

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di PT Cakra Tekno Nusantara, penulis memegang jabatan sebagai *Developer Trainee* di divisi *IT Developer*, dengan supervisi dari Angga Wirapraditya B.Sc, selaku *Chief Technology Officer* (CTO). Instruksi, pengarahan, serta diskusi dilakukan melalui aplikasi *Whatsapp* dan *Google meets*.

3.2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang, tanggung jawab yang diberikan adalah sebagai berikut.

- Mempelajari penggunaan studio <https://demo.cakra.ai/>.
- Membuat rancangan percakapan dan mempelajari *use case* pelaporan masyarakat dari *call center 110 POLRI* lewat *Ms. Excel*.
- Merancang sistem API dan website.
- Mengkoneksikan API dengan *database PostgreSQL*.
- Membuat dan merancang API yang dapat menyaring nama dan kategori kasus tindakan kriminalitas berdasarkan input.
- Membuat *Back-End* web API yang berfungsi untuk melakukan pengujian serta mengontrol data API nama dan kategori.

3.3 Uraian Kerja Magang

Proses kerja magang berlangsung 17 minggu. Selama kerja magang tersebut, *progress* pekerjaan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Uraian Kerja Magang

MINGGU KE	JENIS PEKERJAAN YANG DILAKUKAN MAHASISWA
1	Pengenalan Membuat conversational flow dari use case pelaporan Pengenalan function yang digunakan
2	Membuat conversational flow Memahami konsep decision pada conversational flow Membuat function
3	Integrasi function dengan <i>Database</i> Membuat function Optik Melawai Membuat conversational flow
4	Update kategori BOT Gisa Membuat function Membuat conversational flow
5	Membuat API nama dan kategori Membuat API CRUD <i>database</i> Membuat website untuk mengatur dan testing API
6	Membuat website untuk mengatur dan testing API Membuat endpoint website Troubleshooting website
7	Membuat backend website Memindahkan website ke CodeIgniter 4 Troubleshooting Membuat dynamic role base login
8	Membuat dynamic role base login Membuat backend page website Training meeting
9	Membuat backend page website Membuat speech recognition Evaluasi
10	Membuat speech recognition by upload Auto convert audio dengan ffmpeg Mencari speech recognition long duration

Tabel 3.1 Uraian Kerja Magang (Lanjutan)

MINGGU KE	JENIS PEKERJAAN YANG DILAKUKAN MAHASISWA
11	Auto convert audio dengan ffmpeg Koneksi hasil transkrip ke api BOT
12	Koneksi hasil transkrip ke api BOT cakra.ai Membuat <i>chatbot</i> di platform telegram dengan Telebgram API
13	Mencari dan Mencoba Telebgram API Membuat Telegram <i>chatbot</i> Integrasi Telegram API ke Telebgram <i>chatbot</i>
14	Membuat Telegram <i>chatbot</i> Enhancement Telegram <i>chatbot</i>
15	Enhancement Telegram <i>chatbot</i> Mempelajari dan mencari mengenai BotMan
16	Implementasi BotMan Troubleshooting BotMan Mengkoneksikan BotMan dengan API Cakra.ai
17	Enhancement dan Troubleshooting BotMan yang terkoneksi dengan API Cakra.ai Membersihkan dokumen

Kerja magang terbagi dalam dua tahap, yaitu perancangan dan implementasi, dengan penjelasan sebagai berikut.

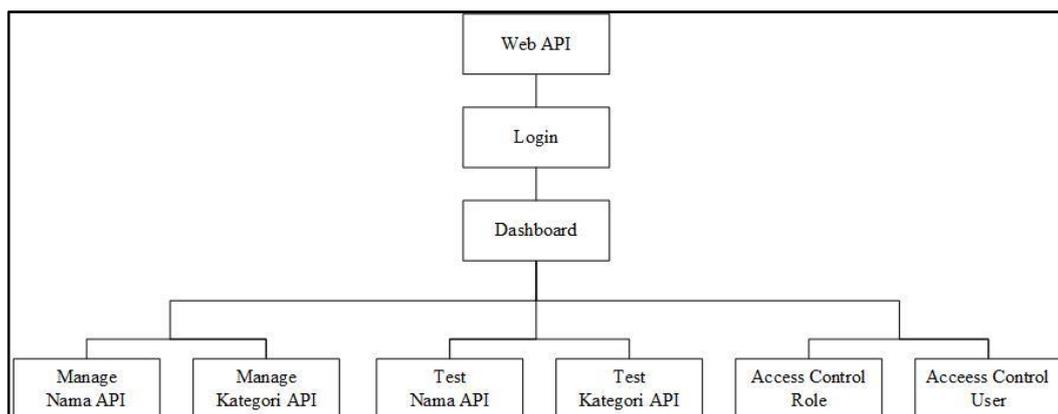
3.3.1 Perancangan Aplikasi

Pada tahap perancangan, dibangun *sitemap* dan *flowchart* yang akan digunakan sebagai acuan pada tahap implementasi.

A. Sitemap

Sitemap merupakan suatu diagram yang menyediakan informasi secara detail perihal seluruh halaman dan konten yang terdapat pada website^[2].

Berikut *sitemap* yang digunakan dalam pembuatan website API untuk *Virtual Assistant* di PT Cakra Tekno Nusantara.



Gambar 3.1 *Sitemap* Website API

Dari Gambar 3.1 dapat dilihat bahwa, website API akan memiliki delapan halaman dengan rincian sebagai berikut.

1. Halaman *Login*

Halaman *login* merupakan halaman utama yang akan memproses login sebelum melanjutkan ke halaman *dashboard*.

2. Halaman *Dashboard*

Halaman *dashboard* akan menampilkan statistik dari API, user, dan *database* yang tersedia. Statistik akan ditampilkan dalam bentuk *chart* dan angka. Selain itu, terdapat juga *side* menu yang dapat digunakan untuk mengakses halaman lainnya.

3. Halaman *Manage Nama API*

Halaman *Manage Nama API* merupakan halaman yang berguna untuk melihat, menambahkan, mengubah, serta menghapus data yang ada.

4. Halaman *Manage Kategori API*

Halaman *Manage Kategori API* merupakan halaman yang berguna untuk melihat, menambahkan, mengubah, serta menghapus data yang ada.

5. Halaman *Test* Nama API

Halaman *Test* Nama API memiliki fungsi untuk melakukan percobaan terhadap Nama API.

6. Halaman *Test* Kategori API

Halaman *Test* Kategori API memiliki fungsi untuk melakukan percobaan terhadap Kategori API.

7. Halaman *Access Control Role*

Halaman *Access Control Role* adalah halaman yang berfungsi untuk menampilkan tabel *role* yang tersedia pada website ini.

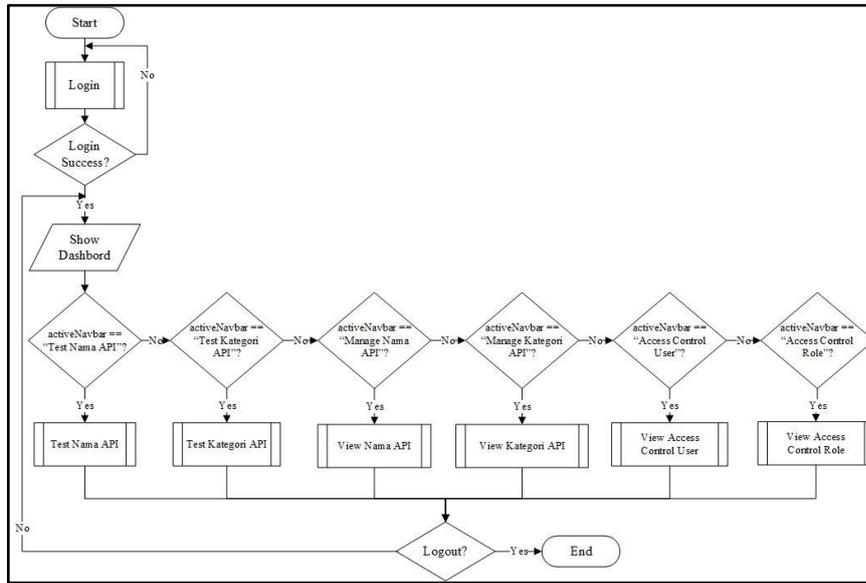
8. Halaman *Access Control User*

Halaman *Access Control User* adalah halaman yang berfungsi untuk menampilkan tabel user yang terdaftar pada *database*. Halaman ini juga memiliki fitur untuk mengubah data maupun menghapus user yang terdaftar.

B. Flowchart

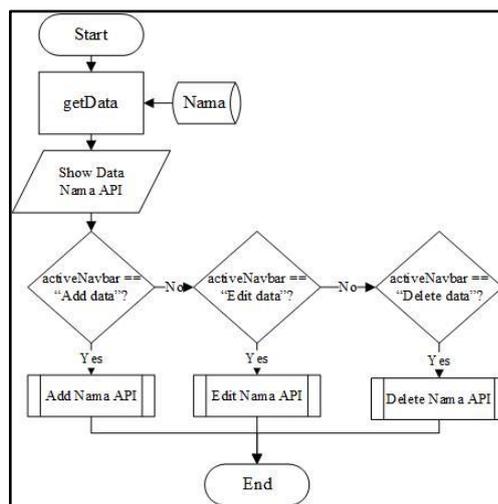
Flowchart dapat diartikan sebagai suatu alat atau sarana yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus^[3]. Berikut beberapa *flowchart* yang digunakan dalam pembuatan API untuk *Virtual Assistant* di PT Cakra Tekno Nusantara.

1. Flowchart View



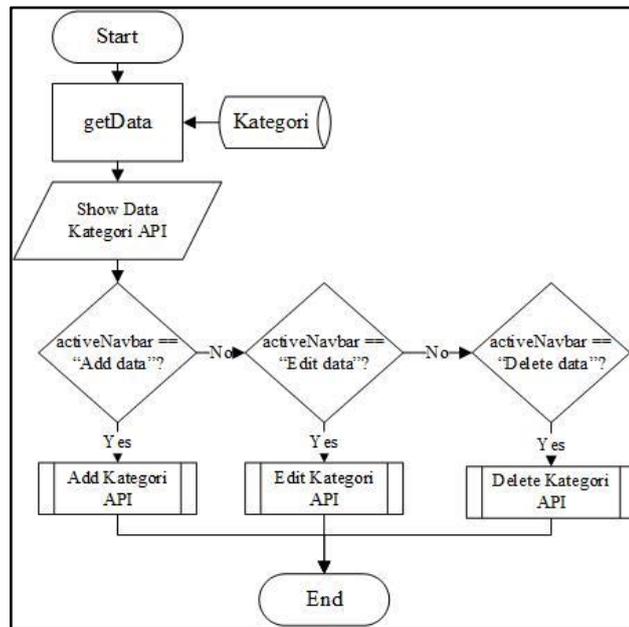
Gambar 3.2 Flowchart View Dashboard

Gambar 3.2 merupakan *flowchart view* untuk halaman *dashboard*. Pada halaman *dashboard*, akan terdapat menu untuk mengakses ke halaman lainnya. Seperti, Test Nama API, Test Kategori API, *Manage* Nama API, *Manage* Kategori API, halaman *Access Control* User, dan halaman *Access Control* Role.



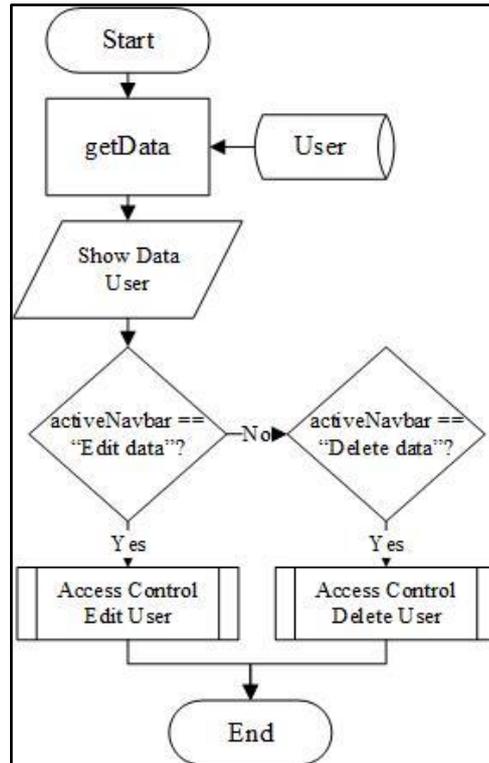
Gambar 3.3 Flowchart View Nama API

Gambar 3.3 merupakan *flowchart view* untuk halaman Nama API. *Flowchart* ini akan menampilkan data *noise* dari Nama API yang ada di *database*. Dari halaman ini, akan terdapat menu untuk mengakses ke halaman *Add Nama API*, *Edit Nama API*, dan *Delete Nama API*.



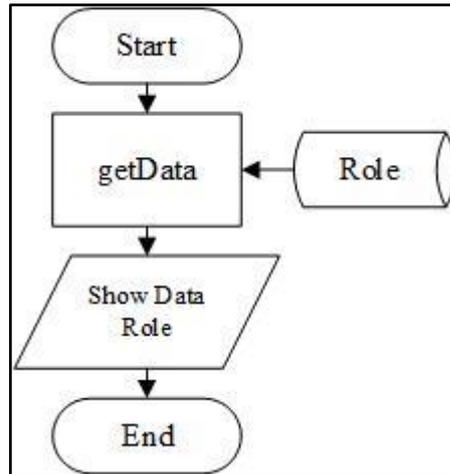
Gambar 3.4 *Flowchart View* Kategori API

Gambar 3.4 merupakan *flowchart view* untuk halaman Kategori API. *Flowchart* ini akan menampilkan data *noise* dari Kategori API yang ada di *database*. Dari halaman ini, akan terdapat menu untuk mengakses ke halaman *Add Kategori API*, *Edit Kategori API*, dan *Delete Kategori API*.



Gambar 3.5 *Flowchart View Access Control User*

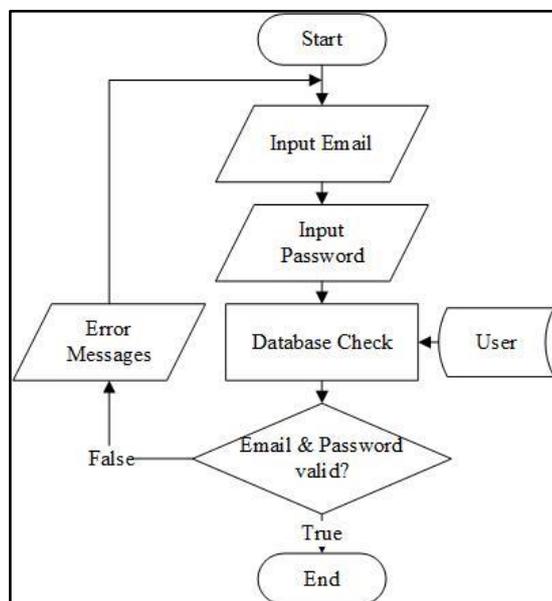
Gambar 3.5 merupakan *flowchart view* untuk halaman *Access Control*. *Flowchart* ini akan menampilkan user yang terdaftar pada *database*. Dari halaman ini, akan terdapat menu untuk mengakses ke halaman *Edit User* dan *Delete User*.



Gambar 3.6 *Flowchart View Access Control Role*

Gambar 3.6 merupakan *flowchart view* untuk halaman *Access Control Role*. *Flowchart* ini akan menampilkan daftar *role* yang ada pada *database*.

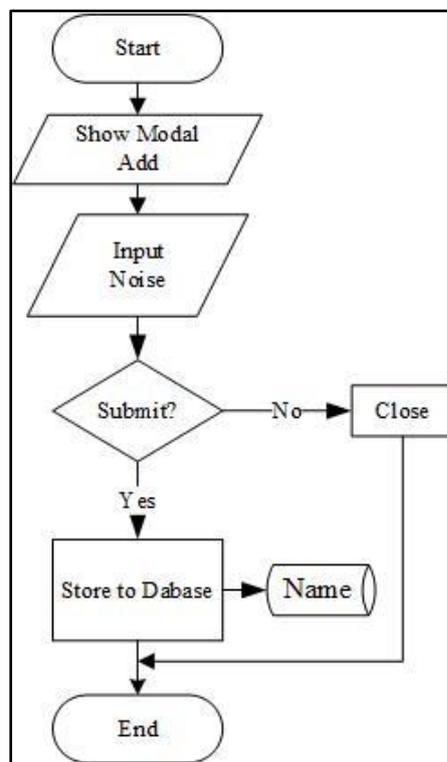
2. *Flowchart Login*



Gambar 3.7 *Flowchart Login*

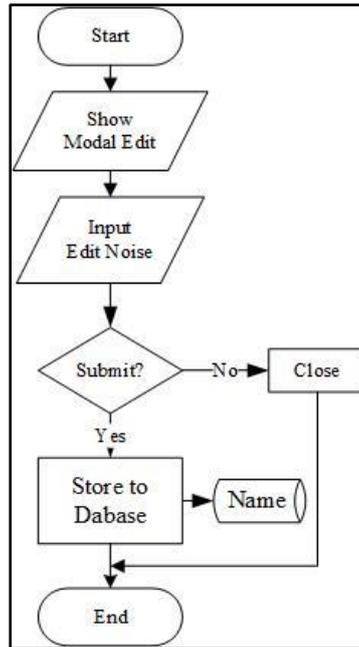
Dari Gambar 3.7 akan terdapat 2 *field input*. Untuk dapat melakukan *login*, user harus menggunakan *email* dan *password* yang telah terdaftar pada *database* CakraDB. Bila *email* atau *password* yang diisi tidak terdaftar atau salah, maka akan tampil suatu *error messages* dan user tidak dapat melanjutkan ke halaman berikutnya.

3. Flowchart Manage Nama API



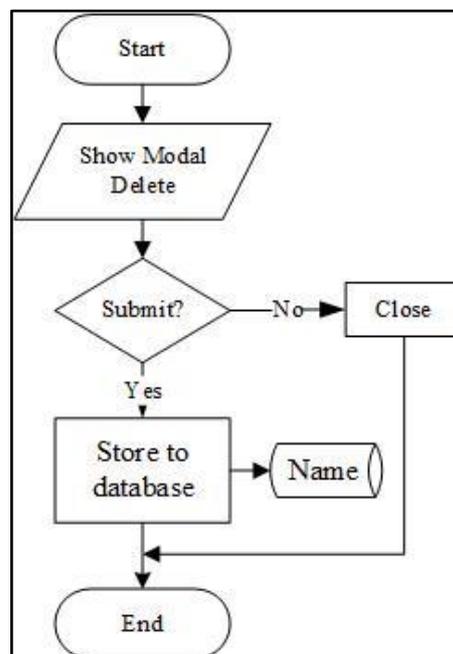
Gambar 3.8 Flowchart Add Nama API

Dari Gambar 3.8 akan terdapat suatu modal yang memiliki *field input noise*. Setelah *noise diinput*, selanjutnya data tersebut akan disimpan ke dalam *database* CakraDB.



Gambar 3.9 *Flowchart Edit Nama API*

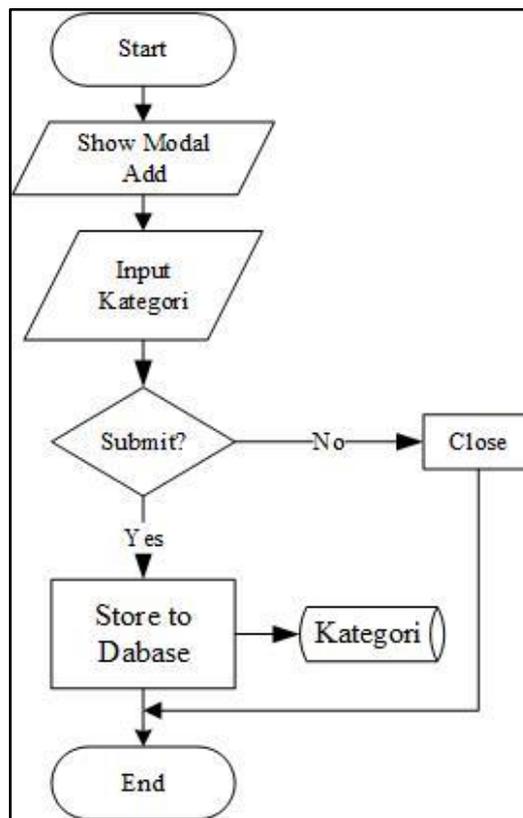
Dari Gambar 3.9 akan terdapat suatu modal yang memiliki *field input* yang sudah terisi dengan data *noise* sebelumnya. Setelah *noise* diubah, data tersebut kemudian akan disimpan ke dalam *database* CakraDB.



Gambar 3.10 *Flowchart Delete Nama API*

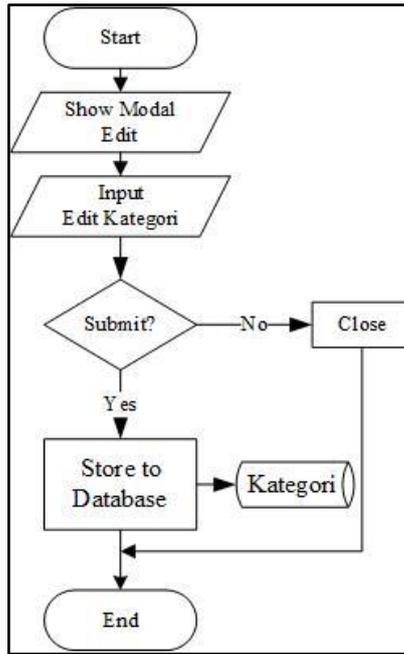
Dari Gambar 3.10 akan terdapat suatu modal yang memiliki 2 *button* yaitu, *submit* dan *cancel*. *Button submit* akan berfungsi untuk menghapus data *noise* dari *database* CakraDB. Sedangkan, *button cancel* akan berfungsi untuk membatalkan proses penghapusan.

4. Flowchart Manage Kategori API



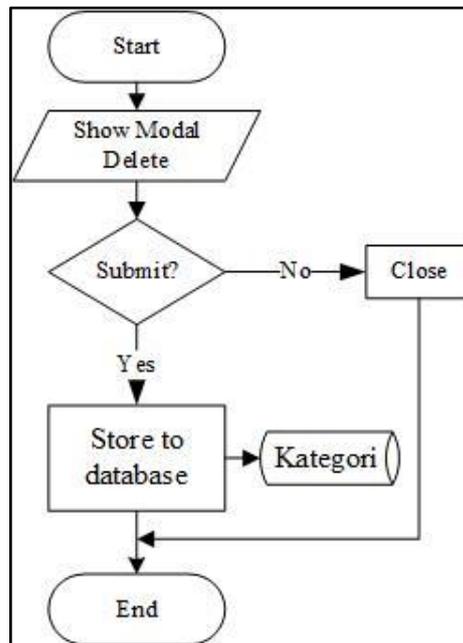
Gambar 3.11 Flowchart Add Kategori API

Dari Gambar 3.11 akan terdapat suatu modal yang memiliki *field input* kategori *noise*. Setelah kategori *noise* diinput, selanjutnya data tersebut akan disimpan ke dalam *database* CakraDB.



Gambar 3.12 *Flowchart Edit Kategori API*

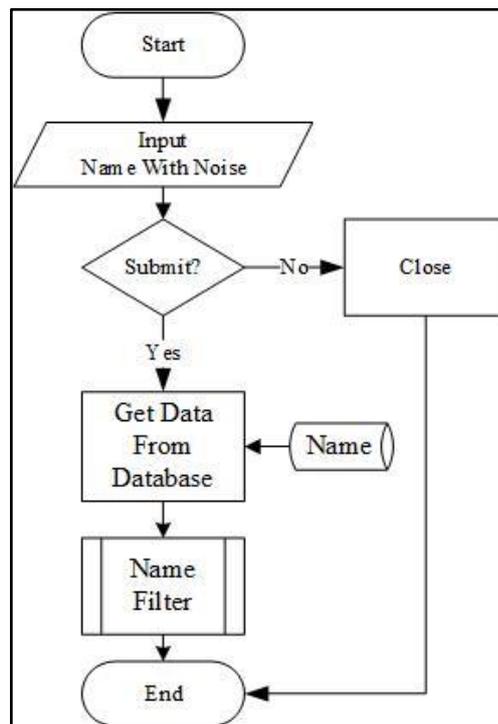
Dari Gambar 3.12 akan terdapat suatu modal yang memiliki *field input* yang sudah terisi dengan data kategori *noise* sebelumnya. Setelah *noise* tersebut diperbaiki, data tersebut akan disimpan ke dalam *database* CakraDB.



Gambar 3.13 *Flowchart Delete Kategori API*

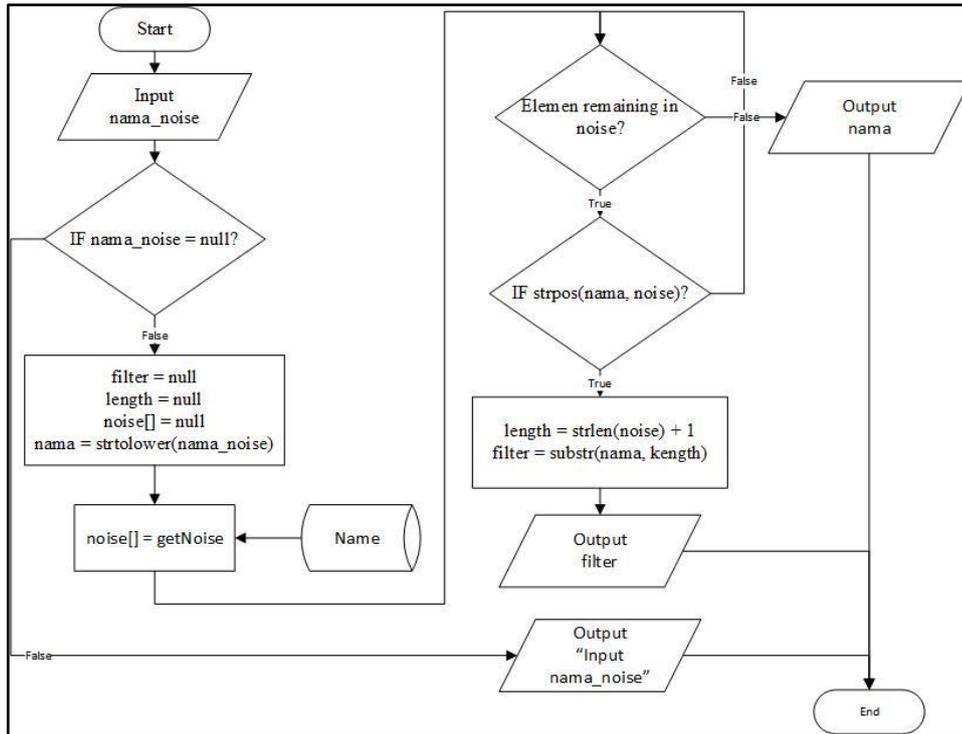
Dari Gambar 3.13 akan terdapat suatu modal yang memiliki 2 *button* yaitu, *submit* dan *cancel*. *Button submit* berfungsi untuk menghapus data kategori *noise* dari *database* CakraDB. Sedangkan, *button cancel* akan berfungsi untuk membatalkan proses penghapusan.

5. Flowchart Testing Nama API



Gambar 3.14 Flowchart Test Nama API

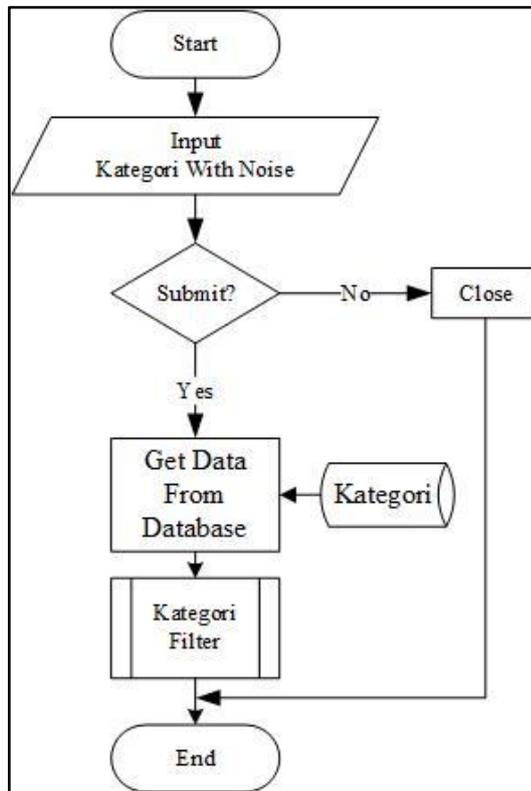
Dari *flowchart testing* Nama API yang dapat dilihat pada Gambar 3.14 akan terdapat suatu *field input*. User yang akan menggunakan Nama API harus *menginput* nama dengan *noise*. Kemudian, data yang mengandung *noise* tersebut akan diproses pada *modul* Name Filter. Flowchart dari *modul* Name Filter dapat dilihat dibawah ini.



Gambar 3.15 *Flowchart* Filter Nama API

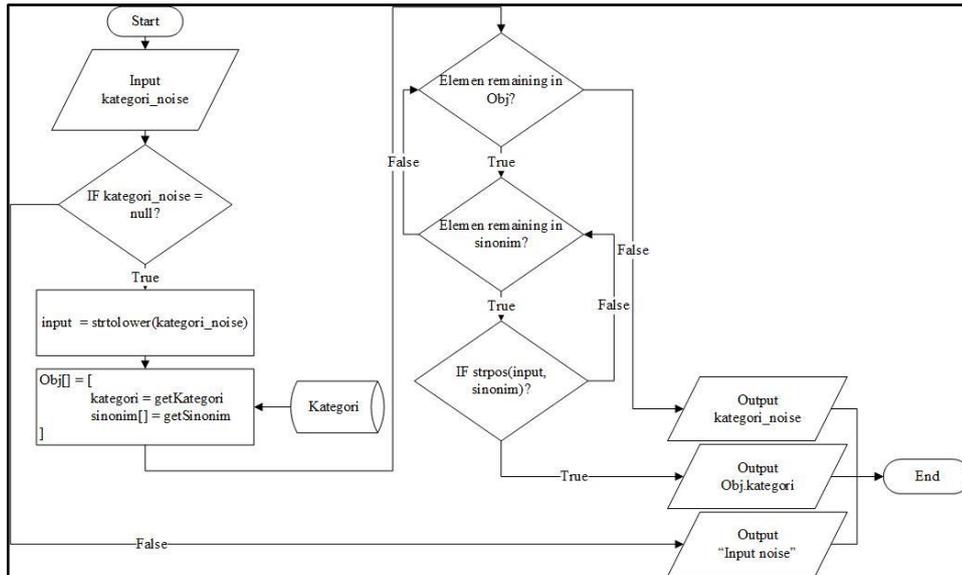
API ini memiliki *requirement* untuk memfilter *noise* yang terdapat pada *input* nama, sehingga *output* yang dihasilkan hanya berupa nama. Sebagai contoh, jika user memasukkan kata “Nama saya Andreas Steven” maka hasil yang didapat adalah “Andreas Steven”. API ini akan mengambil data *noise* dari *database* yang tersedia. Kemudian, API akan membandingkan *noise* tersebut dengan *noise* dari *input* user. Bila ditemukan kesamaan, maka API akan menghapus *noise* tersebut. Sehingga, *output* yang didapatkan hanya berupa nama. Jika antara *noise database* dan *input noise* tidak ditemukan kesamaan. API akan memberikan *output* sesuai dengan *input* tanpa adanya proses.

6. Flowchart Testing Kategori API



Gambar 3.16 Flowchart Test Kategori API

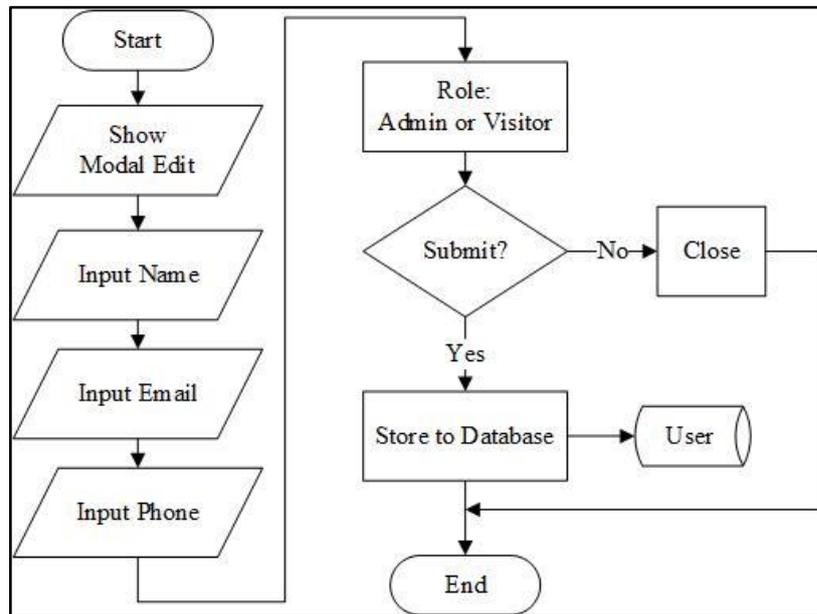
Dari *flowchart* testing kategori API yang dapat dilihat pada Gambar 3.16, akan terdapat suatu *field input*. User yang akan menggunakan Kategori API harus *menginput* kategori dengan *noise*. Kemudian, data yang mengandung *noise* tersebut akan diproses pada *modul* Kategori Filter. *Flowchart* dari *modul* Kategori Filter dapat dilihat di bawah ini.



Gambar 3.17 *Flowchart* Filter Kategori API

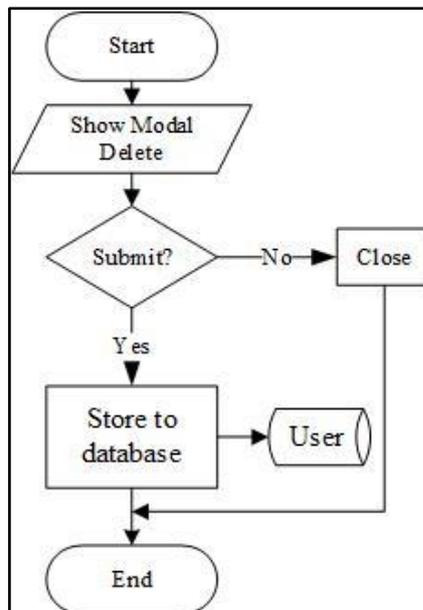
API ini memiliki *requirement* untuk memfilter *noise* yang terdapat pada *input* kategori tindakan kriminalitas yang diberikan oleh user. Sehingga, *output* yang dihasilkan hanya berupa kategori tindak kriminal. Sebagai contoh, jika user memasukkan kata “Ada kasus pembunuhan” maka hasil yang didapat adalah “Pembunuhan”. API ini akan membandingkan *noise* yang terdapat pada *input* dengan *noise* yang terdapat pada *database* yang telah dibuat. Jika antara *noise* yang terdapat pada *input* dengan *noise* yang terdapat pada *database* ditemukan kesamaan. Maka, API akan mengambil kategori yang sesuai. Jika kategori tidak ditemukan, API akan memberikan *output* sesuai dengan *input* tanpa adanya proses.

7. Flowchart Access Control



Gambar 3.18 Flowchart Access Control Edit User

Flowchart ini memiliki requirement untuk mengubah data user yang terdaftar pada database. Data yang diubah antara lain adalah Nama, Email, Phone Number, dan Role.



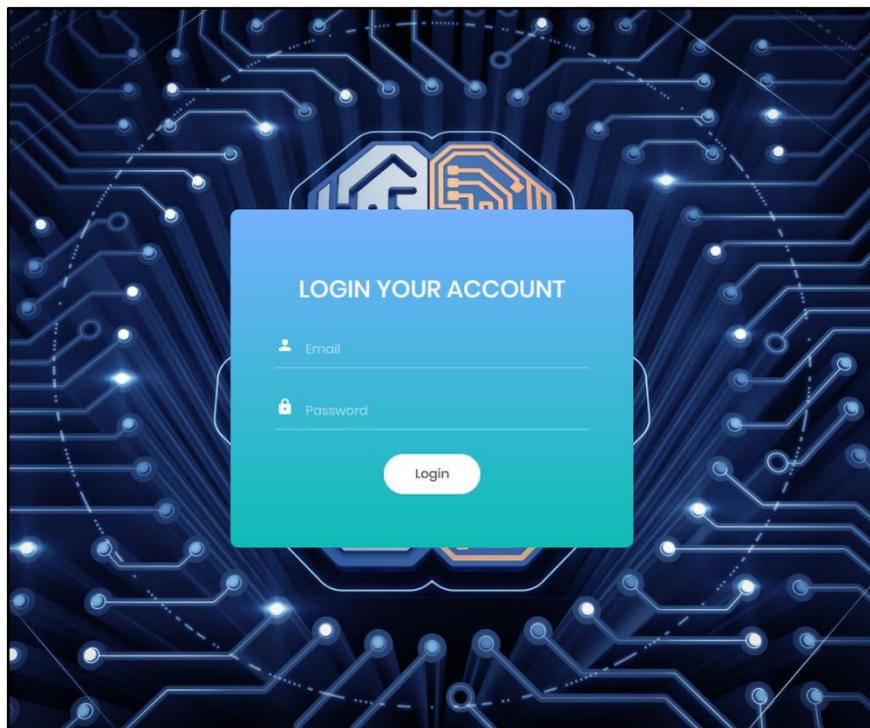
Gambar 3.19 Flowchart Access Control Delete User

Flowchart ini memiliki *requirement* untuk menghapus user yang terdaftar pada *database*. Sedangkan, untuk melakukan penambahan user, dapat dilakukan secara langsung melalui *database*.

3.3.2 Hasil dan Implementasi

Dari *flowchart* yang sudah diberikan, direalisasikan lah *flowchart* tersebut ke dalam suatu website.

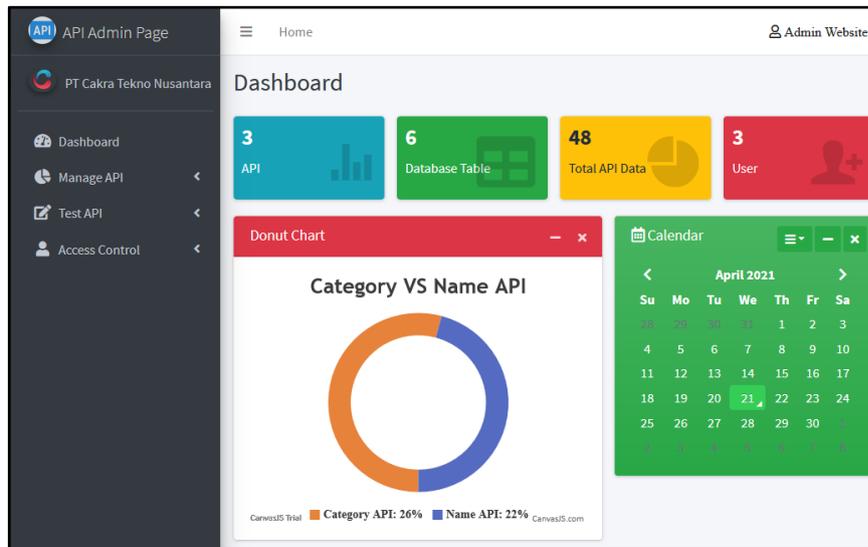
A. Halaman *Login*



Gambar 3.20 Halaman *Login*

Halaman *login* merupakan halaman pertama saat user mengakses website ini. Pada halaman ini, user akan diminta untuk *meninput email* dan *password*. Jika *email* dan *password* yang *diinput* benar maka user akan berpindah ke halaman *dashboard*.

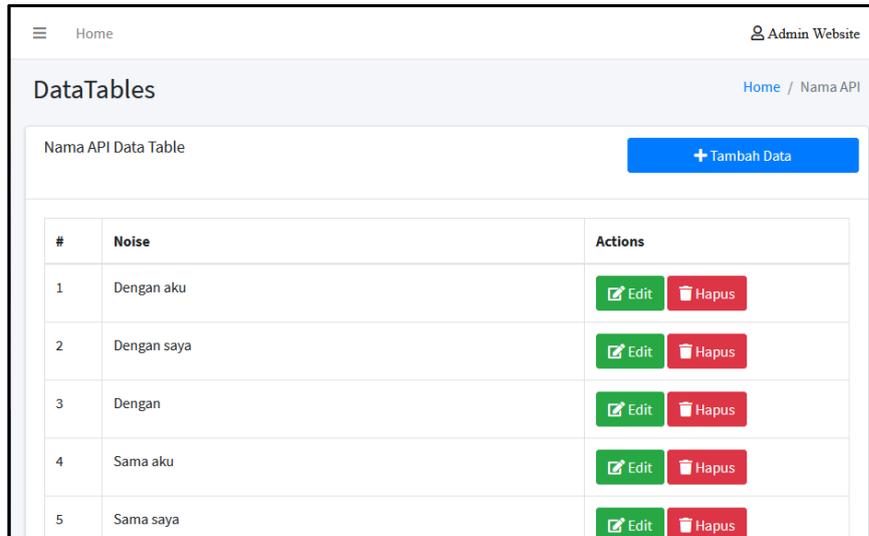
B. Halaman *Dashboard*



Gambar 3.21 Halaman *Dashboard*

Pada halaman *dashboard* ini, terdapat statistik dari Kategori API dan Nama API. Statistik yang ditampilkan diantaranya terdapat jumlah API yang tersedia, jumlah table dalam *database* yang digunakan, jumlah data *noise* secara keseluruhan, jumlah user yang terdaftar, perbandingan antara jumlah data *noise* yang terdapat pada Nama API dengan Kategori API yang ditampilkan dengan *donat chart*, dan kalender yang menunjukkan tanggal. Pada sisi kiri dari halaman *dashboard* terdapat menu yang menjadi akses utama bagi user jika ingin mencoba atau *manage* API yang tersedia.

C. Halaman *Manage* Nama API

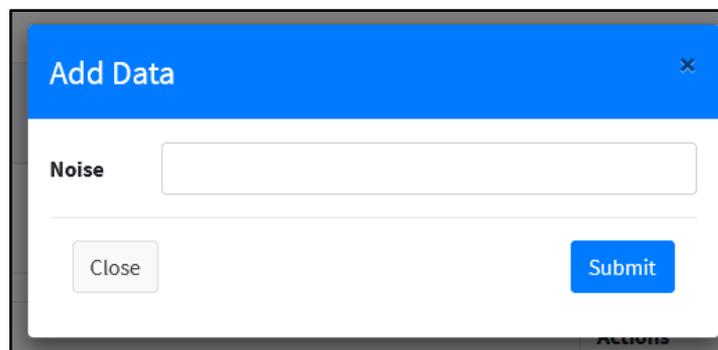


The screenshot shows a web interface for managing API names. At the top, there is a navigation bar with 'Home' and 'Admin Website'. Below that, the page title is 'DataTables' with a breadcrumb 'Home / NamaAPI'. The main content area is titled 'Nama API Data Table' and features a blue '+ Tambah Data' button. Below this is a table with the following data:

#	Noise	Actions
1	Dengan aku	Edit Hapus
2	Dengan saya	Edit Hapus
3	Dengan	Edit Hapus
4	Sama aku	Edit Hapus
5	Sama saya	Edit Hapus

Gambar 3.22 Halaman *Manage* Nama API

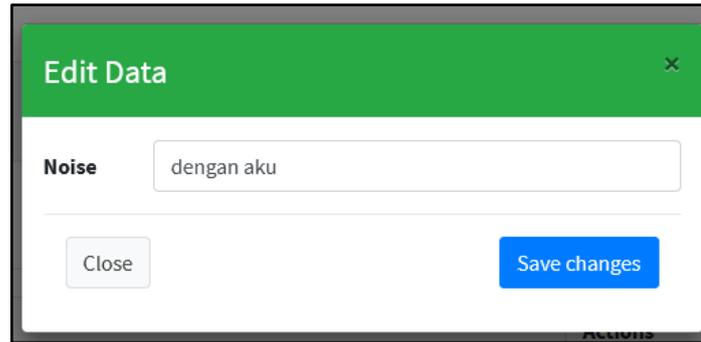
Pada halaman ini, terdapat table yang berisi data *noise* dari *database* yang ada. Terdapat juga fitur *add*, *edit*, dan *delete* yang dapat digunakan oleh admin untuk menambah, mengubah, atau bahkan menghapus data *noise*.



The screenshot shows a modal window titled 'Add Data'. It contains a text input field labeled 'Noise'. Below the input field, there are two buttons: 'Close' and 'Submit'.

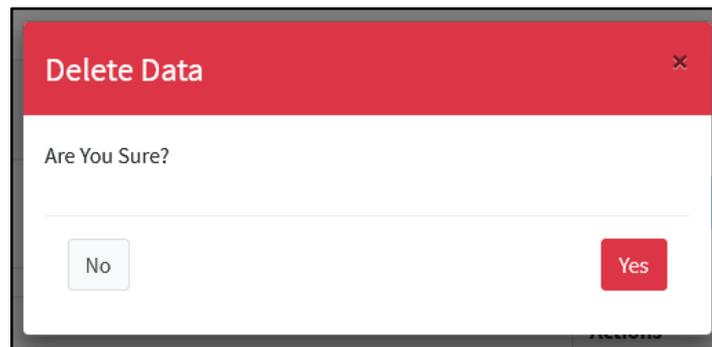
Gambar 3.23 Tampilan *Modal Add* Nama API

Gambar 3.23 merupakan tampilan dari *modal* untuk fitur *add* pada Nama API. Fungsi dari fitur *add* ialah untuk menambahkan *noise* ke dalam *database*.



Gambar 3.24 Tampilan *Modal Edit* Nama API

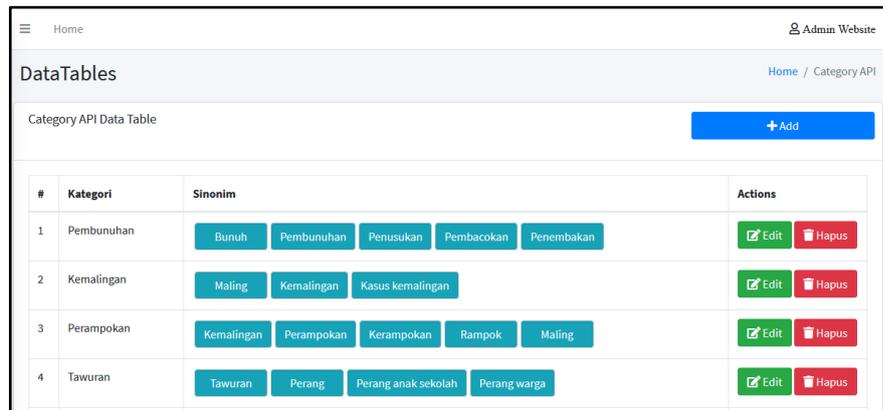
Gambar 3.24 merupakan tampilan dari *modal* untuk fitur *edit* pada Nama API. Fungsi dari fitur *edit* ialah untuk mengubah atau memperbaiki *noise* yang sudah ada.



Gambar 3.25 Tampilan *Modal Delete* Nama API

Gambar 3.25 merupakan tampilan dari *modal* untuk fitur *delete* pada Nama API. Fungsi dari fitur *delete* ialah untuk menghapus *noise* yang sudah dibuat sebelumnya.

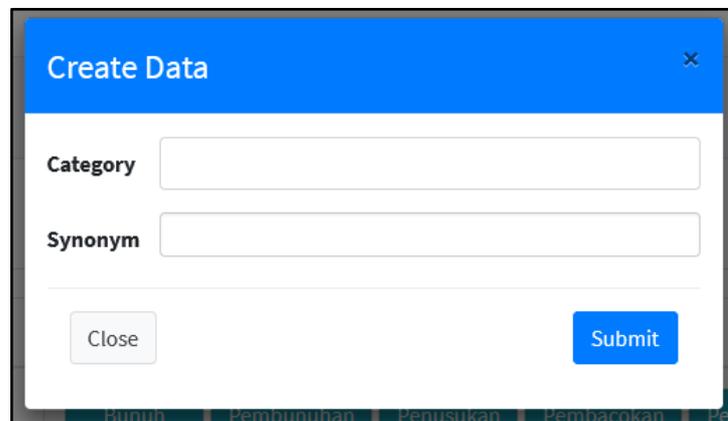
D. Halaman *Manage* Kategori API



#	Kategori	Sinonim	Actions
1	Pembunuhan	Bunuh Pembunuhan Penusukan Pembacokan Penembakan	Edit Hapus
2	Kemalingan	Maling Kemalingan Kasus kemalingan	Edit Hapus
3	Perampokan	Kemalingan Perampokan Kerampokan Rampok Maling	Edit Hapus
4	Tawuran	Tawuran Perang Perang anak sekolah Perang warga	Edit Hapus

Gambar 3.26 Halaman *Manage* Kategori API

Pada halaman ini, terdapat table yang berisi data kategori *noise* dari *database* yang ada. Terdapat juga fitur *add*, *edit*, dan *delete* yang dapat digunakan oleh admin untuk menambah, mengubah, atau bahkan menghapus data kategori *noise*.



Create Data

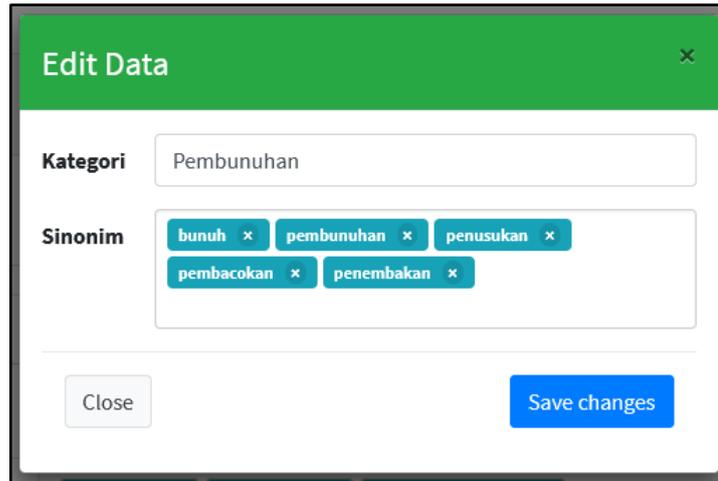
Category

Synonym

Close Submit

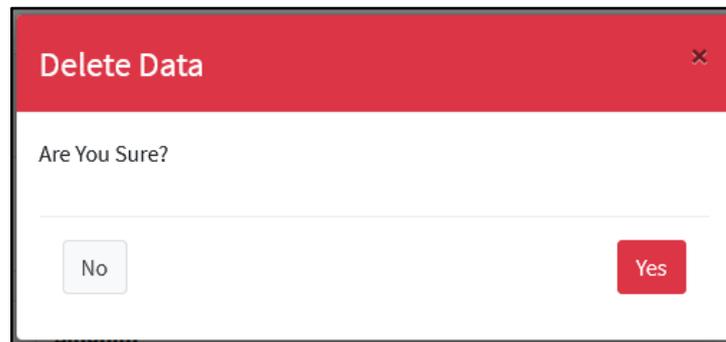
Gambar 3.27 Tampilan *Modal Add* Kategori API

Gambar 3.27 merupakan tampilan dari *modal* untuk fitur *add* pada Kategori API. Fungsi dari fitur *add* ialah untuk menambahkan kategori *noise* ke dalam *database*.



Gambar 3.28 Tampilan *Modal Edit* Kategori API

Gambar 3.28 merupakan tampilan dari modal untuk fitur *edit* pada Kategori API. Fungsi dari fitur *edit* ialah untuk mengubah atau memperbaiki kategori *noise* yang sudah ada.



Gambar 3.29 Tampilan *Modal Delete* Kategori API

Gambar 3.29 merupakan tampilan dari *modal* untuk fitur *delete* pada Kategori API. Fungsi dari fitur *delete* ialah untuk menghapus kategori *noise* yang sudah dibuat sebelumnya.

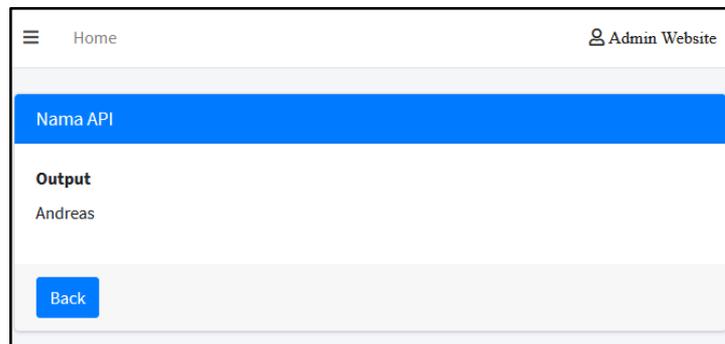
E. Halaman *Test* Nama API



The screenshot shows a web application interface. At the top, there is a navigation bar with a hamburger menu icon on the left, the text 'Home' in the center, and a user profile icon followed by 'Admin Website' on the right. Below the navigation bar is a blue header with the text 'Nama API'. Underneath the header, there is a section titled 'Input'. This section contains a text input field with the text 'Nama saya andreas' and a blue 'Submit' button below it.

Gambar 3.30 Tampilan *Test* Nama API

Gambar 3.30 merupakan tampilan dari halaman *test* Nama API. Halaman ini akan menerima *input* berupa nama yang mengandung *noise* untuk dilakukan proses *noise* filter sehingga hasilnya seperti yang terlihat pada Gambar 3.31 di bawah ini.

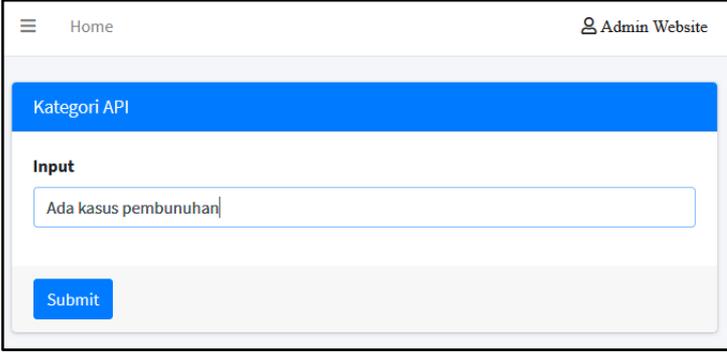


The screenshot shows the same web application interface as in Gambar 3.30. The 'Input' section is now replaced by an 'Output' section. The 'Output' section displays the text 'Andreas'. Below the 'Output' section, there is a blue 'Back' button.

Gambar 3.31 Tampilan Hasil *Test* Nama API

Gambar 3.31 merupakan tampilan hasil dari pemrosesan data yang *diinput* pada Gambar 3.30. Dapat dilihat, API akan menghilangkan *noise* yang terdapat pada *input* sehingga menghasilkan *output* hanya berupa nama saja.

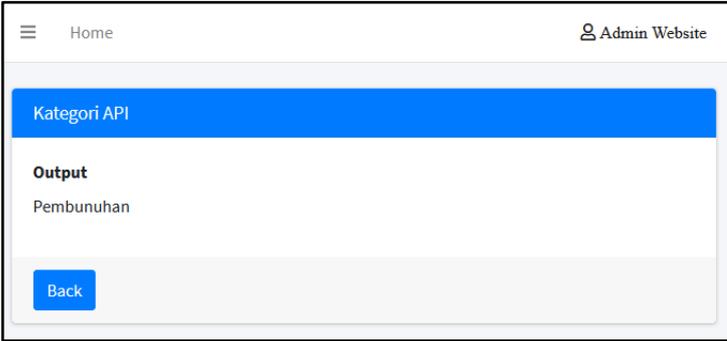
F. Halaman *Test* Kategori API



The screenshot shows a mobile application interface. At the top, there is a navigation bar with a hamburger menu icon on the left, the text 'Home' in the center, and a user profile icon with the text 'Admin Website' on the right. Below the navigation bar is a blue header with the text 'Kategori API'. Underneath the header is a white input field with the text 'Ada kasus pembunuhan'. At the bottom of the input area is a blue button with the text 'Submit'.

Gambar 3.32 Tampilan *Test* Kategori API

Gambar 3.32 merupakan tampilan dari halaman *test* Kategori API. Halaman ini akan menerima *input* berupa kategori tindakan kriminalitas yang mengandung *noise* untuk dilakukan proses *noise* filter. Sehingga, hasilnya seperti yang terlihat pada Gambar 3.33 dibawah ini.

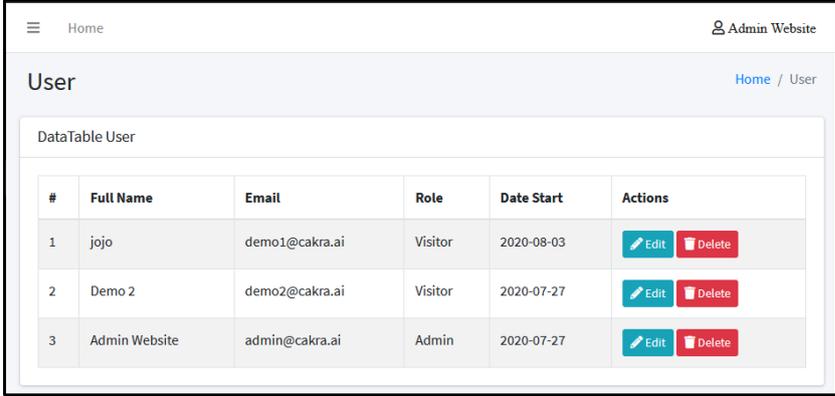


The screenshot shows the same mobile application interface as in Gambar 3.32. The blue header 'Kategori API' is still present. Below it, the text 'Output' is displayed, followed by the text 'Pembunuhan'. At the bottom of the output area is a blue button with the text 'Back'.

Gambar 3.33 Tampilan *Test* Kategori API

Gambar 3.33 merupakan tampilan hasil dari pemrosesan data yang *diinput* pada Gambar 3.32. Dapat dilihat, API akan menghilangkan *noise* yang terdapat pada *input* sehingga menghasilkan *output* hanya berupa kategori tindakan kriminalitas saja.

G. Halaman *Access Control*

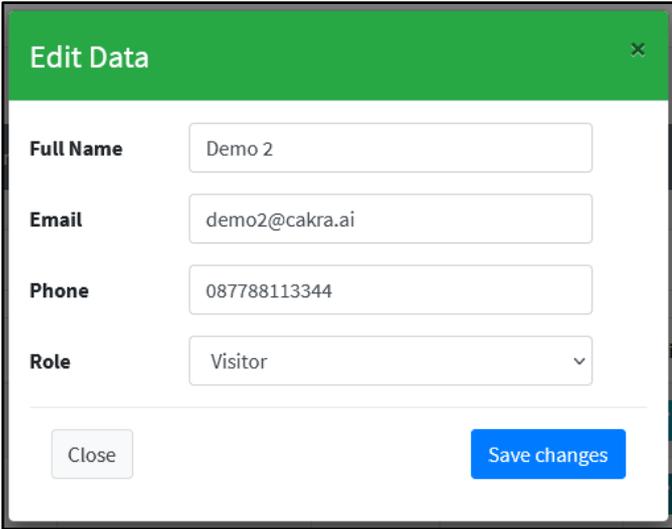


The screenshot shows a web interface for user management. At the top, there is a navigation bar with 'Home' and 'Admin Website'. Below this, the page title is 'User'. A breadcrumb trail shows 'Home / User'. The main content area is titled 'DataTable User' and contains a table with the following data:

#	Full Name	Email	Role	Date Start	Actions
1	jojo	demo1@cakra.ai	Visitor	2020-08-03	Edit Delete
2	Demo 2	demo2@cakra.ai	Visitor	2020-07-27	Edit Delete
3	Admin Website	admin@cakra.ai	Admin	2020-07-27	Edit Delete

Gambar 3.34 Tampilan *Access Control*

Gambar 3.34 merupakan tampilan dari halaman *Access Control*. Halaman ini menampilkan data user yang terdaftar pada *database*. Terdapat juga fitur *edit* dan *delete* yang dapat digunakan oleh admin, untuk mengubah atau menghapus user yang terdaftar.



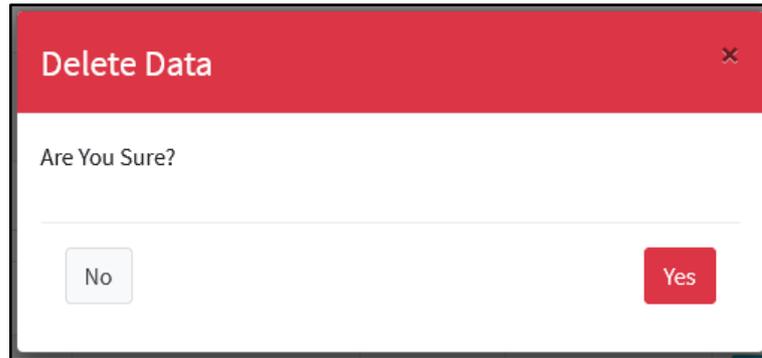
The screenshot shows a modal window titled 'Edit Data' with a close button (X) in the top right corner. The form contains the following fields:

- Full Name:** Demo 2
- Email:** demo2@cakra.ai
- Phone:** 087788113344
- Role:** Visitor (dropdown menu)

At the bottom of the modal, there are two buttons: 'Close' and 'Save changes'.

Gambar 3.35 Tampilan *Access Control Edit User*

Gambar 3.35 merupakan tampilan dari modal untuk fitur *edit* pada *Access Control*. Fungsi dari fitur *edit* ialah untuk mengubah atau memperbaiki data user yang sudah ada.



Gambar 3.36 Tampilan *Access Control Delete User*

Gambar 3.36 merupakan tampilan dari *modal* untuk fitur *delete* pada *Access Control*. Fungsi dari fitur *delete* ialah untuk menghapus user yang terdaftar pada *database*.

#	Role Name	Role Descriptions	Accessible Menu
1	Admin	Full CRUD to Website	Manage API Test API Access Control Speech Recognition
2	Visitor	Full access to Test API pages	Test API Speech Recognition

Gambar 3.37 Tampilan *Access Control Role*

Gambar 3.37 merupakan tampilan dari halaman *Access Control Role*. Halaman ini menampilkan data *role* yang terdaftar pada *database*.

3.4 Kendala Yang Ditemukan

Berikut ini merupakan kendala yang dihadapi dalam pelaksanaan kerja magang.

1. Kesulitan dalam berkoordinasi, berdiskusi, dan mengevaluasi hasil kerja magang secara langsung dengan pembimbing lapangan karena situasi pandemi Covid-19.
2. Kesulitan dalam merancang algoritma Kategori Filter karena menggunakan *noise* lebih dari satu.
3. Kesulitan dalam mempelajari *framework* CodeIgniter 4.

3.5 Solusi Atas Kendala Yang Ditemukan

Berikut ini merupakan solusi atas kendala yang ditemukan dalam pelaksanaan kerja magang.

1. Proses koordinasi, diskusi, dan evaluasi dilakukan secara tidak langsung dengan menggunakan aplikasi *google meets*, dan *whatsapp*.
2. Mencari referensi dari internet untuk membantu memecahkan masalah yang dihadapi.
3. Membaca dan mempelajari *framework* CodeIgniter 4 dari dokumentasi resmi maupun video *tutorial* di *YouTube*.