

Pendekatan Single Page Application pada Sistem Informasi Bimbingan Tugas Akhir Menggunakan Metode Agile Developement

Abdul Rahim*¹, Agus Siswanto², Yulia Arvita³, Suyanti⁴

Program Studi Teknik Informatika, Universitas Dinamika Bangsa

e-mail: ¹abdulrahim@unama.ac.id, ²agussiswanto@unama.ac.id, ³yulia_arvita@unama.ac.id,

⁴suyanti@unama.ac.id

*Penulis Korespondensi

Diterima: 27 September 2024; Direvisi: 25 Mei 2025; Disetujui: 06 Juni 2025

Abstrak

Penelitian ini bertujuan untuk mengembangkan aplikasi bimbingan tugas akhir berbasis Single Page Application (SPA) guna meningkatkan efisiensi dalam proses bimbingan di perguruan tinggi. Permasalahan seperti kurangnya interaksi antara dosen dan mahasiswa serta sulitnya pengelolaan jadwal bimbingan dan revisi menjadi fokus utama. Aplikasi dikembangkan menggunakan metode Agile, yang memungkinkan pengembangan secara bertahap sesuai kebutuhan pengguna. Dengan desain SPA, aplikasi ini memberikan pengalaman yang lebih responsif, mengurangi waktu pemuatan, dan mempercepat komunikasi. Selain itu, aplikasi ini juga dilengkapi dengan fitur QR Code untuk mempermudah proses presensi bimbingan, di mana mahasiswa dapat melakukan presensi dengan cepat dan akurat hanya dengan memindai kode yang dihasilkan sistem. Fitur ini tidak hanya mempercepat proses pencatatan kehadiran, tetapi juga meminimalkan risiko kesalahan pencatatan. Uji coba menunjukkan bahwa aplikasi ini mampu meningkatkan efektivitas bimbingan dengan mengurangi hambatan komunikasi dan meningkatkan akurasi pencatatan. Hasil ini penting karena memperbaiki proses bimbingan tugas akhir secara keseluruhan. Kesimpulannya, aplikasi SPA ini terbukti mampu meningkatkan efisiensi dalam proses bimbingan akademik dan presensi mahasiswa.

Kata kunci: single page application, bimbingan tugas akhir, aplikasi akademik, agile.

Abstract

This study aims to develop a thesis guidance application based on Single Page Application (SPA) to improve the efficiency of the guidance process in universities. Issues such as a lack of interaction between supervisors and students, as well as difficulties in managing guidance schedules and revisions, are the main focus. The application was developed using the Agile method, allowing iterative development based on user needs. With the SPA design, the application provides a more responsive experience, reducing load times and speeding up communication. In addition, the application is equipped with a QR Code feature to facilitate the attendance process during guidance sessions, where students can quickly and accurately register their attendance by scanning the code generated by the system. This feature not only speeds up the attendance recording process but also minimizes the risk of errors. Testing showed that the application successfully enhanced the effectiveness of the guidance process by reducing communication barriers and improving attendance accuracy. These results are significant as they improve the overall thesis guidance process. In conclusion, the SPA-based application has proven to increase efficiency in both the academic guidance process and student attendance recording.

Keywords: *single page application, thesis guidance, academic application, agile.*

1. PENDAHULUAN

Bimbingan tugas akhir merupakan salah satu komponen esensial dalam penyelesaian studi di perguruan tinggi. Proses bimbingan tugas akhir ini memfasilitasi mahasiswa dalam mendapatkan arahan dari dosen pembimbing untuk menyelesaikan penelitian secara sistematis. Namun, keterbatasan waktu pertemuan dan masalah komunikasi antara mahasiswa dan dosen sering kali menjadi kendala utama. Masalah komunikasi yang tidak efektif merupakan salah satu faktor penghambat utama penyelesaian tugas akhir mahasiswa [1]. Pengelolaan bimbingan secara manual juga kerap kali menyebabkan bentrokan jadwal dan ketidakpastian dalam dokumentasi revisi yang dilakukan [2].

Bimbingan tugas akhir merupakan salah satu komponen penting dalam menyelesaikan studi mahasiswa di perguruan tinggi. Namun, terdapat berbagai kendala dalam proses bimbingan ini, seperti keterbatasan waktu pertemuan dan komunikasi yang tidak efektif antara mahasiswa dan dosen pembimbing [1]. Hambatan komunikasi sering kali menjadi penyebab utama keterlambatan penyelesaian skripsi. Selain itu, pengelolaan bimbingan yang masih dilakukan secara manual dapat menimbulkan masalah dalam mengatur jadwal pertemuan dan dokumentasi [3].

Untuk mengatasi permasalahan tersebut, pengembangan aplikasi bimbingan berbasis *Single Page Application* (SPA) dipandang sebagai solusi yang efektif. SPA memungkinkan interaksi yang lebih responsif, karena seluruh konten aplikasi dimuat dalam satu halaman tanpa perlu memuat ulang keseluruhan halaman setiap kali terjadi interaksi. Hal ini mempercepat akses data dan meningkatkan pengalaman pengguna. Lokapitasari (2018) menemukan bahwa penerapan SPA pada sistem informasi akademik mampu meningkatkan efisiensi waktu dan penggunaan sumber daya perangkat lunak [4].

Framework Laravel dan Livewire digunakan dalam pengembangan aplikasi ini karena memberikan keunggulan dalam hal pengelolaan backend dan frontend secara interaktif. Laravel menawarkan fleksibilitas, efisiensi, dan keamanan yang optimal dalam pengembangan aplikasi berbasis web skala besar. Sementara itu, Livewire memudahkan pengembangan tampilan antarmuka interaktif tanpa perlu banyak menggunakan JavaScript, sehingga mempercepat proses pengembangan aplikasi. Implementasi AJAX dalam Laravel juga mendukung pengembangan SPA yang lebih responsif dengan cara menangani permintaan data secara asinkron [5], [6].

Metode pengembangan yang digunakan adalah *Agile Development*. Metode ini memungkinkan pengembangan dilakukan secara bertahap, sehingga mempermudah penyesuaian berdasarkan umpan balik pengguna di setiap iterasi [7], [8]. Nasution (2020) menekankan bahwa pendekatan Agile memberikan fleksibilitas yang lebih tinggi, memungkinkan tim pengembang untuk beradaptasi dengan cepat terhadap perubahan kebutuhan pengguna [9]. Selain itu, metode ini meningkatkan kolaborasi antara pengembang dan pengguna untuk menghasilkan produk akhir yang lebih sesuai dengan kebutuhan mereka [7].

Selain fitur-fitur di atas, aplikasi ini juga dilengkapi dengan teknologi QR Code yang memudahkan proses presensi bimbingan. Fitur QR Code memungkinkan mahasiswa melakukan presensi secara cepat dan akurat dengan cara memindai kode yang dihasilkan sistem setiap kali bimbingan berlangsung. Penggunaan teknologi QR Code dalam sistem presensi terbukti mampu meningkatkan akurasi pencatatan data kehadiran dan meminimalkan kesalahan. Implementasi fitur ini tidak hanya mempercepat proses presensi, tetapi juga membantu mengurangi risiko kesalahan pencatatan dan meningkatkan keteraturan dalam pengelolaan kehadiran mahasiswa [10], [11].

Berdasarkan hasil-hasil penelitian sebelumnya, pengembangan aplikasi bimbingan tugas akhir berbasis SPA dengan dukungan framework Laravel, Livewire, dan teknologi QR Code diharapkan dapat meningkatkan efisiensi manajemen bimbingan akademik, mempercepat komunikasi antara dosen dan mahasiswa, serta mengurangi hambatan yang sering terjadi dalam

proses bimbingan tradisional.

2. METODOLOGI PENELITIAN

Pengumpulan data yang diperlukan dalam penelitian akan dilakukan dengan beberapa cara, yaitu:

1. Metode Observasi digunakan untuk memperoleh data dan informasi dari proses bimbingan tugas akhir, seperti pola komunikasi antara dosen dan mahasiswa, masalah yang sering muncul dalam manajemen jadwal bimbingan, serta kendala dalam pengelolaan revisi. Data ini menjadi dasar dalam proses perancangan aplikasi bimbingan tugas akhir berbasis *Single Page Application* (SPA).
2. Metode Wawancara digunakan untuk mendapatkan data dan informasi yang dibutuhkan, seperti pengalaman dosen dan mahasiswa dalam bimbingan tugas akhir, kebutuhan akan fitur-fitur tertentu dalam aplikasi, serta hambatan yang dihadapi dalam sistem bimbingan yang ada. Wawancara dilakukan dengan dosen pembimbing dan mahasiswa dari beberapa program studi untuk mendapatkan pemahaman yang mendalam terkait proses bimbingan.

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan pendekatan Agile, yang memungkinkan pengembangan aplikasi dilakukan secara bertahap dan berulang berdasarkan umpan balik dari pengguna. Untuk pengembangan aplikasi ini, digunakan framework PHP Laravel dengan Livewire sebagai library untuk membangun arsitektur SPA. Laravel dipilih karena keunggulannya dalam pengelolaan sistem backend yang efisien, sementara Livewire memungkinkan pembuatan komponen interaktif tanpa perlu menulis JavaScript secara langsung.



Gambar 1. Agile development [12]

Tahapan Agile dalam pengembangan aplikasi bimbingan tugas akhir berbasis *Single Page Application* (SPA), dengan framework Laravel dan Livewire, yang ditampilkan pada gambar 1 dapat dijelaskan sebagai berikut:

1. *Requirements* (Identifikasi Kebutuhan)
Pada tahap ini, dilakukan pengumpulan data dan informasi terkait kebutuhan aplikasi, seperti kendala yang dihadapi oleh dosen dan mahasiswa dalam proses bimbingan tugas akhir. Observasi dan wawancara digunakan untuk memahami masalah yang ada dan fitur yang dibutuhkan dalam aplikasi, seperti pengelolaan jadwal, revisi, dan komunikasi.
2. *Design* (Perancangan Sistem)
Setelah kebutuhan teridentifikasi, perancangan sistem dilakukan dengan mempertimbangkan aspek-aspek teknis. Pada tahap ini, arsitektur aplikasi, termasuk antarmuka pengguna (UI) dan pengalaman pengguna (UX), dirancang menggunakan Laravel dan Livewire untuk membangun SPA yang responsif dan interaktif.
3. *Development* (Pengembangan Aplikasi)

Dalam tahap pengembangan, aplikasi dibangun secara iteratif sesuai dengan desain yang telah dibuat. Proses ini meliputi pengkodean fitur-fitur utama, seperti pengelolaan jadwal bimbingan, revisi dokumen, dan komunikasi real-time. Pada setiap iterasi, fitur yang dikembangkan akan diuji dan ditingkatkan berdasarkan umpan balik.

4. *Testing* (Pengujian Perangkat Lunak)

Pengujian dilakukan untuk memastikan setiap fitur dalam aplikasi berjalan sesuai dengan spesifikasi. Dalam konteks ini, digunakan metode Black Box Testing untuk memeriksa fungsionalitas tanpa memeriksa kode internal. Fitur-fitur utama seperti pengelolaan jadwal, revisi, dan komunikasi diuji untuk memastikan aplikasi berjalan dengan baik.

5. *Deployment* (Penerapan Aplikasi)

Setelah aplikasi diuji dan dinyatakan siap, aplikasi di-deploy atau diterapkan dalam lingkungan produksi. Pada tahap ini, aplikasi dapat diakses oleh pengguna, yaitu dosen dan mahasiswa yang terlibat dalam proses bimbingan tugas akhir.

6. *Review* (Evaluasi dan Penyempurnaan)

Setelah implementasi, aplikasi akan dievaluasi berdasarkan umpan balik dari pengguna. Masukan dari dosen dan mahasiswa digunakan untuk melakukan perbaikan pada fitur-fitur yang kurang optimal. Tahapan ini memastikan aplikasi selalu diperbarui dan disesuaikan dengan kebutuhan pengguna yang berkembang.



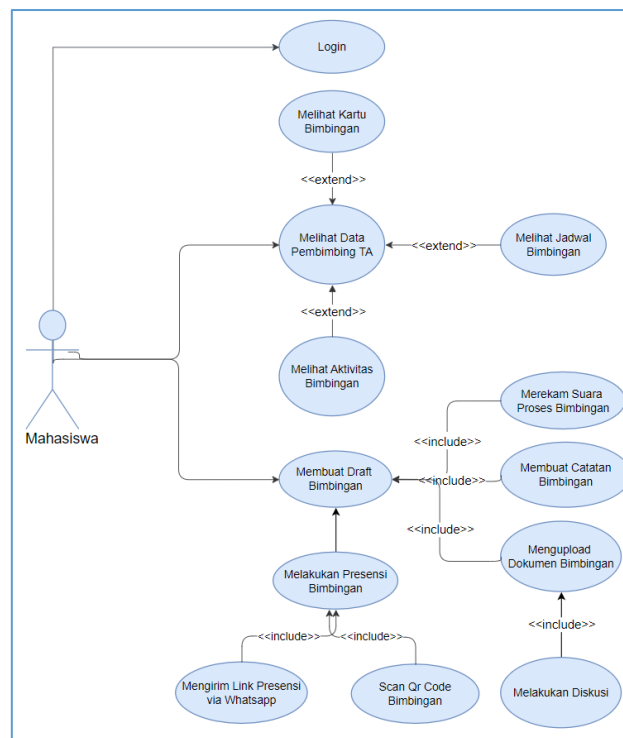
Gambar 2. Traditional page vs SPA [13]

Penelitian ini menggunakan arsitektur *Single Page Application* (SPA) untuk memberikan pengalaman pengguna yang lebih cepat dan responsif. Dengan SPA, semua interaksi dalam aplikasi bimbingan tugas akhir dilakukan di satu halaman tanpa harus memuat ulang seluruh halaman, sehingga akses ke aplikasi menjadi lebih cepat dan efisien. Dalam gambar 2 menjelaskan bagaimana perbedaan antara model halaman tradisional dan SPA: dalam SPA, setelah permintaan awal untuk HTML, semua interaksi selanjutnya dikomunikasikan ke server melalui AJAX yang kemudian merespons dengan JSON, yang langsung diperbarui pada halaman tanpa perlu pemuatan ulang. Aplikasi ini dibangun menggunakan framework Laravel, yang mengelola bagian backend, dan Livewire, yang memungkinkan pembuatan tampilan interaktif tanpa memerlukan banyak JavaScript. Livewire memungkinkan pembaruan antarmuka secara dinamis berdasarkan interaksi pengguna, sehingga mempercepat proses bimbingan tugas akhir tanpa mengganggu pengalaman pengguna. Pengembangan aplikasi ini dilakukan menggunakan metode Agile, yang memungkinkan perubahan atau penambahan fitur dilakukan secara bertahap dan cepat berdasarkan umpan balik pengguna di setiap iterasi. Dengan pendekatan ini, aplikasi dapat diuji dan disesuaikan untuk memastikan fungsionalitas yang optimal dalam mendukung proses bimbingan tugas akhir.

3. PEMBAHASAN DAN HASIL

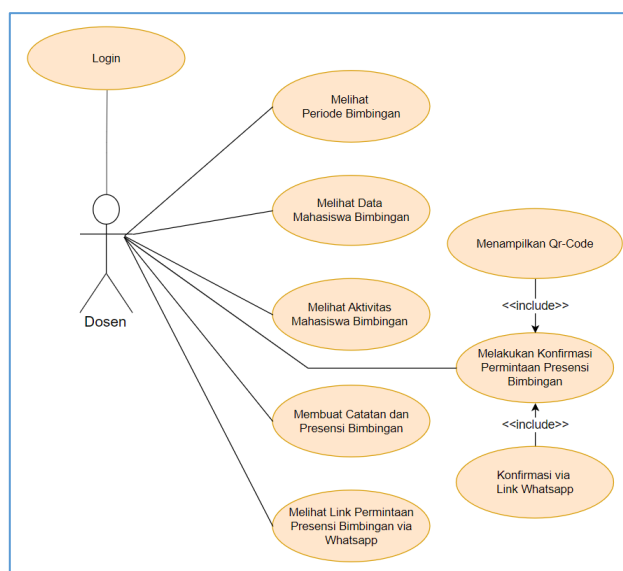
3.1. Use Case Diagram

Pada penelitian menggunakan 2 aktor yaitu dosen dan mahasiswa, berikut *use case diagram* untuk aktor mahasiswa yang ditunjukkan pada gambar 3.



Gambar 3. Use case diagram aktor mahasiswa

Use case diagram ini menggambarkan alur interaksi mahasiswa dengan sistem aplikasi bimbingan tugas akhir, dimulai dari login untuk mengakses fitur-fitur seperti melihat kartu bimbingan, data pembimbing, jadwal bimbingan, dan aktivitas bimbingan sebelumnya. Mahasiswa dapat membuat draft bimbingan yang mencakup mengunggah dokumen, membuat catatan, dan merekam proses bimbingan. Selain itu, mahasiswa diwajibkan melakukan presensi bimbingan melalui berbagai cara, seperti mengirim link presensi via WhatsApp atau memindai QR code. Diagram ini juga mencakup fitur diskusi antara mahasiswa dan dosen pembimbing, yang bertujuan untuk memastikan bimbingan berjalan secara efektif dan terdokumentasi dengan baik.

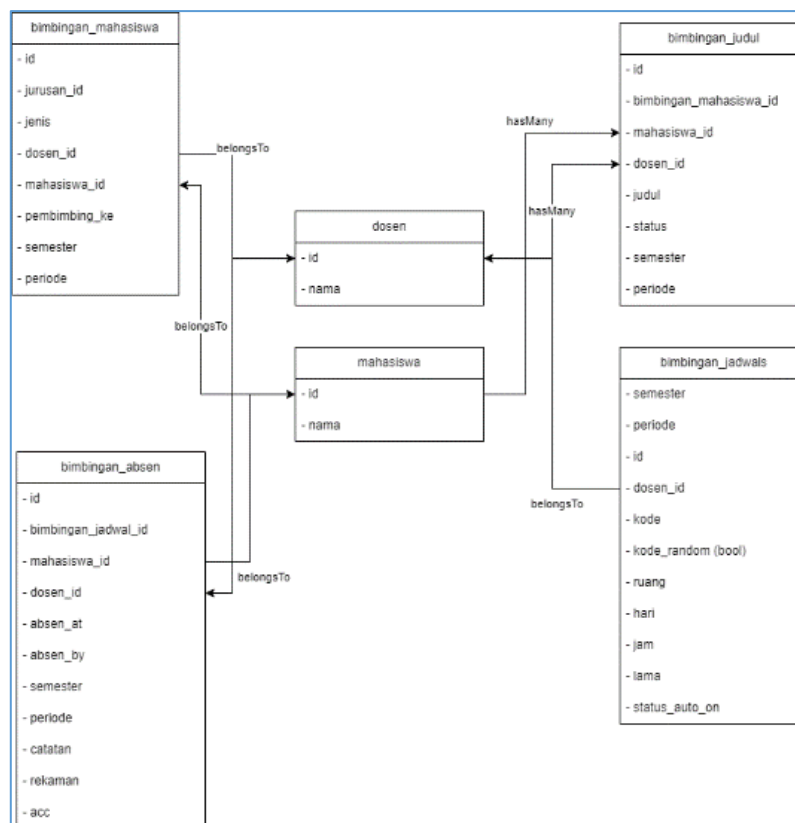


Gambar 4. Use case diagram aktor dosen

Sedangkan pada gambar 4 *use case diagram* menggambarkan interaksi antara dosen dengan sistem aplikasi bimbingan tugas akhir. Dosen dapat login dan mengakses beberapa fitur utama seperti melihat periode bimbingan, data mahasiswa yang dibimbing, serta aktivitas mahasiswa selama proses bimbingan. Selain itu, dosen juga dapat membuat catatan dan presensi bimbingan. Terdapat fitur untuk menampilkan QR code yang memungkinkan mahasiswa melakukan presensi, serta fitur konfirmasi permintaan presensi yang bisa dilakukan melalui link WhatsApp. Diagram ini memfasilitasi alur kerja dosen dalam memantau dan mencatat aktivitas bimbingan dengan lebih efisien dan terorganisir.

3.2. Class Diagram

Pada tahapan ini, penulis menggambarkan hubungan antara model di framework laravel menggunakan class diagram. Berikut pada gambar 5 adalah *class diagram* yang digunakan di penelitian ini.



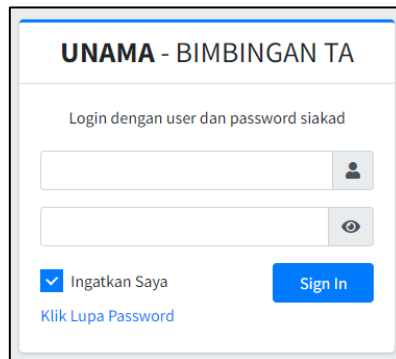
Gambar 5. Class Diagram

Class diagram di atas menggambarkan struktur model basis data untuk sistem bimbingan tugas akhir yang dikembangkan dengan Laravel. Terdapat beberapa entitas utama seperti mahasiswa dan dosen, yang masing-masing memiliki atribut dasar seperti id dan nama. Relasi antara mahasiswa dan dosen diatur dengan relasi *belongsTo*, di mana seorang mahasiswa memiliki dosen pembimbing. Entitas **bimbingan_mahasiswa** menyimpan informasi mengenai proses bimbingan, termasuk atribut seperti jurusan_id, dosen_id, pembimbing_ke, dan periode. Selain itu, entitas **bimbingan_judul** dan **bimbingan_jadwal** mengatur judul dan jadwal bimbingan, dengan relasi *hasMany* ke entitas **bimbingan_mahasiswa**. Tabel **bimbingan_absen** mencatat kehadiran mahasiswa selama bimbingan, termasuk atribut seperti absen_by, catatan, dan rekaman. Secara keseluruhan, diagram ini menunjukkan bagaimana setiap entitas saling terkait untuk mengelola proses bimbingan tugas akhir secara terstruktur dan efisien.

3.3. Implementasi

1. Halaman Login

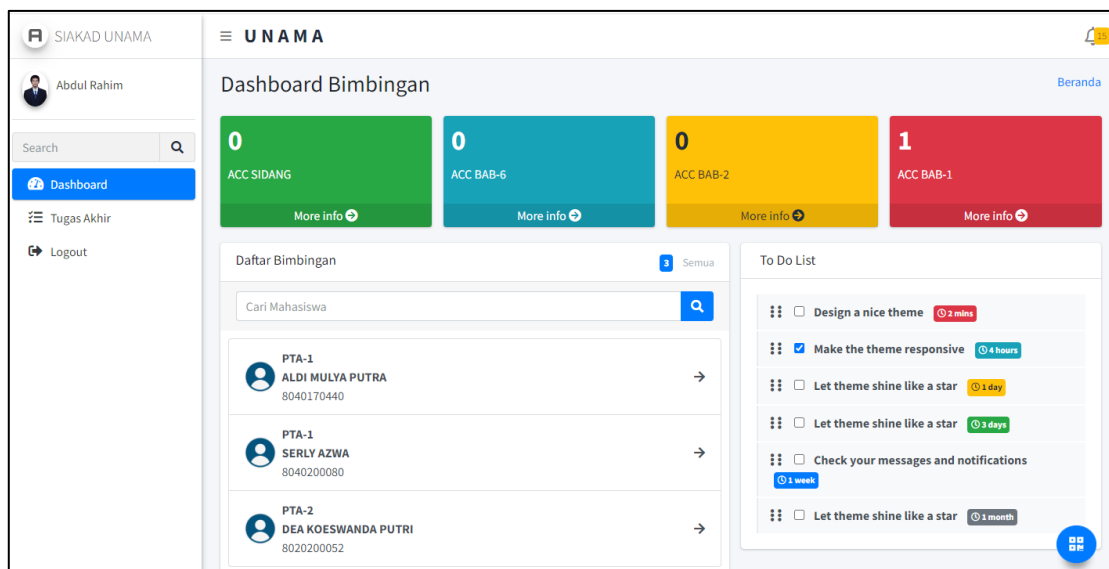
Implementasi halaman login dapat dilihat di gambar 6, pada halaman ini, dosen ataupun mahasiswa dapat login ke sistem untuk bisa menggunakan fitur-fitur di aplikasi bimbingan tugas akhir.



Gambar 6. Login User

2. Halaman Beranda Dosen

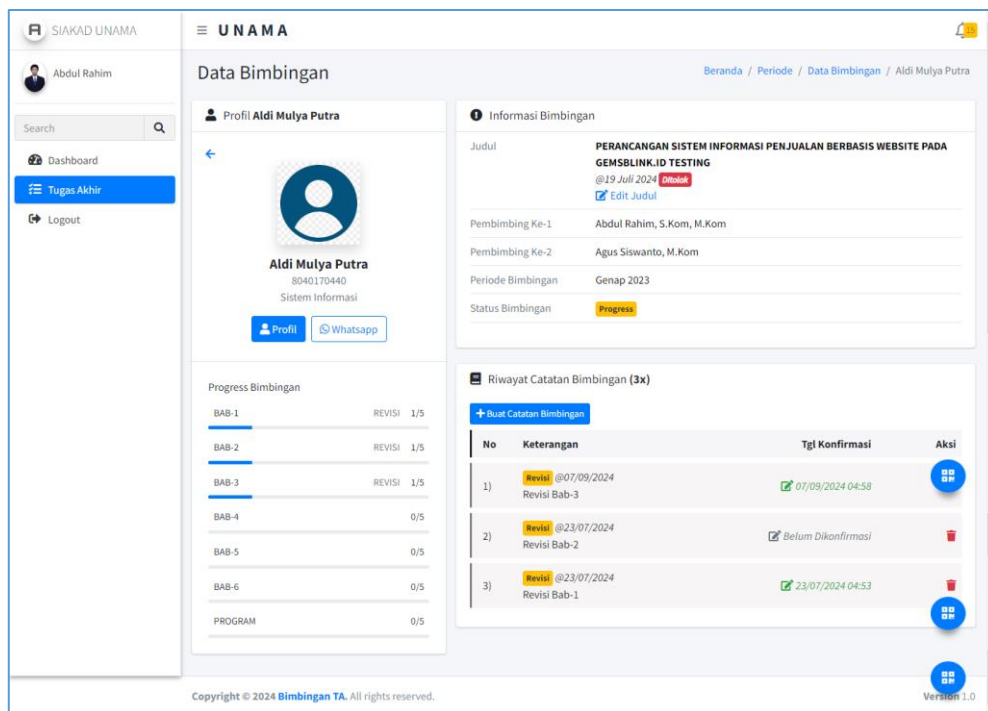
Dosen yang berhasil login akan berpindah ke halaman beranda seperti gambar 7. Pada halaman ini dosen dapat melihat daftar mahasiswa bimbingan, dosen juga bisa melihat berapa total bimbingan yang sudah acc sidang, acc bab 4, bab 2 dan acc bab 1.



Gambar 7. Beranda Dosen

3. Halaman Data Bimbingan Mahasiswa

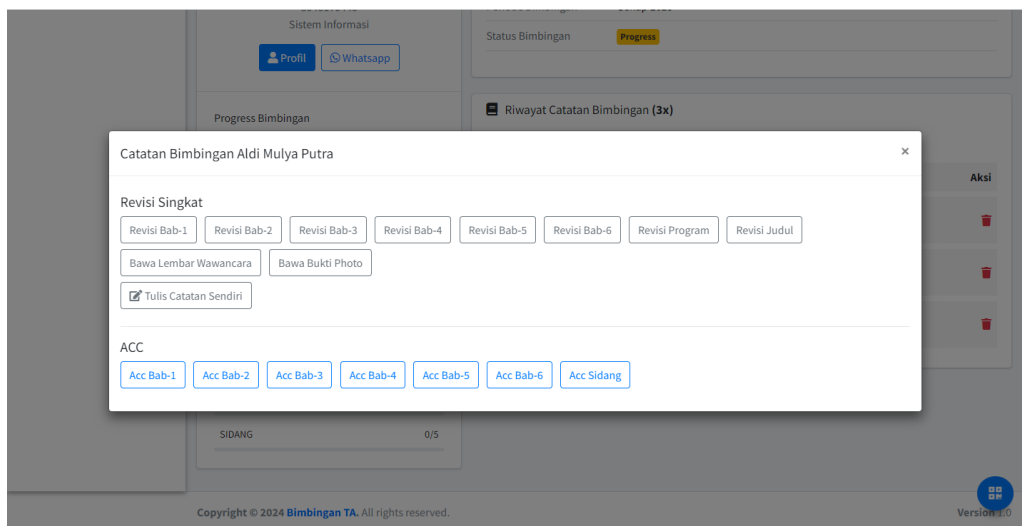
Berikut adalah halaman detail bimbingan mahasiswa, di halaman ini, dosen dapat melihat informasi judul, progress bimbingan dan data bimbingan.



Gambar 8. Detail bimbingan

4. Halaman Bimbingan Manual Dosen

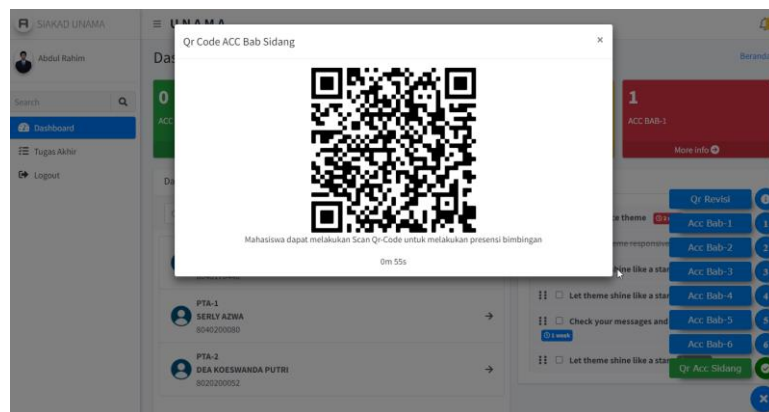
Pada halaman 9, dosen dapat membuat jadwal bimbingan manual dengan memilih salah satu dari link digambar berikut:



Gambar 9. Bimbingan anual

5. Halaman Qr-Code

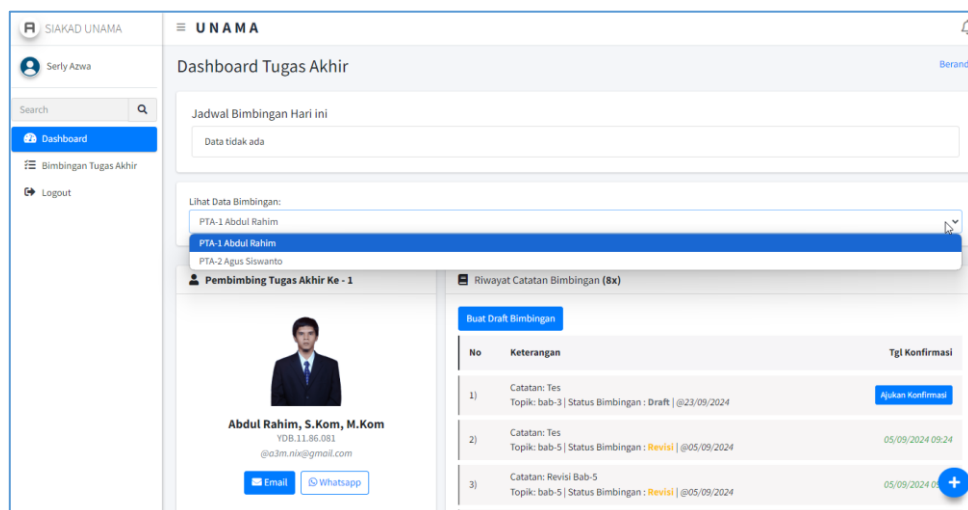
Pada setiap halaman, dosen dapat mengklik floating button untuk menampilkan qr-code yang kemudian dapat di scan oleh mahasiswa yang sedang bimbingan.



Gambar 10. Qr-Code

6. Halaman Beranda Mahasiswa

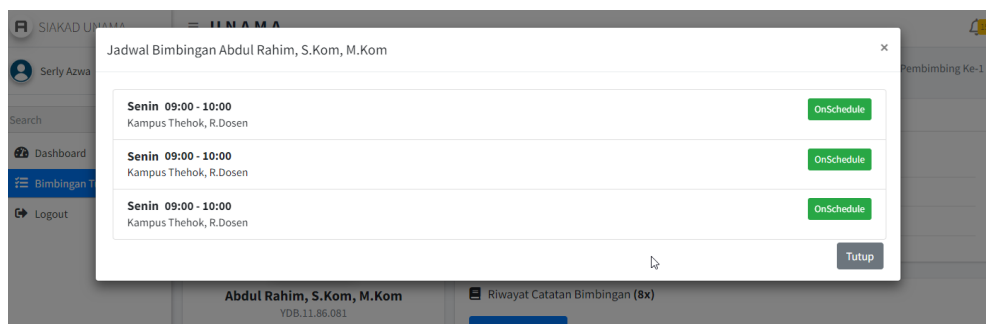
Mahasiswa yang berhasil login akan masuk ke halaman 10, disini mahasiswa bisa melihat informasi bimbingan yang telah dilakukan.



Gambar 11. Beranda mahasiswa

7. Halaman Jadwal Bimbingan

Mahasiswa dapat melihat jadwal bimbingan dengan mengklik link jadwal bimbingan dosen.



Gambar 12. Jadwal bimbingan

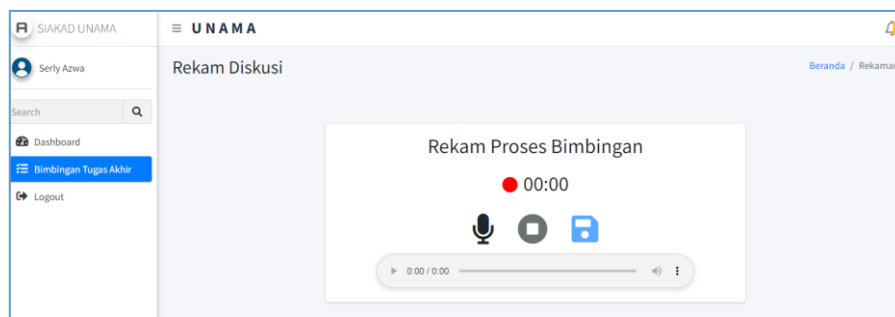
8. Halaman Form Bimbingan

Di halaman form bimbingan, mahasiswa dapat mencatat informasi bimbingan dengan memilih dosen yang topik bimbingan.

Gambar 13. Form bimbingan

9. Halaman rekaman bimbingan

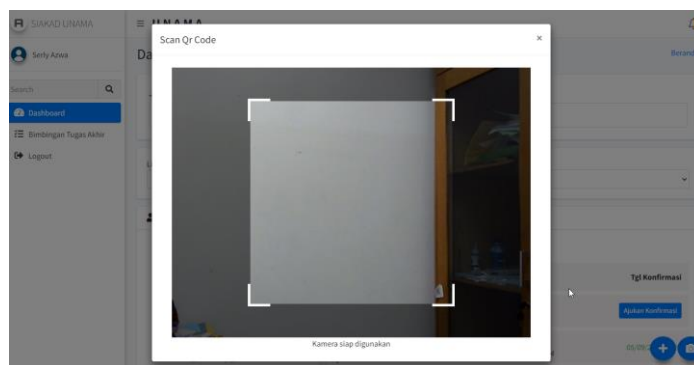
Pada halaman ini, mahasiswa dapat merekam diskusi yang dilakukan dengan dosen, tentunya ini dilakukan sesuai izin yang diberikan oleh dosen.



Gambar 14. Merekam aktivitas bimbingan

10. Halaman Scan Qr Code

Pada halaman ini, mahasiswa dapat melakukan scan qr code yang diberikan oleh dosen, proses presensi bimbingan di tandai dengan mahasiswa berhasil melakukan scan qr code dari dosen.



Gambar 15. Qr code scanner

3.4. Pengujian

Berikut pada tabel aktifitas yang dilakukan pada saat pengujian sistem:

Tabel 1. Pengujian Sistem

No.	Use Case	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Status
1	Login	Menguji apakah mahasiswa atau dosen dapat login ke dalam sistem	- Masukkan username dan password yang valid - Klik tombol "Login"	Pengguna berhasil masuk ke dashboard	Berhasil
2	Melihat Kartu Bimbingan	Menguji apakah mahasiswa dapat melihat kartu bimbingan	- Login sebagai mahasiswa - Akses menu "Kartu Bimbingan"	Kartu bimbingan mahasiswa ditampilkan dengan benar	Berhasil
3	Melihat Data Pembimbing TA	Menguji apakah mahasiswa dapat melihat informasi pembimbing tugas akhir	- Login sebagai mahasiswa - Akses menu "Data Pembimbing"	Informasi dosen pembimbing ditampilkan dengan benar	Berhasil
4	Melihat Jadwal Bimbingan	Menguji apakah mahasiswa dapat melihat jadwal bimbingan	- Login sebagai mahasiswa - Akses menu "Jadwal Bimbingan"	Jadwal bimbingan mahasiswa ditampilkan dengan benar	Berhasil
5	Membuat Draft Bimbingan	Menguji apakah mahasiswa dapat membuat draft bimbingan dan mengunggah dokumen	- Login sebagai mahasiswa - Akses menu "Membuat Draft" - Unggah dokumen	Draft bimbingan berhasil diunggah dan tersimpan dalam sistem	Berhasil
6	Melakukan Presensi Bimbingan	Menguji apakah mahasiswa dapat melakukan presensi bimbingan	- Login sebagai mahasiswa - Pilih sesi bimbingan yang dihadiri - Klik "Presensi"	Presensi berhasil tercatat dalam sistem	Berhasil
7	Melihat Aktivitas Bimbingan	Menguji apakah mahasiswa dapat melihat riwayat aktivitas bimbingan	- Login sebagai mahasiswa - Akses menu "Aktivitas Bimbingan"	Riwayat aktivitas bimbingan ditampilkan dengan benar	Berhasil
8	Membuat Catatan Bimbingan	Menguji apakah dosen dapat membuat dan menyimpan catatan bimbingan	- Login sebagai dosen - Akses kartu bimbingan mahasiswa - Tambahkan catatan bimbingan	Catatan berhasil ditambahkan dan disimpan pada sistem	Berhasil
9	Konfirmasi Permintaan Presensi	Menguji apakah dosen dapat mengonfirmasi permintaan presensi bimbingan mahasiswa	- Login sebagai dosen - Lihat permintaan presensi dari mahasiswa - Klik "Konfirmasi"	Permintaan presensi berhasil dikonfirmasi	Berhasil
10	Mengirim Link Presensi via WhatsApp	Menguji apakah mahasiswa dapat mengirim link presensi ke dosen via WhatsApp	- Login sebagai mahasiswa - Klik "Kirim Link Presensi" - Pilih opsi "Kirim via WhatsApp"	Link presensi berhasil dikirim melalui WhatsApp	Berhasil

No.	Use Case	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Status
11	Melihat Periode Bimbingan	Menguji apakah dosen dapat melihat periode bimbingan yang sedang berlangsung	- Login sebagai dosen - Akses menu "Periode Bimbingan"	Informasi periode bimbingan ditampilkan dengan benar	Berhasil

4. KESIMPULAN

Penelitian ini berhasil mengembangkan sebuah aplikasi bimbingan tugas akhir berbasis SPA dengan memanfaatkan Laravel sebagai framework backend dan Livewire untuk pembuatan tampilan interaktif. Arsitektur SPA dipilih karena mampu menyediakan pengalaman pengguna yang lebih cepat dan responsif, di mana seluruh interaksi di aplikasi dilakukan di satu halaman tanpa perlu memuat ulang seluruh halaman. Hal ini membuat akses ke fitur-fitur aplikasi, seperti melihat kartu bimbingan, membuat draft, dan melakukan presensi, menjadi lebih efisien dan hemat waktu. Penggunaan metode pengembangan *Agile* juga menjadi faktor penting dalam keberhasilan penelitian ini, karena memungkinkan pengembangan aplikasi dilakukan secara bertahap, dengan penyesuaian fitur berdasarkan umpan balik pengguna di setiap iterasi. Pendekatan ini tidak hanya mempermudah proses pengembangan tetapi juga memastikan bahwa aplikasi selalu selaras dengan kebutuhan pengguna dan siap digunakan dalam kondisi optimal. Berdasarkan hasil pengujian, semua fitur yang dikembangkan, seperti login, pengelolaan bimbingan, serta fitur presensi, berhasil diimplementasikan dengan baik dan sesuai dengan yang diharapkan. Dengan demikian, aplikasi ini dapat menjadi solusi yang efektif untuk mempermudah proses bimbingan tugas akhir di perguruan tinggi, dan diharapkan dapat terus dikembangkan dengan penambahan fitur-fitur baru yang semakin menyempurnakan fungsionalitasnya, seperti notifikasi otomatis dan fitur kolaborasi yang lebih canggih antara dosen dan mahasiswa. Aplikasi ini juga membuka peluang untuk penelitian lanjutan, terutama dalam mengevaluasi performa dan kepuasan pengguna di berbagai kondisi penggunaan.

DAFTAR PUSTAKA

- [1] D. E. Pratiwi and A. Roosyanti, "Analisis Faktor Penghambat Skripsi Mahasiswa Jurusan Pendidikan Guru Sekolah Dasar Universitas Wijaya Kusuma Surabaya," *J. Pendidik. Dasar*, vol. 10, no. 1, pp. 101–114, May 2019, doi: 10.21009/jpd.v10i1.11272.
- [2] E. Noviana, O. Kurniawan, and M. N. Huda, "Pengembangan Aplikasi Bimbingan Tugas Akhir Mahasiswa Berbasis Website pada Program Studi Pendidikan Guru Sekolah Dasar Fkip Universitas RiaU," *Prim. J. Pendidik. Guru Sekol. Dasar*, vol. 7, no. 1, p. 1, Apr. 2018, doi: 10.33578/jpkip.v7i1.5334.
- [3] N. Renaningtias and D. Apriliani, "Penerapan Metode Prototype Pada Pengembangan Sistem Informasi Tugas Akhir Mahasiswa," *Rekursif J. Inform.*, vol. 9, no. 1, May 2021, doi: 10.33369/rekursif.v9i1.15772.
- [4] P. L. Lokapitasari Belluano, "Pengembangan Single Page Application pada Sistem Informasi Akademik," *Ilk. J. Ilm.*, vol. 10, no. 1, pp. 38–43, Apr. 2018, doi: 10.33096/ilkom.v10i1.204.38-43.
- [5] Abdul Rahim, "Optimalisasi Proses Sinkronisasi Data Akademik dan Web Services PDDIKTI Menggunakan Fitur Queues pada Framework Laravel," *J. Teknol. Inf. Dan Komun.*, vol. 9, no. 1, pp. 45–55, 2020.
- [6] I. K. Natania, E. D. Wahyuni, and S. F. A. Wati, "Rancang Bangun Aplikasi Donor Darah Darurat Berbasis Website Menggunakan Livewire Laravel," vol. 11, no. 3, 2024.

- [7] I. Ramadhani, "Penerapan Single Page Application untuk Aplikasi Bimbingan Akademik," *J. Teknol. Inf.*, vol. 15, no. 3, pp. 65–78, 2021.
- [8] J. A. Rosman, I. Imron, and M. H. Prasetyo, "Sistem Informasi Bimbingan Tugas Akhir dan Skripsi Online Berbasis Web: Sistem Informasi Bimbingan Tugas Akhir dan Skripsi Online Berbasis Web," *J. INSAN J. Inf. Syst. Manag. Innov.*, vol. 1, no. 1, pp. 61–69, Jun. 2021, doi: 10.31294/jinsan.v1i1.433.
- [9] A. Nasution, M. Siddik, and A. P. Lubis, "Pemanfaatan Agile Development untuk Pengembangan Sistem Informasi Akademik Brainfor Islamic School".
- [10] M. Nishom, T. Abidin, and S. Wiyono, "Pemanfaatan Teknologi Qr-Code untuk Presensi Siswa Di Era Disrupsi Digital," *JMM J. Masy. Mandiri*, vol. 7, no. 2, p. 1984, Apr. 2023, doi: 10.31764/jmm.v7i2.13863.
- [11] M. Handayani, "Efektivitas Penerapan QR Code dalam Presensi Mahasiswa," *J. Sist. Inf. Dan Teknol.*, vol. 9, no. 3, pp. 45–58, 2019.
- [12] Telkom University, "Agile Method Image." [Online]. Available: <https://it.telkomuniversity.ac.id/wp-content/uploads/2022/12/Agile-Method.jpeg>
- [13] V. Puzhevich, "Single-Page Application Vs. Multiple-Page Application." [Online]. Available: <https://scand.com/company/blog/single-page-application-vs-multi-page-application/>