128 | Memanggil sebuah document dengan id “cmd” yang akan di isikan value “edit” |
| 129 | Memanggil sebuah document dengan id “addBaptis” serta memanggil fungsi modal yang akan dipanggil “show” |
| 130 | Memanggil sebuah class dengan nama “modal-title” serta disikan HTML dengan value yang diambil dari library fontawesome5 yakni icon serta diberikan tulisan “Edit Baptis” |
| 131 | Memanggil sebuah class dengan nama “modal-footer” dan “button yang bertipe submit" yang artinya memanggil class name dengan nama modal footer yang berHTML button serta bertipe submit yang akan di isikan sebuah value HTML yang diambil dari library fontawesome5 yakni sebuah icon serta diberikan tulisan “Edit Data” |
| 132 | Memanggil sebuah document dengan id “tampilBaptisModal” yang akan diberikan sebuah attribute action serta diset kemana data ini akan dikirimkan |
| 133 | Memanggil sebuah fungsi ajax dari library jQuery |
| 134 | Memanggil variabel url yang ada didalam fungsi ajax jQuery yang diisikan value dari routingan controller tujuan dengan memanfaatkan fungsi plugin dari framework laravel yakni "url" |
| 135 | Memanggil variabel headers yang ada di dalam fungsi ajax jQuery yang disikan value |
| 136 | Mengisi value dari variabel headers dengan sebuah token dikarenakan method yang digunakan untuk pengiriman data adalah "POST" sehingga adanya penjagaan bahwa data hanya bisa dikirimkan 1x oleh sebab itu diberikan sebuah token |
| 137 | Menutup pemanggilan variabel headers yang dibuka pada line 100 |
| 138 | Memanggil variabel data yang ada didalam fungsi ajax jQuery yang disikan value dari variabel global "current\_page\_index" yang disimpan didalam sebuah varibel baru dengan nama "page" |
| 139 | Memanggil variabel method yang ada didalam fungsi ajax jQuery yang disikan oleh method yang digunakan yakni "POST" |
| 140 | Membuat response apakah pengiriman data berhasil dilakukan atau tidak, pada baris ini akan menampilkan response apabila data berhasil dikirim dengan response yang dikirim akan ditampung oleh sebuah parameter dengan nama "data" |
| 141 | Memanggil sebuah document yang memiliki id "isiBaptisModal" dimana document tersebut akan diisikan sebuah data HTML di dalamnya |
| 142 | Menutup hasil response sukses yang dibuka pada line 140 |
| 143 | Menutup pemanggilan fungsi ajax yang dibuka pada line 133 |
| 144 | Menutup fungsi getDataBaptis yang dibuka pada line 127 |



|  |  |
| --- | --- |
| Line | Penjelasan |
| 145 | Membuat sebuah fungsi pada JavaScript dengan nama “deleteBaptis” yang berisikan sebuah parameter dengan data yang ditampung disimpan oleh variabel “baptism\_id” |
| 146 | Memanggil library sweetalert2 yakni fungsi Swal.fire |
| 147 | Memanggil variabel “title” dari library sweetalert2 dengan isi “Peringatan” |
| 148 | Memanggil variabel “text” dari library sweetalert2 dengan isi “Data akan terhapus secara permanen” |
| 149 | Memanggil variabel “icon” dari library sweetalert2 dengan isi “warning” |
| 150 | Memanggil variabel “showCancelButton” dari library sweetalert2 dengan isi “true” |
| 151 | Memanggil variabel “confirmButtonColor” dari library sweetalert2 dengan isi “#3085d6” |
| 152 | Memanggil variabel “cancelButtonColor” dari library sweetalert2 dengan isi “#d33” |
| 153 | Memanggil variabel “confirmButtonText” dari library sweetalert2 dengan isi “Ya, hapus !” |
| 154 | Memanggil variabel “cancelButtonText” dari library sweetalert2 dengan isi “Batal” |
| 155 | Fungsi dari library sweetalert dengan membuat kondisi atau hasil dengan menampung variabel tersebut kedalam sebuah parameter bernaama “result” |
| 156 | Membuat percabangan dengan mengecek apakah data pada variabel “result” dengan data “isConfirmed” bernilai true atau tidak |
| 157 | Memanggil sebuah fungsi ajax dari library jQuery |
| 158 | Memanggil variabel url yang ada didalam fungsi ajax jQuery yang diisikan value dari routingan controller tujuan dengan memanfaatkan fungsi plugin dari framework laravel yakni "url" |
| 159 | Memanggil variabel headers yang ada di dalam fungsi ajax jQuery yang disikan value |
| 160 | Mengisi value dari variabel headers dengan sebuah token dikarenakan method yang digunakan untuk pengiriman data adalah "POST" sehingga adanya penjagaan bahwa data hanya bisa dikirimkan 1x oleh sebab itu diberikan sebuah token |
| 161 | Menutup pemanggilan variabel headers yang dibuka pada line 159 |
| 162 | Memanggil variabel data yang ada didalam fungsi ajax jQuery yang disikan value dari parameter yang dikirimkan oleh fungsi “deleteBaptis” yakni “baptism\_id” yang disimpan didalam sebuah varibel baru dengan nama "baptism\_id" |
| 163 | Memanggil variabel method yang ada didalam fungsi ajax jQuery yang disikan oleh method yang digunakan yakni "POST" |
| 164 | Membuat response apakah pengiriman data berhasil dilakukan atau tidak, pada baris ini akan menampilkan response apabila data berhasil dikirim dengan response yang dikirim akan ditampung oleh sebuah parameter dengan nama "data" |
| 165 | Melakukan percabangan dari response yang dikembalikan oleh ajax, apakah response dari variabel “data” tersebut mengembalikan tulisan “success” atau tidak |
| 166 | Memanggil library sweetalert2 yakni fungsi Swal.fire |
| 167 | Memanggil variabel “icon” dari library sweetalert2 dengan isi “success” |
| 168 | Memanggil variabel “title” dari library sweetalert2 dengan isi “Berhasil” |
| 169 | Memanggil variabel “text” dari library sweetalert2 dengan isi “Data berhasil dihapus” |
| 170 | Memanggil variabel “showConfirmButton” dari library sweetalert2 dengan isi “false” |
| 171 | Memanggil variabel “timer” dari library sweetalert2 dengan isi “1500” |
| 172 | Menutup fungsi sweetalert2 yang dibuka pada line 166 |
| 173 | Menutup perbandingan yang dibuka pada line 165 |
| 174 | Memanggil fungsi getDataBaptis |
| 175 | Menutup hasil response sukses yang dibuka pada line 165 |
| 176 | Menutup pemanggilan fungsi ajax yang dibuka pada line 157 |
| 177 | Melakukan percabangan kedua jika statement pada line 156 tidak terpenuhi |
| 178 | Memanggil library sweetalert2 yakni fungsi Swal.fire |
| 179 | Memanggil variabel “icon” dari library sweetalert2 dengan isi “error” |
| 180 | Memanggil variabel “title” dari library sweetalert2 dengan isi “Batal” |
| 181 | Memanggil variabel “text” dari library sweetalert2 dengan isi “Data batal dihapus” |
| 182 | Memanggil variabel “showConfirmButton” dari library sweetalert2 dengan isi “false” |
| 183 | Memanggil variabel “timer” dari library sweetalert2 dengan isi “1500” |
| 184 | Menutup fungsi sweetalert2 yang dibuka pada line 178 |
| 185 | Menutup fungsi percabangan pada line 177 |
| 186 | Menutup fungsi response dari sweetalert2 yang dibuka pada line 155 |
| 187 | Menutup pemanggilan fungsi deleteBaptis yang dibuka pada line 145 |
| 188 | Menutup tag </script> sebagai penanda batas akhir dari javaScript |

**Hasil pengujian Web Sion WKO dengan Blackbox Testing**

**CMS (Content Management System)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Module yang Di uji** | **Aktivitas Pengujian** | **Hasil yang diharapkan** | **Hasil Uji** | **Status** |
| 1 | Menu *sidebar* pada CMS untuk tampilan Tata Ibadah | Menu *sidebar* di klik oleh admin | Menampilkan halaman Tata Ibadah | Menampilkan halaman Tata Ibadah | *Valid* |
| Menu sidebar tidak di klik oleh admin | Tidak menampilkan halaman Tata Ibadah | Halaman Tata Ibadah tidak tampil | *Valid* |
| 2 | *Button action* Tambah Ibadah | *Button* Tambah Ibadah di klik | Menampilkan sebuah *form* yang berisikan inputan untuk mengisi data Ibadah dalam bentuk modal | Modal yang berisikan inputan untuk mengisi data Ibadah berhasil tampil | *Valid* |
| *Button* Tambah Ibadah tidak di klik | Tidak akan menampilkan apapun | Tidak keluar modal | *Valid* |
| 3 | *Button action* Lihat Kategori Ibadah | *Button* Lihat Kategori Ibadah di klik | Menampilkan data dari kategori ibadah ke dalam bentuk modal | Muncul sebuah modal yang menampilkan data dari Kategori Ibadah | *Valid* |
| *Button* Lihat Kategori Ibadah tidak di klik | Tidak menampilkan apapun | Tidak memunculkan modal | *Valid* |
| 4 | *Button action Edit* pada Kategori Ibadah | *Button Edit* pada Kategori Ibadah di klik | Menampilkan data sebelumnya dari Kategori Ibadah yang diklik IDnya yang akan ditampilkan didalam sebuah modal | Muncul sebuah modal yang menampilkan data sebelumnya dari Kategori Ibadah yang di klik IDnya | *Valid* |
| *Button Edit* pada Kategori Ibadah tidak di klik | Tidak menampilkan data apapun | Tidak muncul apapun | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | *Button action Delete* pada Kategori Ibadah | *Button Delete* pada Kategori Ibadah di klik | Akan memunculkan sebuah *alert* yang akan memberikan pertanyaan apakah data akan dihapus atau tidak | Muncul sebuah *alert* yang memberikan pilihan untuk menghapus data atau tidak | *Valid* |
| *Button Delete* pada Kategori Ibadah tidak di klik | Tidak memunculkan apapun | Tidak memunculkan apapun | *Valid* |
| 6 | Menu *filter* Ibadah pada halaman Tata Ibadah | *Dropdown* pada menu *filter* ibadah akan di pilih | Menampilkan data berdasarkan menu yang dipilih pada *filter* dengan urutan yang terbaru | Data ibadah tampil berdasarkan menu *filter* yang dipilih | *Valid* |
| *Dropdown* pada menu *filter* ibadah tidak dipilih | Menampilkan semua data ibadah berdasarkan data dengan urutan yang terbaru | Semua data ibadah tampil | *Valid* |
| 7 | Button *Clear* pada halaman Tata Ibadah | Button *Clear* pada halaman Tata Ibadah di klik | Menyetel ulang data filter yang nilai *default* sehingga data yang tampil adalah semua data ibadah | Data Ibadah ditampilkan semua serta menu filter kembali ke nilai default | *Valid* |
| Button *Clear* pada halaman Tata Ibadah tidak di klik | Menampilkan data Ibadah berdasarkan menu filter yang ada | Data Ibadah masih tampil berdasarkan menu filter yang ada | *Valid* |
| 8 | *Button* *download* Data Jemaat pada menu Data Jemat | *Button* *download* Data Jemaat pada menu Data Jemaat di klik | Akan langsung melakukan download semua data Jemaat kedalam bentuk *portable document format* (*.*PDF) | Data jemaat terdownload kedalam bentuk *portable document format* (.PDF) | *Valid* |
|  |  | *Button download* Data Jemaat pada menu Data Jemaat tidak di klik | Tidak akan melakukan proses apapun | Tidak akan melakukan proses download | *Valid* |

**Tampilan User**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Module yang Di uji** | **Aktivitas Pengujian** | **Hasil yang diharapkan** | **Hasil Uji** | **Status** |
| 1 | Menu ibadah pada header | Menu ibadah di klik | Menampilkan jadwal Ibadah pada bulan saat di klik | Jadwal ibadah pada bulan saat di klik ditampilkan | *Valid* |
| Menu Ibadah tidak di klik | Tidak menampilkan jadwal ibadah | Tidak menampilkan apapun | *Valid* |
| 2 | Gambar Struktur organisasi pada bagian Tentang Kami | Gambar Struktur Organisasi di klik | Menampilkan gambar struktur organisasi gereja serta beberapa penjelasan | Gambar struktur organsasi gereja beserta penjelasan berhasil tampil | *Valid* |
| Gambar Struktur Organisasi tidak di klik | Tidak menampilkan apapun | Tidak menampilkan apapun | *Valid* |
| 3 | Form Hubungi Kami | Mengisi form yang ada pada tampilan Hubungi Kami | Data akan tersimpan serta ditampilkan pada menu Hubungi Kami | Data tersimpan serta ditampilkan pada menu Hubungi Kami | Valid |
| Tidak mengisi form pada tampilan Hubungi Kami | Data tidak akan tersimpan | Data tidak tersimpan | Valid |