**BAB IV**

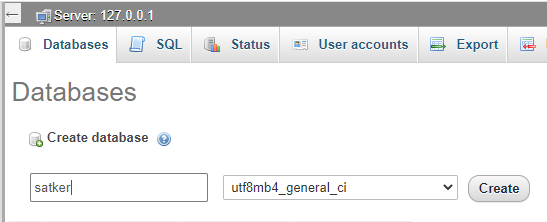
**HASIL DAN PEMBAHASAN**

Sistem Informasi Dinas Pekerjaan Umum dan Perumahan Rakyat Satker Wilayah I Provinsi Nusa Tenggara Timur Berbasis *Website* merupakan sebuah website yang dibuat untuk mempermudah masyarakat dalam memperoleh informasi terbaru pada Dinas Pekerjaan Umum dan Perumahan Rakyat Satker Wilayah I Provinsi Nusa Tenggara Timur. Setelah melakukan desain atau perancangan sistem, tahap selanjutnya adalah melakukan *coding*, pengujian sistem untuk setiap halaman ada pada sistem.

1. **Implementasi Database**
   * 1. **Pembuatan Database dengan Menggunakan PhpmyAdmin**

Langkah awal dalam pembuatan *database* ini adalah dengan menginstal aplikasi XAMPP, setelah terinstal maka phpmyAdmin akan langsung digunakan dengan menghidupkan *service* apache dan *service* mysql pada panel XAMPP itu sendiri, lalu kunjungi http://localhost/phpmyadmin atau http://127.0.0.1/phpmyadmin menggunakan browser.

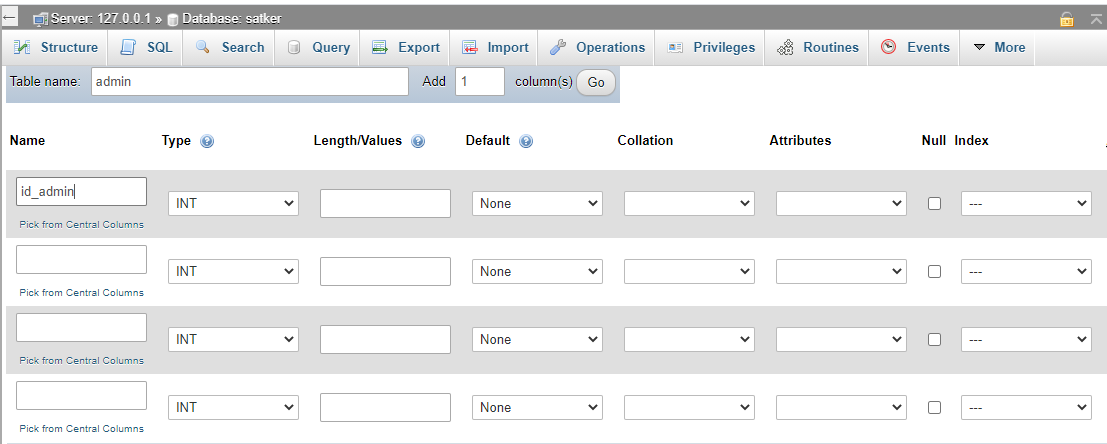
Berikut untuk membuat sebuah *database* baru dengan mengklik *tab* “*Databases*” dan mengetikan nama *database* pada kolom *Database name* selanjutnya dengan menekan tombol *create*.



Gambar 4.1 Pembuatan Database dengan Menggunakan phpmyAdmin

* + 1. **Pembuatan Tabel Database**

Pada perancangan yang ada pada gambar 3.4 ERD (*Entity Relationship Diagram*) terdapat 9 buah tabel, langkah pertama untuk membuat tabel dengan mengklik nama *database* yang sudah dibuat, klik *new*, maka akan muncul sebuah field atau kolom untuk nama tabel dan beberapa *field* untuk penamaan kolom tabel beserta dengan pilihan-pilihan tabel, salah satunya seperti *type*.



Gambar 4.2 Pembuatan Tabel pada *Database*

1. **Implementasi *Interface***
   1. **Halaman Awal Website**

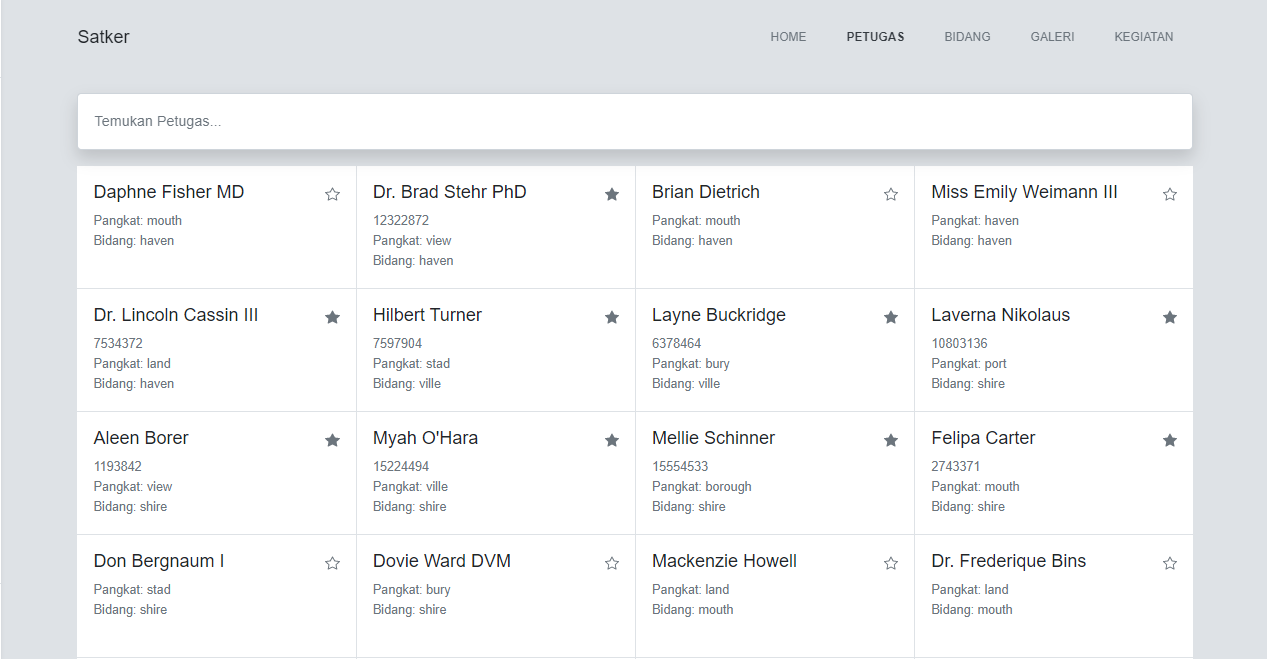
Halaman awal adalah tampilan yang akan dilihat pertama kali oleh pengunjung saat mengunjungi *website* ini, halaman ini berisi mengenai kata sambutan dan beberapa menu navigasi untuk mengakses informasi lainnya pada *website* ini, menu navigasi pada bagian atas terdiri dari navigasi ke halaman awal atau home, navigasi menuju ke halaman daftar informasi petugas, navigasi ke halaman bidang, navigasi ke halaman galeri, dan kegiatan. Sedangkan navigasi sebelah kanan terdapat ikon dengan *tooltip* *home*, struktur organisasi satker, bidang, kegiatan dan navigasi *dashboard* *dashboard* admin atau navigasi login, tergantung pada sesi login pengguna (admin atau pimpinan).



Gambar 4.3 Tampilan Halaman Awal

* 1. **Halaman Petugas**

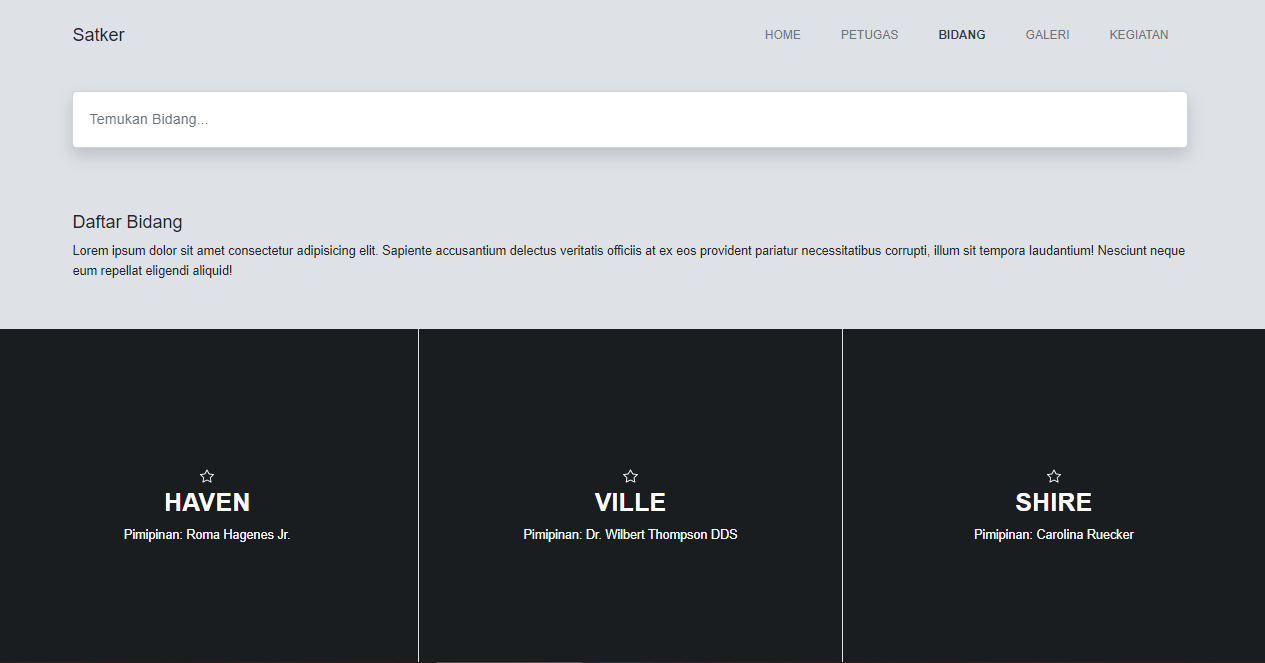
Halaman ini adalah halaman yang dapat diakses oleh siapa saja, dan halaman ini berisi sekilas informasi tentang petugas yang ada, terdapat kolom input yang berfungsi untuk men-filter daftar petugas yang ada dibawahnya, tanda bintang pada petugas menandakan status kerja petugas.



Gambar 4.4 Tampilan Halaman Petugas

* 1. **Halaman Daftar Bidang**

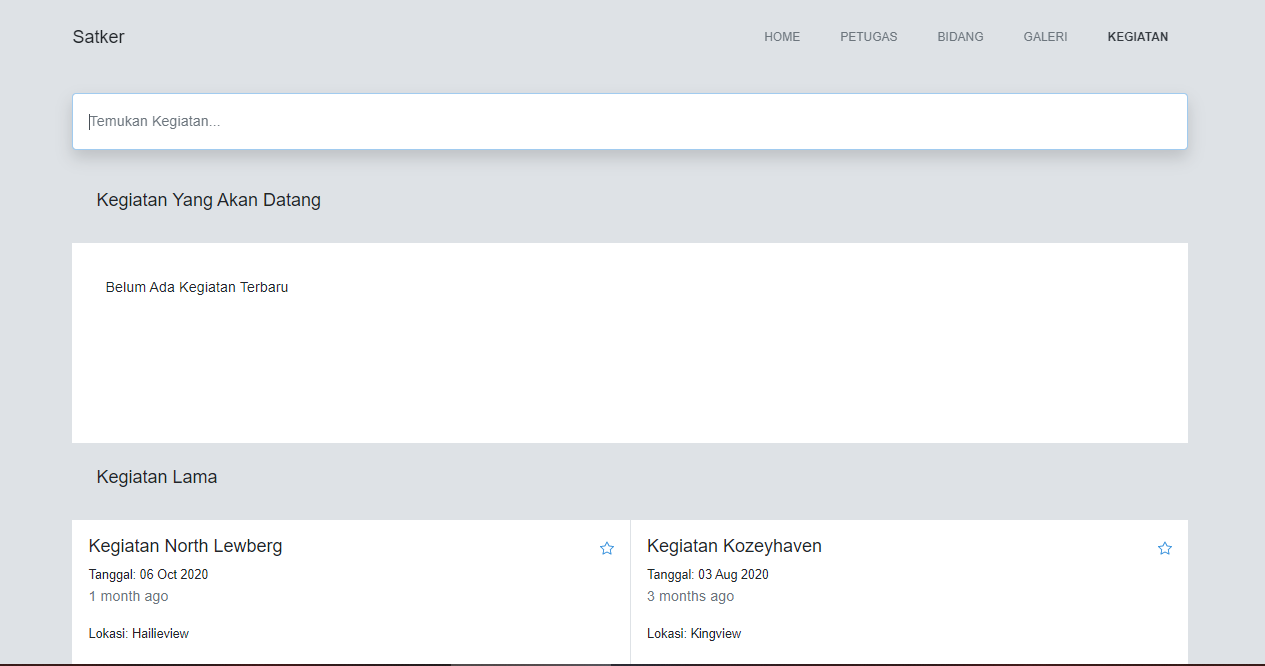
Halaman ini adalah halaman yang dapat diakses oleh siapa saja dan halaman ini berisikan informasi mengenai daftar bidang, nama pemimpin, visi-misi dan tujuan dari tiap bidang, sama seperti pada halaman petugas sebelumnya, halaman ini juga memilik sebuah kolom input untuk men-filter daftar bidang yang ada.



Gambar 4.5 Tampilan Halaman Daftar Bidang

* 1. **Halaman Informasi Kegiatan**

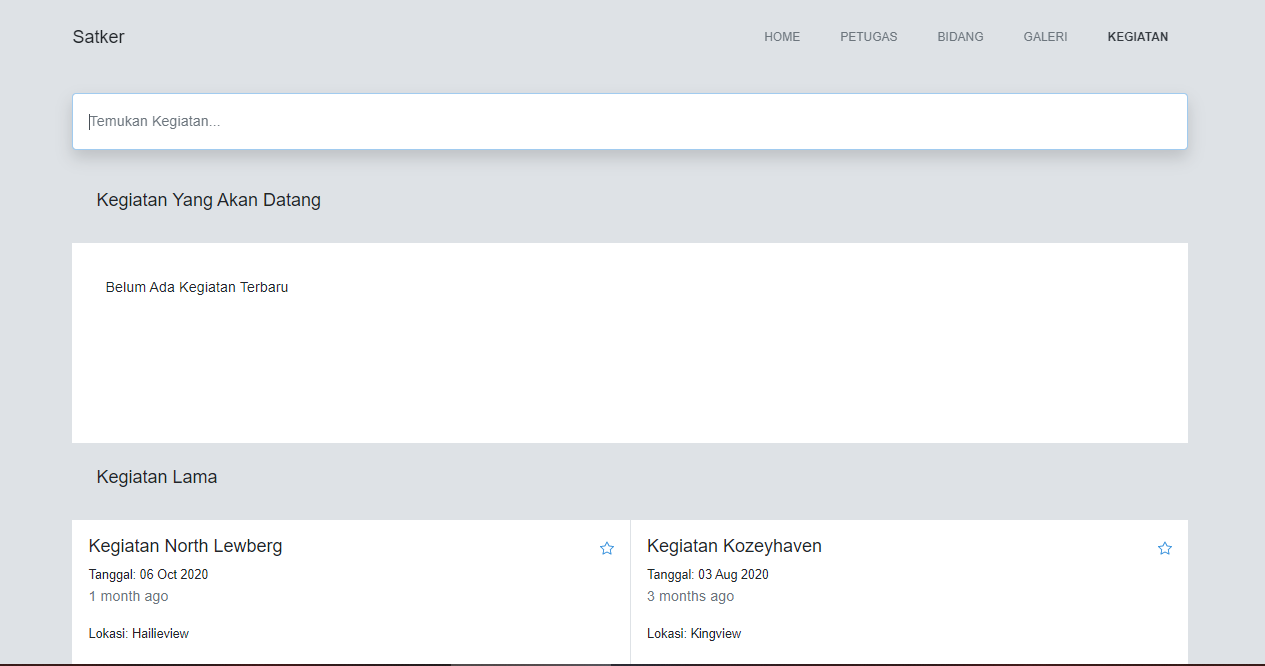
Halaman ini adalah halaman yang dapat diakses oleh siapa saja, dan halaman ini berisikan informasi mengenai kegiatan, baik kegiatan yang akan datang maupun kegiatan lampau, juga terdapat sebuah kolom input berupa pencarian kegiatan untuk menfilter daftar kegiatan yang ditampilkan.



Gambar 4.6 Tampilan Halaman Kegiatan

* 1. **Halaman Informasi Struktur Organisasi**

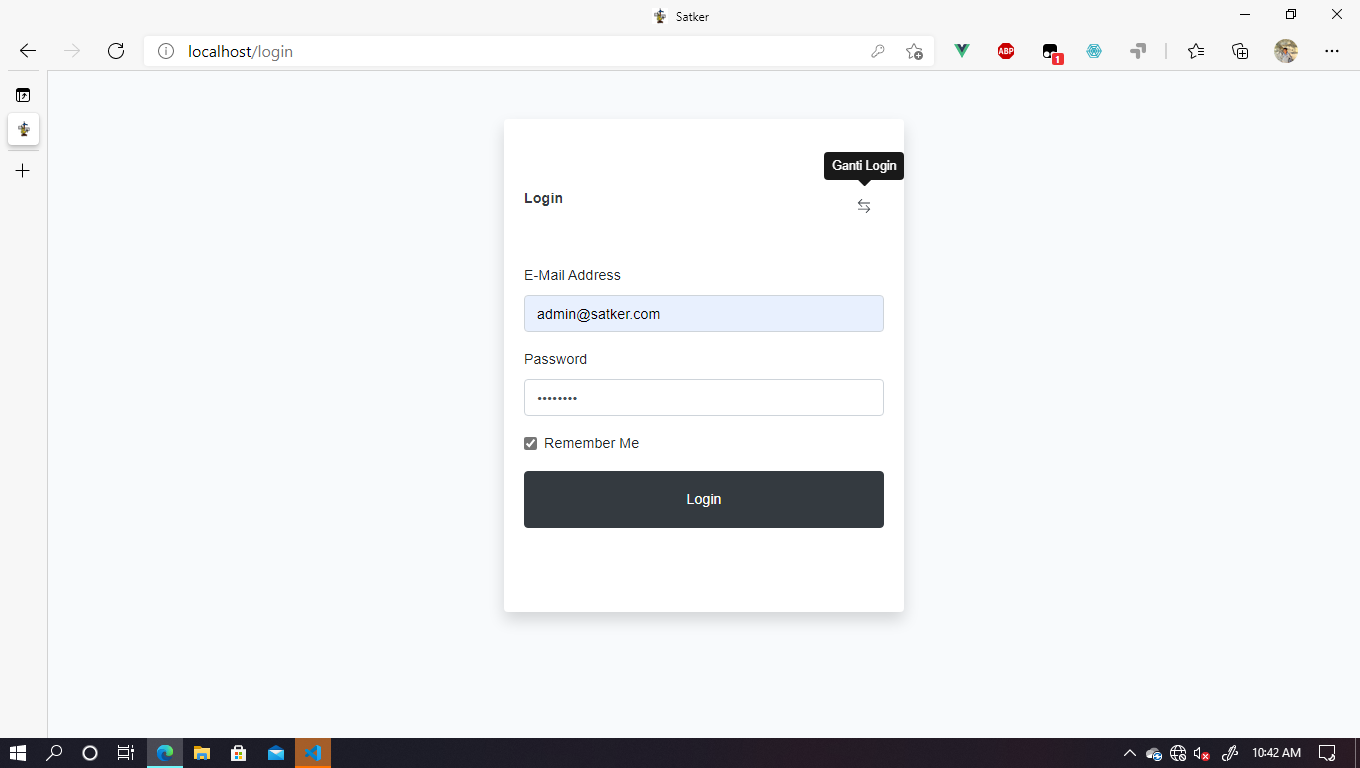
Halaman ini adalah halaman yang dapat diakses oleh siapa saja, dan halaman ini berisikan informasi mengenai struktur organisasi dari satker, juga terdapat sebuah navigasi utama pada bagian atas halaman.



Gambar 4.7 Tampilan Halaman Struktur Organisasi

* 1. **Halaman *Login***

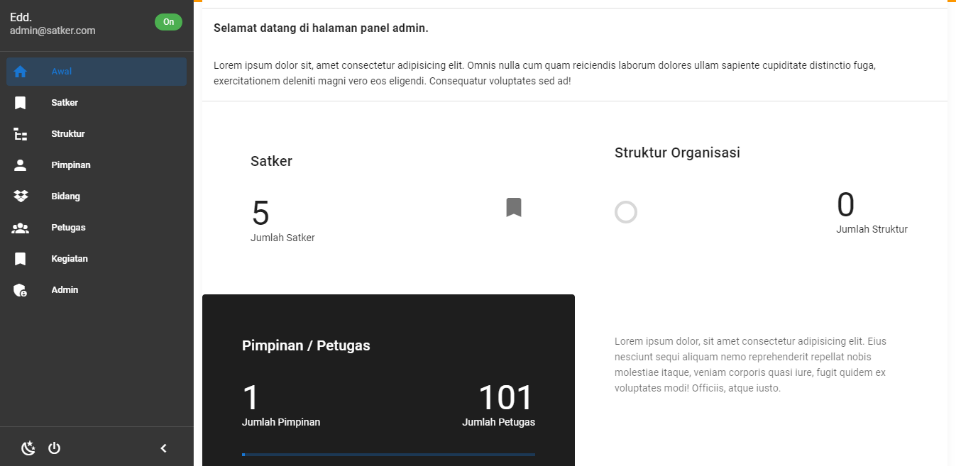
Halaman *login* merupakan halaman yang digunakan pengguna untuk membuka dan mendapatkan akses dalam menggunakan dan mengelola aplikasi, terdapat sebuah ikon 2 (dua) buah panah berlawanan arah untuk mengubah jenis *form* *login*, untuk admin dan pimpinan. Tiap form login terdapat dua buah kolom input *email* dan *password*, dan sebuah *checkbox* dengan label “*remember me*” yang berfungsi untuk mengingat sesi *login* lebih lama bahkan ketika pengguna keluar browser dan mengunjungi aplikasi atau website ini, dan dibawahnya terdapat sebuah aksi untuk memicu proses login. berikut adalah tampilan dari halaman login aplikasi.



Gambar 4.8 Tampilan Halaman *Login*

* 1. **Halaman Panel Admin**

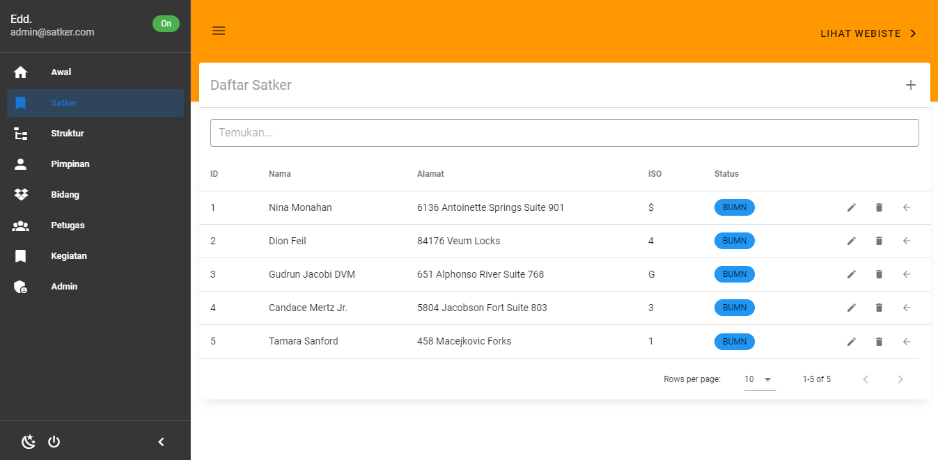
Halaman panel admin adalah halaman awal yang akan dilihat *user* atau admin ketika berhasil melewati proses login dan mendapatkan hak aksesnya dalam mengelola dan menggunakan aplikasi. Halaman ini berisikan informasi sambutan dan beberapa informasi jumlah data yang sudah tercatat pada aplikasi, dan terdapat navigasi digunakan untuk menjelajah halaman panel admin. Terdapat informasi jumlah dari satker maupun pimpinan pada halaman ini.



Gambar 4.9 Tampilan Halaman Panel Admin

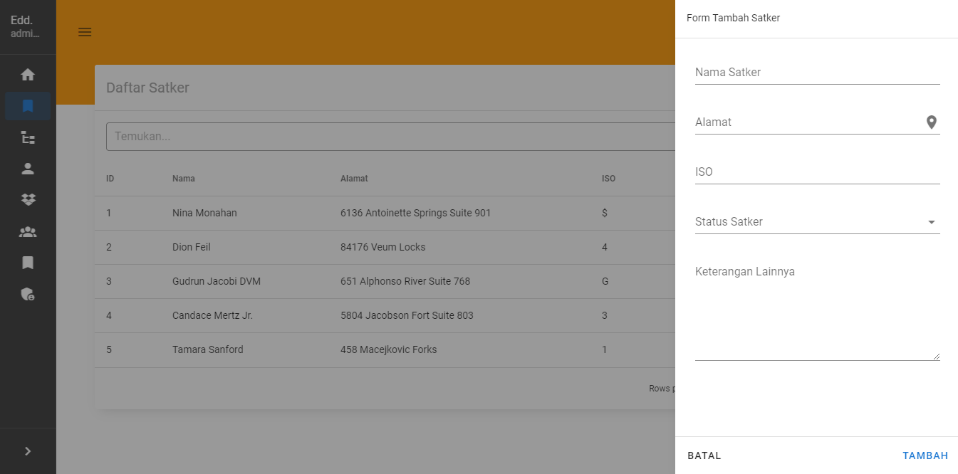
* 1. **Halaman Satker Admin**

Halaman Satker adalah halaman yang berisi aksi-aksi untuk mengelola data satker dan memuat data-data satker, pada sudut kanan atas terdapat aksi dengan ikon tambah atau “*plus*” untuk menambah data satker dan terdapat aksi untuk setiap data satker yang terletak diakhir kolom daftar data satker, terdapat ikon pensil untuk mengubah data satker tersebut, terdapat ikon tempat sampah yang berfungsi untuk menghapus data satker tersebut, sebelum menghapus akan diperlihatkan *dialog* konfirmasi penghapusan data, dan terakhir aksi dengan ikon anak panah ke-arah kiri untuk menampilkan informasi lengkap dari data satker tersebut. Adapun kolom pencarian untuk memudahkan pengguna (admin) dalam mencari data satker, berikut tampilan dari halaman satker.



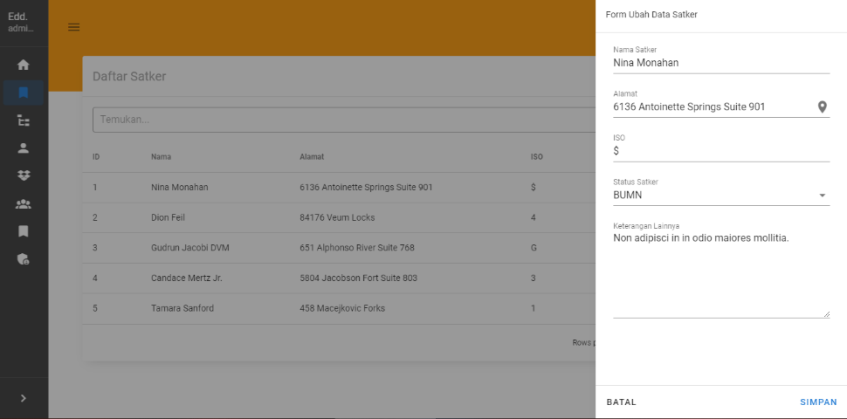
Gambar 4.10 Tampilan Halaman Satker Admin

Pada halaman ini terdapat sebuah *modal* dan *form* untuk menambahkan data satker baru, modal ini akan muncul ketika pengguna mengklik ikon tambah pada bagian sudut kanan atas halaman satker, dalam *form* ini terdapat kolom nama satker, alamat satker, nomor ISO, pilihan status satker, dan keterangan tambahan mengenai satker yang dimaksudkan, di bawahnya terdapat dua buah aksi batal untuk menutup *form* dan tambah untuk menambahkan data ke dalam *database*, berikut tampilan dari *form* tambah data satker baru:



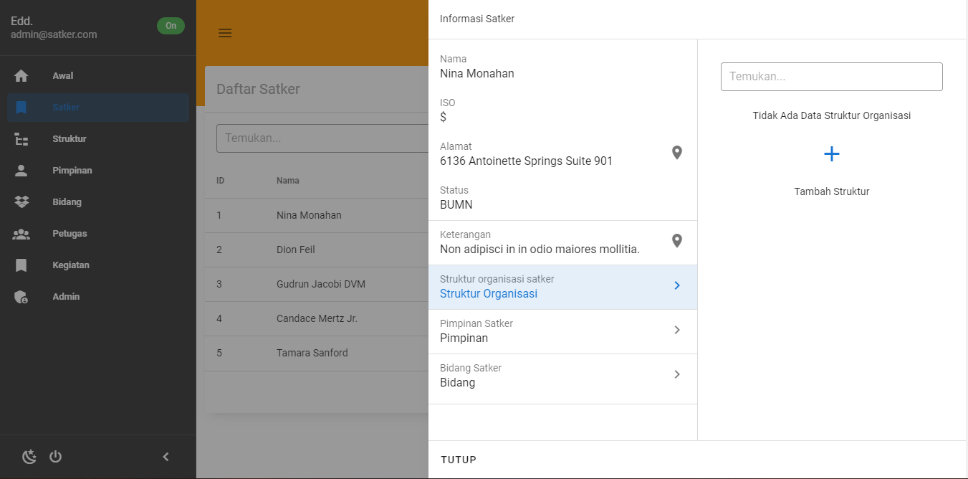
Gambar 4.11 Tampilan Form Tambah Satker

Juga terdapat *modal* dan *form* untuk mengubah data satker, modal ini akan tampil ketika user atau pengguna mengklik aksi edit (tombol dengan ikon pensil) pada setiap data satker, kolom input juga sama seperti pada form tambah satker sebelumnya, terdapat nama satker, alamat satker, ISO, status satker, dan keterangan tambahan satker, berikut tampilan *form* ubah data satker:



Gambar 4.12 Tampilan Form Ubah Data Satker

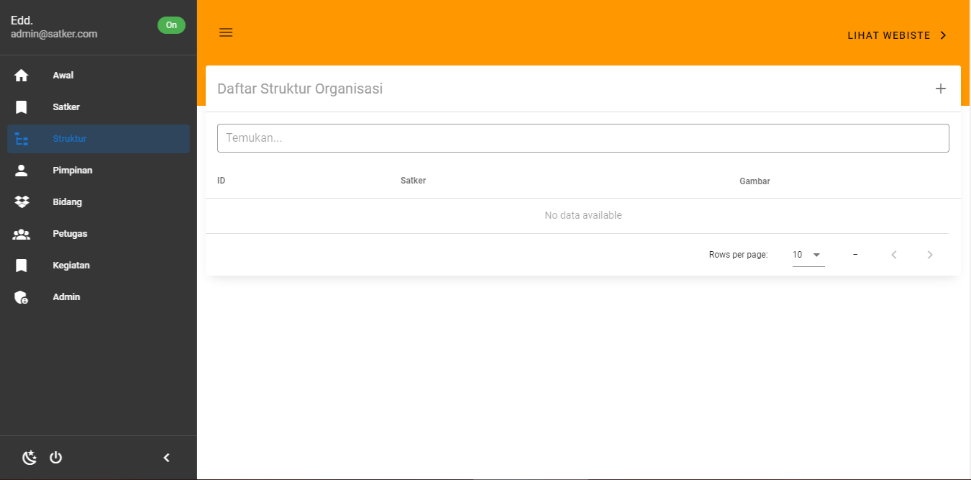
Terdapat juga *modal* untuk menapilkan rincian mengenai satker dengan mengklik tombol rincian (ikon panah kiri) yang terdapat pada setiap data satker. Pada *modal* ini terdapat rincian mengenai data satker, struktur organisasi, data pemimpin, dan juga bidang yang berelasi dengan data satker yang dipilih, berikut tampilan *modal* rincian:



Gambar 4.13 Tampilan Rincian Data Satker

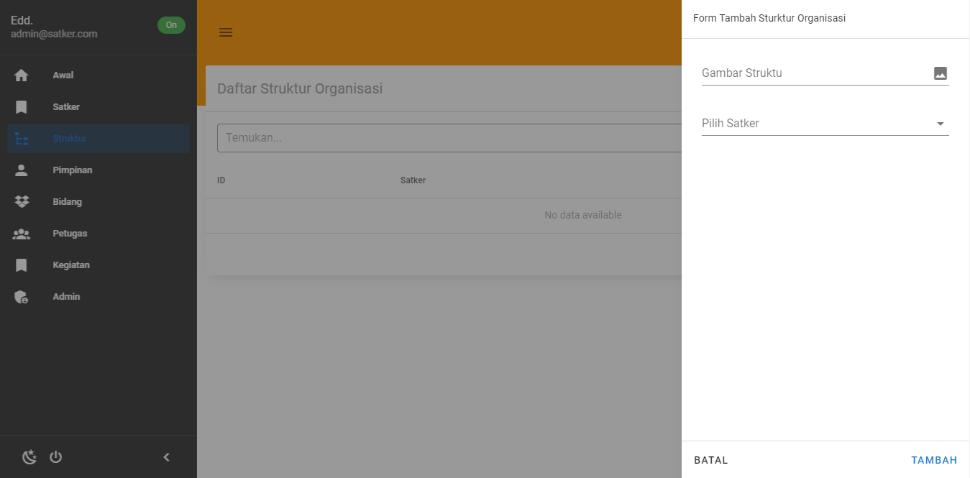
* 1. **Halaman Struktur Organisasi Admin**

Halaman Struktur Organisasi Admin adalah halaman yang berisikan informasi daftar struktur organisasi dari satker dan juga gambar struktur organisasinya, pada sudut kanan atas terdapat aksi untuk menambah data struktur organisasi (ikon tambah) dan terdapat aksi untuk setiap data struktur organisasi yang terletak diakhir kolom daftar data struktur organisasi, berikut tampilan dari halaman struktur organisasi.



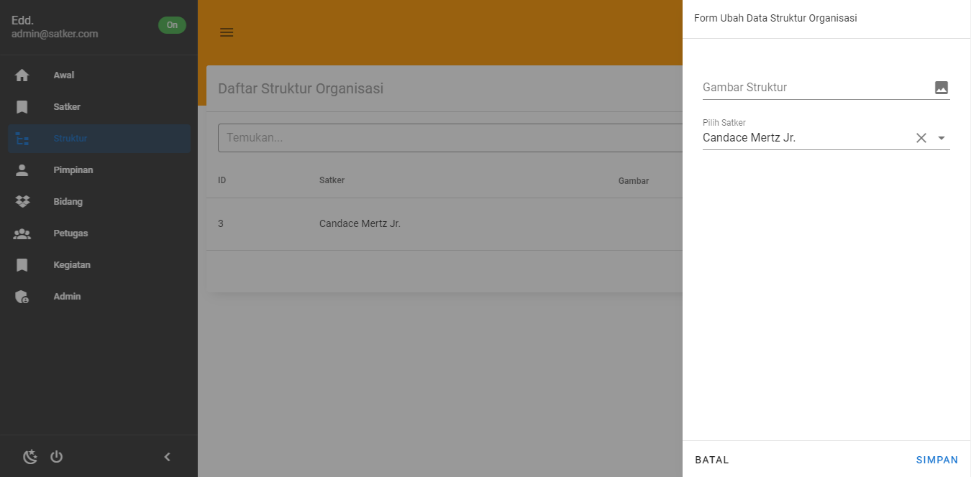
Gambar 4.14 Tampilan Halaman Struktur Organisasi

Juga terdapat *modal* dan *form* untuk menambah data struktur organisasi baru dengan mengklik ikon tambah untuk menampilkan *form* tersebut, dalam *form* tersebut terdapat kolom inputan berupa gambar struktur dan pilihan untuk menentukan pemilik dari gambar struktur, terdapat dua buah aksi batal dan *submit* tambah pada bagian bawah *form*. Berikut tampilan *form* tambah data struktur organisasi:



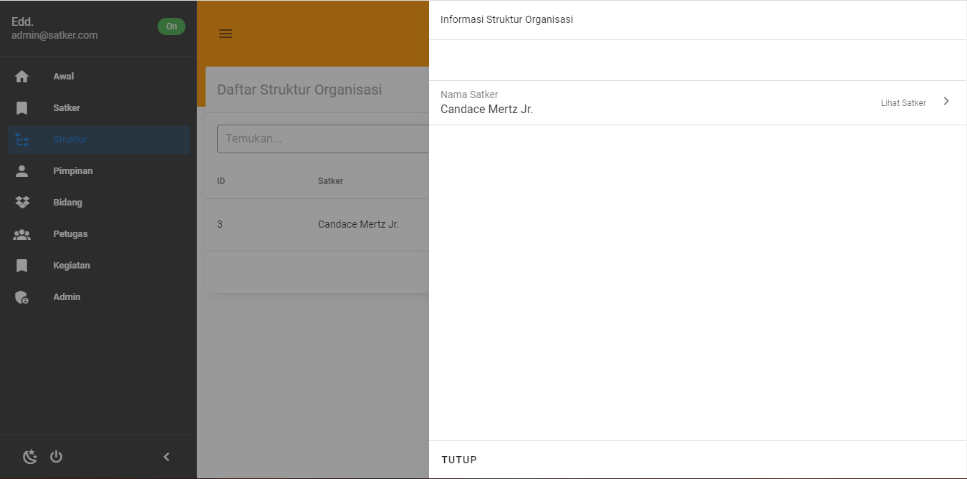
Gambar 4.15 Tampilan Form Tambah Struktur Organisasi

Terdapat juga *form* untuk mengubah data struktur organisasi, dengan mengklik ikon pensil untuk mengaktifkan form ini, *form* ini juga berisi inputan yang sama dengan *form* tambah struktur organisasi sebelumnya, juga terdapat dua buah aksi batal dan *submit* simpan untuk menyimpan perubahan data yang diisi pada inputan *form* berikut adalah tampilan dari *form* ubah data struktur organisasi :



Gambar 4.16 Tampilan *Form* Ubah Struktur Organisasi

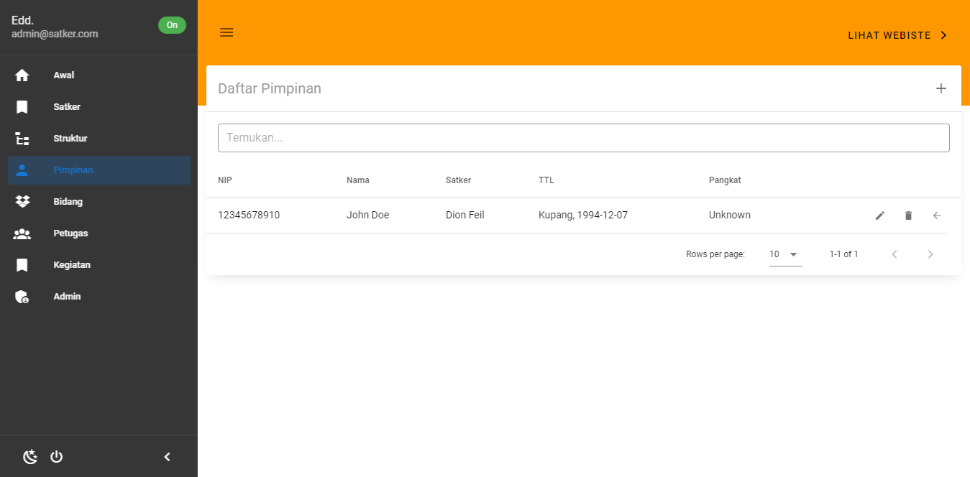
Terdapat juga *modal* untuk menapilkan rincian mengenai struktur organisasi dengan mengklik tombol rincian (ikon panah kiri) yang terdapat pada setiap data struktur organisasi. Pada *modal* ini terdapat rincian mengenai data struktur organisasi, ketika diklik nama satker pemilik struktur organisasi ini maka akan menampilkan rincian satker yang bersangkutan, berikut tampilan *modal* rincian:



Gambar 4.17 Tampilan Rincian Struktur Organisasi

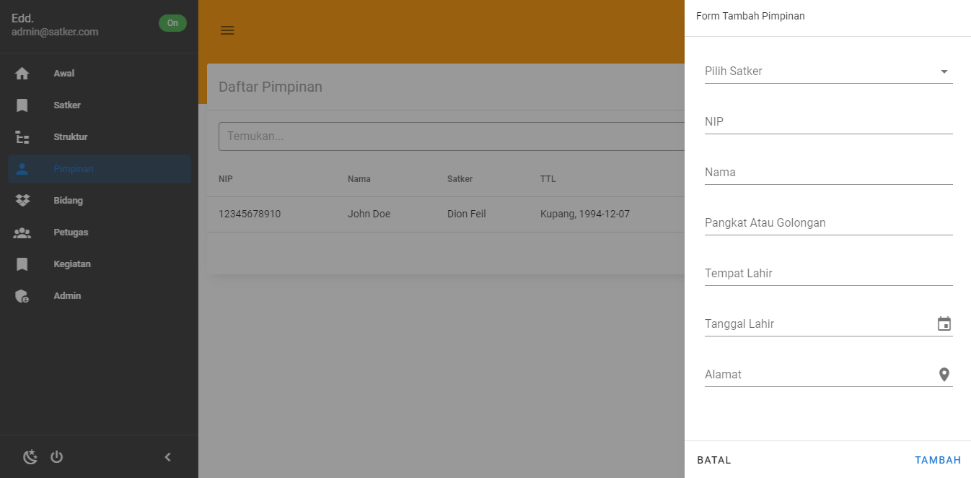
* 1. **Halaman Pimpinan Admin**

Halaman Pimpinan adalah halaman yang berisikan informasi daftar Pimpinan dari satker, sama seperti halaman sebelumnya pada sudut kanan atas terdapat aksi untuk menambah data pimpinan dan terdapat aksi untuk setiap data pimpinan yang terletak diakhir kolom data pimpinan, berikut tampilan dari halaman pimpinan.



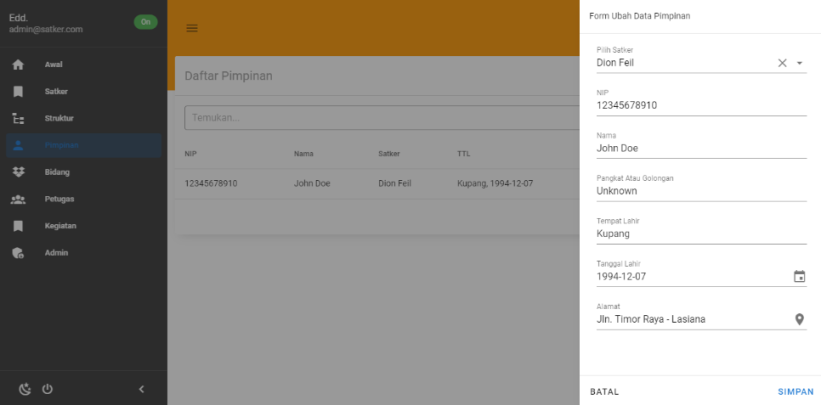
Gambar 4.18 Tampilan Halaman Pimpinan

Juga terdapat *modal* dan *form* untuk menambah data pimpinan baru dengan mengklik ikon tambah untuk menampilkan *form* tersebut, pada *form* ini terdapat beberapa inputan diantanya pilihan satker, NIP, nama pimpinan, pangkat atau golongan, tempat lahir, tanggal lahir dan alamat, dan terdapat dua buah aksi batal dan *submit* tambah pada bagian bawah *form*, berikut tampilan *form* tambah data pimpinan:



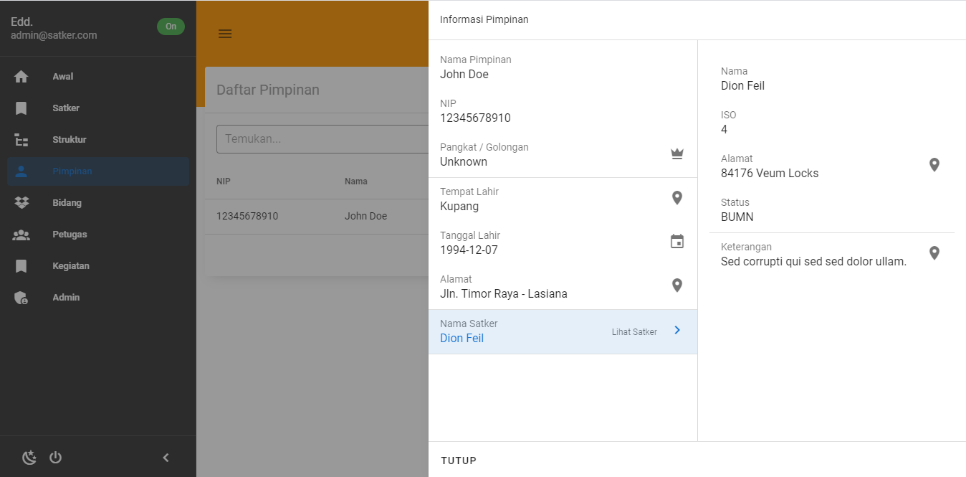
Gambar 4.19 Tampilan *Form* Tambah Pimpinan

Terdapat juga *form* untuk mengubah data pimpinan, dengan mengklik ikon pensil untuk mengaktifkan *form* ini, juga terdapat kolom inputan yang sama dengan form tambah pimpinan, terdapat satker pimpinan, NIP, nama pimpinan, pangkat atau golongan, tempat lahir, tanggal lahir, dan alamat pimpinan, dibawah form terdapat dua buah aksi batal dan submit simpan, untuk menyimpan perubahan data yang diinputkan pada *form*. Berikut adalah tampilan dari *form* ubah data pimpinan :



Gambar 4.20 Tampilan Form Ubah Pimpinan

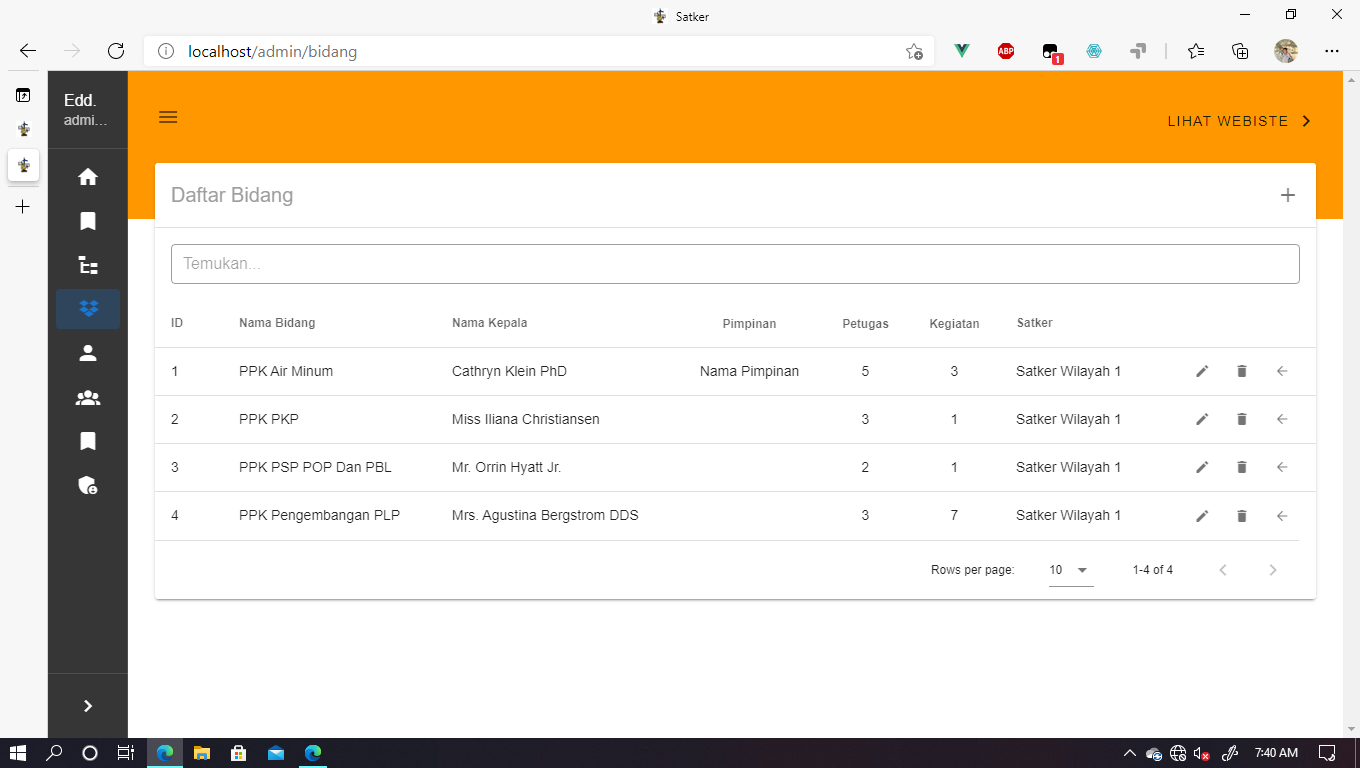
Terdapat juga modal untuk menapilkan rincian mengenai pimpinan dengan mengklik tombol rincian (ikon panah kiri) yang terdapat pada setiap data pimpinan. Pada modal ini terdapat rincian mengenai data pimpinan dan data satker yang berhubungan dengan data pimpinan yang dipilih, berikut tampilan *modal* rincian:



Gambar 4.21 Tampilan Rincian Pimpinan

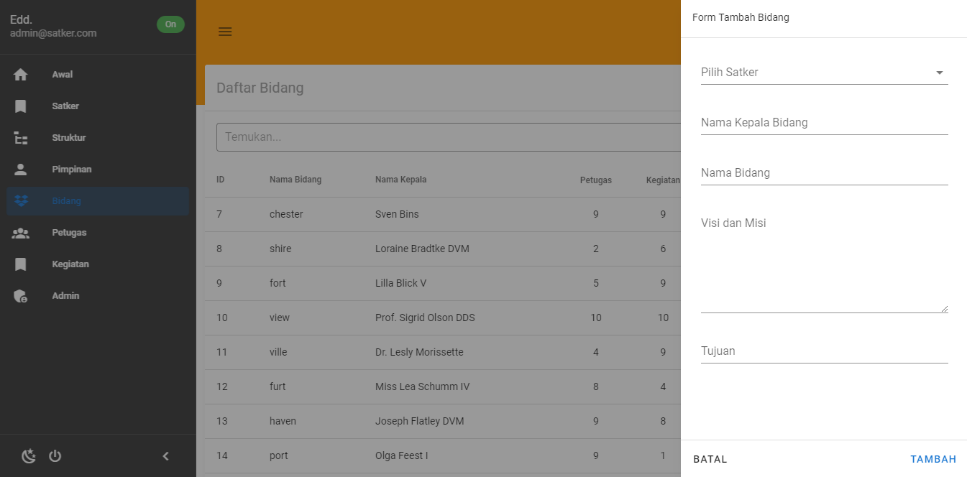
* 1. **Halaman Bidang Admin**

Halaman Bidang adalah halaman yang berisikan informasi daftar Bidang, pada sudut kanan atas terdapat aksi untuk menambah data bidang dan terdapat aksi untuk setiap data bidang yang terletak diakhir kolom daftar data bidang, sama seperti halaman daftar pada admin sebelumnya terdapat aksi dengan ikon pensil untuk mengubah data, tempat sampah untuk menghapus data bidang, dan panah kiri untuk melihat rincian data bidang, berikut tampilan dari halaman bidang.



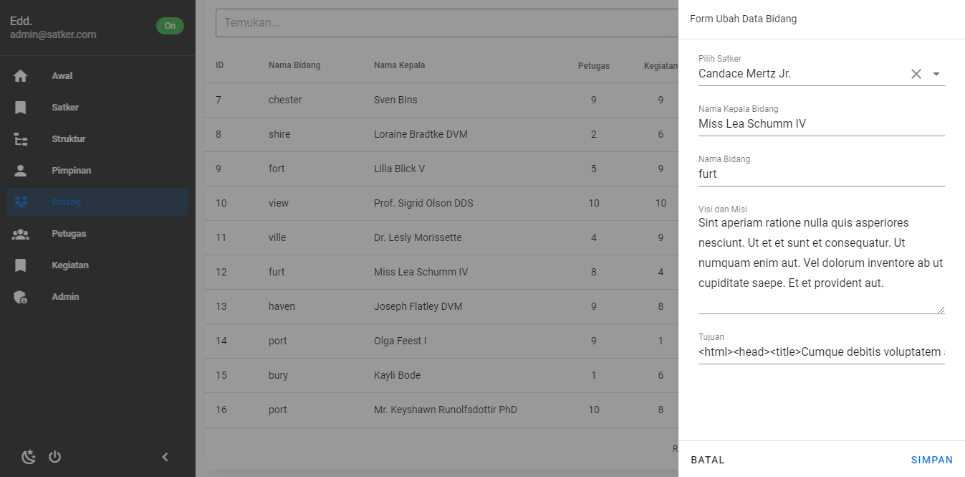
Gambar 4.22 Tampilan Halaman Bidang

Juga terdapat *modal* dan *form* untuk menambah data bidang baru dengan mengklik ikon tambah untuk menampilkan *form* tersebut, di dalam form ini terdapat pilihan satker, nama kepala bidang, nama bidang baru yang akan ditambahkan, visi dan misi bidang, dan terakhir tujuan bidang, berikut tampilan *form* tambah data bidang:



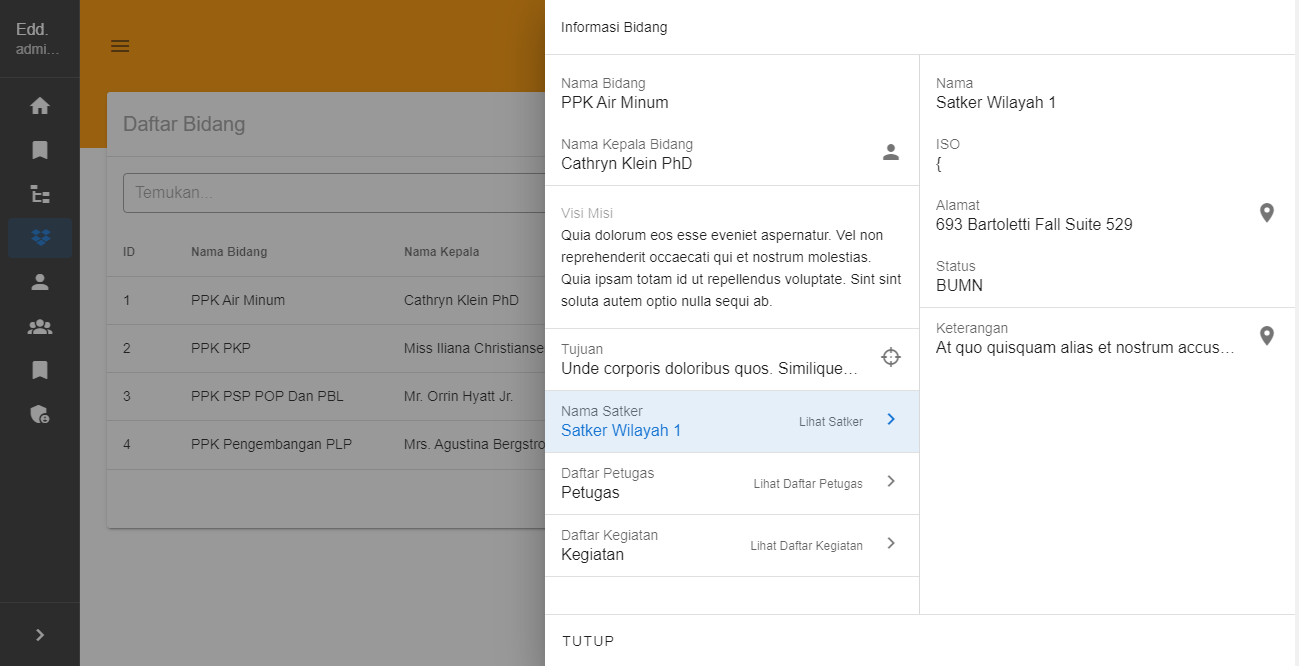
Gambar 4.23 Tampilan *Form* Tambah Bidang

Terdapat juga *form* untuk mengubah data bidang, dengan mengklik ikon pensil untuk mengaktifkan *form* ini, kolom inputan pada *form* ini sama seperti pada *form* tambah bidang sebelumnya dimana terdapat kolom inputan pilihan satker bidang, nama kepala bidang, nama bidang, visi dan misi bidang, dan tujuan bidang, setelah diubah datanya, maka data dapat langsung disimpan dengan mengklik tombol aksi simpan untuk menyimpan perubahan data pada bagian bawah *form*, berikut adalah tampilan dari *form* ubah data bidang:



Gambar 4.24 Tampilan Form Ubah Bidang

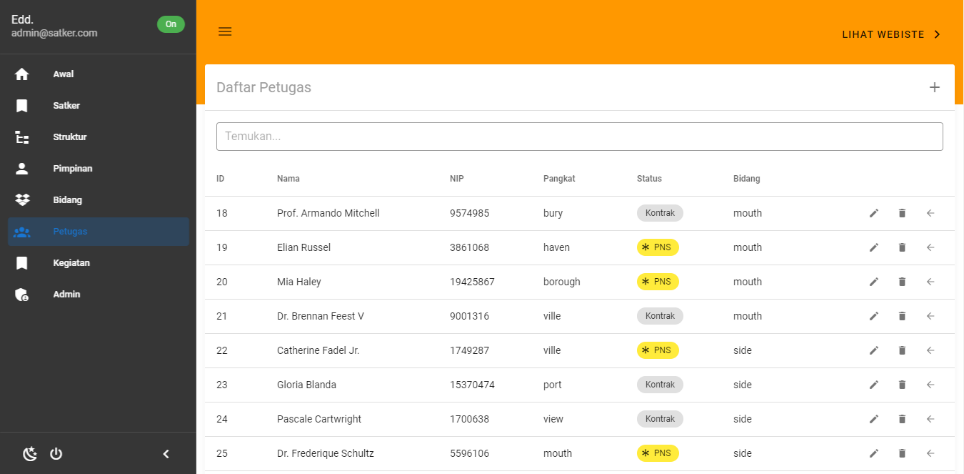
Terdapat juga *modal* untuk menapilkan rincian mengenai bidang dengan mengklik tombol rincian (ikon panah kiri) yang terdapat pada setiap data bidang. Pada *modal* ini terdapat rincian mengenai data bidang, satker, petugas, dan juga data kegiatan yang berelasi dengan data bidang yang dipilih, rincian informasi satker, petugas, dan kegiatan dapat diakses dengan mengklik setiap baris info seperti pada gambar 4.25, ketika baris info satker diklik, berikut tampilan *modal* rincian:



Gambar 4.25 Tampilan Rincian Bidang

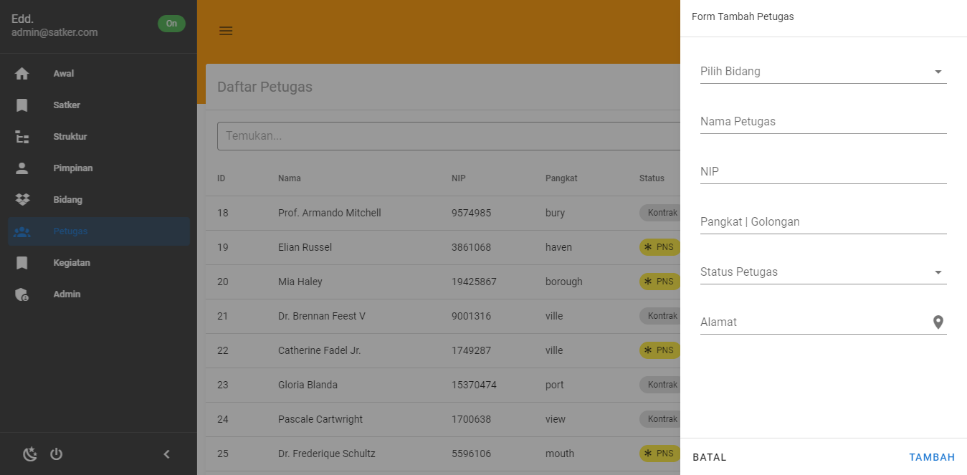
* 1. **Halaman Petugas Admin**

Halaman Petugas adalah halaman yang berisi aksi-aksi untuk mengelola data petugas dan memuat data-data petugas yang tercatat pada *database* aplikasi, pada sudut kanan atas terdapat aksi dengan ikon tambah untuk menambah data petugas dan terdapat aksi untuk setiap data petugas yang terletak diakhir kolom daftar data petugas, dimana terdapat ikon pensil untuk mengubah data petugas, aksi ikon tempat sampah untuk menghapus data petugas, dan panah kiri untuk melihat rincian data petugas tertentu, berikut tampilan dari halaman petugas.



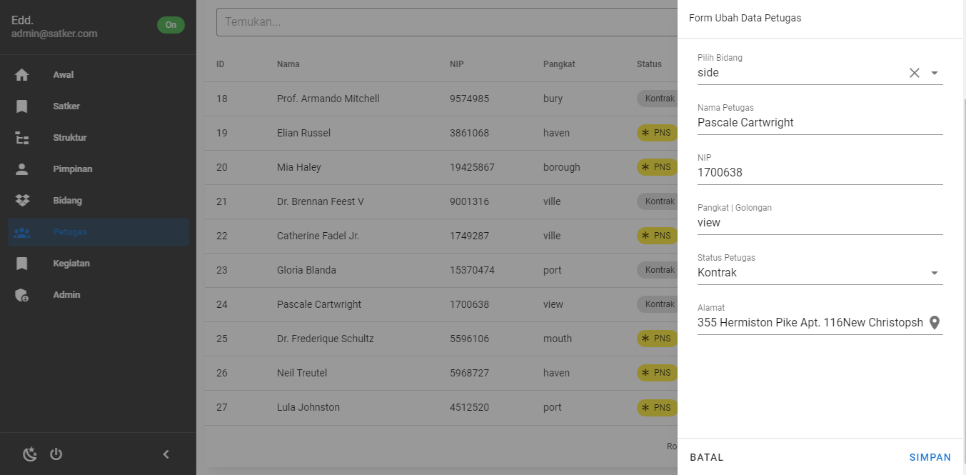
Gambar 4.26 Tampilan Halaman Petugas

Juga terdapat *modal* dan *form* untuk menambah data petugas baru dengan mengklik ikon tambah untuk menampilkan *form* tersebut, didalamnya terdapat kolom inputan berupa pilihan bidang petugas, nama petugas, NIP, pangkat atau golongan, status petugas berupa pilihan PNS dan Kontrak, dan alamat petugas, berikut tampilan *form* tambah data petugas:



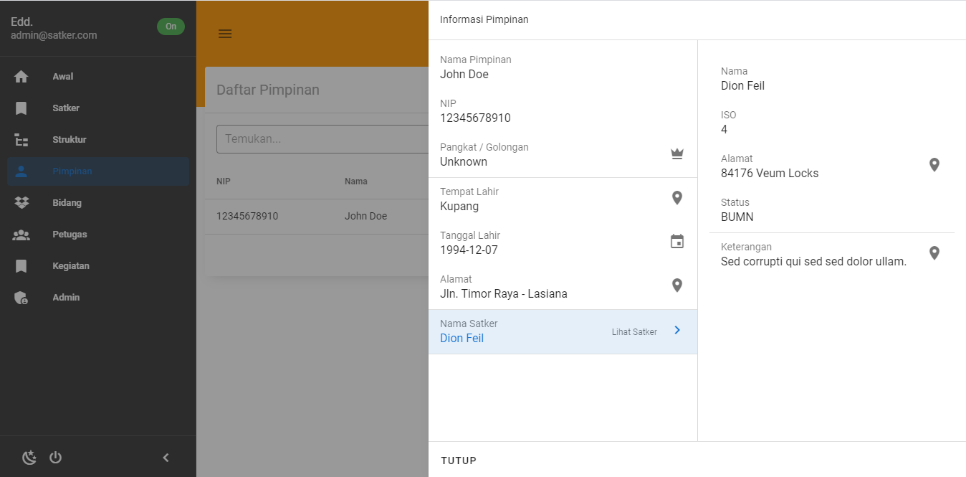
Gambar 4.27 Tampilan Form Tambah Petugas

Terdapat juga *form* untuk mengubah data petugas, dengan mengklik ikon pensil untuk mengaktifkan *form* ini, sama seperti form tambah data petugas sebelumnya, *form* ini juga memiliki kolom inputan berupa pilihan bidang petugas, nama petugas, NIP petugas, pangkat atau golongan petugas, status petugas baik itu PNS atau Kontrak, alamat petugas, dan juga terdapat aksi pada bagian bawah *form* aksi batal untuk membatalkan perubahan dan aksi simpan untuk menyimpan perubahan data petugas, berikut adalah tampilan dari *form* ubah data petugas:



Gambar 4.28 Tampilan *Form* Ubah Petugas

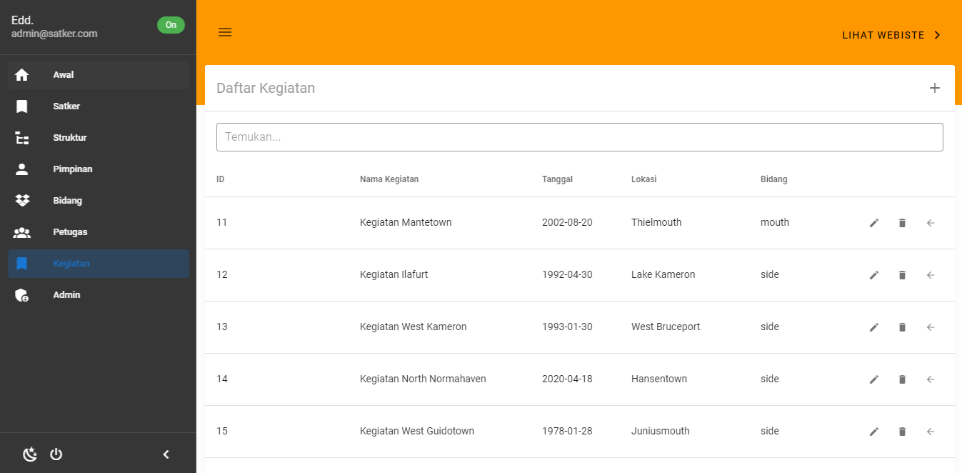
Terdapat juga *modal* untuk menampilkan rincian mengenai petugas dengan mengklik tombol rincian (ikon panah kiri) yang terdapat pada setiap data petugas. Pada modal ini terdapat rincian mengenai data petugas, data satker petugas, dan data bidang yang berhubungan dengan data petugas yang dipilih, berikut tampilan *modal* rincian:



Gambar 4.29 Tampilan Rincian Petugas

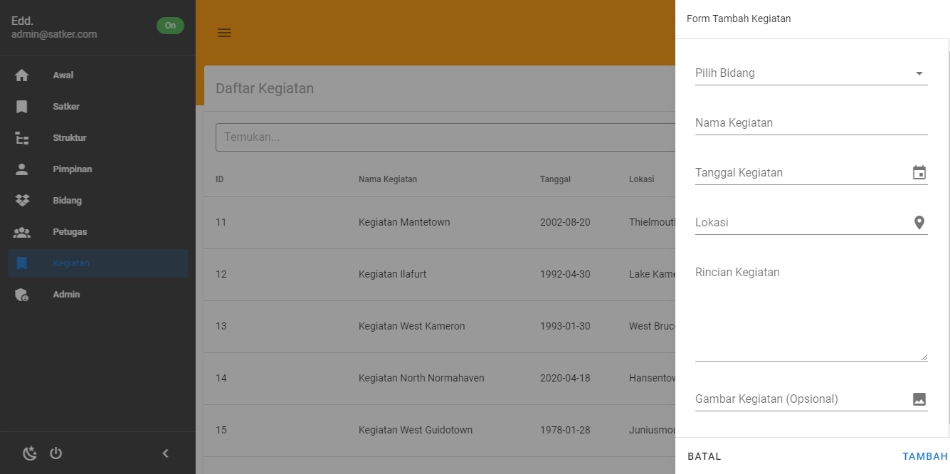
* 1. **Halaman Daftar Kegiatan Admin**

Halaman Daftar Kegiatan adalah halaman yang berisi aksi-aksi untuk mengelola data kegiatan dan memuat data-data kegiatan yang tercatat pada *database* aplikasi, pada sudut kanan atas terdapat aksi dengan ikon tambah atau *plus* untuk menambah data kegiatan dan terdapat aksi untuk setiap data kegiatan yang terletak diakhir kolom daftar data kegiatan, diantaranya terdapat aksi dengan ikon pensil untuk mengubah data kegiatan, aksi dengan ikon tempat sampah untuk menghapus data kegiatan dan panah kiri untuk melihat rincian data kegiatan tertentu, berikut tampilan dari halaman kegiatan pada *dashboard* admin.



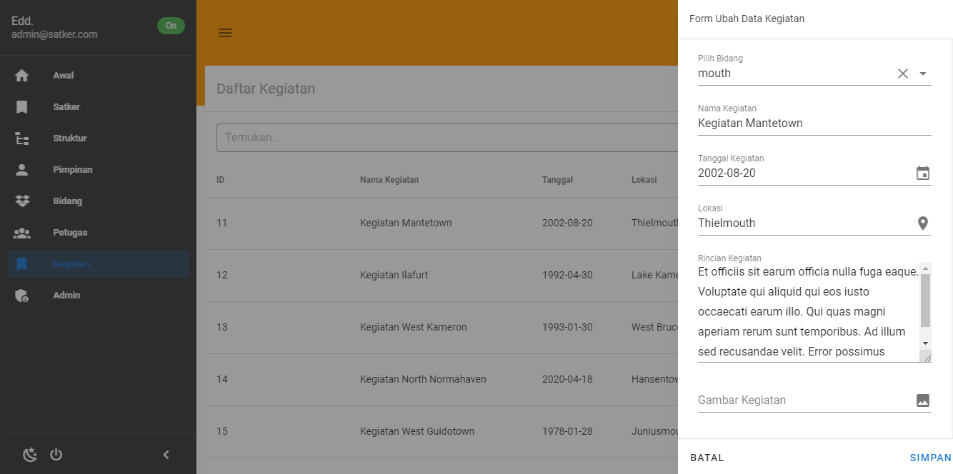
Gambar 4.30 Tampilan Halaman Daftar Kegiatan

Juga terdapat *modal* dan *form* untuk menambah data kegiatan baru dengan mengklik ikon tambah atau *plus* untuk menampilkan *form* tersebut, *form* ini berisikan kolom inputan berupa nama bidang kegiatan, nama kegiatan yang akan ditambahkan, tanggal kegiatan, lokasi kegiatan, rincian kegiatan, gambar kegiatan bisa berupa cover atau poster kegiatan, berikut tampilan *form* tambah data kegiatan:



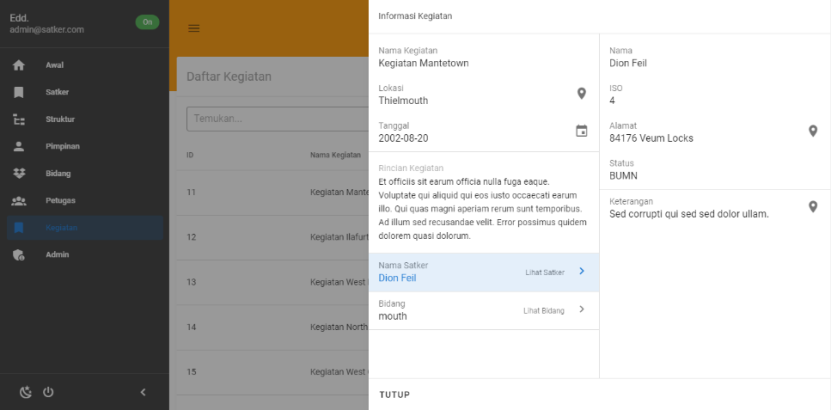
Gambar 4.31 Tampilan *Form* Tambah Kegiatan

Terdapat juga *form* untuk mengubah data kegiatan, dengan mengklik ikon pensil untuk mengaktifkan *form* ini, sama seperti *form* tambah kegiatan sebelumnya, *form* ini juga memiliki kolom inputan pilihan bidang kegiatan, nama kegiatan, tanggal kegiatan, lokasi kegiatan, rincian mengenai kegiatan, gambar kegiatan, dan 2 (dua) buah tombol aksi di bagian bawah *form*, diantaranya terdapat aksi batal dan aksi simpan, berikut adalah tampilan dari *form* ubah data kegiatan:



Gambar 4.32 Tampilan *Form* Ubah Kegiatan

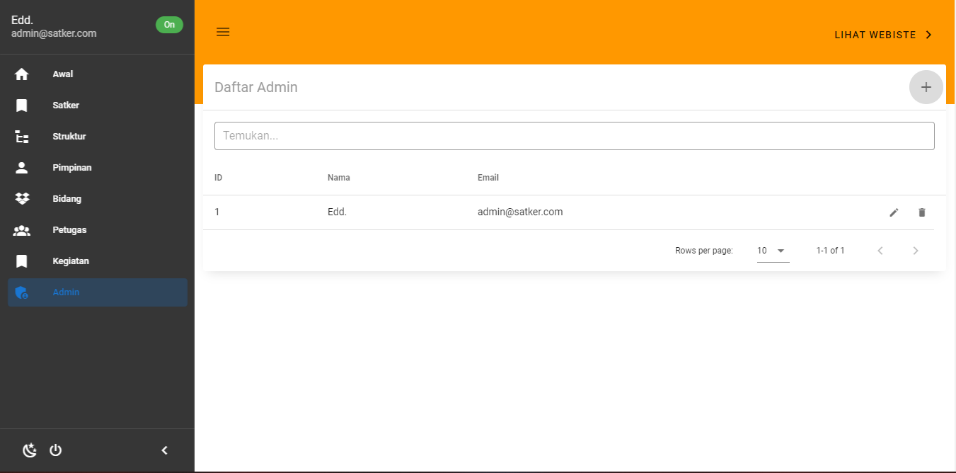
Terdapat juga *modal* untuk menapilkan rincian mengenai kegiatan dengan mengklik tombol rincian (ikon panah kiri) yang terdapat pada setiap data kegiatan. Pada *modal* ini terdapat rincian mengenai data kegiatan, data satker, dan data bidang yang berhubungan dengan data kegiatan yang dipilih, berikut tampilan modal rincian:



Gambar 4.33 Tampilan Rincian Kegiatan

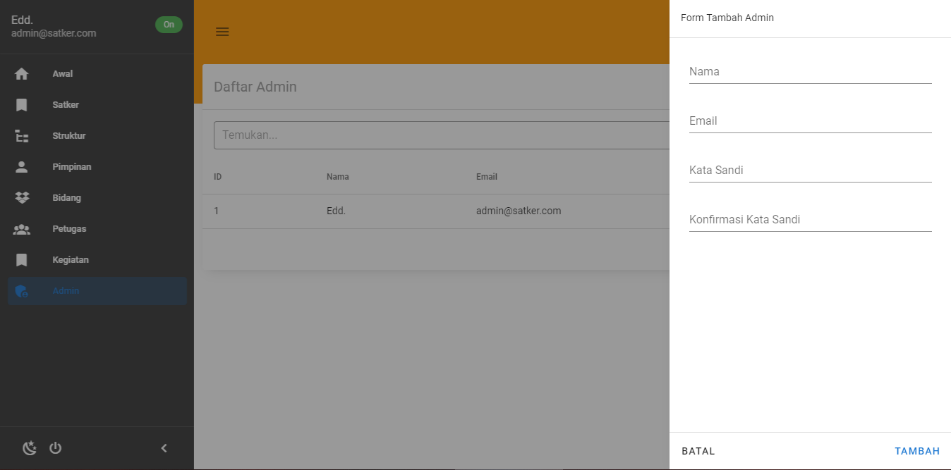
* 1. **Halaman Daftar Admin Admin**

Halaman Daftar Admin adalah halaman yang berisi aksi-aksi untuk mengelola data admin dan memuat data-data admin yang tercatat pada *database* aplikasi, pada sudut kanan atas terdapat aksi untuk menambah data admin dengan tampilan ikon tambah atau *plus* dan terdapat aksi untuk setiap data admin yang terletak diakhir kolom daftar data admin, berupa aksi dengan ikon pensil untuk mengubah data admin dan aksi dengan ikon tempat sampah untuk menghapus data admin, berikut tampilan dari halaman admin.



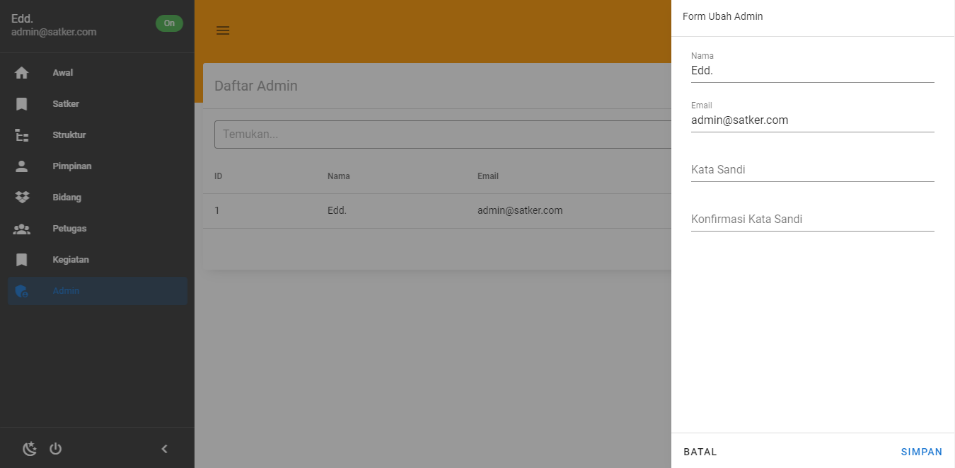
Gambar 4.34 Tampilan Halaman Daftar Admin

Juga terdapat modal dan *form* untuk menambah data admin baru dengan mengklik ikon tambah untuk menampilkan form tersebut, di dalam form tambah data admin ini terdapat beberapa kolom inputan diantaranya, kolom nama admin, email admin yang akan digunakan untuk proses autentikasi, kolom kata sandi yang juga digunakan untuk proses autentikasi login bersamaan dengan email, dan terakhir kolom konfirmasi kata sandi, yaitu kolom untuk mengulang kata sandi yang sebelumnya diinput agar admin yakin kata sandi yang diinputkan adalah kata yang dapat diingat oleh admin, berikut tampilan form tambah data admin:



Gambar 4.35 Tampilan Form Tambah Admin

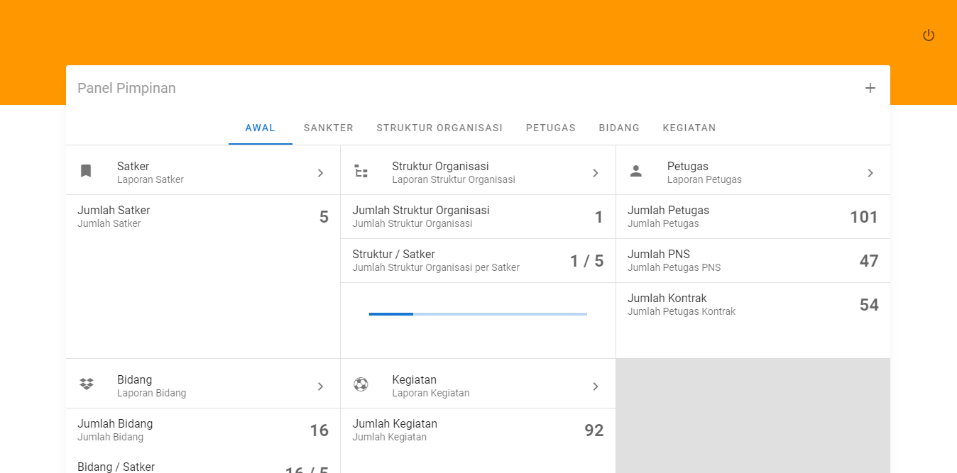
Terdapat juga *form* untuk mengubah data admin, dengan mengklik ikon pensil untuk mengaktifkan *form* ini, sama seperti form tambah admin sebelumnya, terdapat kolom inputan, nama admin, email admin, kata sandi, konfirmasi kata sandi, dan 2 (dua) buah aksi batal dan simpan untuk menyimpan perubahan data, berikut adalah tampilan dari *form* ubah data admin:



Gambar 4.36 Tampilan Form Ubah Admin

* 1. **Halaman Awal Pimpinan**

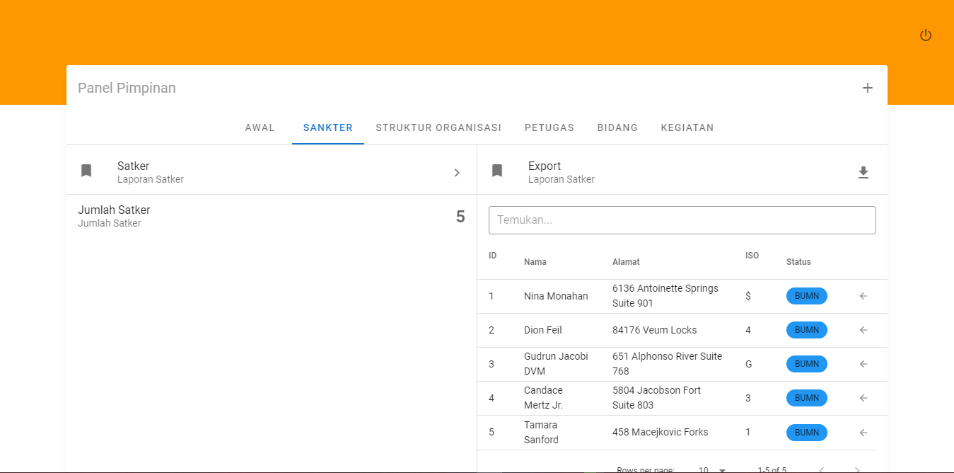
Halaman awal Pimpinan adalah tampilan paling pertama yang tampil saat pimpinan berhasil melakukan login pada aplikasi dan akan diarahkan ke halaman pimpinan, halaman ini berisi jumlah satker, bidang struktur organisasi, petugas, dan kegiatan serta beberapa menu navigasi untuk mengakses informasi lainnya, diantaranya terdapat tab awal, yaitu halaman awal pimpinan, tab satker untuk menampilkan informasi satker pimpinan, tab struktur organisasi, petugas untuk menampilkan informasi petugas, bidang dan kegiatan yang berhubungan dengan pimpinan, terdapat sebuah tombol aksi di sudut kanan atas berupa ikon *power* untuk *logout* dari aplikasi berikut adalah tampilannya.



Gambar 4.37 Tampilan Halaman Awal Pimpinan

* 1. **Halaman Satker Pimpinan**

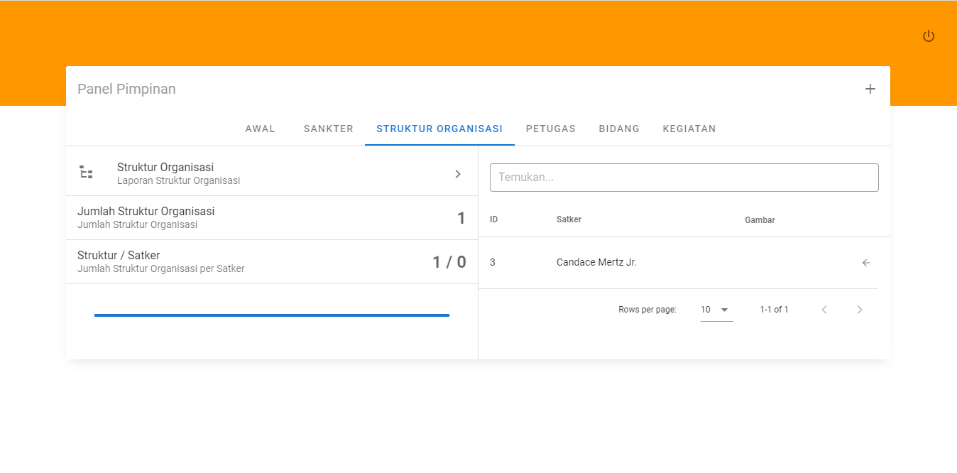
Halaman Satker pada pimpinan adalah halaman yang berisi informasi mengenai informasi satker, pada halaman ini juga terdapat filter untuk menfilter data satker, aksi untuk menampilkan rincian data satker dan mengekspor laporan data satker, berikut adalah tampilan tab satker pada pimpinan.



Gambar 4.38 Tampilan Halaman Satker Pimpinan

* 1. **Halaman Struktur Organisasi Pimpinan**

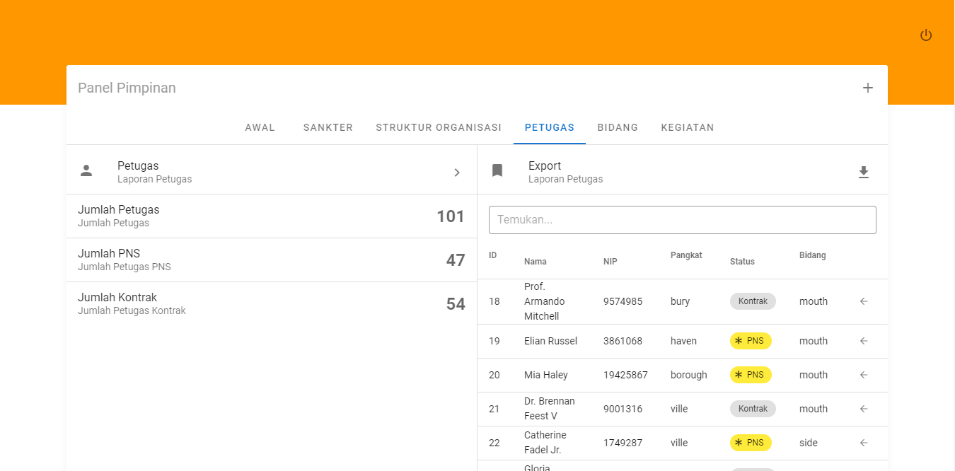
Halaman Struktur organisasi pada pimpinan adalah halaman yang berisi informasi mengenai data struktur organisasi, pada halaman ini juga terdapat aksi untuk menampilkan rincian dan mengekspor laporan data struktur organisasi.



Gambar 4.39 Tampilan Halaman Struktur Organisasi Pimpinan

* 1. **Halaman Petugas Pimpinan**

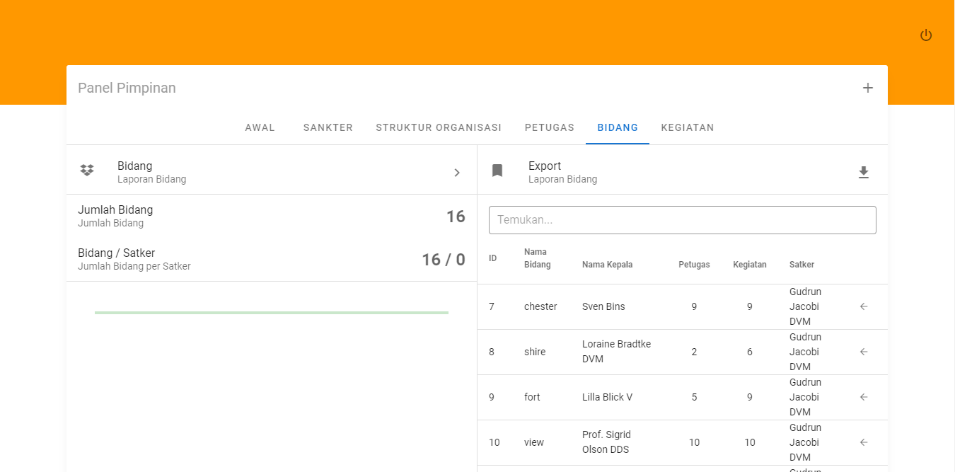
Halaman Petugas pada pimpinan adalah halaman yang berisi informasi mengenai data petugas yang berelasi dengan pimpinan satker, pada halaman ini juga terdapat aksi untuk menampilkan rincian dan mengekspor laporan data petugas.



Gambar 4.40 Tampilan Halaman Petugas Pimpinan

* 1. **Halaman Bidang Pimpinan**

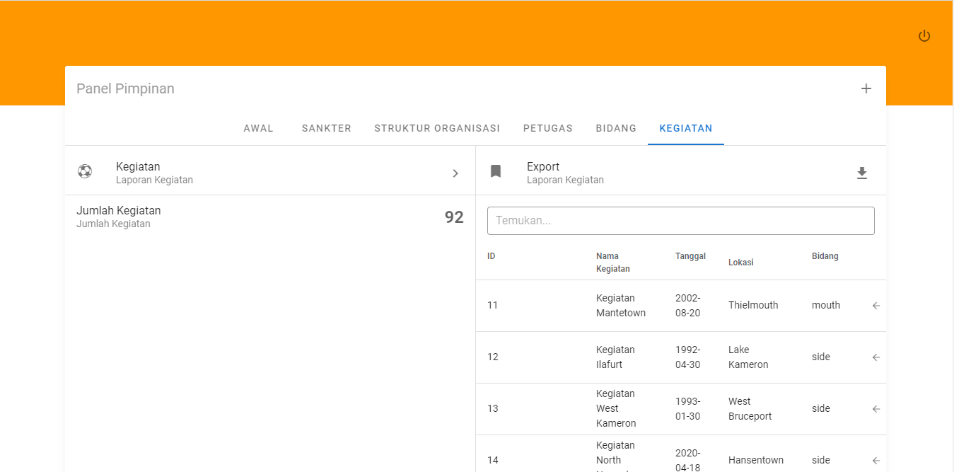
Halaman Bidang pada pada pimpinan adalah halaman yang berisi informasi mengenai data bidang, pada halaman ini juga terdapat aksi untuk menampilkan rincian dan mengekspor laporan data bidang, dan terdapat juga kolom pencarian untuk mempermudah pimpinan dalam menyaring data.



Gambar 4.40 Tampilan Halaman Bidang Pimpinan

* 1. **Halaman Kegiatan Pimpinan**

Halaman Kegiatan pada pimpinan adalah halaman yang berisi informasi mengenai data kegiatan, pada halaman ini juga terdapat aksi untuk menampilkan rincian dan mengekspor laporan data kegiatan.



Gambar 4.41 Tampilan Halaman Kegiatan Pimpinan

1. **Implementasi Program**
   * 1. **Halaman Satker Admin**

Pada halaman satker terdapat beberapa proses utama yang terjadi diantaranya proses menampilkan data satker dan proses menambah dan mengubah data satker. Untuk menambahkan data satker dengan memanggil model dan diakhiri dengan memanggil method paginate(x), x dalam angka jumlah data yang akan ditampilkan per halaman, dan untuk menambahkan sakter dengan memanggil method create(x) dengan x sebagai array informasi data satker yang diinputkan, sedangkan untuk mengedit satker dengan memanggil method update(x) dari sebuah instance satker yang akan diedit, dengan x sebagai data array yang akan di update.

Tabel 4.1 Baris Perintah Menampilkan Data Satker

|  |  |
| --- | --- |
|  | Satker::when(request()->filled('search'), function($query){  $search = request('search');  return $query->where('nama', 'like', "%{$search}%");  })  ->when(request()->filled('id\_satker'), function($query){  return $query->orWhere('id\_satker', (int) request('id\_satker'));  })  ->when(request()->filled('sortBy'), function($query){  $sortBy = is\_array(request('sortBy')) ? request('sortBy')[0] : request('sortBy');  $sortDesc = is\_array(request('sortDesc')) ? request('sortDesc')[0] : request('sortDesc');  return $query->orderBy($sortBy, $sortDesc == 'true' ? 'DESC' : 'ASC');  })  ->paginate(request('itemsPerPage') ?? 10) |

Untuk menambah sebuah data satker baru field / $\_POST yang dikirim dari form di bind ke object $request method validated() untuk mengambil data request yang dikirim dan sudah divalidasi untuk melihat validasinya misalkan untuk validasi satker di directory app/Http/Request/RequestStoreSatker.php didalam file RequestStoreSatker.php terdapat rule untuk setiap field request yang dikirim dari form. lalu untuk menambah dengan memanggil method created() secara static Res::create(), Res::update() hanya untuk parsing hasil return boolean ke array dengan pesan "sukses/gagal" agar lebih mudah dibaca manusia

Tabel 4.2 Baris Perintah Menambahkan Data Satker

|  |  |
| --- | --- |
|  | public function store(RequestStoreSatker $request){  $data = $request->validated();  return Res::store(  Satker::create($data)  );  } |

Untuk mengambil satu data satker berdasarkan id yang dikirim / dilaravel id langsung di cast ke object $satker

Tabel 4.3 Baris Perintah Mengambil Data Spesifik Satker

|  |  |
| --- | --- |
|  | public function show(Satker $satker){  return new SatkerCollection($satker);  } |

Sama seperti store pertama divalidasi dan kemudian data satker yang diparsing/hasil cast dari id (mirip show), panggil method update() dengan mengirim hasil validasi data tadi

Tabel 4.4 Baris Perintah Mengubah Data Satker

|  |  |
| --- | --- |
|  | public function update(RequestUpdateSatker $request, Satker $satker){  $data = $request->validated();  return Res::update(  $satker->update($data)  );  } |

Untuk menghapus salah satu data satker, dapat dilakukan dengan memanggil method destory()

Tabel 4.5 Baris Perintah Menghapus Data Satker

|  |  |
| --- | --- |
|  | public function destroy(Satker $satker){  return Res::delete($satker->delete());  } |

* + 1. **Halaman Struktur Organisasi Admin**

Pada halaman struktur organisasi terdapat beberapa proses utama yang terjadi diantaranya proses menampilkan data struktur organisasi dan proses menambah dan mengubah data struktur organisasi. Untuk menambahkan data struktur organisasi dengan memanggil model dan diakhiri dengan memanggil method paginate(x), x dalam angka jumlah data yang akan ditampilkan per halaman, dan untuk menambahkan sakter dengan memanggil method create(x) dengan x sebagai array informasi data struktur organisasi yang diinputkan, sedangkan untuk mengedit struktur organisasi dengan memanggil method update(x) dari sebuah instance struktur organisasi yang akan diedit, dengan x sebagai data array yang akan di update.

Tabel 4.6 Baris Perintah Menampilkan Data Struktur Organisasi

|  |  |
| --- | --- |
|  | public function index(){  return StrukturCollection::collection(  StrukturOrganisasi::when(request()->filled('search'), function($query){  $search = request('search');  // return $query->where('id\_', 'like', "%{$search}%");  })  ->when(request()->filled('id\_satker'), function($query){  return $query->where('id\_satker', request('id\_satker'));  })  ->when(request()->filled('sortBy'), function($query){  $sortBy = is\_array(request('sortBy')) ? request('sortBy')[0] : request('sortBy');  $sortDesc = is\_array(request('sortDesc')) ? request('sortDesc')[0] : request('sortDesc');  return $query->orderBy($sortBy, $sortDesc == 'true' ? 'DESC' : 'ASC');  })  ->paginate(request('itemsPerPage') ?? 10)  );  } |

Untuk menambah sebuah data struktur organisasi baru field / $\_POST yang dikirim dari form di bind ke object $request method validated() untuk mengambil data request yang dikirim dan sudah divalidasi untuk melihat validasinya misalkan untuk validasi struktur organisasi di directory app/Http/Request/RequestStoreStruktur.php didalam file RequestStoreStruktur.php terdapat rule untuk setiap field request yang dikirim dari form. lalu untuk menambah dengan memanggil method created() secara static Res::create(), Res::update() hanya untuk parsing hasil return boolean ke array dengan pesan "sukses/gagal" agar lebih mudah dibaca manusia

Tabel 4.7 Baris Perintah Menambahkan Data Struktur Organisasi

|  |  |
| --- | --- |
|  | public function store(RequestStoreStruktur $request){  $data = $request->validated();  if($request->gambar){  $nama\_gambar = $request->gambar->store('/struktur', 'public');  $data['gambar'] = $nama\_gambar;  }  return Res::store(  StrukturOrganisasi::create($data)  );  } |

Untuk mengambil satu data struktur organisasi berdasarkan id yang dikirim / dilaravel id langsung di cast ke object $struktur

Tabel 4.8 Baris Perintah Mengambil Data Spesifik Struktur Organisasi

|  |  |
| --- | --- |
|  | public function show(StrukturOrganisasi $struktur){  return new StrukturCollection($struktur);  } |

Sama seperti store pertama divalidasi dan kemudian data struktur organisasi yang diparsing/hasil cast dari id (mirip show), panggil method update() dengan mengirim hasil validasi data tadi

Tabel 4.9 Baris Perintah Mengubah Data Struktur Organisasi

|  |  |
| --- | --- |
|  | public function update(RequestUpdateStruktur $request, StrukturOrganisasi $struktur){  $data = $request->validated();  if($request->gambar){  $nama\_gambar\_old = $struktur->gambar;  $nama\_gambar = $request->gambar->store('/struktur', 'public');  $data['gambar'] = $nama\_gambar;  if($nama\_gambar && Storage::disk('public')->exists($nama\_gambar\_old)){  Storage::disk('public')->delete($nama\_gambar\_old);  }  }  return Res::update(  $struktur->update($data)  );  } |

Untuk menghapus salah satu data struktur organisasi, dapat dilakukan dengan memanggil method destory()

Tabel 4.10 Baris Perintah Menghapus Data Struktur Organisasi

|  |  |
| --- | --- |
|  | public function destroy(StrukturOrganisasi $struktur){  return Res::delete(  $struktur->delete()  );  } |

* + 1. **Halaman Pimpinan Admin**

Pada halaman pimpinan terdapat beberapa proses utama yang terjadi diantaranya proses menampilkan data pimpinan dan proses menambah dan mengubah data pimpinan. Untuk menambahkan data pimpinan dengan memanggil model dan diakhiri dengan memanggil method paginate(x), x dalam angka jumlah data yang akan ditampilkan per halaman, dan untuk menambahkan sakter dengan memanggil method create(x) dengan x sebagai array informasi data pimpinan yang diinputkan, sedangkan untuk mengedit pimpinan dengan memanggil method update(x) dari sebuah instance pimpinan yang akan diedit, dengan x sebagai data array yang akan di update.

Tabel 4.11 Baris Perintah Menampilkan Data Pimpinan

|  |  |
| --- | --- |
|  | public function index(){  return PimpinanCollection::collection(  Pimpinan::when(request()->filled('search'), function($query){  $search = request('search');  return $query->where('nama', 'like', "%{$search}%");  })  ->when(request()->filled('id\_satker'), function($query){  return $query->where('id\_satker', request('id\_satker'));  })  ->when(request()->filled('sortBy'), function($query){  $sortBy = is\_array(request('sortBy')) ? request('sortBy')[0] : request('sortBy');  $sortDesc = is\_array(request('sortDesc')) ? request('sortDesc')[0] : request('sortDesc');  return $query->orderBy($sortBy, $sortDesc == 'true' ? 'DESC' : 'ASC');  })  ->paginate(request('itemsPerPage') ?? 10)  );  } |

Untuk menambah sebuah data pimpinan baru field / $\_POST yang dikirim dari form di bind ke object $request method validated() untuk mengambil data request yang dikirim dan sudah divalidasi untuk melihat validasinya misalkan untuk validasi pimpinan di directory app/Http/Request/RequestStorePimpinan.php didalam file RequestStorePimpinan.php terdapat rule untuk setiap field request yang dikirim dari form. lalu untuk menambah dengan memanggil method created() secara static Res::create(), Res::update() hanya untuk parsing hasil return boolean ke array dengan pesan "sukses/gagal" agar lebih mudah dibaca manusia

Tabel 4.12 Baris Perintah Menambahkan Data Pimpinan

|  |  |
| --- | --- |
|  | public function store(RequestStorePimpinan $request){  $data = $request->validated();    $data['password'] = Hash::make($data['nip']);  return Res::store(  Pimpinan::create($data)  );  } |

Untuk mengambil satu data pimpinan berdasarkan id yang dikirim / dilaravel id langsung di cast ke object $pimpinan

Tabel 4.13 Baris Perintah Mengambil Data Spesifik Pimpinan

|  |  |
| --- | --- |
|  | public function show(Pimpinan $pimpinan){  return new PimpinanCollection($pimpinan);  } |

Sama seperti store pertama divalidasi dan kemudian data pimpinan yang diparsing/hasil cast dari id (mirip show), panggil method update() dengan mengirim hasil validasi data tadi

Tabel 4.14 Baris Perintah Mengubah Data Pimpinan

|  |  |
| --- | --- |
|  | public function update(RequestUpdatePimpinan $request, Pimpinan $pimpinan){  $data = $request->validated();  $data['password'] = Hash::make($data['nip']);    return Res::update(  $pimpinan->update($data)  );  } |

Untuk menghapus salah satu data pimpinan, dapat dilakukan dengan memanggil method destory()

Tabel 4.15 Baris Perintah Menghapus Data Pimpinan

|  |  |
| --- | --- |
|  | public function destroy(Pimpinan $pimpinan){  return Res::delete($pimpinan->delete());  } |

* + 1. **Halaman Petugas Admin**

Pada halaman petugas terdapat beberapa proses utama yang terjadi diantaranya proses menampilkan data petugas dan proses menambah dan mengubah data petugas. Untuk menambahkan data petugas dengan memanggil model dan diakhiri dengan memanggil method paginate(x), x dalam angka jumlah data yang akan ditampilkan per halaman, dan untuk menambahkan sakter dengan memanggil method create(x) dengan x sebagai array informasi data petugas yang diinputkan, sedangkan untuk mengedit petugas dengan memanggil method update(x) dari sebuah instance petugas yang akan diedit, dengan x sebagai data array yang akan di update.

Tabel 4.16 Baris Perintah Menampilkan Data Petugas

|  |  |
| --- | --- |
|  | public function index(){  return PetugasCollection::collection(  Petugas::when(request()->filled('search'), function($query){  $search = request('search');  return $query->where('nama', 'like', "%{$search}%");  })  ->when(request()->filled('id\_bidang'), function($query){  return $query->where('id\_bidang', request('id\_bidang'));  })  ->when(request()->filled('sortBy'), function($query){  $sortBy = is\_array(request('sortBy')) ? request('sortBy')[0] : request('sortBy');  $sortDesc = is\_array(request('sortDesc')) ? request('sortDesc')[0] : request('sortDesc');  return $query->orderBy($sortBy, $sortDesc == 'true' ? 'DESC' : 'ASC');  })  ->paginate(request('itemsPerPage') ?? 10)  );  } |

Untuk menambah sebuah data petugas baru field / $\_POST yang dikirim dari form di bind ke object $request method validated() untuk mengambil data request yang dikirim dan sudah divalidasi untuk melihat validasinya misalkan untuk validasi petugas di directory app/Http/Request/RequestStorePetugas.php didalam file RequestStorePetugas.php terdapat rule untuk setiap field request yang dikirim dari form. lalu untuk menambah dengan memanggil method created() secara static Res::create(), Res::update() hanya untuk parsing hasil return boolean ke array dengan pesan "sukses/gagal" agar lebih mudah dibaca manusia

Tabel 4.17 Baris Perintah Menambahkan Data Petugas

|  |  |
| --- | --- |
|  | public function store(RequestStorePetugas $request){  $data = $request->validated();  return Res::store(  Petugas::create($data)  );  } |

Untuk mengambil satu data petugas berdasarkan id yang dikirim / dilaravel id langsung di cast ke object $petugas

Tabel 4.18 Baris Perintah Mengambil Data Spesifik Petugas

|  |  |
| --- | --- |
|  | public function show(Petugas $petuga){  return new PetugasCollection($petuga);  } |

Sama seperti store pertama divalidasi dan kemudian data petugas yang diparsing/hasil cast dari id (mirip show), panggil method update() dengan mengirim hasil validasi data tadi

Tabel 4.19 Baris Perintah Mengubah Data Petugas

|  |  |
| --- | --- |
|  | public function update(RequestUpdatePetugas $request, Petugas $petuga){  $data = $request->validated();  return Res::update(  $petuga->update($data)  );  } |

Untuk menghapus salah satu data petugas, dapat dilakukan dengan memanggil method destory()

Tabel 4.20 Baris Perintah Menghapus Data Petugas

|  |  |
| --- | --- |
|  | public function destroy(Petugas $petuga){  return Res::delete($petuga->delete());  } |

* + 1. **Halaman Kegiatan Admin**

Pada halaman kegiatan terdapat beberapa proses utama yang terjadi diantaranya proses menampilkan data kegiatan dan proses menambah dan mengubah data kegiatan. Untuk menambahkan data kegiatan dengan memanggil model dan diakhiri dengan memanggil method paginate(x), x dalam angka jumlah data yang akan ditampilkan per halaman, dan untuk menambahkan sakter dengan memanggil method create(x) dengan x sebagai array informasi data kegiatan yang diinputkan, sedangkan untuk mengedit kegiatan dengan memanggil method update(x) dari sebuah instance kegiatan yang akan diedit, dengan x sebagai data array yang akan di update.

Tabel 4.21 Baris Perintah Menampilkan Data Kegiatan

|  |  |
| --- | --- |
|  | public function index(){  return KegiatanCollection::collection(  Kegiatan::when(request()->filled('search'), function($query){  $search = request('search');  return $query->where('nama\_kegiatan', 'like', "%{$search}%");  })  ->when(request()->filled('id\_bidang'), function($query){  return $query->where('id\_bidang', request('id\_bidang'));  })  ->when(request()->filled('id\_kegiatan'), function($query){  return $query->orWhere('id\_kegiatan', request('id\_kegiatan'));  })  ->when(request()->filled('sortBy'), function($query){  $sortBy = is\_array(request('sortBy')) ? request('sortBy')[0] : request('sortBy');  $sortDesc = is\_array(request('sortDesc')) ? request('sortDesc')[0] : request('sortDesc');  return $query->orderBy($sortBy, $sortDesc == 'true' ? 'DESC' : 'ASC');  })  ->paginate(request('itemsPerPage') ?? 10)  );  } |

Untuk menambah sebuah data kegiatan baru field / $\_POST yang dikirim dari form di bind ke object $request method validated() untuk mengambil data request yang dikirim dan sudah divalidasi untuk melihat validasinya misalkan untuk validasi kegiatan di directory app/Http/Request/RequestStoreKegiatan.php didalam file RequestStoreKegiatan.php terdapat rule untuk setiap field request yang dikirim dari form. lalu untuk menambah dengan memanggil method created() secara static Res::create(), Res::update() hanya untuk parsing hasil return boolean ke array dengan pesan "sukses/gagal" agar lebih mudah dibaca manusia

Tabel 4.22 Baris Perintah Menambahkan Data Kegiatan

|  |  |
| --- | --- |
|  | public function store(RequestStoreKegiatan $request){  $data = $request->validated();  if($request->gambar){  $nama\_gambar = $request->gambar->store('/kegiatan', 'public');  $data['gambar'] = $nama\_gambar;  }  return Res::store(  Kegiatan::create($data)  );  } |

Untuk mengambil satu data kegiatan berdasarkan id yang dikirim / dilaravel id langsung di cast ke object $kegiatan

Tabel 4.23 Baris Perintah Mengambil Data Spesifik Kegiatan

|  |  |
| --- | --- |
|  | public function show(Kegiatan $kegiatan){  return new KegiatanCollection($kegiatan);  } |

Sama seperti store pertama divalidasi dan kemudian data kegiatan yang diparsing/hasil cast dari id (mirip show), panggil method update() dengan mengirim hasil validasi data tadi

Tabel 4.24 Baris Perintah Mengubah Data Kegiatan

|  |  |
| --- | --- |
|  | public function update(RequestUpdateKegiatan $request, Kegiatan $kegiatan){  $data = $request->validated();  if($request->gambar){  $nama\_gambar\_old = $kegiatan->gambar;  $nama\_gambar = $request->gambar->store('/kegiatan', 'public');  $data['gambar'] = $nama\_gambar;  if($nama\_gambar && Storage::disk('public')->exists($nama\_gambar\_old)){  Storage::disk('public')->delete($nama\_gambar\_old);  }  }  return Res::update(  $kegiatan->update($data)  );  } |

Untuk menghapus salah satu data kegiatan, dapat dilakukan dengan memanggil method destory()

Tabel 4.25 Baris Perintah Menghapus Data Kegiatan

|  |  |
| --- | --- |
|  | public function destroy(Kegiatan $kegiatan){  return Res::delete($kegiatan->delete());  } |

* + 1. **Halaman Daftar Admin Admin**

Pada halaman admin terdapat beberapa proses utama yang terjadi diantaranya proses menampilkan data admin dan proses menambah dan mengubah data admin. Untuk menambahkan data admin dengan memanggil model dan diakhiri dengan memanggil method paginate(x), x dalam angka jumlah data yang akan ditampilkan per halaman, dan untuk menambahkan sakter dengan memanggil method create(x) dengan x sebagai array informasi data admin yang diinputkan, sedangkan untuk mengedit admin dengan memanggil method update(x) dari sebuah instance admin yang akan diedit, dengan x sebagai data array yang akan di update.

Tabel 4.26 Baris Perintah Menampilkan Data Admin

|  |  |
| --- | --- |
|  | public function index(){  return AdminCollection::collection(  User::when(request()->filled('search'), function($query){  $search = request('search');  return $query->where('nama', 'like', "%{$search}%");  })  ->when(request()->filled('sortBy'), function($query){  $sortBy = is\_array(request('sortBy')) ? request('sortBy')[0] : request('sortBy');  $sortDesc = is\_array(request('sortDesc')) ? request('sortDesc')[0] : request('sortDesc');  return $query->orderBy($sortBy, $sortDesc ? 'DESC' : 'ASC');  })  ->paginate(request('itemsPerPage') ?? 10)  );  } |

Untuk menambah sebuah data admin baru field / $\_POST yang dikirim dari form di bind ke object $request method validated() untuk mengambil data request yang dikirim dan sudah divalidasi untuk melihat validasinya misalkan untuk validasi admin di directory app/Http/Request/RequestStoreAdmin.php didalam file RequestStoreAdmin.php terdapat rule untuk setiap field request yang dikirim dari form. lalu untuk menambah dengan memanggil method created() secara static Res::create(), Res::update() hanya untuk parsing hasil return boolean ke array dengan pesan "sukses/gagal" agar lebih mudah dibaca manusia

Tabel 4.27 Baris Perintah Menambahkan Data Admin

|  |  |
| --- | --- |
|  | public function store(RequestStoreAdmin $request){  $data = $request->validated();  return Res::store(  User::create($data)  );  } |

Untuk mengambil satu data admin berdasarkan id yang dikirim / dilaravel id langsung di cast ke object $admin

Tabel 4.28 Baris Perintah Mengambil Data Spesifik Admin

|  |  |
| --- | --- |
|  | public function show(User $admin){  return new AdminCollection($admin);  } |

Sama seperti store pertama divalidasi dan kemudian data admin yang diparsing/hasil cast dari id (mirip show), panggil method update() dengan mengirim hasil validasi data tadi

Tabel 4.29 Baris Perintah Mengubah Data Admin

|  |  |
| --- | --- |
|  | public function update(RequestUpdateAdmin $request, User $admin){  $data = $request->validated();  $data['password'] = Hash::make($data['password']);  return Res::update(  $admin->update($data)  );  } |

Untuk menghapus salah satu data admin, dapat dilakukan dengan memanggil method destory()

Tabel 4.30 Baris Perintah Menghapus Data Admin

|  |  |
| --- | --- |
|  | public function destroy(User $admin){  return Res::delete(  $admin->delete()  );  } |

* 1. **Pengujian Sistem**

Pengujian sistem merupakan langkah ke lima dalam metode waterfall, pengujian sistem berhubungan dengan proses pengujian sistem yang telah dikerjakan. Pengujian yang akan digunakan untuk menguji sistem adalah metode pengujian sistem black box. Pengujian black box berfokus pada persyaratan fungsional perangkat lunak.

1. **Pengujian Halaman Login**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian validasi inputan, pengujian proses login, dan pengujian autentikasi redirect halaman login.

Tabel 4.31 Pengujian Validasi Input Halaman Login

|  |  |
| --- | --- |
| Pengujian Validasi Input Halaman Login | |
| Prosedur | Memasukan *email* dengan beberapa spasi pada *form login*, dan mengosongkan *password* |
| Input | *Email* dan *password* |
| Pengamatan | Menampilkan pesan error pada kedua kolom |
| Kesimpulan | Validasi input pada proses login bekerja seperti semestinya |

Tabel 4.32 Pengujian Proses Login

|  |  |
| --- | --- |
| Pengujian Proses Login | |
| Prosedur | Memasukan *email* dan *password* yang sudah terdaftar pada databasepada *form login* |
| Input | *Email* dan *password* |
| Pengamatan | Mengalihkan ke halaman Panel Admin |
| Kesimpulan | Proses Login Berhasil |

Tabel 4.33 Pengujian Autentikasi Redirect Halaman Login

|  |  |
| --- | --- |
| Pengujian Autentikasi Redirect Halaman Login | |
| Prosedur | Mengunjungi halaman login setelah berhasil login |
| Input | *-* |
| Pengamatan | Mengalihkan ke halaman Panel Admin |
| Kesimpulan | Berhasil, proses redirect autentikasi pada halaman login bekerja dengan semestinya |

1. **Pengujian Halaman Panel Admin**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian link navigasi panel admin dan pengujian autentikasi pada halaman admin.

Tabel 4.34 Pengujian Link Navigasi Panel Admin

|  |  |
| --- | --- |
| Pengujian Link Navigasi Panel Admin | |
| Prosedur | Mengunjungi setiap link yang ada pada navigasi panel admin. |
| Input | *-* |
| Pengamatan | Menampilkan halaman sesuai link yang diklik |
| Kesimpulan | Berhasil, link navigasi bekerja dengan baik |

Tabel 4.35 Pengujian Autentikasi Panel Admin

|  |  |
| --- | --- |
| Pengujian Autentikasi Panel Admin | |
| Prosedur | Mengunjungi halaman panel admin dalam keadaan sebagai guest atau tamu |
| Input | *-* |
| Pengamatan | Menampilkan halaman login |
| Kesimpulan | Berhasil, proses autentikasi pada halaman panel admin bekerja dengan baik |

1. **Pengujian Halaman Satker Admin**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian pencarian data satker dan pengujian tambah data satker.

Tabel 4.36 Pengujian Pencarian Data Satker Admin

|  |  |
| --- | --- |
| Pengujian Pencarian Data Satker Admin | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama satker |
| Input | *q* dengan nilai “can” |
| Pengamatan | Menampilkan data satker dengan nama “**Can**dace Mertz Jr.” |
| Kesimpulan | Berhasil, proses pencarian data satker bekerja dengan baik |

Tabel 4.37 Pengujian Penambahan Data Satker Admin

|  |  |
| --- | --- |
| Pengujian Penambahan Data Satker Admin | |
| Prosedur | Mengisi semua kolom pada form tambah data satker |
| Input | Nama Satker, Alamat, ISO, Status |
| Pengamatan | Data yang ditambahkan berhasil ditampilkan pada daftar satker |
| Kesimpulan | Berhasil, proses penambahan data satker bekerja dengan baik |

1. **Pengujian Halaman Struktur Organisasi Admin**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian pencarian data struktur organisasi, pengujian tambah data struktur organisasi.

Tabel 4.38 Pengujian Pencarian Data Strukur Organisasi Admin

|  |  |
| --- | --- |
| Pengujian Pencarian Data Struktur Organisasi Admin | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama satker |
| Input | *q* dengan nilai “can” |
| Pengamatan | Menampilkan data struktur oraganisasi dengan nama satker “**Can**dace Mertz Jr.” |
| Kesimpulan | Berhasil, proses pencarian data struktur organisasi bekerja dengan baik |

Tabel 4.39 Pengujian Penambahan Data Struktur Organisasi Admin

|  |  |
| --- | --- |
| Pengujian Penambahan Data Struktur Organisasi Admin | |
| Prosedur | Mengisi semua kolom pada form tambah data satker |
| Input | Gambar Struktur, Nama Satker |
| Pengamatan | Data yang ditambahkan berhasil ditampilkan pada daftar struktur organisasi |
| Kesimpulan | Berhasil, proses penambahan data struktur organisasi bekerja dengan baik |

1. **Pengujian Halaman Pimpinan Admin**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian pencarian data pimpinan, pengujian tambah data pimpinan.

Tabel 4.40 Pengujian Pencarian Data Pimpinan Admin

|  |  |
| --- | --- |
| Pengujian Pencarian Data Pimpinan Admin | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama pimpinan |
| Input | *q* dengan nilai “doe” |
| Pengamatan | Menampilkan data pimpinan dengan nama pimpinan “John **Doe**” |
| Kesimpulan | Berhasil, proses pencarian data pimpinan bekerja dengan baik |

Tabel 4.41 Pengujian Penambahan Data Pimpinan Admin

|  |  |
| --- | --- |
| Pengujian Penambahan Data Pimpinan Admin | |
| Prosedur | Mengisi semua kolom pada form tambah data pimpinan |
| Input | NIP, Nama, Nama Satker, Tempat Tanggal Lahir, Pangkat |
| Pengamatan | Data yang ditambahkan berhasil ditampilkan pada daftar pimpinan |
| Kesimpulan | Berhasil, proses penambahan data pimpinan bekerja dengan baik |

1. **Pengujian Halaman Bidang Admin**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian pencarian data bidang, pengujian tambah data bidang.

Tabel 4.42 Pengujian Pencarian Data Bidang Admin

|  |  |
| --- | --- |
| Pengujian Pencarian Data Bidang Admin | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama bidang |
| Input | *q* dengan nilai “ster” |
| Pengamatan | Menampilkan data bidang dengan nama bidang “Che**ster**” |
| Kesimpulan | Berhasil, proses pencarian data bidang bekerja dengan baik |

Tabel 4.43 Pengujian Penambahan Data Bidang Admin

|  |  |
| --- | --- |
| Pengujian Penambahan Data Bidang Admin | |
| Prosedur | Mengisi semua kolom pada form tambah data bidang |
| Input | Satker, Nama Kepala Bidang, Nama Bidang, Visi dan Misi, Tujuan |
| Pengamatan | Data yang ditambahkan berhasil ditampilkan pada daftar bidang |
| Kesimpulan | Berhasil, proses penambahan data bidang bekerja dengan baik |

1. **Pengujian Halaman Petugas Admin**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian pencarian data petugas, pengujian tambah data petugas.

Tabel 4.44 Pengujian Pencarian Data Petugas Admin

|  |  |
| --- | --- |
| Pengujian Pencarian Data Petugas Admin | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama petugas |
| Input | *q* dengan nilai “rus” |
| Pengamatan | Menampilkan data petugas dengan nama petugas “Elian **Rus**sel” |
| Kesimpulan | Berhasil, proses pencarian data petugas bekerja dengan baik |

Tabel 4.45 Pengujian Penambahan Data Petugas Admin

|  |  |
| --- | --- |
| Pengujian Penambahan Data Petugas Admin | |
| Prosedur | Mengisi semua kolom pada form tambah data petugas |
| Input | Bidang, Nama Petugas, NIP, Pangkat Golongan, Status Petugas, Alamat |
| Pengamatan | Data yang ditambahkan berhasil ditampilkan pada daftar petugas |
| Kesimpulan | Berhasil, proses penambahan data petugas bekerja dengan baik |

1. **Pengujian Halaman Kegiatan Admin**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian pencarian data kegiatan, pengujian tambah data kegiatan.

Tabel 4.46 Pengujian Pencarian Data Kegiatan Admin

|  |  |
| --- | --- |
| Pengujian Pencarian Data Kegiatan Admin | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama kegiatan |
| Input | *q* dengan nilai “ilaf” |
| Pengamatan | Menampilkan data petugas dengan nama kegiatan “Kegiatan **ilaf**urt” |
| Kesimpulan | Berhasil, proses pencarian data kegiatan bekerja dengan baik |

Tabel 4.47 Pengujian Penambahan Data Kegiatan Admin

|  |  |
| --- | --- |
| Pengujian Penambahan Data Kegiatan Admin | |
| Prosedur | Mengisi semua kolom pada form tambah data kegiatan |
| Input | Bidang, Nama Kegiatan, Tanggal Kegiatan, Lokasi Kegiatan, Rincian Kegiatan, Gambar Kegiatan (Opsional) |
| Pengamatan | Data yang ditambahkan berhasil ditampilkan pada daftar kegiatan |
| Kesimpulan | Berhasil, proses penambahan data kegiatan bekerja dengan baik |

1. **Pengujian Halaman Daftar Admin Admin**

Pengujian ini terdapat beberapa tahapan pengujian diantaranya pengujian pencarian data admin, pengujian tambah data admin.

Tabel 4.48 Pengujian Pencarian Data Admin Admin

|  |  |
| --- | --- |
| Pengujian Pencarian Data Admin Admin | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama admin |
| Input | *q* dengan nilai “ed” |
| Pengamatan | Menampilkan data petugas atau admin dengan nama admin “**Ed**d.” |
| Kesimpulan | Berhasil, proses pencarian data admin bekerja dengan baik |

Tabel 4.49 Pengujian Penambahan Data Admin Admin

|  |  |
| --- | --- |
| Pengujian Penambahan Data Admin Admin | |
| Prosedur | Mengisi semua kolom pada form tambah data admin |
| Input | Nama, Email, Kata Sandi, Konfirmasi Kata Sandi |
| Pengamatan | Data yang ditambahkan berhasil ditampilkan pada daftar admin |
| Kesimpulan | Berhasil, proses penambahan data admin bekerja dengan baik |

1. **Pengujian Halaman Petugas Oleh *Guest* atau Tamu**

Pengujian ini terhadap halaman awal bagian halaman petugas, pengunjian ini mengamati bagaimana pencarian halaman petugas bekerja dengan semestinya.

Tabel 4.50 Pengujian Pencarian Data Petugas oleh *Guest*

|  |  |
| --- | --- |
| Pengujian Pencarian Data Petugas oleh *Guest* | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama petugas |
| Input | *q* dengan nilai “rus” |
| Pengamatan | Menampilkan data petugas dengan nama “**Rus**sell Kessler.” |
| Kesimpulan | Berhasil, proses pencarian data petugas bekerja dengan baik |

1. **Pengujian Halaman Bidang Oleh *Guest* atau Tamu**

Pengujian ini terhadap halaman awal bagian halaman bidang bagaimana pengujian ini mengamati pencarian halaman bidang bekerja dengan semestinya.

Tabel 4.51 Pengujian Pencarian Data Bidang oleh *Guest*

|  |  |
| --- | --- |
| Pengujian Pencarian Data Bidang oleh *Guest* | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama bidang |
| Input | *q* dengan nilai “Air minum” |
| Pengamatan | Menampilkan data bidang dengan nama “PPK **AIR MINUM**” |
| Kesimpulan | Berhasil, proses pencarian data bidang bekerja dengan baik |

1. **Pengujian Halaman Kegiatan Oleh *Guest* atau Tamu**

Pengujian ini terhadap halaman awal bagian halaman kegiatan bagaimana pengujian ini mengamati pencarian halaman kegiatan bekerja dengan semestinya.

Tabel 4.51 Pengujian Pencarian Data Kegiatan oleh *Guest*

|  |  |
| --- | --- |
| Pengujian Pencarian Data Kegiatan oleh *Guest* | |
| Prosedur | Mengetikkan satu kata yang terkandung pada beberapa nama kegiatan |
| Input | *q* dengan nilai “east” |
| Pengamatan | Menampilkan data kegiatan dengan nama “Kegiatan **East** Beulah” dam “Kegiatan **East** Alta” |
| Kesimpulan | Berhasil, proses pencarian data kegiatan bekerja dengan baik |

**BAB V**

**PENUTUP**

* 1. **KESIMPULAN**

Dari hasil Tugas Akhir mengenai “Sistem Informasi Berbasis Web pada Kantor Dinas Pekerjaan Umum dan Perumahan Rakyat Satker wilayah 1 provinsi Nusa Tenggara Timur”, maka penulis dapat mengambil kesimpulan sebagai berikut:

1. Merancang serta membangun sistem informasi pada kantor Dinas Pekerjaan Umum dan Perumahan Rakyat Satker Wilayah I Provinsi Nusa Tenggara Timur.
2. Menyajikan informasi kepada masyarakat yang berkaitan dengan kantor Dinas Pekerjaan Umum dan Perumanah Rakyat Satker Wilayah I Provinsi Nusa Tenggara Timur berbasis web.
3. Sehingga menjawab manfaat mempermudahkan pegawai untuk menyampaikan informasi mengenai kantor Dinas Pekerjaan Umum dan Perumanah Rakyat Satker Wilayah I Provinsi Nusa Tenggara Timur berbasis web secara akurat.
   1. **SARAN**

Dalam pembuatan Laporan Tugas Akhir Ini, Penulis akan memberikan saran yaitu:

Dalam pengembangan Website ini perlu adanya aplikasi dari sistem lain seperti Aplikasi Android atau Aplikasi Mobile lainnya.