

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Aktivitas Kerja Profesi pada Kementerian Pekerjaan Umum dan Perumahan Rakyat (PUPR) dilakukan selama kurang lebih 3 (tiga) bulan. Praktikan bekerja menjadi salah satu anggota Pusat Data dan Teknologi Informasi (PUSDATIN) dan dibimbing oleh karyawan PUSDATIN yang bernama Pak Danang, Pak Sam dan Pak Yohannes selaku kepala Bidang Manajemen Teknologi Informasi. Sebelumnya kami melakukan rapat mengenai permasalahan yang ada di PUSDATIN dan mencari solusi apa yang harus dilakukan. Sesudah itu dibuatlah sebuah tim terkait masalah tersebut dengan modul yang berbeda dan pemberian tugas – tugas yang akan dilaksanakan. Setelah melakukan pembagian tugas, Praktikan ditugaskan menjadi UI/UX Designer dan *front-end development* pada perancangan aplikasi “*E-monitoring* SSL”. Adapun kegiatan yang dilakukan sebagai UI/UX dan *front-end Development* yaitu melakukan analisis desain yang sudah dibuat oleh UI/UX (Praktikan), kemudian mengimplementasikan hasil desain menggunakan visual studio code sebagai code editor, HTML 5 serta CSS dan javascript untuk memperindah tampilan.

Aplikasi ini dirancang untuk menyelesaikan dan memudahkan masalah dalam monitoring sertifikat pada website – website PUSDATIN. Lalu sebagai *front-end development* dapat membantu penggunaan aplikasi yang mudah dipahami oleh pengguna. Selama kegiatan pembuatan aplikasi berlangsung, Praktikan selalu memperhatikan keperluan pengguna. Selain itu Praktikan selalu melakukan koordinasi dengan pembimbing dan tim agar tidak terjadinya kesalahpahaman dan selalu mengkonfirmasi kepada bagian Back-end.

3.2 Pelaksanaan Kerja

Praktikan melaksanakan kegiatan tersebut terhitung tiga bulan, yang mana dilaksanakan pada tanggal 13 Juni 2022 hingga 13 September 2022. Ditempatkan di Pusat Data dan Teknologi Informasi bagian Manajemen Teknologi Informasi dengan mengambil posisi UI/UX dan *Frontend Development* untuk merancang aplikasi *E-monitoring* SSL. Aplikasi *E-monitoring* adalah aplikasi yang dapat memberikan kemudahan dalam melakukan pengecekan SSL pada setiap website yang ada di Kementerian PUPR. Tahapan pelaksanaan Kerja Profesi ini dimulai dengan perencanaan sistem yang akan dibuat, setelah itu melakukan analisis terhadap kebutuhan sistem dan *user*, kemudian perancangan desain awal seperti membuat *Activity diagram*, *mockup* aplikasi, lalu tahap implementasi dengan membuat antarmuka aplikasi hingga tahap uji coba sistem dan mensinkronisasikan dengan pihak *back-end*.

3.2.1 Meeting Kerja Profesi

Tahap awal kegiatan Kerja Profesi adalah bertemu dan berbincang tentang Kerja Profesi dengan Bapak Yohannes selaku Kepala Bidang Manajemen Teknologi Informasi. Dalam pertemuan tersebut, dijelaskan secara singkat perkenalan dan pekerjaan apa saja yang dilakukan oleh karyawan di Bidang Manajemen Teknologi Informasi. kemudian Bapak Yohannes memberikan suatu masalah kepada Mahasiswa Universitas Pembangunan Jaya sebagai studi kasus tentang banyaknya website Kementerian PUPR yang tidak dapat dipantau sertifikat SSL-nya secara keseluruhan. Dari permasalahan tersebut, mahasiswa diberikan arahan tentang perancangan aplikasi yang akan dibuat bersama oleh pembimbing yaitu Pak Sam untuk pembagian tugasnya. Tugas yang diberikan oleh pembimbing kepada praktikan adalah UI/UX *Designer* dan *Front-End Development* dan rekan Praktikan Iben Hadi Putro selaku *back-end*.

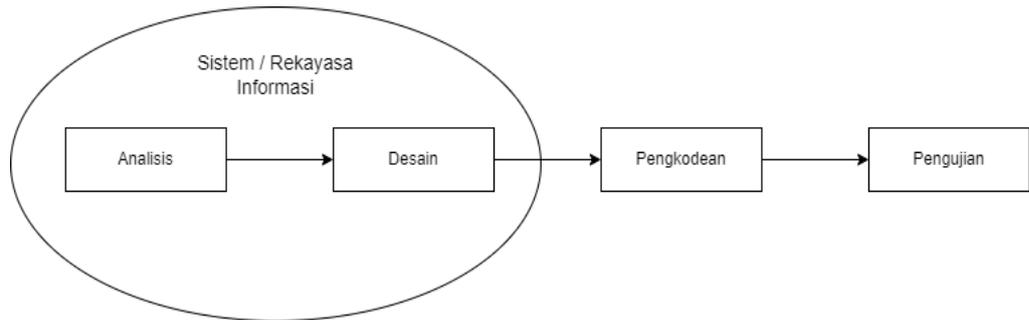
Sebagai UI/UX *Designer* bertugas membuat gambaran *user* interface yang dapat mudah digunakan dan dipahami oleh *user* (pengguna) atau biasa disebut *user-friendly*. Tugas UI/UX mencakup *wireframe*, *prototyping* hingga *mockup* dan juga perencanaan kebutuhan *user*. Pada tugas kali ini Praktikan diharuskan membuat desain *wireframe* dengan tampilan desktop, kemudian membuat

prototyping yang akan menjadi gambaran alur proses aplikasi. Lalu mencocokkan dan memenuhi kebutuhan pengguna pada suatu sistem, yang mana pengguna akan selalu melakukan penilaian pada sistem informasi bukan hanya fungsi saja, akan tetapi dari desain antarmuka. Sebagai UI/UX diperlukan untuk meningkatkan pengalaman pengguna agar aplikasi mudah digunakan.

Dan *Frontend development* melakukan tugas implementasi hasil desain atau interface dalam bentuk code menjadi aplikasi. *Frontend* mengembangkan interface baru untuk para pengguna, sehingga pengguna dapat menggunakan aplikasi dengan mudah dan efisien. Dalam tugasnya frontend diharuskan mempunyai kemampuan seperti penguasaan terhadap HTML, *Javascript*, *Cascading Style Sheets* (CSS) dan *Jquery* serta *figma* sebagai *software editing*.

3.2.2 Perencanaan dan Analisa Aplikasi

Setelah melakukan perencanaan terkait masalah yang dihadapi, Praktikan melakukan analisa terhadap *user requirements* (kebutuhan pengguna) seperti siapa saja yang akan menggunakan aplikasi tersebut, kemudian analisa kebutuhan dari sisi aplikasi (*system requirements*), dan alur proses aplikasi saat digunakan. Pada pengembangan suatu sistem dibutuhkan sebuah metode yang dapat memudahkan dalam melakukan analisa, perancangan, mengidentifikasi masalah, serta membangun aplikasi dengan harapan dapat menyelesaikan masalah. Salah satu metode yang mendekati hal tersebut adalah *Software development Life Cycle* (SDLC) dengan pendekatan air terjun (*waterfall*). *Software Development Life Cycle* (SDLC) merupakan sebuah metode pengembangan pada suatu sistem perangkat lunak yang menggunakan sebuah model dan metode untuk mengembangkan sistem, yang mana terdiri dari beberapa langkah – langkah yang terorganisir dan tersusun. Lalu model *waterfall* adalah pendekatan pengembangan perangkat lunak yang dilakukan secara terurut dan teratur. Pengembangan tersebut, dimulai dari tahap analisis, tahap desain, tahap *coding* (implementasi), dan tahap *testing* (pengujian) (Sukamto & Shalahuddin, 2016). Berikut pada **Gambar 3.1**.



Gambar 3.1 SDLC Pendekatan Waterfall
 Sumber: (Sukamto & Shalahuddin, 2016)

Kemudian pada tahap selanjutnya, praktikan melakukan analisa kebutuhan untuk menentukan proses yang terjadi pada sistem dengan pengguna, sebagai langkah awal dalam pengembangan sistem yang akan dibangun. Analisa kebutuhan merupakan proses mengenai pengumpulan persyaratan secara lengkap dan menyeluruh untuk menentukan persyaratan perangkat lunak untuk memahami apa yang dibutuhkan pengguna (user). Dokumentasi diperlukan pada tahap ini untuk menggambarkan analisis definisi kebutuhan perangkat lunak (Sukamto & Shalahuddin, 2016). Dan berikut gambaran analisa kebutuhan perangkat lunak pada **Tabel 3.1**.

Tabel 3.1 Analisa Kebutuhan

FUNCTIONAL

ANALISA KEBUTUHAN

Saya ingin sistem dapat:

- 1 Sistem menampilkan Form Login
- 2 Sistem Menampilkan *Dashboard*
- 3 Sistem Menampilkan daftar data Website
- 4 Sistem Menampilkan Status SSL Website
- 5 Sistem Menampilkan *DayLeft* SSL
- 6 *User* dapat menambah data website
- 7 *User* dapat mengubah data website
- 8 *User* dapat menghapus data website
- 9 *User* dapat melakukan logout

NON FUNCTIONAL

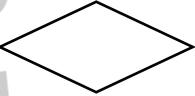
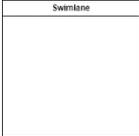
ANALISA KEBUTUHAN

Saya ingin sistem dapat:

- 1 Sistem *User Friendly*
- 2 Sistem akan memberitahukan apabila salah page (halaman)
- 3 Sistem akan menginformasikan apabila salah *username* atau *password*

Sesudah menentukan kebutuhan untuk perangkat lunak selanjutnya praktikan membuat alur proses sistem pada saat digunakan yaitu *Activity diagram*. *Activity diagram* merupakan *workflow* atau sebuah proses bisnis yang ada pada perangkat lunak (Sukamto & Shalahuddin, 2016). Adapun beberapa komponen – komponen yang ada pada *Activity diagram* seperti **Tabel 3.2**.

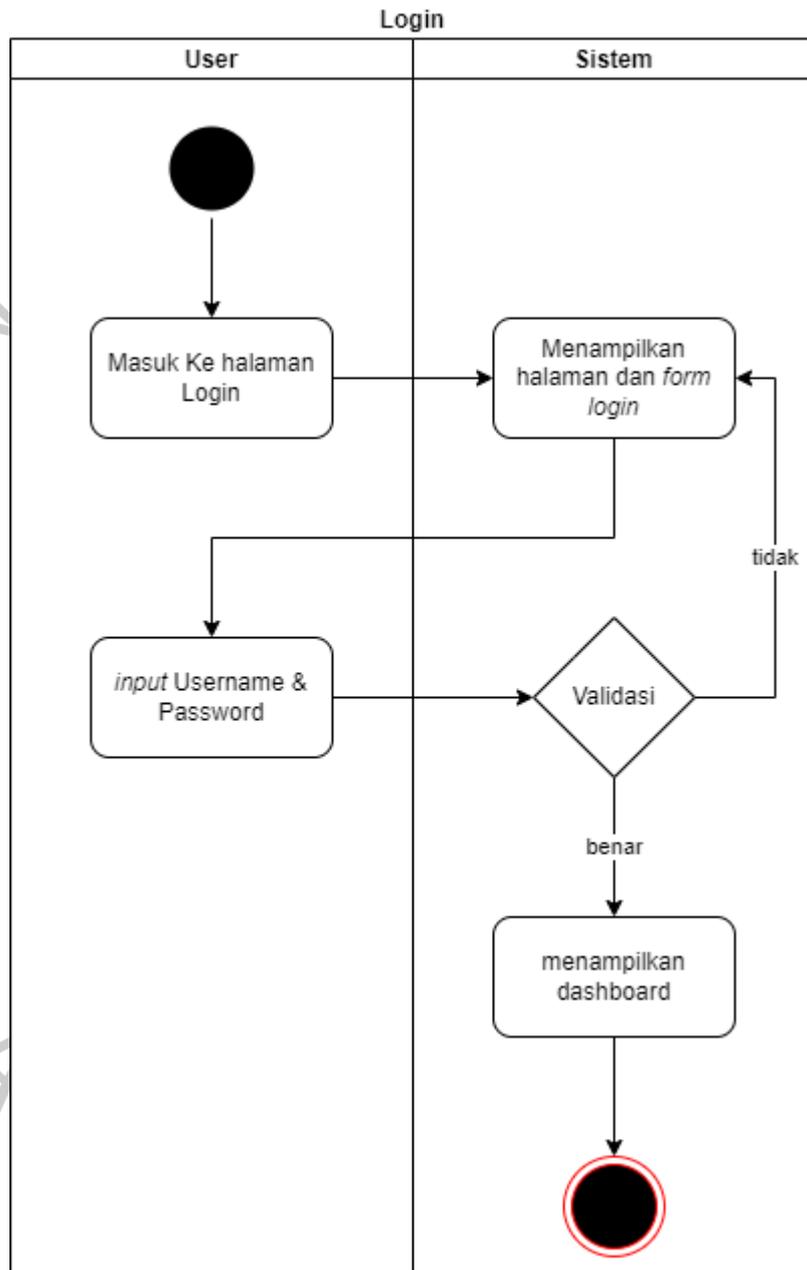
Tabel 3.2 Komponen-Komponen *Activity Diagram*

| Simbol | Nama | Keterangan |
|---|----------------------|--|
|  | Status Awal | sebuah bentuk yang menjadi sebuah penanda awalnya kegiatan atau status awal kegiatan |
|  | Kegiatan/Aktivitas | diagram yang dikerjakan oleh sistem, dan diawali dengan kata kerja |
|  | Percabangan/Decision | sebuah percabangan yang dapat memilih lebih dari satu |
|  | Penggabungan/Join | Penggabungan antara satu kegiatan dengan kegiatan yang lain menjadi satu |
|  | Status Akhir | bentuk penanda bahwa sistem telah berakhir |
|  | Swimlane | sebuah batas proses bisnis tertentu pada tiap kegiatan |

Sumber: (Sukamto & Shalahuddin, 2016)

Selanjutnya, berikut seluruh alur *Activity diagram* dari *e-monitoring* SSL yang dimulai dari **Gambar 3.2** hingga sampai **Gambar 3.7**.

a) *Activity diagram* login

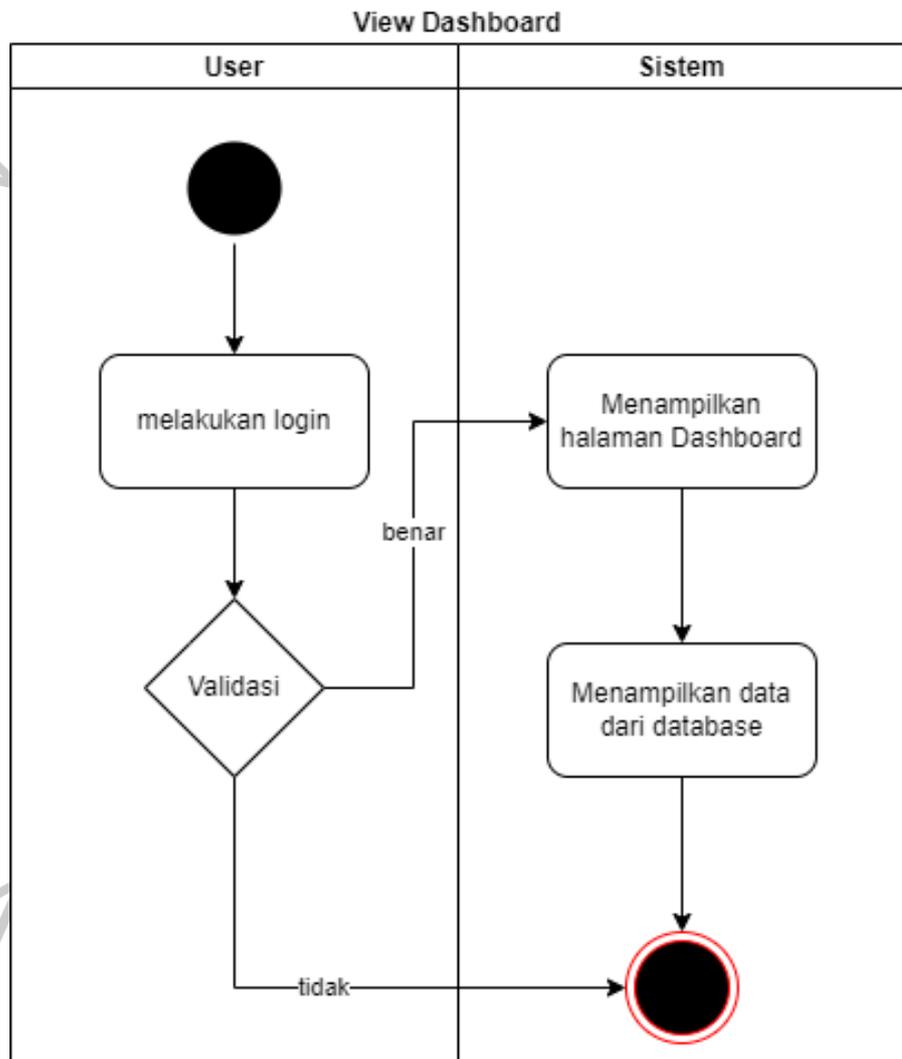


Gambar 3.2 Activity Diagram Login
Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.2** yang merupakan kegiatan awal, *user* (pengguna) akan membuka dan masuk ke halaman web, kemudian sistem menerima respon dan menampilkan halaman sekaligus form login, kemudian pengguna akan melakukan *input* data *username*

dan *password*. Setelah itu, sistem menerima data dan memeriksa data di database. Jika informasi tidak cocok, sistem mengembalikan halaman login dan menampilkan pesan peringatan "Pemberitahuan". bahwa nama pengguna atau kata sandi salah. Jika informasi sesuai maka sistem merespon, user login, dan sistem menampilkan halaman dashboard.

b) Activity diagram view dashboard

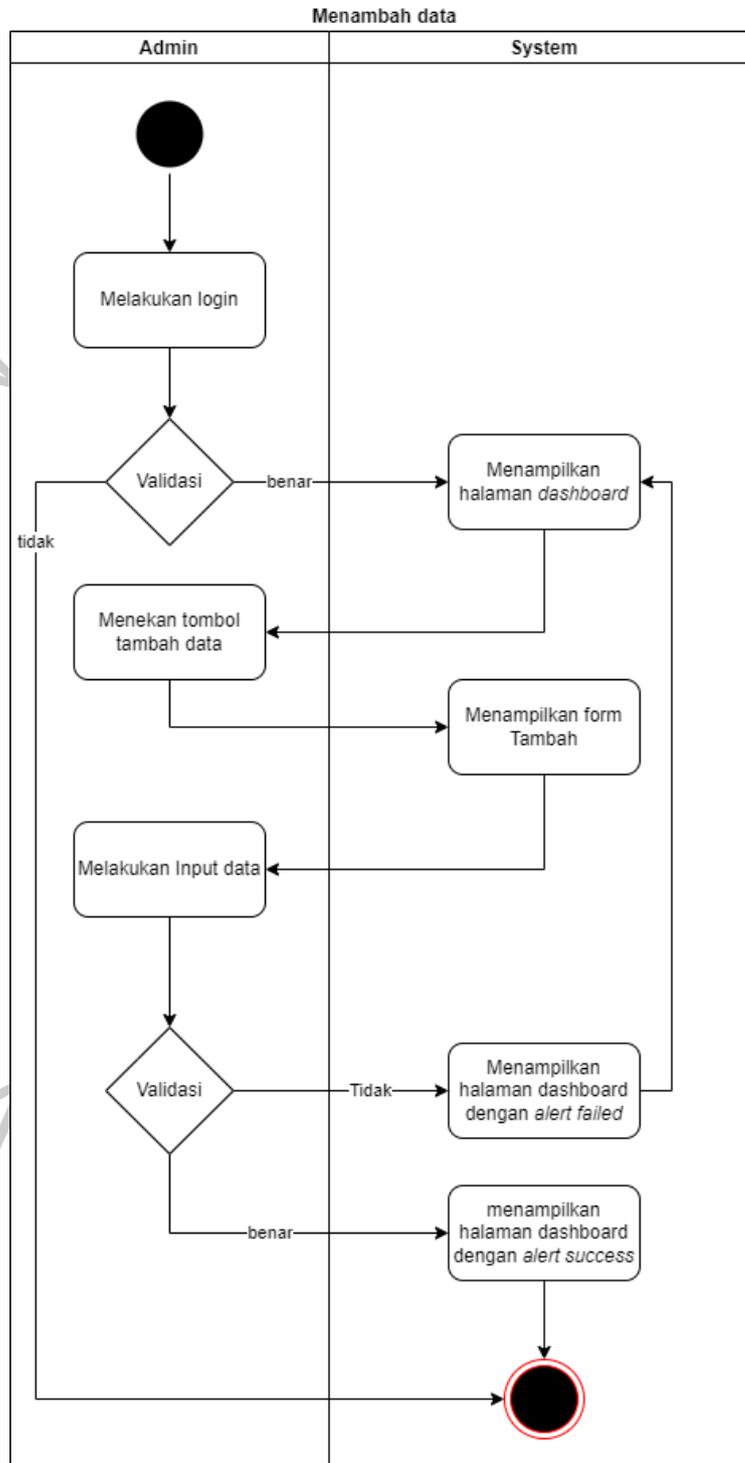


Gambar 3.3 Activity Diagram Dashboard
 Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.3** ini. *User* (pengguna) harus melakukan login terlebih dahulu. Jika tidak maka *user* (pengguna) akan kembali ke halaman login untuk melakukan login. Jika benar, maka sistem akan menampilkan halaman *dashboard*, kemudian menampilkan

semua data seluruh data dari database seperti nama website, Organisasi, Unit kerja dan IP address).

c) Activity diagram add data dashboard



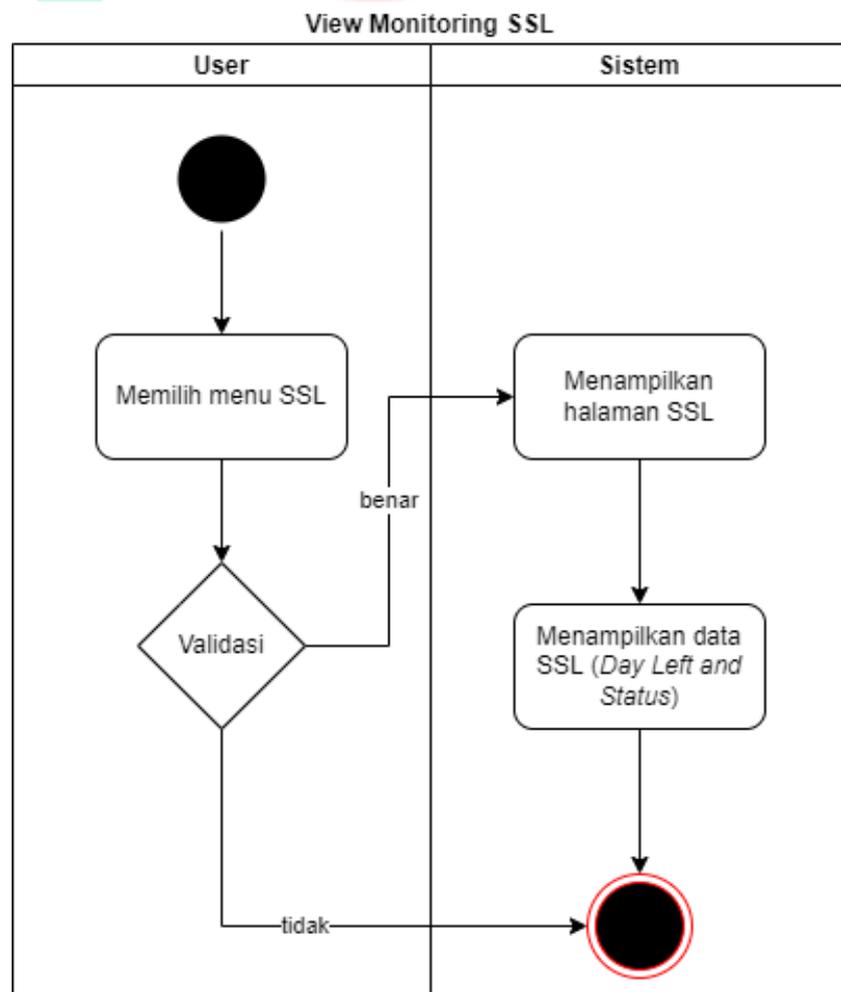
Gambar 3.4 Activity Diagram Add Data Dashboard

Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.4** setelah view data maka seorang *user* dapat melakukan penambahan data dengan melakukan mengklik tombol *add data*, kemudian sistem akan menampilkan sebuah modal yang di dalamnya terdapat form tambah data. Selanjutnya *user* akan mengisi form tersebut sesuai kemauannya. Jika tambah data gagal maka akan dikembalikan ke halaman awal dengan pemberitahuan bahwa proses tambah data gagal. Dan jika tambah data sukses maka akan kembali ke *dashboard* dengan notifikasi tambah data berhasil.

d) Activity diagram view monitoring SSL

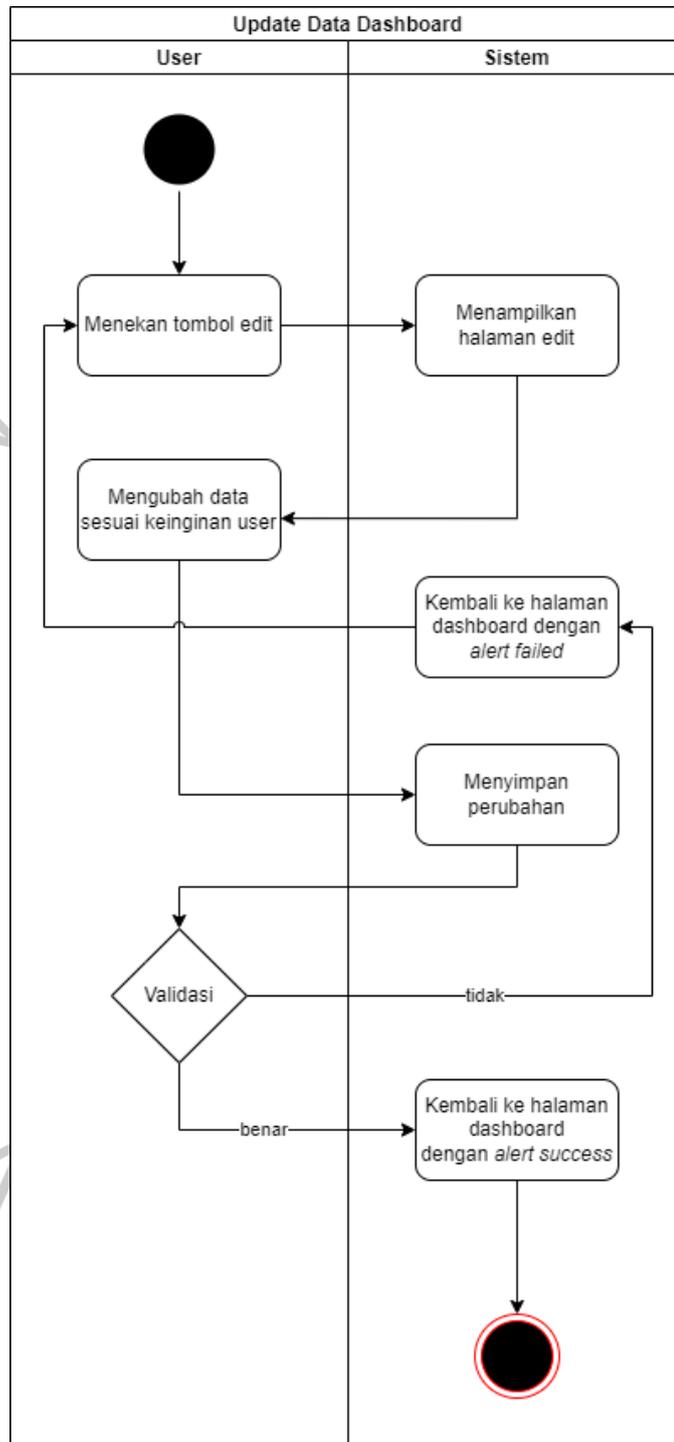
Pada **Gambar 3.5**. Setelah *user* memilih menu SSL maka sistem akan menampilkan halaman SSL yang berisikan nama ip adress di dalamnya terdapat link tiap-tiap website serta dapat melihat data seperti status dan *day left* SSL.



Gambar 3.5 Activity Diagram View Monitoring SSL

Sumber: Hasil Dokumentasi Praktikan

e) *Activity diagram update data dashboard*

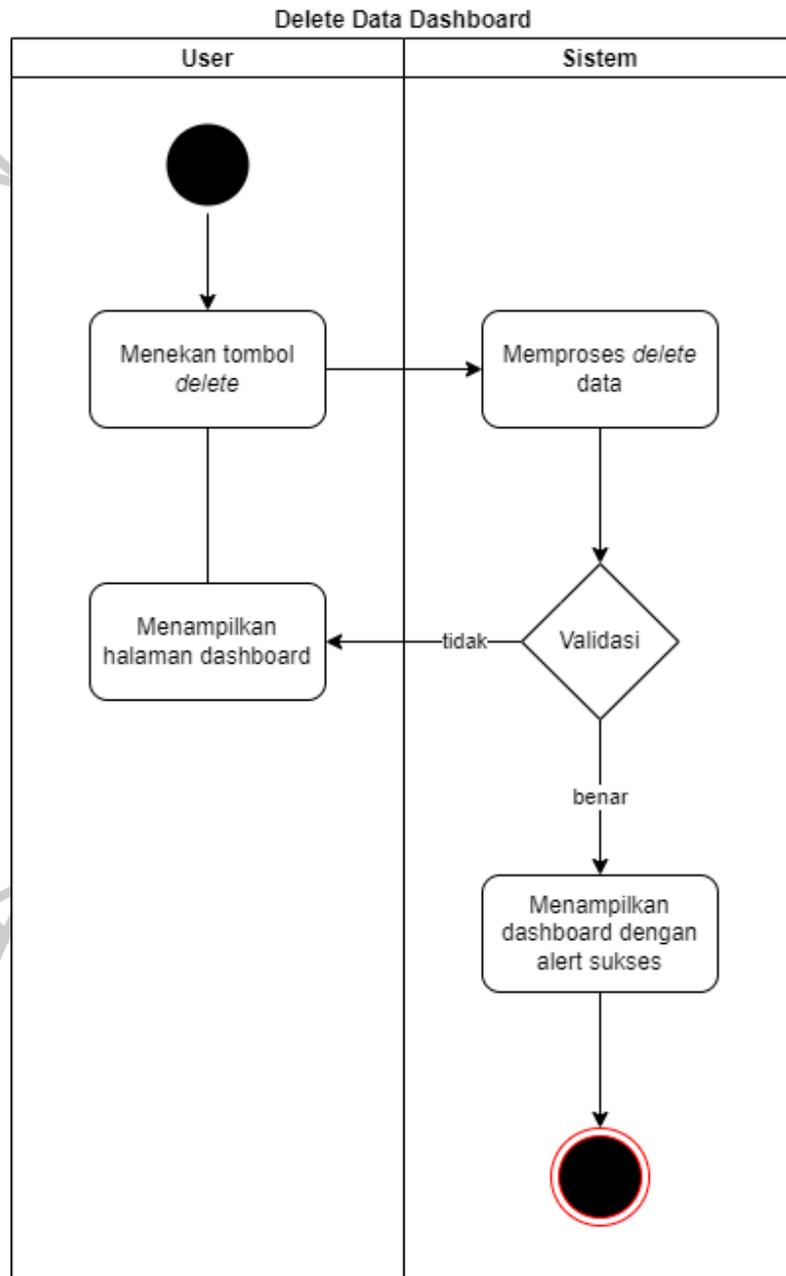


Gambar 3.6 Activity Diagram Update Data Dashboard
 Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.6**. Sesudahnya *user* melakukan login langkah selanjutnya dengan menekan tombol edit untuk melakukan *update*

data pada tabel *dashboard*, setelah itu *user* akan mengubah atau menambah data yang diinginkan. Kemudian setelah selesai *user* dapat menyimpan hasil perubahan dengan menekan tombol *save*. Jika gagal maka akan kembali ke halaman *dashboard* dengan *alert* gagal, jika benar maka akan kembali ke halaman *dashboard* dengan *alert* sukses.

f) **Activity diagram delete data dashboard**



Gambar 3.7 Activity Diagram Delete Data Dashboard
 Sumber: Hasil Dokumentasi Praktikan

Gambar 3.7 di atas merupakan proses penghapusan data pada tabel *dashboard*. Pengguna akan menekan tombol *delete* yang kemudian akan diproses oleh sistem dan sebelum terjadinya penghapusan data, pengguna akan ditanya apakah yakin ingin menghapus data atau tidak. Jika yakin ingin menghapus data, maka data akan terhapus dan pengguna akan kembali ke halaman awal dashboard. Dan jika tidak yakin untuk menghapus, maka data tersebut tidak hilang. Setelah data tersebut berhasil dihapus, sistem akan memberikan sebuah pemberitahuan bahwa data berhasil dihapus dan jika gagal maka akan menampilkan bahwa data gagal dihapus.

Setelah Praktikan menentukan elisitasi dan *activity diagram*. Praktikan mengikuti rapat mengenai koordinasi terhadap perancangan dan implementasi sistem. Oleh karena itu Praktikan membuat timeline pengerjaan menggunakan *gant chart*. Praktikan membuat alur dimulai dengan perencanaan yaitu melakukan perkenalan dengan lingkungan pekerjaan (PUPR dan PUSDATIN), kemudian penjelasan tentang PUSDATIN menaungi website pada unit kerja dan organisasi serta mengikuti rapat koordinasi yang dilakukan pada minggu pertama dan kedua.

Setelah melakukan perencanaan, langkah berikutnya adalah menentukan kebutuhan user berdasarkan hasil rapat tersebut. Kebutuhan user ini menjadi acuan dalam perancangan wireframe aplikasi yang akan dikerjakan oleh UI/UX *designer* sebagai langkah awal pembuatan aplikasi. Kemudian Praktikan membuat alur pengguna atau *userflow* yang bertujuan untuk memudahkan Praktikan saat *prototyping*.

Hasil *prototyping* tersebut, diimplementasikan ke dalam sebuah code program, sehingga dapat berjalan sesuai apa yang sudah dibuat oleh UI/UX *designer*. Lalu Praktikan melakukan tahap ujicoba pada aplikasi untuk mengetahui apa saja fungsi-fungsi yang tidak berjalan pada aplikasi. Kegiatan tersebut Praktikan buat dalam bentuk tabel. Berikut tabel Gantt chart seperti pada

Tabel 3.3 Gantt Chart

| No | Nama Kegiatan | Bulan | | | | | | | | | | | | Ket. | |
|----|---|-------|---|------|---|---|---|------|---|---|---|------|---|------|---|
| | | Juni | | Juli | | | | Agst | | | | Sept | | | |
| | | 1 | 2 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | | |
| 1 | Perkenalan lingkungan kerja, penjelasan dan pembagian divisi yang akan ditempati | | | | | | | | | | | | | | 1) bertemu KA. Manajemen Teknologi Informasi |
| 2 | Rapat dan penjelasan website yang ada di PUSDATIN dan rencana aplikasi yang akan dibuat | | | | | | | | | | | | | | 1) pengenalan website dan masalah yang terjadi pada PUSDATIN |
| 3 | Rapat Koordinasi dengan user dan KA. Manajemen Teknologi Informasi | | | | | | | | | | | | | | 1) membantu dalam perencanaan awal aplikasi yang akan di bangun |
| 4 | Rapat koordinasi rancangan aplikasi monitoring (Uptime, SSL, Dokumentasi) | | | | | | | | | | | | | | 1) melakukan analisis data dan kebutuhan sistem |
| 5 | Pembuatan Wireframe dan asistensi wireframe | | | | | | | | | | | | | | 1) membuat desain awal pada aplikasi |
| 6 | <i>Prototyping</i> dari wireframe dan implementasi koding | | | | | | | | | | | | | | 1) membuat alur aplikasi dan membuat <i>interface</i> website |
| 7 | <i>Testing pertama</i> Aplikasi dengan tim dan melakukan asistensi ke pihak User | | | | | | | | | | | | | | 1) melakukan <i>testing</i> terhadap <i>interface</i> |
| 8 | <i>Testing</i> kedua bersama dengan user dan serah terima | | | | | | | | | | | | | | 1) melakukan <i>testing</i> aplikasi yang sudah direvisi |
| 9 | Mengirim file aplikasi ke pihak IT User | | | | | | | | | | | | | | 1) menupload semua file aplikasi ke github dan diserahkan kepada pihak PUSDATIN |
| 10 | Konsultasi dan Pembuatan Laporan Kerja Profesi | | | | | | | | | | | | | | 1) melakukan bimbingan mengenai kerja profesi dan laporan |

Sumber: Hasil Dokumentasi Praktikan

3.2.3 Perancangan Desain UI/UX Aplikasi

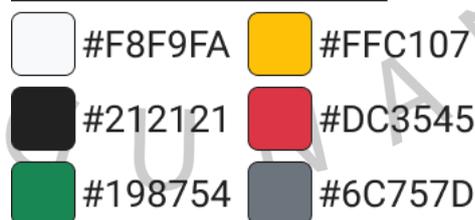
Setelah membuat kebutuhan sistem, tahap selanjutnya adalah membuat perancangan UI/UX *design* pada aplikasi dengan menerapkan konsep *human computer interaction* (HCI) atau Interaksi Manusia dan Komputer (IMK). IMK sendiri sudah praktikan pelajari pada semester 2 (dua) dengan 3 (tiga) sks, sehingga ilmu tersebut dapat diterapkan pada kegiatan ini. Dengan menggunakan konsep IMK yang diterapkan seperti merancang dan mengimplementasi desain agar pengguna dapat dengan mudah menggunakan komputer. Interaksi Manusia Komputer adalah bagian multidisiplin dalam melakukan perancangan produk yang dapat mudah digunakan oleh manusia (Dalle, Mutalib, Shaari, & Salam, 2019).

Pada perancangan ini praktikan membuat *user interface* sebagai gambaran awal aplikasi yang akan dibuat sehingga dapat menjelaskan komponen apa saja yang harus dirancang. Untuk merancang desain *interface* tersebut, Praktikan menggunakan *software* bernama figma. Saat melakukan perancangan *user interface* Praktikan membuat komponen – komponen desain yang terdiri dari *color palette* dan *font*. Berikut contoh *color palette* dan *font* yang Praktikan pada dibawah ini.

Typhography

font: Roboto

Brand Color



Gambar 3.8 Color Palette dan Font
Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.8** Font yang digunakan bernama *Roboto* dan *font weight* yang dipakai adalah 100 (*thin*), 300 (*light*), 400 (*regular*), 500 (*medium*), 700 (*bold*) dan 900 (*black*). Pada komponen warna Praktikan menggunakan warna yang nyaman dipandang, sehingga

memudahkan pengguna dalam menggunakan semua kegiatan pada aplikasi ini. Berikut adalah tampilan keseluruhan desain *user interface* yang telah praktikan desain untuk e-monitoring SSL:

1. Halaman *Login*

The image shows a login form with a white background and a thin border. At the top center is a logo consisting of two overlapping yellow and blue shapes. Below the logo is the word 'LOGIN' in a bold, black, sans-serif font. Underneath are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. At the bottom of the form is a black button with the word 'MASUK' in white, bold, uppercase letters.

Gambar 3.9 *User Interface Login*
Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.9** terdapat beberapa komponen yang praktikan buat seperti *form* yang terdiri dari *background form*, logo perusahaan, *Heading 2*, *field input username*, *field input password* dan *button masuk*. Dari tampilan ini memiliki 2 (dua) warna *primary* yaitu putih dan hitam sebagai warna netral dan tidak terjadi kontras terhadap warna lain. Diluar *form login* tepatnya bagian bawah (*footer*) halaman terdapat teks yang bertuliskan hak cipta pada aplikasi dan tahun. Halaman ini dibuat minimal mungkin agar pengguna yang ingin memakai aplikasi dapat mudah menggunakan terutama pada halaman pertama.

2. Halaman *Dashboard*

Pada **Gambar 3.10** yang merupakan *user interface dashboard* ini menjadi halaman utama setelah melakukan login. Halaman dashboard juga memiliki beberapa *section* yang membuat *interface* menjadi beberapa bagian. Berikut bagian – bagian yang ada di halaman dashboard sebagaimana berikut:

- a) Bagian pertama yaitu *header* atau *topbar*, yang mana bagian ini berada diatas halaman dan berwarna hitam sebagai warna *primary*. Di dalam bagian *topbar* terdapat beberapa komponen yang terdiri dari *icon burger* atau 3 (tiga) garis horizontal berwarna putih dan pada pojok kanan atas

- halaman terdapat *icon* lingkaran berwarna abu – abu untuk pengguna sekaligus menjadi tombol untuk pengguna *logout*.
- b) Bagian kedua yaitu *sidebar*, bagian ini berada disamping halaman dengan warna yang sama seperti *topbar* yaitu berwarna hitam. Bagian *sidebar* berisikan beberapa komponen salah satunya nama aplikasi dengan warna putih dibawahnya terdapat menu aplikasi yang terdiri dari *dashboard* dan *monitoring* SSL serta kotak berwarna abu – abu yang berisikan pengguna yang login saat itu.
 - c) Bagian ketiga yaitu *content*, di bagian ini berisikan tulisan tanda halaman dan ucapan selamat datang kepada pengguna. Kemudian di bawahnya terdapat sebuah table yang berisikan data – data dari database. tabel ini memiliki *button* tambah data, edit data dan hapus data, serta field search data. Lalu di bagian bawah terdapat *pagination* halaman yang menjadikan tabel tersebut dinamis.

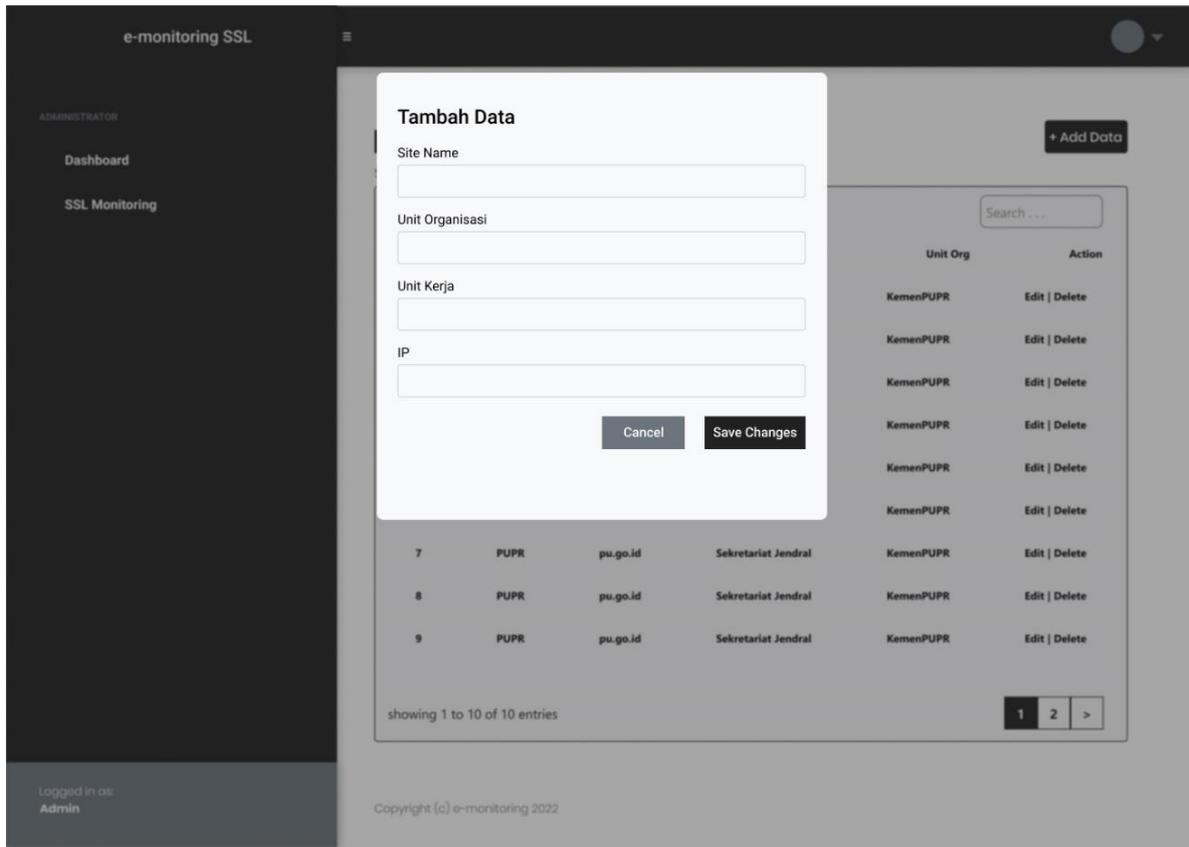
The screenshot shows the 'e-monitoring SSL' dashboard. The sidebar on the left includes 'ADMINISTRATOR', 'Dashboard', and 'SSL Monitoring'. The main content area displays a 'Dashboard' with a 'Selamat Datang Admin' message. Below this is a table with 9 rows of data. The table has columns for 'No', 'Site Name', 'IP', 'Unit Kerja', 'Unit Org', and 'Action'. The data in the table is as follows:

| No | Site Name | IP | Unit Kerja | Unit Org | Action |
|----|-----------|----------|---------------------|-----------|---------------|
| 1 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 2 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 3 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 4 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 5 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 6 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 7 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 8 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 9 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |

At the bottom of the table, it says 'showing 1 to 10 of 10 entries'. The pagination controls show '1', '2', and '>'. The sidebar also shows 'Logged in as: Admin'.

Gambar 3.10 User Interface Dashboard
Sumber: Hasil Dokumentasi Praktikan

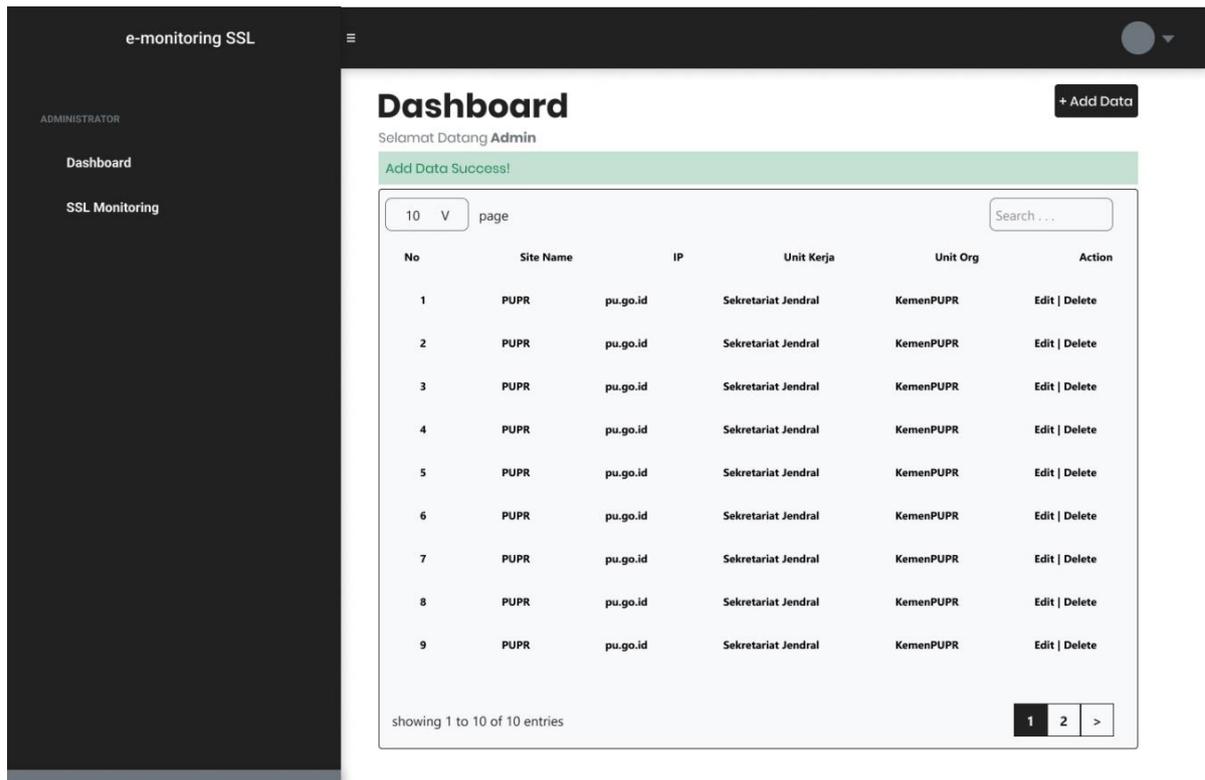
3. Halaman *Add Data*



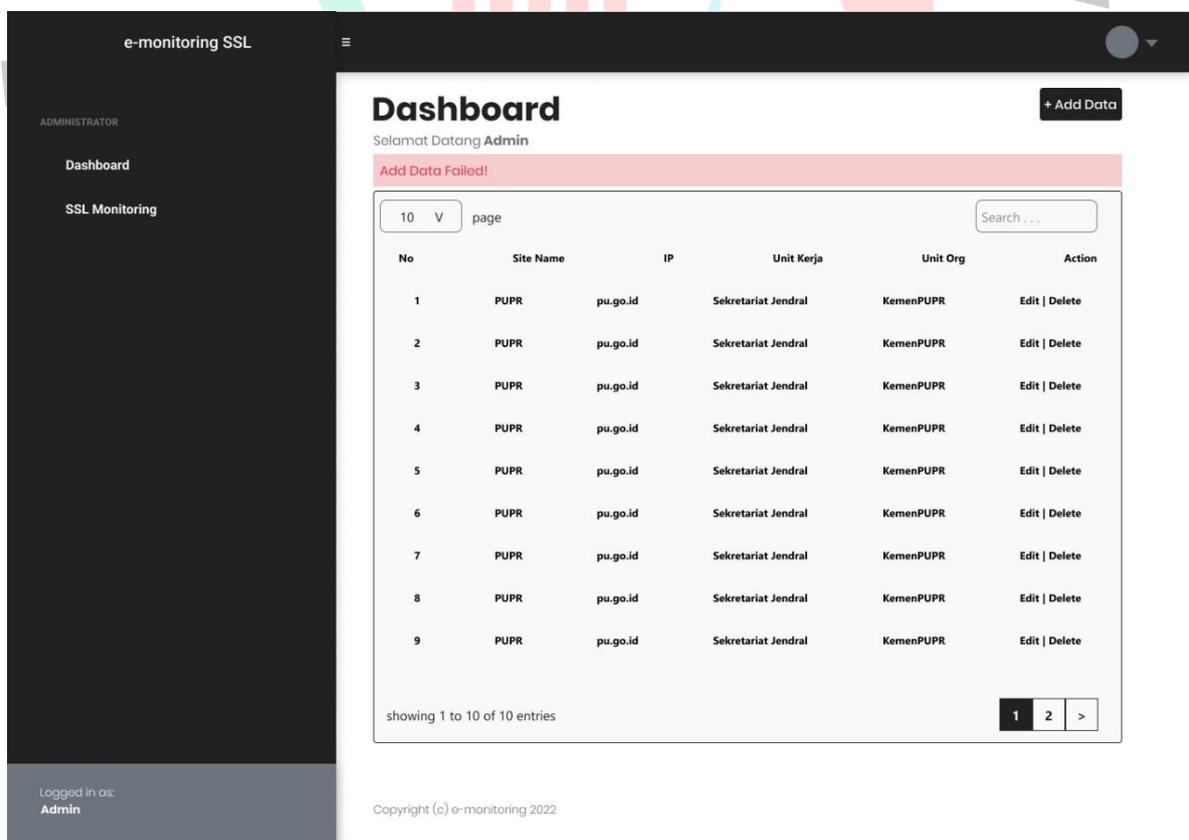
Gambar 3.11 *User Interface Tambah Data*
Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.11**, sebelum tampilan ini muncul pengguna harus menekan *button add data* yang terdapat pada bagian *content* tepatnya di atas tabel. Setelah ditekan maka akan muncul sebuah *modal* yang berisikan sebuah *form* tambah data, dari *modal* tersebut memiliki beberapa komponen yaitu tulisan tambah data, *field input data*, *button cancel* dan *button save/submit*.

Praktikan juga membuat *interface* suatu kondisi jika tambah data berhasil dan gagal. Jika tambah data berhasil maka akan muncul *alert* yang berwarna hijau seperti pada **Gambar 3.12** dan jika tambah data gagal maka akan muncul *alert* berwarna merah seperti pada **Gambar 3.13**.



Gambar 3.12 User Interface Tambah Data Berhasil
Sumber: Hasil Dokumentasi Praktikan



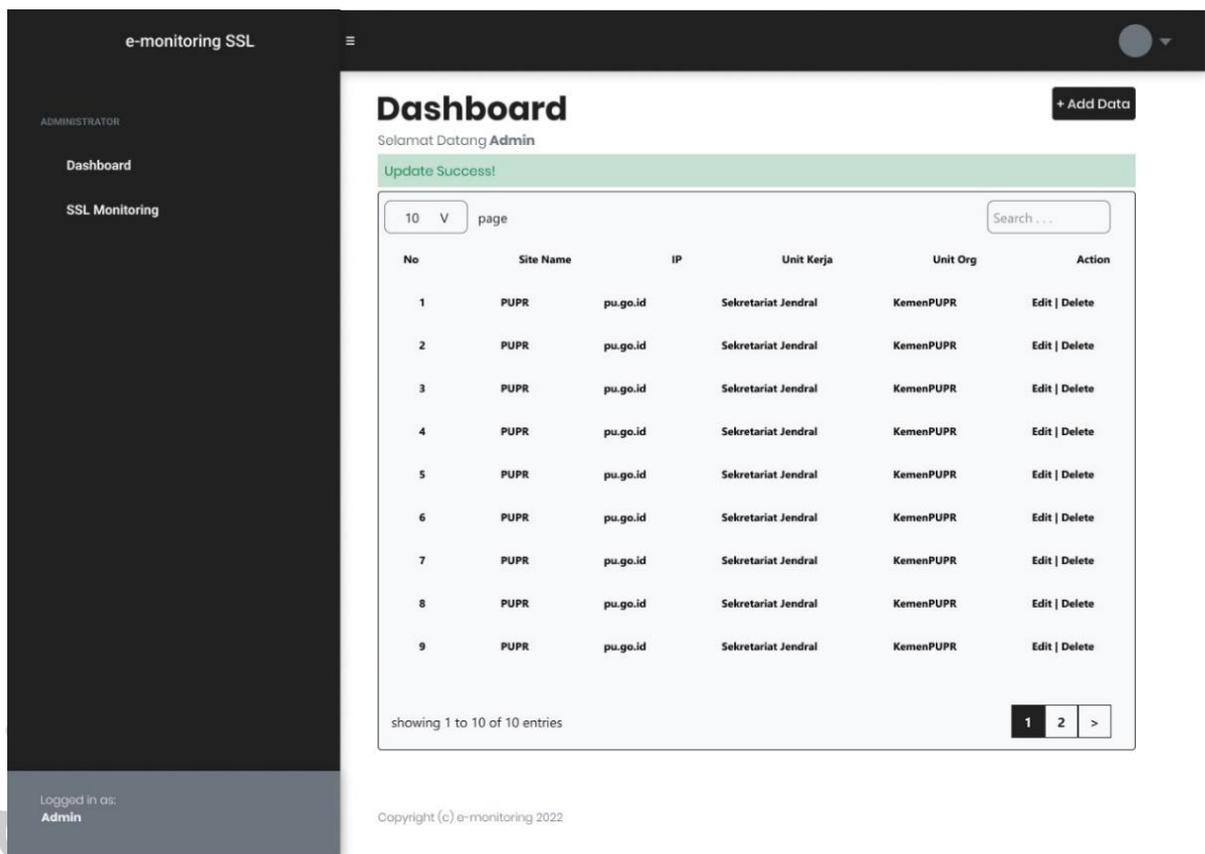
Gambar 3.13 User Interface Tambah Data Gagal
Sumber: Hasil Dokumentasi Praktikan

4. Halaman *Update Data*

Jika pengguna ingin melakukan ubah atau *update* data, maka pengguna harus menekan *button* edit pada tabel, yang mana akan muncul sebuah halaman seperti pada **Gambar 3.14**. Halaman ini memiliki beberapa komponen yang kesamaan terhdap *form* tambah data. Komponen yang terdapat pada form antara lain field ID website, site name, unit organisasi, unit kerja dan IP Adress. Akan tetapi yang membedakan adalah sebuah *button back* berwarna merah dan *button reset* yang berwarna kuning. Langkah selanjutnya Praktikan membuat *interface* terkait kondisi jika *update* data terjadi kegagalan atau berhasil. Berikut *user interface update* data berhasil pada seperti **Gambar 3.15** dan *Update* data gagal pada seperti **Gambar 3.16**. Jika *update* berhasil maka terdapat sebuah *alert* atau pemberitahuan di atas tabel berwarna hijau yang menandakan sukses.

The screenshot displays the 'Edit Page' interface within the 'e-monitoring SSL' application. On the left, a dark sidebar contains the user role 'ADMINISTRATOR' and navigation links for 'Dashboard' and 'SSL Monitoring'. The main content area is titled 'Edit Page' and contains a form with the following fields: 'ID Website', 'Site Name', 'Unit Organisasi', 'Unit Kerja', and 'IP'. A red 'back' button is positioned at the top left of the form, and yellow 'reset' and green 'submit' buttons are located at the bottom right. The footer of the page indicates the user is logged in as 'Admin' and includes the copyright notice 'Copyright (c) e-monitoring 2022'.

Gambar 3.14 User Interface Update Data
Sumber: Hasil Dokumentasi Praktikan



Gambar 3.15 User Interface Update Data Berhasil
 Sumber: Hasil Dokumentasi Praktikan

Sebuah *alert* atau pemberitahuan yang berwarna merah yang bertuliskan “*update failed*” di atas tabel menandakan bahwa *update* suatu data tidak berhasil. Tabel akan menampilkan data dari atas atau *ascending*, di dalam tabel terdapat kolom nomor, nama *website*, *ip address*, unit kerja, unit organisasi dan *action*. Pada baris *action* akan dibuat sebuah tombol *edit* dan *delete* yang berfungsi proses CRUD. *Alert* tersebut menjadi salah satu opsi untuk memberitahukan ada sebuah proses yang tidak berjalan semestinya seperti data yang sudah ada di database. *Alert* tersebut jika pengguna tidak mengisi dengan tidak lengkap, dan koneksi yang lambat. Sehingga *user* atau pengguna dapat mengetahui kesalahan perihal data yang tidak *terupdate* dan dapat memberitahukan kepada pembuat website atau orang yang bertanggung jawab atas hal itu.

The screenshot shows the 'e-monitoring SSL' dashboard. The header includes the title 'e-monitoring SSL' and a user profile icon. The left sidebar lists 'ADMINISTRATOR', 'Dashboard', and 'SSL Monitoring'. The main content area is titled 'Dashboard' and 'Selamat Datang Admin'. A red banner at the top of the main content area displays the message 'Update Failed!'. Below this, there is a table with 10 rows of data. The table has columns for 'No', 'Site Name', 'IP', 'Unit Kerja', 'Unit Org', and 'Action'. The 'Action' column contains 'Edit | Delete' for each row. At the bottom of the table, it says 'showing 1 to 10 of 10 entries'. A pagination control shows '1 2 >'. In the bottom left corner, it says 'Logged in as: Admin'. In the bottom right corner, it says 'Copyright (c) e-monitoring 2022'.

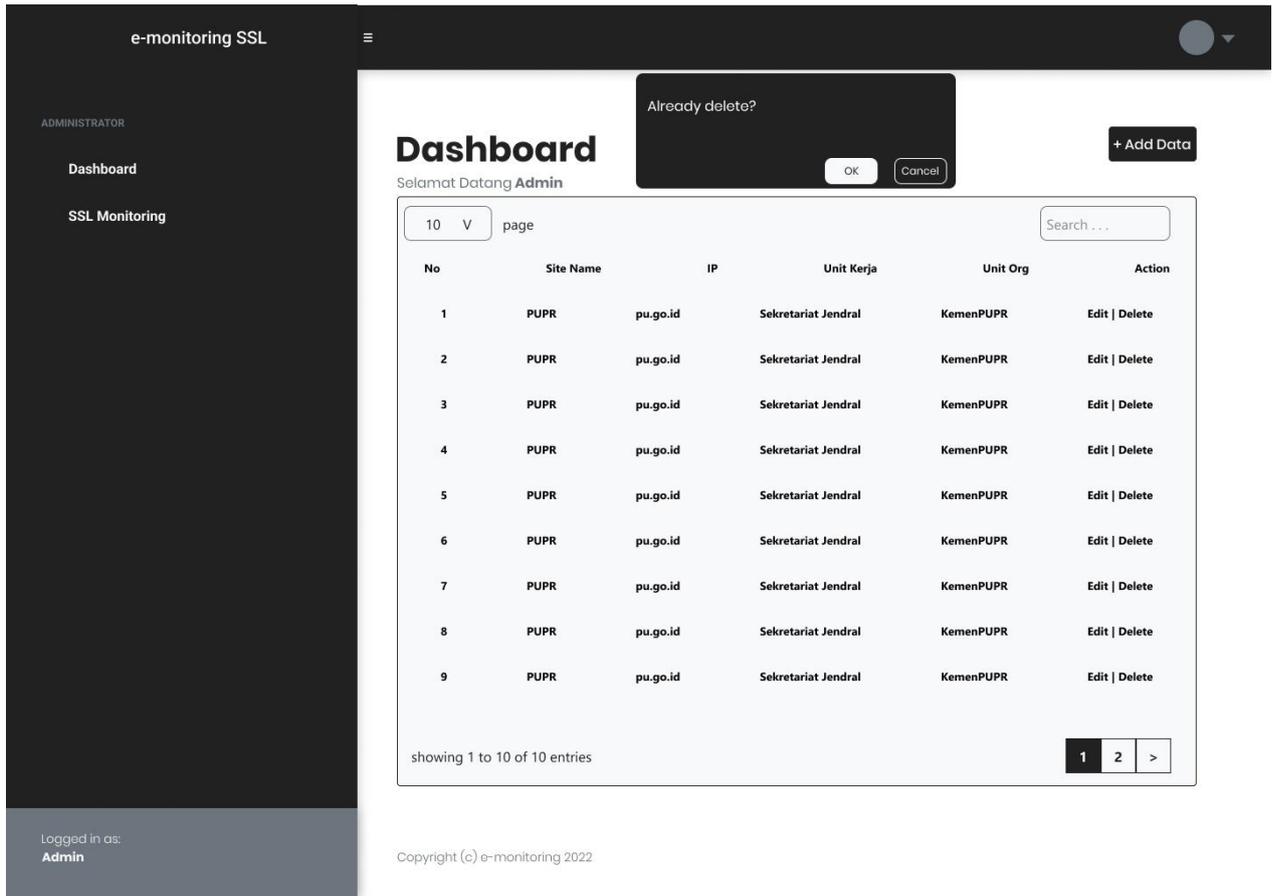
| No | Site Name | IP | Unit Kerja | Unit Org | Action |
|----|-----------|----------|---------------------|-----------|---------------|
| 1 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 2 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 3 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 4 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 5 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 6 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 7 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 8 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |
| 9 | PUPR | pu.go.id | Sekretariat Jendral | KemenPUPR | Edit Delete |

Gambar 3.16 User Interface Update Data Gagal

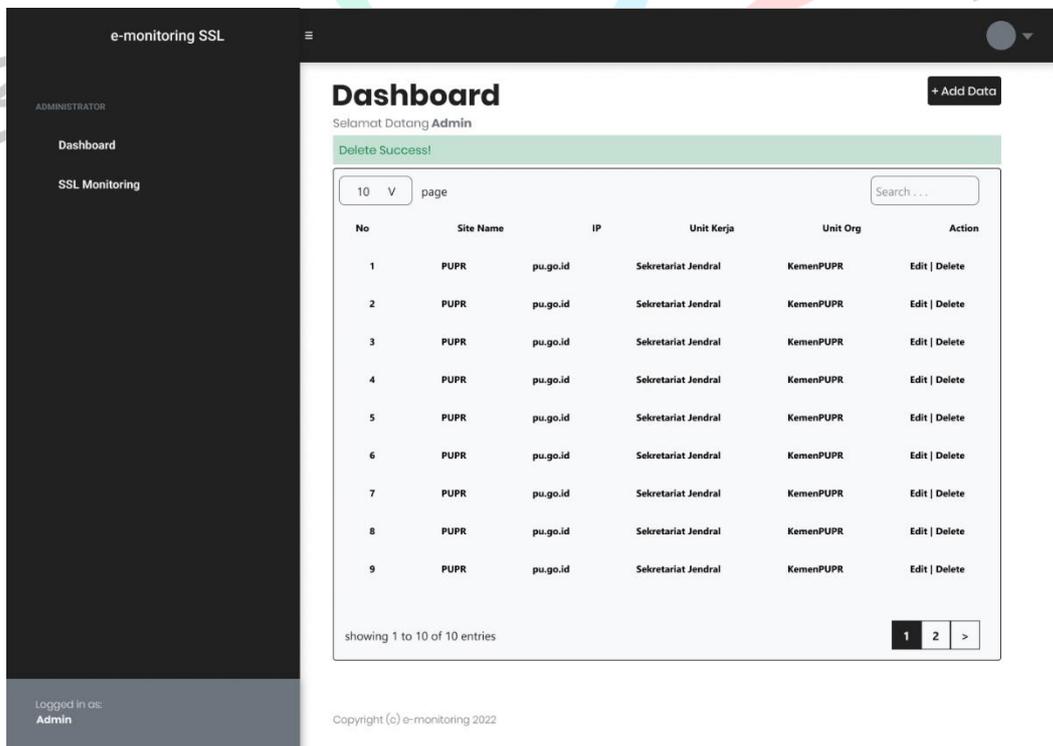
Sumber: Hasil Dokumentasi Praktikan

5. Halaman *Delete Data*

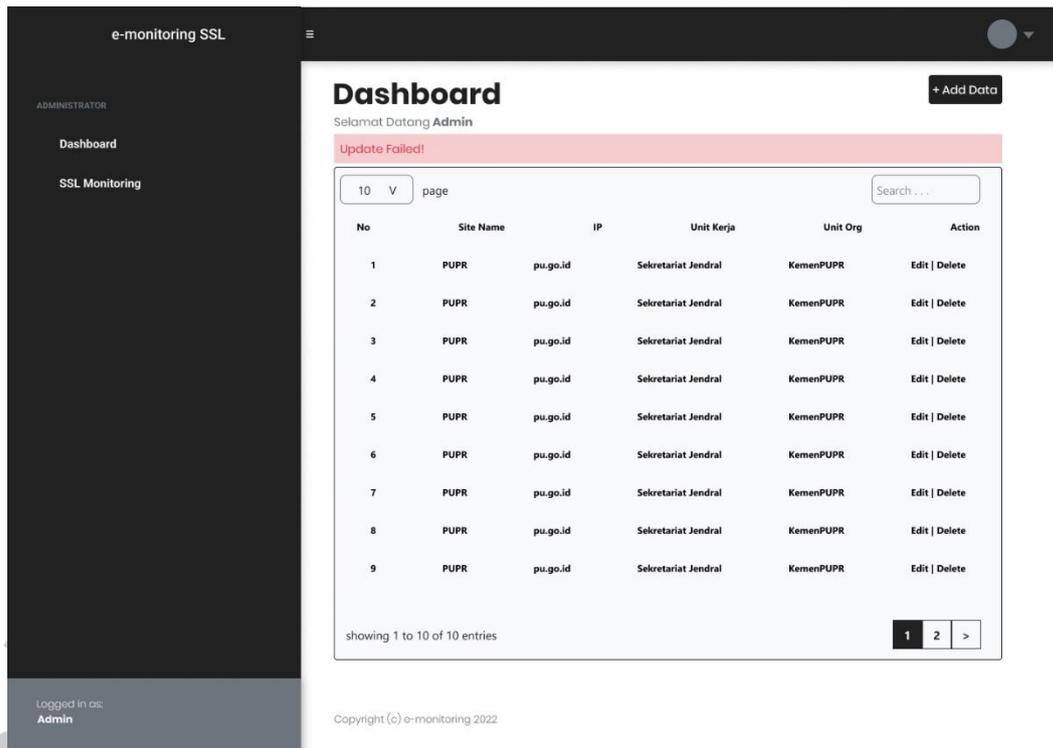
Pada **Gambar 3.18** adalah *user interface* delete yang di dalamnya terdapat modal berwarna hitam jika pengguna menekan *button delete* pada tabel. Maka modal akan muncul dan berisikan pertanyaan pilihan kepada pengguna serta terdapat 2 (dua) *button* yaitu *button cancel* dengan *background* transparan tulisan berwarna putih dan *button OK* berwarna putih. Setelah menentukan pilihan maka akan muncul sebuah *alert* sebagai kondisi apakah berhasil atau gagal. Jika berhasil menghapus data maka hasilnya akan tampilan sebuah pemberitahuan berwarna hijau sebagai data berhasil dihapus seperti pada **Gambar 3.17** dan jika gagal, maka akan menampilkan pemberitahuan berwarna merah yang menyatakan bahwa data gagal dihapus seperti pada **Gambar 3.19**.



Gambar 3.18 User Interface Delete Data
 Sumber: Hasil Dokumentasi Praktikan



Gambar 3.17 User Interface Delete Data Berhasil
 Sumber: Hasil Dokumentasi Praktikan



Gambar 3.19 User Interface Delete Data Gagal
 Sumber: Hasil Dokumentasi Praktikan

6. Halaman Monitoring SSL

Pada **Gambar 3.20** *page* ini merupakan monitoring SSL yang terdapat sebuah tabel pada bagian *content*, kemudian pada tiap tabel terdapat status dan *day left*. Kolom status berisikan informasi apakah website tersebut apakah masih hidup atau sudah mati (expired), yang mana menjadi penanda bahwa jika website tersebut sudah mati (expired) maka diharuskan segera dihidupkan kembali.

Dan kolom *day left* merupakan informasi tentang *countdown* website SSL akan segera habis, yang mana jika sudah habis maka website tersebut tidak aman dan rawan dari serangan siber.

e-monitoring SSL

ADMINISTRATOR

Dashboard

SSL Monitoring

SSL Monitoring

Selamat Datang Admin

10 V page

| No | Site Name | IP | Status | Day Left |
|----|-----------|----------|-----------------------------|------------------------------------|
| 1 | PUPR | pu.go.id | SSL already active! | Certificate will expire in 38 days |
| 2 | PUPR | pu.go.id | SSL already die or expired! | No SSL certificates were found! |
| 3 | PUPR | pu.go.id | SSL already active! | Certificate will expire in 38 days |

showing 1 to 10 of 10 entries

1 2 >

Logged in as: Admin

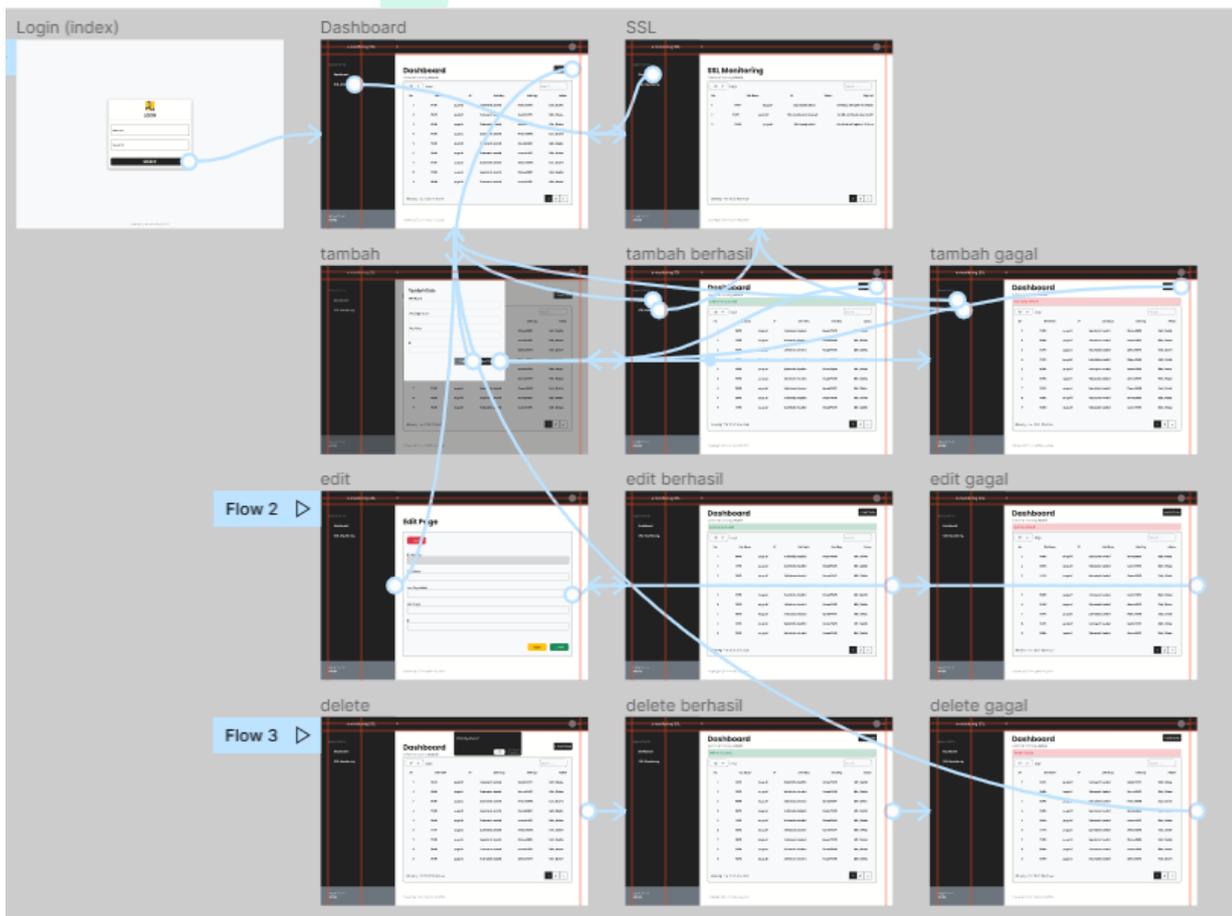
Copyright (c) e-monitoring 2022

Gambar 3.20 User Interface Monitoring SSL
 Sumber: Hasil Dokumentasi Praktikan

3.2.4 Prototyping Desain UI/UX Aplikasi

Setelah membuat semua desain UI/UX aplikasi *e-monitoring* SSL, langkah selanjutnya adalah membuat *prototyping* aplikasi pada *software* figma. *Prototyping* merupakan teknik agar organisasi atau programmer mendapatkan gambaran tentang bagaimana fitur atau produk yang berfungsi tanpa harus menghabiskan waktu dan uang untuk menjelaskan secara detail. *Prototyping* juga memfasilitasi untuk pengguna, dengan melakukan uji coba agar dapat menemukan masalah atau kekurangan dalam desain yang mungkin tidak ditemukan oleh UI/UX (Knight, 2019). Adanya *prototyping* dapat membantu mengurangi kesalahan dalam implementasi yang akan dirancang oleh frontend. Figma sebagai tools membuat wireframe dan mockup terdapat fitur *prototyping*, sehingga figma menjadi tools andalan bagi UI/UX.

Awal alur aplikasi pada *prototyping* desain UI/UX ini dimulai dari halaman login. Pengguna yang dapat masuk ke halaman *dashboard* diharuskan orang/karyawan dari PUSDATIN atau orang memiliki *username* dan *password* yang sudah ditentukan. Jika sudah masuk ke halaman *dashboard* maka terdapat tabel dan beberapa *button* seperti tambah data, edit data dan *delete* data serta menu pada sidebar halaman. Pada bagian sidebar memiliki halaman lain yaitu monitoring SSL. Bagian *button* tambah data hanya menampilkan modal yang berisikan form tambah data, kemudian bagian *button* edit menampilkan halaman form edit data dan *button delete* hanya menghapus data. Berikut gambaran *prototyping* yang praktikan buat menggunakan figma pada **Gambar 3.21** terdapat garis yang menunjukkan alur interaksi pengguna dengan sistem, garis tersebut merupakan penghubung antara halaman ke halaman yang lain.



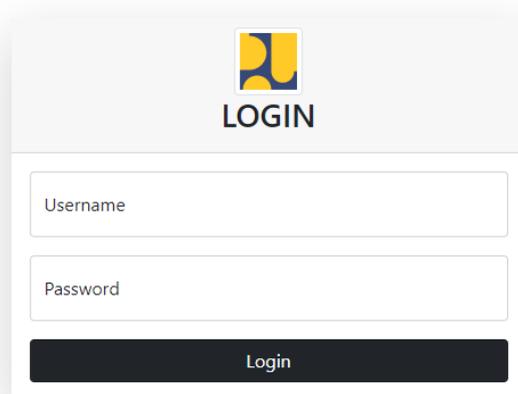
Gambar 3.21 Prototyping Desain UI/UX Menggunakan Figma
 Sumber: Hasil Dokumentasi Praktikan

3.2.5 Implementasi Desain

Setelah melakukan desain UI/UX dan *prototyping*. Praktikan melanjutkan dengan mengimplementasikan hasil desain yang sudah dibuat ke sebuah program. Dalam proses pengembangannya Praktikan menggunakan software Visual Studio Code sebagai *code editor* untuk membantu merancang aplikasi tersebut.

Setelah mendapatkan hasil desain UI/UX aplikasi, kemudian sebagai *frontend development* dilakukan proses implementasi terhadap hasil desain tersebut menjadi tampilan web yang sudah ditentukan pada perencanaan dengan menggunakan HTML, CSS, *Javascript*, *bootstrap* dan *JQuery*. Berikut merupakan hasil terkait desain yang sudah diimplementasi serta penjelasannya:

a. Halaman *login*

The image shows a login form with a light gray background. At the top center is a logo consisting of two overlapping squares, one yellow and one blue. Below the logo is the word "LOGIN" in bold, black, uppercase letters. There are two input fields: the first is labeled "Username" and the second is labeled "Password". Below these fields is a dark gray button with the word "Login" in white text.

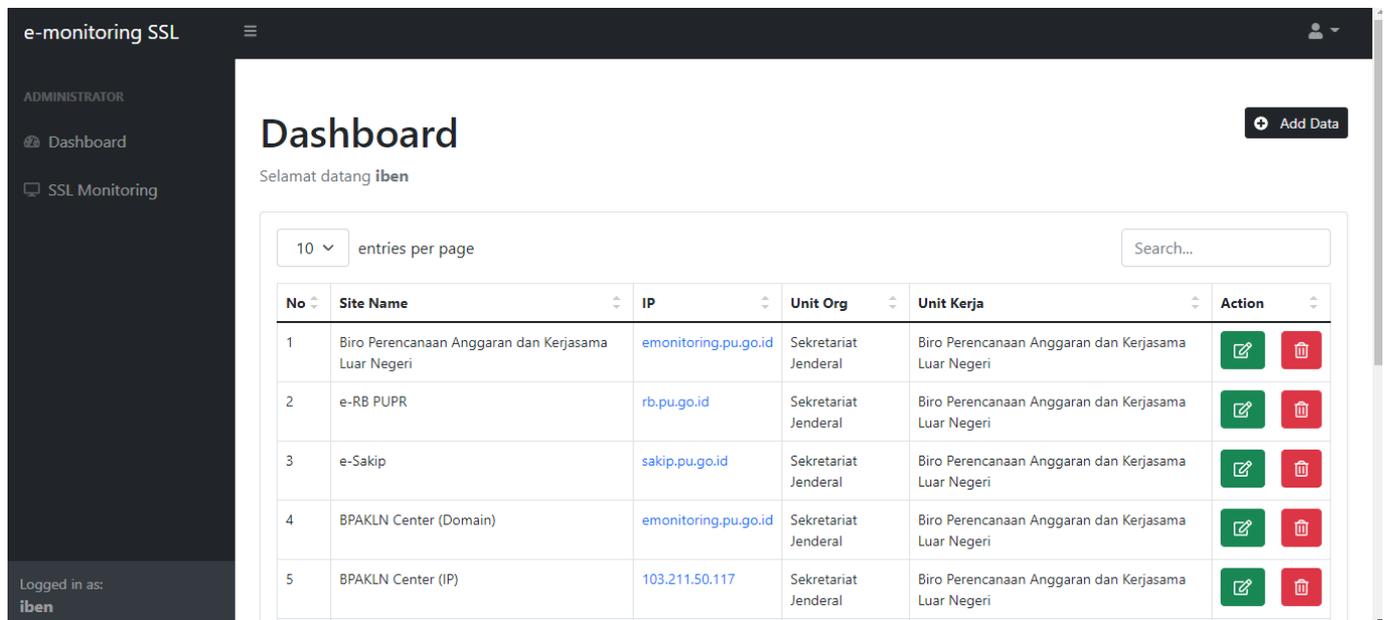
Gambar 3.22 Implementasi Halaman *Login*

Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.22**. form login ini menggunakan CCS bootstrap dengan *class* antara lain di bagian *card(background)*, *field input* dan *button login*. Pengguna diharuskan untuk mengisi form login dengan *username* dan *password* yang sudah ditentukan oleh pihak PUSDATIN. Setelah pengguna melakukan pengisian data pada *field* dapat dilanjutkan dengan menekan tombol *login*. Jika berhasil maka akan berpindah halaman ke dashboard dan jika gagal dengan *username* atau *password* salah maka akan tetap di

halaman login dengan *alert* bahwa gagal untuk melakukan login. Kemudian bagian bawah halaman terdapat tulisan hak cipta menggunakan *tag* HTML yang bernama `<footer>`.

b. Halaman *Dashboard*

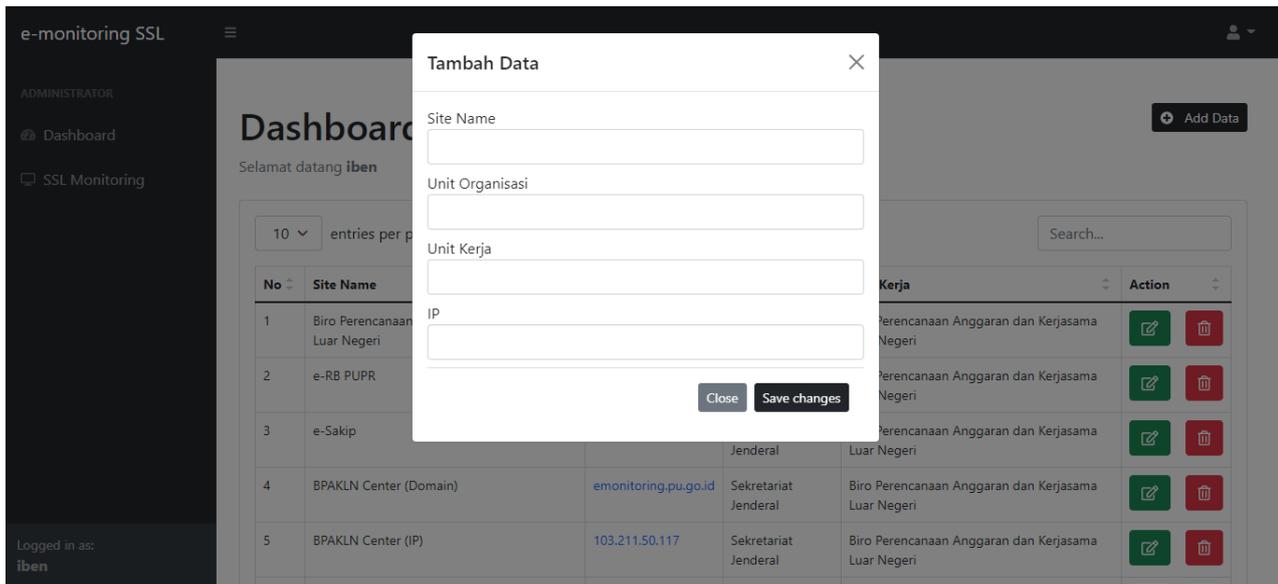


Gambar 3.23 Implementasi Halaman Dashboard

Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.23**. Setelah pengguna melakukan login, pengguna akan dialihkan ke halaman dashboard. Pada halaman dashboard berisikan 3 (tiga) bagian *topbar*, *sidebar* dan *content* yang sudah dijelaskan pada UI/UX. Implementasi pada halaman ini dimulai dari bagian *topbar* yang menggunakan *class* bootstrap yang berisikan *avatar profile* yang berfungsi sebagai *button logout* pengguna dan *toggle bar* untuk menutup bagian *sidebar*, kemudian pada bagian *sidebar* terdapat *class nav-bar* yang berisikan *Dashboard* dan *Monitoring SSL* masing – masing menu tersebut memiliki halaman sendiri. Di bagian *content* terdapat *table* yang berisikan data dari database, di atas *table* memiliki *button add* yang di bawahnya *input search* dan *entries per page*, lalu *table* terdapat 2 (dua) *button* yaitu *button edit* dan *button delete*. *Table* ini mempunyai fitur seperti *sort*, *filter* dan *pagination*. Fitur tersebut menggunakan *jQuery* agar fitur berjalan dengan baik.

c. Halaman Tambah Data

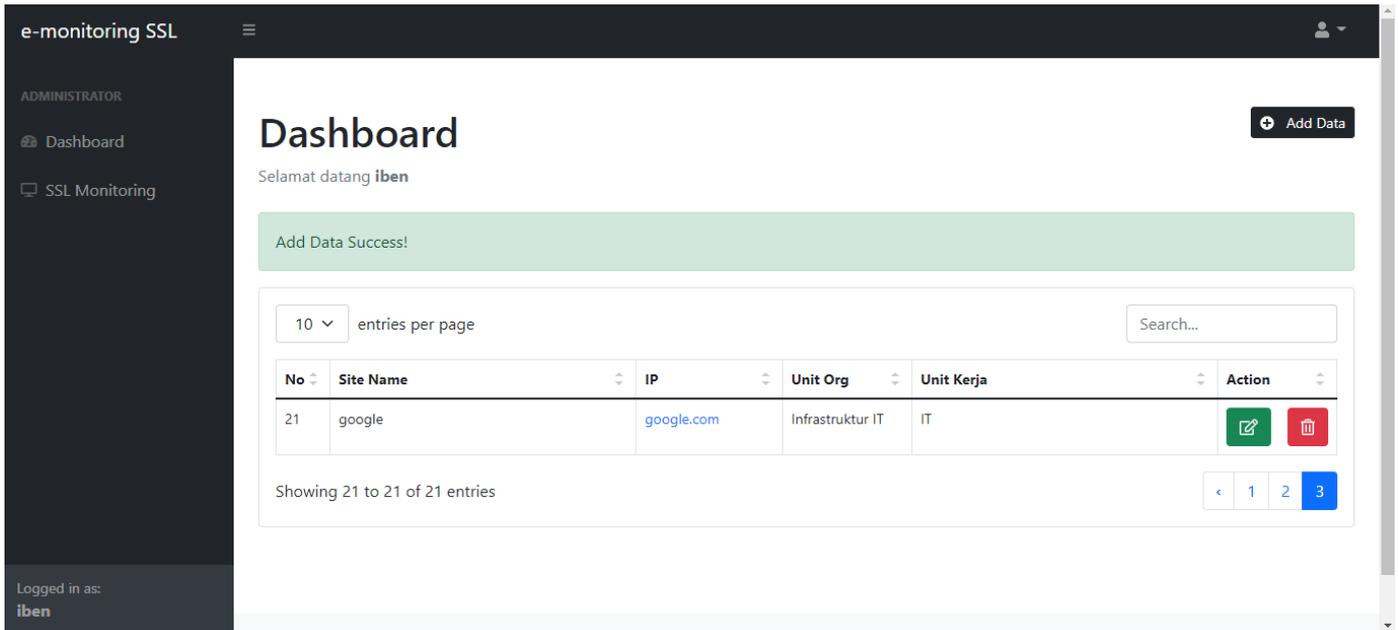


Gambar 3.24 Impelentasi Halaman Tambah Data
Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.24** Tampilan ini merupakan tambah data jika pengguna sudah menekan *button add data* pada *dashboard*, sehingga menampilkan sebuah modal di dalamnya terdapat form yang berisikan site name, unit organisasi, unit kerja dan ip. Jika pengguna sudah mengisi form tersebut maka akan menekan tombol *Save Changes* dan jika pengguna tidak jadi untuk mengisi form maka dapat menekan *button cancel* atau mengklik area diluar modal.

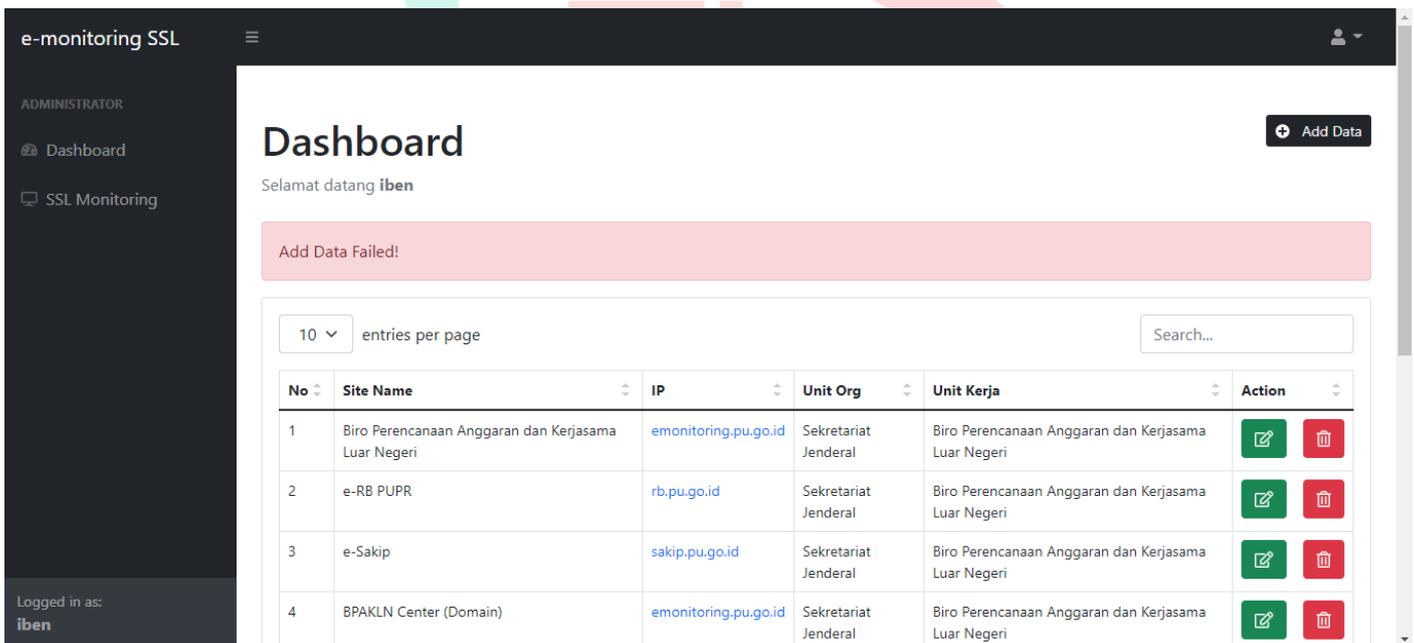
d. Halaman Tambah Data Berhasil

Pada **Gambar 3.25**. Halaman ini pengguna sudah melakukan tambah data dan berhasil, maka sistem akan memunculkan sebuah *alert* dengan tulisan "*Add Data Success!*" yang berwarna hijau. Alert tersebut berasal dari components bootstrap. Data yang sudah disimpan akan muncul pada halaman *dashboard*.



Gambar 3.25 Implementasi Tambah Data Berhasil
Sumber: Hasil Dokumentasi Praktikan

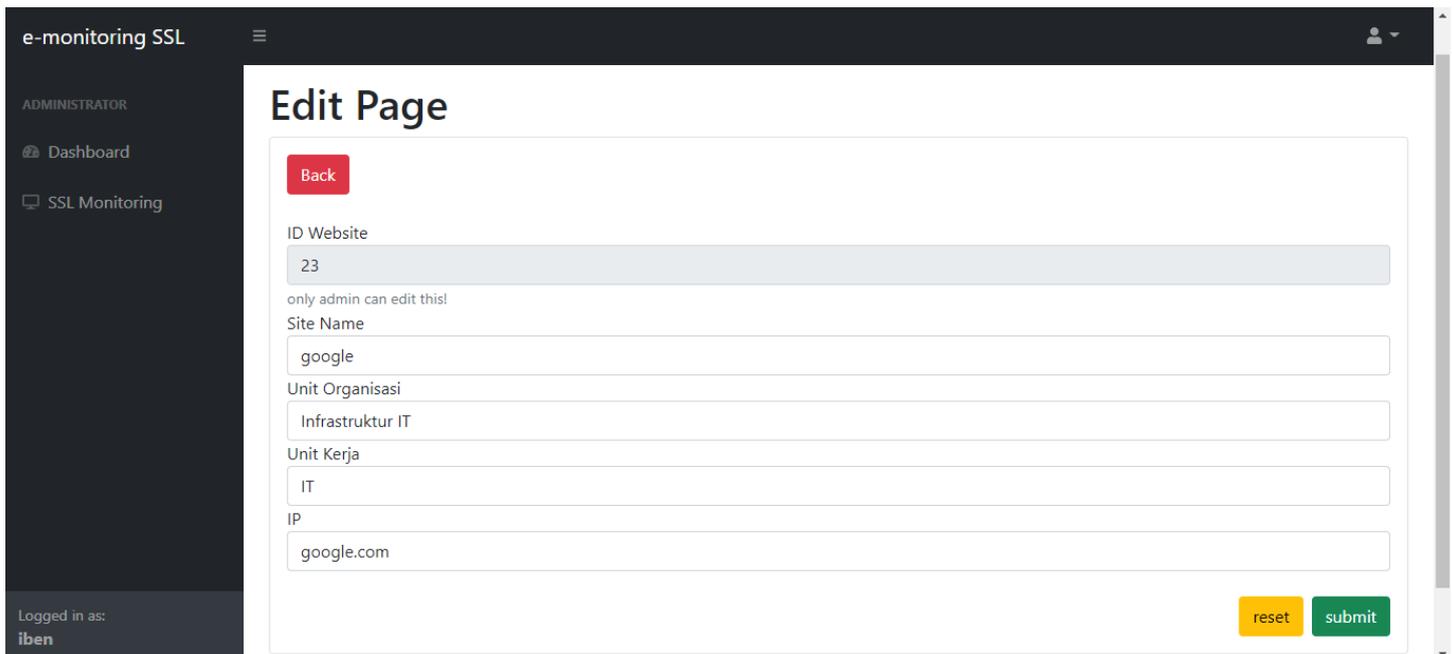
e. Halaman Tambah Data Gagal



Gambar 3.26 Implementasi Tambah Data Gagal
Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.26**. Jika pengguna gagal dalam menginput data, maka akan dikembalikan ke halaman *dashboard* dan akan muncul sebuah *alert* dengan tulisan “*Add Data Failed*” yang berwarna merah.

f. Halaman Edit Data



The screenshot shows the 'Edit Page' interface of the 'e-monitoring SSL' application. The left sidebar contains navigation links for 'ADMINISTRATOR', 'Dashboard', and 'SSL Monitoring'. The main content area is titled 'Edit Page' and contains a form with the following fields and buttons:

- Back**: A red button at the top left of the form.
- ID Website**: A text input field containing the value '23'.
- Site Name**: A text input field containing the value 'google'.
- Unit Organisasi**: A text input field containing the value 'Infrastruktur IT'.
- Unit Kerja**: A text input field containing the value 'IT'.
- IP**: A text input field containing the value 'google.com'.
- reset**: A yellow button at the bottom right of the form.
- submit**: A green button at the bottom right of the form.

Below the form, there is a note: 'only admin can edit this!'.

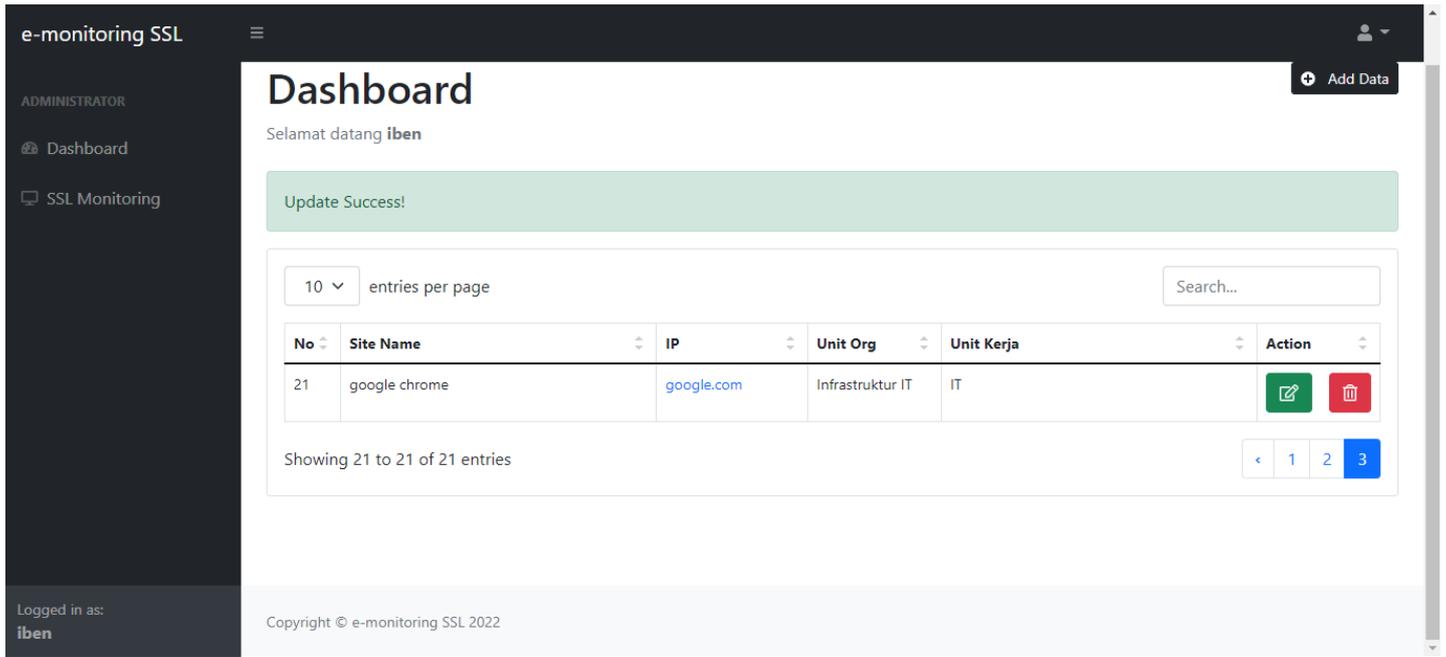
Gambar 3.27 Implementasi Halaman Edit Data

Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.27**. Sebelum masuk ke halaman edit, pengguna harus menekan tombol edit pada tabel yang ada di *dashboard*. Setelah menekan maka akan berpindah ke halaman edit dan pengguna dapat mengedit atau *update* data tersebut. Bagian bawah form terdapat 2 (dua) *button* yaitu *button* reset dan submit. Fungsi pada *button* reset adalah mereset kembali data yang sudah diubah dan fungsi pada *button* submit adalah mensubmit data ke database.

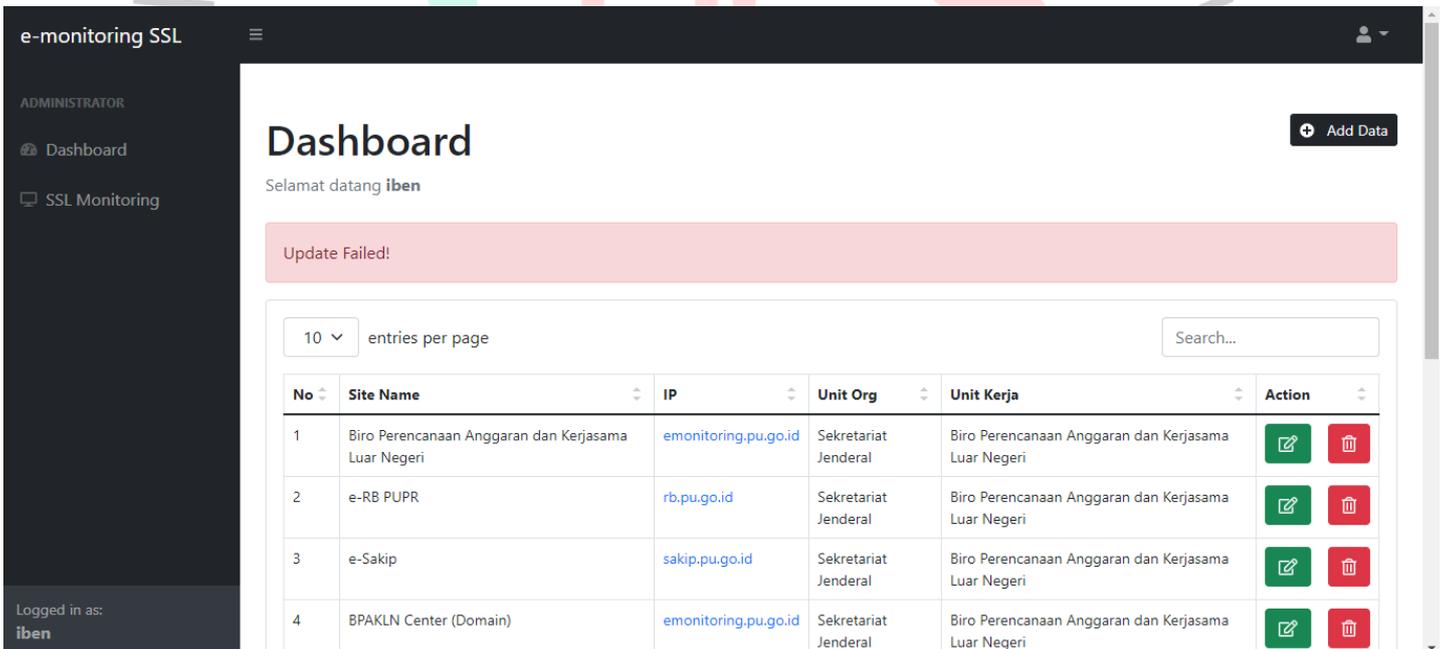
g. Halaman Edit Data Berhasil

Pada **Gambar 3.28** merupakan hasil setelah pengguna melakukan *submit* data yang sudah ubah dan kembali ke halaman *dashboard*, maka akan muncul sebuah *alert* yang menandakan bahwa data tersebut sudah *terupdate*. *Alert* tersebut bertuliskan "*Update Success!*" yang berwarna hijau diatas tabel data.



Gambar 3.28 Implementasi Halaman Update Data Berhasil
 Sumber: Hasil Dokumentasi Praktikan

h. Halaman Edit Data Gagal

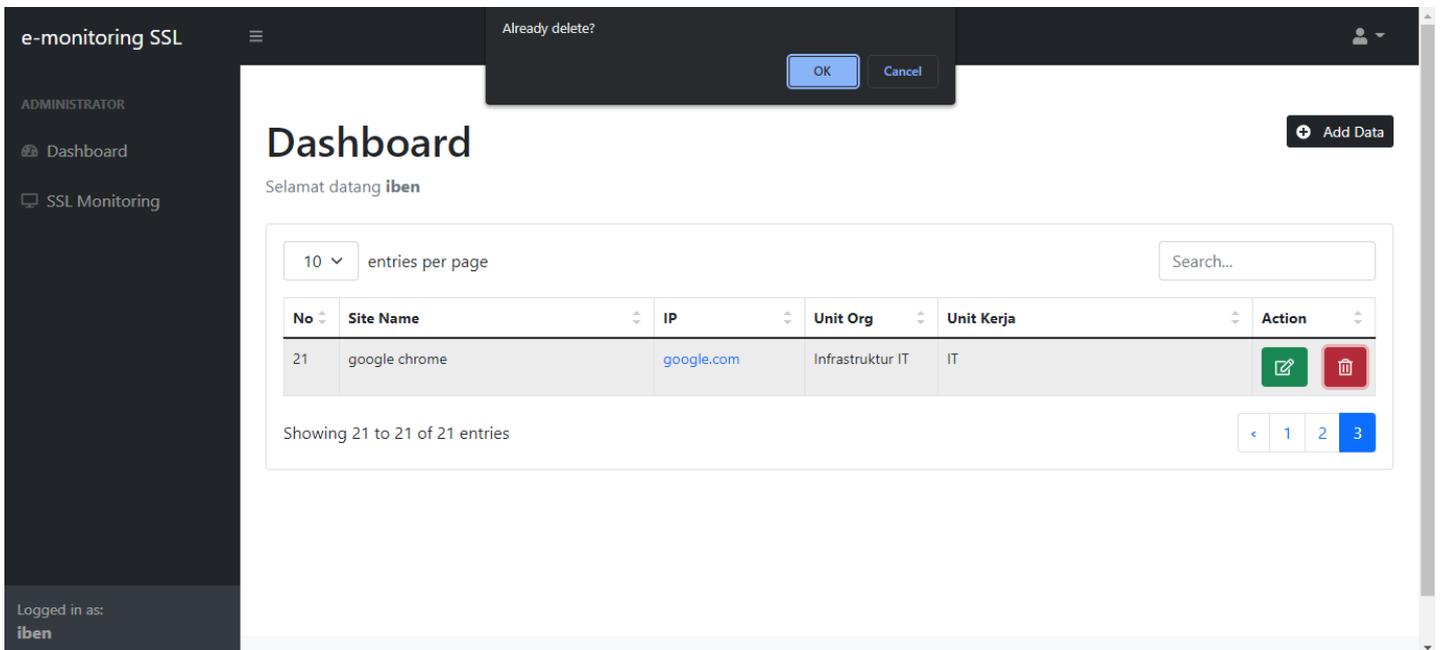


Gambar 3.29 Implementasi Halaman Update Data Gagal
 Sumber: Hasil Dokumentasi Praktikan

Pada. Halaman ini merupakan proses gagal yang terjadi dari *update* data. Jika terjadi kesalahan dari sisi *user* ataupun sistem, maka setelah pengguna mensubmit data yang sudah di*update* hasilnya akan gagal. Pengguna akan kembali ke halaman

dashboard yang kemudian muncul sebuah *alert* bertuliskan “Update Failed!” dengan warna merah.

i. Halaman *Delete*



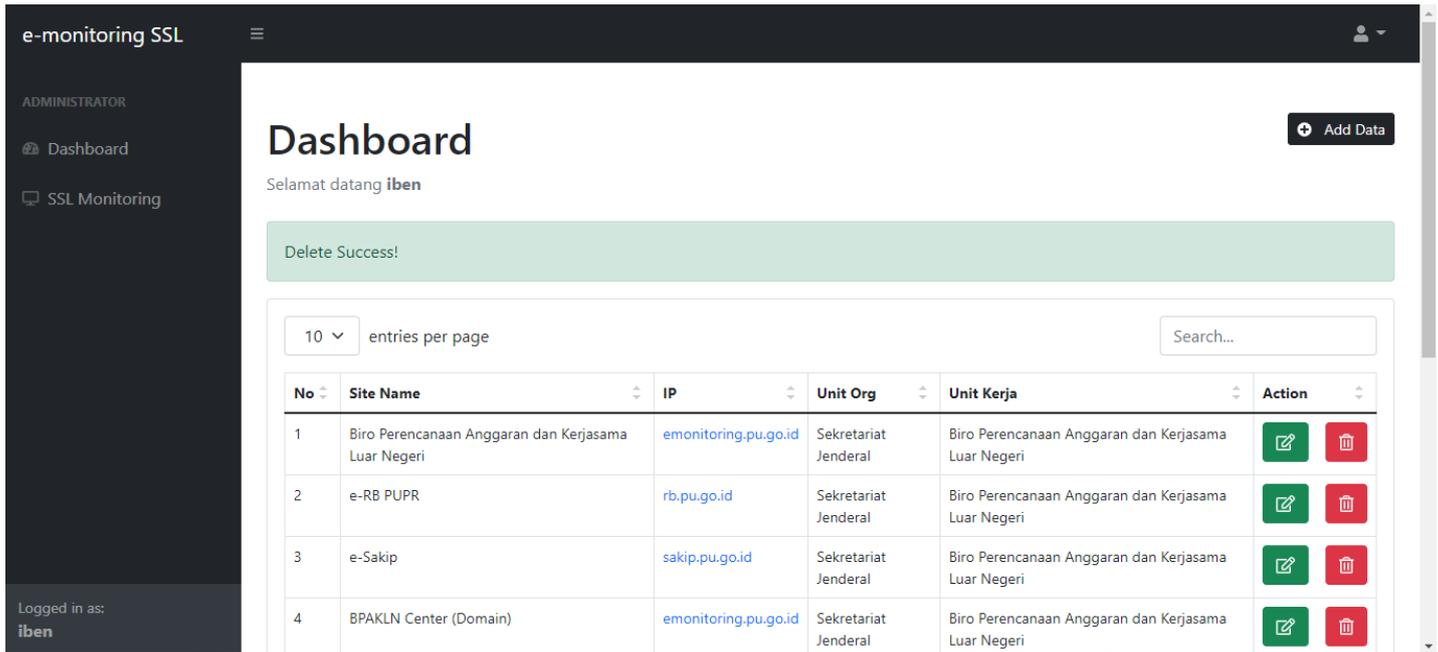
Gambar 3.30 Implementasi Halaman *Delete* Data

Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.30**. Saat pengguna menekan *button delete* pada tabel kolom action, maka sistem akan merespon dan memunculkan sebuah pesan atau dialog menggunakan *javascript*. Pengguna akan memilih, jika OK data akan terhapus dan cancel data tidak jadi terhapus.

j. Halaman *Delete* Berhasil

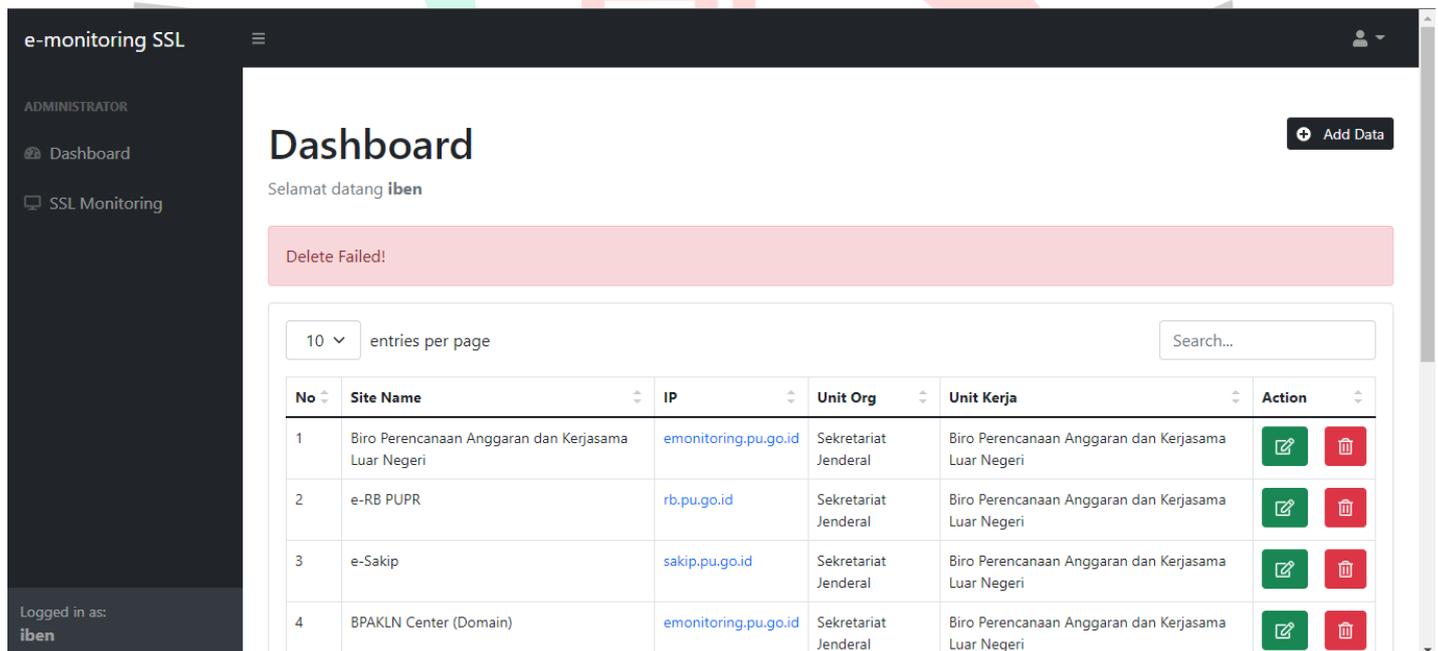
Pada **Gambar 3.31**. merupakan hasil jika pengguna menekan tombol OK pada dialog menghapus data. Pengguna juga akan kembali ke halaman *dashboard* dengan *alert* berwarna hijau bahwa data sudah terhapus dan bertuliskan “Delete Success!”.



Gambar 3.31 Implementasi Halaman Delete Data Berhasil

Sumber: Hasil Dokumentasi Praktikan

k. Halaman Delete Gagal

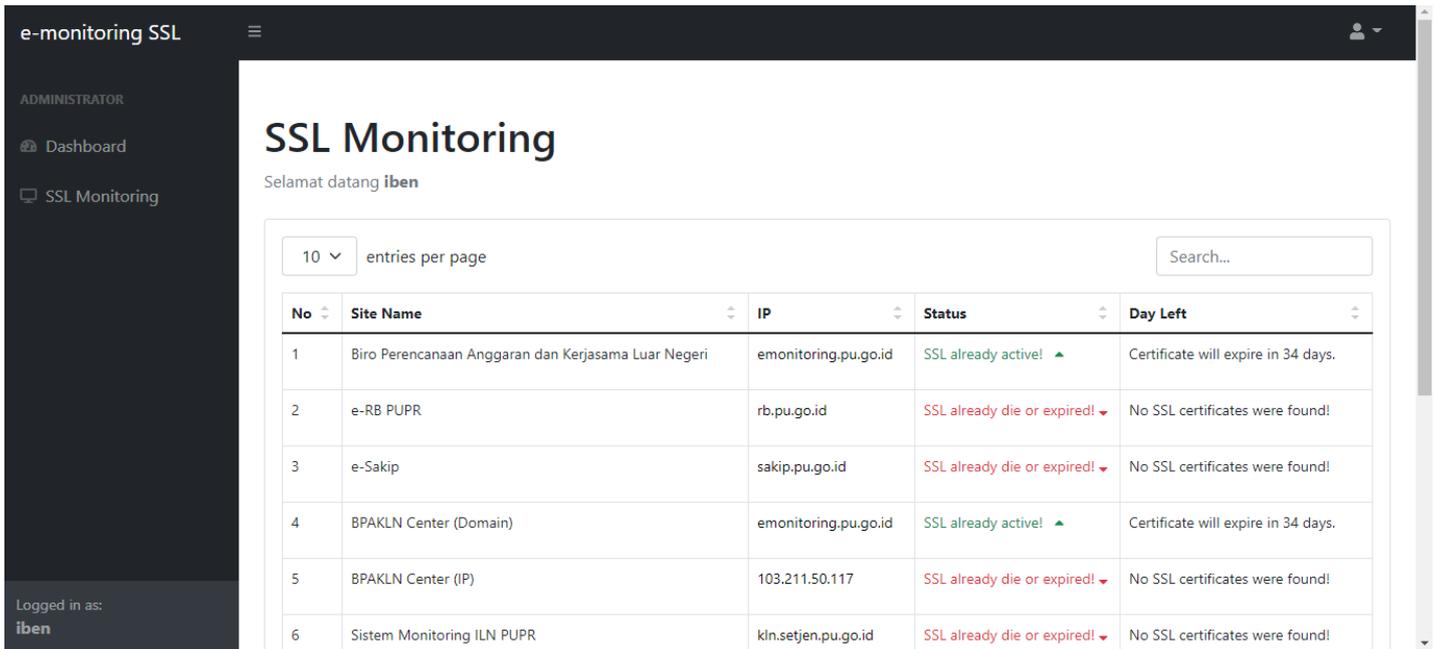


Gambar 3.32 Implementasi Halaman Delete Data Gagal

Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.32** Jika terjadi kesalahan pada sistem atau user maka proses penghapusan data akan berakhir dengan *alert* berwarna merah yang bertulis "*Delete Failed!*".

I. Halaman Monitoring SSL



The screenshot shows a web application interface for monitoring SSL certificates. The header includes the title 'e-monitoring SSL' and a user profile icon. A sidebar on the left contains navigation links for 'ADMINISTRATOR', 'Dashboard', and 'SSL Monitoring'. The main content area is titled 'SSL Monitoring' and includes a greeting 'Selamat datang iben'. Below the title, there is a search bar and a dropdown menu for 'entries per page' set to '10'. The main data is presented in a table with the following columns: No, Site Name, IP, Status, and Day Left.

| No | Site Name | IP | Status | Day Left |
|----|---|----------------------|-------------------------------|-------------------------------------|
| 1 | Biro Perencanaan Anggaran dan Kerjasama Luar Negeri | emonitoring.pu.go.id | SSL already active! ▲ | Certificate will expire in 34 days. |
| 2 | e-RB PUPR | rb.pu.go.id | SSL already die or expired! ▼ | No SSL certificates were found! |
| 3 | e-Sakip | sakip.pu.go.id | SSL already die or expired! ▼ | No SSL certificates were found! |
| 4 | BPAKLN Center (Domain) | emonitoring.pu.go.id | SSL already active! ▲ | Certificate will expire in 34 days. |
| 5 | BPAKLN Center (IP) | 103.211.50.117 | SSL already die or expired! ▼ | No SSL certificates were found! |
| 6 | Sistem Monitoring ILN PUPR | kln.setjen.pu.go.id | SSL already die or expired! ▼ | No SSL certificates were found! |

Gambar 3.33 Implementasi Halaman Monitoring SSL

Sumber: Hasil Dokumentasi Praktikan

Pada **Gambar 3.33**. Merupakan *page* monitoring SSL berisikan tabel yang sedikit berbeda dengan dashboard, yang mana terdapat *field table status* dan *dayleft*. Status berfungsi sebagai pengecekan SSL dan mempunyai kondisi jika SSL tersebut masih ada maka hasilnya akan bertuliskan “*SSL already active!*” yang berwarna hijau. Hijau yang dimaksud menandakan bahwa status website tersebut masih ada. Selanjutnya jika status bertuliskan “*SSL already die or expired!*” dan berwarna merah. Yang mana SSL website tersebut sudah mati atau sudah tidak berlaku kembali. Lalu pada bagian *dayleft* merupakan penanda untuk hitungan mundur sebuah SSL yang akan habis kadaluarsanya.

3.3 Kendala yang Dihadapi

Semasa melaksanakan kegiatan tersebut, Praktikan mendapati kendala saat menjalankan kerja profesi di PUSDATIN, yaitu:

1. Praktikan kurang beradaptasi menggunakan framework seperti Laravel, TailwindCSS dan lainnya.

3.4 Cara Mengatasi Kendala

Dari kendala tersebut Praktikan mencari cara untuk mengatasinya sebagaimana berikut:

1. Praktikan menggunakan php *native*, *javascript* dan bootstrap serta *library* lain sebagai pengganti *framework*, yang mana agar developer selanjutnya dapat mudah mengembangkan dan memahami sturuktur pratikan bangun.

3.5 Pembelajaran yang Diperoleh dari Kerja Profesi

Selama mengikuti kegiatan ini selama 3 (tiga) bulan yang di tempatkan Kementerian PUPR khususnya PUSDATIN, Praktikan mendapatkan beberapa pengalaman baru mengenai pentingnya kerjasama tim dan manajemen kegiatan dalam menyelesaikan aplikasi *e-monitoring* SSL. Kemudian Praktikan juga dapat mengimplementasikan skill dari merancang desain awal hingga implementasi frontend, sehingga Praktikan mendapatkan ilmu baru terutama pada bidang pekerjaan.

