

IMPLEMENTASI ALGORITMA *PRIORITY SCHEDULING* – *FIRST IN FIRST OUT* PADA SISTEM INFORMASI PEMINJAMAN RUANGAN BERBASIS WEB DI UNIVERSITAS ALMA ATA

Rizki Ramadhan, Andri Pramuntadi, Ahmad Subhan Yazid, Dhina Puspasari Wijaya

Teknik Informatika, Universitas Alma Ata

Jalan Brawijaya No.99, Jadan, Tamantirto, Kecamatan Kasihan, Bantul, Daerah Istimewa Yogyakarta

213200203@almaata.ac.id

ABSTRAK

Pengelolaan fasilitas ruangan di Universitas Alma Ata saat ini masih dilakukan secara konvensional, yang mengakibatkan masalah efisiensi seperti bentrok jadwal (*double booking*) dan ketiadaan transparansi informasi ketersediaan ruangan secara *real-time*. Selain itu, penyelesaian konflik jadwal sering kali bersifat subjektif tanpa parameter yang jelas. Penelitian ini bertujuan membangun Sistem Informasi Peminjaman Ruangan Berbasis Web yang menerapkan algoritma hibrida: *Priority Scheduling* dan *First In First Out* (FIFO) untuk menangani konflik jadwal secara otomatis dan objektif. Sistem dikembangkan dengan metode Waterfall, menggunakan bahasa PHP, kerangka kerja *CodeIgniter* 3, dan basis data *PostgreSQL*. Dalam sistem ini, *Priority Scheduling* berfungsi sebagai penentu keputusan utama berdasarkan akumulasi bobot prioritas pemohon dan jenis kegiatan, sedangkan algoritma FIFO digunakan sebagai mekanisme *tie-breaker* (pemecah seri) untuk memprioritaskan pengajuan yang lebih awal jika nilai prioritas setara. Hasil pengujian menggunakan *Black Box Testing* menunjukkan fitur fungsional berjalan sesuai spesifikasi. Pengujian *Algorithm Scenario Testing* membuktikan bahwa logika sistem mampu mendeteksi konflik dan memberikan rekomendasi keputusan yang akurat: memenangkan prioritas yang lebih tinggi secara mutlak, dan menerapkan keadilan waktu (FIFO) saat prioritas seimbang.

Kata kunci : Sistem Informasi, Peminjaman Ruangan, *Priority Scheduling*, FIFO, *CodeIgniter*, Universitas Alma Ata

1. PENDAHULUAN

Pendidikan pada jenjang perguruan tinggi memiliki peran strategis dalam menghasilkan sumber daya manusia yang unggul dan kompeten [1]. Sebagai bagian dari institusi pendidikan tinggi di Indonesia, Universitas Alma Ata senantiasa berupaya meningkatkan mutu layanan pendidikan yang diberikan kepada mahasiswa. Dalam mewujudkan visi tersebut, Universitas Alma Ata tidak hanya berfokus pada pengembangan kurikulum dan kualitas pengajaran, tetapi juga pada penyediaan sarana dan prasarana yang memadai. Salah satu fasilitas penting yang mendukung kelancaran proses akademik dan non-akademik adalah ruang kuliah, ruang seminar, ruang diskusi, dan ruang kegiatan lainnya. Ruangan-ruangan ini tidak hanya digunakan untuk kegiatan perkuliahan rutin, tetapi juga untuk berbagai kegiatan lain seperti sidang skripsi, pelatihan, rapat organisasi kemahasiswaan, seminar, bimbingan kelompok, serta kegiatan unit kegiatan mahasiswa (UKM). Keberadaan sarana dan prasarana yang mencukupi serta dikelola secara optimal berperan penting dalam mendukung terciptanya suasana pembelajaran yang efektif dan nyaman.

Meskipun fasilitas fisik telah tersedia, mekanisme pengelolaan peminjaman ruangan di Universitas Alma Ata saat ini masih dilakukan secara konvensional (manual). Prosedur yang berjalan mengharuskan mahasiswa atau dosen untuk datang langsung ke bagian akademik, mengisi formulir peminjaman secara fisik, dan menyerahkan surat pengantar. Petugas kemudian akan mengecek

ketersediaan ruangan dengan melihat arsip atau agenda harian.

Metode manual ini memiliki kelemahan mendasar dari sisi efisiensi dan akurasi data. Pertama, tidak adanya transparansi informasi secara *real-time*, sehingga sering terjadi asimetri informasi di mana peminjam telah menempuh prosedur administrasi namun ruangan yang dituju ternyata tidak tersedia. Kedua, tingginya risiko human error dalam pencatatan, yang berujung pada kasus jadwal ganda (*double booking*).

Ketika terdapat dua permohonan peminjaman untuk ruangan dan waktu yang sama, petugas akademik sering kali menghadapi dilema dalam menentukan prioritas. Tanpa adanya sistem yang mengatur bobot urgensi, keputusan persetujuan sering kali didasarkan pada "siapa yang datang lebih dulu" atau penilaian subjektif petugas. Hal ini berpotensi menimbulkan ketidakefektifan penggunaan fasilitas, di mana kegiatan yang sangat penting (seperti Seminar atau Pelatihan) bisa saja terhalang oleh kegiatan yang urgensinya lebih rendah (seperti Rapat Rutin) hanya karena pihak kedua mengajukan lebih awal.

Ketiadaan instrumen yang mampu menilai dan mengurutkan tingkat urgensi (*urgency ranking*) menyebabkan petugas akademik sering kali berada dalam posisi dilematis. Pengambilan keputusan menjadi subjektif atau terpaksa menolak kegiatan penting hanya karena ruangan sudah "terkunci" oleh kegiatan yang kurang prioritas. Hal ini berpotensi menyebabkan inefisiensi strategis dalam penggunaan fasilitas kampus. Dengan demikian, diperlukan suatu

sistem yang tidak terbatas pada fungsi pencatatan jadwal semata, tetapi juga memiliki logika untuk mengukur dan mengurutkan prioritas setiap permohonan yang masuk.

Penelitian terdahulu menunjukkan bahwa algoritma Priority Scheduling efektif dalam menentukan urutan pelayanan berdasarkan tingkat kepentingan [2], sementara algoritma FIFO menjamin keadilan antrian berdasarkan waktu [3]. Namun, belum banyak penelitian yang menggabungkan kedua algoritma ini dalam sistem peminjaman ruangan untuk menyeimbangkan urgensi dan keadilan waktu. Penelitian ini bertujuan mengisi celah tersebut dengan mengimplementasikan algoritma hibrida (Priority Scheduling - FIFO) pada sistem berbasis web untuk memberikan rekomendasi keputusan yang objektif, transparan, dan adil.

2. TINJAUAN PUSTAKA

Sejumlah penelitian telah dilakukan untuk mengatasi permasalahan penjadwalan fasilitas umum. Fajri dan Mujiastuti mengembangkan sistem peminjaman ruangan di Fakultas Teknik Universitas Muhammadiyah Jakarta. Meskipun berhasil mendigitalkan proses bisnis, sistem tersebut masih menggunakan alur persetujuan linear tanpa penanganan konflik otomatis [4]. Hal serupa ditemukan pada penelitian Adinugroho di PT. Multi Aneka Pangan Nusantara, yang berfokus pada fitur validasi berjenjang namun belum menyertakan algoritma pendukung keputusan saat terjadi bentrok jadwal [5].

Dalam konteks algoritma penjadwalan, Rohmah dan Gunawan membuktikan efektivitas algoritma Priority Scheduling pada Sistem Pelayanan Administrasi Desa. Penelitian tersebut menyimpulkan bahwa mekanisme prioritas mampu meningkatkan kepuasan layanan bagi kelompok rentan yang membutuhkan penanganan segera [2]. Di sisi lain, Noptrina menerapkan algoritma FIFO (*First In First Out*) pada sistem penjadwalan praktikum. Hasilnya menunjukkan bahwa FIFO menjamin aspek keadilan antrian (*fairness*), namun memiliki kelemahan dalam mengakomodasi situasi mendesak (*urgency*) [3].

Penelitian ini mengisi celah (*gap*) dari studi-studi terdahulu dengan mengintegrasikan kedua algoritma tersebut. Jika penelitian sebelumnya cenderung memilih salah satu antara prioritas atau keadilan waktu, penelitian ini menggabungkan keduanya: Priority Scheduling digunakan untuk menangani urgensi, dan FIFO digunakan untuk menjaga keadilan saat urgensi setara.

2.1. Sistem Informasi

Secara umum, sistem informasi berfungsi untuk mengelola siklus informasi dalam organisasi, mulai dari pengumpulan, pemrosesan, penyimpanan, hingga pendistribusian. Penerapan sistem ini mampu mengoptimalkan operasional menjadi lebih efektif dan efisien, serta memperbaiki kualitas layanan dan

menjembatani interaksi antara pengelola dengan pengguna. Oleh karena itu, di tengah kemajuan era digital, sistem informasi menjadi sarana andalan untuk mendukung pencapaian target, baik bagi institusi maupun perorangan[6]

2.2. Algoritma Priority Scheduling

Algoritma *Priority Scheduling* beroperasi dengan menentukan urutan eksekusi proses berdasarkan tingkat prioritas tertinggi yang dimiliki oleh masing-masing proses[7].

2.3. Algoritma First In First Out (FIFO)

Algoritma *First In First Out* (FIFO) adalah jenis algoritma yang beroperasi menggunakan konsep antrian tanpa tingkatan prioritas. Dengan memanfaatkan basis struktur data, algoritma ini sering diaplikasikan dalam berbagai solusi masalah kehidupan nyata maupun sistem teknologi. Prinsip kerjanya berjalan secara bergiliran dan teratur, di mana elemen yang pertama kali masuk ke dalam sistem akan diproses paling awal mengikuti alur antrian yang ada[8].

2.4. Metode Waterfall

Metode *waterfall* merupakan salah satu pendekatan klasik dalam pengembangan perangkat lunak yang menerapkan alur kerja secara berurutan dan bertahap. Pendekatan ini umumnya terdiri atas lima hingga tujuh fase, di mana setiap fase memiliki fungsi dan tujuan yang jelas. Keseluruhan tahapan tersebut merepresentasikan siklus hidup perangkat lunak, dimulai dari tahap perencanaan hingga proses akhir penyerahan sistem [10]. Adapun tahapan-tahapan pada metode ini sebagai berikut :

- a. Analisis Kebutuhan (Requirement Analysis)
Tahapan awal ini menitikberatkan pada proses identifikasi serta pemahaman kebutuhan pengguna dan sistem. Pada fase ini, analisis sistem berkolaborasi dengan pihak klien untuk merumuskan secara detail fitur, fungsi, serta batasan perangkat lunak yang akan dikembangkan. Luaran dari tahap tersebut berupa dokumen spesifikasi kebutuhan perangkat lunak yang digunakan sebagai dasar pelaksanaan tahapan berikutnya.
- b. Desain Sistem (System Design)
Pada tahap ini, arsitektur sistem dirancang berdasarkan dokumen kebutuhan. Proses desain mencakup perancangan struktur data, arsitektur perangkat lunak, antarmuka pengguna (UI), serta algoritma dan logika sistem. Tujuan utama dari tahap ini adalah menghasilkan blueprint teknis yang akan digunakan oleh tim pengembang dalam proses implementasi.
- c. Penulisan Kode Program (*Coding*)
Tahapan ini merupakan fase implementasi, yaitu proses mengonversi rancangan sistem ke dalam bentuk kode program dengan menggunakan bahasa pemrograman yang telah ditetapkan. Setiap

komponen atau modul yang telah dirancang kemudian diwujudkan oleh tim pengembang sesuai dengan standar pengembangan yang berlaku. Luaran dari tahap ini adalah versi awal perangkat lunak yang siap untuk memasuki proses pengujian

d. Pengujian Program (Testing)

Setelah proses penulisan kode diselesaikan, perangkat lunak menjalani tahap pengujian untuk memastikan seluruh fungsi beroperasi sesuai dengan kebutuhan yang telah ditetapkan. Pengujian ini bertujuan untuk mengidentifikasi serta memperbaiki kesalahan atau cacat (*bug*) yang terdapat dalam sistem. Metode pengujian yang digunakan dapat mencakup unit testing, pengujian integrasi, pengujian sistem, hingga *user acceptance testing* (UAT).

e. Penerapan Program dan Pemeliharaan (*Deployment and Maintenance*)

Tahap akhir adalah penerapan perangkat lunak ke lingkungan nyata atau produksi. Setelah diimplementasikan, perangkat lunak akan terus dipelihara untuk memastikan stabilitas, memperbaiki *bug* yang mungkin muncul, serta menyesuaikan dengan kebutuhan baru. Tahap pemeliharaan meliputi kegiatan pembaruan fitur, optimalisasi kinerja sistem, serta penyesuaian terhadap perkembangan teknologi maupun perubahan regulasi yang berlaku.

2.5. PHP

PHP merupakan bahasa pemrograman skrip yang dijalankan pada sisi *server* (*server-side*) dan pada awalnya dikembangkan untuk kebutuhan pembuatan aplikasi berbasis web, meskipun dalam praktiknya juga dapat dimanfaatkan sebagai bahasa pemrograman serbaguna. Bahasa pemrograman ini pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994. Saat ini, PHP dikenal sebagai singkatan rekursif dari PHP: *Hypertext Preprocessor*. PHP bersifat open source dan dapat digunakan secara bebas berdasarkan PHP License, yaitu lisensi yang memiliki ketentuan tersendiri dan berbeda dari GNU *General Public License* (GPL) secara umum[11].

2.6. PostgreSQL

PostgreSQL merupakan sistem manajemen basis data relasional yang awal pengembangannya dilakukan di Departemen Ilmu Komputer Universitas California, Berkeley. Perangkat lunak ini bersifat bebas lisensi dan kompatibel dengan berbagai platform. Sebagai basis data relasional, *PostgreSQL* mendukung penuh perintah SQL serta bahasa prosedural *PL/pgSQL*. Keunggulan lain dari sistem ini adalah dukungan komunitas yang besar, ketersediaan dokumentasi yang lengkap, serta kompatibilitasnya dengan beragam bahasa pemrograman populer seperti C++, Java, Perl, PHP, Python, dan Tcl[12].

3. METODE PENELITIAN

3.1. Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini adalah model Waterfall. Metode ini dipilih karena pendekatannya yang sistematis dan berurutan, memastikan setiap fase diselesaikan dengan baik sebelum melangkah ke fase berikutnya. Tahapan yang dilakukan meliputi:

- Analisis Kebutuhan: Dilakukan melalui observasi alur peminjaman manual dan wawancara dengan staf akademik Universitas Alma Ata untuk mengidentifikasi permasalahan utama, yaitu bentrok jadwal dan subjektivitas penentuan prioritas.
- Desain Sistem: Merancang arsitektur sistem, basis data, dan antarmuka pengguna.
- Implementasi (Coding): Penulisan kode program menggunakan bahasa PHP dengan framework CodeIgniter 3.
- Pengujian: Memastikan sistem berjalan bebas dari kesalahan (*bug*) dan logika algoritma berfungsi sesuai rancangan.
- Pemeliharaan: Tahap akhir untuk perbaikan jika ditemukan kesalahan pasca-implementasi.

3.2. Implementasi Algoritma Priority Scheduling

Dalam penelitian ini, algoritma Priority Scheduling diimplementasikan menggunakan pendekatan Weighted Scoring Model (Model Penilaian Terbobot). Algoritma ini berfungsi untuk mengkalkulasi tingkat urgensi setiap permohonan peminjaman ruangan yang masuk ke dalam sistem.

a. Jenis Pemohon

Tabel 1. Jenis Pemohon

Pemohon	Nilai
Rektorat	5
Direktorat	4
Dekan	3
Dosen Prodi	2
Mahasiswa	1

b. Jenis Kegiatan

Tabel 2. Kegiatan Rektorat

Rektorat	
Nilai	Kegiatan
5	Rapat Universitas
4	Penandatanganan MoU/MoA
3	Penerimaan Kunjungan Resmi
2	Rapat Pimpinan
1	Briefing Staf Rektorat

Tabel 3. Kegiatan Direktorat

Direktorat	
Nilai	Kegiatan
5	Audit Mutu / Akreditasi
4	Kunjungan Tamu / Kerjasama Eksternal
3	Workshop / Pelatihan Pusat
2	Sosialisasi Kebijakan Pusat
1	Rapat Divisi Internal

Tabel 4. Kegiatan Dekan

Dekan	
Nilai	Kegiatan
5	Akreditasi
4	Rapat Fakultas
3	Seminar Internasional
2	Seminar Nasional
1	Rapat Koordinasi Dosen

Tabel 5. Kegiatan Dosen Prodi

Dosen Prodi	
Nilai	Kegiatan
5	Akreditasi
4	Webinar/Seminar
3	Sosialisasi
2	Rapat Dosen
1	Bimbingan / Konsultasi Akademik

Tabel 6. Kegiatan Mahasiswa

Mahasiswa	
Nilai	Kegiatan
5	Musyawarah Besar (Mubes)
4	Seminar / Event Himpunan
3	Lomba / Kompetisi Mahasiswa
2	Kegiatan UKM
1	Rapat Rutin

Setiap variabel (Pemohon dan Kegiatan) dikonversi terlebih dahulu menjadi nilai skalar 1 sampai 5 sesuai tabel parameter yang telah ditentukan. Selanjutnya, perhitungan Skor Prioritas (S_p) dilakukan menggunakan persamaan berikut:

$$S_p = (N_p * 0,6) + (N_k * 0,4) \quad (1)$$

Dimana :

S_p = Skor Prioritas Akhir (Hasil perhitungan).

N_p = Nilai Jenis Pemohon (Skala 1-5).

N_k = Nilai Jenis Kegiatan (Skala 1-5)

0,6 = Konstanta bobot untuk Pemohon (60%)

0,4 = Konstanta bobot untuk Kegiatan (40%).

3.3. Implementasi Algoritma Priority Scheduling

Pada kasus peminjaman ruangan, algoritma FIFO tidak menggunakan perhitungan prioritas tambahan. Hanya mempertimbangkan waktu kedatangan ($t_{arrival}$), dimana :

$$Urutan = \text{Sort}(t_{arrival}) \quad (2)$$

Dimana $t_{arrival}$ merepresentasikan waktu pencatatan data ke dalam sistem basis data (*entry time*).

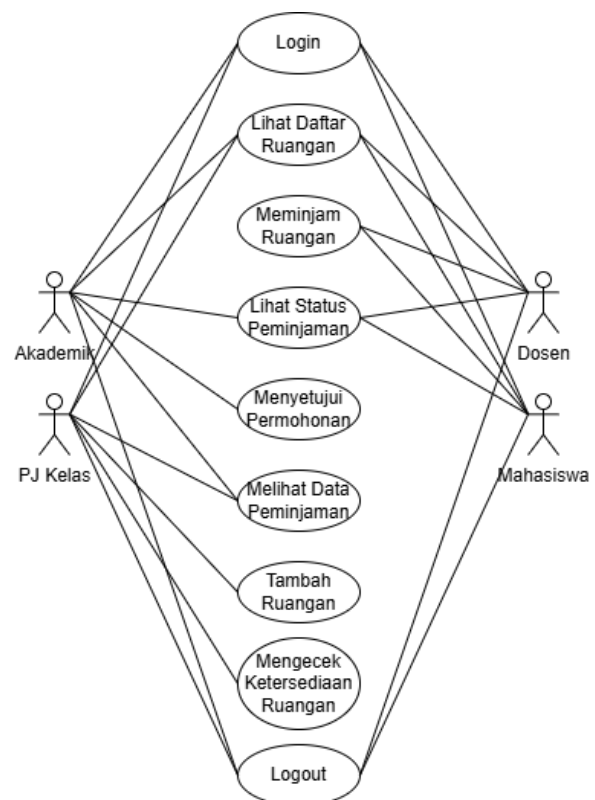
3.4. Kombinasi Implementasi Algoritma Priority Scheduling dan FIFO

Integrasi kedua algoritma dilakukan dengan menetapkan Skor Prioritas (S_p) sebagai kunci pengurutan utama (*primary sort key*) dan Waktu Pengajuan ($t_{arrival}$) sebagai kunci pengurutan kedua (*secondary sort key*).

Mekanisme kombinasi ini bekerja dengan membandingkan dua permohonan (misalkan Permohonan A dan Permohonan B) yang berada dalam satu daftar antrian. Logika komparasi sistem mengikuti aturan keputusan (*decision rule*) sebagai berikut:

- Kondisi 1 (Dominasi Prioritas) : Jika Skor Prioritas A (S_{pA}) lebih besar dari Skor Prioritas B (S_{pB}), maka A ditempatkan di posisi lebih atas daripada B, tanpa mempedulikan kapan waktu pengajuannya.
- Kondisi 2 (Nilai Seri / *Tie-Breaker*): Jika Skor Prioritas A sama persis dengan Skor Prioritas B ($S_{pA} = S_{pB}$), maka sistem akan mengaktifkan algoritma FIFO dengan membandingkan timestamp waktu pengajuan. Permohonan dengan waktu lebih awal akan ditempatkan di posisi atas.

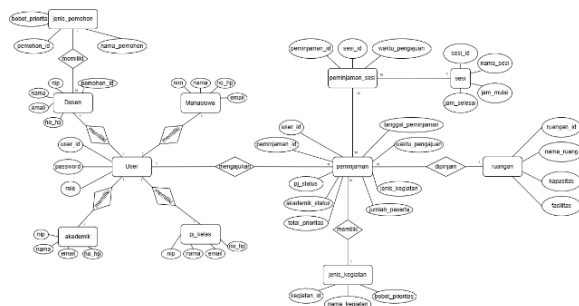
3.5. User Case Diagram



Gambar 1. User Case Diagram

Diagram di atas memvisualisasikan ruang lingkup fungsionalitas sistem yang melibatkan interaksi empat aktor utama, yaitu Mahasiswa dan Dosen sebagai peminjam, PJ Kelas sebagai validator awal, serta Akademik sebagai pengelola utama. Seluruh interaksi diawali dengan use case 'Login', yang bersifat wajib (*include*) bagi semua aktor untuk mendapatkan hak akses ke dalam sistem.

3.6. ERD



Gambar 2. ERD

Struktur basis data sistem dirancang menggunakan Entity Relationship Diagram (ERD) untuk memodelkan relasi antar entitas yang mendukung integritas data dan penerapan algoritma. ERD ini menghubungkan entitas utama yaitu Tabel Pengguna (*Users*), Tabel Ruang, dan Tabel Peminjaman.

3.7. Flowchart Mahasiswa

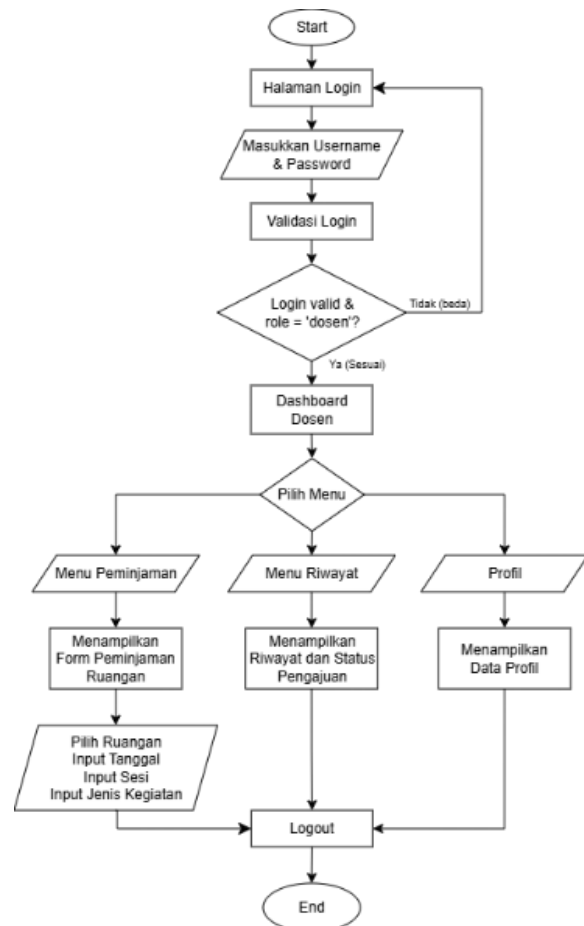


Gambar 3. Flowchart Mahasiswa

Proses penggunaan sistem oleh mahasiswa diawali dengan tahapan otentikasi menggunakan

Nomor Induk Mahasiswa (NIM) dan kata sandi untuk mengakses halaman dashboard. Di dalam sistem, mahasiswa dapat mengakses fitur utama Peminjaman untuk melakukan reservasi ruangan, yang mewajibkan pengunduhan templat surat pengantar dan pengunggahan kembali dokumen tersebut setelah disetujui dosen pembimbing sebagai syarat validasi. Selain itu, mahasiswa dapat menggunakan menu Riwayat untuk memantau status permohonan secara real-time dan menu Profil untuk pemutakhiran data pribadi. Alur aktivitas berakhir ketika mahasiswa menyelesaikan transaksi peminjaman atau melakukan logout dari sistem.

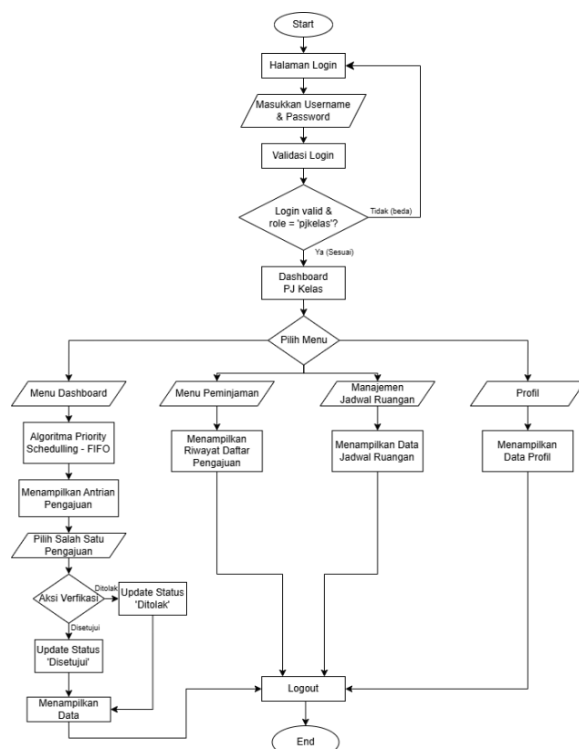
3.8. Flowchart Dosen



Gambar 4. Flowchart Dosen

Aktivitas dosen dimulai dengan proses otentikasi menggunakan NIP dan kata sandi untuk mengakses dashboard utama. Sistem menyediakan tiga fitur utama bagi dosen: menu Peminjaman untuk memilih ruangan dan mengajukan jadwal, menu Riwayat untuk memantau status persetujuan permohonan (waiting, disetujui, atau dibatalkan), serta menu Profil untuk pengelolaan data akun pribadi. Alur sistem berakhir ketika dosen menyelesaikan transaksi peminjaman atau melakukan logout.

3.9. Flowchart PJ Kelas



Gambar 5. Flowchart PJ Kelas

Setelah proses otentikasi berhasil, PJ Kelas mengakses *dashboard* yang menampilkan daftar antrian peminjaman yang telah diurutkan otomatis oleh algoritma Priority Scheduling-FIFO. PJ Kelas bertugas memverifikasi ketersediaan ruangan; jika tersedia, permohonan langsung disetujui. Apabila ruangan yang diminta penuh, sistem merekomendasikan alternatif ruangan sesuai kapasitas untuk disetujui. Namun, jika tidak ada ketersediaan ruangan sama sekali, PJ Kelas menolak pengajuan dan status diperbarui menjadi 'Ditolak'. Proses berakhir saat keputusan tersimpan dalam sistem.

3.10. Flowchart Akademik

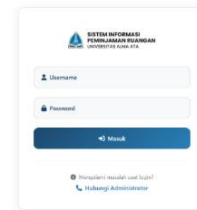


Gambar 6. Flowchart Akademik

Pasca otentikasi, pengguna Akademik mengakses *dashboard* yang menampilkan daftar pengajuan terverifikasi oleh PJ Kelas untuk dilakukan validasi akhir (pengesahan). Selain fungsi otorisasi, Akademik memiliki privilege manajemen data (CRUD) untuk entitas Ruangan, Dosen, dan Mahasiswa, serta dapat memantau riwayat peminjaman yang telah disahkan melalui menu Peminjaman.

4. HASIL DAN PEMBAHASAN

4.1. Tampilan Login



Gambar 7. Tampilan Login

Halaman login dirancang sebagai gerbang keamanan utama yang berfungsi untuk membatasi hak akses sistem hanya kepada pengguna yang telah terdaftar. Pada antarmuka ini, pengguna diwajibkan memasukkan kredensial akun berupa identitas pengguna (NIP atau NIM) serta kata sandi yang valid. Sistem kemudian akan memproses permintaan tersebut melalui mekanisme autentikasi dengan memverifikasi kecocokan data masukan terhadap rekaman yang tersimpan pada tabel users di basis data. Selain memvalidasi kebenaran data, sistem juga menerapkan logika pengalihan (redirect) berbasis peran; setelah login berhasil, pengguna secara otomatis akan diarahkan menuju halaman dashboard yang spesifik sesuai dengan hak aksesnya (role) baik sebagai Admin Akademik, PJ Kelas, Dosen, maupun Mahasiswa, sehingga menjamin keamanan dan isolasi fitur yang tepat sasaran.

4.2. Tampilan Dashboard Mahasiswa dan Dosen

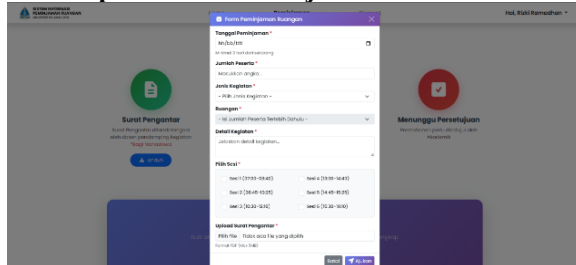


Gambar 8. Tampilan Dashboard Mahasiswa dan Dosen

Halaman utama atau dashboard ini merupakan tampilan awal yang disajikan kepada pengguna (Mahasiswa dan Dosen) setelah berhasil masuk ke dalam sistem. Antarmuka dirancang dengan konsep landing page yang modern dan informatif untuk

memberikan pemahaman awal kepada pengguna mengenai mekanisme kerja sistem. Pada bagian atas (header), terdapat menu navigasi utama yang memudahkan akses menuju halaman form "Peminjaman" dan halaman "Riwayat" transaksi.

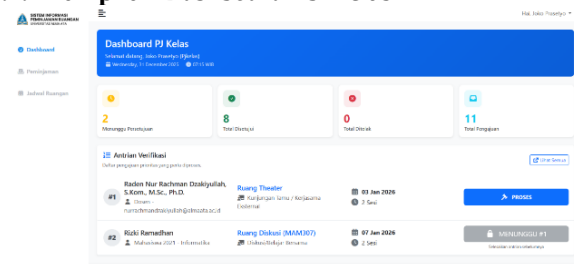
4.3. Tampilan Form Peminjaman



Gambar 9. Form Peminjaman

Antarmuka formulir peminjaman dirancang menggunakan konsep modal popup yang responsif untuk mengumpulkan data variabel utama yang dibutuhkan oleh algoritma *Priority Scheduling* dan FIFO. Secara umum, kedua jenis pengguna diwajibkan mengisi atribut dasar meliputi pilihan ruangan, tanggal peminjaman, jumlah peserta, keterangan, serta pemilihan sesi waktu (*time slots*) berbasis *checkbox*. Pengguna juga wajib memilih "Jenis Kegiatan" yang akan menjadi parameter kunci dalam penentuan bobot prioritas algoritma. Jika diakses oleh Dosen, kolom unggah dokumen tersebut dihilangkan dan digantikan oleh panel informasi berwarna hijau (success alert) yang menyatakan bahwa dosen tidak perlu menyertakan surat pengantar, disertai fleksibilitas waktu peminjaman yang dapat dilakukan minimal 1 hari sebelum kegiatan.

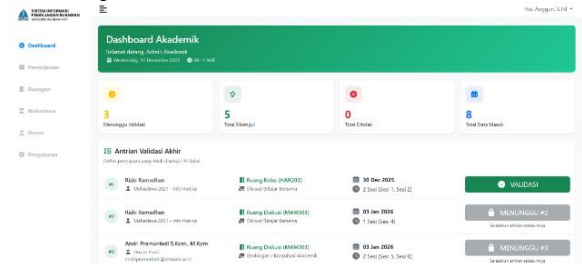
4.4. Tampilan Dashboard PJ Kelas



Gambar 10. Dashboard PJ Kelas

Tampilan *dashboard* PJ Kelas menampilkan daftar pengajuan peminjaman ruangan yang masuk ke dalam sistem dan telah diurutkan secara otomatis berdasarkan hasil penerapan algoritma *Priority Scheduling*-FIFO. Urutan pengajuan ditampilkan berdasarkan tingkat prioritas tertinggi, dan apabila memiliki nilai prioritas yang sama maka diurutkan berdasarkan waktu pengajuan paling awal. Melalui tampilan ini, PJ Kelas dapat dengan mudah melakukan verifikasi terhadap pengajuan yang masuk secara terstruktur dan sistematis.

4.5. Tampilan Dashboard Akademik



Gambar 11. Dashboard Akademik

Tampilan dashboard Akademik merupakan halaman yang digunakan untuk melakukan verifikasi akhir terhadap pengajuan peminjaman ruangan yang telah diproses oleh PJ Kelas. Pada halaman ini ditampilkan daftar pengajuan lengkap dengan informasi pemohon, jadwal peminjaman, ruangan yang diajukan atau direkomendasikan, serta status verifikasi sebelumnya. Melalui dashboard ini, pihak Akademik dapat meninjau setiap pengajuan dan memberikan keputusan persetujuan atau penolakan sebagai bentuk pengesahan akhir, sehingga pengajuan peminjaman dinyatakan sah dan tercatat secara resmi di dalam sistem.

4.6. Implementasi Kode Algoritma

Implementasi kode algoritma merupakan tahap realisasi logika bisnis (*business logic*) ke dalam sintaks pemrograman PHP pada kerangka kerja CodeIgniter 3. Bagian ini menjadi otak dari sistem, di mana aturan-aturan abstrak mengenai prioritas dan antrian diterjemahkan menjadi fungsi matematis yang dieksekusi oleh server untuk mengelola urutan data.

Berbeda dengan sistem konvensional yang menampilkan data secara acak atau sekadar berdasarkan waktu masuk, sistem ini dirancang untuk menyajikan Daftar Antrian Terurut. Sistem menerapkan pendekatan hibrida: *Priority Scheduling* sebagai penentu urutan utama (*primary sort key*) berdasarkan skor urgensi, dan FIFO (*First In First Out*) sebagai penentu urutan sekunder (*secondary sort key*) jika terdapat skor yangimbang.

Berikut adalah tahapan implementasi kode program yang dibagi berdasarkan alur kerja algoritma

4.7. Logika Pengambilan dan Perhitungan Bobot Prioritas

Sebelum data ditampilkan dalam daftar antrian, sistem terlebih dahulu harus mengkuantifikasi nilai urgensi dari setiap permohonan yang masuk. Pada tahap ini, implementasi dilakukan pada Model Peminjaman (*Peminjaman_model.php*). Fungsi ini bertugas mengkalkulasi skor akhir dengan menerapkan bobot persentase sesuai rumus yang ditetapkan: 60% Nilai Pemohon + 40% Nilai Kegiatan.

```
// 1. TENTUKAN NILAI PEMOHON (Bobot 60%)
$role = isset($data['type_peminjam']) ? $data['type_peminjam'] : 'mahasiswa';
$user_id = isset($data['user_id']) ? $data['user_id'] : null;
$nilai_pemohon = 1;
if ($role == 'mahasiswa') {
    $nilai_pemohon = 1;
} elseif ($role == 'dosen') {
    if ($user_id) {
        $this->db->select('jp.bobot_prioritas');
        $this->db->from('dosen d');
        $this->db->join('jenis_pemohon jp', 'd.pemohon_id = jp.pemohon_id', 'left');
        $this->db->where('d.nip', $user_id);
        $dosen_data = $this->db->get()->row();
        $nilai_pemohon = $dosen_data > (float) $dosen_data->bobot_prioritas : 2;
    } else {
        $nilai_pemohon = 2;
    }
}

// 2. TENTUKAN NILAI KEGIATAN (Bobot 40%)
$nilai_kegiatan = 1;
if (!empty($data['jenis_kegiatan_id'])) {
    $q_kegiatan = $this->db->get_where('jenis_kegiatan', [
        'jenis_kegiatan_id' => $data['jenis_kegiatan_id']
    ])->row();
    if ($q_kegiatan) {
        $nilai_kegiatan = (float) $q_kegiatan->bobot_prioritas;
    }
}

// 3. HITUNG SKOR AKHIR (WSM)
// (Nilai Pemohon x 60%) + (Nilai Kegiatan x 40%)
$total_score = ($nilai_pemohon * 0.60) + ($nilai_kegiatan * 0.40);
return (float) $total_score;
}
```

Gambar 12. Logika pengambilan dan perhitungan bobot prioritas

Fungsi di atas memastikan setiap data peminjaman yang disimpan ke dalam database memiliki atribut `total_prioritas` yang akurat. Nilai inilah yang kemudian menjadi parameter utama dalam proses pengurutan.

4.8. Implementasi Algoritma Priority Scheduling dengan Tie-Breaker FIFO

Inti dari penyelesaian masalah "ketidakjelasan prioritas" diimplementasikan pada fungsi pengambilan data (*fetch*). Alih-alih melakukan pemrosesan array

yang kompleks untuk memberikan rekomendasi, sistem memanfaatkan efisiensi *Query Builder CodeIgniter* untuk mengurutkan data langsung dari sumbernya.

```
// --- INTI ALGORITMA HIBRIDA ---

// 1. Priority Scheduling: Prioritas skor tinggi di atas
$this->db->order_by('p.total_prioritas', 'DESC');

// 2. FIFO: Jika skor sama, yang daftar duluan di atas
$this->db->order_by('p.waktu_pengajuan', 'ASC');

// Eksekusi Query
$result = $this->db->get();
return $result->result();
```

Gambar 13. Implementasi Algoritma Priority Scheduling dengan Tie-Breaker FIFO

Dengan implementasi kode tersebut, sistem secara otomatis menyajikan data kepada PJ Kelas maupun Admin Akademik dalam keadaan sudah terurut. Hal ini menghilangkan beban kognitif petugas dalam membandingkan urgensi secara manual. Petugas cukup memproses pengajuan mulai dari urutan teratas (baris pertama tabel), karena secara matematis baris tersebut adalah pengajuan dengan tingkat urgensi tertinggi (atau tercepat jika urgensinya setara).

4.9. Pengujian

Pengujian dilakukan untuk memvalidasi fungsionalitas sistem dan logika algoritma dalam menangani konflik. Metode yang digunakan adalah *Black Box Testing* untuk fitur fungsional.

Tabel 7. Pengujian Black Box

No	Pengujian	Hasil yang diharapkan	Hasil
1	Mengakses sistem dengan data kredensial yang salah	Sistem akan menolak akses dan menampilkan pesan "Username atau Password salah".	Sesuai
2	Mengakses sistem dengan data valid	Sistem menerima akses dan mengarahkan ke Dashboard Mahasiswa.	Sesuai
3	Mengosongkan salah satu field wajib (misal: Keterangan)	Sistem menampilkan pesan validasi "Bidang Keterangan Kegiatan wajib diisi".	Sesuai
4	Mengajukan tanpa mengunggah Surat Pengantar	Sistem menolak dan meminta unggah file "File surat pengantar wajib diupload".	Sesuai
5	Melakukan validasi akhir	Status akhir akademik_status berubah menjadi 'disetujui' (resmi). Sistem menerbitkan/mengirim status ke mahasiswa.	Sesuai
6	Manajemen Data (Dosen, Mahasiswa, Ruangan, Pengaturan)	Data berhasil di tambah, edit, hapus	Sesuai
7	Akses Dashboard	Sistem menampilkan daftar pengajuan yang sudah terurut otomatis	Sesuai
8	PJ Kelas memasukkan jadwal kuliah rutin	Sistem menyimpan jadwal dan slot tersebut menjadi unavailable (tidak bisa dipinjam orang lain).	Sesuai
9	PJ Kelas menyetujui pengajuan	Status pj_status berubah jadi 'Disetujui', namun akademik_status masih 'Waiting'. Data masuk ke dashboard Admin Akademik.	Sesuai
10	PJ Kelas menolak pengajuan	Status berubah jadi 'Ditolak'. Data TIDAK muncul di dashboard Admin Akademik	Sesuai

5. KESIMPULAN DAN SARAN

Berdasarkan seluruh tahapan penelitian yang telah dilalui—mulai dari analisis kebutuhan, perancangan sistem, implementasi kode program, hingga pengujian sistem menggunakan metode *Black Box* serta skenario algoritma—dapat disimpulkan

bahwa penelitian ini berhasil merancang dan membangun Sistem Informasi Peminjaman Ruangan Berbasis Web di Universitas Alma Ata dengan menerapkan algoritma *Priority Scheduling* dan *First In First Out* (FIFO).

Sistem yang dibangun saat ini telah memenuhi kebutuhan dasar dan tujuan utama penelitian. Namun, demi pengembangan sistem yang lebih sempurna di masa mendatang, penulis mengajukan beberapa saran sebagai berikut: Integrasi dengan SIAKAD (Sistem Informasi Akademik) Saat ini, data pengguna (Mahasiswa dan Dosen) masih dikelola secara mandiri (*standalone*) dalam basis data aplikasi. Untuk pengembangan selanjutnya, disarankan agar sistem dapat terintegrasi langsung dengan database SIAKAD Universitas Alma Ata melalui API (*Application Programming Interface*). Hal ini bertujuan agar data pengguna tersinkronisasi secara otomatis tanpa perlu input manual ulang saat ada mahasiswa baru atau pergantian dosen. Untuk pengembangan penelitian selanjutnya, disarankan agar sistem dapat mengimplementasikan metode optimasi yang lebih adaptif.

DAFTAR PUSTAKA

- [1] H. Humiati and D. Budiarti, "Peran Perguruan Tinggi Dalam Meningkatkan Sumber Daya Manusia," *JMM - J. Masy. Merdeka*, vol. 3, no. 1, pp. 13–24, 2024, doi: 10.51213/jmm.v3i1.46.
- [2] A. A. Rohmah and D. Gunawan, "Implementasi Algoritma Priority Scheduling Sistem Informasi Pelayanan Administrasi Kependudukan Desa," *J. Inform. J. Pengemb. IT*, vol. 8, no. 3, pp. 181–187, 2023, doi: 10.30591/jpit.v8i3.4891.
- [3] N. Noptrina, A. Pramuntadi, and D. P. Wijaya, "Perancangan Sistem Penjadwalan Praktikum Menggunakan Algoritma Fifo (First in First Out) Berbasis Website," *J. Inform. Teknol. dan Sains*, vol. 6, no. 2, pp. 375–385, 2024, doi: 10.51401/jinteks.v6i2.4520.
- [4] N. Z. Fajri and R. Mujiastuti, "Sistem Informasi Peminjaman Ruangan dan Barang pada Fakultas Teknik Universitas Muhammadiyah Jakarta Berbasis Website," vol. 15, no. 1, pp. 299–307, 2024.
- [5] M. Adinugroho, T. Herlambang, F. Yudianto, and D. B. Magfira, "Perancangan sistem informasi peminjaman ruangan pada PT. Multi Aneka Pangan Nusantara," *Indones. Berdaya*, vol. 4, no. 3, pp. 1045–1058, 2023, doi: 10.47679/ib.2023518.
- [6] M. F. Adham, "Analisis Implementasi Sistem Informasi: Studi Literatur," *J. Teknol. Sist. Inf.*, vol. 5, no. 1, pp. 264–275, 2024, doi: 10.35957/jtsi.v5i1.7815.
- [7] H. D. Intan, "Sistem Informasi Penjadwalan Perjalanan Dinas Anggota DPRD Kabupaten Kutai Barat Menggunakan Algoritma Priority Scheduling Berbasis Web".
- [8] F. Fitriani and Y. Apridiansyah, "Aplikasi Antrian Pembayaran Uang Kuliah Berbasis Android Menggunakan Algoritma Priority Scheduling Berbasis Web," *JUSIBI (Jurnal Sist. Inf. dan E-Bisnis)*, vol. 3, no. 2, pp. 91–103, 2021, doi: 10.54650/jusibi.v3i2.384.
- [9] R. Yusran, A. Nanda, A. Amalda, R. Luthvia, and R. Fadlan, "Upaya Pemenuhan Kesadaran Masyarakat dan Pemenuhan Gizi Seimbang untuk Mencegah Peningkatan Angka Stunting di Nagari Pariangan 2023," *Inov. J. Pengabd. Masy.*, vol. 1, no. 2, pp. 131–140, 2023, doi: 10.54082/ijpm.138.
- [10] R. Amelia, D. P. Wijaya, A. Pramuntadi, and D. Danianti, "Sistem Monitoring Mahasiswa Magang Prodi Informatika UAA Berbasis Website Menggunakan Metode Waterfall," no. 2022, pp. 699–707, 2024.
- [11] Rina Noviana, "Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan Mysql," *J. Tek. dan Sci.*, vol. 1, no. 2, pp. 112–124, 2022, doi: 10.56127/jts.v1i2.128.
- [12] D. A. B. Prasetyo, "Implementasi Information Schema Database Pada PostgreSQL untuk Pembuatan Tabel Informasi dengan Menggunakan Python Di PT XYZ," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 1961–1972, 2022, doi: 10.35957/jatisi.v9i3.2221.