

PERANCANGAN WEBSITE “PENYELAMAT PINTAR”

Dosen Pengampu : Theresia Wati S.Kom., MTI.



No	NIM	Nama
1	2210501013	Syawalia Nurul Fitri
2	2210501028	Nuzulul Firdaus
3	2210501029	Rafi Rizqallah Andila
4	2210501031	Ghaniragazzo Dzakwan

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAKARTA
TAHUN 2023

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa. Sebab karena limpahan rahmat serta anugerah dari-Nya kami bisa menyelesaikan makalah tugas akhir kami dengan judul “Penyelamat Pintar” tepat pada waktunya.

Dalam penyusunan tugas akhir ini, kami mengucapkan banyak terimakasih kepada semua pihak yang telah membantu dalam penyusunan tugas akhir ini, diantaranya:

1. Theresia Wati S.Kom., MTI.Selaku dosen Mata Kuliah Praktikum dan Teori Pemograman Web.
2. Tri Agus Setiawan selaku dosen Mata Kuliah Kelas Besar Praktikum dan Teori Pemograman Web.

Kami menyadari masih banyak kekurangan dalam penulisan tugas akhir ini. Dengan rendah hati kami meminta kritik dan saran dari pembaca makalah ini supaya selanjutnya dapat kami perbaiki kembali. Karena kami sangat menyadari, bahwa makalah yang telah kami buat ini masih memiliki banyak kekurangan.

Jakarta, 4 Desember 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang Pembuatan Project.....	1
1.2 Tujuan dan Maksud.....	1
1.3 Ruang Lingkup.....	2
BAB II	3
PEMBAHASAN	3
2.1 Teknologi dan Tools Yang Digunakan.....	3
2.2 Rancangan Basis Data Project Aplikasi.....	4
2.3 Koneksi front end dan back end.....	7
2.4 Pembuatan Form Untuk Website.....	12
BAB III	23
KESIMPULAN DAN SARAN	23
3.1 Kesimpulan.....	23
3.2 Saran.....	23

BAB I

PENDAHULUAN

1.1 Latar Belakang Pembuatan Project

Perancangan website “penyelamat pintar” adalah adanya permasalahan dalam sebuah pendidikan yang tidak bisa didapatkan oleh anak - anak yang berada di daerah - daerah terpencil. Permasalahan pendidikan ini selalu menjadi perbincangan bagaimana anak - anak bisa dapat mendapatkan pendidikan yang layak seperti anak - anak pada umumnya, karena dengan tidak adanya pendidikan yang layak untuk anak - anak di daerah terpencil akan menghasilkan anak yang kurang pengetahuan dan keterampilan, selain itu juga anak - anak tidak memahami apa hak dan kewajibannya.

Dalam mengatasi permasalahan pendidikan pada daerah terpencil, diperlukan solusi teknologi informasi yang efektif untuk memberitahu kepada banyak orang bahwa banyak anak - anak yang tidak mendapatkan pendidikan. Rancangan website “Penyelamat Pintar” untuk mempermudah orang - orang yang ingin berpartisipasi berbagi ilmu ataupun berdonasi kepada anak - anak yang di daerah terpencil membutuhkan pendidikan.

Rancangan website “Penyelamat Pintar” juga dapat menjadi alat untuk meningkatkan kesadaran masyarakat tentang keamanan, keselamatan, dan penanganan situasi darurat. Dengan kesadaran masyarakat dapat membantu anak - anak di daerah terpencil mendapatkan pendidikan yang layak dan membantu anak - anak mengembangkan potensinya.

1.2 Tujuan dan Maksud

Tujuan dari project website “Penyelamat Pintar” yaitu memberikan solusi konkret untuk mengatasi permasalahan pendidikan di daerah terpencil dan memastikan bahwa setiap anak memiliki akses yang setara dan peluang yang adil untuk mendapatkan pendidikan yang layak.

Maksud dari website “Penyelamat Pintar” adalah mengajarkan anak-anak mengenai hak-hak mereka dalam mendapatkan pendidikan dan kewajiban-kewajiban yang perlu mereka penuhi sebagai peserta didik.

1.3 Ruang Lingkup

1. Menampilkan dan memberikan akses para pengajar yang ingin mengajar di daerah terpencil dengan hati yang berempati.

2. Membuka donasi untuk membantu anak-anak di daerah terpencil yang minim pendidikan.

BAB II

PEMBAHASAN

2.1 Teknologi dan Tools Yang Digunakan

1. Virtual Studio Code (Vscode)

Visual Studio Code (VSCode) adalah editor kode sumber ringan dan powerful yang dikembangkan oleh Microsoft. VSCode mendukung banyak bahasa pemrograman, memiliki integrasi Git, debugger terintegrasi, ekstensi yang dapat diunduh, serta menyediakan fitur Intellisense untuk meningkatkan produktivitas pengembang. Dengan antarmuka yang bersih dan konfigurasi yang mudah, VSCode cocok digunakan di berbagai platform dan proyek pengembangan.

2. Mysql

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang bersifat open-source. Dikembangkan oleh Oracle Corporation, MySQL digunakan untuk menyimpan dan mengelola data dalam bentuk tabel terhubung. Ini mendukung bahasa query SQL (Structured Query Language) untuk melakukan operasi seperti pengambilan data, penyisipan, pembaruan, dan penghapusan. MySQL banyak digunakan dalam pengembangan aplikasi web dan perangkat lunak karena kecepatan, kehandalan, dan dukungan komunitas yang besar.

3. Laravel

Laravel adalah kerangka kerja pengembangan aplikasi web berbasis PHP yang bersifat open-source. Dikembangkan dengan filosofi sintaksis yang ekspresif dan elegan, Laravel menyediakan alat dan fitur-fitur modern untuk memudahkan pengembangan aplikasi, termasuk manajemen basis data, routing, templating, dan otentikasi pengguna. Laravel juga dikenal dengan fitur-fitur seperti Eloquent ORM untuk berinteraksi dengan basis data, sistem templating Blade, dan Laravel Artisan, yang merupakan utilitas baris perintah untuk tugas pengembangan.

2.2 Rancangan Basis Data Project Aplikasi

Entitas dan atribut

- Tabel User

Nama Table	Tipe Data	Keterangan
ID	BigInt(20)	Primary Key
nama	varchar	
username	text	
password	varchar	
created_at	timestamp	
updated_at	timestamp	

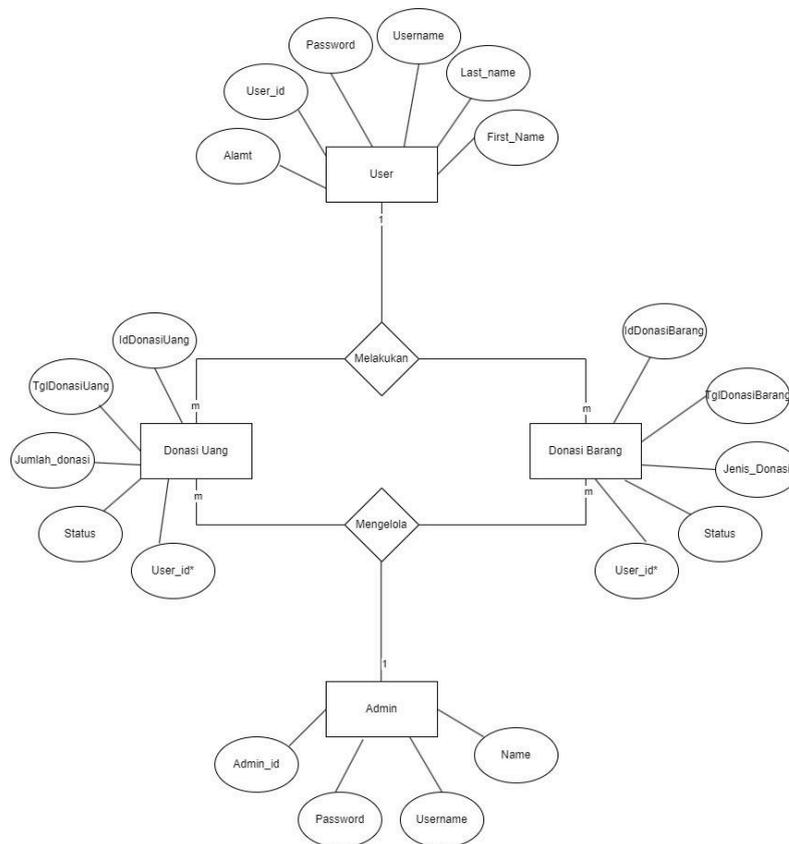
- Table goods_donations

Nama	Tipe Data	Keterangan
ID	BigInt(20)	Primary key
nama_pengirim	varchar	
alamat_pengirim	text	
bentuk_barang	text	
bukti_donasi	varchar	
created_at	timestamp	
updated_at	timestamp	

- Table Financial_donations

Nama	Tipe Data	Keterangan
ID	BigInt(20)	Primary Key
nama_pengirim	varchar	
alamat_pengirim	text	
jumlah_donasi	varchar	
bukti_donasi	varchar	
created_at	timestamp	
updated_at	timestamp	

Entity Relationship Diagram (ERD)



Implementasi Database My SQL

- Table User

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 nama	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	3 username 	text	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	4 password	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	5 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	6 updated_at	timestamp			Yes	NULL		

- Table goods_donations

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 nama_pengirim	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	3 alamat_pengirim	text	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	4 bentuk_barang	text	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	5 bukti_donasi	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	6 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	7 updated_at	timestamp			Yes	NULL		

- Table Financial_donations

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 nama_pengirim	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	3 alamat_pengirim	text	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	4 jumlah_donasi	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	5 bukti_donasi	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	6 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	7 updated_at	timestamp			Yes	NULL		

2.3 Koneksi front end dan back end

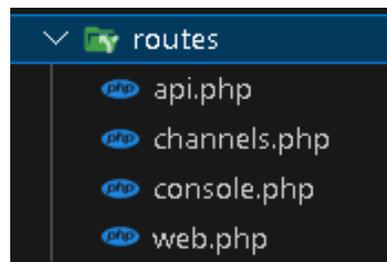
1. Koneksi Database

Langkah awal yang perlu diambil adalah memodifikasi file `.env` yang terdapat dalam kerangka kerja ini. Modifikasi yang diperlukan melibatkan penyesuaian variabel yang diperlukan untuk menghubungkan ke database, sebagaimana dijelaskan dalam ilustrasi di bawah ini:

```
.env
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=penyelamatpintar
15 DB_USERNAME=root
16 DB_PASSWORD=
```

2. Routing

Dalam penanganan URL yang diminta oleh pengguna, akan dikelola dalam direktori rute. Tata letak direktori rute dalam aplikasi ini adalah sebagai berikut:



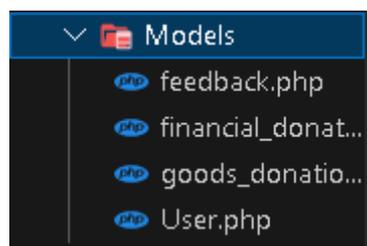
Dalam manajemen rute sebuah situs web, URL terdapat dalam file web.php di dalam direktori ini. Berikut adalah contoh tampilan rute yang diterapkan pada situs web ini:

```
routes - web.php
16 Route::get('/', function () {
17     return view('beranda');
18 });
19
20 Route::get('/profile', function () {
21     return view('profile');
22 });
23
24 Route::get('/visi_dan_misi', function () {
25     return view('visi_dan_misi');
26 });
27
28 Route::get('/cara_kerja', function () {
29     return view('cara_kerja');
30 });
31
32 Route::get('/tentang_dana_pintar', function () {
33     return view('tentang_dana_pintar');
34 });
35
36 Route::get('/amal_pintar', function () {
37     return view('amal_pintar');
38 });
39
40 Route::get('/berita_terkini', function () {
```

Di dalam berkas ini, aplikasi menangani permintaan yang dikirimkan oleh pengguna, seperti endpoint URL dan metode HTTP request yang dilakukan. Setelah menerima permintaan, berkas ini menentukan cara respons yang harus dikirimkan selanjutnya. Pengelolaan respons dapat berupa referensi ke berkas pengontrol atau langsung diatur di dalam rute itu sendiri. Selain itu, rute juga dapat dilindungi oleh middleware, seperti metode otentikasi yang harus dilakukan sebelum menjalankan beberapa fitur yang ada.

3. Models

Model adalah kontrol untuk sebuah tabel dari database yang digunakan untuk mengelola data melalui kode yang dituliskan. Sebuah model ditulis berdasarkan struktur yang ada dalam database, sehingga memudahkan interaksi antara aplikasi dan database.



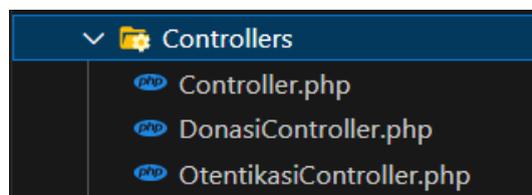
Konten dalam berkas model merupakan konfigurasi yang dilakukan untuk mengelola tabel database yang merepresentasikan model tersebut. Saat menggunakan model dalam

pembuatan aplikasi, hal ini akan mempermudah pengendalian yang terkait dengan tabel dalam database, seperti menetapkan relasi SQL, mengubah format data untuk ditampilkan, dan melakukan kueri untuk menyaring data. Secara umum, tindakan ini dilakukan dalam berkas ini, sebagaimana terlihat pada ilustrasi berkas model `financial_donations` di bawah ini.

```
financial_donations.php X
app > Models > financial_donations.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class financial_donations extends Model
8  {
9      protected $fillable = [
10         'nama_pengirim',
11         'alamat_pengirim',
12         'jumlah_donasi',
13         'bukti_donasi',
14     ];
15 }
16
```

4. Controller

Controller adalah perantara model untuk dapat ditampilkan oleh tampilan. Permintaan atau input yang melewati route kemudian diproses oleh controller, dan selanjutnya controller menentukan berkas tampilan yang ingin ditampilkan. Beberapa controller yang diterapkan pada aplikasi ini adalah seperti yang terlihat pada ilustrasi berikut:



Berkas controller bermanfaat untuk mengelola data yang diambil dari database melalui model yang telah dibuat. Kemudian, data yang telah diambil dari model diproses oleh controller untuk dapat ditampilkan di direktori tampilan sesuai keinginan. Salah satu

controller yang ada dalam aplikasi ini adalah DonasiController, yaitu berkas controller yang bertugas mengelola model financial_donations & goods_donations agar dapat ditampilkan di halaman tampilan.

```
DonasiController.php X
app > Http > Controllers > DonasiController.php
9 class DonasiController extends Controller
10 {
11     public function prosesDonasiUang(Request $request)
12     {
13         // Validasi data formulir
14         $validatedData = $request->validate([
15             'nama_pengirim' => 'required',
16             'alamat_pengirim' => 'required',
17             'jumlah_donasi' => 'required|numeric',
18             'bukti_donasi' => 'required|image|mimes:jpeg,png,jpg|max:2048',
19         ]);
20
21         // Simpan data ke database
22         $donasi = new financial_donations;
23         $donasi->nama_pengirim = $request->nama_pengirim;
24         $donasi->alamat_pengirim = $request->alamat_pengirim;
25         $donasi->jumlah_donasi = $request->jumlah_donasi;
26
27         // Upload dan simpan file bukti_donasi
28         if ($request->hasFile('bukti_donasi')) {
29             $file = $request->file('bukti_donasi');
30             $fileName = time() . '.' . $file->getClientOriginalName();
31             $file->move(public_path('bukti_transfer'), $fileName);
32             $donasi->bukti_donasi = $fileName;
33         }
34
35         $donasi->save();
36
37         // Redirect dengan pesan sukses
38         return redirect('/amal_pintar')->with('success', "Terimakasih $request->nama_pengirim telah berdonasi barang di amal pintar!");

```

Konten dalam berkas controller terdiri dari beberapa metode atau fungsi yang dimiliki oleh berkas controller tersebut. Seperti yang terlihat pada gambar di atas, terdapat metode fungsi prosesDonasiUang yang berfungsi untuk mengatur requirement pada saat input data dan mengirim data tersebut ke dalam table financial_donations. selain itu file foto yang di input akan diambil nama filenya untuk dimasukkan kedalam kolom bukti_donasi, kemudian terakhir save untuk mengirim semua data tersebut jika memenuhi requirement yang ditentukan tiap input boxnya.

5. View

Struktur view sama halnya seperti file html, yaitu untuk menampilkan tampilan antarmuka yang diakses langsung oleh user. Struktur folder view yang ada pada aplikasi ini adalah sebagai berikut:

mengelola cookie dan sesi. Selain itu, fungsi pengulangan sangat diperlukan untuk menampilkan data yang telah di input sebelumnya dengan menggunakan session yang ada didalam Laravel. Contoh penggunaan pengulangan pada berkas blade.php adalah sebagai berikut:

```
beranda.blade.php x
resources > views > beranda.blade.php > ...
177 @if(Session::has('berhasillogin'))
178 <div class="alert alert-success">
179     {{ Session::get('berhasillogin') }}
180 </div>
181 <script>
182     // Refresh halaman setelah 3 detik (3000 milidetik)
183     setTimeout(function() {
184         location.reload();
185     }, 3000);
186 </script>
187 @endif
188
189 @if(Session::has('notiflogout'))
190 <div class="alert alert-success">
191     {{ Session::get('notiflogout') }}
192 </div>
193 <script>
194     // Refresh halaman setelah 3 detik (3000 milidetik)
195     setTimeout(function() {
196         location.reload();
197     }, 3000);
198 </script>
199 @endif
```

Dengan menggunakan logika looping seperti session yang dituliskan pada kode diatas, dapat mempermudah pengembangan tampilan website. Kini notifikasi akan muncul jika if else tersebut terpenuhi dengan memanggil fungsi session tersebut.

2.4 Pembuatan Form Untuk Website

Website Penyelamat Pintar tidak hanya menggunakan form pada saat input saja, tetapi juga pada saat buat akun dan masuk akun. berikut adalah beberapa form yang digunakan pada website Penyelamat Pintar, yaitu:

1. Form Buat Akun

Form ini dibuat untuk user yang akan membuat akun agar bisa akses ke seluruh halaman yang ada pada website Penyelamat Pintar.



Buat Akun

Silahkan lengkapi form untuk buat akun

Nama

Username

Password

Buat

Sudah memiliki akun? [Masuk Akun](#)

Form ini berisi data user yang ingin mendaftarkan akun di website Penyelamat Pintar. Route yang mengelola request form buat akun ini adalah sebagai berikut:

```

62 Route::get('/buat_akun', function () {
63     return view('buat_akun');
64 });
65
66 Route::post('buatakun', [OtentikasiController::class, 'membuat'])->name('membuat');

```

Route dengan method request get adalah route yang memanggil view `buat_akun` yang berfungsi untuk menampilkan halaman form `buat_akun` yang ada pada gambar di atas. Lalu route dengan method request post untuk menggunakan fungsi `membuat` yang ada di controller `OtentikasiControlle`, berikut adalah kode form yang ada di view `buat_akun`:

```

<form action="{{ route('membuat') }}" method="post">
    @csrf
    <div class="text-wrapper">Buat Akun</div>
    <p class="div">Silahkan lengkapi form untuk buat akun</p>
    <div class="form-login">
        <div class="namalengkap">
            <div class="text-nama-lengkap"><label for="name">Nama</label>
            <input type="text" class="input-text" id="nama" name="nama" placeholder="Ketikkan nama">
            </div>
        </div>
        <div class="username">
            <div class="text-wrapper-2"><label for="username">Username</label>
            <input type="text" class="input-text" id="username" name="username" placeholder="Ketikkan username">
            </div>
        </div>
        <div class="password">
            <div class="text-wrapper-4"><label for="password">Password</label>
            <input type="password" class="input-text" id="password" name="password" placeholder="Ketikkan password">
            </div>
        </div>
        <div class="button-masuk">
            <button type="submit" class="masuk-button">Buat</button>
        </div>
    </div>
    <p class="belum-memiliki-akun">
        <span class="span">Sudah memiliki akun? </span> <a href="/masuk_akun" class="buat-akun-button">Masuk Akun</a>
    </p>
</form>

```

Ketika form diakses dan dilakukan submit oleh user, route yang berguna mengarahkan form `buat_akun` adalah route dengan metode request post. Request yang dilakukan oleh user dikirimkan ke url endpoint yang sama dengan request method get yaitu

/buat_akun. Kali ini *controller* yang berfungsi menangani request method post tersebut merupakan fungsi controller yang bernama membuat. Fungsi yang dituliskan adalah sebagai berikut:

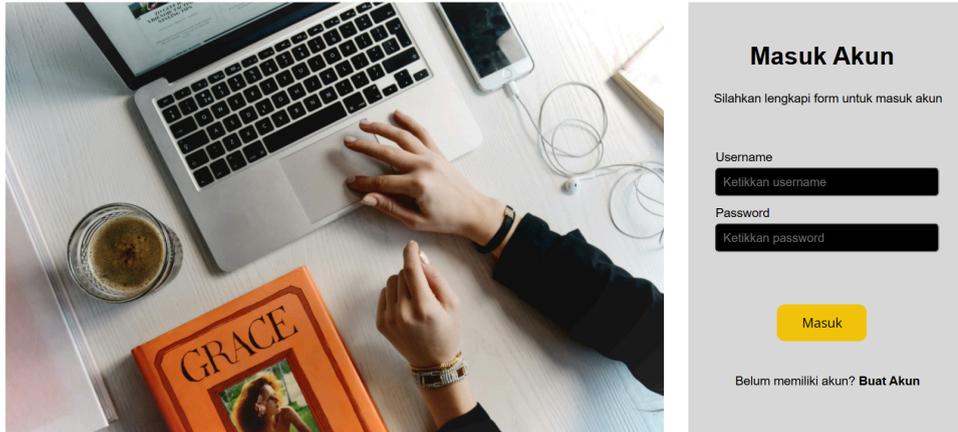
```
30
31 public function membuat(Request $request)
32 {
33     // Validasi data formulir
34     $validated = $request->validate([
35         'nama' => 'required|max:50',
36         'username' => 'required|min:5|max:20|unique:users',
37         'password' => 'required|min:8|max:20'
38     ]);
39
40     // Simpan data ke database
41     $buatakun = new User;
42     $buatakun->nama = $request->nama;
43     $buatakun->username = $request->username;
44     $buatakun->password = $request->password;
45
46     $buatakun->save();
47
48     // Redirect dengan pesan sukses
49     return redirect('/masuk_akun')->with('successCreate', "Akun $request->username berhasil dibuat, silahkan login!");
50 }
```

Fungsi tersebut menerima request dan melakukan validasi data sebelum mengirimkannya ke *database*. Hal ini berguna agar data yang diinputkan oleh user sesuai dengan data yang diminta oleh website ini. Setelah berhasil melakukan register pada bagian return, user akan diarahkan ke url /masuk_akun yaitu merupakan tampilan yang menunjukkan form login. Setelah berhasil melakukan registrasi, user akan diarahkan ke form masuk akun agar melakukan login kedalam website. Gambar dibawah ini merupakan output yang ditampilkan ketika user berhasil melakukan registrasi akun.



2. Form Masuk Akun

Form login berfungsi untuk menjadikan user sebagai pengakses aktif website ini agar dapat menjalankan beberapa fitur yang ada pada website ini. Form login yang ada pada website ini memiliki tampilan sebagai berikut:



Hanya dibutuhkan username dan password yang telah didaftarkan oleh user untuk dapat login kedalam website Penyelamat Pintar. Pada form masuk akun terdapat 2 route yang menangani request yang dilakukan oleh user. Route yang ada pada form masuk akun adalah sebagai berikut:

```
68 Route::get('/masuk_akun', function () {
69     return view('masuk_akun');
70 });
71
72 Route::post('/masuk_akun', [OtentikasiController::class, 'login']->name('login'));
73
```

Route dengan method request get adalah route yang memanggil view masuk_akun yang berfungsi untuk menampilkan halaman form masuk_akun yang ada pada gambar diatas. Lalu route dengan method request post untuk menggunakan fungsi login yang ada di controller OtentikasiController, berikut adalah kode form yang ada di view masuk_akun:

```

205 <form action="{{ route('login') }}" method="post">
206     @csrf
207     <div class="text-wrapper">Masuk Akun</div>
208     <p class="div">Silahkan lengkapi form untuk masuk akun</p>
209     <div class="form-login">
210         <div class="username">
211             <div class="text-wrapper-2">Username
212             <input type="text" class="input-text" id="username" name="username" placeholder="ketikkan username">
213             </div>
214         </div>
215         <div class="password">
216             <div class="text-wrapper-4">Password
217             <input type="password" class="input-text" id="password" name="password" placeholder="ketikkan password">
218             </div>
219         </div>
220         <div class="button-masuk">
221             <button type="submit" class="masuk-button">Masuk</button>
222         </div>
223     </div>
224     <p class="belum-memiliki-akun">
225     <span class="span">Belum memiliki akun? </span> <a href="buat_akun" class="buat-akun-button">Buat Akun</a>
226     </p>
227 </form>

```

Ketika form diakses dan dilakukan submit oleh user, route yang berguna mengarahkan form masuk_akun adalah route dengan metode request post. Request yang dilakukan oleh user dikirimkan ke url endpoint yang sama dengan request method get yaitu /masuk_akun. Kali ini *controller* yang berfungsi menangani request method post tersebut merupakan fungsi controller yang bernama login. Fungsi yang dituliskan adalah sebagai berikut:

```

9 class OtentikasiController extends Controller
10 {
11     public function login(Request $request){
12         $credentials = $request->validate([
13             'username' => 'required',
14             'password' => 'required'
15         ]);
16
17         if(Auth::attempt($credentials)){
18             $request->session()->regenerate();
19             return redirect()->intended('/')->with("berhasilLogin", "Selamat datang $request->username!!!");
20         }else{
21
22             return redirect()->intended('/masuk_akun')->with('fail', "Username atau password salah, silahkan coba kembali!");
23         }
24     }
25 }

```

Data yang dikirimkan oleh user di proses melalui validasi yang dilakukan untuk mencocokkan kesesuaian input dengan data yang diminta. Selanjutnya input di validasi kembali terhadap data yang ada pada *database*, apabila data pada *database* ada maka user akan berhasil masuk akun ke dalam website. Setelah user berhasil melakukan login, maka user akan diarahkan ke halaman home dan memiliki popup pesan berhasil login. Output yang akan ditampilkan adalah sebagai berikut:

Selamat datang rafirizqallah!!!



3. Form Konfirmasi Donasi Keuangan

Form ini dibuat untuk konfirmasi donasi keuangan pada website Penyelamat Pintar, berikut adalah halaman form konfirmasi donasi keuangan.

Konfirmasi Donasi

Form Konfirmasi

Silahkan lengkapi data untuk konfirmasi donasi keuangan

Nama lengkap*

Alamat*

Jumlah nominal*

Bukti transfer*

 Tidak ada file yang dipilih

Ketika form konfirmasi donasi keuangan di input dan submit data tersebut akan dikirimkan kedalam database. Berikut adalah route yang dibutuhkan untuk form konfirmasi donasi keuangan.

```

51
52 Route::get('/konfirmasi_donasi_uang', function () {
53     return view('konfirmasi_donasi_uang');
54 }->middleware('auth');
55
56 Route::post('/proses-kirim-bukti-uang', [DonasiController::class, 'prosesDonasiUang']->name('prosesDonasiUang'));
57

```

Route dengan method request get adalah route yang memanggil view konfirmasi_donasi_keuangan yang berfungsi untuk menampilkan halaman form konfirmasi_donasi_keuangan yang ada pada gambar diatas. Penggunaan middleware dengan fungsi auth adalah untuk verifikasi apakah user sudah login atau belum. Lalu route dengan method request post untuk menggunakan fungsi prosesDonasiUang yang ada di controller DonasiController, berikut adalah kode form yang ada di view konfirmasi_donasi_keuangan:

```

sources > view > @ konfirmasi_donasi_uang.blade.php > ...
188 <form action="{{ route('prosesDonasiUang') }}" method="POST" enctype="multipart/form-data">
189 @csrf
190 <div class="konfirmasi-donasi">
191 <div class="div">
192 <div class="header">
193 <h1 class="jumlah-awal-piatar">Konfirmasi Donasi / 83</h1>
194 <div class="donasi-keuangan">
195 <div class="kotak-kuning">
196 <div class="jumlah-donasi">Form Konfirmasi</div>
197 <div class="silakan-isi-data">Silakan mengisi data untuk konfirmasi donasi keuangan</div>
198 <div class="paris-putih"></div>
199 <div class="form-group">
200 <div class="form">
201 <div class="nama-pengirim" class="text-wrapper-2">Nama lengkap</div>
202 <div class="kolom-nama">
203 <div class="form-group">
204 <input type="text" placeholder="ketikkan nama lengkap" id="nama_pengirim" name="nama_pengirim" class="u-input u-input-rectangle" required="">
205 </div>
206 </div>
207 </div>
208 <div class="form-group">
209 <div class="form">
210 <div class="alamat-pengirim" class="text-wrapper-4">Alamat</div>
211 <div class="kolom-alamat">
212 <div class="form-group">
213 <input type="text" placeholder="ketikkan alamat" id="alamat_pengirim" name="alamat_pengirim" class="u-input u-input-rectangle" required="">
214 </div>
215 </div>
216 <div class="form-group">
217 <div class="form">
218 <div class="jumlah-donasi" class="text-wrapper-5">Jumlah nominal</div>
219 <div class="kolom-jumlah-nominal">
220 <div class="form-group">
221 <input type="text" placeholder="ketikkan jumlah nominal" id="jumlah_donasi" name="jumlah_donasi" class="u-input u-input-rectangle" required="">
222 </div>
223 </div>
224 </div>
225 <div class="form-group">
226 <div class="form">
227 <div class="bukti-donasi" class="text-wrapper-6">Bukti transfer</div>
228 <div class="kolom-bukti-transfer">
229 <div class="form-group">
230 <input type="file" id="bukti_donasi" name="bukti_donasi" class="u-input u-input-rectangle" accept="image/*" required="">
231 </div>
232 </div>
233 <div class="button-metode">
234 <button type="submit">KIRIM BUKTI</button>
235 </div>
236 </div>
237 </div>

```

Ketika form diakses dan dilakukan submit oleh user, route yang berguna mengarahkan form konfirmasi_donasi_keuangan adalah route dengan metode request post. Request yang dilakukan oleh user dikirimkan ke url endpoint yang sama dengan request method get yaitu /konfirmasi_donasi_keuangan. Kali ini *controller* yang berfungsi menangani request method post tersebut merupakan fungsi controller yang bernama prosesDonasiUang. Fungsi yang dituliskan adalah sebagai berikut:

```

11 public function prosesDonasiUang(Request $request)
12 {
13     // Validasi data formulir
14     $validatedData = $request->validate([
15         'nama_pengirim' => 'required',
16         'alamat_pengirim' => 'required',
17         'jumlah_donasi' => 'required|numeric',
18         'bukti_donasi' => 'required|image|mimes:jpeg,png,jpg|max:2048',
19     ]);
20
21     // Simpan data ke database
22     $donasi = new financial_donations;
23     $donasi->nama_pengirim = $request->nama_pengirim;
24     $donasi->alamat_pengirim = $request->alamat_pengirim;
25     $donasi->jumlah_donasi = $request->jumlah_donasi;
26
27     // Upload dan simpan file bukti_donasi
28     if ($request->hasFile('bukti_donasi')) {
29         $file = $request->file('bukti_donasi');
30         $fileName = time() . '.' . $file->getClientOriginalName();
31         $file->move(public_path('bukti_transfer'), $fileName);
32         $donasi->bukti_donasi = $fileName;
33     }
34
35     $donasi->save();
36
37     // Redirect dengan pesan sukses
38     return redirect('/amal_pintar')->with('success', "Terimakasih $request->nama_pengirim telah berdonasi barang di amal pintar!");
39 }

```

Pada fungsi ini data yang di input akan diperiksa apakah sudah memenuhi required yang telah diatur oleh prosesDonasiKeuangan atau tidak. Setelah itu data akan dikirim ke database dengan nama table financial_donations, lalu pada pengiriman foto bukti akan diambil nama filenya untuk dikirim ke kolom bukti_donasi. Kemudian jika semua proses berhasil dijalankan, maka halaman akan dialihkan ke /amal_pintar dengan popup session 'success'. Berikut adalah tampilan dari popup session 'success'.



4. Form Donasi Barang

Form ini dibuat untuk konfirmasi donasi keuangan pada website Penyelamat Pintar, berikut adalah halaman form konfirmasi donasi barang.

Konfirmasi Donasi



Ketika form konfirmasi donasi barang di input dan submit data tersebut akan dikirimkan kedalam database. Berikut adalah route yang dibutuhkan untuk form konfirmasi donasi barang.

```
58 Route::get('/konfirmasi_donasi_barang', function () {
59     return view('konfirmasi_donasi_barang');
60 })->middleware('auth');
61
62 Route::post('/proses-kirim-bukti-barang', [DonasiController::class, 'prosesDonasiBarang']->name('prosesDonasiBarang'));
63
```

Route dengan method request get adalah route yang memanggil view konfirmasi_donasi_barang yang berfungsi untuk menampilkan halaman form konfirmasi_donasi_barang yang ada pada gambar diatas. Penggunaan middleware dengan fungsi auth adalah untuk verifikasi apakah user sudah login atau belum. Lalu route dengan method request post untuk menggunakan fungsi prosesDonasiUang yang ada di controller DonasiController, berikut adalah kode form yang ada di view konfirmasi_donasi_barang:

```

188 <form action="{ route('prosesDonasiBarang') }}" method="POST" enctype="multipart/form-data">
189 @csrf
190 <div class="konfirmasi-donasi">
191 <div class="div">
192 <div class="overlap">
193 <div class="section">
194 <h3 class="judul-smal-pinter">Konfirmasi Donasi</h3>
195 <div class="donasi-keuangan">
196 <div class="bentuk-bundling">
197 <div class="judul-donasi">Form Konfirmasi</div>
198 <p class="p">Silahkan lengkapi data untuk konfirmasi donasi barang anda:</p>
199 <div class="group-input">
200 <div class="u-form-group">
201 <label for="nama_pengirim" class="text-wrapper-2">Nama lengkap</label>
202 <div class="kolom-nama">
203 <div class="overlap-group">
204 <input type="text" placeholder="ketikkan nama lengkap" id="nama_pengirim" name="nama_pengirim" class="u-input u-input-rectangle" required="">
205 </div>
206 </div>
207 </div>
208 <div class="u-form-group">
209 <label for="alamat_pengirim" class="text-wrapper-4">Alamat</label>
210 <div class="kolom-alamat">
211 <div class="overlap-group">
212 <input type="text" placeholder="ketikkan alamat" id="alamat_pengirim" name="alamat_pengirim" class="u-input u-input-rectangle" required="">
213 </div>
214 </div>
215 </div>
216 <div class="u-form-group">
217 <label for="jumlah_donasi" class="text-wrapper-5">Bentuk barang</label>
218 <div class="kolom-jumlah-donasi">
219 <div class="overlap-group">
220 <input type="text" placeholder="ketikkan bentuk barang" id="bentuk_barang" name="bentuk_barang" class="u-input u-input-rectangle" required="">
221 </div>
222 </div>
223 </div>
224 <div class="u-form-group">
225 <label for="bukti_donasi" class="text-wrapper-6">Foto barang</label>
226 <div class="kolom-bukti-transfer">
227 <div class="overlap-group">
228 <input type="file" id="bukti_donasi" name="bukti_donasi" class="u-input u-input-rectangle" accept="image/*" required="">
229 </div>
230 </div>
231 </div>
232 <div class="button-metode">
233 <button type="submit">IKRIM BUKTI</button>
234 </div>
235 </div>

```

Ketika form diakses dan dilakukan submit oleh user, route yang berguna mengarahkan form konfirmasi_donasi_barang adalah route dengan methode request post. Request yang dilakukan oleh user dikirimkan ke url endpoint yang sama dengan request method get yaitu /konfirmasi_donasi_barang. Kali ini *controller* yang berfungsi menangani request method post tersebut merupakan fungsi controller yang bernama prosesDonasiBarang. Fungsi yang dituliskan adalah sebagai berikut:

```

40
41 public function prosesDonasiBarang(Request $request)
42 {
43     // Validasi data formulir
44     $validatedData = $request->validate([
45         'nama_pengirim' => 'required',
46         'alamat_pengirim' => 'required',
47         'bentuk_barang' => 'required',
48         'bukti_donasi' => 'required|image|mimes:jpeg,png,jpg,gif|max:2048',
49     ]);
50
51     // Simpan data ke database
52     $donasi = new goods_donations;
53     $donasi->nama_pengirim = $request->nama_pengirim;
54     $donasi->alamat_pengirim = $request->alamat_pengirim;
55     $donasi->bentuk_barang = $request->bentuk_barang;
56
57     // Upload dan simpan file bukti_donasi
58     if ($request->hasFile('bukti_donasi')) {
59         $file = $request->file('bukti_donasi');
60         $fileName = time() . '-' . $file->getClientOriginalName();
61         $file->move(public_path('bukti_barang'), $fileName);
62         $donasi->bukti_donasi = $fileName;
63     }
64
65     $donasi->save();
66
67     // Redirect dengan pesan sukses
68     return redirect('/amal_pintar')->with('success', "Terimakasih $request->nama_pengirim telah berdonasi barang di amal pintar!");
69 }
70

```

Pada fungsi ini data yang di input akan diperiksa apakah sudah memenuhi required yang telah diatur oleh prosesDonasiBarang atau tidak. Setelah itu data akan dikirim ke database dengan nama table goods_donations, lalu pada pengiriman foto bukti akan diambil nama filenya untuk dikirim ke kolom bukti_barang. Kemudian jika semua proses berhasil dijalankan, maka halaman akan dialihkan ke /amal_pintar dengan popup session 'success'. Berikut adalah tampilan dari popup session 'success'.



BAB III

KESIMPULAN DAN SARAN

3.1 Kesimpulan

Project Pemrograman Web ini dapat menghasilkan sebuah website donasi yang bernama “Penyelamat Pintar”, secara keseluruhan kami telah berhasil membuat website dan membantu masyarakat Indonesia yang kurang mendapatkan pembelajaran secara layak. Dalam pembuatan website “Penyelamat Pintar” terdapat fitur penting yaitu masuk akun, buat akun, konfirmasi donasi keuangan, konfirmasi donasi barang. Selain itu sangat banyak menu yang dapat diakses oleh user.

3.2 Saran

Kami sebagai developer website “Penyelamat Pintar” sangat sadar akan kekurangan website ini, maka dari itu kami memberikan hasil semaksimal mungkin dan akan mengembangkan website “Penyelamat Pintar” secara berkala dikemudian hari dengan mempertimbangkan:

1. Penguatan Aspek Kemanfaatan: Jelaskan lebih rinci dampak positif yang telah dicapai oleh website "Penyelamat Pintar" terhadap masyarakat Indonesia yang membutuhkan pembelajaran layak. Sertakan testimoni atau data konkret yang mendukung pencapaian ini.
2. Peningkatan Fitur dan Aksesibilitas: Sarankan untuk terus memperbaiki dan mengembangkan fitur-fitur yang ada dalam website. Pastikan agar pengguna dapat dengan mudah mengakses dan memanfaatkan berbagai menu yang ditawarkan.
3. Pelibatan Pengguna: Dorong untuk lebih melibatkan pengguna dalam pengembangan selanjutnya. Mungkin dengan menyelenggarakan survey atau forum diskusi online untuk mendapatkan masukan langsung dari pengguna terkait dengan pengalaman penggunaan website.
4. Pemasaran dan Kesadaran Masyarakat: Sarankan untuk meningkatkan upaya pemasaran agar lebih banyak masyarakat yang mengetahui keberadaan dan manfaat dari website ini. Kampanye sosial media atau kerjasama dengan lembaga-lembaga terkait dapat menjadi solusi.

5. Keberlanjutan dan Pemeliharaan: Ingatkan pentingnya pemeliharaan rutin dan pembaruan teknis agar website tetap berfungsi optimal. Pastikan adanya tim yang bertanggung jawab untuk memastikan kelancaran operasional dan keamanan website.
6. Kemitraan dan Kolaborasi: Sarankan untuk mencari potensi kerjasama dengan lembaga atau perusahaan lain yang memiliki visi serupa dalam mendukung pendidikan dan pembelajaran di Indonesia.
7. Evaluasi Periodik: Anjurkan untuk melakukan evaluasi periodik terhadap performa dan dampak website. Ini dapat membantu dalam mengidentifikasi area yang perlu ditingkatkan dan memastikan bahwa website terus berkontribusi secara positif bagi masyarakat.