

LAMPIRAN 1

Tabel L1. 1 Aktivitas Identifikasi Masalah

No	Aktivitas	Hasil
1	Identifikasi Masalah	Tingginya jumlah pengidap penyakit diabetes pada tahun 2021 menyita perhatian masyarakat serta rumah sakit. Menurut data <i>INTERNATIONAL DIABETES FEDERATION</i> , terdapat 19.000.000 angka pengidap diabetes masyarakat Indonesia yang berusia 20-79 tahun sepanjang tahun 2021. Jumlah tersebut menjadi yang terbesar kelima di dunia. Lalu, terdapat 94.000 pengidap diabetes di Kota Surabaya.
2	Observasi	Melakukan observasi lapangan ke RSUD Haji Surabaya, saya memperoleh data dan informasi dari ibu Santi selaku bagian SIMRS, dimana pasien yang melakukan kontrol atau konsultasi di rumah sakit pengidap penyakit diabetes mellitus. Data yang saya peroleh dari SIMRS yaitu terdapat 1.143 rata-rata perbulan pada tahun 2022 pasien pengidap penyakit diabetes mellitus melakukan control atau konsultasi.
3	Wawancara	Lalu melakukan wawancara lanjutan dengan ibu Santi terkait data konsultasi pasien pengidap penyakit diabetes mellitus dan didapatkan bermacam nama penyakit, beserta gejala yang di diagnosa oleh pasien tersebut. Setelah itu untuk memastikan nama penyakit dan juga gejala-gejala dari diabetes mellitus, dilakukan wawancara dengan dokter spesialis penyakit dalam yang paham mengenai diabetes mellitus. Dan didapatkan informasi bahwa benar untuk nama penyakit dan gejala yang ada pada data diagnosa pasien tersebut.

Halaman ini sengaja dikosongkan

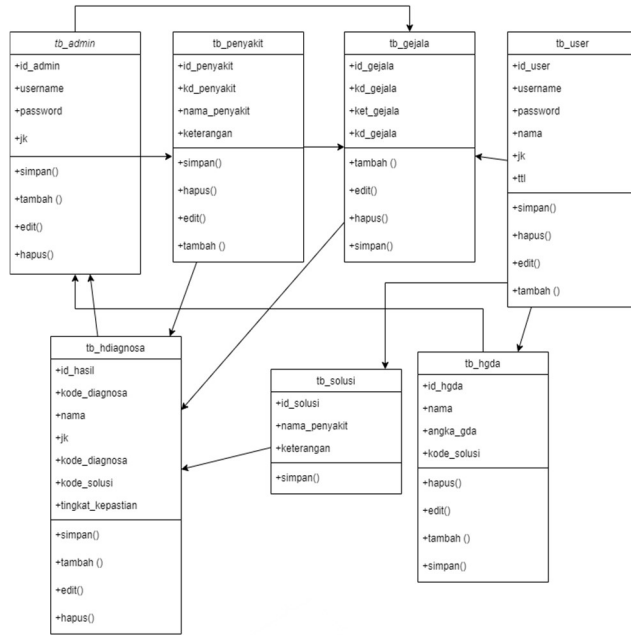
LAMPIRAN 2

Tabel L2. 1 Aktivitas Analisis dan Perancangan Sistem

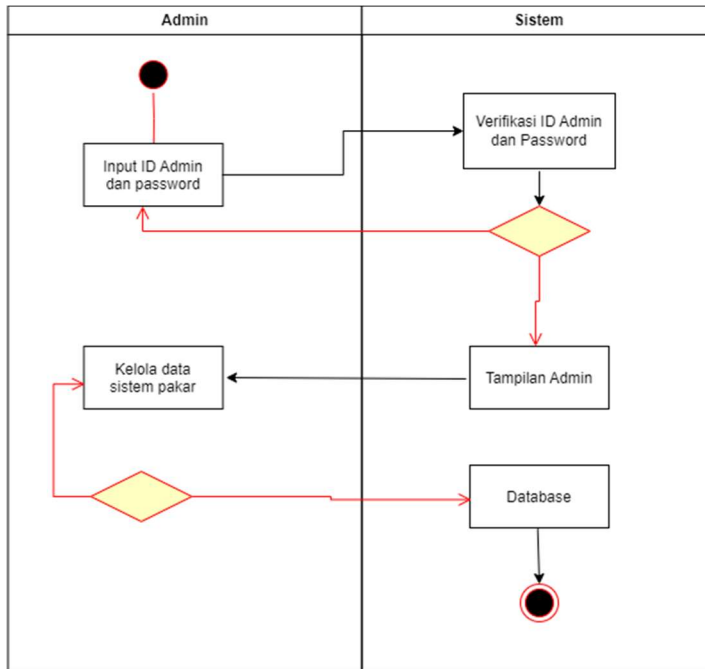
No	Aktivitas
1	<p>Merancang diagram untuk pembentukan kerangka struktur sistem yang akan dibuat seperti class diagram yang terdapat tb_admin, tb_penyakit, tb_gejala, tb_user, tb_hdiagnosa dan tb_solusi. ditunjukkan pada gambar L2.1, activity diagram dibagi menjadi dua yaitu untuk admin dan user. Activity diagram admin mengharuskan admin untuk dinput username dan password, sistem memverifikasi data login admin, jika benar maka akan tampil halaman dashboard admin, jika salah maka akan kembali ke halaman login, di dalam halaman admin, dapat mengelola seluruh data yang ada pada sistem, activity admin ditunjukkan pada gambar L2.2. untuk activity diagram user mengharuskan user melakukan login dengan memasukkan username dan password lalu diverifikasi oleh sistem, setelah itu menuju tampilan home dan dapat menjalankan fungsi yang ada pada sistem, activity user ditunjukkan pada gambar L2.3. Sequence diagram pada gambar L2.4 dan L2.5 dan use case diagram pada gambar L2.6</p>
2	<p>Membuat rancangan sistem. Pada tahap ini dilakukan dengan cara membuat desain mockup yang akan digunakan untuk tampilan sistem. Desain rancangan desain halaman login ditunjukkan pada gambar 3.10. rancangan desain halaman registrasi ditunjukkan pada gambar 3.11. rancangan desain halaman home ditunjukkan pada gambar 3.12. rancangan desain halaman biodata ditunjukkan pada gambar 3.13. rancangan desain halaman diagnose ditunjukkan pada gambar 3.14. rancangan desain halaman hasil diagnose ditunjukkan pada gambar 3.15. rancangan desain halaman info penyakit ditunjukkan pada gambar 3.16. rancangan desain halaman dashboard admin ditunjukkan pada gambar 3.17. rancangan desain halaman gejala ditunjukkan pada gambar 3.18. rancangan desain halaman penyakit ditunjukkan pada gambar 3.19. rancangan desain halaman basis aturan ditunjukkan pada gambar 3.20. rancangan desain halaman laporan diagnose ditunjukkan pada gambar 3.21. rancangan desain halaman daftar user ditunjukkan pada gambar 3.22.</p>
3	<p>Melakukan penentuan Bahasa pemrograman serta framework yang akan digunakan untuk pembuatan sistem. Bahasa pemrograman yang digunakan yaitu Bahasa Pemrograman PHP dengan menggunakan framework Laravel. Menggunakan database MySQL, dan dirunning menggunakan browser chrome.</p>

4	<p>Kemudian, tahap ini mulai pembuatan system yang diawali dengan pembuatan tampilan dari halaman awal atau homescreen hasil pembuatan halaman home ditunjukkan pada gambar 4.10, pembuatan halaman informasi penyakit ditunjukkan pada gambar 4.11, pembuatan halaman data diri pengguna ditunjukkan pada gambar 4.12, pembuatan halaman diagnose ditunjukkan pada gambar 4.13, pembuatan halaman hasil diagnose ditunjukkan pada gambar 4.14, pembuatan halaman login ditunjukkan pada gambar 4.15, pembuatan halaman registrasi ditunjukkan pada gambar 4.16, pembuatan halaman dashboard admin ditunjukkan pada halaman 4.17, pembuatan halaman gejala ditunjukkan pada gambar 4.18, pembuatan halaman penyakit ditunjukkan pada gambar 4.20 pembuatan halaman basis pengetahuan ditunjukkan pada gambar 4.22, pembuatan halaman laporan diagnose ditunjukkan pada gambar 4.23, pembuatan halaman hasil cetak laporan ditunjukkan pada gambar 4.24.</p>
5	<p>Membuat algoritma forward chaining yang akan digunakan untuk pemilihan pertanyaan pada sistem. Pembuatan algoritma forward chaining dapat dilihat pada bab 3 subbab 3.7.1 pemodelan basis pengetahuan A. Tertera Langkah-langkah untuk membuat algoritma forward chaining.</p>
6	<p>Membuat algoritma C4.5 yang akan digunakan untuk pembentukan pohon akar berdasarkan dataset yang sebelumnya sudah di inputkan. Pembuatan algoritma C4.5 dapat dilihat pada bab 3 subbab 3.7.1 pemodelan basis pengetahuan B. Tertera langkah-langkah untuk pembuatan algoritma C4.5 beserta rumus menghitung entropy yang ditunjukkan pada persamaan 3.1, lalu rumus menghitung gain ditunjukkan pada persamaan 3.2, lalu beserta perhitungan manual dari algoritma C4.5.</p>
7	<p>Setelah semua bagian sudah dibuat, kemudian dilakukan pengujian dengan melakukan percobaan satu persatu fungsi dalam sistem. Pengujian fungsi ini dilakukan menggunakan black box dan white box. Untuk pengujian black box ditunjukkan pada tabel 4.1. untuk pengujian white box dapat dilihat pada subbab 4.2, 4.2.1 pengujian whitebox.</p>

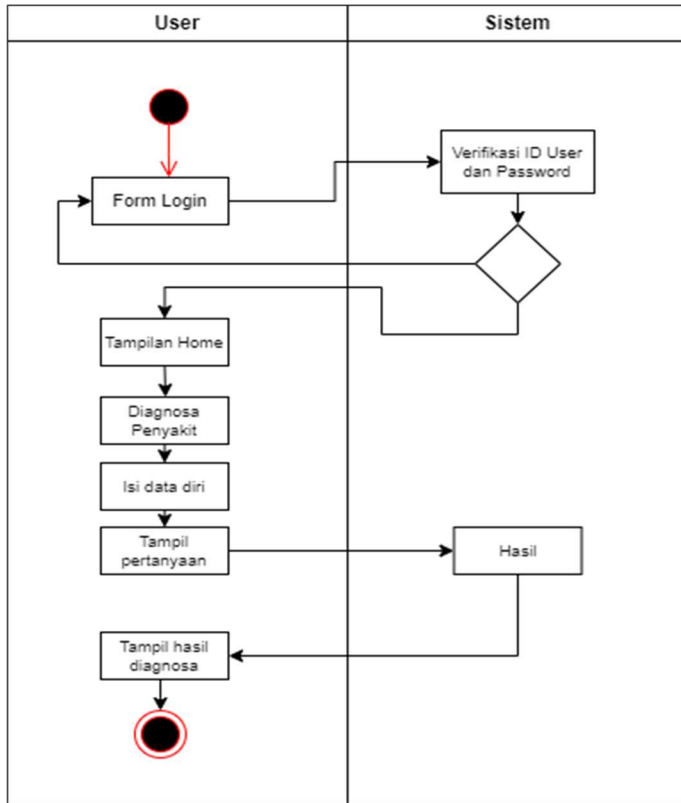
List Gambar Lampiran:



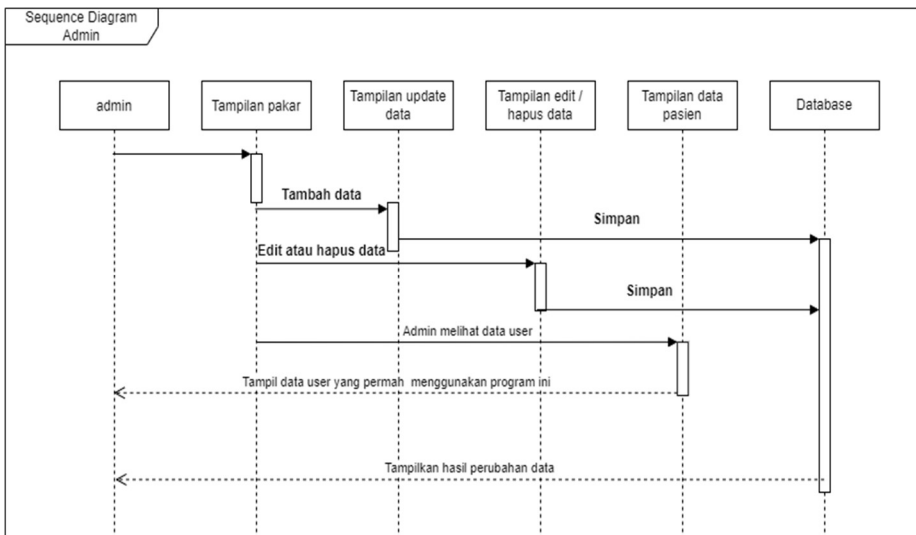
Gambar L2. 1 Class Diagram



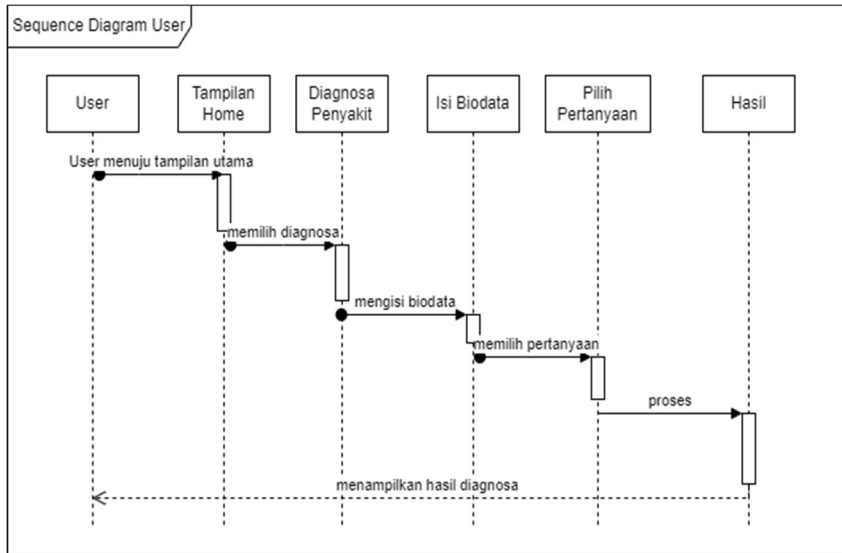
Gambar L2. 2 Activity Diagram Admin



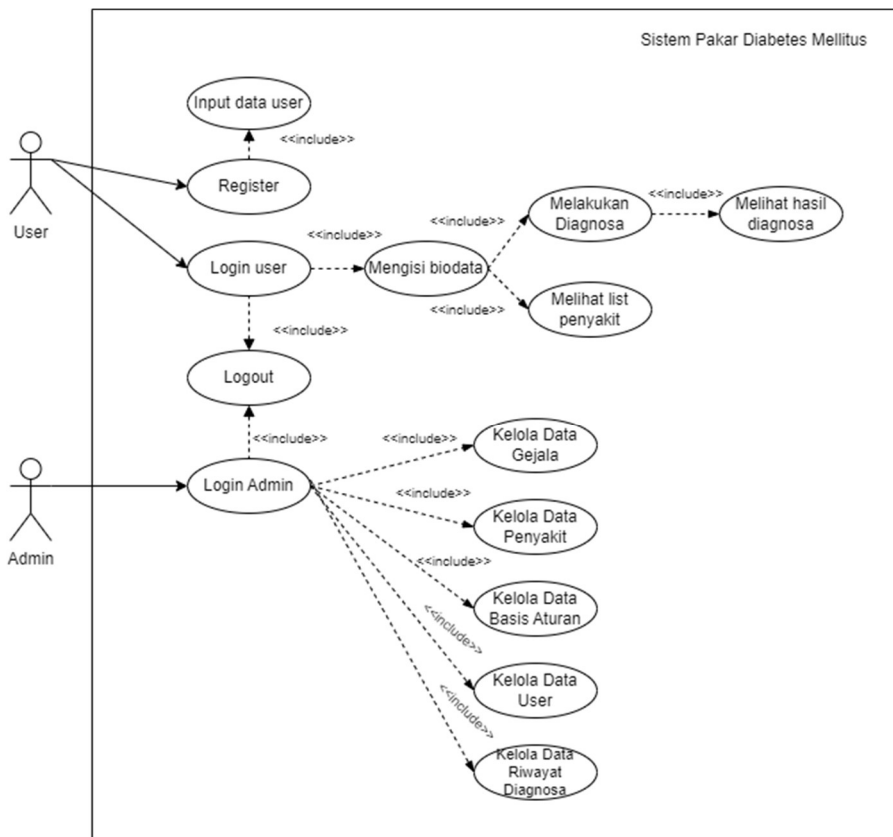
Gambar L2. 3 Activity Diagram User



Gambar L2. 4 Sequence Diagram Admin



Gambar L2. 5 Squence Diagram User


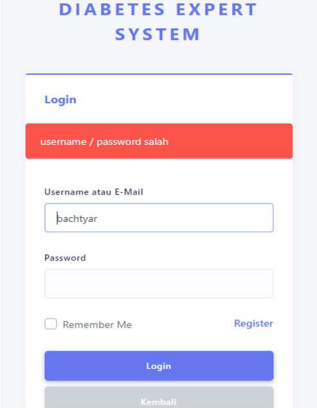


Gambar L2. 6 Use Case Diagram

Halaman ini sengaja dikosongkan

LAMPIRAN 3

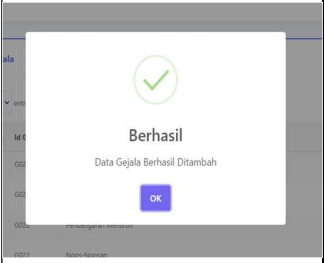
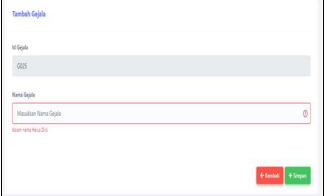
Tabel L3. 1 Uraian Aktivitas Pengujian Sistem

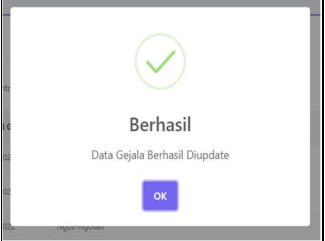
No	Aktivitas	Deskripsi	Code	Screenshoot
1	Pengujian login	<p>Pada pengujian login, hal yang ingin diuji yaitu ketika berhasil melakukan login dan ketika gagal melakukan login. Ketika berhasil melakukan login dengan memasukkan username dan password dengan benar, maka akan dilanjutkan ke halaman home. Ketika gagal melakukan login, maka akan muncul notifikasi bahwa username atau password salah dan akan tetap berada di halaman login</p>	<pre> <div id="app"> <section class="section"> <div class="container mt-5"> <div class="row"> <div class="col-12 col-sm-8 offset-sm-2 col-md-6 offset-md-3 col-lg-6 offset-lg-3 col-xl-4 offset-xl-4"> <div class="login-brand"> <strong class="text- primary"> {{ config('app.name') }} </div> <div class="card card- primary"> <div class="card-header"> <h4>{{ __ ('Login') }}</h4> </div>@error('gagal') <div class="alert alert- danger" role="alert"> {{ \$message }} </div> </div> </div> </div> </div> </section> </div> </pre>	<p>Ketika Berhasil:</p>  <p>Maka akan masuk ke halaman home</p> <p>Ketika gagal:</p> 

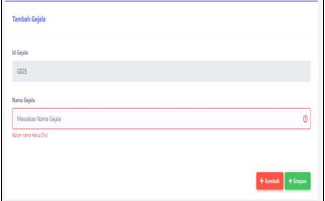
No	Aktivitas	Deskripsi	Code	Screenshoot
			<pre> @enderror <div class="card-body"> <form method="POST" action="{{route('login')}}"> @csrf <div class="form-group"> <label for="username"> {{ __('Username atau E-Mail')}}</label> <input id="username" type="text" class="form-control @error('username') is- invalid @enderror " name="username" value=" {{old('username')}}" tabindex="1" autofocus> @error('username') <div class="invalid- feedback">{{ \$message}} </div> @enderror </div> <div class="form-group"> <div class="d-block"> <label for="password" class="control-label"> </pre>	<p>Maka akan muncul peringatan</p>


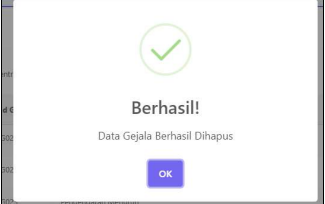
No	Aktivitas	Deskripsi	Code	Screenshot
			<pre> {{ __ ('Password') }}</label> </div> <input id="password" type="password" class="form- control @error('password') is-invalid @enderror" name="password" tabindex="2"> @error('password') <div class="invalidfeedback"> {{ \$message }} </div>@enderror </div> <div class="row"> <div class="col"> <div class="form-group"> <div class="custom-control custom-checkbox"> <input type="checkbox" name="remember" class="custom-control- input" tabindex="3" id="remember" {{ old('remember') ? 'checked' : '' }}> </pre>	


No	Aktivitas	Deskripsi	Code	Screenshot
			<pre> <label class="customcontrol-label" for="remember">{{ __ ('Remember Me') }}</label> </div> </div> </div> <div class="col">{{-- Register --}} {{ __ ('Register') }} </div> </div> <div class="form-group"> <button type="submit" class="btn btn-primary btn-lg btn-block" tabindex="4">{{ __ ('Login') }} </button> {{ __ ('Kembali') }}</div> </form> </div></div> <div class="simple-footer"> Copyright &copy; 2023 {{ </pre>	

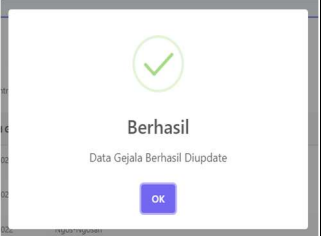
No	Aktivitas	Deskripsi	Code	Screenshoot
			<pre> config('app.name') }}</div> </div></div> </div> </section> </div> </pre>	
2	Penguujian tambah data gejala	<p>Pada penguujian tambah data gejala, dilakukan penginputan kode gejala yang sudah tergenerate secara otomatis dengan tipe variable alphanumeric, nama gejala dengan tipe variable alphanumeric. Tiap kolom pada data gejala tidak boleh kosong, jika ada yang kosong maka akan muncul peringatan data harus diisi.</p>	<pre> <form action="{{route('admin.gejala.store')}}" method="post"> @csrf <div class="card-body"> <div class="form-group"> <label for="id">Id Gejala</label> <input type="text" class="form- control @error('id') is-invalid @enderror" id="id" name="id" value="{{ \$generateId }}" readonly> <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> <div class="form-group"> <label for="nama">Nama Gejala</label> <input type="text" class="form- control @error('nama') is-invalid @enderror" id="nama" name="nama" value="{{ old('nama') }}" </pre>	<p>Ketika Berhasil:</p>  <p>Maka akan muncul modal bertuliskan data gejala berhasil ditambahkan</p> <p>Ketika gagal:</p> 

No	Aktivitas	Deskripsi	Code	Screenshoot
			<pre> placeholder="Masukkan Nama Gejala" autocomplete="off"> @error('nama') <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> </div> <div class="card-footer text-right"> <i class="fas fa- arrow-left"></i> Kembali <button type="submit" class="btn btn- success"><i class="fa fa-plus"></i> Simpan</button> </div> </form> </pre>	<p>Akan muncul peringatan bahwa kolom harus diisi tidak boleh kosong.</p>
3	Pengujian ubah data gejala	Pada pengujian ubah data gejala, Tiap kolom pada data gejala tidak boleh kosong, jika ada yang kosong maka akan muncul peringatan data harus diisi.	<pre> <form action="{{route('admin.gejala.update', \$gejala->id)}}" method="post"> @method('PUT') @csrf <div class="card-body"> <div class="form-group"> <label for="id">Id Gejala</label> <input type="text" class="form-control </pre>	<p>Ketika berhasil:</p> 



No	Aktivitas	Deskripsi	Code	Screenshoot
			<pre> @error('id') is-invalid @enderror" id="id" name="id" value="{{ \$gejala->id }}" readonly> @error('id') <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> <div class="form-group"> <label for="nama">Nama Gejala</label> <input type="text" class="form- control @error('id') is-invalid @enderror" id="nama" name="nama" value="{{ \$gejala->nama }}" placeholder="Masukkan Nama Gejala"> @error('nama') <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> </div> <div class="card-footer text-right"> <i class="fas fa- arrow-left"></i> Kembali </pre>	<p>Akan muncul modal bertuliskan data gejala berhasil diupdate</p> <p>Ketika gagal:</p>  <p>Akan muncul peringatan bahwa kolom harus diisi tidak boleh kosong.</p>

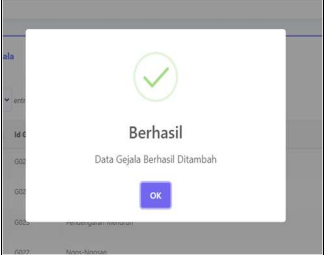
No	Aktivitas	Deskripsi	Code	Screenshoot
			<pre data-bbox="794 285 1479 441"><button type="submit" class="btn btn-success"><i class="fa fa-plus"></i> Simpan</button> </div> </form></pre>	
4	Pengujian hapus data gejala	Pada pengujian ini dilakukan penghapusan data gejala. Yaitu dengan menekan button hapus yang ada pada kolom aksi. Setelah di klik button hapus, maka data gejala yang dipilih akan terhapus dari tabel.	<pre data-bbox="707 450 1512 1170">function hapusData(id) { let url = \$('#delete_{\$id}`).attr('action'); let data = \$('#delete_{\$id}`).serialize(); let method = 'POST'; console.log(url, data, method); \$.ajax({ url: url, type: method, data: data, success: function(response) { console.log(response); Swal.fire('Berhasil!', 'Data Gejala Berhasil Dihapus', 'success') } }) setTimeout(() => { location.reload(); }, 1000); }</pre>	<p data-bbox="1566 454 1870 481">Ketika ingin menghapus:</p>  <p data-bbox="1559 735 1881 831">akan muncul modal konfirmasi apakah ingin menghapus data.</p> <p data-bbox="1559 869 1881 897">Ketika berhasil dihapus:</p> 

No	Aktivitas	Deskripsi	Code	Screenshoot
			<pre> } </script> </pre>	Akan muncul modal bertuliskan data gejala berhasil dihapus
5	Pengujian tambah data penyakit	Pada pengujian tambah data penyakit, dilakukan penginputan kode penyakit yang sudah tergenerate secara otomatis dengan tipe variable alphanumeric, nama penyakit dengan tipe variable alphanumeric. Tiap kolom pada data penyakit tidak boleh kosong, jika ada yang kosong maka akan muncul peringatan data harus diisi.	<pre> <form action="{{route('admin.gejala.store')}}" method="post"> @csrf <div class="card-body"> <div class="form-group"> <label for="id">Id Gejala</label> <input type="text" class="form-control @error('id') is-invalid @enderror" id="id" name="id" value="{{ \$generateId }}" readonly> <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> <div class="form-group"> <label for="nama">Nama Gejala</label> <input type="text" class="form-control @error('nama') is-invalid @enderror" id="nama" name="nama" value="{{old('nama')}}" placeholder="Masukkan Nama Gejala" autocomplete="off"> </pre>	Ketika Berhasil:  Maka akan muncul modal bertuliskan data penyakit berhasil ditambahkan

No	Aktivitas	Deskripsi	Code	Screenshoot
			<pre> @error('nama') <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> </div> <div class="card-footer text-right"> <i class="fas fa- arrow-left"></i> Kembali <button type="submit" class="btn btn- success"><i class="fa fa-plus"></i> Simpan</button> </div> </form> </pre>	
6	Pengujian ubah data penyakit	Pada pengujian ubah data penyakit, Tiap kolom pada data penyakit tidak boleh kosong, jika ada yang kosong maka akan muncul peringatan data harus diisi.	<pre> <form action="{{route('admin.gejala.update', \$gejala->id)}}" method="post"> @method('PUT') @csrf <div class="card-body"> <div class="form-group"> <label for="id">Id Gejala</label> <input type="text" class="form-control @error('id') is-invalid @enderror" id="id" name="id" value="{{ \$gejala->id }}" readonly> </pre>	<p>Ketika berhasil:</p>  <p>Akan muncul modal bertuliskan data penyakit berhasil diupdate</p>

No	Aktivitas	Deskripsi	Code	Screenshot
			<pre> @error('id') <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> <div class="form-group"> <label for="nama">Nama Gejala</label> <input type="text" class="form- control @error('id') is-invalid @enderror" id="nama" name="nama" value="{{ \$gejala->nama }}" placeholder="Masukkan Nama Gejala"> <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> </div> <div class="card-footer text-right"> <i class="fas fa- arrow-left"></i> Kembali <button type="submit" class="btn btn- success"><i class="fa fa-plus"></i> Simpan</button> </pre>	

No	Aktivitas	Deskripsi	Code	Screenshot
			<pre> </div> </form> </pre>	
7	Pengujian hapus data penyakit	Pada pengujian ini dilakukan penghapusan data penyakit. Yaitu dengan menekan button hapus yang ada pada kolom aksi. Setelah di klik button hapus, maka data penyakit yang dipilih akan terhapus dari tabel.	<pre> function hapusData(id) { let url = \$('#delete_\${id}`).attr('action'); let data = \$('#delete_\${id}`).serialize(); let method = 'POST'; console.log(url, data, method); \$.ajax({ url: url, type: method, data: data, success: function(response) { console.log(response); Swal.fire('Berhasil!', 'Data Penyakit Berhasil Dihapus', 'success') setTimeout(() => { location.reload(); }, 1000); } }) } </pre>	<p>Ketika ingin menghapus:</p>  <p>akan muncul modal konfirmasi apakah ingin menghapus data.</p> <p>Ketika berhasil dihapus:</p>  <p>Akan muncul modal bertuliskan data penyakit berhasil dihapus</p>

No	Aktivitas	Deskripsi	Code	Screenshoot
8	Pengujian tambah data pengetahuan	Pada pengujian tambah data pengetahuan, dilakukan pemilihan nama penyakit serta nama gejala. Tiap kolom pada data pengetahuan tidak boleh kosong, jika ada yang kosong maka akan muncul peringatan data harus diisi.	<pre> <form action="{{ route('admin.bp.store') }}" method="post"> @csrf <div class="card-body"> {{ show_error(\$errors) }} <div class="form-group"> <label for="penyakit">Nama Penyakit</label> <select class="form-control cb @error('penyakit_id') is-invalid @enderror id="penyakit" name="penyakit_id"> <option disabled selected>-- Pilih Penyakit --</option> @foreach (\$penyakits as \$penyakit) <option value="{{ \$penyakit->id }}">{{ \$penyakit->nama }}</option> @endforeach </select> @error('penyakit_id') <div class="invalid-feedback"> {{ \$message }} </div> @enderror </div> <div class="form-group row"> <div class="col"> <label for="gejala">Nama Gejala</label> </pre>	<p>Ketika Berhasil:</p>  <p>Maka akan muncul modal bertuliskan data pengetahuan berhasil ditambahkan</p>

No	Aktivitas	Deskripsi	Code	Screenshot
			<pre>@foreach(\$gejalas as \$gejala) <div class="form-check"> <label class="form-check-label" style="text-transform: capitalize"> <input class="form-check-input" type="checkbox" name="gejala[{{ \$gejala->id }}" value="{{ \$gejala->id }}" {{ \$gejala->nama }} - {{ \$gejala->id }} </label> </div> </div> @endforeach </div> </div> </div> <div class="card-footer text-right"> <i class="fas fa-arrow-left"></i> Kembali <button type="submit" class="btn btn- success"><i class="fa fa-plus"></i> Simpan</button> </div> </form></pre>	