

Monitoring Otomatis dan *Priority-Based Alerting* pada Sistem Penjadwalan Kuliah Berbasis Web

Mei Linda Cahyandari¹, Juliah Kurniasih², Dina Yulina Heriyani³

¹Program Studi Informatika, Fakultas Teknik, Universitas Sarjanawiyata Tamansiswa, Yogyakarta, Indonesia

Email: ¹2022018002@ustjogja.ac.id, ²juliah.kurniasih@ustjogja.ac.id, ³dina.heriyani@ustjogja.ac.id

INFORMASI ARTIKEL

Histori artikel:

Naskah masuk, 22 Juni 2026

Direvisi, 27 Juli 2026

Diiterima, 31 Juli 2026

Kata Kunci:

penjadwalan kuliah, monitoring otomatis, priority-based alerting, priority alert scoring, notifikasi WhatsApp.

ABSTRAK

Abstract- Changes to class schedules after publication often delay information because students and lecturers must manually check the academic website. This study develops an automated mechanism to detect schedule changes, assign information priority, and distribute notifications to affected users. The scientific novelty lies in integrating an existing Genetic Algorithm-based scheduling system with a monitoring service, Priority Alert Scoring Algorithm, monitoring log, and website and WhatsApp notifications in one post-scheduling workflow. The system was developed through requirements analysis, design, implementation, and testing. Priority scores combine severity of the changed attribute and urgency based on the remaining time before class. Testing comprised five automated Black Box scenarios using Playwright on Chromium and Firefox, 20 schedule-change detection trials, and 50 notification-delivery trials. All functional scenarios were valid. The average detection time was 0.15 seconds, the average notification response time was 2.3 seconds, and all 50 notifications were delivered successfully. Within the test environment, the integrated mechanism met the operational targets and supported faster, more reliable, and context-sensitive communication of schedule changes.

Abstrak- Perubahan jadwal perkuliahan setelah jadwal diterbitkan sering menimbulkan keterlambatan informasi karena mahasiswa dan dosen harus memeriksa laman akademik secara mandiri. Penelitian ini mengembangkan mekanisme otomatis untuk mendeteksi perubahan jadwal, menentukan prioritas informasi, dan mendistribusikan notifikasi kepada pengguna terdampak. Kebaruan ilmiah penelitian terletak pada integrasi sistem penjadwalan berbasis Genetic Algorithm yang telah tersedia dengan monitoring service, Priority Alert Scoring Algorithm, monitoring log, serta notifikasi website dan WhatsApp dalam satu alur pasca-penjadwalan. Pengembangan sistem meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian. Skor prioritas menggabungkan severity atribut yang berubah dan urgency berdasarkan sisa waktu menuju pelaksanaan kuliah. Pengujian mencakup lima skenario Black Box otomatis menggunakan Playwright pada Chromium dan Firefox, 20 percobaan deteksi perubahan, serta 50 percobaan pengiriman notifikasi. Seluruh skenario fungsional dinyatakan valid. Rata-rata waktu deteksi mencapai 0,15 detik, rata-rata waktu respons notifikasi 2,3 detik, dan seluruh 50 notifikasi berhasil dikirim. Pada lingkungan pengujian, mekanisme terintegrasi memenuhi target operasional serta mendukung penyampaian perubahan jadwal yang lebih cepat, andal, dan sesuai tingkat prioritas.

Copyright © 2026 LPPM - STMIK IKMI Cirebon
This is an open access article under the CC-BY license

Penulis Korespondensi:

Mei Linda Cahyandari
Program Studi Informatika
Fakultas Teknik, Universitas Sarjanawiyata Tamansiswa, Yogyakarta
Indonesia
Email: 2022018002@ustjogja.ac.id

1. Pendahuluan

Perguruan tinggi menggunakan sistem informasi untuk mengelola mata kuliah, dosen, kelas mahasiswa, ruang, hari, dan waktu perkuliahan. Penyusunan jadwal merupakan masalah optimasi karena sistem harus memenuhi *hard constraints* dan *soft constraints* secara bersamaan agar tidak terjadi konflik sumber daya akademik [1]. Sistem yang menjadi objek penelitian telah menggunakan *Genetic Algorithm* untuk membentuk jadwal, sehingga penelitian ini tidak mengubah proses seleksi, crossover, mutasi, fungsi fitness, atau parameter algoritma tersebut.

Penelitian penjadwalan sebelumnya dominan berfokus pada fase pembentukan jadwal. Amrulloh dan Sela mengombinasikan *Genetic Algorithm* dan pencarian tabu untuk menghasilkan jadwal bebas bentrok [2]. Romaguera et al. mengembangkan sistem berbasis web dengan *enhanced genetic algorithm* untuk memenuhi batasan penjadwalan dan meningkatkan efisiensi ruang [3]. Kedua pendekatan tersebut memperkuat proses hulu, tetapi belum mengelola perubahan operasional setelah jadwal dipublikasikan.

Masalah muncul ketika dosen berhalangan, ruang digunakan untuk kegiatan lain, atau waktu perkuliahan perlu disesuaikan. Pada sistem awal, perubahan hari, jam, atau ruang hanya disimpan dalam basis data dan ditampilkan kembali pada *website*. Mahasiswa dan dosen harus melakukan pemeriksaan sendiri. Pola *pull system* ini berisiko menunda penerimaan informasi, terutama ketika perubahan terjadi mendekati waktu kuliah.

Studi notifikasi akademik telah menggunakan WhatsApp Engine, aplikasi Android, dan integrasi *monitoring* absensi untuk mempercepat distribusi pesan [4]-[6]. Namun, pengiriman pesan pada studi tersebut belum menggabungkan deteksi otomatis terhadap perubahan entitas jadwal dengan klasifikasi tingkat prioritas. Akibatnya, perubahan rutin dan perubahan mendesak berpotensi diperlakukan dengan intensitas yang sama.

Pada domain infrastruktur teknologi informasi, *event-based monitoring* dapat menangkap perubahan keadaan secara berkelanjutan [7]. Kajian alert prioritisation menunjukkan bahwa alert perlu diurutkan menurut dampak dan kebutuhan respons [8]. *Smart alerting* juga digunakan untuk menekan *alert fatigue* dengan menyaring pemberitahuan yang kurang penting [9], [11]. Akan tetapi, pendekatan tersebut banyak diterapkan pada insiden teknologi informasi, cloud, dan keamanan siber, bukan pada perubahan jadwal akademik.

Perbandingan tersebut menunjukkan tiga celah penelitian. Pertama, optimasi penjadwalan berhenti pada pembentukan jadwal awal. Kedua, notifikasi

akademik belum dipicu oleh *monitoring* perubahan jadwal secara otomatis. Ketiga, penelitian prioritisasi alert belum mengombinasikan jenis perubahan akademik dan kedekatan waktu pelaksanaan dalam satu skor yang mudah dijelaskan kepada administrator.

Penelitian ini mengisi celah tersebut dengan mengintegrasikan *monitoring service*, *recommendation engine*, *Priority Alert Scoring Algorithm*, *monitoring log*, dan modul notifikasi *website* serta WhatsApp pada sistem penjadwalan yang telah menggunakan *Genetic Algorithm*. Sistem mengubah perubahan data menjadi *change event*, menghitung *severity* dan *urgency*, lalu mengklasifikasikan informasi sebagai Normal, Warning, atau Critical. Integrasi satu alur pasca-penjadwalan tersebut merupakan kebaruan utama penelitian.

Penelitian bertujuan merancang mekanisme deteksi perubahan jadwal secara otomatis, menerapkan scoring prioritas yang transparan, serta mengevaluasi fungsi dan performanya berdasarkan waktu deteksi, waktu respons, kontinuitas layanan, dan keberhasilan pengiriman notifikasi. Evaluasi juga membandingkan karakteristik sistem dengan penelitian sejenis agar posisi kontribusinya dapat dinilai secara lebih jelas.

2. Metode Penelitian

2.1 Desain Penelitian

Penelitian menggunakan metode pengembangan sistem dengan pendekatan modifikasi terhadap aplikasi penjadwalan kuliah berbasis web yang telah tersedia. Pengembangan berfokus pada proses setelah jadwal terbentuk. Modul *Genetic Algorithm* dipertahankan sebagai pembentuk jadwal awal, sedangkan komponen *monitoring*, scoring prioritas, pencatatan log, rekomendasi jadwal pengganti, dan distribusi notifikasi ditambahkan sebagai fungsi pasca-penjadwalan.

Tahapan penelitian meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian. Data kebutuhan diperoleh melalui observasi sistem berjalan, wawancara dengan pihak akademik dan pengguna, dokumentasi sistem, serta studi literatur. Hasil analisis digunakan untuk merumuskan kebutuhan fungsional, arsitektur, aturan deteksi perubahan, hierarki dampak, batas *urgency*, keluaran notifikasi, dan kriteria penerimaan operasional.

2.2 Arsitektur Sistem

Sistem yang dikembangkan terdiri atas lima komponen utama.

1. *Monitoring service* berjalan sebagai *background process* untuk membandingkan keadaan data sebelum dan sesudah

- pembaruan. Perubahan atribut hari, jam, atau ruang dicatat sebagai *change event*.
2. *Recommendation engine* menyediakan alternatif jadwal pengganti. Kandidat hanya ditampilkan apabila memenuhi *hard constraints* berupa ketersediaan dosen, kelas, ruang, kapasitas ruang, dan slot waktu operasional.
 3. *Priority Alert Scoring Algorithm* menghitung tingkat kepentingan perubahan berdasarkan *severity* jenis perubahan dan *urgency* waktu menuju pelaksanaan kuliah.
 4. *Monitoring log* menyimpan kondisi sebelum dan sesudah perubahan, jenis perubahan, waktu kejadian, skor, kategori alert, penerima, dan status pengiriman.
 5. Modul *alerting* membentuk pesan yang memuat jadwal lama, jadwal baru, mata kuliah, waktu, ruang, dan level prioritas, lalu mengirimkannya melalui *website* dan WhatsApp.

Arsitektur menghubungkan aplikasi web, modul penjadwalan, basis data, *monitoring service*, *recommendation engine*, algoritma skoring, *monitoring log*, modul *alerting*, *WhatsApp Gateway*, serta dosen dan mahasiswa sebagai penerima informasi. Rancangan ini mempertahankan fungsi penjadwalan awal dan menambahkan otomasi pada perubahan operasional.

2.3 Mekanisme Priority Alert Scoring

Severity disusun sebagai hierarki ordinal berdasarkan luas dampak perubahan terhadap aktivitas pengguna. Perubahan ruang diberi skor 10 karena hanya mengubah lokasi. Perubahan jam diberi skor 20 karena memengaruhi agenda harian. Perubahan hari diberi skor 30 karena dapat mengubah rencana mingguan. Hierarki ini dirumuskan dari analisis kebutuhan sistem, karakteristik operasional akademik, dan prinsip penggunaan *severity* untuk membedakan dampak insiden [8], [10].

Tabel 1. Bobot *Severity*

Jenis perubahan	Tingkat dampak	Skor
Ruang kuliah	Rendah	10
Jam kuliah	Sedang	20
Hari kuliah	Tinggi	30

Urgency ditentukan dari selisih waktu antara transaksi perubahan dan pelaksanaan kuliah. Perubahan lebih dari tujuh hari diberi skor 10, perubahan tiga sampai tujuh hari diberi skor 20, dan

perubahan kurang dari tiga hari diberi skor 30. Batas tersebut membedakan perubahan yang masih memiliki ruang penyesuaian panjang, menengah, dan singkat.

Tabel 2. Bobot *Urgency*

Selisih waktu	Tingkat urgensi	Skor
Lebih dari 7 hari	Rendah	10
3–7 hari	Sedang	20
Kurang dari 3 hari	Tinggi	30

Nilai prioritas dihitung menggunakan persamaan:

$$PS = S + U$$

PS merupakan *priority score*, S merupakan skor *severity*, dan U merupakan skor *urgency*. Hasil penjumlahan diklasifikasikan menjadi tiga kategori sebagaimana ditunjukkan pada Tabel 3.

Tabel 3. Klasifikasi *Priority Score*

Priority score	Kategori	Mekanisme penyampaian
20–30	Normal	Notifikasi terjadwal
40–50	Warning	Antrean prioritas menengah
60	Critical	Pengiriman instan

Angka 10, 20, dan 30 berfungsi sebagai label ordinal berjarak sama agar kombinasi dampak dan waktu dapat dihitung secara transparan. Angka tersebut bukan koefisien probabilitas. Dasar kategorinya berasal dari analisis kebutuhan melalui observasi, wawancara, dokumentasi, serta literatur *severity* dan alert prioritisation [8]-[11]. Validasi pada penelitian ini terbatas pada kesesuaian keluaran sistem terhadap aturan yang telah ditetapkan melalui *Black Box Testing*. Validasi isi oleh panel pakar lintas institusi belum dilakukan dan dicatat sebagai keterbatasan penelitian.

2.4 Pengujian Sistem

Pengujian fungsional menggunakan *Black Box Testing* otomatis melalui *Playwright* pada *Chromium* dan *Firefox*. Lima skenario mencakup deteksi perubahan, perhitungan *priority score*, penolakan jadwal bentrok, pemilihan rekomendasi, dan pengiriman informasi. Keluaran aktual dibandingkan dengan keluaran yang diharapkan untuk menentukan status valid atau tidak valid.

Pengujian performa mencakup 20 percobaan perubahan jadwal untuk mengukur waktu deteksi dan pengamatan waktu respons pada kondisi jaringan pengujian. Keandalan pengiriman diuji melalui 50 pesan. Target operasional ditetapkan sebesar waktu deteksi tidak lebih dari 2 detik, waktu respons tidak lebih dari 5 detik, keberhasilan pengiriman minimal 95%, dan *monitoring service* berjalan kontinu. Target tersebut digunakan sebagai kriteria penerimaan internal yang selaras dengan konsep time behaviour dan performance efficiency ISO/IEC 25010, bukan sebagai standar universal untuk seluruh sistem [14].

$$KN = \frac{N_b}{N_t} \times 100\%$$

Keberhasilan notifikasi dihitung dengan $KN = (N_b/N_t) \times 100\%$, dengan N_b sebagai jumlah pesan berhasil dan N_t sebagai jumlah pesan yang diuji. Ketidakpastian proporsi keberhasilan dilaporkan

menggunakan interval kepercayaan *Wilson* 95%. Karena data waktu yang tersedia berupa nilai agregat, analisis waktu dilakukan secara deskriptif terhadap target operasional. Evaluasi komparatif dilakukan pada fungsi sistem, bukan pada angka latensi antarpelelitian, karena lingkungan, infrastruktur, dan metrik penelitian terdahulu berbeda.

3. Hasil dan Pembahasan

3.1 Implementasi Sistem

Implementasi menghasilkan sistem SiJadu yang dilengkapi pengajuan jadwal pengganti, pemeriksaan bentrok, rekomendasi ruang dan waktu, *monitoring* perubahan, klasifikasi prioritas, pencatatan log, serta distribusi notifikasi. Ketika dosen mengajukan perubahan, sistem memeriksa ketersediaan dosen, kelas, ruang, kapasitas, dan waktu. Jadwal yang memenuhi seluruh *hard constraints* dapat disimpan, sedangkan konflik memicu *recommendation engine* untuk menampilkan kandidat alternatif.

Setelah jadwal disimpan, *monitoring service* membandingkan data lama dan data baru. Perbedaan atribut diproses sebagai *change event*. Algoritma menentukan *severity* berdasarkan atribut yang berubah dan *urgency* berdasarkan waktu menuju pelaksanaan. Total skor menghasilkan kategori Normal, Warning, atau Critical. Modul *alerting* kemudian membentuk pesan dan mendistribusikannya melalui *website* serta WhatsApp.

Integrasi tersebut mengubah proses penyampaian informasi dari *pull system* menjadi *push system*. Pengguna tetap dapat memeriksa *website*, tetapi sistem juga mengirimkan informasi secara proaktif segera setelah perubahan terdeteksi dan diklasifikasikan.

3.2 Hasil Pengujian Fungsional

Lima skenario fungsional dijalankan pada *Chromium* dan *Firefox*. Seluruh skenario menghasilkan keluaran yang sesuai dengan kebutuhan sistem.

Tabel 4. Hasil Pengujian Fungsional

No.	Skenario	Hasil aktual	Status
1	Perubahan data jadwal	Perubahan tercatat sebagai change event	Valid
2	Perhitungan priority score	Kategori sesuai aturan severity dan urgency	Valid
3	Pengajuan ruang atau waktu bentrok	Kandidat ditolak dan alternatif ditampilkan	Valid
4	Pemilihan rekomendasi	Slot waktu dan ruang tersedia dapat dipilih	Valid
5	Pengiriman informasi	Pesan memuat jadwal lama, jadwal baru, dan level alert	Valid

Hasil pengujian menunjukkan bahwa *monitoring service* mengenali transaksi perubahan dan mencatatnya ke *monitoring log*. Algoritma skoring menghasilkan kategori sesuai kombinasi nilai yang ditanamkan. *Recommendation engine* menolak kandidat yang melanggar *hard constraints*, sedangkan modul *alerting* membentuk pesan dengan informasi sebelum dan sesudah perubahan.

Status valid pada seluruh skenario membuktikan konsistensi logika implementasi pada dua peramban

yang diuji. Hasil ini memvalidasi fungsi aturan, tetapi tidak dapat dipakai sebagai bukti bahwa bobot 10, 20, dan 30 telah memiliki validitas isi untuk seluruh institusi.

3.3 Hasil Pengujian Performa

Pengujian performa menggunakan 20 percobaan perubahan jadwal dan 50 percobaan pengiriman notifikasi. Hasil aktual dibandingkan dengan target operasional penelitian.

Tabel 5. Hasil Pengujian Performa

Parameter	Target	Hasil aktual	Keterangan
Waktu deteksi, n = 20	<= 2 detik	0,15 detik	Memenuhi
Waktu respons	<= 5 detik	2,3 detik	Memenuhi
Keberhasilan, n = 50	>= 95%	100%	Memenuhi
Kontinuitas monitoring	Tanpa kegagalan	Stabil selama pengujian	Memenuhi

Rata-rata waktu deteksi sebesar 0,15 detik setara dengan 7,5% dari batas 2 detik atau 92,5% lebih rendah daripada batas operasional. Hasil ini menunjukkan bahwa *monitoring service* mampu mengenali perubahan dengan latensi rendah pada perangkat, basis data, dan beban pengujian yang digunakan.

Rata-rata waktu respons sebesar 2,3 detik setara dengan 46% dari batas 5 detik atau 54% lebih rendah daripada batas operasional. Rangkaian deteksi, skoring, pembentukan pesan, dan permintaan pengiriman dapat diselesaikan tanpa melewati target pada kondisi jaringan pengujian.

Seluruh 50 pesan berhasil dikirim sehingga tingkat keberhasilan mencapai 100%. Interval kepercayaan *Wilson* 95% untuk 50 keberhasilan dari 50 percobaan berada pada kisaran 92,9% sampai

100%. Nilai titik melampaui target 95%, tetapi batas bawah interval menunjukkan bahwa sampel yang lebih besar masih diperlukan untuk mempersempit ketidakpastian estimasi.

$$KN = \frac{50}{50} \times 100\% = 100\%$$

Monitoring service tetap aktif tanpa crash atau downtime selama periode pengamatan. Capaian tersebut berlaku pada lingkungan uji yang digunakan. Keberhasilan di lingkungan produksi masih dapat dipengaruhi beban serentak, koneksi jaringan, validitas nomor, konfigurasi server, dan ketersediaan layanan WhatsApp pihak ketiga.

3.4 Evaluasi Komparatif dan Statistik Deskriptif

Secara fungsional, penelitian [2] dan [3] mengoptimalkan pembentukan jadwal, sedangkan sistem ini melanjutkan proses setelah jadwal dipublikasikan melalui deteksi perubahan, pencatatan event, dan notifikasi. Dengan demikian, kontribusinya tidak menggantikan algoritma penjadwalan, tetapi melengkapi siklus operasional pasca-penjadwalan.

Dibandingkan sistem notifikasi akademik pada [4]-[6], SiJadu tidak hanya menggunakan kanal digital. Pesan dipicu oleh *monitoring service* dan diberi prioritas sebelum dikirim. Dibandingkan penelitian *smart alerting* [8], [9], [11], metode ini menggunakan *rule-based scoring* yang lebih sederhana dan dapat ditelusuri, tetapi belum adaptif serta belum divalidasi oleh panel pakar lintas institusi.

Analisis deskriptif menunjukkan bahwa waktu deteksi dan respons berada di bawah target operasional, sedangkan proporsi keberhasilan memiliki interval *Wilson* 95% sebesar 92,9% sampai 100%. Perbandingan angka latensi dengan penelitian terdahulu tidak dilakukan karena perbedaan objek, arsitektur, jaringan, dan definisi metrik dapat menghasilkan kesimpulan yang menyedatkan.

3.5 Pembahasan

Kombinasi *severity* dan *urgency* memungkinkan sistem membedakan perubahan yang berdampak rendah dan masih jauh dari waktu kuliah dari perubahan yang berdampak tinggi serta mendesak. Klasifikasi ini mendukung prinsip prioritas untuk mengurangi risiko *alert fatigue* [8], [9], [11]. Keunggulan *rule-based scoring* terletak pada interpretabilitas karena administrator dapat menelusuri alasan suatu perubahan dikategorikan Normal, Warning, atau Critical.

Hasil fungsional dan performa menunjukkan bahwa integrasi komponen baru berjalan sesuai rancangan dan memenuhi kriteria penerimaan internal. Temuan ini mendukung kelayakan teknis mekanisme pada lingkungan uji. Namun, penelitian belum mengukur secara langsung penurunan keterlambatan mahasiswa, pengurangan beban kerja admin, atau peningkatan kepuasan pengguna, sehingga manfaat tersebut diposisikan sebagai implikasi potensial, bukan efek yang telah diuji.

Keterbatasan penelitian meliputi 20 percobaan deteksi, 50 pengiriman pesan, pengujian pada lingkungan terbatas, belum adanya uji beban multi-pengguna, ketergantungan pada *WhatsApp Gateway*, dan belum adanya validasi pakar terhadap bobot serta ambang prioritas. Sistem juga belum memantau kesehatan server, CPU, memori, atau gangguan jaringan.

Pengujian berikutnya perlu melibatkan volume transaksi dan pengguna yang lebih besar, beberapa

kondisi jaringan, analisis distribusi waktu per percobaan, serta validasi isi menggunakan panel akademik dan teknik informasi. Tahap tersebut diperlukan untuk menilai generalisasi performa dan menyesuaikan bobot dengan karakteristik institusi.

4. Kesimpulan

Penelitian berhasil mengintegrasikan *monitoring service*, *recommendation engine*, *Priority Alert Scoring Algorithm*, *monitoring log*, dan notifikasi *website* serta *WhatsApp* ke dalam sistem penjadwalan berbasis web yang telah menggunakan *Genetic Algorithm*. Integrasi ini membentuk satu alur pasca-penjadwalan mulai dari deteksi perubahan hari, jam, atau ruang hingga klasifikasi dan pengiriman informasi kepada pengguna terdampak.

Seluruh lima skenario *Black Box Testing* dinyatakan valid. Pada 20 percobaan, rata-rata waktu deteksi mencapai 0,15 detik. Rata-rata waktu respons mencapai 2,3 detik. Seluruh 50 notifikasi berhasil dikirim, dengan interval kepercayaan *Wilson* 95% sebesar 92,9% sampai 100%, dan *monitoring service* berjalan stabil selama pengamatan. Hasil tersebut memenuhi target operasional pada lingkungan pengujian, tetapi belum mewakili uji produksi berskala besar.

Secara praktis, sistem dapat membantu pengelola akademik mengurangi langkah penyampaian perubahan secara manual, menyediakan *monitoring log* sebagai jejak audit, dan menempatkan informasi mendesak pada prioritas lebih tinggi. Dosen dan mahasiswa juga memperoleh informasi tanpa harus terus memeriksa *website*. Pengembangan selanjutnya perlu mencakup validasi pakar atas bobot, pengujian multi-pengguna, notifikasi multisaluran, *monitoring infrastruktur*, dan optimasi *recommendation engine*.

Daftar Pustaka

- [1] E. Y. Sari, D. Y. Heriyani, and T. Rahmawati, "Pemodelan sistem optimasi penjadwalan matakuliah dengan algoritma genetika," *TEKNIMEDIA: Teknologi Informasi dan Multimedia*, vol. 1, no. 1, pp. 70–78, 2023.
- [2] A. Amrulloh and E. I. Sela, "Optimasi proses penjadwalan mata kuliah menggunakan algoritma genetika dan pencarian tabu," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 3, pp. 157–166, 2021, doi: 10.14710/jtsiskom.2021.14137.
- [3] D. Romaguera, J. Plender-Nabas, J. Matias, and L. Austero, "Development of a web-based course timetabling system based on an enhanced genetic algorithm," *Procedia Computer Science*, vol. 234, pp. 1714–1721, 2024, doi: 10.1016/j.procs.2024.03.177.
- [4] M. A. A. Widya and P. Wahono, "Implementasi sistem notifikasi akademik berbasis *WhatsApp*

- Engine di SMK NU Al-Hidayah Ngimbang,” COMPUTECH: Jurnal Ilmiah Teknologi Informasi dan Komunikasi, vol. 2, no. 1, pp. 1–8, 2022.
- [5] A. D. Jubaedi, S. Dwiyatno, E. Krisnaningsih, Solihin, A. Shafitri, and A. Sutiawan, “Sistem informasi *monitoring* kegiatan absensi siswa dengan notifikasi WhatsApp,” JSiI (Jurnal Sistem Informasi), vol. 10, no. 2, pp. 109–115, 2023, doi: 10.30656/jsii.v10i2.6630.
- [6] R. Amalia, R. Sukmana, and M. Nurcahya, “Pengembangan aplikasi notifikasi informasi akademik *real-time* berbasis Android,” Jurnal Teknologi Edukasi Elektronik, vol. 6, no. 2, pp. 112–120, 2022.
- [7] V. R. Kebande, N. M. Karie, and H. S. Venter, “Adding event-based capability to *real-time* distributed systems *monitoring*,” IEEE Access, vol. 9, pp. 10543–10558, 2021, doi: 10.1109/ACCESS.2021.3051294.
- [8] F. Jalalvand, M. Baruwat Chhetri, S. Nepal, and C. Paris, “Alert prioritisation in security operations centres: A systematic survey on criteria and methods,” ACM Computing Surveys, vol. 57, no. 2, Art. no. 42, pp. 1–36, 2025, doi: 10.1145/3695462.
- [9] F. Voutsas, J. Violos, and A. Leivadreas, “Mitigating *alert fatigue* in cloud *monitoring* systems: A machine learning perspective,” Computer Networks, vol. 250, Art. no. 110543, 2024, doi: 10.1016/j.comnet.2024.110543.
- [10] S. Ahmed, M. Singh, B. Doherty, E. Ramlan, K. Harkin, M. Bucholc, and D. Coyle, “An empirical analysis of state-of-art classification models in an IT incident *severity* prediction framework,” Applied Sciences, vol. 13, no. 6, Art. no. 3843, 2023, doi: 10.3390/app13063843.
- [11] X. Wang, X. Yang, X. Liang, X. Zhang, W. Zhang, and X. Gong, “Combating *alert fatigue* with AlertPro: Context-aware alert prioritization using reinforcement learning for multi-step attack detection,” Computers & Security, vol. 137, Art. no. 103583, 2024, doi: 10.1016/j.cose.2023.103583.
- [12] D. A. Puspita, N. Azise, and A. Lutfi, “Sistem informasi pelayanan masyarakat di Kecamatan Jangkar berbasis web dan via *WhatsApp gateway*,” G-Tech: Jurnal Teknologi Terapan, vol. 7, no. 3, pp. 797–806, 2023, doi: 10.33379/gtech.v7i3.2680.
- [13] A. Rahma, F. Indriyani, and T. A. A. Sandi, “Perancangan dan implementasi *monitoring* perangkat server menggunakan Zabbix pada PT. Rizki Tujuh Belas Kelola,” Jurnal INSAN: Journal of Information System Management Innovation, vol. 3, no. 2, pp. 85–95, 2023, doi: 10.31294/jinsan.v3i2.3009.
- [14] ISO/IEC, “ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models,” International Organization for Standardization, Geneva, Switzerland, 2011.