

**RANCANG BANGUN SISTEM INFORMASI
PENJADWALAN PEMBELAJARAN OTOMATIS
BERBASIS WEB MENGGUNAKAN *CONSTRAINT*
*PROGRAMMING***

PROYEK AKHIR

Disusun Oleh:

Elvan

3312311004

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya Teknik
Informatika



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM**

2026

KATA PENGANTAR

Puji syukur terhadap Tuhan Yang Maha Esa yang selalu memberi rahmat-Nya sehingga penulis dapat menyelesaikan proyek akhir dengan judul “Rancang Bangun Sistem Informasi Penjadwalan Pembelajaran Otomatis Berbasis Web Menggunakan *Constraint Programming*”. Dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Sismanto dan Ibu Min Hun sebagai Orang tua yang selalu menemani dan memberi dukungan selama membuat proyek akhir.
2. Bapak Muhammad Idris, S.Tr., M.Tr.Kom., sebagai dosen pembimbing.
3. Semua pihak yang membantu baik secara langsung maupun tidak langsung sehingga penyusunan proyek akhir dapat terselesaikan.

Laporan ini ditulis untuk memenuhi syarat dalam pengerjaan proyek akhir. Penulis berharap laporan ini dapat memberikan manfaat serta menjadi bahan referensi yang berguna bagi pembaca dalam memahami perencanaan dan pelaksanaan proyek akhir ini. Akhir kata penulis juga mengharapkan saran dan kritik yang membangun dari pembaca demi perbaikan dan penyempurnaan laporan ini.

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI	iii
DAFTAR TABEL	iv
DAFTAR GAMBAR	v
DAFTAR LAMPIRAN.....	vii
ABSTRAK.....	viii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan	3
1.4. Batasan Masalah	3
1.5. Manfaat	3
1.6. Kolaborasi dan Sinergi dalam Pelaksanaan Proyek.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. Penelitian Terkait	6
2.2. Landasan Teori	7
2.3. Metode Pengembangan Produk	10
BAB III ANALISIS DAN PERANCANGAN	13
3.1. Analisis Kebutuhan.....	13
3.2. Perancangan	15
BAB IV IMPLEMENTASI DAN PENGUJIAN	57
4.1. Hasil Implementasi	57
4.2. Pengujian.....	68
BAB V KESIMPULAN.....	72
5.1. Kesimpulan	72
5.2. Saran	72
DAFTAR PUSTAKA.....	73
DAFTAR LAMPIRAN	75

DAFTAR TABEL

Tabel 1. Pembagian Peran dan Tanggung Jawab	4
Tabel 2. Perbandingan Penelitian Sebelumnya	6
Tabel 3. Kebutuhan Fungsional	15
Tabel 4. Kebutuhan Non-Fungsional	16
Tabel 5. Skenario <i>Use Case</i> Buat Akun	18
Tabel 6. Skenario <i>Use Case Login</i>	19
Tabel 7. Skenario <i>Use Case Logout</i>	20
Tabel 8. Skenario <i>Use Case</i> Mengelola Hari	20
Tabel 9. Skenario <i>Use Case</i> Mengelola Jam Pembelajaran	22
Tabel 10. Skenario <i>Use Case</i> Mengelola Kelas	23
Tabel 11. Skenario <i>Use Case</i> Mengelola Ruang Kelas.....	24
Tabel 12. Skenario <i>Use Case</i> Mengelola Pengajar	25
Tabel 13. Skenario <i>Use Case</i> Mengelola Pelajaran	26
Tabel 14. Skenario <i>Use Case</i> Menentukan Batasan.....	28
Tabel 15. Skenario <i>Use Case</i> Membuat Jadwal.....	28
Tabel 16. Skenario <i>Use Case Edit</i> Jadwal.....	30
Tabel 17. Skenario <i>Use Case Export</i> Jadwal (Admin).....	30
Tabel 18. Skenario <i>Use Case</i> Melihat Jadwal (Admin)	31
Tabel 19. Tabel Pengujian	68

DAFTAR GAMBAR

Gambar 1. Metode <i>Waterfall</i>	10
Gambar 2. Gambaran Umum Sistem	14
Gambar 3. Diagram <i>Use Case</i>	17
Gambar 4. <i>Activity Diagram</i> Buat Akun.....	33
Gambar 5. <i>Activity Diagram</i> Login.....	34
Gambar 6. <i>Activity Diagram</i> Logout.....	34
Gambar 7. <i>Activity Diagram</i> Mengelola Hari.....	35
Gambar 8. <i>Activity Diagram</i> Mengelola Jam Pembelajaran.....	36
Gambar 9. <i>Activity Diagram</i> Mengelola Kelas.....	37
Gambar 10. <i>Activity Diagram</i> Mengelola Ruang Kelas	38
Gambar 11. <i>Activity Diagram</i> Mengelola Pengajar	39
Gambar 12. <i>Activity Diagram</i> Mengelola Pelajaran	40
Gambar 13. <i>Activity Diagram</i> Menentukan Batasan	41
Gambar 14. <i>Activity Diagram</i> Membuat Jadwal.....	41
Gambar 15. <i>Activity Diagram</i> Edit Jadwal	42
Gambar 16. <i>Activity Diagram</i> Export Jadwal	42
Gambar 17. <i>Activity Diagram</i> Melihat Jadwal	43
Gambar 18. <i>Entity Relationship Diagram</i>	44
Gambar 19. <i>Wireframe</i> Home	45
Gambar 20. <i>Wireframe</i> Login.....	46
Gambar 21. <i>Wireframe</i> Daftar Akun	46
Gambar 22. <i>Wireframe</i> Data Pengguna.....	47
Gambar 23. <i>Wireframe</i> Data Profil	47
Gambar 24. <i>Wireframe</i> Dashboard.....	48
Gambar 25. <i>Wireframe</i> Data Hari	48
Gambar 26. <i>Wireframe</i> Data Jam Pembelajaran	49
Gambar 27. <i>Wireframe</i> Data Kelas	50
Gambar 28. <i>Wireframe</i> Data Ruang Kelas	51
Gambar 29. <i>Wireframe</i> Data Pengajar	52
Gambar 30. <i>Wireframe</i> Data Pelajaran.....	52

Gambar 31. <i>Wireframe</i> Data Jadwal	53
Gambar 32. <i>Wireframe</i> Buat Jadwal	54
Gambar 33. <i>Wireframe</i> Detail Jadwal	55
Gambar 34. Antarmuka <i>Home</i>	57
Gambar 35. Halaman Daftar	58
Gambar 36. Halaman <i>Login</i>	58
Gambar 37. Halaman Dasbor	59
Gambar 38. Halaman Data Profil.....	59
Gambar 39. Halaman Data Hari.....	60
Gambar 40. Halaman Data Jam	60
Gambar 41. Halaman Data Kelas.....	61
Gambar 42. Halaman Data Ruang	61
Gambar 43. Halaman Data Pengajar.....	62
Gambar 44. Halaman Data Pelajaran.....	62
Gambar 45. Halaman Tambah Data	63
Gambar 46. Halaman Ubah Data	63
Gambar 47. Halaman Hapus Data	64
Gambar 48. Halaman Jadwal	64
Gambar 49. Halaman Tambah Jadwal	65
Gambar 50. Halaman Tambah Data Jadwal.....	65
Gambar 51. Halaman Tambah Batasan Data Jadwal	66
Gambar 52. Halaman Data yang Ditambah	66
Gambar 53. Halaman Data Jadwal.....	67
Gambar 54. Tampilan <i>PDF</i> Jadwal.....	67

DAFTAR LAMPIRAN

Lampiran A. Link Produk.....	75
------------------------------	----

ABSTRAK

Penjadwalan yang baik sangat berpengaruh terhadap seluruh kegiatan belajar-mengajar. Terkadang jadwal yang dibuat manual masih terdapat beberapa bentrokan yang menyebabkan jadwal tersebut perlu dilakukan perbaikan. Di era modern sekarang, proses penjadwalan juga bisa dibantu teknologi dengan merancang sistem informasi penjadwalan yang mampu merancang jadwal yang kompleks dengan lebih cepat dan minim kesalahan. Proyek ini dikembangkan dengan metode *waterfall* dan algoritma *constraint programming* sebagai metode penyelesaian masalahnya. Hasil akhir dari proyek ini adalah sistem informasi berbasis web yang mampu membuat jadwal pembelajaran secara otomatis. Manfaat dari proyek ini adalah menghemat waktu pengerjaan, mengurangi kesalahan dalam membuat jadwal, dan data-data inputan yang dapat digunakan kembali untuk membuat jadwal lain.

Kata Kunci: Penjadwalan, Aplikasi Web, *Constraint Programming*

BAB I PENDAHULUAN

1.1. Latar Belakang

Menurut Undang-Undang Republik Indonesia Nomor 20 Tahun 2003, pendidikan merupakan usaha sadar dan terencana untuk mewujudkan suasana belajar dan proses pembelajaran agar peserta didik secara aktif mengembangkan potensi dirinya. Agar proses pembelajaran dapat berlangsung secara efektif, lembaga pendidikan memerlukan jadwal pembelajaran yang mampu mengatur kegiatan belajar mengajar secara terstruktur sehingga penggunaan waktu, tenaga pengajar, ruang kelas, dan sumber daya lainnya dapat berjalan dengan baik.

Penyusunan jadwal pembelajaran merupakan salah satu permasalahan yang kompleks dalam pengelolaan lembaga pendidikan. Sebuah jadwal harus memenuhi berbagai batasan (*constraint*) secara bersamaan, seperti seorang tenaga pengajar tidak dapat mengajar lebih dari satu kelas pada waktu yang sama, setiap kelas hanya dapat mengikuti satu mata pelajaran dalam satu waktu, serta jumlah jam pembelajaran setiap mata pelajaran harus terpenuhi. Selain itu, terdapat pula batasan lain yang dapat berbeda pada setiap lembaga pendidikan, seperti ketersediaan waktu tenaga pengajar, penggunaan ruang tertentu untuk mata pelajaran tertentu, maupun kebijakan akademik lainnya. Semakin banyak jumlah kelas, tenaga pengajar, ruang kelas, dan mata pelajaran yang dikelola, maka semakin tinggi pula tingkat kompleksitas penyusunan jadwal.

Pada banyak lembaga pendidikan, penyusunan jadwal pembelajaran masih dilakukan secara manual menggunakan *Microsoft Excel* maupun aplikasi *spreadsheet* lainnya (Fadel et al., 2023; Hilmi et al., 2024; Lestari et al., 2023). Proses tersebut memerlukan waktu yang cukup lama karena staf harus menyusun sekaligus memeriksa setiap kemungkinan bentrokan jadwal secara manual. Ketika terjadi perubahan data, seperti pergantian tenaga pengajar, penambahan kelas, atau perubahan alokasi jam pembelajaran, jadwal yang telah dibuat sering kali harus disusun kembali sehingga meningkatkan risiko terjadinya konflik jadwal dan memperlambat proses persiapan kegiatan belajar mengajar.

Selain proses penyusunan jadwal yang kompleks, kebutuhan penjadwalan pada setiap lembaga pendidikan juga tidak selalu sama. Perbedaan jumlah kelas,

tenaga pengajar, ruang kelas, mata pelajaran, maupun aturan penjadwalan menyebabkan sistem yang bersifat kaku sulit diterapkan pada berbagai institusi. Oleh karena itu, diperlukan suatu sistem yang tidak hanya mampu menghasilkan jadwal secara otomatis, tetapi juga memberikan fleksibilitas kepada pengguna untuk mengelola data serta menentukan batasan penjadwalan sesuai dengan kebutuhan masing-masing lembaga pendidikan.

Salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan penjadwalan adalah *Constraint Programming* (CP). *Constraint Programming* merupakan pendekatan yang memodelkan permasalahan sebagai *Constraint Satisfaction Problem* (CSP), yaitu mencari solusi yang memenuhi seluruh batasan yang telah ditentukan. Dengan pendekatan ini, sistem dapat menghasilkan jadwal secara otomatis tanpa melanggar *constraint* yang diberikan, sehingga waktu penyusunan jadwal dapat dipersingkat dan kemungkinan terjadinya konflik jadwal dapat diminimalkan.

Berdasarkan permasalahan tersebut, pada penelitian ini dirancang sebuah sistem informasi penjadwalan pembelajaran otomatis berbasis web menggunakan metode *Constraint Programming*. Sistem yang dikembangkan memungkinkan staf pendidikan mengelola data *master* penjadwalan, menentukan *constraint* sesuai kebutuhan lembaga pendidikan, menghasilkan jadwal pembelajaran secara otomatis, serta mengekspor hasil jadwal agar dapat didistribusikan kepada tenaga pengajar maupun peserta didik. Dengan adanya sistem ini, diharapkan proses penyusunan jadwal menjadi lebih cepat, fleksibel, dan efisien dibandingkan dengan proses penyusunan secara manual.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, terdapat beberapa permasalahan yaitu:

1. Bagaimana membantu proses penjadwalan kegiatan belajar-mengajar secara otomatis?
2. Bagaimana membangun sistem penjadwalan pembelajaran yang dapat melayani kebutuhan penjadwalan di berbagai lembaga pendidikan secara *online*?

3. Bagaimana menyelesaikan masalah penjadwalan yang memiliki banyak *constraint* seperti ketersediaan ruang kelas, tenaga pengajar, dan waktu?

1.3. Tujuan

Adapun tujuan dari pembuatan sistem ini adalah:

1. Membangun sistem informasi berbasis web yang dapat membuat jadwal pembelajaran secara otomatis.
2. Membangun sistem penjadwalan berbasis web untuk lembaga pendidikan.
3. Menerapkan metode *Constraint Programming* dalam menghasilkan jadwal yang memenuhi berbagai *constraint* yang ditentukan.

1.4. Batasan Masalah

Berikut adalah batasan-batasan dari sistem yang akan dikembangkan:

1. Entitas yang tersedia pada sistem merujuk pada sistem pendidikan di Indonesia, yaitu entitas kelas, hari, jam pembelajaran, ruang, pengajar dan pelajaran.
2. Permasalahan diselesaikan menggunakan *constraint programming* yang dimodelkan sebagai *constraint satisfaction problem*.
3. Sistem menangani multi institusi dengan *single role user*.
4. Sistem hanya menangani pembuatan jadwal untuk kegiatan belajar-mengajar / pendidikan.

1.5. Manfaat

Manfaat yang diharapkan dari sistem ini mencakup manfaat teoritis, manfaat praktis, serta kontribusi terhadap Tujuan Pembangunan Berkelanjutan (*SDGs*), yaitu sebagai berikut:

1.5.1. Manfaat Teoritis

Secara teoritis, proyek ini diharapkan memberi kontribusi terhadap pengembangan ilmu pengetahuan di bidang Sistem Informasi. Hasil penelitian ini dapat menjadi referensi keilmuan mengenai penerapan teknologi berbasis web dalam melakukan penjadwalan pembelajaran.

1.5.2. Manfaat Praktis

Secara praktis, proyek ini dapat memberikan manfaat sebagai berikut:

1. Menghemat waktu karena pembuatan jadwal dilakukan secara otomatis oleh sistem yang mampu menangani segala batasan.
2. Data inputan dari pengguna tersimpan, sehingga dapat digunakan ulang untuk pembuatan jadwal lain.
3. Meminimalisir kesalahan penjadwalan, seperti bentroknya waktu antara pengajar, ruang, dan waktu karena sistem memverifikasi semua batasannya.

1.5.3. Kontribusi terhadap Tujuan Pembangunan Berkelanjutan (*Sustainable Development Goals/SDGs*)

Proyek ini turut memberikan kontribusi nyata terhadap pencapaian tujuan *SDGs* melalui SDG 9 (Industri, Inovasi, dan Infrastruktur), sistem ini meningkatkan inovasi teknologi pada lembaga pendidikan. Melalui pengembangan ini, proses pembuatan jadwal menjadi lebih mudah dan cepat.

1.6. Kolaborasi dan Sinergi dalam Pelaksanaan Proyek

Proyek Akhir ini dilaksanakan secara individu, dimana penulis bertindak sebagai pengembang tunggal yang bertanggung jawab penuh atas keseluruhan fitur yang dirancang. Penulis melaksanakan seluruh tahapan siklus pengembangan mulai dari tahap perencanaan, pengumpulan kebutuhan, analisis, desain, implementasi hingga pengujian.

Rincian pembagian peran dan tanggung jawab disajikan pada Tabel 1 berikut.

Tabel 1. Pembagian Peran dan Tanggung Jawab

Pihak Terlibat	Peran	Ruang Lingkup Tanggung Jawab
Penulis: Elvan	Penyusun Laporan dan Pengembang Aplikasi	Bertanggung jawab penuh atas seluruh proses mulai dari perencanaan, pengumpulan kebutuhan, analisis,

		desain, implementasi hingga pengujian dan penyusunan laporan.
Dosen Pembimbing: Bapak Muhammad Idris, S.Tr., M.Tr.Kom	Pembimbing Akademik	Memberikan arahan konseptual, penetapan target penyelesaian proyek, memberi saran dan perbaikan pada laporan dan aplikasi, serta pembimbingan laporan.

Selama pelaksanaan proyek, untuk memastikan kualitas dan relevansi solusi yang dihasilkan, penulis melakukan diskusi rutin dan koordinasi dengan dosen pembimbing, Bapak Muhammad Idris, S.Tr., M.Tr.Kom, terkait kesesuaian pengembangan aplikasi dengan kebutuhan fungsional.

Bentuk kolaborasi ini menunjukkan bahwa meskipun proyek dilaksanakan secara individu, keterlibatan aktif dari dosen pembimbing menjadi bagian penting dalam memastikan kualitas dan keberhasilan proyek.

BAB II TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Berikut adalah beberapa penelitian sebelumnya yang berkaitan dengan penelitian ini:

Tabel 2. Perbandingan Penelitian Sebelumnya

Peneliti/ Tahun	Judul/Topik Penelitian	Metode/ Teknologi	Kelebihan	Kekurangan/ Celah
(Jeriko et al., 2022)	Penerapan Algoritma Genetika dalam Sistem Informasi Penjadwalan Mata Kuliah Berbasis Website pada STIKOM Yos Sudarso Purwokerto	Metode: <i>Prototype</i> dan Algoritma Genetika Teknologi: <i>PHP, Code Igniter 3, Javascript, dan MySQL</i>	Meningkatkan efektivitas dan efisiensi	Tidak memiliki sistem export data
(Azmi & Najiyah, 2023)	Sistem Informasi Penjadwalan Mata Pelajaran Menggunakan Algoritma <i>Artificial Bee Colony</i> Berbasis Web	Metode: <i>Waterfall</i> dan <i>Artificial Bee Colony Algorithm</i>	Mengoptimalkan waktu pembuatan jadwal dan menghindari konflik jadwal	Penjadwalan dibuat per kelas sehingga memakan banyak waktu jika terdapat banyak kelas
(Zahidah & Putri, 2025)	Implementasi Algoritma <i>Constraint Satisfaction Problems</i> dan <i>Backtracking</i> pada Penjadwalan	Metode: <i>Waterfall</i> , Algoritma <i>Constraint Satisfaction Problems</i> , dan Algoritma <i>Backtracking</i>	Menghasilkan jadwal secara otomatis, meningkatkan efisiensi, fleksibilitas, dan akurasi	Pengelolaan jadwal kegiatan lain masih secara manual

	Kegiatan Belajar Mengajar	Teknologi: <i>PHP, Laravel, React JS, dan MySQL</i>	proses penjadwalan	
Penelitian ini	Rancang Bangun Sistem Informasi Penjadwalan Pembelajaran Otomatis Berbasis Web Menggunakan <i>Constraint Programming</i>	Metode: <i>Waterfall dan Constraint Programming</i> Teknologi: <i>Tailwind CSS, JavaScript, Python, Django, dan PostgreSQL</i>	Fitur penambahan batasan dan <i>export data</i>	-

Dari penelitian yang sudah ada, maka penulis juga akan melakukan “Rancang Bangun Sistem Informasi Penjadwalan Pembelajaran Otomatis Berbasis Web Menggunakan *Constraint Programming*”. Penulis melihat hasil aplikasi dan permasalahan yang ada dapat diselesaikan dengan mengoptimalkan fitur-fitur yang sudah ada seperti sistem pembuatan akun guna untuk menyimpan data-data agar bisa digunakan untuk membuat jadwal lainnya dan penambahan batasan untuk penyesuaian masing-masing pengajar. Dengan ini, aplikasi yang dihasilkan memiliki kelebihan dan perbedaan tersendiri dari penelitian-penelitian yang dilakukan sebelumnya.

2.2. Landasan Teori

2.2.1. Penjadwalan

Penjadwalan adalah proses pengalokasian sumber daya yang terbatas ke sejumlah kegiatan (tugas) dalam rentang waktu yang tersedia, dengan tujuan memenuhi beberapa batasan serta mengoptimalkan kinerja,

seperti minimisasi waktu penyelesaian, penggunaan sumber daya maksimal, atau menghindari tumpang tindih (Pinedo, 2022).

2.2.2. Web

Aplikasi web adalah aplikasi yang diakses dengan peramban web melalui jaringan seperti internet atau intranet. Secara khusus, sistem berbasis web digunakan sebagai halaman web informatif atau interaktif (Sitohang et al., 2021).

2.2.3. *Constraint Programming*

Secara umum, *constraint programming* adalah studi tentang sistem komputasi berbasis kendala. Kemunculannya dalam dunia riset operasi, khususnya untuk memodelkan dan memecahkan masalah kombinatorial. Salah satu manifestasi sederhana dari struktur ini adalah bahwa *constraint programming* bekerja secara langsung pada variabel diskrit alih-alih bergantung pada relaksasi model yang berkelanjutan (Bessiere et al., 2023).

Dalam *constraint programming*, terdapat 3 bagian utama, yaitu *variable*, *domain*, dan *constraint*. *Variable* adalah simbol yang mewakili suatu nilai. *Domain* adalah nilai yang dimiliki suatu *variable*. *Constraint* adalah nilai yang bisa diambil oleh sebuah *variable*. Dapat disimpulkan untuk menemukan sekumpulan nilai yang ditetapkan untuk setiap *variable* yang memenuhi semua persyaratan / *constraint* atau bisa didefinisikan menjadi *constraint satisfaction problem (CSP)*.

Berdasarkan jurnal “*Constraint Satisfaction – a Survey*” (Google, 2026), *CSP* dapat diklasifikasikan menjadi:

1. *Classical CSP*, semua *variable*, *domain*, dan *constraint* harus terpenuhi.
2. *Dynamic CSP*, *constraint* dapat berubah seiring waktu.
3. *Distributed CSP*, *constraint* tersebar di beberapa agen / *node*.
4. *Fuzzy / Probabilistic CSP*, *constraint* memiliki derajat kepuasan.
5. *Over-constrained CSP*, tidak semua *constraint* bisa dipenuhi, digunakan *relaxation* atau *optimization*.

Berdasarkan jurnal “*Constraint Satisfaction – a Survey*” (Google, 2026), CSP memiliki beberapa metode penyelesaian:

1. *Systematic Search Methods*, seperti *backtracking*, *forward checking*, dan *arc consistency*.
2. *Inference / Constraint Propagation Methods*, seperti *AC-3*, *AC-4*, dan *path consistency*.
3. *Heuristic & Variable Ordering*, seperti *minimum remaining value (MRV)* dan *degree heuristic*.
4. *Decomposition Methods*, seperti *tree-decomposition* dan *graph-based*.
5. *Local Search Methods*, seperti *hill climbing* dan *simulated annealing*.

2.2.4. HTML

HTML adalah bahasa markah utama *World Wide Web*. HTML menentukan semantik struktural untuk teks dalam dokumen web. Elemen-elemennya memberi tahu peramban web cara menampilkan konten pada halaman web, termasuk teks, gambar, video, dan audio (Putro et al., 2024).

2.2.5. CSS

CSS adalah bahasa tata letak dan pemformatan yang memungkinkan Anda memformat bahasa *markup* seperti HTML. Idealnya, dokumen HTML hanya berisi informasi semantik, dan dengan CSS, informasi ini akan diformat secara formatif dan tipografis (Duckett, 2021).

2.2.6. JavaScript

Bahasa pemrograman JavaScript banyak digunakan untuk pemrograman web dan semakin meningkat, untuk komputasi tujuan umum. JavaScript adalah bahasa berorientasi objek yang dirancang pada tahun 1995 oleh Brendan Eich di Netscape untuk memungkinkan *non-programmer* memperluas situs web dengan kode yang dapat dieksekusi di sisi klien (Ecma International, 2026).

2.2.7. Python

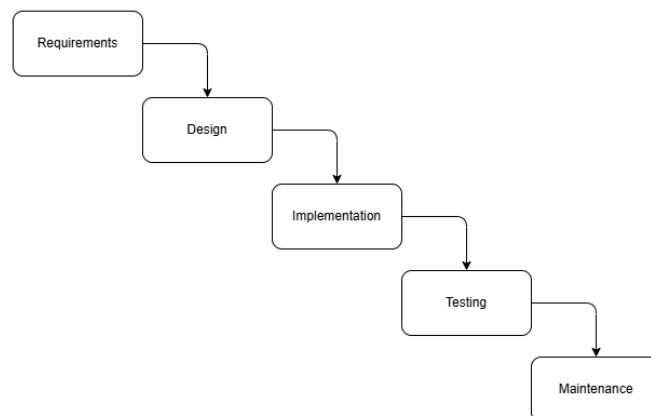
Fitur-fitur pemrograman berorientasi objek tingkat tinggi tersedia dalam *Python*, sebuah bahasa pemrograman serbaguna. Bahasa ini telah mendapatkan popularitas karena sintaksisnya yang mudah dipahami dan tidak rumit, portabilitas, serta kemudahan pembelajarannya. *Python* adalah bahasa komputer dengan fitur-fitur dari *Java* dan *C* (Lutz, 2025).

2.2.8. PostgreSQL

PostgreSQL adalah *DBMS* Relasional Objek canggih yang mendukung konstruksi *SQL* dan dapat diakses dengan protokol standar. Fitur-fiturnya yang berorientasi objek membuatnya memenuhi syarat untuk mengelola data dan tersedia gratis untuk komunitas melalui filosofi sumber terbuka (Zapata, 2024).

2.3. Metode Pengembangan Produk

Metode pengembangan / *Software Development Life Cycle (SDLC)* yang digunakan pada proyek ini adalah metode *waterfall*. Model ini terdiri dari serangkaian fase, dimulai dengan spesifikasi sistem, dengan hasil yang mengalir dari satu tahap ke tahap lainnya (Saravanos & Curinga, 2023). Dengan menggunakan metode ini, pengembangan harus menyelesaikan tahap A untuk melanjutkan ke tahap B. Di dalam metode *waterfall* terdapat beberapa tahapan, yaitu pengumpulan kebutuhan (*requirements*), desain (*design*), implementasi (*implementation*), pengujian (*testing*), dan pengelolaan (*maintenance*).



Gambar 1. Metode Waterfall

2.3.1. *Requirements*

Requirements merupakan tahapan pertama dalam pengembangan sistem berbasis *waterfall*. Fungsionalitas perangkat lunak dan kendala operasinya ditentukan secara rinci (Saravanos & Curinga, 2023). Pada tahap ini, pengembang mencari tahu kebutuhan-kebutuhan yang diperlukan untuk membangun sistem ini.

2.3.2. *Design*

Tahap selanjutnya adalah *design*. Struktur keseluruhan perangkat lunak dirancang dan komponen perangkat lunak spesifik diidentifikasi (Saravanos & Curinga, 2023). Pada tahap ini, pengembang akan merancang gambaran umum dari sistem, struktur dan logika sistem, hingga tampilan antarmuka.

2.3.3. *Implementation*

Setelah melakukan tahap *design*, akan melanjutkan ke tahap *implementation* / pengembangan. Ini diimplementasikan menggunakan beberapa bahasa pemrograman, seringkali oleh individu atau tim yang berbeda (Saravanos & Curinga, 2023). Pada tahap pengembangan, sistem akan dikembangkan dengan bahasa pemrograman *python* dengan *framework django*, *postgreSQL* sebagai basis data, dan kode editor *Visual Studio Code*.

2.3.4. *Testing*

Pada tahap *testing*, sistem akan diuji untuk dipastikan layak dipakai. Modul yang dikembangkan secara individual diintegrasikan ke dalam sistem yang lengkap dan diuji (Saravanos & Curinga, 2023). Metode yang digunakan untuk menguji sistem adalah *black box testing* untuk memastikan sistem berjalan dengan seharusnya dan sesuai dengan kebutuhan yang dikumpulkan.

2.3.5. *Maintenance*

Tahap terakhir dalam metode *waterfall* adalah *maintenance*. Perangkat lunak dimodifikasi untuk memenuhi perubahan kebutuhan dan memperbaiki kesalahan yang ditemukan saat penggunaan (Saravanos & Curinga, 2023). Pemeliharaan sistem akan dilakukan untuk memastikan sistem tetap berjalan dengan baik dan melakukan perbaikan terhadap kesalahan yang ditemukan.

BAB III ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan

Bagian ini merupakan kebutuhan dari sistem yang akan dikembangkan, mulai dari kendala / kondisi yang dihadapi saat ini hingga rancangan / gambaran awal sistem yang akan disusun.

3.1.1. Proses Bisnis Berjalan

Alur kerja adalah satu atau lebih prosedur atau aktivitas terkait yang secara kolektif mewujudkan sasaran bisnis atau sasaran kebijakan, biasanya dalam konteks struktur organisasi yang mendefinisikan peran dan hubungan fungsional (Dumas et al., 2023). Berikut adalah proses bisnis dari masing-masing penelitian yang dijadikan referensi:

1. Jurnal “Sistem Informasi Penjadwalan Pembelajaran pada Sekolah Menggunakan *Genetic Algorithm* Berbasis *Website*” (Hilmi et al., 2024)

Pada jurnal ini, proses bisnis dimulai dengan *login* ke *website*, menambahkan data guru, mata pelajaran, pengampu, ruang, jam, hari, dan waktu tidak tersedia. Kemudian proses membuat jadwal dan *export* ke format *xlsx*. Dari aplikasi yang dibuat, guru perlu *login* ke *website* hanya untuk melihat jadwal sehingga kurang efektif.

2. Jurnal “Sistem Penjadwalan Perkuliahan Menggunakan Algoritma Genetika pada Jurusan Teknik Elektro Universitas Riau” (Lestari et al., 2023)

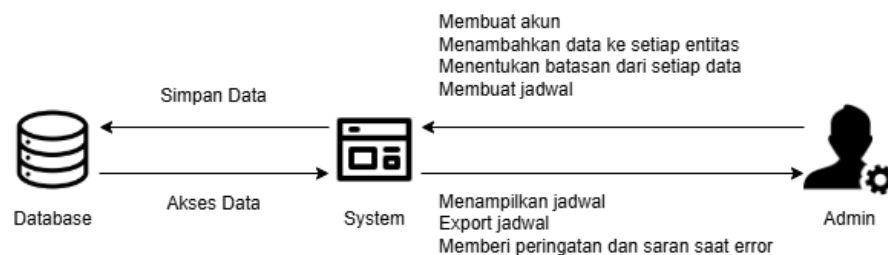
Pada jurnal ini, proses bisnis dimulai dengan *login*, tambah jadwal, memasukkan data mata kuliah, dan buat jadwal. Jadwal yang terbuat bisa di *export* ke format *xlsx* dan diunduh ke format *pdf*. Dari aplikasi yang dikembangkan, jadwal tidak bisa langsung dilihat dari *website* sehingga perlu dilakukan *export* maupun unduh terlebih dahulu.

3. Jurnal “Implementasi Metode Pewarnaan Graf Menggunakan Algoritma *Greedy* untuk Menentukan Jadwal Mata Pelajaran pada SMKN 4 Kendari” (Fadel et al., 2023)

Pada jurnal ini, proses bisnis dimulai dengan *login*, tambah mata pelajaran, guru, dan kelas. Lalu *generate* jadwal untuk masing-masing kelas. Pada aplikasi ini, pengguna tidak bisa melakukan *import* untuk menambahkan data secara massal.

3.1.2. Gambaran Umum Sistem

Sistem merupakan gabungan terpadu dari orang, produk, dan proses yang menyediakan kemampuan untuk memenuhi kebutuhan atau tujuan tertentu (Salado & Kulkarni, 2021). Bagian ini akan menjelaskan bagaimana sistem ini bekerja beserta komponen dan interaksi yang terlibat. Berikut adalah gambaran umum dari sistem ini:



Gambar 2. Gambaran Umum Sistem

Sesuai dengan gambaran umum sistem di atas, pengguna (admin) bisa membuat akun, menambahkan data ke setiap entitas, menentukan batasan dari setiap data, dan membuat jadwal. Kemudian sistem akan menyimpan data ke *database*, dan sistem akan menampilkan jadwal berdasarkan data yang ditambahkan oleh admin. Sistem juga dapat melakukan *export* jadwal dan memberi peringatan dan saran saat terjadi *error*.

3.2. Perancangan

Pada bagian ini, akan memuat rancangan sistem yang akan dikembangkan, yang terdiri dari:

3.2.1. Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan perilaku fungsional sistem (Dennis et al., 2021). Berikut adalah kebutuhan fungsional yang disajikan pada tabel 3.

Tabel 3. Kebutuhan Fungsional

Kode	Deskripsi
FR-01	Admin bisa membuat akun
FR-02	Admin bisa <i>login</i> dan <i>logout</i>
FR-03	Admin bisa mengelola hari
FR-04	Admin bisa mengelola jam pembelajaran
FR-05	Admin bisa mengelola kelas
FR-06	Admin bisa mengelola ruang kelas
FR-07	Admin bisa mengelola pengajar
FR-08	Admin bisa mengelola pelajaran beserta <i>slot</i> waktunya
FR-09	Admin bisa menentukan batasan dari setiap data
FR-10	Admin mampu membuat jadwal secara otomatis
FR-11	Sistem mampu meng- <i>export</i> jadwal dalam format <i>pdf</i>
FR-12	Pengguna bisa menyesuaikan jadwal yang sudah terbuat

3.2.2. Kebutuhan Non-Fungsional

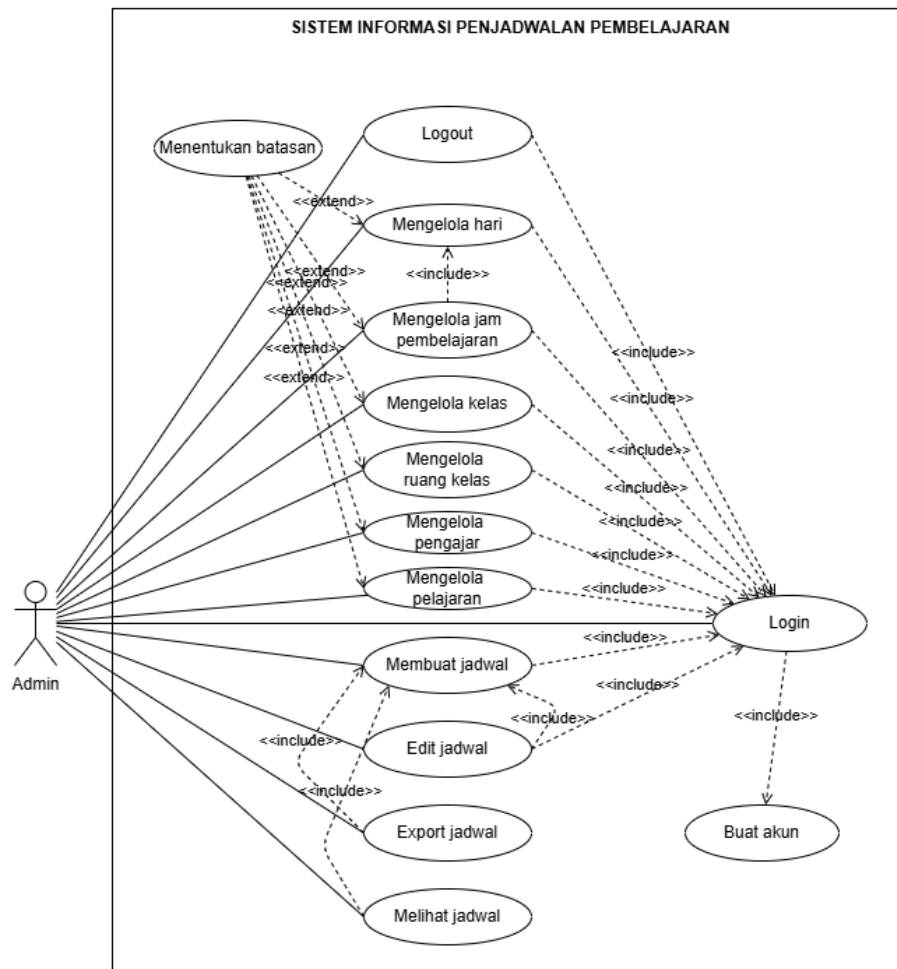
Kebutuhan non-fungsional mengungkapkan seberapa baik sistem harusnya bekerja (Dennis et al., 2021). Berikut adalah kebutuhan non-fungsional yang disajikan pada tabel 4.

Tabel 4. Kebutuhan Non-Fungsional

Kode	Deskripsi
NFR-01	Sistem menggunakan bahasa Indonesia karena sistem yang dikembangkan berfokus pada sistem pendidikan di Indonesia
NFR-02	Sistem mampu ditampilkan secara responsif di layar manapun
NFR-03	Sistem melindungi <i>password</i> dengan keamanan enkripsi data
NFR-04	Sistem beroperasi secara legal, etis, dan aman bagi pengguna dan lingkungan

3.2.3. Diagram Use Case

Use case diagram adalah suatu tindakan dari sudut pandang sistem, yang dimaksudkan oleh perancang sistem, dan diharapkan oleh beberapa pengguna atau pemangku kepentingan lainnya, ketika keadaannya tepat, untuk menghasilkan hasil tertentu yang dapat diamati yang bernilai bagi satu atau lebih pengguna atau pemangku kepentingan lain dari sistem tersebut (Dennis et al., 2021). Berikut adalah diagram *use case* dari sistem yang akan dikembangkan:



Gambar 3. Diagram Use Case

3.2.4. Skenario Use Case

Skenario *use case* adalah serangkaian interaksi yang terjadi dalam kondisi tertentu, dengan tujuan mencapai tujuan utama aktor, dan menghasilkan hasil tertentu terkait tujuan tersebut. Interaksi dimulai dari tindakan pemicu dan berlanjut hingga tujuan tercapai atau dibatalkan, dan sistem menyelesaikan tanggung jawab apa pun yang dimilikinya terkait interaksi tersebut (Dennis et al., 2021). Berikut adalah skenario-skenario dari setiap *use case*:

1. Skenario *Use Case* Buat Akun

Skenario *use case* buat akun menjelaskan proses pembuatan akun pengguna agar dapat menggunakan aplikasi yang ditunjukkan pada tabel 5.

Tabel 5. Skenario Use Case Buat Akun

Nomor	UC001	
Nama	Buat Akun	
Deskripsi	Admin bisa buat akun	
Aktor	Admin	
Kondisi Awal	Admin berada pada halaman buat akun	
Skenario Utama		
Aksi Aktor		Reaksi Sistem
1. Admin memasukkan <i>username, password</i> , dan konfirmasi <i>password</i> 2. Admin menekan tombol daftar 5. Admin memasukkan nama dan nama institusi 6. Admin menekan tombol masuk		3. Sistem melakukan pengecekan data 4. Menampilkan halaman data pengguna 7. Sistem mendaftarkan pengguna 8. Sistem menampilkan halaman dasbor
Skenario Alternatif		
Aksi Aktor		Reaksi Sistem

-	3a. Jika username tidak valid atau sudah ada, atau password tidak valid maka muncul pesan error
Kondisi Akhir	Admin berhasil buat akun

2. Skenario *Use Case Login*

Skenario *use case login* menjelaskan proses verifikasi akun pengguna yang ingin masuk ke sistem seperti yang ditunjukkan pada tabel 6.

Tabel 6. Skenario Use Case Login

Nomor	UC002		
Nama	Login		
Deskripsi	Admin bisa login ke aplikasi		
Aktor	Admin		
Kondisi Awal	Admin berada pada halaman login		
Skenario Utama			
Aksi Aktor		Reaksi Sistem	
1. Admin memasukkan username dan password 2. Admin menekan tombol login		3. Sistem melakukan pengecekan data login 4. Menampilkan halaman admin	
Skenario Alternatif			
Aksi Aktor		Reaksi Sistem	
-		3a. Jika data login tidak valid, maka muncul pesan salah password	
Kondisi Akhir	Admin berhasil login		

3. Skenario *Use Case Logout*

Skenario *use case logout* menjelaskan proses keluar / *logout* dari sistem yang ditunjukkan pada tabel 7.

Tabel 7. Skenario Use Case Logout

Nomor	UC003		
Nama	Logout		
Deskripsi	Admin bisa <i>logout</i> dari aplikasi		
Aktor	Admin		
Kondisi Awal	Admin sudah <i>login</i>		
Skenario Utama			
Aksi Aktor		Reaksi Sistem	
1. Admin menekan tombol <i>logout</i> 3. Admin menekan Ya		2. Sistem melakukan konfirmasi <i>logout</i> 4. Sistem menghapus data <i>login</i> admin	
Skenario Alternatif			
Aksi Aktor		Reaksi Sistem	
3a. Admin menekan Batal		4a. Sistem menutup tampilan konfirmasi <i>logout</i>	
Kondisi Akhir	Admin berhasil <i>logout</i>		

4. Skenario *Use Case Mengelola Hari*

Skenario *use case* mengelola hari menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data hari yang ditunjukkan pada tabel 8.

Tabel 8. Skenario Use Case Mengelola Hari

Nomor	UC004
--------------	-------

Nama	Mengelola Hari
Deskripsi	Admin bisa tambah, lihat, ubah, dan hapus data hari
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Admin menuju halaman data hari 3. Admin tekan Tambah 4. Admin mengisi data 5. Admin tekan Simpan	2. Sistem menampilkan halaman data hari 6. Sistem menyimpan data ke <i>database</i>
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
3a. Admin tekan Ubah 4a. Admin mengubah data 5a. Admin tekan Simpan 3b. Admin tekan Hapus 5b. Admin pilih Ya	6a. Sistem mengubah data ke <i>database</i> 4b. Sistem meminta konfirmasi 6b. Sistem menghapus data dari <i>database</i>
Kondisi Akhir	Admin berhasil mengelola data hari

5. Skenario *Use Case* Mengelola Jam Pembelajaran

Skenario *use case* mengelola jam pembelajaran menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data jam pembelajaran yang ditunjukkan pada tabel 9.

Tabel 9. Skenario Use Case Mengelola Jam Pembelajaran

Nomor	UC005
Nama	Mengelola Jam Pembelajaran
Deskripsi	Admin bisa tambah, lihat, ubah, dan hapus data jam pembelajaran
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Admin menuju halaman data jam pembelajaran 3. Admin tekan Tambah 4. Admin mengisi data 5. Admin tekan Simpan	2. Sistem menampilkan halaman data jam pembelajaran 6. Sistem menyimpan data ke <i>database</i>
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
3a. Admin tekan Ubah 4a. Admin mengubah data 5a. Admin tekan Simpan 3b. Admin tekan Hapus	6a. Sistem mengubah data ke <i>database</i>

5b. Admin pilih Ya	4b. Sistem meminta konfirmasi 6b. Sistem menghapus data dari <i>database</i>
Kondisi Akhir	Admin berhasil mengelola data jam pembelajaran

6. Skenario *Use Case* Mengelola Kelas

Skenario *use case* mengelola kelas menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data kelas yang ditunjukkan pada tabel 10.

Tabel 10. Skenario Use Case Mengelola Kelas

Nomor	UC006		
Nama	Mengelola Kelas		
Deskripsi	Admin bisa tambah, lihat, ubah, dan hapus data kelas		
Aktor	Admin		
Kondisi Awal	Admin sudah <i>login</i>		
Skenario Utama			
Aksi Aktor		Reaksi Sistem	
1. Admin menuju halaman data kelas 3. Admin tekan Tambah 4. Admin mengisi data 5. Admin tekan Simpan		2. Sistem menampilkan halaman data kelas 6. Sistem menyimpan data ke <i>database</i>	
Skenario Alternatif			
Aksi Aktor		Reaksi Sistem	
3a. Admin tekan Ubah			

4a. Admin mengubah data 5a. Admin tekan Simpan 3b. Admin tekan Hapus 5b. Admin pilih Ya	6a. Sistem mengubah data ke <i>database</i> 4b. Sistem meminta konfirmasi 6b. Sistem menghapus data dari <i>database</i>
Kondisi Akhir	Admin berhasil mengelola data kelas

7. Skenario *Use Case* Mengelola Ruang Kelas

Skenario *use case* mengelola ruang kelas menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data ruang kelas yang ditunjukkan pada tabel 11.

Tabel 11. Skenario Use Case Mengelola Ruang Kelas

Nomor	UC007
Nama	Mengelola Ruang Kelas
Deskripsi	Admin bisa tambah, lihat, ubah, dan hapus data ruang kelas
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Admin menuju halaman data ruang kelas 3. Admin tekan Tambah 4. Admin mengisi data 5. Admin tekan Simpan	2. Sistem menampilkan halaman data ruang kelas

	6. Sistem menyimpan data ke <i>database</i>
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
3a. Admin tekan Ubah 4a. Admin mengubah data 5a. Admin tekan Simpan 3b. Admin tekan Hapus 5b. Admin pilih Ya	6a. Sistem mengubah data ke <i>database</i> 4b. Sistem meminta konfirmasi 6b. Sistem menghapus data dari <i>database</i>
Kondisi Akhir	Admin berhasil mengelola data ruang kelas

8. Skenario *Use Case* Mengelola Pengajar

Skenario *use case* mengelola pengajar menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data pengajar yang ditunjukkan pada tabel 12.

Tabel 12. Skenario Use Case Mengelola Pengajar

Nomor	UC008
Nama	Mengelola Pengajar
Deskripsi	Admin bisa tambah, lihat, ubah, dan hapus data pengajar
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Admin menuju halaman data pengajar	

3. Admin tekan Tambah 4. Admin mengisi data 5. Admin tekan Simpan	2. Sistem menampilkan halaman data pengajar 6. Sistem menyimpan data ke <i>database</i>
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
3a. Admin tekan Ubah 4a. Admin mengubah data 5a. Admin tekan Simpan 3b. Admin tekan Hapus 5b. Admin pilih Ya	6a. Sistem mengubah data ke <i>database</i> 4b. Sistem meminta konfirmasi 6b. Sistem menghapus data dari <i>database</i>
Kondisi Akhir	Admin berhasil mengelola data pengajar

9. Skenario *Use Case* Mengelola Pelajaran

Skenario *use case* mengelola pelajaran menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data pelajaran yang ditunjukkan pada tabel 13.

Tabel 13. Skenario Use Case Mengelola Pelajaran

Nomor	UC009
Nama	Mengelola Pelajaran
Deskripsi	Admin bisa tambah, lihat, ubah, dan hapus data pelajaran
Aktor	Admin

Kondisi Awal	Admin sudah <i>login</i>	
Skenario Utama		
Aksi Aktor		Reaksi Sistem
1. Admin menuju halaman data pelajaran 3. Admin tekan Tambah 4. Admin mengisi data 5. Admin tekan Simpan		2. Sistem menampilkan halaman data pelajaran 6. Sistem menyimpan data ke <i>database</i>
Skenario Alternatif		
Aksi Aktor		Reaksi Sistem
3a. Admin tekan Ubah 4a. Admin mengubah data 5a. Admin tekan Simpan 3b. Admin tekan Hapus 5b. Admin pilih Ya		6a. Sistem mengubah data ke <i>database</i> 4b. Sistem meminta konfirmasi 6b. Sistem menghapus data dari <i>database</i>
Kondisi Akhir	Admin berhasil mengelola data pelajaran	

10. Skenario *Use Case* Menentukan Batasan

Skenario *use case* menentukan batasan menjelaskan proses admin menambahkan batasan dari data-data yang dimasukkan untuk membuat jadwal yang ditunjukkan pada tabel 14.

Tabel 14. Skenario Use Case Menentukan Batasan

Nomor	UC010
Nama	Menentukan Batasan
Deskripsi	Admin bisa menentukan batasan dari setiap data saat membuat jadwal
Aktor	Admin
Kondisi Awal	Admin memiliki data-data yang diperlukan untuk membuat jadwal
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Admin menuju halaman jadwal 3. Admin tekan Buat Jadwal 5. Admin menambahkan data-data yang diperlukan 6. Admin menentukan batasan yang diperlukan pada data tertentu	2. Sistem menampilkan halaman jadwal 4. Sistem menampilkan halaman buat jadwal
Kondisi Akhir	Admin berhasil menentukan batasan kepada data

11. Skenario Use Case Membuat Jadwal

Skenario *use case* membuat jadwal menjelaskan proses pembuatan jadwal oleh sistem setelah admin menambahkan data-data yang diperlukan yang ditunjukkan pada tabel 15.

Tabel 15. Skenario Use Case Membuat Jadwal

Nomor	UC011
--------------	-------

Nama	Membuat Jadwal
Deskripsi	Admin bisa membuat jadwal dari data yang ditambahkan
Aktor	Admin
Kondisi Awal	Admin sudah menentukan batasan dari data-data yang ditambahkan
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Admin tekan <i>Generate</i> 5. Admin mengisi nama jadwal dan konfigurasi publikasi 6. Admin tekan Simpan	2. Sistem melakukan pengecekan terhadap data dan batasan yang ditambahkan admin 3. Sistem melakukan proses pembuatan jadwal 4. Sistem menampilkan jadwal
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
5a. Jika jadwal akan dipublikasi, maka terdapat opsi untuk memakai pin untuk akses jadwal	3a. Jika hasil pengecekan tidak valid, maka sistem menampilkan <i>error</i>
Kondisi Akhir	Admin berhasil membuat jadwal

12. Skenario *Use Case Edit Jadwal*

Skenario *use case edit* jadwal menjelaskan proses admin melakukan penyesuaian pada data jadwal yang ditunjukkan pada tabel 16.

Tabel 16. Skenario Use Case Edit Jadwal

Nomor	UC012
Nama	Edit Jadwal
Deskripsi	Admin bisa menyesuaikan jadwal yang sudah terbuat
Aktor	Admin
Kondisi Awal	Admin memiliki jadwal yang sudah terbuat
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Admin memilih jadwal yang akan di <i>edit</i> 2. Admin tekan <i>Edit</i> 4. Admin menyesuaikan data jadwal 5. Admin tekan Simpan	3. Sistem menampilkan detail jadwal 6. Sistem mengubah data pada <i>database</i>
Kondisi Akhir	Admin berhasil <i>edit</i> jadwal

13. Skenario *Use Case Export Jadwal*

Skenario *use case export* jadwal (admin) menjelaskan proses admin melakukan *export* untuk mengunduh jadwal ke format *pdf* atau *xlsx* yang ditunjukkan pada tabel 17.

Tabel 17. Skenario Use Case Export Jadwal (Admin)

Nomor	UC013
Nama	<i>Export Jadwal</i>

Deskripsi	Admin bisa meng- <i>export</i> jadwal yang ada	
Aktor	Admin	
Kondisi Awal	Admin memiliki jadwal yang sudah terbuat	
Skenario Utama		
Aksi Aktor		Reaksi Sistem
1. Admin memilih jadwal yang akan di <i>export</i> 3. Admin tekan <i>Export</i> 5. Admin mengunduh hasil <i>export</i>		2. Sistem menampilkan data jadwal 4. Sistem menyiapkan data ke format yang dipilih
Kondisi Akhir	Admin berhasil melakukan <i>export</i> jadwal	

14. Skenario *Use Case* Melihat Jadwal

Skenario *use case* melihat jadwal (admin) menjelaskan proses admin melihat data jadwal yang sudah dibuat sebelumnya yang ditunjukkan pada tabel 18.

Tabel 18. Skenario *Use Case* Melihat Jadwal (Admin)

Nomor	UC014
Nama	Melihat Jadwal
Deskripsi	Admin bisa melihat jadwal yang di <i>generate</i> sistem
Aktor	Admin
Kondisi Awal	Admin memiliki jadwal yang sudah terbuat
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Admin memilih jadwal yang ingin dilihat	

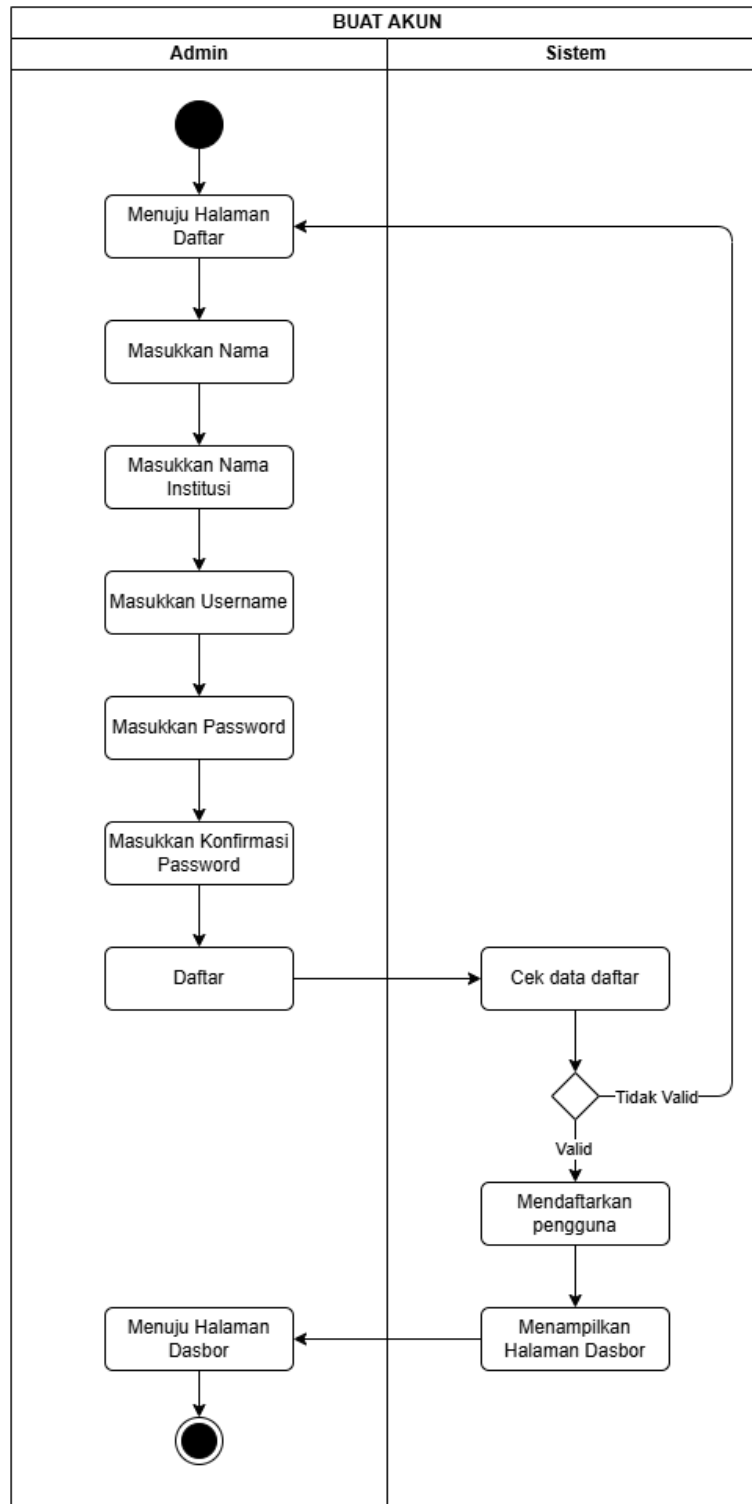
3. Admin melihat data jadwal	2. Sistem menampilkan data jadwal
Kondisi Akhir	Admin bisa melihat data jadwal

3.2.5. *Activity Diagram*

Activity diagram digunakan untuk menangkap detail proses yang dimaksud. Sebaliknya, pemodelan perusahaan umumnya memerlukan berbagai teknik pemodelan yang digunakan bersama-sama untuk menangkap informasi domain yang dibutuhkan (Dennis et al., 2021). Berikut adalah *activity diagram* dari setiap *use case*:

1. Activity Diagram Buat Akun

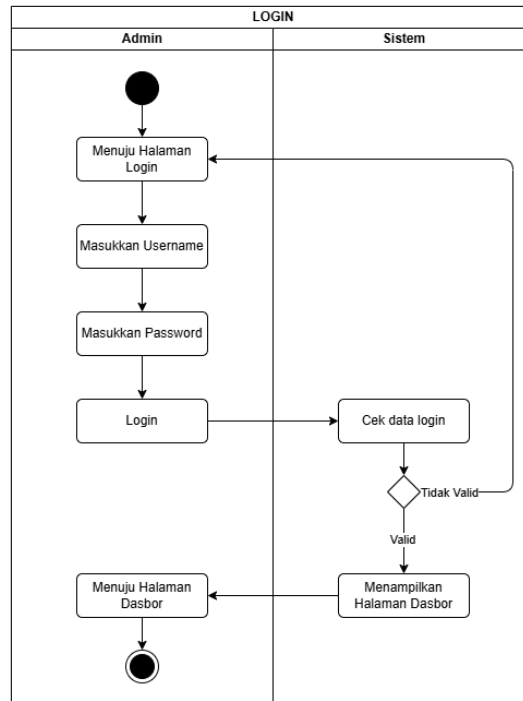
Activity diagram buat akun menjelaskan proses pembuatan akun pengguna agar dapat menggunakan aplikasi.



Gambar 4. Activity Diagram Buat Akun

2. Activity Diagram Login

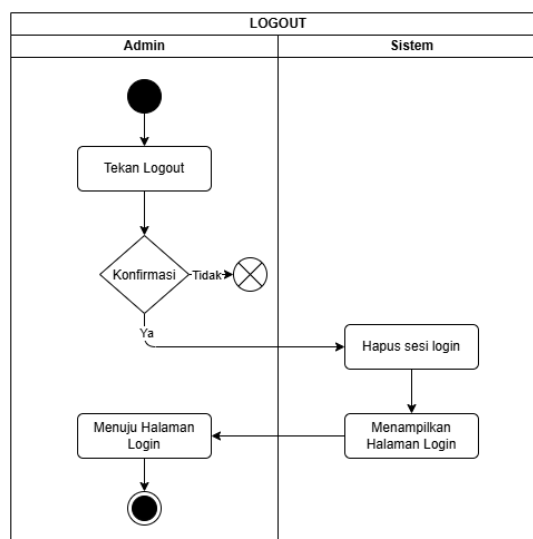
Activity diagram menjelaskan proses verifikasi akun pengguna yang ingin masuk ke sistem.



Gambar 5. Activity Diagram Login

3. Activity Diagram Logout

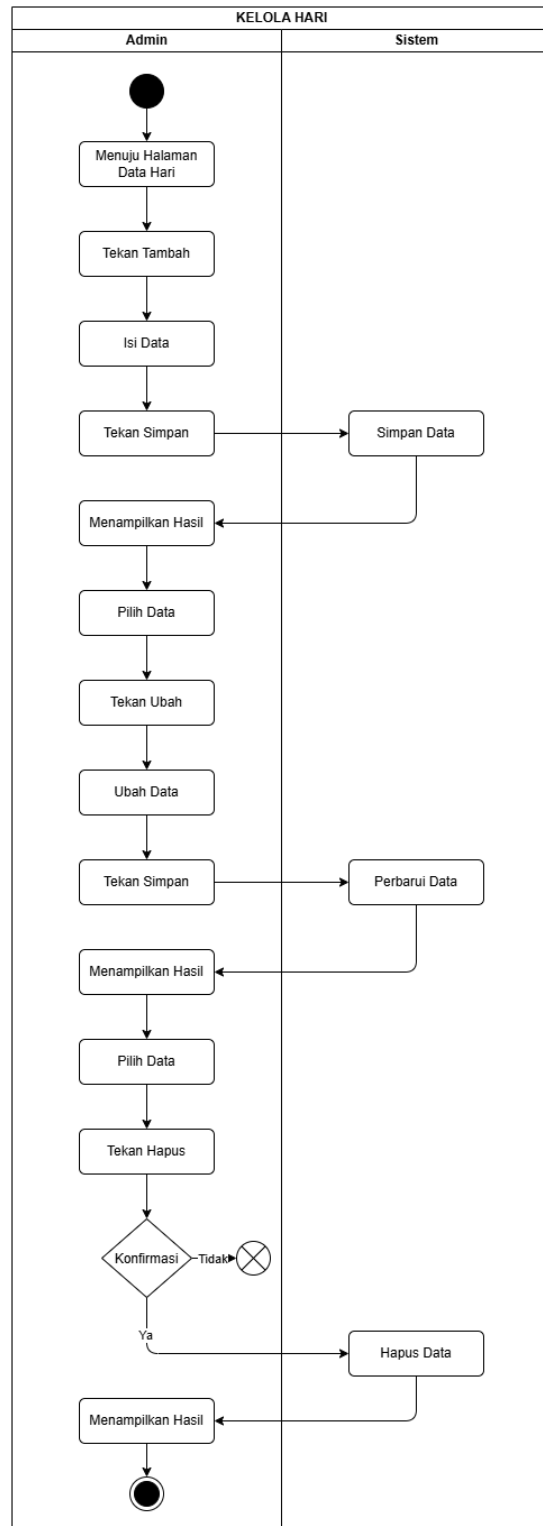
Activity Diagram menjelaskan proses keluar / logout dari sistem.



Gambar 6. Activity Diagram Logout

4. Activity Diagram Mengelola Hari

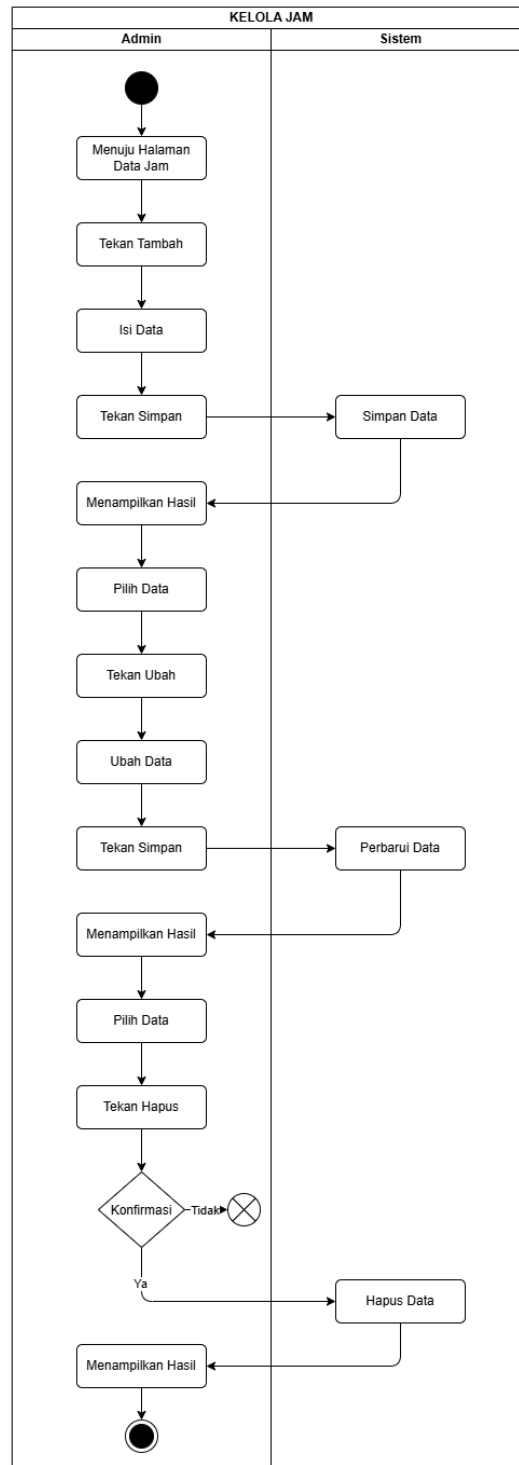
Activity Diagram mengelola hari menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data hari.



Gambar 7. Activity Diagram Mengelola Hari

5. Activity Diagram Mengelola Jam Pembelajaran

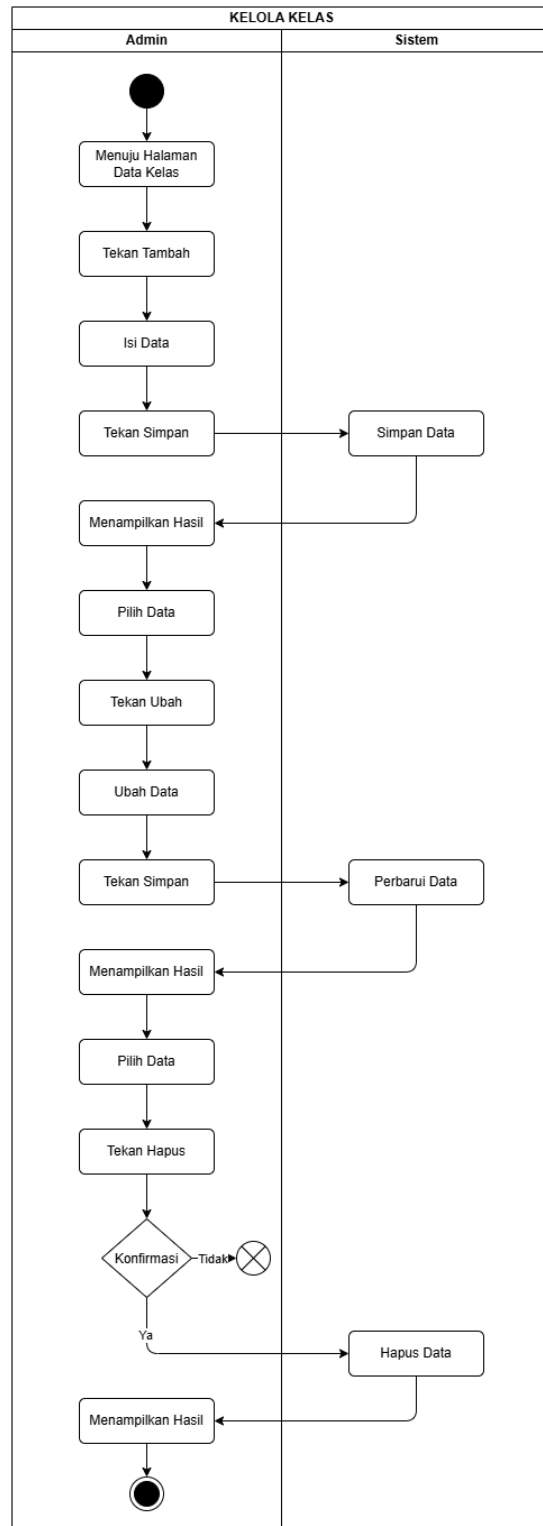
Activity Diagram mengelola jam pembelajaran menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data jam pembelajaran.



Gambar 8. Activity Diagram Mengelola Jam Pembelajaran

6. Activity Diagram Mengelola Kelas

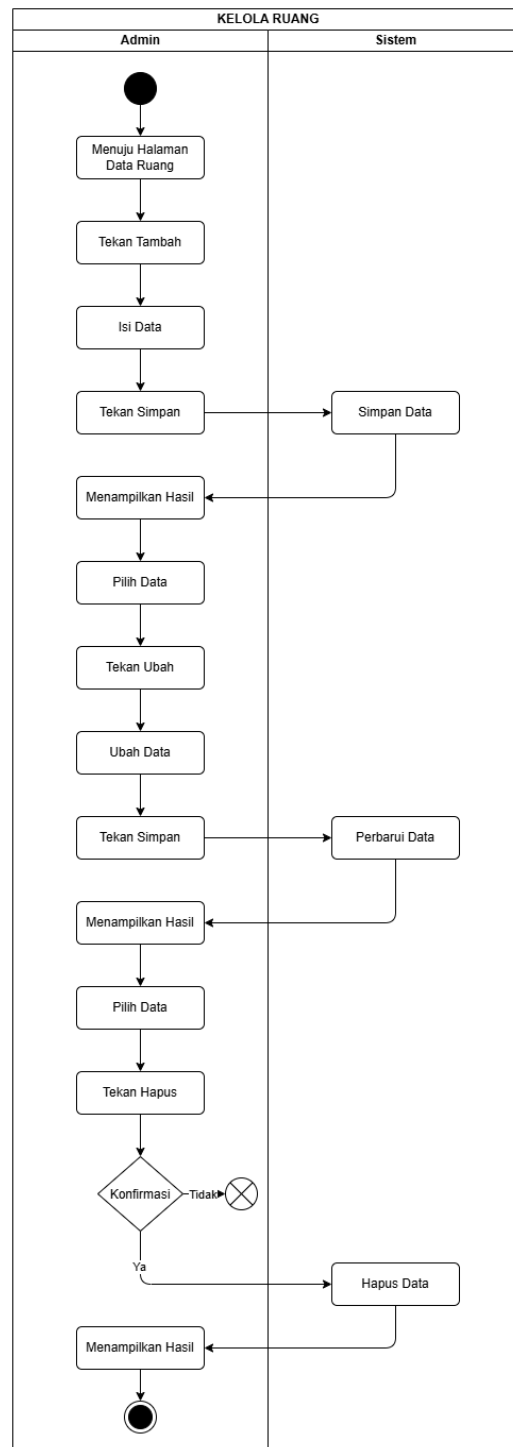
Activity Diagram mengelola kelas menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data kelas.



Gambar 9. Activity Diagram Mengelola Kelas

7. Activity Diagram Mengelola Ruang Kelas

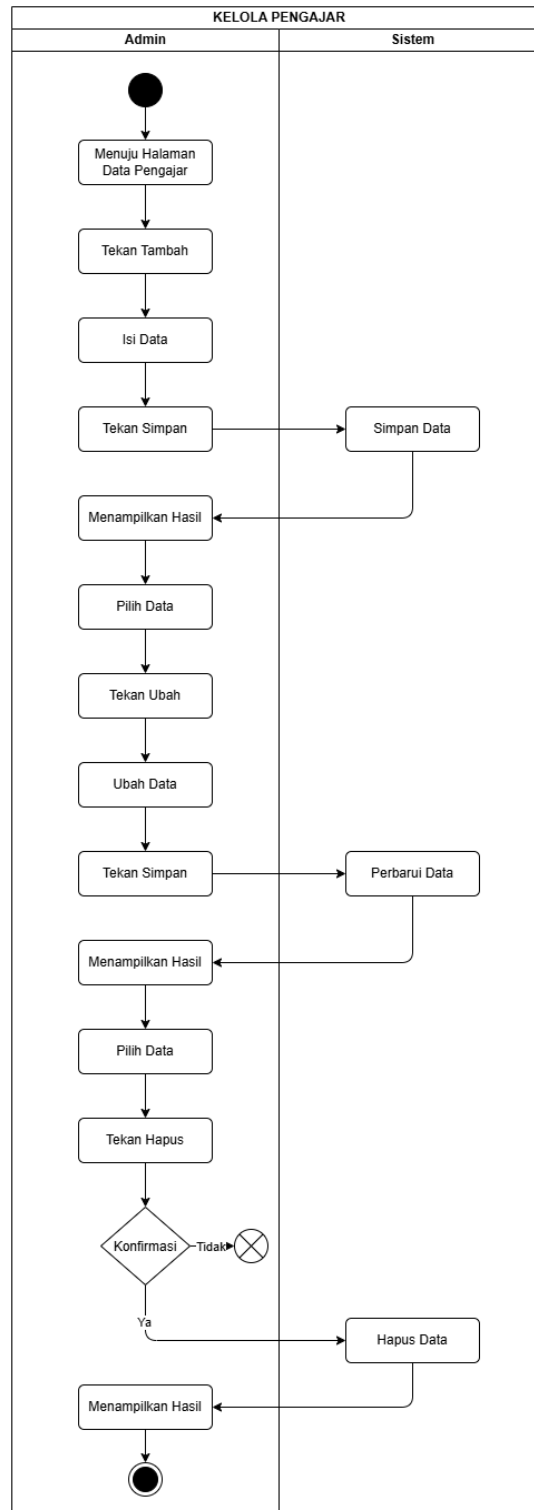
Activity Diagram mengelola ruang kelas menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data ruang kelas.



Gambar 10. Activity Diagram Mengelola Ruang Kelas

8. Activity Diagram Mengelola Pengajar

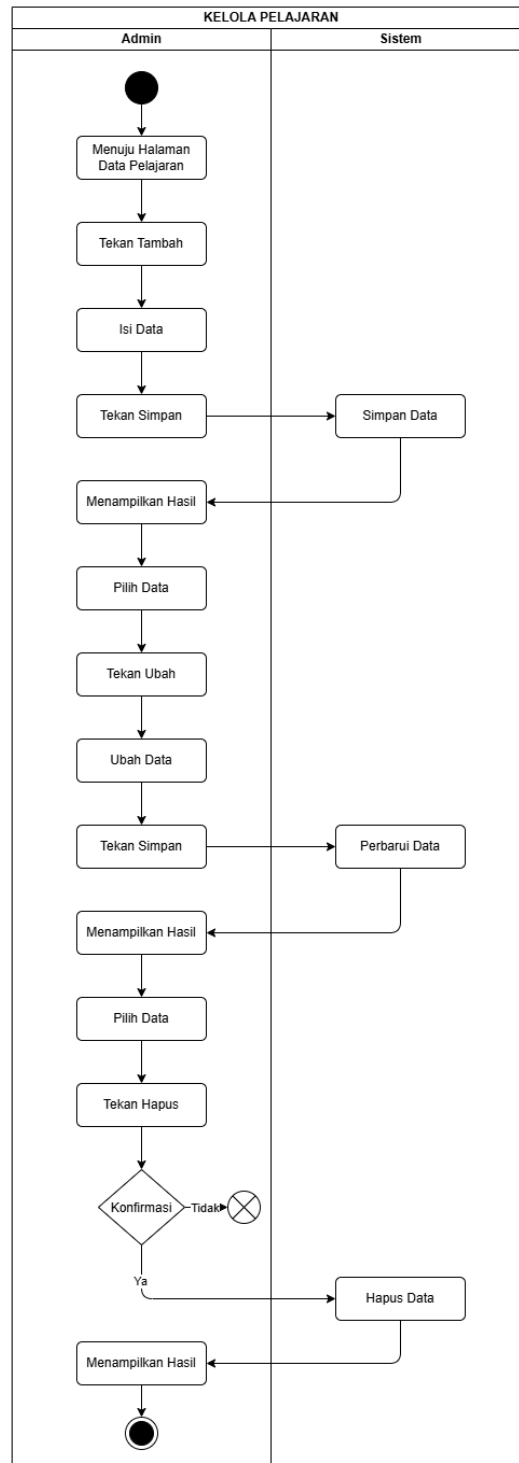
Activity Diagram mengelola pengajar menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data pengajar.



Gambar 11. Activity Diagram Mengelola Pengajar

9. Activity Diagram Mengelola Pelajaran

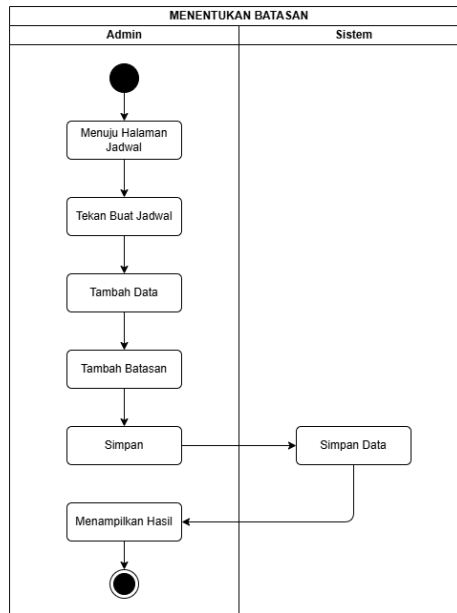
Activity Diagram mengelola pelajaran menjelaskan proses admin menambah, melihat, mengubah, dan menghapus data pelajaran.



Gambar 12. Activity Diagram Mengelola Pelajaran

10. Activity Diagram Menentukan Batasan

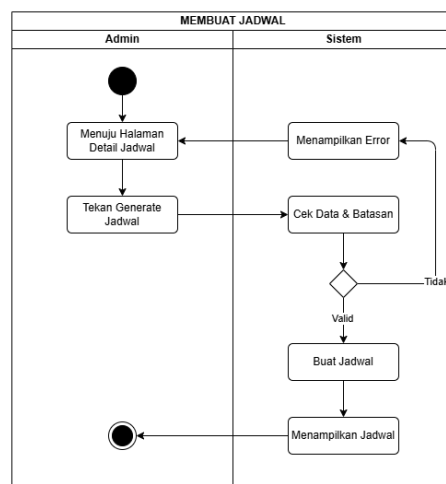
Activity Diagram menentukan batasan menjelaskan proses admin menambahkan batasan dari data-data yang dimasukkan untuk membuat jadwal.



Gambar 13. Activity Diagram Menentukan Batasan

11. Activity Diagram Membuat Jadwal

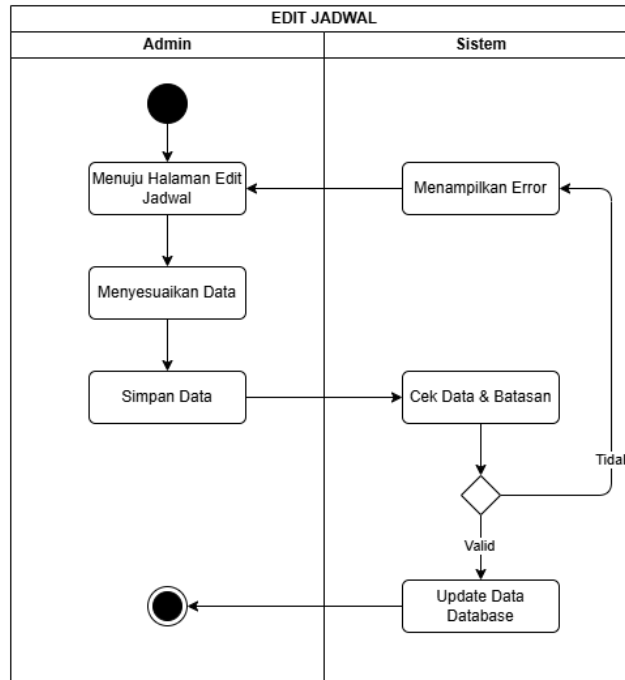
Activity Diagram membuat jadwal menjelaskan proses pembuatan jadwal oleh sistem setelah admin data-data yang diperlukan.



Gambar 14. Activity Diagram Membuat Jadwal

12. Activity Diagram Edit Jadwal

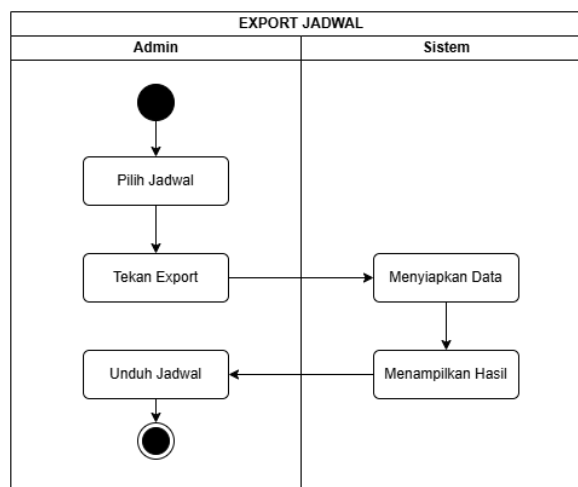
Activity Diagram edit jadwal menjelaskan proses admin melakukan penyesuaian pada data jadwal.



Gambar 15. Activity Diagram Edit Jadwal

13. Activity Diagram Export Jadwal

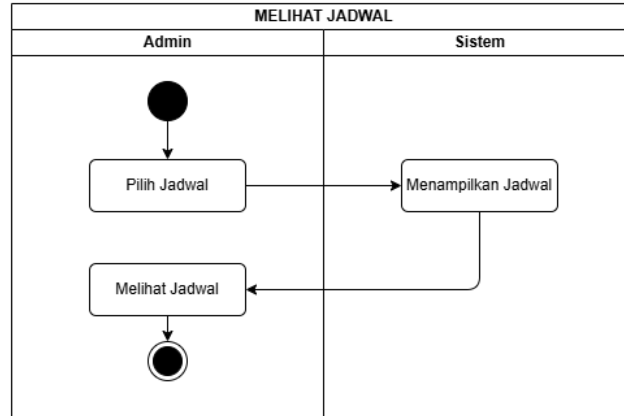
Activity Diagram export jadwal menjelaskan proses admin melakukan export untuk mengunduh jadwal ke format pdf.



Gambar 16. Activity Diagram Export Jadwal

14. Activity Diagram Melihat Jadwal

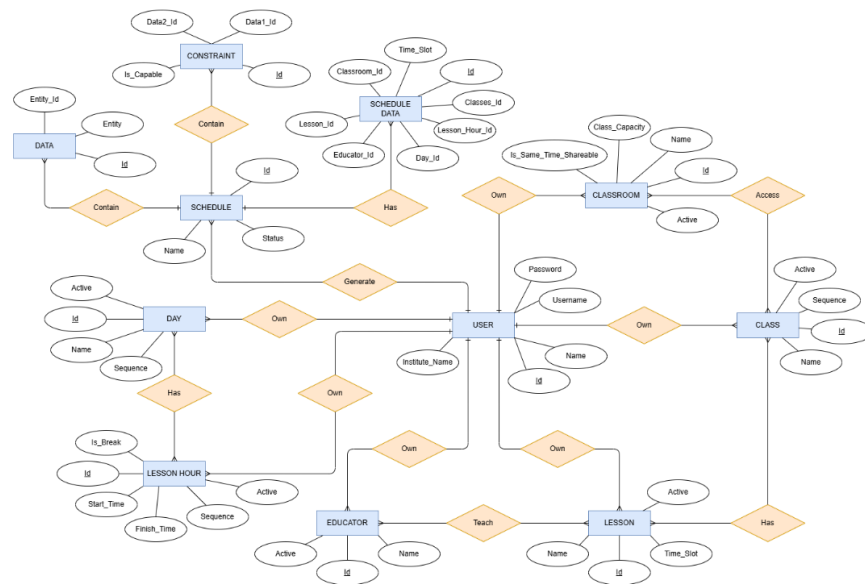
Activity Diagram melihat jadwal menjelaskan proses admin melihat data jadwal yang sudah dibuat sebelumnya.



Gambar 17. Activity Diagram Melihat Jadwal

3.2.6. Entity Relationship Diagram

ERD adalah representasi grafis dari domain masalah yang sedang dimodelkan. *ERD* membantu perancang basis data dalam mengidentifikasi data dan aturan yang akan direpresentasikan dan digunakan dalam basis data. *ERD* merupakan representasi domain masalah yang independen dari implementasi dan memfasilitasi komunikasi antara pengguna akhir dan analis. Komponen dasar *ERD* adalah entitas, properti entitas yang disebut atribut, dan hubungan antar entitas (Pulungan et al., 2022). Berikut adalah rancangan basis data dari sistem yang akan dikembangkan:



Gambar 18. Entity Relationship Diagram

Pada ERD di atas, terdapat entitas *user* yang menampung nama, *username*, *password*, dan nama institusi. Terdapat juga entitas yang diperlukan untuk membuat jadwal, yaitu entitas *day*, *lesson hour*, *class*, *classroom*, *educator*, dan *lesson*. Pada entitas *day*, *lesson hour*, dan *class*, terdapat atribut *sequence* untuk menentukan urutan peletakannya pada jadwal. Pada entitas *lesson* terdapat atribut *time slot* untuk menentukan banyaknya slot waktu yang dipakai untuk *lesson* tersebut, entitas *lesson hour* terdapat atribut *break* untuk menentukan apakah pada jam pembelajaran tersebut adalah jam istirahat, dan entitas *classroom* terdapat atribut *class capacity* untuk menentukan seberapa banyak kelas yang dapat ditampung pada ruang kelas tersebut dan atribut *same time shareable* untuk menentukan apakah ruang kelas tersebut dapat digunakan bersama-sama.

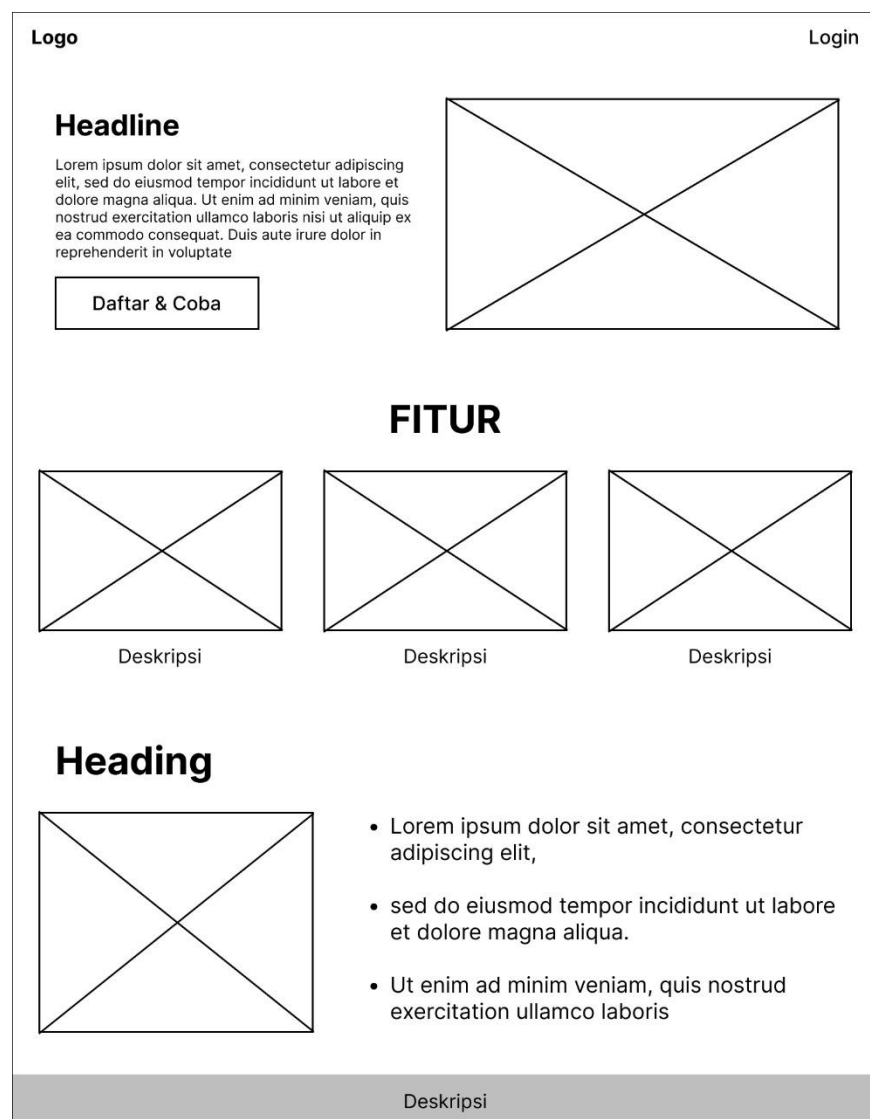
Entitas *schedule* untuk menampung jadwal yang dibuat setiap *user*. Terdapat atribut *status* untuk mengetahui apakah jadwal sudah terbuat atau sedang menentukan data-datanya. Entitas *data* untuk menampung data yang dipilih untuk membuat jadwal. Entitas *constraint* untuk menampung semua batasan dari sebuah *schedule*. Terdapat atribut *is capable* untuk menentukan apakah batasan tersebut adalah bersedia atau tidak bersedia. Entitas *schedule data* adalah data-data dari sebuah jadwal.

3.2.7. Perancangan Antarmuka (*Wireframe*)

Perancangan antarmuka pengguna terdiri dari mendefinisikan elemen antarmuka dengan benar sehingga tugas dan interaksi yang akan dilakukan pengguna mudah dipahami (Diehl et al., 2022). Berikut adalah rancangan antarmuka dari sistem yang akan dikembangkan:

1. *Wireframe Home*

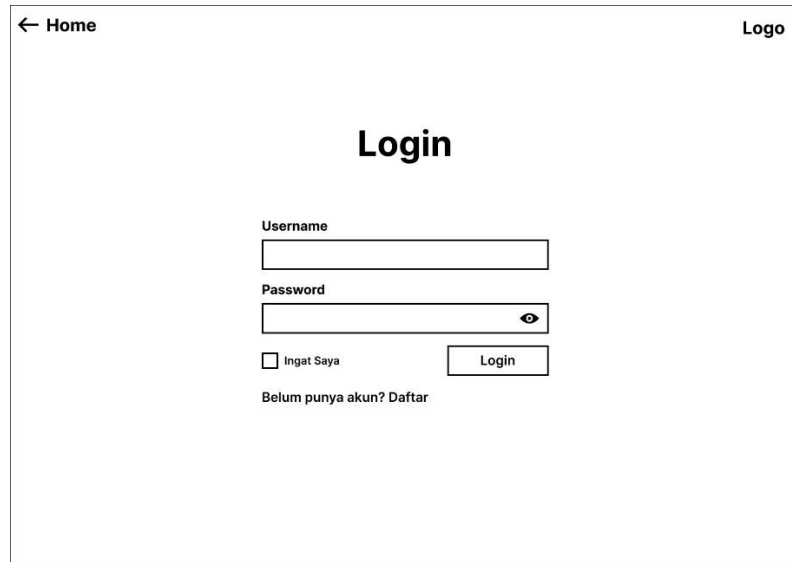
Home adalah halaman pertama saat pengguna akses *website* ini. Pada halaman ini terdapat tombol *login*, deskripsi singkat, fitur, dan keunggulan.



Gambar 19. *Wireframe Home*

2. Wireframe Login

Halaman *login* bertujuan untuk pengguna (admin) dapat masuk ke akunnya dan melakukan pembuatan jadwal. Admin perlu memasukkan *username* dan *password* agar bisa masuk ke akunnya.



The wireframe shows a login page with a header containing a back arrow and 'Home' text on the left, and a 'Logo' on the right. The main heading is 'Login'. Below it are two input fields: 'Username' and 'Password'. The 'Password' field has an eye icon for toggling visibility. Below the password field is a checkbox labeled 'Ingat Saya' and a 'Login' button. At the bottom, there is a link that says 'Belum punya akun? Daftar'.

Gambar 20. Wireframe Login

3. Wireframe Daftar Akun

Halaman daftar akun bertujuan untuk pengguna baru melakukan pendaftaran akun untuk menggunakan fitur yang disediakan. Pengguna perlu mengisi *username*, *password*, dan konfirmasi *password*.



The wireframe shows a registration page with a header containing a back arrow and 'Home' text on the left, and a 'Logo' on the right. The main heading is 'Daftar'. Below it are three input fields: 'Username', 'Password', and 'Konfirmasi Password'. Both the 'Password' and 'Konfirmasi Password' fields have eye icons for toggling visibility. Below the last field is a link that says 'Sudah punya akun? Login' and a 'Daftar' button.

Gambar 21. Wireframe Daftar Akun

4. Wireframe Data Pengguna

Setelah melakukan pendaftaran akun, pengguna akan diarahkan untuk mengisi data nama pengguna dan nama institusi. Setelah semua diisi, pengguna menekan masuk dan sistem akan mengarahkan pengguna ke halaman admin.

Logo

Data Pengguna

Nama Anda

Nama Institusi

Masuk

Gambar 22. Wireframe Data Pengguna

5. Wireframe Data Profil

Halaman data profil berisi seluruh data milik admin yang terdiri dari *username*, nama, nama institusi, dan pengguna juga dapat mengganti *password*.

Logo

Dashboard

Jadwal

Data ^

Hari

Jam Pembelajaran

Kelas

Ruang Kelas

Pengajar

Pelajaran

Data Profil

← Logout

Data Profil

Username

Nama

Nama Institusi

Password Sekarang

Password Baru

Konfirmasi Password Baru

Simpan

Gambar 23. Wireframe Data Profil

6. Wireframe Dashboard

Pada halaman dashboard, admin dapat melihat keseluruhan data yang sudah ditambahkan. Terdapat data jadwal, hari, jam pembelajaran, kelas, ruang kelas, pengajar, dan pelajaran. Pada halaman ini, terdapat tombol untuk langsung melihat data maupun menambah data.

The dashboard wireframe consists of a sidebar menu on the left and a main content area on the right. The sidebar menu includes 'Logo', 'Dashboard' (highlighted), 'Jadwal', 'Data' (with a dropdown arrow), 'Hari', 'Jam Pembelajaran', 'Kelas', 'Ruang Kelas', 'Pengajar', 'Pelajaran', 'Data Profil', and a 'Logout' button. The main content area is titled 'Dashboard' and contains a 'Jadwal' card with a '+' icon and the number '5'. Below this is a 'Data' section with two rows of three cards each. The first row contains 'Hari' (5), 'Jam Pembelajaran' (9), and 'Kelas' (12). The second row contains 'Ruang Kelas' (15), 'Pengajar' (20), and 'Pelajaran' (30). Each card has a '+' icon and a right-pointing arrow.

Gambar 24. Wireframe Dashboard

7. Wireframe Data Hari

Pada halaman data hari, admin bisa melihat, menambah, mengubah, dan menghapus data hari. Terdapat opsi untuk menampilkan data yang diarsip dan fitur *import* untuk menambah data secara massal. Terdapat kolom nama, aktif, dan urutan untuk menentukan urutan data dari jadwal.

The 'Data Hari' wireframe shows a table with 5 entries. Above the table are buttons for 'Tambah', 'Ubah', 'Import', and 'Hapus', along with a checkbox for 'Tampilkan data arsip'. Below the buttons are controls for 'entries per page' (set to 10) and a 'Search' field. The table has columns for 'Nama', 'Urutan', and 'Aktif'. Each row represents a day of the week, with checkboxes for 'Aktif' and a trash icon for deletion. The footer shows 'Showing 1 to 5 of 5 entries' and a pagination control.

Nama	Urutan	Aktif
☒ Senin	1	☒
☒ Selasa	2	☒
☐ Rabu	3	☒
☐ Kamis	4	☒
☐ Jumat	5	☒

Gambar 25. Wireframe Data Hari

8. Wireframe Data Jam Pembelajaran

Pada halaman data jam pembelajaran, admin bisa melihat, menambah, mengubah, dan menghapus data jam pembelajaran. Terdapat opsi untuk menampilkan data yang diarsip dan fitur *import* untuk menambah data secara massal. Terdapat kolom mulai, selesai, aktif, urutan untuk menentukan urutan data dari jadwal, dan kolom istirahat untuk menentukan apakah pada jam tersebut merupakan jam istirahat.

Logo Dashboard Jadwal Data ^ Hari Jam Pembelajaran Kelas Ruang Kelas Pengajar Pelajaran Data Profil ← Logout	Data Jam Pembelajaran Jumlah data: 9							
	Tambah		Ubah		☐ Tampilkan data arsip		Import	
							Hapus	
	10 ▾		entries per page		Search:			
	☐	Mulai	Selesai	Urutan	Istirahat	Aktif		
	☒	08:00	08:40	1		✓	🗑	
	☒	08:40	09:20	2		✓	🗑	
	☐	09:20	10:00	3		✓	🗑	
	☐	10:00	10:40	4	✓	✓	🗑	
	☐	10:40	11:20	5		✓	🗑	
☐	11:20	12:00	6		✓	🗑		
☐	12:00	12:40	7		✓	🗑		
☐	12:40	13:20	8		✓	🗑		

Gambar 26. Wireframe Data Jam Pembelajaran

9. Wireframe Data Kelas

Pada halaman data kelas, admin bisa melihat, menambah, mengubah, dan menghapus data kelas. Terdapat opsi untuk menampilkan data yang diarsip dan fitur *import* untuk menambah data secara massal. Terdapat kolom nama, aktif, dan urutan untuk menentukan urutan data dari jadwal.

Logo

Dashboard

Jadwal

Data ^

Hari

Jam Pembelajaran

Kelas

Ruang Kelas

Pengajar

Pelajaran

Data Profil

← Logout

Data Kelas

Jumlah data: 12

Tambah

Ubah

☐ Tampilkan data arsip

Import

Hapus

10 ▾

entries per page

Search:

☐	Nama	Urutan	Aktif	
☒	7A	1	✓	
☒	7B	2	✓	
☐	7C	3	✓	
☐	8A	4	✓	
☐	8B	5	✓	
☐	8C	6	✓	
☐	9A	7	✓	
☐	9B	8	✓	

Gambar 27. Wireframe Data Kelas

10. Wireframe Data Ruang Kelas

Pada halaman data ruang kelas, admin bisa melihat, menambah, mengubah, dan menghapus data ruang kelas. Terdapat opsi untuk menampilkan data yang diarsip dan fitur *import* untuk menambah data secara massal. Terdapat kolom nama, aktif, kapasitas untuk menentukan apakah ruang kelas tersebut bisa menampung seberapa banyak kelas, dan kolom berbagi kelas untuk menentukan apakah ruang kelas tersebut dapat digunakan secara bersamaan dengan kelas lain.

Logo

Dashboard

Jadwal

Data ^

Hari

Jam Pembelajaran

Kelas

Ruang Kelas

Pengajar

Pelajaran

Data Profil

← Logout

Data Ruang Kelas

Jumlah data: 15

Tambah

Ubah

☐ Tampilkan data arsip

Import

Hapus

10 ▾

entries per page

Search:

☐	Nama	Kapasitas Kelas	Berbagi Kelas	Aktif	
☒	R101	1		✓	
☒	R102	2	✓	✓	
☐	R103	1		✓	
☐	R104	3	✓	✓	
☐	R105	1		✓	
☐	R106	1		✓	
☐	R107	1		✓	
☐	R108	1		✓	

Gambar 28. Wireframe Data Ruang Kelas

11. Wireframe Data Pengajar

Pada halaman data pengajar, admin bisa melihat, menambah, mengubah, dan menghapus data pengajar. Terdapat opsi untuk menampilkan data yang diarsip dan fitur *import* untuk menambah data secara massal. Terdapat kolom nama dan aktif.

Logo	Data Pengajar		Jumlah data: 20																																					
	Dashboard Jadwal Data ^ Hari Jam Pembelajaran Kelas Ruang Kelas Pengajar Pelajaran Data Profil ← Logout	Tambah Ubah ☐ Tampilkan data arsip Import Hapus 10 ▼ entries per page Search: <table border="1"> <thead> <tr> <th>☐</th> <th>Nama</th> <th>Aktif</th> <th></th> </tr> </thead> <tbody> <tr> <td>☒</td> <td>Budi</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☒</td> <td>Anto</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>Herdi</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>Susanto</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>Susi</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>Wati</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>Rusdi</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>Lusi</td> <td>✓</td> <td> Hapus </td> </tr> </tbody> </table>				☐	Nama	Aktif		☒	Budi	✓	Hapus	☒	Anto	✓	Hapus	☐	Herdi	✓	Hapus	☐	Susanto	✓	Hapus	☐	Susi	✓	Hapus	☐	Wati	✓	Hapus	☐	Rusdi	✓	Hapus	☐	Lusi	✓
☐	Nama	Aktif																																						
☒	Budi	✓	Hapus																																					
☒	Anto	✓	Hapus																																					
☐	Herdi	✓	Hapus																																					
☐	Susanto	✓	Hapus																																					
☐	Susi	✓	Hapus																																					
☐	Wati	✓	Hapus																																					
☐	Rusdi	✓	Hapus																																					
☐	Lusi	✓	Hapus																																					

Gambar 29. Wireframe Data Pengajar

12. Wireframe Data Pelajaran

Pada halaman data pelajaran, admin bisa melihat, menambah, mengubah, dan menghapus data pelajaran. Terdapat opsi untuk menampilkan data yang diarsip dan fitur *import* untuk menambah data secara massal. Terdapat kolom nama, aktif, dan slot waktu untuk menentukan berapa jam pembelajaran yang dibutuhkan setiap pelajaran.

Logo	Data Pelajaran		Jumlah data: 30																																														
	Dashboard Jadwal Data ^ Hari Jam Pembelajaran Kelas Ruang Kelas Pengajar Pelajaran Data Profil ← Logout	Tambah Ubah ☐ Tampilkan data arsip Import Hapus 10 ▼ entries per page Search: <table border="1"> <thead> <tr> <th>☐</th> <th>Nama</th> <th>Slot Waktu</th> <th>Aktif</th> <th></th> </tr> </thead> <tbody> <tr> <td>☒</td> <td>Matematika</td> <td>3</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☒</td> <td>B.Indo</td> <td>2</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>B.Ing</td> <td>3</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>IPA</td> <td>3</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>IPS</td> <td>2</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>SBK</td> <td>1</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>PJOK</td> <td>3</td> <td>✓</td> <td> Hapus </td> </tr> <tr> <td>☐</td> <td>PPKn</td> <td>3</td> <td>✓</td> <td> Hapus </td> </tr> </tbody> </table>				☐	Nama	Slot Waktu	Aktif		☒	Matematika	3	✓	Hapus	☒	B.Indo	2	✓	Hapus	☐	B.Ing	3	✓	Hapus	☐	IPA	3	✓	Hapus	☐	IPS	2	✓	Hapus	☐	SBK	1	✓	Hapus	☐	PJOK	3	✓	Hapus	☐	PPKn	3	✓
☐	Nama	Slot Waktu	Aktif																																														
☒	Matematika	3	✓	Hapus																																													
☒	B.Indo	2	✓	Hapus																																													
☐	B.Ing	3	✓	Hapus																																													
☐	IPA	3	✓	Hapus																																													
☐	IPS	2	✓	Hapus																																													
☐	SBK	1	✓	Hapus																																													
☐	PJOK	3	✓	Hapus																																													
☐	PPKn	3	✓	Hapus																																													

Gambar 30. Wireframe Data Pelajaran

13. Wireframe Data Jadwal

Pada halaman data jadwal, admin bisa melihat, menambah, dan menghapus data jadwal. Admin perlu masuk ke salah satu data jadwal untuk mengubah data maupun melakukan *export*.

Logo Dashboard Jadwal Data ^ Hari Jam Pembelajaran Kelas Ruang Kelas Pengajar Pelajaran Data Profil ← Logout	Data Jadwal Jumlah data: 5			
	Tambah		Hapus	
	10 entries per page		Search:	
	☐	Nama	Publikasi	Status
	☒	2025-2026 Ganjil	✓	Draft 🗑
	☒	2024-2025 Genap	✓	Selesai 🗑
	☐	2024-2025 Ganjil	✓	Selesai 🗑
	☐	2023-2024 Genap	✓	Selesai 🗑
	☐	2023-2024 Ganjil	✓	Selesai 🗑
	Showing 1 to 5 of 5 entries		< 1 >	

Gambar 31. Wireframe Data Jadwal

14. Wireframe Buat Jadwal

Halaman buat jadwal berfungsi untuk admin menambahkan data dan batasan dari data-data. Kemudian membuat jadwal dengan menekan tombol Buat Jadwal.

Logo

Dashboard

Jadwal

Data ^

Hari

Jam Pembelajaran

Kelas

Ruang Kelas

Pengajar

Pelajaran

Data Profil

← Logout

Jadwal

Batal

Tambah Data

Status: Draft

Tambah Batasan

Buat Jadwal

Data

Lihat Data Hari

Lihat Data Jam

Lihat Data Kelas

Lihat Data Ruang

Lihat Data Pengajar

Lihat Data Pelajaran

Batasan

10 ▾

entries per page

Search:

Data1	Data2	Bersedia
Pengajar - Rusdi	Jam Pembelajaran - 10:40 - 11:20	☐
Ruang Kelas - R101	Jam Pembelajaran - 10:40 - 11:20	☐
Pengajar - Rusdi	Hari - Kamis	☒
Kelas - 7C	Pelajaran - PJOK	☐
Kelas - 7C	Ruang Kelas - R103	☒
Hari - Senin	Pelajaran - PJOK	☐
Ruang Kelas - R101	Pelajaran - IPA	☒

Gambar 32. Wireframe Buat Jadwal

15. Wireframe Detail Jadwal

Halaman detail jadwal adalah halaman dimana jadwal sudah berhasil terbuat dan admin dapat melihat data jadwal dan mengisi nama dari jadwal. Terdapat fitur *export* untuk admin mengunduh data jadwal ke format *pdf* maupun *xlsx*. Pada halaman ini, admin bisa memilih untuk mempublikasikan jadwal atau tidak dan bisa memilih untuk menggunakan pin atau tidak. Sistem akan otomatis buatkan *link* untuk dibagikan agar pengguna publik dapat mengakses dan lihat jadwal.

Logo

Dashboard

Jadwal

Data

Hari

Jam Pembelajaran

Kelas

Ruang Kelas

Pengajar

Pelajaran

Data Profil

← Logout

Jadwal - 2024-2025 Ganjil

Ubah

Export

Hapus

Status: Selesai

Nama

2025-2026 Ganjil

Publikasi

☒

Pin

Link

https://schedule.com/as9d87as6d9as8d09a8sda

Filter

Batasan

10

▼

entries per page

Search:

Data1	Data2	Bersedia
Pengajar - Rusdi	Jam Pembelajaran - 10:40 - 11:20	☐
Ruang Kelas - R101	Jam Pembelajaran - 10:40 - 11:20	☐
Pengajar - Rusdi	Hari - Kamis	☒
Kelas - 7C	Pelajaran - PJOK	☐
Kelas - 7C	Ruang Kelas - R103	☒
Hari - Senin	Pelajaran - PJOK	☐
Ruang Kelas - R101	Pelajaran - IPA	☒

Gambar 33. Wireframe Detail Jadwal

3.2.8. Rancangan Proses *Generate Jadwal*

Rancangan proses *generate* jadwal ini bertujuan untuk memudahkan penulis dan pembaca dalam memahami alur dan program yang akan dirancang pada tahap implementasi yang disajikan dalam bentuk *pseudocode*, sebagai berikut:

```
INPUT data hari
INPUT data jam pembelajaran
INPUT data kelas
INPUT data ruang kelas
INPUT data pengajar
INPUT data pelajaran
INPUT data batasan

VALIDASI kelengkapan data
VALIDASI ketersediaan slot pembelajaran terhadap total slot pelajaran
VALIDASI ketersediaan ruang kelas terhadap total kelas
VALIDASI ketersediaan pengajar

PEMBENTUKAN model constraint
MEMBENTUK variabel untuk pengambilan keputusan oleh model
MENAMBAHKAN batasan 1 pelajaran per kelas dalam 1 waktu
MENAMBAHKAN batasan agar pelajaran tidak bentrok
MENAMBAHKAN batasan agar pengajar tidak bentrok

MENJALANKAN solver

OUTPUT jadwal
```


BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Hasil Implementasi

Bagian ini menjelaskan hasil implementasi dari pengembangan aplikasi web yang telah dirancang. Tujuan dari penjabaran ini adalah untuk menunjukkan bahwa produk telah berhasil dibangun sesuai dengan rancangan awal.

4.1.1. Lingkungan Implementasi dan Instalasi Sistem

Sistem ini diimplementasikan menggunakan kode editor VS Code dengan teknologi yang digunakan mencakup:

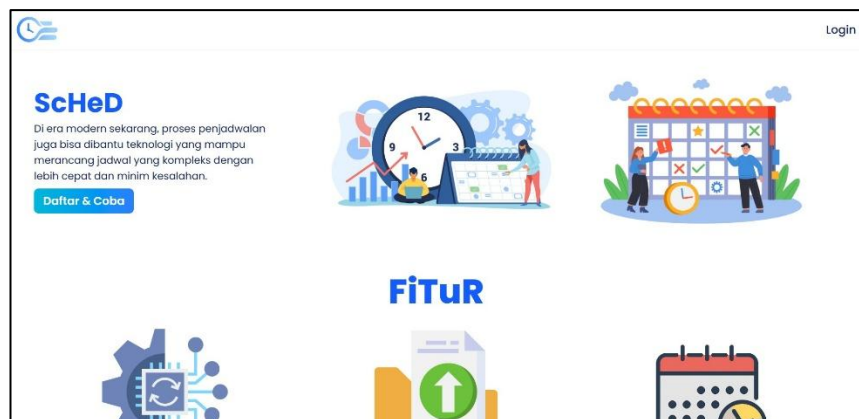
- *HTML*, *CSS*, dan *JavaScript*, sebagai bahasa yang digunakan untuk pengembangan *front end* sistem.
- *Python* sebagai bahasa pemrograman berorientasi objek untuk mendukung logika-logika yang dibutuhkan.
- *PostgreSQL* sebagai penampung data.

4.1.2. Antarmuka dan Alur Penggunaan Sistem

Berikut adalah hasil implementasi antarmuka pengguna beserta penjelasan alur kerja sistem:

1. Halaman *Home*

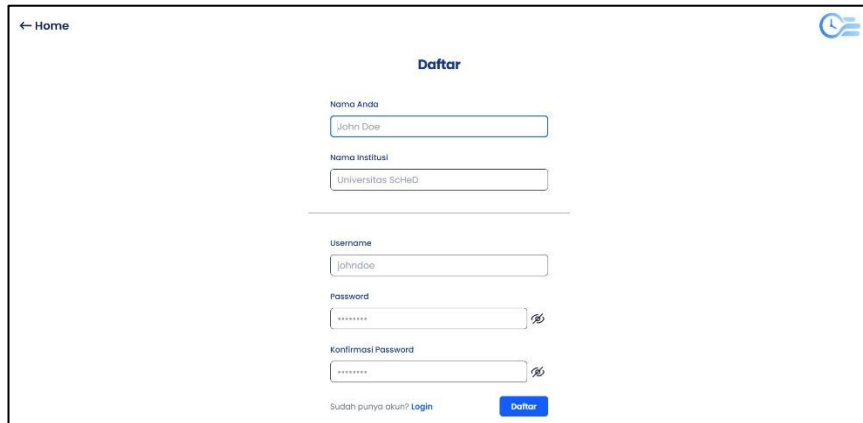
Home adalah halaman pertama saat pengguna akses *website* ini. Pada halaman ini terdapat tombol *login*, daftar, deskripsi singkat, fitur, dan keunggulan.



Gambar 34. Antarmuka Home

2. Halaman Daftar

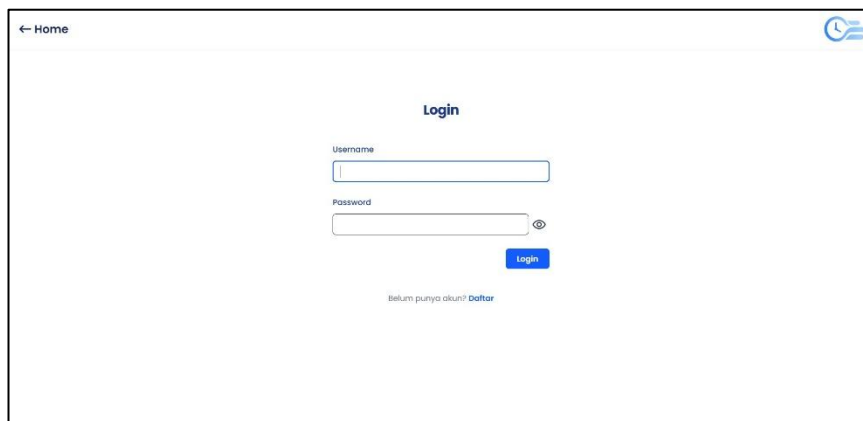
Halaman daftar bertujuan untuk pengguna baru melakukan pendaftaran akun untuk menggunakan fitur yang disediakan. Pengguna perlu mengisi *username*, *password*, dan konfirmasi *password*.

The screenshot shows a web page titled "Daftar" (Registration). At the top left is a "← Home" link, and at the top right is a clock icon. The form contains the following fields: "Nama Anda" (Your Name) with the value "John Doe", "Nama Institusi" (Institution Name) with the value "Universitas SciHeD", "Username" with the value "johndoe", "Password" (masked with dots), and "Konfirmasi Password" (also masked with dots). There are eye icons to toggle password visibility. At the bottom, there is a link "Sudah punya akun? Login" and a blue "Daftar" button.

Gambar 35. Halaman Daftar

3. Halaman Login

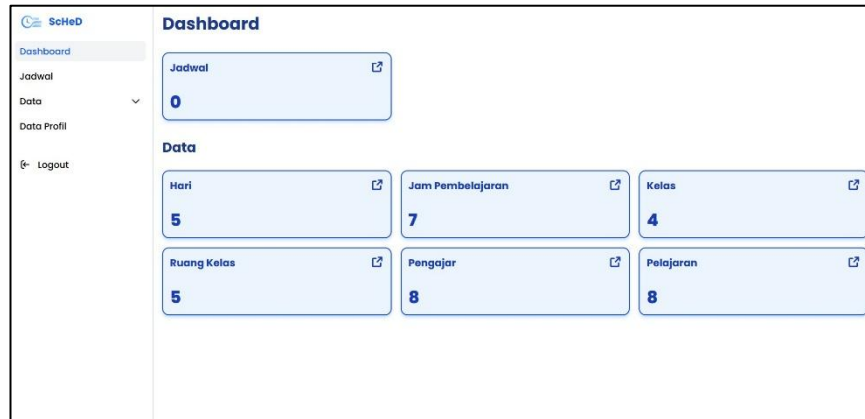
Halaman *login* bertujuan untuk pengguna (admin) dapat masuk ke akunnya dan melakukan pembuatan jadwal. Admin perlu memasukkan *username* dan *password* agar bisa masuk ke akunnya.

The screenshot shows a web page titled "Login". At the top left is a "← Home" link, and at the top right is a clock icon. The form contains the following fields: "Username" and "Password" (masked with dots). There is an eye icon to toggle password visibility. At the bottom, there is a link "Belum punya akun? Daftar" and a blue "Login" button.

Gambar 36. Halaman Login

4. Halaman Dasbor

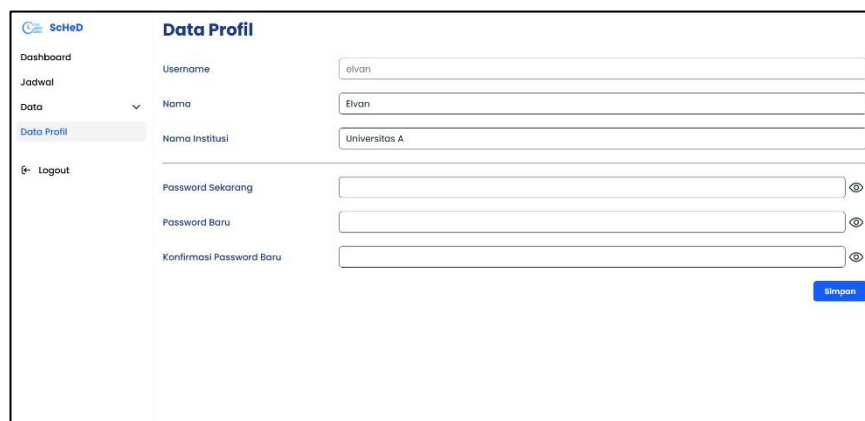
Pada halaman dasbor, admin dapat melihat keseluruhan data yang sudah ditambahkan. Terdapat data jadwal, hari, jam pembelajaran, kelas, ruang kelas, pengajar, dan pelajaran. Pada halaman ini, terdapat tombol untuk langsung melihat data.



Gambar 37. Halaman Dasbor

5. Halaman Data Profil

Halaman data profil berisi seluruh data milik admin yang terdiri dari *username*, nama, nama institusi, dan pengguna juga dapat mengganti *password*.



Gambar 38. Halaman Data Profil

6. Halaman Data Hari

Pada halaman data hari, admin bisa melihat, menambah, mengubah, dan menghapus data hari. Terdapat kolom nama, aktif, dan urutan untuk menentukan urutan data dari jadwal.

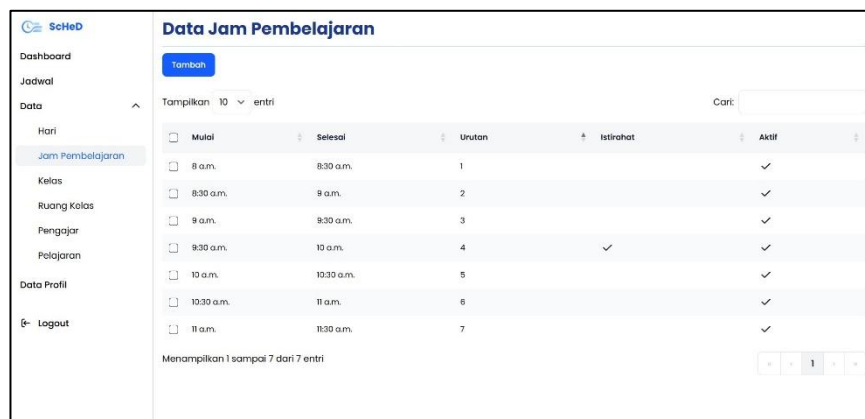


Nama	Urutan	Aktif
☐ Senin	1	✓
☐ Selasa	2	✓
☐ Rabu	3	✓
☐ Kamis	4	✓
☐ Jumat	5	✓

Gambar 39. Halaman Data Hari

7. Halaman Data Jam

Pada halaman data jam, admin bisa melihat, menambah, mengubah, dan menghapus data jam pembelajaran. Terdapat kolom mulai, selesai, aktif, urutan untuk menentukan urutan data dari jadwal, dan kolom istirahat untuk menentukan apakah pada jam tersebut merupakan jam istirahat.



Mulai	Selesai	Urutan	Istirahat	Aktif
☐ 8 a.m.	8:30 a.m.	1		✓
☐ 8:30 a.m.	9 a.m.	2		✓
☐ 9 a.m.	9:30 a.m.	3		✓
☐ 9:30 a.m.	10 a.m.	4	✓	✓
☐ 10 a.m.	10:30 a.m.	5		✓
☐ 10:30 a.m.	11 a.m.	6		✓
☐ 11 a.m.	11:30 a.m.	7		✓

Gambar 40. Halaman Data Jam

8. Halaman Data Kelas

Pada halaman data kelas, admin bisa melihat, menambah, mengubah, dan menghapus data kelas. Terdapat kolom nama, aktif, dan urutan untuk menentukan urutan data dari jadwal.



Gambar 41. Halaman Data Kelas

9. Halaman Data Ruang

Pada halaman data ruang, admin bisa melihat, menambah, mengubah, dan menghapus data ruang kelas. Terdapat kolom nama, aktif, kapasitas untuk menentukan apakah ruang kelas tersebut bisa menampung seberapa banyak kelas, dan kolom berbagi kelas untuk menentukan apakah ruang kelas tersebut dapat digunakan secara bersamaan dengan kelas lain.



Gambar 42. Halaman Data Ruang

10. Halaman Data Pengajar

Pada halaman data pengajar, admin bisa melihat, menambah, mengubah, dan menghapus data pengajar. Terdapat kolom nama dan aktif.



Gambar 43. Halaman Data Pengajar

11. Halaman Data Pelajaran

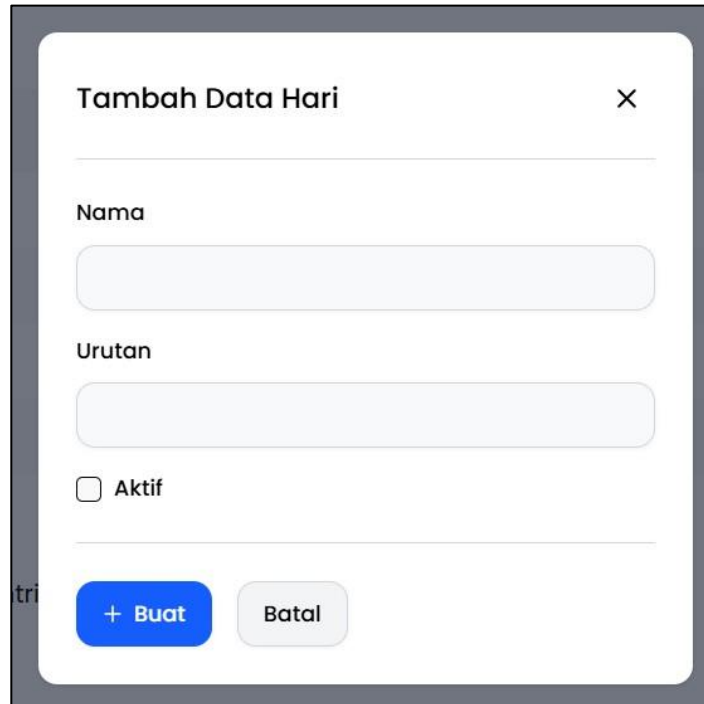
Pada halaman data pelajaran, admin bisa melihat, menambah, mengubah, dan menghapus data pelajaran. Terdapat kolom nama, aktif, dan slot waktu untuk menentukan berapa jam pembelajaran yang dibutuhkan setiap pelajaran.



Gambar 44. Halaman Data Pelajaran

12. Halaman Tambah Data

Pada *popup* ini, pengguna perlu mengisi data untuk bisa menambahkan data.



Tambah Data Hari

Nama

Urutan

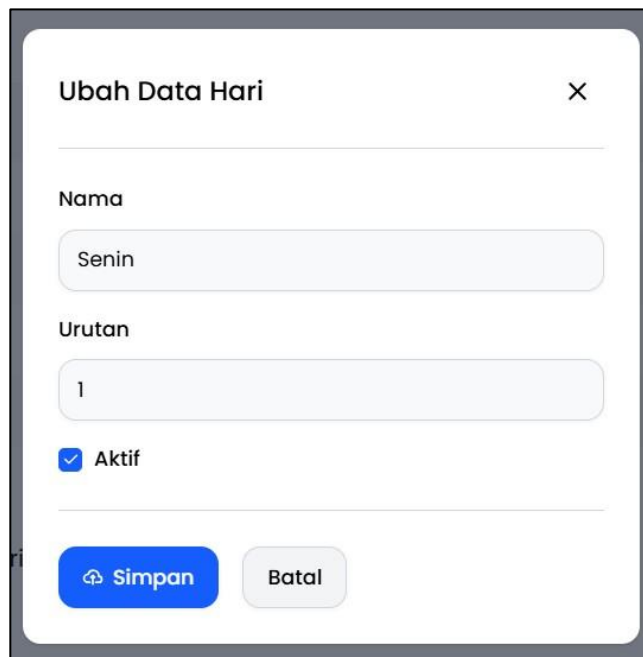
☐ Aktif

+ Buat Batal

Gambar 45. Halaman Tambah Data

13. Halaman Ubah Data

Pada *popup* ini, pengguna bisa mengubah data untuk melakukan perubahan data.



Ubah Data Hari

Nama

Senin

Urutan

1

☒ Aktif

Simpan Batal

Gambar 46. Halaman Ubah Data

14. Halaman Hapus Data

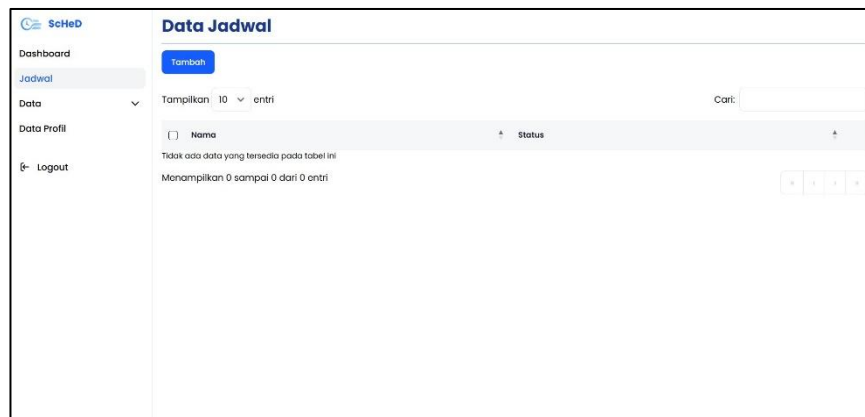
Pada *popup* ini, pengguna bisa melakukan penghapusan data dan bisa menghapus banyak data dalam 1 kali.



Gambar 47. Halaman Hapus Data

15. Halaman Jadwal

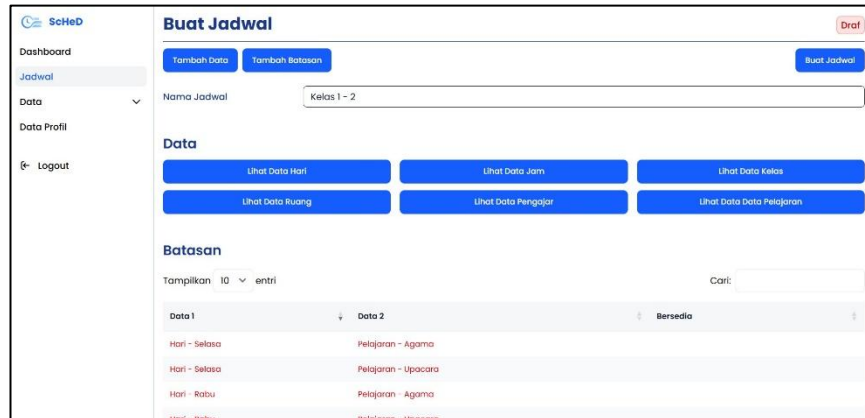
Pada halaman jadwal, admin bisa melihat, menambah, dan menghapus data jadwal. Admin perlu masuk ke salah satu data jadwal untuk mengubah data maupun melakukan *export*.



Gambar 48. Halaman Jadwal

16. Halaman Tambah Jadwal

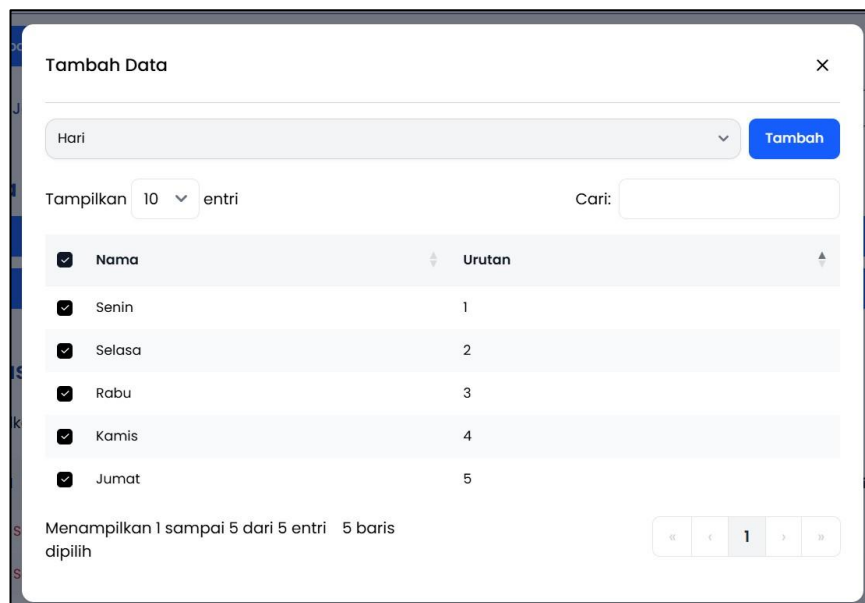
Halaman tambah jadwal berfungsi untuk admin menambahkan data dan batasan dari data-data yang sudah dibuat sebelumnya. Kemudian membuat jadwal dengan menekan tombol Buat Jadwal.



Gambar 49. Halaman Tambah Jadwal

17. Halaman Tambah Data Jadwal

Pada *popup* ini, pengguna dapat menambahkan data yang dibuat sebelumnya untuk membuat jadwal.



Gambar 50. Halaman Tambah Data Jadwal

18. Halaman Tambah Batasan Data Jadwal

Pada *popup* ini, pengguna dapat menambahkan batasan pada data yang dipilih untuk membuat jadwal.

Tambah Batasan

Data 1

Hari

Tampilkan 10 entri

Cari:

☐ Nama	Urutan
☐ Senin	1
☐ Selasa	2
☐ Rabu	3

Menampilkan 1 sampai 3 dari 3 entri

Data 2

Pilih Data

☐ Bersedia

Tambah

Gambar 51. Halaman Tambah Batasan Data Jadwal

19. Halaman Data yang Ditambah

Pada *popup* ini, pengguna dapat melihat data-data yang ditambahkan untuk membuat jadwal.

Data Hari

Nama	Urutan
Senin	1
Selasa	2
Rabu	3

Menampilkan 1 sampai 3 dari 3 entri

Gambar 52. Halaman Data yang Ditambah

20. Halaman Data Jadwal

Halaman data jadwal adalah halaman dimana jadwal sudah berhasil terbuat dan admin dapat melihat data jadwal. Terdapat fitur *export* untuk admin mengunduh data jadwal ke format *pdf*.

The screenshot shows the 'Jadwal | Kelas 1 - 2' interface. It includes a sidebar with 'Dashboard', 'Jadwal', 'Data', 'Data Profil', and 'Logout'. The main area displays a grid of schedule data for Class 1-2, with columns for 'Hari' (Day), 'Jam' (Time), 'Pelajaran' (Subject), 'Pengajar' (Teacher), and 'Ruang' (Room). The schedule is organized by day (Senin, Selasa, Rabu) and time slots (08:00-08:30, 08:30-09:00, 09:00-09:30, 09:30-10:00, 10:00-10:30, 10:30-11:00, 11:00-11:30). Subjects include Agama, MTK, SBK, PJK, TIK, B. Indo, B. Ing, and Upacara. Teachers listed are Budi, Susi, Anto, Herdi, and Susanto. Room numbers are 101, 102, 103, and 104. The interface also includes buttons for 'Export', 'Ubah', 'Selesai', and 'Hapus'.

Gambar 53. Halaman Data Jadwal

21. Tampilan Jadwal dalam PDF

Tampilan jadwal dalam *pdf* bertujuan untuk admin dapat membagikan hasil jadwal kepada orang lain agar dapat melihat jadwal yang akan digunakan untuk kegiatan belajar-mengajar per kelas.

The screenshot shows a PDF schedule for Class 1A. The table has columns for 'Hari' (Day), 'Jam' (Time), 'Pelajaran' (Subject), 'Pengajar' (Teacher), and 'Ruang' (Room). The schedule is organized by day (Senin, Selasa, Rabu) and time slots (08:00-08:30, 08:30-09:00, 09:00-09:30, 09:30-10:00, 10:00-10:30, 10:30-11:00, 11:00-11:30). Subjects include Agama, MTK, SBK, PJK, TIK, B. Indo, B. Ing, and Upacara. Teachers listed are Budi, Susi, Anto, Herdi, and Susanto. Room numbers are 101, 102, 103, and 104. The interface also includes buttons for 'Export', 'Ubah', 'Selesai', and 'Hapus'.

Gambar 54. Tampilan PDF Jadwal

4.2. Pengujian

Bagian ini memuat hasil pengujian sistem yang dilakukan dengan metode *black box testing*, yaitu metode pengujian untuk mengevaluasi fungsionalitas aplikasi tanpa melihat kode atau struktur internal.

Berikut adalah rekapan *black box testing* yang telah dilaksanakan:

Tabel 19. Tabel Pengujian

Test Case ID	Nama Fitur	Kondisi Awal	Skenario Pengujian	Ekspetasi Hasil	Hasil Aktual	Status
TC-01	Melakukan Daftar (Berhasil)	Pengguna berada pada halaman daftar	Pengguna memasukkan <i>username</i> , <i>password</i> , dan konfirmasi <i>password</i>	Berhasil dan menuju halaman data pengguna	Berhasil dan menuju halaman data pengguna	Lulus
TC-02	Melakukan Daftar (Gagal)	Pengguna berada pada halaman daftar	Pengguna memasukkan nama, nama institusi, <i>username</i> yang sudah ada, atau <i>password</i> dan konfirmasi <i>password</i> berbeda / terlalu sederhana	Muncul pesan error	Muncul pesan error	Lulus
TC-03	Melakukan Login (Berhasil)	Pengguna berada pada halaman <i>login</i>	Pengguna memasukkan <i>username</i> dan <i>password</i> dengan benar	Berhasil <i>login</i> dan masuk ke halaman dasbor	Berhasil <i>login</i> dan masuk ke halaman dasbor	Lulus

TC-04	Melakukan <i>Login</i> (Gagal)	Pengguna berada pada halaman <i>login</i>	Pengguna menggunakan <i>username</i> yang salah atau <i>password</i> yang salah	Muncul pesan error	Muncul pesan error	Lulus
TC-05	Perubahan Data Profil (Berhasil)	Pengguna berada pada halaman data profil	Pengguna mengubah nama, nama institusi, atau <i>password</i> dengan tepat	Berhasil melakukan perubahan data profil	Berhasil melakukan perubahan data profil	Lulus
TC-06	Perubahan Data Profil (Gagal)	Pengguna berada pada halaman data profil	Pengguna mengisi data nama, nama institusi, atau <i>password</i> dengan data yang tidak valid	Muncul pesan error	Muncul pesan error	Lulus
TC-07	Melihat Data	Pengguna berada pada salah satu halaman data	Sistem menampilkan data	Pengguna bisa melihat data	Pengguna bisa melihat data	Lulus
TC-08	Menambah Data	Pengguna menekan tombol tambah data	Pengguna mengisi seluruh data yang diperlukan	Berhasil menambahkan data	Berhasil menambahkan data	Lulus
TC-09	Mengubah Data	Pengguna menekan tombol ubah data	Pengguna melakukan perubahan data	Berhasil mengubah data	Berhasil mengubah data	Lulus
TC-10	Menghapus Data	Pengguna menekan tombol hapus data	Pengguna mengonfirmasi penghapusan jumlah data yang dipilih	Berhasil menghapus data	Berhasil menghapus data	Lulus

TC-11	Menambah Jadwal	Pengguna berada pada halaman jadwal	Pengguna menekan tombol tambah	Pengguna akan dibawa ke halaman buat jadwal	Pengguna akan dibawa ke halaman buat jadwal	Lulus
TC-12	Menambah Data Jadwal	Pengguna menekan tombol tambah data	Pengguna memilih tipe data dan data yang akan ditambahkan, kemudian tekan tombol tambah	Berhasil menambahkan data jadwal	Berhasil menambahkan data jadwal	Lulus
TC-13	Menambah Batasan Jadwal	Pengguna menekan tombol tambah batasan	Pengguna memilih data 1 dan data 2 dan menentukan apakah bersedia atau tidak sebagai batasannya	Berhasil menambahkan batasan jadwal	Berhasil menambahkan batasan jadwal	Lulus
TC-14	Lihat Data Jadwal	Pengguna telah menambahkan data jadwal sebelumnya	Pengguna menekan salah satu tombol lihat data	Data yang ditambahkan akan ditampilkan	Data yang ditambahkan akan ditampilkan	Lulus
TC-15	Lihat Batasan Jadwal	Pengguna telah menambahkan batasan jadwal sebelumnya	Pengguna melihat tabel batasan	Data batasan yang ditambahkan akan ditampilkan pada tabel batasan	Data batasan yang ditambahkan akan ditampilkan pada tabel batasan	Lulus
TC-16	Buat Jadwal (Berhasil)	Pengguna telah menambahkan data dan batasan untuk	Pengguna menekan tombol buat jadwal	Jadwal berhasil dibuat dan ditampilkan	Jadwal berhasil dibuat dan ditampilkan	Lulus

		membuat jadwal				
TC- 17	Buat Jadwal (Gagal)	Pengguna telah menambahka n data dan batasan untuk membuat jadwal	Pengguna menekan tombol buat jadwal	Muncul pesan gagal dan kendalanya	Muncul pesan gagal dan kendalanya	Lulus
TC- 18	Ubah Jadwal	Pengguna telah membuat jadwal sebelumnya	Pengguna menekan tombol ubah dan melakukan perubahan	Perubahan dan jadwal berhasil di update	Perubahan dan jadwal berhasil di update	Lulus
TC- 19	Hapus Jadwal	Pengguna telah membuat jadwal sebelumnya	Pengguna menekan tombol hapus dan mengonfirmas i penghapusan jadwal	Data berhasil dihapus	Data berhasil dihapus	Lulus
TC- 20	Export Jadwal	Pengguna telah membuat jadwal sebelumnya	Pengguna menekan tombol export	Jadwal berhasil di export dan di download	Jadwal berhasil di export dan di download	Lulus

BAB V KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian pada proyek akhir yang berjudul "Rancang Bangun Sistem Informasi Penjadwalan Pembelajaran Otomatis Berbasis Web Menggunakan *Constraint Programming*", dapat disimpulkan bahwa tujuan penelitian telah berhasil dicapai. Sistem informasi berbasis web yang dikembangkan mampu mengelola data penjadwalan, meliputi data kelas, tenaga pengajar, ruang kelas, mata pelajaran, hari, dan jam pembelajaran sebagai dasar dalam proses penyusunan jadwal.

Penerapan metode *Constraint Programming* yang dimodelkan sebagai *Constraint Satisfaction Problem* (CSP) berhasil menghasilkan jadwal pembelajaran secara otomatis dengan memenuhi berbagai batasan (*constraint*) yang telah ditentukan, seperti ketersediaan tenaga pengajar, penggunaan ruang kelas, alokasi waktu pembelajaran, serta aturan penjadwalan lainnya. Berdasarkan hasil pengujian fungsional, seluruh fitur sistem dapat berjalan sesuai dengan kebutuhan yang telah ditetapkan, sehingga sistem mampu membantu proses penyusunan jadwal pembelajaran menjadi lebih cepat, mengurangi potensi konflik jadwal, serta memberikan fleksibilitas kepada pengguna dalam mengelola data dan aturan penjadwalan.

5.2. Saran

Walaupun aplikasi berhasil diimplementasikan, namun masih terdapat beberapa fitur dan keterbatasan yang dapat disempurnakan pada pengembangan selanjutnya. Berikut adalah saran yang dapat dipertimbangkan untuk pengembangan selanjutnya:

1. Pada sisi *export* jadwal, dapat ditambahkan opsi lain seperti *export* ke bentuk *xlsx*, dengan tujuan pengguna dapat mengubah data selain melalui aplikasi web.
2. Penambahan opsi *import* untuk menambahkan data agar mempercepat proses penambahan data.

DAFTAR PUSTAKA

- Azmi, I. S., & Najiyah, I. (2023). Sistem Informasi Penjadwalan Mata Pelajaran Menggunakan Algoritma Artificial Bee Colony Berbasis Web. *Jurnal Ilmiah Teknik Informatika Dan Komunikasi*, 3(3), 35–48.
<https://doi.org/10.55606/juitik.v3i3.612>
- Bessiere, C., Raedt, L. D., Guns, T., Kotthoff, L., Nanni, M., Nijssen, S., O’Sullivan, B., Paparrizou, A., Pedreschi, D., & Simonis, H. (2023). *The Inductive Constraint Programming Loop*. 17.
<https://doi.org/https://doi.org/10.48550/arXiv.1510.03317>
- Dennis, A., Wixon, B. H., & Tegarden, D. (2021). *Systems Analysis and Design: An Object-Oriented Approach with UML* (6th ed.). Wiley.
- Diehl, C., Martins, A., Almeida, A., Silva, T., Ribeiro, Ó., Santinha, G., Rocha, N., & Silva, A. G. (2022). Defining Recommendations to Guide User Interface Design: Multimethod Approach. *JMIR Human Factors*, 9(3), e37894. <https://doi.org/10.2196/37894>
- Duckett, J. (2021). *HTML and CSS: Design and Build Websites*. John Wiley & Sons, Inc.
- Dumas, M., Rosa, M. L., Mendling, J., & Reijers, H. A. (2023). *Fundamentals of Business Process Management* (3rd ed.). <https://doi.org/10.1007/978-3-031-08848-8>
- Ecma International. (2026). *ECMAScript® Language Specification*. <https://ecma-international.org/publications-and-standards/standards/ecma-262/>
- Fadel, M., Pramono, B., & Statiswaty. (2023). Implementasi Metode Pewarnaan Graf Menggunakan Algoritma Greedy Untuk Menentukan Jadwal Mata Pelajaran Pada Smkn 4 Kendari. *Animator*, 1(2), 34–41.
- Google. (2026). *Constraint optimization | OR-Tools*. Google for Developers.
<https://developers.google.com/optimization/cp>
- Hilmi, M. A., Fakhriza, M., & Alda, M. (2024). Sistem Informasi Penjadwalan Pembelajaran Pada Sekolah Menggunakan Genetic Algorithm Berbasis Website. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 7171–7177.
<https://doi.org/10.36040/jati.v8i4.10147>

- Jeriko, T. K., Racma, D. F., Widjayanti, C. E., & Setyawan, A. A. (2022). Penerapan Algoritma Genetika Dalam Sistem Informasi Penjadwalan Mata Kuliah Berbasis Website Pada Stikom Yos Sudarso Purwokerto. *Jurnal Muara Sains, Teknologi, Kedokteran, Dan Ilmu Kesehatan*, 6(1), 101–118. <https://doi.org/10.24912/jmstkik.v6i1.17262>
- Lestari, A., Susilo, E., & Anhar. (2023). Sistem Penjadwalan Perkuliahan Menggunakan Algoritma Genetika Pada Jurusan Teknik Elektro Universitas Riau. *Jurnal Teknologi Informatika Dan Komputer*, 9(2), 1424–1434. <https://doi.org/10.37012/jtik.v9i2.1679>
- Lutz, M. (2025). *Learning Python* (6th ed.). O'Reilly Media, Inc.
- Pinedo, M. L. (2022). *Scheduling Theory, Algorithms, and Systems* (6th ed.). Springer Cham. <https://doi.org/10.1007/978-3-031-05921-6>
- Pulungan, S. M., Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (2022). Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database. *I(2)*, 143–147.
- Putro, A. N. S., Alamin, Z., Mutmainah, S., Mu'min, M. A., Suryadi, D., Muflih, G. Z., Trisanti, N., Safitri, Y., & Fachri, F. (2024). *Pemograman web* (A. F. Qohar & A. F. Rohman (eds.)). PT. Penerbit Qriset Indonesia.
- Salado, A., & Kulkarni, A. U. (2021). *An Assessment of the Adequacy of Common Definitions of the Concept of System*. 12. <https://doi.org/10.1002/j.2334-5837.2021.00851.x>
- Saravanos, A., & Curinga, M. X. (2023). *Simulating the Software Development Lifecycle : The Waterfall Model*. 6(108), 19. <https://doi.org/10.3390/asi6060108>
- Sitohang, H., Rosmiati, & Sinaga, E. E. S. (2021). *Aplikasi E-Learning Berbasis Web untuk Pembelajaran Jarak Jauh*. 4(01), 106–115.
- Zahidah, R. G., & Putri, R. A. (2025). *Implementasi Algoritma Constraint Satisfaction Problems dan Backtracking Pada Penjadwalan Kegiatan Belajar Mengajar*. 9, 2025. <https://doi.org/10.47002/metik.v9i1.1042>
- Zapata, J. G. (2024). *MySQL VS PostgreSQL : A Comparative Analysis of Relational Database Management Systems (RDBMS) Technologies Response Time in Web-based E-commerce*. 5.

DAFTAR LAMPIRAN

Lampiran A. Link Produk

https://github.com/Elvan2004/Penjadwalan_Pembelajaran_Otomatis