**BAB III**

**METODOLOGI PENELITIAN**

1. **Data Penelitian**

Data penelitian yang akan digunakan dalam penelitian ini adalah data pegawai, data pelanggan, data *reseller* dan *file* pada CV. Trista Jaya Abadi.

1. **Metode Pembanding**

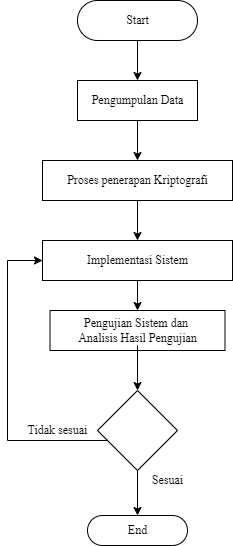
Dalam metode pembanding ini hubungan antara penelitian ini dengan penelitian terdahulu sebagai berikut:

Pada penelitian terdahulu yang bernama Andre Setiawan dan Titin Fatimah dengan judul “Implementasi Algoritma Kriptografi RC4 Untuk Keamanan Database Aplikasi Penggajian Karyawan Berbasis Web Pada PT. Trans Intra Asia” dimana penelitian terdahulu menggunakan metode RC4 untuk mengamankan *database* aplikasi penggajian karyawan yang ada di PT. Trans Intra Asia. Kelebihan aplikasi yang dibuat penelitian terdahulu yaitu Terdapat autentikasi *username* dan *password* pada *form login*. Program yang *user friendly*, karena memiliki tampilan yang sederhana dan jelas. Data perusahaan akan aman jika menggunakan aplikasi ini, karena aplikasi ini sudah menggunakan sistem keamanan *database* dan data yang dihasilkan di dalam *database* dalam bentuk enkripsi sehingga data sulit dicuri. Kekurangan aplikasi yang dibuat oleh penelitian terdahulu yaitu aplikasi ini hanya digunakan pada *database* yang sudah ditentukan oleh *user*. Tidak semua *field* (kolom) dapat dienkripsi.

Pada penelitian ini yang berjudul “Implementasi Algoritma Kriptografi AES-128 dan RC4 Untuk Pengamanan Database Dan File Berbasis Web Pada CV. Trista Jaya Abadi” dimana penelitian ini menggunakan metode AES-128 dan RC4 untuk mengamankan *database* dan menggunakan metode RC4 untuk mengamankan *file* di CV. Trista Jaya Abadi. Kelebihan aplikasi yang dibuat penelitian ini yaitu program aplikasi yang dibuat *user friendly*, karena memiliki tampilan yang sederhana dan jelas untuk digunakan oleh *user*. *Database* dan *file* yang dienkripsi tidak dapat dibaca oleh pihak yang tidak bertanggung jawab dan ukuran *file* yang di enkripsi dan di dekripsi tidak mengubah ukuran *file* asli. Kekurangan aplikasi yang dibuat penelitian ini yaitu aplikasi yang dibuat hanya mengenkripsi *file* yang berekstensi .docx, .pdf, .xlsx, dan .pptx serta semakin besar ukuran *file* yang di enkripsi dan di dekripsi, maka semakin lama prosesnya. apa hubungan antara penelitian terdahulu dengan penelitian saya

1. **Penerapan Metode**

Pada penerapan metode ini digunakan sebagai pedoman penting dalam melaksanakan penelitian. Hal ini dilakukan agar hasil yang dicapai tidak keluar dari tujuan yang telah dilakukan sebelumnya. Tahapan penelitian pada penelitian ini dapat dilihat pada gambar 3.1 berikut:



**Gambar 3.1 Tahapan Penelitian**

1. **Pengumpulan Data**

Pada tahap ini dilakukan pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Wawancara (*interview*)

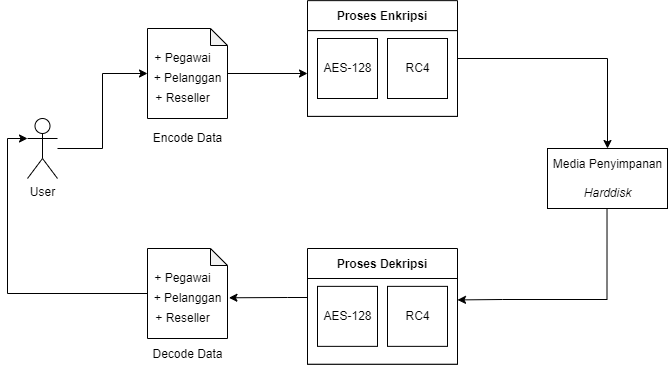
Proses wawancara dilakukan dengan melalui wawancara atau tanya jawab langsung dengan pihak-pihak yang berhubungan dengan aplikasi dan pembuatan program, agar mendapatkan informasi tentang aplikasi dan keamanan yang ada.

1. Observasi (*observation*)

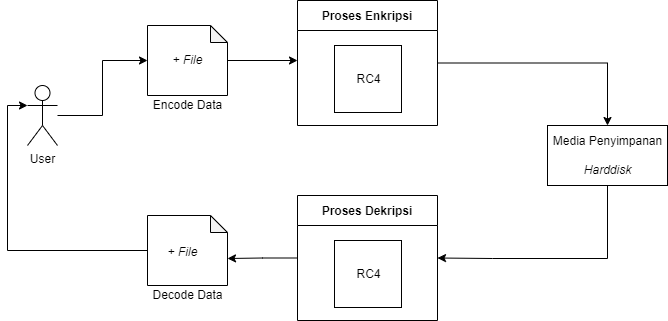
Observasi merupakan salah satu teknik pengumpulan data yang dilakukan melalui pengamatan secara langsung terhadap proses sistem yang sedang berjalan pada CV. Trista Jaya Abadi.

1. **Proses Penerapan Kriptografi**

Dari permasalahan yang sudah dijelaskan, maka diperlukan sebuah aplikasi yang berguna untuk menjaga suatu kerahasiaan data atau file. Aplikasi yang dibuat nantinya dapat mengubah sebuah data menjadi data yang tidak dapat dibaca dan dapat membuat file tidak dapat dibuka setelah melalui proses enkripsi. Data dan file dapat dilihat dan dibuka kembali saat melalui proses dekripsi. Berikut ini adalah proses enkripsi dan dekripsi data dan file berikut:



**Gambar 3.2 Penerapan Metode Kriptografi Data Pegawai, Data Pelanggan dan Data Reseller**



**Gambar 3.2 Penerapan Metode Kriptografi *File***

1. **Implementasi**

Pada tahap implementasi ini akan dilakukan tahap perancangan ke dalam bahasa pemrograman tertentu, dalam hal ini aplikasi yang akan digunakan sebagai berikut:

1. Perangkat lunak yang digunakan dalam penerapan pengamanan data pegawai, data pelanggan dan data *reseller* serta *file* menggunakan bahasa pemrograman PHP dengan *framework CodeIgniter 4* dan DBMS menggunakan MySQL.
2. Perangkat keras yang akan digunakan *Processor* Intel core I5, Ram 8GB, SSD 120 GB dan *Harddisk* 500 GB.
3. **Pengujian Sistem**

Pada tahap pengujian ini akan dilakukan sebuah pengujian yang bertujuan untuk mengetahui sistem yang dibuat sudah sesuai dengan hasil analisis dan perancangan serta sistem yang dibuat sesuai dengan yang diharapkan. Untuk mengetahui sistem sudah berjalan sesuai dengan yang diharapkan, maka dibutuhkan sebuah metode pengujian yang menjadi ukuran atau parameter dalam pengujian sistem.

Metode pengujian yang digunakan adalah *blackbox testing*, yaitu sebuah metode yang akan digunakan untuk melakukan pengujian terhadap aplikasi untuk menemukan kesalahan dan melakukan percobaan pada fungsional aplikasi saat dioperasikan, untuk mengetahui apakah input diterima dengan benar di dalam aplikasi dan output yang dihasilkan di dalam aplikasi telah sesuai dengan yang diharapkan.

1. **Rancangan Pengujian**
2. ***Black Box***

Pengujian *black box* adalah metode pengujian aplikasi yang menguji apakah semua input dan output aplikasi yang dibuat telah berjalan dengan semestinya sesuai dengan kebutuhan *fungsional* aplikasi saat dioperasikan.

Metode pengujian *black box* ini digunakan untuk memastikan bahwa semua *fungsional* aplikasi saat proses enkripsi dan dekripsi data dan file sudah sesuai dengan yang diharapkan. Metode pengujian black box ini juga akan digunakan untuk melakukan pengujian dari sisi kecepatan dan panjang teks aslinya sebelum dan sesudah di enkripsi dalam proses enkripsi dan dekripsi data. Selain itu metode black box ini juga digunakan untuk melakukan pengujian dari sisi kecepatan dalam proses enkripsi dan dekripsi file. Berikut ini adalah tabel rancangan pengujian yang akan digunakan dalam pengujian sebagai berikut:

1. Tabel rancangan pengujian data pegawai

Tabel 3.1 Rancangan pengujian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Karakter Asli | Jumlah Karakter (bit) | Hasil Enkripsi | Jumlah Karakter Hasil Enkripsi (bit) | Waktu Enkripsi |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Tabel rancangan pengujian data pelanggan

Tabel 3.2 Rancangan pengujian data pelanggan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Karakter Asli | Jumlah Karakter (bit) | Hasil Enkripsi | Jumlah Karakter Hasil Enkripsi (bit) | Waktu Enkripsi |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Tabel rancangan pengujian data reseller

Tabel 3.3 Rancangan pengujian data reseller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Karakter Asli | Jumlah Karakter (bit) | Hasil Enkripsi | Jumlah Karakter Hasil Enkripsi (bit) | Waktu Enkripsi |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Tabel rancangan pengujian *file*

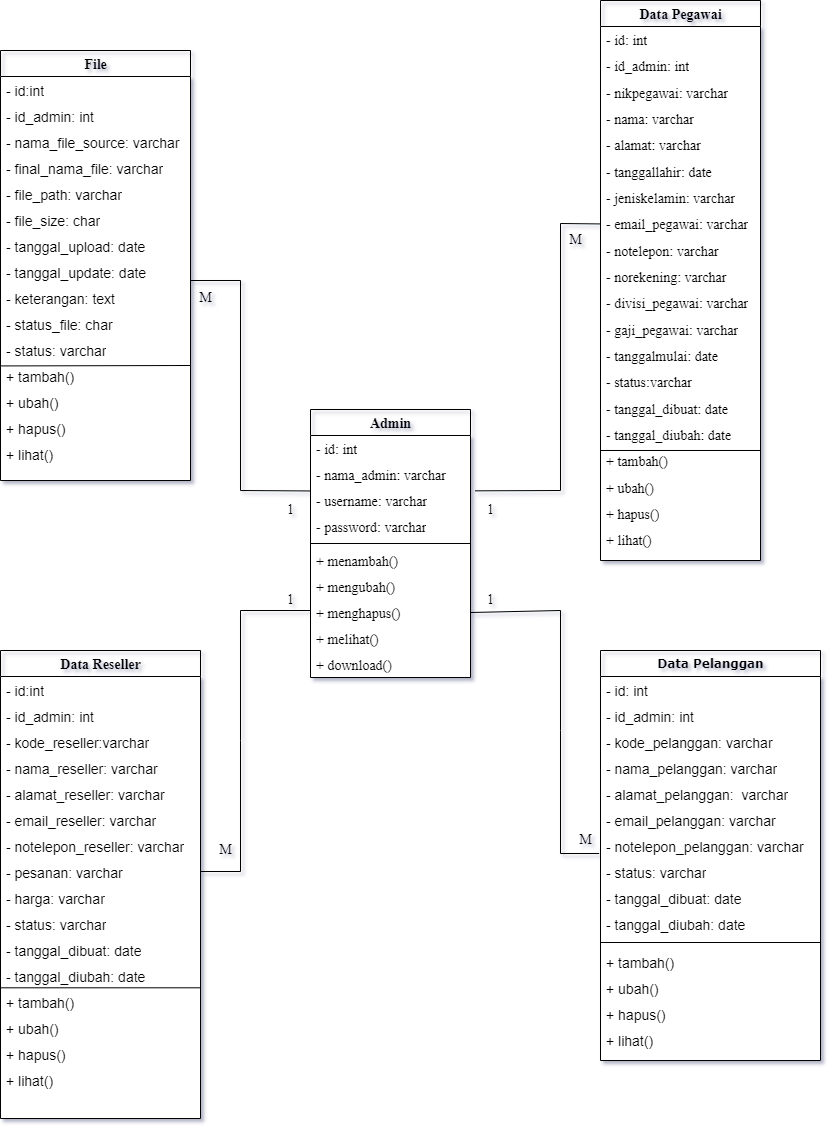
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama File | Ukuran File | | | Waktu | |
| Asli | Enkripsi | Dekripsi | Enkripsi | Dekripsi |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Rancangan Basis Data**

Dalam hal ini rancangan basis data yang akan dibahas yaitu mengenai *Class Diagram*, *Logical Record Structure* (LRS) dan Spesifikasi Basis Data.

1. ***Class Diagram***

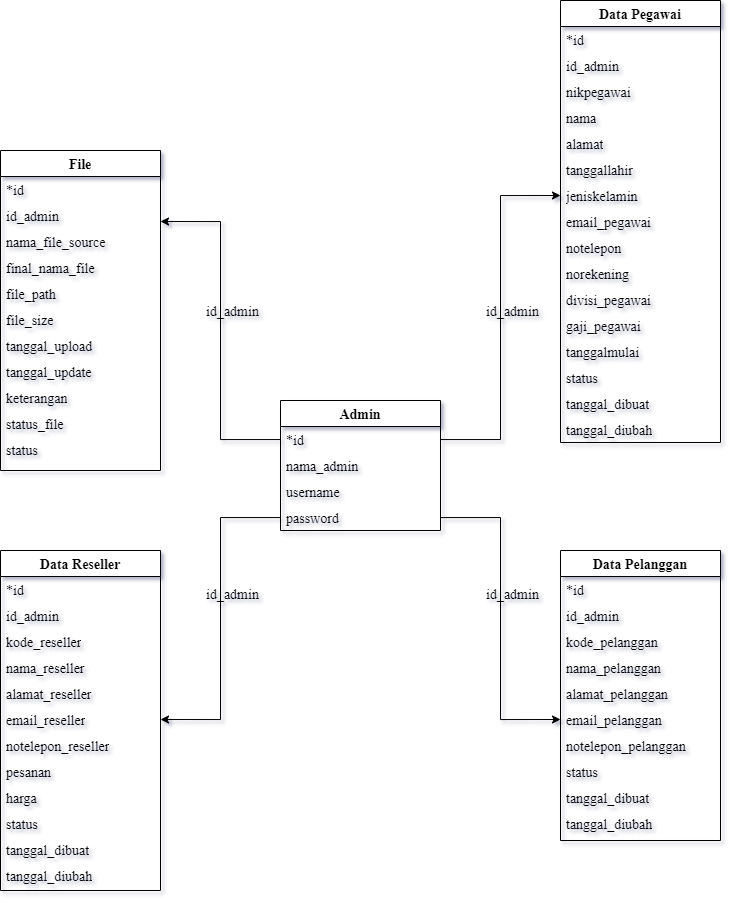
Pada *class diagram* ini menggambarkan struktur-struktur yang ada pada aplikasi. Struktur meliputi atribut dan metode yang ada pada masing-masing *class*. *Class diagram* digambarkan seperti pada gambar 3.2.



**Gambar 3.2 *Class Diagram***

1. ***Logical Record Structure* (LRS)**

Bentuk *Logical Record Structure* (LRS) pada aplikasi yang dibuat sebagai berikut:



**Gambar 3.3 *Logical Record Structure* (LRS)**

1. **Spesifikasi Basis Data**

Berikut adalah struktur-struktur tabel yang terdapat di dalam basis data yang digunakan dalam pembuatan aplikasi ini sebagai berikut :

1. Tabel admin

Nama tabel : admin

Isi : Data admin

Media : *Harddisk*

*Primary Key* : *id*

**Tabel 3.1 Spesifikasi Tabel Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | ***Type*** | **Lebar** | **Keterangan** |
| id | int | 11 | id |
| nama\_admin | varchar | 50 | Nama Admin |
| username | varchar | 50 | Username |
| password | varchar | 100 | Password |

1. Tabel data pegawai

Nama tabel : pegawai

Isi : Data pegawai

Media : *Harddisk*

*Primary Key* : *id*

**Tabel 3.2 Spesifikasi Tabel Pegawai**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | *Type* | Lebar | Keterangan |
| id | int | 11 | Id |
| id\_admin | int | 11 | Dibuat Oleh |
| nikpegawai | varchar | 100 | Nik Pegawai |
| nama | varchar | 50 | Nama Pegawai |
| alamat | varchar | 255 | Alamat Pegawai |
| tanggallahir | date | - | Tanggal Lahir |
| jeniskelamin | varchar | 15 | Jenis Kelamin |
| email\_pegawai | varchar | 100 | Email Pegawai |
| notelepon | varchar | 100 | No Telepon |
| norekening | varchar | 100 | No Rekening |
| divisi\_pegawai | varchar | 100 | Divisi Pegawai |
| gaji\_pegawai | varchar | 100 | Gaji Pegawai |
| tanggalmulai | date | - | Tanggal Mulai |
| status | varchar | 100 | status |
| tanggal\_dibuat | date | - | Tanggal Dibuat |
| tanggal\_diubah | date | - | Tanggal Diubah |

1. Tabel data pelanggan

Nama tabel : pelanggan

Isi : Data pelanggan

Media : *Harddisk*

*Primary Key* : *id*

**Tabel 3.3 Spesifikasi Tabel Pelanggan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | ***Type*** | **Lebar** | **Keterangan** |
| id | int | 11 | Id |
| id\_admin | int | 11 | Dibuat Oleh |
| kode\_pelanggan | varchar | 50 | Kode Pelanggan |
| nama\_pelanggan | varchar | 50 | Nama Pelanggan |
| alamat\_pelanggan | varchar | 255 | Alamat Pelanggan |
| email\_pelanggan | varchar | 100 | Email Pelanggan |
| notelepon\_pelanggan | varchar | 100 | No Telepon |
| status | varchar | 100 | status |
| tanggal\_dibuat | date | - | Tanggal Dibuat |
| tanggal\_diubah | date | - | Tanggal Diubah |

1. Tabel data *reseller*

Nama tabel : *reseller*

Isi : Data *reseller*

Media : *Harddisk*

*Primary Key* : *id*

**Tabel 3.4 Spesifikasi Tabel *Reseller***

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | ***Type*** | **Lebar** | **Keterangan** |
| id | int | 11 | Id |
| id\_admin | int | 11 | Dibuat Oleh |
| kode\_reseller | varchar | 50 | Kode Reseller |
| nama\_reseller | varchar | 50 | Nama Reseller |
| alamat\_reseller | varchar | 255 | Alamat Reseller |
| email\_reseller | varchar | 100 | Email Reseller |
| notelepon\_reseller | varchar | 100 | No Telepon |
| pesanan | varchar | 100 | Jumlah Pesanan |
| harga | varchar | 100 | Harga Pesanan |
| status | varchar | 100 | status |
| tanggal\_dibuat | date | - | Tanggal Dibuat |
| tanggal\_diubah | date | - | Tanggal Diubah |

1. Tabel *file*

Nama tabel : *file*

Isi : Data *file*

Media : *Harddisk*

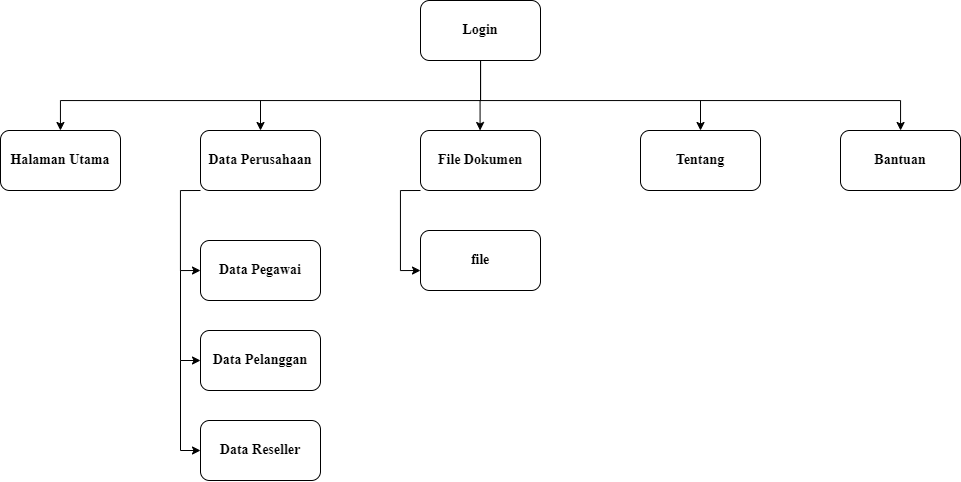
*Primary Key* : *id*

**Tabel 3.5 Spesifikasi Tabel *File***

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Field*** | ***Type*** | **Lebar** | **Keterangan** |
| id | int | 11 | Id |
| id\_admin | int | 11 | Dibuat Oleh |
| nama\_file\_source | varchar | 50 | Nama File Source |
| final\_nama\_file | varchar | 50 | Final Nama File |
| file\_path | varchar | 100 | File Path |
| file\_size | char | 100 | File Size |
| tanggal\_upload | date | - | Tanggal Upload |
| tanggal\_update | date | - | Tanggal Update |
| keterangan | text | - | Keterangan |
| status\_file | char | 1 | Status enkrip |
| status | varchar | 100 | status |

1. **Rancangan Menu**

Rancangan menu pada aplikasi yang akan dibuat terdiri dari beberapa tampilan menu yang dapat dilihat pada gambar 3.4 berikut:



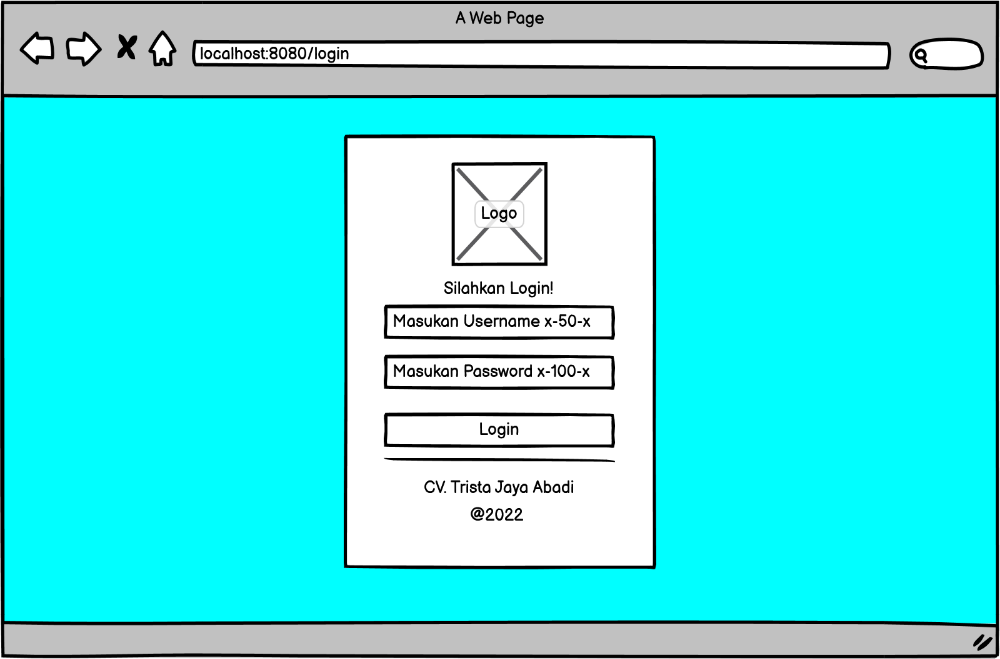
**Gambar 3.4 Rancangan Menu**

1. **Rancangan Layar**

Rancangan layar sangat penting dan diperlukan dalam membuat suatu program. Oleh karena itu, rancangan layar dibuat harus mudah dimengerti dan dipahami, agar tidak membingungkan dan tidak mengalami kesulitan saat *user* menggunakan aplikasi ini. Rancangan layar dari aplikasi yang dibuat adalah sebagai berikut:

1. **Rancangan Layar Halaman *Login***

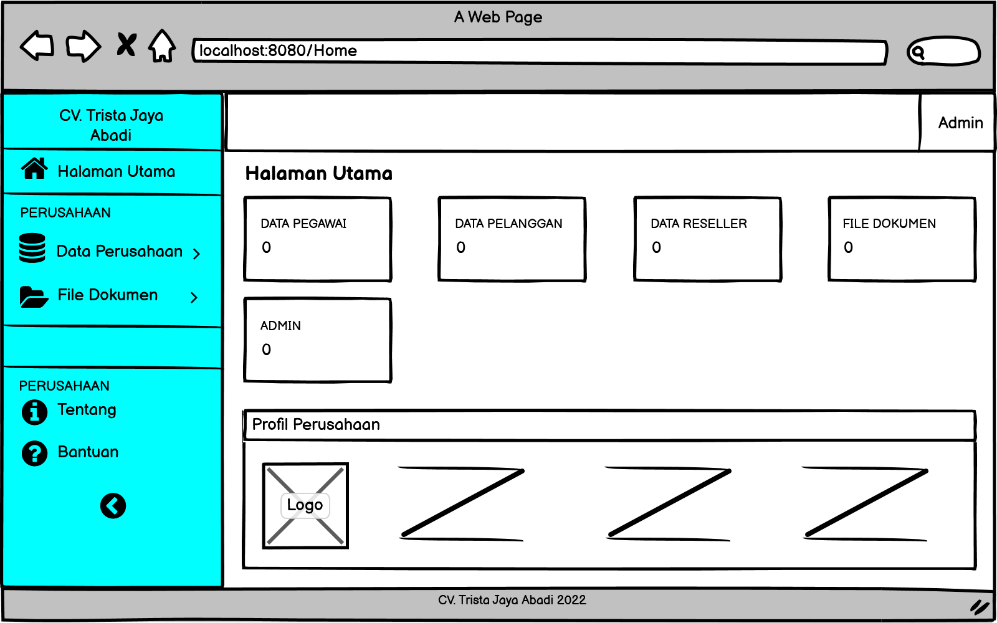
Pada halaman ini merupakan tampilan rancangan layar halaman *login*. Pada *form* *login* berisikan *field* *username* dan *password*. Selain itu terdapat tombol *login* yang berfungsi untuk melakukan proses *login*. Untuk masuk ke dalam aplikasi ini *user* harus melakukan login. Rancangan layar *form login* dapat dilihat pada gambar 3.5 berikut:



**Gambar 3.5 Rancangan Layar Halaman *Login***

1. **Rancangan Layar Halaman Utama**

Pada halaman ini merupakan tampilan rancangan layar halaman utama. Dalam halaman utama berisi informasi tentang jumlah data pegawai, data pelanggan, data *reseller*, *file* dan data admin serta berisi tentang informasi berupa profil perusahaan pada CV. Trista Jaya Abadi. Selain itu terdapat beberapa menu yang dapat digunakan untuk pendataan data perusahaan dan *file* dokumen. Rancangan layar halaman utama dapat dilihat pada gambar 3.6 berikut:



**Gambar 3.6 Rancangan Layar Halaman Utama**

1. **Rancangan Layar Halaman Data Pegawai**

Pada halaman ini merupakan tampilan rancangan layar halaman data pegawai. Pada halaman ini terdapat tombol tambah data yang digunakan untuk menambah data pegawai. Pada halaman ini terdapat tabel data pegawai yang berisikan data dari data pegawai. Selain itu terdapat tombol ubah data yang digunakan untuk mengubah atau melihat data pegawai dan terdapat tombol hapus yang digunakan untuk menghapus data pegawai. Rancangan layar halaman data pegawai dapat dilihat pada gambar 3.7 berikut:

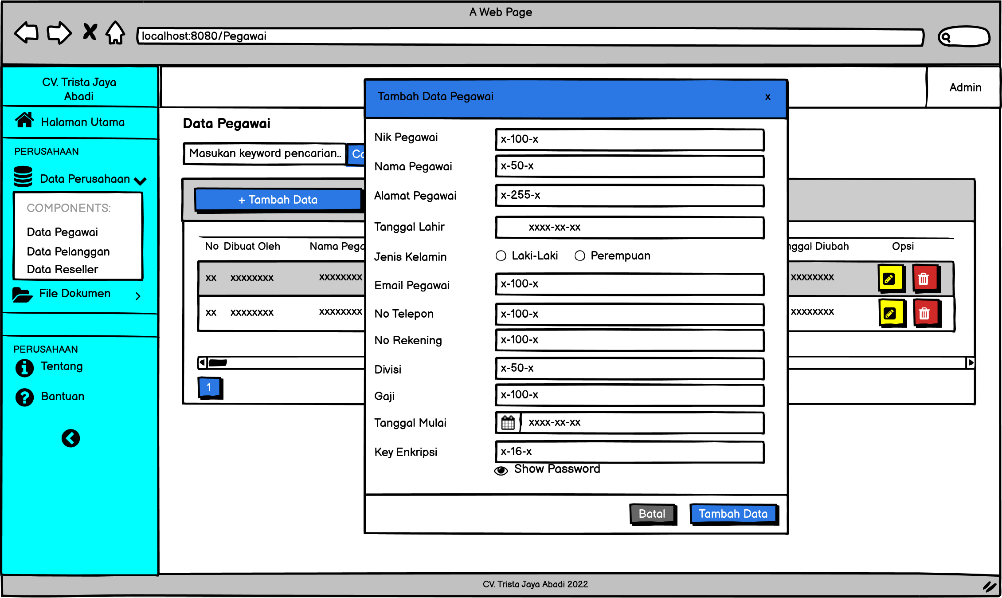
Sebuah gambar berisi teks

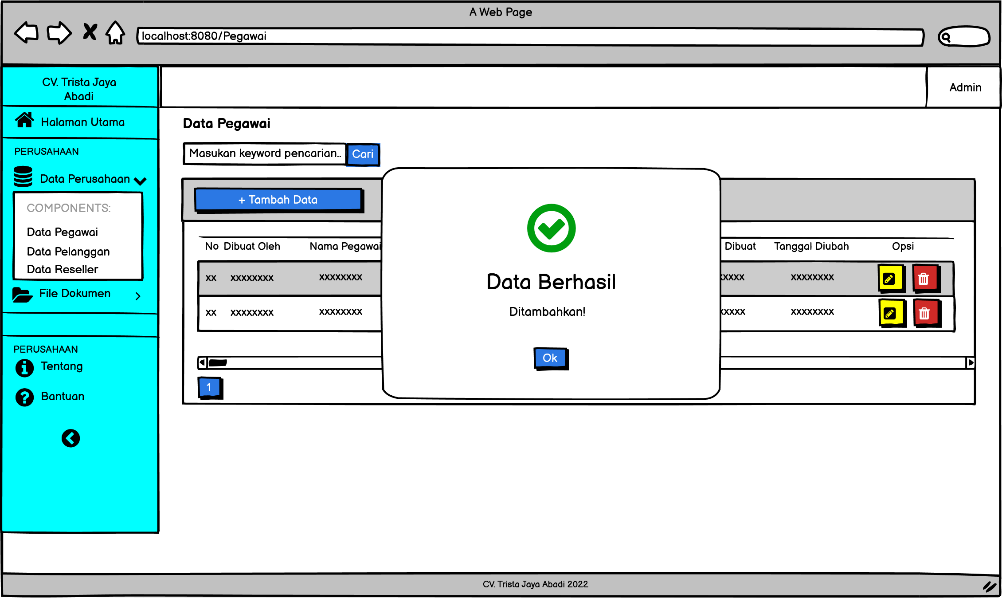
Deskripsi dibuat secara otomatis

**Gambar 3.7 Rancangan Layar Halaman Data Pegawai**

1. **Rancangan Layar *Form* Tambah Data Pegawai**

Pada halaman ini merupakan tampilan rancangan layar *form* tambah data pegawai. Pada *form* ini terdapat *field* yang berisikan nik pegawai, nama pegawai, alamat pegawai, tanggal lahir, jenis kelamin, email pegawai, no telepon, no rekening, divisi, gaji, tanggal mulai dan *key* enkripsi yang harus diisi oleh *user* jika ingin menambah data pegawai. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini juga terdapat tombol tambah data yang digunakan untuk menambah data pegawai yang telah diisi oleh *user* ke dalam *database* dan setelah berhasil tambah data maka akan tampil *pop up* berhasil tambah data seperti gambar 3.9. Selain itu terdapat tombol batal yang digunakan untuk membatalkan tambah data serta menutup *form* tambah data pegawai. Rancangan layar *form* tambah data pegawai dapat dilihat pada gambar 3.8 berikut:

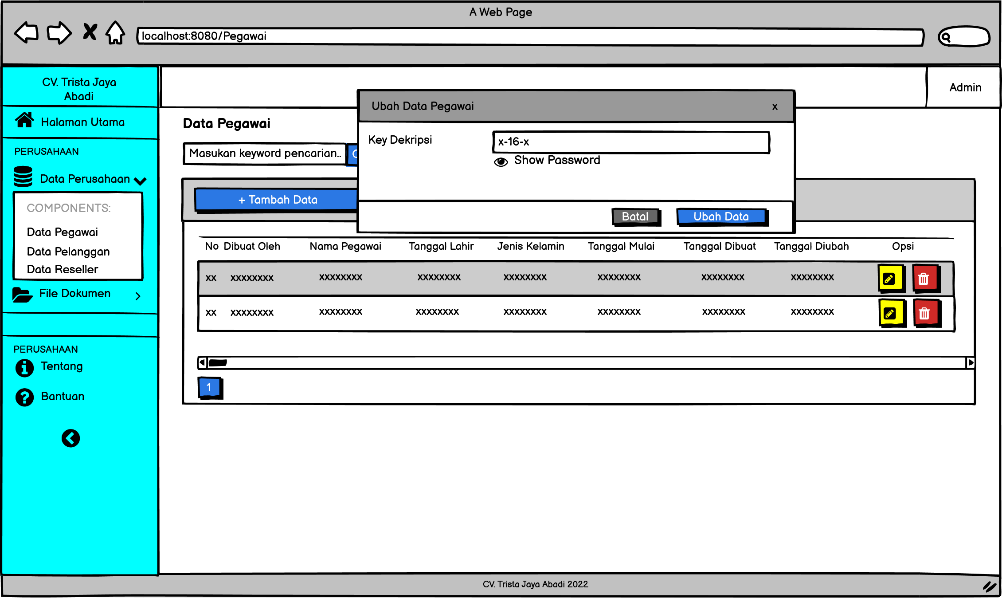


 **Gambar 3.8 Rancangan Layar *Form* Tambah Data Pegawai**

**Gambar 3.9 Rancangan Layar Berhasil Tambah Data Pegawai**

1. **Rancangan Layar *Form* *Key* Dekripsi Ubah Data Pegawai**

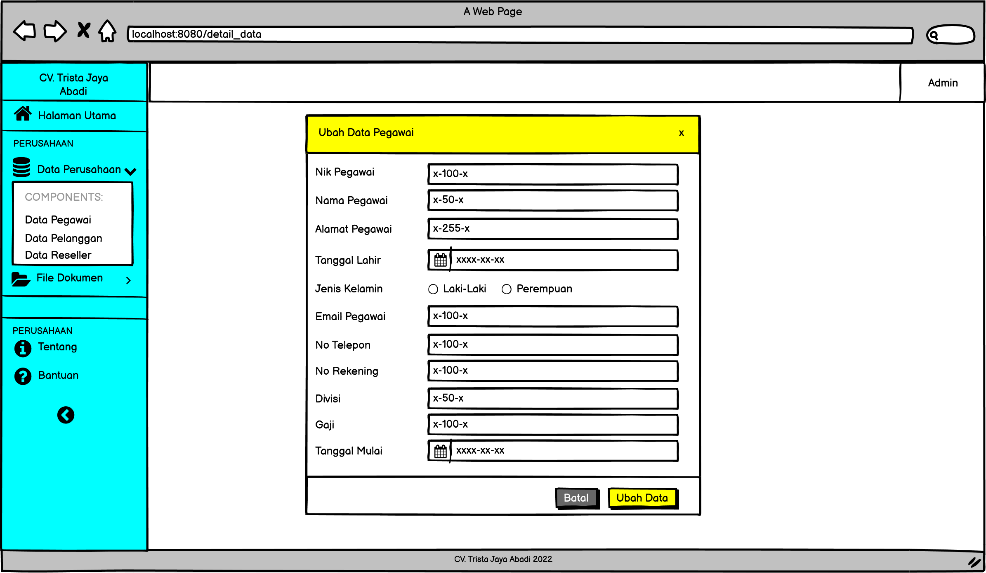
Pada halaman ini merupakan tampilan rancangan layar *form* *key* dekripsi ubah data pegawai. Pada *form* ini terdapat *field* yang berisikan *key* dekripsi yang harus diisi jika user ingin mengubah atau melihat data yang terenkripsi. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini juga terdapat tombol ubah data yang digunakan untuk masuk ke dalam halaman ubah data pegawai jika *key* yang dimasukan benar dan terdapat tombol batal yang digunakan untuk membatalkan jika tidak ingin masuk ke halaman ubah data. Rancangan layar *form* *key* dekripsi ubah data pegawai dapat dilihat pada gambar 3.10 berikut:



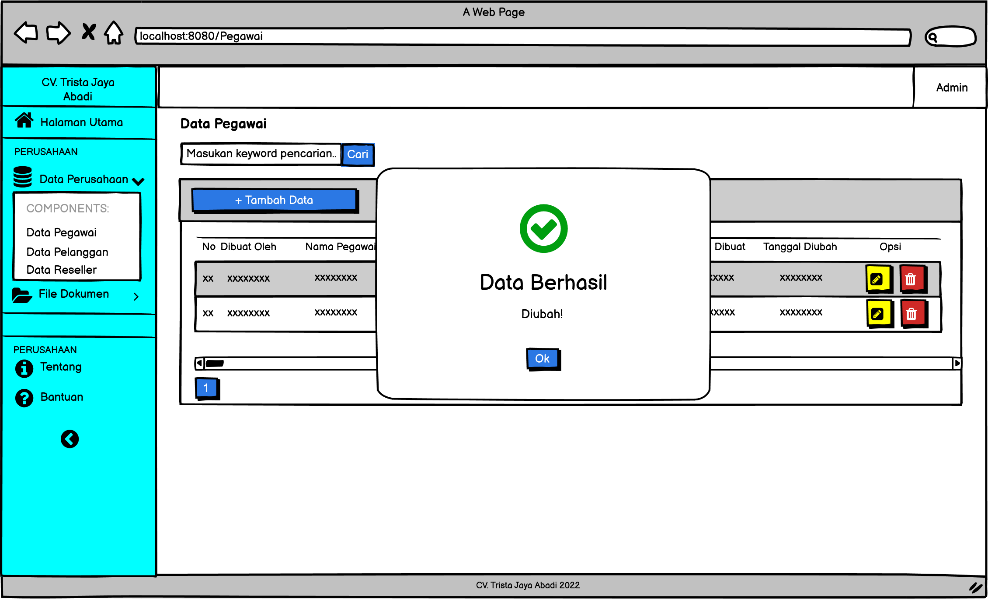
**Gambar 3.10 Rancangan Layar *Form* *Key* Dekripsi Ubah Data Pegawai**

1. **Rancangan Layar Halaman *Form* Ubah Data Pegawai *Key* Benar**

Pada halaman ini merupakan tampilan rancangan layar halaman *form* ubah data pegawai *key* benar. Pada halaman *form* ini terdapat *field* yang berisikan nik pegawai, nama pegawai, alamat pegawai, tanggal lahir, jenis kelamin, email pegawai, no telepon, no rekening, divisi, gaji, dan tanggal mulai yang *user* bisa mengubah atau tidak mengubah data yang sudah ada di dalam *field* ubah data pegawai. Pada *form* ini terdapat tombol ubah data yang digunakan untuk mengubah data pegawai yang sudah diubah oleh *user* dan setelah berhasil ubah data maka akan tampil *pop up* berhasil ubah data seperti gambar 3.12. Selain itu terdapat tombol batal yang digunakan untuk membatalkan mengubah data pegawai dan menutup halaman *form* ubah data pegawai, serta dikembalikan ke halaman data pegawai. Rancangan layar halaman *form* ubah data pegawai *key* benar dapat dilihat pada gambar 3.11 berikut:



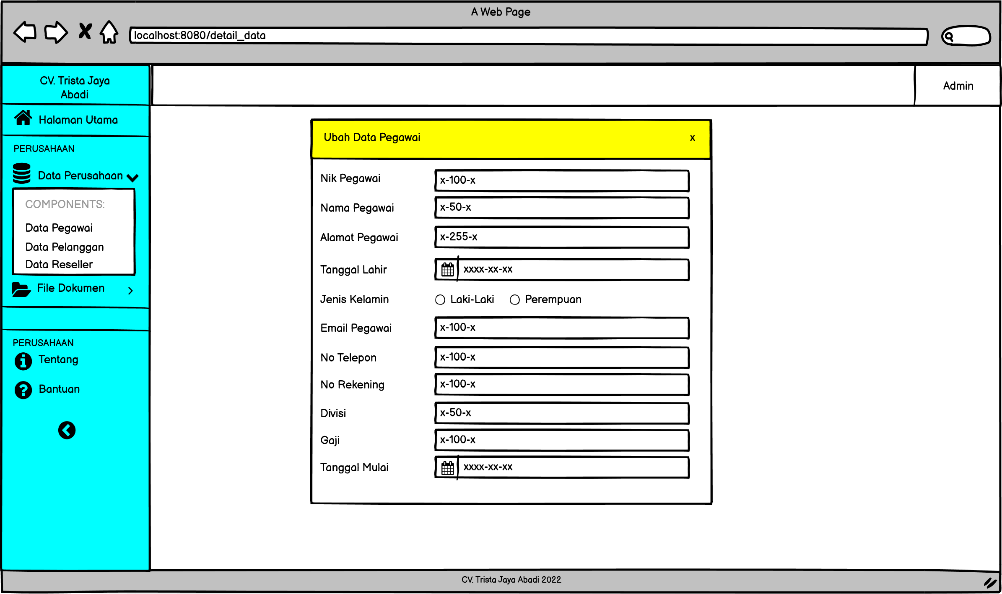
**Gambar 3.11 Rancangan Layar Halaman *Form* Ubah Data Pegawai *Key* Benar**



**Gambar 3.12 Rancangan Layar Berhasil Ubah Data Pegawai**

1. **Rancangan Layar Halaman *Form* Ubah Data Pegawai *Key* Salah**

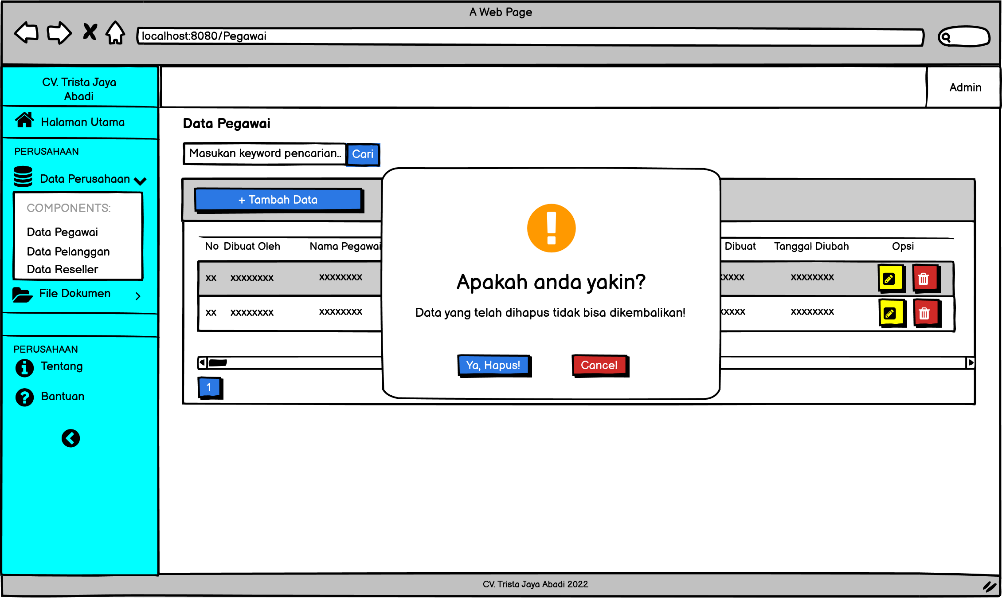
Pada halaman ini merupakan tampilan rancangan layar halaman *form* ubah data pegawai *key* salah. Pada halaman *form* ini terdapat *field* yang berisikan nik pegawai, nama pegawai, alamat pegawai, tanggal lahir, jenis kelamin, email pegawai, no telepon, no rekening, divisi, gaji, dan tanggal mulai yang masih terenkripsi karena *key* yang dimasukan salah. Pada *form* ubah data pegawai, hanya *field* yang bersifat data penting saja yang masih terenkripsi dan pada halaman *form* ini *user* tidak dapat melihat atau mengubah data selain kembali kehalaman data pegawai karena di *form* ini tidak ada tombol batal dan ubah data. Rancangan layar halaman *form* ubah data pegawai *key* salah dapat dilihat pada gambar 3.13 berikut:



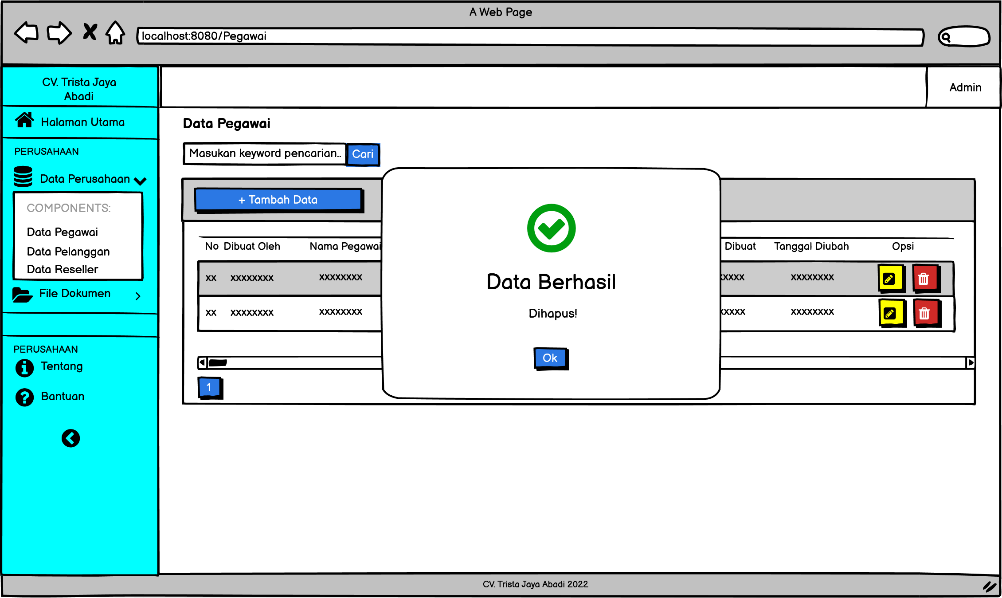
**Gambar 3.13 Rancangan Layar Halaman *Form* Ubah Data Pegawai *Key* Salah**

1. **Rancangan Layar Hapus Data Pegawai**

Pada halaman ini merupakan tampilan rancangan layar hapus data pegawai. Pada halaman data pegawai jika *user* klik ikon hapus, maka akan tampil *pop up* yang berisikan apakah anda yakin. Jika *user* mengklik tombol hapus, maka data yang ingin dihapus akan langsung terhapus di dalam *database* dan tombol *cancel* untuk membatalkan menghapus data pegawai. Rancangan layar hapus data pegawai dapat dilihat pada gambar 3.14 dan gambar 3.15 dibawah ini:



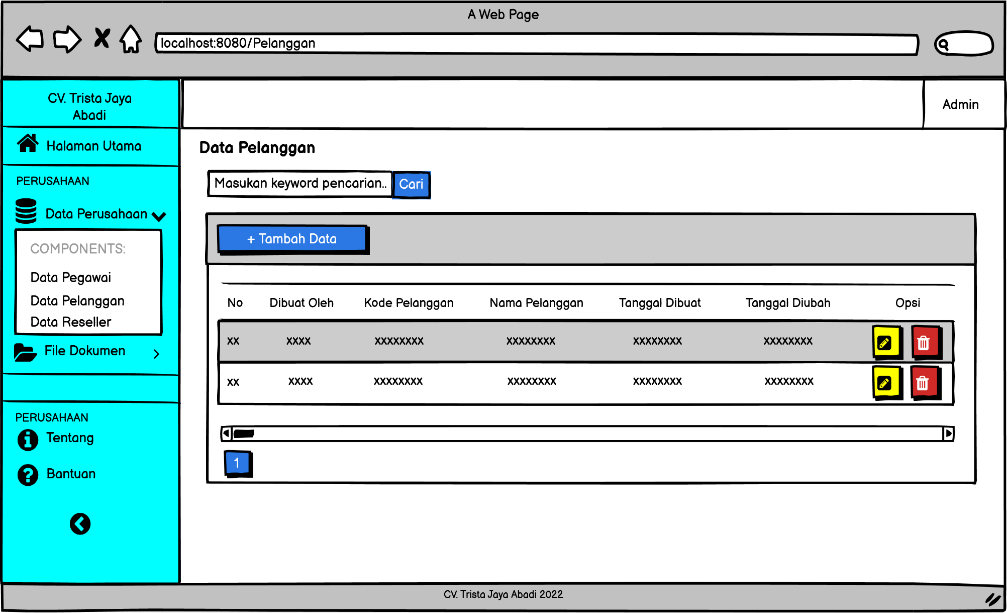
**Gambar 3.14 Rancangan layar Hapus Data Pegawai**



**Gambar 3.15 Rancangan Layar Berhasil Hapus Data Pegawai**

1. **Rancangan Layar Halaman Data Pelanggan**

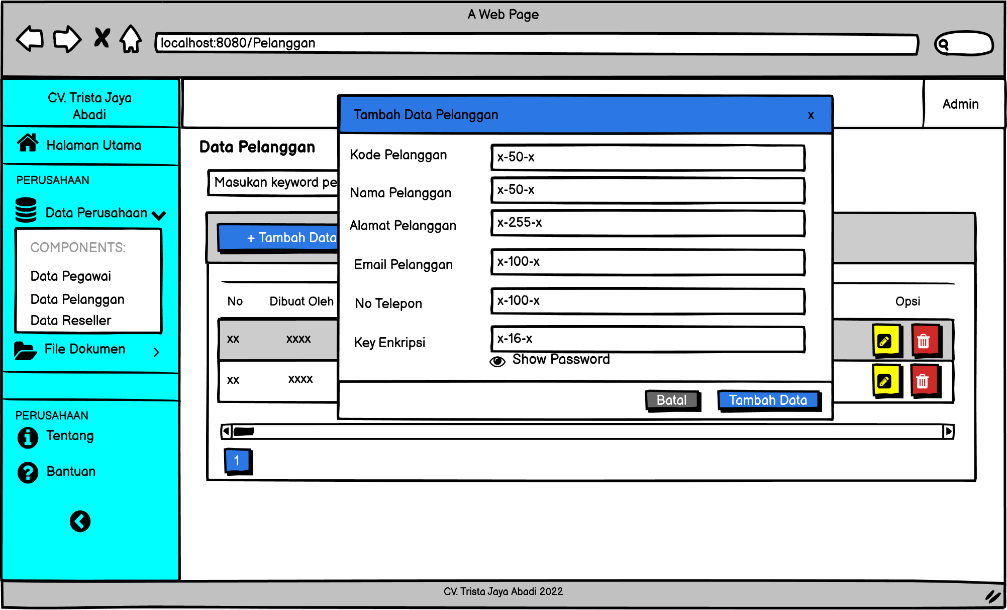
Pada halaman ini merupakan tampilan rancangan layar halaman data pelanggan. Pada halaman ini terdapat tombol tambah data yang digunakan untuk menambah data pelanggan. Pada halaman ini terdapat sebuah tabel data pelanggan yang berisikan data dari data pelanggan. Selain itu terdapat tombol ubah yang digunakan untuk mengubah atau melihat data pelanggan dan terdapat tombol hapus yang digunakan untuk menghapus data pelanggan. Rancangan layar halaman data pelanggan dapat dilihat pada gambar 3.16 berikut:



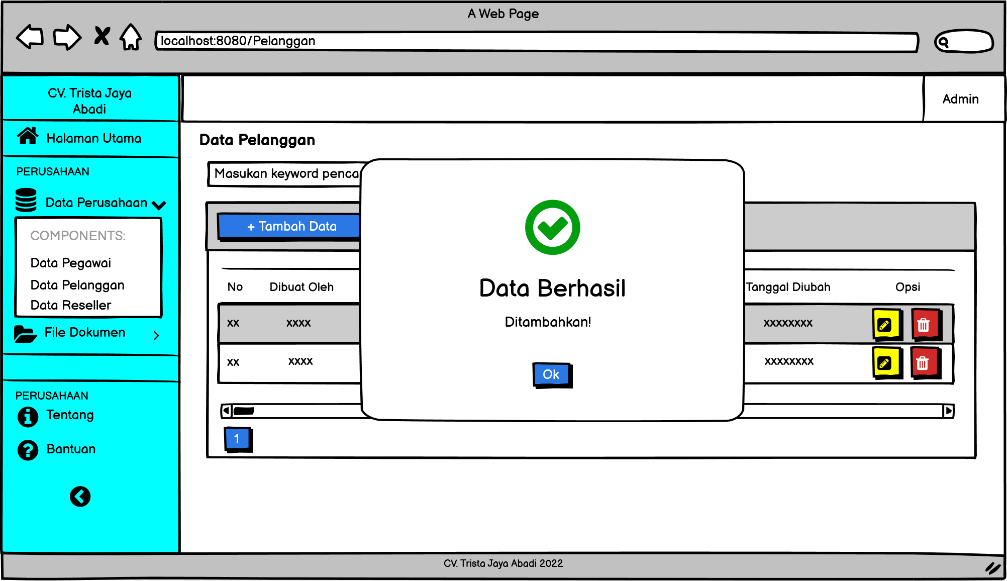
**Gambar 3.16 Rancangan Layar Halaman Data Pelanggan**

1. **Rancangan Layar Form Tambah Data Pelanggan**

Pada halaman ini merupakan tampilan rancangan layar *form* tambah data pelanggan. Pada *form* ini terdapat *field* yang berisikan kode pelanggan, nama pelanggan, alamat pelanggan, email pelanggan, no telepon, dan *key* enkripsi yang harus diisi oleh *user* jika ingin menambah data pelanggan. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini juga terdapat tombol tambah data yang digunakan untuk menambah data pelanggan yang telah diisi oleh *user* ke dalam *database* dan setelah berhasil tambah data maka akan tampil pop up berhasil tambah data seperti gambar 3.18. Selain itu terdapat tombol batal yang digunakan untuk membatalkan tambah data dan menutup *form* tambah data. Rancangan layar *form* tambah data pelanggan dapat dilihat pada gambar 3.17 berikut:



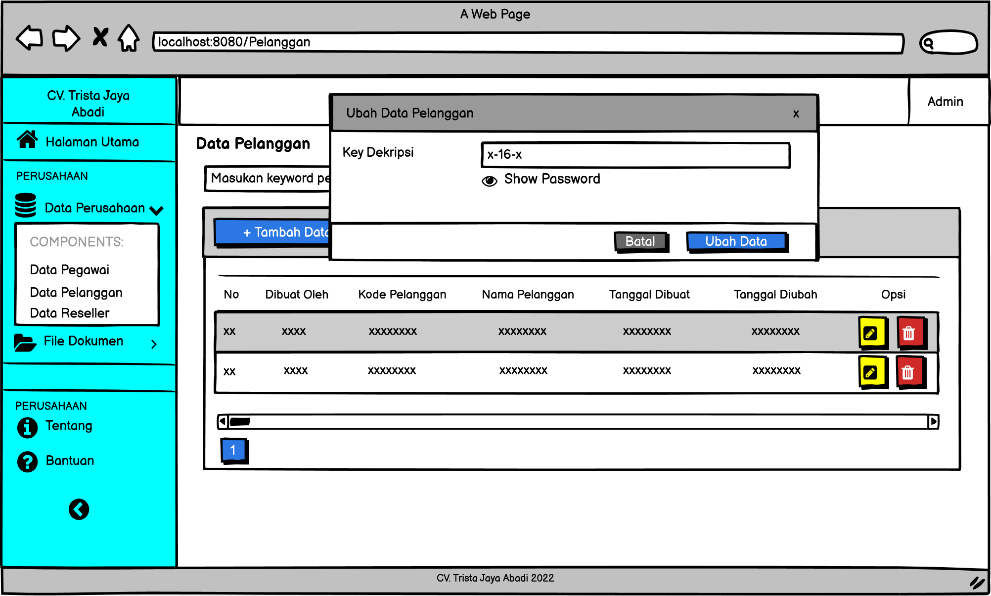
**Gambar 3.17 Rancangan Layar *Form* Tambah Data Pelanggan**



**Gambar 3.18 Rancangan Layar Berhasil Tambah Data Pelanggan**

1. **Rancangan Layar *Form* *Key* Dekripsi Ubah Data Pelanggan**

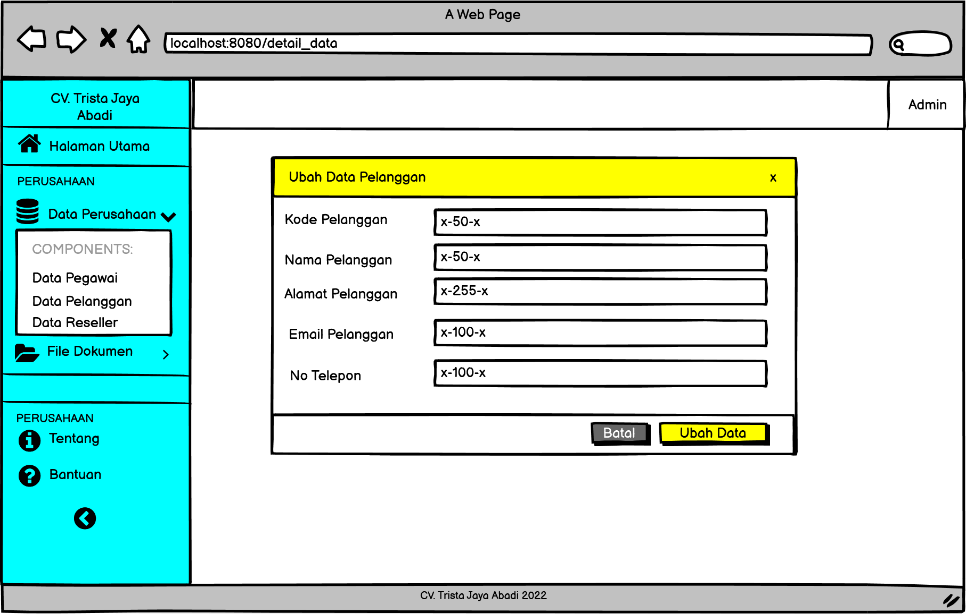
Pada halaman ini merupakan tampilan rancangan layar *form* *key* dekripsi ubah data pelanggan. Pada *form* ini terdapat *field* yang berisikan *key* dekripsi yang harus diisi jika *user* ingin mengubah atau melihat data yang terenkripsi. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini juga terdapat tombol ubah data yang digunakan untuk masuk ke dalam halaman ubah data pelanggan jika *key* yang dimasukan benar dan terdapat tombol batal yang digunakan untuk membatalkan jika tidak ingin masuk ke halaman ubah data. Rancangan layar *form* *key* dekripsi ubah data pelanggan dapat dilihat pada gambar 3.19 berikut:

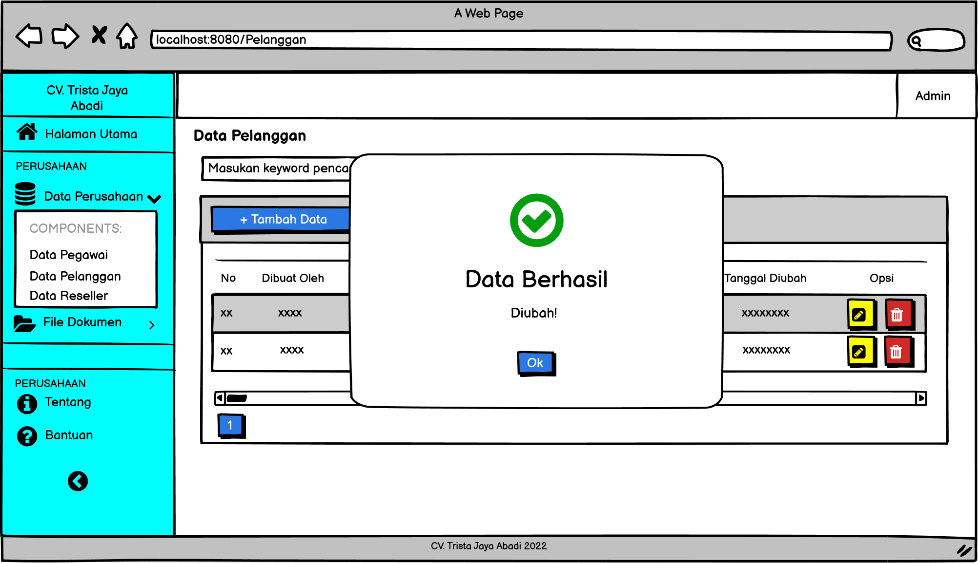


**Gambar 3.19 Rancangan Layar *Form* *Key* Dekripsi Ubah Data Pelanggan**

1. **Rancangan Layar Halaman *Form* Ubah Data Pelanggan *Key* Benar**

Pada halaman ini merupakan tampilan rancangan layar halaman *form* ubah data pelanggan *key* benar. Pada halaman *form* ini terdapat *field* yang berisikan kode pelanggan, nama pelanggan, alamat pelanggan, email pelanggan dan no telepon yang *user* bisa mengubah atau tidak mengubah data yang sudah ada di dalam *field* ubah data pelanggan. Pada *form* ini terdapat tombol ubah data yang digunakan untuk mengubah data pelanggan yang sudah diubah oleh *user* dan setelah berhasil ubah data maka akan tampil *pop up* berhasil ubah data seperti gambar 3.21. Selain itu terdapat tombol batal yang digunakan untuk membatalkan mengubah data pelanggan dan menutup *form* ubah data pelanggan, serta dikembalikan ke halaman data pelanggan. Rancangan layar halaman *form* ubah data pelanggan *key* benar dapat dilihat pada gambar 3.20 berikut:

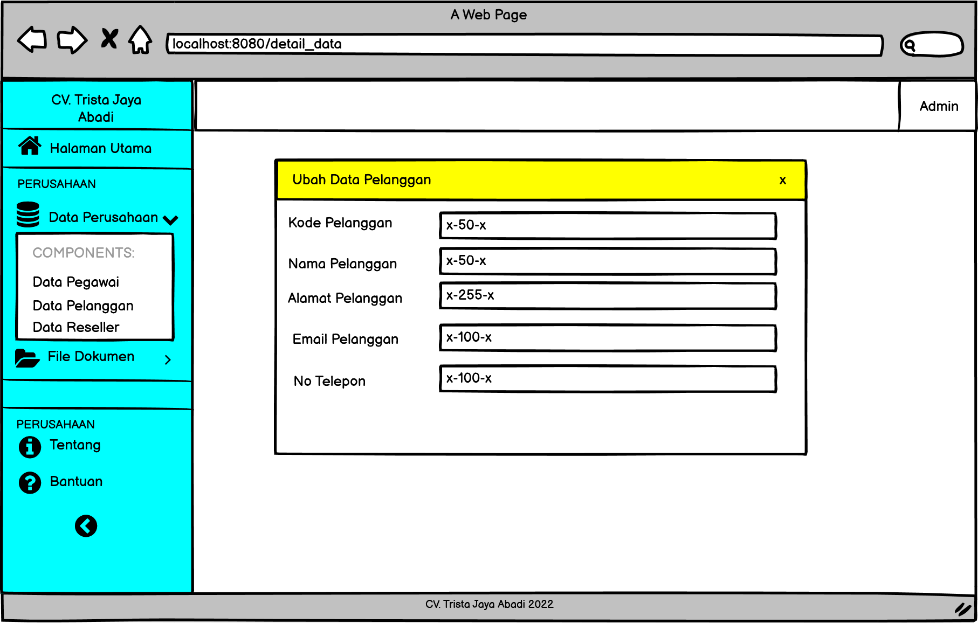


 **Gambar 3.20 Rancangan Layar Halaman *Form* Ubah Data Pelanggan  *Key* Benar**

**Gambar 3.21 Rancangan Layar Berhasil Ubah Data Pelanggan**

1. **Rancangan Layar Halaman *Form* Ubah Data Pelanggan *Key* Salah**

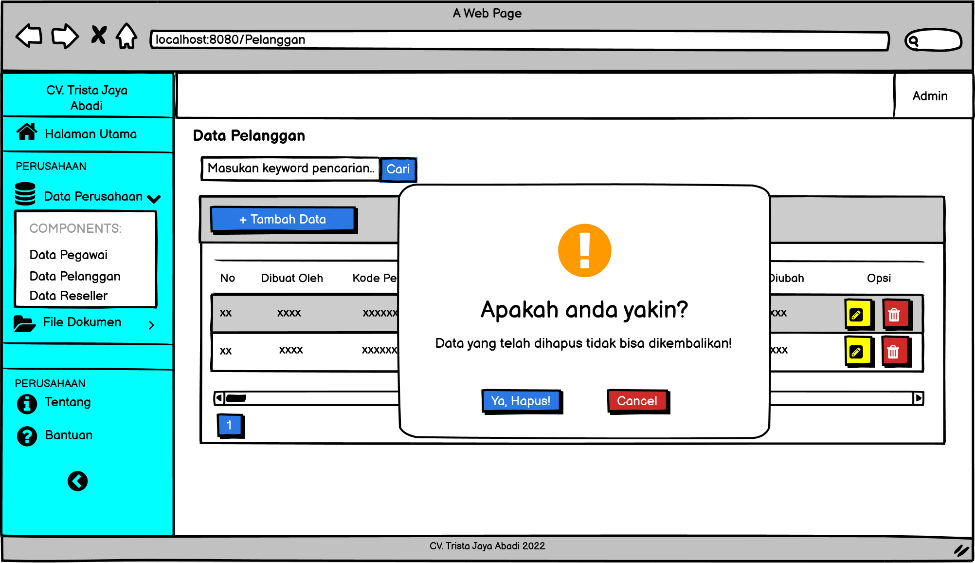
Pada halaman ini merupakan tampilan rancangan layar halaman *form* ubah data pelanggan *key* salah. Pada halaman *form* ini terdapat *field* yang berisikan kode pelanggan, nama pelanggan, alamat pelanggan, email pelanggan dan no telepon yang masih terenkripsi karena *key* yang dimasukan salah. Pada *form* ubah data pelanggan, hanya *field* yang bersifat data penting saja yang masih terenkripsi dan pada halaman *form* ini *user* tidak dapat melihat atau mengubah data selain kembali ke halaman data pelanggan karena di *form* ini tidak ada tombol batal dan ubah data. Rancangan layar halaman *form* ubah data pelanggan *key* salah dapat dilihat pada gambar 3.22 berikut:



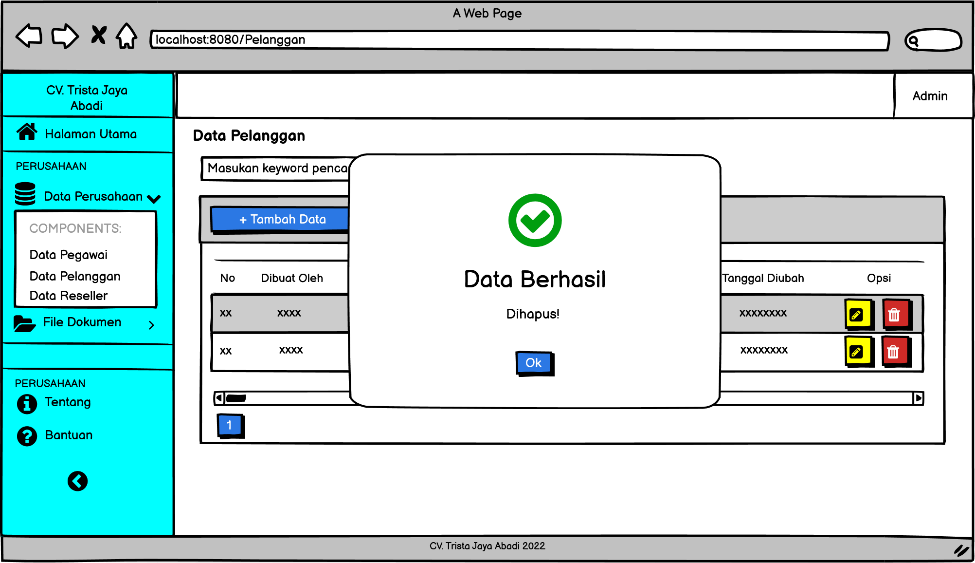
**Gambar 3.22 Rancangan Layar Halaman *Form* Ubah Data Pelanggan *Key* Salah**

1. **Rancangan Layar Hapus Data Pelanggan**

Pada halaman ini merupakan tampilan rancangan layar hapus data pelanggan. Pada halaman data pelanggan jika *user* klik ikon hapus, maka akan tampil *pop up* yang berisikan apakah anda yakin. Jika *user* mengklik tombol hapus, maka data yang ingin dihapus akan langsung terhapus di dalam *database* dan tombol *cancel* untuk membatalkan menghapus data pelanggan. Rancangan layar hapus data pelanggan dapat dilihat pada gambar 3.23 dan gambar 3.24 dibawah ini:



**Gambar 3.23 Rancangan Layar Hapus Data Pelanggan**



**Gambar 3.24 Rancangan Layar Berhasil Hapus Data Pelanggan**

1. **Rancangan Layar Halaman Data *Reseller***

Pada halaman ini merupakan tampilan rancangan layar data *reseller*. Pada halaman ini terdapat tombol tambah data yang digunakan untuk menambah data *reseller*. Pada halaman ini terdapat sebuah tabel data reseller yang berisikan data dari data *reseller*. Selain itu terdapat tombol ubah data yang digunakan untuk mengubah atau melihat data *reseller* dan terdapat tombol hapus yang digunakan untuk menghapus data *reseller*. Rancangan layar halaman data *reseller* dapat dilihat pada gambar 3.25 berikut:

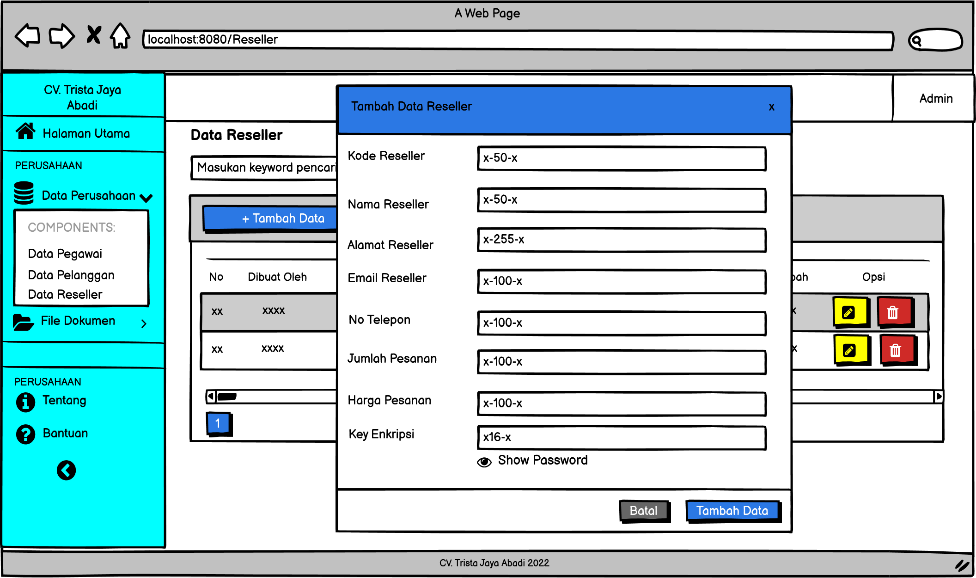
Sebuah gambar berisi teks

Deskripsi dibuat secara otomatis

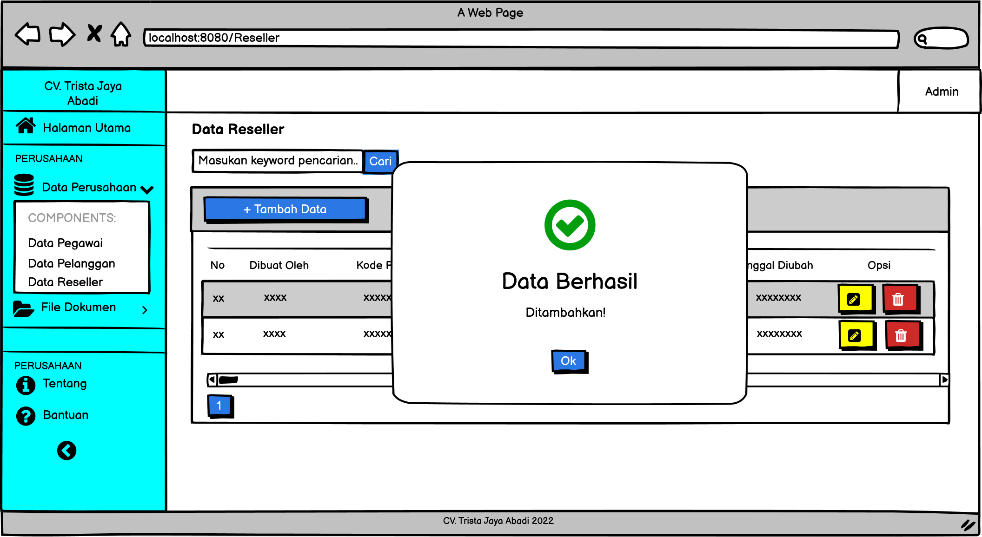
**Gambar 3.25 Rancangan Layar Halaman Data *Reseller***

1. **Rancangan Layar *Form* Tambah Data *Reseller***

Pada halaman ini merupakan tampilan rancangan layar *form* tambah data *reseller*. Pada *form* ini terdapat *field* yang berisikan kode *reseller*, nama *reseller*, alamat *reseller*, email *reseller*, no telepon, jumlah pesanan, harga pesanan, dan *key* enkripsi yang harus diisikan oleh *user* jika ingin menambah data *reseller*. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini juga terdapat tombol tambah data yang digunakan untuk menambah data *reseller* yang telah diisi oleh *user* ke dalam *database* dan setelah berhasil tambah data maka akan tampil *pop up* berhasil tambah data seperti gambar 3.27. Selain itu terdapat tombol batal yang digunakan untuk membatalkan tambah data dan menutup *form* tambah data *reseller*. Rancangan layar *form* tambah data *reseller* dapat dilihat pada gambar 3.26 berikut:



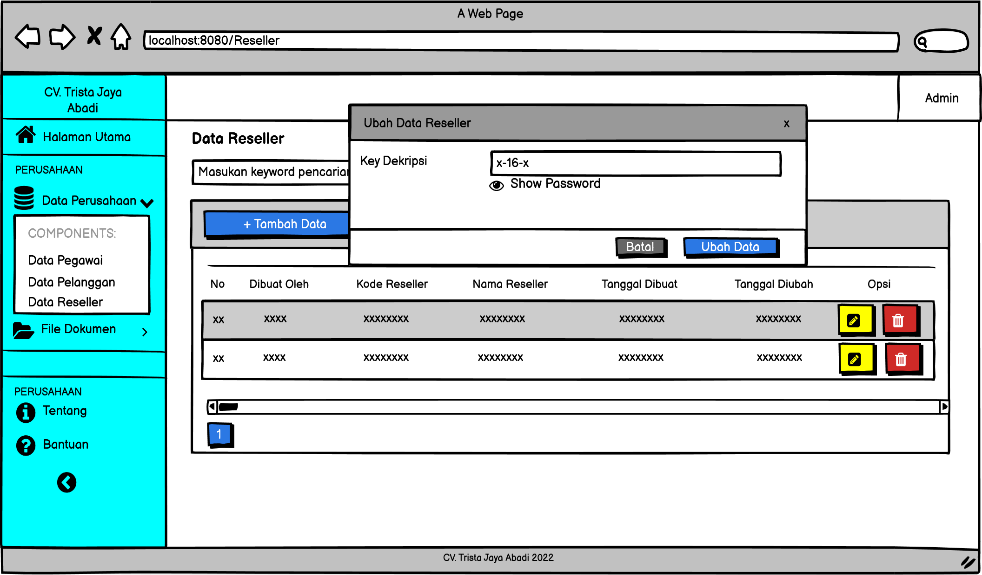
**Gambar 3.26 Rancangan Layar *Form* Tambah Data *Reseller***



**Gambar 3.27 Rancangan Layar Berhasil Tambah Data *Reseller***

1. **Rancangan Layar *Form* *Key* Dekripsi Ubah Data *Reseller***

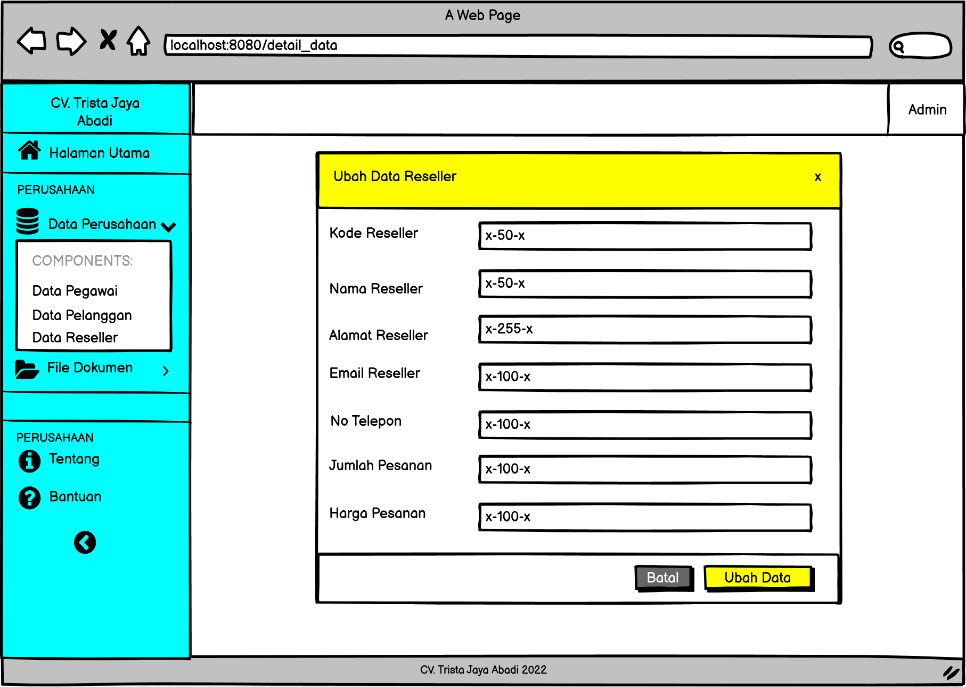
Pada halaman ini merupakan tampilan rancangan layar *form* *key* dekripsi ubah data *reseller*. Pada *form* ini terdapat *field* yang berisikan *key* dekripsi yang harus diisi jika user ingin mengubah atau melihat data yang terenkripsi. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini juga terdapat tombol ubah data yang digunakan untuk masuk ke dalam halaman ubah data *reseller* jika *key* yang dimasukan benar dan terdapat tombol batal yang digunakan untuk membatalkan jika tidak ingin masuk ke halaman ubah data. Rancangan layar *form* *key* dekripsi ubah data *reseller* dapat dilihat pada gambar 3.28 berikut:



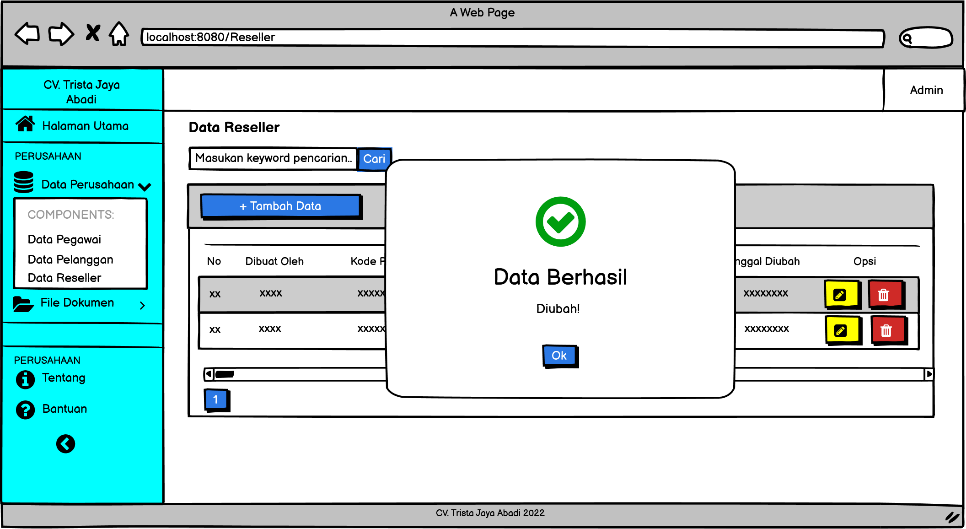
**Gambar 3.28 Rancangan Layar *Form* *Key* Dekripsi Ubah Data *Reseller***

1. **Rancangan Layar Halaman *form* Ubah Data Reseller Key Benar**

Pada halaman ini merupakan tampilan rancangan layar halaman *form* ubah data *reseller key* benar. Pada halaman *form* ini terdapat *field* yang berisikan kode *reseller*, nama *reseller*, alamat *reseller*, email *reseller*, no telepon, jumlah pesanan dan harga pesanan yang *user* bisa mengubah atau tidak mengubah data yang sudah ada di dalam *field* ubah data *reseller*. Pada *form* ini terdapat tombol ubah data yang digunakan untuk mengubah data *reseller* yang sudah diubah oleh *user* dan setelah berhasil ubah data maka akan tampil *pop up* berhasil ubah data seperti gambar 3.30. Selain itu terdapat tombol batal yang digunakan untuk membatalkan mengubah data *reseller* dan menutup *form* ubah data *reseller,* serta dikembalikan ke halaman data *reseller*. Rancangan layar halaman *form* ubah data *reseller key* benar dapat dilihat pada gambar 3.29 berikut:



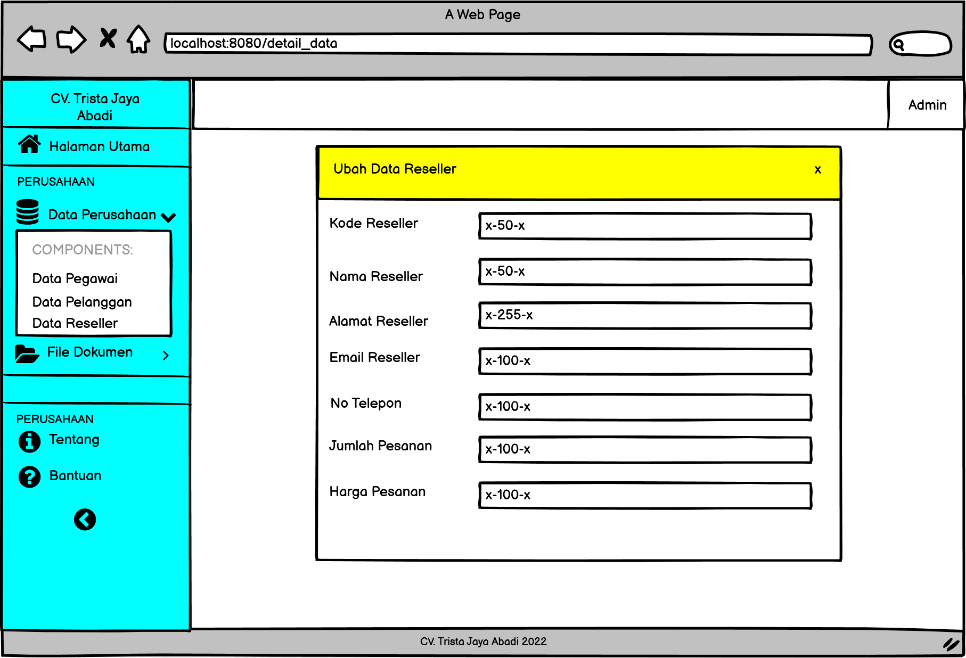
**Gambar 3.29 Rancangan Layar Halaman *Form* Ubah Data *Reseller Key* Benar**



**Gambar 3.30 Rancangan Layar Berhasil Ubah Data *Reseller***

1. **Rancangan Layar Halaman *Form* Ubah Data *Reseller Key* Salah**

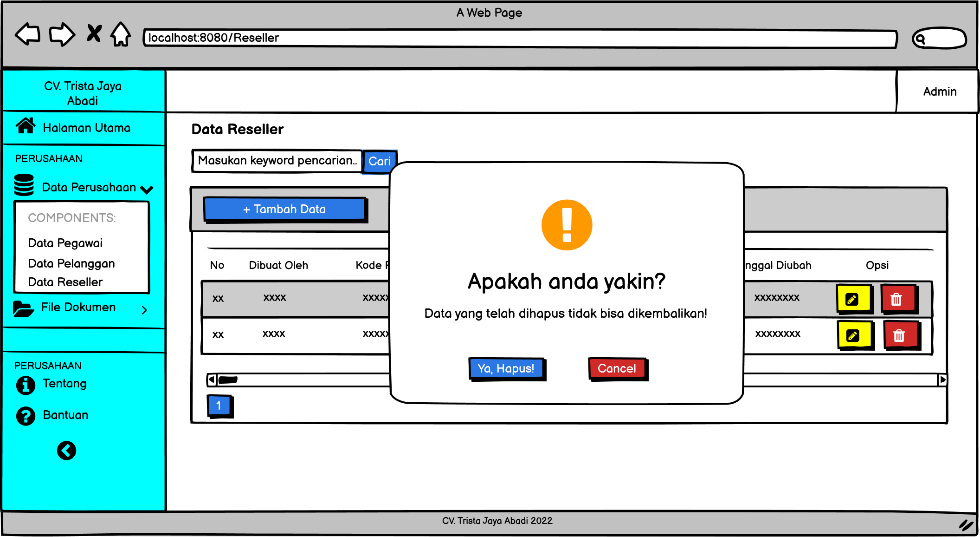
Pada halaman ini merupakan tampilan rancangan layar halaman *form* ubah data *reseller key* salah. Pada halaman *form* ini terdapat *field* yang berisikan kode *reseller*, nama *reseller*, alamat *reseller*, email *reseller*, no telepon, jumlah pesanan dan harga pesanan yang masih terenkripsi karena *key* yang dimasukan salah. Pada *form* ubah data *reseller*, hanya *field* yang bersifat data penting saja yang masih terenkripsi dan pada halaman *form* ini *user* tidak dapat melihat atau mengubah data selain kembali ke halaman data *reseller* karena di *form* ini tidak ada tombol batal dan ubah data. Rancangan layar halaman *form* ubah data *reseller key* salah dapat dilihat pada gambar 3.31 berikut:



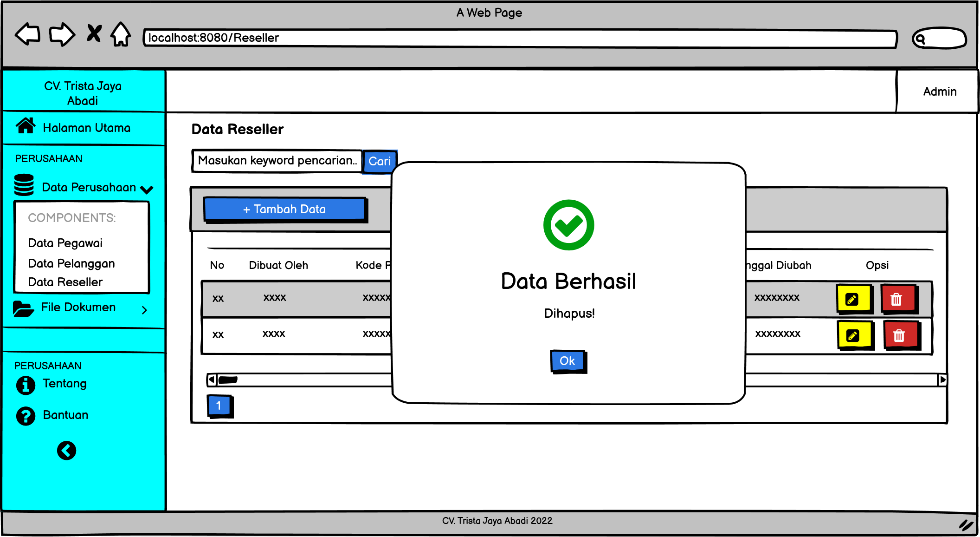
**Gambar 3.31 Rancangan Layar Halaman *Form* Ubah Data *Reseller Key* Salah**

1. **Rancangan Layar Hapus Data *Reseller***

Pada halaman ini merupakan tampilan rancangan layar hapus data *reseller*. Pada halaman data *reseller* jika user klik ikon hapus, maka akan tampil *pop up* yang berisikan apakah anda yakin. Jika *user* mengklik tombol hapus, maka data yang ingin dihapus akan langsung terhapus di dalam *database* dan tombol *cancel* untuk membatalkan menghapus data *reseller*. Rancangan layar hapus data *reseller* dapat dilihat pada gambar 3.32 dan gambar 3.33 dibawah ini:



**Gambar 3.32 Rancangan Layar Hapus Data *Reseller***



**Gambar 3.33 Rancangan Layar Berhasil Hapus Data *Reseller***

1. **Rancangan Layar Halaman *File***

Pada halaman ini merupakan tampilan rancangan layar halaman *file*. Pada halaman ini terdapat tombol enkripsi *file* yang digunakan untuk enkripsi *file* dengan cara menginput data *file*. Pada halaman ini terdapat sebuah tabel *file* yang berisikan data dari data *file*. Pada halaman ini terdapat tombol *download* untuk *download* *file* enkripsi dan dekripsi. Pada halaman ini terdapat tombol ubah data *file* untuk mengubah atau melihat data *file*. Selain itu terdapat tombol info untuk dekripsi *file* dan terdapat tombol hapus untuk menghapus data *file*. Rancangan layar halaman *file* dapat dilihat pada gambar 3.34 berikut:

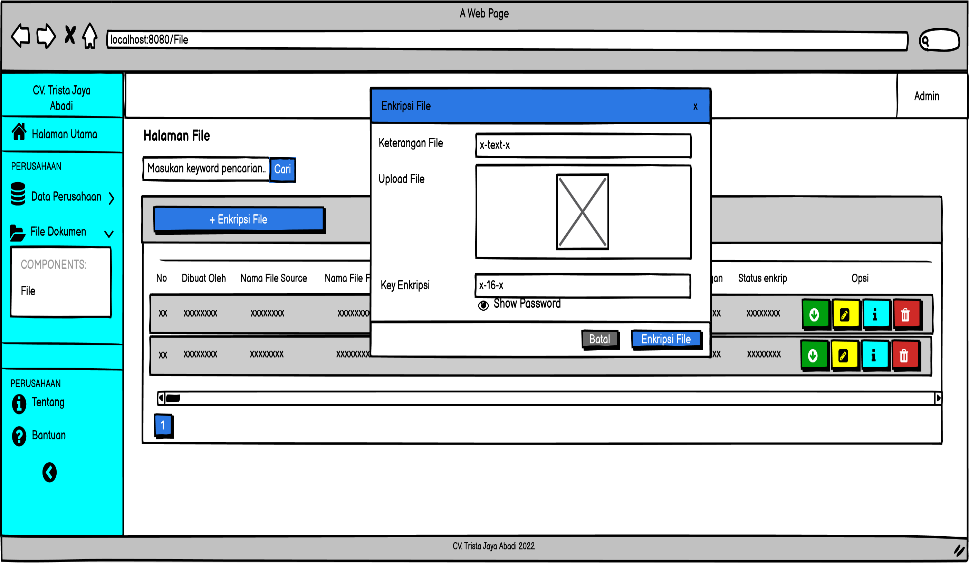
Sebuah gambar berisi teks

Deskripsi dibuat secara otomatis

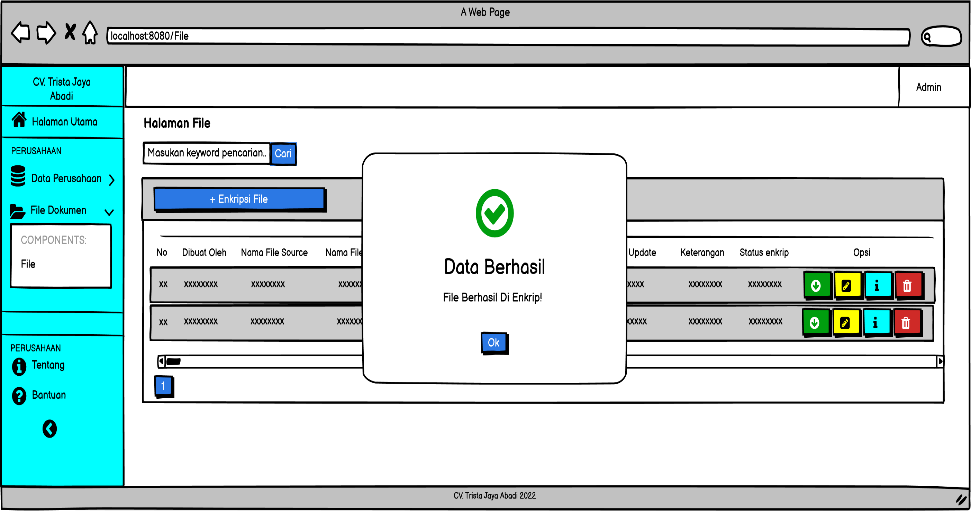
**Gambar 3.34 Rancangan Layar Halaman *File***

1. **Rancangan Layar *Form* Enkripsi *File***

Pada halaman ini merupakan tampilan rancangan layar *form* enkripsi *file*. Pada *form* ini terdapat *field* yang berisikan keterangan *file*, *upload* *file* dan *key* enkripsi yang harus diisi oleh *user* jika ingin enkripsi *file*. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini juga terdapat tombol enkripsi *file* yang digunakan untuk enkripsi *file* dan menambah data *file* yang telah diisi oleh *user* ke dalam *database* serta menyimpan *file* enkripsi ke dalam penyimpanan. Selain itu ketika *user* mengklik tombol enkripsi *file* maka akan tampil *pop up* berhasil enkripsi *file* seperti gambar 3.36. Pada *form* ini juga terdapat tombol batal yang digunakan untuk membatalkan enkripsi *file* dan menutup *form* enkripsi *file*. Rancangan layar *form* enkripsi *file* dapat dilihat pada gambar 3.35 berikut:



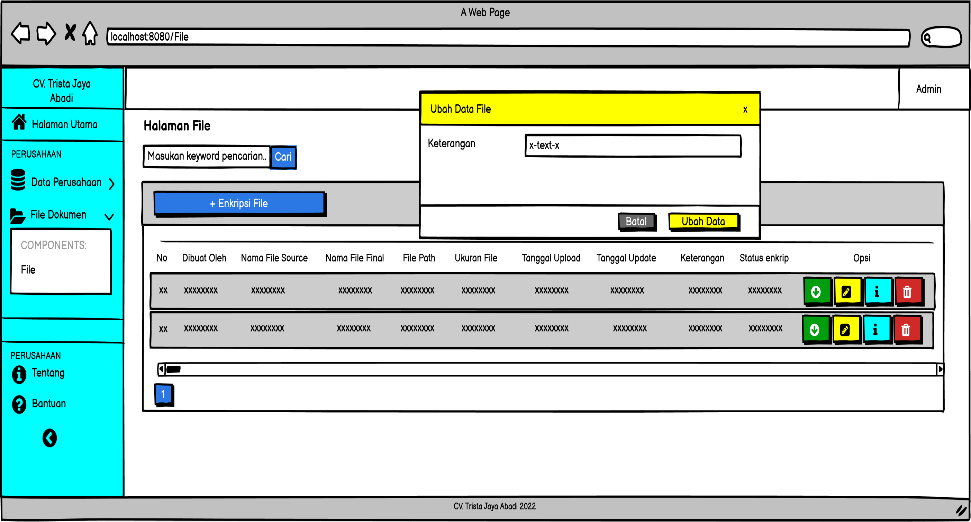
**Gambar 3.35 Rancangan Layar *Form* Enkripsi *File***



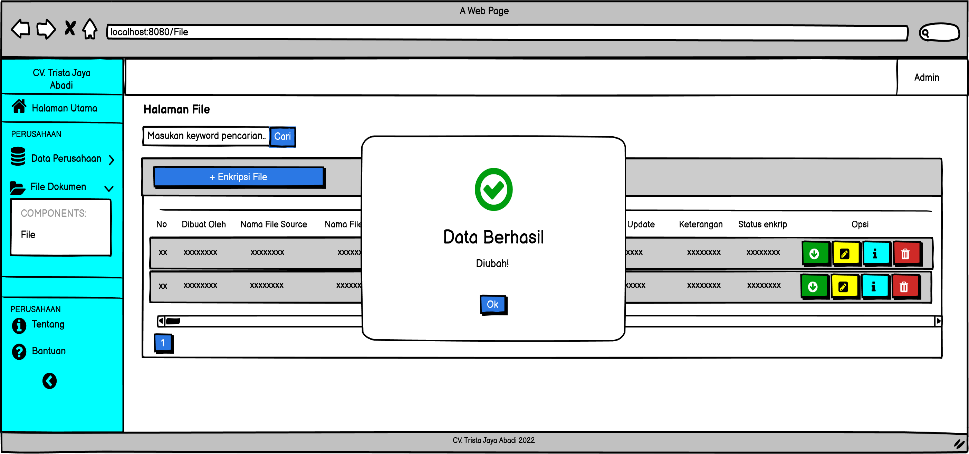
**Gambar 3.36 Rancangan Layar Berhasil Enkripsi *File***

1. **Rancangan Layar *Form* Ubah Data *File***

Pada halaman ini merupakan tampilan rancangan layar *form* ubah data *file*. Pada *form* ini terdapat *field* yang berisikan keterangan yang *user* bisa mengubah atau tidak mengubah data yang sudah ada di dalam *field* ubah data *file*. Pada *form* ini terdapat tombol ubah data yang digunakan untuk mengubah data *file* yang sudah diubah oleh *user* dan ketika *user* klik tombol ubah data maka akan tampil *pop up* berhasil ubah data seperti gambar 3.38. Selain itu terdapat tombol batal yang digunakan untuk membatalkan mengubah data *file* dan menutup *form* ubah data *file*. Rancangan layar *form* ubah data *file* dapat dilihat pada gambar 3.37 berikut:



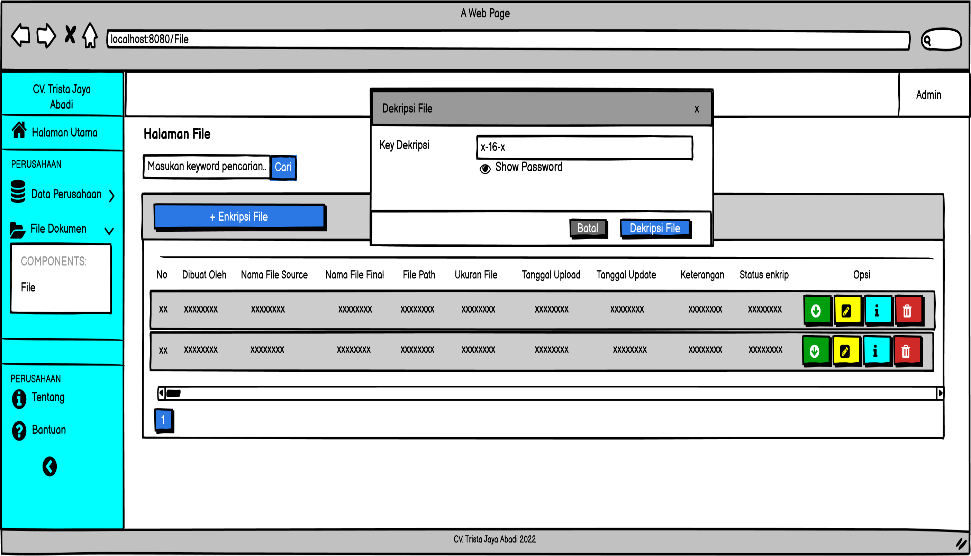
**Gambar 3.37 Rancangan Layar *Form* Ubah Data *File***



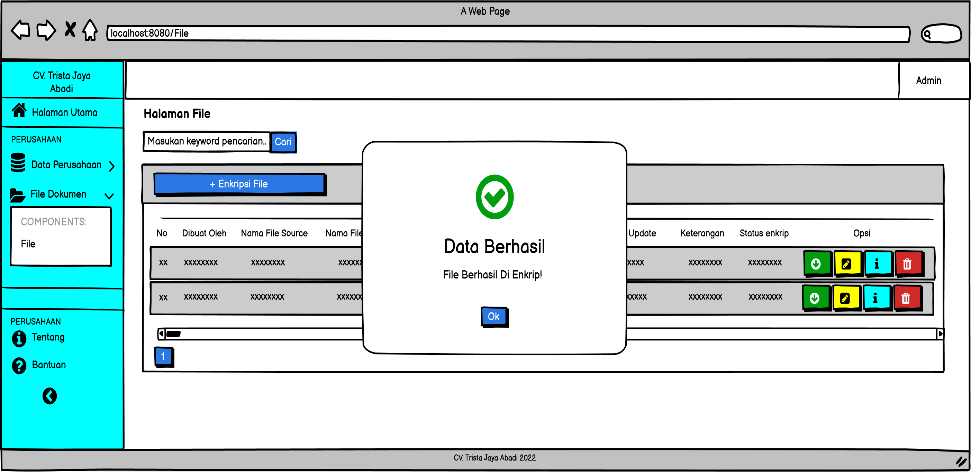
**Gambar 3.38 Rancangan Layar Berhasil ubah Data *File***

1. **Rancangan Layar *Form* Dekripsi *File***

Pada halaman ini merupakan tampilan rancangan layar *form* dekripsi *file*. Pada *form* ini terdapat *field* yang berisikan *key* dekripsi yang harus diisi jika *user* ingin mendekripsi *file*. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini juga terdapat tombol dekripsi *file* untuk dekripsi *file* jika *key* dekripsi yang dimasukan benar dan ketika *user* klik dekripsi file maka akan tampil *pop up* berhasil dekripsi *file* seperti gambar 3.40. Selain itu terdapat tombol batal yang digunakan untuk membatalkan dekripsi *file*. Rancangan layar *form* dekripsi *file* dapat dilihat pada gambar 3.39 berikut:



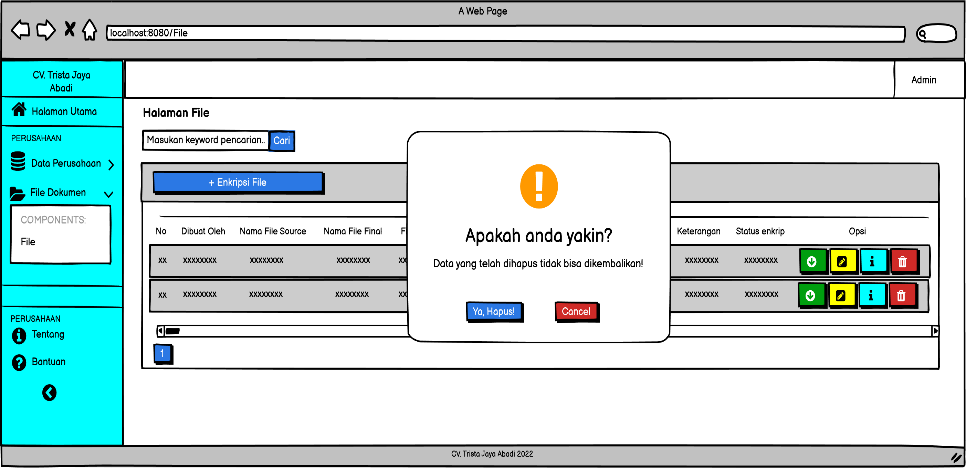
**Gambar 3.39 Rancangan Layar *Form* Dekripsi *File***



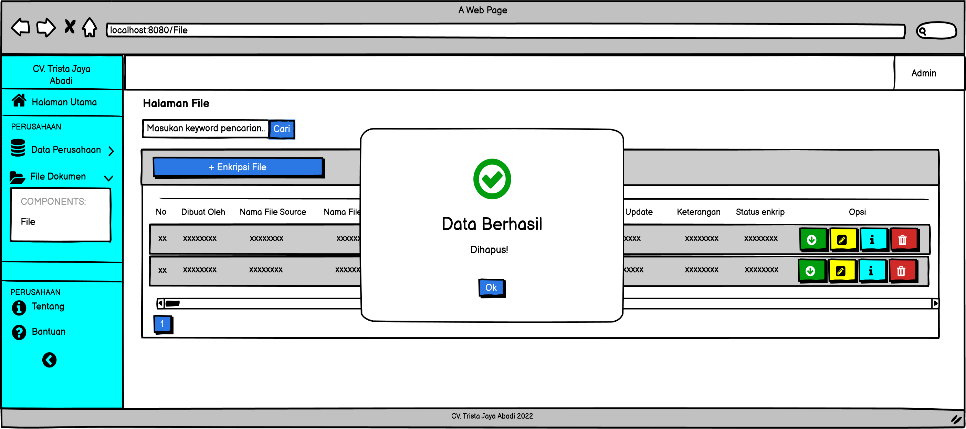
**Gambar 3.40 Rancangan Layar Berhasil Dekripsi *File***

1. **Rancangan Layar Hapus Data *File***

Pada halaman ini merupakan tampilan rancangan layar hapus data *file*. Pada halaman *file* jika user klik ikon hapus, maka akan tampil *pop up* yang berisikan apakah anda yakin. Jika user mengklik tombol hapus, maka data yang ingin dihapus akan langsung terhapus di dalam *database* dan penyimpanan file serta tombol *cancel* untuk membatalkan menghapus data *file*. Rancangan layar hapus data *file* dapat dilihat pada gambar 3.41 dan gambar 3.42 dibawah ini:



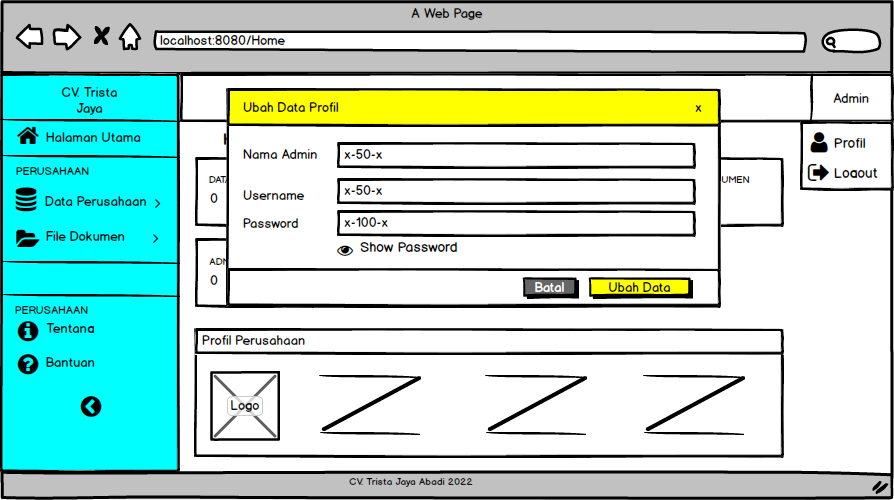
**Gambar 3.41 Rancangan Layar Hapus Data *File***



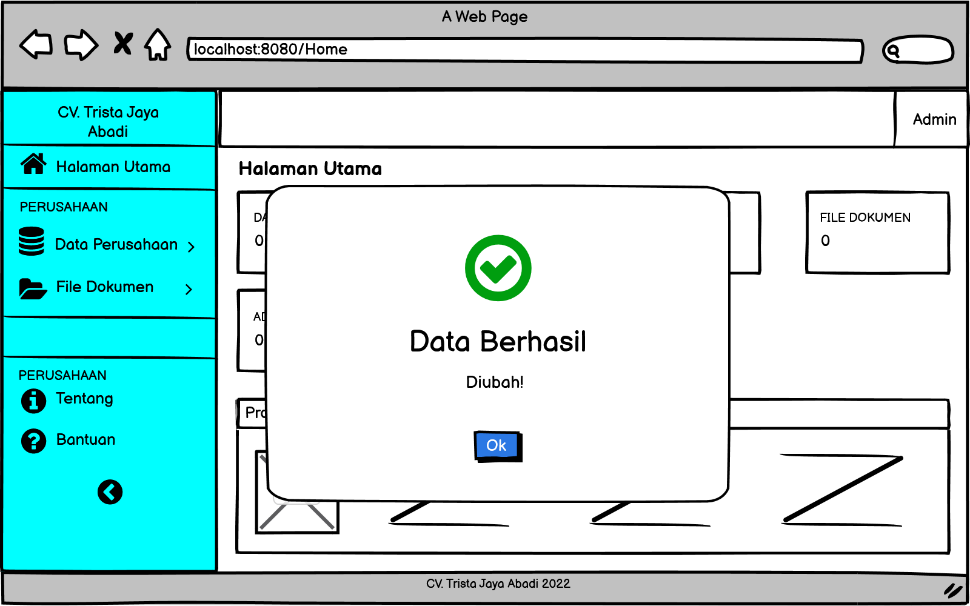
**Gambar 3.42 Rancangan Layar Berhasil Hapus Data *File***

1. **Rancangan Layar *Form* Ubah Data Profil**

Pada halaman ini merupakan tampilan rancangan layar *form* ubah data profil. Pada *form* ini terdapat *field* yang berisikan nama admin, *username* dan *password*. Untuk dibagian *password*, jika *user* ingin mengubah *password*, maka isi *password* baru di *field* *password* dan jika tidak ingin mengubah *password* untuk *field* *password* dikosongkan saja. Pada *form* ini terdapat *show password* untuk melihat *password* yang diinputkan sudah benar atau belum. Pada *form* ini terdapat tombol ubah data yang digunakan untuk mengubah atau melihat data profil dan ketika *user* klik ubah data maka akan muncul *pop up* data berhasil di ubah seperti gambar 3.44. Selain itu tombol batal digunakan untuk membatalkan mengubah data. Rancangan layar *form* ubah data profil dapat dilihat pada gambar 3.43 berikut:



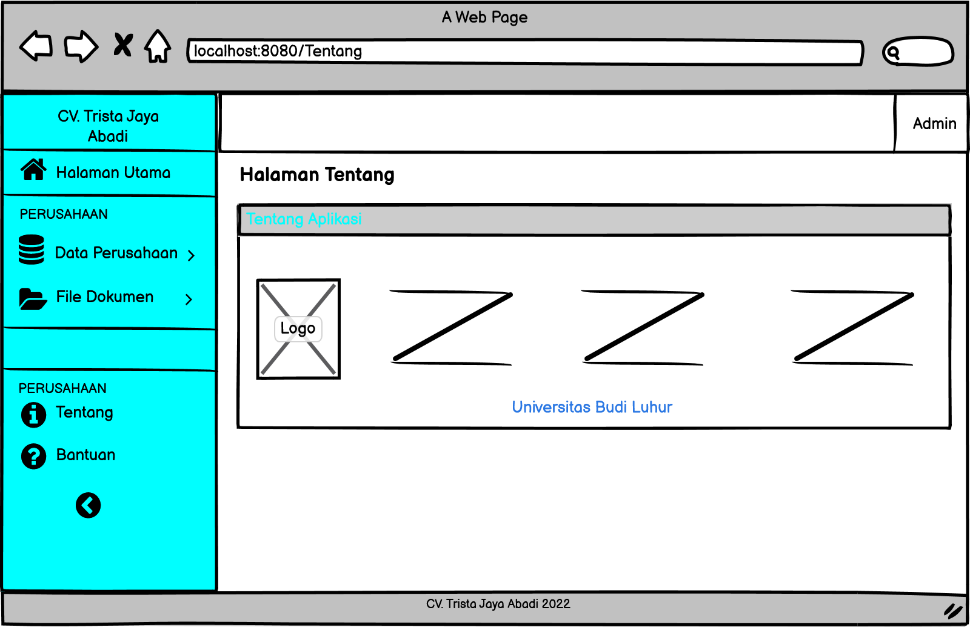
**Gambar 3.43 Rancangan Layar *Form* Ubah Data Profil**



**Gambar 3.44 Rancangan Layar Berhasil Ubah Data Profil**

1. **Rancangan Layar Halaman Tentang**

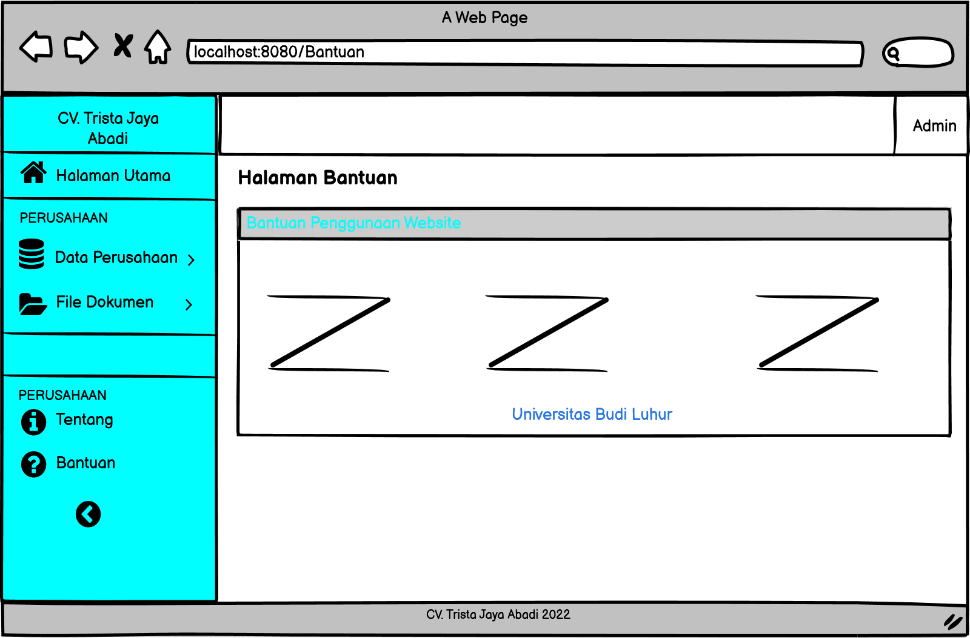
Pada halaman ini merupakan tampilan rancangan layar tentang aplikasi yang dibuat. Pada halaman ini terdapat penjelasan tentang aplikasi dan tujuan dibuatnya aplikasi. Rancangan layar halaman tentang dapat dilihat pada gambar 3.45 berikut:



**Gambar 3.45 Rancangan Layar Halaman Tentang**

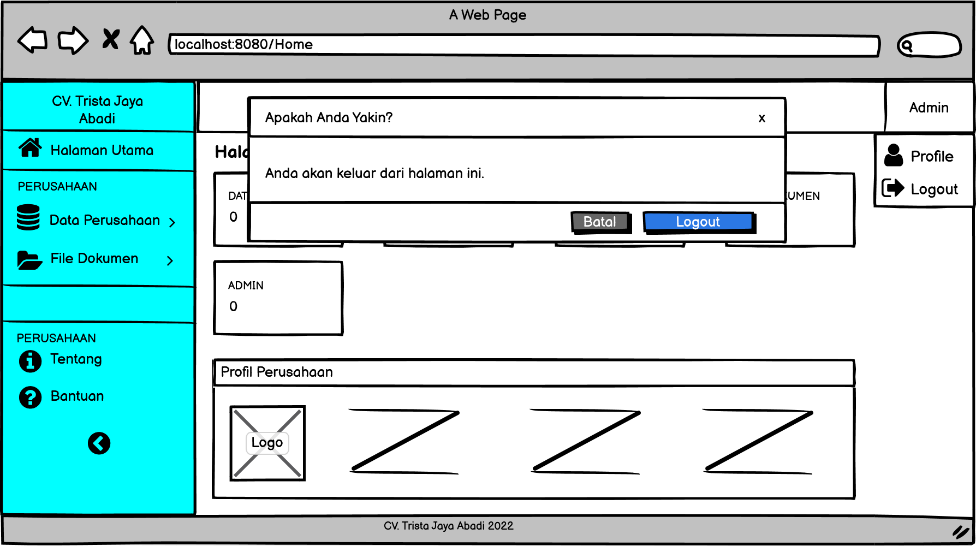
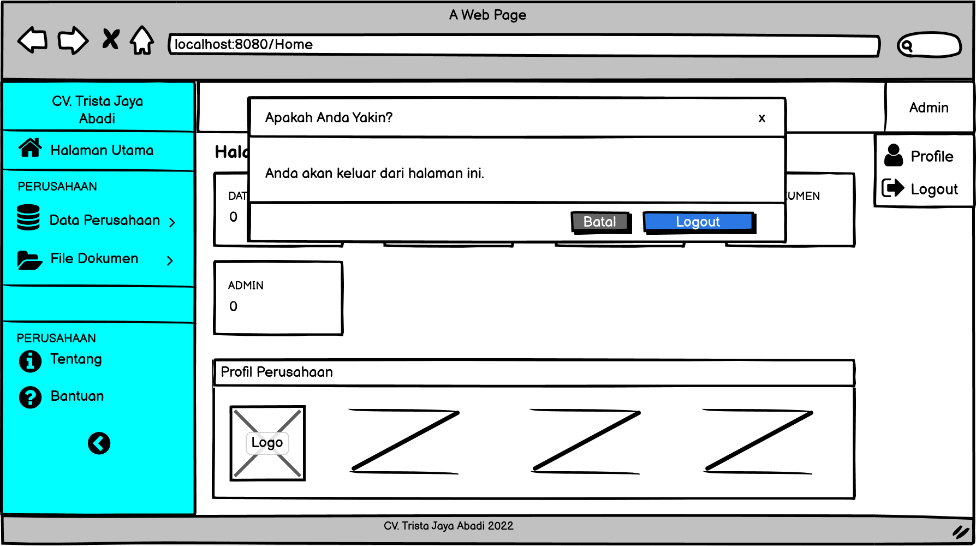
1. **Rancangan Layar Halaman Bantuan**

Pada halaman ini merupakan tampilan rancangan layar bantuan penggunaan aplikasi. Pada halaman ini terdapat penjelasan cara menggunakan aplikasi ini dengan benar. Rancangan layar halaman bantuan dapat dilihat pada gambar 3.46 berikut:



**Gambar 3.46 Rancangan Layar Halaman Bantuan**

1. **Rancangan Layar *Logout***

 Pada rancangan layar *logout* ini user dapat melakukan *logout* dari aplikasi. Dengan klik nama *username* di sebelah kanan paling atas maka akan muncul menu profil dan *logout*, setelah klik ikon *logout*, maka akan menampilkan dialog yang berisikan apakah anda yakin. Jika *user* mengklik *logout*, maka *user* langsung keluar dari aplikasi dan akan dikembalikan ke halaman *login*. Selain itu ketika *user* klik batal maka akan membatalkan proses *logout* dari aplikasi. Rancangan layar *logout* dapat dilihat pada gambar 3.47 berikut:

**Gambar 3.47 Rancangan Layar *Logout***