

PERANCANGAN SISTEM OPTIMALISASI PENJADWALAN MATA PELAJARAN MENGGUNAKAN ALGORITMA DYNAMIC PROGRAMMING

Shifa Fauziah¹, Khanif Faozi²

Universitas Pamulang; Jl. Raya Puspitek, Buaran, Kec. Pamulang, Kota Tangerang Selatan,
Banten 15310, (021) 7412566

Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang
e-mail: 1fauziahshifa862@gmail.com, 2dosen0241@unpam.ac.id

ABSTRAK : *Penjadwalan mata pelajaran di sekolah sering menghadapi kendala seperti bentrokan jadwal guru, alokasi ruang yang tidak seimbang, serta waktu penyusunan yang lama akibat proses manual. Penelitian ini merancang sistem penjadwalan berbasis web dengan algoritma Dynamic Programming untuk mengoptimalkan penyusunan jadwal di Madrasah Ibtidaiyah Ar-Robiatul Adawiyah. Algoritma ini memecah masalah besar menjadi submasalah yang saling terkait melalui prinsip optimal substructure dan overlapping subproblems. Metode penelitian yang digunakan adalah deskriptif kuantitatif dengan teknik observasi, wawancara, dan dokumentasi. Hasil pengembangan menunjukkan bahwa sistem mampu mempercepat proses penjadwalan, mengurangi bentrokan jadwal, serta meningkatkan akurasi dan efisiensi dibandingkan metode manual. Sistem ini membantu sekolah dalam meningkatkan efektivitas administrasi dan kualitas pengelolaan jadwal belajar.*

Kata kunci: *Penjadwalan, Dynamic Programming, Sistem Informasi, Web, Optimalisasi.*

ABSTRACT: *The process of scheduling subjects in schools often faces challenges such as time conflicts, mismatched schedules between teachers and students, and uneven teaching load distribution. These problems usually occur because the scheduling process is still carried out manually, which is time-consuming and prone to human error. This study aims to design a web-based subject scheduling system that optimizes the scheduling process using the Dynamic Programming algorithm. Dynamic Programming is applied due to its ability to decompose complex problems into smaller interrelated subproblems. Through the principles of optimal substructure and overlapping subproblems, the system can generate efficient scheduling solutions, avoid conflicts between teachers and classrooms, and maximize the utilization of school resources. This research employs a descriptive quantitative approach with data collection methods including observation, interviews, and documentation at Madrasah Ibtidaiyah Ar-Robiatul Adawiyah. The implementation results show that the designed system can accelerate the scheduling process and reduce errors commonly found in conventional methods. By applying the Dynamic Programming algorithm, scheduling becomes more flexible, accurate, and easier to update according to needs. This system is expected to assist schools in improving administrative efficiency, ensuring smooth teaching and learning activities, and enhancing the overall quality of schedule management.*

Keywords: *Information System, Subject Scheduling, Dynamic Programming, Web-Based System, Optimization.*

PENDAHULUAN

Penjadwalan mata pelajaran adalah kegiatan untuk mengatur proses belajar- mengajar pada siswa dan guru. Dengan adanya penjadwalan mata pelajaran, maka mata pelajaran dapat terorganisir berdasarkan waktu, ruang kelas, maupun pengajar atau guru. Aplikasi penjadwalan ini bertujuan untuk mempermudah, mengatur dan meningkatkan kedisiplinan dalam proses kegiatan belajar-mengajar (Zen, Untomo, & Hamdi, 2023). Di Madrasah Ibtidaiyah Ar-Robiatul Adawiyah yang terletak di Jl. Sunan Giri Gg. Masjid Biturrahman Rt. 01/02, Pondok Bahar, Kec. Karang Tengah, Kota Tangerang, Penjadwalan mata pelajaran di sekolah memainkan peran yang sangat penting dalam memastikan kelancaran dan keteraturan proses belajar mengajar. Penjadwalan ini bertujuan tidak hanya untuk meningkatkan efisiensi operasional, tetapi juga untuk mendukung peningkatan kualitas pendidikan. Dengan jadwal yang tersusun rapi, proses belajar mengajar dapat berjalan tanpa hambatan, mendorong disiplin siswa, dan memastikan penggunaan waktu yang efektif sehingga setiap siswa dapat mengikuti kegiatan pembelajaran dengan optimal (Riyanti, Heriansyah, & Ramadhan, 2024).

Permasalahan dalam penyusunan jadwal pembelajaran sering kali menjadi tantangan di dunia pendidikan, termasuk di Madrasah Ibtidaiyah Ar-Robiatul Adawiyah. Berdasarkan hasil evaluasi dan wawancara dengan kepala sekolah, diperlukan sebuah aplikasi yang dapat mempermudah tugas staf tata usaha dengan menggunakan metode yang tepat agar penjadwalan dapat berjalan lebih optimal. Salah satu algoritma yang dapat digunakan untuk menyelesaikan masalah ini adalah Dynamic Programming. Keterbatasan sistem yang ada juga menyebabkan kesulitan dalam memenuhi kebutuhan siswa secara tepat waktu dan menyelaraskan jadwal dengan ketersediaan guru. Waktu yang diperlukan untuk menyusun jadwal di sekolah ini dapat bervariasi tergantung pada faktor-faktor seperti kompleksitas kurikulum, jumlah kelas dan guru, serta pengalaman staf. Proses penjadwalan biasanya memakan waktu sekitar satu hingga dua minggu, terutama pada awal tahun. Untuk mengatasi masalah ini, diperlukan sebuah website penjadwalan matapelajaran yang mampu memperhitungkan semua variabel kompleks ini. Website tersebut dapat mengoptimalkan penyusunan jadwal dengan mempertimbangkan ketersediaan guru, alokasi waktu, serta meminimalisir jadwal. Dengan penerapan algoritma Dynamic Programming, proses penyusunan jadwal dapat diselesaikan dalam waktu yang lebih singkat dan efisien, sekitar satu hingga dua minggu, dibandingkan dengan metode konvensional yang lebih memakan waktu. Di dalam perancangan sistem ini penulis menggunakan Dynamic Programming merupakan metode yang sangat fleksibel dan handal dalam menyelesaikan berbagai permasalahan pemrograman matematis. Dibandingkan algoritma optimasi lainnya, pendekatan ini menawarkan keluasan tinggi dalam penerapannya. Dalam konteks sistem ini, algoritma Dynamic Programming diaktifkan ketika timer untuk menjalankan polisi mulai berjalan. Sistem akan mencatat posisi pengguna, lalu menghitung jalur tercepat menuju lokasi pengguna dengan memanfaatkan prinsip- prinsip Dynamic Programming. (Fawwaz, Winarta, Selvianna, Ramli, & Waruwu, 2019).

Dynamic programming merupakan algoritma pendekatan untuk memecahkan suatu permasalahan menjadi masalah-masalah yang lebih sederhana dan menjadikannya sebagai pondasi untuk membangun solusi. Dalam permasalahan untuk menentukan prioritas tugas mata kuliah mahasiswa, algoritma dynamic programming dapat diterapkan untuk menentukan nilai optimal dari seluruh parameter/variabel yang ditentukan sebagai bentuk perlakuan untuk menciptakan urutan yang optimal dalam penyelesaian tugas dengan mempertimbangkan semua bobot yang relevan. (Hidayat, Ferdinan, Saputra, Christian, & Pranatawijaya, 2024). Menyadari pentingnya permasalahan ini, penulis berkomitmen untuk mengangkatnya dalam tugas akhir dengan judul PERANCANGAN SISTEM OPTIMALISASI PENJADWALAN MATA PELAJARAN MENGGUNAKAN ALGORITMA DYNAMIC PROGRAMMING. Diharapkan bahwa sistem ini akan memberikan solusi yang efektif dalam meningkatkan efisiensi penjadwalan, serta berkontribusi pada peningkatan kualitas pendidikan di madrasah.

METODA

Metode pelaksanaan penelitian ini terdiri atas beberapa tahapan yang dilakukan secara sistematis untuk menghasilkan sistem penjadwalan mata pelajaran yang optimal. Adapun tahapan pelaksanaan penelitian adalah sebagai berikut:

1. Tahap Persiapan
 - Mengidentifikasi permasalahan penjadwalan mata pelajaran yang terjadi di sekolah.
 - Menentukan tujuan penelitian dan ruang lingkup pengembangan sistem.
 - Mengumpulkan referensi terkait algoritma Dynamic Programming dan sistem penjadwalan.
2. Tahap Pengumpulan Data
Data diperoleh melalui:
 - Observasi
Mengamati langsung proses penjadwalan manual yang dilakukan oleh pihak sekolah.
 - Wawancara
Melakukan wawancara dengan kepala sekolah dan guru kurikulum untuk mengetahui kebutuhan, kendala, serta aturan penjadwalan.
 - Dokumentasi
Mengumpulkan data guru, mata pelajaran, jam pelajaran, kelas, dan jadwal tahun-tahun sebelumnya sebagai bahan analisis.
3. Tahap Analisis
 - Menganalisis alur penjadwalan berjalan dan permasalahan yang muncul, seperti bentrok jadwal atau alokasi waktu guru.
 - Mengidentifikasi kebutuhan sistem (input, proses, output).
 - Menentukan parameter algoritma Dynamic Programming yang digunakan.
4. Tahap Perancangan Sistem
 - Mendesain arsitektur sistem berbasis web.
 - Membuat rancangan database (ERD dan LRS).
 - Mendesain diagram UML seperti use case, activity diagram, sequence diagram, dan class diagram.
 - Menyusun rancangan tampilan antarmuka (UI).
5. Tahap Implementasi
 - Mengembangkan sistem menggunakan PHP, CodeIgniter, MySQL, HTML, CSS, jQuery, dan AJAX.
 - Mengintegrasikan algoritma Dynamic Programming untuk mengolah jadwal secara optimal.
 - Mengimplementasikan fitur admin dan guru sesuai kebutuhan operasional sekolah.
6. Tahap Pengujian
 - Melakukan pengujian Black Box untuk memastikan semua fungsi sistem berjalan sesuai kebutuhan pengguna.
 - Melakukan pengujian White Box untuk memastikan alur logika berjalan benar dan bebas error.
 - Melakukan perbaikan jika ditemukan kesalahan dalam proses pengujian.
7. Tahap Evaluasi dan Penyusunan Laporan
 - Mengevaluasi hasil implementasi sistem berdasarkan kriteria efektivitas, efisiensi, dan akurasi.
 - Menyusun laporan penelitian dalam bentuk skripsi atau jurnal ilmiah.

Algoritma Dynamic Programming

Algoritma *Dynamic Programming* adalah sebuah metode yang termasuk dalam teori optimasi, dapat dikatakan pula sebagai metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sedemikian rupa sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Dalam konteks penjadwalan mata pelajaran siswa, metode ini dapat digunakan untuk mengoptimalkan jadwal pelajaran dengan mempertimbangkan beberapa faktor seperti ketersediaan guru untuk mengajar, beban ajar/jam mengajar guru, waktu yang

tersedia dan persyaratan khusus untuk mata pelajaran tertentu. Penerapan metode *dynamic programming* dalam penjadwalan mata pelajaran memungkinkan pencarian solusi jadwal yang optimal berdasarkan fungsi tujuan yang ditetapkan. Dengan memperhitungkan semua variabel dan parameter yang relevan, serta memanfaatkan kemampuan pemrograman dinamis, metode ini dapat membantu meningkatkan efisiensi dan kualitas penjadwalan dalam konteks pendidikan. (Riyanti, Heriansyah, & Ramadhan, 2024).

Algoritma tersebut sudah terlebih dahulu membuat sesuatu populasi awal merupakan langkah pertama dalam menyusun solusi dari suatu masalah, seperti penjadwalan mata pelajaran menggunakan *Dynamic Programming*. Populasi awal terdiri dari sekumpulan solusi individu yang merupakan jadwal-jadwal yang dapat dioptimalkan lebih lanjut. Setiap individu dalam populasi ini dihasilkan berdasarkan kombinasi faktor-faktor penting seperti mata pelajaran, ketersediaan guru, waktu pelajaran, ruangan, dan beban kerja.

Langkah-langkah Perhitungan algoritma *dynamic programming*:

Dalam konteks penjadwalan mata pelajaran, Dynamic Programming digunakan untuk mencari kombinasi jadwal terbaik (paling optimal) dengan meminimalkan biaya konflik seperti:

- Guru bentrok (mengajar lebih dari satu kelas pada saat yang sama)
- Ruang tidak tersedia
- Pelajaran dobel atau tidak sesuai

Notasi	Keterangan
N	Jumlah total slot waktu (misal: 5 hari × 6 jam = 30 slot)
n	Indeks slot waktu ke-n ($1 \leq n \leq N$)
S_n	Keadaan pada slot ke-n (misalnya: pelajaran yang bisa ditempatkan di slot ini)
S_{nk}	Pilihan ke-k pada slot n (misalnya guru/pelajaran/ruang yang tersedia)
X_n	Keputusan mata pelajaran/guru yang dijadwalkan pada slot ke-n
X_n^*	Nilai keputusan optimal pada slot ke-n
$C(S_n, X_n)$	Biaya konflik atau penalti penjadwalan (misal: guru bentrok, ruangan tidak tersedia)
$F_n^*(S_n)$	Biaya minimum total dari slot n hingga akhir
$f_n^*(S_n) = \min \{ C(S_n, X_n) + f_{(n+1)}^*(X_n) \}$	Rumus umum DP

Rumus Penjadwalan Mata Pelajaran dengan Dynamic Programming

- Definisi Variabel
 $C(S_n, X_n) \rightarrow$ Cost function (biaya konflik) jika mata pelajaran atau guru X_n dijadwalkan pada slot waktu S_n . Biaya ini dihitung berdasarkan: Ketersediaan guru \rightarrow Jika guru tersedia pada slot tersebut, biaya = 0; jika tidak, biaya = nilai penalti besar.
Ketersediaan ruangan \rightarrow Jika ruang tersedia, biaya = 0; jika tidak, biaya = penalti. Tidak terjadi bentrok guru \rightarrow Jika guru tidak mengajar di dua kelas berbeda pada waktu yang sama, biaya = 0; jika bentrok, biaya = penalti. Aturan tambahan (opsional) \rightarrow Misalnya pelajaran tertentu hanya boleh dijadwalkan di pagi hari. $f_n^*(S_n) \rightarrow$ Biaya minimum total mulai dari slot ke-n hingga slot terakhir (optimal cost from step n to the end). $n \rightarrow$ Nomor urutan slot atau langkah penjadwalan (misalnya jam ke-1, jam ke-2, dst.). $X_n \rightarrow$ Pilihan guru/pelajaran yang dijadwalkan di slot S_n .

- Rumus Umum Dynamic Programming
Penjelasan:
Untuk setiap slot waktu S_n , kita menghitung semua kemungkinan penempatan guru/pelajaran X_n .
 $C(S_n, X_n) \rightarrow$ Menghitung biaya konflik untuk kombinasi guru/pelajaran tersebut di slot waktu tersebut.
 $f^*_{n+1}(S_{n+1}) \rightarrow$ Mengambil biaya minimum dari slot berikutnya hingga akhir, berdasarkan keputusan sekarang. Fungsi min \rightarrow Memilih kombinasi guru/pelajaran yang memberikan total biaya paling rendah.
- Kondisi Dasar (Base Case)
Jika sudah sampai slot terakhir ($n = N$, di mana N = jumlah total slot), maka: $f^*_N(S_N) = \min_{X_N \in \text{Pilihan Artinya, di slot terakhir kita hanya menghitung biaya konflik untuk semua pilihan dan memilih yang terkecil.}}$
- Proses Penyelesaian
 - a) Mulai dari slot terakhir \rightarrow Hitung $f^*_N(S_N)$ untuk semua pilihan.
 - b) Lanjut ke slot sebelumnya ($n = N-1$) \rightarrow Gunakan hasil dari slot berikutnya untuk menghitung $f^*_n(S_n)$.
 - c) Ulangi proses ini mundur hingga sampai di slot pertama ($n = 1$)
 - d) Ambil urutan jadwal yang memberikan total biaya minimum.
 - e) Proses Penyelesaian (Algoritma Mundur / Backward Iteration)
- Langkah-langkah:
 - a) Inisialisasi Slot Terakhir
Hitung $f^*_N(S_N)$ untuk Semua pilihan X_N . Simpan nilai biaya minimum.
 - b) Iterasi ke Slot Sebelumnya

Untuk slot ke- $(N-1)$, hitung:

Lanjutkan Mundur hingga Slot Pertama

Ulangi proses ini hingga slot ke-1 selesai dihitung.

Ekonstruksi Jadwal Optimal

Setelah semua slot dihitung, pilih kombinasi guru/pelajaran yang menghasilkan biaya minimumkeseluruhan.

- Kelebihan Pendekatan Dynamic Programming
- Menjamin solusi optimal.
- Bisa menampung banyak aturan dan constraint.
- Dapat digunakan untuk penjadwalan dengan banyak variabel seperti guru, ruang, dan kelas sekaligus.

HASIL DAN PEMBAHASAN

1. Gambaran Umum Sistem

Sistem penjadwalan mata pelajaran yang dikembangkan adalah aplikasi berbasis web yang digunakan untuk membantu sekolah dalam mengelola jadwal guru, mata pelajaran, ruang kelas, serta alokasi waktu secara otomatis. Sistem ini dirancang untuk mengatasi permasalahan bentrok jadwal, mengurangi proses manual yang memakan waktu, serta meningkatkan efisiensi penyusunan jadwal di Madrasah Ibtidaiyah Ar-Robiatul Adawiyah.

2. Analisis Permasalahan yang Dihadapi Sekolah

Berdasarkan hasil observasi dan wawancara, ditemukan beberapa permasalahan sebagai berikut:

- a. Penjadwalan masih dilakukan secara manual sehingga membutuhkan waktu lama.
 - b. Sering terjadi bentrokan jadwal antara guru yang mengajar di beberapa kelas. Pemanfaatan ruang kelas belum optimal, beberapa ruang overload dan sebagian tidak terpakai.
 - c. Perubahan jadwal sulit dilakukan, karena harus menyesuaikan seluruh jadwal lain. Kurangnya alat bantu digital untuk mengelola data jadwal secara terstruktur. Permasalahan tersebut menjadi dasar kebutuhan sistem yang lebih efektif dan otomatis.
3. Penerapan Algoritma Dynamic Programming
- Dynamic Programming (DP) digunakan sebagai metode optimasi untuk menghasilkan jadwal terbaik. Penerapan DP dilakukan melalui langkah-langkah berikut:
- a. State dan Subproblem

Setiap kombinasi jadwal (guru–hari–jam–kelas–ruang) diperlakukan sebagai state.

Subproblem berupa penentuan alokasi waktu setiap guru berdasarkan:

Ketersediaan guru

Beban jam mengajar

Jumlah mata pelajaran

Kapasitas kelas

Aturan sekolah

b. Transition Function

Algoritma mencari solusi terbaik dengan memilih jadwal yang tidak menyebabkan konflik, kemudian menambahkannya pada jadwal optimal sebelumnya.

c. Memoization

Setiap hasil subproblem disimpan agar tidak dihitung ulang, sehingga proses menjadi lebih cepat dan efisien.

d. Optimal Substructure

Solusi akhir terbentuk dari solusi-solusi kecil yang optimal, menghasilkan jadwal lengkap yang bebas bentrok.

4. Implementasi Sistem

Sistem diimplementasikan menggunakan PHP, CodeIgniter, MySQL, HTML, CSS, JavaScript, dan jQuery. Fitur utama meliputi:

- a. Modul Admin
 - Manajemen data guru
 - Manajemen kelas
 - Data mata pelajaran
 - Pengaturan jam belajar
 - Pengelolaan tahun ajaran
 - Proses pembuatan jadwal otomatis
 - Riwayat jadwal
- b. Modul Guru
 - Melihat jadwal mengajar
 - Mengakses informasi kelas dan jam pelajaran
 - Menerima perubahan jadwal
- c. Dashboard Monitoring

Menampilkan rekap:
Jumlah guru
Jumlah kelas
Jadwal yang telah terbentuk
Statistik bentrokan yang berhasil diatasi

5. Pengujian Sistem

Pengujian dilakukan dengan dua pendekatan:

1. Pengujian Black Box

Fokus pada fungsi-fungsi sistem. Hasilnya:
Semua fitur berhasil dijalankan sesuai kebutuhan.
Input–output sistem bekerja dengan benar.
Proses generate jadwal otomatis berjalan tanpa error.

2. Pengujian White Box

Pengujian pada logika program dan alur algoritma:
Tidak ditemukan alur logika yang menyebabkan crash.
Proses DP berjalan sesuai skenario perhitungan.

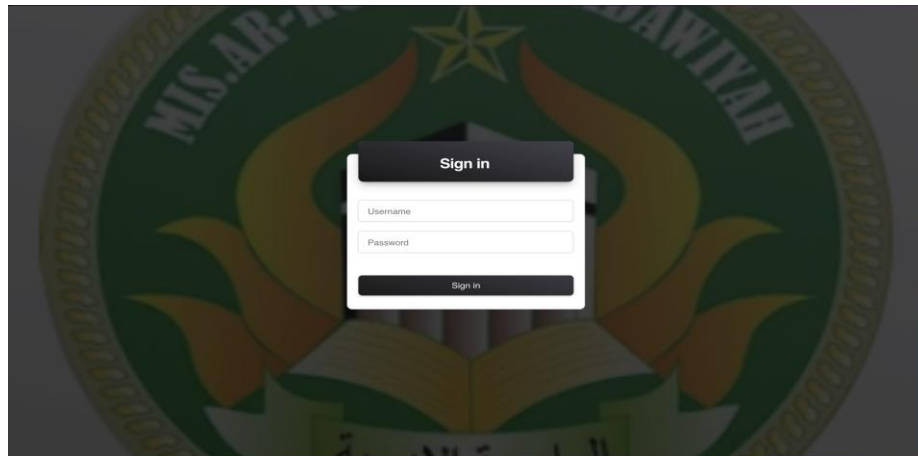
6. Hasil Implementasi

Setelah sistem diterapkan, diperoleh hasil sebagai berikut:
Waktu penyusunan jadwal lebih cepat, dari 1–2 minggu menjadi hanya beberapa menit.
Bentrokan jadwal berkurang signifikan karena algoritma menghindari konflik otomatis.
Pemanfaatan ruang kelas lebih optimal, tidak ada jam yang kosong berlebih.
Perubahan jadwal lebih fleksibel, hanya mengatur bagian tertentu tanpa mengulang seluruh jadwal.
Guru dapat melihat jadwal secara real-time, meningkatkan efektivitas komunikasi.

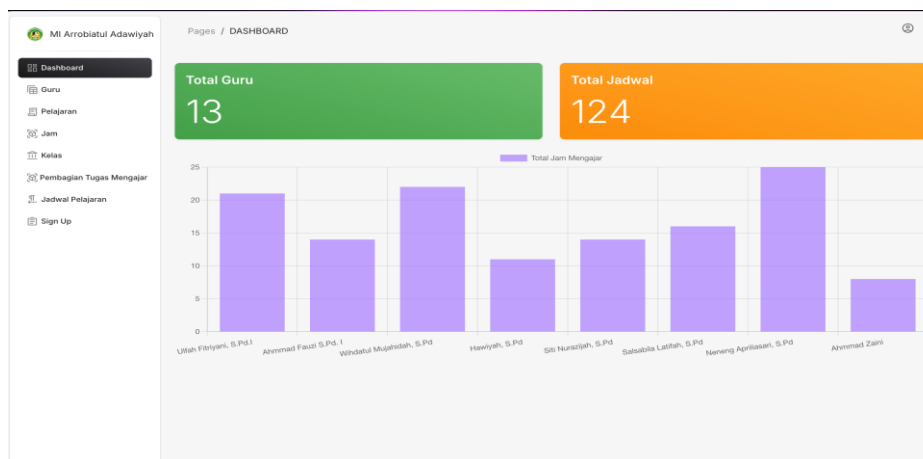
7. Pembahasan

Dari hasil pengembangan, dapat disimpulkan bahwa penggunaan Dynamic Programming memberikan keuntungan berikut:
Mengoptimalkan penjadwalan dengan hasil yang lebih akurat.
Mengurangi redundansi perhitungan, sehingga proses menjadi cepat.
Menghasilkan solusi terbaik dari berbagai kemungkinan jadwal.
Memberikan fleksibilitas untuk penyesuaian kembali tanpa merusak struktur jadwal.

A. Hasil Implementasi Sistem



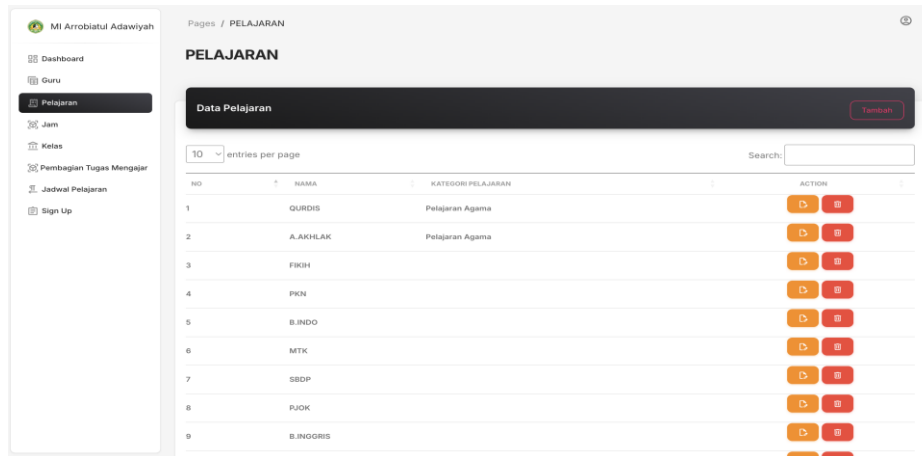
Gambar 2. Implementasi Login Admin



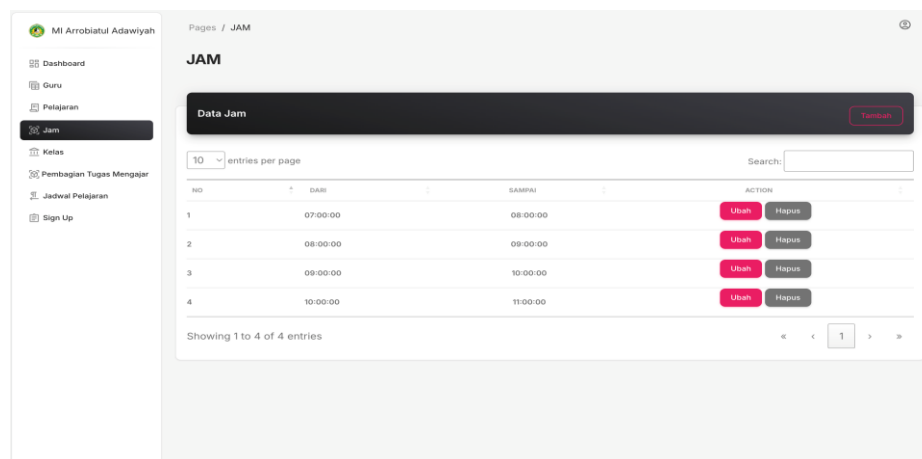
Gambar 3. Implementasi Dashboard Admin

NIP	NAMA	WAKU KELAS	TELEPON	TTL	ACTION
	Linda Iestari S.Pd	IV B	08345678543	tangerang, 2002-05-07	[Edit] [View] [Delete] [Add]
	Lisa Aulia S.Pd	III A		tangerang, 2003-05-07	[Edit] [View] [Delete] [Add]
	Abdul Fikri S.Pd	V		tangerang, 1998-01-20	[Edit] [View] [Delete] [Add]
06444	Ahmad Fauzi S.Pd.I	VI	0892929	tangerang, 1888-11-10	[Edit] [View] [Delete] [Add]
34543	Ulfa Fitriyani, S.Pd.I	I A	333333	Tangerang, 1989-06-05	[Edit] [View] [Delete] [Add]
45667	Hawiyah, S.Pd	II A	0854443	tangerang, 2025-08-12	[Edit] [View] [Delete] [Add]
4444444	Wihdatul Mujahidah, S.Pd	I B	504444	tangerang, 2000-06-19	[Edit] [View] [Delete] [Add]
7878923	sifa		32323	tangerang, 2025-06-02	[Edit] [View] [Delete] [Add]
08329283	Salsabila Latifah, S.Pd	III B	08345678543	tangerang, 2001-02-02	[Edit] [View] [Delete] [Add]
08433333	Ahmad Zaini		086363636	tangerang, 2006-11-02	[Edit] [View] [Delete] [Add]

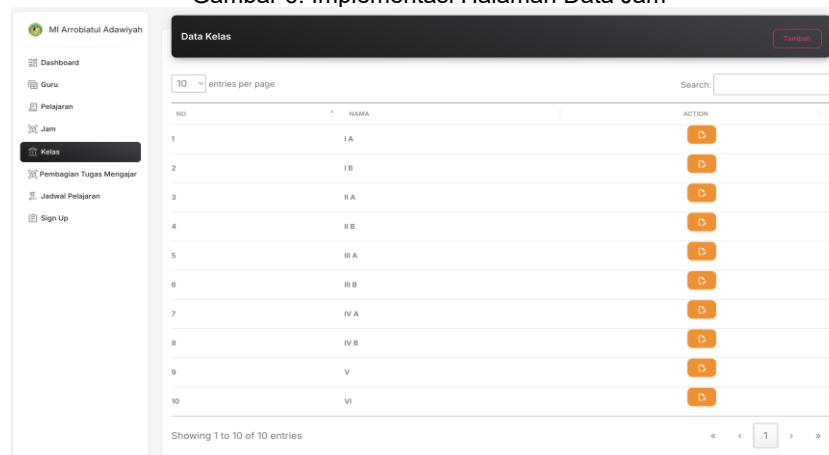
Gambar 4. Implementasi halaman Data Guru



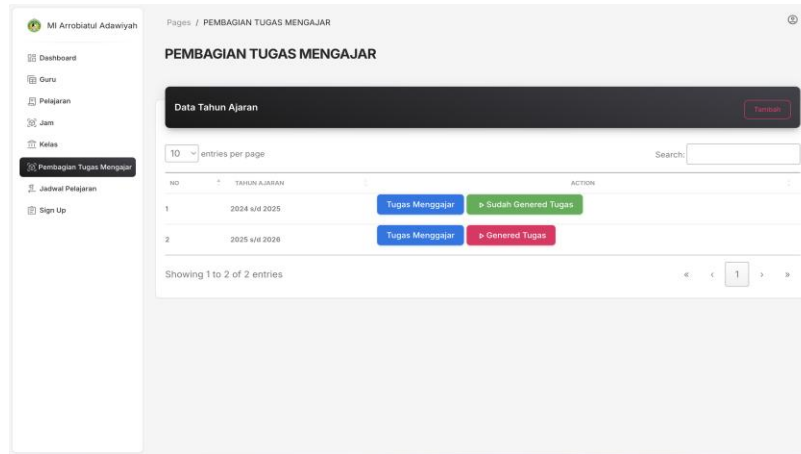
Gambar 5. Implementasi Halaman Data Pelajaran



Gambar 6. Implementasi Halaman Data Jam



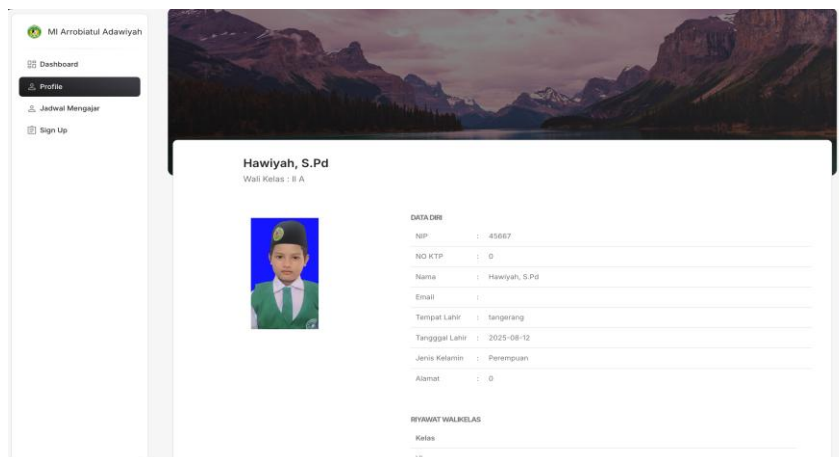
Gambar 7. Implementasi Halaman Data Kelas



Gambar 8. Implementasi Halaman Data Tahun Ajar



Gambar 9. Implementasi Halaman Dashbord Guru



Gambar 10. Implementasi Halaman Profil Guru

KELAS	HARI	WAKTU	MATA PELAJARAN
II A	Senin	08:00:00 - 09:00:00	BINGGRIS
II A	Selasa	07:00:00 - 10:00:00	MTK
II A	Rabu	07:00:00 - 10:00:00	PKN
II A	Kamis	08:00:00 - 09:00:00	SBOP
II A	Jumat	08:00:00 - 09:00:00	AKHLAK
II B	Jumat	07:00:00 - 08:00:00	QURDIS
III B	Kamis	08:00:00 - 09:00:00	FIKH

Gambar 11. Implementasi Halaman Jadwal Mengajar

B. Pengujian Sistem

- Pengujian Sistem *Black Box*

Tabel 1. Pengujian Black Box Login

Me nu	Test case ID	Test Case	Test Step	Expected Result	Actual Result	Res ult
Lo gin	TC_LOG IN_001	Login dengan usernam e dan passwor d benar (Admin/ Guru)	1.Buka halaman login. 2. Masukkan username dan password yang benar. 3. Klik tombol Login.	Pengguna berhasil masuk ke dashboard sesuai peran (Admin/Guru)	Sistem menampilkan halaman dashboard sesuai peran.	Don e
Lo gin	TC_LOG IN_002	Usernam e benar, passwor d salah	1. Masukkan username benar dan password salah. 2. Klik Login.	Sistem menampilkan pesan "Gagal masuk ke dalam sistem! Harap tunggu sebentar!"	Pesan kesalahan muncul sesuai harapan.	Don e
Lo gin	TC_LOG IN_003	Usernam e salah, passwor d benar	1. Masukkan username salah dan password benar. 2. Klik Login.	Sistem menampilkan pesan "Username atau password salah".	Pesan kesalahan tampil dengan benar.	Don e
	TC_LOG	Kolom	1. Biarkan	Sistem	Pesan	

Lo gin	IN_004	usern ame dan passwor d kosong	kolom kosong. 2. Klik Login.	menampilkan pesan validasi "Kolom tidak boleh kosong".	validasi muncul.	Don e
-----------	--------	---	---------------------------------------	---	---------------------	----------

Tabel 2. Pengujian Black Box Dashboard Admin

Menu	Test Case ID	Test Case	Test Step	Expected Result	Actual Result	Re sult
Dash board Admin	TC_D ASH _001	Akses dashboar d setelah login	1. Login sebagai Admin. 2. Arahkan Ke dashboard utama.	Dashboard menampilkan ringkasan data guru dan jadwal.	Dashboard tampil sesuai desain.	Do ne
Dash board Admin	TC_D ASH _002	Menampi lkan total guru	1. Buka Dashboard. 2. Lihat bagian total guru.	Jumlah guru tampil sesuai data.	Jumlah tampil dengan benar.	Do ne
Dash board Admin	TC_D ASH _003	Klik menu sidebar (Guru, Jadwal, Logout)	1. Klik menu satu per satu.	Halaman sesuai menu terbuka.	Semua menu berfungsi.	Do ne

Tabel 3. Pengujian Black Box Halaman Data Guru

Menu	Test Case ID	Test Case	Test Step	Expected Result	Actual Result	Res ult
Data	TC_G	Tambah	1. Klik	Data guru	Data	

Guru	URU_001	data guru baru	Tambah Data Guru. 2. Isi semua kolom. 3. Klik Simpan.	tersimpan dan tampil di tabel.	tampil dengan benar di tabel.	Done
Data Guru	TC_GURU_002	Edit data guru	1. Klik Edit pada salah satu guru. 2. Ubah data. 3. Klik Simpan.	Perubahan tersimpan dan tampil.	Data berubah sesuai input.	Done
Data Guru	TC_GURU_003	Hapus data guru	1. Klik Hapus pada guru tertentu. 2. Konfirmasi.	Data guru terhapus dari sistem.	Data terhapus.	Done

Tabel 4. Pengujian Black Box Halaman Jadwal Mengajar

Menu	Test Case ID	Test Case	Test Step	Expected Result	Actual Result	Result
Jadwal Mengajar	TC_JADWAL_001	Menampilkan jadwal semua kelas	1. Login sebagai Admin. 2. Buka menu Jadwal Mengajar	Tabel jadwal menampilkan seluruh kelas, hari, jam, dan mapel.	Data jadwal tampil lengkap.	Done
Jadwal Mengajar	TC_JADWAL_002	Edit jadwal pelajaran	1. Klik Edit. 2. Ubah hari/jam/mapel. 3. Klik Simpan.	Jadwal berubah sesuai input.	Perubahan berhasil disimpan.	Done
Jadwal Mengajar	TC_JADWAL_003	Cetak jadwal	1. Klik tombol Print. 2. Sistem membuka tampilan cetak.	Jadwal muncul dalam format cetak.	Jadwal berhasil dicetak.	Done

- Pengujian Sistem White Box

Tabel 5. Tabel White Box Login

No	Pengujian	Text Case Benar	Text Case Salah
1	Sistem memeriksa kecocokan username dan password pada tabel login.	Data valid → sistem menampilkan halaman dashboard admin.	Data salah → sistem menampilkan pesan "Username atau password salah".
2	Pembuatan sesi login untuk pengguna yang berhasil masuk.	Session terbentuk dan admin diarahkan ke dashboard.	Session tidak terbentuk dan admin tetap di halaman login.
3	Pengecekan peran pengguna untuk menentukan hak akses (admin atau guru).	Admin diarahkan ke halaman admin, guru ke halaman guru.	Jika role tidak dikenali → sistem menampilkan pesan error akses.

Tabel 5. Tabel White box Dashboard

No	Pengujian	Text Case Benar	Text Case Salah
1	Sistem memeriksa status <i>session</i> pengguna untuk memastikan yang mengakses	Session aktif → sistem menampilkan halaman dashboard admin.	Session tidak ditemukan → sistem mengalihkan ke halaman login dengan pesan "Silakan login terlebih

	adalah admin.		dahulu.”
2	Proses pengambilan data guru, siswa, jadwal, dan laporan dari database menggunakan perintah <i>SELECT</i> .	Data berhasil diambil dan ditampilkan di tabel dashboard.	Query gagal → sistem menampilkan pesan error atau tabel kosong.
3	Perhitungan jumlah data (guru, siswa, jadwal) melalui fungsi <i>COUNT()</i> untuk ringkasan informasi.	Hasil perhitungan sesuai dengan jumlah aktual di database.	Hasil perhitungan tidak sesuai karena kesalahan query.
4	Pemanggilan fungsi statistik (misalnya jumlah guru aktif, siswa aktif).	Nilai statistik muncul sesuai data di database.	Statistik tidak tampil akibat kesalahan pemanggilan fungsi.
5	Pengujian tombol navigasi (menu ke Data Guru, Data Jadwal, dan Laporan).	Klik menu menampilkan halaman yang sesuai.	Klik menu gagal mengarahkan ke halaman tujuan.

Tabel 6. Tabel White Box Halaman Data Guru

No	Pengujian	Text Case Benar	Text Case Salah
1	Sistem menampilkan seluruh data guru dari tabel guru menggunakan perintah <i>SELECT * FROM guru</i>	Data guru tampil lengkap sesuai isi database.	Data tidak muncul akibat kesalahan query atau koneksi database.
2	Proses penambahan data guru baru melalui form input dan validasi <i>`if(empty(\$nama)</i>		<i>empty(\$nip))`</i> .

3	Proses pengeditan data guru dengan query UPDATE guru SET ... WHERE id_guru=....	Data berhasil diperbarui sesuai perubahan terbaru.	Query gagal atau ID tidak ditemukan → sistem menampilkan pesan error pembaruan.
4	Penghapusan data guru dengan kondisi if(isset(\$_GET['hapus'])).	Data guru terhapus dari database dan tabel diperbarui.	Data tidak terhapus karena ID tidak ditemukan atau terjadi kesalahan koneksi.
5	Validasi duplikasi data berdasarkan NIP guru menggunakan kondisi if(\$cek > 0).	Sistem menolak data jika NIP sudah digunakan dan menampilkan peringatan.	Sistem menyimpan data ganda tanpa validasi (error logika).

Tabel 7. Tabel White Box Halaman Jadwal Mengajar

No	Pengujian	Text Case Benar	Text Case Salah
1	Sistem menampilkan daftar jadwal mengajar dari tabel jadwal menggunakan query SELECT * FROM jadwal.	Semua data jadwal tampil lengkap sesuai database.	Data tidak muncul akibat kesalahan query atau koneksi.
2	Proses penambahan jadwal baru dengan validasi `if(empty(\$guru))`		empty(\$kelas)
3	Logika pengecekan bentrok waktu mengajar dengan kondisi if(\$guru_sama && \$jam_sama).	Sistem menolak jadwal bentrok dan menampilkan pesan "Guru sudah memiliki jadwal di jam tersebut."	Sistem tetap menyimpan data meskipun jam bentrok (error logika).
4	Proses pengeditan jadwal dengan query UPDATE jadwal SET ...	Data jadwal diperbarui sesuai	Query gagal atau ID jadwal tidak

	WHERE id_jadwal=....	perubahan terbaru.	ditemukan → sistem menampilkan pesan gagal memperbarui.
5	Algoritma penjadwalan otomatis menggunakan metode Dynamic Programming untuk menentukan kombinasi jadwal terbaik.	Jadwal tersusun tanpa bentrok dan sesuai ketersediaan guru serta ruang kelas.	Sistem menghasilkan jadwal tumpang tindih atau tidak sesuai (error algoritma).
6	Penghapusan jadwal mengajar berdasarkan ID menggunakan kondisi <code>if(isset(\$_GET['hapus']))</code> .	Data jadwal berhasil dihapus dari tabel.	Data tidak terhapus karena ID tidak ditemukan atau query error.

SIMPULAN DAN SARAN

Berdasarkan hasil penelitian, perancangan, serta implementasi sistem penjadwalan mata pelajaran yang telah dilakukan, dapat disimpulkan bahwa sistem ini mampu membantu pihak sekolah dalam mengelola jadwal mengajar guru dengan lebih efektif dan optimal. Proses yang sebelumnya dilakukan secara manual dan membutuhkan banyak waktu kini dapat dikerjakan secara otomatis melalui sistem berbasis web. Sistem ini juga mampu meminimalisir terjadinya bentrok jadwal antar guru maupun antar mata pelajaran, sehingga kegiatan belajar mengajar dapat berlangsung dengan lebih teratur dan terkoordinasi.

DAFTAR PUSTAKA

- Ginting, F. A. (2019). Dynamic Programming dalam Penyelesaian Masalah Penjadwalan.
- Gunawan, Y. A. (2023). Penerapan Algoritma Dynamic Programming Dalam Pencarian Judul Skripsi. Teknologi Sistem Informasi dan Sistem Komputer TGD.
- Sri Novida Sari, R. K. (2022). Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Metode Hybrid Artificial Bee Colony (HABC). Nasional Teknologi Komputer.
- Khairunnisa Samosir, D. N. (2020). Perancangan Sistem Informasi Pengolahan Jadwal Mata Pelajaran Siswa Secara Online Di SMPN 31 Padang Berbasis Web. Teknik Informatika dan Sistem Informasi.
- Hidayat, F. N., Ferdinan, B. D., Saputra, R., Christian, E., & Pranatawijaya, V. H. (2024). Penerapan Algoritma Dynamic Programming Dalam Penentuan Prioritas Pengerjaan Tugas Kuliah Mahasiswa. Informatika : Fakultas Sains dan Teknologi.

- Zen, M., Untomo, R. B., & Hamdi, N. (2023). Sistem Informasi Penjadwalan Mata Pelajaran Berbasis Web Menggunakan Metode Prototyping Pada SMKN 9 Medan. RESOLUSI : Rekayasa Teknik Informatika dan Informasi, 106-113.
- Riyanti, R., Heriansyah, R., & Ramadhan, M. (2024). Penjadwalan Mata Pelajaran Menggunakan Metode Dynamic Programming (Studi Kasus SD Negeri 1 Babat Toman). Jurnal Pengembangan Sistem Informasi dan Informatika.
- Fawwaz, I., Winarta, A., Selvianna, Ramli, J. J., & Waruwu, L. M. (2019). PENERAPAN ALGORITMA DYNAMIC PROGRAMMING PADA PERGERAKAN LAWAN DALAM PERMAINAN POLICE & THIEF. Journal of Informatics and Telecommunication Engineering.
- Amaliah, M. D., Lubis, I., & Lubis, H. (2022). SISTEM PENJADWALAN MATAPELAJARAN PADA PONDOK PESANTREN DAARUL ISTIQLAL MEDAN. Krisnadana. 153
- L, F. A., & Ginting, R. (2019). Dynamic Programming dalam Penyelesaian Masalah Penjadwalan. TALENTA Conference Series: Energy & Engineering (EE).
- Wahyono, A. t., Novianto, D. J., Nugroho, T. A., & Agusti, F. (2024). OPTIMALISASI RUTE PERJALANAN MENUJU KAMPUS UNIVERSITAS DUTA BANGSA SURAKARTA DENGAN METODE DYNAMIC PROGRAMMING. JournalScience Innovation and Technology (SINTECH).
- Wang, M., Liu, G., & Lin, X. (2022). Dynamic Optimization of the Multi-Skilled Resource-Constrained Project Scheduling Problem with Uncertainty in Resource Availability. Mathematics (MDPI).
- Qori Azmi Ayasy Siagian, M. S. (2024). SISTEM PENJADWALAN MATA PELAJARAN MEMAKAI ALGORITMA GENETIKA BERBASIS WEB Jurnal Informatika, Manajemen dan Teknologi, 26.
- Denisha Trihapningsari, M. A. (2024). Perancangan Basis Data Sistem Reservasi Wisma Universitas Terbuka dengan Metode Database Life Cycle. Jurnal Teknologi Sistem Informasi dan Aplikasi, 1638-1646.
- Anisa, E. (2025). Sistem Notifikasi Penjadwalan Perawatan Kendaraan BerbasisWeb dengan WhatsApp Gateway di PT Indokarya Teknik. JURNAL ILMIAH INFORMATIKA GLOBAL, VOLUME 16 No. 1.
- Fathi, H. (2023). PERANCANGAN SISTEM INFORMASI PENJADWALAN MATAKULIAH MENGGUNAKAN ALGORITMA GENETIK. JURNAL RAMATEKNO, 61–80.
- Aida Fitriyani, H. L. (2025). SISTEM PENJADWALAN MATA PELAJARAN MENGGUNAKAN ALGORITMA GENETIKA SMK PGRI RAWALUMBU. Jurnal Manajemen Informatika Jayakarta, 205-219.
- Gun Gun Gunawan Rahmat, R. D. (2024). IMPLEMENTASI SISTEM INFORMASI PENGOLAHAN JADWAL MATA PELAJARAN DI MTS WIJAYA SURYA BERBASIS WEBSITE. (Jurnal Mahasiswa Teknik Informatika).
- Aris Fajrianto, I. R. (2021). SISTEM PENJADWALAN MATA PELAJARAN SEKOLAH DASAR BERBASIS WEB. Jurnal Komputer dan Aplikasi. 154
- Lintang Aji Pangestu, S. H. (2023). Penerapan Algoritma Genetika Dalam Penjadwalan Mata Pelajaran . JURNAL INFORMATIKA, 194-205.

