
PENGEMBANGAN SISTEM PENJADWALAN SIDANG SKRIPSI BERBASIS WEB DENGAN RAPID APPLICATION DEVELOPMENT DAN ALGORITMA BACKTRACKING

Roberto Kaban¹⁾, Nurhafiz Ahmad Rangkuti²⁾, Ita Margareta Br Tarigan³⁾, David JM Sembiring⁴⁾
Teknik Informatika

Institut Teknologi dan Bisnis Indonesia

Jl. T. Amir Hamzah Km 31,5 Desa Tandem Hilir Kecamatan Hamparan Perak

e-mail: roberto.kaban@yahoo.com¹⁾, nurhafiz26ahmadrangkuti@gmail.com²⁾, itamargareta1997@gmail.com³⁾, davidjms366@gmail.com⁴⁾

Abstrak

Proses penjadwalan sidang skripsi merupakan salah satu kegiatan akademik yang memiliki tingkat kompleksitas tinggi karena melibatkan berbagai pihak dan sumber daya dengan keterbatasan waktu tertentu. Permasalahan yang sering terjadi pada proses penjadwalan meliputi bentrokan jadwal dosen, keterbatasan ketersediaan ruangan, serta kesulitan dalam menyesuaikan jadwal apabila terjadi perubahan. Penelitian ini bertujuan untuk mengembangkan sistem penjadwalan sidang skripsi berbasis web menggunakan algoritma backtracking dengan studi kasus di Institut Teknologi dan Bisnis Indonesia (ITB Indonesia). Metode pengembangan sistem yang digunakan adalah Rapid Application Development (RAD) yang menekankan pengembangan sistem secara iteratif dan melibatkan pengguna secara aktif. Proses penjadwalan dimodelkan sebagai permasalahan penjadwalan berbasis kendala, di mana setiap mahasiswa harus memperoleh satu jadwal sidang, dosen tidak diperkenankan terlibat dalam lebih dari satu sidang pada waktu yang sama, dan setiap ruangan hanya dapat digunakan untuk satu kegiatan dalam satu slot waktu tertentu. Algoritma backtracking diterapkan untuk mengevaluasi setiap kemungkinan penjadwalan berdasarkan kendala yang telah ditetapkan. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu menghasilkan jadwal sidang skripsi yang valid dan bebas konflik. Penerapan sistem ini diharapkan dapat membantu pengelola akademik dalam menyusun jadwal secara lebih efisien dan akurat, mengurangi kesalahan penjadwalan, serta meningkatkan kualitas layanan akademik.

Kata kunci: Algoritma Backtracking; Penjadwalan sidang Skripsi; Rapid Application Development; sistem berbasis web.

1. Pendahuluan

Perkembangan teknologi informasi telah mendorong perguruan tinggi untuk memanfaatkan sistem berbasis web dalam mendukung pengelolaan proses akademik. Pemanfaatan teknologi tersebut bertujuan untuk meningkatkan efisiensi, akurasi, dan transparansi layanan akademik [1], [2]. Salah satu proses akademik yang memiliki tingkat kompleksitas tinggi adalah penjadwalan sidang skripsi mahasiswa. Proses ini melibatkan berbagai pihak seperti mahasiswa, dosen pembimbing, dosen penguji, serta pengelola akademik, yang masing-masing memiliki keterbatasan waktu dan aturan tertentu.

Institut Teknologi dan Bisnis Indonesia (ITB Indonesia) sebagai perguruan tinggi yang menyelenggarakan program pendidikan di bidang teknologi dan bisnis melaksanakan sidang skripsi secara rutin pada setiap periode akademik. Proses penjadwalan sidang skripsi di ITB Indonesia melibatkan banyak komponen, antara lain data mahasiswa, ketersediaan dosen pembimbing dan dosen penguji, serta ketersediaan ruangan dan slot waktu. Dengan meningkatnya jumlah mahasiswa dan aktivitas akademik, proses penjadwalan sidang skripsi menjadi semakin kompleks.

Pada praktiknya, penjadwalan sidang skripsi yang dilakukan secara manual atau semi manual berpotensi menimbulkan berbagai permasalahan. Permasalahan penjadwalan sidang skripsi di ITB Indonesia dapat dipandang sebagai masalah penjadwalan berbasis kendala. Setiap mahasiswa harus memperoleh satu jadwal sidang. Setiap dosen tidak diperkenankan terlibat dalam lebih dari satu sidang pada waktu yang sama. Setiap ruangan hanya dapat digunakan untuk satu kegiatan dalam satu slot waktu tertentu. Apabila salah satu kendala tersebut tidak terpenuhi, maka jadwal yang dihasilkan menjadi tidak valid. Kondisi ini dapat menyebabkan keterlambatan pelaksanaan sidang skripsi dan berdampak pada efektivitas layanan akademik. Diperlukan suatu sistem penjadwalan yang mampu mengelola seluruh komponen tersebut secara terstruktur dan sistematis.

Algoritma backtracking merupakan salah satu algoritma yang dapat digunakan untuk menyelesaikan permasalahan penjadwalan dengan banyak kendala. Algoritma ini bekerja dengan membangun solusi secara bertahap dan melakukan pengecekan terhadap setiap kendala yang telah ditetapkan. Apabila pada suatu tahap ditemukan pelanggaran kendala, algoritma akan kembali ke tahap sebelumnya untuk mencoba alternatif solusi lain [3], [4]. Pendekatan ini memungkinkan sistem untuk menghasilkan jadwal sidang skripsi yang valid dan bebas konflik.

Sistem yang dibangun diharapkan mampu mengelola data mahasiswa, dosen pembimbing, dosen penguji, ruangan, dan slot waktu secara terintegrasi. Sistem diharapkan dapat menghasilkan jadwal sidang skripsi yang sesuai dengan seluruh kendala yang telah ditentukan. Perancangan dan pengembangan sistem dilakukan menggunakan metode Rapid Application Development (RAD) yang menekankan proses pengembangan secara iteratif dan cepat melalui keterlibatan aktif pengguna [5], [6], [7].

Hasil yang diharapkan dari penelitian ini adalah tersedianya sistem penjadwalan sidang skripsi berbasis web yang mampu membantu pengelola akademik di ITB Indonesia dalam menyusun jadwal secara lebih efisien dan akurat. Sistem ini diharapkan mampu mengurangi kesalahan penjadwalan, mempercepat proses penentuan jadwal, serta meningkatkan kualitas layanan akademik.

2. Landasan Teori

Sistem Informasi

Sistem informasi merupakan suatu sistem yang terdiri dari komponen-komponen yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, dan menyajikan informasi guna mendukung pengambilan keputusan dan pengelolaan suatu organisasi [8], [9]. Dalam perguruan tinggi, sistem informasi akademik berperan penting dalam mengelola data mahasiswa, dosen, dan aktivitas akademik lainnya. Penerapan sistem informasi yang baik dapat meningkatkan efisiensi operasional, mengurangi kesalahan administrasi, serta meningkatkan kualitas layanan akademik (Agustiani et al., 2023; Fatah & Mufarroha, 2022).

Sistem Berbasis Web

Sistem berbasis web adalah sistem yang dapat diakses melalui jaringan internet atau intranet menggunakan peramban web [10]. Sistem ini memungkinkan pengguna untuk mengakses informasi dan layanan tanpa bergantung pada perangkat atau lokasi tertentu [11], [12], [13]. Dalam lingkungan perguruan tinggi, sistem berbasis web memberikan kemudahan akses bagi pengelola akademik, dosen, dan mahasiswa.

Penjadwalan Sidang Skripsi

Penjadwalan merupakan proses pengalokasian sumber daya terhadap sekumpulan aktivitas dalam periode waktu tertentu dengan memperhatikan berbagai batasan [14]. Sumber daya dalam penjadwalan sidang skripsi meliputi dosen, ruangan, dan slot waktu, sedangkan aktivitas yang dijadwalkan adalah pelaksanaan sidang skripsi mahasiswa. Tujuan utama penjadwalan adalah menghasilkan susunan jadwal yang memenuhi seluruh aturan yang ditetapkan serta meminimalkan konflik [15]. Penjadwalan sidang skripsi adalah proses penentuan waktu dan tempat pelaksanaan sidang skripsi bagi mahasiswa dengan melibatkan dosen pembimbing dan dosen penguji.

Constraint Satisfaction Problem (CSP)

Constraint Satisfaction Problem (CSP) merupakan permasalahan yang terdiri dari sekumpulan variabel, domain nilai, dan kendala yang harus dipenuhi. Tujuan CSP adalah menemukan nilai untuk setiap variabel sehingga seluruh kendala terpenuhi [16], [17]. Dalam penjadwalan sidang skripsi, variabel dapat berupa mahasiswa atau jadwal sidang, domain berupa slot waktu dan ruangan, serta kendala berupa aturan yang membatasi penggunaan dosen dan ruangan.

Algoritma Backtracking

Algoritma backtracking adalah salah satu algoritma pencarian solusi yang digunakan untuk menyelesaikan masalah dengan banyak kendala [3], [4]. Algoritma ini bekerja dengan cara membangun solusi secara bertahap dan melakukan pengecekan terhadap kendala pada setiap tahap. Apabila suatu solusi parsial tidak memenuhi kendala, algoritma akan kembali ke langkah sebelumnya dan mencoba alternatif solusi lain [16]. Backtracking sangat efektif digunakan pada permasalahan yang memiliki ruang solusi terbatas namun kompleks, seperti penjadwalan [18], [19].

Unified Modeling Language (UML)

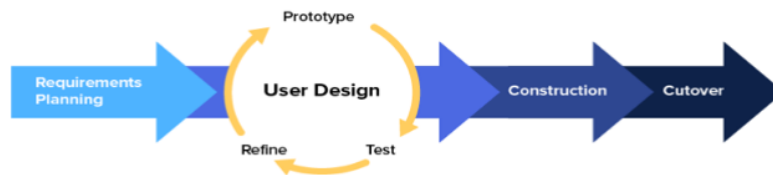
Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem perangkat lunak. UML menyediakan sekumpulan diagram yang digunakan untuk menggambarkan struktur dan perilaku sistem secara jelas dan terstruktur [20], [21]. Dengan UML, pengembang dapat memahami kebutuhan sistem dan hubungan antar komponen sebelum sistem diimplementasikan.

Dalam pengembangan sistem penjadwalan sidang skripsi berbasis web, UML digunakan sebagai alat bantu perancangan sistem. Diagram UML membantu menggambarkan interaksi antara pengguna dan sistem, alur proses penjadwalan, serta struktur data yang digunakan. Diagram yang umum digunakan meliputi use case diagram untuk menggambarkan fungsi sistem, activity diagram untuk memodelkan alur proses penjadwalan, sequence diagram untuk menunjukkan interaksi antar objek, serta class diagram untuk merepresentasikan struktur kelas dan relasi antar kelas [22]. Penggunaan UML dalam penelitian ini bertujuan untuk meminimalkan kesalahan pada tahap implementasi dan memastikan bahwa sistem yang dibangun sesuai dengan kebutuhan pengguna. Dengan adanya pemodelan UML, proses pengembangan sistem menjadi lebih sistematis dan mudah dipahami oleh seluruh pihak yang terlibat.

Rapid Application Development (RAD)

Rapid Application Development (RAD) merupakan metode pengembangan perangkat lunak yang menekankan pada kecepatan pengembangan dan keterlibatan pengguna secara aktif dalam proses perancangan dan implementasi sistem [23].

Metode RAD berfokus pada pengembangan sistem secara iteratif dan inkremental, sehingga perubahan kebutuhan dapat segera diakomodasi [24].

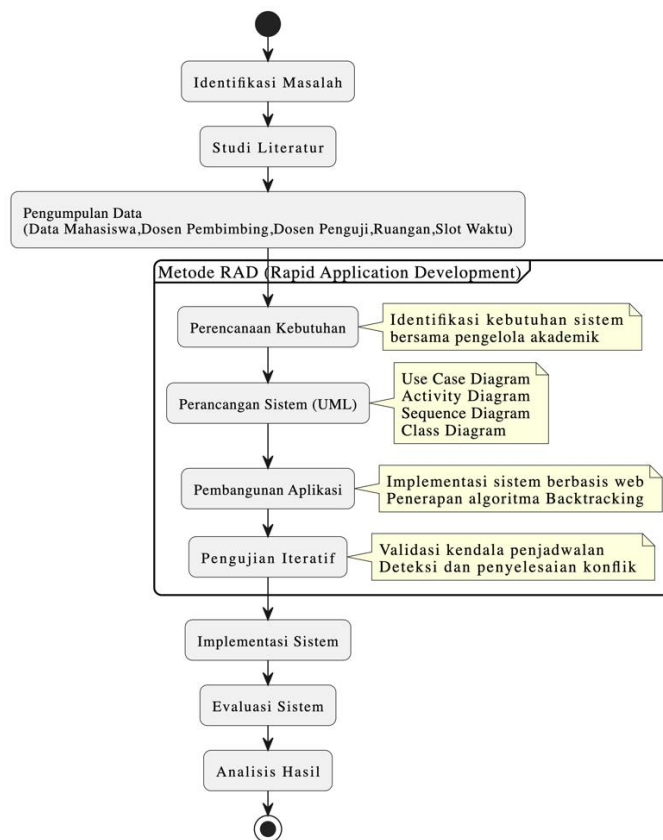


Gambar 1. Tahapan dalam metode RAD

Metode RAD terdiri dari beberapa tahapan utama, yaitu perencanaan kebutuhan, desain sistem, pembangunan aplikasi, dan implementasi [25]. Pada tahap perencanaan kebutuhan, pengembang dan pengguna bekerja sama untuk mengidentifikasi kebutuhan sistem. Tahap desain dilakukan dengan membuat rancangan sistem secara cepat menggunakan alat bantu seperti UML. Tahap pembangunan aplikasi dilakukan dengan mengimplementasikan rancangan ke dalam bentuk sistem yang dapat diuji. Tahap implementasi dilakukan dengan melakukan pengujian dan penerapan sistem ke lingkungan pengguna [6].

3. Metode Penelitian Tahapan Penelitian

Diagram berikut menggambarkan secara sistematis tahapan penelitian yang dilakukan dalam pengembangan sistem penjadwalan sidang skripsi berbasis web menggunakan algoritma backtracking dengan metode Rapid Application Development (RAD).



Gambar 2. Metode penelitian

Pada gambar 2 menunjukkan rangkaian tahapan penelitian yang dimulai dari identifikasi masalah hingga analisis hasil penelitian. Pada tahap identifikasi masalah, dilakukan pengamatan terhadap proses penjadwalan sidang skripsi di Institut Teknologi dan Bisnis Indonesia (ITB Indonesia) untuk menemukan permasalahan yang terjadi, khususnya terkait bentrokan jadwal dan keterbatasan sumber daya.

Tahap selanjutnya adalah studi literatur yang bertujuan untuk memperoleh landasan teoritik yang relevan, meliputi konsep sistem informasi, penjadwalan berbasis kendala, algoritma backtracking, metode Rapid Application Development, serta pemodelan sistem menggunakan UML. Studi literatur ini menjadi dasar dalam merancang sistem

yang sesuai dengan kebutuhan dan permasalahan yang dihadapi. Setelah itu, dilakukan pengumpulan data yang meliputi data mahasiswa, data dosen pembimbing, data dosen penguji, data ruangan, serta data slot waktu sidang. Data tersebut digunakan sebagai input utama dalam proses penjadwalan dan menjadi dasar dalam penerapan algoritma backtracking.

Tahapan pengembangan sistem dilakukan dengan menggunakan metode Rapid Application Development (RAD). Pada tahap perencanaan kebutuhan, peneliti bersama pihak pengelola akademik mengidentifikasi kebutuhan sistem dan menentukan kendala penjadwalan yang harus dipenuhi. Tahap perancangan sistem dilakukan dengan memanfaatkan diagram UML, seperti use case diagram, activity diagram, sequence diagram, dan class diagram, untuk menggambarkan fungsi, alur proses, serta struktur sistem.

Tahap pembangunan aplikasi merupakan tahap implementasi perancangan sistem ke dalam sistem berbasis web. Pada tahap ini, algoritma backtracking diterapkan untuk menghasilkan jadwal sidang skripsi yang memenuhi seluruh kendala yang telah ditentukan. Dilakukan pengujian secara iteratif untuk memastikan bahwa sistem mampu menghasilkan jadwal yang valid dan bebas konflik. Tahap akhir dari metode penelitian adalah implementasi sistem, evaluasi sistem, dan analisis hasil. Implementasi dilakukan dengan menerapkan sistem pada lingkungan pengguna. Evaluasi sistem bertujuan untuk menilai kinerja dan keakuratan hasil penjadwalan. Analisis hasil dilakukan untuk mengetahui sejauh mana sistem yang dikembangkan mampu menyelesaikan permasalahan penjadwalan sidang skripsi di ITB Indonesia.

Pemodelan Masalah Penjadwalan Sidang Skripsi

Permasalahan penjadwalan sidang skripsi pada Institut Teknologi dan Bisnis Indonesia dimodelkan sebagai Constraint Satisfaction Problem (CSP). CSP merupakan permasalahan pencarian solusi yang terdiri atas sekumpulan variabel, domain nilai, dan kendala yang harus dipenuhi secara simultan. Variabel merepresentasikan jadwal sidang setiap mahasiswa, sedangkan domain nilai merupakan kombinasi antara ruangan dan slot waktu yang tersedia. Kendala dibentuk berdasarkan aturan akademik yang berlaku, seperti ketersediaan dosen pembimbing dan dosen penguji, serta penggunaan ruangan.

Misalkan terdapat n mahasiswa yang akan mengikuti sidang skripsi, maka didefinisikan himpunan variabel sebagai berikut:

$$X = x_1, x_2, x_3, \dots, x_n \quad (1)$$

dengan x_i menyatakan jadwal sidang skripsi mahasiswa ke- i . Setiap variabel x_i memiliki domain sebagai berikut.

$$D_i = R \times T \quad (2)$$

dengan $R = r_1, r_2, r_3, \dots, r_m$ dan $T = t_1, t_2, t_3, \dots, t_k$. R adalah himpunan ruangan sidang, T adalah himpunan slot waktu sidang, sehingga setiap nilai variabel memenuhi:

$$x_i \in D_i \quad (3)$$

atau secara eksplisit merupakan pasangan:

$$x_i = (r, t) \quad (4)$$

Agar jadwal sidang yang dihasilkan valid, maka setiap solusi harus memenuhi kendala-kendala berikut: Kendala Ketersediaan Dosen, setiap dosen pembimbing dan dosen penguji harus tersedia pada slot waktu sidang yang dijadwalkan.

$$Availability(d_j, t_k) = 1 \quad (5)$$

Jika nilai fungsi bernilai 0, maka penjadwalan dianggap tidak valid.

Kendala Konflik Waktu Dosen, seorang dosen tidak diperkenankan terlibat dalam lebih dari satu sidang pada slot waktu yang sama.

$$x_i(t_k) \neq x_l(t_k), i \neq l \quad (6)$$

Kendala Konflik Ruang, satu ruangan hanya dapat digunakan untuk satu sidang pada satu slot waktu tertentu.

$$x_i(r_m, t_k) \neq x_l(r_m, t_k), i \neq l \quad (7)$$

Dalam proses pencarian solusi, algoritma backtracking mengevaluasi penjadwalan secara bertahap. Validitas solusi parsial didefinisikan sebagai berikut.

$$Valid(x_1, x_2, \dots, x_k) = \bigwedge_{i=1}^k C(x_i) \quad (8)$$

$C(xi)$ menyatakan seluruh kendala yang harus dipenuhi oleh variabel xi . Apabila pada tahap ke- k terdapat kendala yang dilanggar, maka proses pencarian dihentikan dan algoritma kembali ke tahap sebelumnya untuk mencoba alternatif nilai lain.

Algoritma backtracking bekerja secara rekursif untuk menentukan nilai setiap variabel berdasarkan domain dan kendala yang ada. Formulasi rekursif algoritma backtracking dapat dituliskan sebagai berikut.

$$BT(k) = \bigcup_{v \in D_k} BT(k+1), \text{ for } k < n \text{ and Valid}(x_k = v) \quad (9)$$

Proses ini dilakukan secara berulang hingga seluruh variabel memperoleh nilai yang valid atau seluruh kemungkinan solusi telah dievaluasi.

Kompleksitas waktu algoritma backtracking bersifat eksponensial dan dapat dinyatakan sebagai berikut.

$$T(n) = O(b^n) \quad (10)$$

dengan, n adalah jumlah mahasiswa, b adalah jumlah kemungkinan kombinasi ruangan dan slot waktu. Meskipun kompleksitas teoritis tergolong tinggi, dalam implementasi nyata kompleksitas dapat ditekan melalui pembatasan domain berdasarkan ketersediaan dosen dan ruangan.

4. Hasil Penelitian

Pemodelan Data Penjadwalan Sidang Skripsi

Setiap mahasiswa harus dijadwalkan tepat satu kali sidang dengan melibatkan dua dosen pembimbing dan dua dosen penguji. Data mahasiswa, dosen pembimbing dan penguji seperti tabel 1 berikut:

Tabel 1. Data mahasiswa, dosen pembimbing dan penguji

NIM	Nama Mahasiswa	Pembimbing	Penguji
21040028	Agustinus Aguskrastian Waruwu	Asprina Br Surbakti, Ita Margaretta Raheliya Br Ginting, Safarul Ilham	Roberto Kaban, Meiliyani Br Ginting
21040030	Alvinus Laia	Ilham	Dr. David Sembiring, Roberto Kaban
21040032	Andriono Tumanggor	Dr. Sumarlin, Siti Jamilah	Dr. David Sembiring, Ita Margaretta
21040036	Cindy Angeline	Roberto Kaban, Igamawati	Romulo Aritonang, Polin Ramles
21040038	Dedi Irawan	Giawa	Dr. David Sembiring, Roberto Kaban
21040042	Gregorius Gulo	Dr. David Sembiring, Dewi Yohana	Sinek Mehuli, Siti Jamilah Nirwan Sinuhaji, Meiliyani Br Ginting
...

Dosen berperan sebagai pembimbing dan/atau penguji, serta tidak boleh terjadwal pada dua sidang di waktu yang sama. Tabel 2 berikut adalah data Dosen:

Tabel 2. Ketersediaan Dosen

NIP	Nama Dosen	Program Studi	Status
113128403	Roberto Kaban, M.Kom	Teknik Informatika	Aktif
103127302	Rusty Br Perangin-angin, M.Si	Sistem Informasi	Aktif
110096902	Dr. David Jm Sembiring, M.Kom	Teknik Informatika	Aktif
112058802	Meiliyani Br Ginting, M.Kom	Teknik Informatika	Aktif
...

Setiap ruangan hanya dapat digunakan satu sidang pada satu slot waktu. Tabel 3 berikut adalah data ruangan:

Tabel 3. Ketersediaan ruangan

Kode Ruang	Nama Ruang	Kapasitas	Status
AL4	Aula Lt 4	20 orang	Tersedia

RA101	Ruangan 101	10 orang	Tersedia
RA205	Ruangan 205	10 orang	Tersedia
RA304	Ruangan 304	10 orang	Tersedia
...

Setiap sidang berdurasi 120 menit dan sudah ditentukan slot pelaksanaan sidang skripsi seperti tabel 4 berikut ini:

Tabel 4. Ketersediaan slot pelaksanaan sidang

Kode Slot	Hari	Waktu
S1	Senin	08.00–10.00
S2	Senin	10.00–12.00
S3	Senin	13.00–15.00
S4	Senin	15.00–17.00
S5	Selasa	08.00–10.00
...

Berdasarkan data diatas, pada tahap awal, sistem melakukan inisialisasi dengan mengurutkan mahasiswa berdasarkan tingkat kompleksitas penjadwalan, yaitu jumlah dan ketersediaan dosen yang terlibat. Mahasiswa yang memiliki dosen dengan ketersediaan waktu paling terbatas akan dijadwalkan terlebih dahulu untuk mengurangi potensi konflik di tahap selanjutnya. Setelah proses inisialisasi, sistem melakukan penempatan awal jadwal untuk mahasiswa pertama. Sebagai contoh, mahasiswa Agustinus Aguskrastian Waruwu memiliki dosen pembimbing Asprina dan Ita, serta dosen penguji Roberto dan Meiliyani. Sistem mencoba menempatkan sidang pada slot waktu Senin pukul 08.00 hingga 10.00 dengan menggunakan ruangan RA101. Pada tahap ini, sistem melakukan pengecekan terhadap seluruh kendala, termasuk ketersediaan semua dosen dan status ruangan. Karena seluruh dosen tersedia dan ruangan masih kosong, maka jadwal tersebut diterima sebagai solusi sementara.

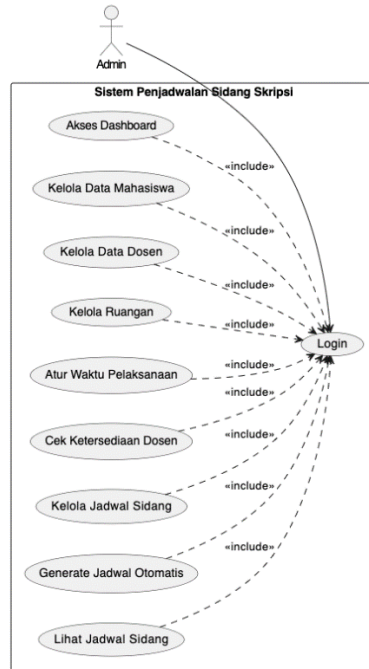
Pada tahap berikutnya, sistem melanjutkan penjadwalan untuk mahasiswa berikutnya, yaitu Alvinus Laia, yang memiliki dosen pembimbing Raheliya dan Safarul, serta dosen penguji David dan Roberto. Apabila sistem mencoba menggunakan slot waktu yang sama, yaitu Senin pukul 08.00 hingga 10.00, maka terjadi konflik karena dosen Roberto Kaban telah terjadwal pada slot tersebut. Konflik ini menyebabkan solusi tidak valid sehingga sistem tidak dapat melanjutkan pada pilihan tersebut. Ketika konflik ditemukan, algoritma melakukan proses *backtracking* dengan membatalkan pilihan slot waktu yang menyebabkan konflik, kemudian mencoba slot waktu alternatif berikutnya, misalnya Senin pukul 10.00 hingga 12.00. Jika pada slot alternatif tersebut seluruh dosen yang terlibat tersedia dan ruangan masih kosong, maka jadwal baru diterima sebagai solusi sementara. Proses ini dilakukan secara berulang dan rekursif hingga seluruh mahasiswa memperoleh jadwal sidang yang valid. Apabila pada suatu tahap tidak ditemukan kombinasi ruang dan waktu yang memenuhi seluruh kendala, maka sistem akan kembali ke solusi sebelumnya dan mencoba alternatif lain sampai solusi yang valid ditemukan atau seluruh kemungkinan telah dievaluasi. Hasil jadwal sidang skripsi seperti yang dihasilkan oleh sistem adalah seperti tabel 5 berikut ini:

Tabel 5. Hasil jadwal sidang skripsi

Mahasiswa	Hari	Waktu	Ruang
Agustinus A. Waruwu	Senin	08.00–10.00	RA101
Alvinus Laia	Senin	10.00–12.00	RA205
Andriono Tumanggor	Selasa	08.00–10.00	RA304
Cindy Angeline	Selasa	10.00–12.00	RA321
...

Use Case

Untuk memberikan gambaran menyeluruh mengenai proses dalam sistem penjadwalan sidang skripsi, berikut disajikan diagram use case sistem. Diagram ini menunjukkan interaksi antara aktor dan fitur-fitur utama yang tersedia dalam sistem.

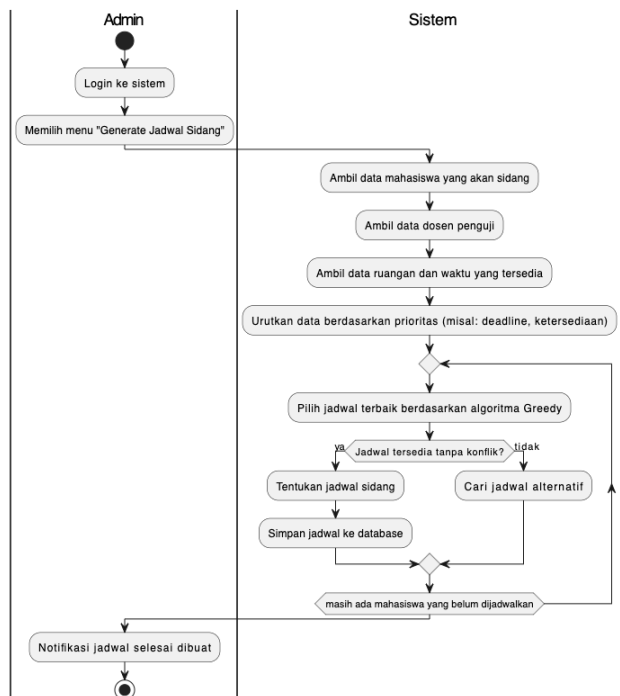


Gambar 3. Use case diagram

Pada diagram tersebut, Admin bertindak sebagai aktor yang berinteraksi langsung dengan sistem. Sebelum mengakses semua fungsi seperti mengelola data mahasiswa, dosen, ruangan, hingga menjadwalkan sidang secara otomatis, Admin terlebih dahulu harus melakukan proses login. Setiap use case yang tersedia memiliki dependensi terhadap proses login ini, yang ditandai dengan relasi include. Hal ini menjamin bahwa seluruh proses dalam sistem hanya dapat dijalankan oleh pengguna yang memiliki hak akses sah. Diagram ini menggambarkan bahwa sistem dirancang terpusat dan terkontrol secara administratif untuk memastikan keamanan dan ketertiban penjadwalan sidang skripsi.

Activity Case diagram Generate jadwal

Berikut adalah activity diagram yang menggambarkan proses generate jadwal sidang skripsi secara otomatis menggunakan algoritma greedy dalam sistem.

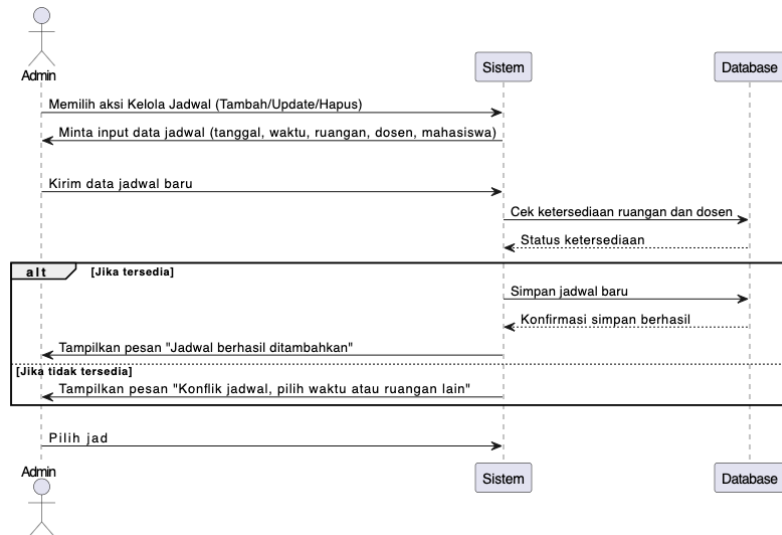


Gambar 4. Activity Case diagram Generate jadwal

Proses dimulai saat admin login ke dalam sistem dan memilih menu generate jadwal sidang. Sistem kemudian mengambil data lengkap mahasiswa yang akan sidang, data dosen penguji, serta informasi ruangan dan waktu yang tersedia. Sistem mengurutkan data tersebut berdasarkan prioritas tertentu, seperti batas waktu sidang atau ketersediaan dosen dan ruangan. Dengan menggunakan algoritma greedy, sistem memilih jadwal terbaik yang memungkinkan tanpa menimbulkan konflik antara dosen, ruangan, dan waktu sidang. Jika jadwal yang dipilih sudah memenuhi syarat tanpa konflik, jadwal tersebut disimpan ke database. Namun jika terjadi konflik, sistem akan mencari alternatif jadwal yang lain. Proses ini akan berulang hingga semua mahasiswa mendapatkan jadwal sidang. Setelah selesai, sistem mengirimkan notifikasi kepada admin bahwa proses generate jadwal telah berhasil dilakukan.

Sequence Pengelolaan jadwal

Sequence diagram ini menggambarkan interaksi antara admin, sistem, dan database saat admin mengelola jadwal sidang skripsi.

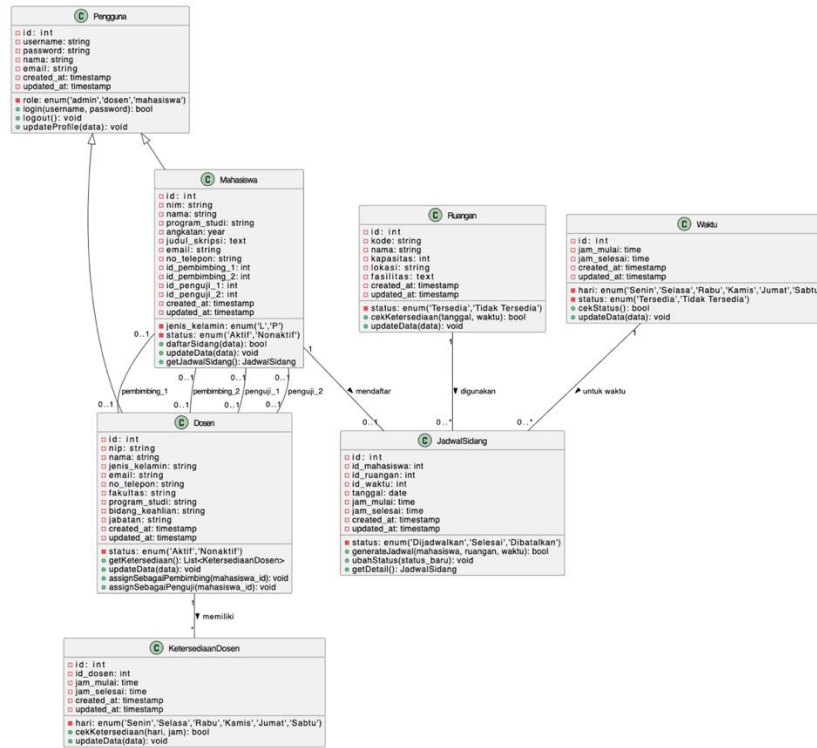


Gambar 5. Sequence diagram pengelolaan jadwal

Proses dimulai ketika admin memilih aksi pengelolaan jadwal, seperti menambah, mengubah, atau menghapus jadwal sidang. Untuk penambahan jadwal, admin mengirimkan data jadwal baru ke sistem, yang kemudian memeriksa ketersediaan ruangan dan dosen melalui database. Jika tersedia, jadwal disimpan dan admin mendapat konfirmasi sukses, tetapi jika ada konflik jadwal, admin diberi tahu untuk memilih waktu atau ruangan lain. Pada proses update, admin memilih jadwal yang ingin diubah, data jadwal diambil dari database dan ditampilkan ke admin untuk diedit. Setelah admin mengirimkan data baru, sistem kembali memeriksa ketersediaan sebelum menyimpan perubahan. Proses hapus jadwal dilakukan dengan admin memilih jadwal yang ingin dihapus, lalu sistem menghapusnya di database dan mengonfirmasi keberhasilan penghapusan.

Class Diagram

Class diagram ini menggambarkan struktur utama sistem pengelolaan sidang skripsi di lingkungan akademik. Pada diagram ini, terdapat kelas Pengguna sebagai kelas induk yang merepresentasikan semua jenis pengguna sistem, seperti dosen dan mahasiswa.



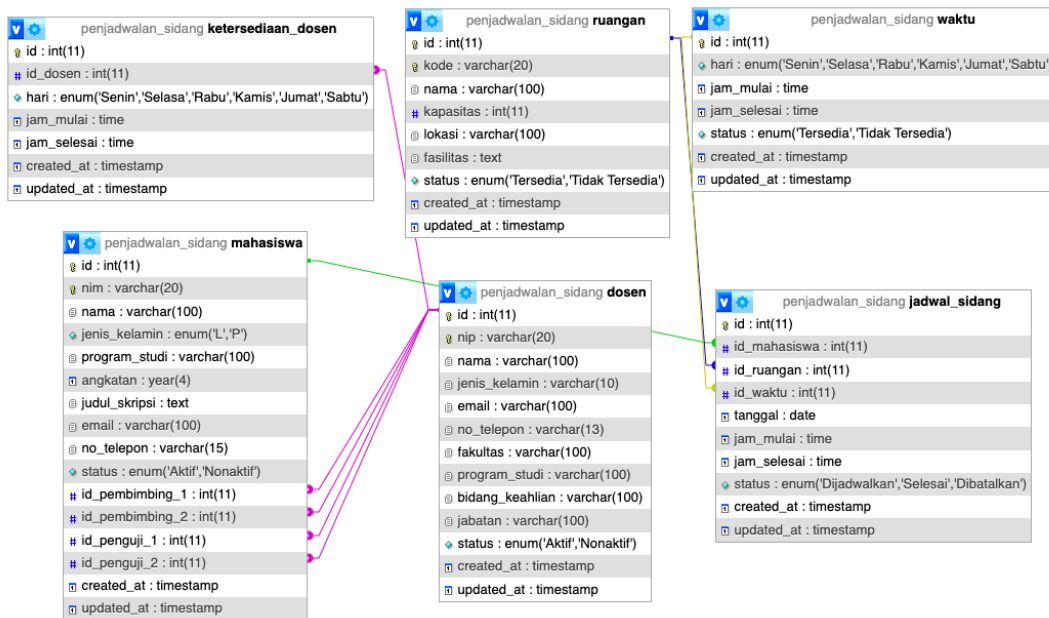
Gambar 6. Class diagram

Kelas Dosen dan Mahasiswa merupakan turunan dari kelas Pengguna, di mana masing-masing memiliki atribut dan fungsi khusus sesuai perannya. Dosen menyimpan data seperti NIP, jabatan, serta bidang keahlian, dan memiliki jadwal ketersediaan yang tersimpan di kelas terpisah bernama KetersediaanDosen. Mahasiswa memiliki atribut terkait data akademik, seperti NIM, program studi, judul skripsi, serta relasi dengan dosen pembimbing dan penguji.

Terdapat kelas Ruang yang menyimpan informasi tentang ruangan sidang, seperti kapasitas dan fasilitas, serta kelas Waktu yang merepresentasikan jadwal waktu sidang dengan detail hari dan jam. Kelas JadwalSidang menghubungkan mahasiswa, ruangan, dan waktu dalam sebuah jadwal sidang yang spesifik, sekaligus menyimpan status dari jadwal tersebut, apakah sudah dijadwalkan, selesai, atau dibatalkan. Sistem ini memungkinkan mahasiswa untuk mendaftar sidang dan mendapatkan jadwal yang terkoordinasi dengan ketersediaan dosen, ruangan, dan waktu yang ada. Relasi antara mahasiswa dengan dosen sebagai pembimbing dan penguji memberikan gambaran proses akademik yang lengkap dalam pelaksanaan sidang skripsi.

Relasi antar tabel dalam database

Sistem yang dibangun menggunakan database MariaDB yang sudah di sertakan pada paket installer XAMPP. Berikut ini adalah relasi antar tabel dalam database pada sistem yang dikembangkan.



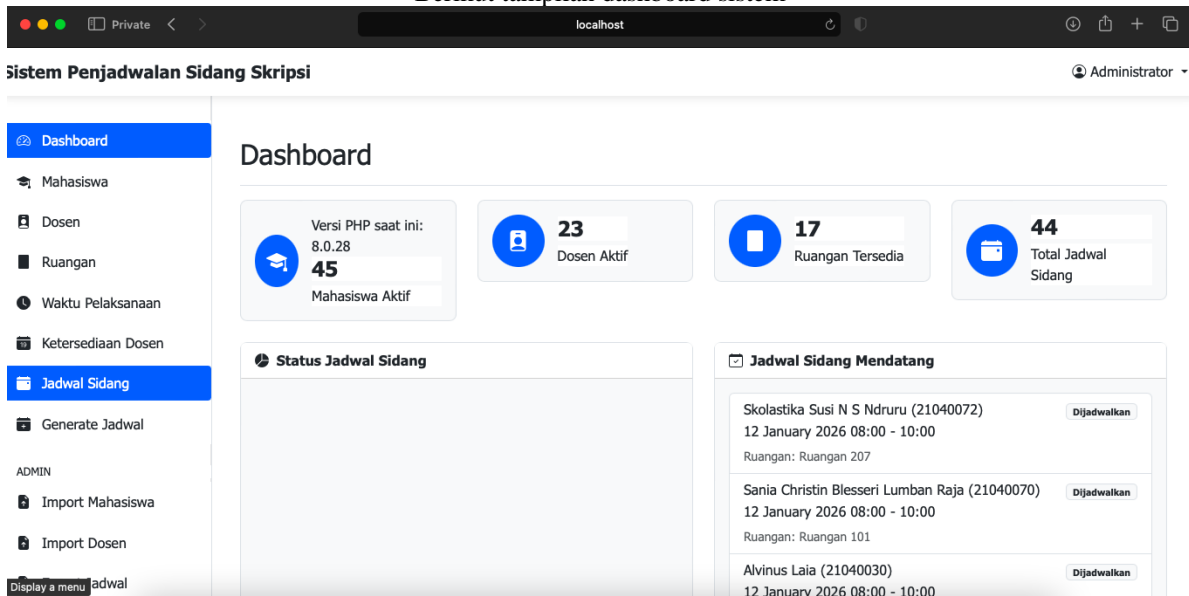
Gambar 7. Relasi antar tabel dalam database

Implementasi sistem

Dashboard sistem

Dashboard sistem menampilkan informasi jumlah mahasiswa, dosen, ruangan dan jadwal aktif pada semester berjalan.

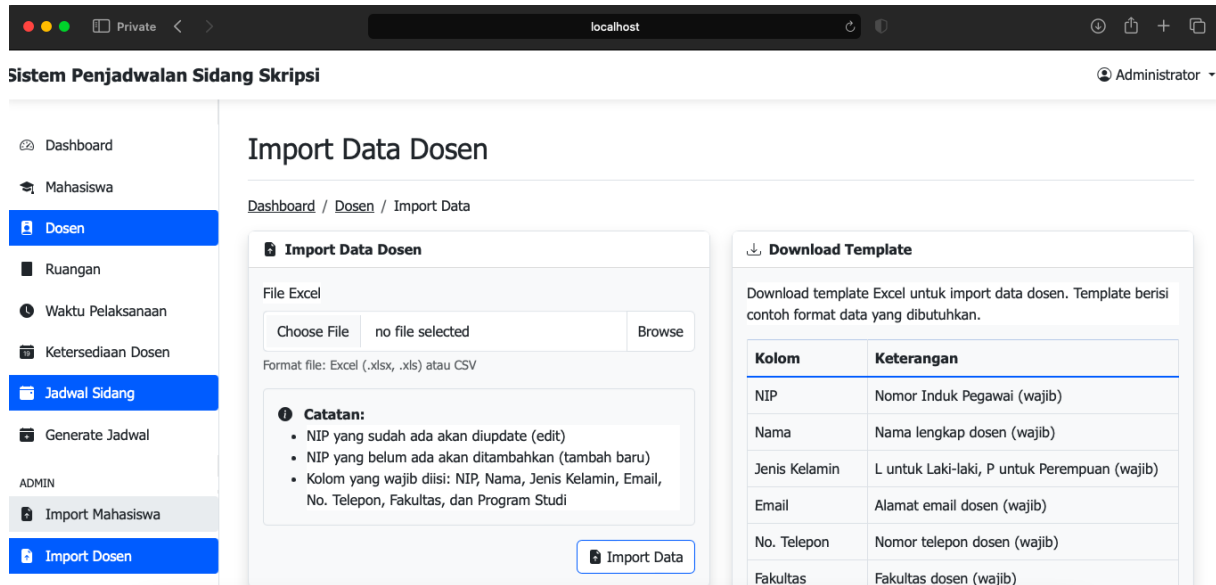
Berikut tampilan dashboard sistem



Gambar 8. Dashboard sistem pada menu admin

Halaman Import data dosen

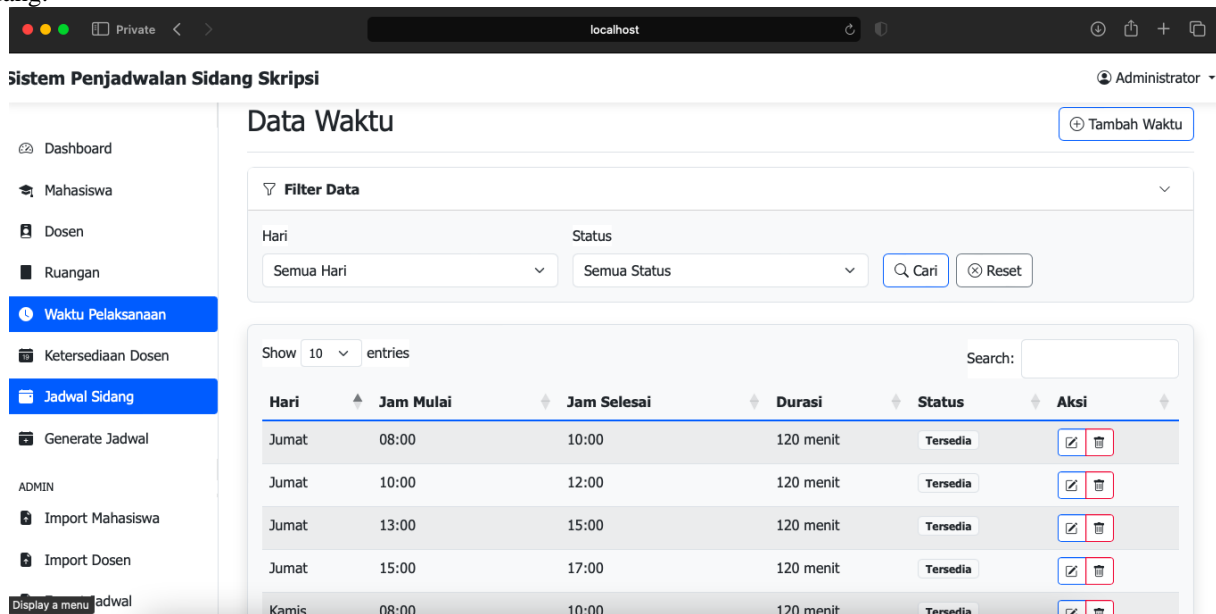
Sistem menyediakan fitur untuk input data dosen melalui file Excel sehingga proses pengelolaan data menjadi lebih cepat. Pengguna cukup mengunggah file sesuai dengan format yang telah ditentukan agar data dapat diproses secara otomatis oleh sistem. Fitur ini dilengkapi dengan validasi struktur dan isi data untuk memastikan tidak terjadi kesalahan format maupun duplikasi informasi. Berikut tampilan import dosen:



Gambar 9. Halaman import data dosen

Halaman jadwal sidang skripsi

Halaman Jadwal Sidang Skripsi menampilkan daftar jadwal sidang mahasiswa secara lengkap berdasarkan tanggal, waktu, ruang, serta dosen pembimbing dan penguji. Halaman ini dilengkapi fitur pencarian dan penyaringan data untuk memudahkan pengguna dalam menemukan informasi jadwal secara cepat dan akurat. Berikut tampilan halaman jadwal sidang:



Gambar 10. Halaman informasi jadwal sidang

Halaman ketersediaan waktu dosen

Halaman Ketersediaan Waktu Dosen digunakan untuk mengelola dan menampilkan jadwal waktu luang dosen yang dapat digunakan dalam proses penjadwalan sidang atau perkuliahan. Melalui halaman ini, dosen atau administrator dapat mengatur slot waktu tersedia sehingga sistem dapat menghindari benturan jadwal dan memastikan proses penjadwalan berjalan lebih efektif. Berikut tampilan ketersediaan waktu dosen:

Ketersediaan Dosen

Filter Data

Dosen: -- Semua Dosen -- Hari: -- Semua Hari -- Filter Reset

Nama Dosen	Hari	Jam Mulai	Jam Selesai	Durasi (menit)	Aksi
Asprina Br Surbakti, M.Kom	Senin	08:00	17:00	540	[Edit] [Hapus]
	Selasa	08:00	17:00	540	[Edit] [Hapus]
	Rabu	08:00	17:00	540	[Edit] [Hapus]
	Kamis	08:00	17:00	540	[Edit] [Hapus]
	Jumat	08:00	17:00	540	[Edit] [Hapus]

Gambar 11. Halaman informasi ketersediaan dosen

Halaman generate jadwal sidang

Halaman Generate Jadwal Sidang digunakan untuk membuat jadwal sidang secara otomatis berdasarkan data mahasiswa, ketersediaan dosen, dan ketersediaan ruang. Sistem akan memproses parameter yang dipilih dan menghasilkan susunan jadwal yang terstruktur guna meminimalkan benturan waktu serta mengoptimalkan penggunaan sumber daya. Berikut tampilan halaman generate jadwal sidang:

Generate Jadwal Sidang

Dashboard / Jadwal Sidang / Generate Jadwal

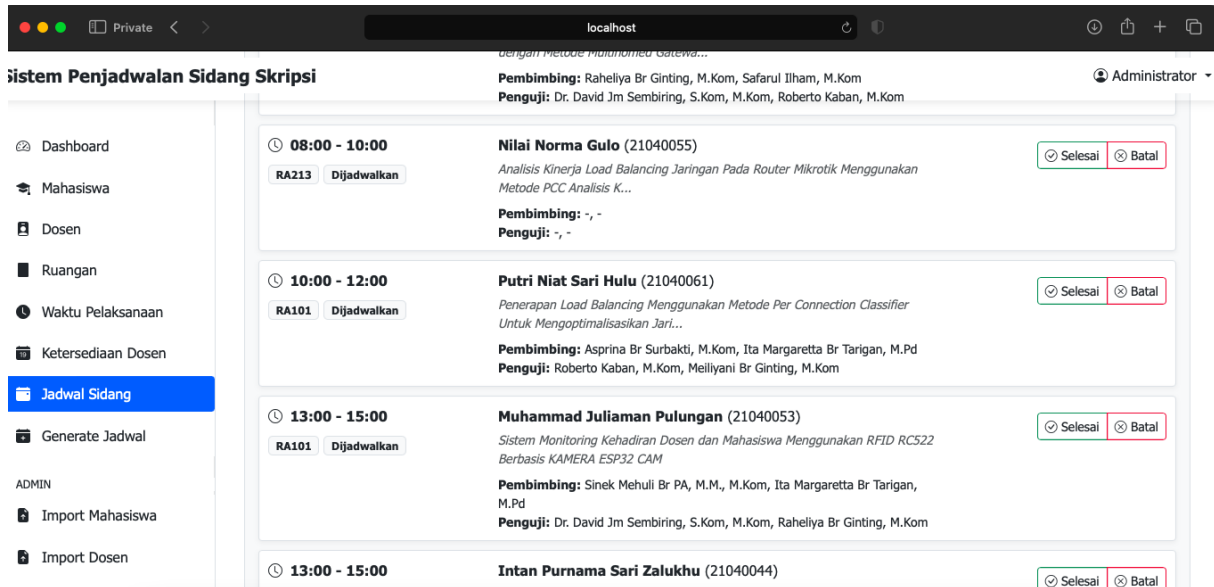
Statistik Data untuk Penjadwalan

- 1 Mahasiswa Belum Dijadwalkan
- 44 Jadwal Sudah Terdaftar
- 23 Dosen Tersedia
- 17 Ruangan Tersedia
- 22 Slot Waktu Tersedia

Gambar 12. Halaman generate jadwal sidang

Halaman Informasi pelaksanaan sidang

Halaman Informasi Pelaksanaan Sidang menampilkan detail jadwal sidang yang mencakup tanggal, waktu, ruang, serta daftar dosen pembimbing dan penguji. Halaman ini dirancang untuk memberikan informasi yang jelas dan terstruktur sehingga mahasiswa dan dosen dapat mempersiapkan pelaksanaan sidang dengan baik. Berikut tampilan halaman pelaksanaan sidang:



Gambar 13. Informasi pelaksanaan sidang

Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode Blackbox untuk memastikan setiap fungsi berjalan sesuai dengan kebutuhan pengguna dan spesifikasi yang telah ditetapkan. Tabel pengujian sistem dengan Blackbox menyajikan skenario pengujian, langkah pengujian, hasil yang diharapkan, serta status keberhasilan dari masing-masing fitur yang diuji.

Tabel 6. Hasil pengujian sistem

Fitur	Input	Proses	Output yang Diharapkan	Hasil
Dashboard	Login user valid / Login user tidak valid	Klik menu Dashboard Sistem	Jika valid: ringkasan tampil, jika tidak valid: tampil pesan error / redirect login	Pass / Fail
Mahasiswa	Klik menu, filter nama mahasiswa ada / tidak ada	menampilkan daftar mahasiswa sesuai filter	Jika ada: tampil mahasiswa, jika tidak ada: tampil pesan "Data tidak ditemukan"	Pass
Dosen	Klik menu, cari dosen berdasarkan NIDN / NIDN tidak ada	Sistem menampilkan data dosen	Jika ada: data muncul, jika tidak: tampil pesan "Dosen tidak ditemukan"	Pass
Ruangan	Klik menu, pilih ruangan aktif / ruangan penuh	Sistem menampilkan ruangan sesuai kondisi	Ruangan tersedia: tampil, ruangan penuh: tampil status "Penuh"	Pass
Waktu Pelaksanaan	Pilih tanggal sidang valid / tanggal libur	Sistem menampilkan waktu yang tersedia	Jika valid: tampil jadwal, jika libur: tampil pesan "Tidak tersedia"	Pass
Ketersediaan Dosen	Pilih dosen tersedia / dosen sedang ada jadwal lain	Sistem cek ketersediaan	Dosen tersedia: tampil hijau, jika tidak: tampil merah / pesan error	Pass
Jadwal Sidang	Klik menu, filter jurusan / filter tanggal	Sistem menampilkan jadwal sidang sesuai filter	Filter berhasil: tampil jadwal sesuai, jika data kosong: tampil pesan "Tidak ada jadwal"	Pass
Generate Jadwal	Data lengkap / data tidak lengkap (misal: dosen kosong)	Sistem generate jadwal otomatis	Data lengkap: jadwal muncul, data tidak lengkap: tampil pesan error	Pass / Fail
ADMIN	Login admin valid / user non-admin	Klik menu ADMIN	Admin: submenu muncul, user non-admin: akses ditolak	Pass / Fail
Import Mahasiswa	File CSV/Excel valid / file corrupt / format salah	Klik Import Mahasiswa	File valid: data masuk, file corrupt: pesan error	Pass / Fail

Import Dosen	File CSV/Excel valid / file kosong / format salah	Klik Import Dosen	File valid: data masuk, file kosong: pesan "File kosong", format salah: pesan error	Pass / Fail
--------------	---	-------------------	---	-------------

5. Kesimpulan

Sistem yang dikembangkan mampu memodelkan proses penjadwalan sidang skripsi sebagai permasalahan penjadwalan berbasis kendala secara terstruktur dan sistematis. Penerapan algoritma backtracking terbukti mampu menghasilkan jadwal sidang skripsi yang valid dan bebas konflik, terutama dalam menghindari bentrokan jadwal dosen dan penggunaan ruangan pada waktu yang sama. Algoritma ini mengevaluasi setiap kemungkinan penjadwalan berdasarkan kendala yang telah ditetapkan sehingga jadwal yang dihasilkan sesuai dengan aturan akademik yang berlaku. Metode Rapid Application Development (RAD) juga memberikan kontribusi positif dalam pengembangan sistem melalui pendekatan iteratif dan keterlibatan aktif pengguna, sehingga sistem dapat dikembangkan secara fleksibel dan sesuai dengan kebutuhan pengelola akademik. Dengan diterapkannya sistem penjadwalan sidang skripsi berbasis web, proses penyusunan jadwal di ITB Indonesia dapat dilakukan secara lebih efisien, akurat, dan terintegrasi. Sistem ini diharapkan mampu mengurangi kesalahan penjadwalan, mempercepat penentuan jadwal sidang, serta meningkatkan kualitas layanan akademik.

Sebagai saran untuk pengembangan selanjutnya, sistem yang telah dibangun dapat ditingkatkan dengan menambahkan fitur penyesuaian jadwal secara otomatis apabila terjadi perubahan ketersediaan dosen atau ruangan. Pengembangan lebih lanjut juga dapat dilakukan dengan mengombinasikan algoritma backtracking dengan algoritma lain, seperti algoritma heuristik atau algoritma optimasi, guna meningkatkan efisiensi penjadwalan pada skala data yang lebih besar. Integrasi sistem dengan sistem informasi akademik yang telah ada serta pengembangan ke platform mobile dapat dipertimbangkan untuk meningkatkan aksesibilitas dan kemudahan penggunaan sistem.

Daftar Pustaka

- [1] R. Lavhe, "Academic Management through a Web-Based Student Profile System," *IJRASET*, vol. 13, no. 5, pp. 7347–7354, May 2025, doi: 10.22214/ijraset.2025.71890.
- [2] A. N. Wulandari, D. Rosadi, E. Rahmawati, and A. T. Utami, "Web-Based Thesis Title Submission Information System Using Codeigniter Framework Case Study at STMIK Mardira Indonesia," *ebmtj*, vol. 3, no. 1, pp. 31–40, Aug. 2025, doi: 10.55208/n4p5gb28.
- [3] M. Comlan and C. Allohoumbo, "Constraint satisfaction algorithms: edition of timetables in the license-master-doctorate system," *Comput. Sci. Inf. Technol.*, vol. 4, no. 3, pp. 217–225, Nov. 2023, doi: 10.11591/csit.v4i3.p217-225.
- [4] P. P. M and J. R. Mary, "Graph Colouring, Class Scheduling and Application in Modern Computer Science," *IJRASET*, vol. 11, no. 3, pp. 1065–1069, Mar. 2023, doi: 10.22214/ijraset.2023.49590.
- [5] A. Ardiansyah, S. J. Kuryanti, E. A. Pratama, and R. A. Anggraini, "PERANCANGAN SISTEM INFORMASI PENGGAJIAN PADA JASA DIGITAL PRINTING DENGAN METODE RAPID APPLICATION DEVELOPMENT," *conten*, vol. 2, no. 2, pp. 118–126, Dec. 2022, doi: 10.31294/conten.v2i2.1720.
- [6] R. Kaban, D. J. Sembiring, and I. M. B. Tarigan, "Monitoring System for Student Internships Using the Rapid Application Development (RAD) Method".
- [7] S. Suhardi, A. R. Rizqiah, N. Ichsan, R. Ratnawati, L. Hakim, and S. Ayumida, "SISTEM INFORMASI ALUMNI BERBASIS WEB DENGAN METODE RAPID APPLICATION DEVELOPMENT," *Technologia*, vol. 15, no. 1, p. 161, Jan. 2024, doi: 10.31602/tji.v15i1.13587.
- [8] M. F. Adham, "Analisis Implementasi Sistem Informasi: Studi Literatur," *JTSI*, vol. 5, no. 1, pp. 264–275, Jun. 2024, doi: 10.35957/jtsi.v5i1.7815.
- [9] M. Farid, I. Safitri, N. Nabila, E. S. Putri, and M. T. I. Rahmayani, "Perancangan Sistem Informasi Pemesanan Tiket Travel Bengkulu (GOTralis) Berbasis Android," *JUTIN*, vol. 7, no. 1, pp. 351–358, Jan. 2024, doi: 10.31004/jutin.v7i1.22655.
- [10] R. Kaban and D. J. Sembiring, *PEMROGRAMAN WEB INTERAKTIF DENGAN JAVASCRIPT*. Karya Techno Solusindo, 2025. [Online]. Available: https://www.google.co.id/books/edition/PEMROGRAMAN_WEB_INTERAKTIF_DENGAN_JAVASC/inRjEQAAQBAJ?hl=id&gbpv=0
- [11] N. Y. D. Setyaningsih, R. Ningsih, and B. C. Wibowo, "Sistem Informasi Presensi Digital Menggunakan Validasi E-KTP," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 1, pp. 1–11, May 2022, doi: 10.12928/biste.v4i1.5978.
- [12] S. Sintaro, "Permodelan Sistem Informasi Pembelian dan Penjualan Berbasis Website," *jima-ilkom*, vol. 1, no. 1, pp. 25–32, Mar. 2022, doi: 10.58602/jima-ilkom.v1i1.5.
- [13] E. Yanuarti, S. Sarwindah, E. B. Perkasa, and A. Briliantza, "PENERAPAN MODEL RAD DALAM SISTEM ADMINISTRASI LAYANAN TV KABEL BERBASIS WEB," *JIKA*, vol. 6, no. 3, p. 220, Oct. 2022, doi: 10.31000/jika.v6i3.6229.

- [14] A. Suwondo, A. W. Mansur, and Mardinawati, "PENERAPAN EXTREME PROGRAMMING DALAM PENGEMBANGAN SISTEM INFORMASI PENJADWALAN KULIAH JURUSAN AKUNTANSI POLINES," *JTIKP*, vol. 14, no. 1, pp. 1–7, Mar. 2023, doi: 10.51903/jtikp.v14i1.341.
- [15] F. P. Diallo and C. Tudose, "Optimizing the Scheduling of Teaching Activities in a Faculty," *Applied Sciences*, vol. 14, no. 20, p. 9554, Oct. 2024, doi: 10.3390/app14209554.
- [16] X. Han and D. Wang, "Gradual Optimization of University Course Scheduling Problem Using Genetic Algorithm and Dynamic Programming," *Algorithms*, vol. 18, no. 3, p. 158, Mar. 2025, doi: 10.3390/a18030158.
- [17] C. Juvin, E. Hebrard, L. Houssin, and P. Lopez, "An Efficient Constraint Programming Approach to Preemptive Job Shop Scheduling," *LIPICs, Volume 280, CP 2023*, vol. 280, p. 19:1-19:16, 2023, doi: 10.4230/LIPICS.CP.2023.19.
- [18] Angel Caroline Billan and Tata Sutabri, "Restorasi Penjadwalan Sumur Minyak Yang Mengalami Off-Time Menggunakan Algoritma Backtracking Dalam Upaya Optimasi Produksi," *bulletincsr*, vol. 5, no. 3, pp. 228–234, Apr. 2025, doi: 10.47065/bulletincsr.v5i3.507.
- [19] K. Zhang, Y. Dong, L. Cheng, X. Chen, Q. Cheng, and Z. Liu, "A Backtracking Algorithm for Solving the Nearly Equitable Strong Edge-coloring Problem on Transportation Network," *Netw Spat Econ*, vol. 25, no. 1, pp. 219–248, Mar. 2025, doi: 10.1007/s11067-024-09661-z.
- [20] F. A. Feichas and R. D. Seabra, "Evaluation of Perception of Use of a Gamified Platform from the Student Perspective: An Approach for Studying Unified Modeling Language," *Informatics in Education*, Jan. 2023, doi: 10.15388/infedu.2023.22.
- [21] Dr. S. Suriya and N. S., "Design of UML Diagrams for WEBMED - Healthcare Service System Services," *EAI Endorsed Trans e-Learn*, vol. 8, no. 1, p. e5, Feb. 2023, doi: 10.4108/eetel.v8i1.3015.
- [22] A. Gadhi, R. M. Gondu, C. M. Bandaru, K. C. Reddy, and O. Abiona, "Applying UML and Machine Learning to Enhance System Analysis and Design," *IJCNS*, vol. 16, no. 05, pp. 67–76, 2023, doi: 10.4236/ijcns.2023.165005.
- [23] Y. A. Singgalen, "Penerapan Metode TOPSIS Sebagai Pendukung Keputusan Pemilihan Layanan Akomodasi di Destinasi Wisata Pulau," *mib*, vol. 7, no. 3, p. 1386, Jul. 2023, doi: 10.30865/mib.v7i3.6530.
- [24] M. Melvin, J. Wiratama, R. Sutomo, and S. A. Sanjaya, "A Web-based Point of Sales for Automotive Component Industry using Rapid Application Development model," *Journal of Information System*, vol. 8, no. 2, pp. 167–176, Nov. 2023, doi: 10.33633/joins.v8i2.9383.
- [25] A. A. Shaker *et al.*, "Facilitating In-House Mobile App Development Within Psychiatric Outpatient Services for Patients Diagnosed With Borderline Personality Disorder: Rapid Application Development Approach," *JMIR Hum Factors*, vol. 10, p. e46928, Nov. 2023, doi: 10.2196/46928.