

## LAMPIRAN

### Lampiran 1 Surat Kesediaan Pembimbing

#### SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Tri Hesti Wahyuningsih  
NIM : 20090082  
Program Studi : Teknik Informatika

Pihak Kedua

Nama : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.  
Status : Dosen  
NIDN : 0613028601  
Jabatan Fungsional : Lektor Kepala  
Pangkat/Golongan : Penata Tk. I/III d

Pada hari ini senin tanggal 18 maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I/II Skripsi Pihak Pertama dengan **syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua.** Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 18 Maret 2024

Pihak Pertama



Tri Hesti Wahyuningsih

Pihak Kedua



Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Anjani, S.T., M.Kom.  
NIPY.09.015.225

## SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Tri Hesti Wahyuningsih  
NIM : 20090082  
Program Studi : Teknik Informatika

Pihak Kedua

Nama : Dyah Apriliani, S.T., M.Kom.  
Status : Dosen Tetap  
NIDN : 0614049002  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : III d

Pada hari ini kamis tanggal 14 maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan **Bersedia bimbingan secara rutin untuk menyelesaikan Skripsi minimal 1 minggu 1 kali, apabila dalam waktu 40 hari pertama setelah kesepakatan ini dibuat pihak pertama tidak konsisten untuk menyanggupi maka pihak kedua berhak membatalkan kesepakatan ini dan tidak melanjutkan untuk membimbing pihak pertama dengan tidak akan memberikan rekomendasi untuk mengikuti ujian skripsi.** Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 14 Maret 2024

Pihak Pertama



Tri Hesti Wahyuningsih

Pihak Kedua



Dyah apriliani, S.T., M.Kom.

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.T., M.Kom.  
NIPY.09.015.225

## Lampiran 2 Surat Keterangan Penelitian



**POLITEKNIK HARAPAN BERSAMA**  
The True Vocational Campus

Sarjana Terapan Teknik Informatika

Nomor : 42.03/TI.PHB/VII/2024  
Lampiran : -  
Hal : Permohonan Izin Observasi  
Kepada :  
Yth. : **Pemilik Apotek Husni Farma Tegal**  
di Tegal

Dengan hormat, mahasiswa dengan identitas berikut ini:

nama : Tri Hesti Wahyuningsih  
NIM : 20090082  
prodi : Sarjana Terapan Teknik Informatika

Bermaksud melakukan penelitian untuk keperluan Skripsi dengan judul "**Virtual Assistant Pada Apotek Husni Farma Tegal Dengan Framework BotMan**". Kami memohon Bapak/Ibu memberikan izin kepada mahasiswa yang bersangkutan agar memperoleh data, keterangan, dan bahan yang diperlukan.

Demikian permohonan ini disampaikan, atas perhatian kami ucapkan terima kasih.

Tegal, 16 Juli 2024  
Ka. Prodi S.Tr. Teknik Informatika,

**Dyan Apriyanti, S.T., M.Kom**  
NIPY : 09.015.225

## Lampiran 3 Surat Pernyataan Pengajuan HKI

### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Tri Hesti Wahyuningsih  
Kewarganegaraan : Indonesia  
Alamat : Desa Plumbungan RT/RW 04/01 Kecamatan Kramat Kabupaten Tegal, Jawa Tengah, 52181
2. Nama : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jln. Raya Kluwut Timur No. 24, Kecamatan Bulakamba, Kabupaten Brebes, 52253
3. Nama : Dyah Apriliani, S.T., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Perumahan mutiara indah blok c4, jalan angka gang 2 rt 02 rw 02 kel. Procot kec. Slawi Kab. Tegal, 52412

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:  
Berupa : Program Komputer  
Berjudul : Virtual Asisstant pada Sistem Inventory Apotek Husni Farma dengan Framework BotMan
  - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
  - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
  - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
  - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
  - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
  - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:

- a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
- b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 7 Juli 2024



(Tri Hesti Wahyuningsih)  
Pemegang Hak Cipta \*

(Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom)  
Pemegang Hak Cipta \*

(Dyah Apriliani, S.T., M.Kom.)  
Pemegang Hak Cipta \*

\* Semua pemegang hak cipta agar menandatangani di atas materai.



## Lampiran 4 Surat Pengalihan HKI

### SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Tri Hesti Wahyuningsih  
Kewarganegaraan : Indonesia  
Alamat : Desa Plumbungan RT/RW 04/01 Kecamatan Kramat Kabupaten Tegal, Jawa Tengah, 52181
2. Nama : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jln. Raya Kluwut Timur No. 24, Kecamatan Bulakamba, Kabupaten Brebes, 52253
3. Nama : Dyah Apriliani, S.T., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Perumahan mutiara indah blok c4, jalan angka gang 2 rt 02 rw 02 kel. Procot kec. Slawi Kab. Tegal, 52412

Adalah Pihak I selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)  
Politeknik Harapan Bersama  
Alamat : Jl. Mataram No. 9 Pesurungan Lor Kota Tegal

Adalah Pihak II selaku Pemegang Hak Cipta berupa Program Komputer dengan judul **"Virtual Assistant pada Sistem Inventory Apotek Husni Farma dengan Framework BotMan"** untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jendral Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, 8 Juli 2024

Pemegang Hak Cipta

Ketua P3M  


(Dr. Aldi Budi Riyanta S.Si, M.T)

Pencipta

  
32AK82461607

(Tri Hesti Wahyuningsih)

(Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.)

(Dyah Apriliani, S.T., M.Kom.)

Lampiran 5 Syarat Pengajuan HKI (manual book, dan lain lain)

**Manual Book**  
**APLIKASI *VIRTUAL ASISSTANT* PADA SISTEM *INVENTORY* APOTEK**  
**HUSNI FARMA DENGAN *FRAMWORK BOTMAN***

**A. Website (User)**

1. Halaman *Landing Page* User

- a. Membuka website apotek <http://husnifarma.my.id/>
  - Buka browser web anda (seperti *Google Chrome*, *Mozila Firefox*, atau *Safari*)
  - Ketikkan URL website apotek di bilah alamat browser anda dan tekan tombol '*Enter*'
  - Setelah berhasil masuk ke halaman landing page website apotek makan user dapat mengakses fitur *chatbot* yang canggih untuk memeriksa ketersediaan obat secara *realtime* di apotek Husni Farma.

User setelah masuk ke website akan ditampilkan halaman *landing page* seperti berikut.



Gambar 1. Halaman landing page user

2. Halaman *Chatbot*

- a. Klik ikon atau *widjet chatbot* untuk membuka jendela obrolan

- Jendela obrolan akan muncul dilayar anda, ada dibagian kanan bawah



Gambar 2. widget chatbot

b. Memulai percakapan dengan *chatbot*

- Setelah jendela obrolan termuka, anda akan melihat pesan sambutan *chatbot*



Gambar 3. Pesan sambutan pada chatbot

- Ketikkan pertanyaan atau permintaan anda di kolom teks yang tersedia dibagian bawah jendela obrolan.
- Setelah anda mengirim pesan, tunggu beberapa saat untuk menerima respon dari *chatbot*
- *Chatbot* akan merespon dengan menampilkan kategori, dan anda bisa memilih kategori obat yang akan anda cari.





Gambar 4. Tampilan pilihan kategori pada chatbot

- Setelah anda memilih kategori, *chatbot* akan menampilkan pilihan obat berdasarkan kategori obat yang anda pilih.



Gambar 5. Tampilan pilihan obat pada chatbot

- Setelah anda memilih obat, maka *chatbot* akan merespon dengan menampilkan detail ketersediaan obat yang anda pilih



Gambar 6. Tampilan detail ketersediaan obat yang dipilih

## B. Website (Admin)

### 1. Halaman Awal dan Login

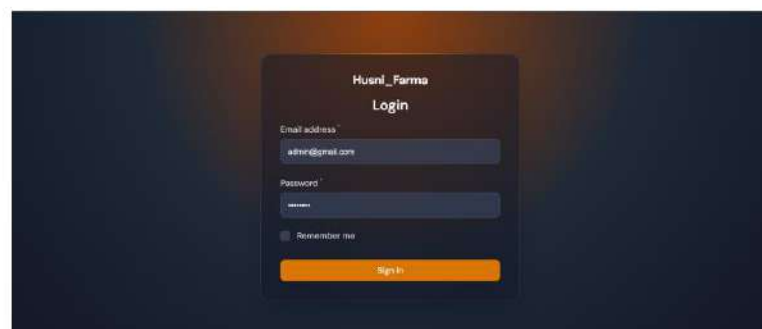
- a. Membuka website apotek <http://husnifarma.my.id/admin>
  - Buka browser web anda (seperti *Google Chrome*, *Mozilla Firefox*, atau *Safari*)
  - Ketikkan URL website apotek di bilah alamat browser anda dan tekan tombol 'Enter'
  - Setelah berhasil masuk ke halaman landing page website apotek maka admin dapat login pada button 'Login Admin'.



Gambar 7. Tampilan button login admin

### b. Login Admin

- Input Username dan password
- *Username* : ketikkan *username* yang telah diberikan
- *Password* : Ketikkan *password* yang sesuai



Gambar 8. Tampilan halaman login admin

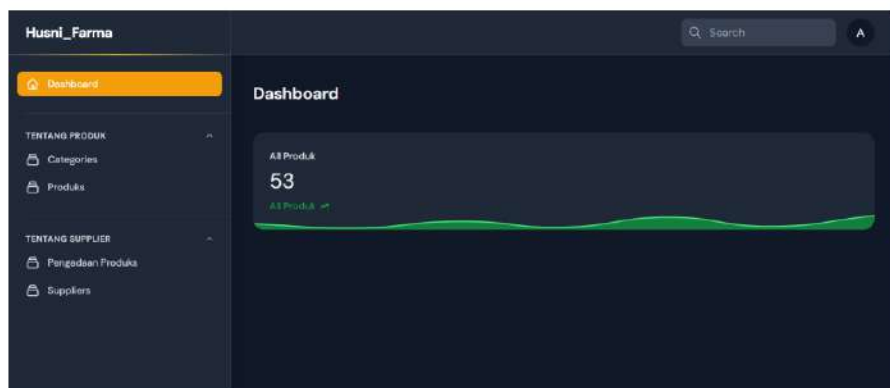
## 2. Halaman Dashboard

### a. Informasi singkat

Dashboard menampilkan ringkasan informasi berupa jumlah produk pada apotek

### b. Navigasi

Dibagian samping, terdapat menu *navigasi* untuk mengakses halaman-halaman lain.



Gambar 9. Tampilan halaman dashboard admin

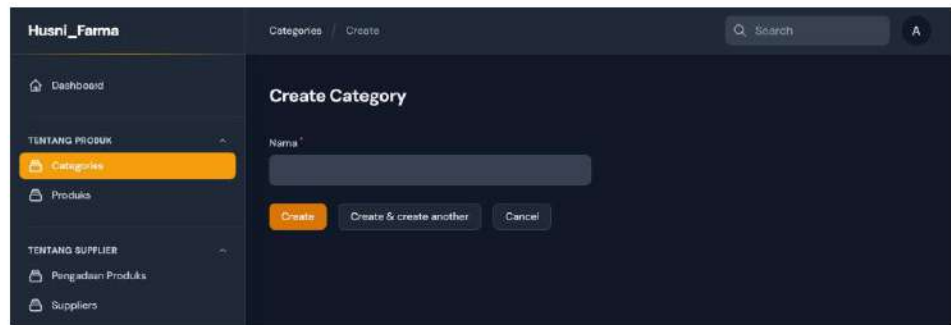
## 3. Halaman *Category*

### a. Mengakses halaman *category*

- Klik menu '*category*' di menu navigasi, klik opsi '*category*'

### b. Create atau tambah *category* baru

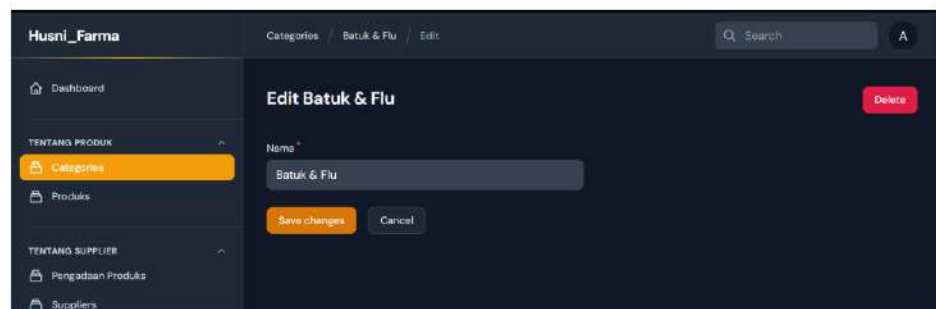
- Klik button '*New Category*' yang terletak dibagian kanan atas halaman
- Isi formulir dengan mengetikkan nama *category* baru
- Klik button '*create*', maka akan menampilkan '*new category created*'.



Gambar 10. Tampilan halaman create category produk

c. Edit atau ubah *category*

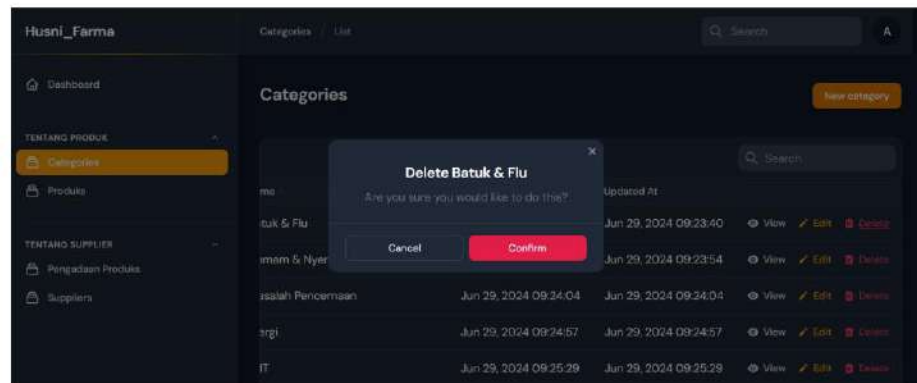
- Pilih atau cari *category* yang ingin diedit dalam daftar
- Klik button '*Edit*' diantara button '*view*' dan '*delete*'
- Edit atau ubah informasi yang diperlukan
- Klik button '*save change*', maka nama *category* yang ubah akan terupdate.



Gambar 11. Tampilan halaman edit category produk

d. Delete atau hapus *category*

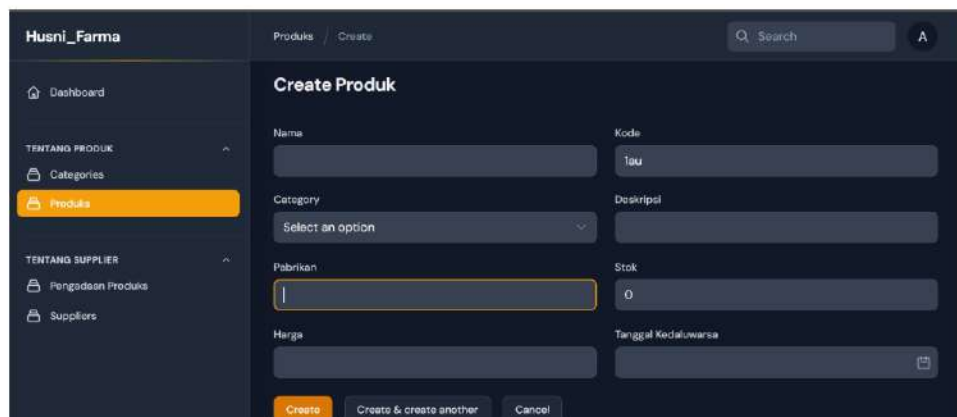
- Piiih atau cari *category* yang ingin dihapus dari daftar
- Klik button '*Delete*' yang terletak disamping button '*Edit*'.
- Konfirmasi data yang akan dihapus, klik '*confirm*' untuk menghapus *category* dari daftar



Gambar 12. Tampilan halaman delete category produk

#### 4. Halaman Produk

- a. Mengakses halaman produk
  - Klik menu 'produk' dimenu navigasi, klik opsi 'produk'
- b. Create atau tambah produk baru
  - Klik button '*New Produk*' yang terletak dibagian kanan atas halaman
  - Isi formular dengan mengisi nama, pilih *category*, deskripsi, pabrik, stok, harga, dan tanggal kadaluwarsa sesuai produk yang akan diinput.
  - Klik button '*create*', maka akan menampilkan '*new produk created*'.

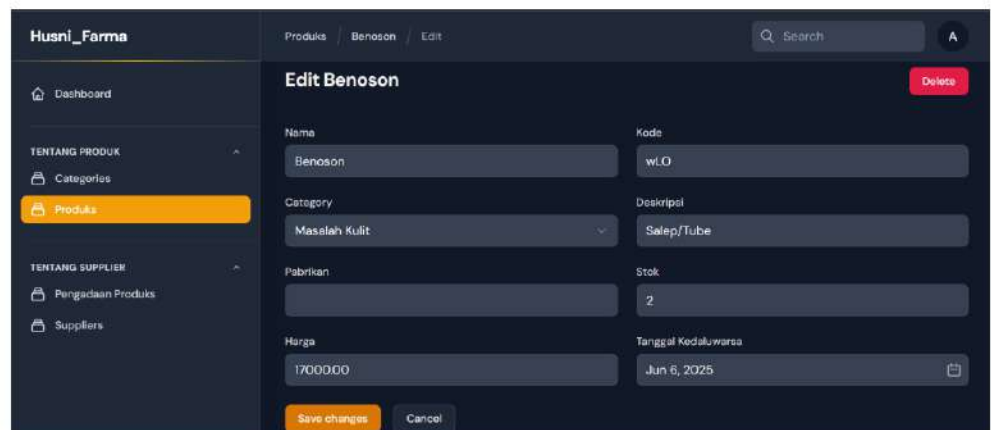


Gambar 13. Tampilan halaman create produk



c. Edit atau ubah produk

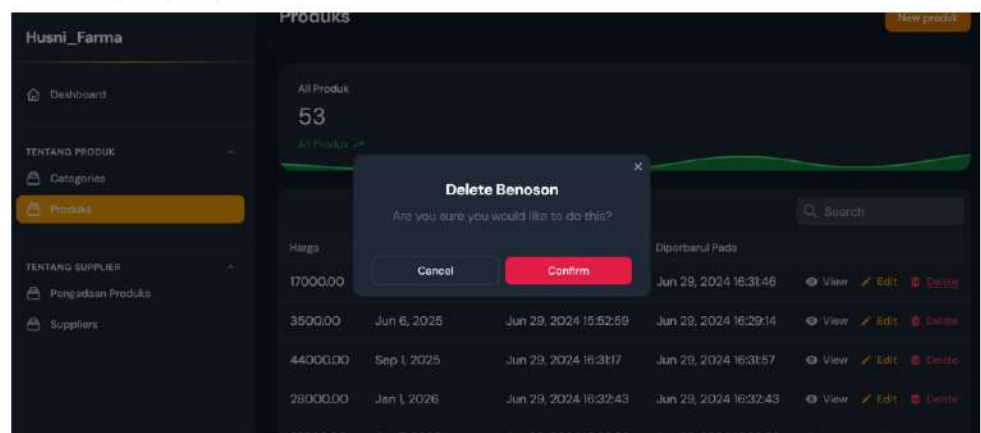
- Pilih atau cari produk yang ingin diedit dalam daftar
- Klik button '*Edit*' diantara button '*view*' dan '*delete*'
- Edit atau ubah informasi yang diperlukan
- Klik button '*save change*', maka informasi produk yang diubah akan terupdate.



Gambar 14. Tampilan halaman edit produk

d. Delete atau hapus produk

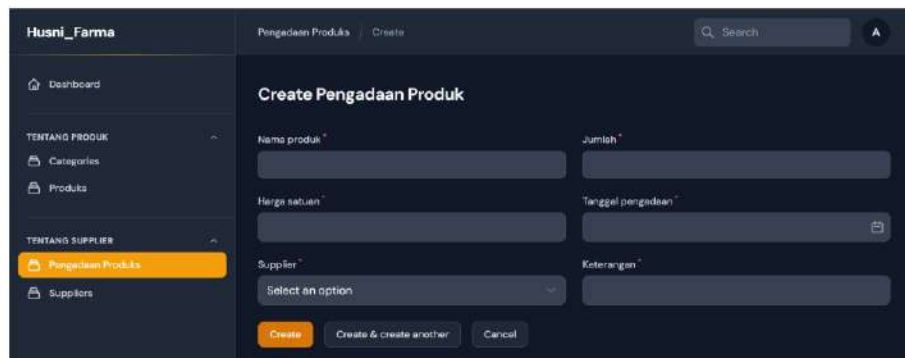
- Piiih atau cari produk yang ingin dihapus dari daftar
- Klik button '*Delete*' yang terletak disamping button '*Edit*'.
- Konfirmasi data yang akan dihapus, klik '*confirm*' untuk menghapus produk dari daftar.



Gambar 15. Tampilan halaman delete produk

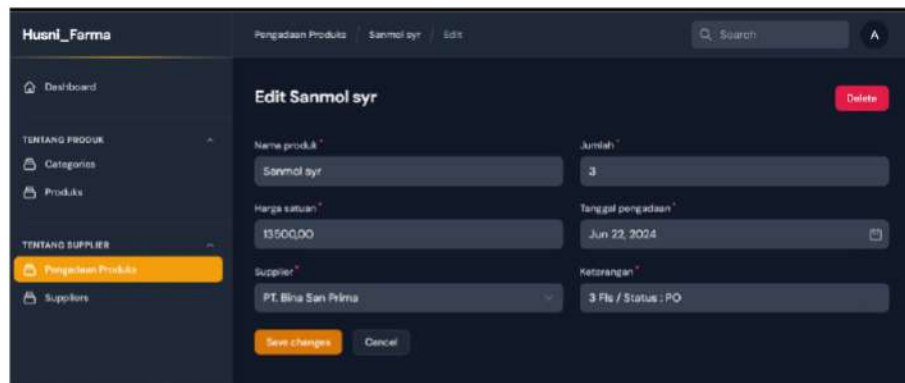
## 5. Halaman Pengadaan Produk

- a. Mengakses halaman pengadaan produk
  - Klik menu 'pengadaan produk' dimenu navigasi, klik opsi 'pengadaan produk'
- b. *Create* atau tambah pengadaan produk baru
  - Klik button '*New Pengadaan Produk*' yang terletak dibagian kanan atas halaman
  - Isi formular dengan mengisi nama, jumlah, harga satuan, tanggal pengaduan, pilih supplier, keterangan sesuai dengan surat pesanan apotek.
  - Klik button '*create*', maka akan menampilkan '*new pengadaan produk creted*'.



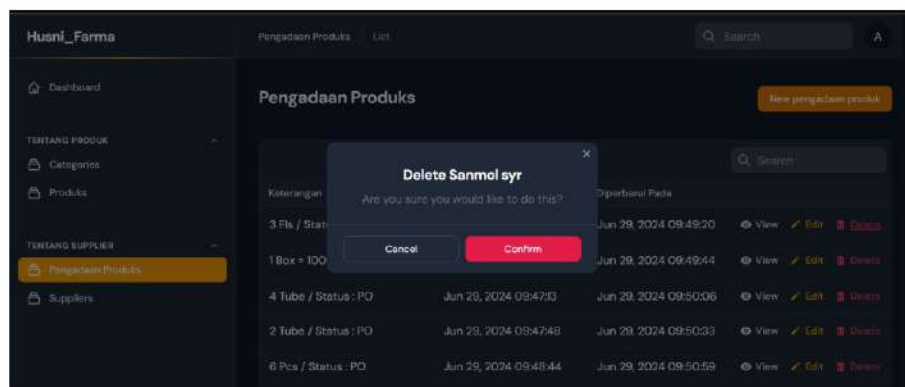
Gambar 16. Tampilan halaman create category pengaduan produk

- c. Edit atau ubah pengadaan barang
  - Pilih atau cari pengadaan barang yang ingin diedit dalam daftar
  - Klik button '*Edit*' diantara button '*view*' dan '*delete*'
  - Edit atau ubah informasi yang diperlukan
  - Klik button '*save change*', maka informasi pengedaan produk akan terupdate sesuai yang diubah



Gambar 17. Tampilan halaman edit pengadaan produk

- d. *Delete* atau hapus pengadaan barang
- Pilih atau cari pengadaan barang yang ingin dihapus dari daftar
  - Klik button '*Delete*' yang terletak disamping button '*Edit*'.
  - Konfirmasi data yang akan dihapus, klik '*confirm*' untuk menghapus pengadaan barang dari daftar

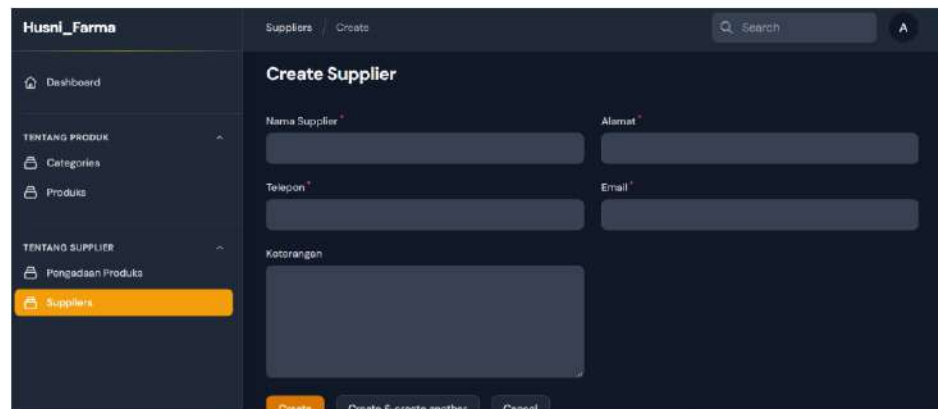


Gambar 18. Tampilan halaman delete pengadaan produk

## 6. Halaman Supplier

- a. Mengakses halaman supplier
  - Klik menu 'supplier' dimenu navigasi, klik opsi 'supplier'
- b. *Create* atau tambah supplier baru
  - Klik button '*New Supplier*' yang terletak dibagian kanan atas halaman

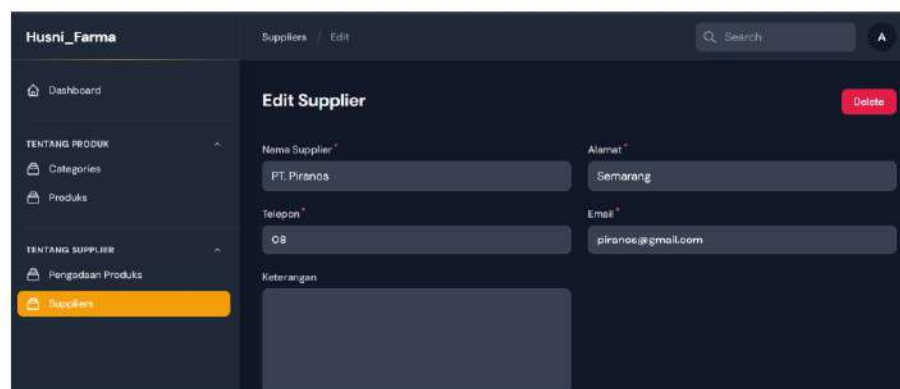
- Isi formulir dengan mengisi nama, alamat, telepon, email dan keterangan.
- Klik button *'create'*, maka akan menampilkan *'new supplier created'*.



Gambar 19. Tampilan halaman create supplier

c. *Edit* atau ubah supplier

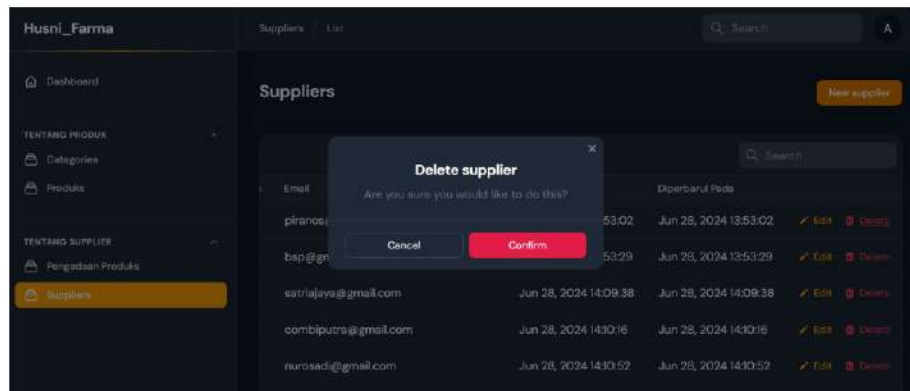
- Pilih atau cari supplier yang ingin diedit dalam daftar
- Klik button *'Edit'* disamping button *'delete'*
- *Edit* atau ubah informasi yang diperlukan
- Klik button *'save change'*, maka informasi supplier yang diubah akan terupdate.



Gambar 20. Tampilan halaman edit supplier

d. *Delete* atau hapus supplier

- Pilih atau cari supplier yang ingin dihapus dari daftar
- Klik button '*Delete*' yang terletak disamping button '*Edit*'.
- Konfirmasi data yang akan dihapus, klik '*confirm*' untuk menghapus supplier dari daftar.



Gambar 21. Tampilan halaman delete supplier



**DOKUMEN TEKNIKAL**  
**APLIKASI *VIRTUAL ASISSTANT* PADA SISTEM *INVENTORY* APOTEK**  
**HUSNI FARMA DENGAN *FRAMWORK BOTMAN***

**1. Profile**

Aplikasi *Virtual Assistant* pada Sistem *Inventory* Apotek Husni Farma dengan framework *BotMan* merupakan solusi inovatif yang mengintegrasikan teknologi chatbot dengan sistem pengelolaan *inventory*. Aplikasi ini tidak hanya meningkatkan efisiensi dan akurasi dalam pengelolaan stok tetapi juga memberikan kemudahan akses informasi bagi pelanggan, sehingga operasional apotek menjadi lebih lancar dan responsif terhadap kebutuhan pelanggan.

**Fitur Utama :**

- a. Sistem *Inventory*: Sistem *inventory* ini dirancang untuk memudahkan pengelolaan stok produk di apotek melalui fitur *CRUD (Create, Read, Update, Delete)* yang mencakup kategori, produk, pengadaan barang, dan supplier.
- b. *Virtual assistant* dengan *chatbot* : *Chatbot* yang dibangun menggunakan *BotMan* ini berfungsi sebagai *virtual assistant* yang dapat memberikan informasi terkait *inventory*, membantu pelanggan dalam mencari tahu ketersediaan obat, serta menjawab pertanyaan seputar produk dan layanan apotek.

**2. Latar Belakang**

Perkembangan teknologi yang sangat pesat memberikan banyak pengaruh pada kehidupan. Manusia mampu memanfaatkan teknologi dengan efektif dan efisien dalam berbagai bidang, salah satunya dalam bidang pengobatan. Dalam bidang pengobatan salah satunya adalah dalam pengelolaan dan penyimpanan data. Apotek salah satu jenis usaha dibidang pengobatan yang asngat memerlukan adanya sistem informasi pengelolaan

data untuk mempermudah dan memperlancar kinerjanya. Teknologi lain yang dapat dimanfaatkan yaitu asisten virtual yang memudahkan tujuan manusia dengan melakukan tugas-tugas secara otomatis sesuai dengan program yang ditentukan.

Asisten virtual merupakan sebuah sistem yang dapat memberikan informasi kepada penggunanya. Asisten virtual dapat berbentuk teks, suara, gambar maupun gabungan ketiganya. Bentuk asisten virtual yang akan dibuat dalam sistem ini adalah fitur chat yang merupakan percakapan dari user dan bot yaitu sebuah program yang memiliki data didalamnya, jika bot diberi masukkan maka bot akan dapat memberikan sebuah jawaban.

Apotek merupakan suatu usaha yang bergerak pada bidang kesehatan khususnya dalam menyalurkan obat pada masyarakat salah satunya yaitu Apotek Husni Farma. Selama ini manajemen stok apotek masih menggunakan secara manual seperti pendataan obat pada buku. Pegawai harus menulis tangan pendataan obat dan setelah itu dijadikan *microsoft excel* untuk dijadikan laporan, yang dikhawatirkan memungkinkan buku rusak atau hilang. Pengelolaan data merupakan kegiatan yang dilakukan dengan menggunakan masukan berupa data dan menghasilkan informasi yang bermanfaat dan bertujuan sesuai dengan yang direncanakan.

BotMan merupakan *framework chatbot php* yang didesain untuk memudahkan dalam membangun bot yang inovatif ke dalam banyak platform olahpesan. Botman juga mendukung fitur penting lain seperti *middleware system*. Botman juga dapat membuat chatbot real time yang tidak perlu menggunakan API atau aplikasi pihak ketiga (pendukung).

Beberapa penelitian sebelumnya, sistem manajemen apotek hanya berfokus dengan layanan apotek dari sisi manajemen apotek, sedangkan penelitian yang sekarang melibatkan asisten virtual berupa chatbot berbasis tombol atau menu, dimana nanti user memilih pilihan yang ditampilkan tanpa user mengetik sendiri. Implementasi asisten virtual bertujuan untuk menyampaikan informasi seperti stok obat kepada pelanggan sesuai dengan

obat yang dicari. Botman yang saya implementasikan menggunakan fitur database middleware untuk terhubung langsung dengan database admin, memastikan chatbot memberikan informasi yang real time dan akurat kepada pengguna tanpa ketergantungan pada platform tertentu.

### 3. Manfaat

Adapun beberapa manfaat yang diharapkan oleh penulis adalah sebagai berikut:

- a. Memberikan efisiensi proses *inventory* dalam pengecekan dan pencatatan stok obat, mengurangi kebutuhan untuk pengecekan manual oleh staf apotek, sehingga mempercepat dan menyederhanakan pengelolaan *inventory*.
- b. Dengan asisten virtual pada *web profile* apotek dapat memudahkan pengguna untuk mendapatkan informasi terkini tentang ketersediaan obat di Apotek Husni Farma secara cepat kapanpun dan dimanapun.

### 4. Spesifikasi Teknis

#### a. Perangkat Keras

Perangkat keras yang digunakan untuk melakukan perkembangan pada aplikasi “*virtual assistant* pada sistem *inventory* apotek Husni Farma dengan *Framwork BotMan*” ini adalah sebagai berikut :

Komputer / Laptop dengan spesifikasi processor AMD Ryzen 7 5000 series, RAM 8 GB dan penyimpanan kapasitas SSD 256 GB yang dilakukan development aplikasi.

#### b. Perangkat Lunak

Perangkat lunak yang digunakan untuk melakukan perkembangan pada aplikasi “*virtual assistant* pada sistem *inventory* apotek Husni Farma dengan *Framwork BotMan*” ini adalah sebagai berikut :

1. Sistem operasi windows 11
2. *Visual Studio Code*
3. *Xampp*
4. *Laravel Framework*

## 5. MySQL

## 6. BotMan FrameWork

### 5. Source Code

Pada source code ini akan menjelaskan beberapa kode yang ada pada aplikasi ini yang berisi tentang html dan php. Berikut penjelasan dan beberapa contoh kode tersebut.

#### a) Landing page user

- Landing page

```
1 <!-- container untuk header -->
<section class="py-4 pt-md-8 pb-8 pb-sm-11 mt-lg-0">
  <div class="container px-md-5">
    <div class="row">
      <div class="col-md-9 col-lg-7">
        <div class="fs-4 fs-md-6 fs-xl-7 sh-5 sh-lg-8 position-relative z-index-2">ApotekHusni <span
          class="fw-thin text-100">Cek obat mudah, </span><span class="text-400">kesehatan
          terjaga</span></div>
        </div>
      </div>
      <div class="row mt-md-6">
        <div class="col-2 d-none d-lg-block mt-auto"></div>
        <div class="col-1 mt-auto d-none d-lg-block"><a class="scroll-indicator text-warning"
          href="#footer"> <span>SCROLL </span></a></div>
        <div class="col-lg-7 position-relative">
          <div class="position-relative overflow-hidden overflow-visible">
          <div class="img-circle p-5 p-md-7 rounded-circle"></div>
          <div class="img-circle-2 p-5 p-md-7 rounded-circle"></div>
        </div>
        <div class="position-absolute bottom-080 col-11 mt-4 d-none d-md-block">
          <p class="fs-2 lh-1 text-800">kemudahan Cek Stok Obat Dengan Chatbot Apotek Husni Farma
            </p>
          <p class="text-900">Selamat datang di Apotek Husni Farma, solusi terpercaya untuk kebutuhan
            kesehatan Anda.
            Kami menyediakan berbagai jenis obat dan layanan kesehatan terbaik untuk Anda dan
            keluarga.
            Gunakan fitur Chatbot kami yang canggih untuk memeriksa ketersediaan obat secara
            real-time
            dan mendapatkan informasi kesehatan dengan cepat dan mudah. Apotek Husni Farma, untuk
            kesehatan Anda yang terjaga setiap saat.</p>
        </div>
        <div class="position-absolute start-0 bottom-0 d-lg-none"><a
          class="scroll-indicator text-warning href="#footer"> <span>SCROLL </span><span></a></div>
      </div>
      <div class="col-2 d-none d-lg-block sh-auto"></div>
    </div>
  </div>
</section>
```

Kode diatas merupakan tampilan untuk landing page apotek untuk user

- Konfigurasi chatbot widget di landing page

```

<script src="https://cdn.jsdelivr.net/npm/botman-web-widget@0/bulid/js/widget.js" defer>
</script>
//Di dalam tag <script> yang memiliki file JavaScript dari CDN (Content Delivery Network) untuk BotMan Web Widget.
//File ini merupakan fungsi-fungsi utama yang diperlukan untuk menjalankan dan mengoperasikan chatbot di halaman web.

var botmanWidget = { //variabel botmanWidget digunakan untuk mengorganisir widget chatbot.
  frameContainer: 'botmanChatbot', //frameContainer: Menentukan element atau ID tempat chatbot akan dibuat. Bisa contoh ini, chatbot diharapkan dapat diakses melalui javascript
  title: 'Chatbot', //title: judul atau nama untuk chatbot.
  iframeMessage: 'halo, Saya asocce chatbot, ada yang bisa saya bantu', //iframeMessage: Pesan pembantar yang ditampilkan saat chatbot pertama kali dibuat.
  widthColor: '#F0F0F0', //widthColor, backgroundColor, headerTextColor: Pengaturan warna untuk tampilan chatbot.
  widthText: '',
  backgroundColor: '#F0F0F0',
  headerTextColor: 'black',
};

window.addEventListener('message', function() {
  //Ini adalah event listener yang mendengarkan pesan yang dikirimkan dari chatbot.
  //window.addEventListener('message', ...) mengaktifkan fungsi yang akan dieksekusi setiap kali ada pesan (message) yang dikirimkan ke jendela (window).
  if (event.origin === window.location) {
    //event.origin === window.location: Memastikan bahwa pesan berasal dari sumber yang sama dengan halaman web saat ini.
    return;
  }
  var dataObject = event;
  data; //var dataObject = event.data; Menyimpan data pesan yang diterima dari chatbot.
  userReply(
    dataObject); //userReply(dataObject): Memanggil fungsi userReply dengan menggunakan data pesan untuk ditanggapi web saat ini.
});

function userReply(
  dataObject) { //userReply adalah fungsi yang digunakan untuk memproses pesan yang diterima dari chatbot.
  botmanWidget.openDataForm = 2000.toString(); //dataObject = "" + dataObject.value + "<!-->";
  //botmanWidget.openDataForm()...: Memanggil metode open dari botmanChatbot.js, yang digunakan untuk menampilkan pesan dari chatbot ke pengguna.
  //page data form "" + 2000.toString(); //dataObject = "" + dataObject.value + "<!-->"; Memuat elemen HTML pesan untuk menampilkan isi pesan dari chatbot.
  //2000.toString(); //Pengguna akan melihat pesan dengan string 2000 untuk diisi pesan ke dalam elemen input.
  //return;
}

```

Kode diatas merupakan variable untuk integrasi chatbot widget pada landing page apotek

## b) Chatbot (Framework BotMan)

- Chatbot widget

```

1 <!doctype html>
2 <html>
3
4 <head>
5   <title>BotMan Widget</title>
6   <meta charset="UTF-8">
7   <link href="/css/botman.css" rel="stylesheet">
8 </head>
9
10 <body>
11   <script id="botmanWidget" src="https://cdn.jsdelivr.net/npm/botman-web-widget@0/bulid/js/chat.js"></script>
12 </body>
13 </html>
14
15

```

Kode diatas memuat skrip JavaScript yang dibutuhkan untuk menjalankan BotMan dan menangani interaksi antara pengguna dan bot.

- BotMan Controller



```
Botman->hears('message', function (BotMan $bot, $message) use ($botman) {

    $productCategories = Category::all(); //Jika pesan adalah "halo", mengambil semua kategori produk dari model Category.
    if ($productCategories->isEmpty()) {
        $bot->reply('Maaf tidak ada obat yang tersedia untuk saat ini.');//Jika kategori kosong, membalas dengan pesan bahwa
    } else {
        $this->showCategory($botman, $productCategories);//Jika ada kategori, memanggil fungsi showCategory untuk menampilkan
    }
    $responseFound = true;

    if (strpos($message, '<span>' === 0) {
        if (preg_match('/<span[\"']*data-json-[\"']*([^\"]*)/i', $message, $matches)) {
            $jsonData = htmlspecialchars_decode($matches[1]);
            $data = json_decode($jsonData);//Jika pesan berisi elemen <span> dengan data JSON tersembunyi, decode JSON ters
            if (json_last_error() === JSON_ERROR_NONE) {
                if ($data->type == 'category') {
                    if ($responseFound = true;
                    $products = Produk::where('category_id', $data->id)->get();
                    if ($products->isEmpty()) {
                        $bot->reply('Maaf obat dengan kategori ' . $data->value . ' tidak tersedia.');//Jika tipe data adalah "category", mengambil semua produk dalam
                    } else {
                        $this->showObat($botman, $products);
                    }
                } elseif ($data->type == 'obat') {
                    $responseFound = true;
                    $product = Produk::where('id', $data->id)->first();
                    if (!$product) {
                        $bot->reply('Maaf obat ' . $data->value . ' tidak ditemukan.');//Jika tipe data adalah "obat", menampilkan detail obat dengan
                    } else {
                        $this->askForObat($botman, $product);
                    }
                }
            }
        }
    }

    //Terima melalui FAQ: foreach ($faq as $item) menunjukkan bahwa setiap item dalam daftar FAQ diproses satu per satu.
    //Terima melalui Pola Pertanyaan: foreach ($patterns as $pattern) menunjukkan bahwa setiap pola pertanyaan dalam item
    else {
        foreach ($faq as $item) {
            $patterns = explode('|', $item->pertanyaan);
            foreach ($patterns as $pattern) {
                if (preg_match('/$pattern/i', $message)) { // Menggunakan preg_match untuk pencocokan pola
                    $bot->reply($item->jawaban); // Respon ditemukan
                    $responseFound = true;
                    break 2; // Keluar dari kedua loop setelah menemukan respon
                }
            }
        }
    }
}

//Mengecek apakah pesan cocok dengan salah satu pola dalam FAQ dan membalas dengan jawaban yang sesuai.
```

```
if (!$responseFound) {
    $bot->reply('Saya tidak memahami pernyataan Anda.');//Jika tidak ada respon yang ditemukan, membalas dengan pesan bahwa bot tidak memahami pernyataan pengguna.
}
$botman->listen();

//tampilkan() @private
public function showCategory(Botman, $productCategories)
{
    $title = 'id="btn-category">divPilih kategori Obat</div>'; //Menyebut bagian HTML untuk judul kategori.
    $buttons = $title; //Menambahkan tombol kategori: setiap string kategori, membuat tombol dengan data JSON yang berisi informasi kategori.
    foreach ($productCategories as $category) { //Menyisipkan bilasan dengan tombol-tombol kategori kepada pengguna.
        $data = json_encode([
            'type' => 'category',
            'value' => $category->name,
            'id' => $category->id,
        ]);
        $button = 'div onclick="jarum.postMessage(' . htmlspecialchars($data) . ', window.location.origin)" class="btn"> . $category->name . '</div>';
    }
    $button .= '</div>';
    $botman->reply($button);
}

//tampilkan() @private
public function showObat(Botman, $products)
{
    $title = 'id="btn-obat">divPilih Obat</div>'; //Menyebut bagian HTML untuk judul pilihan obat.
    $buttons = $title; //Menyebut setiap produk, membuat tombol dengan data JSON yang berisi informasi produk.
    foreach ($products as $product) { //Menyisipkan bilasan dengan tombol-tombol produk kepada pengguna.
        $data = json_encode([
            'type' => 'obat',
            'value' => $product->name,
            'id' => $product->id,
        ]);
        $button = 'div onclick="jarum.postMessage(' . htmlspecialchars($data) . ', window.location.origin)" class="btn"> . $product->name . '</div>';
    }
    $button .= '</div>';
    $botman->reply($button);
}

//tampilkan() @private
public function askForObat(Botman, $product)
{
    $text = 'id="product-name">{product->name}</div>'; //Menampilkan detail obat seperti judul, stok, dan harga dalam format HTML.
    $text .= 'id="product-stock">{product->stock}</div>';
    $text .= 'id="product-price">{product->price}</div>';
    $text .= 'id="product-description">{product->description}</div>';
    $text .= 'id="category">{product->category->name}</div>';
    //Menyisipkan bilasan dengan detail obat tersebut kepada pengguna.
    $botman->reply($text);
}
```

Kode diatas digunakan untuk controller custom botman dalam mengatur percakapan antara bot dan pengguna

### c) Login Admin

```
0 references | 0 implementations
class Login extends Component implements HasForms
{
    use InteractsWithForms;
    use WithRateLimiting;

    0 references
    public $email = '';

    0 references
    public $password = '';

    0 references
    public $remember = false;

    0 references | 0 overrides
    public function mount(): void ...
    }

    0 references | 0 overrides
    public function authenticate(): ?LoginResponse ...
    }

    0 references | 0 overrides
    protected function getFormSchema(): array
    {
        return [
            TextInput::make('email')
                ->label(__('filament::login.fields.email.label'))
                ->email()
                ->required()
                ->autocomplete(),
            TextInput::make('password')
                ->label(__('filament::login.fields.password.label'))
                ->password()
                ->required(),
            Checkbox::make('remember')
                ->label(__('filament::login.fields.remember.label')),
        ];
    }

    0 references | 0 overrides
    public function render(): View
    {
        return view('filament::login')
            ->layout('filament::components.layouts.card', [
                'title' => __('filament::login.title'),
            ]);
    }
}
```

Kode diatas digunakan untuk mengonfigurasi atau menyesuaikan proses login ke admin panel.

### d) Dashboard Admin

```
<?php

namespace App\Filament\Resources\ProdukResource\Widgets;

use App\Models\Produk;
use Filament\Widgets\Widget;
use Filament\Widgets\StatsOverviewWidget\Card;
use Filament\Widgets\StatsOverviewWidget as BaseWidget;

0 references | 0 implementations
class StatsOverview extends BaseWidget
{
    //protected static string $view = 'filament.resources.produk-resource.widgets.stats-overview';
    0 references | 0 overrides
    protected function getCards(): array
    {
        return [
            Card::make('All Produk', Produk::all()->count())
                ->description('All Produk')
                ->descriptionIcon('heroicon-s-trending-up')
                ->chart([7, 2, 10, 3, 15, 4, 17])
                ->color('success')
                ->extraAttributes([
                    'class' => 'cursor-pointer',
                    'wire:click' => '$emitUp("setStatusFilter", "processed");'],
                );
        ];
    }
}
```

Kode diatas merupakan halaman dashboard yang menampilkan jumlah total produk yang ada pada akun admin.

e) CRUD category

- Database migration category

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('categories');
    }
};
```

Kode diatas digunakan untuk mendefinisikan dan mengelola skema database tabel category

-

- Controller category

```

0 references | 0 overrides
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('nama')
                ->required()
                ->maxLength(255),
        ]);
}

0 references | 0 overrides
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('nama')->label('Name')->sortable()->searchable(),
            Tables\Columns\TextColumn::make('description')->label('Description'),
            Tables\Columns\TextColumn::make('created_at')->label('Created At')->dateTime(),
            Tables\Columns\TextColumn::make('updated_at')->label('Updated At')->dateTime(),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\ViewAction::make(),
            Tables\Actions>EditAction::make(),
            Tables\Actions>DeleteAction::make(),
        ])
        ->bulkActions([
            Tables\Actions>DeleteBulkAction::make(),
        ]);
}

0 references | 0 overrides
public static function getRelations(): array
{
    return [
        //
    ];
}

0 references | 0 overrides
public static function getPages(): array
{
    return [
        'index' => Pages\ListCategories::route('/'),
        'create' => Pages\CreateCategory::route('/create'),
        'edit' => Pages>EditCategory::route('/{record}/edit'),
    ];
}

```

Kode diatas digunakan untuk controller mengatur routing yang akan dilakukan yang mencakup CRUD untuk entitas aplikasi

- Model category

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

8 references | 0 implementations
class Category extends Model
{
    use HasFactory;

    0 references
    protected $fillable = ['nama'];

    0 references | 0 overrides
    public function Produk()
    {
        return $this->hasMany(Produk::class);
    }
}

```

Kode diatas digunakan untuk menampilkan data yang sesuai dengan settingan controller

f) CRUD Produk

- Database migration Produk

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('produks', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('code')->unique();
            $table->foreignId('category_id')->constrained()->onDelete('cascade');
            $table->text('description')->nullable();
            $table->string('manufacturer')->nullable();
            $table->integer('stock')->default(0);
            $table->decimal('price', 8, 2);
            $table->date('expiry_date')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('produks');
    }
};
```

Kode diatas digunakan untuk mendefinisikan dan mengelola skema database tabel produk

- Controller Produk

```
0 references 0 overrides
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('name')->label('Nama'),
            Forms\Components\TextInput::make('code')->label('Kode')->defaultValue(function(){ return \Str::random(3);}),
            Forms\Components\Select::make('category_id')->relationship('category', 'name'),
            Forms\Components\TextEditor::make('description')->label('Deskripsi')->nullable(),
            Forms\Components\TextEditor::make('manufacturer')->label('Pabrikan')->nullable(),
            Forms\Components\TextEditor::make('stock')->label('Stok')->defaultValue(0),
            Forms\Components\TextEditor::make('price')->label('Harga')->defaultValue(0000000000),
            Forms\Components\DatePicker::make('expiry_date')->label('Tanggal Kedaluwarsa')->nullable(),
        ])
}

0 references 0 overrides
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('name')->label('Nama')->sortable()->searchable(),
            Tables\Columns\TextColumn::make('code')->label('Kode'),
            Tables\Columns\TextColumn::make('category.name')
                ->sortable()
                ->searchable(),
            Tables\Columns\TextColumn::make('description')->label('Deskripsi'),
            Tables\Columns\TextColumn::make('manufacturer')->label('Pabrikan'),
            Tables\Columns\TextColumn::make('stock')->label('Stok'),
            Tables\Columns\TextColumn::make('price')->label('Harga'),
            Tables\Columns\TextColumn::make('expiry_date')->label('Tanggal Kedaluwarsa')->date(),
            Tables\Columns\TextColumn::make('created_at')->label('Dibuat Pada')->dateTime(),
            Tables\Columns\TextColumn::make('updated_at')->label('Diperbarui Pada')->dateTime(),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\ViewAction::make(),
            Tables\Actions\EditAction::make(),
            Tables\Actions\DeleteAction::make(),
        ])
        ->bulkActions([
            Tables\Actions\DeleteBulkAction::make(),
        ])
}

0 references 0 overrides
public static function getRelations(): array
{
    return [
        //
    ];
}
```

Kode diatas digunakan untuk controller mengatur routing yang akan dilakukan yang mencakup CRUD untuk entitas aplikasi

- Model Produk

```
0 references | 0 migrations
class Produk extends Model
{
    use HasFactory;

    0 references
    protected $fillable = [
        'name', 'code', 'category_id', 'description', 'manufacturer', 'stock', 'price', 'expiry_date'
    ];

    0 references | 0 overrides
    public function Category()
    {
        return $this->belongsTo(Category::class);
    }
}
```

Kode diatas digunakan untuk menampilkan data yang sesuai dengan settingan controller

g) CRUD pengadaan produk

- Database migration pengadaan produk

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('pengadaan_produk', function (Blueprint $table) {
            $table->id();
            $table->string('nama_produk');
            $table->integer('jumlah');
            $table->decimal('harga_satuan', 10, 2);
            $table->date('tanggal_pengadaan');
            $table->foreignId('supplier_id')->constrained()->onDelete('cascade');
            $table->text('keterangan')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('pengadaan_produk');
    }
};
```

Kode diatas digunakan untuk mendefinisikan dan mengelola skema database tabel pengadaan produk



- CRUD pengadaan produk

```
class PengadaanProdukResource extends Resource
{
    #resources
    protected static $string Model = 'PengadaanProdukResource';

    #resources
    protected static $string RecordIdentifier = 'nama_produk';

    #resources
    protected static $string NavigationIcon = 'heroicon-o-collection';

    #resources
    protected static $string NavigationGroup = 'tentang Supplier';

    #resources Controller
    public static function form(Form $form): Form
    {
        return $form
            >>schema()
                >>formComponents(TextInput::make('nama_produk')
                    >>required()
                    >>maxLength(255),
                    >>formComponents(TextInput::make('jumlah')
                    >>required()
                    >>numeric(),
                    >>formComponents(TextInput::make('harga_satuan')
                    >>required()
                    >>numeric(),
                    >>formComponents(DatePicker::make('tanggal_pengadaan')
                    >>required(),
                    >>formComponents(Select::make('supplier_id')
                    >>relationship('supplier', 'nama_supplier')
                    >>required(),
                    >>formComponents(TextInput::make('keterangan')->required(),
                    >>formComponents(Select::make('kategori')->required(),
                    >>relationship('kategori', 'nama')
                    >>required(),
                );
    }

    #resources Controller
    public static function table(Table $table): Table
    {
        return $table
            >>columns([
                Table\Column::make('nama_produk')->label('Nama Produk')->sortable()->searchable(),
                Table\Column::make('jumlah')->label('Jumlah'),
                Table\Column::make('harga_satuan')->label('Harga Satuan'),
                Table\Column::make('supplier.nama_supplier')
                >>sortable(),
                >>searchable(),
                Table\Column::make('tanggal_pengadaan')->label('Tanggal Pengadaan')->date(),
                Table\Column::make('keterangan')->label('Keterangan'),
                >>searchable(),
                Table\Column::make('created_at')->label('Dibuat Pada')->datetime(),
                Table\Column::make('updated_at')->label('Diperbarui Pada')->datetime(),
            ])
            >>filters([
                //
            ])
            >>actions([
                Table\Actions\ViewAction::make(),
                Table\Actions\EditAction::make(),
                Table\Actions>DeleteAction::make(),
            ])
            >>bulkActions([
                Table\Actions>DeleteBulkAction::make(),
            ]);
    }
}
```

Kode diatas digunakan untuk controller mengatur routing yang akan dilakukan yang mencakup CRUD untuk entitas aplikasi

- Model pengadaan produk

```

class PengadaanProduk extends Model
{
    use HasFactory;

    0 references
    protected $fillable = [
        'nama_produk',
        'jumlah',
        'harga_satuan',
        'tanggal_pengadaan',
        'supplier_id',
        'keterangan',
        //'Surat Pesanan'
    ];

    0 references | 0 overrides
    public function supplier()
    {
        return $this->belongsTo(Supplier::class);
    }
}

```

Kode diatas digunakan untuk menampilkan data yang sesuai dengan settingan controller

#### h) CRUD Supplier

- Database migration supplier

```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('suppliers', function (Blueprint $table) {
            $table->id();
            $table->string('nama_supplier');
            $table->string('alamat');
            $table->string('telepon');
            $table->string('email')->unique();
            $table->text('keterangan')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('suppliers');
    }
};

```

Kode diatas digunakan untuk mendefinisikan dan mengelola skema database tabel supplier

- CRUD supplier

```
4 Controller | Implementation
class SupplierResource extends Resource
{
  #attributes
  protected $attr = ['index' => Supplier::class,];

  #attributes
  protected $attr = ['show' => Supplier::class,];

  #attributes
  protected $attr = ['show' => Supplier::class,];

  #references | 0 overrides
  public static function form($form $form): Form
  {
    return $form
    {
      return $form
      {
        -required()
        -width(200)
        -label('Nama Supplier'),
        -required()
        -width(200)
        -label('Alamat'),
        -required()
        -width(200)
        -label('Telepon'),
        -required()
        -width(200)
        -label('Email'),
        -required()
        -width(200)
        -label('Keterangan'),
      }
    };
  }

  #references | 0 overrides
  public static function table($table): Table
  {
    return $table
    {
      -width(200)
      -label('Nama Supplier')->sortable()-sortable(),
      -width(200)
      -label('Alamat'),
      -width(200)
      -label('Telepon'),
      -width(200)
      -label('Email'),
      -width(200)
      -label('Keterangan'),
      -width(200)
      -label('Dibuat Pada')->datetime(),
      -width(200)
      -label('Diubah Pada')->datetime(),
    }
    {
      //
    }
    {
      -width(200)
      -width(200)
      -width(200)
      -width(200)
    }
  }
}
}
```

Kode diatas digunakan untuk controller mengatur routing yang akan dilakukan yang mencakup CRUD untuk entitas aplikasi

- Model supplier

```
4 Entity | Implementation
class Supplier extends Model
{
  use HasFactory;

  #references
  protected $fillable = [
    'nama_supplier',
    'alamat',
    'telepon',
    'email',
    'keterangan',
  ];

  #references | 0 overrides
  public function pengadaanProduk()
  {
    return $this->hasMany(PengadaanProduk::class);
  }
}
```

Kode diatas digunakan untuk menampilkan data yang sesuai dengan settingan controller

## Lampiran 6 Sertifikat HKI yang Terbit

  
**REPUBLIK INDONESIA**  
**KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA**

### SURAT PENCATATAN CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202465306, 15 Juli 2024

**Pencipta**

Nama : **Tri Hesti Wahyu ningsih, Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom. dkk**

Alamat : **Dukuh Kembangpang Desa Plumbungan RT/RW 04/01 Kec. Kramat Kab. Tegal Provinsi Jawa Tengah 52181, Kramat, Tegal, Jawa Tengah, 52181**

Kewarganegaraan : **Indonesia**

**Pemegang Hak Cipta**

Nama : **Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama**

Alamat : **Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142, Margadana, Tegal, Jawa Tengah 52142**

Kewarganegaraan : **Indonesia**

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **VIRTUAL ASISSTANT PADA SISTEM INVENTORY APOTEK HUSNIFARMA DENGAN FRAMWORK BOTMAN**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : **15 Juli 2024, di Tegal**

Jangka waktu pelindungan : **Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.**

Nomor pencatatan : **000640661**

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.  
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA  
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL  
u.b  
Direktur Hak Cipta dan Desain Industri

  
**IGNATIUS M.T. SILALAH**  
NIP. 196812301996031001



Disclaimer:  
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

**LAMPIRAN PENCIPTA**

No	Nama	Alamat
1	Tri Hesti Wahyuningsih	Dukuh Kembangpang Desa Plumbungan RT/RW 04/01 Kec. Kramat Kab. Tegal Provinsi Jawa Tengah 52181, Kramat, Tegal
2	Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.	Jln. Raya Kluwut Timur No. 24 Kec.Bulakamba Kab. Brebes Provinsi Jawa Tengah 52253, Bulakamba, Brebes
3	Dyah Apriliani, S.T., M.Kom.	Perumahan Mutiara Indah Blok C4, Jalan Nangka Gang 2 Rt 02 Rw 02 Kel. Procot Kec. Slawi Kab.Tegal Provinsi Jawa Tengah 52412, Slawi, Tegal





Lampiran 7 Lembar Bimbingan



SARJANA TERAPAN TEKNIK INFORMATIKA  
POKITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Tri Hesti Wahyuningsih  
 Nim : 20090082  
 No. Ponsel : 085225778051  
 Judul TA : Virtual Asistant Pada Sistem Inventory Apotek Husni  
 Farma Tegal Dengan Algoritma Sequential Search  
 Dosen Pembimbing I : Ir. Ginanjar Wiro Sasmito, M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	21/3/24	proposai skripsi	perbaikan	f.
2.	09/09/2024	proposal skripsi	oke. lanjut ke UI/UX	f.
3.	22/09/2024	konsep chat bot	lanjut	f.
4.	30/24/5	chatbot	oke. lanjut.	f.
5.	1/7/24	prohan	oke.	f.
6.	12/7/24	Laporan	revisi	f.





**SARJANA TERAPAN TEKNIK INFORMATIKA  
POKITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
7.	17/21 /7	Laporan	Revisi	b.
		Laporan	Revisi	b.
		Ok.	yang uji	b.

Tegal, 2024  
Dosen Pembimbing I

Ir. Ghanjar Wiro Sasmito, M.Kom.  
NIPY. 10.007.032



SARJANA TERAPAN TEKNIK INFORMATIKA  
POKITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Tri Hesti Wahyuningsih  
Nim : 20090082  
No. Ponsel : 085225778051  
Judul TA : Virtual Asistant Pada Sistem Inventory Apotek Husni  
Farma Tegal Dengan-Algorithm Sequential Search  
Dosen Pembimbing II : Dyah Aprilianti, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	21/3/2024		- Data → dilengkapi - Penambahan roll	
2.	3/4/2024	Aplikasi	- Data → dilengkapi - Aplikasi dibuat dengan kepi lagi → Wiley.	
3.	22/4/2024	Aplikasi	- Konfirmasi - Aplikasi dilanjutkan front end - Back end.	
4.	30/4/2024	Aplikasi	- Data dilengkapi manajemen obat dengan - dilengkapi	
5.	28/6/2024	Aplikasi	- Data dilengkapi. Aplikasi.	



SARJANA TERAPAN TEKNIK INFORMATIKA  
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
6.	7/2024	Aplikasi	- lengkapi aplikasi - layout laporan	[Signature]
7.	17/2024	Aplikasi	- doc manual + foto foto - book - pendayagunaan	[Signature]
8.	17/2024	Laporan	Revisi Lap.	[Signature]
9.	17/2024	Laporan	Acc	[Signature]



**SARJANA TERAPAN TEKNIK INFORMATIKA  
POKITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing

Tegal, 2024  
Dosen Pembimbing I

Dyah Apriliani, S.T., M.Kom.  
NIPY.09.015.225