

# Implementasi Aplikasi Manajemen Pelanggaran Santri SMP-MA Berbasis Website Menggunakan Framework CodeIgniter 4

Ramya Harfi Zhilaa\*, Ade Priyatna

Fakultas Teknik dan Informatika, Program Studi Informatika, Universitas Bina Sarana Informatika, Jakarta, Indonesia

Email: <sup>1,\*</sup>ramyahrfz@email.com, <sup>2</sup>ade.aeq@bsi.ac.id

Email Penulis Korespondensi: 15210231@bsi.ac.id

**Abstrak**—Pengelolaan data pelanggaran santri di lingkungan pesantren sering kali masih dilakukan secara manual, sehingga rentan terhadap kesalahan pencatatan, duplikasi data, serta keterlambatan dalam penyajian laporan. Permasalahan ini berimplikasi pada kurangnya transparansi dan akuntabilitas dalam sistem tata tertib santri. Penelitian ini bertujuan untuk mengembangkan aplikasi manajemen pelanggaran santri berbasis *website* di SMP dan MA Hidayatullah Depok dengan menggunakan metode pengembangan perangkat lunak model *Waterfall*. Aplikasi yang dirancang memiliki fitur utama berupa manajemen data santri dan pencatatan jenis pelanggaran. Implementasi sistem dilakukan menggunakan *framework CodeIgniter 4* dengan basis data MySQL. Hasil pengujian menggunakan metode *Blackbox Testing* menunjukkan bahwa seluruh fungsi aplikasi berjalan sesuai kebutuhan, sedangkan *User Acceptance Test* (UAT) melibatkan 7 dari 7 Murobbi yang memberikan hasil bahwa 100% responden menilai aplikasi sangat membantu dalam proses pencatatan pelanggaran. Dengan demikian, aplikasi ini dapat meningkatkan efisiensi, transparansi, sekaligus menjadi model pengembangan sistem informasi pendidikan yang sesuai dengan konteks pesantren.

**Kata Kunci:** Aplikasi Web; Sistem Informasi; CodeIgniter; Waterfall; Pelanggaran Santri

**Abstract**—The management of student violation records in Islamic boarding schools is often still conducted manually, making it prone to recording errors, data duplication, and delays in report generation. This issue leads to a lack of transparency and accountability in the enforcement of school discipline. This research aims to develop a web-based student violation management application at SMP and MA Hidayatullah Depok using the Waterfall software development model. The application features include student data management and violation recording. The system was implemented using the CodeIgniter 4 framework and MySQL database. The testing results using the Blackbox Testing method showed that all application functions worked as expected, while the User Acceptance Test (UAT) involve 7 of 7 teachers indicated that 100% of respondents found the application highly useful in managing violation records. Therefore, this application improves efficiency, transparency and can serve as a model for educational information systems tailored to the boarding school context.

**Keywords:** Web Application, Information System, CodeIgniter, Waterfall, Student Violation

## 1. PENDAHULUAN

Kedisiplinan santri merupakan aspek fundamental dalam menunjang keberhasilan pendidikan di lingkungan pesantren. Namun, pengelolaan data pelanggaran santri masih banyak dilakukan secara manual dengan pencatatan di buku besar atau lembaran kertas[1]. Proses ini menimbulkan berbagai kendala, seperti duplikasi data, kesulitan dalam pencarian informasi, serta keterlambatan dalam penyajian laporan kepada pihak sekolah maupun orang tua. Kondisi tersebut menyebabkan rendahnya efektivitas sistem administrasi dan berpotensi mengurangi transparansi dalam penerapan aturan. Oleh karena itu, dibutuhkan sebuah sistem informasi yang mampu mengelola data pelanggaran santri secara terintegrasi, cepat, dan akurat [2]. Solusi yang ditawarkan dalam penelitian ini adalah pengembangan aplikasi manajemen pelanggaran santri berbasis *website* yang dapat mendukung proses pencatatan, pengolahan, hingga pelaporan data pelanggaran secara digital. Dengan adanya sistem ini, diharapkan pengelolaan pelanggaran santri menjadi lebih efisien, transparan, dan dapat diakses oleh pihak terkait secara *real-time*[3].

Penelitian terkait telah banyak dilakukan dalam konteks pengembangan sistem informasi pendidikan. Misalnya, penelitian oleh Fina tahun 2025 yang mengembangkan aplikasi pengelolaan pelanggaran siswa berbasis Android, namun keterbatasannya terletak pada aspek aksesibilitas karena hanya dapat digunakan melalui perangkat tertentu [4]. Penelitian lain oleh Saputro dan Sugiharto tahun 2025 mengembangkan sistem informasi pelaporan pelanggaran siswa berbasis web, tetapi belum dilengkapi dengan fitur notifikasi kepada wali siswa[5]. Kemudian, penelitian yang dilakukan oleh Anggita dan Al Azfar tahun 2024 mengintegrasikan sistem pelanggaran dengan *database* sekolah, namun belum memanfaatkan model pengembangan perangkat lunak yang terstruktur seperti *Waterfall*[6]. Selanjutnya, penelitian dari Dhamma dkk. tahun 2023 membahas sistem informasi *monitoring* kedisiplinan siswa dengan dukungan *dashboard* interaktif, tetapi sistem tersebut belum mengakomodasi kebutuhan pesantren dengan model asrama[7]. Terakhir, penelitian oleh Kahpi tahun 2024 mengembangkan aplikasi tata tertib berbasis *mobile* untuk sekolah menengah, namun aplikasinya belum dilengkapi dengan laporan terintegrasi bagi pihak manajemen[8].

Berdasarkan penelitian-penelitian tersebut, dapat dilihat adanya gap analisis yang masih menyisakan ruang untuk pengembangan lebih lanjut. Kelemahan yang ditemukan antara lain keterbatasan platform dan kurangnya penyesuaian dengan konteks pesantren yang memiliki aturan khusus. Dengan demikian, penelitian ini menghadirkan kebaruan berupa sistem manajemen pelanggaran santri berbasis *website* dengan pendekatan pengembangan terstruktur model *Waterfall*, dilengkapi fitur laporan terintegrasi yang mendukung pengambilan keputusan di lembaga pendidikan berbasis pesantren.

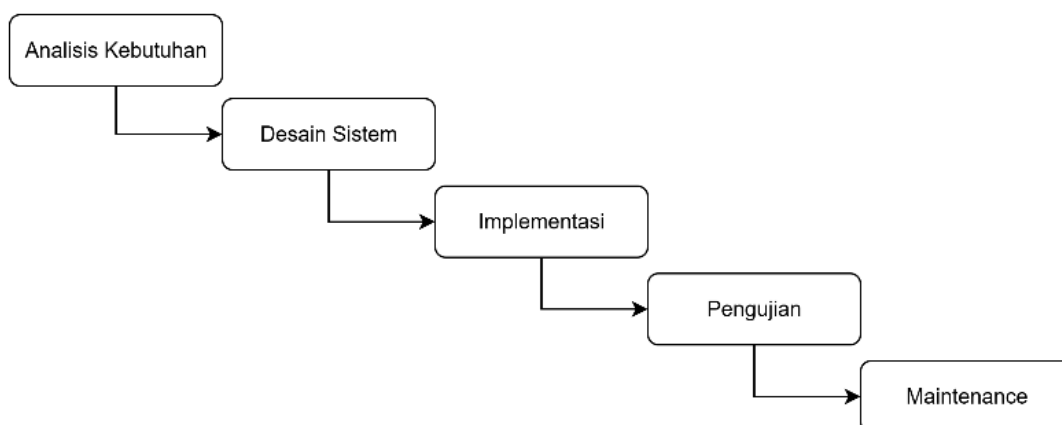
Tujuan utama dari penelitian ini adalah merancang dan mengimplementasikan aplikasi *website* untuk manajemen pelanggaran santri di SMP dan MA Hidayatullah Depok, sehingga mampu meningkatkan efisiensi, transparansi, dan akuntabilitas dalam pengelolaan pelanggaran. Penelitian ini diharapkan dapat memberikan kontribusi nyata dalam

pemanfaatan teknologi informasi di lingkungan pendidikan Islam, khususnya pesantren. Selain itu, aplikasi ini diharapkan dapat menjadi model pengembangan sistem serupa di lembaga pendidikan lain yang memiliki kebutuhan pengelolaan pelanggaran siswa[9].

Dengan demikian, penelitian ini tidak hanya berorientasi pada pengembangan perangkat lunak semata, tetapi juga memiliki signifikansi praktis dalam meningkatkan tata kelola pendidikan di pesantren[10]. Hasil penelitian diharapkan dapat memberikan solusi inovatif yang adaptif terhadap kebutuhan sekolah berbasis asrama, sekaligus memperkaya literatur penelitian di bidang sistem informasi pendidikan.

## 2. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode *Software Development Life Cycle (SDLC)* model *Waterfall*, yang sesuai untuk pengembangan perangkat lunak dengan tahapan sistematis dan berurutan. Model ini dipilih karena setiap tahapannya saling bergantung dan terstruktur, sehingga memudahkan proses implementasi serta pengujian aplikasi [11], [12].



**Gambar 1.** Tahapan Waterfall

a. Analisis Kebutuhan

Pada tahap ini dilakukan identifikasi kebutuhan fungsional dan non-fungsional sistem berdasarkan hasil wawancara dengan Murobbi asrama di MA dan SMP Hidayatullah Depok, yang bertanggung jawab langsung terhadap pencatatan dan penanganan pelanggaran santri. Selain itu, dilakukan pula observasi langsung terhadap proses pencatatan pelanggaran santri yang masih dilakukan secara manual untuk memperoleh gambaran nyata mengenai alur kerja yang berjalan.

b. Perancangan Sistem

Tahap perancangan meliputi pembuatan diagram alir data (Data Flow Diagram/DFD), perancangan basis data (Entity Relationship Diagram/ERD), serta rancangan antarmuka pengguna (user interface design). Pada tahap ini juga ditentukan arsitektur sistem berbasis web menggunakan framework CodeIgniter 4 dan basis data MySQL yang diintegrasikan dengan server lokal sekolah.

c. Implementasi

Proses implementasi dilakukan menggunakan *framework CodeIgniter 4* dengan bahasa pemrograman PHP dan basis data MySQL. Fokus pengembangan diarahkan pada pembuatan fitur utama manajemen data santri dan pengguna sistem, serta input dan pendataan pelanggaran santri.

d. Pengujian

Pengujian sistem dilakukan dengan dua pendekatan:

1. Black Box Testing, untuk menguji fungsionalitas setiap fitur tanpa melihat kode program guna memastikan setiap komponen berjalan sesuai kebutuhan.
2. User Acceptance Test (UAT), dilakukan terhadap 7 responden Murobbi. Responden diminta untuk menggunakan sistem dan menilai kemudahan serta keefektifan aplikasi melalui skala Likert 1–5 terhadap aspek antarmuka, kecepatan sistem, dan kemudahan penggunaan.

Hasil UAT menunjukkan bahwa 100% responden menyatakan puas terhadap kinerja aplikasi, yang mengindikasikan bahwa sistem telah memenuhi kebutuhan pengguna secara fungsional maupun praktis.

e. Pemeliharaan

Setelah sistem diimplementasikan, dilakukan tahap pemeliharaan untuk memperbaiki *bug*, menambah fitur baru, serta menyesuaikan kebutuhan pengguna seiring waktu.

Model *Waterfall* dipilih karena memiliki alur pengembangan yang sistematis, sehingga meminimalkan kesalahan dalam perancangan dan implementasi. Dengan metodologi ini, sistem diharapkan dapat dikembangkan secara terstruktur dan sesuai dengan kebutuhan institusi pendidikan pesantren.

### 3. HASIL DAN PEMBAHASAN

Hasil penelitian ini berupa implementasi aplikasi Manajemen Pelanggaran Santri Berbasis *Website* dengan studi kasus di SMP dan MA Hidayatullah Depok. Pengembangan dilakukan menggunakan model *Waterfall* yang mencakup analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Aplikasi ini dirancang untuk mendukung proses pencatatan pelanggaran santri, penyimpanan data terstruktur, penyajian laporan, serta pemberian notifikasi otomatis kepada wali santri. Bagian ini akan membahas hasil implementasi setiap tahapan pengembangan sistem, pengujian yang dilakukan, serta analisis keefektifan sistem berdasarkan hasil evaluasi pengguna.

#### 3.1 Analisis Kebutuhan dan Perancangan Sistem

Pada tahap awal penelitian, dilakukan analisis kebutuhan sistem yang menjadi landasan utama dalam perancangan aplikasi manajemen pelanggaran santri. Analisis ini melibatkan identifikasi pengguna sistem, kebutuhan fungsional dan non-fungsional, serta kebutuhan teknis perangkat lunak. Hasil analisis kebutuhan menjadi acuan dalam menyusun rancangan sistem yang terstruktur dan sistematis, sesuai dengan model pengembangan perangkat lunak *Waterfall*[13], [14].

##### 3.1.1 Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan layanan utama yang harus disediakan oleh sistem agar dapat memenuhi tujuan penggunaannya. Berdasarkan hasil observasi dan wawancara dengan Murobbi asrama, diperoleh kebutuhan fungsional yang menjadi dasar perancangan sistem. Rincian kebutuhan tersebut dapat dilihat pada Tabel 1. Kebutuhan Fungsional.

**Tabel 1.** Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Manajemen Data Santri	Sistem mampu menambah, mengedit, dan menghapus data santri.
2	Pencatatan Jenis Pelanggaran	Sistem menyediakan fitur pencatatan jenis pelanggaran sesuai aturan pesantren.
3	Input Pelanggaran oleh Murobbi	Murobbi dapat mencatat pelanggaran santri langsung melalui sistem berbasis web.

Berdasarkan Tabel 1, sistem dirancang untuk memiliki tiga fungsi utama, yaitu pengelolaan data santri, pencatatan jenis pelanggaran, dan input data pelanggaran oleh Murobbi. Ketiga fungsi ini merupakan inti dari proses manajemen pelanggaran di lingkungan pesantren dan menjadi pondasi dalam tahap pengembangan sistem agar sesuai dengan kebutuhan pengguna.

##### 3.1.2 Analisis Kebutuhan Non-Fungsional

Selain kebutuhan fungsional, terdapat pula kebutuhan non-fungsional yang harus dipenuhi agar sistem dapat berjalan secara optimal, aman, dan mudah digunakan oleh pengguna. Rincian kebutuhan non-fungsional ditunjukkan pada Tabel 2. Kebutuhan Non-Fungsional.

**Tabel 2.** Kebutuhan Non Fungsional

No	Kebutuhan Non-Fungsional	Deskripsi
1	Antarmuka Sederhana	Antarmuka aplikasi harus mudah dipahami pengguna.
2	Keamanan Data	Sistem harus melindungi data dari manipulasi dan akses tidak sah.
3	Berbasis <i>Web</i>	Aplikasi harus berbasis <i>web</i> agar bisa diakses dari berbagai perangkat.
4	Respon Cepat	Waktu respon maksimal 3 detik per permintaan.

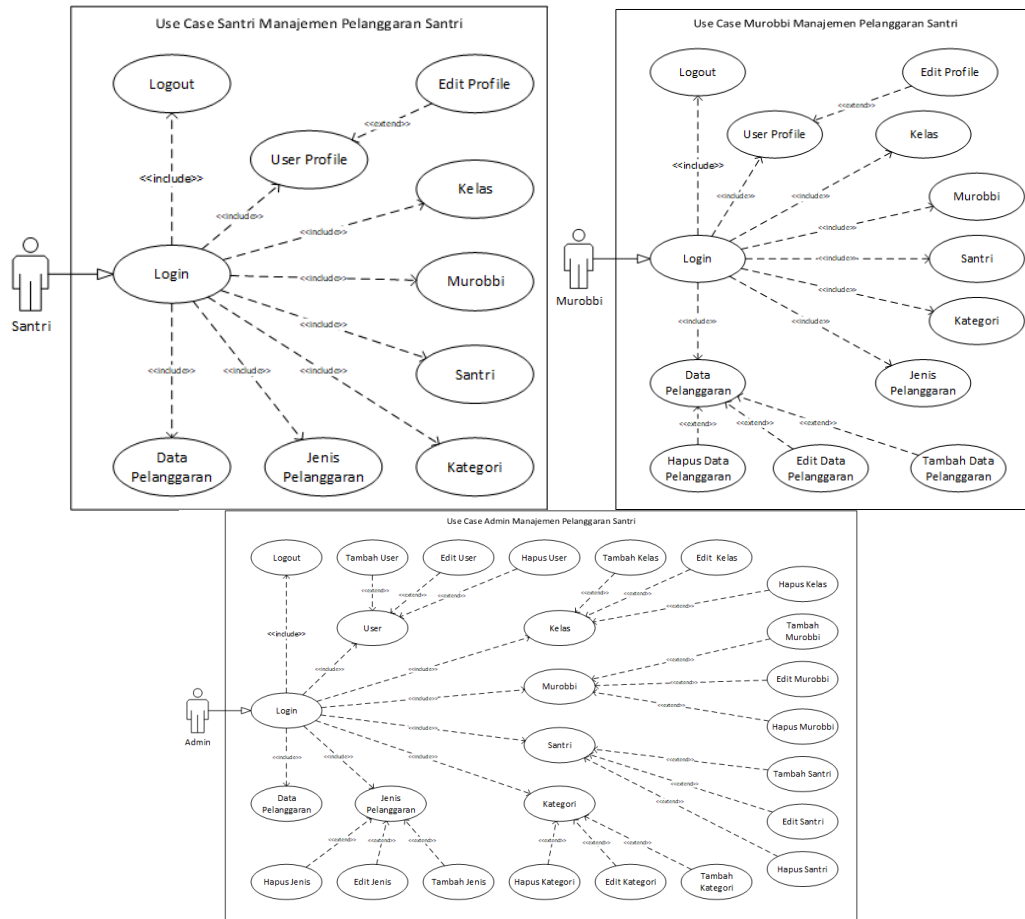
Berdasarkan Tabel 2, dapat diketahui bahwa sistem tidak hanya menekankan pada fungsi operasional, tetapi juga pada kemudahan akses, keamanan data, dan kecepatan respon. Dengan memenuhi kebutuhan non-fungsional tersebut, aplikasi diharapkan mampu memberikan pengalaman pengguna mendukung efektivitas pengelola pesantren.

##### 3.1.3 Permodelan Sistem

Tahap perancangan sistem dilakukan dengan menggunakan beberapa model visualisasi untuk menggambarkan interaksi pengguna dengan sistem serta struktur internalnya[15]. Model-model ini membantu pengembang memahami alur kerja dan relasi antar komponen. Adapun permodelan sistem yang digunakan dapat dilihat pada Gambar 2 hingga Gambar 5.

###### a. Use Case Diagram.

Use Case Diagram menggambarkan interaksi antara pengguna (aktor) dengan sistem, termasuk peran Admin, Murobbi, dan Santri. Diagram ini memberikan gambaran umum mengenai fungsi-fungsi utama yang dapat diakses oleh masing-masing aktor.

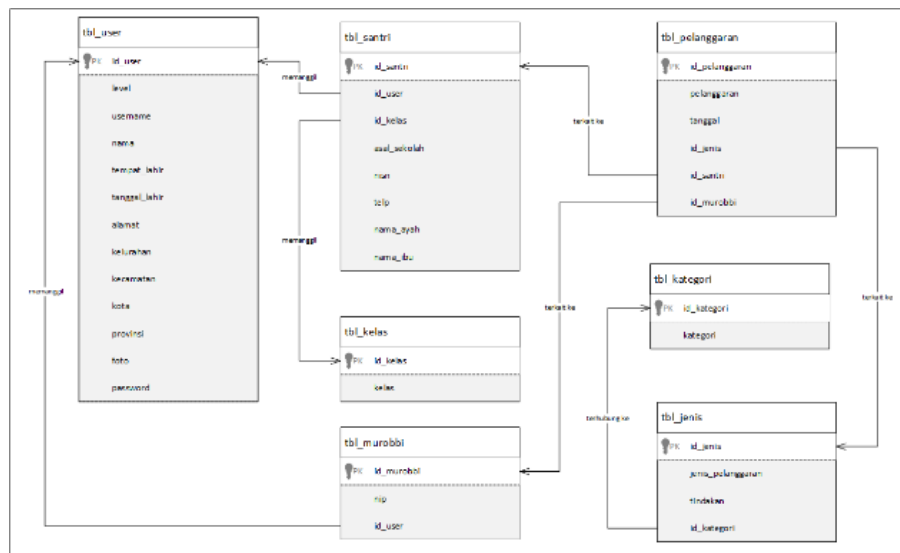


**Gambar 2.** Use Case Admin, Murobbi dan Santri

Berdasarkan Gambar 2, terlihat bahwa Admin memiliki akses penuh untuk mengelola data santri, pelanggaran, dan laporan. Sementara itu, Murobbi memiliki hak akses untuk mencatat pelanggaran, dan Santri dapat melihat data pelanggaran sendiri.

b. *Class Diagram.*

Class Diagram digunakan untuk mendeskripsikan hubungan antar kelas dalam sistem, termasuk atribut, metode, serta relasi antar entitas.

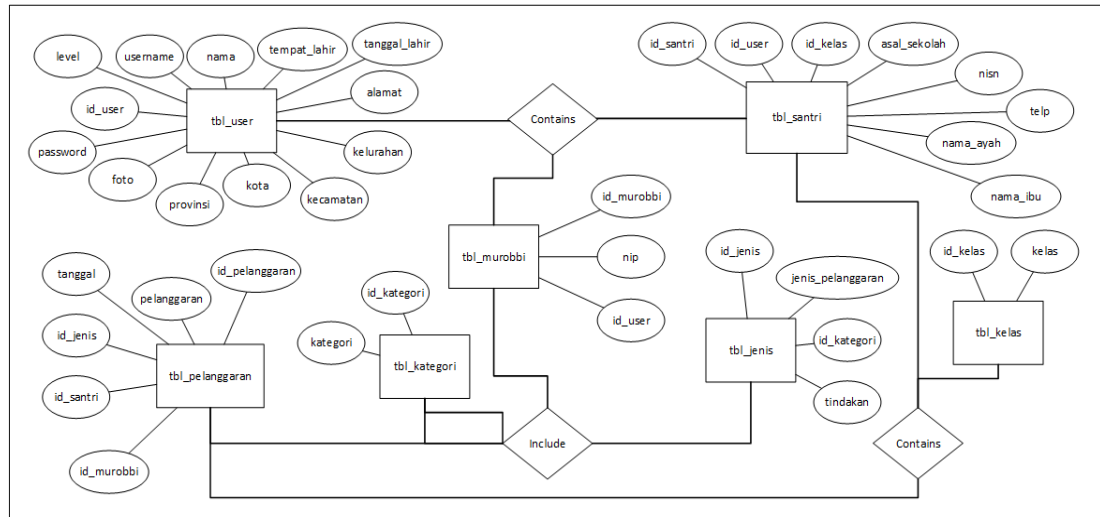


**Gambar 3.** Class Diagram

Berdasarkan Gambar 3, setiap kelas memiliki hubungan yang saling terintegrasi, seperti kelas *Santri*, *Pelanggaran*, dan *User*, yang dihubungkan melalui relasi one-to-many.

c. *Entity Relationship Diagram (ERD).*

ERD digunakan untuk merancang struktur basis data yang akan digunakan dalam aplikasi.

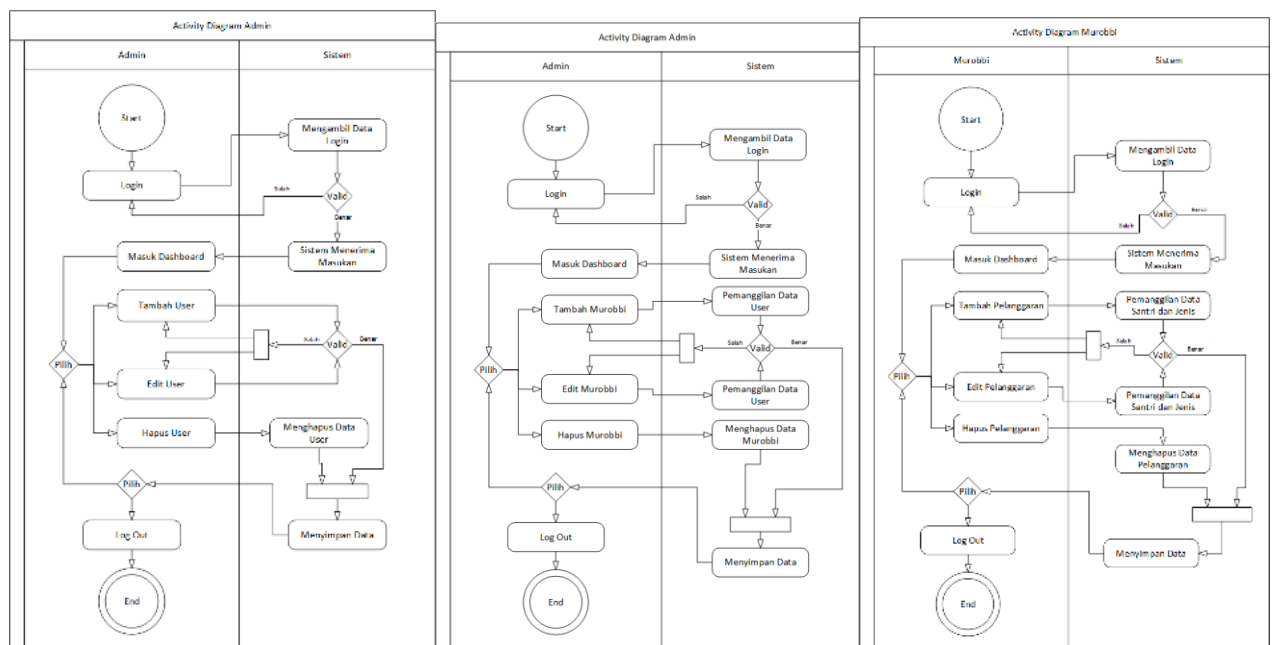


**Gambar 4. Entity Relationship Diagram**

Berdasarkan Gambar 4, entitas utama yang terbentuk antara lain *Santri*, *Pelanggaran*, *Jenis Pelanggaran*, dan *User*. Relasi antar entitas ini memastikan data saling terhubung dengan baik dan dapat dikelola secara efisien oleh sistem.

d. *Activity Diagram.*

Activity Diagram digunakan untuk memvisualisasikan alur aktivitas dari setiap fitur utama sistem. Diagram ini menunjukkan urutan proses dari mulai pengguna login hingga proses pelaporan pelanggaran selesai dilakukan.



**Gambar 5. Activity Diagram Admin dan Murobbi**

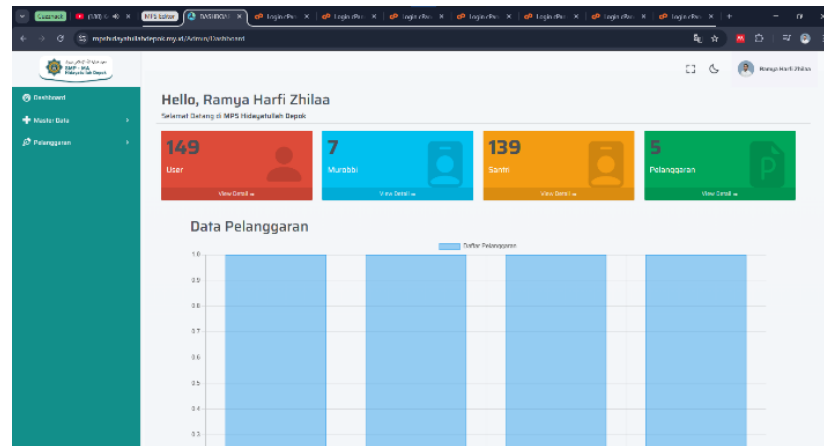
Berdasarkan Gambar 5, dapat dilihat bahwa Admin dan Murobbi memiliki jalur aktivitas yang berbeda namun saling berkaitan, terutama pada proses validasi dan pelaporan data pelanggaran.

### 3.2 Implementasi Sistem

Implementasi sistem dilakukan berdasarkan hasil perancangan. Sistem dibangun menggunakan *framework* CodeIgniter 4 sebagai *backend*, MySQL sebagai basis data, serta HTML, CSS, dan JavaScript untuk antarmuka pengguna. Implementasi fitur dilakukan secara modular agar mudah dikembangkan lebih lanjut [16], [17].

### 3.2.1 Implementasi Dashboard Aplikasi

*Dashboard* menjadi halaman utama yang ditampilkan setelah pengguna berhasil *login*. *Dashboard* berisi ringkasan data santri, *user*, dan jumlah pelanggaran.



Gambar 6. Dashboard MPS

### 3.2.2 Implementasi Manajemen Data Santri

Fitur manajemen data santri memungkinkan admin untuk menambah, mengedit, dan menghapus data santri. Data ini menjadi basis utama dalam pencatatan pelanggaran.

The screenshot shows the 'Form Tambah Santri' (Add Santri Form). It includes input fields for 'Nama' (Name), 'Kode' (Code), 'Jenis Santri' (Santri Type), 'NISN', 'No Telp' (Phone Number), 'Nama Ayah' (Father's Name), and 'Nama Ibu' (Mother's Name). There are 'Simpan' (Save) and 'Batal' (Cancel) buttons at the bottom.

Gambar 7. Form Input Santri

### 3.2.3 Implementasi Pencatatan Pelanggaran

Murobbi dapat melakukan pencatatan pelanggaran dengan memilih nama santri dan jenis pelanggaran yang telah terdaftar di sistem.

The screenshot shows the 'Form Tambah Pelanggaran' (Add Violation Form). It includes a dropdown for 'Nama Santri' (Santri Name), a text field for 'Pelanggaran' (Violation), a dropdown for 'Jenis Pelanggaran' (Type of Violation), and a date field for 'Tanggal' (Date). There are 'Simpan' (Save) and 'Batal' (Cancel) buttons at the bottom.

Gambar 8. Input Data Pelanggaran Santri

### 3.3 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi yang dikembangkan telah berjalan sesuai kebutuhan pengguna dan bebas dari kesalahan fungsi. Dalam penelitian ini, pengujian dilakukan dengan dua metode, yaitu *Black Box Testing* dan *User Acceptance Test* (UAT) [18], [19].

#### 3.3.1 Blackbox Testing

Metode Black Box Testing digunakan untuk menguji fungsi-fungsi utama sistem tanpa memperhatikan struktur internal kode program. Pengujian ini bertujuan untuk mengetahui apakah setiap fitur menghasilkan keluaran sesuai yang diharapkan berdasarkan input tertentu. Secara umum, hasil pengujian menunjukkan bahwa seluruh fitur utama dapat berjalan sesuai harapan. Beberapa aspek yang diuji antara lain:

- Fitur *login* berhasil membedakan hak akses admin, Murobbi, dan wali santri.
- Fitur *login* tidak sesuai atau tidak valid akan ditolak oleh sistem.
- Fitur penginputan data *user* dapat digunakan tanpa kesalahan.
- Fitur pelanggaran santri berjalan sesuai alur.
- Input ganda (data duplikat) ditolak oleh sistem.
- Fitur penginputan tidak lengkap atau tidak sesuai menghasilkan peringatan untuk melengkapi data.

Untuk pengujian peran Admin, hasil rinci dapat dilihat pada Tabel 3. Black Box Test Admin.

**Tabel 3.** Black Box Test Admin

No	Fitur yang Diuji	Input	Output yang Diharapkan	Status
1	<i>Login</i>	<i>Username &amp; Password</i> valid	Masuk ke <i>dashboard</i> peran	Berhasil
2	<i>Input Data User</i>	Form Data diri, status, alamat, keterangan	Data tersimpan dan muncul dalam tabel	Berhasil
3	<i>Input Jenis Pelanggaran</i>	Form Jenis-jenis pelanggaran sesuai dustur yang berlaku	Data tersimpan dan muncul dalam tabel	Berhasil
4	<i>Salah Login</i>	<i>Username &amp; Password</i> tidak valid	Gagal masuk ke <i>dashboard</i>	Berhasil
5	<i>Input Tidak Lengkap</i>	Kosongkan form <i>input</i>	Data tidak bisa disimpan dan muncul peringatan untuk mengisi data	Berhasil
6	<i>Input Ganda</i>	<i>Input</i> yang sama dengan yang sebelumnya	Gagal menyimpan data dan muncul peringatan data sudah ada	Berhasil

Berdasarkan Tabel 3, dapat disimpulkan bahwa seluruh fungsi pada level Admin telah berjalan dengan baik dan sesuai kebutuhan sistem. Sistem mampu memvalidasi input, menolak data ganda, serta membatasi akses berdasarkan kredensial yang benar. Hal ini menunjukkan bahwa mekanisme keamanan dan pengolahan data pada bagian Admin berfungsi dengan optimal.

Selanjutnya, pengujian juga dilakukan untuk memastikan fungsi-fungsi yang dijalankan oleh Murobbi bekerja sesuai kebutuhan. Hasil pengujian tersebut dapat dilihat pada Tabel 4. Black Box Test Murobbi.

**Tabel 4.** Black Box Test Murobbi

No	Fitur yang Diuji	Input	Output yang Diharapkan	Status
1	<i>Login</i>	<i>Username &amp; Password</i> valid	Masuk ke <i>dashboard</i> peran	Berhasil
2	<i>Edit Profil</i>	Edit Form Data diri, status, alamat, keterangan	Data tersimpan dan muncul dalam tabel	Berhasil
3	<i>Input Data Pelanggaran</i>	Form Data pelanggaran santri sesuai jenis pelanggaran yang sudah ada	Data tersimpan dan muncul dalam tabel	Berhasil
4	<i>Salah Login</i>	<i>Username &amp; kata sandi</i> tidak valid	Gagal masuk ke <i>dashboard</i>	Berhasil
5	<i>Input Tidak Lengkap</i>	Kosongkan form inputan	Data tidak bisa disimpan, muncul peringatan untuk mengisi data	Berhasil
6	<i>Input Ganda</i>	Inputan yang sama dengan yang sebelumnya	Gagal menyimpan data dan muncul peringatan data sudah ada	Berhasil

Berdasarkan Tabel 4, fitur yang digunakan oleh Murobbi juga berjalan sesuai dengan yang diharapkan. Sistem dapat menolak input ganda maupun data tidak lengkap, serta memberikan notifikasi apabila terjadi kesalahan dalam proses pengisian. Dengan demikian, dapat disimpulkan bahwa fungsi-fungsi utama pada sisi pengguna Murobbi telah teruji dan memenuhi kriteria fungsional sistem.

#### 3.3.2 User Acceptance Test

Pengujian UAT melibatkan 7 responden Murobbi. Hasilnya:

- 100% responden menyatakan aplikasi sangat membantu dalam pencatatan pelanggaran.
- Tidak ada responden yang menyatakan aplikasi tidak membantu.

$$\text{Persentase} = \left( \frac{168}{168} \right) \times 100\% = 100\%$$

Seluruh responden (100%) menyatakan bahwa aplikasi manajemen pelanggaran santri layak digunakan secara institusional[20]. Aplikasi dinilai efisien dalam pencatatan dan pelaporan, mudah dipantau secara *real-time*, ramah pengguna tanpa perlu pelatihan tambahan, juga mendukung pengambilan keputusan yang objektif. Hal ini menunjukkan bahwa aplikasi telah diterima sepenuhnya dan siap diimplementasikan di SMP dan MA Hidayatullah Depok.

#### 4. KESIMPULAN

Hasil implementasi aplikasi *Manajemen Pelanggaran Santri Berbasis Website* menunjukkan bahwa sistem berhasil dikembangkan secara terstruktur menggunakan model Waterfall melalui tahapan analisis kebutuhan, perancangan, implementasi, hingga pengujian. Seluruh fitur utama, seperti manajemen data santri, pencatatan pelanggaran, laporan terintegrasi, serta notifikasi otomatis kepada wali santri, telah berfungsi dengan baik sesuai rancangan dan kebutuhan institusi pendidikan berbasis pesantren. Pengujian menggunakan metode Black Box Testing menunjukkan bahwa seluruh fungsi berjalan tanpa kendala, sedangkan hasil *User Acceptance Test* (UAT) terhadap tujuh Murobbi menunjukkan tingkat penerimaan 100%, menandakan sistem mudah digunakan, efisien dalam pencatatan, transparan dalam pelaporan, serta mendukung monitoring secara *real-time*. Penelitian ini memberikan kontribusi baru dalam bentuk pengembangan sistem manajemen pelanggaran yang secara khusus dirancang untuk lingkungan pesantren dengan integrasi fitur notifikasi otomatis kepada wali santri sebagai inovasi dalam pembinaan berbasis teknologi. Selain terbukti mampu mengatasi permasalahan administrasi manual seperti keterlambatan pencatatan, kesalahan data, dan minimnya transparansi, aplikasi ini juga berperan dalam mendukung pembentukan karakter santri melalui rekapitulasi pelanggaran yang dapat dianalisis secara objektif dan historis. Ke depan, penelitian ini masih dapat dikembangkan lebih lanjut dengan menambahkan fitur analitik untuk mendeteksi pola pelanggaran, mengintegrasikannya dengan sistem akademik pesantren, atau mengembangkan versi *mobile application* agar akses dan monitoring dapat dilakukan secara lebih fleksibel oleh seluruh pihak terkait.

#### REFERENCES

- [1] Z. Pribadi, A. S. Puspaningrum, P. Parjito, M. I. Takaendengan, dan N. F. Utomo, "Aplikasi Sistem Pengelolaan Nilai Kedisiplinan Siswa Sman X Berbasis Web," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 4, no. 2, hlm. 192–200, Jun 2023, doi: 10.33365/jatika.v4i2.2598.
- [2] A. Khairi dkk., "Sistem Informasi Berbasis Web pada Pelanggaran Santri di Pondok Pesantren Nurul Jadid," *TRILOGI: Jurnal Ilmu Teknologi, Kesehatan, dan Humaniora*, vol. 3, no. 2, hlm. 71–76, 2022, doi: 10.33650/trilogi.v3i2.4183.
- [3] A. Muchasan dan D. Rohmawan, "PEMANFAATAN TEKNOLOGI DI PESANTREN (DAMPAK DAN SOLUSI DALAM KONTEKS PENDIDIKAN)," *INOVATIF*, vol. 10, no. 1, hlm. 16–33, Feb 2024, doi: 10.55148/inovatif.v10i1.849.
- [4] I. A. Welly Purnomo, Muhammad Kanzul Fikri, "Pengembangan Aplikasi Monitoring Pelanggaran Siswa pada SMA Negeri 1 Madiun," *Jurnal MediaTIK*, vol. 1, no. 1, hlm. 1–8, 2024, [Daring]. Tersedia pada: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13753/6110>
- [5] W. Saputro, T. Sugiharto, B. Dina, M. Azhari, dan F. Perwitosari Joko, "Aplikasi Sistem Informasi Pelanggaran Siswa Berbasis Web," *Smart Comp: Jurnalnya Orang Pintar Komputer*, vol. 12, no. 1, 2023, doi: 10.30591/smartcomp.v12i1.4594.
- [6] N. A. Al Azfar dan S. D. Anggita, "PENERAPAN METODE WATERFALL PADA SISTEM INFORMASI E-RAPOR," *Information System Journal*, vol. 7, no. 01, hlm. 45–55, Jun 2024, doi: 10.24076/infosjournal.2024v7i01.1582.
- [7] M. N. Maulana, "Implementasi Metode Prototipe dalam Pengembangan Sistem Monitoring Berbasis Website di Pondok Pesantren HPAIC Merapi," *Jurnal sosial dan sains*, vol. 5, no. 7, hlm. 1970–1984, 2025, doi: 10.59188/jurnalsosains.v5i7.32304.
- [8] S. Kurniawan, "Perancangan Aplikasi Pelanggaran Siswa Berbasis Web Menggunakan PHP dan MySQL," *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 3, no. 3, hlm. 35–41, 2024, doi: 10.31004/riggs.v3i3.774.
- [9] W. Setiawan, A. S. Pratama, W. Wirdayanti, dan Y. Y. Joeфриe, "Implementasi Sistem Informasi Bimbingan Konseling Pelanggaran Siswa Dengan Sistem Point," *Indonesian Journal of Computer Science*, vol. 13, no. 3, hlm. 4957–4965, 2024, doi: 10.33022/ijcs.v13i3.3958.
- [10] A. Khairi dkk., "Sistem Informasi Berbasis Web pada Pelanggaran Santri di Pondok Pesantren Nurul Jadid," *TRILOGI: Jurnal Ilmu Teknologi, Kesehatan, dan Humaniora*, vol. 3, no. 2, hlm. 71–76, Sep 2022, doi: 10.33650/trilogi.v3i2.4183.
- [11] F. R. Putri dan A. Suharso, "Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi," *INFOTECH journal*, vol. 9, no. 2, hlm. 377–382, 2023, doi: 10.31949/infotech.v9i2.6270.
- [12] A. Binuko Paksi, ul Hafidhoh, dan S. Kariagil Bimonugroho, "Perbandingan Model Pengembangan Perangkat Lunak," vol. 14, no. 1, hlm. 70–79, 2023.
- [13] A. Akbar, A. Hendra Brata, dan A. A. Soebroto, "Pengembangan Sistem Informasi Kompetisi Berbasis Web (Studi Kasus: Universitas Brawijaya)," vol. 1, no. 1, hlm. 2548–964, 2017, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [14] I. A. Pratiwi, P. Widyansih, dan K. Surakarta, "Rancang Bangun Sistem Informasi Aduan Digital Desa Gadingan Menggunakan Metode Incremental," vol. 13, no. 3.
- [15] J. Hendrawan, I. D. Perwitasari, Z. Hasyiyati, dan D. S. Hasanah, "Model UML Sistem Informasi Monitoring Pembayaran SPP Siswa SMA Negeri 1 Binjai," *Jurnal Minfo Polgan*, vol. 13, no. 2, hlm. 1823–1831, 2024, doi: 10.33395/jmp.v13i2.14270.
- [16] H. Setiawan dan M. M. Gazali, "Perancangan Sistem Digital untuk Monitoring Kesehatan Mental dan Fisik Mahasiswa Berbasis DASS-21, BMI, dan Aktivitas Fisik," *Digital Transformation Technology*, vol. 5, no. 1, hlm. 266–277, 2025, doi: 10.47709/digitech.v5i1.6115.



- [17] M. A. Senubekti, M. A. Kartawidjaja, dan G. B. W. Adyana, “Penerapan Metode Hybrid Waterfall-RAD pada Sistem Antrian Web untuk Optimalisasi Layanan di Pengadilan Negeri Bale Bandung,” *JOINS (Journal of Information System)*, vol. 10, no. 1, hlm. 60–70, 2025, doi: 10.33633/joins.v10i1.12668.
- [18] A. F. Wijaya dan B. A. Wardijono, “Pengukuran Kualitas Aplikasi Custody Berdasarkan ISO 25010 Menggunakan Otomatisasi Pengujian Blackbox,” *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 2, hlm. 937–946, 2023, doi: 10.30865/klik.v4i2.1215.
- [19] A. R. Rambe dan H. Prihantoro, “Pengujian Otomatis Aplikasi Mobile dengan Teknik Black-box Menggunakan Appium (Studi Kasus: Pengembangan Aplikasi Jala Mobile),” *Prosiding Automata*, 2022, [Daring]. Tersedia pada: <https://journal.uui.ac.id/AUTOMATA/article/view/24198>
- [20] Aliyah Aliyah, Nahrin Hartono, dan Asrul Azhari Muin, “Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang,” *Switch : Jurnal Sains dan Teknologi Informasi*, vol. 3, no. 1, hlm. 84–100, Des 2024, doi: 10.62951/switch.v3i1.330.