

Penjadwalan dan Pelaporan Menggunakan *Dynamic Priority Scheduling* dan *Geolocation* untuk Keamanan Lingkungan

Scheduling and Reporting using Dynamic Priority Scheduling and Geolocation for Environmental Security

Faiz Hilmawan Masyfa¹, Dany Primanita Kartikasari², Tibyani³

^{1,2,3}Teknologi Informasi, Universitas Brawijaya

E-mail: ¹faizhm@student.ub.ac.id, ²dany.jalin@ub.ac.id, ³tibyani@ub.ac.id

Abstrak

Angka kriminalitas di Indonesia yang meningkat menimbulkan inisiatif dari warga untuk meningkatkan keamanan lingkungan tempat tinggal melalui ronda. Penjadwalan yang masih manual menjadi salah satu permasalahan kegiatan ronda. Belum adanya portal pelaporan tindak kejahatan yang sudah terjadi maupun laporan daerah yang berpotensi tindak kejahatan juga menjadi permasalahan. Hal ini menimbulkan tidak adanya rekam jejak tindak kejahatan yang telah terjadi serta menimbulkan kerentanan keamanan lingkungan. Sehingga tujuan penelitian ini adalah mengembangkan sistem informasi penjadwalan ronda dan pelaporan keamanan lingkungan. Penjadwalan pada sistem ini dijalankan otomatis dengan mengimplementasikan algoritma *dynamic priority scheduling* serta dilengkapi reminder jadwal yang terintegrasi dengan telegram. Pelaporan yang berbasis lokasi dikembangkan menggunakan teknologi *geolocation*. Hasil pengujian sistem telah menunjukkan penjadwalan memiliki akurasi perbandingan sebesar 99% dan akurasi optimasi 97.04% yang dapat diraih dengan waktu eksekusi 24.72 detik pada data dengan persebaran yang cukup merata. Hal ini menunjukkan penjadwalan yang otomatis memiliki ketepatan penjadwalan dan efisiensi waktu yang lebih baik dari penjadwalan manual. Sistem pelaporan mampu mengidentifikasi lokasi pengguna dengan toleransi akurasi 3 meter. Validasi akhir dengan SUS menunjukkan bahwa sistem sudah dapat diterima oleh pengguna. Metode yang diajukan berhasil membuat proses penjadwalan dan pelaporan menjadi lebih mudah dan dinamis.

Kata kunci: *Dynamic priority scheduling, geolocation, keamanan lingkungan, penjadwalan, pelaporan*

Abstract

The increasing crime rate in Indonesia has led initiatives from residents to improve the security of their living environment through patrols. Manual scheduling is one of the patrol problems. The absence of a portal for reporting crimes that occurred or reporting areas that have the crimes potential is also a problem. The purpose of this research is to develop a patrol scheduling and environmental security reporting. Scheduling system run automatically by implementing a dynamic priority scheduling algorithm with addition reminder that is integrated with Telegram. Location-based reporting is developed using geolocation technology. The test results have shown that scheduling has a comparison accuracy of 99% and an optimization accuracy of 97.04% with an execution time of 24.72 seconds on data with a fairly even distribution. This shows that automatic scheduling has better scheduling accuracy and faster than manual scheduling. The reporting system can identify the user's location with an accuracy tolerance of 3 meters. SUS indicates that the system is acceptable. The proposed method has succeeded in making the scheduling and reporting process easier and more dynamic.

Keywords: *Dynamic priority scheduling, environmental security, geolocation, reporting, scheduling*

1. PENDAHULUAN

Angka kriminalitas di Indonesia menunjukkan peningkatan pada masa pandemi dan setelahnya. Peningkatan angka kriminalitas ini disinyalir terjadi karena dampak dari masa pandemi yang mempengaruhi berbagai faktor kehidupan seperti ekonomi dan sosial [1]. Pembebasan narapidana melalui kebijakan asimilasi juga membuat keresahan di tengah masyarakat, hal ini disebabkan terdapat kasus-kasus kejahatan yang dilakukan oleh beberapa narapidana yang tidak memiliki rasa jera [2]. Kasus kriminalitas yang banyak terjadi adalah pencurian dan angka kriminalitas meningkat sebesar 19.72% dari masa sebelum pandemi [3]. Peningkatan jumlah tindak kriminal pada masa pandemi terjadi pada minggu ke 19 dan ke 20 yang meningkat menjadi 3481 dan 3726 kasus [4].

Pada lingkup kehidupan bermasyarakat sudah menjadi kewajiban setiap individu untuk mewujudkan kondisi yang aman dan tertib. Dalam upaya pencegahan terjadinya tindak kriminal, ronda malam dilakukan secara bergilir setiap hari sesuai jadwal yang sudah ditentukan. Dengan dorongan sosial sebagai bentuk pengabdian, para warga yang tergolong laki-laki dewasa dengan rentang usia 17-50 tahun secara sukarela mengikuti ronda malam sesuai jadwal masing-masing.

Namun keberanekaragaman jenis pekerjaan menjadi faktor terjadinya beberapa kendala pada berlangsungnya ronda, seperti petani yang tidak dapat hadir di ronda malam ketika masa panen tiba [5]. Selain faktor beragamnya pekerjaan, faktor kesadaran dan kemauan berperan penting sebagai dorongan untuk melakukan ronda [5]. Kesadaran dan kemauan memungkinkan untuk dapat diwujudkan dengan pemilihan jadwal ronda yang sesuai dengan kemauannya sendiri. Ritme kehidupan yang dinamis khususnya pada masyarakat perkotaan juga dapat menyebabkan tingginya potensi dilakukannya pergantian jadwal dari hari yang sudah ditetapkan. Sehingga dibutuhkan jadwal yang dapat diubah secara dinamis menyesuaikan kebutuhan masyarakat. Sehingga pembuatan jadwal ronda yang fleksibel diupayakan untuk mendorong masyarakat lebih tertib dalam mengikuti ronda sesuai jadwal.

Kemitraan masyarakat dan kepolisian juga menjadi penting sebagai bentuk monitoring terhadap keamanan wilayah [6]. Tidak adanya portal pelaporan terkait tindak kejahatan yang sudah terjadi menimbulkan tidak adanya rekam jejak terkait tindak kejahatan tersebut. Serta minimnya informasi terkait daerah yang rentan akan tindak kejahatan juga menjadi permasalahan yang harus diselesaikan dalam upaya memitigasi resiko tindak kejahatan. Maka dari itu perlu dikembangkannya sebuah sistem untuk mendorong tindak pelaporan yang dilakukan. Dengan bentuk partisipasi warga melakukan pemetaan lokasi yang beresiko terjadi kejahatan. Maupun melaporkan tindak kejahatan yang telah terjadi dengan lokasi sebagai rekam jejak kejadian tindak kejahatan. Hal ini berhubungan pernyataan pada penelitian [7] yang mengatakan bahwa untuk menjawab permasalahan pada pelaporan tindak kejahatan diperlukan sebuah peran teknologi sebagai bentuk mediasi antara masyarakat dan kepolisian.

Berkaitan dengan permasalahan ini penelitian terkait pembuatan sistem informasi siskamling telah dilakukan dan menjadi sebuah solusi dari permasalahan pada kegiatan siskamling [8]. Namun pada penelitian tersebut masih terdapat beberapa permasalahan yang masih dapat dikembangkan lebih lanjut seperti pada proses penjadwalan yang belum otomatis. Penggunaan *sms gateway* untuk tukar jadwal dan presensi mulai kurang relevan dengan perkembangan teknologi saat ini. Serta sistem pelaporan yang masih dapat dikembangkan lebih baik lagi dengan informasi poin *geolocation* agar data pelaporan lebih lengkap dan akurat. Penelitian terkait pelaporan tindak yang menggunakan *geolocation* sebagai validasi sudah dilakukan sebelumnya [9,10]. Pada penelitian oleh Permadi, Arwani, dan Akbar [9] masih belum dipertimbangan terkait akurasi dari lokasi yang dihasilkan, serta tidak terdapat pemetaan lokasi dengan rekomendasi terhadap pelaporan tindak kejahatan yang sudah dilakukan.

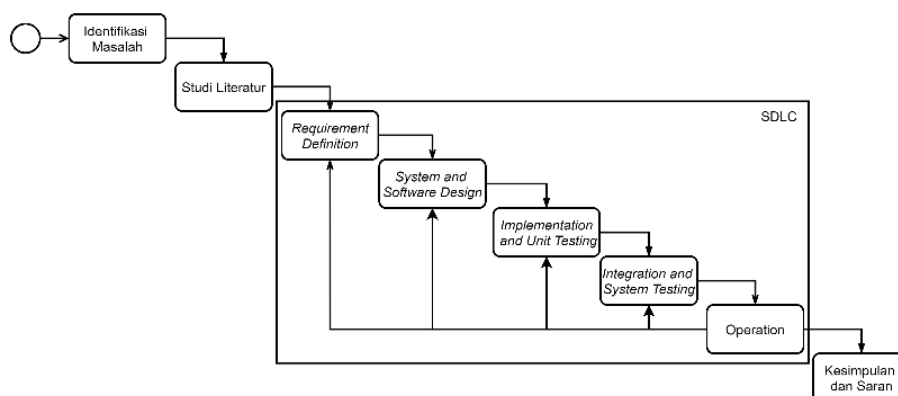
Penggunaan algoritma *dynamic priority scheduling* pada sistem penjadwalan yang dikembangkan didasari pada sistem *queue* atau antrian pada algoritma tersebut. Antrian ini akan digunakan untuk membuat *rolling* jadwal setiap minggunya. Penjadwalan menggunakan algoritma tersebut didasarkan pada nilai prioritas yang dimiliki setiap proses [11,12]. Dari

penelitian sebelumnya yang menerapkan algoritma ini untuk penjadwalan *co-working space* [11], dan manajemen sumber daya [12] maka akan dijadikan sebagai rujukan utama dalam membangun sistem penjadwalan yang disesuaikan dengan studi kasus pada penelitian ini. Proses penjadwalan ini juga akan dilengkapi dengan fungsional *reminder* yang dibuat melalui integrasi dengan *telegram*. Sedangkan penggunaan *geolocation* diimplementasikan menggunakan *HTML5 Geolocation API* sebagai pengambil titik lokasi *gps* dan *Openstreetmap* sebagai penyedia *map layer*. *HTML5 Geolocation API* sendiri merupakan layanan yang disediakan *HTML5* untuk mengambil lokasi, penggunaan layanan ini mempertimbangkan dukungan *cross-platform mobile web app* dan *location-based service*. Sedangkan *Openstreetmap* merupakan penyedia layanan server basis data geospasial yang bersifat *open source* dan dapat diakses dengan *openstreetmap API* [13]. Penerapan *geolocation* pada pelaporan yang dilakukan oleh peneliti terdahulu untuk pemetaan absensi [14] dijadikan rujukan untuk implementasi *geolocation* pada penelitian ini.

Didasari oleh permasalahan sebelumnya, penelitian ini mengajukan solusi berupa pengembangan sistem yang memiliki fungsional utama untuk membantu penjadwalan piket ronda malam dan pelaporan keamanan lingkungan. Implementasi proses penjadwalan dilakukan dengan mempertimbangkan akurasi dan efisiensi waktu menggunakan algoritma *dynamic priority scheduling*. Sedangkan pelaporan yang berbasis lokasi akan mempertimbangkan akurasi poin lokasi yang dihasilkan sistem. Diharapkan pengembangan sistem ini dapat menjadi solusi permasalahan penjadwalan dan pelaporan.

1. METODE PENELITIAN

Metodologi yang digunakan pada penelitian ini bertipe implementatif dengan alur digambarkan pada Gambar 1. Tahapan metodologi penelitian diawali dengan identifikasi masalah, dilanjutkan dengan studi literatur. Metode pengumpulan data digunakan dengan teknik wawancara kepada narasumber tertentu sebagai data primer dan studi literatur sebagai data sekunder. Sedangkan metode pengembangan perangkat lunak yang digunakan berjenis Waterfall dari Sommerville [15]. Alur yang dilakukan pada metode pengembangan adalah *requirement definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation* beserta evaluasinya. Pemilihan metode ini mempertimbangkan *requirement* yang dibutuhkan sudah jelas dan tidak diperlukan untuk melakukan evaluasi di tengah pengembangan aplikasi dilakukan. Hal ini juga mempertimbangkan pengembangan sistem yang dibuat memiliki interaksi pengguna yang sederhana dan lebih berfokus pada pengembangan di sisi *backend*. Setelah selesai tahap dari metode pengembangan, ditutup dengan kesimpulan dan saran untuk penelitian ini.



Gambar 1 Metodologi Penelitian

Identifikasi masalah akan dilakukan dengan cara melakukan wawancara kepada tiga narasumber yaitu ketua RW sebagai pengelola kegiatan ronda dan warga sebagai pelaksana

kegiatan ronda. Proses wawancara ini dilakukan pada Kel. Lowokwaru, Kec. Lowokwaru, Kota Malang dan dengan topik seputar kegiatan ronda. Proses identifikasi masalah akan didukung dengan studi literatur yang dilakukan sebagai pengambilan data sekunder yang menunjang penelitian ini. Hal ini dilakukan dengan cara mengumpulkan data dan informasi dengan cara membaca jurnal dan buku referensi yang dapat dijadikan sebagai dasar penelitian.

Requirement definition dilakukan dengan mendeskripsikan daftar kebutuhan pengguna dari wawancara dan studi literatur yang dilakukan. Daftar kebutuhan fungsional dan non-fungsional dari masing masing aktor akan didefinisikan dalam sebuah *Software Requirement Specifications (SRS)*. Selanjutnya dilakukan transformasi menjadi sebuah model yang merepresentasikan interaksi antara pengguna terhadap sistem informasi yang dikembangkan. Pemodelan tersebut diwakili dengan pembuatan sebuah use case diagram serta use case *skenario* untuk setiap proses.

System and software desgin digunakan untuk memvisualisasikan rancangan dari kebutuhan-kebutuhan yang dibuat. Terdapat rancangan arsitektur sistem yang menggambarkan rancangan dari sistem secara keseluruhan. Serta rancangan tiga algoritma yang dibangun pada sistem.

Implementation adalah tahapan untuk mengubah desain yang sudah dibuat menjadi barisan kode pemrograman. Kode program disusun dengan menggunakan arsitektur *model, view, controller (MVC)* dengan *framework Laravel*. Subecz [16] merekomendasikan penggunaan *Laravel* untuk mengembangkan web dengan cepat dan modern karena didukung dengan karakteristik utamanya mulai dari routing yang sederhana dan cepat, arsitektur *MVC, blade template*, interaksi database yang lebih mudah dengan adanya *eloquent* yang memungkinkan untuk tidak perlu melakukan *sql query*, dan keamanan aplikasi. Setelah implementasi dilakukan maka dilanjutkan dengan proses *unit testing*. *Unit testing* dilakukan dengan melakukan *test* pada sebuah *method* atau *function* yang terpisah dengan yang lain (*terisolasi*)[17]. Pengujian dengan unit testing dilakukan dengan cara pembuatan *test case* terlebih dahulu. Pembuatan *test case* ini diawali dengan penentuan nilai input dan nilai output yang diekspektasikan. Ketika sudah ditentukan maka pada *class* tes yang dibuat akan dilakukan dengan eksekusi pengujian untuk melihat apakah hasil yang diberikan sesuai dengan ekspektasi atau tidak.

Integration dilakukan dengan integrasi pada dua sistem eksternal guna mendukung berjalannya sistem. Integrasi pertama dilakukan dengan *Telegram Bot API* sebagai penunjang fitur *reminder*. Sedangkan pada fitur pelaporan dan presensi akan diintegrasikan dengan sistem penyedia peta digital. Teknologi *geolocation* berbasis *HTML5 Geolocation API* juga digunakan untuk pengambilan lokasi. Untuk memvalidasi sistem dapat berjalan dengan baik setelah dilakukan integrasi maka dilakukan *system testing*. *System testing* terfokus pada pengujian ketiga algoritma yang dikembangkan beserta integrasinya dengan layanan yang digunakan. *Test case* akan dibuat dengan pendekatan *equivalence partitioning*. *Test case* dibagi menjadi kelas-kelas dengan menyesuaikan tujuan yang ingin dicapai dari fungsionalitas tersebut [18].

Operation diawali dengan melakukan deploy terlebih dahulu sistem yang sudah dibuat kepada sebuah server. Jenis web server yang digunakan adalah *LiteSpeed* dengan mempertimbangkan performa yang lebih baik dari *Apache* [19]. Server ini berjalan menggunakan *CPU* dua core, *RAM* 4 GB, dan tipe penyimpanan *SSD*. Setelah itu masuk pada tahap evaluasi yang berguna untuk memvalidasi pemenuhan kebutuhan pengguna sistem. Evaluasi ini dilakukan dengan menggunakan *System Usability Scale (SUS)* untuk mengukur *usability* dari sistem. Terdapat 10 instrumen pertanyaan *SUS* yang digunakan pada penelitian ini, ditunjukkan pada Tabel 1.

Tabel 1 Instrumen Pertanyaan *System Usability Scale (SUS)*

No	Pertanyaan	Skala
1	Saya berfikir untuk menggunakan sistem ini lagi	1-5
2	Saya merasa sistem ini sangat rumit dan sulit untuk digunakan	1-5
3	Saya rasa sistem ini mudah untuk digunakan	1-5
4	Saya membutuhkan bantuan dari orang lain ataupun teknisi untuk menggunakan sistem ini	1-5

5	Saya merasa fitur-fitur pada sistem ini berjalan dengan semestinya	1-5
6	Saya merasa banyak hal yang berjalan tidak konsisten dan tidak sesuai	1-5
7	Saya merasa orang lain akan mudah memahami sistem ini	1-5
8	Saya merasa sistem ini membingungkan	1-5
9	Saya merasa tidak hambatan dan kesulitan dalam menggunakan sistem ini	1-5
10	Saya perlu waktu untuk membiasakan diri menggunakan sistem ini	1-5

Sedangkan Tabel 2 menunjukkan skala penerimaan berdasarkan *SUS*. Cara perhitungan dari hasil kuesioner yang didapatkan dapat dilakukan dengan cara instrumen pertanyaan nomor ganjil nilainya dikurangi 1, sedangkan instrumen pertanyaan nomor genap, konstanta 5 akan dikurangi nilai dari instrumen pertanyaan nomor genap. Total dari kedua instrument pertanyaan baik genap maupun ganjil akan diakumulasi dan dikalikan dengan 2.5 untuk mendapatkan skor *SUS* setiap partisipan. Untuk mendapatkan skor *SUS* secara keseluruhan dilakukan perhitungan rata-rata keseluruhan. Pada pengujian dan perhitungan *SUS* yang dilakukan cukup rumit namun jumlah sampel yang dibutuhkan untuk pengujian hanya sedikit [20]. Pada proses evaluasi ini ditentukan jumlah sample pengguna sebagai partisipan pada *testing* ini sebanyak 10 orang. Dari 10 orang tersebut sudah mewakili role dari pengguna sistem dan distribusi usia pengguna.

Tabel 2 Skala System Usability Scale (*SUS*)

Kriteria	Rentang Skor	Kategori	Skor
Non-acceptable	0-50	Worst imaginable	20
		Poor	30
		Ok	50
Marginal	51-70	Good	70
		Best imaginable	100
Acceptable	71-100		

2. HASIL DAN PEMBAHASAN

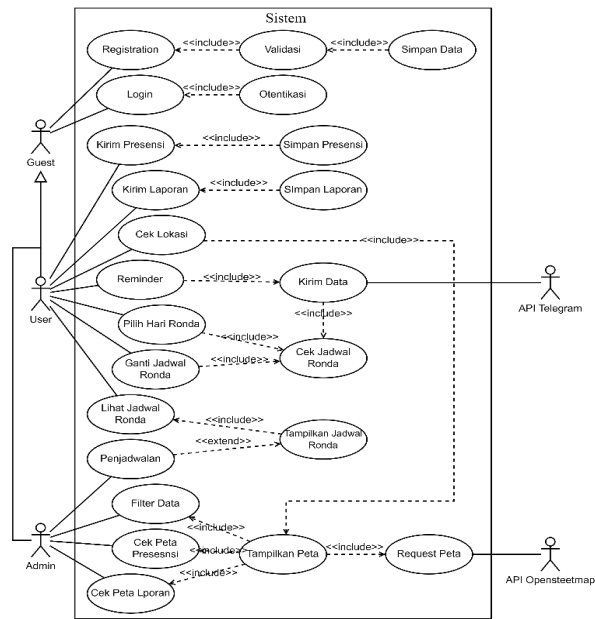
3.1 Pendefinisian Kebutuhan

Terdapat tujuh proses bisnis yang harus ada pada sistem, mencakup registrasi, login, penjadwalan, *reminder*, presensi, pelaporan serta pengelolaan presensi dan laporan. Serta terdapat tiga aktor yaitu *guest*, *admin* dan *user* yang akan berinteraksi dengan sistem. Kebutuhan fungsional dan non-fungsional yang dihasilkan dari setiap aktor disajikan dalam sebuah *Software Requirement Specifications (SRS)* pada Tabel 3.

Tabel 3 *Software Requirement Specifications*

Kode	Software Requirement Statement	Level
Kebutuhan Fungsional		
SRS-SIKL- F-01	Sistem menyediakan layanan registrasi	Normal
SRS-SIKL- F-02	Sistem menyediakan layanan login	Normal
SRS-SIKL- F-03	Sistem menyediakan fitur untuk menampilkan jadwal ronda	Normal
SRS-SIKL- F-04	Sistem menyediakan layanan reminder jadwal ronda	Expected
SRS-SIKL- F-05	Sistem menyediakan fitur presensi menggunakan lokasi	Normal
SRS-SIKL- F-06	Sistem menyediakan fitur pelaporan menggunakan lokasi	Normal
SRS-SIKL- F-07	Sistem menyediakan fitur pemilihan dan pergantian jadwal	Expected
SRS-SIKL- F-08	Sistem menyediakan fasilitas pengelolaan jadwal ronda	Normal
SRS-SIKL- F-09	Sistem menyediakan fitur untuk menampilkan presensi <i>user</i> melalui peta	Normal
SRS-SIKL- F-10	Sistem menyediakan fitur untuk menampilkan laporan dari <i>user</i> melalui peta	Normal
Kebutuhan Non-Fungsional		
SRS-SIKL-NF-01	Sistem memiliki keamanan untuk otentikasi	Normal
SRS-SIKL-NF-02	Sistem bisa mengambil lokasi dari <i>gps</i> dengan akurasi tinggi untuk presensi	Expected
SRS-SIKL-NF-03	Sistem bisa mengambil lokasi dari <i>gps</i> dengan akurasi tinggi untuk pelaporan	Expected
SRS-SIKL-NF-04	Sistem dapat membuat jadwal otomatis secara cepat	Excellent

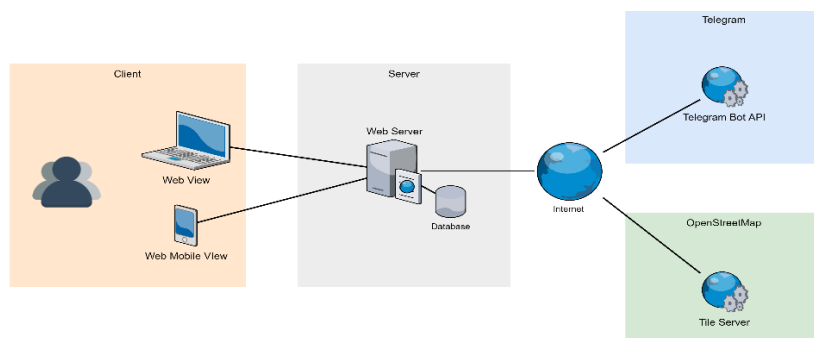
Dari *SRS* tersebut dilakukan pemodelan yang digambarkan dengan *use case diagram* pada Gambar 2. Pada *use case diagram* tersebut terdapat 13 interaksi yang dapat dilakukan oleh ketiga aktor utama. Serta terdapat dua aktor pendukung yaitu pihak ketiga penyedia layanan yang digunakan oleh sistem. Layanan untuk *reminder* pada *channel telegram* dengan bantuan *API Telegram* dan layanan peta digital yang disediakan oleh *OpenStreetMap*.



Gambar 2 Use Case Diagram Sistem

3.2 Perancangan

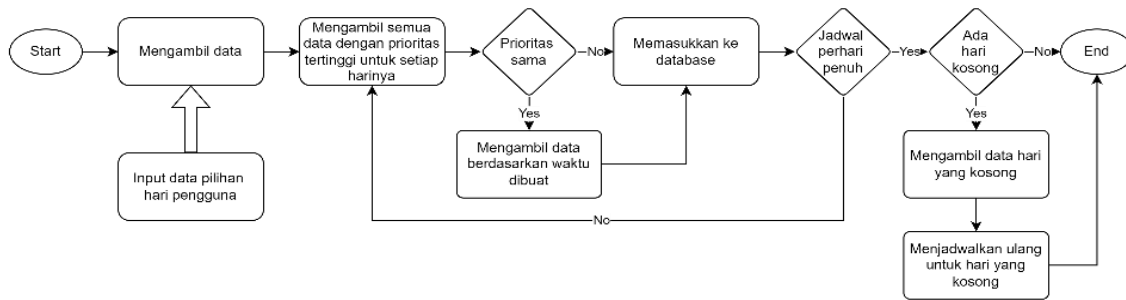
Pada tahap *system and software design*, hasil perancangan yang pertama adalah arsitektur dari sistem yang digambarkan pada Gambar 3. Terdapat empat lingkungan sistem yang saling terhubung. Lingkungan pertama adalah sisi *client* yang memiliki dua aksesibilitas yaitu *web view* yang merupakan halaman web untuk layar desktop, serta terdapat *web mobile view* untuk layar mobile. Sistem yang dikembangkan akan secara dinamis menyesuaikan jenis perangkat yang digunakan oleh pengguna baik dalam bentuk desktop maupun mobile. Lingkungan selanjutnya adalah *server*, yang mencakup *web server* beserta database yang digunakan. *Web server* akan berperan untuk melayani pengguna dengan menerima request maupun pengiriman respon. *Web server* ini akan terhubung dengan dua sistem yaitu *telegram* dan *OpenStreetMap*. Layanan dari *telegram* berupa *telegram bot API* yang berfungsi untuk mengirim pesan dari *web server* menuju aplikasi *telegram*. Sedangkan layanan dari *OpenStreetMap* berupa *tile server* sebagai penyedia data raster atau peta digital yang digunakan pada sistem.



Gambar 3 Arsitektur Sistem

Terdapat tiga algoritma yang digunakan pada penelitian ini. Masing-masing algoritma tersebut adalah algoritma untuk penjadwalan dengan *dynamic priority scheduling*, algoritma untuk penerapan *geolocation*, dan algoritma untuk proses *reminder*. Algoritma penjadwalan pada Gambar 4, diawali dengan pengambilan data. Data ini berupa *input* empat pilihan hari dari *user* yang diurutkan berdasarkan prioritas. Dilanjutkan dengan mengambil data berdasarkan prioritas tertinggi untuk mengisi jadwal di masing-masing ketujuh hari. Ketika terdapat prioritas

yang sama maka akan data yang diambil didasarkan pada waktu dibuat atau diperbarui. Setelah data dimasukkan ke database selanjutnya dilakukan pengecekan terhadap pemenuhan jumlah *user* setiap harinya. Pengecekan ini juga berfungsi untuk membatasi jumlah *user* yang diambil, misalkan terdapat kondisi prioritas hari dan waktu dibuat atau diupdate sama maka akan tetap diambil sesuai batasan maksimal jumlah *user* di hari tersebut. Setelah dicek dan jadwal belum penuh maka akan dilakukan pengambilan data lagi berdasarkan urutan prioritas. Namun jika sudah penuh maka akan dilakukan pengecekan apakah terdapat hari yang belum terisi. Jika terdapat hari yang kosong maka akan menjadwalkan ulang dengan memprioritaskan pengisian hari yang kosong.



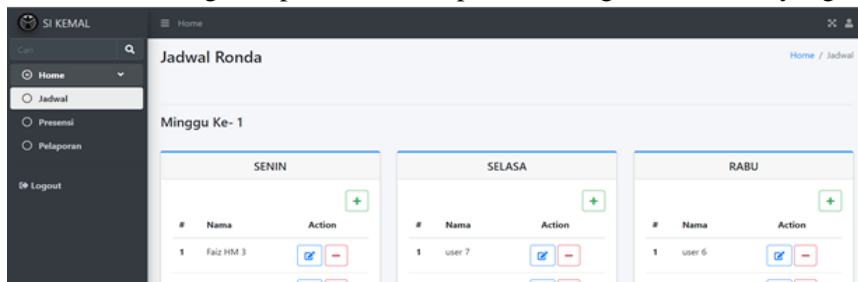
Gambar 4 Alur Algoritma *Dynamic Priority Scheduling*

Pada algoritma *reminder*, ketika sistem *reminder* di-trigger maka proses pertama yang dilakukan adalah mengambil tanggal hari ini. Tanggal hari ini akan digunakan menjadi parameter untuk mengambil semua pengguna yang memiliki jadwal di hari ini. Ketika jadwal kosong maka proses *reminder* tidak dijalankan, namun ketika jadwal di hari ini ada maka akan dikirim *reminder*. Pengecekan ini bertujuan agar tidak mengirim *reminder* tanpa terdapat pengguna, sekalipun proses pembuatan jadwal sudah dipastikan tidak pernah kosong.

Sedangkan Alur pada algoritma untuk penerapan *geolocation* diawali dengan melakukan *request* peta kepada penyedia layanan. Peta dari *openstreetmap* akan ditampilkan pada sebuah halaman. Setelah itu sistem akan mengambil lokasi dari pengguna dengan persetujuan dari pengguna untuk ditampilkan pada peta. Pengambilan lokasi ini dilakukan dengan mempertimbangkan pengaturan akurasi tinggi jika *device* yang digunakan oleh pengguna mendukung. Namun ketika pengguna tidak mengizinkan akses lokasi maka akan terdeteksi *error*. Selain itu pengecekan *error* juga dilakukan apakah pada waktu *time out* serta posisi dapat ditemukan atau tidak.

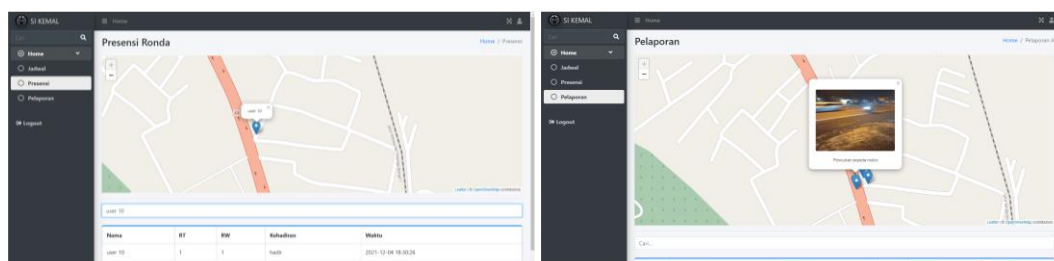
3.3 Implementasi

Tahap implementasi menghasilkan tiga halaman khusus untuk admin dan tiga halaman khusus untuk *user*. *User* dan admin akan login melalui halaman yang sama. Halaman awal untuk admin adalah halaman home yang digunakan untuk menavigasi halaman lain. Halaman pengelolaan jadwal di Gambar 5 terdapat tiga fungsionalitas untuk admin yaitu menampilkan keseluruhan jadwal, menambah jadwal, mengubah jadwal, dan menghapus jadwal. Fungsionalitas untuk menambahkan jadwal secara manual tetap dibuat meskipun penjadwalan dilakukan secara otomatis, agar implementasi tetap sesuai dengan kebutuhan yang ditentukan.



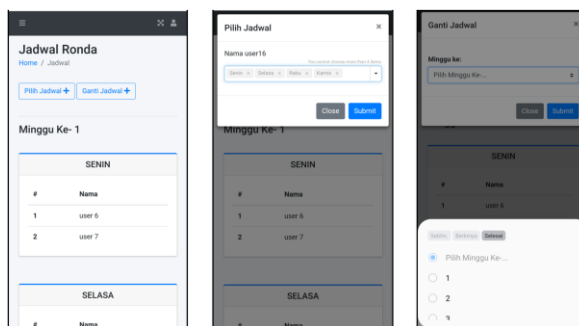
Gambar 5 Halaman Pengelolaan Jadwal

Pada Gambar 6 terdapat halaman untuk pencatatan presensi dan laporan yang dilakukan oleh *user*. Kedua halaman ini dibahas bersamaan karena memiliki kesamaan yaitu pengimplementasian *geolocation*. Halaman presensi berisi peta dengan titik lokasi presensi *user* yang dilengkapi data kehadiran dengan fitur *search* sebagai pelengkap. Tampilan peta dan titik lokasi juga menjadi konten utama pada halaman laporan. Halaman laporan juga akan memuat tabel data laporan dari *user* yang dilengkapi dengan fitur *search* untuk memudahkan pencarian.



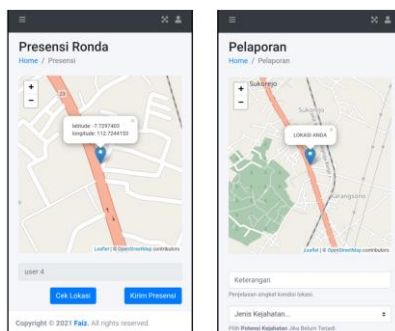
(a) (b)
Gambar 6 (a) Halaman Pencatatan Presensi dan (b) Halaman Laporan

Halaman *home* untuk *user* secara umum sama dengan tampilan *home* untuk *admin*. Hanya saja untuk menyesuaikan penggunaan *mobile device* oleh *user* maka tampilan ini dirancang responsif untuk dapat menyesuaikan jenis perangkat yang digunakan. Sedangkan halaman jadwal untuk *user* ditunjukkan pada Gambar 7. Halaman ini memuat informasi jadwal secara keseluruhan dalam satu bulan. Namun yang ditampilkan hanya hari yang memuat *user* mengakses jadwal. Pada halaman yang sama terdapat kolom untuk melakukan pemilihan hari yang disediakan pada menu *dropdown*. Pemilihan hari diurutkan berdasarkan prioritas, dengan hari berprioritas paling tinggi dipilih terlebih dahulu urut hingga yang prioritas terendah. Serta terdapat kolom untuk mengganti jadwal sesuai minggu yang diinginkan.



(a) (b) (c)
Gambar 7 (a) Halaman Jadwal, (b) Pilih Jadwal dan (c) Ganti Jadwal

Selanjutnya pada Gambar 8 memuat dua tampilan sekaligus, yaitu tampilan presensi dan pelaporan. Kedua halaman ini memiliki kesamaan yaitu mengimplementasikan peta dan pengambilan titik lokasi pengguna. Untuk halaman presensi yang berbasis lokasi akan terdapat pengecekan lokasi *user* dan jika berjarak lebih dari 100 meter dari pos ronda maka akan menampilkan pesan bahwa lokasi terlalu jauh dari pos ronda. Serta pada halaman laporan juga terdapat *panic button* untuk memfasilitasi pelaporan darurat. Ketika *panic button* ditekan maka akan mengirimkan *spam* pesan darurat menuju *channel telegram* dengan berisi keterangan kejadian dan lokasi kejadian terjadi.



Gambar 8 (a) Halaman Presensi dan (b) Halaman Pelaporan

Tahap implementasi dilakukan bersama dengan pengujian unit atau *unit testing*. Pengujian ini dilakukan pada setiap *function* dalam enam *controller class*. Melalui pengujian ini didapatkan bahwa tidak terdapat *failure* untuk setiap *function* yang diuji. Salah satu contoh *test case* yang digunakan pada pengujian unit ini ditunjukkan pada Tabel 4, yang merupakan pengujian unit untuk fungsionalitas jadwal. Semua *function* dapat memberikan *feedback* sesuai yang diharapkan, baik pada fungsionalitas registrasi, login, penjadwalan, *reminder*, presensi dan pelaporan. Sehingga dapat diverifikasi bahwa semua fungsionalitas pada sistem berjalan sesuai.

Tabel 4 *Unit Test* Fungsionalitas Penjadwalan

Judul	Unit Test Jadwal
Tujuan	Memverifikasi fungsional jadwal (memilih jadwal, membuat jadwal otomatis, dan CRUD jadwal)
Target Class	JadwalController
Ekspektasi	showJadwal(): redirect halaman jadwal, kode status (302) showTambahJadwal(): redirect halaman tambah jadwal, kode status (302) cekTanggal(): melakukan cek tanggal, kode status (200) pilihJadwal(): redirect halaman home, kode status (302) buatJadwal(): melakukan pembuatan jadwal, kode status (200) cekJadwal(): melakukan cek kekosongan jadwal, kode status (200) gantiJadwal(): redirect halaman jadwal, kode status (320) tambahJadwal(): redirect halaman jadwal, kode status (320) editJadwal(): redirect halaman jadwal, kode status (320) hapusJadwal(): redirect halaman jadwal, kode status (320)
Hasil	OK (10 test, 10 assertion)
Status	Valid

3.4 Integrasi

Tahap selanjutnya dilakukan pengecekan integrasi dengan sistem penyedia layanan lain. Sistem penyedia layanan lain mencakup *HTML5 geolocation*, *OpenStreetMap*, serta *telegram*. Pengecekan ini dilakukan dengan melakukan *request* layanan pada penyedia dan melihat *log* yang dihasilkan. Data *response* dari setiap *request* yang tercatat di *log* tidak ada yang menunjukkan *error* dan menghasilkan data yang diharapkan. Sehingga dapat dikatakan bahwa semua proses integrasi berhasil dilakukan.

System testing dilakukan ketika keseluruhan sistem telah terintegrasi. Pengujian ini dilakukan pada tiga fungsional utama sistem yaitu algoritma penjadwalan, proses *reminder*, serta pelaporan dan presensi menggunakan *geolocation*. *Test case* pada penjadwalan dibuat dengan membagi data masukan menjadi data dengan pemilihan prioritas tertinggi merata, cukup merata dan tidak merata. Data prioritas yang merata pada Tabel 5 adalah kondisi ketika tidak ada *user* yang memilih hari prioritas tertinggi sama, semua memilih hari yang berbeda.

Tabel 5 Distribusi Prioritas Tertinggi Merata

Senin	Selasa	Rabu	Kamis	Jumat	Sabtu	Minggu
User A	User C	User E	User G	User I	User K	User M
User B	User D	User F	User H	User J	User N	User N

Data cukup merata adalah kondisi terdapat sedikit *user* memilih hari prioritas tertinggi

yang sama. Sedangkan data dengan pilihan yang tidak merata pada Tabel 6 merupakan kondisi banyak *user* yang memilih hari prioritas tertinggi yang sama sehingga saling bertabrakan dan menumpuk dihari tertentu. Penggunaan tiga jenis persebaran data ini didasarkan pada pemilihan prioritas tertinggi yang merata dirasa kecil kemungkinan dapat terjadi. Sehingga untuk menguji apakah sistem tetap dapat menghasilkan optimasi yang baik maka dibuatkan *skenario* uji menggunakan data persebaran pemilihan prioritas tertinggi cukup merata dan tidak merata.

Tabel 6 Distribusi Prioritas Tertinggi Tidak Merata

Senin	Selasa	Rabu	Kamis	Jumat	Sabtu	Minggu
User A	User D	User E	User G	-	User K	User M
User B	-	User F	User H	-	User L	User N
User C	-		User I	-	User J	-

Terdapat dua parameter pengujian pada proses pengujian ini yaitu akurasi dan lama waktu eksekusi. Akurasi dari penjadwalan ini diukur dengan dua teknik yaitu, membandingkan hasil penjadwalan menggunakan algoritma *dynamic priority scheduling* dari sistem dengan penjadwalan dengan perhitungan manual. Teknik kedua adalah dengan menghitung perbandingan optimasi yang dihasilkan oleh sistem dengan optimasi akurasi paling maksimal. Persamaan untuk menghitung akurasi perbandingan hasil sitem dan manual adalah (*jumlah penjadwalan yang tidak sesuai / jumlah keseluruhan jadwal * 100%*). Sedangkan persamaan untuk perhitungan akurasi optimasi adalah (*jumlah prioritas pada satu minggu / maksimal prioritas pada satu minggu * 100%*). Hasil dari pengujian algoritma penjadwalan berdasarkan ketiga *test case* menunjukkan bahwa rata-rata dari lima percobaan menggunakan 14 data dengan distribusi prioritas tertinggi yang merata menghasilkan akurasi perbandingan sebesar 99%, akurasi optimasi 99% dan waktu eksekusi 6.59 detik. Ketika menggunakan data dengan distribusi prioritas tertinggi yang cukup merata menghasilkan akurasi perbandingan sebesar 99%, akurasi optimasi 97.04% dan waktu eksekusi 24.72 detik. Sedangkan percobaan menggunakan data dengan distribusi prioritas tertinggi tidak merata menghasilkan akurasi perbandingan sebesar 77.13%, akurasi optimasi 90.39% dan waktu eksekusi 54.66 detik.

System test pada fungsionalitas *reminder* jadwal ronda mengukur waktu eksekusi pengiriman *reminder* ke *channel telegram* dengan membagi menjadi dua *test case* dengan lima kali percobaan. Pada *case* positif jadwal yang akan dikirim oleh *reminder* akan tersedia sedangkan untuk *case* negatif jadwal tidak tersedia. Pengujian ini bertujuan untuk memastikan bahwa tidak akan ada *reminder* kosong (tanpa daftar *user*) yang terkirim. Pada *case* negatif *reminder* tidak terkirim, rata-rata waktu eksekusi 0,014 detik. Sedangkan untuk *case* positif, *reminder* terkirim dengan daftar *user*, rata-rata waktu eksekusi 2,522 detik.

Terdapat dua *test case* pada pengujian pelaporan dan presensi dengan *geolocation*, yaitu akses *gps* diberikan dan tidak. Sehingga pada *test case* positif juga digunakan untuk melihat poin lokasi yang dihasilkan oleh sistem. Sedangkan *case* negatif menunjukkan sistem dapat menangkap *error* atau tidak. Percobaan *case* negatif, menghasilkan pesan *error*. Percobaan *case* positif, poin lokasi dapat ditampilkan pada peta. Perhitungan jarak menggunakan rumus *haversine* rata-rata dari lima kali pengukuran menghasilkan toleransi sebesar 3 meter.

3.5 Operasi

Tahap operasi menghasilkan sistem yang sudah siap digunakan dengan kondisi sistem yang sudah diunggah pada sebuah server. Setelah sistem siap digunakan maka selanjutnya dilakukan evaluasi terhadap *usability* dari sistem kepada pengguna. Hasil dari perhitungan skor SUS pada Tabel 7 menunjukkan nilai *usability* sistem yang dibangun. Skor akhir SUS sistem ini menunjukkan angka 77.8 yang tergolong pada kriteria *acceptable* dan berkategori *good*.

Tabel 7 Hasil Perhitungan Skor SUS

Partisipan	Usia	Instrumen Pertanyaan										Skor
		1	2	3	4	5	6	7	8	9	10	
Admin	50	4	4	4	3	3	3	3	3	3	2	80
User 1	22	3	2	4	3	4	3	3	3	4	3	80

User 2	45	3	2	3	3	3	3	2	4	1	3	67.5
User 3	21	4	4	4	3	4	4	4	4	4	3	95
User 4	43	3	4	4	3	3	3	4	4	3	2	82.5
User 5	32	3	4	3	2	3	3	4	4	4	3	82.5
User 6	48	3	4	4	2	4	4	3	3	1	2	75
User 7	21	4	1	3	2	3	3	3	3	4	2	70
User 8	43	3	3	3	1	4	4	3	4	4	1	75
User 9	21	3	3	4	3	3	3	3	3	0	3	70
Rata-rata Skor SUS												77.8

3. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang dilakukan dengan mengembangkan sistem informasi keamanan lingkungan, dapat ditarik kesimpulan bahwa penelitian ini sudah dapat menghasilkan sistem dengan proses penjadwalan yang memiliki akurasi baik ketika data yang diolah memiliki distribusi pilihan prioritas tertinggi yang merata. Akurasi perbandingan 99% dan akurasi optimasi 99% diraih ketika data memiliki distribusi pemilihan hari dengan prioritas tertinggi yang merata. Ketika data yang diolah memiliki distribusi cukup merata maka akurasi perbandingan yang dihasilkan adalah 99% dengan akurasi optimasi 97.04%. Sedangkan ketika data yang digunakan tidak merata menghasilkan akurasi perbandingan yang cukup rendah di angka 77.13%, namun akurasi optimasi tetap tergolong baik yaitu 90.39%. Proses penjadwalan pada data yang memiliki distribusi pilihan prioritas hari tertinggi merata, cukup merata dan tidak merata secara berrutan waktu eksekusinya sebesar 6.59 detik, 24.72 detik, dan 54.66 detik. Dibandingkan dengan penjadwalan yang dilakukan secara manual tentunya ini akan lebih cepat.

Poin lokasi berbasis *geolocation* pada fitur pelaporan dan presensi berhasil ditampilkan ketika akses lokasi diberikan. Sedangkan ketika akses lokasi yang tidak diberikan, maka sistem menrim pesan *error*. Hasil perhitungan selisih poin rumus *haversine* rata-rata menunjukkan selisih 3 meter. Selisih jarak yang kecil menunjukkan keakuratan poin lokasi yang dihasilkan.

Validasi akhir dengan evaluasi menggunakan SUS menunjukkan nilai *usability* sistem pada angka 77.8. Nilai tersebut tergolong pada kriteria *acceptable* dan berkategori *good*. Sehingga dapat dikatakan sistem sudah dapat diterima oleh pengguna. Sistem dapat diaplikasikan untuk menyelesaikan permasalahan penjadwalan dan pelaporan.

Prospek penelitian selanjutnya adalah melakukan pengembangan pada proses penjadwalan maupun penerapan teknologi *geolocation*. Pengembangan pada penjadwalan dilakukan dengan mempertimbangkan metode optimasi lain. Seperti penambahan metode *fuzzy logic* untuk menentukan jadwal yang lebih dinamis sehingga *user* tidak perlu memilih secara manual. Algoritma penjadwalan yang digunakan pada kasus ini juga dapat diterapkan pada kasus lain. Sedangkan untuk *geolocation* penelitian lanjutan yang dapat dilakukan adalah proses presensi otomatis dengan *tracking real-time* maupun pelaporan keamanan lingkungan yang diklasifikasikan tingkat bahayanya dengan metode *mechine learning*.

DAFTAR PUSTAKA

- [1] A. A. Triana and A. M. Fauzi, "Dampak Pandemi Corona Virus Diserse 19 Terhadap Meningkatnya Kriminalitas Pencurian Sepeda Motor di Surabaya," *Syiah Kuala Law Journal*, vol. 4, no. 3, pp. 302–309, Desember 2020.
- [2] M. Anwar, "Asimilasi Dan Peningkatan Kriminalitas Di Tengah Pembatasan Sosial Berskala Besar Pandemi Corona," *ADALAH*, vol. 4, no. 1, pp. 101–106, 2020.
- [3] L. Mamluchah and N. Mubarok, "Peningkatan Angka Kejahatan Pencurian Pada Masa Pandemi Dalam Tinjauan Kriminologi Dan Hukum Pidana Islam," *Al-Jinayah: Jurnal Hukum Pidana Islam*, vol. 6, no. 1, pp. 1–26, Juni 2020.
- [4] I. A. Millah, "Penanggulangan Kejahatan Di Masa Pandemi Covid-19 (Dalam Perspektif Kriminologi Dan Viktimologi)," *Jurnal Komunikasi Hukum (JKH)*, vol. 6, no. 2, pp. 497–513, Agustus 2020.

- [5] N. Khoirunisa, “Peran Warga Dalam Revitalisasi Siskamling Di Tengah Pandemi Covid-19 Sebagai Upaya Menjaga Keamanan Lingkungan,” *Saskara : Indonesian Journal of Society Studies*, vol. 1, no. 1, pp. 27–39, Juni 2021.
- [6] O. S. Hermarani and K. Kuswardani, “Sistem Keamanan Dan Ketertiban Masyarakat Swakarsa Sebagai Upaya Penanggulangan Kejahatan di Kecamatan Ngargoyoso, Karanganyar,” *DiH: Jurnal Ilmu Hukum*, vol. 18, no. 2, pp. 212–222, Agustus 2022.
- [7] A. Nursikuwagus, F. W. Sari, and A. S. Sitanggung, “Pendekatan Teknologi Melalui Aplikasi Cerdas Penolong Masyarakat Untuk Meningkatkan Kepercayaan Masyarakat Terhadap Kepolisian Atas Tindak Pidana/Kejahatan Di Kecamatan Cobleng Bandung,” *Dharmakarya*, vol. 8, no. 2, p. 74, Juni 2019.
- [8] M. K. Anwar and U. Fadlillah, “Sistem Informasi Siskamling Berbasis Web,” Thesis, Universitas Muhammadiyah Surakarta, Surakarta, 2018.
- [9] S. N. Permadi, I. Arwani, and M. A. Akbar, “Pengembangan Aplikasi Berbasis Mobile Untuk Pelaporan Ancaman Kejahatan di Perumahan Panorama Cibodas,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 2, pp. 561–571, 2020.
- [10] W. Bagye, M. Ashari, and M. T. A. Zaen, “Prototipe Alat Kirim Pesan Singkat Tindak Kejahatan Sebagai Solusi Peningkatan Keamanan Berbasis Lokasi,” *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 19, no. 1, pp. 100–108, November 2019.
- [11] S. H. Hidayatullah and I. Prihandi, “Design and Build A Web-Based Co-Working Space System Using the Dynamic Priority Scheduling Algorithm Case Study: PT Permata Bank,” *International Journal of Computer Techniques*, vol. 7, no. 6, pp. 1–12, November 2020.
- [12] A. Maiti, U. Anand, S. Raj, S. Ahuja, and R. S. Sinha, “Resource Management During Disasters Using Priority Scheduling,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 3, no. 5, pp. 3257–3263, Mei 2021.
- [13] M. A. F. Nugroho, Y. W. Syaifudin, and D. Puspitasari, “Penentuan Jarak Terpendek Menggunakan Metode Dijkstra Pada Data Spasial Openstreetmap (Studi Kasus : Pada Perusahaan Pengantaran Barang Wahana Logistik Kota Malang),” *SMATIKA JURNAL*, vol. 9, no. 01, pp. 45–50, April 2019.
- [14] Nurkholis and M. A. Sobarnas, “Penerapan Geolokasi Pada Absensi Fasilitator Program Padat Karya Pemerintah Yang Tersebar Di Seluruh Wilayah Indonesia,” *INFOTECH : Jurnal Informatika & Teknologi*, vol. 1, no. 2, pp. 116–126, Desember 2020.
- [15] I. Sommerville, *Software Engineering*, 10th ed. London: Pearson Education Limited, 2016.
- [16] Z. Subecz, “Web-development with Laravel framework,” *Gradus*, vol. 8, no. 1, pp. 211–218, 2021.
- [17] M. A. Rizkyana, Y. Yunanto, Y. Yoga, and S. R. Widiyanto, “Implementasi Unit Testing Menggunakan Metode Test-First Development,” *MULTINETICS*, vol. 7, no. 1, pp. 37–47, Mei 2021.
- [18] H. Hendri, J. W. Hasiholan Manurung, R. A. Ferian, W. F. Hanaatmoko, and Y. Yulianti, “Pengujian Black Box Pada Aplikasi Sistem Informasi Pengelolaan Masjid Menggunakan Teknik Equivalence Partitions,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 3, no. 2, p. 107, April 2020.
- [19] Lukman and M. Yudhiastari, “Analisis Kinerja Web Server Apache Dan Litespeed Menggunakan Httpperf Pada Virtual Private Server (VPS),” *Respati*, vol. 16, no. 2, pp. 24–32, Juli 2021.
- [20] U. Ependi, T. B. Kurniawan, and F. Panjaitan, “System Usability Scale Vs Heuristic Evaluation: A Review,” *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 10, no. 1, pp. 65–74, April 2019.