

LAMPIRAN

Lampiran 1. 1 Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama	: Nabilla Auly Zahra
NIM	: 21090094
Program Studi	: Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama	: Ir. Ginanjar Wiro Sasmito, M.Kom. IPM., ASEAN Eng
Status	: Dosen Tetap
NIDN	: 0613028601
Jabatan Fungsional	: Lektor Kepala
Pangkat/Golongan	: Penata Tk.I/III D

Pada hari ini Rabu tanggal 12 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali pertemuan kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir

Tegal, 12 Maret 2025

Pihak Pertama	Pihak Kedua
	
Nabilla Auly Zahra	Ir. Ginanjar Wiro Sasmito, M.Kom. IPM., ASEAN Eng


Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika

Dyah Apriliani, S.T., M.Kom.
NIPY: 09.0151225

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Nabilla Auly Zahra
NIM : 21090094
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

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

Pada hari ini Kamis tanggal 13 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 2 minggu sekali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

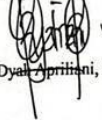
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir

Tegal, 13 Maret 2025

Pihak Pertama


Nabilla Auly Zahra

Pihak Kedua


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

Mengesahkan
Ketua Program Studi Sarjana Terapan Teknik Informatika


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

Lampiran 1. 2 Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Nabilla Auly Zahra
Kewarganegaraan : Indonesia
Alamat : Jln. Perintis Kemerdekaan Gang 27 No. 10 RT 09 RW 06 Kelurahan Panggung Kecamatan Tegal Timur
NIK : 3376025205030004
2. Nama : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jln. Raya Kluwut Timur No. 24, Kecamatan Bulakamba, Kabupaten Brebes.
NIK : -
3. Nama : Dyah Apriliani, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Slawi, Jalan Nangka No.2, Procot, Procot Blok C4 SLAWI SLAWI JAWA TENGAH.
NIK : -

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Sistem Rekomendasi ORMAWA pada Poltek Harber menggunakan *Content Based Filtering* dan algoritma *Cosine Similarity*
 - 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, 23 Juni 2025



(Nabilla Auly Zahra)
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 1. 3 Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Nabilla Auly Zahra
Kewarganegaraan : Indonesia
Alamat : Jln. Perintis Kemerdekaan Gang 27 N0. 10 RT 09 RW 06 Kelurahan Panggung Kecamatan Tegal Timur
NIK : 3376025205030004
2. Nama : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jln. Raya Kluwut Timur No. 24, Kecamatan Bulakamba, Kabupaten Brebes.
NIK : -
3. Nama : Dyah Apriliani, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Slawi, Jalan Nangka No.2, Procot, Procot Blok C4 SLAWI SLAWI JAWA TENGAH.
NIK : -

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

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

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Sistem Rekomendasi ORMAWA pada Poltek Harber menggunakan *Content Based Filtering* dan algoritma *Cosine Similarity*". 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.

Tegal, 23 Juni 2025

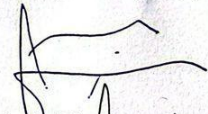
Pemegang Hak Cipta
Kepala P3M



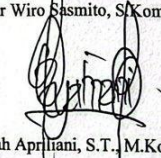
(Muhammad Fikri Hidayatullah, S.T., M.Kom.)



(Nabilla Auly Zahra)



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



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



PENDAHULUAN



Selamat datang di Sistem Rekomendasi ORMAWA Poltek Harber, sebuah aplikasi pintar yang dirancang untuk membantu mahasiswa menemukan organisasi dan Unit Kegiatan Mahasiswa (UKM) yang paling sesuai dengan minat, bakat, dan preferensi pribadi mereka. Dalam lingkungan kampus yang dinamis dan penuh peluang, memilih organisasi yang tepat dapat menjadi langkah penting dalam pengembangan diri mahasiswa.

Sistem ini memanfaatkan pendekatan Content-Based Filtering dan algoritma Cosine Similarity untuk menganalisis karakteristik mahasiswa dan mencocokkannya dengan profil masing-masing ORMAWA secara akurat. Dengan teknologi ini, rekomendasi yang diberikan bersifat personal, relevan, dan mampu mendukung mahasiswa dalam mengambil keputusan yang tepat untuk pengembangan akademik maupun non-akademik.

Sistem Rekomendasi ORMAWA Poltek Harber dirancang untuk dapat diakses dengan mudah oleh seluruh mahasiswa, dan memberikan pengalaman yang menyeluruh dan adaptif terhadap kebutuhan individu. Aplikasi ini menjadi sarana efektif dalam menjembatani mahasiswa dengan organisasi yang mampu mengasah potensi serta menumbuhkan semangat berorganisasi di lingkungan kampus.



MEMULAI PENGGUNAAN APLIKASI

2

Bab ini memberikan panduan awal bagi pengguna untuk mulai mengakses dan menggunakan Sistem Rekomendasi ORMAWA di Politeknik Harapan Bersama. Proses ini mencakup langkah-langkah untuk membuka aplikasi hingga pengguna berhasil masuk ke halaman utama sistem.

- Cara Membuka Aplikasi

Langkah awal yang perlu dilakukan adalah membuka peramban internet seperti Google Chrome, Mozilla Firefox, atau browser lainnya pada perangkat komputer Anda.

Kemudian, ketik alamat sistem berikut pada kolom pencarian browser:

URL Aplikasi: <http://ukm-ta.test/>

Catatan: Alamat ini bersifat lokal dan akan disesuaikan jika aplikasi sudah tersedia secara online.

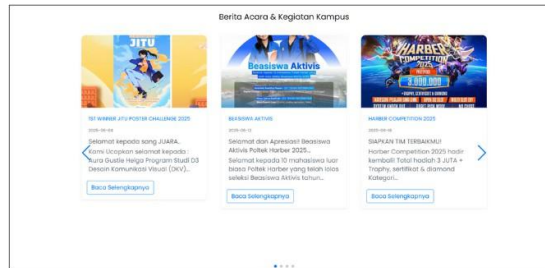
Setelah itu, tekan Enter. Sistem akan menampilkan halaman login sebagai pintu masuk ke dalam aplikasi.



Gambar 1. Beranda

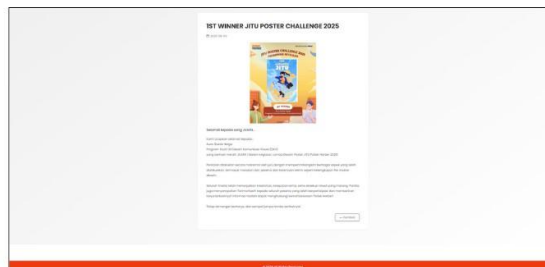
Saat pertama kali mengakses platform ORMAWA CONNECT, pengguna akan diarahkan ke halaman beranda. Pada halaman ini, pengguna akan disambut dengan banner utama dan judul platform yang mencerminkan identitas serta tujuan ORMAWA CONNECT. Terdapat pula navigasi yang memudahkan pengguna untuk mengakses berbagai fitur yang tersedia. Selain itu, beranda menampilkan berita terbaru terkait organisasi mahasiswa, deskripsi singkat mengenai fungsi dan manfaat platform, serta video informatif mengenai kehidupan kampus di Politeknik Harapan Bersama.

TAMPILAN BERANDA



Gambar 1. Berita

Pada halaman beranda, fitur Berita Terbaru menyajikan informasi terkini seputar kemahasiswaan, seperti kegiatan organisasi, pengumuman penting, dan berita kampus lainnya di Politeknik Harapan Bersama. Untuk melihat detail berita, pengguna dapat mengklik tombol "Baca Selengkapnya" pada setiap artikel yang ditampilkan.



Gambar 2. Detail Berita

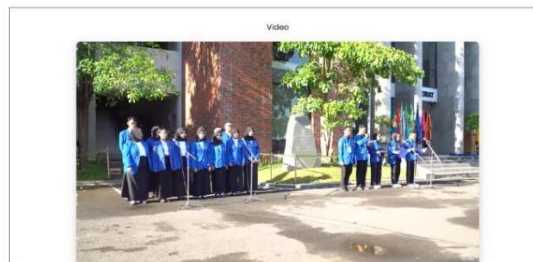
klik tombol "kembali" untuk kembali ke halaman awal dan melihat fitur dan tampilan lainnya.

TAMPILAN BERANDA



Gambar 3. Tentang

Setelah Berita Terbaru, terdapat bagian Deskripsi Singkat yang menjelaskan tujuan, manfaat, dan fungsi utama ORMAWA CONNECT dalam mendukung pengembangan diri dan partisipasi mahasiswa di Politeknik Harapan Bersama.



Gambar 4. Video

Sebagai penutup halaman beranda, ditampilkan video dari YouTube resmi Politeknik Harapan Bersama yang memperlihatkan kegiatan kemahasiswaan untuk memberi gambaran nyata dan menarik minat terhadap ORMAWA CONNECT. Untuk melihat lebih lanjut, pengguna dapat mengklik "Video" yang akan mengarahkan langsung ke kanal YouTube resmi Poltek Harber.

FITUR TENTANG

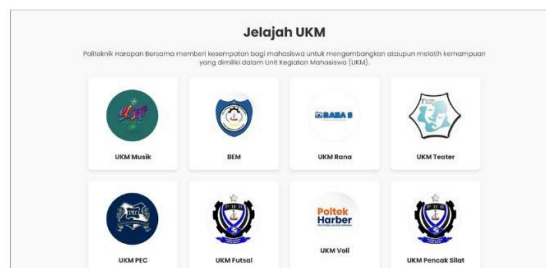
5

Klik menu "Tentang" pada navigasi untuk menuju halaman yang berisi informasi lengkap.



Gambar 5. Fitur Tentang

Pada Fitur ‘Tentang’ terdapat tampilan jelajah ormawa. Tampilan Jelajah ORMAWA menampilkan daftar organisasi dan UKM aktif di Politeknik Harapan Bersama. Fitur ini memungkinkan pengguna mengeksplorasi informasi seperti nama, jenis kegiatan, deskripsi singkat, dan status keaktifan setiap ORMAWA.



Gambar 6. Jelajah Ormawa

Untuk melihat detail tiap organisasi, klik yang ada dalam card atau klik “logo” nanti akan menampilkan detail dari ormawa tersebut.



Gambar 7. Detail Ormawa

FITUR REKOMENDASI



Fitur ini merupakan fitur utama dalam aplikasi yang berfungsi memberikan rekomendasi organisasi atau UKM yang sesuai dengan preferensi dan karakteristik mahasiswa.

Langkah Memulai Rekomendasi:

1. Masuk ke Halaman Rekomendasi
2. Pilih salah satu dari 2 opsi
3. Isi Form Preferensi (Pengguna diminta untuk menjawab beberapa pertanyaan seputar minat, jenis kegiatan yang disukai, tujuan berorganisasi, dan komitmen.)
4. Setelah semua pertanyaan dijawab, klik tombol “Kirim” untuk memproses preferensi.
5. Lihat Hasil Rekomendasi

Sistem akan menampilkan daftar organisasi atau UKM yang paling sesuai dengan preferensi pengguna berdasarkan proses dari *Content-Based Filtering* dengan perhitungan algoritma *Cosine Similarity*.

The screenshot shows the 'Poltek Harber' website interface. The top navigation bar includes 'BERANDA', 'TENTANG', 'REKOMENDASI', and 'RECRUITMENT'. The main heading is 'Temukan Rekomendasi Organisasi Mahasiswa yang Cocok untuk Anda'. Below this, there are two orange buttons: 'UKM' and 'Ormanwa'. The lower part of the image shows a form titled 'Temukan UKM yang Cocok untuk Anda' with several questions and a 'Lihat Rekomendasi' button. The form includes a dropdown for 'Apakah kamu berminat untuk mencari organisasi?', a dropdown for 'Apakah kamu berminat untuk mencari UKM?', a dropdown for 'Apakah kamu berminat untuk mencari Ormanwa?', and a dropdown for 'Apakah kamu berminat untuk mencari organisasi?'. The 'Lihat Rekomendasi' button is highlighted in blue.

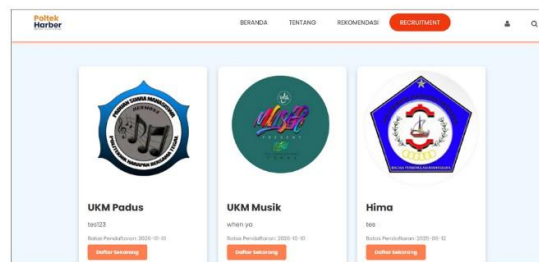
Gambar 8. Rekomendasi

Setelah menerima hasil rekomendasi yang sesuai dengan preferensi, pengguna akan mendapatkan saran organisasi yang cocok. Selanjutnya, pengguna dapat melanjutkan ke halaman "Recruitment" untuk melakukan pendaftaran pada organisasi yang dipilih.

FITUR RECRUITMENT

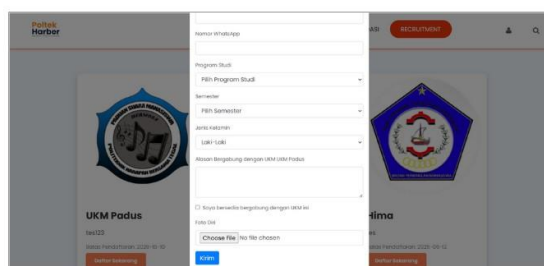
7

Fitur Recruitment merupakan bagian dari aplikasi yang digunakan untuk mendaftarkan diri ke ORMAWA atau UKM yang sedang membuka open recruitment. Adapun langkah-langkah pendaftaran sebagai berikut :



Gambar 9. Fitur Recruitment

1. Akses Halaman Recruitment
2. Klik menu “Recruitment” pada navigasi utama untuk melihat daftar ORMAWA yang sedang membuka pendaftaran.
3. Pilih Organisasi yang Direkomendasikan
4. Tinjau daftar ORMAWA berdasarkan hasil rekomendasi atau eksplorasi mandiri.
5. Isi Formulir Pendaftaran
6. Lengkapi data diri dan pertanyaan tambahan yang dibutuhkan oleh organisasi terkait.
7. Kirim Formulir
8. Setelah semua kolom terisi, klik tombol “Daftar” untuk mengirimkan pendaftaran.
9. Tunggu Konfirmasi
10. Penerimaan akan diinformasikan melalui notifikasi email aktif yang terdaftar yang akan dikirim secara manual oleh admin.



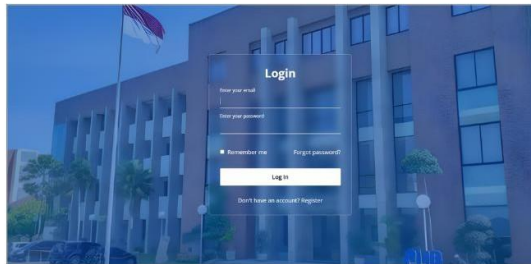
Gambar 10. Registrasi

FITUR LOGIN



Fitur Login pada aplikasi ini disediakan khusus untuk dua jenis pengguna:

1. Super Admin – Memiliki akses penuh untuk mengelola akun user dan menambahkan konten berita saja.
2. Admin ORMAWA – Merupakan admin dari masing-masing organisasi atau UKM, yang hanya dapat mengakses dan mengelola data terkait organisasinya sendiri seperti data rekomendasi, recruitment, dan dokumentasi kegiatan lainnya.



Gambar 11. Registrasi

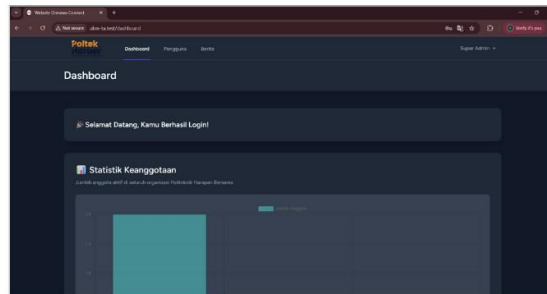
Langkah Login Super admin:

1. Masukkan email
2. Masukkan Password
3. Klik Tombol Login
4. Jika berhasil akan menampilkan halaman dashboard Super Admin

Langkah Login Admin :

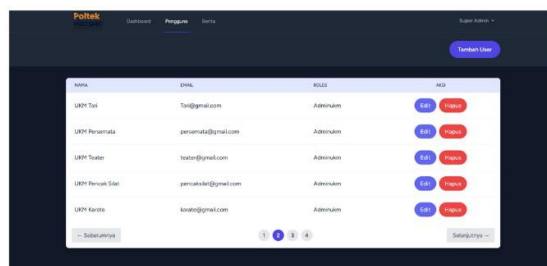
1. Minta buat akun pada Super Admin
2. Masukkan email yang sudah dibuat oleh Super Admin
3. Masukkan Password yang sudah dibuat oleh Super Admin
4. Klik Tombol Login
5. Jika berhasil akan menampilkan halaman dashboard Admin

Super Admin



Gambar 12. View Dashboard Super Admin

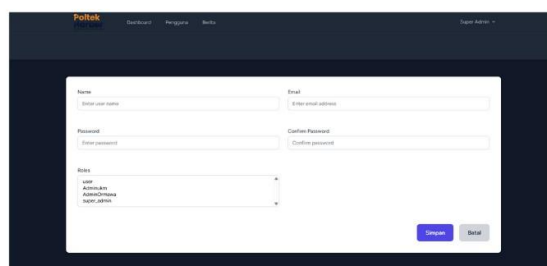
Pada tampilan awal super admin terdapat dashboard yang menampilkan grafik diagram yang merupakan jumlah anggota dari setiap organisasi yang aktif.



Gambar 13. View akun pengguna

Pada tampilan ini untuk mengelola seperti tambah, edit, hapus akun pengguna. adapun langkah-langkah membuat akun di super admin sebagai berikut :

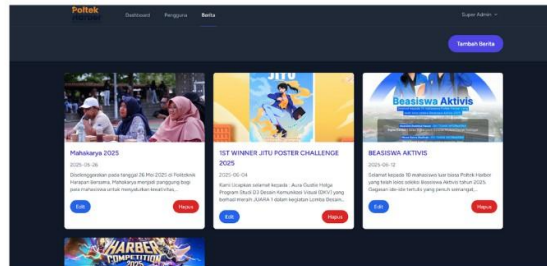
1. Klik tambah User
2. Isi format nama, email, password, konfirmasi password, roles
3. Klik simpan



Gambar 14. View tambah akun

Super Admin

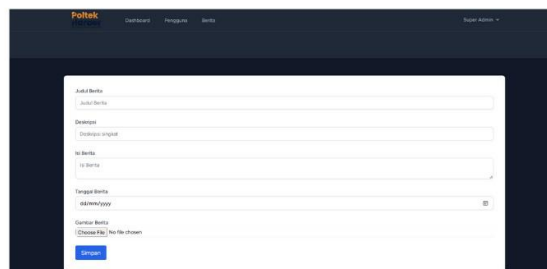
10



Gambar 15. View Berita

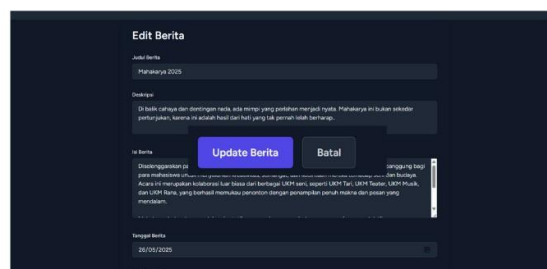
Pada tampilan ini view hasil dari berita yang telah ditambahkan untuk terhubung pada tampilan berita user. super admin dapat membuat berita, mengedit dan menghapus berita sesuai kebutuhan. Berikut langkah-langkah membuat berita baru:

1. Klik “Tambah Berita”
2. Isi format Judul berita, Deskripsi, Isi berita, Tanggal berita, dan gambar
3. Setelah itu klik “Simpan”



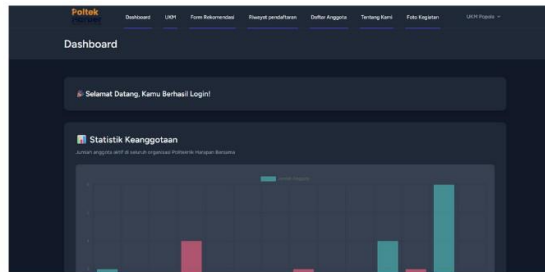
Gambar 16. View tambah Berita

Super admin juga dapat mengedit dan menghapus berita yang sudah tersimpan atau dibuat sesuai kebutuhan.



Gambar 17. View edit Berita

Admin tiap Ormawa



Gambar 18. View edit Berita

Pada tampilan awal admin tiap ORMAWA terdapat dashboard yang menampilkan grafik diagram yang merupakan jumlah anggota dari setiap organisasi yang aktif.

Gambar 18. View form Rekomendasi

Pada tampilan ini admin wajib mengisi formulir input yang sudah disediakan untuk memperoleh data yang nantinya akan di proses untuk sistem rekomendasinya. Berikut langkah-langkah untuk mengisi :

1. Klik navbar “Form Rekomendasi”
2. Isi semua formulir input sesuai preferensi tiap organisasi
3. Setelah itu klik “Simpan Data Admin UKM”

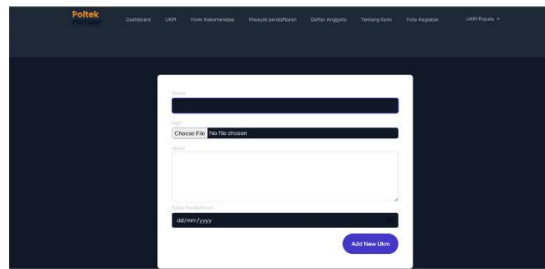
Jika sudah tersimpan, datanya akan langsung tersimpan pada database dan menampilkan view hasil jawaban yang sudah diisi pada formulir inputnya

	Pendaftaran 1	Pendaftaran 2	Pendaftaran 3	Pendaftaran 4	Pendaftaran 5	Pendaftaran 6	Pendaftaran 7	Pendaftaran 8
Organisasi pengembangan usaha lokal (Pusat pengembangan usaha lokal, lembaga usaha, lembaga keuangan)	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal
Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal
Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal
Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal
Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal
Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal
Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal	Pusat pengembangan usaha lokal

Gambar 19. View jawaban Rekomendasi

Admin tiap Ormawa

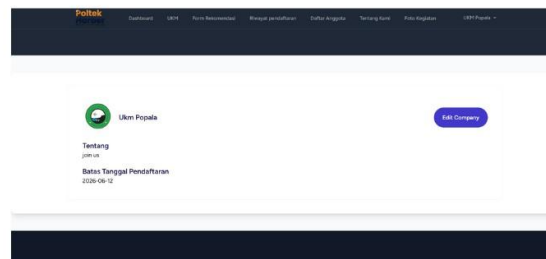
12



Gambar 20. View tambah Recruitment

Tampilan ini admin dapat menambahkan recruitment nya untuk membuka pendaftaran anggota baru. Berikut langkah-langkah :

1. Klik navigasi “UKM”
2. Isi format recruitment berupa nama organisasi, file logo, deskripsi promosi organisasi dan batas pendaftaran.
3. Klik “Add new ukm” untuk menyimpan dan agar tampil pada halaman recruitment pada user.

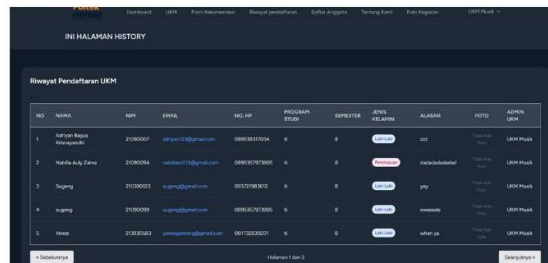


Gambar 20. View Recruitment

Setelah berhasil menambah akan berupa seperti tampilan diatas, admin juga dapat mengedit jika ada kesalahan.

Admin tiap Ormawa

13



INI HALAMAN HISTORY

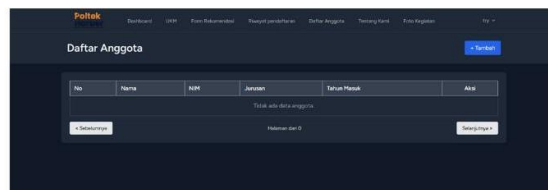
Riwayat Pendaftaran URM

NO	NAMA	NIM	EMAIL	NILAI IP	PROGRAM STUDI	SEKOLAH	JANIS KELAM	ALASAN	FOTO	AKTIF
1	Adrian Bapa Karyawati	21030001	adrian21@gmail.com	000000000000	0	0	0	0	0	0
2	Nakila Auli Zana	21030004	nakila21@gmail.com	000000000000	0	0	0	0	0	0
3	Ruging	21030003	ruging@gmail.com	000000000000	0	0	0	0	0	0
4	Ingung	21030009	ingung@gmail.com	000000000000	0	0	0	0	0	0
5	Nana	21030003	nana@gmail.com	000000000000	0	0	0	0	0	0

Halaman 1 dari 2

Gambar 21. View history pendaftar

Pada tampilan ini admin dapat melihat daftar riwayat pendaftaran mahasiswa yang mendaftar pada organisasinya.



Daftar Anggota

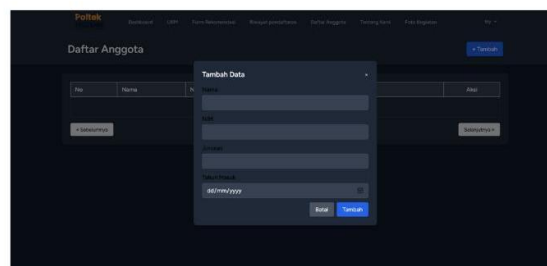
No	Nama	NIM	Jumlah	Tahun Masuk	Asal
Tidak ada data anggota					

Halaman 1 dari 0

Gambar 21. View daftar anggota

Tampilan ini berupa daftar anggota yang aktif pada organisasinya. Admin dapat mengelola dengan menambahkan, mengedit dan menghapus daftar anggota yang sudah tidak aktif lagi. Berikut langkah-langkah :

1. Klik “tambah” untuk menambah anggota yang aktif
2. Muncul pop up
3. Isi format nama anggota, NIM, jurusan, dan tahun masuk organisasi
4. Klik “Tambah” untuk menambah dan menyimpan hasil daftar anggota yang sudah ditambahkan



Daftar Anggota

Tambah Data

No	Nama	NIM	Jumlah	Tahun Masuk	Asal
Tidak ada data anggota					

Halaman 1 dari 0

Tambah Data

Nama

NIM

Jumlah

Tahun Masuk

Asal

000000000000

Tahun

Tambah

Gambar 21. View tambah anggota

Admin tiap Ormawa

14

Gambar 21. View jelajah ormawa

Pada tampilan ini, admin dapat menambahkan isi konten yang ada pada jelajah ormawa. Berikut langkah-langkah :

1. Klik navigasi “Tentang Kami”
2. Isi untuk menambahkan konten jelajah ormawa berupa format nama organisasi, file logo, deskripsi informasi tentang organisasi dan info yang dapat dihubungi.
3. Klik “Submit” untuk menyimpan dan agar tampil pada halaman jelajah ormawa pada user.

Gambar 21. View hasil jelajah ormawa

Setelah admin menambahkan isi untuk bagian jelajah ormawa, nanti hasilnya akan tampil pada halaman jelajah ormawa untuk informasi terkait organisasi.

Gambar 21. View jelajah ormawa

Admin tiap Ormawa

15

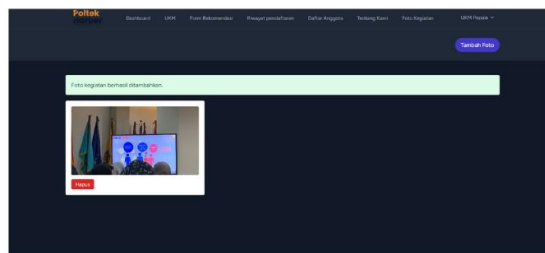
Pada tampilan ini admin dapat menambahkan foto kegiatan untuk di detail halaman jelajah ormawa. Berikut langkah-langkahnya

1. Klik navigasi “Foto Kegiatan”
2. Klik “Tambah Foto”
3. Klik “Choose File” untuk memilih foto kegiatan yang akan di upload sesuai kebutuhan
4. Klik “Simpan” untuk menyimpan dan agar tampil pada halaman detail jelajah ormawa pada user.

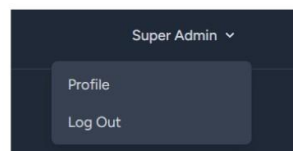


Gambar 21. View tambah foto kegiatan

Jika foto berhasil disimpan maka akan tampil seperti gambar berikut.



Gambar 21. View tambah foto kegiatan



Gambar 21. View Logout

Pada tampilan ini ketika super admin maupun admin per organisasi jika ingin melihat tampilan user-nya dapat klik “Profile”, jika super admin maupun admin per organisasi akan keluar dari sesi halaman admin bisa untuk mengklik “logout”.



PENDAHULUAN



Organisasi Kemahasiswaan (ORMAWA) merupakan wadah bagi mahasiswa dalam mengembangkan minat, bakat, kemampuan sosial, dan aspirasi di lingkungan kampus. Di Politeknik Harapan Bersama, terdapat 29 ORMAWA dengan beragam bidang seperti seni, olahraga, teknologi, keagamaan, dan organisasi internal kampus. Namun, hanya 19 ORMAWA yang aktif. Permasalahan yang sering muncul adalah mahasiswa mengalami kesulitan dalam mengakses informasi terkait open recruitment serta kebingungan dalam memilih organisasi yang sesuai dengan minat mereka. Hal ini berdampak pada rendahnya partisipasi mahasiswa dalam kegiatan organisasi.

Sebagai solusi, dikembangkan sistem rekomendasi ORMAWA berbasis machine learning yang menerapkan metode Content-Based Filtering. Sistem ini menggunakan pendekatan One Hot Encoding untuk merepresentasikan minat mahasiswa dalam bentuk vektor biner dan algoritma Cosine Similarity untuk menghitung tingkat kemiripan antara profil mahasiswa dan profil ORMAWA. Dengan demikian, sistem mampu memberikan rekomendasi yang relevan dan personal. Selain itu, fitur integrasi open recruitment memudahkan mahasiswa mengetahui ORMAWA mana saja yang sedang membuka pendaftaran, sehingga mereka dapat langsung mendaftar melalui satu platform yang terpusat.



SPEKIFIKASI TEKNIS

2

Spesifikasi Teknis meliputi :

1. Modul Pengguna

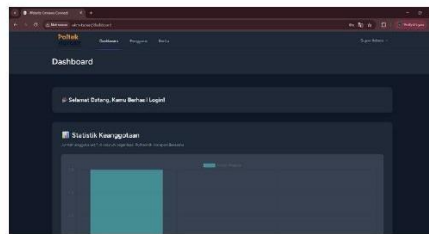
2. Source Code

Berikut uraian spesifikasi untuk pembangunan sistem :

1. Windows 11
2. VS Code
3. Web Browser

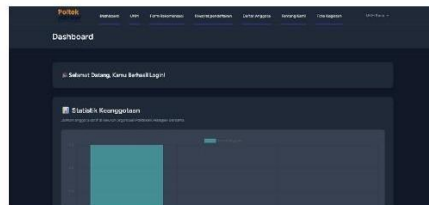
Berikut uraian spesifikasi modul :

1. Modul untuk super admin



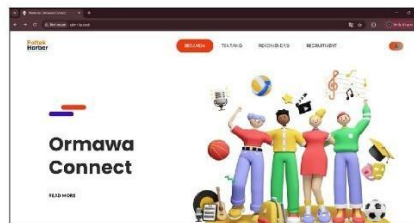
Bagian ini hanya dapat diakses oleh super admin untuk mengelola akun tiap organisasi dan mengelola berita seputar kegiatan yang ada di kampus.

2. Modul untuk admin per organisasi



Bagian ini hanya dapat diakses oleh admin per organisasi untuk mengelola isi konten dari platform seperti tampilan untuk jelajah ormawa, rekomendasi, histori pendaftar, dan daftar anggota aktif untuk ditampilkan pada statistik yang ada pada dashboard

3. Modul untuk Pengguna



Bagian ini adalah tampilan dari sistem ORMAWA CONNECT yang akan ditampilkan untuk pengguna beserta fitur tambahan yang tersedia seperti, fitur rekomendasi, fitur recruitment, fitur jelajah ormawa dsb.

SOURCE CODE

3

Source code back-end dari platform ORMAWA CONNECT menggunakan framework laravel yang digunakan sebagai route untuk melakukan login serta menampilkan isi konten super admin dan admin tiap organisasi seperti, untuk super admin yakni pembuatan akun untuk tiap organisasi dan menambahkan konten berita, untuk admin tiap organisasi yakni menambahkan konten pada fitur jelajah ormawa, menambahkan data formulir input untuk fitur rekomendasi, mengelola untuk bagian fitur recruitment.

1. Fitur Login

```
1 public function create(): View
2 {
3     return view('auth.login');
4 }
5
6 /**
7  * Handle an incoming authentication request.
8  */
9 public function store(LoginRequest $request): RedirectResponse
10 {
11     $request->authenticate();
12     $request->session()->regenerate();
13     return redirect()->intended(route('dashboard', absolute: false));
14 }
15
16 /**
17  * Destroy an authenticated session.
18  */
19 public function destroy(Request $request): RedirectResponse
20 {
21     Auth::guard('web')->logout();
22     $request->session()->invalidate();
23     $request->session()->regenerateToken();
24     return redirect('/');
25 }
26
27
28
29
30
```

Gambar 1 Controller Login

Pada bagian public function create() digunakan untuk menampilkan view pada tampilan UI login, lalu pada function store() digunakan sebagai pengiriman autentifikasi berdasarkan username dan password tersebut apakah sudah terdaftar atau belum. pada function destroy() digunakan oleh super admin untuk menghapus request username dan password yang terekam pada server untuk proses logout.

```
1 Route::get('login', [AuthenticatedSessionController::class, 'create'])
2     ->name('login');
```

Gambar 2. Route Login

'Route::get('login')' untuk mengambil controller "authenticatedSessionController" di controller tersebut, ia mengambil class yg bernama 'create' untuk masuk ke halaman login.


```
1 Route::post('logout', [AuthenticatedSessionController::class, 'destroy'])
2     ->name('logout');
```

'Route::get('logout')' untuk mengambil controller "authenticatedSessionController" di controller tersebut, ia mengambil class yg bernama 'destroy' untuk mengakhiri sesi login.

2. Dashboard super admin

[illegible]

ini merupakan tampilan dashboard awal pada super admin yang menampilkan statistik keanggotaan organisasi yang aktif berupa diagram batang dengan bantuan javascript untuk menampilkan diagram batang.

```
1 Route::middleware(['auth', 'can: manage pengguna'])->group(function () {
2     Route::resource('users', UserController::class);
3     Route::resource('berita', BeritaController::class);
4 });
```

ini merupakan route yang digunakan oleh super admin untuk mengelola data user dan berita.

SOURCE CODE

5

```

1 public function index()
2 {
3     $users = User::with('roles')->get();
4     return view('user.index', compact('users'));
5 }
6
7 public function create()
8 {
9     $roles = Role::all();
10    return view('user.create', compact('roles'));
11 }
12
13 public function store(Request $request)
14 {
15     $request->validate([
16         'name' => 'required|string|max:255',
17         'email' => 'required|email|unique:users,email',
18         'password' => 'required|string|min:8|confirmed', // Memastikan validasi untuk password
19         'roles' => 'required|array',
20     ]);
21
22     // Menentukan data user
23     $data = $request->only('name', 'email');
24
25     // Hash password
26     $data['password'] = Hash::make($request->password);
27
28     // Menetapkan user dan menetapkan role
29     $user = User::create($data);
30     $user->assignRole($request->roles);
31
32     return redirect()->route('users.index')->with('success', 'User created successfully.');
```

Gambar 3. Controller roles

Controller ini digunakan oleh super admin untuk mengelola user dengan fitur untuk menampilkan, menambah, mengedit, menghapus user beserta perannya, mengatur avatar, serta menampilkan riwayat pendaftaran organisasi. Fungsi index() untuk menampilkan semua user beserta peran (role)-nya. Fungsi create() Menampilkan form untuk membuat user baru. store(): Menyimpan user baru ke database, termasuk hashing password dan menetapkan peran. Fungsi edit(\$id): Menampilkan form edit user berdasarkan ID. Fungsi update(\$request, \$id): Memperbarui data user dan perannya. Jika password diisi, maka akan diubah juga. Fungsi destroy(\$id): Menghapus user kecuali jika user tersebut memiliki peran super_admin.

```

class BeritaController extends Controller
{
    // Menampilkan semua berita
    public function index()
    {
        $berita = Berita::all();
        return view('Berita.index', compact('berita'));
    }

    // Menampilkan berita detail
    public function show($id)
    {
        $berita = Berita::findOrFail($id);
        return view('Berita.show', compact('berita'));
    }

    // Menampilkan form untuk menambah berita
    public function create()
    {
        return view('Berita.create');
    }

    // Menyimpan berita baru
    public function store(Request $request)
    {
        $request->validate([
            'judul' => 'required|string|max:255',
            'isi' => 'required|string|max:1000',
            'kategori' => 'required|string',
            'gambar' => 'image|max:1024',
        ]);

        $data = [
            'judul' => $request->judul,
            'isi' => $request->isi,
            'kategori' => $request->kategori,
            'gambar' => $request->gambar,
        ];

        $berita = Berita::create($data);

        // Jika ada gambar yang diupload
        if ($request->hasFile('gambar')) {
            $file = $request->file('gambar');
            $path = $file->store('storage/foto_berita');
            $data['gambar'] = $path;
            $berita->update($data);
        }

        return redirect()->route('Berita.index');
    }

    // Menampilkan berita yang sudah diupdate
    public function update($id)
    {
        $berita = Berita::findOrFail($id);
        return view('Berita.update', compact('berita'));
    }

    // Menampilkan form untuk mengupdate berita
    public function edit($id)
    {
        $berita = Berita::findOrFail($id);
        return view('Berita.edit', compact('berita'));
    }

    // Memperbarui berita
    public function update(Request $request, $id)
    {
        $request->validate([
            'judul' => 'required|string|max:255',
            'isi' => 'required|string|max:1000',
            'kategori' => 'required|string',
            'gambar' => 'image|max:1024',
        ]);

        $data = [
            'judul' => $request->judul,
            'isi' => $request->isi,
            'kategori' => $request->kategori,
            'gambar' => $request->gambar,
        ];

        // Jika ada gambar yang diupload
        if ($request->hasFile('gambar')) {
            $file = $request->file('gambar');
            $path = $file->store('storage/foto_berita');
            $data['gambar'] = $path;
            $berita->update($data);
        }

        return redirect()->route('Berita.index');
    }

    // Menghapus berita
    public function destroy($id)
    {
        $berita = Berita::findOrFail($id);
        $berita->delete();

        // Jika ada gambar yang diupload
        if ($request->hasFile('gambar')) {
            $file = $request->file('gambar');
            $path = $file->store('storage/foto_berita');
            $data['gambar'] = $path;
            $berita->update($data);
        }

        return redirect()->route('Berita.index');
    }
}

```

Gambar 3. Controller Berita

BeritaController adalah controller yang mengelola semua proses terkait data berita, mulai dari menampilkan semua berita ke halaman admin dan publik, menampilkan form tambah dan edit, menyimpan data berita baru, memperbarui berita lama, hingga menghapus berita beserta gambar kegiatannya. Saat menyimpan atau memperbarui berita, controller ini juga menangani proses upload gambar secara manual ke folder storage/foto_kegiatan, dan menyimpan path-nya di database. Semua input divalidasi agar data yang masuk sesuai format yang ditentukan, dan jika ada gambar lama saat update atau delete, gambar tersebut ikut dihapus dari penyimpanan.

7

```

1  // Load the module and create the callback
2
3  // Create the object which implements the API
4  public function init() {
5
6      $user = $this->get('user');
7
8      $this->add('before', 'before_initialize', $user->id(), false);
9
10     return $this->create('user.create');
11
12     // if (false) {
13         return $this->create('user.create');
14     }
15
16     return $this->create('user.create');
17
18     // if (false) {
19         return $this->create('user.create');
20     }
21
22     return $this->create('user.create');
23
24     // if (false) {
25         return $this->create('user.create');
26     }
27
28     return $this->create('user.create');
29
30     // if (false) {
31         return $this->create('user.create');
32     }
33
34     return $this->create('user.create');
35
36     // if (false) {
37         return $this->create('user.create');
38     }
39
40     return $this->create('user.create');
41
42     // if (false) {
43         return $this->create('user.create');
44     }
45
46     return $this->create('user.create');
47
48     // if (false) {
49         return $this->create('user.create');
50     }
51
52     return $this->create('user.create');
53
54     // if (false) {
55         return $this->create('user.create');
56     }
57
58     return $this->create('user.create');
59
60     // if (false) {
61         return $this->create('user.create');
62     }
63
64     return $this->create('user.create');
65
66     // if (false) {
67         return $this->create('user.create');
68     }
69
70     return $this->create('user.create');
71
72     // if (false) {
73         return $this->create('user.create');
74     }
75
76     return $this->create('user.create');
77
78     // if (false) {
79         return $this->create('user.create');
80     }
81
82     return $this->create('user.create');
83
84     // if (false) {
85         return $this->create('user.create');
86     }
87
88     return $this->create('user.create');
89
90     // if (false) {
91         return $this->create('user.create');
92     }
93
94     return $this->create('user.create');
95
96     // if (false) {
97         return $this->create('user.create');
98     }
99
100    return $this->create('user.create');
101
102    // if (false) {
103        return $this->create('user.create');
104    }
105
106    return $this->create('user.create');
107
108    // if (false) {
109        return $this->create('user.create');
110    }
111
112    return $this->create('user.create');
113
114    // if (false) {
115        return $this->create('user.create');
116    }
117
118    return $this->create('user.create');
119
120    // if (false) {
121        return $this->create('user.create');
122    }
123
124    return $this->create('user.create');
125
126    // if (false) {
127        return $this->create('user.create');
128    }
129
130    return $this->create('user.create');
131
132    // if (false) {
133        return $this->create('user.create');
134    }
135
136    return $this->create('user.create');
137
138    // if (false) {
139        return $this->create('user.create');
140    }
141
142    return $this->create('user.create');
143
144    // if (false) {
145        return $this->create('user.create');
146    }
147
148    return $this->create('user.create');
149
150    // if (false) {
151        return $this->create('user.create');
152    }
153
154    return $this->create('user.create');
155
156    // if (false) {
157        return $this->create('user.create');
158    }
159
160    return $this->create('user.create');
161
162    // if (false) {
163        return $this->create('user.create');
164    }
165
166    return $this->create('user.create');
167
168    // if (false) {
169        return $this->create('user.create');
170    }
171
172    return $this->create('user.create');
173
174    // if (false) {
175        return $this->create('user.create');
176    }
177
178    return $this->create('user.create');
179
180    // if (false) {
181        return $this->create('user.create');
182    }
183
184    return $this->create('user.create');
185
186    // if (false) {
187        return $this->create('user.create');
188    }
189
190    return $this->create('user.create');
191
192    // if (false) {
193        return $this->create('user.create');
194    }
195
196    return $this->create('user.create');
197
198    // if (false) {
199        return $this->create('user.create');
200    }
201
202    return $this->create('user.create');
203
204    // if (false) {
205        return $this->create('user.create');
206    }
207
208    return $this->create('user.create');
209
210    // if (false) {
211        return $this->create('user.create');
212    }
213
214    return $this->create('user.create');
215
216    // if (false) {
217        return $this->create('user.create');
218    }
219
220    return $this->create('user.create');
221
222    // if (false) {
223        return $this->create('user.create');
224    }
225
226    return $this->create('user.create');
227
228    // if (false) {
229        return $this->create('user.create');
230    }
231
232    return $this->create('user.create');
233
234    // if (false) {
235        return $this->create('user.create');
236    }
237
238    return $this->create('user.create');
239
240    // if (false) {
241        return $this->create('user.create');
242    }
243
244    return $this->create('user.create');
245
246    // if (false) {
247        return $this->create('user.create');
248    }
249
250    return $this->create('user.create');
251
252    // if (false) {
253        return $this->create('user.create');
254    }
255
256    return $this->create('user.create');
257
258    // if (false) {
259        return $this->create('user.create');
260    }
261
262    return $this->create('user.create');
263
264    // if (false) {
265        return $this->create('user.create');
266    }
267
268    return $this->create('user.create');
269
270    // if (false) {
271        return $this->create('user.create');
272    }
273
274    return $this->create('user.create');
275
276    // if (false) {
277        return $this->create('user.create');
278    }
279
280    return $this->create('user.create');
281
282    // if (false) {
283        return $this->create('user.create');
284    }
285
286    return $this->create('user.create');
287
288    // if (false) {
289        return $this->create('user.create');
290    }
291
292    return $this->create('user.create');
293
294    // if (false) {
295        return $this->create('user.create');
296    }
297
298    return $this->create('user.create');
299
300    // if (false) {
301        return $this->create('user.create');
302    }
303
304    return $this->create('user.create');
305
306    // if (false) {
307        return $this->create('user.create');
308    }
309
310    return $this->create('user.create');
311
312    // if (false) {
313        return $this->create('user.create');
314    }
315
316    return $this->create('user.create');
317
318    // if (false) {
319        return $this->create('user.create');
320    }
321
322    return $this->create('user.create');
323
324    // if (false) {
325        return $this->create('user.create');
326    }
327
328    return $this->create('user.create');
329
330    // if (false) {
331        return $this->create('user.create');
332    }
333
334    return $this->create('user.create');
335
336    // if (false) {
337        return $this->create('user.create');
338    }
339
340    return $this->create('user.create');
341
342    // if (false) {
343        return $this->create('user.create');
344    }
345
346    return $this->create('user.create');
347
348    // if (false) {
349        return $this->create('user.create');
350    }
351
352    return $this->create('user.create');
353
354    // if (false) {
355        return $this->create('user.create');
356    }
357
358    return $this->create('user.create');
359
360    // if (false) {
361        return $this->create('user.create');
362    }
363
364    return $this->create('user.create');
365
366    // if (false) {
367        return $this->create('user.create');
368    }
369
370    return $this->create('user.create');
371
372    // if (false) {
373        return $this->create('user.create');
374    }
375
376    return $this->create('user.create');
377
378    // if (false) {
379        return $this->create('user.create');
380    }
381
382    return $this->create('user.create');
383
384    // if (false) {
385        return $this->create('user.create');
386    }
387
388    return $this->create('user.create');
389
390    // if (false) {
391        return $this->create('user.create');
392    }
393
394    return $this->create('user.create');
395
396    // if (false) {
397        return $this->create('user.create');
398    }
399
400    return $this->create('user.create');
401
402    // if (false) {
403        return $this->create('user.create');
404    }
405
406    return $this->create('user.create');
407
408    // if (false) {
409        return $this->create('user.create');
410    }
411
412    return $this->create('user.create');
413
414    // if (false) {
415        return $this->create('user.create');
416    }
417
418    return $this->create('user.create');
419
420    // if (false) {
421        return $this->create('user.create');
422    }
423
424    return $this->create('user.create');
425
426    // if (false) {
427        return $this->create('user.create');
428    }
429
430    return $this->create('user.create');
431
432    // if (false) {
433        return $this->create('user.create');
434    }
435
436    return $this->create('user.create');
437
438    // if (false) {
439        return $this->create('user.create');
440    }
441
4
```

UkmController ini mengelola seluruh proses CRUD untuk data UKM yang dibuat oleh admin masing-masing UKM. Fungsi index() menampilkan data UKM milik admin yang sedang login. create() dan store() digunakan untuk menampilkan form dan menyimpan data UKM baru, dengan validasi dari StoreukmRequest dan proses upload logo secara manual. edit() dan update() memungkinkan admin untuk mengedit data UKM mereka, termasuk memperbarui logo. Semua proses simpan dan update dilakukan dalam transaksi database (DB::transaction) untuk menjaga konsistensi data. Fungsi destroy() digunakan untuk menghapus UKM dari database dengan aman.

```
1 Route::middleware(['auth', 'can:manage ukm'])->group(function () {
2     Route::resource('ukm', UkmController::class);
3     Route::resource('abouts', AboutOrganisasiController::class);
4     Route::resource('kegiatanukm', KegiatanUKMController::class);
5 });
```

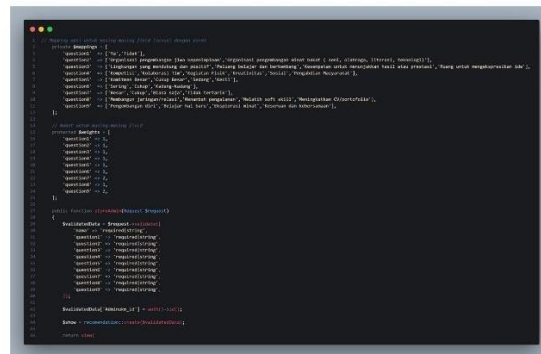
Gambar 5. Route UKM

Kode route tersebut menetapkan sekelompok route yang hanya bisa diakses oleh pengguna yang sudah login (auth) dan memiliki izin manage ukm. Di dalam grup ini, terdapat tiga resource controller: UkmController untuk mengelola data UKM. AboutOrganisasiController untuk mengatur informasi "Tentang Organisasi". KegiatanUKMController untuk mengelola kegiatan atau event dari UKM. Karena menggunakan Route::resource, otomatis akan dibuat route standar untuk fungsi CRUD: index, create, store, show, edit, update, dan destroy untuk masing-masing controller.

```
1 public function index()
2 {
3     return view('ukm.recommendation');
4 }
5
6 public function show()
7 {
8     $user = Auth::user();
9
10    $show = recommendation::with('Adminukm')->where('Adminukm_id', $user->id)->first();
11    if (!$show) {
12        return redirect()->route('ukm.recommendation');
13    }
14    return view('ukm.show_recommendation', compact('show'));
15 }
```

Gambar 5. Controller Rekomendasi

Bagian function index() digunakan untuk menampilkan halaman dari 'ukm.recommendation' lalu pada function show() membuat variabel bernama user untuk memvalidasi login sebagai user yang aktif, lalu \$show memuat model 'recommendation' dengan memilih berdasarkan Adminukm_id yang sedang login, jika data yang berasal dari user yang sedang login kosong maka akan dikembalikan pada halaman 'ukm.recommendation' untuk mengisi data terlebih dahulu, namun jika data sudah ada maka data tersebut akan ditampilkan pada halaman.



Gambar 6. Controller Rekomendasi

Function `storeAdmin()` menerima input dari form rekomendasi yang diisi oleh admin UKM. Isian tersebut terdiri dari 9 pertanyaan (`question1` hingga `question9`) ditambah satu kolom nama. Data ini kemudian divalidasi agar tidak kosong dan harus berupa string. Setelah validasi berhasil, sistem menambahkan `Adminukm_id` (ID admin yang sedang login) ke dalam data, lalu menyimpan seluruh data ke tabel `recomendations` menggunakan `recommendation::create()`. Properti `$mappings` berisi daftar semua opsi jawaban untuk setiap pertanyaan. Ini digunakan nanti saat membentuk vektor one-hot untuk menghitung kemiripan antara jawaban user dan jawaban admin. Properti `Sweights` menetapkan bobot untuk setiap pertanyaan, agar pertanyaan yang dianggap lebih penting (seperti `question7` dan `question9`) memberi pengaruh lebih besar dalam perhitungan kemiripan (cosine similarity). Setelah data berhasil disimpan, controller mengarahkan admin ke halaman `ukm.show_recomendation` untuk melihat kembali data jawaban rekomendasinya.

```

1 public function match($request)
2 {
3     $validatedData = $request->validated();
4     'question1' => 'required|string',
5     'question2' => 'required|string',
6     'question3' => 'required|string',
7     'question4' => 'required|string',
8     'question5' => 'required|string',
9     'question6' => 'required|string',
10    'question7' => 'required|string',
11    'question8' => 'required|string',
12    'question9' => 'required|string',
13 }
14
15 $userVector = $this->createOneHotVector($validatedData);
16
17 $admins = $recommendation->all();
18 $recommendations = [];
19
20 foreach ($admins as $admin) {
21     $adminData = [
22         'question1' => $admin->question1,
23         'question2' => $admin->question2,
24         'question3' => $admin->question3,
25         'question4' => $admin->question4,
26         'question5' => $admin->question5,
27         'question6' => $admin->question6,
28         'question7' => $admin->question7,
29         'question8' => $admin->question8,
30         'question9' => $admin->question9,
31     ];
32
33     $adminVector = $this->createOneHotVector($adminData);
34
35     $similarity = $this->cosineSimilarity($userVector, $adminVector);
36
37     $recommendations[] = [
38         'admin_id' => $admin->id,
39         'name' => $admin->name,
40         'score' => round($similarity, 4)
41     ];
42 }
43
44 usort($recommendations, fn($a, $b) => $a['score'] <=> $b['score']);
45
46 return response()->json([
47     'message' => 'Rekomendasi berdasarkan cosine similarity',
48     'recommendations' => $recommendations
49 ]);
50
51 private function createOneHotVector(array $data)
52 {
53     $vector = [];
54
55     foreach ($this->mappings as $field => $options) {
56         $index = array_search($data[$field], $options, true);
57         $answer = $data[$field] ?? null;
58         $index = array_search($answer, $options);
59
60         if ($index !== false) {
61             $weight = $this->weights[$field] ?? 1;
62             $oneHot[$index] = 1 * $weight;
63         }
64
65         $vector = array_merge($vector, $oneHot);
66     }
67 }

```

Gambar 7. Controller Rekomendasi

Fungsi `match()` menerima input dari pengguna berupa jawaban terhadap sembilan pertanyaan (`question1`–`question9`) yang telah divalidasi. Jawaban ini dikonversi menjadi one-hot vector melalui fungsi `createOneHotVector()`, yang membentuk representasi vektor numerik dari setiap jawaban berdasarkan opsi yang telah ditentukan dalam properti `$mappings`, serta bobot pertanyaan yang ditentukan dalam `$weights`. Setelah itu, sistem mengambil seluruh data rekomendasi admin UKM dari database, mengubah masing-masing jawaban admin menjadi one-hot vector, dan menghitung tingkat kemiripan antara vektor pengguna dan vektor admin menggunakan metode Cosine Similarity. Hasil kemiripan ini disimpan dalam array rekomendasi berisi ID admin, nama, dan skor kemiripan. Seluruh rekomendasi kemudian diurutkan dari skor tertinggi ke terendah, dan hasil akhir dikirimkan dalam bentuk respons JSON yang berisi daftar rekomendasi UKM paling sesuai berdasarkan kemiripan preferensi.

```

1 public function match(request $request)
2 {
3     $validatedData = $request->validate([
4         'question1' => 'required|string',
5         'question2' => 'required|string',
6         'question3' => 'required|string',
7         'question4' => 'required|string',
8         'question5' => 'required|string',
9         'question6' => 'required|string',
10        'question7' => 'required|string',
11        'question8' => 'required|string',
12        'question9' => 'required|string',
13    ]);
14
15    $userVector = $this->createOneHotVector($validatedData);
16
17    $admins = recommendation::all();
18    $recommendations = [];
19
20    foreach ($admins as $admin) {
21        $adminData = [
22            'question1' => $admin->question1,
23            'question2' => $admin->question2,
24            'question3' => $admin->question3,
25            'question4' => $admin->question4,
26            'question5' => $admin->question5,
27            'question6' => $admin->question6,
28            'question7' => $admin->question7,
29            'question8' => $admin->question8,
30            'question9' => $admin->question9,
31        ];
32
33        $adminVector = $this->createOneHotVector($adminData);
34
35        $similarity = $this->cosineSimilarity($userVector, $adminVector);
36
37        $recommendations[] = [
38            'admin_id' => $admin->id,
39            'name' => $admin->name,
40            'score' => round($similarity, 4)
41        ];
42    }
43
44    usort($recommendations, function($a, $b) { return $b['score'] <=> $a['score']; });
45
46    return response()->json([
47        'message' => 'Rekomendasi berdasarkan cosine similarity',
48        'recommendations' => $recommendations
49    ]);
50 }
51
52 private function createOneHotVector(array $data)
53 {
54     $vector = [];
55
56     foreach ($this->mappings as $field => $options) {
57         $vector = array_fill(0, count($options), 0);
58         $answer = $data[$field] ?? null;
59         $index = array_search($answer, $options);
60
61         if ($index !== false) {
62             $weight = $this->weights[$field] ?? 1;
63             $vector[$index] = 1 * $weight;
64         }
65
66         $vector = array_merge($vector, $vector);
67     }
68 }

```

Gambar 8. Controller Rekomendasi

Fungsi `match()` menerima input dari pengguna berupa jawaban terhadap sembilan pertanyaan (`question1`–`question9`) yang telah divalidasi. Jawaban ini dikonversi menjadi one-hot vector melalui fungsi `createOneHotVector()`, yang membentuk representasi vektor numerik dari setiap jawaban berdasarkan opsi yang telah ditentukan dalam properti `$mappings`, serta bobot pertanyaan yang ditentukan dalam `$weights`. Setelah itu, sistem mengambil seluruh data rekomendasi admin UKM dari database, mengubah masing-masing jawaban admin menjadi one-hot vector, dan menghitung tingkat kemiripan antara vektor pengguna dan vektor admin menggunakan metode Cosine Similarity. Hasil kemiripan ini disimpan dalam array rekomendasi berisi ID admin, nama, dan skor kemiripan. Seluruh rekomendasi kemudian diurutkan dari skor tertinggi ke terendah, dan hasil akhir dikirimkan dalam bentuk respons JSON yang berisi daftar rekomendasi UKM paling sesuai berdasarkan kemiripan preferensi.

```

1 public function match($request $request)
2 {
3     $validatedData = $request->validate([
4         'question1' => 'required|string',
5         'question2' => 'required|string',
6         'question3' => 'required|string',
7         'question4' => 'required|string',
8         'question5' => 'required|string',
9         'question6' => 'required|string',
10        'question7' => 'required|string',
11        'question8' => 'required|string',
12        'question9' => 'required|string',
13    ]);
14
15    $userVector = $this->createOneHotVector($validatedData);
16
17    $admins = recommendation::all();
18    $recommendations = [];
19
20    foreach ($admins as $admin) {
21        $adminData = [
22            'question1' => $admin->question1,
23            'question2' => $admin->question2,
24            'question3' => $admin->question3,
25            'question4' => $admin->question4,
26            'question5' => $admin->question5,
27            'question6' => $admin->question6,
28            'question7' => $admin->question7,
29            'question8' => $admin->question8,
30            'question9' => $admin->question9,
31        ];
32
33        $adminVector = $this->createOneHotVector($adminData);
34
35        $similarity = $this->cosineSimilarity($userVector, $adminVector);
36
37        $recommendations[] = [
38            'admin_id' => $admin->id,
39            'nama' => $admin->nama,
40            'score' => round($similarity, 4)
41        ];
42    }
43
44    usort($recommendations, function($a, $b) { return $b['score'] <=> $a['score']; });
45
46    return response()->json([
47        'message' => 'Rekomendasi berdasarkan cosine similarity',
48        'recommendations' => $recommendations
49    ]);
50 }
51
52 private function createOneHotVector(array $data)
53 {
54     $vector = [];
55
56     foreach ($this->mappings as $field => $options) {
57         $onehot = array_fill(0, count($options), 0);
58         $answer = $data[$field] ?? null;
59         $index = array_search($answer, $options);
60
61         if ($index !== false) {
62             $weight = $this->weights[$field] ?? 1;
63             $onehot[$index] = 1 + $weight;
64         }
65
66         $vector = array_merge($vector, $onehot);
67     }
68 }

```

Gambar 9. Controller Rekomendasi

Fungsi `match()` menerima input dari pengguna berupa jawaban terhadap sembilan pertanyaan (`question1`–`question9`) yang telah divalidasi. Jawaban ini dikonversi menjadi one-hot vector melalui fungsi `createOneHotVector()`, yang membentuk representasi vektor numerik dari setiap jawaban berdasarkan opsi yang telah ditentukan dalam properti `$mappings`, serta bobot pertanyaan yang ditentukan dalam `$weights`. Setelah itu, sistem mengambil seluruh data rekomendasi admin UKM dari database, mengubah masing-masing jawaban admin menjadi one-hot vector, dan menghitung tingkat kemiripan antara vektor pengguna dan vektor admin menggunakan metode Cosine Similarity. Hasil kemiripan ini disimpan dalam array rekomendasi berisi ID admin, nama, dan skor kemiripan. Seluruh rekomendasi kemudian diurutkan dari skor tertinggi ke terendah, dan hasil akhir dikirimkan dalam bentuk respons JSON yang berisi daftar rekomendasi UKM paling sesuai berdasarkan kemiripan preferensi.

```
1 private function createOneHotVector(array $data)
2 {
3     $vector = [];
4
5     foreach ($this->mappings as $field => $options) {
6         $oneHot = array_fill(0, count($options), 0);
7         $answer = $data[$field] ?? null;
8         $index = array_search($answer, $options);
9
10        if ($index !== false) {
11            $weight = $this->weights[$field] ?? 1;
12            $oneHot[$index] = 1 * $weight;
13        }
14
15        $vector = array_merge($vector, $oneHot);
16    }
17
18    return $vector;
19 }
20
```

Gambar 10. Controller Rekomendasi

Fungsi `createOneHotVector()` digunakan untuk mengubah data input (jawaban pengguna atau admin UKM) menjadi sebuah vektor numerik (one-hot vector) yang siap diproses dalam perhitungan kemiripan (cosine similarity).

Fungsi ini bekerja dengan cara membaca setiap field (seperti `question1`, `question2`, dst.) dan mencocokkan nilai jawaban terhadap daftar pilihan (`$options`) yang sudah ditentukan dalam properti `$mappings`. Untuk setiap pilihan jawaban, dibuat array dengan nilai 0 (`array_fill`), lalu jika jawaban ditemukan dalam opsi, maka nilai pada posisi indeks jawaban tersebut diubah menjadi 1, dikalikan dengan bobot (`$weights`) yang sudah ditentukan untuk pertanyaan itu. Setelah itu, setiap hasil one-hot vector digabung ke dalam satu array panjang bernama `$vector`. Hasil akhir dari fungsi ini adalah vektor representasi numerik dari seluruh jawaban, yang akan digunakan dalam algoritma cosine similarity untuk membandingkan kemiripan antara pengguna dan admin UKM.

SOURCE CODE

14

```
1 //Perhitungan rekomendasinya
2 private function cosineSimilarity(array $vec1, array $vec2)
3 {
4     $dotProduct = 0;
5     $normVec1 = 0;
6     $normVec2 = 0;
7
8     foreach ($vec1 as $i => $value) {
9         $dotProduct += $value * $vec2[$i];
10        $normVec1 += $value * $value;
11        $normVec2 += $vec2[$i] * $vec2[$i];
12    }
13
14    if ($normVec1 == 0 || $normVec2 == 0) {
15        return 0;
16    }
17
18    return $dotProduct / (sqrt($normVec1) * sqrt($normVec2));
19 }
20 }
```

Gambar 11. Controller Rekomendasi

tampilan ini merupakan perhitungan proses rekomendasinya. Fungsi `cosineSimilarity()` digunakan untuk menghitung tingkat kemiripan antara dua vektor — yaitu vektor dari jawaban pengguna dan vektor dari jawaban admin UKM — dengan menggunakan rumus cosine similarity. Fungsi ini bekerja dengan menjumlahkan hasil perkalian elemen yang bersesuaian dari kedua vektor (`$dotProduct`), lalu menghitung panjang (norma) dari masing-masing vektor (`$normVec1` dan `$normVec2`) menggunakan rumus akar kuadrat jumlah kuadrat elemen-elemen vektor. Jika salah satu norma bernilai nol (yang berarti vektor kosong atau tidak bernilai), maka hasilnya akan dikembalikan sebagai 0 untuk menghindari pembagian nol. Jika keduanya valid, maka skor cosine similarity dihitung dengan membagi nilai dot product dengan hasil kali kedua norma tersebut. Nilai ini menunjukkan tingkat kemiripan dalam skala 0 sampai 1, di mana nilai mendekati 1 berarti sangat mirip.

```

1 route::post('/get-recommendations', [RecommendationController::class, 'match']);
2 route::get('/show-recommendation', [RecommendationController::class, 'show'])->name('list-recommendation');
3
4 route::get('/recommendation', [RecommendationController::class, 'index'])->name('show-recommendation');
5 route::get('/recommendation-ormawa', [RecommendationController::class, 'ormawa'])->name('ormawa-recommendation');
6 route::post('/admin/store', [RecommendationController::class, 'storeAdmin']);
7 route::get('/show-recmawa', [RecommendationController::class, 'show'])->name('list-recmawa');
8
9 route::post('/add-rekomawa', [RecommendationController::class, 'add']);
10 route::post('/get-rekomawa', [RecommendationController::class, 'match']);

```

Gambar 12. Controller Rekomendasi

Kumpulan route tersebut mengatur alur sistem rekomendasi UKM, di mana route `/recommendation` dan `/recommendation-ormawa` menampilkan halaman form untuk admin dan pengguna umum, `/admin/store` menyimpan data jawaban dari admin UKM, `/show-recmawa` menampilkan hasil rekomendasi berdasarkan jawaban pengguna, sedangkan `/add-rekomawa` menerima input pengguna dan `/get-rekomawa` memproses pencocokan jawaban menggunakan algoritma seperti cosine similarity untuk menghasilkan rekomendasi UKM yang paling sesuai.

```

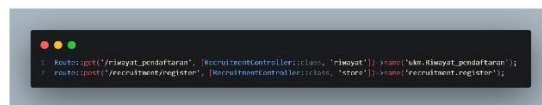
1 class RecruitmentController extends Controller
2 {
3     public function index()
4     {
5         $user = Auth::user();
6
7         $recruitment = Recruitment::where('AdminId', $user->id->first());
8         if ($recruitment) {
9             return redirect()->route('show-recmawa');
10        }
11        return view('show-recmawa', compact('recruitment'));
12    }
13
14    public function index_ormawa()
15    {
16        $user = Auth::user();
17
18        $recruitment = Recruitment::where('AdminId', $user->id->first());
19        if ($recruitment) {
20            return redirect()->route('show-recmawa');
21        }
22        return view('show-recmawa', compact('recruitment'));
23    }
24
25    public function show()
26    {
27        $user = Auth::user(); // Admin user yang sedang login
28
29        // Ambil data recruitment nama untuk user yang sedang login
30        $recruitment = Recruitment::where('AdminId', $user->id->first());
31        return view('show-recmawa', compact('recruitment'));
32    }
33
34    public function store(Request $request)
35    {
36        $validated = $request->validate([
37            'name' => 'required|string',
38            'nim' => 'required|string',
39            'email' => 'required|string',
40            'phone' => 'required|string',
41            'study_program' => 'required|string',
42            'semester' => 'required|integer',
43            'gender' => 'required|in:Male,Female',
44            'password' => 'required|string',
45            'photo' => 'nullable|string',
46            'AdminId' => 'required|integer|exists:users,AdminId', // Admin user yang dipilih user
47        ]);
48
49        // Ambil AdminId user yang sedang login
50        $user = Auth::user();
51        $recruitment = Recruitment::create([
52            'name' => $validated['name'],
53            'nim' => $validated['nim'],
54            'email' => $validated['email'],
55            'phone' => $validated['phone'],
56            'study_program' => $validated['study_program'],
57            'semester' => $validated['semester'],
58            'gender' => $validated['gender'],
59            'password' => $validated['password'],
60            'photo' => $validated['photo'],
61            'AdminId' => $user->id,
62        ]);
63
64        // Tampilkan recruitment ke dalam data yang akan dikirimkan
65        $validated['AdminId'] = $user->id;
66        $recruitment = Recruitment::create($validated);
67        return view('show-recmawa');
68
69        return redirect()->route('show-recmawa')->with('success', 'Berhasil mendaftar.');
```

Gambar 13. Controller Recruitment

SOURCE CODE

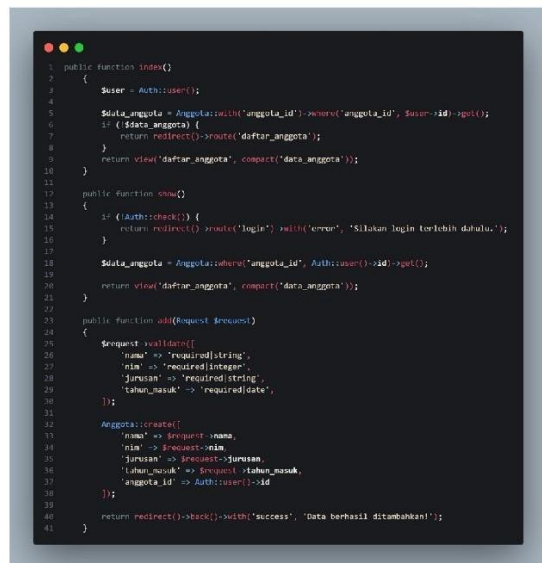
16

Controller RecruitmentController ini mengelola data pendaftaran ke UKM dan Ormawa, di mana method index() dan index_ormawa() menampilkan halaman riwayat pendaftaran masing-masing berdasarkan admin yang login dan hanya jika datanya ditemukan, riwayat() mengambil seluruh data pendaftar milik admin UKM yang sedang login, sedangkan method store() memvalidasi inputan form pendaftaran, mengecek keberadaan UKM berdasarkan Adminukm_id, lalu menyimpan data mahasiswa ke dalam tabel recruitment jika valid.



Gambar 14. Route Recruitment

Route ini digunakan untuk mengakses recruitmentcontroller



Gambar 15. Route Recruitment

```

1 public function update(Request $request)
2 {
3     $request->validate([
4         'id' => 'required|exists:daftar_anggota,id',
5         'nama' => 'required|string',
6         'nim' => 'required|integer',
7         'jurusan' => 'required|string',
8         'tahun_masuk' => 'required|date',
9     ]);
10
11     $data_anggota = Anggota::find($request->id);
12     $data_anggota->nama = $request->nama;
13     $data_anggota->nim = $request->nim;
14     $data_anggota->jurusan = $request->jurusan;
15     $data_anggota->tahun_masuk = $request->tahun_masuk;
16     $data_anggota->anggota_id = Auth::user()->id;
17     $data_anggota->save();
18
19     return redirect()->back()->with('success', 'Data berhasil diubah');
20 }
21
22 public function delete(Request $request)
23 {
24     $request->validate([
25         'id' => 'required|exists:daftar_anggota,id',
26     ]);
27
28     $data_anggota = Anggota::find($request->id);
29
30     // Validasi agar hanya pemilik data yang bisa menghapus
31     if ($data_anggota && $data_anggota->anggota_id == Auth::user()->id) {
32         $data_anggota->delete();
33         return redirect()->back()->with('success', 'Data berhasil dihapus');
34     }
35
36     return redirect()->back()->with('error', 'Gagal menghapus data');
37 }
38
39 public function dashboard()
40 {
41     $anggota = DB::table('daftar_anggota')
42         ->join('users', 'daftar_anggota.anggota_id', '=', 'users.id')
43         ->select('users.name as nama_anggota', DB::raw('count(*) as total'))
44         ->groupBy('users.name')
45         ->get();
46
47     $labels = $anggota->pluck('nama_anggota');
48     $data = $anggota->pluck('total');
49
50     return view('dashboard', compact('labels', 'data'));
51 }

```

Gambar 16. Route Anggota

Controller `AnggotaController` ini menangani manajemen data anggota UKM/organisasi oleh pengguna yang login, di mana method `index()` dan `show()` menampilkan data anggota berdasarkan `anggota_id` dari user yang sedang login, method `add()` digunakan untuk menambahkan anggota baru dengan validasi input terlebih dahulu, method `update()` memperbarui data anggota berdasarkan ID dan memastikan data hanya bisa diubah oleh pemiliknya, method `delete()` menghapus data anggota dengan validasi kepemilikan data, dan method `dashboard()` mengambil data agregat jumlah anggota per user untuk ditampilkan dalam bentuk visualisasi pada dashboard

```

1 Route::get('/daftar-anggota', [AnggotaController::class, 'show'])->name('daftar_anggota');
2 Route::post('/tambah-anggota', [AnggotaController::class, 'add'])->name('tambah_anggota');
3 Route::put('/edit-anggota', [AnggotaController::class, 'update'])->name('edit_anggota');
4 Route::delete('/hapus-anggota', [AnggotaController::class, 'delete'])->name('hapus_anggota');

```

Gambar 17. Route anggota

SOURCE CODE

18

```

1 public function index()
2 {
3     $user = Auth::user();
4
5     $abouts = About::where('user_id' => $user->id, $user->id)->first();
6     if (!$abouts) {
7         return redirect()->route('abouts.create');
8     }
9     return view('abouts.index', compact('abouts'));
10 }
11
12 public function create()
13 {
14     return view('abouts.create');
15 }
16
17 // Simulasi data about
18 public function store(Request $request)
19 {
20     $request->validate([
21         'nama_organisasi' => 'required|string|max:255',
22         'logo' => 'required|image|mimes:jpeg,png,jpg|max:2048',
23         'slogan' => 'required|string|max:255',
24         'tentang_kami' => 'required|string',
25         'email' => 'required|string',
26         'nomor_telepon' => 'required|string',
27         'lokal' => 'required|string|max:255',
28     ]);
29
30     $data = $request->except([
31         'nama_organisasi',
32         'logo',
33         'tentang_kami',
34         'email',
35         'nomor_telepon',
36         'lokal'
37     ]);
38
39     // Tambahkan user yang login
40     $data['user_id'] = Auth::id();
41
42     // Simulasi gambar secara manual jika ada
43     if ($request->hasFile('logo')) {
44         $file = $request->file('logo');
45         $filename = time() . '-' . $file->getClientOriginalName();
46         $destination = public_path('storage/logo');
47
48         // Buat folder jika belum ada
49         if (!file_exists($destination)) {
50             mkdir($destination, 0755, true);
51         }
52
53         $file->move($destination, $filename);
54
55         // Simulasi path manual ke database
56         $data['logo'] = 'logo/' . $filename;
57     }
58
59     About::create($data);
60
61     return redirect()->route('abouts.index')->with('success', 'Berita berhasil ditambahkan.');
```

Gambar 18. Controller About

Method dalam controller ini mengatur data profil organisasi, dimulai dari index() yang menampilkan data about berdasarkan user yang sedang login (jika belum ada, diarahkan ke halaman create), lalu create() menampilkan form input, dan store() menyimpan data baru dengan validasi input, termasuk proses unggah logo yang disimpan ke folder storage/logo secara manual dan disimpan path-nya di database, serta menambahkan ID user yang sedang login sebagai pemilik data.

SOURCE CODE

19

```
1 public function edit($id)
2 {
3     $abouts = about::findOrFail($id);
4     return view('abouts.edit', compact('abouts'));
5 }
6
7 public function update(Request $request, $id)
8 {
9     $request->validate([
10         'nama_organisasi' => 'required|string|max:255',
11         'logo' => 'nullable|image|mimes:png,jpg|max:2048',
12         'slugen' => 'required|string|max:255',
13         'tentang_kami' => 'required|string',
14         'email' => 'required|string',
15         'nomor_telepon' => 'required|string',
16         'lokasi' => 'required|string|max:255'
17     ]);
18
19     $abouts = about::findOrFail($id);
20
21     $data = $request->only([
22         'nama_organisasi',
23         'logo',
24         'slugen',
25         'tentang_kami',
26         'email',
27         'nomor_telepon',
28         'lokasi',
29     ]);
30
31     // Jika gambar baru di upload
32     if ($request->hasFile('logo')) {
33         $file = $request->file('logo');
34         $filename = time() . '-' . $file->getClientOriginalName();
35         $destination = public_path('storage/logo');
36
37         if (!file_exists($destination)) {
38             mkdir($destination, 0755, true);
39         }
40
41         $file->move($destination, $filename);
42         $data['logo'] = 'logo/' . $filename;
43     }
44
45     $abouts->update($data);
46
47     return redirect()->route('abouts.index')->with('success', 'Berita berhasil diperbarui.');
```

Gambar 19. Controller About

Fungsi edit() dan update() di controller ini digunakan untuk mengelola proses pembaruan data profil organisasi. Method edit() mengambil data berdasarkan ID dan menampilkannya ke dalam form edit, sementara update() memproses data yang dikirim dari form tersebut. Di dalam update(), input divalidasi terlebih dahulu, lalu jika ada gambar logo baru diunggah, sistem menyimpannya ke folder storage/logo dan memperbarui path-nya di database. Setelah semua data disiapkan, sistem akan mengupdate data about yang bersangkutan dan mengarahkan pengguna kembali ke halaman index dengan pesan sukses.

SOURCE CODE

20

```

1 public function index()
2 {
3     $user = Auth::user();
4     $fotos = FotoKegiatan::where('user_id', $user->id) ->get();
5     return view('kegiatan.index', compact('fotos'));
6 }
7
8 public function create()
9 {
10    return view('kegiatan.create');
11 }
12
13 // Simpan foto kegiatan
14 public function store(Request $request)
15 {
16     $request->validate([
17         'foto_kegiatan' => 'required|image|mimes:jpg,jpg,log|max:1048',
18     ]);
19
20     $data = [];
21
22     // Tambahkan user_id
23     $data['user_id'] = Auth::id();
24
25     // Simpan file ke folder publik
26     if ($request->hasFile('foto_kegiatan')) {
27         $file = $request->file('foto_kegiatan');
28         $filename = time() . '-' . $file->getClientOriginalName();
29         $destination = public_path('storage/foto_kegiatan');
30         if (!file_exists($destination)) {
31             mkdir($destination, 0755, true);
32         }
33         $file->move($destination, $filename);
34         $data['foto_kegiatan'] = 'foto_kegiatan/' . $filename;
35     }
36
37     FotoKegiatan::create($data);
38
39     return redirect()->route('kegiatan.index')->with('success', 'Foto kegiatan berhasil ditambahkan.');
```

Gambar 20. Controller Kegiatan

Controller KegiatanUKMController bertanggung jawab untuk mengelola data foto kegiatan UKM, dimulai dari menampilkan seluruh foto milik user yang sedang login (index()), menampilkan form unggah foto (create()), menyimpan foto baru ke direktori publik (store()), hingga menghapus foto dari direktori dan database (destroy()). Pada saat menyimpan, foto akan divalidasi terlebih dahulu, lalu disimpan ke dalam folder storage/foto_kegiatan, dan path-nya dicatat dalam database bersama dengan ID user. Fungsi destroy() memastikan file juga dihapus dari sistem file sebelum menghapus record-nya di database.

Lampiran 1. 6 Sertifikat HKI

 REPUBLIK 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	: EC002025078571, 30 Juni 2025
Pencipta	
Nama	: Nabilla Auly Zahra, Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom. dkk
Alamat	: Jalan Perintis kemerdekaan gg 27 no 10 rt 09 rw 06 kelurahan panggung, Tegal Timur, Kota Tegal, Jawa Tengah, 52122
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	: Jalan Mataram No. 9, Pesunungan Lor, Kecamatan Margadana 52142, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Sistem Rekomendasi ORMAWA pada Poltek Harber menggunakan metode Content Based Filtering dan algoritma Cosine Similarity
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 30 Juni 2025, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	: 000918832
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 Damarasongko, 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	Nabilla Auly Zahra	Jalan Perintis kemerdekaan gg 27 no 10 rt 09 rw 06 kelurahan panggung Tegal Timur, Kota Tegal
2	Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.	Jln. Raya Kluwut Timur No. 24, Kecamatan Bulakamba, Kabupaten Brebes.
3	Dyah Apriliani, S.T., M.Kom.	Perumahan Mutiara Indah Slawi, Jalan Nangka No.2, Procot, Procot Blok C4 SLAWI SLAWI JAWA TENGAH Slawi, Kab. Tegal



Lampiran 1. 7 Lembar Bimbingan



**D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

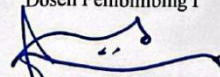
LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Nabilla Auly Zahra
NIM : 21090094
No. Ponsel : 0895357973905
Judul TA : SISTEM REKOMENDASI ORMAWA PADA
POLTEK HARBER MENGGUNAKAN
CONTENT BASED FILTERING DAN
ALGORITMA COSINE SIMILARITY
Dosen Pembimbing I : Ir. Ginanjar Wiro Sasmito, M.Kom. IPM.,
ASEAN Eng

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	23/25 1	Uraian	Ok.	b
2.	26/25 5	pro du	lanjut	b
3.	18/25 6	pro du	Ok.	b
4.	26/25 6	Laporan	perisi	b
5.	07/25 07	Laporan	perisi	b

6.	08/07 ²⁵	Laporan	revisi	16
7.	09/07 ²⁵	Laporan	revisi	16
8.	10/07 ²⁵	Laporan	OK	16
	7	Acc	Siag	16

Tegal,
Dosen Pembimbing I



Ir. Gijanmar Wiro Sasmito,
M.Kom. IPM., ASEAN Eng
NIPY. 10.007.032



**D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Nabilla Auly Zahra
NIM : 21090094
No. Ponsel : 0895357973905
Judul TA : SISTEM REKOMENDASI ORMAWA PADA
POLTEK HARBER MENGGUNAKAN
CONTENT BASED FILTERING DAN
ALGORITMA COSINE SIMILARITY
Dosen Pembimbing I : Dyah Apriliani, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	16/04 ²⁵	Aplikasi	Revisi	
2.	21/5 ²⁵	Aplikasi	- Uke UI - - pengelak - + proses path Manual - Raton + Mouse	
3.	21/6 ²⁵	Aplikasi	tambah no urut hasil rekomendasi	
4.	24/6 ²⁵	Aplikasi	OK	

5	25/6	Laporan	Revis	
6	8/8	Laporan	Revis	
7	13/7	Laporan	Revis	
8	14/7	Laporan	Revis	

Tegal,
Dosen Pembimbing I


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