

**PENGEMBANGAN SISTEM BERBASIS WEB UNTUK
DIGITALISASI ADMINISTRASI DESA RIAS**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

RIZQI PRATAMA NIM: 1062255

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2025**

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

**PENGEMBANGAN SISTEM BERBASIS WEB
UNTUK DIGITALISASI ADMINISTRASI DESA RIAS**

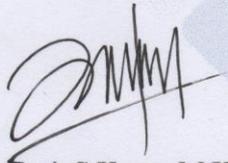
Oleh:

Rizqi Pratama / 1062255

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

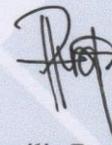
Menyetujui,

Pembimbing 1



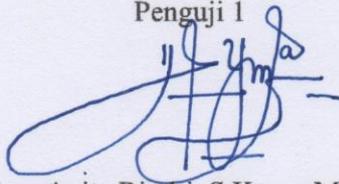
Ahmat Josi, S.Kom., M.Kom.
NIP. 198908202019031015

Pembimbing 2



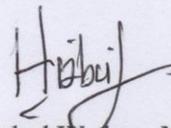
Putri Armilia Prayesy, M.Kom.
NIP. 199501042024062001

Penguji 1



Yang Agita Rindri, S.Kom., M.Eng.
NIP. 198609282022032003

Penguji 2



M. Hizbul Wathan, M.Kom.
NIP. 198904182024061001

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa : Rizqi Pratama NIM: 1062255

Dengan Judul : Pengembangan Sistem Berbasis Web untuk Digitalisasi Administrasi Desa Rias

Menyatakan bahwa laporan akhir ini adalah hasil kerja saya sendiri dan bukan merupakan plagiat. Pernyataan ini saya buat dengan sebenarnya dan bila ternyata di kemudian hari ternyata melanggar pernyataan ini, saya bersedia menerima sanksi yang berlaku.

Sungailiat, 21 Juli 2025

Nama Mahasiswa

Tanda Tangan

Rizqi Pratama



.....

ABSTRAK

Administrasi desa merupakan aspek vital dalam penyelenggaraan pemerintahan tingkat desa. Namun, di Desa Rias, Kecamatan Toboali, proses administrasi masih dilakukan secara manual sehingga menghambat efisiensi layanan, meningkatkan potensi kesalahan pencatatan, dan membatasi akses masyarakat terhadap dokumen penting. Untuk mengatasi permasalahan ini, dikembangkanlah sistem informasi administrasi berbasis web yang bertujuan untuk mendigitalisasi pelayanan surat-menyurat di desa. Sistem ini memungkinkan masyarakat untuk melakukan pengajuan surat secara daring, dilanjutkan dengan proses verifikasi dan tindak lanjut oleh operator desa, serta pemberitahuan status pengajuan yang dapat diakses melalui fitur notifikasi. Selain itu, sistem ini mendukung pengunduhan dokumen digital hasil pengajuan, pengelolaan akun pengguna oleh admin, manajemen jenis surat administrasi yang tersedia, serta pengarsipan surat yang dapat diakses oleh admin dan operator. Pengembangan sistem menggunakan metode Rapid Application Development (RAD) melalui tahapan analisis, desain, konstruksi, dan implementasi. Hasil pengujian dengan metode Black-Box menunjukkan bahwa seluruh fitur berjalan sesuai fungsinya. Sementara itu, pengujian User Acceptance Testing (UAT) memperoleh nilai sebesar 91,42% yang termasuk dalam kategori “Sangat Layak”. Dengan diterapkannya sistem ini, proses administrasi di Desa Rias menjadi lebih efisien, transparan, dan mudah diakses tanpa perlu kunjungan langsung ke kantor desa.

Kata kunci: Digitalisasi Administrasi, Pelayanan Surat, Sistem Berbasis Web

ABSTRACT

Village administration is a vital aspect of governance at the village level. However, in Rias Village, Toboali Subdistrict, administrative processes are still carried out manually, which hinders service efficiency, increases the risk of recording errors, and limits public access to important documents. To address these issues, a web-based administrative information system was developed to digitize letter-based public services in the village. This system enables citizens to submit letter requests online, which are then verified and followed up by village operators, with submission statuses delivered through a notification feature. Additionally, the system supports the downloading of digital documents, user account management by administrators, administrative letter type management, and archiving of completed letters accessible to admins and operators. The system was developed using the Rapid Application Development (RAD) method, which includes the stages of analysis, design, construction, and implementation. Testing using the Black-Box method showed that all features functioned as expected. Meanwhile, User Acceptance Testing (UAT) yielded a score of 91.42%, which falls into the “Highly Feasible” category. The implementation of this system has made the administrative service process in Rias Village more efficient, transparent, and accessible without requiring citizens to visit the village office in person.

Keywords: Administrative Digitalization, Letter Submission, Web-Based System

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan proyek akhir dan penyusunan laporan proyek akhir dengan judul “Pengembangan Sistem Berbasis Web untuk Digitalisasi Administrasi Desa Rias” dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan program studi serta sebagai bentuk implementasi ilmu dan keterampilan yang telah diperoleh selama masa perkuliahan di bangku pendidikan tinggi. Penulisan laporan ini tidak hanya bertujuan untuk memenuhi kewajiban akademik, tetapi juga sebagai wujud nyata dari penerapan teori ke dalam praktik dalam bidang yang telah dipelajari selama proses pembelajaran.

Dalam proses penyusunan laporan ini, penulis menyadari bahwa keberhasilan proyek ini tidak terlepas dari dukungan moral, bimbingan akademik, serta bantuan teknis dari berbagai pihak yang terlibat secara langsung maupun tidak langsung. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak I Made Andik Setiawan, M.Eng., Ph.D. sebagai Direktur Politeknik Manufaktur Negeri Bangka Belitung.
2. Ibu Yang Agita Rindri, S.Kom., M.Eng. sebagai Ketua Jurusan Informatika dan Bisnis.
3. Bapak Sidhiq Andriyanto, S.T., M.Kom. sebagai Koordinator Program Studi Sarjana Terapan Teknologi Rekayasa Perangkat Lunak.
4. Bapak Ahmat Josi, S.Kom., M.Kom. sebagai Dosen Pembimbing I yang telah dengan tulus dan penuh komitmen meluangkan waktu, tenaga, serta memberikan bimbingan dan masukan yang sangat berarti selama proses penyusunan dan penyelesaian proyek akhir ini.

5. Ibu Putri Armilia Prayesy, M.Kom. sebagai Dosen Pembimbing II yang telah memberikan saran, arahan, serta solusi yang konstruktif kepada penulis selama proses penyusunan laporan proyek akhir ini.
6. Pihak Kantor Desa Rias, yang telah memberikan data, informasi, serta kesempatan untuk melakukan observasi dan wawancara dalam rangka mendukung pengembangan sistem.
7. Ayah, Ibu, dan saudara yang senantiasa memberikan doa, dukungan moral, serta kasih sayang yang tiada henti, yang menjadi sumber kekuatan dan semangat selama penyusunan proyek akhir ini.
8. Rekan-rekan seperjuangan yang telah memberikan bantuan, semangat, serta saling mendukung dalam proses penyelesaian proyek akhir dan penyusunan laporan ini.

Penulis juga menyadari bahwa laporan ini masih jauh dari kata sempurna. Keterbatasan pengetahuan, pengalaman, serta waktu yang tersedia tentu menjadi faktor yang memengaruhi hasil akhir dari laporan ini. Oleh karena itu, penulis membuka diri terhadap segala bentuk kritik dan saran yang bersifat membangun demi penyempurnaan laporan ini di masa mendatang.

Akhir kata, penulis berharap agar laporan ini dapat memberikan manfaat bagi pembaca, serta menjadi bahan referensi dalam penelitian selanjutnya yang relevan dengan topik yang dibahas. Semoga laporan ini dapat memberikan kontribusi positif dalam dunia akademik maupun praktis.

Sungailiat, 21 Juli 2025



Rizqi Pratama

DAFTAR ISI

DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Proyek Akhir	3
1.4 Batasan Masalah.....	3
BAB II DASAR TEORI.....	4
2.1 Tinjauan Pustaka.....	4
2.2 Pengertian Administrasi.....	7
2.2.1 Pelayanan Masyarakat.....	8
2.2.2 Administrasi Surat.....	8
2.3 Visual Studio Code	9
2.4 PHP.....	11
2.5 Tailwind.....	11
2.6 Laragon.....	12
2.7 MySQL.....	13
2.8 Laravel.....	14
2.9 Website	15
2.10 Metode RAD	16
2.11 Perancangan Sistem.....	17

2.11.1	<i>Use Case Diagram</i>	17
2.11.2	<i>Activity Diagram</i>	18
2.11.3	<i>Class Diagram</i>	19
2.11.4	<i>Sequence Diagram</i>	20
BAB III METODE PELAKSANAAN		21
3.1	Diagram Alir	21
3.2	Pengumpulan Data	21
3.3	Analisis	22
3.4	Pengembangan	23
3.4.1	Perencanaan Kebutuhan	23
3.4.2	Desain Sistem	24
3.4.3	Pengembangan	63
3.4.4	Implementasi	67
3.5	Laporan	67
BAB IV PEMBAHASAN		69
4.1	Analisis Kebutuhan dan Perancangan Sistem	69
4.2	Rancangan Database	70
4.2.1	Database db_rias	71
4.2.2	Tabel <i>users</i>	72
4.2.3	Tabel <i>jenis_surat</i>	72
4.2.4	Tabel <i>pengajuan_surat</i>	73
4.2.5	Tabel <i>pengajuan_selesai</i>	74
4.2.6	Tabel <i>notifikasi</i>	75
4.3	Tampilan Antarmuka	76
4.3.1	Tampilan <i>Login</i>	76

4.3.2	Tampilan <i>Register</i>	76
4.3.3	Tampilan Verifikasi Email.....	77
4.3.4	Tampilan Halaman Utama.....	77
4.3.5	Tampilan <i>Form</i> Surat	79
4.3.6	Tampilan Status Pengajuan Surat.....	79
4.3.7	Tampilan Riwayat Pengajuan Surat	80
4.3.8	Tampilan Detail Riwayat Pengajuan Surat	81
4.3.9	Tampilan Profil Pengguna.....	82
4.3.10	Tampilan <i>Dashboard Admin</i>	83
4.3.11	Tampilan <i>User Management</i>	84
4.3.12	Tampilan Edit Data <i>User</i>	84
4.3.13	Tampilan Data Jenis Surat.....	85
4.3.14	Tampilan Tambah Jenis Surat	86
4.3.15	Tampilan Edit Jenis Surat.....	87
4.3.16	Tampilan Arsip Surat (<i>Admin</i>)	88
4.3.17	Tampilan Tambah Arsip Surat (<i>Admin</i>).....	88
4.3.18	Tampilan Edit Arsip Surat (<i>Admin</i>).....	89
4.3.19	Tampilan <i>Dashboard Operator</i>	90
4.3.20	Tampilan Antrian Pengajuan Surat.....	91
4.3.21	Tampilan Detail Pengajuan Surat.....	92
4.3.22	Tampilan Detail Proses Pengajuan Surat	93
4.3.23	Tampilan Arsip Surat (<i>Operator</i>)	94
4.3.24	Tampilan Tambah Arsip Surat (<i>Operator</i>)	95
4.3.25	Tampilan Edit Arsip Surat (<i>Operator</i>)	96

4.4	Implementasi	97
4.4.1	Pengujian Validasi Ahli Media.....	97
4.5	Hasil Kuesioner	109
4.6	Hasil Perhitungan <i>User Acceptance Testing</i> (UAT)	111
BAB V KESIMPULAN DAN SARAN.....		112
5.1	Kesimpulan.....	112
5.2	Saran.....	113
DAFTAR PUSTAKA		114
LAMPIRAN.....		116



DAFTAR TABEL

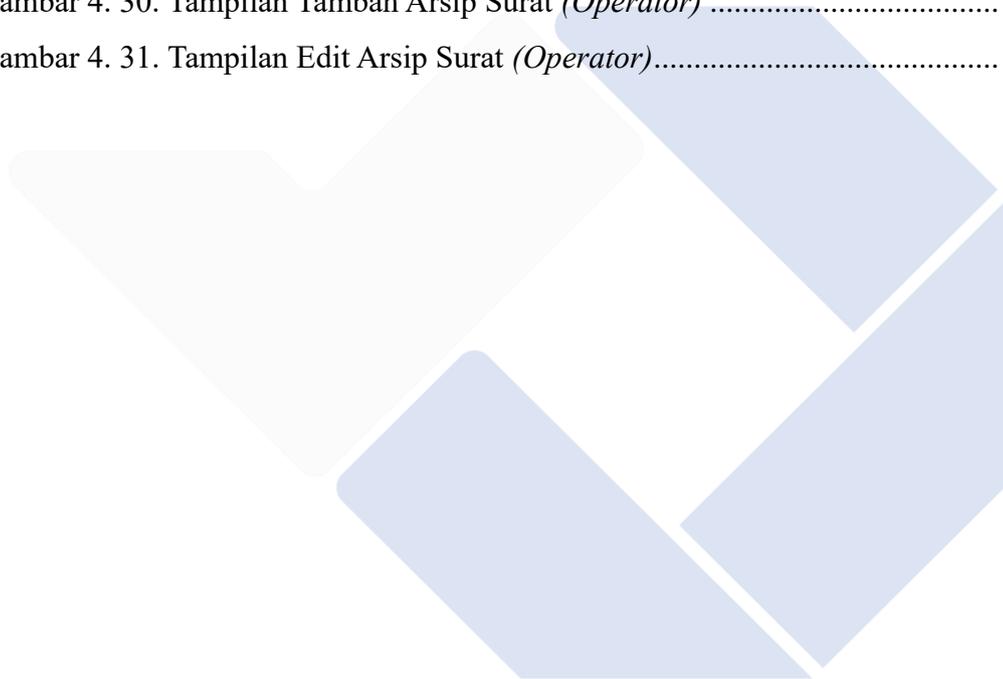
Tabel 2. 1. Tinjauan Pustaka	4
Tabel 2. 2. Simbol <i>Use Case Diagram</i>	17
Tabel 2. 3. Simbol <i>Activity Diagram</i>	18
Tabel 2. 4. Simbol <i>Class Diagram</i>	19
Tabel 2. 5. Simbol <i>Sequence Diagram</i>	20
Tabel 3. 1. Kriteria Kelayakan	66
Tabel 3. 2. Validasi <i>User Acceptance Testing (UAT)</i>	66
Tabel 4. 1. Validasi <i>Login Admin</i>	97
Tabel 4. 2. Validasi <i>Dashboard Admin</i>	98
Tabel 4. 3. Validasi <i>Data User</i>	98
Tabel 4. 4. Validasi Halaman <i>Data Jenis Surat</i>	99
Tabel 4. 5. Validasi <i>Tambah, Edit, Hapus Jenis Surat Admin</i>	100
Tabel 4. 6. Validasi Halaman <i>Arsip Surat Admin</i>	101
Tabel 4. 7. Validasi Halaman <i>Login Operator</i>	102
Tabel 4. 8. Validasi Halaman <i>Dashboard Operator</i>	102
Tabel 4. 9. Validasi Halaman <i>Data Pengajuan Surat Operator</i>	103
Tabel 4. 10. Validasi <i>Ubah Status Pengajuan Surat Operator</i>	103
Tabel 4. 11. Validasi <i>Dokumen Pengajuan Operator</i>	104
Tabel 4. 12. Validasi Halaman <i>Arsip Surat Operator</i>	105
Tabel 4. 13. Validasi Halaman <i>Login User</i>	106
Tabel 4. 14. Validasi Halaman <i>Formulir Pengajuan Surat</i>	107
Tabel 4. 15. Validasi <i>Status Pengajuan Surat</i>	107
Tabel 4. 16. Validasi Halaman <i>Riwayat Pengajuan Surat</i>	108
Tabel 4. 17. Validasi Halaman <i>Profil User</i>	108
Tabel 4. 18. Validasi Halaman <i>Edit Profil User</i>	109
Tabel 4. 19. Hasil <i>Kuesioner User</i>	110

DAFTAR GAMBAR

Gambar 3. 1. Diagram Alir.....	21
Gambar 3. 2. Metode RAD	23
Gambar 3. 3. Desain Halaman Utama.....	26
Gambar 3. 4. Desain Pengajuan Surat.....	27
Gambar 3. 5. Desain Pengajuan Surat.....	28
Gambar 3. 6. Desain Profil Pengguna.....	29
Gambar 3. 7. Desain Dashboard <i>Admin/Operator</i>	29
Gambar 3. 8. Desain Data <i>User</i>	30
Gambar 3. 9. Desain Edit Data <i>User</i>	31
Gambar 3. 10. Desain Data Surat.....	32
Gambar 3. 11. Desain Tambah/Edit Jenis Surat.....	32
Gambar 3. 12. Desain Data Pengajuan.....	33
Gambar 3. 13. Desain Detail Pengajuan Surat (<i>Pending</i>)	34
Gambar 3. 14. Desain Detail Pengajuan Surat (<i>Diproses</i>).....	35
Gambar 3. 15. Desain Arsip Surat.....	36
Gambar 3. 16. Desain Tambah Arsip Surat.....	37
Gambar 3. 17. <i>Use Case Diagram</i>	38
Gambar 3. 18. <i>Activity Diagram Register</i>	39
Gambar 3. 19. <i>Activity Diagram Login</i>	40
Gambar 3. 20. <i>Activity Diagram</i> Pengajuan Surat	41
Gambar 3. 21. <i>Activity Diagram</i> Status Pengajuan Surat	42
Gambar 3. 22. <i>Activity Diagram</i> Riwayat Pengajuan Surat.....	43
Gambar 3. 23. <i>Activity Diagram</i> Edit Profil	44
Gambar 3. 24. <i>Activity Diagram User Management</i>	45
Gambar 3. 25. <i>Activity Diagram</i> Data Jenis Surat	46
Gambar 3. 26. <i>Activity Diagram</i> Arsip Surat	47
Gambar 3. 27. <i>Activity Diagram</i> Validasi Pengajuan Surat	49

Gambar 3. 28. <i>Class Diagram</i>	50
Gambar 3. 29. <i>Sequence Diagram Register</i>	51
Gambar 3. 30. <i>Sequence Diagram Login</i>	52
Gambar 3. 31. <i>Sequence Diagram Pengajuan Surat</i>	53
Gambar 3. 32. <i>Sequence Diagram Status Pengajuan Surat</i>	54
Gambar 3. 33. <i>Sequence Diagram Riwayat Pengajuan Surat</i>	55
Gambar 3. 34. <i>Sequence Diagram Edit Profil User</i>	56
Gambar 3. 35. <i>Sequence Diagram User Management</i>	57
Gambar 3. 36. <i>Sequence Diagram Data Jenis Surat</i>	58
Gambar 3. 37. <i>Sequence Diagram Arsip Surat</i>	60
Gambar 3. 38. <i>Sequence Diagram Validasi Pengajuan Surat</i>	62
Gambar 4. 1. Database db_rias	71
Gambar 4. 2. Tabel <i>users</i>	72
Gambar 4. 3. Tabel <i>jenis_surat</i>	72
Gambar 4. 4. Tabel <i>pengajuan_surat</i>	73
Gambar 4. 5. Tabel <i>pengajuan_selesai</i>	74
Gambar 4. 6. Tabel <i>notifikasi</i>	75
Gambar 4. 7. Tampilan <i>Login</i>	76
Gambar 4. 8. Tampilan <i>Register</i>	76
Gambar 4. 9. Tampilan Verifikasi Email.....	77
Gambar 4. 10. Tampilan Halaman Utama.....	78
Gambar 4. 11. Tampilan <i>Form Surat</i>	79
Gambar 4. 12. Tampilan Status Pengajuan Surat	80
Gambar 4. 13. Tampilan Riwayat Pengajuan Surat	80
Gambar 4. 14. Tampilan Detail Riwayat Pengajuan Surat.....	81
Gambar 4. 15. Tampilan Profil Pengguna	82
Gambar 4. 16. Tampilan <i>Dashboard Admin</i>	83
Gambar 4. 17. Tampilan <i>User Management</i>	84
Gambar 4. 18. Tampilan Edit Data <i>User</i>	85
Gambar 4. 19. Tampilan Data Jenis Surat	85
Gambar 4. 20. Tampilan Tambah Jenis Surat.....	86

Gambar 4. 21. Tampilan Edit Jenis Surat.....	87
Gambar 4. 22. Tampilan Arsip Surat (<i>Admin</i>).....	88
Gambar 4. 23. Tampilan Tambah Arsip Surat (<i>Admin</i>).....	89
Gambar 4. 24. Tampilan Edit Arsip Surat (<i>Admin</i>).....	90
Gambar 4. 25. Tampilan <i>Dashboard Operator</i>	90
Gambar 4. 26. Tampilan Antrian Pengajuan Surat.....	91
Gambar 4. 27. Tampilan Detail Pengajuan Surat.....	92
Gambar 4. 28. Tampilan Detail Proses Pengajuan Surat.....	93
Gambar 4. 29. Tampilan Arsip Surat (<i>Operator</i>).....	94
Gambar 4. 30. Tampilan Tambah Arsip Surat (<i>Operator</i>).....	95
Gambar 4. 31. Tampilan Edit Arsip Surat (<i>Operator</i>).....	96



DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup.....	116
Lampiran 2 : Lembar Validasi Ahli.....	117
Lampiran 3 : Hasil Kuisisioner UAT	121
Lampiran 4 : Kuisisioner Penilaian Admin.....	124
Lampiran 5 : Kuisisioner Penilaian Operator	125
Lampiran 6 : Dokumentasi Penyerahan Website	126



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Administrasi surat-menyurat di desa merupakan salah satu aspek penting dalam penyelenggaraan pemerintahan di tingkat desa. Namun, di Desa Rias, Kecamatan Toboali, proses layanan administrasi berbasis surat masih dilakukan secara manual dan konvensional. Hal ini menyebabkan berbagai permasalahan, seperti keterlambatan pelayanan, potensi kesalahan pencatatan, serta kurangnya transparansi dan efisiensi dalam pengelolaan dokumen.

Di era digital saat ini, penggunaan teknologi informasi sudah seharusnya diintegrasikan ke dalam sistem pelayanan publik, termasuk di tingkat desa. Digitalisasi administrasi surat-menyurat di desa dapat menjadi solusi untuk meningkatkan efisiensi layanan, mempermudah akses masyarakat terhadap dokumen penting, serta mempercepat proses pengolahan data secara sistematis dan terdokumentasi. Selain itu, sistem berbasis web memungkinkan masyarakat untuk mengajukan permohonan administrasi persuratan secara daring tanpa harus datang langsung ke kantor desa.

Sistem Administrasi Desa diharapkan dapat membangun sistem yang efisien untuk mengelola data dan informasi desa, mempercepat proses administrasi, dan meningkatkan partisipasi masyarakat yang aktif dalam pengambilan keputusan desa. (Safrudin, et al., 2024). Pegawai desa dapat memberikan pelayanan tanpa perlu melakukan interaksi langsung secara tatap muka dengan masyarakat, sementara warga desa dapat mengajukan permohonan surat secara daring tanpa harus mendatangi kantor desa. Hal ini menjadikan proses pelayanan administrasi menjadi lebih cepat, efisien, dan praktis. (Fitriasari, et al., 2023).

Berdasarkan hasil observasi dan wawancara dengan perangkat Desa Rias, ditemukan bahwa sebagian besar warga mengalami kesulitan dalam mengakses

pelayanan administrasi persuratan karena waktu pelayanan yang terbatas dan ketergantungan pada prosedur manual.

Berdasarkan permasalahan diatas, dirancanglah proyek "Pengembangan Sistem Berbasis Web untuk Digitalisasi Administrasi Desa Rias". Sistem ini akan mencakup fitur utama seperti registrasi dan *login*, pengajuan surat administrasi secara digital, verifikasi dan tindak lanjut oleh *operator* desa, hingga pemberitahuan status pengajuan serta pengunduhan dokumen hasil. Sistem ini juga akan mendukung pengelolaan akun pengguna oleh *admin*, manajemen jenis surat administrasi yang tersedia, serta pengelolaan arsip surat yang dapat diakses oleh *admin* dan *operator*.

Dengan alur sistem yang terstruktur, diharapkan sistem ini dapat meningkatkan kualitas pelayanan publik di Desa Rias. Selain mempercepat proses pengelolaan surat-menyurat, sistem ini juga akan memberikan transparansi, akuntabilitas, dan kemudahan akses bagi seluruh warga desa. Dengan demikian, diharapkan pengembangan sistem ini dapat dijadikan sebagai model digitalisasi layanan administrasi surat di desa yang bisa diadopsi oleh desa-desa lain di wilayah Kabupaten Bangka Selatan.

1.2 Perumusan Masalah

Adapun perumusan masalah yang diangkat dalam proyek akhir ini adalah sebagai berikut:

- a. Bagaimana rancangan sistem *website* yang dapat digunakan untuk proses administrasi surat-menyurat di Desa Rias?
- b. Bagaimana sistem *website* dapat memfasilitasi masyarakat dalam mengajukan permohonan surat secara daring?
- c. Bagaimana sistem *website* dapat membantu perangkat desa dalam memproses dokumen administrasi surat-menyurat secara lebih efisien?

1.3 Tujuan Proyek Akhir

Adapun tujuan yang ingin dicapai melalui pengembangan proyek akhir ini adalah sebagai berikut:

- a. Mengembangkan sistem berbasis web untuk digitalisasi proses administrasi di Desa Rias.
- b. Merancang alur pengajuan surat secara daring guna mempermudah masyarakat tanpa harus datang langsung ke kantor desa.
- c. Membangun fitur verifikasi dan distribusi dokumen digital untuk meningkatkan efisiensi kerja perangkat desa dalam mengelola administrasi persuratan.

1.4 Batasan Masalah

Adapun batasan sistem yang dikembangkan dalam proyek akhir ini meliputi:

- a. Sistem hanya mencakup administrasi pelayanan pengajuan surat.
- b. Sistem tidak menangani fitur-fitur lain di luar ruang lingkup surat-menyurat, seperti pencatatan data kependudukan, laporan anggaran keuangan desa, kegiatan pembangunan, atau layanan lainnya.

BAB II DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka dalam penelitian ini berfokus pada sejumlah penelitian terdahulu yang relevan dengan topik yang diangkat. Beberapa poin yang dapat disimpulkan adalah sebagai berikut:

Tabel 2. 1. Tinjauan Pustaka

No	Judul	Kesimpulan
1	Implementasi Sistem Informasi Administrasi Desa Penggung Berbasis Website	Dari hasil penelitian yang telah dilakukan oleh (Safrudin, et al., 2024), dapat disimpulkan bahwa penerapan Sistem Informasi Administrasi Desa berbasis web mampu membantu aparat desa dalam mengelola data kependudukan yang sebelumnya dilakukan secara manual. Sistem ini mempercepat proses administrasi, meminimalkan kesalahan dalam data, serta mendorong transparansi dan keterlibatan masyarakat dalam proses pengambilan keputusan di tingkat desa.
2	Pengembangan Pelayanan Administrasi Aplikasi Website Desa Lontar	Dari hasil pengabdian yang telah dilakukan oleh (Fitriasari, et al., 2023), dapat disimpulkan bahwa dengan memanfaatkan e-government, aplikasi web Desa Lontar dapat memudahkan masyarakat untuk mengajukan surat tanpa harus datang

3	<p>Pengembangan Sistem Informasi Pelayanan Surat Menyurat (SIRA) Berbasis <i>Website</i> Dengan Menggunakan Framework Codeigniter</p>	<p>langsung ke kantor desa. Selain itu, pegawai administrasi desa dapat menyediakan layanan dengan lebih cepat dan efektif tanpa harus bertemu secara pribadi. Selain itu, sistem yang terintegrasi memungkinkan <i>operator</i> desa untuk dengan mudah mengakses data administrasi yang disimpan di database dengan menggunakan <i>username</i> dan <i>password</i> yang telah didaftarkan di database.</p> <p>Dari hasil implementasi yang dilakukan oleh (Syaebani, Tyasmala, Maulani, Utami, & Wahyuni, 2021), dapat disimpulkan bahwa <i>website</i> pelayanan surat menyurat Kelurahan Mendawai memudahkan warga untuk mengajukan permohonan surat secara mandiri tanpa harus pergi ke kantor kelurahan. Akun warga masing-masing memungkinkan Anda melacak seluruh proses, mulai dari pengajuan hingga surat yang ditandatangani oleh kepala desa. Selain itu, melalui sistem digital yang telah terintegrasi, kelurahan dapat memberikan layanan secara online dan membuat arsip surat lebih mudah diurus.</p>
4	<p>Perancangan Sistem Informasi Pengelolaan Surat Menyurat Berbasis Web Desa Sumberkarang</p>	<p>Dari hasil penelitian yang telah dilakukan oleh (Umami, Adam, & Winarti, 2022), dapat disimpulkan bahwa Sistem informasi yang dibangun untuk mengelola surat menyurat Desa Sumberkarang telah</p>

5 Rancang Bangun Sistem Informasi Layanan Administrasi Desa Berbasis Web Di Desa Dukuh

berjalan dengan baik dan memenuhi kebutuhan. Dengan demikian, sistem ini dapat digunakan untuk mengelola surat masuk dan surat keluar secara lebih terorganisir dan efisien, yang akan menghasilkan peningkatan kualitas layanan administrasi desa.

Dari hasil implementasi yang telah dilakukan oleh (Romadhon & Maryam, 2023), dapat disimpulkan bahwa masyarakat sekarang dapat mengajukan surat secara *online* tanpa harus pergi ke kantor desa berkat pengembangan Sistem Informasi Layanan Administrasi Desa berbasis web di Desa Dukuh. Perangkat desa juga dapat menyediakan pelayanan administrasi secara lebih sistematis. Hasil pengujian metode *blackbox* menunjukkan bahwa semua fitur sistem telah berjalan sesuai dengan rencananya.

Berdasarkan analisis dari beberapa penelitian sebelumnya, penulis merancang dan mengembangkan sebuah sistem administrasi berbasis web yang bertujuan untuk mendigitalisasi proses pelayanan surat-menyurat di Desa Rias. Sistem ini menggabungkan berbagai fitur penting dan dikembangkan lebih lanjut untuk mengatasi keterbatasan pada sistem terdahulu. Salah satu keunggulan dari sistem ini adalah fleksibilitas dalam pengelolaan jenis surat, di mana *admin* dapat menambah, mengedit, atau menghapus jenis surat sesuai kebutuhan pelayanan. Sistem juga menyediakan fitur arsip surat yang dapat diakses oleh *operator* dan *admin*. Setelah surat selesai diproses, sistem akan otomatis mengunggah dokumen ke dalam arsip. Di sisi lain, *admin* dan *operator* dapat menambah arsip secara

manual. Fitur ini dilengkapi pencarian berdasarkan nama pengaju dan jenis surat, sehingga memudahkan pelacakan dokumen secara terstruktur. Untuk menjaga keamanan dan validitas data pengguna, sistem juga menerapkan proses verifikasi email bagi warga yang baru melakukan pendaftaran akun.

Selain itu, sistem ini dilengkapi dengan tampilan *dashboard* interaktif, yang menampilkan data visual seperti total pengguna, jumlah pengajuan surat hari ini, serta statistik pengajuan perbulan. Fitur manajemen pengguna juga disediakan, lengkap dengan fungsi pencarian dan filter untuk mempermudah pengelolaan data warga. Sistem ini tidak hanya mempermudah proses pengajuan surat oleh masyarakat, namun juga membantu perangkat desa dalam mengelola data administrasi secara efisien, terstruktur, dan terintegrasi dalam satu platform digital yang responsif.

2.2 Pengertian Administrasi

Administrasi didefinisikan sebagai suatu kegiatan ketatausahaan yang mencakup pencatatan, pendokumentasian, penggandaan, dan penyimpanan data, sekaligus bagian dari proses kerjasama antarindividu untuk mencapai tujuan organisasi yang telah ditetapkan (Hidayat & Fresh Yani, 2022). Administrasi merupakan suatu proses yang melibatkan berbagai aktivitas pengelolaan untuk mencapai tujuan tertentu secara terstruktur. Administrasi tidak hanya terbatas pada pencatatan atau dokumentasi, tetapi juga mencakup kegiatan yang bersifat menyeluruh, seperti perencanaan, pengorganisasian, pengarahan, koordinasi, dan pengendalian terhadap sumber daya yang tersedia, baik itu manusia, informasi, maupun alat bantu lainnya.

Dalam proyek sistem *website* administrasi Desa Rias, administrasi yang dimaksud secara khusus mengacu pada administrasi pelayanan surat-menyurat, yaitu proses pengajuan, pengelolaan, verifikasi, hingga pengarsipan dokumen atau surat yang berkaitan dengan kebutuhan masyarakat di Desa Rias. Digitalisasi proses ini melalui sistem berbasis web bertujuan untuk meningkatkan efisiensi pelayanan,

mengurangi ketergantungan pada prosedur manual, dan mempercepat alur komunikasi antara masyarakat dengan perangkat desa.

2.2.1 Pelayanan Masyarakat

Menurut Undang-Undang Nomor 25 Tahun 2009 tentang Pelayanan Publik, pelayanan publik dapat diartikan sebagai suatu bentuk kegiatan atau serangkaian aktivitas yang dilakukan secara sistematis dan bertanggung jawab oleh penyelenggara layanan, baik di tingkat pusat maupun daerah, dalam rangka memenuhi kebutuhan masyarakat. Pelayanan ini mencakup penyediaan barang, jasa, maupun layanan administratif yang diberikan kepada setiap warga negara dan penduduk, sesuai dengan ketentuan dan peraturan yang berlaku. Tujuan dari pelayanan publik adalah untuk menjamin hak-hak dasar masyarakat dalam memperoleh akses terhadap layanan yang transparan, cepat, dan akuntabel yang telah ditetapkan dalam peraturan perundang-undangan.

2.2.2 Administrasi Surat

Administrasi surat merupakan bagian penting dalam sistem tata kelola pemerintahan desa yang berperan dalam pencatatan, pengelolaan, dan penyampaian dokumen resmi kepada masyarakat. Administrasi surat tidak hanya berkaitan dengan proses pengiriman dan penerimaan surat, tetapi juga mencakup pengolahan dokumen administrasi yang dibutuhkan warga dalam berbagai keperluan. Surat-surat yang dikeluarkan oleh pihak desa berfungsi sebagai dokumen legal yang menyatakan atau membuktikan status, kondisi, atau pernyataan tertentu dari warga, yang biasanya dibutuhkan sebagai persyaratan administrasi lanjutan di instansi lain.

Di Kantor Desa Rias, terdapat sembilan jenis pelayanan surat yang umum diajukan oleh masyarakat, antara lain:

- a. Surat Keterangan Domisili: Digunakan sebagai bukti tempat tinggal resmi warga.
- b. Surat Keterangan Kehilangan: Sebagai bukti kehilangan barang atau dokumen, untuk keperluan pengurusan ulang.

- c. Surat Keterangan Penghasilan Orang Tua: Dibutuhkan untuk pengurusan beasiswa atau keperluan administratif lain yang memerlukan informasi pendapatan keluarga.
- d. Surat Keterangan Usaha: Digunakan untuk menyatakan keberadaan usaha warga yang berada di wilayah desa.
- e. Surat Keterangan Tidak Mampu: Digunakan untuk keperluan bantuan sosial atau keringanan biaya tertentu.
- f. Surat Keterangan Pindah: Diperlukan saat warga hendak pindah tempat tinggal secara administratif.
- g. Surat Keterangan Beda Nama: Digunakan untuk menyatakan adanya perbedaan penulisan nama pada dokumen tertentu.
- h. Surat Keterangan Nikah: Sebagai pelengkap administrasi pernikahan nonresmi atau adat sebelum pengesahan secara negara.
- i. Surat Keterangan Pengantar Nikah: digunakan sebagai syarat pengajuan pernikahan ke Kantor Urusan Agama (KUA).

2.3 Visual Studio Code

Menurut (Huda N. , 2022), Visual Studio Code adalah aplikasi yang dibuat oleh Microsoft, dapat digunakan di semua perangkat *desktop* secara gratis dan memiliki banyak fitur dan ekstensi yang membuatnya menjadi pilihan utama pengembang. Bahkan hampir semua sistem operasi, seperti Windows, Mac OS, dan Linux, didukung oleh Visual Studio Code.

Berikut ini adalah beberapa fitur unggulan yang dimiliki oleh Visual Studio Code:

1. *Basic Editing*

Visual Studio Code memiliki fitur editing dasar yang mendukung aktivitas penulisan kode secara efisien. Dibandingkan dengan perangkat lunak sejenis, fitur ini menjadi salah satu yang paling lengkap. Untuk kenyamanan pengguna, Visual Studio Code menawarkan fitur seperti format kode dan *auto-save*. Selain itu, penggunaan *hotkey*, yang dapat melakukan perintah tertentu, seperti menyalin dan menempelkan beberapa baris kode, mencari

keyword atau nama file, dan berbagai *shortcut* keyboard lainnya, mempercepat dan memudahkan proses pengeditan kode.

2. *Debugging*

Fitur *debugging* Visual Studio Code membantu pengguna mengelola program yang sedang dikembangkan dengan mempercepat proses penyuntingan, penyusunan, dan eksekusi kode berulang. Jika terjadi kesalahan penulisan kode, peringatan kesalahan ditampilkan dengan cukup jelas sehingga pengguna dapat dengan mudah menemukan lokasi kesalahan. Secara *default*, Visual Studio Code memiliki kemampuan *debugging* untuk bahasa pemrograman JavaScript, terutama Node.js. Namun, bagi pengguna bahasa pemrograman lain, ada opsi untuk mengunduh ekstensi tambahan melalui extension marketplace.

3. *Extension Marketplace*

Dengan fitur *extension marketplace*, pengguna dapat menginstal berbagai alat atau ekstensi dengan cepat tanpa perlu melakukan konfigurasi manual. Fitur ini memungkinkan pengguna untuk mempelajari dan menambahkan komponen tambahan di luar fitur bawaan Visual Studio Code. Beberapa ekstensi populer yang dapat diunduh termasuk ESLint, Prettier, Live Server, Vetur, dan Debuggers.

4. IntelliSense

Menulis kode dari awal hingga akhir secara manual memerlukan banyak waktu dan tenaga. Namun, Visual Studio Code memiliki fitur IntelliSense yang secara otomatis memberikan saran atau prediksi penulisan kode berdasarkan bahasa pemrograman atau basis data yang digunakan. Jika pengguna membutuhkan dukungan untuk bahasa pemrograman tertentu, seperti HTML, CSS, JavaScript, dan TypeScript.

5. Integrasi dengan GitHub

Visual Studio Code menyediakan dukungan integrasi dengan berbagai platform manajemen proyek, salah satunya adalah GitHub, yang saat ini merupakan salah satu platform kolaborasi pengembangan perangkat lunak paling populer. Melalui integrasi ini, pengguna dapat membagikan kode dan

bekerja sama dengan anggota tim secara terpusat dalam satu repositori. Untuk memanfaatkan fitur ini, pengguna diwajibkan memiliki akun GitHub. Setelah akun tersedia, pengguna dapat menginstal ekstensi seperti GitHub Pull Request and Issues melalui extension marketplace. Ekstensi tersebut memungkinkan pengguna untuk terhubung dan berinteraksi secara langsung dengan repositori GitHub melalui antarmuka Visual Studio Code.

2.4 PHP

PHP merupakan singkatan rekursif dari "PHP: *Preprocessor Hypertext*," adalah bahasa pemrograman yang digunakan secara luas untuk menangani pembuatan dan pengembangan situs web (Akbar, 2021). Fungsi PHP adalah membuat atau mengembangkan situs web statis atau dinamis serta aplikasi web.

Berikut merupakan keunggulan dari PHP:

- a. PHP memungkinkan pengembangan situs web yang bersifat dinamis, sehingga konten dapat disesuaikan berdasarkan interaksi pengguna.
- b. Bersifat *open source*, yang berarti dapat digunakan secara bebas dan tanpa biaya oleh siapa pun.
- c. Dapat dijalankan pada berbagai sistem operasi selama perangkat tersebut memiliki web *browser*, termasuk perangkat *mobile*.
- d. Mendukung integrasi dengan berbagai jenis basis data, seperti MySQL, Oracle, PostgreSQL, dan lainnya, yang memberikan fleksibilitas dalam pengelolaan data.
- e. Tidak memerlukan proses kompilasi (*compile*) dalam penggunaannya, sehingga dapat langsung dijalankan setelah penulisan kode selesai.

2.5 Tailwind

Berdasarkan pendapat (Prasatya, 2024), Tailwind adalah *framework* CSS dengan utilitas yang memungkinkan pengembang membuat desain yang rumit dengan mudah. Berbeda dengan *framework* CSS lainnya yang menyediakan komponen siap pakai, Tailwind menawarkan kelas utilitas yang dapat digunakan untuk menyesuaikan tampilan elemen HTML dengan cepat.

Berdasarkan ringkasan yang disampaikan oleh (RevoU, n.d.) dari sumber Hubspot, Fungsi Tailwind CSS adalah sebagai berikut:

- a. Mengurangi penulisan CSS khusus: *developer* dapat mengatur tampilan elemen dengan menggunakan kelas-kelas yang sudah ada di HTML, dan mereka juga dapat membuat desain unik tanpa menulis CSS.
- b. Bisa menjaga ukuran file CSS tetap kecil: Jika pengembang tidak menggunakan *framework* seperti Tailwind, mereka harus terus menulis CSS saat menambahkan fitur dan komponen baru, yang berarti file CSS akan lebih besar dan berat. Namun, dengan menggunakan utilitas seperti *flexbox* dan *padding* di Tailwind, sebagian besar *style* dapat digunakan kembali, sehingga tidak perlu menulis CSS baru.
- c. Tidak perlu menciptakan nama kelas sendiri: pengembang tidak perlu bingung mencari nama kelas untuk *style* dan komponen tertentu atau bahkan mengingat nama yang rumit.
- d. Dapat melakukan perubahan dengan tingkat risiko yang lebih rendah: pada pendekatan konvensional, modifikasi terhadap berkas CSS berpotensi menyebabkan gangguan pada tampilan keseluruhan situs. Namun, penggunaan kelas-kelas utilitas langsung dalam HTML bersifat lokal, sehingga perubahan yang dilakukan hanya berdampak pada elemen terkait tanpa memengaruhi komponen lain di dalam halaman web.

2.6 Laragon

(Puguh, 2023) menyatakan bahwa Laragon merupakan sebuah aplikasi yang digunakan untuk membangun dan menjalankan program atau aplikasi pada server lokal (komputer). Salah satu keunggulan utama Laragon adalah kelengkapan fiturnya yang melebihi aplikasi sejenis, serta kemudahan dalam pengoperasiannya karena didukung oleh antarmuka grafis (GUI) yang intuitif. Dengan Laragon, pengguna dapat mengembangkan aplikasi menggunakan Git serta melakukan manajemen basis data melalui antarmuka PHPMyAdmin. Selain itu, Laragon juga

mendukung pengembangan aplikasi dengan berbagai teknologi populer, seperti Node.js dengan MongoDB, Python dengan Django atau Flask dan PostgreSQL, serta bahasa pemrograman lainnya seperti Ruby, Java, dan Go.

Dalam proyek Sistem website administrasi persuratan di Desa Rias, Laragon dimanfaatkan sebagai lingkungan pengembangan lokal untuk membangun dan menguji sistem. Dengan menggunakan Laragon, proses instalasi *framework* seperti Laravel menjadi lebih cepat dan efisien. Pengelolaan dapat dilakukan secara lokal melalui PHPMyAdmin. Hal ini menjadikan Laragon sebagai solusi yang ideal dalam tahap pengembangan awal sistem administrasi desa yang responsif, aman, dan terintegrasi dengan berbagai fitur modern.

2.7 MySQL

Menurut (Fauzi, 2024), MySQL merupakan salah satu sistem manajemen basis data relasional (*Relational Database Management System/RDBMS*) bersifat *open source* yang digunakan untuk menyimpan dan mengelola data secara efisien. MySQL menggunakan bahasa pemrograman SQL (*Structured Query Language*), yang berfungsi untuk melakukan operasi terhadap data, seperti pengambilan, pembaruan, penghapusan, serta manipulasi data lainnya dalam basis data relasional. Sebagai sistem basis data relasional, MySQL menyimpan data dalam bentuk tabel yang terdiri dari baris dan kolom, dan diatur dalam sebuah skema. Skema ini berperan dalam mendefinisikan struktur penyimpanan data serta hubungan antar tabel dalam database.

Berikut ini adalah beberapa fungsi utama dari MySQL:

- a. Penyimpanan Data: MySQL berfungsi sebagai media penyimpanan data dalam bentuk tabel, sesuai dengan prinsip database relasional.
- b. Pengambilan Data: Pengguna dapat mengambil data yang telah disimpan menggunakan perintah *query* sesuai dengan kebutuhan informasi.
- c. Pembaruan Data: Data yang terdapat dalam basis data dapat diperbarui atau dimodifikasi apabila terjadi perubahan atau penyesuaian informasi.

- d. Penghapusan Data: MySQL menyediakan fitur untuk menghapus data yang sudah tidak diperlukan, guna menjaga relevansi dan efisiensi penyimpanan.
- e. Pencarian Data: Sistem memungkinkan pencarian data berdasarkan kriteria tertentu, baik melalui nilai spesifik, rentang nilai, maupun kata kunci, sehingga memudahkan proses pencarian informasi secara akurat.
- f. Pengelolaan Hak Akses: MySQL menyediakan sistem manajemen hak akses bagi pengguna. *Administrator* dapat mengatur hak akses setiap pengguna sesuai peran masing-masing, seperti hanya membaca data, menambah data, atau menghapus data, guna menjaga keamanan dan integritas data dalam sistem.

2.8 Laravel

Laravel merupakan salah satu *framework* PHP *open source* yang dirancang untuk memudahkan dan mempercepat proses pengembangan aplikasi web (Sutisna, 2024). Laravel merupakan salah satu *framework* PHP yang menyediakan berbagai fitur bawaan untuk mendukung pengembangan aplikasi web. Beberapa fitur utama yang ditawarkan antara lain autentikasi, pengelolaan *routing*, serta manajemen basis data. Keberadaan fitur-fitur tersebut dapat mempercepat proses pengembangan aplikasi dan meningkatkan efisiensi kerja pengembang. Laravel menerapkan arsitektur *Model-View-Controller* (MVC), yang berfungsi untuk memisahkan logika bisnis, tampilan antarmuka, dan pengelolaan data. Pendekatan ini membantu menghasilkan struktur kode yang lebih terorganisir, rapi, dan lebih mudah untuk dikembangkan atau dipelihara dalam jangka panjang.

Berikut beberapa fitur utama dari Laravel:

- a. *Eloquent ORM*: *Eloquent* merupakan sistem *Object-Relational Mapping* (ORM) bawaan Laravel yang menyediakan cara intuitif dan elegan dalam berinteraksi dengan basis data. Dengan *Eloquent*, pengembang dapat memanipulasi data menggunakan sintaksis berbasis objek tanpa perlu menulis *query* SQL secara langsung.

- b. *Routing*: Laravel menyediakan sistem *routing* yang fleksibel dan mudah diimplementasikan. Melalui fitur ini, pengembang dapat dengan cepat menentukan rute atau *endpoint* dalam aplikasi dengan cara yang bersih dan mudah dibaca.
- c. *Blade Templating Engine*: *Blade* adalah mesin *templating* ringan milik Laravel yang memungkinkan pembuatan antarmuka pengguna (*user interface*) secara dinamis. *Blade* mendukung penggunaan sintaks yang sederhana serta memungkinkan pembuatan komponen tampilan yang dapat digunakan kembali.
- d. *Middleware*: *Middleware* pada Laravel digunakan untuk memproses dan memfilter HTTP request yang masuk sebelum diteruskan ke logika aplikasi. Fitur ini sangat bermanfaat dalam mengelola autentikasi, otorisasi, serta validasi input pengguna secara efisien.
- e. *Artisan CLI*: *Artisan* adalah antarmuka baris perintah (*Command Line Interface*) milik Laravel yang menyediakan berbagai perintah bawaan untuk mempercepat proses pengembangan. Dengan *Artisan*, pengembang dapat membuat file seperti *controller*, *model*, dan *migration* hanya dengan satu baris perintah.
- f. *Migration* dan *Seeders*: Laravel menyediakan sistem *migration* untuk mengelola struktur skema basis data secara terkontrol dan konsisten. Selain itu, Laravel juga menyediakan fitur *seeder* untuk mengisi basis data dengan data awal atau data *dummy* yang berguna dalam proses pengujian dan pengembangan.

2.9 Website

Berdasarkan informasi yang diperoleh dari situs (Biznetgio, 2025), *Website* adalah kumpulan halaman yang saling berhubungan yang terhubung ke satu domain dan dapat diakses melalui *browser* internet seperti Chrome, Firefox, atau Safari. Setiap halaman di dalam *website* mengandung berbagai jenis informasi, seperti teks,

gambar, video, atau audio, yang disusun secara sistematis dan membuatnya mudah diakses.

Dalam proyek sistem *website* administrasi persuratan di Desa Rias, *website* berperan sebagai media utama dalam mendigitalisasi proses pelayanan administrasi kepada masyarakat. Melalui *website* ini, masyarakat dapat melakukan pengajuan surat secara daring tanpa harus datang langsung ke kantor desa. Selain itu, pengelolaan data administrasi seperti data warga, jenis surat, serta riwayat pengajuan dapat dilakukan secara terpusat dan efisien. Dengan desain yang responsif, aman, dan terintegrasi dengan berbagai fitur modern, *website* ini menjadi solusi yang tepat dalam meningkatkan kualitas pelayanan publik di tingkat desa.

2.10 Metode RAD

Rapid Application Development (RAD) merupakan salah satu model proses pengembangan perangkat lunak yang menekankan kecepatan dalam membangun sistem melalui pendekatan iteratif dan *prototyping* yang intensif. Menurut (Musyaffa, 2021), Metode pengembangan perangkat lunak *Rapid Application Development* (RAD) adalah metode pengembangan perangkat lunak yang berfokus pada siklus pengembangan dalam waktu yang singkat.

Adapun tahapan umum dalam model RAD meliputi:

1. Perencanaan Kebutuhan

Tahap ini bertujuan untuk mengidentifikasi dan menganalisis kebutuhan sistem secara menyeluruh, serta merumuskan tujuan proyek yang akan dicapai.

2. Perancangan Antarmuka Pengguna (*Design*)

Pada tahap ini dilakukan perancangan awal sistem melalui pembangunan prototipe secara cepat.

3. Pengembangan

Tahap ini meliputi pengembangan sistem secara iteratif berdasarkan prototipe yang telah disetujui. Umpan balik dari pengguna digunakan untuk menyempurnakan fitur-fitur dan fungsionalitas sistem.

4. Implementasi

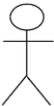
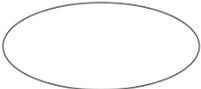
Pada tahap ini, sistem yang telah dikembangkan diimplementasikan ke dalam lingkungan operasional yang sebenarnya.

2.11 Perancangan Sistem

2.11.1 Use Case Diagram

Use Case Diagram merupakan salah satu jenis diagram dalam yang digunakan untuk menggambarkan hubungan antara aktor (pengguna) dan fungsionalitas sistem secara menyeluruh. Diagram ini menyajikan bagaimana pengguna eksternal (seperti *admin*, *operator*, dan *user*) berinteraksi dengan sistem dalam berbagai skenario penggunaan yang mungkin terjadi.

Tabel 2. 2. Simbol *Use Case Diagram*

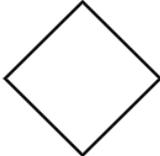
Simbol	Nama	Uraian
	<i>Actor</i>	Merupakan entitas pengguna yang berinteraksi dengan sistem dan memiliki peran tertentu.
	<i>Use Case</i>	Mengilustrasikan aksi atau layanan yang dilakukan sistem sebagai respons atas interaksi aktor.
<u><<include>></u>	<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> selalu menyertakan perilaku dari <i>use case</i> lain sebagai bagian dari prosesnya.
<u><<extend>></u>	<i>Extend</i>	Digunakan untuk mendefinisikan skenario opsional yang dapat memperluas fungsi utama pada <i>use case</i> tertentu.

	<i>Association</i>	Menggambarkan hubungan atau keterkaitan antara aktor dan <i>use case</i> dalam sistem.
	<i>Generalization</i>	Menyatakan hubungan hierarkis antara aktor yang menunjukkan turunan dengan perilaku atau tanggung jawab yang lebih spesifik.

2.11.2 Activity Diagram

Activity diagram adalah salah satu jenis diagram yang digunakan untuk memodelkan alur kerja atau proses bisnis dari suatu sistem. Diagram ini menggambarkan urutan aktivitas atau tindakan, termasuk kondisi percabangan, pengulangan, dan paralelisme yang mungkin terjadi dalam proses sistem.

Tabel 2. 3. Simbol *Activity Diagram*

Simbol	Nama	Uraian
	<i>Start State</i>	Menandakan titik awal dimulainya proses atau alur aktivitas dalam sistem.
	<i>Activity</i>	Menunjukkan suatu tindakan atau proses yang terjadi dalam sistem.
	<i>Decision</i>	Menyajikan titik percabangan logika yang bergantung pada suatu kondisi (benar atau salah).
	<i>Transition State</i>	Mengilustrasikan perpindahan atau hubungan alur dari satu aktivitas ke aktivitas lain.



Swimlane

Menjelaskan pembagian aktivitas berdasarkan peran atau tanggung jawab yang berbeda.



End State

Merepresentasikan titik akhir dari serangkaian aktivitas dalam sistem.

2.11.3 Class Diagram

Class diagram adalah diagram yang digunakan untuk merepresentasikan struktur sistem secara statis dalam bentuk kelas-kelas beserta atribut, metode, dan relasi antar kelas. Diagram ini berfungsi sebagai dasar untuk mendefinisikan blueprint atau kerangka dasar dari perangkat lunak yang akan dibangun.

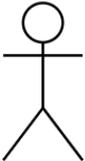
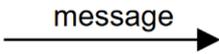
Tabel 2. 4. Simbol *Class Diagram*

Simbol	Nama	Uraian
	<i>Class</i>	Menunjukkan kelompok objek yang memiliki atribut dan metode yang serupa.
	<i>Association</i>	Menyatakan relasi tetap antara dua buah kelas dalam struktur sistem.
	<i>Aggregation</i>	Menjelaskan relasi di mana sebuah kelas terdiri atas komponen dari kelas lain, namun keduanya tetap dapat berdiri sendiri.
	<i>Generalization</i>	Menunjukkan relasi hierarki antar kelas di mana satu kelas merupakan bentuk umum dari kelas lain yang lebih spesifik.

2.11.4 Sequence Diagram

Sequence diagram adalah diagram yang digunakan untuk menggambarkan bagaimana objek-objek dalam sistem saling berinteraksi satu sama lain dalam urutan waktu tertentu. Fokus utamanya adalah pada kronologi pesan atau interaksi antar objek.

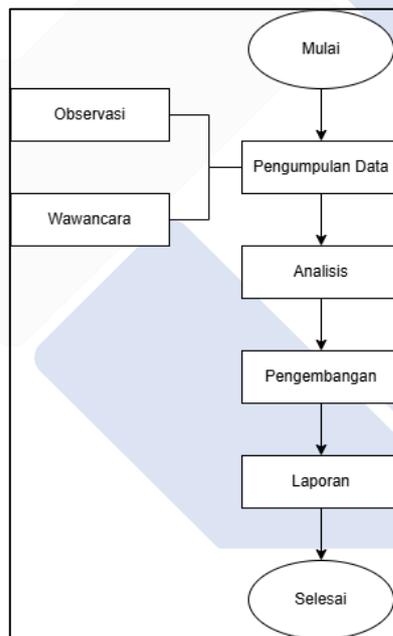
Tabel 2. 5. Simbol *Sequence Diagram*

Simbol	Nama	Uraian
	<i>Actor</i>	Merepresentasikan entitas luar (pengguna atau sistem lain) yang melakukan interaksi dengan sistem.
	<i>Object</i>	Menunjukkan entitas dalam sistem yang memiliki tanggung jawab tertentu dan terlibat dalam alur komunikasi.
	<i>Lifeline</i>	Menampilkan durasi keberadaan suatu objek selama proses berlangsung dalam diagram.
	<i>Activation Box</i>	Menandakan periode saat suatu objek sedang menjalankan operasi atau fungsi tertentu.
	<i>Message</i>	Menyimbolkan pertukaran informasi atau instruksi antar objek dalam urutan tertentu.
	<i>Alternative</i>	Merepresentasikan percabangan logika atau kondisi tertentu dalam alur interaksi antar objek.

BAB III METODE PELAKSANAAN

3.1 Diagram Alir

Diagram alir digunakan untuk menggambarkan tahapan-tahapan dalam proses pengembangan sistem berbasis web pada penelitian ini. Diagram ini berfungsi untuk memberikan gambaran alur kerja sistem secara keseluruhan, mulai dari interaksi pengguna hingga pemrosesan data oleh sistem. Berikut merupakan rancangan diagram alir pada penelitian “Pengembangan Sistem Berbasis Web untuk Digitalisasi Administrasi Desa Rias”:



Gambar 3. 1. Diagram Alir

3.2 Pengumpulan Data

Dalam penelitian ini, pengumpulan data dilakukan melalui beberapa metode guna memperoleh informasi yang relevan dan mendukung proses pengembangan sistem berbasis web untuk digitalisasi administrasi desa. Adapun metode pengumpulan data yang digunakan meliputi:

1. Observasi

Metode observasi dilakukan dengan mendatangi langsung kantor desa guna mengamati proses administrasi yang berlangsung, khususnya dalam pengelolaan surat menyurat, kemudian mencatat dan mendokumentasikan alur kerja administrasi mulai dari pengajuan surat oleh warga, hingga tahap proses surat selesai.

2. Wawancara

Metode wawancara dilakukan dengan melakukan sesi tanya jawab bersama perangkat desa yang terlibat langsung dalam proses administrasi, seperti sekretaris dan *operator*. Wawancara ini bertujuan untuk menggali informasi mengenai kendala yang dihadapi dalam proses administrasi manual, serta kebutuhan sistem yang diharapkan untuk menunjang efektivitas dan efisiensi kerja. Hasil wawancara kemudian dijadikan acuan dalam perancangan fitur-fitur utama pada sistem berbasis web yang dikembangkan.

3.3 Analisis

Analisis kebutuhan dilakukan guna memperoleh data yang relevan terkait sistem yang akan dirancang. Analisis dilakukan berdasarkan data yang diperoleh melalui wawancara dengan perangkat desa dan hasil observasi terhadap proses administrasi yang berlangsung di kantor desa. Hasil dari tahap ini menjadi dasar dalam perancangan sistem yang sesuai dengan kebutuhan di lapangan serta mendukung proses digitalisasi administrasi secara efektif dan efisien.

Adapun aspek-aspek yang dianalisis dalam penelitian ini meliputi:

1. Analisis Kebutuhan Pengguna

- a. Mengidentifikasi target pengguna utama, yaitu warga desa dan perangkat desa, serta memahami kebutuhan mereka dalam melakukan proses administrasi secara cepat dan efisien melalui sistem digital.
- b. Menganalisis antarmuka sistem agar ramah pengguna, mudah dioperasikan oleh semua kalangan, termasuk warga dengan literasi digital yang terbatas.

2. Analisis Fitur Sistem

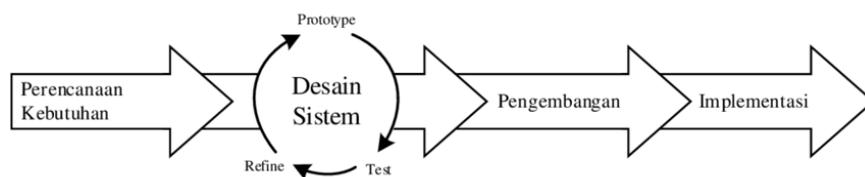
- a. Menentukan fitur utama yang akan diterapkan pada sistem, seperti kemampuan untuk menambah, mengedit, dan menghapus jenis surat agar lebih fleksibel dan dapat disesuaikan dengan kebutuhan administrasi desa.
- b. Sistem dilengkapi dengan fitur arsip surat untuk menyimpan dokumen surat yang telah diproses secara terorganisir guna mempermudah pelacakan dan pengelolaan dokumen.

3. Penyesuaian Desain Antarmuka Sistem

- a. Mengidentifikasi tantangan dalam perancangan antarmuka pengguna (UI) agar tetap responsif, mudah dipahami, dan dapat diakses melalui berbagai perangkat seperti komputer maupun perangkat mobile.
- b. Merancang solusi teknis agar sistem dapat berfungsi dengan baik, termasuk optimalisasi performa dan tampilan antarmuka yang sederhana namun fungsional.

3.4 Pengembangan

Dalam pengembangan sistem ini menggunakan metode *Rapid Application Development* (RAD). Metode ini dipilih karena mampu mempercepat proses pengembangan sistem. Metode ini mencakup beberapa tahapan penting dalam proses pengembangan yaitu:



Gambar 3. 2. Metode RAD

3.4.1 Perencanaan Kebutuhan

Pada tahap ini, dilakukan proses identifikasi terhadap kebutuhan sistem, yang didasarkan pada hasil analisis sebelumnya. Beberapa aspek utama yang menjadi fokus dalam tahapan ini meliputi:

1. Kebutuhan Fungsional: Fitur seperti pengajuan surat, verifikasi, hingga pemberitahuan status pengajuan serta pengunduhan dokumen.
2. Kebutuhan Non-Fungsional: Tampilan antarmuka yang menarik, responsif, dan mudah dipahami oleh pengguna.
3. Platform yang Digunakan: *Website* ini dikembangkan menggunakan *framework* Laravel, sebuah kerangka kerja PHP yang bersifat *open-source* dan banyak digunakan dalam pengembangan aplikasi web modern.

3.4.2 Desain Sistem

Desain sistem adalah proses merancang struktur dan komponen-komponen dari sebuah sistem berdasarkan kebutuhan yang telah dianalisis sebelumnya. Tujuannya adalah untuk menentukan bagaimana sistem akan bekerja, termasuk alur data, struktur database, antarmuka pengguna, serta hubungan antar komponen, agar sistem dapat berfungsi secara efektif dan efisien sesuai tujuan yang diinginkan. Menurut (Huda A. M., 2025), Desain sistem adalah proses mendesain elemen-elemen sistem seperti arsitektur, modul dan komponen, antarmuka yang berbeda dari komponen-komponen tersebut, dan data yang melewati sistem tersebut.

Dalam pengembangan sistem berbasis web ini, desain sistem memiliki peran penting dalam memastikan bahwa seluruh komponen yang dibangun dapat bekerja secara terintegrasi dan sesuai dengan kebutuhan pengguna, berikut beberapa elemen utama pada desain sistem:

1. Arsitektur Sistem

Sistem berbasis web ini dikembangkan menggunakan *framework* Laravel, dengan MySQL sebagai sistem basis data yang berfungsi untuk menyimpan dan mengelola data secara terstruktur. Sistem ini dirancang dengan membagi peran pengguna ke dalam tiga komponen utama yang saling terhubung, yaitu:

- a. *Admin*: Bertugas untuk mengelola data *operator*, *user*, jenis surat, dan arsip.
- b. *Operator*: Bertugas untuk melakukan verifikasi pengajuan surat dan mengunggah surat yang sudah diproses, serta mengelola arsip surat.

- c. *User*: Memiliki peran untuk mengajukan permohonan surat melalui sistem

2. Database

Sistem berbasis web ini mengelola database yang terdiri sebagai berikut:

- a. Data *User*: Berisi informasi pribadi seperti nama, email, *password*, dan nomor handphone.
- b. Data Jenis Surat: Berisi data mengenai berbagai jenis surat yang tersedia untuk diajukan.
- c. Data Pengajuan Surat: Mencatat data permohonan surat yang diajukan oleh user.
- d. Data Pengajuan Selesai: Menyimpan data surat yang sudah selesai diproses.
- e. Data Arsip Surat: Menyimpan seluruh surat yang telah diproses, baik yang ditambahkan secara otomatis oleh sistem maupun yang dimasukkan secara manual

3. Interaksi Sistem dan Pengguna

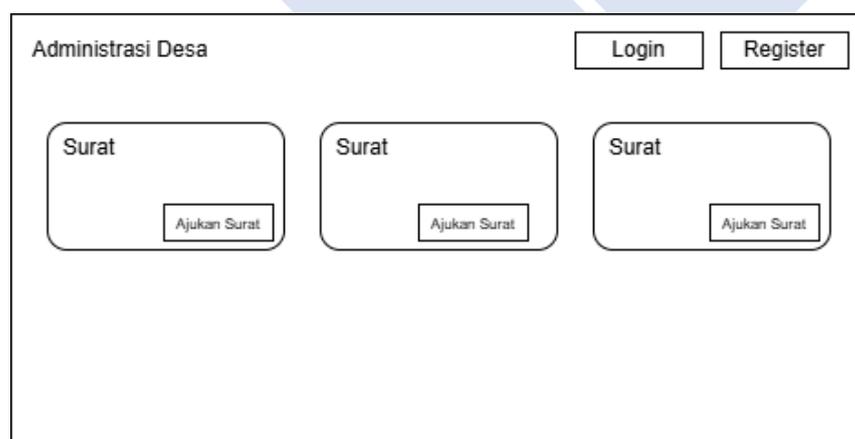
- a. Pengajuan Surat: Pengguna dapat memilih jenis surat yang ingin diajukan, kemudian mengisi data yang dibutuhkan dan mengunggah dokumen pendukung.
- b. Pemantauan Status Pengajuan: Pengguna dapat memantau perkembangan proses surat melalui fitur status pengajuan yang tersedia dalam sistem.
- c. Verifikasi Dokumen: *Operator* memverifikasi kelengkapan. Jika dokumen yang diajukan valid, maka pengajuan diproses; jika tidak valid, maka operator akan menolak pengajuan disertai dengan pesan penolakan yang akan ditampilkan ke pengguna.
- d. Penyelesaian dan Unggah Surat: Jika proses pembuatan surat telah selesai, maka *operator* dapat mengunggah dokumen agar dapat diakses oleh pengguna.

- e. Pengunduhan surat: Pengguna yang mengajukan surat dapat mengunduh surat yang telah selesai dibuat melalui akun mereka di dalam sistem.
 - f. Pengelolaan Arsip Surat: *Operator* memiliki akses ke arsip surat untuk melihat, menambah, dan mengelola arsip dari surat yang telah diproses.
4. Interaksi Sistem dan *Admin*
- a. Manajemen Jenis Surat: *Admin* dapat menambahkan, mengedit, dan menghapus jenis surat sesuai dengan kebutuhan pelayanan desa.
 - b. Pengelolaan Data *User*: *Admin* dapat mengelola akun pengguna yang terdaftar dalam sistem
 - c. Pengelolaan Arsip Surat: *Admin* dapat mengelola arsip surat yang ditambahkan secara otomatis maupun manual

3.4.2.1 Rancangan Desain *Website*

1. Desain Halaman Utama

Halaman utama merupakan tampilan awal yang akan dilihat oleh pengguna saat mengakses sistem informasi administrasi desa. Desain halaman ini dirancang dengan antarmuka yang sederhana namun informatif untuk memudahkan pengguna dalam memahami fungsi dan navigasi awal dari sistem.



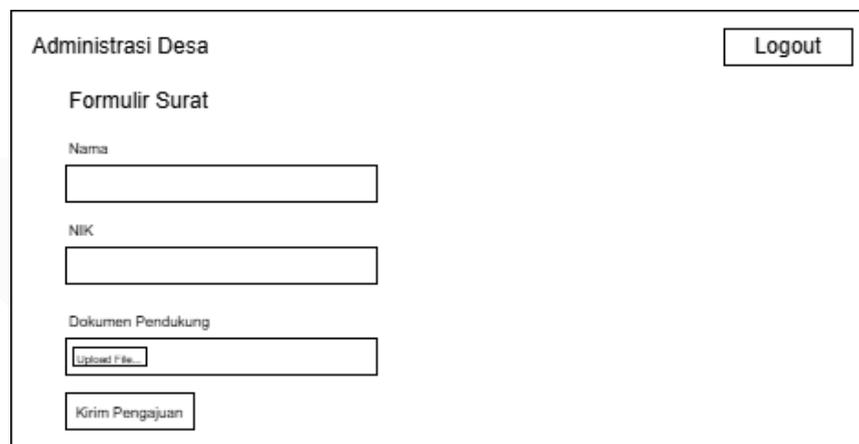
Gambar 3. 3. Desain Halaman Utama

Pada Gambar 3.3 ditampilkan desain halaman utama sistem yang memuat judul utama di bagian atas sebagai identitas sistem. Tersedia tombol *Login*

dan *Register* yang memungkinkan pengguna untuk masuk ke dalam sistem atau membuat akun baru. Pada bagian tengah halaman, terdapat tampilan jenis-jenis surat yang dapat diajukan oleh pengguna, sehingga memudahkan akses langsung terhadap layanan administrasi yang tersedia. Desain ini bertujuan untuk memberikan kesan awal yang jelas, ramah pengguna, dan fungsional.

2. Desain Pengajuan Surat

Halaman pengajuan surat dirancang untuk memfasilitasi pengguna dalam mengisi dan mengirim permohonan surat secara daring.



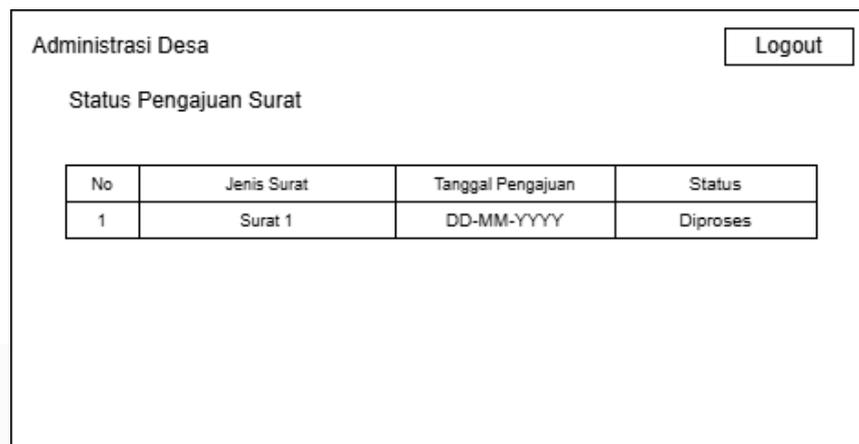
The image shows a web interface for 'Administrasi Desa' (Village Administration). It features a 'Logout' button in the top right corner. The main content is a 'Formulir Surat' (Letter Form) with the following fields: 'Nama' (Name) with a text input box, 'NIK' (National Identity Number) with a text input box, and 'Dokumen Pendukung' (Supporting Documents) with an 'Upload File...' button. At the bottom of the form is a 'Kirim Pengajuan' (Submit Application) button.

Gambar 3. 4. Desain Pengajuan Surat

Pada Gambar 3.4 ditampilkan desain halaman pengajuan surat yang memuat formulir isian sesuai jenis surat yang dipilih oleh pengguna. Setiap kolom isian disesuaikan dengan kebutuhan data dari masing-masing surat. Di bagian bawah formulir, terdapat tombol “Kirim Pengajuan” yang berfungsi untuk mengirim permohonan surat ke sistem agar diproses lebih lanjut oleh pihak desa. Desain ini dirancang agar proses pengajuan surat menjadi lebih praktis, efisien, dan dapat dilakukan tanpa harus datang langsung ke kantor desa.

3. Desain Status Pengajuan Surat

Status pengajuan surat merupakan fitur yang dirancang untuk memberikan transparansi dan kemudahan bagi pengguna dalam memantau perkembangan proses surat yang telah diajukan. Dengan adanya tampilan ini, pengguna dapat mengetahui sejauh mana surat mereka diproses oleh pihak desa.



No	Jenis Surat	Tanggal Pengajuan	Status
1	Surat 1	DD-MM-YYYY	Diproses

Gambar 3. 5. Desain Status Pengajuan Surat

Pada Gambar 3.5 ditampilkan desain halaman status pengajuan surat yang memuat tabel berisi informasi penting seperti jenis surat yang diajukan, tanggal pengajuan, dan status dari masing-masing surat. Desain ini bertujuan agar pengguna dapat dengan mudah memantau dan menindaklanjuti pengajuan mereka tanpa perlu melakukan konfirmasi secara langsung ke kantor desa.

4. Desain Profil Pengguna

Profil pengguna merupakan bagian dari sistem yang dirancang untuk memberikan kemudahan bagi pengguna dalam melihat dan memperbarui informasi pribadinya. Fitur ini menjadi bagian penting dari sistem agar data pengguna selalu akurat dan terkini.

Gambar 3. 6. Desain Profil Pengguna

Pada Gambar 3.6 ditampilkan desain halaman profil pengguna yang memuat informasi pengguna. Pengguna dapat melakukan pembaruan terhadap informasi tersebut langsung melalui formulir yang tersedia. Setelah melakukan perubahan, pengguna dapat menekan tombol “Simpan” untuk memperbarui data di dalam sistem. Desain ini mendukung kenyamanan pengguna dalam mengelola data pribadi secara mandiri.

5. Desain *Dashboard Admin/Operator*

Dashboard merupakan halaman utama yang ditampilkan setelah *admin* atau *operator* berhasil *login* ke dalam sistem. Halaman ini dirancang untuk menyajikan ringkasan informasi penting yang dapat membantu dalam pemantauan dan pengelolaan aktivitas sistem secara menyeluruh dan efisien.

Gambar 3. 7. Desain *Dashboard Admin/Operator*

Pada Gambar 3.7 ditampilkan desain *dashboard* untuk *admin* dan *operator*. *Dashboard* ini menampilkan data ringkas seperti total pengguna, jumlah pengajuan surat pada hari ini, total pengajuan dalam bulan ini, serta riwayat aktivitas terbaru yang dilakukan pengguna. Tampilan ini bertujuan untuk memberikan gambaran umum terkait aktivitas sistem secara *real-time* dan mempermudah pengambilan keputusan operasional.

6. Desain Data *User*

Fitur manajemen data pengguna disediakan untuk *admin* guna mengelola akun-akun yang terdaftar dalam sistem. Halaman ini memungkinkan *admin* untuk melihat, mengedit, maupun menghapus data pengguna secara langsung, sehingga memastikan bahwa informasi pengguna tetap akurat dan terkontrol.

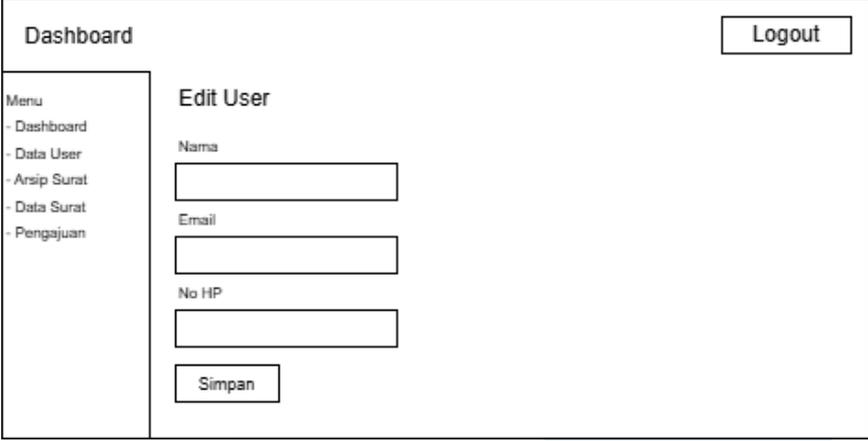
Dashboard		Logout			
Menu	Daftar User				
- Dashboard	Nama	Email	Role	Status	Aksi
- Data User	user	user@email.com	user	Aktif	Edit Hapus
- Arsip Surat					
- Data Surat					
- Pengajuan					

Gambar 3. 8. Desain Data *User*

Pada Gambar 3.8 ditampilkan desain halaman data *user* yang diperuntukkan bagi *admin*. Dalam desain ini, terdapat daftar pengguna yang telah terdaftar dalam sistem lengkap dengan informasi seperti nama, email, peran pengguna, dan status. Selain itu, tersedia tombol aksi seperti “Edit” dan “Hapus” di setiap baris data yang memungkinkan *admin* melakukan pembaruan maupun penghapusan data pengguna sesuai kebutuhan.

7. Desain Edit Data *User*

Untuk menjaga akurasi data pengguna dalam sistem, *admin* diberikan akses ke fitur edit data *user*. Melalui halaman ini, *admin* dapat memperbarui informasi akun pengguna jika terdapat kesalahan input atau perubahan data.



The image shows a web interface for editing user data. It features a sidebar menu on the left with options: Dashboard, Data User, Arsip Surat, Data Surat, and Pengajuan. The main content area is titled 'Edit User' and contains three input fields labeled 'Nama', 'Email', and 'No HP'. Below these fields is a 'Simpan' button. In the top right corner of the dashboard area, there is a 'Logout' button.

Gambar 3. 9. Desain Edit Data *User*

Pada Gambar 3.9 ditampilkan desain halaman edit data *user* yang dapat diakses oleh *admin*. Dalam tampilan ini, *admin* dapat melihat informasi pengguna yang dipilih secara detail. Seluruh data akan ditampilkan dalam kolom input yang bisa langsung diperbarui sesuai kebutuhan. Setelah melakukan perubahan, *admin* dapat menekan tombol “Simpan” untuk memperbarui data pengguna ke dalam sistem.

8. Desain Data Jenis Surat

Untuk memastikan kelengkapan dan fleksibilitas dalam pengelolaan administrasi surat menyurat, sistem menyediakan fitur pengelolaan jenis surat. Fitur ini memungkinkan *admin* untuk menambah, mengubah, atau menghapus jenis surat yang tersedia sesuai kebutuhan desa.

No	Jenis Surat	Deskripsi	Jumlah Field	Aksi
1	Surat 1	Deskripsi	Aksi	Edit Hapus

Gambar 3. 10. Desain Data Jenis Surat

Pada Gambar 3.10 ditampilkan desain halaman data jenis surat yang dapat diakses oleh *admin*. Tampilan ini menyajikan daftar seluruh jenis surat yang telah terdaftar dalam sistem. *Admin* dapat menggunakan tombol “Tambah” untuk menambahkan jenis surat baru, serta tombol aksi seperti “Edit” dan “Hapus” untuk mengelola data yang sudah ada.

9. Desain Tambah/Edit Jenis Surat

Untuk mendukung fleksibilitas dalam pengelolaan jenis surat administrasi, sistem menyediakan halaman khusus bagi *admin* untuk menambahkan maupun mengedit data jenis surat. Halaman ini dirancang agar proses pengelolaan dapat dilakukan dengan mudah, cepat, dan terstruktur.

Gambar 3. 11. Desain Tambah/Edit Jenis Surat

Pada Gambar 3.11 ditampilkan tampilan halaman tambah dan edit jenis surat yang dapat diakses oleh *admin*. Dalam halaman ini, tersedia formulir

input yang mencakup nama jenis surat, deskripsi singkat mengenai surat tersebut, serta *field* dinamis yang digunakan untuk menentukan isian atau data yang perlu dimasukkan oleh pengguna saat mengajukan surat tersebut. Fitur ini memberikan kemudahan bagi *admin* untuk menyesuaikan kebutuhan administrasi desa yang mungkin berubah dari waktu ke waktu.

10. Desain Data Pengajuan

Sistem menyediakan halaman khusus untuk *operator* dalam mengelola data pengajuan surat dari pengguna. Halaman ini dirancang agar *operator* dapat memantau setiap pengajuan secara efisien.

No	Nama Pengaju	Jenis Surat	Tanggal Pengajuan	Status	Detail
1	User	Surat 1	DD-MM-YYYY	Diproses	Detail

Gambar 3. 12. Desain Data Pengajuan

Pada Gambar 3.12 ditampilkan desain halaman data pengajuan surat yang dapat diakses oleh *operator*. Tampilan ini menyajikan daftar antrian pengajuan surat lengkap dengan informasi seperti nama pengaju, jenis surat yang diajukan, tanggal pengajuan, status pengajuan, serta tombol “Detail” untuk melihat informasi lebih lanjut dari setiap pengajuan. Fitur ini bertujuan untuk membantu *operator* dalam mengelola proses pengajuan surat secara terstruktur dan akurat.

11. Desain Detail Pengajuan Surat (*Pending*)

Pengajuan surat yang masih berstatus *pending* menjadi fokus utama halaman ini, yang dirancang khusus untuk digunakan oleh *operator*. *Operator* dapat meninjau informasi pengaju dan rincian pengajuan sebelum memberikan keputusan lebih lanjut.

The image shows a web dashboard interface. At the top left is the word 'Dashboard'. At the top right is a 'Logout' button. On the left side, there is a 'Menu' section with a list of items: Dashboard, Data User, Arsip Surat, Data Surat, and Pengajuan. The main content area is titled 'Detail Pengajuan Surat' and contains the following information: 'Nama Pemohon: User', 'Jenis Surat: Surat 1', 'Tanggal Pengajuan: DD-MM-YYYY', 'Status: Pending', and 'Email: user@email.com'. Below this is a section for 'Data Pengajuan' with fields for 'Nama:', 'No KIK:', and 'Dokumen Pendukung: Download'. There are two buttons, 'Setuju' and 'Tolak', positioned below the 'Data Pengajuan' section. At the bottom, there is a 'Pesan' section with a text input field labeled 'Tulis Pesan Disini.....'.

Gambar 3. 13. Desain Detail Pengajuan Surat (*Pending*)

Pada Gambar 3.13 ditampilkan desain halaman validasi pengajuan surat yang memungkinkan *operator* memeriksa data dari warga yang mengajukan surat. Informasi yang ditampilkan mencakup data diri pengaju dan isian dari formulir surat yang telah dikirimkan. Terdapat dua tombol utama yaitu “Setuju” dan “Tolak”, serta kolom untuk menambahkan pesan atau catatan terkait keputusan yang diambil. Desain ini bertujuan untuk mempermudah proses verifikasi dan pengambilan keputusan oleh *operator* secara cepat dan terstruktur.

12. Desain Detail Pengajuan Surat (Diproses)

Halaman ini ditujukan bagi *operator* untuk menangani pengajuan surat yang sedang dalam tahap pemrosesan. Setelah pengajuan disetujui, *operator* dapat menindaklanjuti dengan mengunggah berkas surat yang telah diselesaikan.

Dashboard		Logout
Menu	- Dashboard	Detail Pengajuan Surat Nama Pemohon: User Jenis Surat: Surat 1 Tanggal Pengajuan: DD-MM-YYYY Status: Diproses Email: user@email.com Data Pengajuan Nama : No KK: Dokumen Pendukung: Download <input type="button" value="Upload File"/> <input type="button" value="Kirim"/>
	- Data User	
	- Arsip Surat	
	- Data Surat	
	- Pengajuan	

Gambar 3. 14. Desain Detail Pengajuan Surat (Diproses)

Pada Gambar 3.14 ditampilkan desain halaman detail pengajuan surat yang telah masuk tahap pemrosesan. Tampilan ini mirip dengan halaman detail pengajuan sebelumnya, di mana informasi pengaju dan data formulir tetap ditampilkan secara lengkap. Perbedaannya terletak pada fitur tambahan berupa tombol “*Upload File*” yang digunakan untuk mengunggah hasil surat yang telah selesai, serta tombol “*Kirim*” untuk menyelesaikan proses pengajuan dan mengirim surat kepada warga. Desain ini membantu *operator* menyelesaikan administrasi surat secara efisien dan sistematis.

13. Desain Arsip Surat

Halaman arsip surat berfungsi sebagai tempat penyimpanan seluruh surat yang telah selesai diproses. Fitur ini memudahkan *admin* dan *operator* dalam melakukan pencatatan serta penelusuran surat secara sistematis.

No	Nama Pengaju	NIK	Jenis Surat	Detail
1	User	NIK	Surat 1	Detail

Gambar 3. 15. Desain Arsip Surat

Pada Gambar 3.15 ditampilkan desain halaman arsip surat yang dapat diakses oleh *admin* maupun *operator*. Dalam tampilan ini, tersedia daftar arsip surat yang telah selesai diproses lengkap dengan informasi terkait surat yang sudah selesai. Terdapat tombol “Tambah” yang berfungsi untuk menambahkan arsip secara manual jika diperlukan, serta tombol “Detail” yang digunakan untuk melihat informasi lebih lanjut mengenai arsip surat tertentu. Desain ini membantu memastikan dokumentasi surat tersimpan dengan rapi dan mudah diakses kembali.

14. Desain Tambah Arsip Surat

Fitur tambah arsip surat disediakan untuk mempermudah *admin* atau *operator* dalam mencatat surat-surat yang telah diselesaikan secara manual ke dalam sistem arsip surat. Hal ini penting agar seluruh dokumentasi surat tetap tersimpan dan terdigitalisasi dengan baik.

The image shows a web application interface for adding a letter archive. It features a sidebar menu on the left with options: Dashboard, Data User, Arsip Surat, Data Surat, and Pengajuan. The main content area is titled 'Tambah Arsip Surat' and contains the following form elements:

- A 'Logout' button in the top right corner.
- A 'Nama Pengaju' (Applicant Name) text input field.
- A 'NIK' (National Identity Card Number) text input field.
- A 'Jenis Surat' (Letter Type) dropdown menu with the placeholder text '--Pilih Jenis Surat--'.
- A 'Dokumen' (Document) section with an 'Upload File....' button.
- A 'Simpan' (Save) button at the bottom right of the form.

Gambar 3. 16. Desain Tambah Arsip Surat

Pada Gambar 3.16 ditampilkan desain halaman tambah arsip surat. Desain ini menyajikan sebuah formulir yang memuat *input* data arsip surat, termasuk *dropdown* untuk memilih jenis surat sesuai dengan data yang tersedia pada fitur data jenis surat. Selain itu, terdapat fitur unggah dokumen yang berfungsi untuk melampirkan *file* surat, serta tombol “Simpan” untuk menyimpan data arsip ke dalam sistem. Tampilan ini dirancang untuk sederhana namun tetap fungsional agar proses pengarsipan berjalan cepat dan efisien.

3.4.2.2 Use Case Diagram

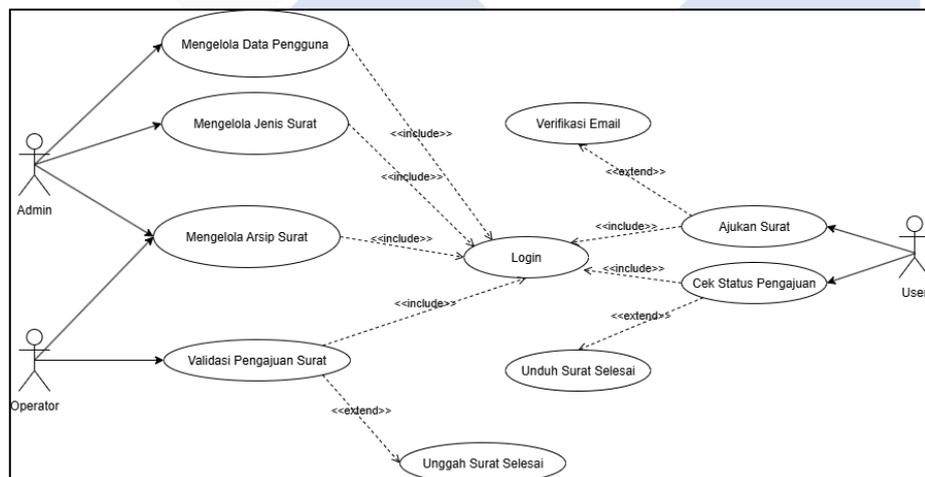
Berikut merupakan gambaran umum sistem dalam bentuk *Use Case Diagram*, yang menggambarkan interaksi antara sistem dan tiga aktor utama, yaitu *Admin*, *Operator*, dan *User* (Warga). Masing-masing aktor memiliki hak akses dan fungsi yang berbeda sesuai dengan peran dan tanggung jawabnya dalam sistem.

- Admin*

Admin memiliki akses penuh terhadap sistem. *Admin* dapat melakukan proses *login* dan *logout*, serta memiliki kewenangan untuk mengelola data pengguna, mengelola jenis surat, dan mengelola arsip surat yang tersimpan dalam sistem.
- Operator*

Operator merupakan petugas yang bertanggung jawab dalam proses pengelolaan surat. *Operator* dapat *login* dan *logout* dari sistem, melakukan validasi terhadap pengajuan surat yang diajukan oleh warga, serta turut mengelola arsip surat yang telah selesai diproses.
- User*

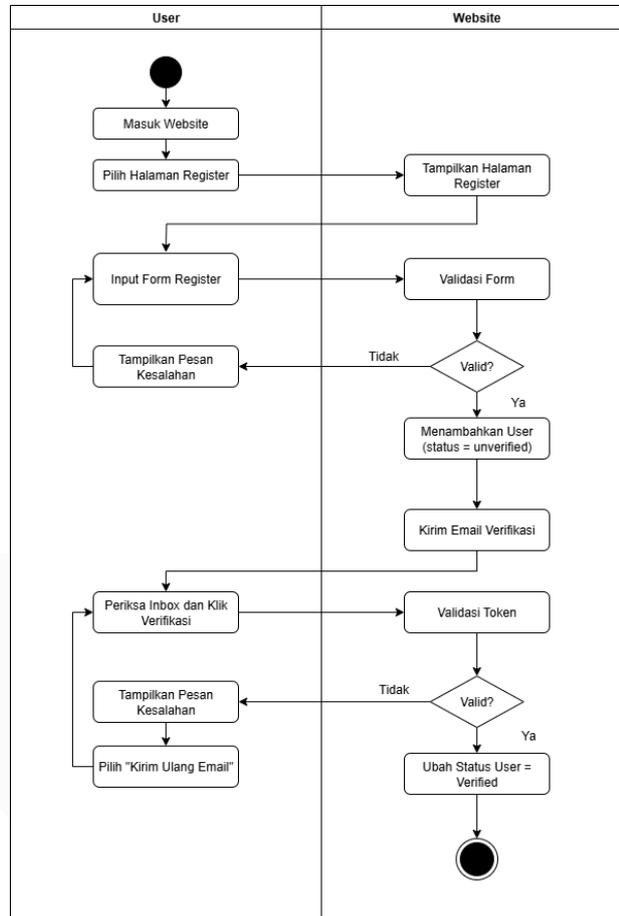
User adalah warga desa yang menggunakan sistem untuk keperluan layanan administrasi. *User* dapat melakukan registrasi, *login*, dan *logout*, melakukan verifikasi email, mengajukan permohonan surat, memantau status pengajuan, serta mengunduh surat yang telah selesai diverifikasi dan diproses oleh *operator*.



Gambar 3. 17. Use Case Diagram

3.4.2.3 Activity Diagram

1. Activity Diagram Register

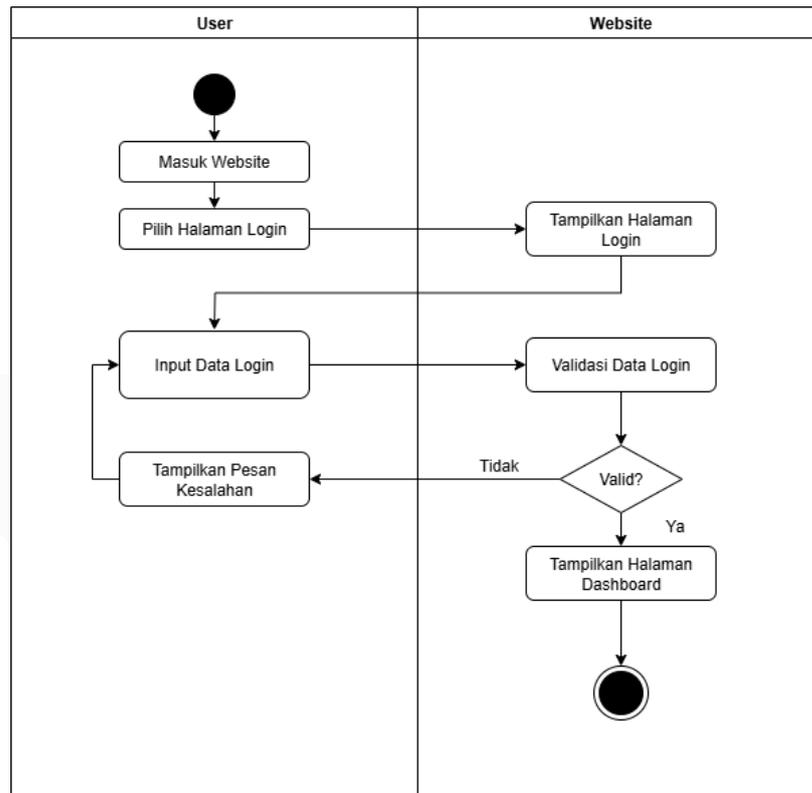


Gambar 3. 18. Activity Diagram Register

Pada Gambar 3.18 dijelaskan bahwa aktivitas registrasi pengguna dimulai dengan mengakses situs web dan memilih menu *Register*. Selanjutnya, sistem akan menampilkan halaman formulir registrasi yang dapat diisi oleh pengguna. Setelah pengguna mengisi formulir, sistem akan melakukan proses validasi data. Apabila data tidak valid, maka sistem akan menampilkan pesan kesalahan. Namun, jika data dinyatakan valid, maka proses pendaftaran akan berhasil dan status pengguna akan tercatat sebagai belum terverifikasi. Sistem kemudian akan mengirimkan email verifikasi ke alamat yang didaftarkan. Pengguna diharapkan memeriksa *inbox* dan mengklik tautan verifikasi yang dikirimkan. Setelah itu, sistem akan

melakukan validasi terhadap token verifikasi. Jika token tersebut valid, maka status pengguna akan diperbarui menjadi terverifikasi. Sebaliknya, apabila token tidak valid, sistem akan menampilkan pesan kesalahan dan memberikan opsi kepada pengguna untuk mengirim ulang email verifikasi.

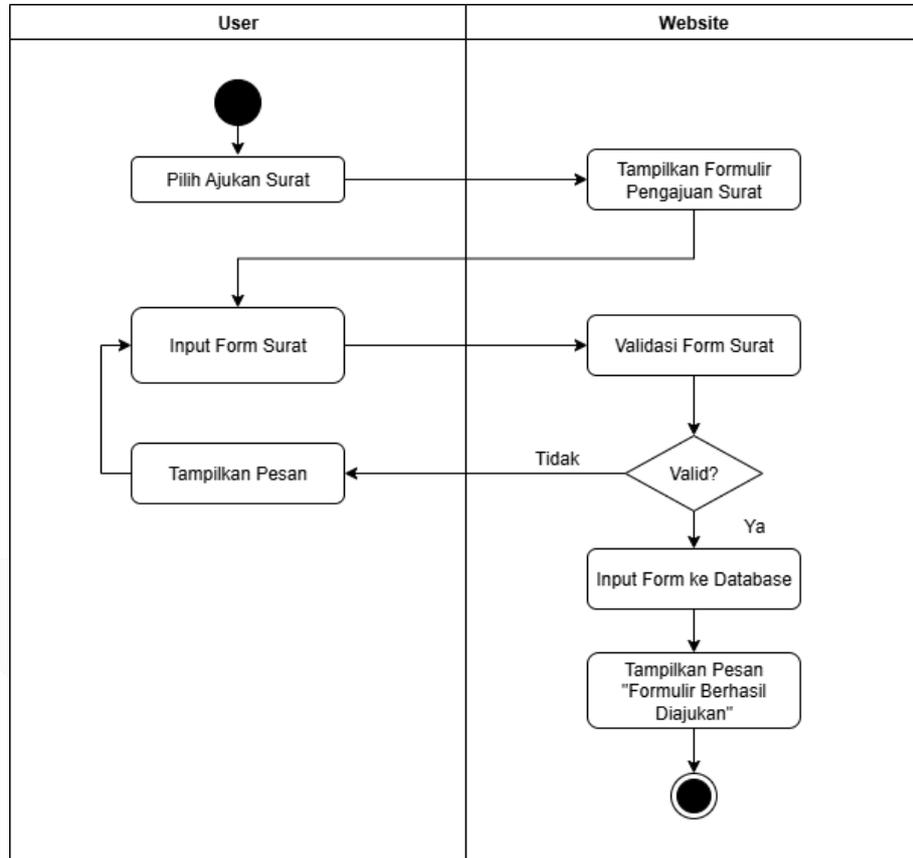
2. Activity Diagram Login



Gambar 3. 19. Activity Diagram Login

Pada Gambar 3.19 dijelaskan bahwa aktivitas *login* diawali dengan pengguna mengakses situs web dan memilih menu *Login*. Selanjutnya, sistem akan menampilkan halaman *login*. Pengguna kemudian mengisi data *login* yang telah disediakan. Setelah itu, sistem akan melakukan proses validasi terhadap data yang dimasukkan. Apabila data *login* dinyatakan valid, maka pengguna akan diarahkan ke halaman *dashboard*. Namun, jika data tidak valid, sistem akan menampilkan pesan kesalahan kepada pengguna.

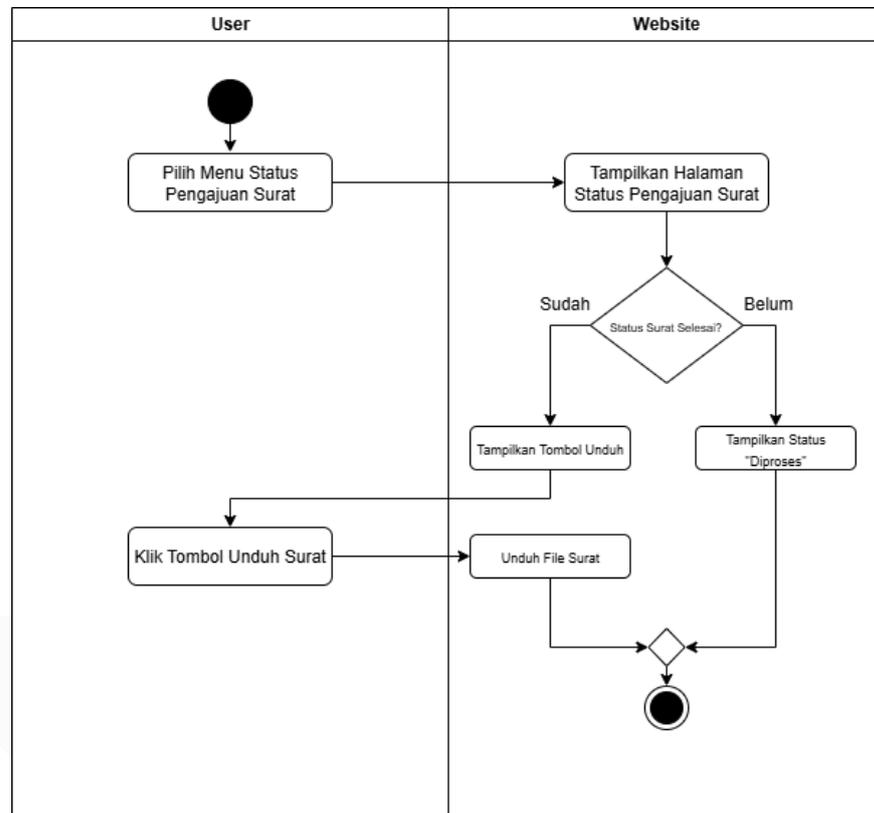
3. Activity Diagram Pengajuan Surat



Gambar 3. 20. Activity Diagram Pengajuan Surat

Pada Gambar 3.20 dijelaskan bahwa aktivitas pengajuan surat oleh warga diawali dengan memilih menu “Ajukan Surat”. Selanjutnya, sistem akan menampilkan halaman formulir pengajuan surat. Pengguna kemudian mengisi formulir yang tersedia, dan sistem akan melakukan validasi terhadap data yang dimasukkan. Apabila data valid, maka data pengajuan akan disimpan ke dalam basis data dan sistem akan menampilkan pesan bahwa formulir berhasil diajukan. Namun, jika data tidak valid, maka sistem akan menampilkan pesan kesalahan kepada pengguna.

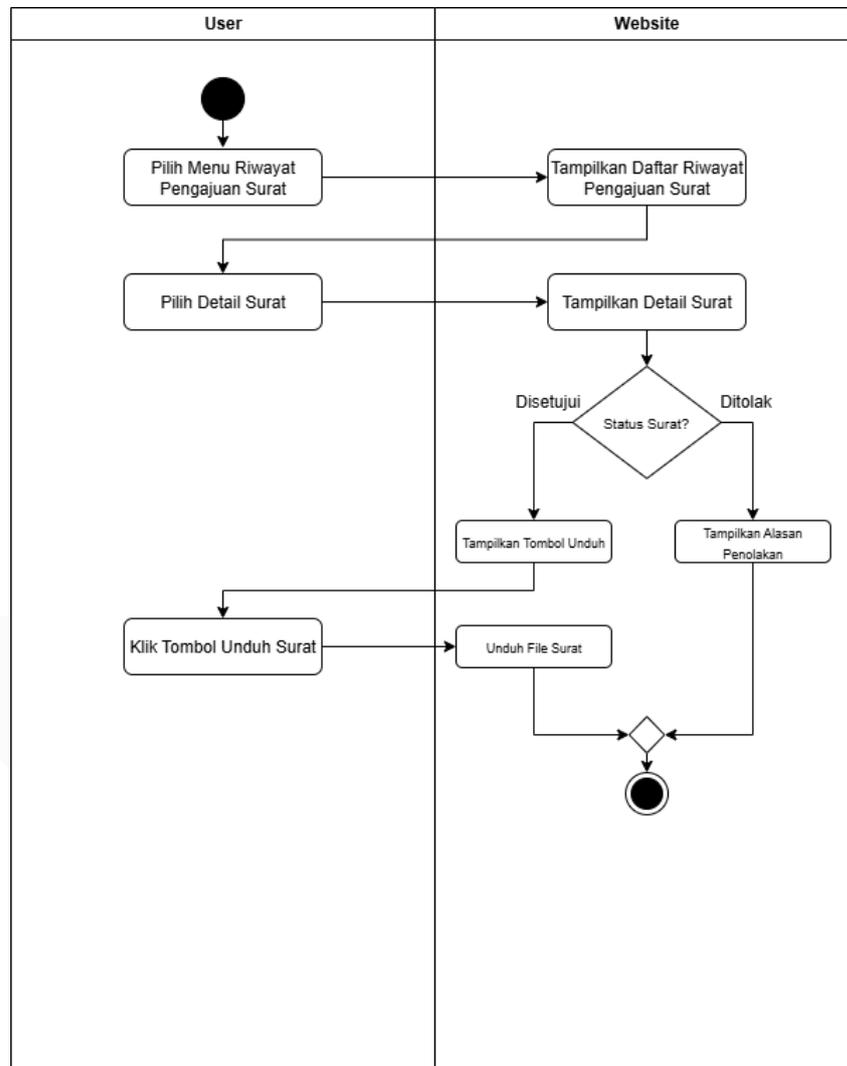
4. Activity Diagram Status Pengajuan Surat



Gambar 3. 21. Activity Diagram Status Pengajuan Surat

Pada Gambar 3.21 dijelaskan bahwa aktivitas melihat status pengajuan surat oleh warga diawali dengan memilih menu “Status Pengajuan Surat”. Selanjutnya, sistem akan menampilkan halaman status pengajuan surat. Apabila status surat telah selesai diproses, maka sistem akan menampilkan tombol “Unduh Surat”, yang dapat digunakan oleh pengguna untuk mengunduh dokumen surat tersebut. Namun, apabila surat masih dalam proses, maka sistem akan menampilkan informasi bahwa surat sedang diproses.

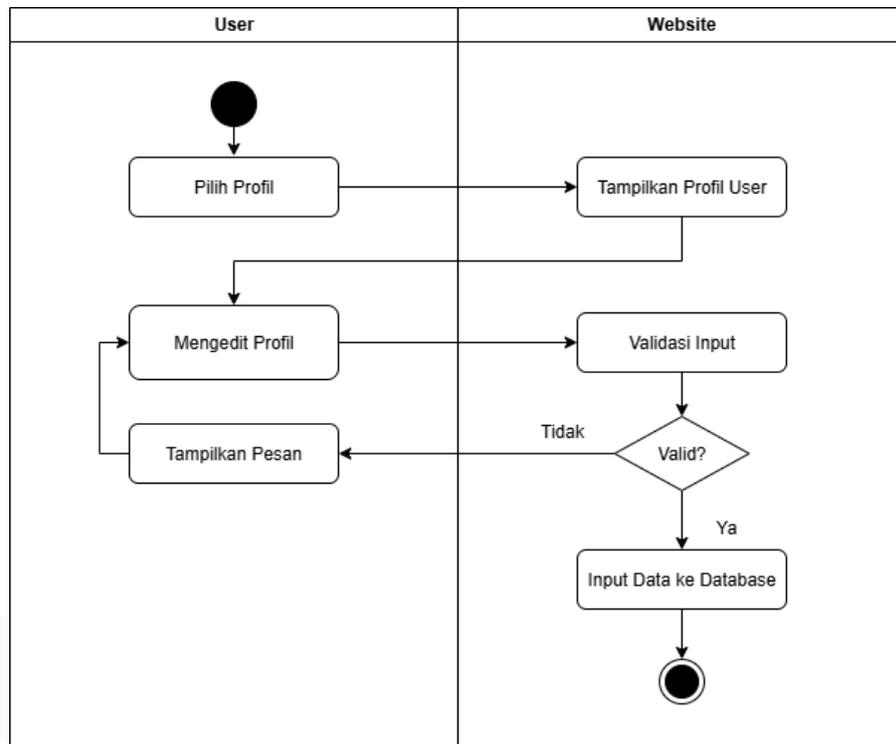
5. Activity Diagram Riwayat Pengajuan Surat



Gambar 3. 22. Activity Diagram Riwayat Pengajuan Surat

Pada Gambar 3.22 dijelaskan bahwa aktivitas melihat riwayat pengajuan surat oleh warga dimulai dengan memilih menu “Riwayat Pengajuan Surat”. Setelah itu, sistem akan menampilkan halaman daftar riwayat pengajuan surat. Pengguna dapat melihat detail dari setiap pengajuan surat dengan memilih tombol “Detail”, dan sistem akan menampilkan informasi lengkap mengenai pengajuan tersebut. Jika status surat disetujui, maka pada halaman detail akan ditampilkan tombol “Unduh Surat” untuk mengunduh dokumen surat. Namun, jika status surat ditolak, maka sistem akan menampilkan alasan penolakan surat tersebut.

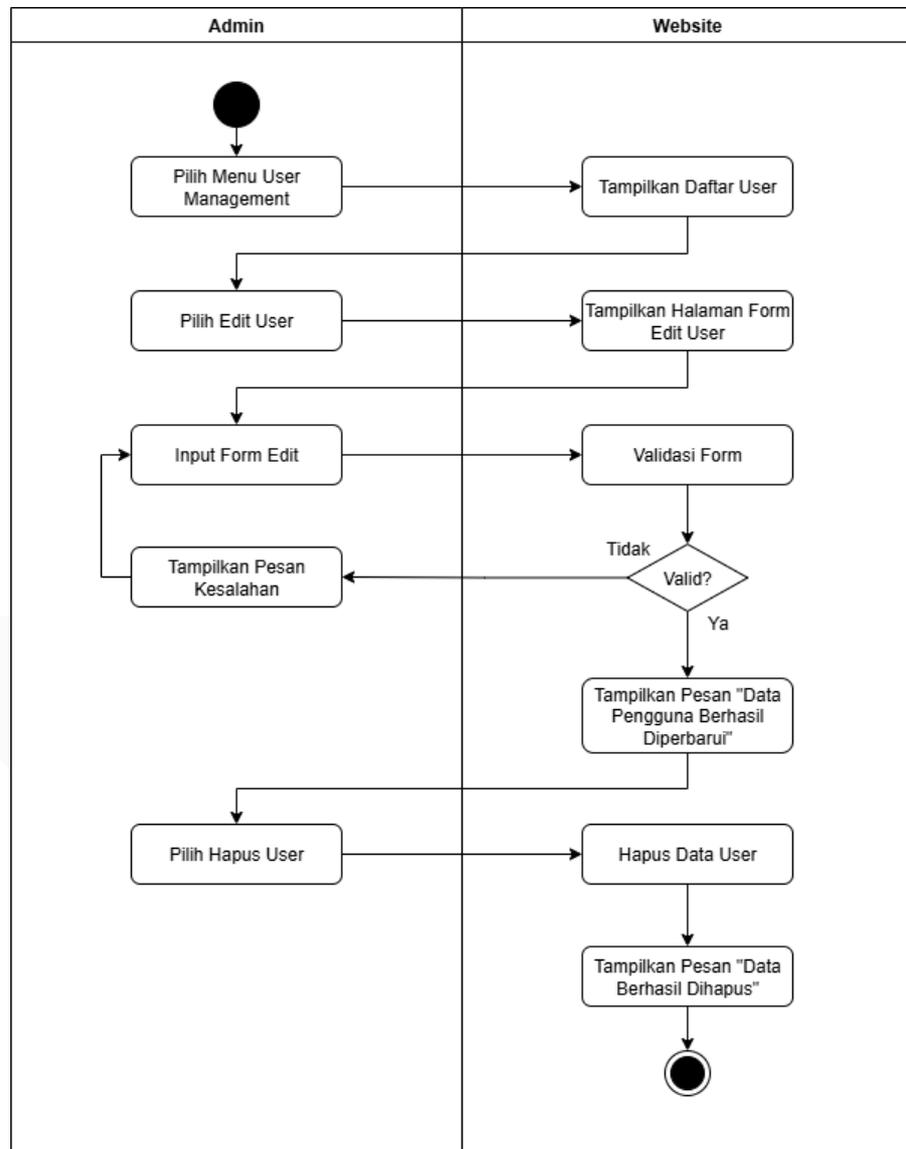
6. Activity Diagram Edit Profil



Gambar 3. 23. Activity Diagram Edit Profil

Pada Gambar 3.23 dijelaskan bahwa aktivitas edit profil dimulai dengan memilih menu “Profil”. Setelah itu, sistem akan menampilkan halaman profil pengguna. Selanjutnya, pengguna dapat melakukan perubahan data dengan mengedit informasi profil. Proses ini akan melalui tahapan validasi input. Apabila data yang dimasukkan valid, maka sistem akan menyimpan perubahan ke dalam basis data dan menampilkan pesan bahwa perubahan berhasil dilakukan. Namun, apabila data tidak valid, maka sistem akan menampilkan pesan kesalahan.

7. Activity Diagram User Management

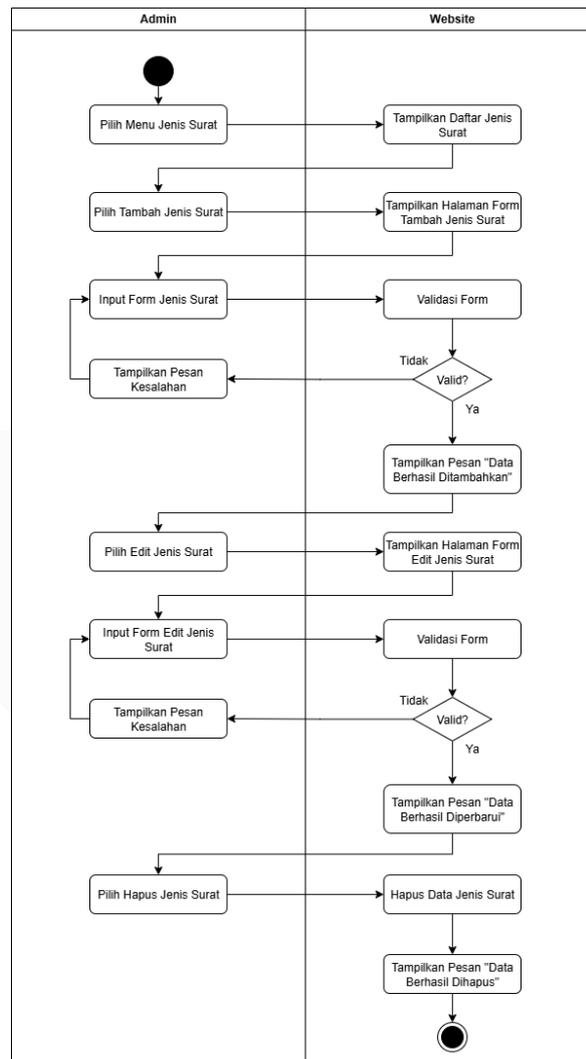


Gambar 3. 24. Activity Diagram User Management

Pada Gambar 3.24 dijelaskan bahwa aktivitas manajemen pengguna oleh *admin* dimulai dengan memilih menu “*User Management*”. Setelah itu, sistem akan menampilkan halaman daftar pengguna. Selanjutnya, *admin* dapat melakukan perubahan data pengguna melalui fitur edit. Proses ini akan melalui tahapan validasi input. Apabila data yang dimasukkan valid, maka sistem akan memperbarui data dan menampilkan pesan bahwa data pengguna berhasil diperbarui. Namun, apabila data tidak valid, sistem akan

menampilkan pesan kesalahan. Selain itu, *admin* juga dapat menghapus data pengguna, yang mana akan direspons oleh sistem dengan menghapus data tersebut dan menampilkan pesan bahwa data berhasil dihapus.

8. Activity Diagram Data Jenis Surat

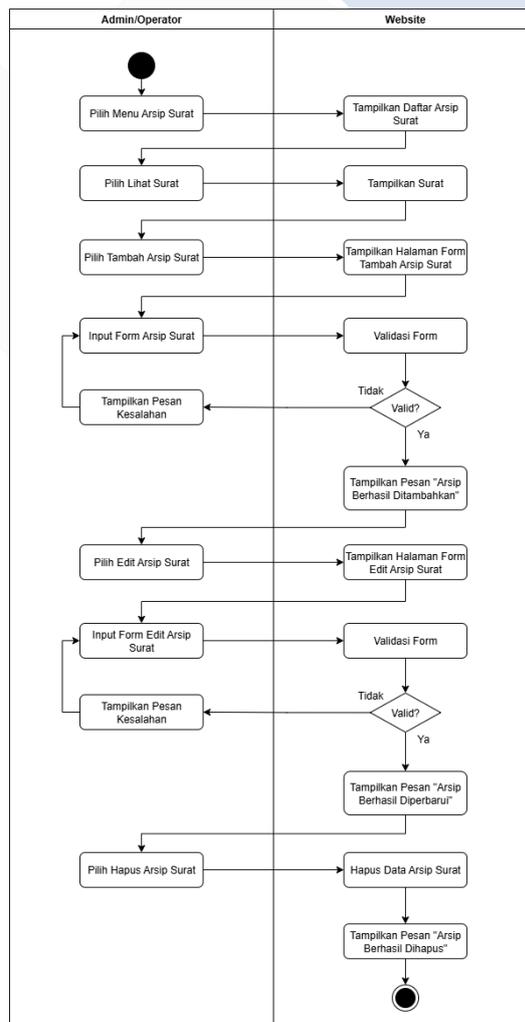


Gambar 3. 25. Activity Diagram Data Jenis Surat

Pada Gambar 3.25 dijelaskan bahwa aktivitas pengelolaan data jenis surat oleh *admin* dimulai dengan memilih menu “Jenis Surat”. Setelah itu, sistem akan menampilkan halaman daftar jenis surat yang tersedia. *Admin* dapat menambahkan data jenis surat dengan memilih opsi tambah, dan sistem akan menampilkan halaman formulir penambahan jenis surat. *Admin* kemudian mengisi formulir tersebut, yang akan divalidasi oleh sistem. Jika

data yang dimasukkan valid, maka sistem akan menyimpan data dan menampilkan pesan bahwa data berhasil ditambahkan. Jika data tidak valid, maka sistem akan menampilkan pesan kesalahan. Selanjutnya, *admin* juga dapat melakukan perubahan terhadap data jenis surat yang ada dengan memilih opsi edit, dan sistem akan menampilkan halaman formulir edit jenis surat. Proses validasi kembali dilakukan, dan jika data valid, sistem akan menampilkan pesan bahwa data berhasil diperbarui. Jika tidak valid, maka akan ditampilkan pesan kesalahan. Selain itu, *admin* juga dapat menghapus data jenis surat, yang akan membuat sistem menghapus data tersebut dan menampilkan pesan bahwa data berhasil dihapus.

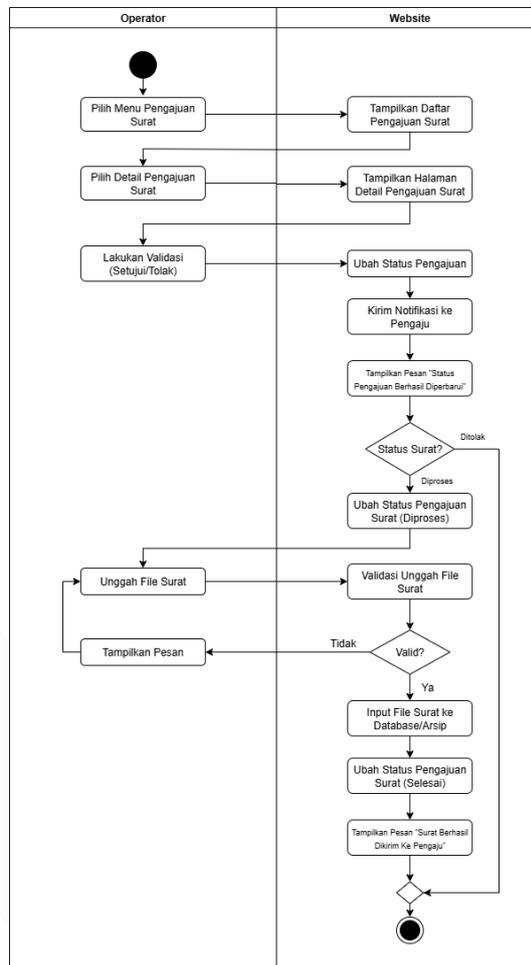
9. Activity Diagram Arsip Surat



Gambar 3. 26. Activity Diagram Arsip Surat

Pada Gambar 3.26 dijelaskan bahwa aktivitas pengelolaan data arsip surat oleh *admin* atau *operator* dimulai dengan memilih menu “Arsip Surat”. Setelah itu, sistem akan menampilkan halaman daftar arsip surat yang tersedia. *Admin* atau *operator* dapat melihat arsip surat dengan memilih salah satu data arsip, dan sistem akan menampilkan isi atau dokumen dari arsip tersebut. *Admin* atau *operator* juga dapat menambahkan arsip surat dengan memilih tombol “Tambah”, yang akan memunculkan halaman formulir penambahan arsip. Formulir tersebut kemudian diisi dan divalidasi oleh sistem. Jika data yang dimasukkan valid, maka sistem akan menyimpan data dan menampilkan pesan bahwa arsip berhasil ditambahkan. Jika data tidak valid, maka sistem akan menampilkan pesan kesalahan. Selanjutnya, *admin* atau *operator* juga dapat melakukan pengeditan terhadap arsip surat yang ada. Sistem akan menampilkan halaman formulir edit arsip, dan setelah pengisian dilakukan, sistem akan memvalidasi data tersebut. Jika data valid, maka akan ditampilkan pesan bahwa arsip berhasil diperbarui, sedangkan jika tidak valid, sistem akan menampilkan pesan kesalahan. Terakhir, *admin* atau *operator* dapat menghapus data arsip surat, dan sistem akan menghapus data tersebut serta menampilkan pesan bahwa arsip berhasil dihapus.

10. Activity Diagram Validasi Pengajuan Surat



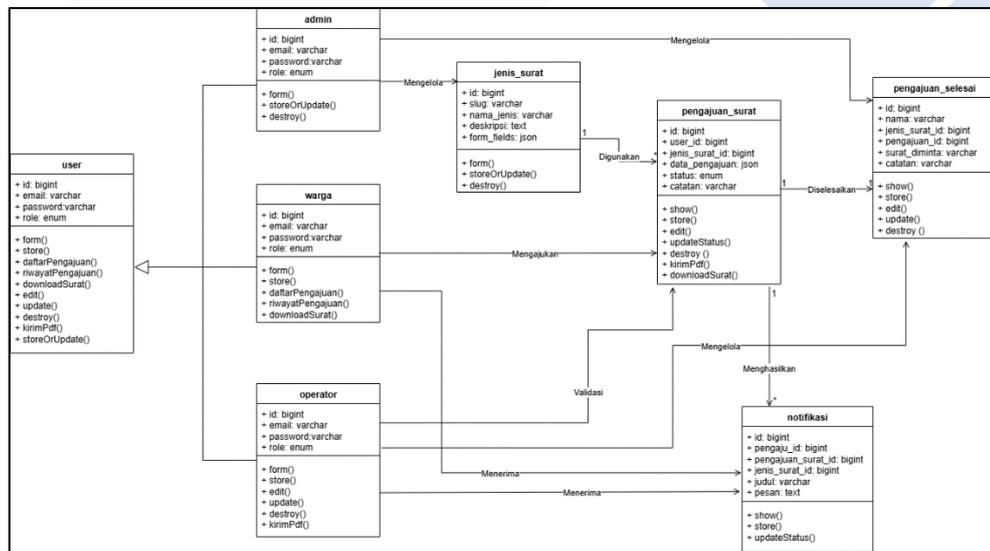
Gambar 3. 27. Activity Diagram Validasi Pengajuan Surat

Pada Gambar 3.27 dijelaskan bahwa aktivitas validasi pengajuan surat oleh *operator* dimulai dengan memilih menu “Antrian Pengajuan Surat”. Sistem kemudian menampilkan daftar pengajuan yang masuk. *Operator* dapat memilih salah satu pengajuan untuk melihat detailnya, yang ditampilkan dalam halaman khusus. Melalui halaman ini, *operator* dapat melakukan validasi dengan memilih untuk menyetujui atau menolak pengajuan. Jika validasi dilakukan, sistem akan memperbarui status pengajuan sesuai keputusan, mengirimkan notifikasi kepada pengaju, serta menampilkan pesan konfirmasi. Apabila surat telah selesai diproses, *operator* dapat mengunggah dokumen melalui halaman yang sama. Setelah file diunggah, sistem akan memverifikasi validitasnya. Jika valid, sistem akan menyimpan

dokumen ke dalam basis data dan arsip, memperbarui status pengajuan menjadi “selesai”, serta menampilkan pesan bahwa surat berhasil dikirim kepada pengaju. Sebaliknya, jika file tidak valid, sistem akan menampilkan pesan kesalahan.

3.4.2.4 Class Diagram

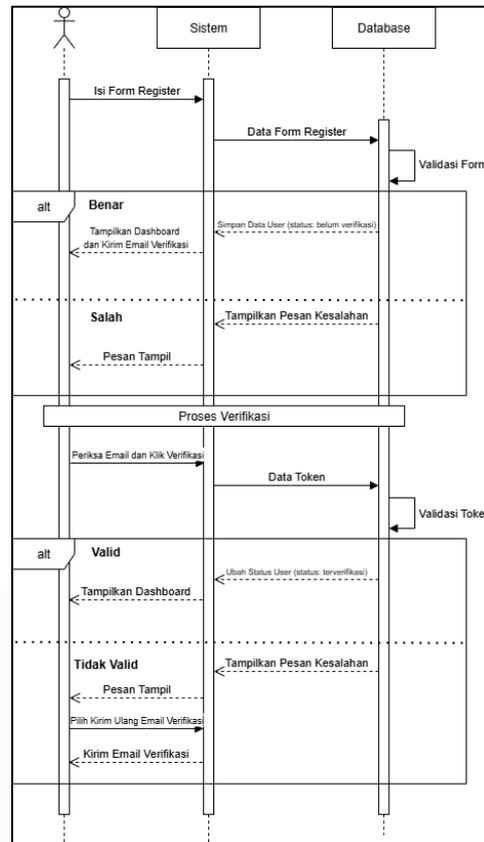
Class diagram berikut menggambarkan struktur data serta hubungan antar objek dalam sistem website administrasi pelayanan surat di Desa Rias. Diagram ini memvisualisasikan bagaimana masing-masing aktor, yaitu *Admin*, *Operator*, dan *Warga*, berinteraksi dengan berbagai entitas seperti Jenis Surat, Pengajuan Surat, Pengajuan Selesai, dan Notifikasi. Setiap kelas dalam diagram dilengkapi dengan atribut dan metode yang mendukung alur kerja serta fungsi sistem, sesuai dengan tanggung jawab dan peran pengguna dalam proses pelayanan administrasi desa secara menyeluruh.



Gambar 3. 28. *Class Diagram*

3.4.2.5 Sequence Diagram

1. Sequence Diagram Register



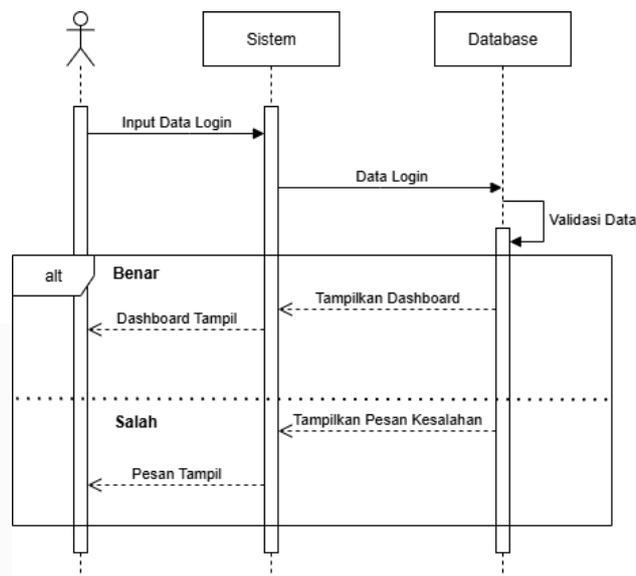
Gambar 3. 29. Sequence Diagram Register

Pada Gambar 3.29 dijelaskan bahwa aktivitas registrasi dimulai ketika pengguna mengisi formulir pendaftaran. Setelah itu, sistem akan mengirimkan data yang diinput ke database untuk divalidasi. Jika data valid, maka informasi pengguna akan disimpan dengan status “belum diverifikasi”, sistem akan menampilkan halaman *dashboard*, serta mengirimkan email verifikasi kepada pengguna. Namun, jika data tidak valid, maka sistem akan menampilkan pesan kesalahan.

Selanjutnya, pada proses verifikasi, pengguna akan memeriksa email dan menekan tombol verifikasi yang telah dikirim. Sistem kemudian akan memvalidasi token verifikasi. Jika token dinyatakan valid, maka status pengguna akan diperbarui menjadi “terverifikasi” dan sistem akan

menampilkan *dashboard*. Sebaliknya, jika token tidak valid, sistem akan menampilkan pesan kesalahan dan memberikan opsi kepada pengguna untuk mengirim ulang email verifikasi. Jika pengguna memilih untuk mengirim ulang, maka sistem akan mengirimkan ulang email verifikasi ke alamat email yang terdaftar.

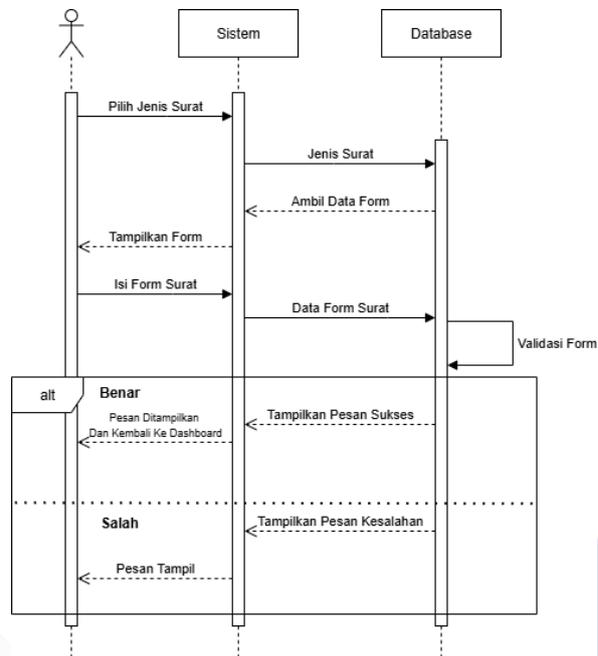
2. Sequence Diagram Login



Gambar 3. 30. Sequence Diagram Login

Pada Gambar 3.30 dijelaskan bahwa proses *login* dimulai ketika pengguna mengisi data *login*. Selanjutnya, sistem akan mengirimkan data tersebut ke basis data untuk dilakukan validasi. Apabila data yang dimasukkan benar, maka pengguna akan diarahkan ke halaman *dashboard*. Namun, jika data tidak valid, sistem akan menampilkan pesan kesalahan.

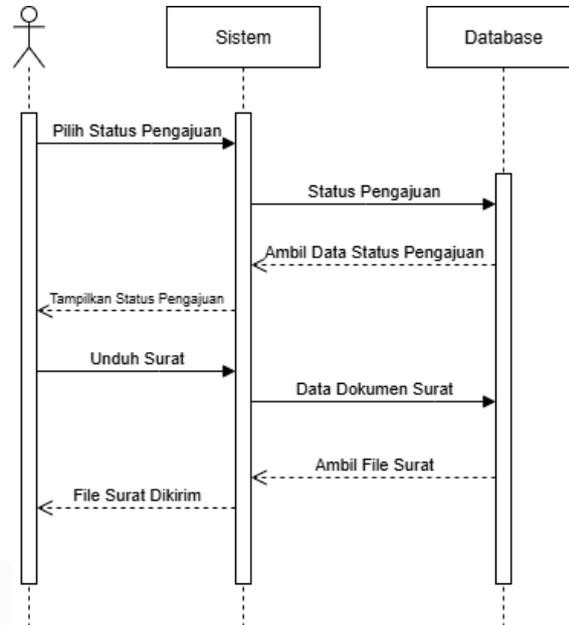
3. Sequence Diagram Pengajuan Surat



Gambar 3. 31. Sequence Diagram Pengajuan Surat

Pada Gambar 3.31 dijelaskan bahwa proses pengajuan surat dimulai ketika pengguna memilih jenis surat yang diinginkan. Selanjutnya, sistem akan mengambil jenis surat tersebut dan basis data akan menyediakan data formulir yang sesuai. Sistem kemudian menampilkan formulir kepada pengguna untuk diisi. Setelah formulir diisi, sistem melakukan proses validasi. Apabila data yang diinput valid, maka akan ditampilkan pesan sukses. Sebaliknya, jika terdapat kesalahan pada data, sistem akan menampilkan pesan kesalahan.

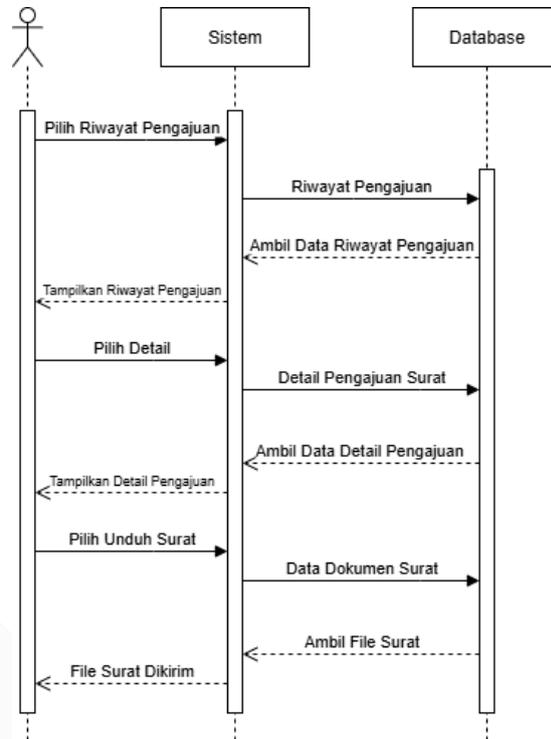
4. Sequence Diagram Status Pengajuan Surat



Gambar 3. 32. Sequence Diagram Status Pengajuan Surat

Pada Gambar 3.32 dijelaskan bahwa proses melihat status pengajuan surat dimulai ketika pengguna memilih menu “Status Pengajuan”. Selanjutnya, basis data akan mengirimkan data status pengajuan, dan sistem akan menampilkannya kepada pengguna. Apabila surat telah selesai diproses, pengguna dapat mengunduh surat tersebut. Pada tahap ini, basis data akan mengambil *file* surat, kemudian sistem akan mengirimkan *file* tersebut kepada pengguna.

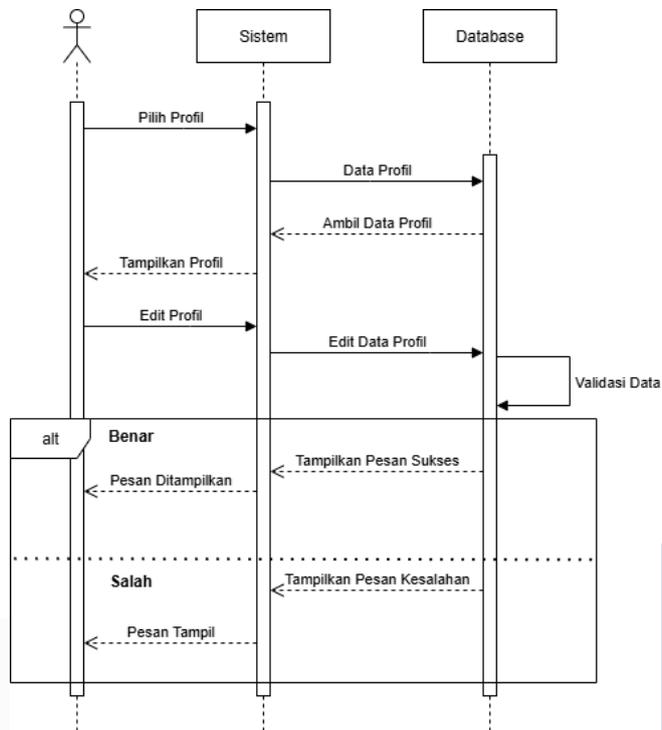
5. Sequence Diagram Riwayat Pengajuan Surat



Gambar 3. 33. Sequence Diagram Riwayat Pengajuan Surat

Pada Gambar 3.33 dijelaskan bahwa proses melihat riwayat pengajuan surat dimulai ketika pengguna memilih menu “Riwayat Pengajuan”. Selanjutnya, basis data akan mengambil data riwayat pengajuan, kemudian sistem akan menampilkannya kepada pengguna. Pengguna dapat memilih salah satu pengajuan untuk melihat detailnya, dan sistem akan mengambil data detail tersebut dari basis data. Apabila status riwayat surat telah disetujui, pengguna dapat mengunduh surat tersebut. Pada tahap ini, basis data akan mengambil *file* surat, lalu sistem akan mengirimkan *file* tersebut kepada pengguna.

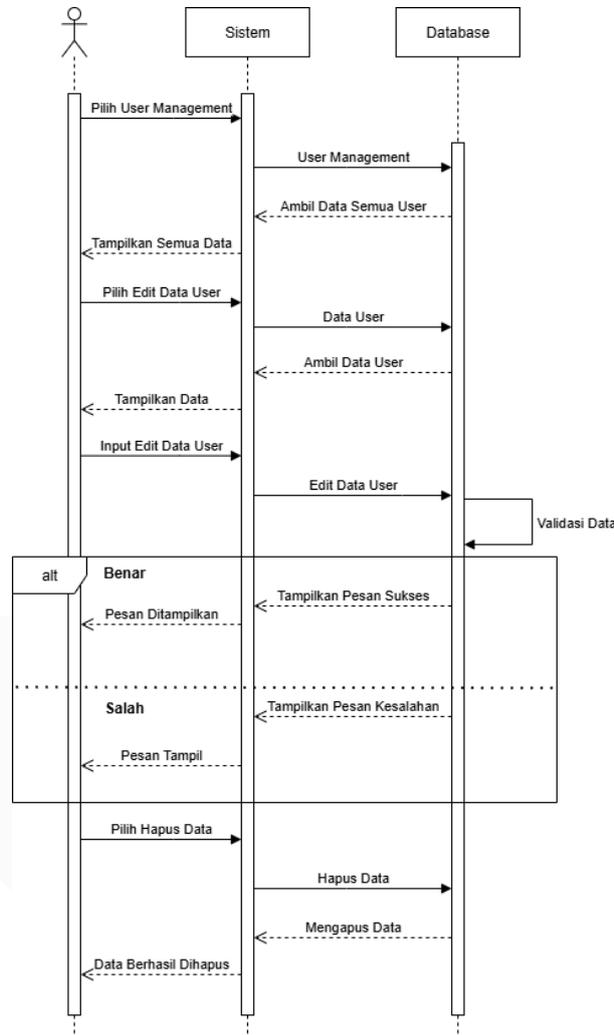
6. Sequence Diagram Edit Profil User



Gambar 3. 34. Sequence Diagram Edit Profil User

Pada Gambar 3.34 dijelaskan bahwa proses melihat profil pengguna dimulai ketika pengguna memilih menu “Profil”. Selanjutnya, basis data akan mengambil data profil, dan sistem akan menampilkan halaman profil kepada pengguna. Apabila pengguna melakukan pembaruan data profil, sistem akan mengirimkan data tersebut ke basis data untuk divalidasi. Jika data yang dimasukkan valid, maka sistem akan menampilkan pesan sukses. Sebaliknya, jika data tidak valid, sistem akan menampilkan pesan kesalahan.

7. Sequence Diagram User Management

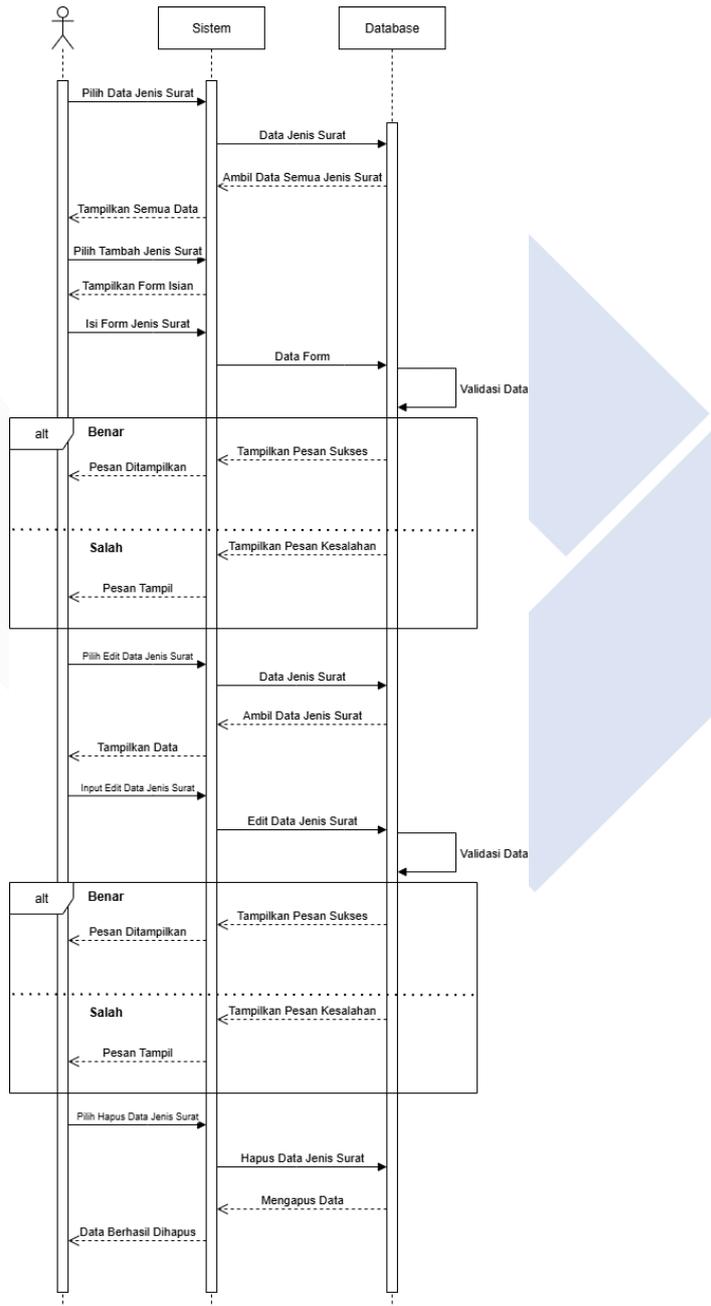


Gambar 3. 35. Sequence Diagram User Management

Pada Gambar 3.35 dijelaskan bahwa proses manajemen pengguna oleh *admin* dimulai ketika admin memilih menu “*User Management*”. Selanjutnya, basis data akan mengambil seluruh data pengguna, dan sistem akan menampilkannya kepada *admin*. Apabila *admin* ingin memperbarui data pengguna, maka *admin* dapat memilih tombol “*Edit Data User*”, sehingga sistem akan mengambil data pengguna dari basis data dan menampilkan halaman formulir edit. Setelah *admin* memperbarui data, sistem akan mengirimkan data tersebut ke basis data untuk divalidasi. Jika data yang dimasukkan valid, sistem akan menampilkan pesan sukses.

Sebaliknya, jika data tidak valid, maka akan ditampilkan pesan kesalahan. Selain itu, *admin* juga dapat menghapus data pengguna dengan menekan tombol “Hapus Data”, sehingga basis data akan menghapus data tersebut dan sistem akan menampilkan pesan bahwa data berhasil dihapus.

8. Sequence Diagram Data Jenis Surat



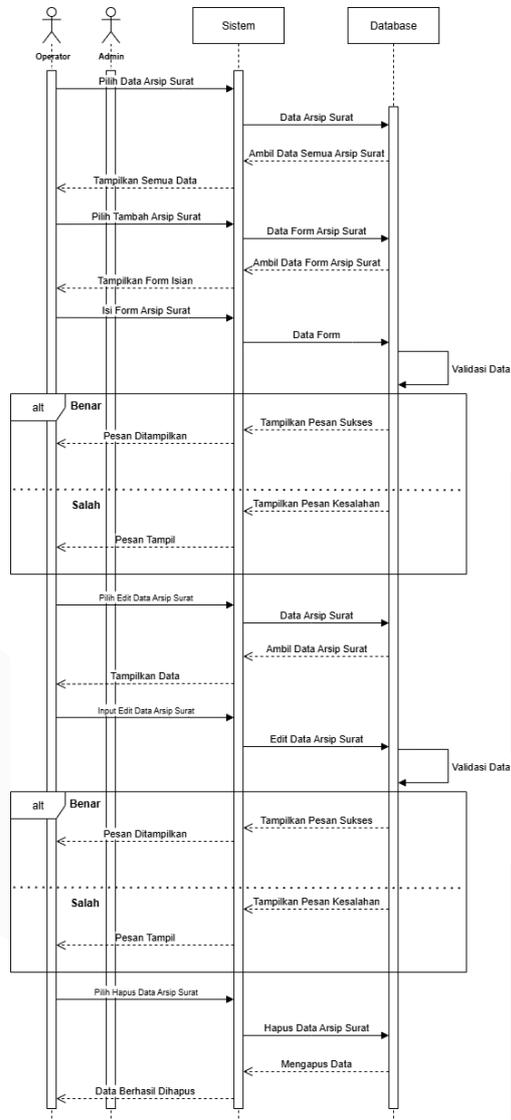
Gambar 3. 36. Sequence Diagram Data Jenis Surat

Pada Gambar 3.36 dijelaskan bahwa proses manajemen data jenis surat oleh *admin* dimulai ketika *admin* memilih menu “Jenis Surat”. Selanjutnya, basis data akan mengambil seluruh data jenis surat dan sistem akan menampilkannya kepada *admin*. Apabila *admin* ingin menambahkan jenis surat baru, maka *admin* dapat menekan tombol “Tambah Jenis Surat”, dan sistem akan menampilkan formulir isian. Setelah formulir diisi, data akan dikirimkan ke basis data untuk dilakukan validasi. Jika data valid, sistem akan menampilkan pesan sukses. Sebaliknya, jika data tidak valid, akan ditampilkan pesan kesalahan.

Jika *admin* ingin memperbaiki data jenis surat, maka *admin* dapat memilih tombol “Edit Data Jenis Surat”, sehingga sistem akan mengambil data terkait dari basis data dan menampilkan halaman formulir edit. Setelah perubahan dilakukan, sistem akan mengirimkan data ke basis data untuk divalidasi. Jika data valid, sistem akan menampilkan pesan sukses. Jika tidak, sistem akan menampilkan pesan kesalahan.

Selain itu, *admin* juga dapat menghapus data jenis surat dengan menekan tombol “Hapus Data Jenis Surat”. Sistem akan menghapus data tersebut dari basis data dan menampilkan pesan bahwa data berhasil dihapus.

9. Sequence Diagram Arsip Surat



Gambar 3. 37. Sequence Diagram Arsip Surat

Pada Gambar 3.37 dijelaskan bahwa proses manajemen arsip surat oleh *admin* atau *operator* dimulai ketika pengguna memilih menu “Arsip Surat”. Selanjutnya, basis data akan mengambil daftar data arsip surat, dan sistem akan menampilkannya kepada pengguna.

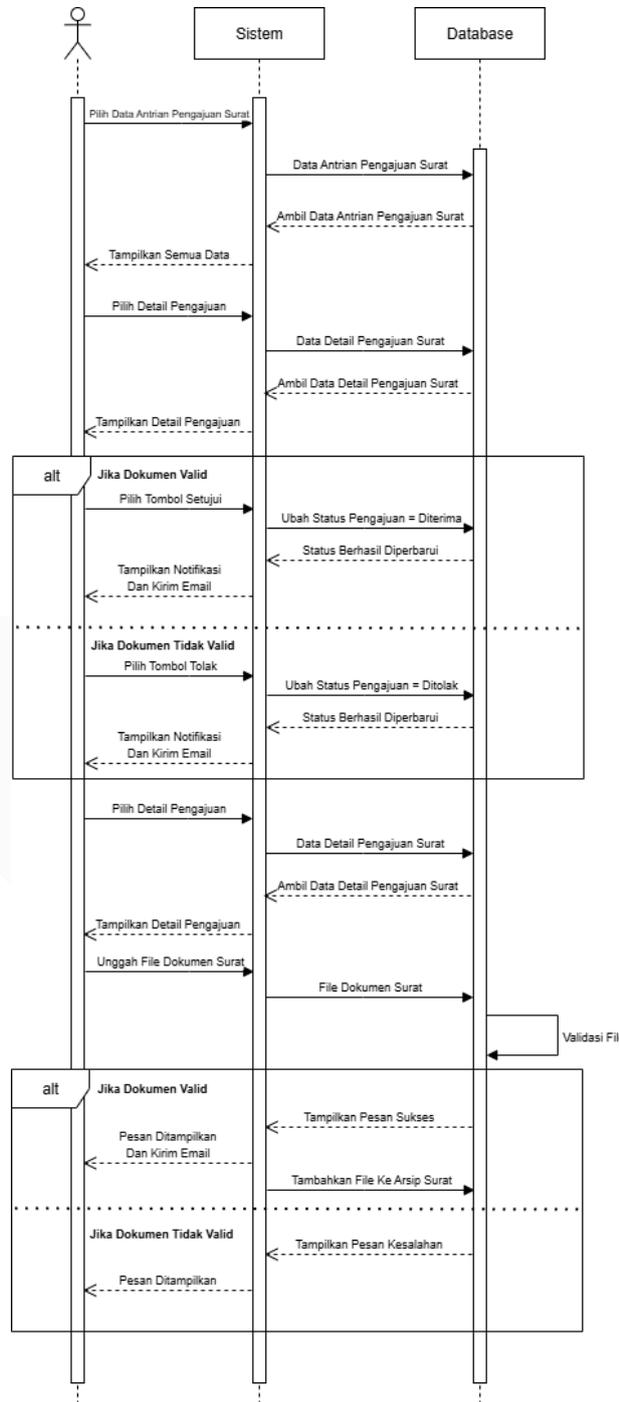
Apabila *admin* atau *operator* ingin menambahkan arsip surat baru, maka dapat menekan tombol “Tambah Arsip Surat”, kemudian sistem akan menampilkan formulir isian yang diambil dari basis data. Setelah formulir

diisi, sistem akan mengirimkan data ke basis data untuk divalidasi. Jika data valid, maka sistem akan menampilkan pesan sukses. Jika tidak valid, sistem akan menampilkan pesan kesalahan.

Jika *admin* atau *operator* ingin memperbarui data arsip surat, maka dapat memilih tombol “Edit Data Arsip Surat”, dan sistem akan mengambil data terkait dari basis data lalu menampilkan halaman formulir edit. Setelah perubahan dilakukan, sistem akan mengirimkan data ke basis data untuk divalidasi. Apabila data valid, akan ditampilkan pesan sukses. Jika tidak, maka akan ditampilkan pesan kesalahan.

Selain itu, *admin* atau *operator* juga dapat menghapus data arsip surat dengan menekan tombol “Hapus Data”. Sistem akan menghapus data dari basis data dan menampilkan pesan bahwa data berhasil dihapus.

10. Sequence Diagram Validasi Pengajuan Surat



Gambar 3. 38. Sequence Diagram Validasi Pengajuan Surat

Pada Gambar 3.38 dijelaskan bahwa proses manajemen antrian pengajuan surat oleh operator dimulai ketika operator memilih menu “Antrian Pengajuan Surat”. Selanjutnya, basis data akan mengambil seluruh data

antrian pengajuan, dan sistem akan menampilkannya kepada *operator*. Apabila *operator* ingin melihat detail pengajuan, *operator* dapat menekan tombol “Detail Pengajuan Surat”, kemudian sistem akan mengambil data detail dari basis data dan menampilkannya.

Operator dapat melakukan validasi terhadap pengajuan tersebut. Jika *operator* memilih tombol “Setujui”, maka sistem akan mengirimkan status persetujuan ke basis data, menampilkan notifikasi, dan mengirimkan email pemberitahuan kepada pemohon. Sebaliknya, jika *operator* memilih tombol “Tolak”, maka sistem akan mengirimkan status penolakan ke basis data, disertai notifikasi dan email kepada pemohon.

Apabila surat telah selesai dibuat, *operator* dapat mengunggah dokumen surat melalui halaman detail pengajuan. Setelah *file* diunggah, sistem akan mengirimkan dokumen ke basis data untuk divalidasi. Jika dokumen valid, sistem akan menampilkan pesan sukses serta mengirimkan notifikasi dan email kepada pemohon. Namun, jika dokumen tidak valid, maka sistem akan menampilkan pesan kesalahan.

3.4.3 Pengembangan

Pengembangan dilakukan berdasarkan hasil rancangan yang telah disusun sebelumnya. Setiap pengembangan dilakukan secara terstruktur untuk memastikan bahwa sistem yang dibangun sesuai dengan kebutuhan. Metode pengembangan yang digunakan terdiri dari beberapa tahapan utama sebagai berikut:

1. Pengembangan Antarmuka (*Frontend*)
 - a. Dikembangkan menggunakan *framework* Laravel, Tailwind dan bahasa pemrograman PHP.
 - b. Antarmuka dirancang agar responsif dan mudah digunakan.
 - c. Fitur verifikasi email guna memastikan validitas akun pengguna baru.
 - d. Sistem notifikasi untuk memberikan informasi status pengajuan surat
 - e. Tampilan *dashboard* untuk *admin* dan *operator* yang menampilkan data statistik

- f. Menyediakan fitur manajemen pengguna
 - g. *Admin* dapat mengelola jenis surat yang tersedia
 - h. *Operator* dapat melakukan verifikasi dokumen dan mengunggah surat yang telah diproses
 - i. *Admin* dan *Operator* dapat mengelola arsip surat
2. Pengembangan *Backend*
- a. Menggunakan basis data MySQL untuk menyimpan seluruh data, termasuk data *user*, jenis surat, pengajuan surat, serta penyimpanan dokumen yang telah diproses.
 - b. Sistem autentikasi untuk membedakan hak akses antara *user*, *operator*, dan *admin*.
3. Implementasi Fitur Utama
- a. Manajemen Surat: Pengguna dapat memilih jenis surat dan melakukan pengajuan secara digital
 - b. Verifikasi dan Proses Surat: *Operator* melakukan verifikasi kelengkapan data dan dokumen pendukung, kemudian mengunggah surat yang telah diproses.
 - c. Status Pengajuan: Pengguna dapat melihat perkembangan proses surat melalui halaman status pengajuan surat.
 - d. Unduh Surat: Setelah surat selesai diproses, pengguna dapat mengunduh surat dari sistem.
 - e. Arsip Surat: *Admin* dan *Operator* dapat mengelola data arsip surat yang mencakup penyimpanan, pengeditan, dan penghapusan surat-surat yang telah diproses untuk keperluan dokumentasi dan pencatatan administrasi.
4. Pengujian
- a. Pengujian dilakukan untuk memastikan bahwa setiap fitur pada sistem dapat berfungsi sesuai dengan yang telah dirancang, serta memberikan hasil yang akurat dan konsisten. Pengujian ini mencakup *Black Box Testing* serta *User Acceptance Testing (UAT)* untuk menilai apakah sistem telah memenuhi kebutuhan dan harapan pengguna akhir.

- b. Apabila ditemukan fitur yang tidak berjalan sebagaimana mestinya, maka akan dilakukan proses identifikasi kesalahan (*debugging*) dan perbaikan kode (*refactoring*) sebelum sistem memasuki tahap implementasi akhir.

3.4.3.1 Perhitungan Hasil Pengujian

Dalam proyek “Pengembangan Sistem Berbasis Web untuk Digitalisasi Administrasi Desa Rias”, dilakukan tahapan pengujian yang melibatkan tiga jenis pengguna, yaitu admin, operator, dan user (warga), dengan menggunakan metode *Black Box Testing* dan *User Acceptance Testing (UAT)*. Pengujian *Black Box* digunakan untuk memverifikasi keakuratan fungsi-fungsi utama dalam sistem berdasarkan skenario yang telah ditentukan tanpa memeriksa kode program secara langsung. Pengujian dilakukan oleh masing-masing aktor untuk memastikan bahwa seluruh fitur berjalan sesuai fungsinya dan layak digunakan dalam konteks layanan administrasi pelayanan surat di desa. Sementara itu, pengujian UAT bertujuan untuk memperoleh evaluasi secara langsung dari perspektif pengguna akhir terhadap pengalaman penggunaan sistem yang telah dikembangkan.

Dalam perhitungan hasil pengujian ini, terdapat dua aspek yang dievaluasi, yaitu persentase hasil dari pengujian *Black Box* dan hasil dari *User Acceptance Testing*. Untuk mengetahui persentase tersebut digunakan perhitungan sebagai berikut:

- a. Rumus Mencari Total Skor Maksimal

Skor Maksimal

$$= \text{Bobot tertinggi} \times \text{Jumlah pertanyaan} \times \text{Responden}$$

- b. Rumus Menghitung Hasil Pengujian

$$\text{Hasil Pengujian} = \left(\frac{\text{Total skor yang diperoleh}}{\text{Total skor maksimal}} \right) \times 100\%$$

Dari hasil pengujian yang sudah diperoleh, selanjutnya dilakukan analisis perbandingan terhadap kriteria kelayakan sistem. Kriteria kelayakan tersebut dapat dilihat sebagai berikut:

Tabel 3. 1. Kriteria Kelayakan

Rentang Kriteria	Kriteria
0%-20%	Sangat Tidak Layak
21%-40%	Tidak Layak
41%-60%	Kurang Layak
61%-80%	Layak
81%-100%	Sangat Layak

3.4.3.2 Validasi *Black Box*

Validasi *Black Box* disajikan dalam bentuk tabel pengujian berdasarkan masing-masing jenis pengguna (*admin*, *operator*, dan *user*). Setiap skenario diuji berdasarkan *input* dan *output* yang diharapkan untuk memastikan bahwa sistem dapat berjalan sesuai dengan kebutuhan fungsional. Pengujian ini bertujuan untuk memperoleh bukti bahwa seluruh fitur sistem telah bekerja dengan benar dan bebas dari kesalahan selama pengoperasian.

3.4.3.3 Validasi *User Acceptance Testing (UAT)*

Validasi *User Acceptance Testing (UAT)* disusun dalam bentuk kuisisioner yang diberikan kepada responden dengan peran sebagai pengguna sistem. Kuisisioner terdiri dari serangkaian pertanyaan yang mengevaluasi aspek-aspek seperti kemudahan penggunaan, kejelasan tampilan, kecepatan sistem, serta kepuasan terhadap fitur yang tersedia. Setiap pertanyaan menggunakan skala penilaian 1–5 untuk menilai tingkat persetujuan responden. Hasil dari UAT digunakan untuk menilai sejauh mana sistem telah memenuhi ekspektasi dan kebutuhan pengguna.

Tabel 3. 2. Validasi *User Acceptance Testing (UAT)*

No.	Pertanyaan	Skala Penilaian				
		1	2	3	4	5
1	Apakah Anda dapat login ke dalam sistem dengan mudah dan cepat?					

- 2 Apakah antarmuka sistem admin mudah dipahami dan digunakan?
- 3 Apakah fitur pengelolaan data pengguna berjalan dengan baik dan sesuai kebutuhan?
- 4 Apakah Anda dapat menambahkan, mengedit, dan menghapus jenis surat tanpa kendala?
- 5 Apakah sistem memudahkan Anda dalam mengelola arsip surat secara digital?
- 6 Apakah data pengguna dan jenis surat ditampilkan secara jelas dan akurat di dalam sistem?
- 7 Apakah Anda merasa puas menggunakan sistem ini untuk membantu pekerjaan administratif desa?

3.4.4 Implementasi

Implementasi merupakan tahap akhir dalam proses pengembangan sistem, di mana sistem yang telah dibangun dan diuji kemudian diterapkan secara nyata di lingkungan pengguna.

Dalam proyek “Pengembangan Sistem Berbasis Web untuk Digitalisasi Administrasi Desa Rias”, tahap implementasi dilakukan dengan cara mengunggah sistem ke *server hosting*, sehingga dapat diakses secara *online* oleh warga desa, *operator*, maupun *admin*. Selain itu, dilakukan juga pelatihan bagi aparat desa agar mampu mengoperasikan sistem secara mandiri, terutama dalam hal pengelolaan data warga, verifikasi pengajuan surat, dan pengunggahan surat yang sudah diproses.

3.5 Laporan

Laporan adalah dokumen yang menyajikan hasil dari seluruh proses penelitian atau pengembangan sistem, mulai dari identifikasi masalah, analisis kebutuhan,

pengembangan, implementasi, dan pengujian. Laporan bertujuan untuk memberikan gambaran menyeluruh mengenai proses yang telah dilakukan serta hasil yang dicapai.

Dalam proyek “Pengembangan Sistem Berbasis Web untuk Digitalisasi Administrasi Desa Rias”, laporan ini mencakup dokumentasi lengkap mengenai bagaimana sistem dirancang, fitur-fitur yang dikembangkan, hasil uji coba sistem, serta dampaknya terhadap efisiensi layanan surat-menyurat di lingkungan desa. Laporan ini juga menjadi bukti pertanggungjawaban terhadap pengembangan sistem dan dapat digunakan sebagai acuan untuk pengembangan lanjutan di masa mendatang.



BAB IV

PEMBAHASAN

4.1 Analisis Kebutuhan dan Perancangan Sistem

Pada tahap pengembangan sistem, langkah utama yang dilakukan adalah analisis kebutuhan dan perancangan sistem. Analisis ini bertujuan untuk memahami karakteristik sistem yang akan dibangun, mengidentifikasi fungsionalitas yang dibutuhkan, serta menggali harapan pengguna terhadap sistem yang dikembangkan.

Proses analisis dilakukan melalui metode observasi dan wawancara langsung dengan pihak Desa Rias, guna memperoleh pemahaman yang komprehensif mengenai alur administrasi yang berjalan saat ini serta tantangan yang dihadapi.

Hasil dari analisis menunjukkan bahwa terdapat tiga jenis aktor dalam sistem administrasi surat ini, yaitu *admin*, *operator*, dan *user* (warga). Adapun kebutuhan fungsional yang dibutuhkan dalam sistem ini adalah sebagai berikut:

A. Admin

- Melihat informasi statistik sistem, seperti jumlah total pengguna, aktivitas pengguna, serta jumlah pengajuan surat.
- Melihat notifikasi terbaru yang berkaitan dengan status pengajuan.
- Mengelola data pengguna, termasuk menambah, memperbarui, dan menghapus data pengguna.
- Mengelola data jenis surat, yang mencakup penambahan, pengeditan, dan penghapusan jenis surat sesuai kebutuhan administrasi desa.
- Mengakses, menambah, mengedit, dan menghapus arsip surat.

B. Operator

- Melihat informasi statistik sistem, seperti jumlah total pengguna, aktivitas pengguna, serta jumlah pengajuan surat.
- Melihat notifikasi terbaru mengenai status pengajuan.

- Mengelola proses validasi terhadap pengajuan surat yang masuk dari pengguna.
- Memberikan keputusan terhadap pengajuan surat, berupa menerima atau menolak pengajuan dengan menyertakan alasan.
- Mengunggah dokumen surat yang telah selesai diproses.
- Mengakses, menambah, mengedit, dan menghapus arsip surat.

C. User

- Melakukan verifikasi email setelah proses registrasi akun.
- Mengajukan permohonan surat dengan memilih jenis surat yang diinginkan dan mengisi data yang diperlukan.
- Melihat status pengajuan surat secara berkala.
- Mengunduh dokumen surat yang telah selesai diproses.
- Melihat riwayat pengajuan surat sebelumnya yang telah dilakukan.
- Mengubah atau memperbarui informasi profil pengguna.

4.2 Rancangan Database

Rancangan database merupakan tahap penting dalam pengembangan sistem, karena struktur database yang baik akan mendukung proses pengolahan data yang efisien dan terorganisir. Dalam sistem administrasi pelayanan surat di Desa Rias, database dirancang untuk menyimpan berbagai data yang berkaitan dengan pengguna, pengajuan surat, jenis surat, status pengajuan, serta arsip surat.

4.2.1 Database db_rias

Tabel	Tindakan
cache	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
cache_locks	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
failed_jobs	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
jenis_surat	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
jobs	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
job_batches	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
migrations	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
model_has_permissions	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
model_has_roles	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
notifications	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
notifikasi	★ Jelajahi Struktur Cari Tambahkan Hapus
password_reset_tokens	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
pengajuan_selesai	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
pengajuan_surat	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
permissions	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
roles	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
role_has_permissions	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
sessions	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
users	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus

Gambar 4. 1. Database db_rias

Dalam pengembangan sistem, database yang digunakan adalah db_rias yang terdiri dari 19 tabel. Beberapa tabel merupakan bagian inti dari sistem seperti *users*, *jenis_surat*, *pengajuan_surat*, *pengajuan_selesai*, dan *notifikasi*, sedangkan tabel lainnya merupakan bawaan dari Laravel.

Tabel seperti *cache*, *sessions*, *failed_jobs*, dan *migrations* dihasilkan secara otomatis oleh Laravel untuk keperluan manajemen migrasi, *cache*, sesi pengguna, serta sistem antrian. Meski tidak digunakan secara langsung dalam fungsionalitas sistem, tabel-tabel tersebut tetap diperlukan untuk memastikan stabilitas dan performa *framework* Laravel.

Dengan demikian, meskipun tidak semua tabel digunakan secara eksplisit dalam implementasi fitur, seluruh struktur tetap dipertahankan untuk mendukung ekosistem Laravel secara keseluruhan.

4.2.2 Tabel *users*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id	bigint		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
3	username	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
4	email	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
5	number_phone	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
6	email_verified_at	timestamp			Ya	NULL		
7	password	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
8	role	enum('admin', 'operator', 'user')	utf8mb4_unicode_ci		Tidak	user		
9	status	enum('aktif', 'nonaktif')	utf8mb4_unicode_ci		Tidak	aktif		
10	remember_token	varchar(100)	utf8mb4_unicode_ci		Ya	NULL		
11	created_at	timestamp			Ya	NULL		
12	updated_at	timestamp			Ya	NULL		

Gambar 4. 2. Tabel *users*

Gambar 4.2 menunjukkan struktur tabel *users* yang berfungsi untuk menyimpan data seluruh pengguna sistem, baik itu *admin*, *operator*, maupun *user*. Tabel ini merupakan bagian penting dari sistem karena digunakan dalam proses autentikasi dan manajemen pengguna. Beberapa atribut utama dalam tabel ini antara lain *id*, *name*, *email*, *password*, dan *role*, yang berfungsi untuk membedakan jenis pengguna.

4.2.3 Tabel *jenis_surat*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id	bigint		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
2	slug	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
3	nama_jenis	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
4	deskripsi	text	utf8mb4_unicode_ci		Ya	NULL		
5	form_fields	json			Tidak	Tidak ada		
6	created_at	timestamp			Ya	NULL		
7	updated_at	timestamp			Ya	NULL		

Gambar 4. 3. Tabel *jenis_surat*

Gambar 4.3 menunjukkan struktur tabel *jenis_surat* yang digunakan untuk menyimpan data seluruh jenis surat yang tersedia dalam sistem, yang nantinya dapat diajukan oleh pengguna (warga). Tabel ini memiliki peran penting dalam mendukung fleksibilitas sistem, karena setiap entri pada tabel ini menentukan jenis surat yang dapat diproses secara daring, termasuk struktur isian formulirnya.

Tabel ini memiliki beberapa atribut penting, antara lain *id* sebagai *primary key* yang berfungsi untuk mengidentifikasi setiap jenis surat secara unik. Atribut *slug* digunakan sebagai representasi teks dari *nama_jenis* yang diformat agar ramah terhadap URL. Nilai *slug* dibentuk dari nama jenis surat dengan mengganti spasi

antar kata menjadi tanda hubung (-), misalnya “Surat Keterangan Usaha” akan menjadi “surat-keterangan-usaha”. Atribut `nama_jenis` menyimpan nama surat yang akan ditampilkan kepada pengguna, sedangkan deskripsi digunakan untuk memberikan penjelasan singkat mengenai fungsi atau konteks penggunaan dari jenis surat tersebut.

Salah satu atribut yang paling penting dalam tabel ini adalah `form_fields`, yang bertipe JSON. Kolom ini digunakan untuk menyimpan struktur dinamis dari formulir pengajuan surat berdasarkan jenis surat tertentu. Melalui pendekatan ini, sistem dapat menghasilkan formulir secara otomatis tanpa perlu mengubah kode program secara langsung. Misalnya, untuk jenis surat “Surat Keterangan Usaha”, sistem dapat menampilkan isian seperti pengantar RT, KTP, dan jenis usaha berdasarkan definisi struktur yang tersimpan dalam kolom `form_fields`. Pendekatan ini memberikan fleksibilitas tinggi dalam pengelolaan jenis surat di masa mendatang, karena penambahan atau pengubahan formulir dapat dilakukan cukup dengan memperbarui isi dari kolom JSON tersebut.

4.2.4 Tabel pengajuan_surat

#	Nama	Jenis	Penyortiran	Atribut	Tak Terilai	Bawaan	Komentar	Ekstra
1	<code>id</code>	bigint		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
2	<code>user_id</code>	bigint		UNSIGNED	Tidak	Tidak ada		
3	<code>jenis_surat_id</code>	bigint		UNSIGNED	Tidak	Tidak ada		
4	<code>data_pengajuan</code>	json			Tidak	Tidak ada		
5	<code>status</code>	enum('pending', 'ditolak', 'diproses', 'selesai')	utf8mb4_unicode_ci		Tidak	pending		
6	<code>catatan</code>	varchar(255)	utf8mb4_unicode_ci		Ya	NULL		
7	<code>created_at</code>	timestamp			Ya	NULL		
8	<code>updated_at</code>	timestamp			Ya	NULL		

Gambar 4. 4. Tabel pengajuan_surat

Gambar 4.4 menunjukkan struktur tabel `pengajuan_surat` yang berfungsi untuk menyimpan data setiap pengajuan surat yang dilakukan oleh pengguna sistem, khususnya warga desa. Tabel ini memuat informasi penting seperti `user_id` untuk mengidentifikasi pengaju, serta `jenis_surat_id` untuk menandai jenis surat yang diajukan. Atribut `data_pengajuan` bertipe JSON digunakan untuk menyimpan isian data dinamis dari form pengajuan, yang berbeda-beda tergantung jenis suratnya.

Status pengajuan dikelola melalui atribut `status` yang menggunakan enumerasi dengan nilai seperti `pending`, `ditolak`, `diproses`, dan `selesai`, yang mencerminkan

tahapan proses pengajuan tersebut. Tabel ini juga memiliki kolom catatan yang memungkinkan *operator* menyisipkan pesan atau keterangan tambahan kepada pengaju. Jika *operator* tidak mengisi catatan, maka sistem akan menampilkan pesan default seperti “Pengajuan surat kamu telah diterima dan sedang diproses.” Data dalam tabel ini ditambahkan secara otomatis ketika warga mengajukan surat melalui sistem, di mana identitas pengaju, jenis surat, dan data pengajuan dikirim dan disimpan ke dalam database melalui proses *backend*.

4.2.5 Tabel pengajuan_selesai

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id 🔑	bigint		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
2	nama	varchar(255)	utf8mb4_unicode_ci		Ya	NULL		
3	jenis_surat_id 🔑	bigint		UNSIGNED	Ya	NULL		
4	pengajuan_id 🔑	bigint		UNSIGNED	Ya	NULL		
5	surat_diminta	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
6	created_at	timestamp			Ya	NULL		
7	updated_at	timestamp			Ya	NULL		

Gambar 4. 5. Tabel pengajuan_selesai

Gambar 4.5 menunjukkan struktur tabel pengajuan_selesai yang digunakan untuk menyimpan data pengajuan surat yang telah selesai diproses oleh *operator*. Tabel ini memuat beberapa atribut penting seperti nama yang merepresentasikan nama pengaju, jenis_surat_id yang menunjukkan jenis surat, pengajuan_id yang merujuk pada id pengajuan di tabel pengajuan_surat, serta surat_diminta yang berisi nama *file* dokumen surat yang telah dibuat.

Data pada tabel ini dapat ditambahkan secara otomatis oleh sistem saat *operator* menyelesaikan suatu pengajuan dengan mengunggah *file* surat melalui halaman detail pengajuan. Dalam proses ini, sistem akan menyimpan data terkait ke dalam tabel pengajuan_selesai dan sekaligus mengubah status pengajuan menjadi selesai. Selain secara otomatis, tabel ini juga dapat diisi secara manual oleh *operator* atau *admin* melalui halaman arsip surat, yang berguna jika terdapat surat yang diproses tidak melalui alur pengajuan digital.

4.2.6 Tabel notifikasi

#	Nama	Jenis	Penyortiran	Atribut	Tak Terilai	Bawaan	Komentar	Ekstra
1	id	bigint		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
2	pengaju_id	bigint		UNSIGNED	Tidak	Tidak ada		
3	pengajuan_surat_id	bigint		UNSIGNED	Ya	NULL		
4	jenis_surat_id	bigint		UNSIGNED	Tidak	Tidak ada		
5	judul	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
6	pesan	text	utf8mb4_unicode_ci		Tidak	Tidak ada		
7	tipe	enum('info', 'success', 'warning', 'error')	utf8mb4_unicode_ci		Tidak		info	
8	sudah_dibaca_operator	tinyint(1)			Tidak	0		
9	sudah_dibaca_masyarakat	tinyint(1)			Tidak	0		
10	dibaca_operator_pada	timestamp			Ya	NULL		
11	dibaca_masyarakat_pada	timestamp			Ya	NULL		
12	created_at	timestamp			Ya	NULL		
13	updated_at	timestamp			Ya	NULL		

Gambar 4. 6. Tabel notifikasi

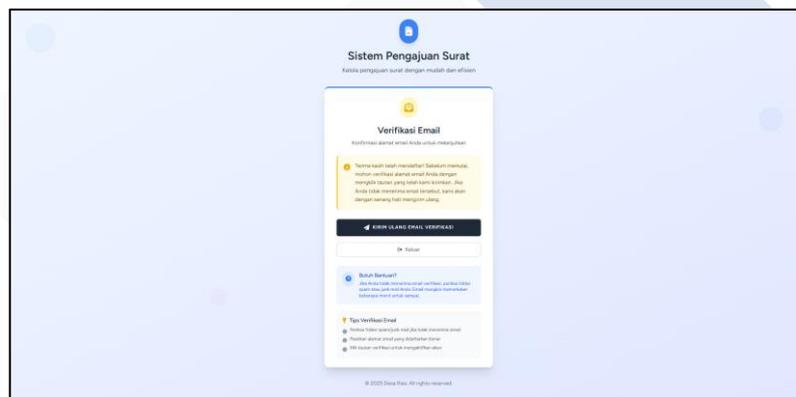
Gambar 4.6 menunjukkan struktur tabel notifikasi yang digunakan untuk menyimpan data notifikasi pengajuan surat yang tersedia dalam sistem. Tabel ini berfungsi sebagai sarana komunikasi internal antara warga sebagai pengaju dan *operator* sebagai pengelola pengajuan surat. Beberapa atribut penting dalam tabel ini antara lain *pengaju_id*, *pengajuan_surat_id*, dan *jenis_surat_id* yang masing-masing merepresentasikan relasi dengan pengguna, data pengajuan surat, serta jenis surat yang diajukan. Atribut *judul* dan *pesan* digunakan untuk menampilkan isi notifikasi kepada *operator*, yang secara otomatis dibuat saat warga melakukan pengajuan surat.

Selain itu, terdapat atribut *tipe* bertipe *enum* yang menyatakan jenis notifikasi seperti *info*, *success*, *warning*, atau *error*. Untuk mendukung pelacakan status pembacaan notifikasi, disediakan atribut *sudah_dibaca_operator* dan *sudah_dibaca_masyarakat* sebagai penanda apakah notifikasi telah dibaca oleh *operator* atau *user*. Dengan demikian, sistem dapat secara otomatis menandai notifikasi sebagai sudah dibaca ketika *operator* atau masyarakat membuka detail pengajuan terkait.

Pada Gambar 4.8 menunjukkan bahwa pada halaman registrasi, pengguna diwajibkan untuk mengisi beberapa data pribadi, yaitu nama lengkap, alamat email, nomor handphone, dan kata sandi. Data tersebut diperlukan untuk keperluan pembuatan akun dan proses autentikasi pengguna dalam sistem. Selain itu, terdapat tulisan navigasi bertuliskan “Sudah Punya Akun” yang berfungsi untuk menuju halaman login bagi pengguna yang sebelumnya telah memiliki akun dalam sistem.

4.3.3 Tampilan Verifikasi Email

Setelah proses pendaftaran selesai, sistem melakukan verifikasi email sebagai langkah keamanan tambahan. Proses ini wajib dilakukan sebelum pengguna dapat mengakses fitur pengajuan surat.

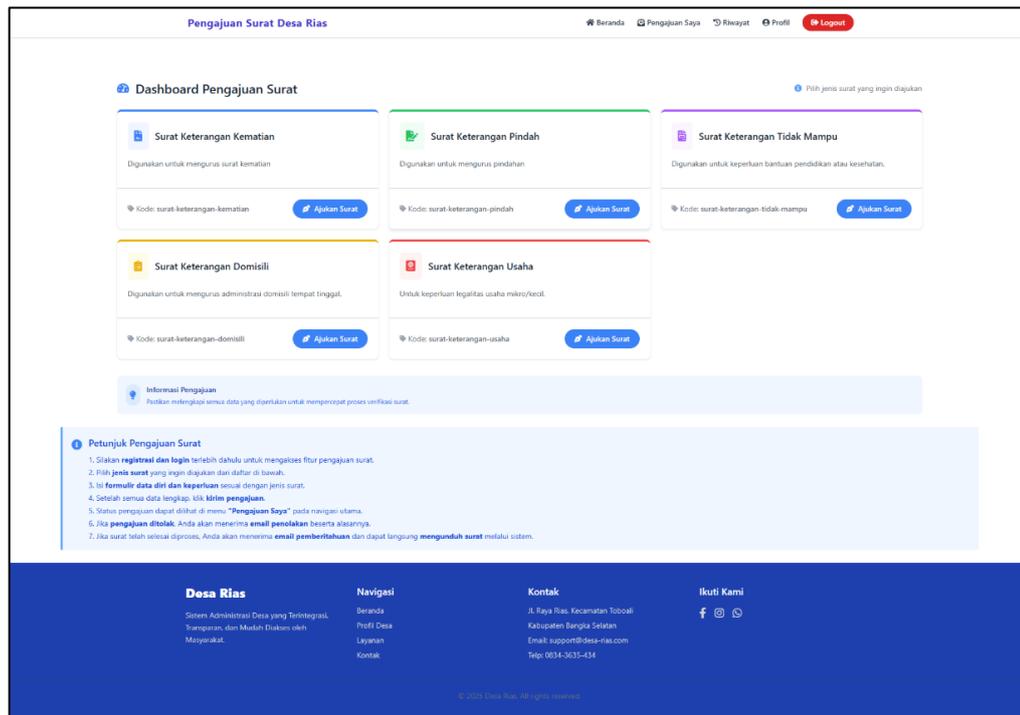


Gambar 4. 9. Tampilan Verifikasi Email

Pada Gambar 4.9 menunjukkan bahwa halaman verifikasi email akan ditampilkan ketika pengguna yang baru saja melakukan pendaftaran mencoba mengajukan surat melalui sistem. Pada tahap ini, pengguna diwajibkan untuk melakukan verifikasi terlebih dahulu dengan cara memeriksa kotak masuk pada alamat email yang telah didaftarkan, kemudian mengklik tautan verifikasi yang dikirimkan oleh sistem. Proses ini bertujuan untuk memastikan validitas identitas pengguna sebelum mengakses layanan administrasi lebih lanjut.

4.3.4 Tampilan Halaman Utama

Saat masuk ke dalam situs, pengguna diarahkan ke halaman utama pengajuan surat. Halaman ini menjadi pusat layanan bagi warga dalam melakukan permohonan surat secara daring.



Gambar 4. 10. Tampilan Halaman Utama

Pada gambar 4.10 menunjukkan tampilan halaman utama pengajuan surat. Pada halaman ini, pengguna dapat memilih jenis surat yang ingin diajukan, disertai dengan informasi serta petunjuk terkait proses pengajuan. Selain itu, halaman ini juga dilengkapi dengan menu navigasi yang terdiri atas “Beranda”, “Pengajuan Saya”, “Riwayat”, dan “Profil” yang memungkinkan pengguna untuk melihat data pribadi serta mengubah informasi akun. Di bagian bawah halaman terdapat *footer* yang memuat informasi kontak resmi Desa Rias sebagai media komunikasi antara warga dan pihak desa.

4.3.5 Tampilan *Form* Surat

Setelah memilih jenis surat yang akan diajukan, pengguna akan diarahkan ke halaman formulir pengajuan. Halaman ini dirancang agar proses pengisian data dapat dilakukan secara terstruktur dan mudah dipahami.

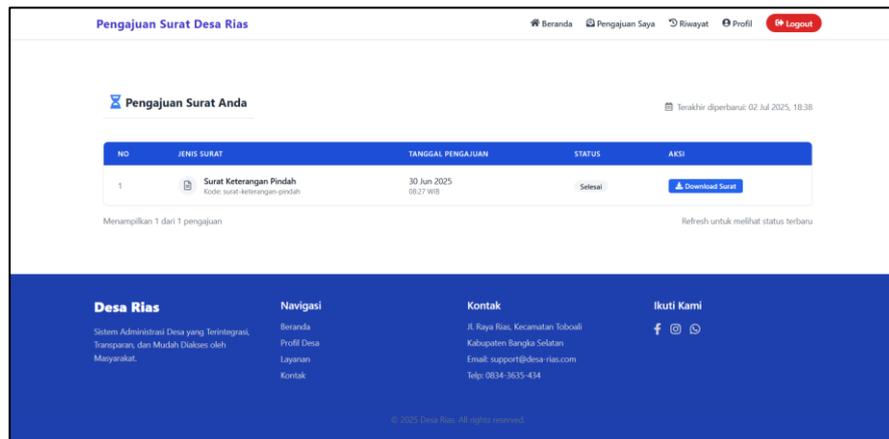
The screenshot displays a web application interface for a 'Formulir Pengajuan - Surat Keterangan Pindah'. The form is structured with several input fields: 'Nama' (text), 'Jenis Kelamin' (dropdown menu), 'Agama' (dropdown menu), 'Status Perkawinan' (dropdown menu), 'Pekerjaan' (text), 'Alamat Asal' (text area), 'Pindah ke' (text), 'Pada tanggal' (date picker), and 'Alasan Pindah' (text). A blue 'Kirim Pengajuan' button is located at the bottom left of the form. On the right side, there is a sidebar titled 'Alur Pengajuan' which lists five steps: 1. Isi formulir dengan lengkap dan benar, 2. Upload dokumen pendukung (jika ada), 3. Tekan tombol Kirim, 4. Tunggu konfirmasi dari admin desa, and 5. Ambil surat di kantor desa / Unduh PDF. The top navigation bar includes 'Beranda', 'Pengajuan Saya', 'Riwayat', 'Profil', and 'Logout'. The footer contains four columns of information: 'Desa Rias' (System Administration), 'Navigasi' (Home, Profile, Logout, Contact), 'Kontak' (Address, Email, Phone), and 'Ikuti Kami' (Social media icons).

Gambar 4. 11. Tampilan *Form* Surat

Pada Gambar 4.11 menunjukkan tampilan halaman formulir pengajuan surat. Pengguna dapat mengisi data yang diperlukan sesuai dengan jenis surat yang dipilih. Setelah seluruh data terisi dengan lengkap dan benar, pengguna dapat mengirimkan permohonan surat dengan menekan tombol “Kirim Pengajuan”. Selain itu, pada halaman ini juga disertakan informasi mengenai alur pengajuan sebagai panduan, agar pengguna memahami setiap tahapan dalam proses pengajuan surat secara digital.

4.3.6 Tampilan Status Pengajuan Surat

Untuk memudahkan pengguna dalam memantau proses administrasi, sistem menyediakan fitur pelacakan status pengajuan surat. Fitur ini dapat diakses melalui tombol navigasi “Pengajuan Saya” yang terletak di bagian atas halaman utama.

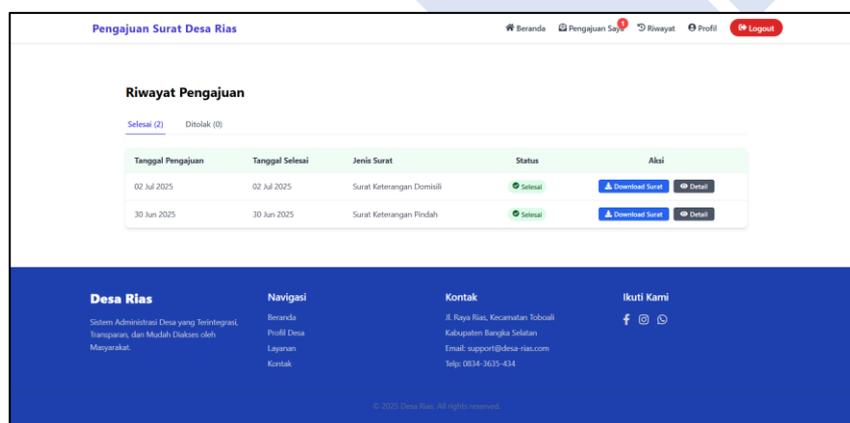


Gambar 4. 12. Tampilan Status Pengajuan Surat

Pada Gambar 4.12 menampilkan halaman status pengajuan surat yang telah diajukan oleh pengguna. Melalui halaman ini, pengguna dapat memantau perkembangan status pengajuan, apakah surat masih dalam proses atau telah selesai diverifikasi oleh pihak desa. Apabila surat telah diproses dan dinyatakan selesai, sistem akan menampilkan tombol “Download Surat” yang memungkinkan pengguna untuk mengunduh dokumen surat secara langsung dalam format digital.

4.3.7 Tampilan Riwayat Pengajuan Surat

Sebagai bentuk dokumentasi digital, sistem menyediakan halaman riwayat pengajuan surat yang memungkinkan pengguna untuk mengakses seluruh arsip pengajuan yang telah dilakukan. Fitur ini dapat diakses melalui menu navigasi “Riwayat” pada bagian atas halaman utama.

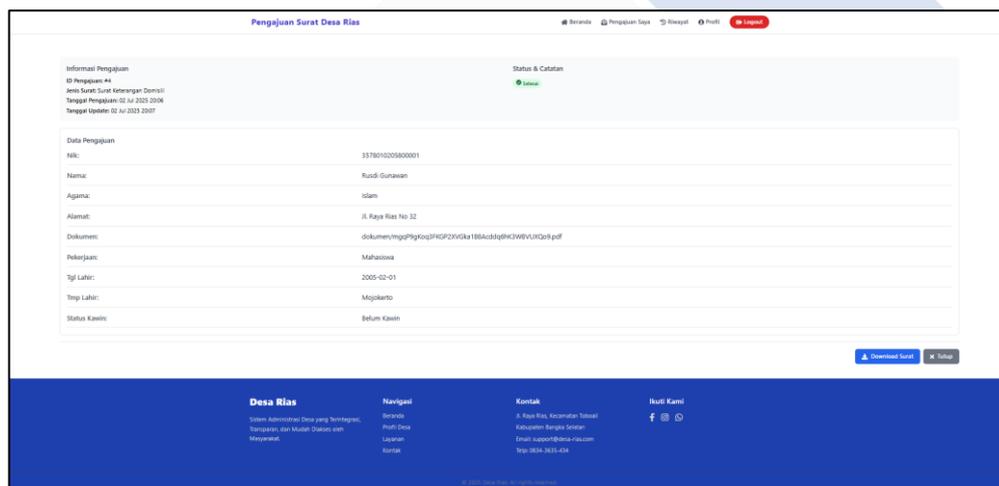


Gambar 4. 13. Tampilan Riwayat Pengajuan Surat

Pada Gambar 4.13 menampilkan halaman riwayat pengajuan surat milik pengguna. Pada halaman ini, pengguna dapat melihat daftar seluruh surat yang pernah diajukan, baik yang telah selesai diproses maupun yang ditolak oleh pihak desa. Selain itu, pengguna juga diberikan opsi untuk mengunduh surat yang telah selesai atau melihat detail informasi dari setiap pengajuan yang telah dilakukan. Fitur ini memudahkan pengguna dalam melakukan pelacakan terhadap dokumen administratif secara digital dan terorganisir.

4.3.8 Tampilan Detail Riwayat Pengajuan Surat

Untuk memberikan informasi yang lebih rinci terkait setiap pengajuan surat, sistem menyediakan halaman detail pengajuan. Halaman ini dapat diakses melalui tombol “Detail” pada daftar riwayat pengajuan surat.



Gambar 4. 14. Tampilan Detail Riwayat Pengajuan Surat

Pada Gambar 4.14 menampilkan halaman detail dari riwayat pengajuan surat. Pada halaman ini ditampilkan informasi lengkap terkait pengajuan. Selain itu, ditampilkan pula data pengajuan yang sebelumnya telah diisi oleh pengguna melalui formulir. Apabila surat telah selesai diproses, sistem akan menyediakan tombol “Download Surat” yang memungkinkan pengguna untuk mengunduh dokumen surat dalam format digital. Halaman ini dirancang untuk memberikan transparansi dan kemudahan akses terhadap informasi pengajuan secara menyeluruh.

4.3.9 Tampilan Profil Pengguna

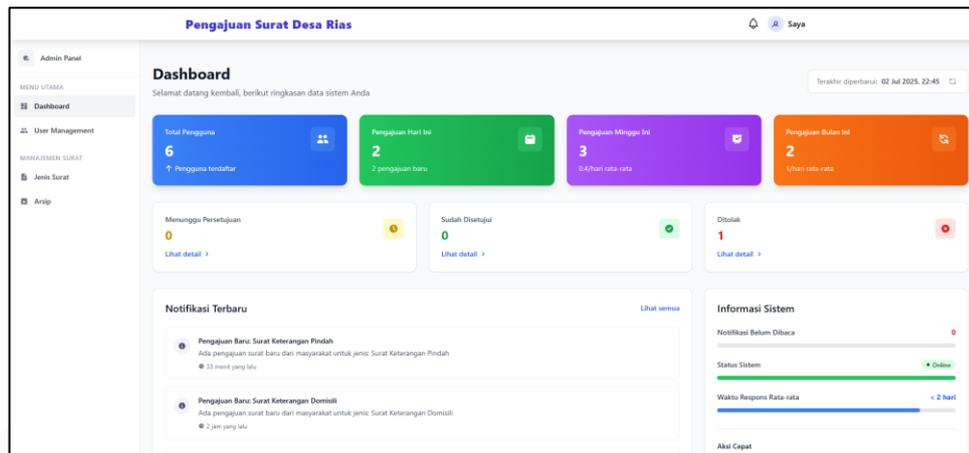
Sistem ini menyediakan halaman profil pengguna yang dapat diakses melalui tombol “Profil” pada navigasi di bagian atas halaman utama. Fitur ini memungkinkan pengguna untuk mengelola dan memperbarui informasi akun secara mandiri.

Gambar 4. 15. Tampilan Profil Pengguna

Pada Gambar 4.15 menampilkan halaman profil pengguna yang dimana pengguna dapat melakukan perubahan terhadap data pribadi seperti nama, alamat email, dan kata sandi. Setelah melakukan pengubahan data, pengguna dapat menyimpan perubahan tersebut dengan menekan tombol “Simpan”. Fitur ini disediakan untuk memberikan fleksibilitas bagi pengguna dalam memperbarui informasi akun secara mandiri dan aman.

4.3.10 Tampilan *Dashboard Admin*

Untuk mempermudah pengelolaan sistem, *admin* disediakan halaman *dashboard* yang menyajikan berbagai informasi penting secara ringkas dan terstruktur. Halaman ini dapat diakses oleh *admin* setelah berhasil *login* ke dalam sistem.



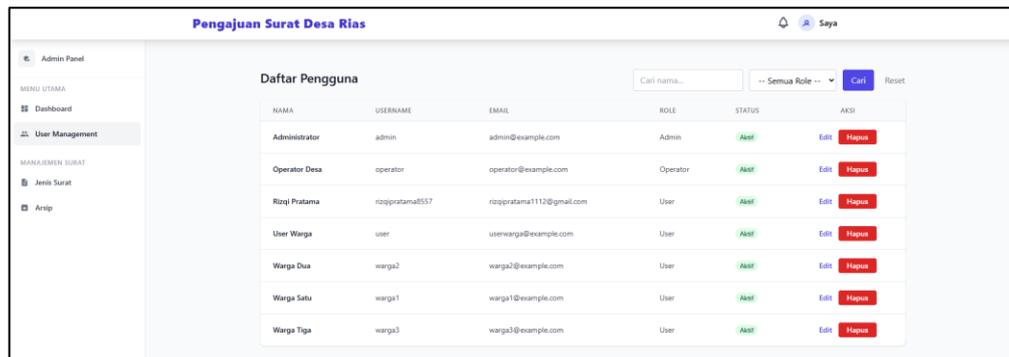
Gambar 4. 16. Tampilan *Dashboard Admin*

Pada Gambar 4.16 menampilkan tampilan halaman *dashboard admin*. Pada halaman ini, *admin* dapat melihat ringkasan informasi penting berupa data total pengguna, jumlah pengajuan surat pada hari ini, minggu ini, dan bulan ini. Selain itu, ditampilkan pula status pengajuan surat yang diklasifikasikan ke dalam tiga kategori, yaitu surat yang menunggu persetujuan, surat yang telah disetujui, dan surat yang ditolak.

Dashboard ini juga dilengkapi dengan fitur notifikasi terbaru yang menampilkan informasi terkait pengajuan baru dari warga, notifikasi sistem yang belum dibaca, status sistem saat ini, serta rata-rata waktu respons sistem. Untuk menunjang efisiensi kerja, tersedia pula tombol aksi cepat (*quick action*) yang memungkinkan *admin* langsung diarahkan ke halaman tertentu sesuai fungsi yang dibutuhkan. Fitur-fitur ini dirancang untuk memberikan kemudahan dan kecepatan dalam pengelolaan administrasi secara digital.

4.3.11 Tampilan *User Management*

Untuk mendukung pengelolaan akun secara menyeluruh, sistem menyediakan fitur *User Management* yang dapat diakses oleh *admin* melalui tombol “*User Management*” pada *sidebar* navigasi di bagian kanan halaman.



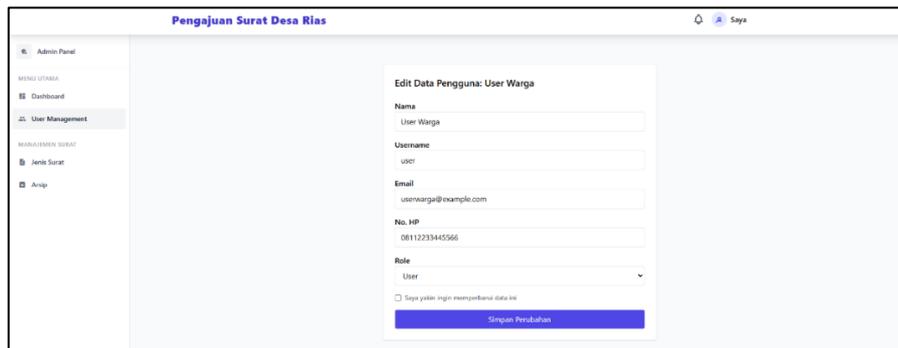
NAMA	USERNAME	EMAIL	ROLE	STATUS	AKSI
Administrator	admin	admin@example.com	Admin	Aktif	Edit Hapus
Operator Desa	operator	operator@example.com	Operator	Aktif	Edit Hapus
Rizqi Pratama	rizqipratama8557	rizqipratama1112@gmail.com	User	Aktif	Edit Hapus
User Warga	user	userwarga@example.com	User	Aktif	Edit Hapus
Warga Dua	warga2	warga2@example.com	User	Aktif	Edit Hapus
Warga Satu	warga1	warga1@example.com	User	Aktif	Edit Hapus
Warga Tiga	warga3	warga3@example.com	User	Aktif	Edit Hapus

Gambar 4. 17. Tampilan *User Management*

Pada Gambar 4.17 menampilkan tampilan halaman manajemen pengguna yang berfungsi untuk mengelola data pengguna dalam sistem. Melalui halaman ini, *admin* dapat melakukan pencarian pengguna secara spesifik dengan menggunakan *filter* berdasarkan nama maupun peran (*role*) pengguna. Selain itu, *admin* juga diberikan hak akses untuk melakukan perubahan terhadap data pengguna yang telah terdaftar, serta menghapus data pengguna apabila diperlukan. Fitur ini dirancang untuk mempermudah pengelolaan akun dalam sistem administrasi secara efektif dan terstruktur.

4.3.12 Tampilan Edit Data *User*

Sebagai bagian dari fitur manajemen pengguna, sistem menyediakan halaman khusus untuk melakukan pembaruan data pengguna. Halaman ini diakses oleh *admin* ketika memilih opsi edit pada daftar pengguna di halaman *user management*.



Gambar 4. 18. Tampilan Edit Data *User*

Pada Gambar 4.18 menampilkan tampilan halaman pengelolaan data pengguna yang digunakan oleh *admin* untuk memperbarui informasi pengguna dalam sistem. *Admin* dapat melakukan perubahan terhadap beberapa atribut pengguna, seperti nama, *username*, email, nomor handphone, serta *role* pengguna yang tersedia, yaitu *user*, *operator*, dan *admin*. Sebelum melakukan penyimpanan perubahan, terdapat *checkbox* untuk konfirmasi sebagai bentuk validasi tindakan. Setelah *checkbox* diaktifkan, *admin* dapat menekan tombol “Simpan Perubahan” untuk menyelesaikan proses pembaruan data.

4.3.13 Tampilan Data Jenis Surat

Untuk mengakomodasi berbagai keperluan administrasi, sistem menyediakan fitur pengelolaan jenis surat. Fitur ini dapat diakses oleh *admin* melalui menu jenis surat pada *sidebar*, dan digunakan untuk menyesuaikan jenis-jenis surat yang tersedia dalam sistem.

#	Nama Jenis	Deskripsi	Jumlah Field	Aksi
1	Surat Keterangan Nikah	Digunakan untuk mengurus pernikahan	21	Edit Hapus
2	Surat Keterangan Pindah	Digunakan untuk mengurus pindahan	9	Edit Hapus
3	Surat Keterangan Tidak Mampu	Digunakan untuk keperluan bantuan pendidikan atas...	7	Edit Hapus
4	Surat Keterangan Domisili	Digunakan untuk mengurus administrasi domisili tem...	9	Edit Hapus
5	Surat Keterangan Usaha	Untuk keperluan legalitas usaha mikro/kecil.	3	Edit Hapus

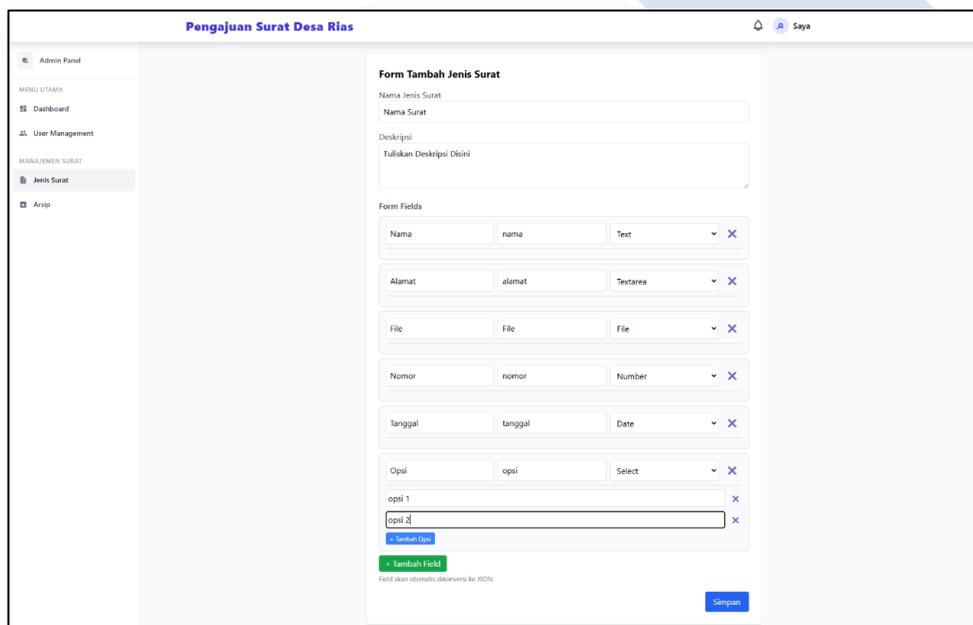
Gambar 4. 19. Tampilan Data Jenis Surat

Pada Gambar 4.19 menampilkan tampilan halaman data jenis surat, di mana tersedia tabel yang memuat daftar jenis surat yang telah terdaftar dalam sistem. Tabel ini menyajikan informasi penting seperti nama jenis surat, deskripsi singkat,

serta jumlah *form field* yang diperlukan untuk setiap jenis surat. *Admin* memiliki akses untuk menambahkan jenis surat baru dengan menekan tombol “Tambah”, serta dapat melakukan tindakan lebih lanjut melalui tombol aksi seperti “Edit” untuk memperbarui data, dan “Hapus” untuk menghapus jenis surat yang tidak lagi digunakan. Fitur ini memberikan kemudahan bagi *admin* untuk mengelola jenis surat secara fleksibel dan terstruktur sesuai kebutuhan layanan administrasi desa.

4.3.14 Tampilan Tambah Jenis Surat

Untuk menambah jenis surat baru yang sesuai dengan kebutuhan administrasi, sistem menyediakan halaman khusus bagi *admin* untuk mendefinisikan detail dan struktur form dari surat yang akan ditambahkan. Halaman ini dapat diakses melalui tombol “Tambah” pada halaman Data Jenis Surat.



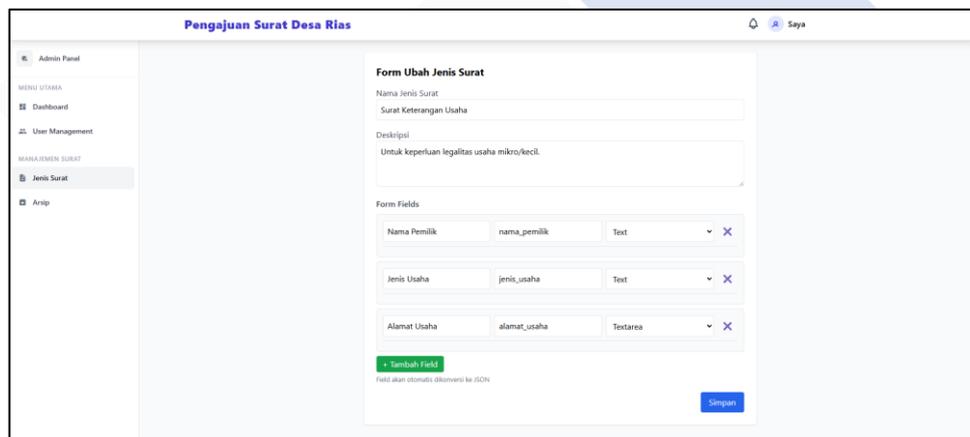
Gambar 4. 20. Tampilan Tambah Jenis Surat

Pada Gambar 4.20 menampilkan tampilan halaman tambah jenis surat yang digunakan oleh *admin* saat ingin menambahkan jenis surat baru ke dalam sistem. Pada halaman ini, terdapat beberapa kolom input yang harus diisi, seperti nama jenis surat dan deskripsi surat. Selain itu, *admin* juga dapat menambahkan *form field* sesuai kebutuhan masing-masing jenis surat, dengan berbagai opsi tipe *input*, antara lain *text* untuk isian teks umum, *textarea* untuk alamat atau keterangan panjang, *file*

untuk unggahan dokumen, *number* untuk angka, *date* untuk tanggal, serta *select* untuk pilihan yang telah ditentukan. Setelah seluruh data terisi, *admin* dapat menyimpan jenis surat baru tersebut dengan menekan tombol “Simpan”. Fitur ini memberikan fleksibilitas dalam mengatur struktur form surat sesuai kebutuhan administrasi desa.

4.3.15 Tampilan Edit Jenis Surat

Untuk melakukan perubahan pada jenis surat yang telah terdaftar, sistem menyediakan halaman edit yang dapat digunakan *admin* untuk memperbarui informasi secara langsung tanpa harus membuat ulang. Halaman ini dapat diakses melalui tombol “Edit” pada daftar jenis surat yang tersedia.



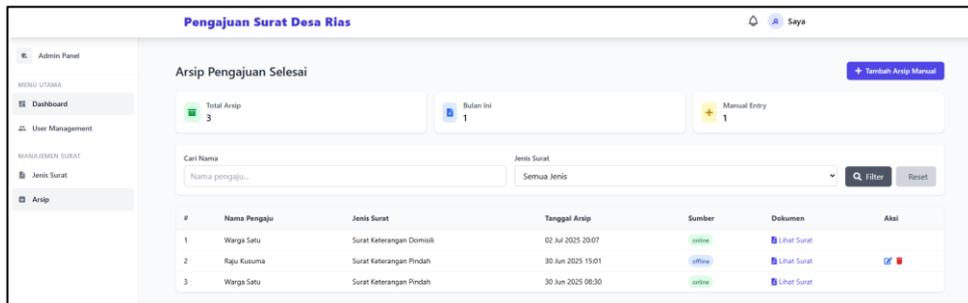
The screenshot displays the 'Form Ubah Jenis Surat' interface. It features a sidebar on the left with a menu including 'Admin Panel', 'MENU UTAMA' (Dashboard, User Management), and 'MANAJEMEN SURAT' (Jenis Surat, Arsip). The main form area is titled 'Form Ubah Jenis Surat' and contains the following fields: 'Nama Jenis Surat' (filled with 'Surat Keterangan Usaha'), 'Deskripsi' (filled with 'Untuk keperluan legalitas usaha mikro/kecil'), and a 'Form Fields' section. This section lists three fields: 'Nama Pemilik' (type: Text), 'Jenis Usaha' (type: Text), and 'Alamat Usaha' (type: Textarea). A '+ Tambah Field' button is located below the list. At the bottom right of the form is a blue 'Simpan' button.

Gambar 4. 21. Tampilan Edit Jenis Surat

Pada Gambar 4.21 menampilkan tampilan halaman edit jenis surat yang sudah ada di dalam sistem. Untuk pengisian data, tata letaknya serupa dengan halaman tambah jenis surat, hanya saja seluruh kolom sudah terisi dengan data sebelumnya sehingga *admin* dapat melakukan pembaruan terhadap nama surat, deskripsi, maupun struktur *form field* yang ada. Setelah perubahan dilakukan, *admin* dapat menekan tombol “Simpan” untuk memperbarui data jenis surat yang bersangkutan. Fitur ini memudahkan *admin* dalam melakukan penyesuaian terhadap jenis surat tanpa harus membuat ulang dari awal.

4.3.16 Tampilan Arsip Surat (*Admin*)

Sebagai bagian dari pengelolaan dokumen digital, sistem menyediakan halaman arsip surat untuk *admin* yang berfungsi untuk menyimpan dan mengatur surat-surat yang telah diproses. Halaman ini membantu *admin* dalam pelacakan, pencarian, dan pengelolaan arsip secara terstruktur.



#	Nama Pengaju	Jenis Surat	Tanggal Arsip	Sumber	Dokumen	Aksi
1	Warga Satu	Surat Keterangan Domisili	02 Jul 2025 20:07	Online	Lihat Surat	
2	Raju Kusuma	Surat Keterangan Pindah	30 Jun 2025 15:01	Offline	Lihat Surat	Edit Hapus
3	Warga Satu	Surat Keterangan Pindah	30 Jun 2025 08:30	Online	Lihat Surat	

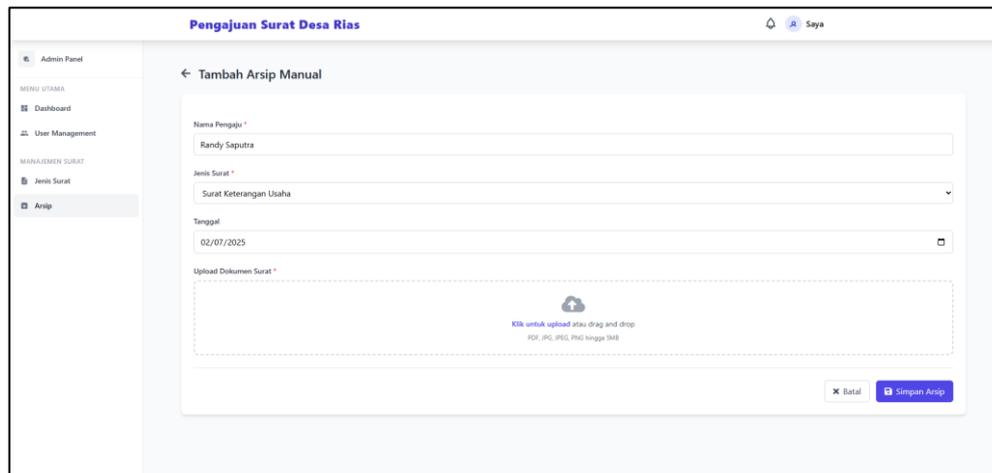
Gambar 4. 22. Tampilan Arsip Surat (*Admin*)

Pada Gambar 4.22 menampilkan tampilan halaman arsip surat pada sistem. Terdapat ringkasan data berupa total arsip yang tersimpan, jumlah arsip yang ditambahkan pada bulan ini, serta jumlah arsip yang dimasukkan secara manual. Terdapat pula fitur pencarian untuk mencari arsip berdasarkan nama pengaju atau jenis surat.

Data arsip ditampilkan dalam bentuk tabel yang memuat informasi seperti nama pengaju, jenis surat, tanggal pengarsipan, serta sumber asal arsip (apakah berasal dari sistem atau ditambahkan secara manual). Pada setiap entri arsip, tersedia tombol aksi seperti edit dan hapus, namun tombol tersebut hanya aktif apabila arsip yang dimaksud merupakan arsip manual. Jika *admin* ingin menambahkan arsip surat secara manual, maka dapat dilakukan dengan menekan tombol “Tambah Arsip Manual”.

4.3.17 Tampilan Tambah Arsip Surat (*Admin*)

Untuk mengakomodasi surat-surat yang diperoleh di luar proses pengajuan digital, sistem menyediakan fitur penambahan arsip secara manual. Fitur ini penting agar semua dokumen tetap terintegrasi dalam sistem, termasuk surat yang diterima secara langsung oleh pihak desa.

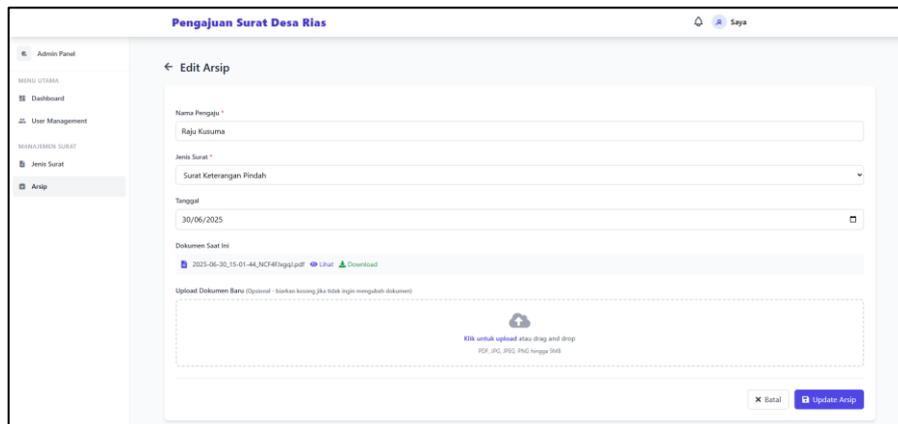


Gambar 4. 23. Tampilan Tambah Arsip Surat (*Admin*)

Pada Gambar 4.23 menampilkan tampilan halaman tambah arsip surat, yang digunakan ketika *admin* ingin menambahkan arsip surat secara manual ke dalam sistem. Pada halaman ini tersedia beberapa kolom input, antara lain nama pengaju, jenis surat yang datanya diambil langsung dari database yang telah terdaftar di sistem, tanggal penyimpanan surat, serta fitur unggah dokumen yang memungkinkan *admin* mengunggah *file* surat dalam format digital. Fitur ini disediakan untuk mendukung fleksibilitas dalam pengelolaan arsip, terutama untuk surat-surat yang tidak melalui proses pengajuan secara langsung oleh pengguna melalui sistem.

4.3.18 Tampilan Edit Arsip Surat (*Admin*)

Untuk menjaga keakuratan dan keterbaruan data arsip, sistem menyediakan halaman edit arsip yang dapat digunakan untuk melakukan pembaruan terhadap arsip surat yang telah ditambahkan sebelumnya. Halaman ini dapat diakses dengan menekan tombol edit pada entri arsip manual.

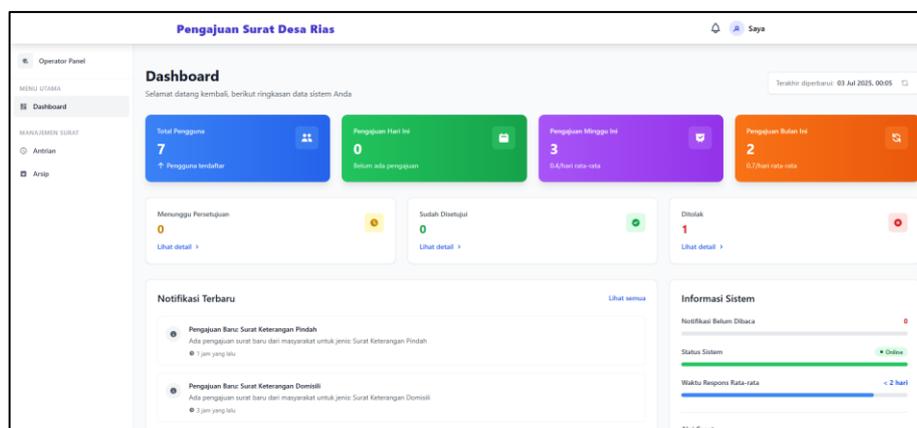


Gambar 4. 24. Tampilan Edit Arsip Surat (*Admin*)

Pada Gambar 4.24 menampilkan tampilan halaman edit arsip surat ketika *admin* menekan tombol edit pada arsip yang sudah ada. Halaman ini memungkinkan *admin* untuk memperbarui data arsip, seperti nama pengaju, jenis surat, tanggal penyimpanan, serta dokumen surat yang telah diunggah sebelumnya. Seluruh kolom *input* akan terisi secara otomatis dengan data arsip yang telah ada, sehingga *admin* hanya perlu mengubah bagian yang diperlukan. Setelah melakukan perubahan, *admin* dapat menekan tombol “Simpan” untuk memperbarui data arsip dalam sistem.

4.3.19 Tampilan *Dashboard Operator*

Untuk mendukung tugas *operator*, sistem menyediakan *dashboard* yang dirancang agar *operator* dapat bekerja secara efisien tanpa terganggu oleh menu yang tidak relevan dengan tugasnya.



Gambar 4. 25. Tampilan *Dashboard Operator*

Pada Gambar 4.25 menampilkan tampilan halaman *dashboard* bagi *operator*. *Operator* dapat melihat ringkasan informasi utama yang mencakup total jumlah pengguna, jumlah pengajuan surat pada hari ini, minggu ini, dan bulan ini. Selain itu, ditampilkan juga rekap status pengajuan surat yang terbagi menjadi tiga kategori, yaitu surat yang masih menunggu validasi, surat yang telah disetujui, serta surat yang ditolak.

Dashboard operator turut dilengkapi dengan fitur notifikasi terbaru yang memberikan informasi terkait pengajuan masuk dari warga, notifikasi sistem yang belum dibaca, status sistem secara umum, dan rata-rata waktu respons. Untuk meningkatkan efisiensi kerja *operator*, disediakan tombol aksi cepat (*quick action*) yang memudahkan navigasi langsung menuju halaman yang berkaitan dengan pengelolaan pengajuan surat.

Berbeda dari tampilan *dashboard admin*, pada *dashboard operator sidebar* hanya menampilkan menu yang relevan dengan tugas *operator*, yaitu *Dashboard*, *Antrian Pengajuan Surat*, dan *Arsip Surat*. Penyederhanaan menu ini bertujuan agar *operator* dapat lebih fokus dalam melakukan validasi dan pengelolaan surat secara efektif dan terarah.

4.3.20 Tampilan Antrian Pengajuan Surat

Untuk membantu *operator* dalam menangani permohonan surat yang masuk, sistem menyediakan halaman khusus antrian pengajuan surat. Halaman ini merupakan pusat kontrol bagi *operator* dalam melakukan proses validasi dan tindak lanjut terhadap setiap pengajuan yang diterima.

The screenshot shows a dashboard titled 'Pengajuan Surat Desa Rias' with a user profile 'Saya' in the top right. On the left is a sidebar menu with 'Operator Panel', 'MENU UTAMA', 'Dashboard', 'MANAJEMEN SURAT', 'Antrian', and 'Arsip'. The main content area is titled 'Daftar Pengajuan Surat' and includes a filter bar with buttons for 'Semua', 'Pending', 'Ditolak', 'Diproses', and 'Selesai'. Below the filter is a table with the following data:

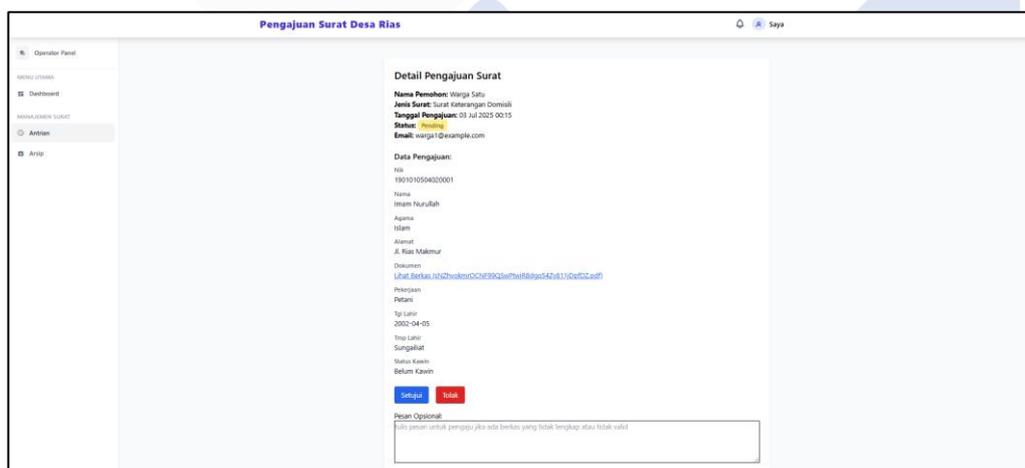
#	Nama Pengaju	Jenis Surat	Tanggal Pengajuan	Status	Detail
1	Warga Satu	Surat Keterangan Pindah	30 Jun 2025	Selesai	Detail
2	Warga Satu	Surat Keterangan Domisili	02 Jul 2025	Selesai	Detail
3	Warga Satu	Surat Keterangan Pindah	02 Jul 2025	Ditolak	Detail
4	Warga Satu	Surat Keterangan Domisili	03 Jul 2025	Pending	Detail

Gambar 4. 26. Tampilan Antrian Pengajuan Surat

Pada Gambar 4.26 menampilkan tampilan halaman antrian pengajuan surat yang diperuntukkan bagi *operator*. Pada halaman ini, *operator* dapat menggunakan fitur *filter* untuk menyaring pengajuan surat berdasarkan status tertentu, seperti semua, *pending*, ditolak, diproses, atau selesai. Data pengajuan ditampilkan dalam bentuk tabel yang terdiri dari beberapa kolom informasi, antara lain nama pengaju, jenis surat, tanggal pengajuan, status pengajuan, serta tombol “Detail” yang dapat digunakan untuk mengakses informasi lengkap dan melakukan proses validasi terhadap pengajuan surat tersebut. Fitur ini dirancang untuk mempermudah *operator* dalam mengelola dan menindaklanjuti setiap permohonan surat yang masuk secara efisien dan terstruktur.

4.3.21 Tampilan Detail Pengajuan Surat

Setelah menekan tombol “Detail” pada halaman antrian pengajuan, *operator* akan diarahkan ke halaman detail pengajuan. Halaman ini berfungsi sebagai ruang verifikasi utama di mana *operator* dapat meninjau seluruh informasi yang diperlukan sebelum mengambil keputusan terhadap pengajuan yang masuk.



Gambar 4. 27. Tampilan Detail Pengajuan Surat

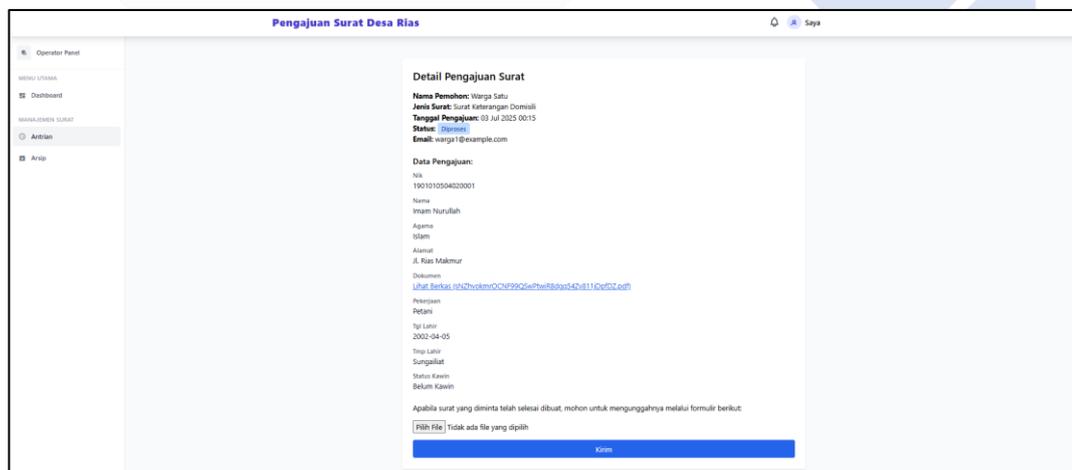
Pada Gambar 4.27 menampilkan tampilan halaman detail pengajuan surat yang diakses oleh *operator*. Pada halaman ini, ditampilkan informasi lengkap mengenai pengajuan surat, antara lain nama pemohon, jenis surat yang diajukan, tanggal pengajuan, status pengajuan, alamat email pemohon, serta data formulir yang telah diisi oleh pengguna saat melakukan pengajuan. Selain itu, *operator* juga dapat

melihat dokumen pendukung yang telah diunggah oleh pengguna sebagai syarat pengajuan.

Melalui halaman ini, *operator* diberikan opsi untuk melakukan validasi terhadap pengajuan, yaitu dengan menyetujui atau menolak permohonan surat berdasarkan kelengkapan dan kebenaran data. *Operator* juga dapat menambahkan pesan atau catatan kepada pengguna sebagai bentuk komunikasi atau pemberitahuan terkait hasil proses validasi. Fitur ini dirancang untuk mendukung proses administrasi secara transparan, akuntabel, dan efisien.

4.3.22 Tampilan Detail Proses Pengajuan Surat

Setelah pengajuan surat disetujui, sistem akan menampilkan versi lanjutan dari halaman detail pengajuan. Pada tahap ini, fokus utama bergeser dari proses validasi ke proses unggah dokumen sebagai bentuk tindak lanjut dari permohonan yang telah disetujui.



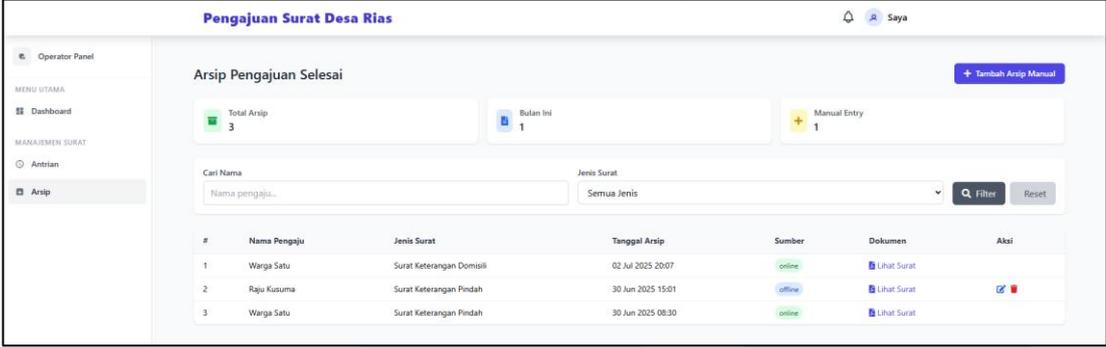
Gambar 4. 28. Tampilan Detail Proses Pengajuan Surat

Pada Gambar 4.28 menampilkan tampilan halaman detail pengajuan surat yang diakses oleh *operator* ketika status pengajuan berada dalam kondisi diproses atau telah disetujui. Perbedaan utama dibandingkan dengan halaman detail sebelumnya adalah pada bagian aksi yang tersedia. Jika sebelumnya terdapat tombol “Setujui” dan “Tolak” untuk proses validasi, maka pada tahap ini elemen tersebut berubah menjadi *form* unggah dokumen.

Apabila surat yang diajukan telah selesai dibuat, *operator* dapat mengunggah *file* surat tersebut melalui *form* unggah yang telah disediakan. Setelah dokumen berhasil diunggah, sistem akan secara otomatis mengirimkan informasi kepada pengguna bahwa surat telah tersedia dan dapat diunduh melalui akun masing-masing. Fitur ini dirancang untuk mempercepat proses distribusi dokumen serta meningkatkan efisiensi dalam layanan administrasi surat menyurat secara digital.

4.3.23 Tampilan Arsip Surat (*Operator*)

Sebagai bagian dari tugas pengelolaan dokumen digital, sistem menyediakan halaman arsip surat khusus untuk *operator* yang berfungsi untuk memantau dan mengelola surat-surat yang telah diproses. Halaman ini memudahkan *operator* dalam melakukan pelacakan, pencarian, dan pengelolaan arsip secara terstruktur.



#	Nama Pengaju	Jenis Surat	Tanggal Arsip	Sumber	Dokumen	Aksi
1	Warga Satu	Surat Keterangan Domisili	02 Jul 2025 20:07	online	Lihat Surat	
2	Raju Kusuma	Surat Keterangan Pindah	30 Jun 2025 15:01	offline	Lihat Surat	Edit Hapus
3	Warga Satu	Surat Keterangan Pindah	30 Jun 2025 08:30	online	Lihat Surat	

Gambar 4. 29. Tampilan Arsip Surat (*Operator*)

Pada Gambar 4.29 menampilkan tampilan halaman arsip surat pada sistem dari sisi *operator*. Halaman ini menyajikan informasi ringkas berupa total arsip yang tersimpan, jumlah arsip yang ditambahkan pada bulan ini, serta jumlah arsip yang dimasukkan secara manual. Terdapat juga fitur pencarian yang memudahkan *operator* untuk mencari arsip berdasarkan nama pengaju atau jenis surat yang diarsipkan.

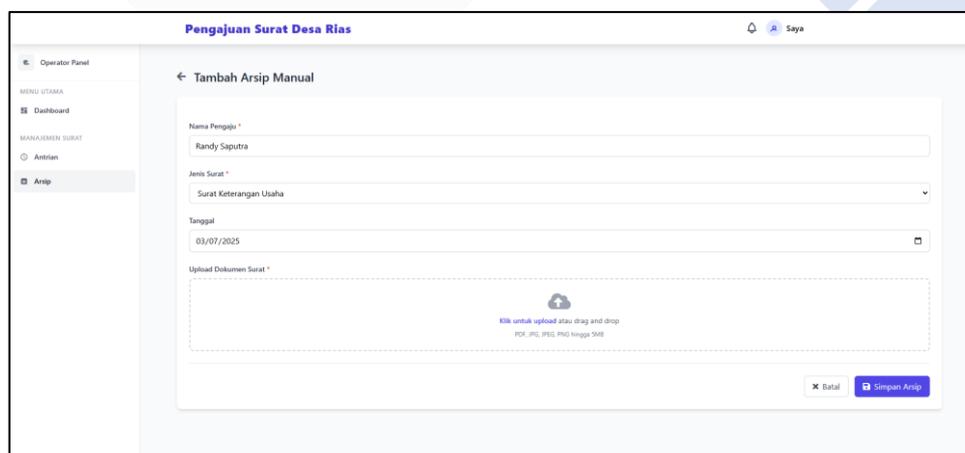
Data arsip ditampilkan dalam bentuk tabel yang berisi informasi seperti nama pengaju, jenis surat, tanggal pengarsipan, serta sumber asal arsip, baik yang berasal dari sistem melalui proses pengajuan pengguna, maupun yang ditambahkan secara manual. Setiap baris data arsip dilengkapi dengan tombol aksi seperti edit dan

hapus, namun tombol tersebut hanya dapat digunakan pada arsip yang ditambahkan secara manual.

Secara tampilan dan fungsionalitas, sistem arsip surat ini sama dengan yang digunakan oleh *admin*. *Operator* diberikan akses untuk mengelola arsip guna mendukung kelancaran proses layanan administrasi, termasuk apabila perlu menambahkan arsip secara manual dengan menekan tombol “Tambah Arsip Manual”. Fitur ini dirancang untuk memberikan fleksibilitas dalam pengelolaan dokumen surat secara digital, baik oleh *admin* maupun *operator*.

4.3.24 Tampilan Tambah Arsip Surat (*Operator*)

Untuk mendukung fleksibilitas dalam pengelolaan dokumen, sistem menyediakan halaman tambah arsip surat yang dapat diakses oleh *operator*. Halaman ini memungkinkan *operator* menambahkan arsip surat secara manual ke dalam sistem, khususnya untuk dokumen yang tidak melalui proses pengajuan daring oleh pengguna.



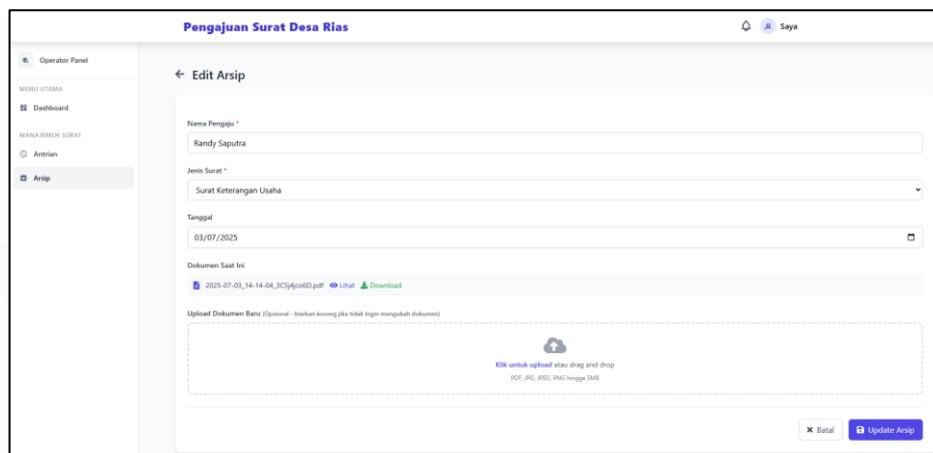
Gambar 4. 30. Tampilan Tambah Arsip Surat (*Operator*)

Pada Gambar 4.30 menampilkan tampilan halaman tambah arsip surat yang diakses oleh *operator* ketika ingin menambahkan arsip surat secara manual ke dalam sistem. Pada halaman ini tersedia beberapa kolom *input*, seperti nama pengaju, jenis surat yang datanya diambil langsung dari database jenis surat yang telah terdaftar dalam sistem, tanggal penyimpanan surat, serta fitur unggah dokumen yang memungkinkan *operator* mengunggah *file* surat dalam format digital. Fitur ini

memberikan fleksibilitas bagi *operator* dalam mengelola arsip surat, terutama untuk dokumen yang tidak melalui proses pengajuan oleh pengguna melalui sistem.

4.3.25 Tampilan Edit Arsip Surat (*Operator*)

Untuk memastikan data arsip surat tetap akurat dan terkini, sistem menyediakan halaman edit arsip surat yang dapat diakses oleh *operator*. Melalui halaman ini, *operator* dapat memperbarui informasi arsip secara efisien sesuai kebutuhan, sehingga pengelolaan dokumen digital tetap terjaga dengan baik.



Gambar 4. 31. Tampilan Edit Arsip Surat (*Operator*)

Pada Gambar 4.31 menampilkan tampilan halaman edit arsip surat yang diakses oleh *operator* ketika menekan tombol edit pada salah satu arsip yang telah tersimpan dalam sistem. Halaman ini memungkinkan *operator* untuk memperbarui data arsip, seperti nama pengaju, jenis surat, tanggal penyimpanan, serta dokumen surat yang sebelumnya telah diunggah.

Seluruh kolom *input* pada halaman ini akan otomatis terisi dengan data arsip yang telah ada, sehingga *operator* hanya perlu melakukan perubahan pada bagian yang diperlukan. Setelah proses pengeditan selesai, *operator* dapat menekan tombol “Update Arsip” untuk memperbarui data arsip ke dalam sistem. Fitur ini mempermudah *operator* dalam menjaga akurasi dan kelengkapan data arsip surat yang dikelola secara digital.

4.4 Implementasi

Pada tahap implementasi, dilakukan proses pengujian untuk memastikan semua fitur bekerja dengan baik dan sesuai dengan fungsinya. Metode pengujian *Black Box Testing* digunakan untuk menguji keluaran dari setiap fungsi sistem tanpa memperhatikan struktur internal atau kode program.

Untuk menguji sistem ini melibatkan *validator* dari kalangan ahli media dan teknologi informasi, yang memiliki pengalaman dalam pengembangan sistem informasi digital. Mereka bertugas untuk mengevaluasi kesesuaian antarmuka pengguna, rute navigasi, dan kemampuan setiap fitur yang tersedia. *Validator* memastikan bahwa sistem berfungsi sesuai dengan rancangan dan bahwa *Admin*, *Operator*, dan *User* dapat menggunakannya dengan mudah.

4.4.1 Pengujian Validasi Ahli Media

Adapun hasil dari pengujian yang telah dilakukan tersebut adalah sebagai berikut.

a. Pengujian Halaman *Admin*

Tabel 4. 1. Validasi *Login Admin*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Admin</i> mengakses halaman <i>login</i>	Halaman <i>login</i> muncul dengan <i>input</i> email & <i>password</i>	Valid	[✓] Diterima [] Ditolak
2	<i>Admin</i> memasukkan email & <i>password</i> benar	Sistem <i>redirect</i> ke <i>dashboard admin</i>	Valid	[✓] Diterima [] Ditolak
3	<i>Admin</i> memasukkan data salah	Sistem menampilkan pesan <i>error</i> "Email/ <i>Password</i> salah"	Valid	[✓] Diterima [] Ditolak

Tabel 4. 2. Validasi *Dashboard Admin*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Admin</i> berhasil login	<i>Dashboard</i> ditampilkan	Valid	[✓] Diterima [] Ditolak
2	<i>Dashboard</i> menampilkan ringkasan data sistem	Ringkasan data sistem ditampilkan	Valid	[✓] Diterima [] Ditolak

Tabel 4. 3. Validasi Data *User*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Admin</i> klik tombol edit <i>user</i>	<i>Form</i> edit muncul dengan data <i>user</i> terisi	Valid	[✓] Diterima [] Ditolak
2	<i>Admin</i> menyimpan perubahan data <i>user</i>	Data <i>user</i> terupdate dan sistem tampilkan notifikasi	Valid	[✓] Diterima [] Ditolak
3	<i>Admin</i> klik tombol hapus <i>user</i>	Data <i>user</i> terhapus dan tidak muncul lagi di tabel	Valid	[✓] Diterima [] Ditolak

4	<i>Admin</i> menggunakan <i>filter</i> pencarian	Sistem menampilkan data sesuai kata kunci	Valid	[✓] Diterima [] Ditolak
---	--	---	-------	-----------------------------

Tabel 4. 4. Validasi Halaman Data Jenis Surat

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Admin</i> membuka menu Jenis Surat	Sistem menampilkan tabel jenis surat	Valid	[✓] Diterima [] Ditolak
2	<i>Admin</i> menambahkan data jenis surat	Data jenis surat terisi	Valid	[✓] Diterima [] Ditolak
3	<i>Admin</i> mengedit data jenis surat	Data jenis surat terupdate dan sistem tampilkan notifikasi	Valid	[✓] Diterima [] Ditolak
4	<i>Admin</i> menghapus data jenis surat	Data jenis surat terhapus dan tidak muncul lagi di tabel	Valid	[✓] Diterima [] Ditolak

Tabel 4. 5. Validasi Tambah, Edit, Hapus Jenis Surat *Admin*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Admin</i> klik tombol tambah jenis surat	<i>Form</i> tambah muncul dan bisa diisi	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	<i>Admin</i> simpan jenis surat baru	Data baru muncul di tabel jenis surat	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	<i>Admin</i> klik tombol edit jenis surat	<i>Form</i> edit muncul dengan data yang dipilih	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	<i>Admin</i> menyimpan perubahan jenis surat	Data surat diperbarui di tabel	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5	<i>Admin</i> klik tombol hapus jenis surat	Data dihapus dan hilang dari tabel	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4. 6. Validasi Halaman Arsip Surat *Admin*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Admin</i> melihat daftar arsip surat	Sistem menampilkan daftar arsip surat yang tersimpan	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	<i>Admin</i> menambahkan arsip surat secara manual	Arsip surat berhasil disimpan dan muncul di daftar	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	<i>Admin</i> mengedit arsip surat manual	Data arsip diperbarui dan perubahan ditampilkan	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	<i>Admin</i> menghapus arsip surat	Arsip terhapus dari daftar dan tidak bisa diakses lagi	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5	<i>Admin</i> melakukan pencarian berdasarkan nama pengaju	Menampilkan arsip yang sesuai dengan nama yang diketik	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
6	<i>Admin</i> melakukan pencarian berdasarkan jenis surat	Daftar arsip menampilkan arsip dengan jenis surat yang dipilih	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

b. Pengujian Halaman *Operator*

Tabel 4. 7. Validasi Halaman *Login Operator*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Operator</i> mengakses halaman <i>login</i>	Halaman <i>login</i> muncul dengan <i>input</i> email & <i>password</i>	Valid	[✓] Diterima [] Ditolak
2	<i>Operator</i> memasukkan email & <i>password</i> benar	Sistem <i>redirect</i> ke <i>dashboard operator</i>	Valid	[✓] Diterima [] Ditolak
3	<i>Operator</i> memasukkan data salah	Pesan <i>error</i> ditampilkan “Email/ <i>Password</i> salah”	Valid	[✓] Diterima [] Ditolak

Tabel 4. 8. Validasi Halaman *Dashboard Operator*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Operator login</i> berhasil	Halaman <i>dashboard</i> tampil dengan menu yang tersedia	Valid	[✓] Diterima [] Ditolak
2	Tampilan <i>dashboard</i> memuat ringkasan data sistem	Informasi data sistem ditampilkan	Valid	[✓] Diterima [] Ditolak

Tabel 4. 9. Validasi Halaman Data Pengajuan Surat *Operator*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Operator</i> membuka menu Data Pengajuan	Tabel daftar pengajuan surat warga ditampilkan	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	<i>Operator</i> mencari pengajuan tertentu	Tabel menampilkan hasil <i>filter</i> berdasarkan status yang dipilih	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4. 10. Validasi Ubah Status Pengajuan Surat *Operator*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Operator</i> klik tombol “Proses” atau “Tolak”	Status pengajuan berubah di database dan di tabel	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Sistem menampilkan pesan notifikasi berhasil ubah status	Notifikasi seperti “Status berhasil diubah” muncul	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4. 11. Validasi Dokumen Pengajuan *Operator*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Operator</i> klik tombol “Tambah Dokumen”	<i>Form upload</i> dokumen muncul	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	<i>Operator</i> memilih dokumen dan klik <i>upload</i>	Sistem validasi dokumen (format, ukuran, dll)	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Dokumen valid	Dokumen tersimpan ke database dan tabel	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Dokumen tidak valid (misal ukuran besar)	Sistem menampilkan pesan <i>error</i> validasi	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5	Setelah ubah status atau <i>upload</i> dokumen	Sistem menampilkan pesan konfirmasi keberhasilan	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
6	Gagal <i>upload</i> atau <i>error</i> lainnya	Sistem menampilkan pesan <i>error</i> yang informatif	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4. 12. Validasi Halaman Arsip Surat *Operator*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Operator</i> melihat daftar arsip surat	Sistem menampilkan daftar arsip surat yang tersimpan	Valid	[✓] Diterima [] Ditolak
2	<i>Operator</i> menambahkan arsip surat secara manual	Arsip surat berhasil disimpan dan muncul di daftar	Valid	[✓] Diterima [] Ditolak
3	<i>Operator</i> mengedit arsip surat manual	Data arsip diperbarui dan perubahan ditampilkan	Valid	[✓] Diterima [] Ditolak
4	<i>Operator</i> menghapus arsip surat	Arsip terhapus dari daftar dan tidak bisa diakses lagi	Valid	[✓] Diterima [] Ditolak
5	<i>Operator</i> melakukan pencarian berdasarkan nama pengaju	Menampilkan arsip yang sesuai dengan nama yang diketik	Valid	[✓] Diterima [] Ditolak
6	<i>Operator</i> melakukan pencarian berdasarkan jenis surat	Daftar arsip menampilkan arsip dengan jenis surat yang dipilih	Valid	[✓] Diterima [] Ditolak

c. Pengujian Halaman *User*

Tabel 4. 13. Validasi Halaman *Login User*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>User</i> mengakses halaman <i>login</i>	Halaman <i>login</i> ditampilkan dengan <i>form</i> <i>username</i> dan <i>password</i>	Valid	[✓] Diterima [] Ditolak
2	<i>User</i> mengisi <i>form</i> dengan data benar & terverifikasi	<i>User</i> diarahkan ke halaman utama	Valid	[✓] Diterima [] Ditolak
3	<i>User</i> mengisi <i>form</i> dengan <i>username/password</i> salah	Pesan kesalahan “ <i>Username</i> atau <i>Password</i> salah” ditampilkan	Valid	[✓] Diterima [] Ditolak
4	<i>User</i> belum melakukan verifikasi akun	Pesan “Akun belum diverifikasi” muncul dan <i>login</i> ditolak	Valid	[✓] Diterima [] Ditolak
5	<i>User</i> melakukan <i>login</i> setelah akun diverifikasi	<i>User</i> dapat <i>login</i> dan diarahkan ke halaman utama	Valid	[✓] Diterima [] Ditolak
6	<i>Input field login</i> dikosongkan lalu <i>submit</i>	Muncul pesan validasi bahwa <i>field</i> tidak boleh kosong	Valid	[✓] Diterima [] Ditolak

Tabel 4. 14. Validasi Halaman Formulir Pengajuan Surat

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	User memilih menu "Ajukan Surat"	Halaman formulir pengajuan surat tampil	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	User mengisi semua <i>field</i> dan klik <i>Submit</i>	Jika valid, data tersimpan ke database, muncul pesan sukses	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	User mengosongkan <i>field</i> wajib	Validasi menolak, muncul pesan "Harap lengkapi <i>form</i> "	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Format <i>input</i> tidak sesuai (contoh: huruf di <i>field</i> NIK)	Validasi gagal, tampilkan pesan kesalahan format	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4. 15. Validasi Status Pengajuan Surat

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	User memilih menu "Status Pengajuan"	Halaman status tampil dan menampilkan semua pengajuan surat	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

2	Terdapat status “Diproses / Ditolak / Disetujui”	Status sesuai dengan yang diubah oleh <i>operator/admin</i>	Valid	[✓] Diterima [] Ditolak
---	--	---	-------	-----------------------------

Tabel 4. 16. Validasi Halaman Riwayat Pengajuan Surat

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>User</i> memilih menu “Riwayat”	Sistem menampilkan daftar riwayat pengajuan surat yang selesai atau ditolak	Valid	[✓] Diterima [] Ditolak
2	<i>User</i> melihat detail surat selesai	Sistem menampilkan detail surat (tanggal pengajuan, tanggal selesai, jenis surat, <i>file</i> pdf, dan detail)	Valid	[✓] Diterima [] Ditolak
3	<i>User</i> melihat detail surat ditolak	Sistem menampilkan detail surat (tanggal pengajuan, tanggal ditolak, jenis surat, alasan penolakan, dan detail)	Valid	[✓] Diterima [] Ditolak

Tabel 4. 17. Validasi Halaman Profil *User*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>User</i> klik menu “Profil”	Halaman profil muncul berisi data <i>user</i>	Valid	[✓] Diterima [] Ditolak

2	Data tampil sesuai database	Data <i>user</i> sesuai dengan informasi yang tersimpan	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
---	-----------------------------	---	-------	--

Tabel 4. 18. Validasi Halaman Edit Profil *User*

No	Aktivitas yang diuji	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>User</i> klik tombol edit profil	<i>Form</i> edit profil tampil dengan data yang bisa diubah	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	<i>User</i> mengubah data dan klik simpan	Validasi data dan jika berhasil, simpan ke database	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Data kosong atau salah format (misal nama angka)	Validasi gagal dan tampil pesan <i>error</i> format/kosong	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Setelah sukses edit, muncul pesan sukses	Pesan “Profil berhasil diperbarui” ditampilkan	Valid	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

4.5 Hasil Kuesioner

Untuk mengevaluasi sejauh mana sistem yang telah dikembangkan dapat diterima dan digunakan dengan baik oleh pengguna, dilakukan pengujian melalui metode *User Acceptance Testing* (UAT). Pengujian ini dilakukan dengan membagikan kuesioner kepada beberapa pengguna yang telah mencoba langsung fitur-fitur sistem.

Kuesioner terdiri dari tujuh pertanyaan yang dirancang untuk mengukur kemudahan penggunaan, kejelasan alur, serta kepuasan pengguna terhadap sistem. Setiap pertanyaan diberi skor antara 1 hingga 5, dengan skor 5 menunjukkan tingkat kepuasan tertinggi. Hasil pengisian kuesioner dari 13 responden disajikan dalam tabel berikut:

Tabel 4. 19. Hasil Kuesioner *User*

No	Nama	Pertanyaan							Total
		1	2	3	4	5	6	7	
1	Zainudin	5	4	5	5	4	5	5	33
2	Andriyanto	5	5	4	5	4	3	5	31
3	Abdi Pratama	5	3	4	4	5	5	4	30
4	Parto Sulistyو	5	4	3	5	5	5	4	31
5	Barto Novendi	5	5	4	4	5	5	4	32
6	Ikhsan	5	5	5	5	5	5	5	35
7	Agus Samsudin	4	5	4	4	5	5	4	31
8	Usman	5	5	5	5	5	5	5	35
9	Parjo Mawarman	5	4	5	5	4	5	4	32
10	Nurdin	5	5	5	5	5	5	5	35
11	Astuti	5	5	5	5	5	4	5	34
12	Yuliana	4	4	4	4	4	4	4	28
13	Ratih Devi	5	4	3	4	3	5	5	29
Total									416

Berdasarkan hasil di atas, total skor keseluruhan yang diperoleh adalah 416 dari maksimal skor 455 (jika semua responden memberikan skor 5 di setiap pertanyaan). Hal ini menunjukkan bahwa sebagian besar pengguna merasa puas terhadap kinerja sistem, terutama dalam hal kemudahan penggunaan, kejelasan alur, dan efektivitas dalam mendukung proses administrasi pelayanan surat di desa secara digital. Skor tinggi yang diperoleh mencerminkan bahwa sistem telah memenuhi ekspektasi pengguna dan layak untuk digunakan secara luas dalam operasional administrasi pelayanan surat-menyurat di desa.

4.6 Hasil Perhitungan *User Acceptance Testing* (UAT)

Berikut merupakan hasil perhitungan dari pengujian yang sudah dilakukan:

$$\text{Total} = 416$$

$$\text{Skor Maksimal} = 5 \times 7 \times 13 = 455$$

$$\begin{aligned} \text{Hasil Pengujian} &= \left(\frac{416}{455} \right) \times 100\% \\ &= 91,42\% \end{aligned}$$

Hasil perhitungan menunjukkan bahwa website administrasi pelayanan surat-menyurat di Desa Rias merupakan solusi yang ideal dalam meningkatkan kualitas pelayanan administrasi secara digital. Sistem ini dinilai telah memenuhi kebutuhan pengguna dari segi kemudahan akses, kecepatan layanan, serta efisiensi dalam pengelolaan data administrasi. Berdasarkan hasil pengujian yang dilakukan melalui metode *User Acceptance Testing* (UAT), diperoleh nilai akhir dari pengguna sebesar 91,42%. Nilai ini menunjukkan bahwa sistem berada dalam kategori “Sangat Layak” untuk digunakan dalam mendukung proses administrasi yang lebih transparan, efisien, dan terstruktur.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, pengembangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa:

1. Sistem administrasi pelayanan surat berbasis web yang dikembangkan untuk Desa Rias berhasil menjawab permasalahan utama dalam pengelolaan administrasi yang sebelumnya masih dilakukan secara manual.
2. Sistem ini menyediakan fitur utama seperti registrasi pengguna, pengajuan surat secara daring, verifikasi oleh *operator*, pengelolaan arsip surat, serta manajemen pengguna dan jenis surat yang dapat diakses oleh *admin*.
3. Pengujian menggunakan metode *Black Box Testing* menunjukkan bahwa seluruh fitur berjalan sesuai dengan fungsinya tanpa adanya kesalahan teknis.
4. Hasil *User Acceptance Testing* (UAT) memperoleh nilai sebesar 91,42%, yang berada pada kategori “Sangat Layak”, menandakan bahwa sistem ini telah diterima dengan baik oleh pengguna dan sesuai dengan kebutuhan administrasi desa.
5. Dengan diterapkannya sistem ini, proses pelayanan administrasi surat-menyerurat di Desa Rias menjadi lebih efisien, transparan, mudah diakses, serta mendukung pelayanan publik tanpa harus datang langsung ke kantor desa.

5.2 Saran

Agar *website* administrasi pelayanan surat ini dapat terus berkembang dan digunakan secara berkelanjutan, terdapat beberapa saran sebagai berikut:

1. Fitur notifikasi otomatis melalui WhatsApp dapat ditambahkan guna memberikan informasi lebih cepat kepada pengguna mengenai status pengajuan surat.
2. Terintegrasi dengan tanda tangan elektronik (*digital signature*) untuk meningkatkan keamanan dan keabsahan dokumen.
3. Perlu dilakukan pemeliharaan sistem secara berkala agar performa dan keamanan aplikasi tetap terjaga.



DAFTAR PUSTAKA

- Safrudin, L. I., Sutopo, R. A., Rabani, H. F., Aji, S. N., Indriyanto, Z. F., & Widiati, I. S. (2024). Implementasi sistem informasi administrasi Desa Penggung berbasis website. *Seminar Nasional Hasil Penelitian & Pengabdian Masyarakat Bidang Ilmu Komputer*.
- Fitriasari, N. S., Ariawan, I., Arifin, W. A., Salim, H., Fawaz, A., Apriansyah, M. R., & Lipalda, F. (2023). Pengembangan pelayanan administrasi aplikasi website Desa Lontar. *Jurnal Mandala Pengabdian Masyarakat*, 290–296. <https://www.jurnal-pharmaconmw.com/jmpm/index.php/jmpm/article/view/140>
- Syaebani, A., Tyasmala, D. V., Maulani, R., Utami, E. D., & Wahyuni, S. N. (2021). Pengembangan sistem informasi pelayanan surat menyurat (SIRA) berbasis website dengan menggunakan framework CodeIgniter. *Journal of Information System Management (JOISM)*, 3(2), 32–38. <https://doi.org/10.24076/joism.2021v3i2.446>
- Umami, I., Adam, M. T., & Winarti. (2022). Perancangan sistem informasi pengelolaan surat menyurat berbasis web Desa Sumberkarang. *ULIL ALBAB: Jurnal Ilmiah Multidisiplin*, 2880–2885. <https://ulilalbabinstitute.id/index.php/JIM/article/view/663>
- Romadhon, A. L., & Maryam. (2023). Rancang bangun sistem informasi layanan administrasi desa berbasis web di Desa Dukuh. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 8(2), 514–524. <https://doi.org/10.29100/jipi.v8i2.3553>
- Hidayat, A., & Fresh, Y. S. Z. (2022). Peran administrasi pembangunan desa. *Jurnal Inovasi Penelitian*, 3(7), 7091–7100. <https://doi.org/10.47492/jip.v3i7.2215>
- Huda, N. (2022). *Visual Studio Code: Pengertian, fitur, keunggulan dan jenisnya*. Dewaweb. <https://www.dewaweb.com/blog/mengenal-visual-studio-code/>
- Akbar, A. B. (2021). *Pengertian PHP, fungsinya, dan kelebihanannya*. Peta Network. <https://www.peta-network.com/pengertian-php/>
- Prasatya. (2024). *Apa itu Tailwind? Sejarah, tools, contoh syntax*. CODEPOLITAN. <https://www.codepolitan.com/blog/apa-itu-tailwind-sejarah-tools-contoh-syntax/>
- RevoU. (n.d.). *Apa itu Tailwind? Arti, fungsi, contoh, FAQs 2025*. RevoU. <https://www.revou.co/kosakata/tailwind>
- Puguh, A. (2023). *Laragon: Pengertian, fungsi, download, dan cara instalasinya*. Rumahweb. <https://blog.rumahweb.com/laragon-adalah/>
- Fauzi, H. (2024). *Apa itu MySQL? Definisi, manfaat, dan cara kerjanya*. Nevacloud. <https://nevacloud.com/blog/mysql/>
- Sutisna, N. (2024). *Kenalan dengan Laravel: Framework PHP yang keren dan serbaguna*. Dicoding. <https://www.dicoding.com/blog/kenalan-dengan-laravel-framework-php-yang-keren-dan-serbaguna/>
- Biznetgio. (2025). *Apa itu website? Pengertian, fungsi dan jenis-jenisnya*. Biznetgio. <https://www.biznetgio.com/news/apa-itu-website>

Musyaffa, I. (2021). *Metode pengembangan RAD (Rapid Application Development)*. Agus Hermanto. <https://agus-hermanto.com/blog/detail/metode-pengembangan-rad-rapid-application-development>

Huda, A. M. (2025). *Apa itu desain sistem dan mengapa hal itu diperlukan?* Diklatkerja. <https://www.diklatkerja.com/blog/apa-itu-desain-sistem-dan-mengapa-hal-itu-diperlukan>



LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Rizqi Pratama
Tempat & tanggal lahir : Sungailiat, 11 Desember 2003
Alamat rumah : Jl. Bukit Betung II
Hp : 085269903837
Email : rizqi1112@gmail.com
Jenis kelamin : Laki - Laki
Agama : Islam



2. Riwayat Pendidikan

- a. 2010-2016 : SD Negeri 10 Sungailiat
- b. 2016-2019 : SMP Negeri 2 Sungailiat
- c. 2019-2022 : SMA Negeri 1 Sungailiat

Sungailiat, 21 Juli 2025

Rizqi Pratama

Lampiran 2 : Lembar Validasi Ahli

PENGEMBANGAN SISTEM BERBASIS WEB UNTUK DIGITALISASI ADMINISTRASI DESA RIAS (Penguji Blackbox)				
Nama Lengkap : <u>Biadela Annandora Wasesa, S.Kom., M.Kom.</u>				
Tanggal : <u>03 Juli 2025</u>				
4.1 Tabel Pengujian Halaman Login Admin				
No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Admin mengakses halaman login	Halaman login muncul dengan input email & password	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Admin memasukkan email & password benar	Sistem redirect ke dashboard admin	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Admin memasukkan data salah	Sistem menampilkan pesan error "Email/Password salah"	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4.2 Tabel Pengujian Halaman Dashboard				
No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Admin berhasil login	Dashboard ditampilkan	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

2	Dashboard menampilkan ringkasan data sistem	Ringkasan data sistem ditampilkan	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
---	---	-----------------------------------	---------	--

4.3 Tabel Pengujian Data User

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Admin klik tombol edit user	Form edit muncul dengan data user terisi	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Admin menyimpan perubahan data user	Data user terupdate dan sistem tampilkan notifikasi	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Admin klik tombol hapus user	Data user terhapus dan tidak muncul lagi di tabel	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Admin menggunakan filter pencarian	Sistem menampilkan data sesuai kata kunci	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

4.4 Tabel Pengujian Halaman Data Jenis Surat

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Admin membuka menu Jenis Surat	Sistem menampilkan tabel jenis surat	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Admin menambahkan data jenis surat	Data jenis surat terisi	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Admin mengedit data jenis surat	Data jenis surat terupdate dan sistem tampilkan notifikasi	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Admin menghapus data jenis surat	Data jenis surat terhapus dan tidak muncul lagi di tabel	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

4.5 Tabel Pengujian Tambah, Edit, Hapus Jenis Surat Admin

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Admin klik tombol tambah jenis surat	Form tambah muncul dan bisa diisi	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

2	Admin simpan jenis surat baru	Data baru muncul di tabel jenis surat	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Admin klik tombol edit jenis surat	Form edit muncul dengan data yang dipilih	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
4	Admin menyimpan perubahan jenis surat	Data surat diperbarui di tabel	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
5	Admin klik tombol hapus jenis surat	Data dihapus dan hilang dari tabel	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4.6 Pengujian Halaman Arsip Surat Admin

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Admin melihat daftar arsip surat	Sistem menampilkan daftar arsip surat yang tersimpan	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Admin menambahkan arsip surat secara manual	Arsip surat berhasil disimpan dan muncul di daftar	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

3	Admin mengedit arsip surat manual	Data arsip diperbarui dan perubahan ditampilkan	✓ [] Ditolak	✓ [] Ditolak
4	Admin menghapus arsip surat	Arsip terhapus dari daftar dan tidak bisa diakses lagi	✓ [] Ditolak	✓ [] Ditolak
5	Admin melakukan pencarian berdasarkan nama pengaju	Menampilkan arsip yang sesuai dengan nama yang diketik	✓ [] Ditolak	✓ [] Ditolak
6	Admin melakukan pencarian berdasarkan jenis surat	Daftar arsip menampilkan arsip dengan jenis surat yang dipilih	✓ [] Ditolak	✓ [] Ditolak

4.7 Tabel Pengujian Halaman Login Operator

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Operator mengakses halaman login	Halaman login muncul dengan input email & password	✓ [] Ditolak	✓ [] Ditolak
2	Operator memasukkan email & password benar	Sistem redirect ke dashboard operator	✓ [] Ditolak	✓ [] Ditolak
3	Operator memasukkan data salah	Pesan error ditampilkan "Email/Password salah"	✓ [] Ditolak	✓ [] Ditolak

4.8 Tabel Pengujian Halaman Dashboard Operator

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Operator login berhasil	Halaman dashboard tampil dengan menu yang tersedia	✓ [] Ditolak	✓ [] Ditolak
2	Tampilan dashboard memuat ringkasan data sistem	Informasi data sistem ditampilkan	✓ [] Ditolak	✓ [] Ditolak

4.9 Tabel Pengujian Halaman Data Pengajuan Surat Operator

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Operator membuka menu Data Pengajuan	Tabel daftar pengajuan surat warga ditampilkan	✓ [] Ditolak	✓ [] Ditolak
2	Operator mencari pengajuan tertentu	Tabel menampilkan hasil filter berdasarkan status yang dipilih	✓ [] Ditolak	✓ [] Ditolak

4.10 Tabel Pengujian Ubah Status Pengajuan Surat Operator

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Operator klik tombol "Proses" atau "Tolak"	Status pengajuan berubah di database dan di tabel	✓ [] Ditolak	✓ [] Ditolak
2	Sistem menampilkan pesan notifikasi berhasil ubah status	Notifikasi seperti "Status berhasil diubah" muncul	✓ [] Ditolak	✓ [] Ditolak

4.11 Tabel Pengujian Dokumen Pengajuan Operator

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Operator klik tombol "Tambah Dokumen"	Form upload dokumen muncul	✓ [] Ditolak	✓ [] Ditolak
2	Operator memilih dokumen dan klik upload	Sistem validasi dokumen (format, ukuran, dll)	✓ [] Ditolak	✓ [] Ditolak
3	Dokumen valid	Dokumen tersimpan ke database dan tabel	✓ [] Ditolak	✓ [] Ditolak
4	Dokumen tidak valid (misal ukuran besar)	Sistem menampilkan pesan error validasi	✓ [] Ditolak	✓ [] Ditolak
5	Setelah ubah status atau upload dokumen	Sistem menampilkan pesan konfirmasi keberhasilan	✓ [] Ditolak	✓ [] Ditolak
6	Gagal upload atau error lainnya	Sistem menampilkan pesan error yang informatif	✓ [] Ditolak	✓ [] Ditolak

Tabel 4.12 Pengujian Halaman Arsip Surat Operator

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Operator melihat daftar arsip surat	Sistem menampilkan daftar arsip surat yang tersimpan	Selesai	[✓] Diterima [] Ditolak
2	Operator menambahkan arsip surat secara manual	Arsip surat berhasil disimpan dan muncul di daftar	Selesai	[✓] Diterima [] Ditolak
3	Operator mengedit arsip surat manual	Data arsip diperbarui dan perubahan ditampilkan	Selesai	[✓] Diterima [] Ditolak
4	Operator menghapus arsip surat	Arsip terhapus dari daftar dan tidak bisa diakses lagi	Selesai	[✓] Diterima [] Ditolak
5	Operator melakukan pencarian berdasarkan nama pengaju	Menampilkan arsip yang sesuai dengan nama yang diketik	Selesai	[✓] Diterima [] Ditolak

6	Operator melakukan pencarian berdasarkan jenis surat	Daftar arsip menampilkan arsip dengan jenis surat yang dipilih	Selesai	[✓] Diterima [] Ditolak
---	--	--	---------	-----------------------------

Tabel 4.13 Pengujian Halaman Login User

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	User mengakses halaman login	Halaman login ditampilkan dengan form username dan password	Selesai	[✓] Diterima [] Ditolak
2	User mengisi form dengan data benar & terverifikasi	User diarahkan ke halaman utama	Selesai	[✓] Diterima [] Ditolak
3	User mengisi form dengan username/password salah	Pesan kesalahan "Username atau Password salah" ditampilkan	Selesai	[✓] Diterima [] Ditolak
4	User belum melakukan verifikasi akun	Pesan "Akun belum diverifikasi" muncul dan login ditolak	Selesai	[✓] Diterima [] Ditolak

5	User melakukan login setelah akun diverifikasi	User dapat login dan diarahkan ke halaman utama	Selesai	[✓] Diterima [] Ditolak
6	Input field login dikosongkan lalu submit	Muncul pesan validasi bahwa field tidak boleh kosong	Selesai	[✓] Diterima [] Ditolak

Tabel 4.14 Pengujian Halaman Formulir Pengajuan Surat

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	User memilih menu "Ajukan Surat"	Halaman formulir pengajuan surat tampil	Selesai	[✓] Diterima [] Ditolak
2	User mengisi semua field dan klik Submit	Jika valid, data tersimpan ke database, muncul pesan sukses	Selesai	[✓] Diterima [] Ditolak
3	User mengosongkan field wajib	Validasi menolak, muncul pesan "Harap lengkapi form"	Selesai	[✓] Diterima [] Ditolak
4	Format input tidak sesuai (contoh: huruf di field NIK)	Validasi gagal, tampilan pesan kesalahan format	Selesai	[✓] Diterima [] Ditolak

Tabel 4.15 Pengujian Status Pengajuan Surat

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	User memilih menu "Status Pengajuan"	Halaman status tampil dan menampilkan semua pengajuan surat	Selesai	[✓] Diterima [] Ditolak
2	Terdapat status "Diproses / Ditolak / Disetujui"	Status sesuai dengan yang diubah oleh operator/admin	Selesai	[✓] Diterima [] Ditolak

Tabel 4.16 Pengujian Riwayat Pengajuan Surat

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	User memilih menu "Riwayat"	Sistem menampilkan daftar riwayat pengajuan surat yang selesai atau ditolak	Selesai	[✓] Diterima [] Ditolak
2	User melihat detail surat selesai	Sistem menampilkan detail surat (tanggal pengajuan, tanggal selesai, jenis surat, file pdf, dan detail)	Selesai	[✓] Diterima [] Ditolak
3	User melihat detail surat ditolak	Sistem menampilkan detail surat (tanggal pengajuan, tanggal ditolak, jenis surat, alasan penolakan, dan detail)	Selesai	[✓] Diterima [] Ditolak

Tabel 4.17 Pengujian Halaman Profil User

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	User klik menu "Profil"	Halaman profil muncul berisi data user	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Data tampil sesuai database	Data user sesuai dengan informasi yang tersimpan	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4.18 Pengujian Edit Profil User

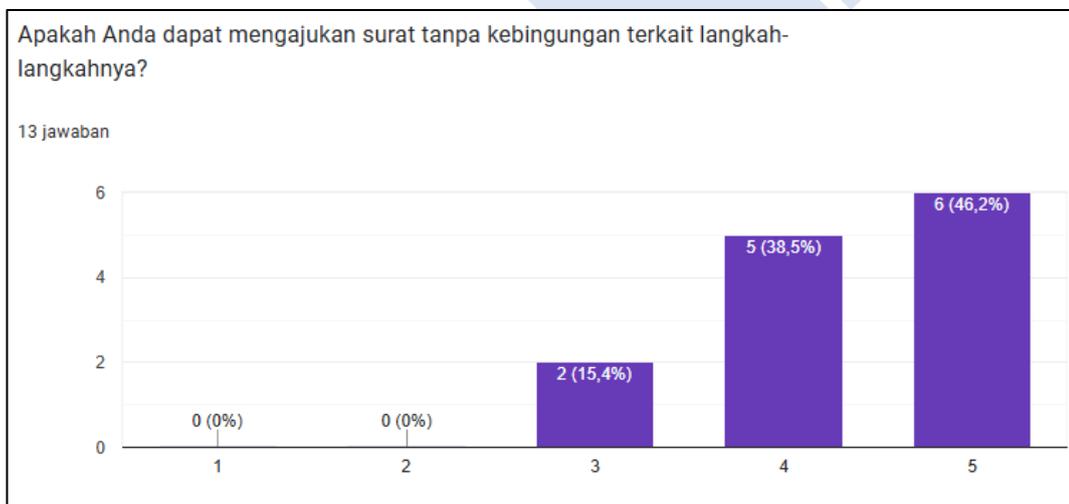
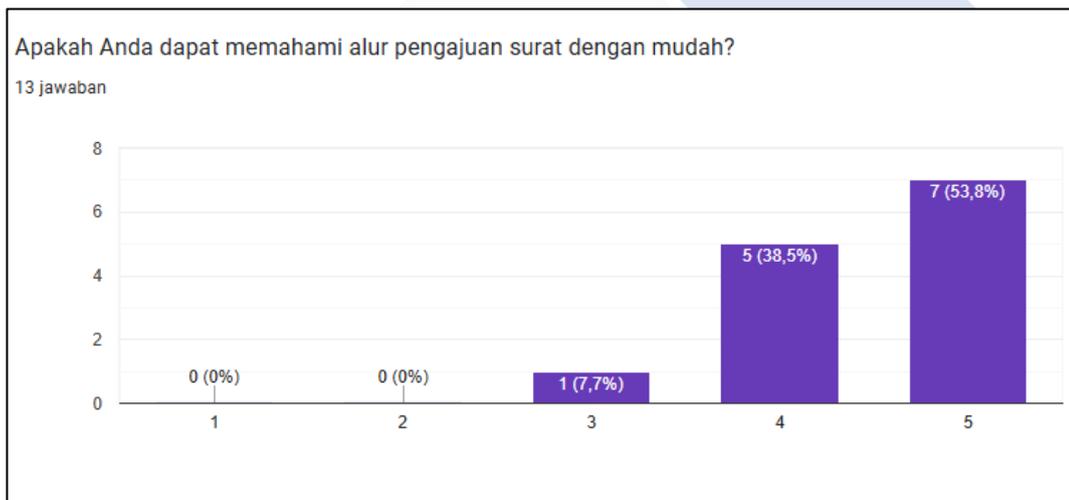
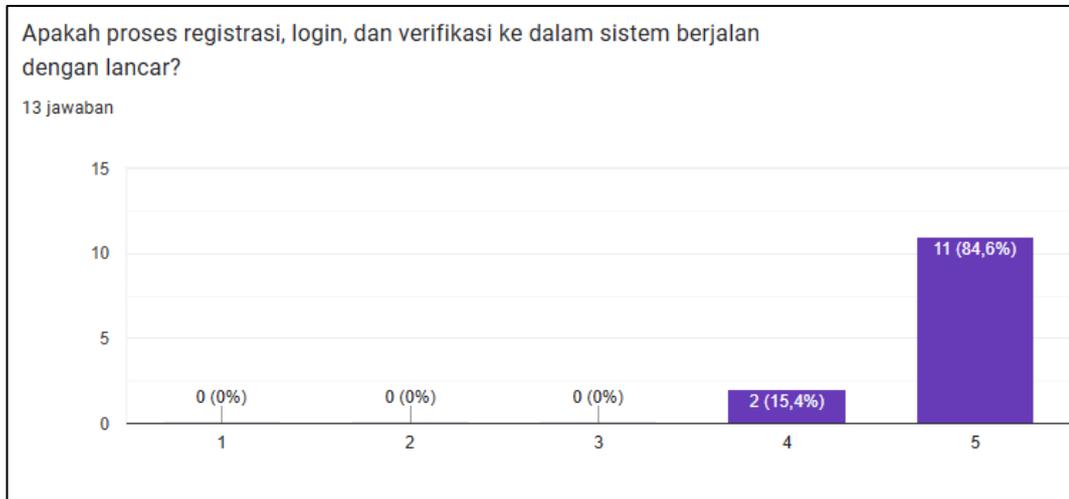
No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	User klik tombol edit profil	Form edit profil tampil dengan data yang bisa diubah	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	User mengubah data dan klik simpan	Validasi data dan jika berhasil, simpan ke database	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Data kosong atau salah format (misal nama angka)	Validasi gagal dan tampil pesan error format/kosong	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

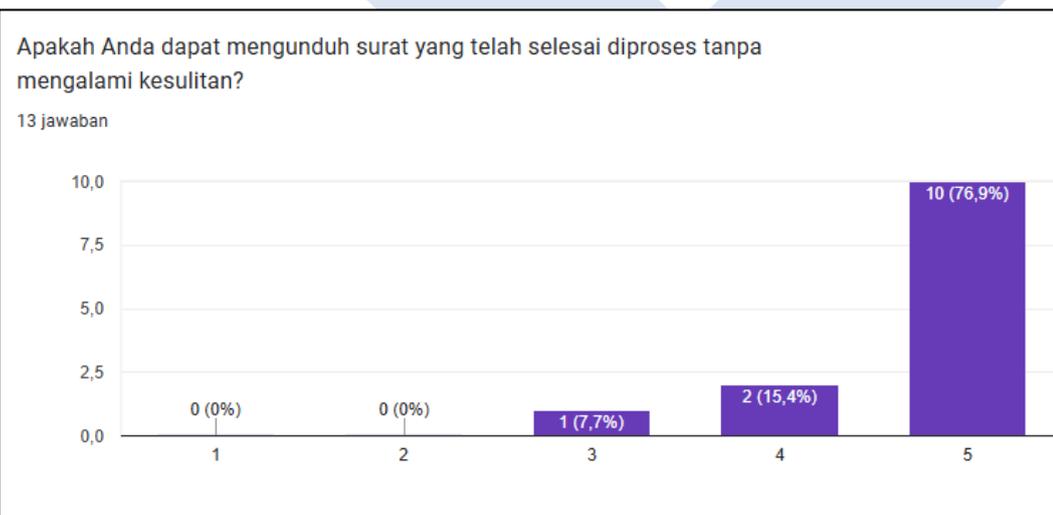
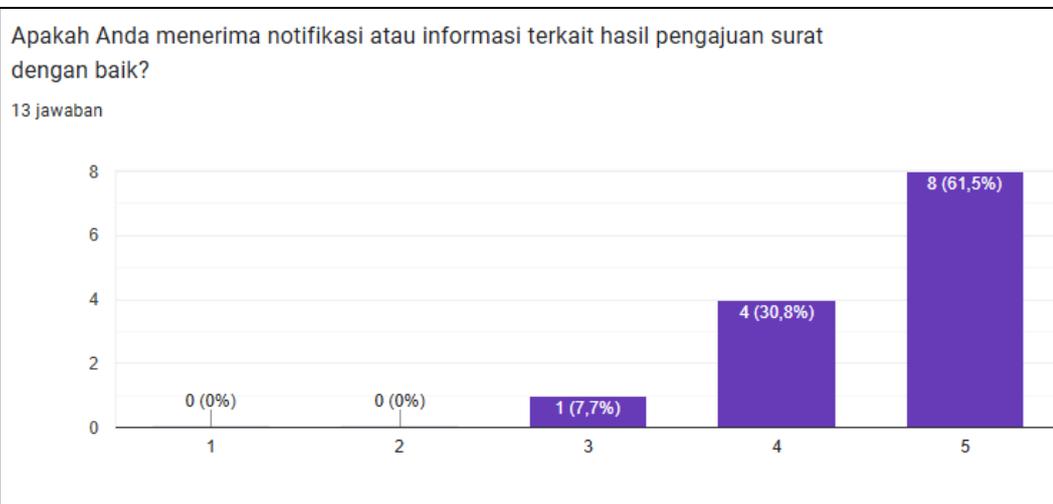
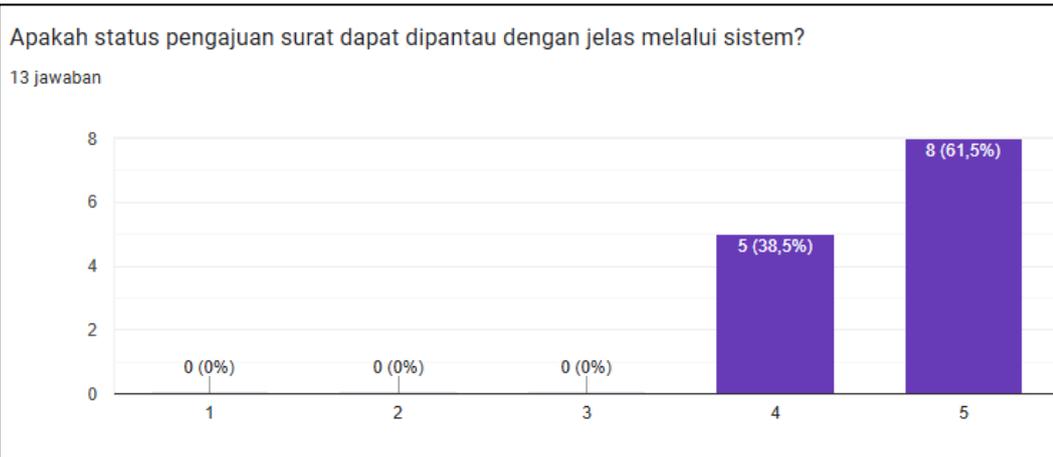
4	Setelah sukses edit, muncul pesan sukses	Pesan "Profil berhasil diperbarui" ditampilkan	Selesai	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
---	--	--	---------	--

03 Juli 2025 Validator,

B
Benedicta Ananda Wisoso,
S.Kom., M.Kom.

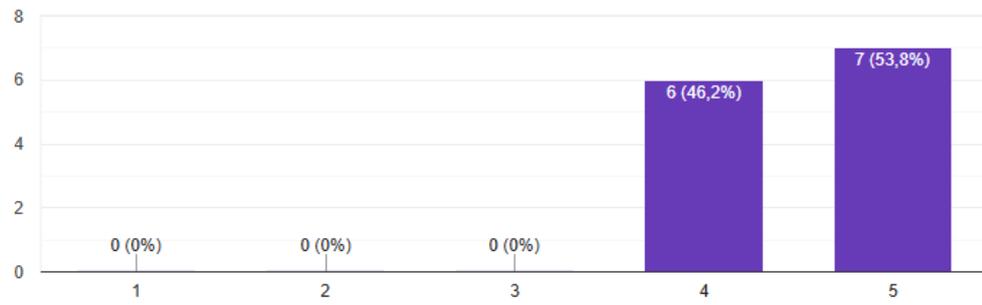
Lampiran 3 : Hasil Kuisisioner UAT





Apakah sistem ini membantu memudahkan Anda dalam mengurus keperluan administrasi desa?

13 jawaban



Lampiran 4 : Kuisiener Penilaian Admin

Kuisiener Penilaian Admin

Nama Lengkap: *Iman Syaifulin*

No	Pertanyaan	Skala Penilaian				
		1	2	3	4	5
1	Apakah Anda dapat login ke dalam sistem dengan mudah dan cepat?					✓
2	Apakah antarmuka sistem admin mudah dipahami dan digunakan?				✓	
3	Apakah fitur pengelolaan data pengguna berjalan dengan baik dan sesuai kebutuhan?					✓
4	Apakah Anda dapat menambahkan, mengedit, dan menghapus jenis surat tanpa kendala?					✓
5	Apakah sistem memudahkan Anda dalam mengelola arsip surat secara digital?					✓
6	Apakah data pengguna dan jenis surat ditampilkan secara jelas dan akurat di dalam sistem?					✓
7	Apakah Anda merasa puas menggunakan sistem ini untuk membantu pekerjaan administrative desa?					✓

Keterangan:

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Lampiran 5 : Kuisiener Penilaian Operator

Kuisiener Penilaian Operator

Nama Lengkap: Della Puspitasari

No	Pertanyaan	Skala Penilaian				
		1	2	3	4	5
1	Apakah Anda dapat login ke dalam sistem tanpa hambatan?				✓	
2	Apakah proses validasi pengajuan surat oleh warga mudah dilakukan melalui sistem ini?				✓	
3	Apakah Anda dapat dengan mudah mengunggah surat yang telah diproses ke dalam sistem?			✓		
4	Apakah sistem menampilkan status pengajuan secara jelas dan akurat?				✓	
5	Apakah fitur arsip surat berfungsi dengan baik dan mudah digunakan?				✓	
6	Apakah sistem membantu mempercepat alur kerja dalam menangani pengajuan surat dari warga?					✓
7	Apakah Anda puas dengan kinerja sistem dalam mendukung tugas anda sebagai operator desa?					✓

Keterangan:

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Lampiran 6 : Dokumentasi Penyerahan *Website*

