

LAMPIRAN

Lampiran 1. Lembar Kesediaan Pembimbing

PEMBIMBING 1

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Muhammad Haidar Rasyiq
NIM : 20090087
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 28 maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan **minimal bimbingan satu kali dalam seminggu**. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 28 Maret 2024

Pihak Pertama



Muhammad Haidar Rasyiq

Pihak Kedua



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

Mengetahui
Ketua Program Studi Sajana Terapan Teknik Informatika



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

PEMBIMBING 2

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan dibawah ini :

Pihak Pertama

Nama : Muhammad Haidar Rasyiq
NIM : 20090087
Program Studi : Teknik Informatika

Pihak Kedua

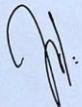
Nama : Arif Hidayah, M.Tr.Kom.
Status : Dosen/Praktisi
NIDN : -
Jabatan Fungsional : -
Pangkat/Golongan : -

Pada hari ini senin tanggal 29 april 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing 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, 29 April 2024

Pihak Pertama



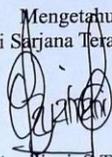
Muhammad Haidar Rasyiq

Pihak Kedua



Arif Hidayah, M.Tr.Kom.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



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

Lampiran 2. Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta :

1. Nama : Muhammad Haidar Rasyiq
Kewarganegaraan : Indonesia
Alamat : Jln Cissanggarung Dalam No.06 Rt.4/Rw.4, Kec. Brebes,
Kab.Brebes, Jawa Tengah, ID. 52217
2. Nama : Dyah Apriliani, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Blok C4, Jalan Nangka Gang 2 Rt 02
Rw 02 Kel. Procot Kec. Slawi Kab. Tegal
3. Nama : Arif Hidayah, M.Tr.Kom
Kewarganegaraan : Indonesia
Alamat : Jl. Bulu Emas No. 08 RT.01 RW.01 Kel. Lawatan
Kec. Dukuhturi, Kab. Tegal

Dengan ini menyatakan bahwa:

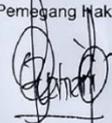
1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Aplikasi Rekomendasi Usaha Mikro, Kecil dan Menengah (UMKM) Kuliner Berbasis Website Menggunakan Algoritma KNN
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya 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.
4. 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, 12 Maret 2025



Muhammad Haidar Rasyiq
Pemegang Hak Cipta*



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



Arif Hidayah, M.Tr.Kom
Pemegang Hak Cipta*

Pemegang Hak Cipta*

* Semua pemegang hak cipta agar menandatangani di atas materai.

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Muhammad Haidar Rasyiq
Kewarganegaraan : Indonesia
Alamat : Jln. Cissanggarung Dalam No.06 Rt.4/Rw.4, Kec. Brebes,
Kab.Brebes, Jawa Tengah, ID. 52217
2. Nama : Dyah Apriliani, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Blok C4, Jalan Nangka Gang 2 Rt 02
Rw 02 Kel. Procot Kec. Slawi Kab. Tegal
3. Nama : Arif Hidayah, M.Tr.Kom
Kewarganegaraan : Indonesia
Alamat : Jl. Bulu Emas No. 08 RT.01 RW.01 Kel. Lawatan
Kec. Dukuhturi, Kab. Tegal

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 **"IMPLEMENTASI CHATBOT PADA SISTEM INFORMASI ORGANISASI MAHASISWA BERBASIS WEBSITE (STUDI KASUS : ORGANISASI MAHASISWA POLITEKNIK HARAPAN BERSAMA TEGAL)"** untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

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

Pemegang Hak Cipta
Kepala P3M

(Muhammad Fikri Hidayatullah, S.T., M.Kom)



Tegal, 12 Maret 2025
Pencipta



(Muhammad Haidar Rasyiq)

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

(Arif Hidayah, M.Tr.Kom)

Lampiran 4. Syarat Pengajuan HKI



IMPLEMENTASI CHATBOT PADA SISTEM
INFORMASI ORGANISASI MAHASISWA
BERBASIS WEBSITE
TECHNICAL BOOK

Muhammad Haidar Rasyiq, Dyah Apriliani., M.Kom., Arif Hidayah, M.Tr.Kom.

Sarjana Terapan Teknik Informatika

A. Profile

Website ini dirancang untuk menjadi platform yang menyatukan seluruh informasi terkait organisasi mahasiswa di Politeknik Harapan Bersama Tegal. Website ini memiliki fitur diantaranya lain yaitu pendaftaran calon anggota ormawa, arsip document, berita kegiatan setiap unit, jadwal kegiatan ukm, chatbot. Dengan fitur chatbot yang terintegrasi, website ormawa tidak hanya menyajikan informasi yang diperlukan oleh mahasiswa, tetapi juga memberikan pengalaman interaktif yang memudahkan mahasiswa dalam mencari informasi terkait kegiatan, acara, keanggotaan, dan berbagai informasi penting lainnya.

B. Latar Belakang

Website dapat menyediakan berbagai informasi dalam bentuk teks, suara, gambar, video, file dan lain-lain (Hidayat, 2010). Pemanfaatan website digunakan oleh kampus untuk meningkatkan keunggulan kompetitif dan pelayanan baik dilingkungan kampus maupun luar. Website Politeknik Harapan Bersama yaitu <https://poltekharber.ac.id> memiliki berbagai informasi seperti tentang kampus, akademik, penerimaan mahasiswa, layanan, kontak dan lain-lain. Hal ini membuat website sangat penting dalam memenuhi kebutuhan informasi dan layanan yang menunjang kegiatan kampus. Akan tetapi, informasi organisasi mahasiswa pada website tersebut tidak disajikan. Sehingga, dapat menimbulkan kebingungan yang menyebabkan ketidakpuasan pengguna, terutama pengguna yang baru pertama kali berkunjung ke website.

Organisasi mahasiswa (Ormawa) di Politeknik Harapan Bersama Tegal terdiri dari beberapa unit yaitu Badan Perwakilan Mahasiswa (BPM), Badan Eksekutif Mahasiswa (BEM), Himpunan Mahasiswa Prodi (HMP), dan Unit Kegiatan Mahasiswa (UKM). Ormawa KMPHB (Keluarga Mahasiswa Politeknik Harapan Bersama Tegal) ini adalah sebuah wadah yang bergerak di bidang pelayanan informasi dan kegiatan bagi mahasiswa Politeknik Harapan Bersama Tegal, akan tetapi dalam hal keaktifan anggota pada setiap unit susah buat diketahui dan dalam penyimpanan dokumen proposal kegiatan, laporan pertanggung jawaban kurang efektif.

Sebagian besar mahasiswa ingin mendapat informasi yang mudah, cepat dan akurat. Pendaftaran calon anggota pada setiap unit organisasi mahasiswa memerlukan banyak informasi prosedur pendaftaran di setiap unitnya. Di Politeknik Harapan Bersama Tegal sarana informasi pendaftaran membutuhkan komunikasi dua arah untuk menanyakan informasi lebih detail. Calon anggota yang ingin mendaftar juga tentu mencari informasi lebih dahulu tentang setiap unit pada organisasi mahasiswa yang ada di Politeknik Harapan Bersama tersebut. Untuk mendukung kebutuhan informasi yang akurat dan update terkait

dengan pendaftaran mahasiswa baru, maka diperlukan sebuah sistem informasi berbasis website yang dilengkapi dengan chatbot dapat merangkum dengan baik berbagai data pendaftaran serta menampilkan informasi lainnya pada pengguna.

Chatbot adalah sebuah program yang dibuat berbasis Artificial Intelligence (AI) atau kecerdasan buatan, yang dapat menggenerasi sebuah percakapan atau obrolan dengan pengguna lain layaknya manusia melalui situs web. Chatbot sendiri bisa dijadikan alternatif untuk memberikan informasi mengenai masalah pendaftaran calon anggota pada setiap unit sehingga bisa mengarahkan mahasiswa untuk mendaftar dan dokumen apa saja yang perlu dikumpulkan sebagai persyaratan mendaftar. Chatbot rule-based dipilih dikarenakan bisa merespon berdasarkan aturan yang telah ditetapkan terlebih dahulu, berdasarkan pengenalan bentuk leksikal teks masukan tanpa membuat jawaban teks baru. Pengetahuan yang digunakan dalam chatbot adalah data atau informasi yang didapatkan secara manual dan diatur serta disajikan dengan pola percakapan. Basis data dengan aturan yang lebih komprehensif memungkinkan chatbot membalas ke lebih banyak jenis input pengguna (Adamopoulou & Moussiades, 2020).

Chatbot ini sederhana karena tidak menggunakan machine learning sehingga, lebih mudah untuk dilakukan pembaruan pada databasenya dan lebih cepat merespon pertanyaan pengguna. Machine learning tidak genting untuk digunakan karena setiap pertanyaan memiliki jawaban, tidak memiliki kandidat-kandidat jawaban seperti pada sistem. Chatbot dapat memberikan pilihan topik, setelah pengguna menanyakan sesuatu. Penawaran topik pada chatbot berupa button/menu untuk mengarahkan ke topik lebih detail atau jawaban dengan cepat.

Dari apa yang sudah dipaparkan diatas maka website ini hadir untuk membantu aktifitas organisasi mahasiswa menjalankan kewajibannya yaitu melayani, menyampaikan dan membina mahasiswa/mahasiswi di Politeknik Harapan Bersama Tegal.

C. Manfaat

Website ini membantu dalam menyimpan dan mengelola data anggota, kegiatan, dan administrasi organisasi mahasiswa secara terstruktur dan terorganisir.

D. Spesifikasi Teknis

Spesifikasi teknis meliputi:

- Source Code

Berikut uraian spesifikasi untuk pembangunan Website Organisasi Mahasiswa:

1. Visual Studio Code
2. Laravel
3. HTML
4. PHP
5. JavaScript
6. Web Browser

E. Source Code

a. Bagian Chatbot

- Main.blade.php

```
{{!-- Chatbot --}}
<button class="chatbot-toggler">
  <span class="material-symbols-outlined">chat</span>
  <span class="material-symbols-outlined">close</span>
</button>
<div class="chatbot">
  <header>
    <h2>Chatbot</h2>
  </header>
  <ul class="chatbox active">
    <li class="chat incoming">
      <span class="material-symbols-outlined">smart_toy</span>
      <p>Hai, Silahkan bertanya seputar Organisasi👋👋</p>
    </li>
  </ul>
  <div class="chat-input">
    <textarea placeholder="Masukan angka sesuai urutan pertanyaan" required</textarea>
    <span id="send-btn" class="material-symbols-outlined">send</span>
  </div>
</div>
```

1) Button Toggle Chatbot

Sebuah tombol dengan dua ikon : satu untuk membuka (chat) dan satu untuk menutup (close) chatbot.

2) Chatbot dimulai dengan header "Chatbot"

3) chatbox: Daftar pesan yang muncul dalam chatbot, disini ada sebuah pesan awal yaitu "Hai, silahkan bertanya seputar organisasi" yang akan muncul pertama kali ketika chatbot dibuka.

4) Chat-input : Sebuah textarea untuk mengetikkan pertanyaan atau pesan yang akan dikirim ke chatbot. Ada juga tombol kirim dengan ikon "send"

```

{{!-- Footer or additional scripts --}}
<script src="{{ asset(path: 'js/landing.js') }}"></script>

```

Landing.js dimuat untuk menambahkan interaktivitas atau fungsi lainnya pada halaman, seperti fungsi chatbot. Asset() adalah helper di laravel untuk memanggil file direktori public

- Landing.js

```

let orgs = document.querySelectorAll('.list-org');
let title = document.querySelector('.title');
const units = document.querySelectorAll('.container-unit');
const dropdown = document.querySelector('.dropdown');
const links = document.querySelectorAll('header a');

```

- 1) querySelectorAll() ini digunakan untuk memilih semua elemen yang cocok dengan selector yang diberikan, dan mengembalikannya dalam bentuk NodeList.
- 2) querySelector() ini digunakan untuk memilih elemen pertama yang cocok dengan selector yang diberikan.
- 3) Ini berarti, elemen dengan kelas .list-org, .title, .container-unit, dan .dropdown diambil, secara bersamaan dengan semua tautan (<a>) yang ada di elemen <header>.

```

links.forEach((link) => {
  link.addEventListener('click', () => {
    links.forEach(link => link.classList.remove('active'));
    link.classList.add('active');
    document.body.classList.remove('show-chatbot');
    document.querySelector('.dropdown').classList.remove('active');
  })
});

```

- 1) Menggunakan forEach() digunakan untuk iterasi melalui semua tautan.
- 2) Pada setiap tautan, ditambahkan Event Listener untuk mendeteksi klik. Ketika tautan diklik, beberapa Tindakan dilakukan :
 - Menghapus kelas active dari semua tautan untuk memastikan hanya satu tautan yang aktif setiap saat.
 - Menambahkan kelas active ke tautan yang diklik, menandakan bahwa tautan tersebut adalah yang saat ini aktif.
 - Menghilangkan kelas show-chatbot dari body, yang mungkin terkait dengan menampilkan atau menyembunyikan chatbot di halaman.

- Menghilangkan kelas active dari dropdown, yang menunjukkan bahwa dropdown tidak lagi aktif.

```
document.querySelector('.showOrg').addEventListener('click', function(){
  dropdown.classList.toggle('active');
});
```

- 1) Kode tersebut mencari elemen pertama yang memiliki kelas .showOrg menggunakan document.querySelector().
- 2) Setelah elemen tersebut ditemukan, kode menambahkan Event Listener untuk menemukan “click”. Ini berarti Ketika elemen .showOrg diklik, fungsi yang ditentukan akan dijalankan.
- 3) Di dalam fungsi yang di eksekusi ketika elemen diklik, kode dropdown.classList.toggle('active'); digunakan, fungsi toggle ini memiliki dua efek :
 - Jika elemen yang diwakili oleh dropdown sudah memiliki kelas active, maka kelas tersebut akan dihapus (dihilangkan).
 - Jika elemen tidak memiliki kelas active, maka kelas tersebut akan ditambahkan.

```
function toggleLink() {
  units.forEach((unit) => unit.classList.remove('active'));
  let header = this.dataset.name;
  title.innerHTML = `INFORMASI KEGIATAN<br> UNIT ${header.toUpperCase()}`;
  document.querySelector(`#${header}`).classList.add('active');
  document.querySelector('.dropdown').classList.remove('active');
}
```

fungsi toggleLink() ini digunakan untuk memperbarui tampilan halaman ketika pengguna mengklik tautan untuk memilih unit tertentu. Ini mengubah tampilan judul, menandai unit yang dipilih sebagai aktif, dan menyembunyikan dropdown jika sedang ditampilkan.

```

orgs.forEach((org) => {
  document.body.classList.remove('show-chatbot');
  org.addEventListener('click', toggleLink);
})

document.querySelector('.dropdown-options li:last-child').addEventListener('click', () => {
  dropdown.classList.remove('active');
});

```

- 1) Kode ini menggunakan forEach() untuk mengiterasi melalui semua elemen yang disimpan dalam variabel orgs, yang diambil sebelumnya menggunakan document.querySelectorAll('.list-org').
- 2) Dengan setiap elemen dalam daftar orgs, kode akan melakukan aksi tertentu dalam blok {} yang akan mengikuti.
- 3) Di dalam fungsi callback dari forEach, kode ini akan langsung menghapus kelas show-chatbot dari elemen body setiap kali salah satu elemen org diklik.
- 4) Setelah menghapus kelas show-chatbot, kode ini menambahkan Event Listener pada setiap elemen org.
- 5) Sebuah Event Listener ditambahkan untuk mendeteksi klik pada elemen terakhir ini. Ketika elemen terakhir diklik, kelas active akan dihapus dari dropdown dengan kode dropdown.classList.remove('active');. Ini berfungsi untuk menyembunyikan dropdown ketika opsi terakhir dipilih.

```

const chatbotToggler = document.querySelector('.chatbot-toggler');
chatbotToggler.addEventListener('click', () => {
  document.body.classList.toggle('show-chatbot');
});

```

- 1) Kode diatas mencari elemen pertama dihalaman yang memiliki kelas .chatbot-toggler dan menyimpannya dalam variable chatbotToggler.
- 2) Menambahkan sebuah Event Listener pada elemen chatbotToggler. Event Listener ini akan mendengar peristiwa klik. Ketika pengguna mengklik elemen ini, fungsi di dalam blok {} akan dijalankan.
- 3) Di dalam fungsi yang dijalankan saat klik terjadi, kode ini menggunakan metode toggle() untuk kelas show-chatbot pada document.body.
- 4) Fungsi toggle() berfungsi untuk menambahkan kelas show-chatbot jika kelas tersebut belum ada. Sebaliknya, jika kelas show-chatbot sudah ada, maka fungsi ini akan menghapusnya. Dengan demikian, ini akan mengubah tampilan

chatbot antara terlihat dan tersembunyi setiap kali pengguna mengklik elemen pengubah.

```
const chatInput = document.querySelector('.chat-input textarea');
const senderChat = document.querySelector('.chat-input span');
const chatBox = document.querySelector('.chatbox');
const botChat = document.querySelector('.chatbox incoming');
```

- 1) Kode diatas mencari elemen textarea (area pengetikan) didalam elemen yang memiliki kelas .chat-input. Elemen ini digunakan untuk memasukan teks chat oleh pengguna.
- 2) Mengambil elemen span yang ada di dalam .chat-input. Biasanya, elemen ini berfungsi sebagai tombol atau indikator untuk mengirim pesan, yang dapat berisi teks atau ikon yang menunjukkan fungsi pengiriman.
- 3) Elemen dengan kelas .chatbox, yang merupakan area tempat semua pesan chat ditampilkan. Ini adalah bagian utama dari antarmuka chat di mana pesan pengguna dan chatbot akan muncul.
- 4) Mencari elemen dalam galeri .chatbox yang memiliki kelas .incoming, yang biasanya digunakan untuk menandai pesan-pesan yang dikirim oleh chatbot. Ini bisa berarti elemen yang akan menampilkan chat dari bot ketika bot memberikan respons kepada pengguna.

```
let answers = [];
answers['visi'] = ['Dengan visi kami "Menjadikan BEM Politeknik Harapan Bersama Sebagai Wadah Untuk M'];
answers['misi'] = ['Misi: <br>1. Mendorong Pengembangan Kualitas Sumber daya Mahasiswa PHB. <br>2. Me'];
answers['ciri'] = 'Terdapat beberapa ciri organisasi antara lain terdiri dari dua orang atau lebih, m';
answers['struktur'] = [
```

- 1) Kata kunci let untuk mendeklarasikan variable baru yang Bernama answers. Setelah nama variabel, tanda sama dengan = digunakan untuk menginisialisasi answers dengan sebuah array kosong. Dengan kata lain, variabel answers sekarang adalah sebuah array yang tidak memiliki elemen di dalamnya saat diciptakan.
- 2) Elemen dengan kunci visi ditambahkan ke dalam variabel answers. Nilai yang diberikan adalah sebuah array yang berisi dua string, masing-masing mendeskripsikan visi di mana informasi tersebut menggunakan HTML seperti
 untuk menyisipkan baris baru.

- 3) Elemen dengan kunci misi ditambahkan, yang juga berupa array. Array ini berisi dua string mendetail tentang misi, di mana setiap misi ditulis dengan HTML untuk format tampilan yang lebih baik.
- 4) Kunci ciri ditambahkan dengan sebuah string yang menjelaskan karakteristik umum dari sebuah organisasi. Ini disimpan sebagai string tunggal, tidak dalam array.
- 5) Elemen dengan kunci struktur ditambahkan, yang juga berupa array. Array ini berisi dua string mendetail tentang struktur, Dimana setiap struktur ditulis dengan HTML untuk format tampilan yang lebih baik.

```

console.log(answers['tugas']);
answers['pengertian'] = 'Menurut Prof. Dr. Sondang P. Siagian, organisasi adalah suatu bentuk persekutuan
answers['daftar'] = 'untuk mendaftar sebagai anggota baru, kamu bisa klik <a href="/register" class=
answers['regist'] = 'untuk mendaftar sebagai anggota baru, kamu bisa klik <a href="/register" class=
answers['tujuan'] = 'Tujuan dibentuknya organisasi secara umum antara lain meningkatkan kemandirian,
const defaultAnswer = 'Maaf, untuk saat ini kami belum bisa menjawab pertanyaan itu, silahkan bertanya
let userMessage;
console.log(answers['visi'][1])

```

- 1) Fungsi console.log() digunakan untuk mencetak atau menampilkan nilai dari kata kunci tugas ke konsol, sehingga dapat melihat secara langsung apa isi dari answers['tugas'] pada saat kode itu dijalankan.
- 2) Menambahkan kata kunci pengertian ke dalam objek answers dan memberikan nilai berupa string.
- 3) Menambahkan kata kunci daftar yang berisi pesan instruksi untuk mendaftar sebagai anggota baru.
- 4) Menambahkan kata kunci regist dengan nilai yang persis sama seperti yang ada pada kunci daftar.
- 5) Menambahkan kata kunci tujuan ke dalam objek answers dan memberikan nilai berupa string.
- 6) Mendefinisikan sebuah variabel defaultAnswer yang menyimpan string sebagai respons standar ketika pertanyaan yang diajukan tidak dapat dijawab.
- 7) Mendeklarasikan sebuah variabel bernama userMessage menggunakan kata kunci let, namun variabel ini belum diisi dengan nilai apapun, sehingga saat ini userMessage akan berisi undefined. Variabel ini mungkin akan digunakan nanti untuk menyimpan pesan atau informasi dari pengguna.
- 8) Mengambil nilai yang terkait dengan kunci visi dari objek answers, kemudian menggunakan notasi indeks [1], kode ini mengakses elemen kedua dalam array

yang terdapat di bawah kunci visi. Selanjutnya, nilai yang diambil tersebut dicetak ke konsol menggunakan `console.log()`. Ini memungkinkan Anda untuk melihat informasi yang terdapat di elemen kedua array visi.

```
const createChatLi = (message, className) => {
  const chatLi = document.createElement('li');
  chatLi.classList.add('chat', className);
  let chatContent = className === 'outgoing' ? `<p>${message}</p>` : `<span class="material-symbols-outlined">smart_toy</span><p>${message}</p>`;
  chatLi.innerHTML = chatContent;
  return chatLi;
};
```

- 1) Di sini, kita mendeklarasikan sebuah fungsi bernama `createChatLi` menggunakan notasi arrow function (`=>`). Fungsi ini menerima dua parameter: `message` dan `className`. `message` sendiri adalah teks pesan yang akan ditampilkan dalam elemen obrolan. Sedangkan `className` adalah nama kelas yang akan ditambahkan ke elemen `li`, yang menentukan apakah pesan tersebut merupakan pesan yang keluar (dari pengguna) atau masuk (dari chatbot).
- 2) Kode diatas membuat elemen HTML baru bertipe `li` (list item) yang akan digunakan untuk menampilkan pesan dalam daftar obrolan.
- 3) Menambahkan kelas CSS `chat` dan kelas tambahan yang diterima dari parameter `className` ke elemen `li` yang baru dibuat. Ini memungkinkan styling yang berbeda tergantung pada jenis pesan.
- 4) Menggunakan operator ternary untuk menentukan konten HTML yang akan ditampilkan di dalam `li`.
 - Jika `className` adalah `'outgoing'`, maka konten hanya berupa elemen paragraf (`<p>`) yang berisi pesan.
 - Jika tidak, (artinya berarti pesan tersebut adalah jenis masuk), maka konten akan terdiri dari sebuah ikon yang diwakili oleh `` dengan kelas `material-symbols-outlined` (yang biasanya menampilkan ikon dari pustaka Material Icons) dan diikuti dengan elemen paragraf (`<p>`) yang sama dengan pesan.
- 5) Mengatur properti `innerHTML` dari elemen `li` (`chatLi`) ke nilai `chatContent`, sehingga konten pesan ditambahkan ke dalam elemen.
- 6) Akhirnya, fungsi ini mengembalikan elemen `li` yang telah dibuat lengkap dengan kontennya.

```
function linkDaftar() {
  document.body.classList.remove('show-chatbot');
}
```

fungsi linkDaftar berfungsi untuk menyembunyikan atau menonaktifkan tampilan chatbot dengan menghapus kelas CSS yang mengontrol tampilan chatbot tersebut. Ini berguna untuk mengalihkan perhatian pengguna ke bagian lain dari halaman web atau untuk membersihkan tampilan setelah tindakan tertentu dilakukan.

```
const moveDown = chatBox.scrollTo(0, chatBox.scrollHeight);
```

Kode `const moveDown = chatBox.scrollTo(0, chatBox.scrollHeight);` adalah cara yang efisien untuk memastikan kotak obrolan selalu menampilkan pesan terbaru dengan menggulirnya secara otomatis ke bagian paling bawah setiap kali ada konten baru ditambahkan.

```
const handleChat = () => {
  userMessage = chatInput.value.trim();
  let keywords = ['visi', 'misi', 'pengertian', 'ciri', 'tujuan', 'daftar', 'regist', 'struktur', 'tugas'];
  let keywordOrgs = ['bea', 'bpa', 'akuntansi', 'asp', 'dkv', 'elektro', 'farmasi', 'kebidanan', 'komputer', 'mesin', 'perhotelan', 'informasi'];
  let isKeywordExist = false;
  let isUndefined = true;
  if(userMessage) return;
}
```

Fungsi handleChat dimulai dengan mengambil nilai dari input obrolan (`chatInput.value`), menghapus spasi di awal dan akhir (`trim()`) dan menyimpannya dalam variabel `userMessage`. Kemudian, mendefinisikan dua array: `keyword` yang berisi kata kunci umum seperti 'visi', 'misi', dll, dan `keywordOrgs` yang berisi daftar nama organisasi. `isKeywordExist` diatur ke `false`. Variabel ini digunakan untuk melacak apakah kata kunci yang relevan telah ditemukan dalam pesan pengguna. `isUndefined` diatur ke `true` dan digunakan untuk mengecek apakah jawaban dari organisasi yang spesifik tersedia atau tidak. Jika `userMessage` kosong, fungsi akan berhenti (`return`) tanpa melakukan apapun.

```
chatBox.appendChild(createChatLi(userMessage, "outgoing"));
moveDown;
```

Pesan pengguna ditambahkan ke kotak obrolan dengan menggunakan fungsi `createChatLi` untuk membuat elemen daftar (`li`) dengan kelas `outgoing`, menunjukkan bahwa pesan tersebut berasal dari pengguna. `moveDown` dipanggil untuk menggulir kotak obrolan ke bagian paling bawah, memastikan pesan terbaru selalu terlihat.

```

keywords.forEach((k)=> {
  if (userMessage.toLowerCase().includes(k)){
    keywordOrgs.forEach((org)=> {
      let index = keywordOrgs.indexOf(org)
      if (userMessage.toLowerCase().includes(org)) {
        if (answers[k][index] !== undefined) {
          setTimeout(()=> {
            chatBox.appendChild(createChatLi(answers[k][index], "incoming"));
            chatBox.scrollTo(0, chatBox.scrollHeight);
            isKeywordExist = true;
          }, 600)
        } else {
          isUndefined = false;
        }
      }
    })
  }
})

```

```

setTimeout(()=>{
  if(!isKeywordExist) {
    if(isUndefined) {
      chatBox.appendChild(createChatLi(answers[k], "incoming"));
      chatBox.scrollTo(0, chatBox.scrollHeight);
      isKeywordExist = true;
    } else {
      chatBox.appendChild(createChatLi("maaf ${k} untuk organisasi tersebut belum ada", "incoming"));
      chatBox.scrollTo(0, chatBox.scrollHeight);
      isKeywordExist = true;
    }
  }
}, 700)
}

```

Menggunakan array keywords. Jika userMessage mengandung salah satu kata kunci (diubah menjadi huruf kecil agar tidak case-sensitive), maka akan masuk ke loop berikutnya untuk mencari nama organisasi. Di dalam loop organisasi, kode memeriksa apakah userMessage juga mengandung nama organisasi dari array keywordOrgs. Jika kedua kata kunci (umum dan organisasi) ditemukan, kode mencoba mencari jawaban yang sesuai dalam objek answers menggunakan answers[k][index], dimana k adalah kata kunci umum, dan index adalah index dari organisasi yang ditemukan. Jika jawaban ditemukan (tidak undefined), jawaban tersebut ditambahkan ke kotak obrolan sebagai pesan "incoming" setelah penundaan. Jika jawaban tidak ditemukan (undefined), maka isUndefined diubah menjadi false. Setelah loop organisasi selesai, kode menunggu 700ms. Jika isKeywordExist masih false, yang berarti tidak ada jawaban spesifik yang ditemukan, kode memeriksa nilai dari isUndefined. Jika isUndefined bernilai true,

maka jawaban umum dari kata kunci umum (answers[k]) akan ditampilkan. Jika isUndefined bernilai false, maka pesan "maaf [kata kunci] untuk organisasi tersebut belum ada" akan ditampilkan.

```
setTimeout(() => {  
  if (!isKeywordExist) {  
    chatBox.appendChild(createChatLi(defaultAnswer, "incoming"));  
    chatBox.scrollTo(0, chatBox.scrollHeight);  
    isKeywordExist = false;  
  }  
}, 1000)
```

Setelah 1000ms (1 detik), jika isKeywordExist masih false, yang berarti tidak ada kata kunci yang cocok ditemukan dalam pesan pengguna, maka pesan defaultAnswer ditambahkan ke kotak obrolan sebagai pesan "incoming". isKeywordExist diatur kembali ke false.

```
chatBox.scrollTo(0, chatBox.scrollHeight);  
chatInput.value = '';
```

Kotak obrolan digulir ke bagian paling bawah lagi untuk memastikan pesan terbaru terlihat. Nilai dari chatInput diatur menjadi kosong ("") untuk menghapus teks yang telah dimasukkan oleh pengguna.

```
senderChat.addEventListener('click', handleChat);
```

senderChat.addEventListener('click', handleChat); adalah penghubung penting yang mengaktifkan seluruh alur kerja obrolan. Tanpa baris kode ini, mengklik tombol "Kirim" tidak akan melakukan apa pun. Fungsi handleChat adalah otak dari aplikasi obrolan, yang memproses pesan, mencari jawaban, dan memperbarui tampilan obrolan.

b. Login

- LoginController.php

```
class LoginController extends Controller
{
    /* ...
    use AuthenticatesUsers;
    /* ...
    0 references
    protected $redirectTo = '/home';
    /* ...
    0 references | 0 overrides
    public function __construct()
    {
        $this->middleware(middleware: 'guest')->except(methods: 'logout');
        $this->middleware(middleware: 'auth')->only(methods: 'logout');
    }
}
```

Kelas bernama LoginController. extends Controller berarti kelas ini memfungsikan dari kelas Controller yang merupakan kelas dasar untuk semua controller. use AuthenticatesUsers; menggunakan trait bernama AuthenticatesUsers. protected \$redirectTo = '/home'; berarti setelah login berhasil, pengguna akan diarahkan ke route /home. \$this->middleware('guest')->except('logout'); menerapkan middleware guest ke semua method di controller ini kecuali method logout. Middleware guest akan mengarahkan pengguna yang sudah login kembali ke halaman utama jika mereka mencoba mengakses halaman login atau register. Intinya, hanya pengguna yang belum login yang bisa mengakses halaman login dan register. \$this->middleware('auth')->only('logout'); menerapkan middleware auth hanya ke method logout. Middleware auth akan memastikan bahwa hanya pengguna yang sudah login yang bisa mengakses method logout. Ini mencegah pengguna yang belum login untuk mencoba logout.

- Login.blade.php

```

<div class="d-flex justify-content-center align-items-center" style="height: 500px;">
  <div class="w-50 center border rounded px-3 py-3 mx-auto">
    <form method="POST" action="{{ route(name: 'login') }}">
      @csrf
      <div style="display: flex; justify-content: center; align-items: center;">
        
      </div>
      <div class="mb-3">
        <h1 class="text-center">Login</h1>
        <label for="email" class="form-label">Email</label>
        <input id="email" type="email" class="form-control @error('email') is-invalid @enderror"
          name="email" value="{{ old(key: 'email') }}" autocomplete="email" autofocus>
        @error('email')
          <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
          </span>
        @enderror
      </div>
      <div class="mb-3">
        <label for="password" class="form-label">Password</label>
        <input id="password" type="password" class="form-control @error('password') is-invalid @enderror"
          name="password" autocomplete="current-password">
        @error('password')
          <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
          </span>
        @enderror
      </div>
      <div class="mb-3 d-grid">
        <button name="submit" type="submit" class="btn btn-primary">Login</button>
      </div>
      <p>Belum Punya Akun? <span><a href="/register">Buat Akun</a></span></p>
      <p>Kamu Admin? jika iya masuk sini min hih <span><a href="/admin/login">Login Admin</a></span></p>
      <a href="/" class="btn btn-secondary">Kembali</a>
    </form>
  </div>
</div>

```

Kode diatas adalah formulir login HTML yang dibuat dengan menggunakan Bootstrap dan Blade (template engine Laravel). Formulir ini memiliki input untuk email dan password, validasi error, tombol submit, dan tautan ke halaman registrasi, login admin, dan halaman utama. Tata letaknya responsif karena menggunakan kelas-kelas Bootstrap untuk flexbox dan grid. Kode ini juga aman karena menggunakan token CSRF untuk mencegah serangan.

c. Register

- RegisterController.php

```
class RegisterController extends Controller
{
    /* ...

    use RegistersUsers;

    /** ...
    0 references
    protected $redirectTo = '/home';

    /** ...
    0 references | 0 overrides
    public function __construct()
    {
        $this->middleware(middleware: 'guest');
    }

    /**
     * Get a validator for an incoming registration request.
     *
     * @param array $data
     * @return \Illuminate\Contracts\Validation\Validator
     */
    protected function validator(array $data): Validator
    {
        return Validator::make(data: $data, rules: [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
            'password' => ['required', 'string', 'min:8', 'confirmed'],
            'nim' => ['required', 'string', 'max:20', 'unique:users,nim'], // NIM wajib unik
            'prodi' => ['required', 'string', 'max:100'], // Program studi
            'semester' => ['required', 'integer', 'min:1', 'max:14'], // Semester minimal 1 dan maksimal 14
            'nomor' => ['required', 'string', 'regex:/^(\\+62|0)[0-9]{9,13}$/'], // Validasi nomor telepon
        ], messages: [
            // Custom error messages
            'nim.required' => 'Nim harus di isi.',
            'nim.unique' => 'Nim sudah digunakan.',
            'prodi.required' => 'Program studi harus di isi.',
            'semester.required' => 'Semester harus di isi.',
            'semester.integer' => 'Semester harus berupa angka.',
            'semester.min' => 'Semester tidak boleh kurang dari 1.',
            'semester.max' => 'Semester tidak boleh lebih dari 14.',
            'nomor.required' => 'Nomor telepon harus di isi.',
            'nomor.regex' => 'Format nomor telepon tidak valid.',
            'password.required' => 'Password harus di isi.'
        ]
    }
}
```

```

        'nomor.regex' => 'Format nomor telepon tidak valid.',
        'name.required' => 'Nama harus di isi',
        'email.required' => 'Email harus di isi',
        'password.required' => 'Password wajib di isi',
        'password-confirm.required' => 'Konfirmasi password wajib di isi',
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\Models\User
 */
0 references | 0 overrides
protected function create(array $data): Model\User
{
    $userData = [
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make(value: $data['password'])
    ];

    // Cek dan masukkan data yang ada (jika tidak null)
    if (isset($data['nim'])) {
        $userData['nim'] = $data['nim'];
    }
    if (isset($data['prodi'])) {
        $userData['prodi'] = $data['prodi'];
    }
    if (isset($data['semester'])) {
        $userData['semester'] = $data['semester'];
    }
    if (isset($data['nomor'])) {
        $userData['nomor'] = $data['nomor'];
    }

    return User::create(attributes: $userData);
}

```

Kode diatas adalah RegisterController adalah controller Laravel yang menangani proses pendaftaran pengguna baru. Ia menggunakan trait RegistersUsers untuk

menyediakan fungsionalitas pendaftaran dasar. Method validator() mendefinisikan aturan validasi untuk data pendaftaran, termasuk validasi untuk NIM, program studi, semester, dan nomor telepon dengan format yang sesuai. Method create() membuat akun pengguna baru di database setelah data berhasil divalidasi, dan mengenkripsi password sebelum menyimpannya. Dengan ini, controller memastikan bahwa hanya data yang valid yang disimpan ke database, dan password pengguna disimpan dengan aman.

- Register.blade.php

```

<div class="container py-5">
  <div class="d-flex justify-content-center align-items-center" style="height: 50%;">
    <div class="w-50 center border rounded px-3 py-3 mx-auto">
      <form method="POST" action="{{ route(name: 'register') }}">
        @csrf
        <div style="display: flex; justify-content: center; align-items: center;">
          
        </div>
        <div class="mb-3">
          <h1 class="text-center">Daftar</h1>
          <label for="name" class="form-label">Nama Lengkap <span style="color: red;">*</span>
          <input type="text" value="{{ old(key: 'name') }}" name="name" class="form-control">
          @if ($errors->has('name'))
            <div class="text-danger">
              {{ $errors->first('name') }}
            </div>
          @endif
        </div>
        @if ($errors->has('nim'))
          <div class="text-danger">
            {{ $errors->first('nim') }}
          </div>
        @endif
      </div>
      <div class="mb-3">
        <label for="email" class="form-label">Email<span style="color: red;">*</span>
        <input type="email" value="{{ old(key: 'email') }}" name="email" class="form-control">
        @if ($errors->has('email'))
          <div class="text-danger">
            {{ $errors->first('email') }}
          </div>
        @endif
      </div>
      <div class="mb-3">
        <label for="nomor" class="form-label">WhatsApp<span style="color: red;">*</span>
      </div>
    </div>
  </div>
</div>

```

```

<input type="text" maxlength="13" value="{{ old(key: 'nomor') }}" name="nomor"
  class="form-control"/>
@endif ($errors->has('nomor'))
<div class="text-danger">
  {{ $errors->first('nomor') }}
</div>
@endif
</div>

<div class="mb-3">
<label class="form-label" for="prodi">Program Studi: <span style="color: red;">
<select name="prodi" id="prodi" class="form-control">
  <option value="DIV Akuntansi Sektor Publik">DIV Akuntansi Sektor Publik</option>
  <option value="DIV Teknik Informatika">DIV Teknik Informatika</option>
  <option value="DIII Akuntansi">DIII Akuntansi</option>
  <option value="DIII Desain Komunikasi Visual">DIII Desain Komunikasi Visu
  <option value="DIII Desain Komunikasi Visual">DIII Desain Komunikasi Visu
  <option value="DIII Farmasi">DIII Farmasi</option>
  <option value="DIII Kebidanan">DIII Kebidanan</option>
  <option value="DIII Keperawatan">DIII Keperawatan</option>
  <option value="DIII Perhotelan">DIII Perhotelan</option>
  <option value="DIII Teknik Elektronika">DIII Teknik Elektronika</option>
  <option value="DIII Teknik Komputer">DIII Teknik Komputer</option>
  <option value="DIII Teknik Mesin">DIII Teknik Mesin</option>
</select>
@endif ($errors->has('prodi'))
<div class="text-danger">
  {{ $errors->first('prodi') }}
</div>
@endif
</div>

<div class="mb-3">
<label for="semester" class="form-label">Semester<span style="color: red;">*
<select name="semester" id="semester" class="form-control">
  <!-- Semester options will be dynamically populated -->
</select>
@endif ($errors->has('semester'))
<div class="text-danger">
  {{ $errors->first('semester') }}
</div>
@endif
</div>

<script>
document.addEventListener('DOMContentLoaded', function() {
  const prodiSelect = document.getElementById('prodi');
  const semesterSelect = document.getElementById('semester');

  function updateSemesterOptions() {

```

```

function updateSemesterOptions() {
  const selectedProdi = prodiSelect.value;
  const maxSemester = selectedProdi.startsWith('DIV') ? 6 : 4;

  // Clear current options
  semesterSelect.innerHTML = '';

  // Populate new options
  for (let i = 1; i <= maxSemester; i++) {
    const option = document.createElement('option');
    option.value = i;
    option.textContent = i;
    semesterSelect.appendChild(option);
  }
}

// Initialize semester options on page load
updateSemesterOptions();

// Update semester options when prodi changes
prodiSelect.addEventListener('change', updateSemesterOptions);
});
</script>

<div class="mb-3">
  <label for="password">password:<span style="color: red;">*</span></label>
  <input type="password" id="password" name="password" class="form-control">
  <p1 style="margin-top:4px" class="ket">kekuatan password: <span id="power-poi
    karakter</span></p1>
  @if ($errors->has('password'))
  <div class="text-danger">
    {{ $errors->first('password') }}
  </div>
  @endif
  </div>
  <div class="mb-3">
  <label for="password-confirm">Konfirmasi Password<span style="color: red;">*</span>
  <input id="password-confirm" type="password" class="form-control" name="passw
    autocomplete="new-password">
  </div>
  <div class="mb-3 d-grid">
  <button name="submit" type="submit" class="btn btn-primary">Daftar</button>
  </div>
  <p><span>Kembali ke <span> <a href="/login">Halaman Login</a></span></p>
</form>
</div>
</div>

```

```

<script>
  // cek kekuatan password
  let password = document.getElementById("password");
  let power = document.getElementById("power-point");
  password.oninput = function() {
    let point = 0;
    let value = password.value;
    let widthPower = ["Sangat Lemah", "Lemah", "Cukup", "Kuat", "Sangat Kuat"];
    let colorPower = ["#D73F40", "#DC6551", "#F2B84F", "#BDE952", "#3ba62f"];

    if (value.length >= 6) {
      let arrayTest = [/[0-9]/, /[a-z]/, /[A-Z]/, /^[^0-9a-zA-Z]/];
      arrayTest.forEach((item) => {
        if (item.test(value)) {
          point += 1;
        }
      });
    }
    power.textContent = widthPower[point];
    power.style.color = colorPower[point];
  };
};

```

Kode diatas adalah form pendaftaran yang lengkap dengan validasi. Validasi tersebut mencakup pada pengisian setiap form dan pengecekan kekuatan password serta memberi feedback visual.

d. Admin

- Admin.controller.php

```

function adminView(): Factory\View
{
  if (Auth::user()->role == 'admin') {
    $org = Auth::user()->nama_organisasi;
    $kegiatan = Agenda::where('nama_organisasi', operator: $org)->get();
    $sanggota = User::where('nama_organisasi', operator: $org)->get();
    $rutin = Rutin::all();
    // $sanggota = User::orderBy('name')->get();
    $user = Admin::find(id: Auth::user()->id);
    return view('admin.dashboard.index', data: ['user' => $user, 'rutin' => $rutin, 'anggota' => $sanggota, 'kegiatan' => $keg]);
  }
}

```

Kode diatas menjelaskan apakah peran pengguna yang sedang login adalah 'admin'. `$org = Auth::user()->nama_organisasi`; kode tersebut mengambil nama organisasi dari data pengguna yang sedang login. `$kegiatan = Agenda::where('nama_organisasi', $org)->get()`; kode tersebut mengambil data kegiatan atau agenda yang memiliki nama organisasi yang sama dengan organisasi admin tersebut. `$sanggota = User::where('nama_organisasi', $org)->get()`; kode tersebut mengambil semua data user yang terdaftar dalam organisasi yang sama dengan admin. `$rutin = Rutin::all()`; kode tersebut mengambil semua data rutinan dari beberapa kegiatan unit kegiatan mahasiswa (UKM). `$user = Admin::find(Auth::user()->id)`; kode tersebut Mengambil data admin (Admin) berdasarkan ID pengguna yang sedang login. `return view('admin.dashboard.index', ['user' => $user, 'rutin' => $rutin, 'anggota' => $sanggota, 'kegiatan' => $kegiatan]);` Menampilkan tampilan (view) `admin.dashboard.index` dan mengirimkan data

\$user, \$rutin, \$sanggota, dan \$kegiatan ke tampilan tersebut. Data ini akan digunakan untuk ditampilkan di dashboard admin.

```
public function news(): Factory\View
{
    if (Auth::user()->role == 'admin') {
        $org = Auth::user()->nama_organisasi;
        $kegiatan = Agenda::where(column: 'nama_organisasi', operator: $org)->get();
        $sanggota = User::where(column: 'nama_organisasi', operator: $org)->get();
        $rutin = Rutin::all();
        // $sanggota = User::orderBy('name')->get();
        $user = Admin::find(id: Auth::user()->id);
        return view(view: 'admin.news.index', data: ['user' => $user, 'rutin' => $rutin, 'anggota' => $sanggota, 'kegiatan' => $kegiatan]);
    }
}
```

Fungsi news() menampilkan halaman berita untuk admin. Halaman berita ini menampilkan data kegiatan, anggota, dan informasi setiap unit yang berada di organisasi mahasiswa Politeknik Harapan Bersama Tegal.

```
public function arsip(): Factory\View
{
    if (Auth::user()->role == 'admin') {
        $org = Auth::user()->nama_organisasi;
        $kegiatan = Agenda::where(column: 'nama_organisasi', operator: $org)->get();
        $sanggota = User::where(column: 'nama_organisasi', operator: $org)->get();
        $rutin = Rutin::all();
        // $sanggota = User::orderBy('name')->get();
        $user = Admin::find(id: Auth::user()->id);
        return view(view: 'admin.arsip.index', data: ['user' => $user, 'rutin' => $rutin, 'anggota' => $sanggota, 'kegiatan' => $kegiatan]);
    }
}
```

Fungsi arsip() menampilkan halaman arsip untuk unit, yang menampilkan data kegiatan, anggota, dan informasi unit yang terkait dengan unit tersebut. Data kegiatan kemungkinan adalah data yang telah diarsipkan (misalnya, kegiatan yang sudah selesai).

```
lic function absensi(): Factory\View
{
    if (Auth::user()->role == 'admin') {
        $org = Auth::user()->nama_organisasi;
        $kegiatan = Agenda::where(column: 'nama_organisasi', operator: $org)->get();
        $sanggota = User::where(column: 'nama_organisasi', operator: $org)->get();
        $rutin = Rutin::all();
        // $sanggota = User::orderBy('name')->get();
        $user = Admin::find(id: Auth::user()->id);
        return view(view: 'admin.pengurus.index', data: ['user' => $user, 'rutin' => $rutin, 'anggota' => $sanggota, 'kegiatan' => $kegiatan]);
    }
}
```

Fungsi absensi() menampilkan halaman yang berhubungan dengan absensi anggota pada setiap unit. Fungsi ini mengambil data kegiatan, anggota, dan informasi unit.

```
lic function calon(): Factory\View
{
    if (Auth::user()->role == 'admin') {
        $org = Auth::user()->nama_organisasi;
        $kegiatan = Agenda::where(column: 'nama_organisasi', operator: $org)->get();
        $sanggota = User::where(column: 'nama_organisasi', operator: $org)->get();
        $rutin = Rutin::all();
        // $sanggota = User::orderBy('name')->get();
        $user = Admin::find(id: Auth::user()->id);
        return view(view: 'admin.calon.index', data: ['user' => $user, 'rutin' => $rutin, 'anggota' => $sanggota, 'kegiatan' => $kegiatan]);
    }
}
```

Fungsi calon () menampilkan halaman untuk mengelola data calon anggota yang akan di seleksi bagi setiap unit.

```

ic function wawancara(): Factory\View
{
    $org = Auth::user()->nama_organisasi;
    $kegiatan = Agenda::where(column: 'nama_organisasi', operator: $org)->get();
    $anggota = User::where(column: 'nama_organisasi', operator: $org)->get();
    $rutin = Rutin::all();
    // $anggota = User::orderBy('nama')->get();
    $user = Admin::find(id: Auth::user()->id);
    return view('admin.wawancara.index', data: ['user' => $user, 'rutin' => $rutin, 'anggota' => $anggota, 'kegiatan' => $kegiatan]);
}

```

Fungsi wawancara() menampilkan halaman untuk mengelola data wawancara calon anggota yang akan di seleksi bagi setiap unit .

```

ic function profile(): Factory\View
{
    $org = Auth::user()->nama_organisasi;
    $kegiatan = Agenda::where(column: 'nama_organisasi', operator: $org)->get();
    $anggota = User::where(column: 'nama_organisasi', operator: $org)->get();
    $rutin = Rutin::all();
    // $anggota = User::orderBy('nama')->get();
    $user = Admin::find(id: Auth::user()->id);
    return view('admin.profile.index', data: ['user' => $user, 'rutin' => $rutin, 'anggota' => $anggota, 'kegiatan' => $kegiatan]);
}

```

Fungsi profile() menampilkan halaman profil untuk admin.

- Sidebar.blade.php

```

<div class="link-bar d-flex">
    <div class="link-bar d-flex">
        <li class="{{ Route::is(patterns: 'admin.dashboard') ? 'active' : '' }}">
            <a href="{{ route('admin.dashboard') }}">
                <span class="material-symbols-outlined">home</span>Dashboard
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.news') ? 'active' : '' }}">
            <a href="{{ route('admin.news') }}">
                <span class="material-symbols-outlined">campaign</span>NEWS
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.arsip') ? 'active' : '' }}">
            <a href="{{ route('admin.arsip') }}">
                <span class="material-symbols-outlined">bookmark</span>Arsip
            </a>
        </li>
        @if ($user->nama_organisasi == 'bea')
            <li class="{{ Route::is(patterns: 'rutin.index') ? 'active' : '' }}">
                <a href="{{ route('rutin.index') }}">
                    <span class="material-symbols-outlined">routine</span>Rutinitas
                </a>
            </li>
        @endif
        <li class="{{ Route::is(patterns: 'admin.abseksi') ? 'active' : '' }}">
            <a href="{{ route('admin.abseksi') }}">
                <span class="material-symbols-outlined">diversity_3</span>Pengurus
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.calon') ? 'active' : '' }}">
            <a href="{{ route('admin.calon') }}">
                <span class="material-symbols-outlined">inventory</span>Seleksi Administrasi
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.wawancara') ? 'active' : '' }}">
            <a href="{{ route('admin.wawancara') }}">
                <span class="material-symbols-outlined">inventory</span>Seleksi Nawancara
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.profile') ? 'active' : '' }}">
            <a href="{{ route('admin.profile') }}">
                <span class="material-symbols-outlined">person</span>Data Profile
            </a>
        </li>
    </div>
    <form action="{{ route('admin.logout') }}" method="POST">
        @csrf
        <button type="submit" style="display: flex; align-items: center; justify-content: center; color: black; border: none; background: none;">
            <span class="material-symbols-outlined">close</span>Keluar
        </button>
    </form>
</div>
</div>

```

Kode diatas adalah sebuah side bar (bilah sisi) untuk halaman admin. Sidebar ini berisi tautan (link) navigasi ke berbagai bagian panel admin, seperti dashboard, news, arsip, rutinitas, pengurus, seleksi administrasi, seleksi wawancara, data profile. Ada juga pengecekan peran (\$user->nama_organisasi == 'bem') untuk menampilkan tautan "Rutinitas" hanya jika admin berasal dari unit "bem".

e. Super Admin

- Superadmin.controller

```

} elseif (Auth::user()->role == 'super_admin') {
    $kegiatan = Agenda::all();
    $anggota = User::all();
    $admin = Admin::all();
    // $anggota = User::orderBy('name')->get();
    $user = User::find(id: Auth::user()->id);
    return view('superadmin.dashboard.index', data: ['user' => $user, 'anggota' => $anggota, 'kegiatan' => $kegiatan, 'admin' => $admin]);
}

```

Kode diatas akan menjelaskan jika seorang setelah login sebagai super admin maka akan diarahkan ke dashboard superadmin. Superadmin memiliki akses ke semua data kegiatan, data anggota pada setiap unit, dan admin dari seluruh unit.

```

} elseif (Auth::user()->role == 'super_admin') {
    $kegiatan = Agenda::all();
    $anggota = User::all();
    $admin = Admin::all();
    // $anggota = User::orderBy('name')->get();
    $user = User::find(id: Auth::user()->id);
    return view('superadmin.createuser.index', data: ['user' => $user, 'anggota' => $anggota, 'kegiatan' => $kegiatan, 'admin' => $admin]);
}

```

Kode diatas menjelaskan halaman untuk membuat pengguna baru sebagai admin (create user) khusus untuk super admin.

```

} elseif (Auth::user()->role == 'super_admin') {
    $kegiatan = Agenda::all();
    $anggota = User::all();
    $admin = Admin::all();
    // $anggota = User::orderBy('name')->get();
    $user = User::find(id: Auth::user()->id);
    return view('superadmin.news.index', data: ['user' => $user, 'anggota' => $anggota, 'kegiatan' => $kegiatan, 'admin' => $admin]);
}

```

Kode diatas menjelaskan halaman sebuah news. Superadmin hanya bisa melihat, tanpa adanya menghapus atau mengedit.

```

} elseif (Auth::user()->role == 'super_admin') {
    $kegiatan = Agenda::all();
    $anggota = User::all();
    $admin = Admin::all();
    // $anggota = User::orderBy('name')->get();
    $user = User::find(id: Auth::user()->id);
    return view('superadmin.arsip.index', data: ['user' => $user, 'anggota' => $anggota, 'kegiatan' => $kegiatan, 'admin' => $admin]);
}

```

Kode diatas menjelaskan halaman untuk mengelola arsip khusus untuk superadmin. Super admin pada halaman arsip hanya dapat menerima informasi dan mendownload untuk kebutuhan dokumen.

```

) elseif (Auth::user()->role == 'super_admin') {
    $kegiatan = Agenda::all();
    $anggota = User::all();
    $admin = Admin::all();
    // $anggota = User::orderBy('name')->get();
    $user = User::find(id: Auth::user()->id);
    return view('superadmin.dataormawa.index', data: ['user' => $user, 'anggota' => $anggota, 'kegiatan' => $kegiatan, 'admin' => $admin]);
}

```

Kode diatas menjelaskan halaman data ormawa untuk super admin. Data ormawa berisi semua anggota dari berbagai unit organisasi, super admin hanya dapat melihat nama, jabatan, unit organisasi apa.

- Sidebar.superadmin

```

<div class="sidebar">
    <div class="link-bar d-flex">
        <li class="{{ Route::is(patterns: 'admin.dashboard') ? 'active' : '' }}">
            <a href="{{ route('admin.dashboard') }}">
                <span class="material-symbols-outlined">home</span> Dashboard
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.news') ? 'active' : '' }}">
            <a href="{{ route('admin.news') }}">
                <span class="material-symbols-outlined">campaign</span> NEWS
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.arsip') ? 'active' : '' }}">
            <a href="{{ route('admin.arsip') }}">
                <span class="material-symbols-outlined">bookmark</span> Arsip
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.absensi') ? 'active' : '' }}">
            <a href="{{ route('admin.absensi') }}">
                <span class="material-symbols-outlined">inventory</span> Data Ormawa
            </a>
        </li>
        <li class="{{ Route::is(patterns: 'admin.tambahAdminView') ? 'active' : '' }}">
            <a href="{{ route('admin.tambahAdminView') }}">
                <span class="material-symbols-outlined">person</span> Create User
            </a>
        </li>
    </div>
    <div class="form actions"{{ route('admin.logout') }}" method="POST">
        <button type="submit">
            <span class="material-symbols-outlined">close</span> Keluar
        </button>
    </div>
</div>
</div>

```

Kode ini adalah struktur HTML untuk sebuah sidebar pada Super admin. Sidebar ini berisi tautan (link) navigasi ke berbagai bagian panel Super admin, seperti dashboard, news, arsip, data ormawa, create user.



IMPLEMENTASI CHATBOT PADA SISTEM
INFORMASI ORGANISASI MAHASISWA
BERBASIS WEBSITE
MANUAL BOOK

Muhammad Haidar Rasyiq, Dyah Apriliani., M.Kom., Arif Hidayah, M.Tr.Kom.
Sarjana Terapan Teknik Informatika

A. Pendahuluan

1. Tujuan dan Manfaat

1.1 Tujuan

Tujuan dibuatnya website ini adalah membantu dalam menyimpan dan mengelola data anggota, kegiatan, dan administrasi organisasi mahasiswa secara terstruktur dan terorganisir.

1.2 Manfaat

1. Manfaat Bagi Mahasiswa

- 1) Mahasiswa dapat mengetahui cara mendaftar sebagai calon anggota organisasi mahasiswa.
- 2) Mahasiswa dapat mengenal pada setiap unit yang tergabung dalam organisasi mahasiswa.
- 3) Mahasiswa dapat mengetahui kegiatan pada setiap unit organisasi mahasiswa.

2. Manfaat Bagi Organisasi Mahasiswa

- 1) Dapat membantu setiap unit agar mengetahui keaktifan anggota/pengurus pada setiap program kerja.
- 2) Dapat mengelola administrasi penerimaan anggota baru pada setiap unit.
- 3) Dapat mengetahui keaktifan setiap unit pada program kerja yang terencana.
- 4) Dapat membantu dalam penyimpanan, pencarian dokumen proposal kegiatan dan laporan pertanggung jawaban.

2. Deskripsi Umum Sistem

2.1. Deskripsi Umum Aplikasi

Deskripsi umum Implementasi Chatbot pada Sistem Informasi Organisasi Mahasiswa Berbasis Website meliputi deskripsi, fungsi utama website ini adalah untuk me-manage sebuah organisasi mahasiswa yang berada di Politeknik Harapan Bersama Tegal.

2.2. Deskripsi Umum Kebutuhan Aplikasi

Deskripsi umum kebutuhan aplikasi yang akan di implementasikan meliputi semua informasi yang berada di unit organisasi mahasiswa Politeknik Harapan Bersama Tegal yang menjadi acuan dalam pengembangan aplikasi.

B. Menu dan Cara Penggunaan

1. Tampilan Landing Page

Pada Halaman awal website terdapat header yang terdapat beberapa navigasi menu yaitu 'Beranda', 'Jadwal Kegiatan', 'Organisasi', 'Masuk' dan pada pojok kanan bawah terdapat 'Chatbot'.

1.1. Beranda

Pada Beranda terdapat ucapan selamat datang pada website dan logo organisasi mahasiswa. Tampilan sebagai berikut.



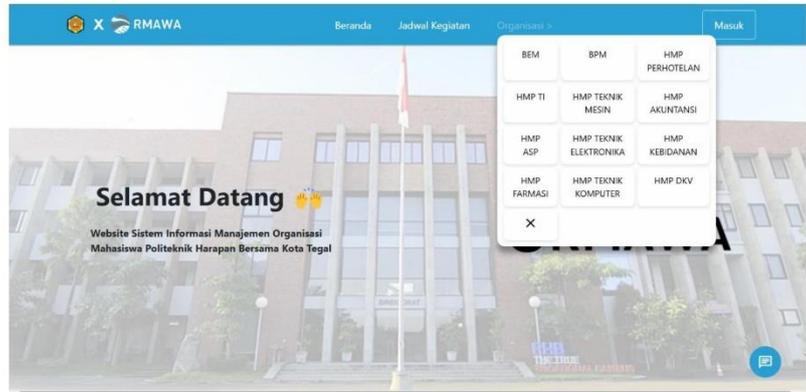
1.2. Jadwal Kegiatan

Pada Jadwal kegiatan terdapat tabel jadwal kegiatan unit kegiatan mahasiswa (UKM) agar mahasiswa mengetahui jadwal pada setiap unit kegiatan mahasiswa (UKM). Tampilan seperti berikut.

No	Hari	UKM	Lokasi
3	Sabtu	UKM Musik	Sekretariat Ormawa UKM Musik
4	Sabtu	UKM Karate	Depan Asrama dan SC
5	Rabu	UKM Pencak Silat	Depan Asrama dan SC
6	Sabtu	UKM Robotika	Gedung B
7	Sabtu	UKM Tari	Sport Center
8	Sabtu	UKM Formasi	Majid
9	Sabtu	UKM Ransol	Sport Center
10	Selasa & Jumat	UKM Basket	Sport Center
11	Rabu	UKM Futsal	Sport Center
12	Sabtu	UKM Popala	Sekretariat Ormawa UKM Popala
13	Sabtu	UKM Teater	Sport Center

1.3. Organisasi

Pada Organisasi terdapat beberapa navigasi setiap unit pada organisasi mahasiswa. Tampilan seperti berikut.

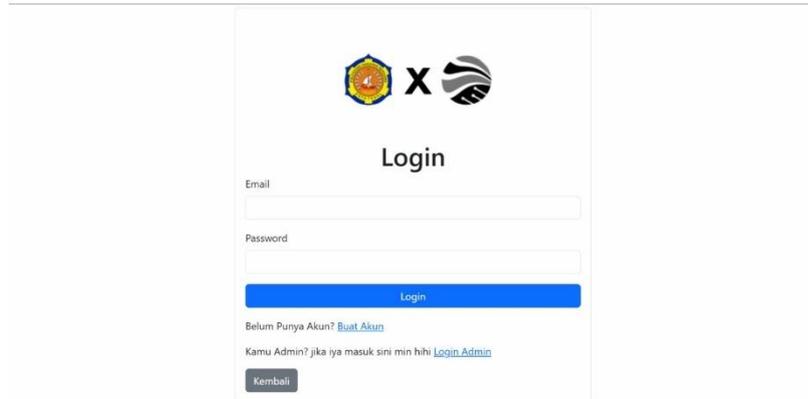


1.3.1 Jika sudah mengklik tersebut silahkan pilih salah satu, misal milih 'HMP TI' nanti akan diarahkan kepada halaman yang berisi 'Logo', 'Visi', 'Misi'. Tampilan seperti berikut.



1.4. Masuk

Pada Navigasi ini nanti akan diarahkan ke halaman Login. Tampilan seperti berikut.

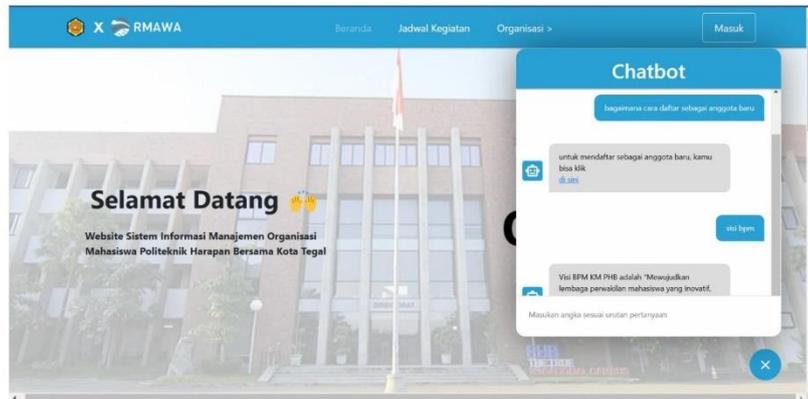


The screenshot shows a login interface with the following elements:

- Logo of RMAWA (Rahmatul Mahasiswa) at the top center.
- The word "Login" in a large, bold font.
- An "Email" input field.
- A "Password" input field.
- A blue "Login" button.
- Text: "Belum Punya Akun? [Buat Akun](#)"
- Text: "Kamu Admin? jika iya masuk sini min hihi [Login Admin](#)"
- A "Kembali" button at the bottom left.

1.5. Chatbot

Pada chatbot ini mahasiswa bisa menanyakan seputar informasi unit organisasi mahasiswa dan cara mendaftar sebagai anggota baru. Tampilan sebagai berikut.



2. User

2.1. Halaman Awal dan Login

Yang pertama user akan diarahkan ke halaman dashboard aplikasi dengan mengakses domain website ini dengan cara sebagai berikut.

- a) Buka browser (Chrome S Mozilla Firefox)
- b) Masukan url : <https://www.ormawapoltekharber.my.id/> yang merupakan alamat website, seperti berikut :



- c) Kemudian user bisa melakukan login dengan mengklik tombol "Masuk" dan nanti akan diarahkan ke halaman Login, jika user sudah mempunyai akun maka masukan 'Email' dan 'Password', akan tampil seperti berikut.

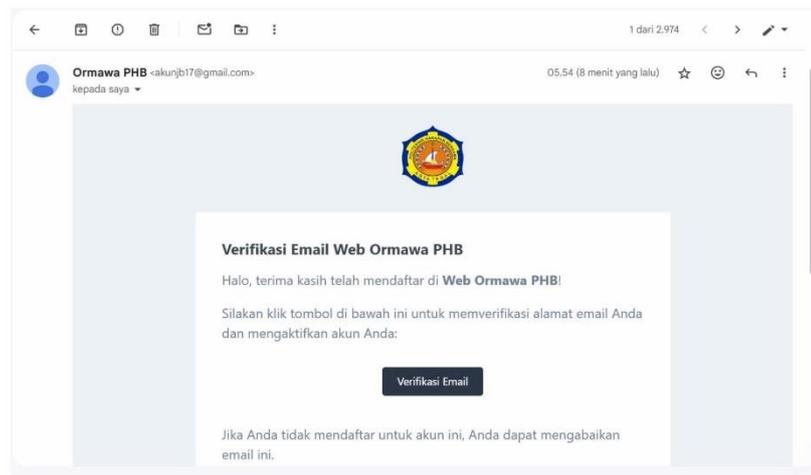
The image shows the login page of the ORMAWA website. At the top, there are three logos: a yellow and red emblem, a black 'X', and the ORMAWA logo. Below the logos is the word 'Login' in a bold, black font. There are two input fields: 'Email' and 'Password'. Below the 'Password' field is a blue 'Login' button. Underneath the button, there are two lines of text: 'Belum Punya Akun? [Buat Akun](#)' and 'Kamu Admin? jika iya masuk sini min hihi [Login Admin](#)'. At the bottom left, there is a small grey button labeled 'Kembali'.

- d) Jika user belum memiliki akun maka klik tombol “Buat Akun” maka akan di arahkan ke halaman Daftar dan selanjutnya mengisi form ‘Nama Lengkap’, ‘Nim’, ‘Email’, ‘Whatsapp’, ‘Program Studi’, ‘Semester’, ‘Password’, ‘Konfirmasi Password’. Tampilan seperti berikut.



The screenshot shows a registration form titled "Daftar" with the following fields: Nama Lengkap, NIM, Email, WhatsApp, Program Studi (with a dropdown menu showing "DIY Akuntansi Sektor Publik"), Semester (with a dropdown menu showing "1"), password, and Konfirmasi Password. A blue "Daftar" button is located at the bottom of the form.

- e) Jika user sudah mendaftarkan akun nanti di arahkan pada email untuk verifikasi email pada akun tersebut, lalu klik tombol ‘Verifikasi Email’. Tampilan seperti berikut.



2.2. Halaman Dashboard User

- a) Setelah user melakukan verifikasi email, user akan diarahkan ke dashboard user yang berisikan menu yaitu 'Form', 'Tahap Pendaftaran', 'Riwayat'. Tampilan seperti Berikut.

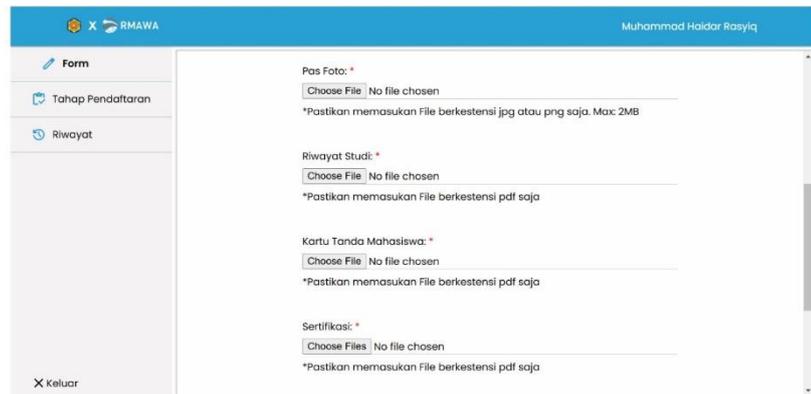


The screenshot shows a web application interface for 'RMAWA'. The user is logged in as 'Muhammad Haidar Rasyiq'. The left sidebar contains three menu items: 'Form', 'Tahap Pendaftaran', and 'Riwayat'. The main content area is titled 'FORM PENDAFTARAN' and contains the following fields:

- Nama Lengkap: Muhammad Haidar Rasyiq
- NIM: 20090087
- Whatsapp: 089872136764
- Semester: 1

At the bottom left of the sidebar, there is a 'Keluar' button.

- b) Kemudian user melengkapi dokumen pada "Form Pendaftaran", dokumen yang harus dilengkapi yaitu 'Pas Foto', 'Riwayat Studi', 'Kartu Tanda Mahasiswa', 'Sertifikat', lalu memilih organisasi yang di tuju, jika sudah dilengkapi semuanya user klik tombol "Daftar". Tampilan seperti berikut.



The screenshot shows the same 'FORM PENDAFTARAN' page, but with file upload fields for documents. The fields are:

- Pas Foto: *
Choose File | No file chosen
*Pastikan memasukan File berkeestensi jpg atau png saja. Max: 2MB
- Riwayat Studi: *
Choose File | No file chosen
*Pastikan memasukan File berkeestensi pdf saja
- Kartu Tanda Mahasiswa: *
Choose File | No file chosen
*Pastikan memasukan File berkeestensi pdf saja
- Sertifikasi: *
Choose Files | No file chosen
*Pastikan memasukan File berkeestensi pdf saja

The 'Keluar' button is still visible at the bottom left of the sidebar.

- c) Setelah user mendaftar akan diarahkan pada menu “Tahap Pendaftaran”, di menu tersebut user dapat melihat informasi proses tahapan pendaftaran pada organisasi mahasiswa. Tampilan seperti berikut.

- d) Selain itu juga user akan mendapatkan notifikasi email terkait proses tahapan pendaftaran. Tampilan seperti berikut.

- e) Kemudian pada menu “Riwayat” user bisa melihat kekurangan jika tidak lolos pada tahapan pendaftaran, dan bisa melihat berapa kali mendaftar pada unit ormawa. Tampilan seperti berikut.



No	Nama	Organisasi Tujuan	Status	Keterangan	Tanggal
1	Muhammad Haidar Rasyiq	himp si	Lolos ke Wawancara	Berkas lengkap	01 Jan 2025
2	Muhammad Haidar Rasyiq	himp si	gagal tahap wawancara	Tidak lolos	01 Jan 2025
3	Muhammad Haidar Rasyiq	himp si	Lolos ke Wawancara	Berkas lengkap	09 Jan 2025
4	Muhammad Haidar Rasyiq	himp si	aktif	100 Work	09 Jan 2025

3. Admin

Pada website Admin untuk tampilan awal dashboard dan login sama dengan user hanya saja untuk akun admin dibuatkan oleh Super Admin sehingga admin bisa langsung Login dengan akun yang telah dibuatkan oleh Super Admin sebelumnya. Cara mengakses website admin sama dengan cara mengakses website user melalui link yang sama namun untuk tampilan Dashboard admin berbeda dengan user. Untuk admin ada fitur news, arsip, pengurus, seleksi administrasi, seleksi wawancara, data profile.

3.1. Halaman Awal dan Login

Yang pertama admin akan diarahkan ke halaman dashboard aplikasi dengan mengakses domain website ini dengan cara sebagai berikut.

- Buka browser (Chrome S Mozilla Firefox)
- Masukan url : <https://www.ormawapoltekharber.my.id/> yang merupakan alamat website, seperti berikut :



- c) Kemudian admin bisa melakukan login dengan mengklik tombol “Masuk” dan nanti akan diarahkan ke halaman login, jika admin maka mengklik tombol “Login Admin” yang nanti diarahkan ke halaman login sebagai admin. Tampilan sebagai berikut.



Login Admin

Email

Password

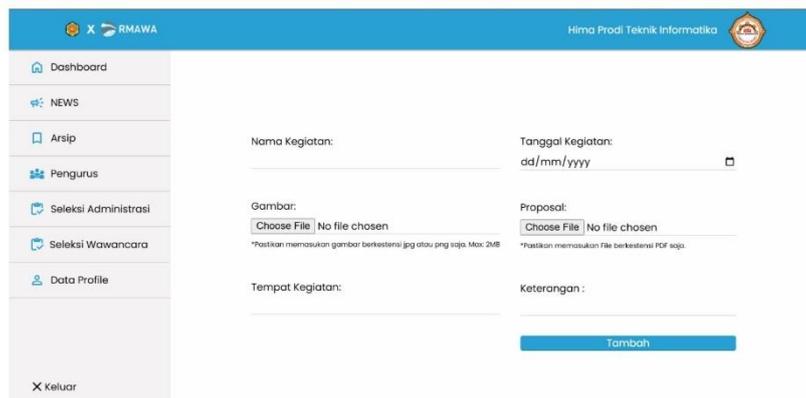
[Kembali](#)

3.2. Halaman Dashboard Admin

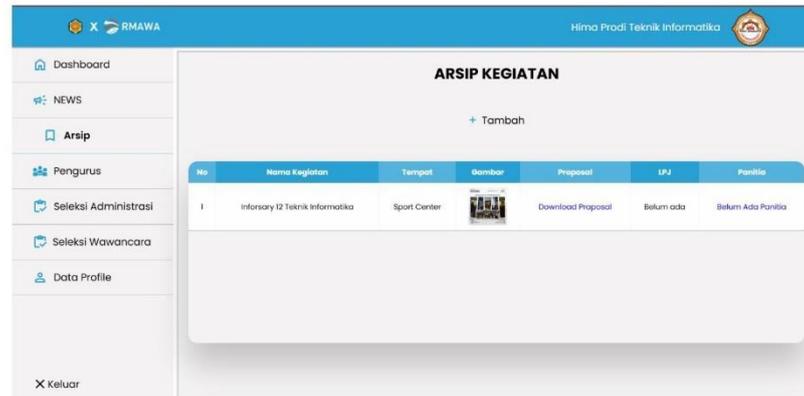
- a) Setelah admin melakukan login dengan akun admin maka akan diarahkan ke dashboard admin yang memiliki fitur berbeda dengan dashboard user. Berikut beberapa fitur yang ada pada dashboard admin yaitu 'Dashboard', 'News', 'Arsip', 'Pengurus', 'Seleksi Administrasi', 'Seleksi Wawancara', 'Data Profile'. Tampilan sebagai berikut.



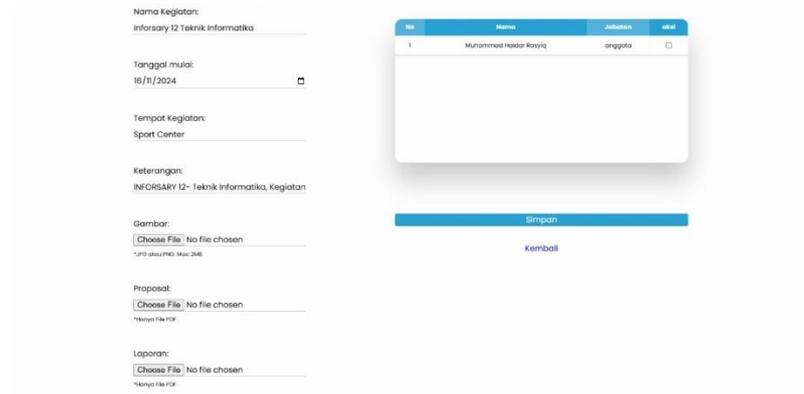
- b) Pada fitur Arsip admin bisa menambahkan 'Nama Kegiatan', 'Tanggal Kegiatan', 'Gambar', 'Proposal', 'Tempat Kegiatan', 'Keterangan'. Setelah semuanya sudah terisi dengan lengkap mengklik tombol 'Tambah'. Tampilan seperti berikut.



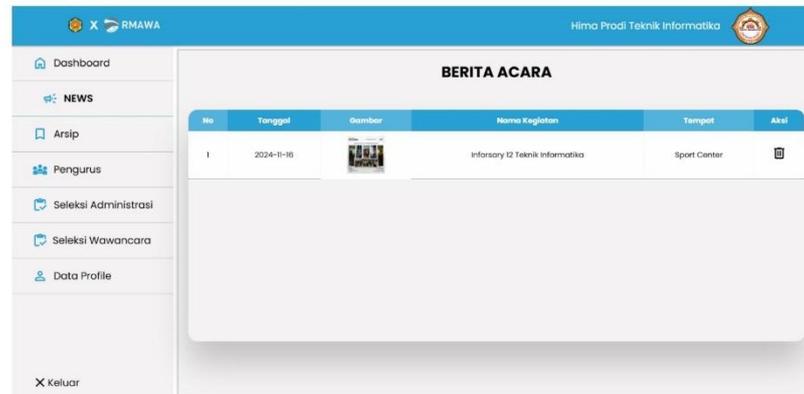
- c) Setelah berhasil menambahkan arsip dan terlaksananya kegiatan tersebut dibuatlah laporan pertanggungjawaban, untuk menambahkan dokumen laporan pertanggungjawaban yaitu mengklik tombol “Belum Ada Panitia”. Tampilan seperti berikut.



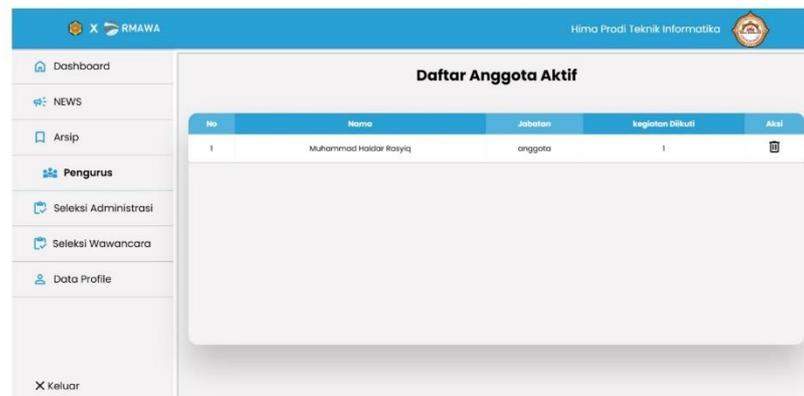
- d) Kemudian akan di arahkan ke halaman ‘Panitia’, admin bisa mengubah, menambahkan laporan pertanggungjawaban dan menambahkan panitia pada kegiatan tersebut. Kalo semuanya sudah ditambahkan silahkan mengklik tombol ‘Simpan’. Tampilan seperti berikut.



- e) Setelah sudah menambahkan dokumen arsip, informasi tersebut akan muncul pada fitur “News”, di fitur tersebut hanya bisa menghapus informasi saja. Tampilan sebagai berikut.



- f) Kemudian pada fitur “Pengurus” admin hanya dapat menghapus dan melihat berapa banyak pengurus tersebut mengikuti kegiatan, cara untuk mengetahuinya yaitu mengklik pada “Angka” di kegiatan diikuti. Tampilan seperti berikut.



- g) Selanjutnya yaitu fitur “Seleksi Administrasi” pada fitur ini terdapat tabel yang berisi ‘Nama’, ‘Nomer whatsapp’, ‘Aksi’, jika ingin mengetahui kelengkapan dokumen calon anggota maka mengklik tombol ‘pensil’ pada kolom aksi. Tampilan seperti berikut.



No	Nama	Nomer Whatsapp	Aksi
1	Muhammad Haidar Rasyiq	089872136764	 

- h) Setelah mengklik tombol tersebut akan diarahkan pada halaman administrasi calon anggota yang terdiri dari yaitu ‘Prodi’, ‘Nim’, ‘Semester’, ‘Nomor Whatsapp’, dan bisa mendownload kelengkapan dokumentasi calon anggota caranya yaitu mengklik pada tombol masing-masing dokumen. Tampilan seperti berikut.



Muhammad Haidar Rasyiq



Prodi: DIV Akuntansi Sektor Publik
NIM: 20090087
Semester: 1
Nomor Wa: 089872136764

[Kembali](#)

Download [file-foto/_nas-foto](#)
Download [file-ktm/_ktm](#)
Download [file-studi..._studi](#)
Download [file-serti..._sertif](#)

[Terima](#) [Tolak](#)

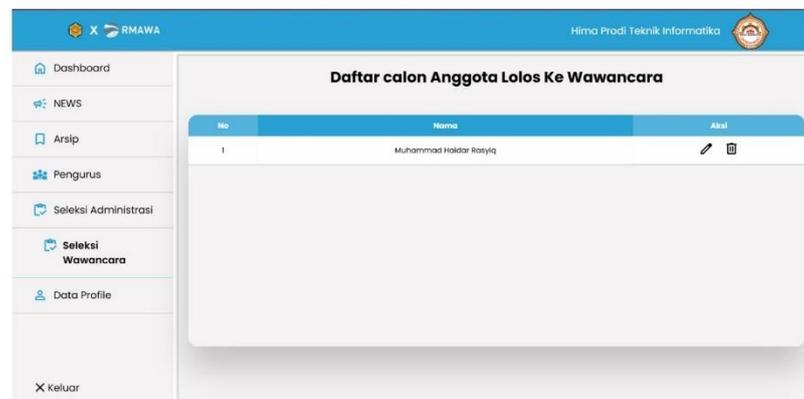
- i) Jika administrasi tersebut lengkap dan sesuai persyaratan, maka admin dapat menerima calon anggota tersebut untuk melakukan ke tahap selanjutnya. Cara untuk menerimanya yaitu mengklik tombol “Terima”, setelah mengklik tombol tersebut akan diarahkan pada Popup yang berisikan beberapa kolom yaitu kolom ‘Masukan alasan’, kolom ‘Tempat wawancara’, kolom ‘Tanggal Wawancara’, kolom ‘Jam Wawancara’, kolom tersebut wajib di isi semua. Jika semuanya sudah di isi dan sesuai jadwal maka mengklik tombol “Kirim”. Tampilan sebagai berikut.

The screenshot shows a modal window titled "Masukan Keterangan Aksi" with a close button (X). It contains four input fields: "Masukan Alasan", "Tempat Wawancara" (with a sub-label "Masukkan tempat wawancara"), "Tanggal Wawancara" (with a date picker icon and placeholder "dd/mm/yyyy"), and "Jam Wawancara" (with a time picker icon and placeholder "--:--"). At the bottom of the modal are two buttons: "Tutup" (Close) and "Kirim" (Send). The background shows a user profile with "Semester: 1" and "Nomor Wa: 089872136764". There are also "Kembali" (Back), "Terima" (Accept), and "Tolak" (Reject) buttons visible.

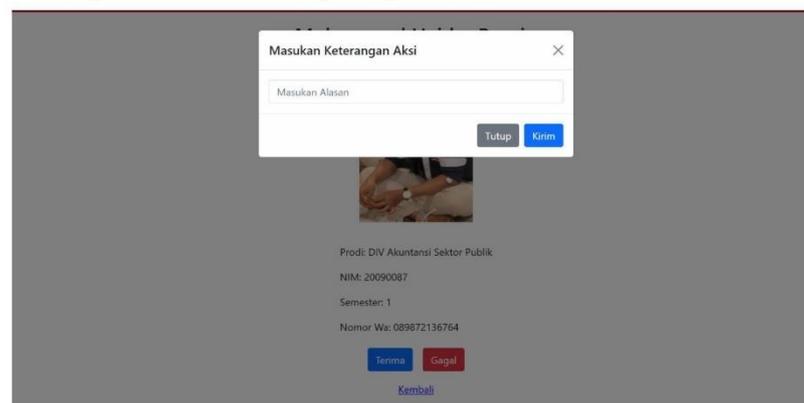
- j) Jika administrasi tersebut ada yang kurang atau tidak sesuai persyaratan, maka admin dapat menolak calon anggota tersebut. Cara untuk menolaknya yaitu mengklik tombol “Tolak” maka akan diarahkan pada Popup yang berisikan kolom ‘Masukan Alasan’, jika kolom tersebut sudah terisi maka mengklik tombol “Kirim”. Tampilan seperti berikut.

The screenshot shows a modal window titled "Masukan Keterangan Aksi" with a close button (X). It contains a single input field labeled "Masukan Alasan". At the bottom of the modal are two buttons: "Tutup" (Close) and "Kirim" (Send). The background shows a user profile with "Prodi: DIV Akuntansi Sektor Publik" and "NIM: 20090087". There are also "Kembali" (Back), "Terima" (Accept), and "Tolak" (Reject) buttons visible.

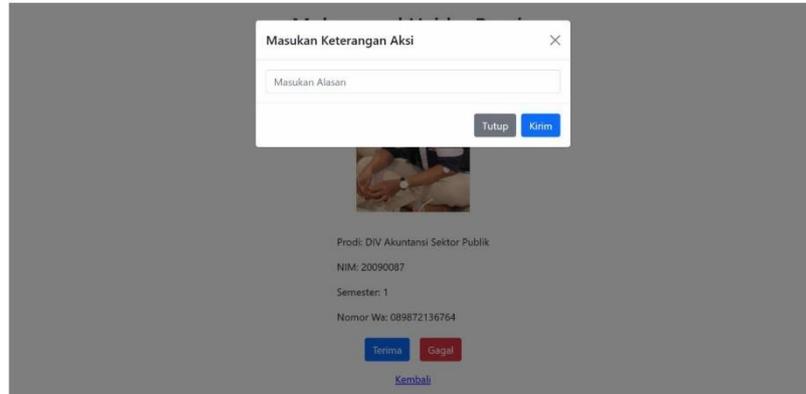
- k) Selanjutnya fitur “Seleksi Wawancara”, pada fitur tersebut terdapat tabel berisi ‘No’, ‘Nama’, ‘Aksi’, jika ingin mengetahui identitas calon anggota dapat mengklik tombol ‘Pencil’. Tampilan sebagai berikut.



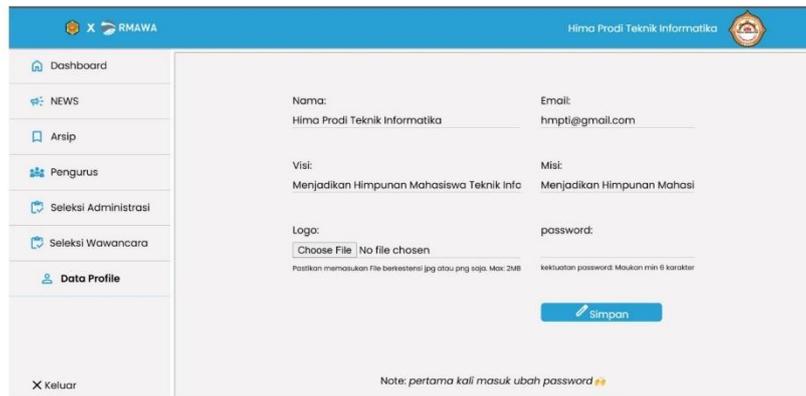
- l) Jika sudah mengklik tombol tersebut maka akan diarahkan pada halaman identitas, kemudian setelah calon anggota melakukan seleksi wawancara jika ingin di terima maka mengklik tombol “Terima”, nanti akan diarahkan pada popup yang berisikan kolom ‘Masukan Alasan’ yang wajib di isi, jika sudah terisi maka mengklik tombol “Kirim”. Tampilan seperti berikut.



- m) Kemudian setelah calon anggota melakukan seleksi wawancara jika ingin ditolak maka mengklik tombol “Gagal”, nanti akan diarahkan pada popup yang berisikan kolom ‘Masukan Alasan’ yang wajib di isi, jika sudah terisi maka mengklik tombol “Kirim”. Tampilan seperti berikut.



- n) Selanjutnya yaitu fitur “Data Profile”, didalam fitur tersebut terdapat beberapa kolom yaitu ‘Nama’, ‘Email’, ‘Visi’, ‘Misi’, ‘Logo’, ‘Password’. Kolom tersebut dapat di perbarui jika ada perubahan terhadap unit organisasi mahasiswa, jika semuanya sudah di perbarui maka mengklik tombol “Simpan”. Tampilan seperti berikut.



4. Super Admin

Pada Website Super Admin untuk tampilan awal Dashboard dan Login sama dengan admin hanya saja untuk akun Super admin dibuatkan oleh Developer sehingga Super admin bisa langsung Login dengan akun yang telah dibuatkan oleh Developer sebelumnya. Cara mengakses website Super Admin sama dengan cara mengakses website admin melalui link yang sama namun untuk tampilan Dashboard Super Admin berbeda dengan admin. Untuk Super Admin ada fitur News, Arsip, Data Ormawa, Create User.

4.1 Halaman Awal dan Login

Yang pertama Super admin akan diarahkan ke halaman dashboard aplikasi dengan mengakses domain website ini dengan cara sebagai berikut.

- a) Buka browser (Chrome S Mozilla Firefox)
- b) Masukan url : <https://www.ormawapoltekharber.my.id/> yang merupakan alamat website, seperti berikut :



- c) Kemudian super admin bisa melakukan login dengan mengklik tombol “Masuk” dan nanti akan diarahkan ke halaman login, jika admin maka mengklik tombol “Login Admin” yang nanti diarahkan ke halaman login sebagai admin. Tampilan sebagai berikut.



The screenshot shows a login form titled "Login Admin". At the top, there is a logo with a shield, the letter "X", and a hand icon. Below the logo, the text "Login Admin" is centered. There are two input fields: "Email" and "Password". A blue button labeled "Login" is positioned below the password field. A link labeled "Kembali" is located at the bottom left of the form area.

4.2 Halaman Dashboard Super Admin

- a) Setelah Super admin melakukan login dengan akun super admin maka akan diarahkan ke dashboard super admin yang memiliki fitur berbeda dengan dashboard user dan admin. Berikut beberapa fitur yang ada pada dashboard Super admin yaitu 'Dashboard', 'News', 'Arsip', 'Data Ormawa', 'Create User'. Tampilan sebagai berikut.



- b) Kemudian fitur “News”, fitur “News” pada super admin berbeda dengan admin yaitu hanya bisa melihat informasi saja. Tampilan seperti berikut.

No	Tanggal	Gambar	Nama Kegiatan	Tempat
1	2024-08-24		Seminar mental Health	Aula
2	2024-11-16		Inforsary 12 Teknik Informatika	Sport Center

- c) Selanjutnya fitur “Arsip”, fitur “Arsip” pada super admin berbeda dengan admin yaitu terdapat tabel yang berisi ‘No’, ‘Nama Kegiatan’, ‘Tempat’, ‘Gambar’, ‘Proposal’, ‘Lpj’, ‘Panitia’. Pada fitur ini super admin hanya dapat mendownload pada kolom proposal dan lpj, dengan cara mengklik pada setiap kolom proposal dan lpj kegiatan yang akan di download, selain itu bisa melihat berapa banyak panitia yang mengikuti pada setiap kegiatan dengan cara mengklik angka pada kolom panitia. Tampilan seperti berikut.

No	Nama Kegiatan	Tempat	Gambar	Proposal	LPJ	Panitia
1	Seminar mental Health	Aula		file-proposal/1FF8Vs...	file-lpj/a45Jmu23JG...	Belum Ada Panitia
2	Inforsary 12 Teknik Informatika	Sport Center		file-proposal/7v9T5D...	file-lpj/sH0ikpHf8W...	1

- d) Fitur “Data Ormawa”, pada fitur ini terdapat tabel anggota organisasi mahasiswa yang terdaftar dalam kepengurusan setiap unit. Tampilan seperti berikut.

No	Nama	WhatsApp	Unit	Jabatan
1	Muhammad Haidar Rosyiq	08987236764	Himp. TI	Ketua Himpahprodi

- e) Fitur “Create User”, pada fitur ini terdapat beberapa kolom yaitu ‘Nama’, ‘Unit’, ‘Email’. Kolom tersebut wajib di isi dan jika sudah di isi semua maka mengklik Tombol “Tambah”. Fitur ini untuk menambahkan akun admin (unit ormawa). Tampilan seperti berikut.

Nama: _____ Unit: _____

Email: _____

Note: Tolong double check untuk data yang di input

Lampiran 5. Sertifikat HKI yang Terbit

REPUBLIC INDONESIA KEMENTERIAN HUKUM	
SURAT PENCATATAN CIPTAAN	
Dalam rangka perlindungan 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	: EC002025033401, 20 Maret 2025
Pencipta	
Nama	: Muhammad Haidar Rasyiq, Dyah Apriliani, S.T., M.Kom. dkk
Alamat	: Jl. Cissanggarung Dalam no.06 RT/RW 04/04, Kaligangsa Wetan, Brebes, Kab. Brebes, Jawa Tengah, 52217
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat Politeknik Harapan Bersama
Alamat	: Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: IMPLEMENTASI CHATBOT PADA SISTEM INFORMASI ORGANISASI MAHASISWA BERBASIS WEBSITE (STUDI KASUS : ORGANISASI MAHASISWA POLITEKNIK HARAPAN BERSAMA TEGAL)
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 20 Maret 2025, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	: 000873662
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 DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL u.b Direktur Hak Cipta dan Desain Industri	
 Agung Damarsasongko,SH.,MH. NIP. 196912261994031001	
	
	Disclaimer: 1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan. 2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara. 3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Muhammad Haidar Rasyiq	Jl. Cissanggarung Dalam no.06 RT/RW 04/04, Kaligangsa Wetan Brebes, Kab. Brebes
2	Dyah Apriliani, S.T., M.Kom.	Perumahan Mutiara Indah Blok C4, Jalan Nangka Gang 2 Rt 02 Rw 02 Kel. Procot Slawi, Kab. Tegal
3	Arif Hidayah, M.Tr.Kom	Jl. Bulu Emas No. 08 RT.01 RW.01 Kel. Lawatan Dukuhturi, Kab. Tegal



Lampiran 6. Lembar Bimbingan

PEMBIMBING I



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Muhammad Haidar Rasyiq
Nim : 20090087
No. Ponsel : 085712378089
Judul TA : Implementasi *Chatbot* pada Sistem Informasi Organisasi Mahasiswa Berbasis *Website* (Studi Kasus : Organisasi Mahasiswa Politeknik Harapan Bersama Tegal)
Dosen Pembimbing I : Dyah Apriliani, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	19/2024 29/2024 /2	Perencanaan	- Alur aplikasi - Ulor - Diagram aplikasi	
2.	19/2024 /4	Aplikasi	- Tampilan aplikasi	
3.	9/23 /6	Aplikasi	- penuh dan lanjutkan.	



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
4	1/7 1/7 2024	Aplikasi	- Tampilan	
5	1/7 2024	Aplikasi	- Tampilan - Chatbot	
6	22/7 2024	Aplikasi	- Tampilan - Chatbot - Chatbot	
7	18/8 2024	Aplikasi	- Jadwal \rightarrow Ulem - button max ben - Validasi - proses akhir registrasi - laporan de ②	
8	26/8 2024	Aplikasi	* laporan dan - item frame form Register profil * dan seleksi	



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
9.	20/2024 9	Aplikasi.	- anal. - sen. - ui/ux. - Tahapan rektn. -	
10.	28/10 24	Aplikasi.	- Prodi → sem → tugas - Verifikasi → login. - Spiker	
11.			ui/ux tampilan - Penambahan user. - Himp → adm- - Revisi Menu folder	
12.	3/1 2025	Aplikasi		



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
	21/2025 /1	Dok. Manu Bode.	Perbaiki f Dok. <u>teknik</u>	
	11/2025 /3	Pok Mac dan doir tekn	Revisi	
	16/2025 /4	Paporan 1.2.3	Revisi Gab 1 Lamput 2.3.	
	21/2025 /5	Laporan	Revisi	
	24/2025 /6	Laporan	Revisi	
	23/2025 /6	Laporan	Revisi	



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
	24/2025 /6	Laporan	Acc.	

Tegal, 24 ^{Jun} 2025
Dosen Pembimbing I

Dyah Apriliani, S.T., M.Kom.
NIPY. 09.015.255

PEMBIMBING II



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Muhammad Haidar Rasyiq
 Nim : 20090087
 No. Ponsel : 085712378089
 Judul TA : Implementasi *Chatbot* pada Sistem Informasi Organisasi Mahasiswa Berbasis *Website* (Studi Kasus : Organisasi Mahasiswa Politeknik Harapan Bersama Tegal)
 Dosen Pembimbing II : Arif Hidayah, M.Tr.Kom

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1	29/7/2024	Proposal	- lebih jelas dalam penyampaian	
2	16/7/2024	Aplikasi	- perbaiki konsep ORSP dan absensi	
3	12/8/2024	Aplikasi	- form validation - redirect after submit - Strong Kekuatan password. - captcha - Captcha	
4	22/8/2024	Aplikasi		
5	16/12/2024	Aplikasi		



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
	7/1 2025	Aplikasi	- Perbaikan Bug - Riwayat Study diberi contoh.	
	22/1 2025	Aplikasi	Lanjutan //	
	13/3 2025	Pengajuan HKI	Pengajuan	
	17/3 2025	Perbaikan Pengajuan HKI	Perbaikan.	
	23/5 2025	Laporan	- Perbaiki - lanjut bab selanjutnya.	
	13/6 2025	Laporan	- Perbaiki: Use case	
	19/6 2025	Laporan	- Lanjutkan //	



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
	24/6 2025	relaporan	ACC	

24 Juni 2025
Tegal, 29 April 2024
Dosen Pembimbing II

Arif Hidayah, M.Tr.Kom
NIPY.