

**DIGITALISASI PELAYANAN ADMINISTRASI DAN
KETAHANAN EKONOMI MASYARAKAT DESA
DENGAN TEKNIK *CROWDSOURCING***



Disusun oleh

Dwi Hapsari 3.34.18.0.08

Muhammad Irhamul Umam 3.34.18.0.16

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI SEMARANG
2021**

**DIGITALISASI PELAYANAN ADMINISTRASI DAN
KETAHANAN EKONOMI MASYARAKAT DESA
DENGAN TEKNIK *CROWDSOURCING***



**Tugas akhir ini disusun untuk melengkapi sebagian persyaratan
menjadi Ahli Madya**

Disusun oleh

Dwi Hapsari 3.34.18.0.08

Muhammad Irhamul Umam 3.34.18.0.16

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI SEMARANG**

2021

PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini, saya:

Nama / NIM / Kelas : Dwi Hapsari / 3.34.18.0.08 / IK 3A

Nama / NIM / Kelas : Muhammad Irhamul Umam / 3.34.18.0.16 / IK 3A

Jurusan : Teknik Elektro

Program Studi : Teknik Informatika

Menyatakan dengan sesungguhnya bahwa Tugas Akhir dengan judul **DIGITALISASI PELAYANAN ADMINISTRASI DAN KETAHANAN EKONOMI MASYARAKAT DESA DENGAN TEKNIK CROWDSOURCING** yang dibuat untuk melengkapi sebagian persyaratan menjadi Ahli Madya pada Program Studi Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Semarang, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari Tugas Akhir yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar Ahli Madya di lingkungan Politeknik Negeri Semarang maupun di perguruan tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Semarang, 16 September 2021

Mahasiswa,

Mahasiswa,

Dwi Hapsari

NIM 3.34.18.0.08

Muhammad Irhamul Umam

NIM 3.34.18.0.16

HALAMAN PERSETUJUAN

Tugas akhir dengan judul **DIGITALISASI PELAYANAN ADMINISTRASI DAN KETAHANAN EKONOMI MASYARAKAT DESA DENGAN TEKNIK CROWDSOURCING** dibuat untuk melengkapi sebagian persyaratan menjadi Ahli Madya pada Program Studi Teknik Informatika, Jurusan Teknik Elektro Politeknik Negeri Semarang dan disetujui untuk diajukan dalam sidang ujian Tugas Akhir.

Semarang, 16 September 2021

Pembimbing I,



Mardiyono, S.Kom., M.Sc.
NIP. 197403112000121001

Pembimbing II,



Dr. Ir. Kurnianingsih, S.T.,M.T.
NIP. 197904262003122002

Mengetahui,
Ketua Program Studi Teknik Informatika



Idhawati Hestiningsih, S.Kom., M.Kom.
NIP. 197711192008012013

HALAMAN PENGESAHAN

Tugas akhir dengan judul **DIGITALISASI PELAYANAN ADMINISTRASI DAN KETAHANAN EKONOMI MASYARAKAT DESA DENGAN TEKNIK CROWDSOURCING** Telah dipertahankan dalam ujian wawancara dan diterima sebagai syarat untuk menjadi Ahli Madya pada Program Studi Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Semarang pada tanggal 16 September 2021.

TIM PENGUJI

Penguji I,

Tri Raharjo Yudantoro S.Kom., M.Kom.
NIP. 196810252000121001

Penguji II,

Liliek Triyono, S.T., M. Kom.
NIP. 198404202015041003

Penguji III,

Slamet Handoko, S.Kom., M.Kom
NIP. 197501302001121001

Ketua Penguji,

Mardiyono, S.Kom., M.Sc.
NIP. 197403112000121001

Sekretaris,

Muhammad Irwan Yanwari, S.Kom.,M.Eng.
NIP. 199001072019031020

Mengesahkan



Yusnan Badruzzaman, S.T.,M.Eng
NIP. 197210271999031002

ABSTRAK

Dwi Hapsari dan Muhammad Irhamul Umam. "Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik Crowdsourcing", Tugas Akhir DIII Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Semarang, di bawah bimbingan Mardiyono, S.Kom., M.Sc. dan Dr. Ir. Kurnianingsih, S.T, M.T., September 2021, 164..

Pandemi covid-19 mengharuskan masyarakat untuk menjaga jarak (*social distancing*) khususnya dalam mengurus administrasi di kantor desa. Hal ini juga mengganggu UMKM dalam mempromosikan dan menjual produknya yang berdampak pada penurunan omsetnya. Tujuan dari pembuatan tugas akhir ini adalah membangun Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing*. Metode yang digunakan dalam pembangunan sistem ini adalah metode *Scrum* yang memiliki *tahapan product backlog, sprint planning meeting, sprint backlog, sprint execution, finished work*. Pembangunan sistem ini menerapkan teknik *crowdsourcing* yang berfungsi untuk menarik minat penduduk terhadap produk UMKM ketika pelaku UMKM mempromosikan produknya pada aplikasi dan dihubungkan dengan aplikasi *marketplace*. Sedangkan *database* yang digunakan dalam sistem ini adalah MySQL. Fitur utama dari sistem ini adalah untuk pengajuan administrasi surat dan aspirasi oleh penduduk desa kepada pemerintah desa serta promosi produk UMKM. Selain itu juga terdapat fitur yang berfungsi untuk melihat berita dan informasi mengenai desa. Aplikasi ini sudah diuji dengan metode *blackbox* dengan hasil semua fungsi sudah berjalan dengan baik. Pengujian meliputi pengujian tampilan antarmuka sistem dan fungsionalitas fitur sistem. Berdasarkan hasil kuesioner dari 10 responden yang terdiri atas pemerintah desa, pelaku UMKM dan masyarakat, didapatkan hasil persentase kepuasan pengguna sebesar 88,3% yang berarti sangat memuaskan.

Kata kunci: Administrasi, *Crowdsourcing*, Desa, Pemerintah, Penduduk, UMKM, *Scrum*

ABSTRACT

Dwi Hapsari and Muhammad Irhamul Umam. "Digitalization of Administrative Services and Economic Resilience of Rural Community with Crowdsourcing Technique", Final Project of DIII Informatics Engineering Study Program, Electrical Engineering Department, Semarang State Polytechnic, under the guidance of Mardiyono, S.Kom., M.Sc. and Dr. Ir. Kurnianingsih, S.T, M.T., September 2021, 164..

The COVID-19 pandemic requires people to maintain social distance, especially in managing administration at the village office. This also interferes with MSMEs in promoting and selling their products which has an impact on decreasing their turnover. The purpose of making this final project is to build the Digitization of Administrative Services and Economic Resilience of Village Communities with Crowdsourcing Techniques. The method used in the development of this system is the Scrum method which has stages of product backlog, sprint planning meeting, sprint backlog, sprint execution, finished work. The development of this system applies crowdsourcing techniques that function to attract people's interest in MSME products when MSME actors promote their products on applications and are linked to marketplace applications. While the database used in this system is MySQL. The main feature of this system is for submitting administrative letters and aspirations by villagers to the village government as well as promoting MSME products. In addition there is also a feature that serves to view news and information about the village. This application has been tested using the blackbox method with the results that all functions are running well. Testing includes testing the system interface and system feature functionality. Based on the results of the questionnaire from 10 respondents consisting of the village government, MSME actors and the community, it was found that the percentage of user satisfaction was 88,3%, which means very satisfying.

Keywords: Administration, Crowdsourcing, Village, Government, Population, MSME, Scrum

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT, atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini. dengan judul **DIGITALISASI PELAYANAN ADMINISTRASI DAN KETAHANAN EKONOMI MASYARAKAT DESA DENGAN TEKNIK CROWDSOURCING** disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Program Studi Diploma III Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Semarang.

Pada kesempatan ini penulis menyampaikan terimakasih kepada:

1. Bapak Ir. Supriyadi, M.T, selaku Direktur Politeknik Negeri Semarang.
2. Bapak Yusnan Badruzzaman, S.T.,M.Eng, selaku Ketua Jurusan Teknik Elektro.
3. Ibu Idhawati Hestiningsih, S.Kom., M.Kom. selaku Ketua Program Studi D3 Teknik Informatika.
4. Bapak Mardiyono, S.Kom., M.Sc. dan Ibu Dr. Ir. Kurnianingsih, S.T., M.T. selaku Dosen Pembimbing yang memberikan masukan dan solusi kepada penulis mengenai pembuatan alat dan penyusunan laporan.
5. Kedua orang tua beserta saudara-saudara yang senantiasa mendukung penulis dan memberikan bantuan secara moril maupun materil dengan penuh keikhlasan.
6. Teman-teman Kelas IK-3A yang telah membantu dan mendukung Penulis.

Semoga Tugas Akhir ini dapat bermanfaat bagi Penulis khususnya dan pihak yang berkepentingan.

Mahasiswa,

Semarang, 16 September 2021

Mahasiswa,

Dwi Hapsari
NIM 3.34.18.0.08

Muhammad Irhamul Umam
NIM 3.34.18.0.16

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	ii
	iii
ABSTRAK	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xv
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan.....	3
1.4. Manfaat.....	3
1.5. Batasan Masalah.....	3
1.6. Sistematika	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Penelitian Terkait	5
2.2. Teknologi yang Digunakan	18
2.2.1. Crowdsourcing	18
2.2.2. Android	18
2.2.2. Visual Studio Code	18
2.2.3. MySQL.....	19
2.2.4. Firebase	20

2.2.5.	Laravel.....	21
2.2.6.	Flutter	22
2.2.7.	Vue Js	22
2.2.8.	Github.....	23
2.2.9.	Postman.....	24
2.2.10.	FCM.....	24
BAB III KEGIATAN PELAKSANAAN.....		26
3.1.	<i>Product Backlog</i>	27
3.2.	<i>Sprint Planning Meeting</i>	28
3.3.	<i>Sprint Backlog</i>	28
3.3.1.	<i>Sprint 1</i>	28
3.3.2.	<i>Sprint 2</i>	29
3.3.3.	<i>Sprint 3</i>	30
3.3.4.	<i>Sprint 4</i>	32
3.3.5.	<i>Sprint 5</i>	36
3.4.	<i>Sprint Execution</i>	37
3.4.1.	<i>Sprint 1:</i> Analisis Kebutuhan Sistem	37
3.4.2.	<i>Sprint 2:</i> Perancangan Diagram Sistem.....	38
3.4.3.	<i>Sprint 3:</i> Perancangan Antarmuka Sistem.....	59
3.4.4.	<i>Sprint 4:</i> Pembuatan dan Pengujian Fitur Aplikasi.....	72
3.4.5.	<i>Sprint 5:</i> Perancangan Uji Kepuasan Pengguna.....	117
BAB IV ANALISIS DAN PEMBAHASAN		119
4.1.	Analisis dan Pembahasan Sistem	119
4.1.1.	Analisis Aplikasi Android.....	119
4.1.2.	Analisis Website Administrator	119
4.2.	Pembahasan Sistem	121

4.2.1.	Rekapitulasi Uji Tampilan dan Fungsionalitas Aplikasi Android	121
4.2.2.	Rekapitulasi Uji Tampilan dan Fungsionalitas Website Administrator 132	
4.2.3.	Pengujian Tingkat Kepuasan Pengguna.....	157
BAB V KESIMPULAN		161
DAFTAR PUSTAKA		162
LAMPIRAN.....		165

DAFTAR GAMBAR

Gambar 2.1. Logo Visual Studio Code	19
Gambar 2.2. Logo MySQL	20
Gambar 2.3. Logo <i>Firebase</i>	21
Gambar 2.4. Logo Laravel	21
Gambar 2.5. Logo Flutter.....	22
Gambar 2.6. Logo Vue Js.....	23
Gambar 2.7. Logo Figma	Error! Bookmark not defined.
Gambar 2.8. Logo Github	24
Gambar 2.9. Logo Postman.....	24
Gambar 3.1. Cara kerja <i>Scrum</i> menggunakan metode <i>Agile</i>	26
Gambar 3.2. <i>Block Diagram</i>	38
Gambar 3.3. <i>Use Case Diagram</i> Sistem	40
Gambar 3.4. <i>Activity Diagram Login</i> Administrator	41
Gambar 3.5. <i>Activity Diagram</i> Menampilkan Dashboard.....	42
Gambar 3.6. <i>Activity Diagram</i> Manajemen Penduduk	43
Gambar 3.7. <i>Activity Diagram</i> Manajemen Inventaris Desa	44
Gambar 3.8. <i>Activity Diagram</i> Manajemen Keuangan Desa	45
Gambar 3.9. <i>Activity Diagram</i> Manajemen Berita	46
Gambar 3.10. <i>Activity Diagram</i> Manajemen Agenda.....	47
Gambar 3.11. <i>Activity Diagram</i> Manajemen Aspirasi	47
Gambar 3.12. <i>Activity Diagram</i> Manajemen UMKM	48
Gambar 3.13. <i>Activity Diagram</i> Manajemen Surat.....	49
Gambar 3.14. <i>Activity Diagram</i> Manajemen Profil Desa	50
Gambar 3.15. <i>Activity Diagram</i> Login dan Verifikasi <i>User</i>	51
Gambar 3.16. <i>Activity Diagram</i> Akses Infromasi Desa.....	51
Gambar 3.17. <i>Activity Diagram</i> Mendaftar UMKM.....	52
Gambar 3.18. <i>Activity Diagram</i> Akses Produk UMKM.....	53
Gambar 3.19. <i>Activity Diagram</i> Mengajukan Aspirasi.....	53
Gambar 3.20. <i>Activity Diagram</i> Mengajukan Surat.....	54
Gambar 3.21. <i>Activity Diagram</i> Manajemen Profil	55

Gambar 3.22. <i>Activity Diagram</i> Membuat Produk	55
Gambar 3.23. <i>Activity Diagram</i> Mempromosikan Produk	56
Gambar 3.24. <i>Activity Diagram</i> Mengubah Lokasi Toko.....	57
Gambar 3.25. <i>Entity Relationalship Diagram</i>	58
Gambar 3. 26 Tampilan Halaman <i>Splash Screen</i>	59
Gambar 3. 27 Tampilan Halaman <i>Login</i> dan Verifikasi.....	59
Gambar 3. 29 Tampilan Halaman <i>Dashboard</i>	60
Gambar 3. 30 Tampilan Halaman Berita	60
Gambar 3. 31 Tampilan Halaman Surat.....	61
Gambar 3. 32 Tampilan Halaman Detail Surat.....	61
Gambar 3. 33 Tampilan Halaman Profil	62
Gambar 3. 35 Tampilan Halaman Ubah Nomor WhatsApp.....	62
Gambar 3. 36 Tampilan Halaman UMKM	63
Gambar 3. 37 Tampilan Halaman Kelola UMKM.....	63
Gambar 3. 38 Tampilan Halaman Kelola Produk UMKM	64
Gambar 3. 39 Tampilan Halaman Produk UMKM.....	64
Gambar 3. 40 Tampilan Halaman Detail Produk.....	65
Gambar 3. 41 Tampilan Halaman Info Desa	65
Gambar 3. 42 Tampilan Halaman <i>Login</i> Administrator	66
Gambar 3. 43 Tampilan Halaman <i>Dashboard</i> Administrator.....	66
Gambar 3. 44 Tampilan Halaman Manajemen Penduduk	67
Gambar 3. 45 Tampilan Halaman Manajemen Inventaris	67
Gambar 3. 46 Tampilan Halaman Manajemen Keuangan	68
Gambar 3. 47 Tampilan Halaman Manajemen Berita	69
Gambar 3. 48 Tampilan Halaman Manajemen Agenda.....	69
Gambar 3. 49 Tampilan Halaman Manajemen Aspirasi.....	69
Gambar 3. 50 Tampilan Halaman Manajemen Surat.....	70
Gambar 3. 51 Tampilan Halaman Manajemen UMKM	70
Gambar 3. 52 Tampilan Halaman Manajemen Produk UMKM.....	71
Gambar 3. 53 Tampilan Halaman Manajemen Keahlian Penduduk.....	71
Gambar 3. 54 Tampilan Halaman Manajemen Profil Desa	72
Gambar 4.1. Parameter <i>Request API</i>	120

Gambar 4.2. Contoh <i>Response API</i>	121
Gambar 4.3. Halaman <i>Landing Page</i> pada Google Chrome.....	133
Gambar 4.4. Halaman <i>Landing Page</i> pada Mozilla Firefox	133
Gambar 4.5. Halaman <i>Login</i> pada Google Chrome	133
Gambar 4.6. Halaman <i>Login</i> pada Mozilla Firefox	134
Gambar 4.7. Halaman <i>Dashboard</i> pada Google Chrome	134
Gambar 4.8. Halaman <i>Dashboard</i> pada Mozilla Firefox.....	134
Gambar 4.9. Halaman Manajemen Penduduk pada Google Chrome	135
Gambar 4.10. Halaman Manajemen Penduduk pada Mozilla Firefox	135
Gambar 4.11. Halaman Tambah Data Penduduk pada Google Chrome.....	136
Gambar 4.12. Halaman Tambah Data Penduduk pada Mozilla Firefox	136
Gambar 4.13. Halaman Edit Data Penduduk pada Google Chrome	136
Gambar 4.14. Halaman Edit Data Penduduk pada Mozilla Firefox	137
Gambar 4.15. Halaman Detail Data Penduduk pada Google Chrome	137
Gambar 4.16. Halaman Detail Data Penduduk pada Mozilla Firefox	137
Gambar 4.17. Halaman Manajemen Inventaris pada Google Chrome	138
Gambar 4.18. Halaman Manajemen Inventaris pada Mozilla Firefox.....	138
Gambar 4.19. Halaman Manajemen Inventaris pada Google Chrome	139
Gambar 4.20. Halaman Manajemen Inventaris pada Mozilla Firefox	139
Gambar 4.21. Halaman Edit Data Inventaris pada Google Chrome	139
Gambar 4.22. Halaman Edit Data Inventaris pada Mozilla Firefox	140
Gambar 4.23. Halaman Detail Data Inventaris pada Google Chrome	140
Gambar 4.24. Halaman Detail Data Inventaris pada Mozilla Firefox	140
Gambar 4.25. Halaman Manajemen Keuangan pada Google Chrome	141
Gambar 4.26. Halaman Manajemen Keuangan pada Mozilla Firefox.....	141
Gambar 4.27. Halaman Tambah Data Keuangan pada Google Chrome	142
Gambar 4.28. Halaman Tambah Data Keuangan pada Mozilla Firefox.....	142
Gambar 4.29. Halaman Edit Data Keuangan pada Google Chrome.....	142
Gambar 4.30. Halaman Edit Data Keuangan pada Mozilla Firefox	143
Gambar 4.31. Halaman Detail Data Keuangan pada Google Chrome.....	143
Gambar 4.32. Halaman Detail Data Keuangan pada Mozilla Firefox	143
Gambar 4.33. Halaman Manajemen Berita pada Google Chrome.....	144

Gambar 4.34. Halaman Manajemen Berita pada Mozilla Firefox	144
Gambar 4.35. Halaman Tambah Data Berita pada Google Chrome	145
Gambar 4.36. Halaman Tambah Data Berita pada Mozilla Firefox	145
Gambar 4.37. Halaman Edit Data Berita pada Google Chrome	145
Gambar 4.38. Halaman Edit Data Berita pada Mozilla Firefox.....	146
Gambar 4.39. Halaman Detail Berita pada Google Chrome.....	146
Gambar 4.40. Halaman Detail Berita pada Mozilla Firefox	146
Gambar 4.41. Halaman Manajemen Agenda pada Google Chrome	147
Gambar 4.42. Halaman Manajemen Agenda pada Mozilla Firefox	147
Gambar 4.43. Halaman Tambah Data Agenda pada Google Chrome	148
Gambar 4.44. Halaman Tambah Data Agenda pada Mozilla Firefox.....	148
Gambar 4.45. Halaman Edit Data Agenda pada Google Chrome.....	148
Gambar 4.46. Halaman Tambah Data Agenda pada Mozilla Firefox.....	149
Gambar 4.47. Halaman Detail Agenda pada Google Chrome	149
Gambar 4.48. Halaman Detail Agenda pada Mozilla Firefox	149
Gambar 4.49. Halaman Manajemen Aspirasi pada Google Chrome	150
Gambar 4.50. Halaman Manajemen Aspirasi pada Mozilla Firefox	150
Gambar 4.51. Halaman Manajemen Surat pada Google Chrome.....	151
Gambar 4.52. Halaman Manajemen Surat pada Mozilla Firefox	151
Gambar 4.53. Halaman Detail Surat pada Google Chrome	151
Gambar 4.54. Halaman Detail Surat pada Mozilla Firefox	152
Gambar 4.55. Halaman Manajemen UMKM pada Google Chrome	152
Gambar 4.56. Halaman Manajemen UMKM pada Mozilla Firefox	152
Gambar 4.57. Halaman Manajemen Produk pada Google Chrome.....	153
Gambar 4.58. Halaman Manajemen Produk pada Mozilla Firefox	153
Gambar 4.59. Halaman Manajemen Keahlian pada Google Chrome	154
Gambar 4.60. Halaman Manajemen Keahlian pada Mozilla Firefox	154
Gambar 4.61. Halaman Manajemen Profil pada Google Chrome	154
Gambar 4.62. Halaman Manajemen Profil pada Mozilla Firefox.....	155

DAFTAR TABEL

Tabel 2.1. Tabel Perbandingan Sistem.....	14
Tabel 3.1. Daftar <i>Sprint</i> Kebutuhan	28
Tabel 3.2. <i>Sprint</i> 1: Analisis Kebutuhan Sistem	29
Tabel 3.3. <i>Sprint</i> 2: Analisis Perancangan Diagram Sistem	29
Tabel 3.4. <i>Sprint</i> 3: Perancangan Antarmuka Sistem	30
Tabel 3.5. <i>Sprint</i> 5: Pembuatan dan Pengujian Fitur Aplikasi.....	32
Tabel 3.6. Perancangan Uji Kepuasan Pengguna	36
Tabel 3.7 Pengujian Kompatibilitas Fitur <i>Splash Screen</i>	73
Tabel 3.8 Pengujian Fungsionalitas Fitur <i>Splash Screen</i>	73
Tabel 3.9 Pengujian Kompatibilitas Fitur <i>Login</i> dan Verifikasi.....	74
Tabel 3.10 Pengujian Fungsionalitas Fitur <i>Login</i> dan verifikasi	75
Tabel 3.11 Pengujian Kompatibilitas Fitur <i>Dashboard</i>	76
Tabel 3.12 Pengujian Fungsionalitas Fitur <i>Dashboard</i>	76
Tabel 3.13 Pengujian Kompatibilitas Fitur Berita	77
Tabel 3.14 Pengujian Fungsionalitas Fitur Berita.....	78
Tabel 3.15 Pengujian Kompatibilitas Fitur Administrasi Surat.....	79
Tabel 3.16 Pengujian Fungsionalitas Fitur Administrasi Surat	80
Tabel 3.17 Pengujian Kompatibilitas Fitur Aspirasi.....	81
Tabel 3.18 Pengujian Fungsionalitas Fitur Aspirasi	81
Tabel 3.19 Pengujian Kompatibilitas Fitur Profil	82
Tabel 3.20 Pengujian Fungsionalitas Fitur Profil	83
Tabel 3.21 Pengujian Kompatibilitas Fitur UMKM	84
Tabel 3.22 Pengujian Fungsionalitas Fitur UMKM.....	85
Tabel 3.23 Pengujian Kompatibilitas Fitur Keahlian.....	86
Tabel 3.24 Pengujian Fungsionalitas Fitur Keahlian Desa	87
Tabel 3.25 Pengujian Kompatibilitas Fitur <i>Login</i>	88
Tabel 3.26 Pengujian Fungsionalitas Fitur <i>Login</i> Administrator.....	89
Tabel 3.27 Pengujian Kompatibilitas Fitur <i>Dashboard</i> Administrator.....	89
Tabel 3.28 Pengujian Fungsionalitas Fitur <i>Dashboard</i> Administrator	90
Tabel 3.29 Pengujian Kompatibilitas Fitur Manajemen Penduduk	91
Tabel 3.30 Pengujian Fungsionalitas Fitur Manajemen Penduduk.....	92

Tabel 3. 31 Pengujian Kompatibilitas Fitur Manajemen Penduduk	94
Tabel 3. 32 Pengujian Fungsionalitas Fitur Manajemen Inventaris Desa.....	96
Tabel 3. 33 Pengujian Kompatibilitas Fitur Manajemen Keuangan	98
Tabel 3. 34 Pengujian Fungsionalitas Fitur Manajemen Keuangan	99
Tabel 3. 35 Pengujian Kompatibilitas Fitur Manajemen Berita	100
Tabel 3. 36 Pengujian Fungsionalitas Fitur Manajemen Berita.....	102
Tabel 3. 37 Pengujian Kompatibilitas Fitur Manajemen Agenda.....	103
Tabel 3. 38 Pengujian Fungsionalitas Fitur Manajemen Agenda	104
Tabel 3. 39 Pengujian Kompatibilitas Fitur Manajemen Aspirasi.....	106
Tabel 3. 40 Pengujian Fungsionalitas Fitur Manajemen Aspirasi	107
Tabel 3. 41 Pengujian Kompatibilitas Fitur Manajemen Pengajuan Surat.....	108
Tabel 3. 42 Pengujian Fungsionalitas Fitur Manajemen Pengajuan Surat.....	108
Tabel 3. 43 Pengujian Kompatibilitas Fitur Manajemen UMKM	110
Tabel 3. 44 Pengujian Fungsionalitas Fitur Manajemen UMKM.....	110
Tabel 3. 45 Pengujian Kompatibilitas Fitur Manajemen Produk UMKM.....	111
Tabel 3. 46 Pengujian Fungsionalitas Fitur Manajemen Produk UMKM	112
Tabel 3. 47 Pengujian Kompatibilitas Fitur Manajemen Keahlian.....	114
Tabel 3. 48 Pengujian Fungsionalitas Fitur Manajemen Keahlian	114
Tabel 3. 49 Pengujian Kompatibilitas Fitur Manajemen Profil Desa	115
Tabel 3. 50 Pengujian Fungsionalitas Fitur Manajemen Profil Desa.....	116
Tabel 3. 51. Daftar Pertanyaan.....	117
Tabel 3. 52. Indikator Penilaian	117
Tabel 3. 53. Indikator Kepuasan Pengguna	118
Tabel 4. 1. Tabel Spesifikasi <i>Smartphone</i> Penguinji Aplikasi Android.....	121
Tabel 4. 2. Rekap Pengujian Tampilan Aplikasi Android	122
Tabel 4. 5 Hasil Kuisioner Responden.....	158

DAFTAR LAMPIRAN

Lampiran 1 Dokumentasi	166
Lampiran 2 Lembar Kontrol Bimbingan Tugas Akhir	167
Lampiran 3 Form Revisi Tugas Akhir	173
Lampiran 4 Surat Keterangan Selesai Revisi Tugas Akhir.....	176
Lampiran 5 Kode <i>Main File</i>	177
Lampiran 6 Kode Fitur <i>Splash Screen</i>	178
Lampiran 7 Kode Fitur <i>Login</i>	180
Lampiran 8 Kode Validasi OTP.....	183
Lampiran 9 Kode Fitur Beranda	188
Lampiran 10 Kode Fitur Berita	198
Lampiran 11 Kode Fitur Detail Berita	201
Lampiran 12 Kode Fitur Administrasi Surat.....	203
Lampiran 13 Kode Fitur Detail Surat.....	207
Lampiran 14 Kode Fitur Profil.....	212
Lampiran 15 Kode Fitur UMKM.....	223
Lampiran 16 Kode Fitur Produk UMKM	228
Lampiran 17 Kode Fitur Keahlian Penduduk	233
Lampiran 18 Kode Fitur Login Administrator.....	237
Lampiran 19 Kode Fitur Dashboard Administrator	237
Lampiran 20 Kode Fitur Manajemen Penduduk	238
Lampiran 21 Kode Fitur Manajemen Inventaris Desa.....	242
Lampiran 22 Kode Fitur Manajemen Keuangan.....	243
Lampiran 23 Kode Fitur Manajemen Berita	243
Lampiran 24 Kode Fitur Manajemen Agenda	245
Lampiran 25 Kode Fitur Manajemen Aspirasi.....	246
Lampiran 26 Kode Fitur Manajemen Pengajuan Surat.....	247
Lampiran 27 Kode Fitur Manajemen UMKM	248
Lampiran 28 Kode Fitur Manajemen Produk UMKM	250
Lampiran 29 Kode Fitur Manajemen Keahlian	251
Lampiran 30 Kode Fitur Profil Desa.....	252

BAB I PENDAHULUAN

1.1. Latar Belakang

Dalam rangka mewujudkan upaya pemerintah dalam mengurangi penyebaran *Coronavirus disease 2019* atau Covid-19, maka harus *social distancing* sesuai dengan Undang-Undang Nomor 6 Tahun 2018 yaitu tentang Kekarantinaan Kesehatan, dimana masyarakat perlu membatasi kegiatan sosial atau *social distancing* (Telaumbanua 2020). Misalkan dalam kegiatan pengurusan administrasi yang sebelumnya penduduk harus datang langsung ke kantor kepala desa, sekarang harus dikurangi. Jumlah penduduk yang berkunjung ke kantor desa untuk mengurus kepentingan administrasi maupun hanya berkunjung ke kantor desa berbeda-beda setiap hari. Hal ini dapat dibuktikan dari data buku agenda keluar dan buku tamu yang diperoleh penulis dari salah satu desa yakni Desa Sinungrejo, Kecamatan Ambal, Kabupaten Kebumen, jumlah pengunjung dengan tujuan mengurus administrasi dan berkunjung ke kantor desa berkisar 1 sampai 10 orang per hari dengan jumlah penduduk desa 2.123 orang pada bulan Agustus tahun 2021. Hal tersebut akan berdampak melambatnya pengurusan administrasi dan pelayanan yang diberikan oleh pemerintah desa. Dalam penerapan *social distancing* tersebut, secara tidak langsung kegiatan sosial masyarakat juga akan berkurang. Para pelaku Usaha Mikro, Kecil, dan Menengah (UMKM) juga kesulitan dalam mempromosikan dan menjual produknya secara langsung. Keadaan tersebut akan menimbulkan ketahanan ekonomi masyarakat tersebut akan semakin melemah.

Saat ini sudah terdapat beberapa penelitian yang diciptakan untuk menangani masalah digitalisasi pelayanan administrasi. Sistem informasi administrasi kependudukan berbasis web desa dengan menggunakan analisis PIECES (*Performance, Information, Economy, control, Efficiency and Service*) (Paryanta, Sutariyani dan Susilowati, 2017). Sistem yang lain yaitu sistem informasi manajemen desa berbasis web dengan menggunakan PHP dan MySQL (Rohaeti dan Zaliluddin, 2018). Sistem informasi desa dengan pendataan penduduk desa berdasarkan Kartu Keluarga yang dimiliki penduduk setempat (Sihombing 2018). Penelitian menggunakan metode *prototype* sebagai pembuatan sistem informasi

pelayanan masyarakat desa berbasis web (Syukron 2019). Pembuatan sistem pelayanan administrasi berbasis web untuk membantu dalam mengurus administrasi desa (Rahmawati dan Fatmawati, 2020). Namun, dari sistem yang dibangun tersebut belum ada yang terintegrasi dengan *smartphone* dan belum adanya fitur untuk membantu UMKM dalam mempromosikan produknya menggunakan teknik *crowdsourcing* guna menjaga ketahanan ekonomi masyarakat desa.

Laporan Tugas Akhir ini membahas mengenai pembangunan sistem digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing* secara *mobile* sebagai solusi dari permasalahan di atas. Pada sistem ini terdapat fitur untuk mengelola administrasi guna meningkatkan kualitas pelayanan kepada masyarakat desa dan fitur untuk mempromosikan dan menjual produk UMKM yang dihubungkan aplikasi *marketplace* dengan memanfaatkan teknik *crowdsourcing*. Selain itu, sistem ini juga dapat membantu menyalurkan aspirasi masyarakat desa kepada pemerintah desa. Gagasan tersebut menjadi solusi dalam menciptakan desa digital untuk membantu kebutuhan pencatatan dan pengelolaan administrasi serta diharapkan dapat menyejahterakan masyarakat desa dikala pandemi.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada laporan TA ini adalah sebagai berikut:

- 1) Belum tersedianya sistem pengelolaan administrasi secara online untuk mengurangi potensi terjadi penularan covid-19.
- 2) Belum tersedianya media penyampaian secara *online* terhadap permasalahan/ide masyarakat.
- 3) Belum tersedianya aplikasi yang dapat mempromosikan dan menjual produk secara *online*.

1.3. Tujuan

Tujuan pembuatan sistem ini adalah:

- 1) Membuat sistem pengolahan data pemerintahan desa untuk melayani proses administrasi masyarakat desa.
- 2) Membangun sistem yang dapat membantu masyarakat desa dalam menyampaikan aspirasinya secara *online*.
- 3) Membangun sistem yang dapat membantu UMKM dalam mempromosikan dan menjual produknya secara *online*.

1.4. Manfaat

Manfaat pembuatan sistem ini adalah:

- 1) Membantu pelayanan pemerintah kepada masyarakat desa melalui kemudahan akses informasi baik bagi masyarakat.
- 2) Membantu pembangunan dan pengembangan desa sehingga menciptakan masyarakat berbasis komunitas informasi (masyarakat digital) yang lebih berkualitas.
- 3) Mempermudah pelaku UMKM dalam mempromosikan produknya dengan menghubungkan pada aplikasi *marketplace*.

1.5. Batasan Masalah

Agar tidak meluas dari pembahasan, dalam penulisan ini penulis membatasi pada ruang lingkup sebagai berikut :

- 1) Sistem yang terdapat dalam administrator berbentuk website ini terdiri dari pengelolaan administrasi dan keuangan desa, data produk UMKM, berita, aspirasi masyarakat dan surat desa.
- 2) Aplikasi android untuk pengguna digunakan untuk kepentingan warga desa seperti meminta surat keterangan, menyampaikan aspirasi, mendapatkan informasi berita dan agenda, serta membantu pelaku UMKM masyarakat desa dalam mempromosikan produknya.
- 3) Sistem ini dilengkapi dengan *landing page* yang dapat diakses masyarakat umum untuk mengetahui informasi umum mengenai desa.

1.6. Sistematika

Sistematika penulisan pada Tugas Akhir ini dibagi menjadi beberapa bab yang meliputi:

BAB I PENDAHULUAN

Berisi latar belakang yang juga meliputi perumusan masalah, tujuan, dan manfaat pembuatan Tugas Akhir, batasan masalah sistem, dan sistematika penulisan laporan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Berisi beberapa penelitian yang telah dilakukan oleh ahli dan berkaitan dengan Tugas Akhir serta membantu dalam pembuatan Tugas Akhir. Selain itu terdapat perbandingan sistem dengan yang telah ada serta ditambahkan dengan dasar teori seputar teknologi yang digunakan.

BAB III KEGIATAN PELAKSANAAN

Berisi kegiatan pelaksanaan pembuatan sistem menggunakan metode Scrum, aktivitas pembuatan dan implementasinya.

BAB IV ANALISIS DAN PEMBAHASAN

Berisi hasil dari analisis dan rekapitulasi pengujian terhadap sistem yang telah dikerjakan dalam Tugas Akhir ini.

BAB V KESIMPULAN

Berisi kesimpulan dibuatnya laporan Tugas Akhir.

DAFTAR PUSTAKA

Daftar yang menampilkan sumber pustaka yang dijadikan panduan penggerjaan laporan Tugas Akhir.

LAMPIRAN

Dokumen tambahan yang ditambahkan (dilampirkan) ke laporan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas tentang penelitian yang sebelumnya telah dilakukan oleh peneliti serta berkaitan dengan Tugas Akhir yang dibuat tentang sistem digitalisasi desa. Selain itu, terdapat berbagai perbandingan yang telah dibuat peneliti sebelumnya dengan Tugas Akhir Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa Dengan Teknik *Crowdsourcing*. Beberapa teknologi yang ditambahkan ke dalam sistem juga dipaparkan dalam bab ini.

2.1. Penelitian Terkait

Penelitian dengan judul “Sistem Informasi Administrasi Kependudukan Berbasis Web Pada Desa Bogangin Sumpiuh” menghasilkan sistem informasi berbasis web yang dirancang untuk mempermudah kantor kepala desa dalam hal mengelola data penduduk secara cepat dan tepat. Pada sistem ini, web hanya dapat diakses oleh *user* yaitu warga dan administrator. Metode yang digunakan dalam pengembangan sistem ini yaitu SDLC (*System Development Life Cycle*) dengan model proses *waterfall*. Untuk pengunjung hanya bisa mengakses *website* utamanya saja karena sistem informasi administrasi berbasis web ini hanya dapat diakses oleh *user* yaitu warga Desa Bogangin Sumpiuh dan administrator (Imaniawan dan Wati, 2017). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Pemanfaatan *E-Government* Pada Desa Wonokarto Untuk Meningkatkan Akurasi dan Informasi Potensi Desa” merupakan elektronik kepemerintahan berbasis web yang bertujuan meningkatkan kualitas pelayanan pemerintah kepada publik. Pengembangan *e-government* merupakan upaya untuk mengembangkan penyelenggaraan pemerintah yang berbasis elektronik dalam rangka meningkatkan kualitas pelayanan publik secara efektif dan efisien. Selama ini pelayanan yang ada di desa Wonokarto masih menggunakan metode manual sehingga dalam pelayanan masyarakat tentang informasi desa Wonokarto kurang

optimal, dengan adanya hal ini maka untuk mengatasinya di buatlah suatu sistem yang baru bisa membantu aparatur pemerintah desa Wonokarto dalam memberikan informasi kepada masyarakat (Rachman dan Noviyanto, 2017). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “*E-government* Pada Pekon Srimulyo Kabupaten Lampung Tengah” merupakan salah satu media yang berbentuk *website* yang dibuat untuk menyampaikan informasi tentang keseluruhan mengenai Pekon Srimulyo yaitu potensi Pekon Srimulyo, dan kegiatan Pekon Srimulyo. Proses pembuatan *e-government* pada Pekon Srimulyo ini dilakukan dengan menganalisis sistem yang telah ada pada Pekon Srimulyo, kemudian pembuatan beberapa diagram yaitu, diagram konteks. Dilanjutkan dengan mendesain web dan teknik pemrograman terstruktur yang menggunakan bahasa pemrograman PHP dan basis data MySQL (Muntofiah dan Fitrian, 2017). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Aplikasi Sistem Informasi Desa Sebagai Teknologi Tepat Guna untuk Pendataan Penduduk dan Potensi Desa” membuat sistem informasi desa yang komunikatif dengan *user* sehingga memudahkan dalam penginputan data dan mengekspor data dalam sebuah laporan. Sistem dibangun berdasarkan garis besar peraturan pemerintah dan masukan atau saran dari pejabat setempat sehingga menghasilkan beberapa menu *input-output* meliputi data profil desa, data kependudukan, data pendidikan, data potensi wisata desa, data administrasi termasuk surat masuk dan keluar, data inventaris desa, dan data pendapatan dan pengeluaran desa (Syaharuddin 2017). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Pengembangan *Website* dan Sistem Informasi Desa Di Kabupaten Tulungagung” memanfaatkan sebuah *open source* yaitu OpenSID. Web dan SID ini nantinya akan dapat digunakan dengan baik oleh aparat desa dalam menjalankan pekerjaannya. Tujuan pelaksanaan program ini adalah untuk memberikan media untuk mempublikasikan segala potensi yang dimiliki desa, tertatanya pengelolaan administrasi yang telah berbasis digital serta meningkatnya kemampuan dan keterampilan aparatur desa. Sebagai hasil program berupa peningkatan kualitas pelayanan kepada masyarakat dan publikasi ilmiah (Rozi dan Listiawan, 2017). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Aplikasi Simade (Sistem Informasi Manajemen Desa Dalam Meningkatkan Pelayanan Administrasi Di Kepenghuluan Bakti Makmur Kecamatan Bagan Sinembah Kab. Rokan Hilir Riau” pelayanan dikantor kepenghuluan bakti makmur mempermudah mengolah data penduduk, baik itu data kelahiran, kematian, perangkat desa dan potensi desa yang dimiliki dikepenghuluan bakti makmur. Selain itu juga manajemen yang dimiliki dalam aplikasi SIMADE ini surat menyurat dan pengarsipan yang akan disimpan di dalam *database*. Pendataan penduduk di kepenghuluan bakti makmur ini berdasarkan Kartu Keluarga yang dimiliki penduduk setempat (Sihombing 2018). Namun dalam sistem ini belum menampilkan alokasi dana sehingga kurang transparan. Selain itu sistem ini juga belum bisa menampilkan laporan yang dapat diolah atau dikonversi dalam bentuk *Ms. Excel*.

Penelitian dengan judul “Implementasi Web *Mobile* Sebagai Media Informasi Pemberdayaan Masyarakat Di Desa Pirngadi” dilakukan karena Desa Pirngadi masih menggunakan proses manual dalam memberikan pelayanan pada pemerintah desa, sehingga kemungkinan data yang diajukan kurang akurat dan lambat, untuk memfasilitasi pemerintah desa dan penduduk desa Pirngadi dalam mengakses informasi sehingga kualitas pelayanan umum menjadi lebih responsif, efektif,

efisien dan *accountable*, dapat dijelaskan atau dimintai pertanggung jawaban. Sebuah sistem *e-government* berbasis web *mobile* yang meliputi perencanaan, analisis, perancangan dan implementasi dari sistem (Usmanto et al. 2018). Pada sistem ini belum terdapat fitur untuk mempromosikan produk UMKM desa serta wadah untuk menyalurkan aspirasi warga desa.

Penelitian dengan judul “Implementasi *E-Goverment* Sebagai Upaya Peningkatan Potensi Desa Di Desa Bumirejo Menggunakan Web *Mobile*” yaitu perancangan sistem informasi implementasi *e-government* sebagai upaya peningkatan potensi desa pada Desa Bumirejo. Sistem *e-government* peningkatan potensi tersebut sebelumnya masih menggunakan metode konvensional, kini akan lebih dikembangkan dengan metode-metode lain untuk mempermudah proses upaya peningkatan potensi desa dengan menggunakan aplikasi web *mobile* ini memudahkan masyarakat mengakses data pada android atau *smartphone*. (Pratiwi dan Muslihudin, 2018). Namun penelitian ini belum menggunakan *platform* android sebagai media informasi masyarakat desa dan meningkatkan potensi desa. Penelitian dengan judul “Implementasi Web *Government* Dalam Meningkatkan Potensi Produk Unggulan Desa Berbasis Android” berdasarkan dari hasil penelitian dan pembahasan dalam implementasi web *government* dalam meningkatkan potensi produk unggulan desa berbasis android pada Pekon Sukoyoso dapat disimpulkan bahwa implementasi web *government* berbasis android berjalan dengan baik dengan sistem operasi android dengan *user interface* yang menarik dan dinamis, sehingga pengguna merasa nyaman dan mendapatkan kemudahan dalam menggunakannya. Aplikasi web *government* berbasis android yang telah dibangun merupakan salah satu penggunaan teknologi informasi yang tepat dalam mempromosikan produk unggul Pekon maupun sumber daya lainnya kepada masyarakat luas sehingga dapat menjadi jembatan para pengusaha dan pembeli produk unggul Pekon (Samsudin dan Muslihudin, 2018). Pada sistem yang dibuat fitur untuk meningkatkan potensi produk masih perlu dikembangkan, seperti dihubungkan melalui aplikasi *marketplace*.

Penelitian dengan judul “Sistem Informasi Administrasi Desa di Kantor Desa Sumbersekar” sistem yang dibangun bisa membantu kantor desa dalam pembuatan surat keterangan yang diminta oleh warga desa, mampu memberikan notifikasi mengenai permintaan surat yang diminta oleh warga, menampilkan dan mencetak Kartu Tanda Penduduk (KTP) dan Kartu Keluarga (KK), dapat menampilkan laporan mengenai jumlah surat yang telah dicetak, dapat difilter sesuai bulan, tahun, dan jenis surat yang dipilih, dan juga dapat menampilkan *dashboard* berbentuk grafik yang digunakan untuk memberikan informasi kepada tingkat eksekutif secara akurat, mudah, dan cepat (Susilowati dan Chandra, 2018). Tetapi sistem ini hanya ditujukan untuk kantor desa belum ditujukan kepada masyarakat desa. Oleh karena itu, perlu pengembangan *platform* android sebagai media informasi untuk masyarakat desa.

Penelitian dengan judul “Implementasi Pelayanan Administrasi Kependudukan Desa Kalimanah Wetan Purbalingga Berbasis *Online*” telah diimplementasikan dan dapat memudahkan dalam memberikan pelayanan dan informasi kepada masyarakat secara lebih cepat, laporan kependudukan lebih mudah dan efisien karena data-data yang diperlukan dalam proses penyimpanan dengan *database* dan terdapat fasilitas cetak surat pengantar dan surat keterangan bagi warga Desa Kalimanah Wetan, Purbalingga. Sistem ini dapat mengelola data warga sehingga setiap pengelolaan surat selalu terhubung pada *database* warga sehingga surat yang dikeluarkan sesuai dengan data warga di kelurahan dan format surat telah diatur oleh sistem sehingga terbantu dalam pengarsipan yang tertata dengan baik. Sistem ini juga telah memudahkan petugas desa untuk mensosialisasikan program-program desa sehingga lebih efektif dan efisien dalam pelayanan, dan memudahkan warga dalam mendapat informasi seputar kegiatan dan kejadian di desa Kalimanah Wetan (Setyowati, Anton dan Radiyah, 2019). Sistem ini belum terdapat aplikasi android, dimana akan lebih memudahkan penduduk dalam melakukan pembuatan surat dan mendapatkan informasi secara lebih cepat.

Penelitian dengan judul “Inovasi Pelayanan Administrasi Kependudukan Secara *Online* di Kabupaten Pati” Pemerintah Kabupaten Pati melalui Dinas

Kependudukan dan Pencatatan Sipil (Disdukcapil) telah menciptakan inovasi dengan memanfaatkan kemajuan teknologi. Pemanfaatan tersebut diwujudkan dalam kepengurusan administrasi kependudukan di Kabupaten Pati yang dapat dilakukan secara *online*. Disdukcapil Kabupaten Pati menciptakan aplikasi dalam kepengurusan administrasi kependudukan yang disebut dengan Tarjilu Okke (Daftar Siji Enthuk Telu). Tarjilu Okke dapat dimanfaatkan dalam tiga kepengurusan administrasi kependudukan seperti E-KTP, akta kelahiran, dan akta kematian (Soraya 2019). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala)” menghasilkan sebuah *website* informasi desa, yang dapat membantu masyarakat khususnya desa Netpala kabupaten Timor Tengah Selatan dan umumnya kepada semua masyarakat yang membutuhkan informasi tentang desa Netpala. Pada *website* ini terdapat empat menu utama yaitu, profil, lembaga desa, statistik dan potensi keunggulan desa. Dengan adanya *website* ini akses informasi akan semakin mudah, cepat dan akurat dan peningkatan pelayanan kepada masyarakat akan lebih baik lagi dengan harapan, Implementasi dari teknologi informasi berbasis *website* ini harus didukung dengan infrastruktur yang memadai dan sumber daya manusia yang handal (Jimi 2019). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Aplikasi Pelayanan Administrasi Desa Karang Dima Berbasis Web” bertujuan menyelesaikan masalah yang ada dalam pelayanan administrasi di desa Karang Dima dengan merancang dan membangun aplikasi pelayanan administrasi penduduk. Aplikasi ini fokus pada pelayanan administrasi (pembuatan surat) agar proses yang dilakukan lebih mudah dan cepat sehingga dapat memuaskan masyarakat dalam pelayanan administrasi oleh *staff* desa Karang

Dima. Aplikasi berbasis web ini dikembangkan menggunakan bahasa pemrograman PHP dengan menggunakan Mysql sebagai *database*. Metode pengembangan perangkat lunak menggunakan metode *waterfall*. Pengujian perangkat lunak dilakukan melalui pengujian *Black-Box*. Hasil akhir dari penelitian ini adalah aplikasi pelayanan administrasi desa Karang Dima berbasis web yang dapat membantu admin desa dalam memberikan pelayanan administrasi kepada warga desa secara cepat dan tepat (Putra, Hidayatullah dan Juniarta, 2019). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Rancang Bangun Sistem Administrasi Desa Berbasis Android di Kantor Kepala Desa Keramat Jaya Kabupaten Majalengka” menghasilkan aplikasi administrasi desa berbasis android yang dapat menyediakan informasi dan administrasi desa kepada masyarakat luas. Sehingga aplikasi tersebut dapat diakses kapanpun dan dimanapun oleh masyarakat dengan teknologi internet. Selain itu juga mampu memetakan keadaan desa Keramat Jaya sehingga bisa terpublikasikan kepada masyarakat luas dan memudahkan pegawai desa dalam pengelolaan data administrasi desa serta memudahkan masyarakat dalam pembuatan data yang dibutuhkan oleh masyarakat (Nurmalasari dan Sujadi, 2019). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Pengembangan Sistem Informasi Pelayanan Desa Berbasis Web Dengan Metode Prototyping Pada Desa Leran” mengembangkan sistem dengan menggunakan metode *prototyping*. Sistem ini nantinya akan memiliki berberapa fitur, di antaranya adalah: pembuatan laporan mandiri, pengajuan surat pengantar dari desa secara online, pembuatan biodata masyarakat desa, dan juga memuat profil desa Leran. Hasil dari penilitian ini adalah sistem informasi pelayanan desa yang dapat membantu pekerjaan para pegawai

pemerintahan di desa Leran, dan juga memudahkan masyarakat desa dalam medapatkan informasi tentang layanan ataupun informasi lainnya di desa Leran (Kurniawan, Chabibi dan Dewi, 2020). Namun sistem ini masih perlu dikembangkan lagi dengan ditambahkan *platform android* sebagai sistem informasi dan pelayanan administrasi untuk masyarakat desa.

Penelitian dengan judul “Sistem Informasi Pelayanan Masyarakat Berbasis Android Pada Kantor Desa Kaliba Mamase” dirancang menggunakan model UML terdiri dari rancangan diagram aktivitas, *diagram sequence*, *diagram class*. Rancangan diagram ini menghasilkan rancangan *login*, rancangan halaman kelola pelayanan, rancangan halaman kelola berita, rancangan halaman kelola galeri, rancangan halaman kegiatan desa, rancangan halaman *home user*. Dengan diterapkannya sistem informasi pelayanan masyarakat berbasis android pada kantor Desa Kaliba Mamase memberikan dampak yang baik, serta efektif bagi masyarakat dalam pengurusannya dan memudahkan aparat desa dalam pengolahan data informasi guna memberikan pelayanan yang maksimal pada warga. Hal ini juga dapat menjadi contoh atau motivasi bagi kantor desa lainnya. (Ritnawati, Suppa dan Muhallim, 2020). Namun penelitian belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Sistem Informasi Pengolahan Data Kependudukan Pada Kantor Desa Sampean Berbasis Android” sistem informasi kependudukan yang telah bangun dapat menggantikan sistem kependudukan yang masih dilakukan dengan menggunakan aplikasi *Microsoft Excel* menjadi sistem informasi kependudukan secara online dengan menggunakan *smartphone android*. Sistem informasi kependudukan yang dibangun dapat memudahkan admin kantor desa dalam melakukan pengolahan data penduduk secara cepat dan mudah melalui *smartphone android* (Alda 2020). Namun pada sistem ini hanya mengembangkan *platform android* saja, sehingga perlu ditambahkan sistem berbasis *website* untuk digunakan dalam pengolahan data kependudukan.

Penelitian dengan judul “Rancangan Bangunan Sistem Informasi Administrasi desa Berbasis Web menggunakan PHP dan MySQL” di latar belakangi perkembangan teknologi yang begitu pesat, khususnya di bidang sistem informasi. Desa Sukamulya masih belum memiliki Sistem Informasi yang memadai dalam menunjang pelayanannya terhadap masyarakat, terutama di bidang administrasi surat. Metode penelitian yang digunakan adalah metode *Extreme Programing* (XP), metode ini memiliki 4 tahapan yaitu: *Planning, Design, Coding dan Testing*. Untuk metode pengumpulan data yang digunakan adalah metode observasi dan metode literatur. Hasil dari penelitian ini berupa sistem informasi administrasi desa yang dapat menunjang pelayanan di desa Sukamulya. Dari hasil pengujian yang dilakukan Sistem Informasi ini dibuat menggunakan bahasa pemrograman PHP dan Mysql sebagai *databasenya*, Sistem Informasi ini dapat menyajikan data penduduk dan surat desa (Winanjar dan Susanti, 2021). Namun pada sistem ini belum manambahkan aspirasi masyarakat desa dan mempromosikan produk UMKM dengan menggunakan *platform android*.

Penelitian dengan judul “Rancang Bangun Sistem Informasi Manajemen Kependudukan Desa Adat Terintegrasi Berbasis Web” berfokus kepada pendataan krama desa adat yang ada di Bali, seperti pendataan kelahiran, perkawinan, kematian, perceraian, adopsi, krama mipil, krama tamiu, tamiu, dan pendataan keluarga krama desa adat. Sistem ini dapat digunakan oleh Dinas Pemajuan Masyarakat Adat sebagai pihak yang mengelola sistem secara keseluruhan, Prajuru Desa Adat sebagai pengelola sistem pada masing-masing desa adat, serta krama sebagai objek pendataan yang dapat melakukan perubahan data krama melalui sistem. Sistem ini dikembangkan dengan menggunakan *framework Laravel 7, Android Studio*, serta *database MySQL*. (Purnawan 2021). Namun penelitian ini masih menggunakan *platform website* dan belum menggunakan *platform android* sebagai sarana informasi untuk masyarakat desa. Disamping itu sistem ini juga belum menyediakan fitur untuk mempromosikan produk UMKM.

Penelitian dengan judul “Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis *Website* Kelurahan Banaran” dibuat untuk memberikan kemudahan

kepada masyarakat dalam pengajuan pelayanan dan pengelolaan administrasi oleh petugas, penelitian ini memaksimalkan dalam penambahan fitur utama dari hasil penelitian sebelumnya, seperti memberikan pemberitahuan secara langsung yang dikirimkan ke alamat email pemohon dengan melampirkan file surat yang sudah berhasil terverifikasi petugas dan melakukan rekap surat secara otomatis dengan berbentuk laporan dan arsip surat. Pengembangan sistem menggunakan metode *waterfall* yang meliputi perencanaan, analisis, perancangan, implementasi, dan pemeliharaan sistem (Al Hasri dan Sudarmilah, 2021). Namun secara keseluruhan sistem informasi pelayanan administrasi kependudukan masih belum sempurna sehingga dapat dikembangkan lagi. Perbaikan sistem dapat dilakukan dengan mengembangkan sistem android *webview* agar memberikan kenyamanan terhadap pengguna dalam mengoperasikan dengan perangkat android secara langsung.

Tabel 2.1. Tabel Perbandingan Sistem

No	Sistem	Metode/Fitur/Teknologi					
		1	2	3	4	5	6
1	Sistem Informasi Administrasi Kependudukan Berbasis Web Pada Desa Bogangin Sumpiuh (Imaniawan dan Wati, 2017)	√	-	-	-	√	-
2	Pemanfaatan <i>E-Government</i> Pada Desa Wonokarto Untuk Meningkatkan Akurasi Dan Informasi Potensi Desa (Rachman dan Noviyanto, 2017)	√	-	-	-	√	-
3	<i>E-government</i> Pada Pekon Srimulyo Kalirejo Kabupaten Lampung Tengah (Muntofiah dan Fitrian, 2017)	√	-	-	-	√	-
4	Aplikasi Sistem Informasi Desa Sebagai Teknologi Tepat Guna untuk Pendataan Penduduk dan Potensi Desa (Syaharuddin 2017).	√	-	-	-	√	√

5	Pengembangan <i>Website</i> dan Sistem Informasi Desa Di Kabupaten Tulungagung (Rozi dan Listiawan, 2017)	√	-	-	-	√	-
6	Implementasi Web <i>Mobile</i> Sebagai Media Informasi Pemberdayaan Masyarakat Di Desa Pirngadi (Usmanto et al. 2018)	√	-	-	-	√	√
7	Implementasi <i>E-Goverment</i> Sebagai Upaya Peningkatan Potensi Desa Di Desa Bumirejo Menggunakan Web <i>Mobile</i> (Pratiwi dan Muslihudin, 2018)	√	-	-	-	√	√
8	Implementasi Web <i>Government</i> Dalam Meningkatkan Potensi Produk Unggulan Desa Berbasis Android (Samsudin dan Muslihudin, 2018)	√	-	-	-	√	-
9	Sistem Informasi Administrasi Desa di Kantor Desa Sumbersekar (Susilowati dan Chandra, 2018)	√	-	-	-	√	-
10	Implementasi Pelayanan Administrasi Kependudukan Desa Kalimanah Wetan Purbalingga Berbasis <i>Online</i> (Setyowati, Anton dan Radiyah, 2019)	√	-	-	-	√	-
11	Inovasi Pelayanan Administrasi Kependudukan Secara <i>Online</i> di Kabupaten Pati (Soraya 2019)	√	-	-	-	√	-
12	Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala) (Jimi 2019)	√	-	-	-	√	-
13	Aplikasi Pelayanan Administrasi Desa Karang Dima Berbasis Web (Putra, Hidayatullah dan Juniarta, 2019)	√	-	-	-	√	-
14	Rancang Bangun Sistem Administrasi Desa Berbasis Android di Kantor Kepala Desa	√	-	-	-	-	√

	Keramat Jaya Kabupaten Majalengka (Nurmalasari dan Sujadi, 2019)					
15	Pengembangan Sistem Informasi Pelayanan Desa Berbasis Web Dengan Metode <i>Prototyping</i> Pada Desa Leran (Kurniawan, Chabibi dan Dewi, 2020)	√	-	-	-	√ -
16	Sistem Informasi Pelayanan Masyarakat Berbasis Android Pada Kantor Desa Kaliba Mamase (Ritnawati, Suppa dan Muhallim, 2020)	√	-	-	-	√
17	Sistem Informasi Pengolahan Data Kependudukan Pada Kantor Desa Sampean Berbasis Android (Alda 2020)	√	-	-	-	√
18	Rancangan Bangunan Sistem Informasi Administrasi desa Berbasis Web menggunakan PHP dan MySQL (Winanjar dan Susanti, 2021)	√	-	-	-	√ -
19	Rancang Bangun Sistem Informasi Manajemen Kependudukan Desa Adat Terintegrasi Berbasis Web (Purnawan 2021)	√	-	-	-	√ -
20	Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis <i>Website</i> Kelurahan Banaran (Al Hasri dan Sudarmilah, 2021)	√	-	-	-	√ -
21	Digitalisasi Pelayanan Administrasi Dan Ketahanan Ekonomi Masyarakat Desa Dengan Teknik <i>Crowdsourcing</i>	√	√	√	√	√ √

Keterangan:

- 1) Layanan Administrasi
- 2) Layanan Berita
- 3) Pengaduan dan Aspirasi
- 4) *Marketplace*
- 5) *Website* Desa

6) Aplikasi Android

Berdasarkan Tabel 2.1 dapat dilihat penelitian dengan judul “Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa Dengan Teknik *Crowdsourcing*” memiliki kesamaan fitur dengan semua penelitian yang sudah dilakukan sebelumnya yaitu fitur layanan administrasi dan informasi (Imaniawan dan Wati, 2017), (Rachman dan Noviyanto, 2017), (Muntofiah dan Fitrian, 2017), (Syaharuddin 2017), (Rozi dan Listiawan, 2017), (Usmanto et al. 2018), (Pratiwi dan Muslihudin, 2018), (Samsudin dan Muslihudin, 2018), (Susilowati dan Chandra, 2018), (Setyowati, Anton dan Radiyah, 2019), (Soraya 2019), (Jimi 2019), (Putra, Hidayatullah dan Juniarta, 2019), (Nurmalasari dan Sujadi, 2019), (Kurniawan, Chabibi dan Dewi, 2020), (Ritnawati, Suppa dan Muallim, 2020), (Alda 2020), (Winanjar dan Susanti, 2021), (Purnawan 2021), (Al Hasri dan Sudarmilah, 2021), penerapan website desa (Imaniawan dan Wati, 2017), (Rachman dan Noviyanto, 2017), (Muntofiah dan Fitrian, 2017), (Syaharuddin 2017), (Rozi dan Listiawan, 2017), (Usmanto et al. 2018), (Pratiwi dan Muslihudin, 2018), (Samsudin dan Muslihudin, 2018), (Susilowati dan Chandra, 2018), (Setyowati, Anton dan Radiyah, 2019), (Soraya 2019), (Jimi 2019), (Putra, Hidayatullah dan Juniarta, 2019), (Kurniawan, Chabibi dan Dewi, 2020), (Winanjar dan Susanti, 2021), (Purnawan 2021), (Al Hasri dan Sudarmilah, 2021), menggunakan *platform* android (Syaharuddin 2017), (Usmanto et al. 2018), (Pratiwi dan Muslihudin, 2018), (Nurmalasari dan Sujadi, 2019), (Ritnawati, Suppa dan Muallim, 2020), (Alda 2020). Sistem ini telah menggabungkan beberapa fitur tersebut dan menambahkan inovasi fitur menampilkan produk UMKM desa yang dihubungkan aplikasi *marketplace* dengan memanfaatkan teknik *crowdsourcing*. Selain itu, fitur pengaduan dan aspirasi kepada pemerintah desa akan ditambahkan dalam aplikasi yang akan dibuat. Beberapa kelebihan lainnya yang dimiliki aplikasi digitalisasi desa ini yaitu menyediakan fitur untuk manajemen kependudukan, mengolah dana desa, surat menyurat dan berita desa.

2.2. Teknologi yang Digunakan

Bagian ini membahas tentang teknologi yang digunakan dalam mengembangkan digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*.

2.2.1. Crowdsourcing

Crowdsourcing merupakan suatu metode/teknik untuk mendapatkan suatu kebutuhan dengan memanfaatkan kontribusi dari sekumpulan orang yang berasal dari komunitas *online*. Pada sistem ini, teknik *Crowdsourcing* digunakan untuk mengumpulkan *user* ketika ada pelaku UMKM yang mempromosikan produknya dan pengajuan aspirasi masyarakat.

2.2.2 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi.

2.2.2. Visual Studio Code

Visual Studio Code adalah editor *source code* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi. Visual Studio Code gratis dan *open-source*, meskipun unduhan resmi berada di bawah lisensi proprietary. Kode Visual Studio didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk desktop yang berjalan pada Blinklayout. Meskipun menggunakan kerangka Elektron, Visual Studio Code tidak menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode

nama "Monaco") yang digunakan dalam Visual Studio Team Services yang sebelumnya disebut Visual Studio Online. Dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing* Visual Studio Code digunakan sebagai editor penulisan *source code* dengan menggunakan bahasa pemrograman PHP dan Dart.



Gambar 2.1. Logo Visual Studio Code

2.2.3. MySQL

MySQL adalah RDBMS *open source* dan *multithreaded* yang dibuat oleh Michael “Monty” Widenius pada 1995. Pada tahun 2000, MySQL dirilis dengan lisensi ganda yang mengijinkan publik untuk menggunakannya secara gratis di bawah lisensi GNU GPL (General Public License) yang menyebabkan popularitasnya melambung. Banyaknya fitur MySQL membuat *database* ini tetap menjadi sistem basis data yang hebat. Kecepatan adalah salah satu fiturnya yang menonjol. Dalam perbandingan oleh eWeek pada beberapa basis data (MySQL, Oracle, MS SQL, IBM DB2, dan Sybase ASE), MySQL dan Oracle menunjukkan performa dan skalabilitas terbaik. MySQL mampu menangani puluhan ribu tabel dan miliaran baris data dengan cepat dan lancar (Russell J.T. Dyer 2008). Dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing* MySQL digunakan sebagai tempat menyimpan data.



Gambar 2.2. Logo MySQL

2.2.4. Firebase

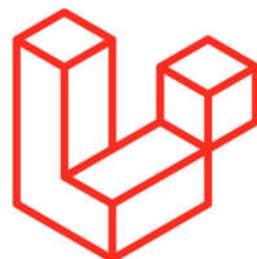
Firebase adalah salah satu layanan yang dikenal bagian dari perusahaan Google untuk memudahkan para pengembang perangkat lunak dalam mengembangkan aplikasi mereka. Pengembangan aplikasi khususnya untuk aplikasi *mobile*. Firebase didirikan oleh Andrew Lee 10 dan James Tamplin dengan nama perusahaan Envolve pada tahun 2011. Firebase Realtime Database adalah salah satu produk yang pertama kali mereka kembangkan. Pengembangan aplikasi Android menggunakan Firebase sangat membantu para developers untuk membuat aplikasi yang berkualitas tinggi dan dapat menumbuhkan basis pengguna yang interaktif. Berbagai fitur canggih dapat diterapkan pada aplikasi yang dibuat, seperti *backend system*, analisis, serta fitur pertumbuhan dan monetisasi. Menggunakan Firebase dapat menghemat waktu karena memungkinkan untuk mengelola sedikit integrasi dalam aplikasi. Hal tersebut dapat terjadi karena Firebase memiliki API intuitif dikemas dalam satu paket SDK. Dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*, Firebase digunakan sebagai penambahan fitur notifikasi pada aplikasi android.



Gambar 2.3. Logo *Firebase*

2.2.5. Laravel

Laravel membuat lebih hemat biaya dan waktu. *Framework* ini terhitung lebih hemat biaya dibandingkan *framework* lainnya. Laravel juga mudah dikembangkan, karena aturan-aturan yang ada dalam Laravel, maka memudahkan pengembang lain yang akan mengembangkan *website* tersebut. Laravel juga memiliki dokumentasi yang lengkap. Dokumen Laravel bisa diakses di id-laravel.com, learninglaravel.net, laracast.com dan Laravel-news.com. *User Friendly* merupakan kelebihan Laravel. Laravel dapat digunakan dengan mudah. Sehingga *framework* ini dikatakan *user friendly*. Selain itu, keamanan Laravel terjamin. Laravel memberi jaminan akan keamanan dari *framework* ini. Selain itu, ada juga komunitas Laravel. Penyelesaian masalah berkaitan dengan penggunaan Laravel terbilang banyak dan mudah ditemukan. Laravel mensupport metode MVC. MVC merupakan metode yang paling sering digunakan dalam penggunaan *website* saat ini. Laravel menggunakan metode MVC, sehingga memudahkan untuk pengembangan *website*. Dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*, Laravel digunakan sebagai *backend*.



Gambar 2.4. Logo Laravel

2.2.6. Flutter

Flutter adalah Software Development Kit (SDK) buatan Google yang berfungsi untuk membuat aplikasi *mobile* menggunakan bahasa pemrograman Dart, baik untuk Android maupun iOS. Dengan Flutter, aplikasi Android dan iOS dapat dibuat menggunakan basis kode dan bahasa pemrograman yang sama yaitu Dart, bahasa pemrograman yang juga diproduksi oleh Google pada tahun 2011. Dart adalah sebuah bahasa pemrograman yang dikembangkan oleh Google, dirancang oleh Lars Bak dan Kasper Lund. Dart pertama kali dikenalkan pada 10 Oktober 2011. Versi 1.0 dari bahasa pemrograman ini baru dirilis pada bulan November 2013. Versi stabil terbaru adalah Dart 2.1, yang dirilis pada tanggal 15 November 2018. Dart dapat digunakan untuk membuat aplikasi server (berbentuk *commandline interface*), web, maupun *mobile* (Android dan iOS). Dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*, Flutter digunakan sebagai *framework* pembuatan aplikasi android..



Gambar 2.5. Logo Flutter

2.2.7. Vue Js

Vue.js adalah suatu *library* Javascript yang digunakan untuk membangun antar muka sebuah *website* yang interaktif. Vue difokuskan hanya pada *view layer* sehingga sangat mudah diimplementasikan dan diintegrasikan dengan *library* lain ataupun juga dengan project yang sudah ada sebelumnya. Dalam hal lainnya Vue js juga dapat membantu programmer untuk membuat web dengan aplikasi *single page* yang canggih dan dapat di kombinasikan dengan *library browser* lainnya.

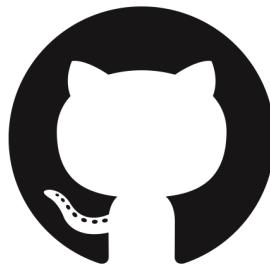
Selain itu Vue js menggunakan template sintaks berbasis HTML yang memungkinkan pengguna untuk mendeklarasikan data / state kedalam DOM. Semua template Vue js adalah HTML yang valid yang dapat diuraikan oleh *browser* sesuai spesifikasi dan parser HTML. Dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*, Vue js digunakan sebagai *front-end*.



Gambar 2.6. Logo Vue Js

2.2.8. Github

Github adalah sebuah *storehouse online* yang digunakan untuk melakukan proyek secara kolaborasi. Github juga dikenal sebagai salah satu *storehouse* yang terbesar di dunia. Github sendiri berupa aplikasi yang bekerja berbasis *website* yang dilengkapi oleh VCS (Version Control System). Fungsi utama Github adalah untuk memudahkan manajemen proyek. Dengan Github akan bisa bekerja dan berkomunikasi dengan banyak orang dari beragam tempat, melacak progress proyek, melacak *bug*, membuat rencana proyek, manajemen tugas, dan lainnya. Selain itu, Github memberikan paket layanan penyimpanan *repository* secara gratis dan pribadi. Inilah mengapa Github banyak dipakai dalam proses pengembangan *software* serta untuk kode komputer. Di tahun 2018, Microsoft telah menjadi pihak yang mengakuisisi Github. Dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*, Github digunakan sebagai manajemen *project* aplikasi website dan android.



Gambar 2.8. Logo Github

2.2.9. Postman

Postman adalah *platform* kolaborasi untuk pengembangan API. Dibuat oleh Abhinav Asthana, seorang programmer dan desainer yang berbasis di Bangalore, India. Postman memudahkan dalam menguji, mengembangkan, dan mendokumentasikan API. Fitur Postman yang sederhana membuat pengujian API dapat dilakukan dengan baik dan cepat. Cara kerja Postman dengan mengklasifikasi *request* berdasarkan *request* method, URL dan parameter-parameter *request*. Dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*, Postman digunakan sebagai penguji API



Gambar 2.9. Logo Postman

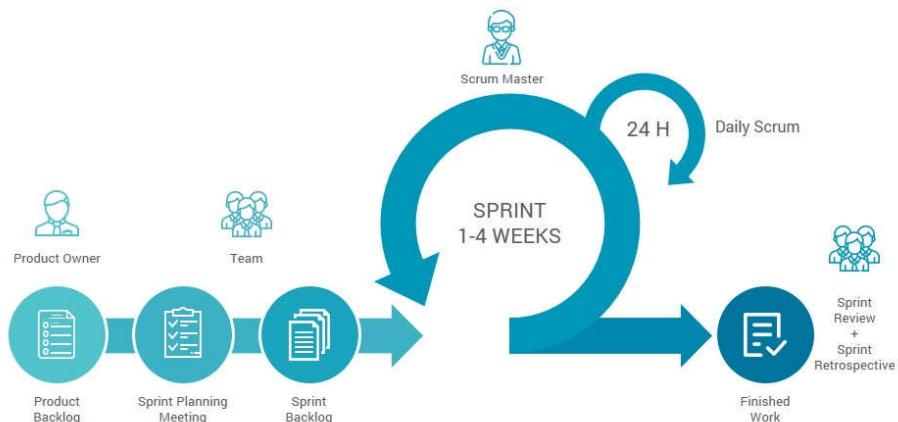
2.2.10. FCM

Firebase Cloud Messaging (FCM) adalah *platform* untuk membuat sebuah aplikasi *mobile* atau aplikasi web dengan peralatan dan infrastruktur yang memungkinkan pengembangan lebih mudah dan menghasilkan aplikasi berkualitas tinggi. FCM ini adalah salah satu fitur yang disediakan oleh Google. Awalnya Firebase ini

digunakan sebagai *database* yang *real-time* yang memberi sebuah API untuk menyinkronkan dan menyimpan data di berbagai perusahaan. Salah satu fitur yang terdapat pada Firebase adalah Firebase Cloud Messaging (FCM), fitur ini adalah platform yang digunakan untuk pesan dan notifikasi untuk android, ios, dan aplikasi web tanpa biaya.

BAB III KEGIATAN PELAKSANAAN

Bab ini membahas tentang pelaksanaan Tugas Akhir yang dikembangkan menggunakan metode *Agile*. Menurut Sommerville (2011) metode *Agile* merupakan metode pengembangan *incremental* yang fokus pada perkembangan yang cepat, perangkat lunak yang dirilis bertahap, mengurangi *overhead* proses, dan menghasilkan kode berkualitas tinggi dan pada proses perkembangannya melibatkan pelanggan secara langsung. Karena *Agile* sebuah sifat, maka perlu kerangka kerja untuk mewujudkan sifat-sifat yang terdapat pada metode *Agile* menjadi langkah-langkah yaitu menggunakan model *Scrum*. *Scrum* adalah salah satu metode rekayasa perangkat lunak dengan menggunakan prinsip-prinsip pendekatan *Agile*, yang bertumpu pada kekuatan kolaborasi tim, *incremental product* dan proses iterasi untuk mewujudkan hasil akhir.



Gambar 3.1. Cara kerja *Scrum* menggunakan metode *Agile*

Gambar 3.1 merupakan tahapan-tahapan yang dilakukan dalam pengembangan sistem menggunakan metode *Scrum*. Tahapan-tahapannya antara lain: *product backlog*, *sprint planning meeting*, *sprint backlog*, *sprint execution*, *finished work*.

1) *Product Backlog*

Tahapan ini pengembang sistem mendapatkan daftar fitur yang akan dikembangkan sesuai kebutuhan dan dikelompokkan dalam *sprint* dengan cara

melakukan wawancara. Kebutuhan - kebutuhan ini didapatkan dari hasil wawancara atau diskusi dengan aparat desa dengan jabatan di kantor desa Sinungrejo sebagai Kepala Seksi (Kasi) yang bernama Sutadi dengan yang beralamat di Desa Sinungrejo, RT 01/03, Kecamatan Ambal, Kabupaten Kebumen.

2) *Sprint Planning Meeting*

Tahapan ini pengembang sistem melakukan pembahasan mengenai pembuatan *sprint* pada proyek yang akan dikerjakan.

3) *Sprint Backlog*

Pada *sprint backlog*, pengembang sistem melakukan pembuatan *task* yang dikelompokkan berdasarkan *sprint*. *Task* disini merupakan pemecahan permasalahan ke dalam bentuk yang lebih sederhana.

4) *Sprint Execution*

Tahapan ini, pengembang sistem melakukan pelaksanaan pengembangan fitur yang telah ditentukan dalam *sprint* yang terdapat *task* pada bagian *sprint backlog*.

5) *Finished Work*

Tahapan ini merupakan bagian dari *sprint retrospective* dan *sprint review* atau pembahasan tentang apa saja yang perlu ditingkatkan terkait proses kerja.

3.1. Product Backlog

Tahapan ini menentukan kebutuhan yang diperlukan dalam pengembangan sistem Tugas Akhir yang meliputi:

- 1) Analisis kebutuhan sistem
- 2) Perancangan diagram sistem
- 3) Perancangan antarmuka sistem
- 4) Pembuatan dan pengujian fitur aplikasi
- 5) Perancangan uji kepuasan pengguna

3.2. Sprint Planning Meeting

Bagian ini menentukan estimasi waktu dalam penggerjaan setiap *sprint* atau kebutuhan sehingga dapat menjadi acuan dalam proses *Scrum*. Daftar *sprint* ditunjukkan pada Tabel 3.1.

Tabel 3.1. Daftar *Sprint* Kebutuhan

No	Sprint	Product Backlog	Prioritas	Estimasi Waktu (Jam)	Waktu yang digunakan (jam)
1.	1	Analisis kebutuhan sistem	Sangat Tinggi	2	2
2.	2	Perancangan diagram sistem	Sangat Tinggi	24	22
3.	3	Perancangan antarmuka sistem	Sangat Tinggi	96	88
4.	4	Pembuatan dan pengujian fitur aplikasi	Sangat Tinggi	602	558
5.	5	Perancangan uji kepuasan pengguna	Tinggi	8	5
TOTAL				732	675

Berdasarkan Tabel 3.1 didapatkan jumlah *sprint* sebanyak lima *sprint* dengan mempertimbangkan prioritas serta estimasi waktu yang digunakan dalam penggerjaan *sprint*. Estimasi waktu yang digunakan untuk penggerjaan lima *sprint* adalah empat minggu lebih dua hari dan waktu penggerjaan yang digunakan sebanyak empat minggu untuk menyelesaikan *sprint* yang sudah direncanakan.

3.3. Sprint Backlog

Pada tahapan ini ditentukan *task* pada setiap *sprint* yang dibuat pada sebelumnya.

3.3.1. Sprint 1

Pada *sprint* ini mengerjakan kebutuhan pada bagian *product backlog* yaitu analisis tentang kebutuhan sistem, *task* yang dikerjakan terdapat pada Tabel 3.2.

Tabel 3.2. *Sprint 1: Analisis Kebutuhan Sistem*

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Analisis kebutuhan perangkat	Analisis kebutuhan perangkat keras	1	1
	Analisis kebutuhan perangkat lunak	1	1
TOTAL		2	2

Dari Tabel 3.2 didapatkan jumlah *task*, estimasi waktu penggerjaan, dan waktu penggerjaan sebenarnya dalam mengerjakan *task – task* pada *backlog* analisis kebutuhan perangkat.

3.3.2. *Sprint 2*

Pada *sprint* ini mengerjakan kebutuhan pada bagian *product backlog* yaitu perancangan diagram sistem, *task* yang dikerjakan terdapat pada Tabel 3.3.

Tabel 3.3. *Sprint 2: Analisis Perancangan Diagram Sistem*

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Analisis perancangan diagram sistem	Analisis <i>block diagram</i>	2	2
	Analisis <i>use case diagram</i>	8	7
	Analisis <i>activity diagram</i>	8	7
	Analisis <i>entity relationship diagram</i>	6	6
TOTAL		24	22

Tabel 3.3 menjelaskan tentang jumlah *task*, estimasi waktu pekerjaan, dan waktu penggerjaan sebenarnya dalam mengerjakan *task – task* pada *backlog* analisis perancangan diagram sistem.

3.3.3. Sprint 3

Pada *sprint* ini mengerjakan kebutuhan pada bagian *product backlog* yaitu perancangan antarmuka sistem yang terdiri dari aplikasi *website* administrator dan pengguna. Pada *sprint* ini terdiri dari fitur-fitur yang terdapat dalam sistem, *task* yang dikerjakan terdapat pada Tabel 3.4.

Tabel 3.4. *Sprint* 3: Perancangan Antarmuka Sistem

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Perancangan antarmuka sistem	Antarmuka Aplikasi Android		
	Halaman <i>Splash Screen</i>	1	1
	Halaman <i>Login</i> dan Verifikasi	2	2
	Halaman Beranda	2	2
	Halaman Surat	2	2
	Halaman Detail Surat	1	1
	Halaman Aspirasi	2	2
	Halaman Berita	2	2
	Halaman Profil	2	2
	Halaman Ubah Profil	1	1
	Halaman Ubah Nomor WhatsApp	1	1
	Halaman UMKM	2	2
	Halaman Kelola UMKM	2	2
	Halaman Kelola Produk UMKM	2	2
	Halaman Produk UMKM	2	2
	Halaman Potensi Desa	2	2
	Halaman Tentang Desa	1	2
	Halaman Pemberitahuan	1	2
Antarmuka Website			
Halaman <i>Landing Page</i>	3	2	

	Halaman Login	2	2
	Halaman Dashboard	3	3
	Halaman Penduduk	2	2
	Halaman Tambah Penduduk	2	2
	Halaman Detail Penduduk	2	1,5
	Halaman Edit Penduduk	2	1,5
	Halaman Inventaris	2	2
	Halaman Tambah Inventaris	2	2
	Halaman Detail Inventaris	2	1,5
	Halaman Edit Inventaris	2	1,5
	Halaman Keuangan	2	2
	Halaman Tambah Keuangan	2	2
	Halaman Detail Keuangan	2	1,5
	Halaman Edit Keuangan	2	1,5
	Halaman Berita	2	2
	Halaman Tambah Berita	2	2
	Halaman Detail Berita	2	1,5
	Halaman Edit Berita	2	1,5
	Halaman Agenda	2	2
	Halaman Tambah Agenda	2	2
	Halaman Detail Agenda	2	1,5
	Halaman Edit Agenda	2	1,5
	Halaman Aspirasi	2	2
	Halaman Detail Aspirasi	2	1,5
	Halaman Surat Masuk	2	2
	Halaman Detail Surat Masuk	2	1,5
	Halaman UMKM	2	2
	Halaman Detail UMKM	2	1,5
	Halaman Produk UMKM	2	2
	Halaman Detail Produk UMKM	2	1,5

	Halaman Keahlian	2	2
	Halaman Detail Keahlian	2	1,5
	Halaman Profil Desa	2	2
	TOTAL	96	88

3.3.4. Sprint 4

Sprint ini mengerjakan kebutuhan pada bagian *product backlog* Pembuatan dan pengujian fitur aplikasi, *task* yang dikerjakan terdapat pada Tabel 3.5.

Tabel 3.5. Sprint 5: Pembuatan dan Pengujian Fitur Aplikasi

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Pembuatan dan pengujian fitur aplikasi	Fitur Aplikasi Android		
	Pembuatan dan Pengujian Fitur Halaman <i>Splash Screen</i>	5	4
	Pembuatan dan Pengujian Fitur Halaman <i>Login</i> dan Verifikasi	8	7
	Pembuatan dan Pengujian Fitur Halaman Beranda	9	8
	Pembuatan dan Pengujian Fitur Halaman Surat	8	7
	Pembuatan dan Pengujian Fitur Halaman Detail Surat	8	7
	Pembuatan dan Pengujian Fitur Halaman Aspirasi	9	7
	Pembuatan dan Pengujian Fitur Halaman Berita	8	7
	Pembuatan dan Pengujian Fitur Halaman Profil	8	7
	Pembuatan dan Pengujian Fitur Halaman Ubah Profil	8	7

	Pembuatan dan Pengujian Fitur Halaman Ubah Nomor WhatsApp	8	7
	Pembuatan dan Pengujian Fitur Halaman UMKM	8	7
	Pembuatan dan Pengujian Fitur Halaman Kelola UMKM	8	7
	Pembuatan dan Pengujian Fitur Halaman Kelola Produk UMKM	8	7
	Pembuatan dan Pengujian Fitur Halaman Produk UMKM	8	7
	Pembuatan dan Pengujian Fitur Halaman Potensi Desa	8	7
	Pembuatan dan Pengujian Fitur Halaman Tentang Desa	5	4
	Pembuatan dan Pengujian Fitur Halaman Pemberitahuan	7	5
Antarmuka Website			
	Pembuatan dan Pengujian Fitur Halaman <i>Landing Page</i>	10	8
	Pembuatan dan Pengujian Fitur Halaman <i>Login</i>	10	9
	Pembuatan dan Pengujian Fitur Halaman <i>Dashboard</i>	15	13
	Pembuatan dan Pengujian Fitur Halaman Penduduk	15	12
	Pembuatan dan Pengujian Fitur Halaman Tambah Penduduk	10	9
	Pembuatan dan Pengujian Fitur Halaman Detail Penduduk	10	9

	Pembuatan dan Pengujian Fitur Halaman Edit Penduduk	10	9
	Pembuatan dan Pengujian Fitur Hapus Data Penduduk	8	7
	Pembuatan dan Pengujian Fitur Halaman Inventaris	10	9
	Pembuatan dan Pengujian Fitur Halaman Tambah Inventaris	10	8
	Pembuatan dan Pengujian Fitur Halaman Detail Inventaris	10	9
	Pembuatan dan Pengujian Fitur Halaman Edit Inventaris	10	12
	Pembuatan dan Pengujian Fitur Hapus Data Inventaris	8	7
	Pembuatan dan Pengujian Fitur Halaman Keuangan	10	9
	Pembuatan dan Pengujian Fitur Halaman Tambah Keuangan	10	8
	Pembuatan dan Pengujian Fitur Halaman Detail Keuangan	10	9
	Pembuatan dan Pengujian Fitur Halaman Edit Keuangan	10	9
	Pembuatan dan Pengujian Fitur Hapus Data Keuangan	8	7
	Pembuatan dan Pengujian Fitur Halaman Berita	10	9
	Pembuatan dan Pengujian Fitur Halaman Tambah Berita	10	8
	Pembuatan dan Pengujian Fitur Halaman Detail Berita	10	9
	Pembuatan dan Pengujian Fitur Halaman Edit Berita	10	9

Pembuatan dan Pengujian Fitur Hapus Data Berita	8	7
Pembuatan dan Pengujian Fitur Halaman Agenda	10	9
Pembuatan dan Pengujian Fitur Halaman Tambah Agenda	10	9
Pembuatan dan Pengujian Fitur Halaman Detail Agenda	10	9
Pembuatan dan Pengujian Fitur Halaman Edit Agenda	10	8
Pembuatan dan Pengujian Fitur Hapus Data Agenda	8	7
Pembuatan dan Pengujian Fitur Halaman Aspirasi	10	9
Pembuatan dan Pengujian Fitur Halaman Detail Aspirasi	10	9
Pembuatan dan Pengujian Fitur Halaman Surat Masuk	10	9
Pembuatan dan Pengujian Fitur Halaman Detail Surat Masuk	10	9
Pembuatan dan Pengujian Fitur Halaman UMKM	10	9
Pembuatan dan Pengujian Fitur Halaman Tambah UMKM	10	9
Pembuatan dan Pengujian Fitur Halaman Detail UMKM	10	9
Pembuatan dan Pengujian Fitur Halaman Edit UMKM	10	9
Pembuatan dan Pengujian Fitur Hapus Data UMKM	8	7
Pembuatan dan Pengujian Fitur Halaman Produk UMKM	10	9

	Pembuatan dan Pengujian Fitur Halaman Tambah Produk UMKM	10	9
	Pembuatan dan Pengujian Fitur Halaman Detail Produk UMKM	10	9
	Pembuatan dan Pengujian Fitur Halaman Edit Produk UMKM	10	9
	Pembuatan dan Pengujian Fitur Hapus Data Produk UMKM	8	7
	Pembuatan dan Pengujian Fitur Halaman Keahlian	10	9
	Pembuatan dan Pengujian Fitur Halaman Tambah Keahlian	10	9
	Pembuatan dan Pengujian Fitur Halaman Detail Keahlian	10	9
	Pembuatan dan Pengujian Fitur Halaman Edit Keahlian	10	9
	Pembuatan dan Pengujian Fitur Hapus Data Keahlian	8	7
	Pembuatan dan Pengujian Fitur Halaman Profil Desa	8	7
TOTAL		610	568

3.3.5. Sprint 5

Pada *sprint* ini mengerjakan kebutuhan pada bagian *product backlog* yaitu Perancangan uji kepuasan pengguna, *task* yang dikerjakan terdapat pada Tabel 3.6.

Tabel 3.6. Perancangan Uji Kepuasan Pengguna

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)

Perancangan uji kepuasan pengguna	Pembuatan Pertanyaan	4	3
	Pembuatan Indikator Penilaian	2	1
	Pembuatan Indikator Kepuasan	2	1
	Pengguna		
TOTAL		8	5

3.4. Sprint Execution

Tahapan ini melakukan penggerjaan *task-task* yang telah ditentukan pada bagian *sprint backlog*.

3.4.1. Sprint 1: Analisis Kebutuhan Sistem

Sprint ini membahas mengenai analisis kebutuhan perangkat berupa kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*) yang diperlukan dalam pembuatan tugas akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*.

1) Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk membangun digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing* adalah sebagai berikut:

- a. Laptop
- b. Mouse

2) Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membangun digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing* adalah sebagai berikut:

- a. Windows 10
- b. Visual Studio Code
- c. Chrome Browser
- d. XAMPP
- e. Framework Laravel
- f. Flutter
- g. Postman

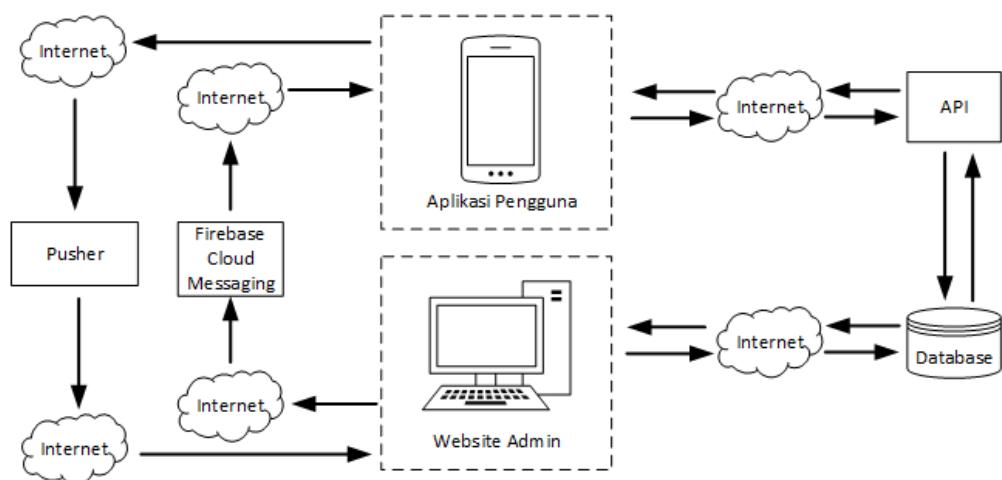
- h. MySQL
- i. Firebase

3.4.2. Sprint 2: Perancangan Diagram Sistem

Pada *sprint* ini merupakan gambaran dari sebuah sistem yang dirancang menggunakan diagram yang terdiri dari *block diagram*, *usecase diagram*, *ERD* (*Entity Relationship Diagram*).

1) Block Diagram

Block diagram digunakan sebagai gambaran umum tentang sistem Tugas Akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing* ini bekerja dengan blok-blok yang saling terhubung seperti Gambar 3.2.



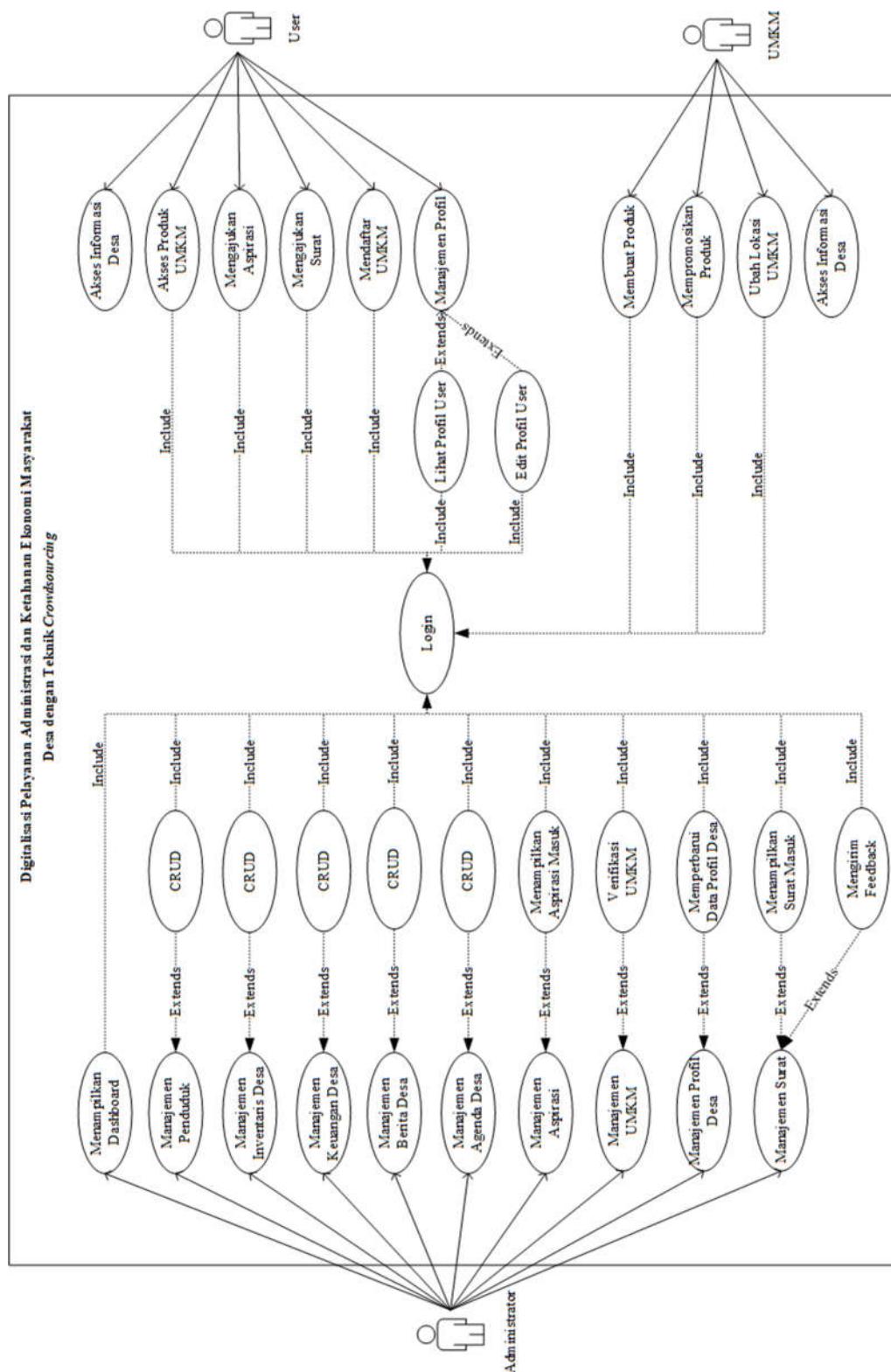
Gambar 3.2. *Block Diagram*

Pada Gambar 3.2 dapat diketahui bahwa aplikasi pengguna berinteraksi dengan *website admin* melalui *API* dan *database*. *Database* berfungsi sebagai tempat penyimpanan semua data yang ada pada sistem. *Database* akan berinteraksi dengan *API* yang berada di internet agar bisa menampilkan data yang diminta aplikasi pengguna atau *website admin*. *API* menggunakan *protocol* yang disediakan oleh *Firebase Cloud Messaging* agar bisa menampilkan notifikasi pada aplikasi pengguna. Sedangkan untuk menampilkan notifikasi pada *website admin*, menggunakan *protocol* yang disediakan oleh *Pusher*.

2) *Use Case Diagram*

Use Case Diagram merupakan sebuah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antara “aktor” dengan sistem yang ada. *Use Case Diagram* digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing* akan dipaparkan pada Gambar 3.3.

Pada Gambar 3.3 terdapat 3 aktor yang terdapat dalam sistem yaitu administrator, *user* penduduk dan pelaku UMKM. Setiap aktor memiliki tugas masing-masing. Administrator dapat menampilkan dashboard, manajemen penduduk, manajemen inventaris desa, manajemen keuangan desa, manajemen berita desa, manajemen agenda desa, manajemen surat, manajemen UMKM dan manajemen aspirasi. Penduduk sebagai *user* dapat melakukan akses informasi desa, akses produk UMKM, mengajukan aspirasi, mengajukan surat dan dapat melakukan manajemen profil berupa mengubah nomor WhatsApp *user*. Sedangkan pelaku UMKM dapat membuat produk, mempromosikan produk, mengubah lokasi UMKM, serta mengakses informasi desa. Admin wajib melakukan *log in* dengan *email* dan *password*, sedangkan *user* dan UMKM melakukan *log in* dengan menggunakan nomor telepon dan *password* yang telah ditentukan sebelum melakukan proses – proses tersebut.



Gambar 3.3. Use Case Diagram Sistem

3) Activity Diagram

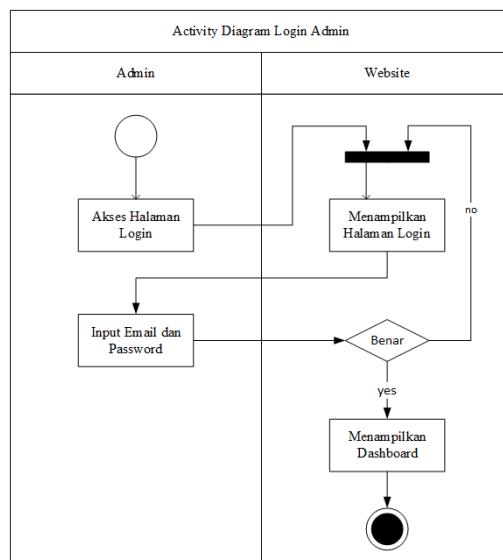
Activity Diagram merupakan sebuah diagram yang digunakan untuk menggambarkan alur dari aktifitas-aktifitas yang ada dalam sistem ini. Berikut merupakan diagram yang terdapat pada sistem ini berdasarkan administrator, *user* dan UMKM.

a) Administrator

Activity Diagram administrator terdiri dari alur yang ditampilkan berdasarkan pada gambar berikut.

1) Activity Diagram Login

Activity Diagram login ditampilkan pada Gambar 3.4.

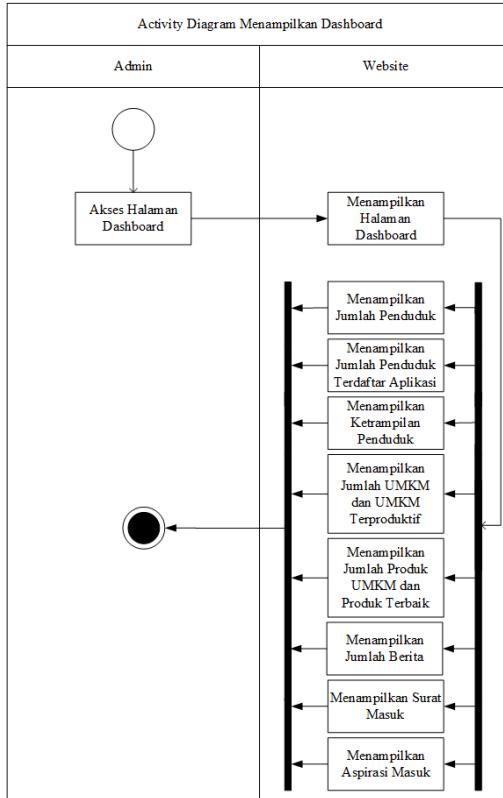


Gambar 3.4. *Activity Diagram Login* Administrator

Berdasarkan Gambar 3.4 administrator dapat melakukan *login* dengan mengakses halaman *login* terlebih dahulu. Setelah itu administrator memasukkan *email* dan *password* yang dimilikinya. Jika proses *login* telah berhasil, sistem akan menampilkan *dashboard* awal administrator. Namun jika proses *login* gagal, administrator akan tetap berada di halaman *login*.

2) Activity Diagram Menampilkan Dashboard

Activity Diagram manampilkan *dashboard* ditampilkan pada Gambar 3.5.



Gambar 3.5. *Activity Diagram* Menampilkan Dashboard

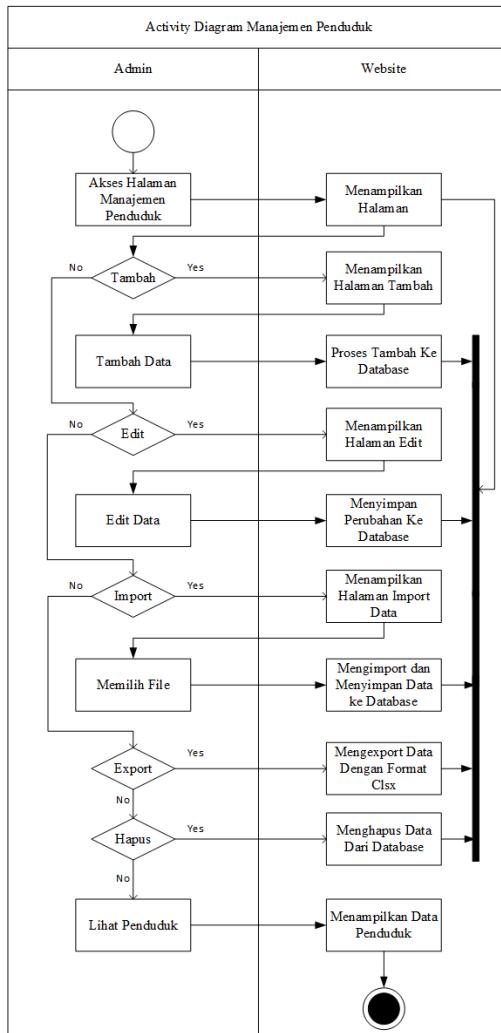
Berdasarkan Gambar 3.5 administrator dapat melihat kalkulasi jumlah penduduk, jumlah penduduk yang terdaftar pada aplikasi *user*, ketramplian penduduk, jumlah UMKM dan UMKM terproduktif, jumlah produk UMKM dan produk terbaik, jumlah berita, surat masuk dan aspirasi masuk. Kalkulasi pada *dashboard* akan bertambah seiring adanya penambahan penduduk, UMKM, berita, surat dan aspirasi.

3) *Activity Diagram* Manajemen Penduduk

Activity Diagram manajemen penduduk ditampilkan pada Gambar 3.6.

Berdasarkan Gambar 3.6 ketika administrator melakukan manajemen penduduk dengan mengakses halaman penduduk terlebih dahulu. Setelah ditampilkan halaman penduduk, administrator dapat menambahkan, mengedit, mengimport, mengexport dan menghapus data penduduk yang akan disimpan

oleh *database*. Jika administrator tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman penduduk.



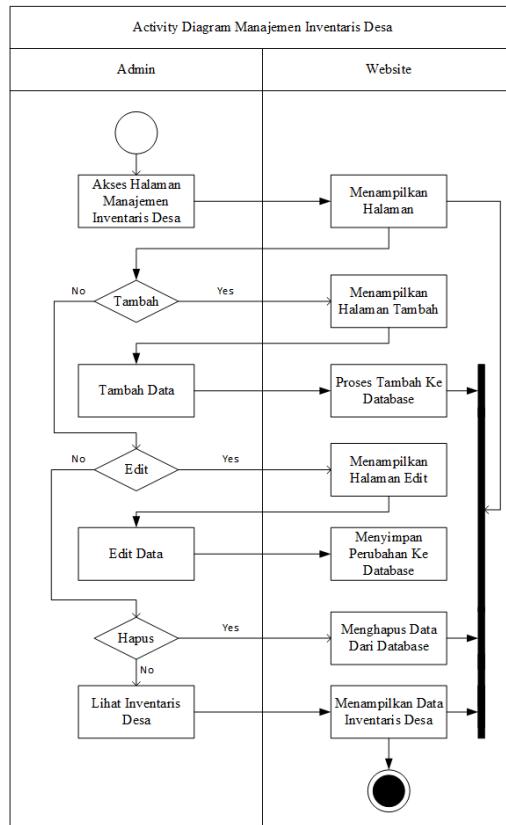
Gambar 3.6. *Activity Diagram* Manajemen Penduduk

4) *Activity Diagram* Manajemen Inventaris Desa

Activity Diagram manajemen inventaris desa ditampilkan pada Gambar 3.7.

Berdasarkan Gambar 3.7 administrator melakukan manajemen inventaris desa dengan mengakses halaman inventaris desa terlebih dahulu. Setelah ditampilkan halaman inventaris desa, administrator dapat menambahkan, mengedit, dan menghapus inventaris desa yang akan disimpan oleh *database*.

Jika administrator tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman manajemen inventaris desa.

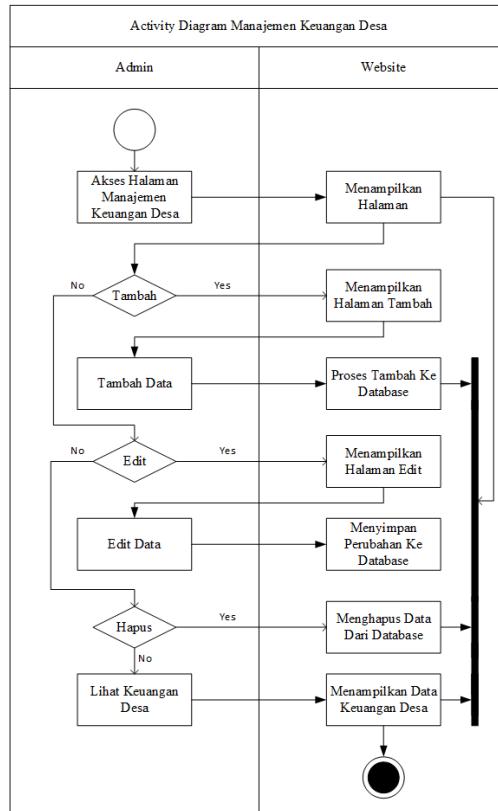


Gambar 3.7. *Activity Diagram* Manajemen Inventaris Desa

5) *Activity Diagram* Manajemen Keuangan Desa

Activity Diagram manajemen keuangan desa ditampilkan pada Gambar 3.8.

Berdasarkan Gambar 3.8 administrator melakukan manajemen keuangan desa dengan mengakses halaman keuangan desa terlebih dahulu. Setelah ditampilkan halaman keuangan desa, administrator dapat menambahkan, mengedit, dan menghapus keuangan desa yang akan disimpan oleh *database*. Jika administrator tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman manajemen keuangan desa

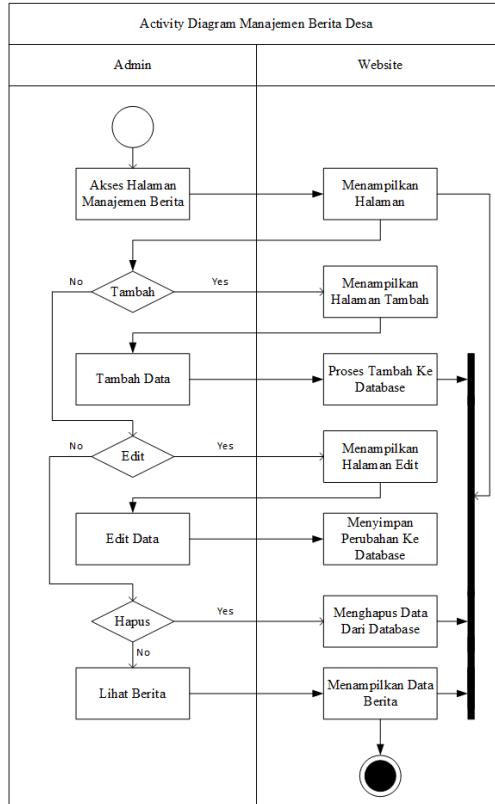


Gambar 3.8. *Activity Diagram* Manajemen Keuangan Desa

6) *Activity Diagram* Manajemen Berita

Activity Diagram manajemen berita ditampilkan pada Gambar 3.9.

Berdasarkan Gambar 3.9 administrator melakukan manajemen berita dengan mengakses halaman berita terlebih dahulu. Setelah ditampilkan halaman berita, administrator dapat menambahkan, mengedit, dan menghapus berita yang akan disimpan oleh *database*. Jika administrator tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman manajemen berita.

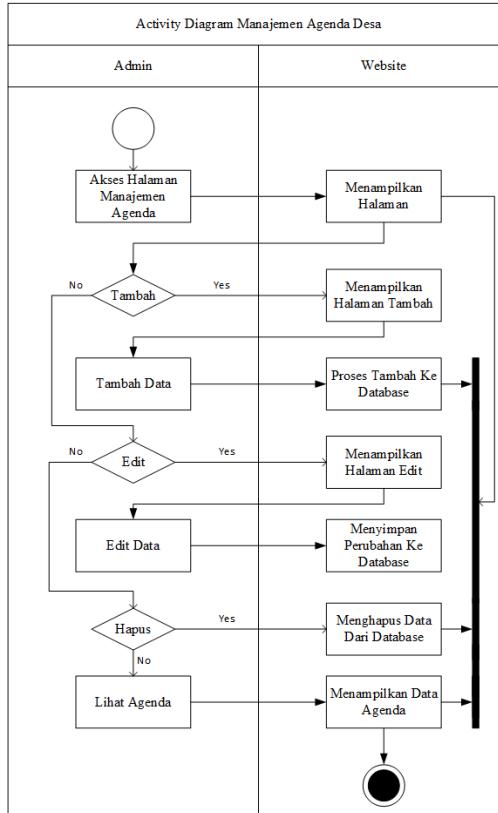


Gambar 3.9. *Activity Diagram Manajemen Berita*

7) *Activity Diagram Manajemen Agenda*

Activity Diagram manajemen agenda ditampilkan pada Gambar 3.10.

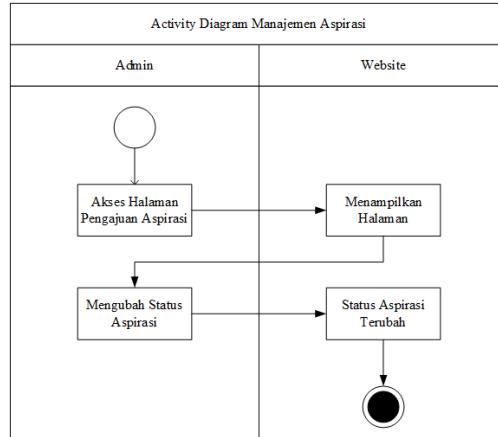
Berdasarkan Gambar 3.10 administrator melakukan manajemen agenda dengan mengakses halaman agenda terlebih dahulu. Setelah ditampilkan halaman agenda, administrator dapat menambahkan, mengedit, dan menghapus berita yang akan disimpan oleh *database*. Jika administrator tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman manajemen agenda.



Gambar 3.10. *Activity Diagram* Manajemen Agenda

8) *Activity Diagram* Manajemen Aspirasi

Activity Diagram Manajemen Aspirasi ditampilkan pada Gambar 3.11.

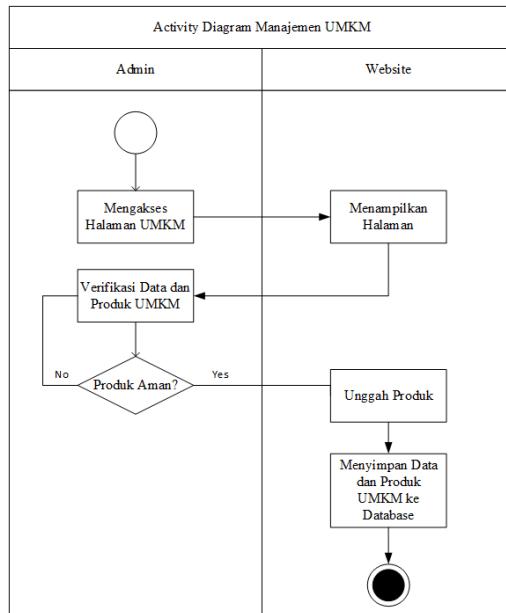


Gambar 3.11. *Activity Diagram* Manajemen Aspirasi

Berdasarkan Gambar 3.11 administrator melakukan manajemen aspirasi dengan mengakses halaman pengajuan aspirasi yang telah diajukan *user*. Setelah ditampilkan halaman aspirasi, administrator dapat mengubah status aspirasi. Apabila aspirasi sudah diterima dan terbaca, maka status pengajuan aspirasi akan terupdate dalam *website* dan aplikasi *user*.

9) *Activity Diagram* Manajemen UMKM

Activity Diagram Manajemen UMKM ditampilkan pada Gambar 3.12

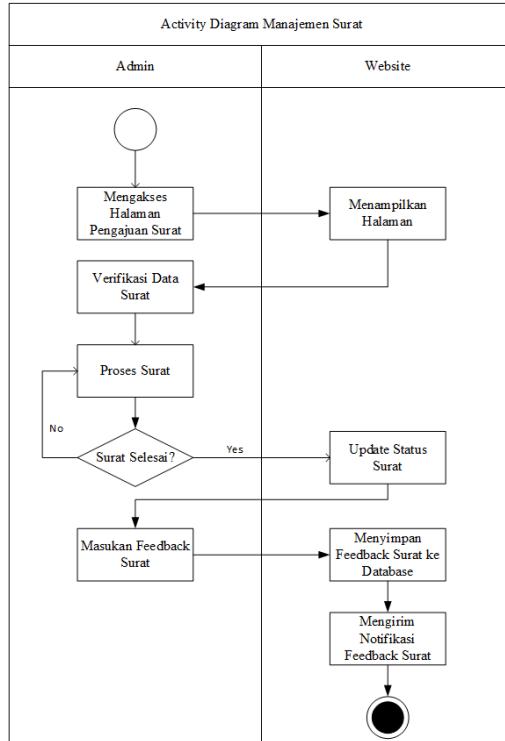


Gambar 3.12. *Activity Diagram* Manajemen UMKM

Berdasarkan Gambar 3.12 administrator melakukan manajemen UMKM dengan mengakses halaman notifikasi UMKM yang telah diajukan pelaku UMKM. Setelah ditampilkan halaman UMKM, administrator dapat memverifikasi data dan produk UMKM. Apabila produk aman, maka data dan produk UMKM akan diunggah serta akan tersimpan di *database*.

10) *Activity Diagram* Manajemen Surat

Activity Diagram manajemen surat ditampilkan pada Gambar 3.13.

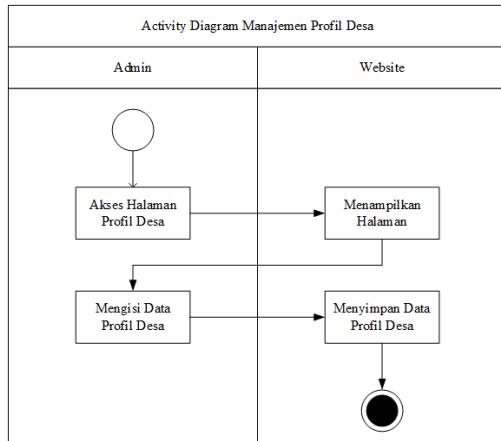


Gambar 3.13. *Activity Diagram* Manajemen Surat

Berdasarkan Gambar 3.13 administrator melakukan manajemen surat dengan mengakses halaman pengajuan surat yang telah diajukan *user*. Setelah ditampilkan halaman surat, administrator dapat memverifikasi surat dan memprosesnya. Apabila surat sudah selesai, maka status surat akan terupdate dalam *website* dengan memasukan *feedback* surat. Kemudian *feedback* surat akan tersimpan ke *database*, dan *website* akan mengirim notifikasi *feedback* surat.

11) *Activity Diagram* Manajemen Profil Desa

Activity Diagram manajemen profil desa ditampilkan pada Gambar 3.14.



Gambar 3.14. *Activity Diagram* Manajemen Profil Desa

Berdasarkan Gambar 3.14 administrator melakukan manajemen profil desa dengan mengakses halaman profil. Setelah ditampilkan halaman profil, administrator dapat mengisi data profil desa seperti menambah logo desa, nama desa, *tagline*, alamat dan letak desa berdasarkan latitude dan longitude.

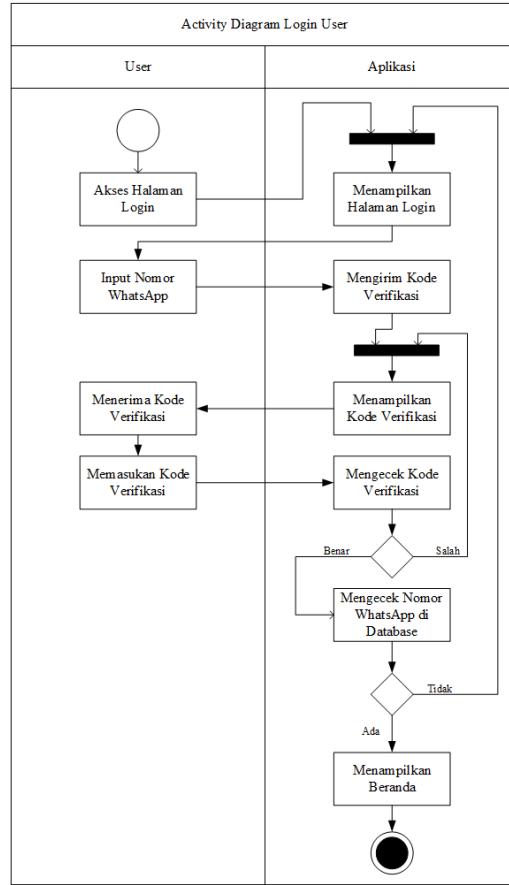
b) *User*

Activity Diagram user terdiri dari alur yang ditampilkan berdasarkan pada gambar berikut.

1) *Activity Diagram Login* dan Verifikasi

Activity Diagram login dan verifikasi *user* ditampilkan pada Gambar 3.15.

Berdasarkan Gambar 3.15 *user* dapat melakukan *login* dengan mengakses halaman *login* terlebih dahulu. Setelah itu *user* memasukkan nomor WhatsApp kemudian *user* mendapatkan nomor verifikasi melalui WhatsApp. *User* dapat memasukkan kode verifikasi yang didapatkan pada halaman verifikasi. Apabila *user* belum terdaftar, maka *user* dapat memasukkan data diri pada halaman registrasi terlebih dahulu sebelum ke halaman beranda. Sedangkan jika verifikasi gagal pengguna akan kembali ke halaman *login*.

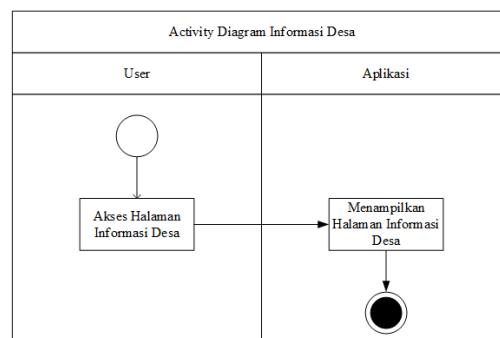


Gambar 3.15. *Activity Diagram* Login dan Verifikasi User

2) *Activity Diagram* Akses Informasi Desa

Activity Diagram akses informasi desa ditampilkan pada Gambar 3.16.

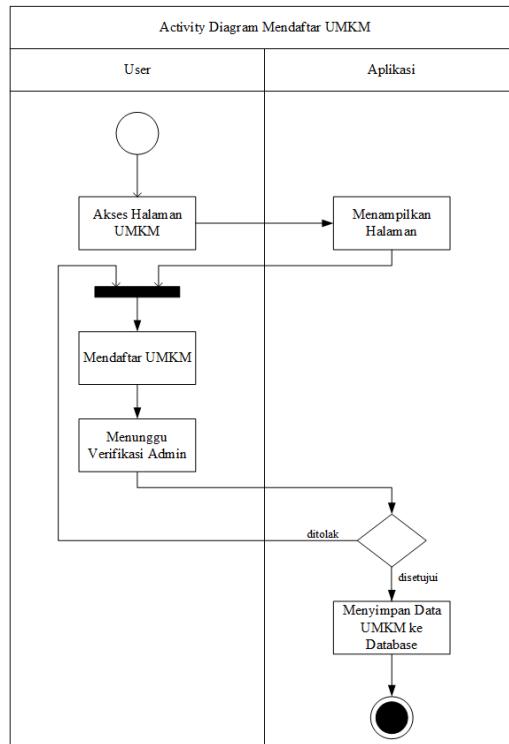
Berdasarkan Gambar 3.16 *user* dapat mengakses halaman informasi desa untuk melihat informasi yang terdapat didalamnya.



Gambar 3.16. *Activity Diagram* Akses Infromasi Desa

3) *Activity Diagram* Mendaftar UMKM

Activity Diagram mendaftar UMKM ditampilkan pada Gambar 3.17.

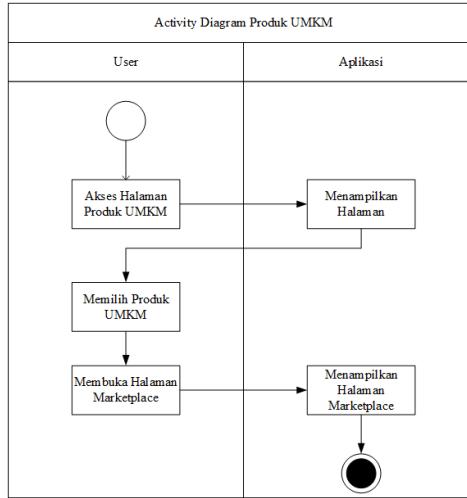


Gambar 3.17. *Activity Diagram* Mendaftar UMKM

Berdasarkan Gambar 3.17 *user* dapat mendaftar UMKM dengan mengakses halaman UMKM. Setelah itu *user* mengisi data UMKM kemudian data tersebut akan diverifikasi oleh administrator. Apabila data UMKM valid dan disetujuai administrator maka data tersebut akan tersimpan di *database*, jika data tidak valid maka data tidak tersimpan di *database*.

4) *Activity Diagram* Akses Produk UMKM

Activity Diagram mendaftar UMKM ditampilkan pada Gambar 3.18.

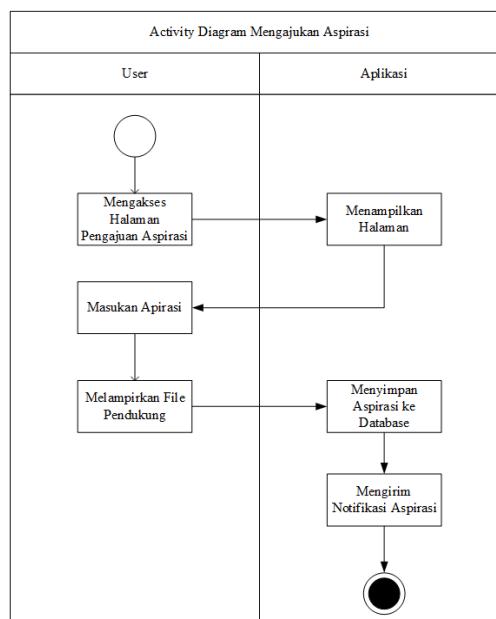


Gambar 3.18. *Activity Diagram* Akses Produk UMKM

Berdasarkan Gambar 3.18 *user* dapat mengakses halaman produk UMKM dengan menampilkan halaman UMKM. Kemudian *user* dapat memilih produk UMKM dan dihubungkan dengan aplikasi *marketplace*.

5) *Activity Diagram* Mengajukan Aspirasi

Activity Diagram mengajukan aspirasi ditampilkan pada Gambar 3.19.

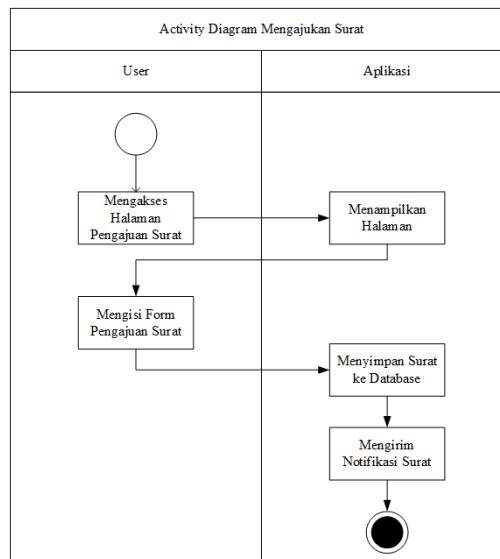


Gambar 3.19. *Activity Diagram* Mengajukan Aspirasi

Berdasarkan Gambar 3.19 *user* dapat mengajukan aspirasi desa dengan mengakses halaman pengajuan aspirasi. Kemudian *user* dapat mengisi aspirasi dengan melampirkan file pendukung. Setelah itu aspirasi akan tersimpan di database dan akan mengirimkan notifikasi ke administrator.

6) *Activity Diagram* Mengajukan Surat

Activity Diagram mengajukan surat ditampilkan pada Gambar 3.20.



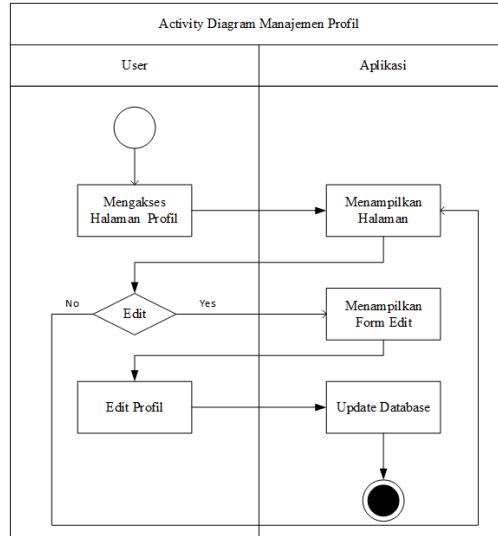
Gambar 3.20. *Activity Diagram* Mengajukan Surat

Berdasarkan Gambar 3.20 *user* dapat mengajukan surat dengan mengakses halaman pengajuan surat. Kemudian *user* dapat mengisi form pengajuan surat. Setelah itu surat yang diajukan akan tersimpan di *database* dan kemudian mengirimkan notifikasi ke administrator.

7) *Activity Diagram* Manajemen Profil

Activity Diagram manajemen profil ditampilkan pada Gambar 3.21.

Berdasarkan Gambar 3.21 *user* dapat melakukan manajemen profil dengan mengakses halaman profil terlebih dahulu. Setelah ditampilkan halaman profil apabila *user* akan mengubah informasi profil maka akan disimpan ke *database*. Jika tidak maka akan ditampilkan kembali halaman profil.



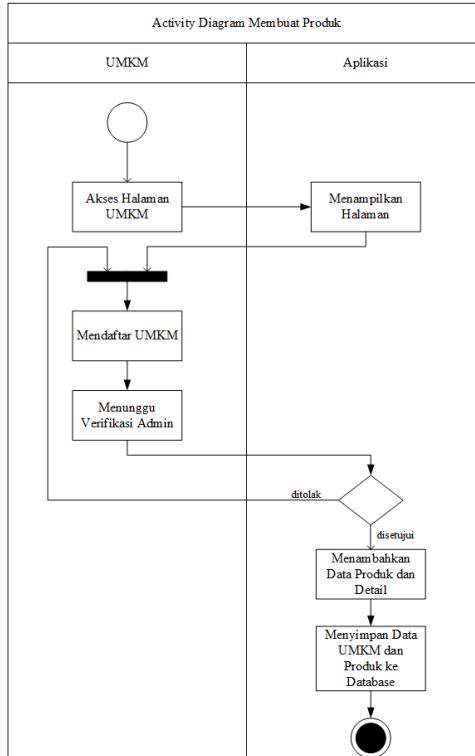
Gambar 3.21. *Activity Diagram* Manajemen Profil

c. UMKM

Berikut beberapa diagram yang terdapat pada bagian UMKM.

1) *Activity Diagram* Membuat Produk

Activity Diagram membuat produk ditampilkan pada Gambar 3.22.

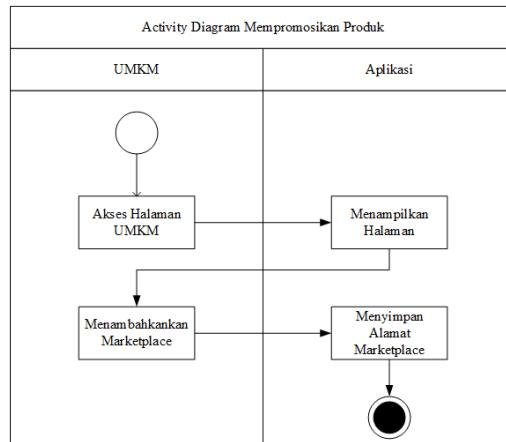


Gambar 3.22. *Activity Diagram* Membuat Produk

Berdasarkan Gambar 3.22 UMKM dapat membuat produk dengan mengakses halaman UMKM terlebih dahulu. Setelah itu UMKM dapat mendaftar UMKM dan menunggu verifikasi dari administrator. Apabila data UMKM valid dan distujui maka UMKM dapat menambahkan data produk dan detail dan kemudian data tersebut akan tersimpan di *database*.

2) *Activity Diagram* Mempromosikan Produk

Activity Diagram mempromosikan produk ditampilkan pada Gambar 3.23.



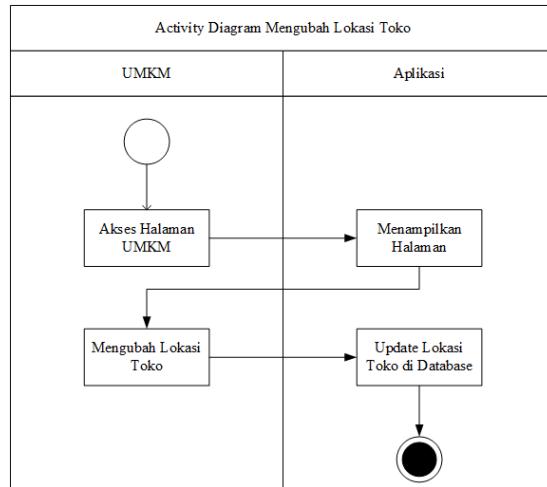
Gambar 3.23. *Activity Diagram* Mempromosikan Produk

Berdasarkan Gambar 3.23 UMKM dapat mempromosikan produk dengan mengakses halaman UMKM, setelah itu UMKM dapat menambahkan alamat *marketplace* yang digunakan untuk mempromosikan produknya.

3) *Activity Diagram* Mengubah Lokasi UMKM

Activity Diagram mengubah lokasi UMKM ditampilkan pada Gambar 3.24.

Berdasarkan Gambar 3.24 UMKM dapat mengubah lokasi UMKM dengan cara mengakses halaman UMKM. Selanjutnya UMKM dapat mengubah lokasi dan kemudian data lokasi UMKM tersebut akan diperbarui di *database*.

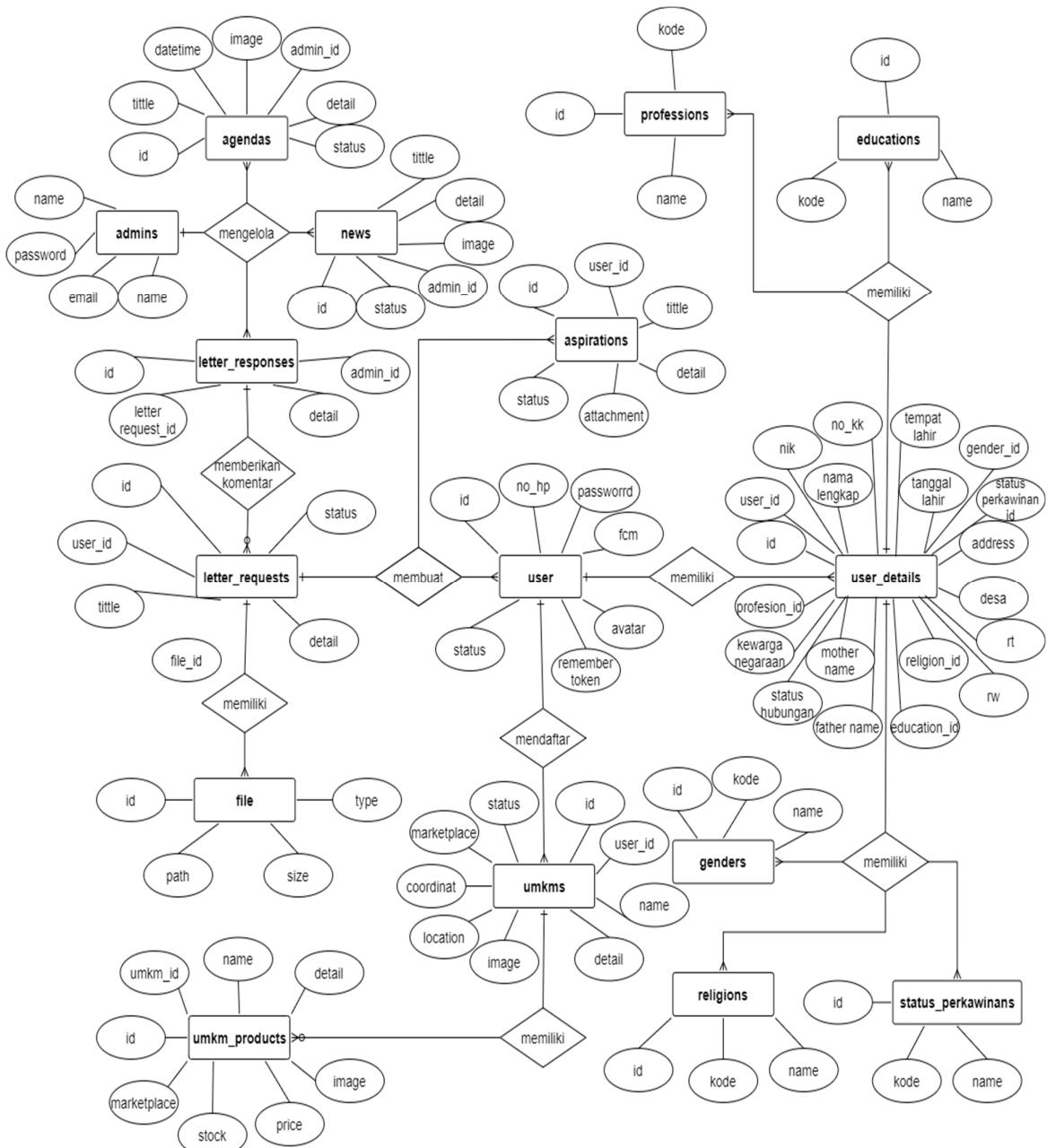


Gambar 3.24. *Activity Diagram* Mengubah Lokasi Toko

4) Entity Relationalship Diagram (ERD)

ERD (*Entity Relationship Diagram*) ditampilkan pada Gambar 3.25.

Dalam Gambar 3.25 ERD aplikasi ini terdapat enam belas entitas yaitu admins, aspirations, users, news, professions, agendas, letter_request, users, letter_responses, feedback, genders, status_perkawinans, educations, umkm dan umkm_products. Entitas users memiliki relasi *one-to-many* terhadap entitas users_details, aspirations, dan letter_request. Entitas admins memiliki relasi *one-to-many* terhadap entitas news, agendas dan letter_respons. Entitas letter_request memiliki relasi *one-to-one* terhadap entitas letter_respons. Entitas aspirations memiliki relasi *one-to-one* terhadap entitas feedback. Entitas genders, status_perkawinans, religions, professions dan education memiliki relasi *one-to-one* terhadap entitas users_details. Entitas umkm memiliki relasi *one-to-one* terhadap entitas umkm_products.



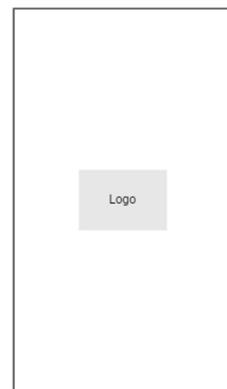
Gambar 3.25. Entity Relationship Diagram

3.4.3. Sprint 3: Perancangan Antarmuka Sistem

Pada bagian *sprint* ini akan dibahas mengenai perancangan tampilan *User Interface* (UI) dari Tugas Akhir digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*.

1) Aplikasi User

a) Halaman *Splash Screen*

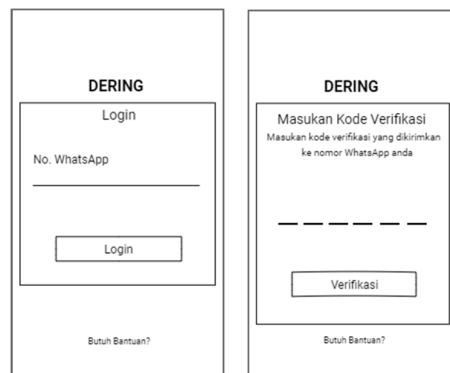


Gambar 3. 26 Tampilan Halaman *Splash Screen*

Gambar 3.26 merupakan tampilan dari halaman *splash screen*, tampilan ini akan muncul pada saat aplikasi dijalankan pertama kali oleh *user*.

b) Halaman *Login* dan Verifikasi

Gambar 3.27 merupakan halaman *login* yang digunakan *user* untuk masuk ke aplikasi. Untuk bisa *login* *user* harus memasukkan akun yang sudah terdaftar sebelumnya dengan memasukkan nomor WhatsApp.



Gambar 3. 27 Tampilan Halaman *Login* dan Verifikasi

c) Halaman *Dashboard*

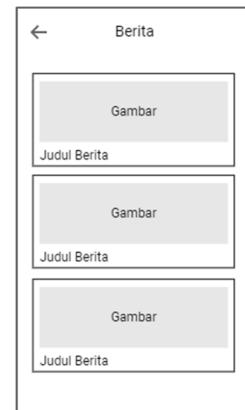
Gambar 3.29 menunjukkan tampilan *dashboard* dari aplikasi. Pada halaman ini akan ditampilkan menu untuk mengakses layanan yang diberikan administrator seperti administrasi, pengaduan dan informasi desa.



Gambar 3. 29 Tampilan Halaman *Dashboard*

d) Halaman Berita

Gambar 3.30 merupakan halaman berita yang digunakan *user* untuk mengakses berita.



Gambar 3. 30 Tampilan Halaman Berita

e) Halaman Surat

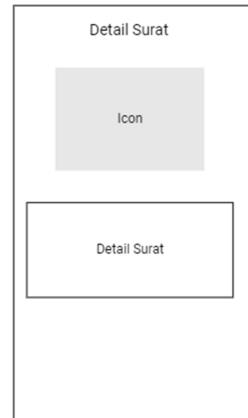
Gambar 3.31 merupakan halaman surat yang digunakan *user* untuk mengajukan administrasi surat. Pada halaman ini menampilkan semua surat dengan kategori draft surat dan surat di proses.



Gambar 3. 31 Tampilan Halaman Surat

f) Halaman Detail Surat

Gambar 3.32 menunjukkan tampilan halaman halaman detail surat. Pada halaman ini akan ditampilkan keterangan dan detail surat yang diajukan oleh *user* kepada administrator.



Gambar 3. 32 Tampilan Halaman Detail Surat

g) Halaman Profil

Gambar 3.33 menunjukkan tampilan dari halaman profil *user*. Halaman ini berisikan menu-menu yang dapat diakses oleh *user*. Dalam halaman ini pengguna juga dapat mengubah informasi pribadi, ubah nomor WhatsApp,

ubah *password*, mengelola UMKM, mengisi keahlian, melihat informasi desa serta menu *logout*.



Gambar 3. 33 Tampilan Halaman Profil

h) Halaman Ubah Nomor WhatsApp

Gambar 3.35 menunjukkan tampilan dari halaman ubah nomor WhatsApp. Dalam halaman ini *user* dapat mengubah nomor WhatsApp yang baru dengan menginputkan kode konfirmasi yang dikirim melalui nomor WhatsApp *user* yang baru.



Gambar 3. 35 Tampilan Halaman Ubah Nomor WhatsApp

i) Halaman UMKM

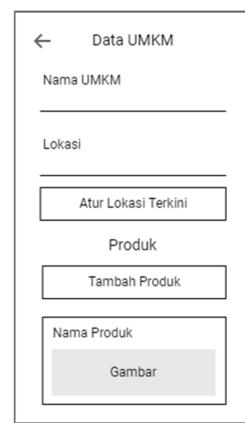
Gambar 3.36 menunjukkan tampilan dari halaman UMKM. Dalam halaman menampilkan menu yang memungkinkan *user* untuk mengelola UMKM dan melihat produk UMKM yang ada di desa.



Gambar 3. 36 Tampilan Halaman UMKM

j) Halaman Kelola UMKM

Gambar 3.37 menunjukkan tampilan dari halaman kelola UMKM. Dalam halaman ini *user* dapat membuat UMKM dengan mengisi nama UMKM serta lokasi UMKM. Selain itu, *user* juga dapat menambah produk yang akan dipromosikan.



Gambar 3. 37 Tampilan Halaman Kelola UMKM

k) Halaman Kelola Produk UMKM

Gambar 3.38 menunjukkan tampilan dari halaman kelola produk UMKM. Dalam halaman ini *user* dapat menambahkan data produk UMKM dengan mengisi nama produk, kategori, stok, harga, dan satuan kemudian menyimpannya.

The form is titled 'Data Produk' with a back arrow icon. It contains five input fields: 'Nama Produk', 'Kategori', 'Stok', 'Harga', and 'Satuan'. Below the fields is a 'Simpan Produk' button.

Gambar 3. 38 Tampilan Halaman Kelola Produk UMKM

l) Halaman Produk UMKM

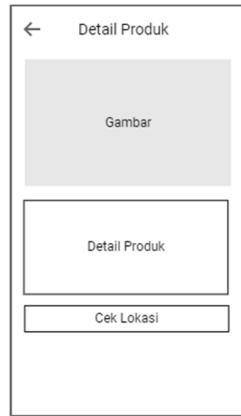
Gambar 3.39 menunjukkan tampilan dari halaman produk UMKM. Dalam halaman ini menunjukkan produk UMKM yang dikelola oleh masyarakat desa.

The form is titled 'Produk UMKM' with a back arrow icon. It features two buttons: 'Kategori 1' and 'Kategori 2'. Below them is a list of three products, each with a 'Nama Produk' field and a 'Gambar' placeholder. The first product's 'Nama Produk' field is highlighted with a grey background.

Gambar 3. 39 Tampilan Halaman Produk UMKM

m) Halaman Detail Produk

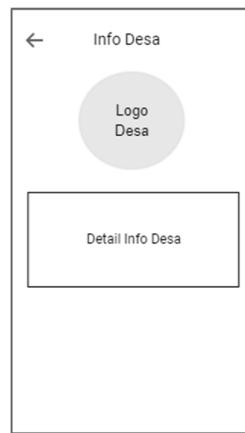
Gambar 3.40 menunjukkan tampilan dari halaman detail produk. Dalam halaman ini menunjukkan detail produk UMKM yang dikelola oleh masyarakat desa beserta lokasi UMKM.



Gambar 3. 40 Tampilan Halaman Detail Produk

n) Halaman Info Desa

Gambar 3.41 menunjukkan tampilan dari halaman informasi mengenai desa. Dalam halaman ini menampilkan logo desa, detail mengenai desa, peraturan desa, visi misi desa dan struktur kepengurusan desa.



Gambar 3. 41 Tampilan Halaman Info Desa

2) Website

a) Halaman *Login*

Gambar 3.42 menunjukkan tampilan halaman *login* yang digunakan untuk melakukan proses manajemen data dari *user* aplikasi. Apabila data yang dimasukkan sesuai maka akan diteruskan akses menuju halaman selanjutnya. Dalam *login* ini hanya dilakukan oleh administrator dengan menginputkan *email* dan *password*.

The screenshot shows a login interface for the 'DERING' application. At the top center is the word 'DERING'. Below it is a rectangular form containing fields for 'Email' and 'Kata Sandi' (Password), each with its own input box. At the bottom of the form is a 'Login' button.

Gambar 3. 42 Tampilan Halaman *Login* Administrator

b) Halaman *Dashboard* Administrator



Gambar 3. 43 Tampilan Halaman *Dashboard* Administrator

Gambar 3.43 menunjukkan tampilan halaman *dashboard*. Pada halaman ini akan ditampilkan kalkulasi dari jumlah penduduk, jumlah penduduk laki-laki, jumlah penduduk perempuan, jumlah *online user*, jumlah UMKM, jumlah produk UMKM, jumlah ketrampilan dan jumlah berita. Halaman ini juga menampilkan statistic keahlian penduduk desa dan daftar penduduk paling

aspiratif. Pada halaman ini juga terdapat statistic UMKM terproduktif per bulan, daftar produk UMKM terbaik, surat masuk terbaru dan aspirasi penduduk yang masuk.

c) Halaman Manajemen Penduduk

No. KK	NIK	Nama	No. HP	TTL	Aksi
					<button>Edit</button> <button>Detail</button>

Gambar 3. 44 Tampilan Halaman Manajemen Penduduk

Gambar 3.44 menunjukkan tampilan halaman manajemen penduduk. Pada halaman ini berisi informasi data penduduk meliputi no KK, NIK, nama, nomor handphone dan tempat tanggal lahir. Dalam halaman ini administrator dapat melakukan penambahan data penduduk, *import* dan *export* data penduduk, menghapus data penduduk, mengedit data penduduk dan melihat data penduduk secara detail.

d) Halaman Manajemen Inventaris

Tipe	Nama	Ukuran	Jumlah	Status	Aksi
					<button>Edit</button> <button>Detail</button> <button>Hapus</button>

Gambar 3. 45 Tampilan Halaman Manajemen Inventaris

Gambar 3.45 menunjukkan tampilan halaman manajemen inventaris desa. Pada halaman ini berisi informasi data inventaris meliputi tipe, nama, ukuran, jumlah dan status. Dalam halaman ini administrator juga dapat melakukan penambahan data inventaris desa, mengedit data inventaris, melihat data inventaris secara detail, dan menghapus data inventaris.

e) Halaman Manajemen Keuangan

Gambar 3.46 menunjukkan tampilan halaman manajemen keuangan desa. Pada halaman ini berisi informasi data keuangan meliputi tanggal, pemasukan, pengeluaran, detail dan status. Dalam halaman ini administrator juga dapat melakukan penambahan data keuangan desa, mengedit data keuangan, melihat data keuangan secara detail dan menghapus data.

Tanggal	Pemasukan	Pengeluaran	Detail	Status	Aksi
					<input type="button" value="Edit"/> <input type="button" value="Detail"/> <input type="button" value="Hapus"/>

Gambar 3. 46 Tampilan Halaman Manajemen Keuangan

f) Halaman Manajemen Berita

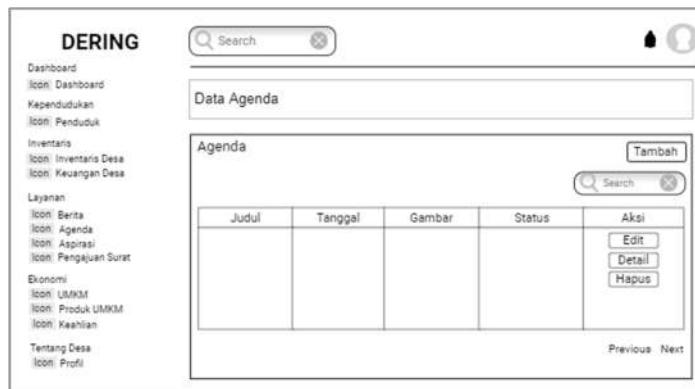
Judul	Gambar	Status	Dilihat	Author	Aksi
					<input type="button" value="Edit"/> <input type="button" value="Detail"/> <input type="button" value="Hapus"/>

Gambar 3. 47 Tampilan Halaman Manajemen Berita

Gambar 3.47 menunjukkan tampilan halaman manajemen berita. Pada halaman ini berisi informasi berita meliputi judul, gambar, status, dilihat dan *author*. Dalam halaman ini administrator juga dapat melakukan penambahan berita baru, mengedit berita, melihat detail berita dan menghapus berita.

g) Halaman Manajemen Agenda

Gambar 3.48 menunjukkan tampilan halaman manajemen agenda. Pada halaman ini berisi informasi agenda meliputi judul, tanggal, gambar, dan status. Dalam halaman ini administrator juga dapat melakukan penambahan agenda, mengedit agenda, melihat detail agenda dan menghapus agenda.



Gambar 3. 48 Tampilan Halaman Manajemen Agenda

h) Halaman Manajemen Aspirasi



Gambar 3. 49 Tampilan Halaman Manajemen Aspirasi

Gambar 3.49 menunjukkan tampilan halaman manajemen aspirasi. Pada halaman ini berisi informasi aspirasi meliputi nama pengirim aspirasi, judul, detail dan status yang dikirimkan oleh *user*.

i) Halaman Manajemen Surat

Gambar 3.50 menunjukkan tampilan halaman manajemen surat. Pada halaman ini berisi informasi pengajuan surat meliputi nama penduduk, judul, detail dan status yang dikirimkan oleh *user*.

Nama Penduduk	Judul	Detail	Status	Aksi
				Edit Detail Hapus

Gambar 3. 50 Tampilan Halaman Manajemen Surat

j) Halaman Manajemen UMKM

Pemilik	Nama UMKM	Gambar	Lokasi	Koordinat	Jumlah Produk	Aksi
						Edit Detail Hapus

Gambar 3. 51 Tampilan Halaman Manajemen UMKM

Gambar 3.51 menunjukkan tampilan halaman manajemen UMKM. Pada halaman ini berisi informasi UMKM meliputi pemilik, nama UMKM, gambar, lokasi, koordinat, dan jumlah produk. Dalam halaman ini administrator juga dapat mengedit UMKM, melihat detail UMKM dan menghapus UMKM.

k) Halaman Manajemen Produk UMKM

Nama UMKM	Nama Produk	Detail	Gambar	Harga	Stok	Dilihat	Aksi
							<button>Edit</button> <button>Detail</button> <button>Hapus</button>

Gambar 3. 52 Tampilan Halaman Manajemen Produk UMKM

Gambar 3.52 menunjukkan tampilan halaman manajemen produk UMKM. Pada halaman ini berisi informasi produk UMKM meliputi nama UMKM, nama produk UMKM, detail, gambar, harga, stok, dan dilihat. Dalam halaman ini administrator juga dapat mengedit produk UMKM, melihat detail produk UMKM dan menghapus produk UMKM.

l) Halaman Manajemen Keahlian

Nama Pengguna	Judul	Lampiran	Dibuat Pada	Aksi
				<button>Edit</button> <button>Detail</button> <button>Hapus</button>

Gambar 3. 53 Tampilan Halaman Manajemen Keahlian Penduduk

Gambar 3.53 menunjukkan tampilan halaman manajemen keahlian penduduk. Pada halaman ini berisi informasi keahlian penduduk meliputi nama, judul, lampiran dan tanggal pembuatan. Dalam halaman ini administrator juga dapat mengedit keahlian penduduk, melihat detail keahlian penduduk dan menghapus keahlian penduduk.

m) Halaman Manajemen Profil Desa

The screenshot shows a web-based application interface titled 'DERING'. On the left, there's a sidebar with various menu items such as Dashboard, Kependudukan, Inventaris, Layanan, Ekonomi, and Tentang Desa. The main content area is titled 'Profil Desa'. It contains several input fields: a circular 'Logo' placeholder, a file input field 'Pilih Logo Desa' with a 'Browse' button, a text input field 'Nama Desa', a text input field 'Tagline', a text input field 'Alamat', and two text input fields for 'Latitude' and 'Longitude'. At the bottom right is a 'Simpan' button.

Gambar 3. 54 Tampilan Halaman Manajemen Profil Desa

Gambar 3.54 menunjukkan tampilan halaman manajemen profil desa. Pada halaman ini administrator dapat menambahkan logo desa, nama desa, *tagline*, alamat serta latitude dan longitude untuk menunjukkan peta desa.

3.4.4. *Sprint 4: Pembuatan dan Pengujian Fitur Aplikasi*

Sprint ini merupakan pembahasan dari perancangan halaman dari setiap fitur, pembuatan, dan pengujian fitur aplikasi yang terdiri dari aplikasi Android untuk *user* maupun Web untuk administrator. Berikut beberapa fitur yang diuraikan dalam pembahasan *sprint* ini.

1) Aplikasi Pengguna

Pada aplikasi pengguna terdiri dari beberapa fitur berikut.

a) *Splash Screen*

Berikut hasil pembuatan dan pengujian fitur *splash screen*:

1) Kode Program

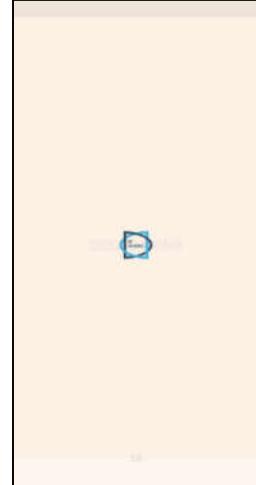
Keterangan kode program sebagai berikut:

(Lampiran 5 Kode Fitur *Splash Screen*)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3.7 Pengujian Kompatibilitas Fitur *Splash Screen*

Halaman Aplikasi	Android
<i>Splash Screen</i>	
Keterangan: Tampilan <i>splash screen</i> pada android sesuai dengan desain yang direncanakan.	

Tabel 3.8 Pengujian Fungsionalitas Fitur *Splash Screen*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Splash Screen</i>	Pengguna dapat melihat logo aplikasi saat pertama kali dijalankan	Pengguna membuka aplikasi yang ada pada <i>smartphone</i>	Pengguna dapat mengetahui logo atau tampilan awal dari aplikasi	✓

Berdasarkan pengujian Tabel 3.8 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalananya fitur *splash screen* saat aplikasi pengguna dijalankan.

b) Fitur *Login* dan Verifikasi

Berikut hasil pembuatan dan pengujian fitur *login* dan verifikasi:

1) Kode Program

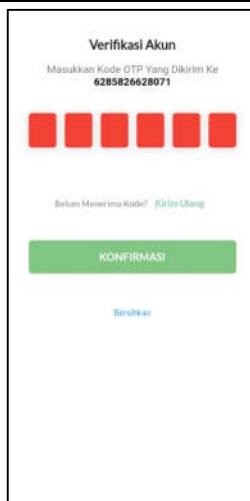
Keterangan kode program sebagai berikut:

(Lampiran 6 Kode Fitur *Login* dan verifikasi)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3.9 Pengujian Kompatibilitas Fitur *Login* dan Verifikasi

Halaman Aplikasi	Android
<i>Login</i>	
Verifikasi	

Keterangan: Tampilan *login* dan verifikasi pada android sesuai dengan desain yang direncanakan.

Tabel 3. 10 Pengujian Fungsionalitas Fitur *Login* dan verifikasi

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Melakukan proses <i>login</i>	Pengguna dapat menuju halaman verifikasi	Pengguna harus memasukan akun yang sudah terdaftar sebelumnya dengan memasukan nomor WhatsApp dan <i>password</i> yang baru	Pengguna diarahkan ke halaman verifikasi	✓
Melakukan proses verifikasi	Pengguna dapat menuju halaman <i>dashboard</i>	Pengguna memasukan kode verifikasi yang telah dikirimkan melalui nomor WhatsApp yang diinputkan	Pengguna diarahkan ke halaman <i>dashboard</i>	✓

Dari hasil pengujian Tabel 3.10 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur *login* dan verifikasi saat pengguna menginputkan nomor WhatsApp dan menerima kode OTP melalui WhatsApp.

c) Fitur *Dashboard*

Berikut hasil pembuatan dan pengujian fitur *dashboard*:

1) Kode Program

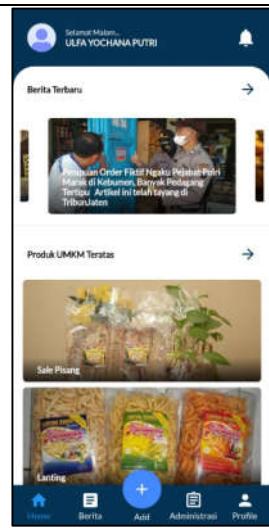
Keterangan kode program sebagai berikut:

(Lampiran 7 Kode Fitur *Dashboard*)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 11 Pengujian Kompatibilitas Fitur *Dashboard*

Halaman Aplikasi	Android
<i>Dashboard</i>	
<p>Keterangan: Tampilan <i>dashboard</i> pada android sesuai dengan desain yang direncanakan.</p>	

Tabel 3. 12 Pengujian Fungsionalitas Fitur *Dashboard*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Melihat halaman <i>dashboard</i>	Pengguna dapat mengakses	Pengguna memilih fitur yang ada pada	Halaman <i>dashboard</i> tampil	✓

	halaman <i>dashboard</i>	halaman <i>dashboard</i>		
--	-----------------------------	-----------------------------	--	--

Dari hasil pengujian Tabel 3.12 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur *dashboard* ketika pengguna dapat mengakses fitur yang ada pada halaman *dashboard*.

d) Fitur Berita

Berikut hasil pembuatan dan pengujian fitur berita:

1) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 8 Kode Fitur Berita)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 13 Pengujian Kompatibilitas Fitur Berita

Halaman Aplikasi	Android
Berita	 <p>The screenshot displays the news feed interface of the application. At the top, there is a search bar labeled 'Cari Berita' and three filter buttons: 'Semua', 'Berita Dicoba', and 'Dikira'. Below these are three news items, each with a thumbnail image and a brief title. The first item shows two people in a formal setting, the second shows a group of people outdoors, and the third shows two men at a podium. At the bottom of the screen, there is a navigation bar with five icons: Home, Berita (highlighted in blue), Add (+), Adminstrasi, and Profil.</p>

Detail Berita	<p>Detail Berita</p> <p>Penipuan Order Fiktif Ngaku Pejabat Polri Marak di Kebumen, Banyak Pedagang Terjipu Artikel ini telah tayang di TribunJateng</p> <p>8 September 2021 Kepahitan dirasakan akibat order-fiktif melalui SMS atau pesan WhatsApp yang mengatasnamakan petinggi Polisi. Meskipun Polres sedang marak aktivitas izin.</p> <p>Pelaku seringkali memanfaatkan kesempatan makar, berengah pada kepada warga dengan mengatakan namakan petinggi Polisi agar akinya mudah.</p> <p>Ini sangat meresahkan masyarakat, terutama para pedagang yang mengalami kerugian.</p> <p>Tak ingin akibat itu terus berlanjut, Polres Kebumen melalui Polsek melakukan sosialisasi tentang modus penipuan banu ini agar diwaspada bersama.</p> <p>Sing ini Polsek Petambahan menggelar sosialisasi modus penipuan order-fiktif dengan cara membagikan poster.</p> <p>Poster itu berisi pesan untuk melakukan cek dan risiko order yang menggunakan keadaan zona pedagang di kompleks Pasar Petambahan, Serang (de 9/2021).</p> <p>Dipelopori Kapolda Jateng, AKBP Peter Yandriana melalui Kepala Muspika Polres Tegalrejo, Iptu Tegmens, hal ini agar masyarakat lebih teliti dalam menghadapi</p>
Keterangan: Tampilan berita pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 14 Pengujian Fungsionalitas Fitur Berita

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Melihat daftar berita	Pengguna dapat melihat daftar berita untuk mengetahui berita terbaru	Pengguna membuka menu berita yang terdapat pada halaman	Halaman daftar berita tampil	✓
Melihat detail berita	Pengguna dapat melihat detail berita	Pengguna memilih berita yang terdapat pada halaman	Halaman detail berita tampil	✓

Dari hasil pengujian Tabel 3.14 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur berita ketika pengguna memilih berita yang terdapat pada halaman.

- e) Fitur Administrasi Surat

Berikut hasil pembuatan dan pengujian fitur administrasi surat:

1) Kode Program

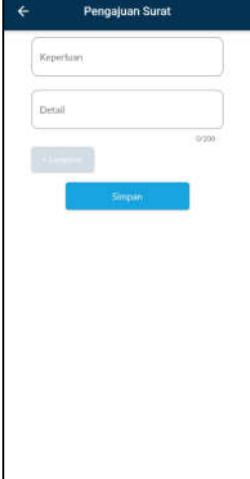
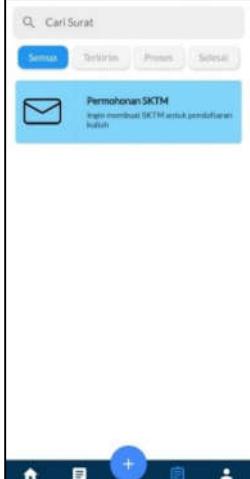
Keterangan kode program sebagai berikut:

(Lampiran 9 Kode Fitur Administrasi Surat)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 15 Pengujian Kompatibilitas Fitur Administrasi Surat

Halaman Aplikasi	Android
Pengajuan Surat	
Daftar Surat	

Keterangan: Tampilan administrasi surat pada android sesuai dengan desain yang direncanakan.

Tabel 3. 16 Pengujian Fungsionalitas Fitur Administrasi Surat

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Mengajukan surat	Pengguna dapat mengajukan surat	Pengguna mengisi judul, detail dan lampiran untuk mengajukan surat	Surat dikirimkan ke administrator	✓
Menampilkan surat	Pengguna dapat melihat daftar surat	Pengguna melihat daftar surat yang diinginkan	Halaman daftar surat tampil	✓

Berdasarkan hasil pengujian Tabel 3.16 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur administrasi surat ketika pengguna mengajukan surat dan melihat daftar surat.

f) Fitur Aspirasi

Berikut hasil pembuatan dan pengujian fitur aspirasi:

1) Kode Program

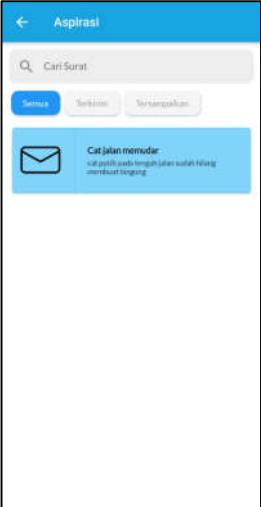
Keterangan kode program sebagai berikut:

(Lampiran 10 Kode Fitur Aspirasi)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 17 Pengujian Kompatibilitas Fitur Aspirasi

Halaman Aplikasi	Android
Pengajuan Aspirasi	
Daftar Aspirasi	
Keterangan: Tampilan aspirasi pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 18 Pengujian Fungsionalitas Fitur Aspirasi

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Mengajukan aspirasi	Pengguna dapat	Pengguna mengisi judul, detail dan	Aspirasi dikirimkan ke administrator	✓

	mengajukan aspirasi	lampiran untuk mengajukan aspirasi		
Menampilkan aspirasi	Pengguna dapat melihat daftar aspirasi	Pengguna melihat daftar aspirasi yang diinginkan	Halaman daftar aspirasi tampil	✓

Dari hasil pengujian Tabel 3.18 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur pengajuan aspirasi dan daftar aspirasi.

g) Fitur Profil

Berikut hasil pembuatan dan pengujian fitur profil:

1) Kode Program

Keterangan kode program sebagai berikut:

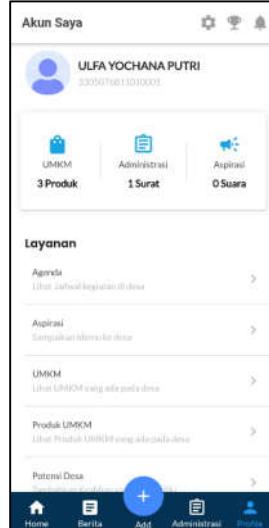
(Lampiran 11 Kode Fitur Profil)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 19 Pengujian Kompatibilitas Fitur Profil

Halaman Aplikasi	Android

Profil		<p>Keterangan: Tampilan profil pada android sesuai dengan desain yang direncanakan.</p>
--------	--	---

Tabel 3. 20 Pengujian Fungsionalitas Fitur Profil

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan profil	Pengguna dapat melihat profil dan menu-menu yang dapat diakses oleh pengguna	Pengguna memilih menu profil	Halaman profil tampil	✓
Mengubah nomor WhatsApp	Pengguna dapat mengubah nomor WhatsApp	Pengguna memilih menu ubah nomor WhatsApp kemudian melakukan perubahan no	Nomor WhatsApp diperbarui	✓

		WhatsApp dan menyimpannya		
Melakukan <i>logout</i>	Pengguna dapat keluar dari aplikasi	Pengguna memilih menu <i>logout</i>	Menampilkan konfirmasi <i>logout</i> dan menampilkan halaman <i>login</i>	✓

Dari hasil pengujian Tabel 3.20 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur *profil* ketika pengguna menggunakan menu yang ada pada halaman profil seperti menampilkan profil, mengubah profil, mengubah nomor WhatsApp, mengubah *password* dan melakukan *logout*.

h) Fitur UMKM

Berikut hasil pembuatan dan pengujian fitur detail surat:

1) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 12 Kode Fitur UMKM)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 21 Pengujian Kompatibilitas Fitur UMKM

Halaman Aplikasi	Android

UMKM		
Produk UMKM		
Keterangan: Tampilan UMKM pada android sesuai dengan desain yang direncanakan.		

Tabel 3. 22 Pengujian Fungsionalitas Fitur UMKM

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Melihat halaman UMKM	Pengguna dapat menggunakan fitur UMKM	Pengguna memilih menu UMKM	Halaman UMKM tampil	✓

Mengelola UMKM	Pengguna dapat mengelola UMKM	Pengguna mengisi data UMKM dan menyimpannya	Data UMKM berhasil ditambahkan	✓
Menambahkan data produk UMKM	Pengguna dapat menambahkan produk UMKM	Pengguna mengisi data produk UMKM dan menyimpannya	Data produk UMKM berhasil ditambahkan	✓
Melihat produk UMKM	Pengguna dapat melihat produk UMKM desa	Pengguna memilih menu produk UMKM	Halaman produk UMKM tampil	✓

Dari hasil pengujian Tabel 3.22 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur UMKM ketika pengguna menggunakan menu yang ada pada halaman UMKM seperti menampilkan halaman UMKM, mengelola UMKM, menambahkan data produk UMKM dan melihat produk UMKM.

i) Fitur Keahlian

Berikut hasil pembuatan dan pengujian fitur keahlian:

1) Kode Program

Keterangan kode program sebagai berikut:

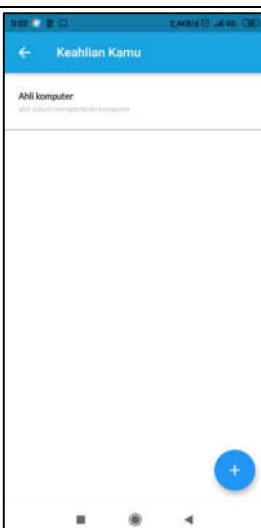
(Lampiran 13 Kode Fitur Keahlian)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 23 Pengujian Kompatibilitas Fitur Keahlian

Halaman Aplikasi	Android
------------------	---------

			
			
Keterangan: Tampilan keahlian desa pada android sesuai dengan desain yang direncanakan.			

Tabel 3. 24 Pengujian Fungsionalitas Fitur Keahlian Desa

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menambahkan data keahlian	Pengguna dapat menambahkan keahlian	Pengguna mengisi data keahlian dan menyimpannya	Data keahlian berhasil ditambahkan	✓

Melihat daftar keahlian	Pengguna dapat melihat daftar keahlian	Pengguna melihat daftar keahlian	Halaman daftar keahlian tampil	✓
-------------------------	--	----------------------------------	--------------------------------	---

Dari hasil pengujian Tabel 3.24 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur keahlian ketika pengguna menambahkan data keahlian dan menampilkannya.

2) Website Administrator

Pada website administrator terdiri dari beberapa fitur berikut.

a. Login Administrator

Berikut hasil pembuatan dan pengujian fitur *login* administrator:

1) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 14 Kode Fitur *Login* Administrator)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 25 Pengujian Kompatibilitas Fitur *Login*

Halaman Aplikasi	Website
<i>Login</i> Administrator	

Keterangan: Tampilan halaman *login* administrator pada website sesuai dengan desain yang direncanakan.

Tabel 3. 26 Pengujian Fungsionalitas Fitur *Login* Administrator

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Login</i>	Administrator dapat masuk ke halaman selanjutnya	Administrator dapat memasukan <i>email</i> dan <i>password</i>	Manampilkan halaman <i>dashboard</i> administrator	✓

Dari hasil pengujian Tabel 3.26 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur *login* administrator ketika administrator memasukan *email* dan *password*.

b. Fitur *Dashboard*

Berikut hasil pembuatan dan pengujian fitur *dashboard* administrator:

1) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 15 Kode Fitur *Dashboard* Administrator)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 27 Pengujian Kompatibilitas Fitur *Dashboard* Administrator

Halaman Aplikasi	Website

Dashboard
Administrator

Keterangan: Tampilan halaman *dashboard* administrator pada website sesuai dengan desain yang direncanakan.

Tabel 3. 28 Pengujian Fungsionalitas Fitur *Dashboard* Administrator

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan halaman <i>dashboard</i> administrator	Administrator dapat melihat informasi pada halaman <i>dashboard</i>	Administrator dapat mengklik menu <i>dashboard</i> pada sidebar	Manampilkan informasi pada halaman <i>dashboard</i>	✓

Dari hasil pengujian Tabel 3.28 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur *dashboard* administrator seperti menampilkan kalkulasi jumlah penduduk, jumlah penduduk laki-laki, jumlah penduduk perempuan, jumlah *user online*, jumlah UMKM, jumlah produk UMKM, jumlah ketrampilan diinput, dan jumlah berita dipublikasikan. Kemudian halaman *dashboard* administrator juga menampilkan statistic keahlian penduduk, data penduduk paling aspiratif, data UMKM terproduktif, produk terbaik, surat masuk terbaru dan aspirasi dari penduduk.

c. Fitur Manajemen Penduduk

Berikut hasil pembuatan dan pengujian fitur manajemen penduduk:

- 1) Kode Program

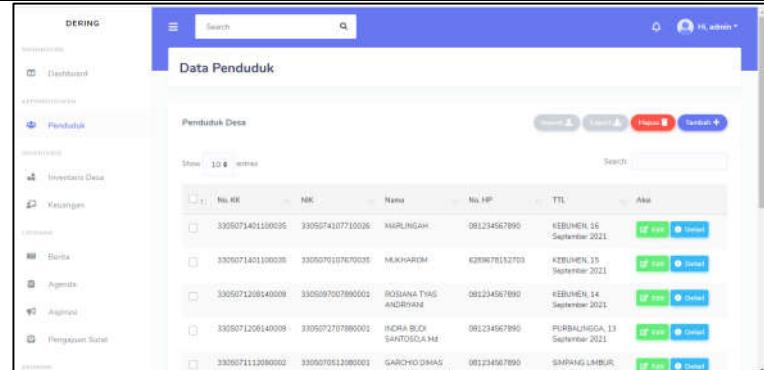
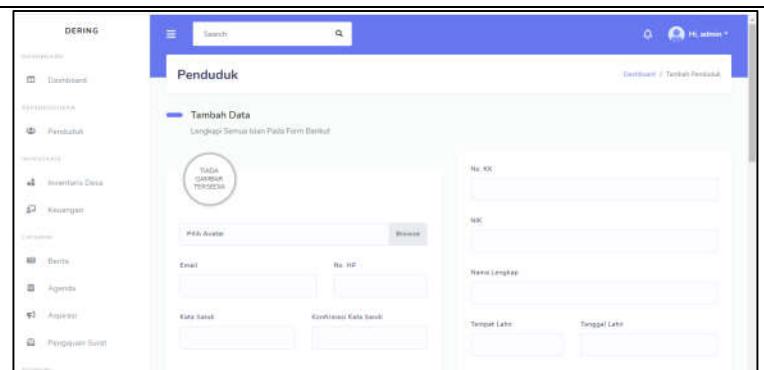
Keterangan kode program sebagai berikut:

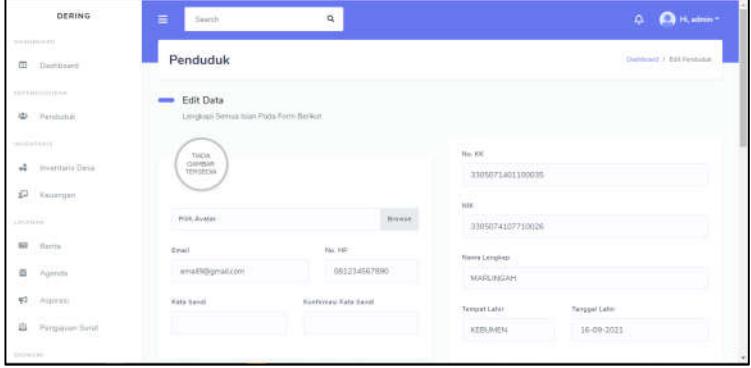
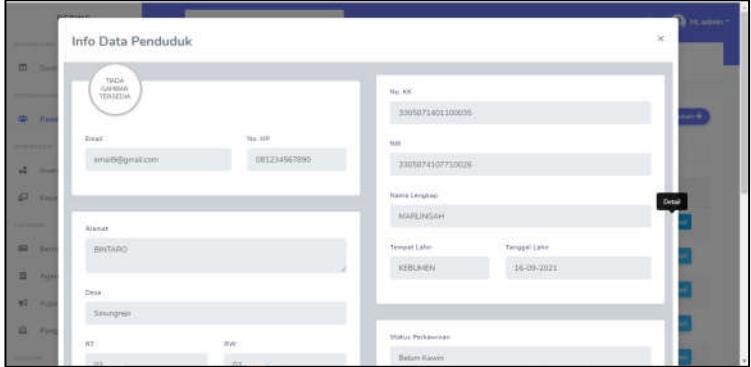
(Lampiran 16 Kode Fitur Manajemen Penduduk)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 29 Pengujian Kompatibilitas Fitur Manajemen Penduduk

Halaman Aplikasi	Website																																				
Manajemen Penduduk	 <p>DERING</p> <p>Data Penduduk</p> <p>Penduduk Desa</p> <p>Show: 10 of 1 entries</p> <table border="1"> <thead> <tr> <th>No. KK</th> <th>NIK</th> <th>Nama</th> <th>No. HP</th> <th>TTL</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>3305071401100035</td> <td>3305071407710026</td> <td>MURUNGAH</td> <td>081234567890</td> <td>KELUHAN, 15 September 2021</td> <td>Detail Hapus</td> </tr> <tr> <td>3305071401100036</td> <td>3305070107670035</td> <td>MUKHAROM</td> <td>0283678152703</td> <td>KELUHAN, 15 September 2021</td> <td>Detail Hapus</td> </tr> <tr> <td>3305071208140009</td> <td>3305080707880001</td> <td>RODANA THIS ANDRIYAH</td> <td>081234567890</td> <td>KELUHAN, 14 September 2021</td> <td>Detail Hapus</td> </tr> <tr> <td>3305071206140009</td> <td>3305072707880001</td> <td>INCHA BUDI SANTOSOAMI</td> <td>081234567890</td> <td>PURBAHAGGA, 13 September 2021</td> <td>Detail Hapus</td> </tr> <tr> <td>3305071112080020</td> <td>3305070512080021</td> <td>GABRIEL DMAS ALTAINDRAZKA</td> <td>081234567890</td> <td>SARWANG LAMUR, 13 September 2021</td> <td>Detail Hapus</td> </tr> </tbody> </table>	No. KK	NIK	Nama	No. HP	TTL	Aksi	3305071401100035	3305071407710026	MURUNGAH	081234567890	KELUHAN, 15 September 2021	Detail Hapus	3305071401100036	3305070107670035	MUKHAROM	0283678152703	KELUHAN, 15 September 2021	Detail Hapus	3305071208140009	3305080707880001	RODANA THIS ANDRIYAH	081234567890	KELUHAN, 14 September 2021	Detail Hapus	3305071206140009	3305072707880001	INCHA BUDI SANTOSOAMI	081234567890	PURBAHAGGA, 13 September 2021	Detail Hapus	3305071112080020	3305070512080021	GABRIEL DMAS ALTAINDRAZKA	081234567890	SARWANG LAMUR, 13 September 2021	Detail Hapus
No. KK	NIK	Nama	No. HP	TTL	Aksi																																
3305071401100035	3305071407710026	MURUNGAH	081234567890	KELUHAN, 15 September 2021	Detail Hapus																																
3305071401100036	3305070107670035	MUKHAROM	0283678152703	KELUHAN, 15 September 2021	Detail Hapus																																
3305071208140009	3305080707880001	RODANA THIS ANDRIYAH	081234567890	KELUHAN, 14 September 2021	Detail Hapus																																
3305071206140009	3305072707880001	INCHA BUDI SANTOSOAMI	081234567890	PURBAHAGGA, 13 September 2021	Detail Hapus																																
3305071112080020	3305070512080021	GABRIEL DMAS ALTAINDRAZKA	081234567890	SARWANG LAMUR, 13 September 2021	Detail Hapus																																
Tambah Data Penduduk	 <p>DERING</p> <p>Penduduk</p> <p>Tambah Data</p> <p>Lengkap Semua Isian Pada Form Berikut.</p> <p>NIK: <input type="text"/></p> <p>Nama: <input type="text"/></p> <p>Email: <input type="text"/></p> <p>No. HP: <input type="text"/></p> <p>Kata Sandi: <input type="password"/></p> <p>konfirmasi Kata Sandi: <input type="password"/></p> <p>Tanggal Lahir: <input type="date"/></p> <p>Tanggal Lahir: <input type="date"/></p>																																				

Edit Data Penduduk	
Detail Data Penduduk	
<p>Keterangan: Tampilan halaman manajemen penduduk pada website sesuai dengan desain yang direncanakan.</p>	

Tabel 3. 30 Pengujian Fungsionalitas Fitur Manajemen Penduduk

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data penduduk	Administrator dapat melihat data penduduk	Administrator dapat mengklik menu penduduk pada sidebar	Manampilkan data penduduk	✓
<i>Import</i> data penduduk	Administrator dapat melakukan <i>import</i> data penduduk	Administrator dapat mengklik tombol <i>import</i> kemudian	Data penduduk berhasil di <i>import</i>	✓

		memilih file yang akan di <i>import</i>		
<i>Export</i> data penduduk	Administrator dapat melakukan <i>export</i> data penduduk	Administrator dapat mengklik tombol <i>export</i>	Data penduduk berhasil di <i>export</i>	✓
Menambahkan data penduduk	Administrator dapat menambahkan data penduduk	Administrator dapat mengklik tombol tambah dan mengisi data penduduk kemudian menyimpannya	Data penduduk berhasil ditambahkan	✓
Mengedit data penduduk	Administrator dapat mengedit data penduduk	Administrator dapat mengklik tombol edit dan mengubah data penduduk kemudian menyimpannya	Data penduduk berhasil diperbarui	✓
Menghapus data penduduk	Administrator dapat menghapus data penduduk	Administrator dapat mengklik tombol hapus	Data penduduk berhasil dihapus	✓
Melihat detail data penduduk	Administrator dapat melihat	Administrator dapat	Detail data penduduk	✓

	detail data penduduk	mengklik tombol detail	berhasil ditampilkan	
Mencari data penduduk	Administrator dapat mencari data penduduk	Administrator dapat mencari data penduduk dengan menuliskan nama pada kotak pencarian	Data penduduk yang dicari berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.30 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen penduduk seperti menampilkan data penduduk, *import* data penduduk, *export* data penduduk, menambahkan data penduduk, mengedit data penduduk, menghapus data penduduk, melihat detail data penduduk dan mencari data penduduk.

d. Fitur Manajemen Inventaris Desa

Berikut hasil pembuatan dan pengujian fitur manajemen inventaris desa:

1) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 17 Kode Fitur Manajemen Inventaris Desa)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 31 Pengujian Kompatibilitas Fitur Manajemen Penduduk

Halaman Aplikasi	Website

<h3>Manajemen Inventaris Desa</h3>	<p>Data Inventaris Desa</p> <table border="1"> <thead> <tr> <th>Type</th> <th>Nama</th> <th>Ukuran</th> <th>Jumlah</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>Alat</td> <td>Gelas</td> <td>-</td> <td>50</td> <td>Baik</td> <td> </td> </tr> <tr> <td>Alat</td> <td>Piring</td> <td>-</td> <td>100</td> <td>Baik</td> <td> </td> </tr> <tr> <td>Alat Kebutuhan</td> <td>Sapu</td> <td>-</td> <td>5</td> <td>Baik</td> <td> </td> </tr> </tbody> </table>	Type	Nama	Ukuran	Jumlah	Status	Aksi	Alat	Gelas	-	50	Baik		Alat	Piring	-	100	Baik		Alat Kebutuhan	Sapu	-	5	Baik	
Type	Nama	Ukuran	Jumlah	Status	Aksi																				
Alat	Gelas	-	50	Baik																					
Alat	Piring	-	100	Baik																					
Alat Kebutuhan	Sapu	-	5	Baik																					
<h3>Tambah Data Inventaris</h3>	<p>Tambah Data Inventaris</p> <p>Type: Alat</p> <p>Name: Gelas</p> <p>Ukuran: -</p> <p>Jumlah: 50</p> <p>Status: Baik</p>																								
<h3>Edit Data Inventaris</h3>	<p>Edit Data Inventaris</p> <p>Type: Alat</p> <p>Name: Gelas</p> <p>Ukuran: -</p> <p>Jumlah: 50</p> <p>Status: Baik</p>																								
<h3>Detail Data Inventaris</h3>	<p>Info Data Inventaris</p> <p>Type: Alat</p> <p>Name: Gelas</p> <p>Ukuran: -</p> <p>Jumlah: 50</p> <p>Status: Baik</p>																								

Keterangan: Tampilan halaman manajemen inventaris desa pada website sesuai dengan desain yang direncanakan.

Tabel 3. 32 Pengujian Fungsionalitas Fitur Manajemen Inventaris Desa

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data inventaris desa	Administrator dapat melihat data inventaris desa	Administrator dapat mengklik menu inventaris desa pada sidebar	Manampilkan data inventaris desa	✓
Menambahkan data inventaris desa	Administrator dapat menambahkan data inventaris desa	Administrator dapat mengklik tombol tambah dan mengisi data inventaris desa kemudian menyimpannya	Data inventaris desa berhasil ditambahkan	✓
Mengedit data inventaris desa	Administrator dapat mengedit data inventaris desa	Administrator dapat mengklik tombol edit dan mengubah data inventaris desa kemudian menyimpannya	Data inventaris desa berhasil diperbarui	✓
Menghapus data inventaris desa	Administrator dapat menghapus data	Administrator dapat mengklik tombol hapus	Data inventaris desa berhasil dihapus	✓

	inventaris desa			
Melihat detail data inventaris desa	Administrator dapat melihat detail data inventaris desa	Administrator dapat mengklik tombol detail	Detail data inventaris desa berhasil ditampilkan	✓
Mencari data inventaris desa	Administrator dapat mencari data inventaris desa	Administrator dapat mencari data inventaris desa dengan menuliskan nama pada kotak pencarian	Data inventaris desa yang dicari berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.32 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen inventaris desa seperti menampilkan data inventaris desa, menambahkan data inventaris desa, mengedit data inventaris desa, menghapus data inventaris desa, melihat detail data inventaris desa dan mencari data inventaris desa.

e. Fitur Manajemen Keuangan

Berikut hasil pembuatan dan pengujian fitur manajemen keuangan:

1) Kode Program

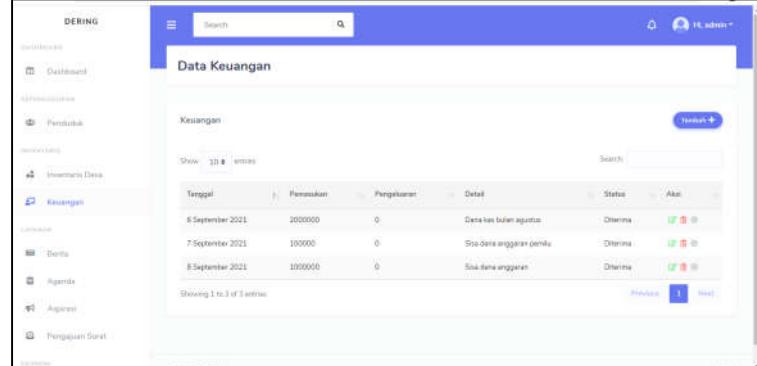
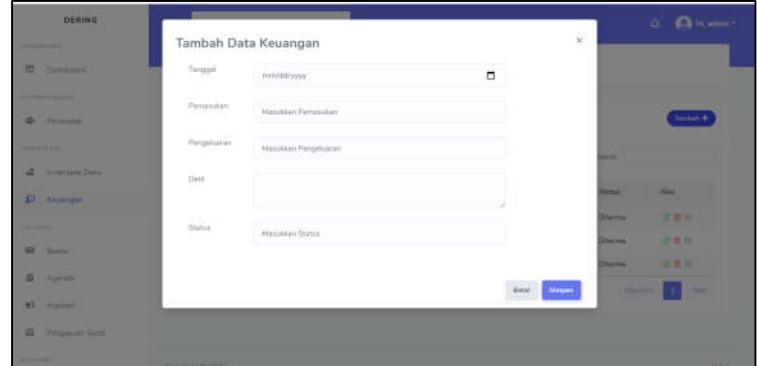
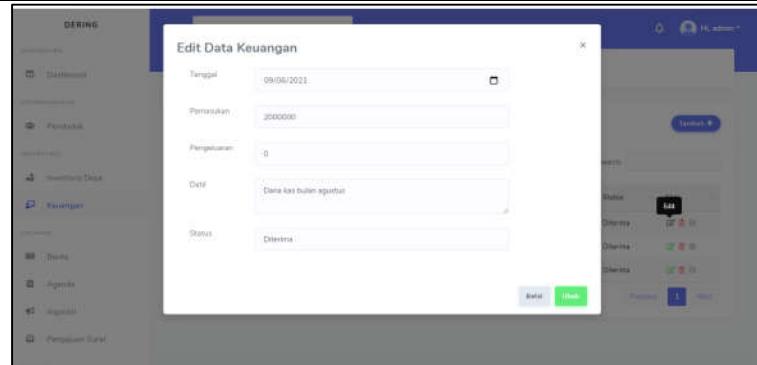
Keterangan kode program sebagai berikut:

(Lampiran 18 Kode Fitur Manajemen Keuangan)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 33 Pengujian Kompatibilitas Fitur Manajemen Keuangan

Halaman Aplikasi	Website
Manajemen Keuangan	 <p>The screenshot shows a table titled 'Data Keuangan' with three entries. The columns are: Tanggal (Date), Pemasukan (Income), Pengeluaran (Expense), Detail (Description), Status (Status), and Aksi (Action). The first entry is '6 September 2021 2000000 0 Dana ke bulan agustus Diterima', the second is '7 September 2021 100000 0 Sisa dana pengeluaran Diterima', and the third is '8 September 2021 1000000 0 Sisa dana pengeluaran Diterima'. There are buttons for 'Previous' and 'Next' at the bottom.</p>
Tambah Data Keuangan	 <p>The modal has fields for: Tanggal (Date) set to '09/08/2021', Pemasukan (Income) 'Masukkan Pemasukan', Pengeluaran (Expense) 'Masukkan Pengeluaran', Detail (Description) 'Dana ke bulan agustus', Status (Status) 'Diterima', and two buttons 'Batal' (Cancel) and 'Simpan' (Save).</p>
Edit Data Keuangan	 <p>The modal displays the same fields as the add form, but with different values: Tanggal (Date) '09/08/2021', Pemasukan (Income) '2000000', Pengeluaran (Expense) '0', Detail (Description) 'Dana ke bulan agustus', Status (Status) 'Diterima', and two buttons 'Batal' (Cancel) and 'Simpan' (Save).</p>

Detail Data Keuangan	
<p>Keterangan: Tampilan halaman manajemen keuangan pada website sesuai dengan desain yang direncanakan.</p>	

Tabel 3. 34 Pengujian Fungsionalitas Fitur Manajemen Keuangan

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data keuangan	Administrator dapat melihat data keuangan	Administrator dapat mengklik menu keuangan pada sidebar	Manampilkan data keuangan	✓
Menambahkan data keuangan	Administrator dapat menambahkan data keuangan	Administrator dapat mengklik tombol tambah dan mengisi data keuangan kemudian menyimpannya	Data keuangan berhasil ditambahkan	✓
Mengedit data keuangan	Administrator dapat mengedit data keuangan	Administrator dapat mengklik tombol edit	Data keuangan berhasil diperbarui	✓

		dan mengubah data keuangan kemudian menyimpannya		
Menghapus data keuangan	Administrator dapat menghapus data keuangan	Administrator dapat mengklik tombol hapus	Data keuangan berhasil dihapus	✓
Melihat detail data keuangan	Administrator dapat melihat detail data keuangan	Administrator dapat mengklik tombol detail	Detail data keuangan berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.34 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen keuangan seperti menampilkan data keuangan, menambahkan data keuangan, mengedit data keuangan, menghapus data keuangan, dan melihat detail data keuangan.

f. Fitur Manajemen Berita

Berikut hasil pembuatan dan pengujian fitur manajemen berita:

1) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 19 Kode Fitur Manajemen Berita)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 35 Pengujian Kompatibilitas Fitur Manajemen Berita

Halaman Aplikasi	Website

<h2>Manajemen Berita</h2>	
<h2>Tambah Berita</h2>	
<h2>Edit Berita</h2>	
<h2>Detail Berita</h2>	
<p>Keterangan: Tampilan halaman manajemen berita pada website sesuai dengan desain yang direncanakan.</p>	

Tabel 3. 36 Pengujian Fungsionalitas Fitur Manajemen Berita

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data berita	Administrator dapat melihat data berita	Administrator dapat mengklik menu berita pada <i>sidebar</i>	Manampilkan data berita	✓
Menambahkan data berita	Administrator dapat menambahkan data berita	Administrator dapat mengklik tombol tambah dan mengisi data berita kemudian menyimpannya	Data berita berhasil ditambahkan	✓
Mengedit data berita	Administrator dapat mengedit data berita	Administrator dapat mengklik tombol edit dan mengubah data berita kemudian menyimpannya	Data berita berhasil diperbarui	✓
Menghapus data berita	Administrator dapat menghapus data berita	Administrator dapat mengklik tombol hapus	Data berita berhasil dihapus	✓
Melihat detail data berita	Administrator dapat melihat	Administrator dapat	Detail data berita	✓

	detail data berita	mengklik tombol detail	berhasil ditampilkan	
--	--------------------	------------------------	----------------------	--

Dari hasil pengujian Tabel 3.36 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen berita seperti menampilkan data berita, menambahkan data berita, mengedit data berita, menghapus data berita, dan melihat detail data berita.

g. Fitur Manajemen Agenda

Berikut hasil pembuatan dan pengujian fitur manajemen agenda:

1) Kode Program

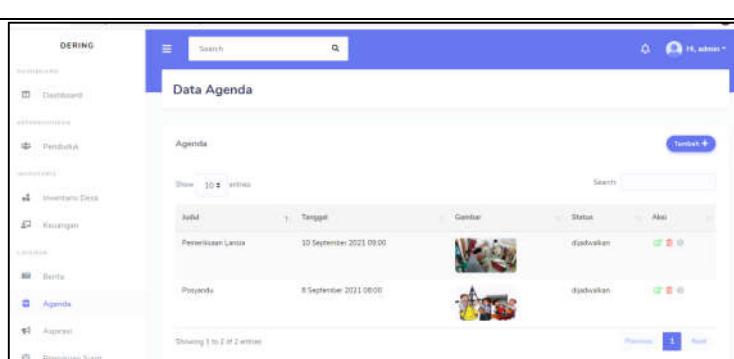
Keterangan kode program sebagai berikut:

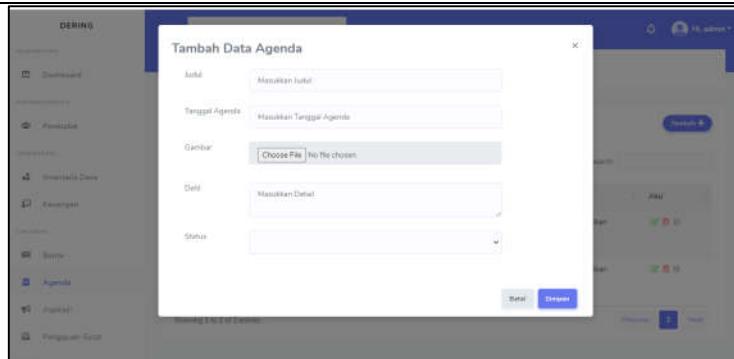
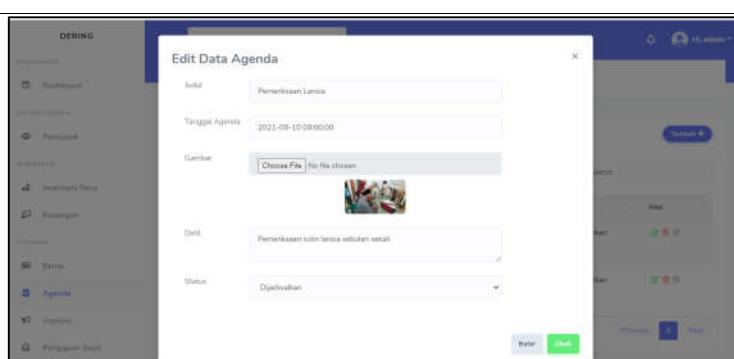
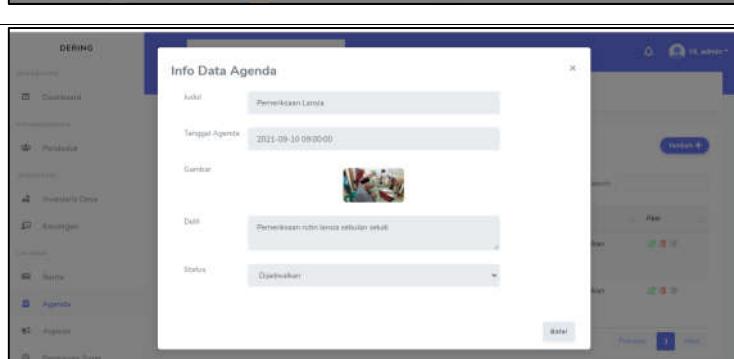
(Lampiran 20 Kode Fitur Manajemen Agenda)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 37 Pengujian Kompatibilitas Fitur Manajemen Agenda

Halaman Aplikasi	Website
Manajemen Agenda	

Tambah Agenda	
Edit Agenda	
Detail Agenda	
<p>Keterangan: Tampilan halaman manajemen agenda pada website sesuai dengan desain yang direncanakan.</p>	

Tabel 3. 38 Pengujian Fungsionalitas Fitur Manajemen Agenda

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data agenda	Administrator dapat melihat data agenda	Administrator dapat mengklik	Manampilkan data agenda	✓

		menu agenda pada <i>sidebar</i>		
Menambahkan data agenda	Administrator dapat menambahkan data agenda	Administrator dapat mengklik tombol tambah dan mengisi data agenda kemudian menyimpannya	Data agenda berhasil ditambahkan	✓
Mengedit data agenda	Administrator dapat mengedit data agenda	Administrator dapat mengklik tombol edit dan mengubah data agenda kemudian menyimpannya	Data agenda berhasil diperbarui	✓
Menghapus data agenda	Administrator dapat menghapus data agenda	Administrator dapat mengklik tombol hapus	Data agenda berhasil dihapus	✓
Melihat detail data agenda	Administrator dapat melihat detail data agenda	Administrator dapat mengklik tombol detail	Detail data agenda berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.38 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen agenda seperti menampilkan data agenda, menambahkan data agenda, mengedit data agenda, menghapus data agenda, dan melihat detail data agenda.

h. Fitur Manajemen Aspirasi

Berikut hasil pembuatan dan pengujian fitur manajemen aspirasi:

1) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 21 Kode Fitur Manajemen Aspirasi)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 39 Pengujian Kompatibilitas Fitur Manajemen Aspirasi

Halaman Aplikasi	Website
Manajemen Aspirasi	
Detail Aspirasi	<p>Keterangan: Tampilan halaman manajemen aspirasi pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 40 Pengujian Fungsionalitas Fitur Manajemen Aspirasi

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data aspirasi	Administrator dapat melihat data aspirasi	Administrator dapat mengklik menu aspirasi pada sidebar	Manampilkan data aspirasi	✓
Melihat detail data aspirasi	Administrator dapat melihat detail data aspirasi	Administrator dapat mengklik tombol detail	Detail data aspirasi berhasil ditampilkan	✓
Mencari data aspirasi	Administrator dapat mencari data aspirasi	Administrator dapat mencari data aspirasi dengan menuliskan nama pada kotak pencarian	Data aspirasi yang dicari berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.40 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen aspirasi seperti menampilkan data aspirasi, menghapus data aspirasi, melihat detail data aspirasi dan mencari data aspirasi.

i. Fitur Manajemen Pengajuan Surat

Berikut hasil pembuatan dan pengujian fitur manajemen pengajuan surat:

1) Kode Program

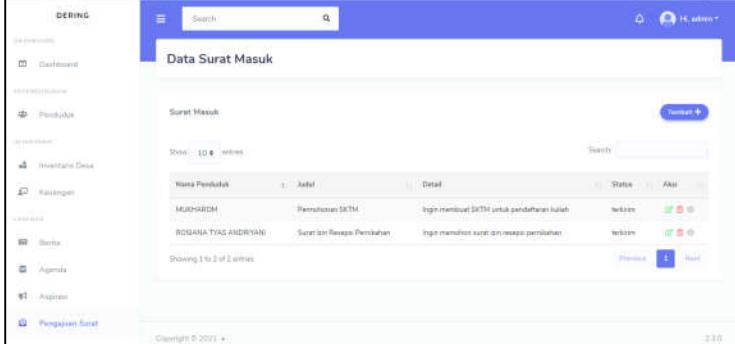
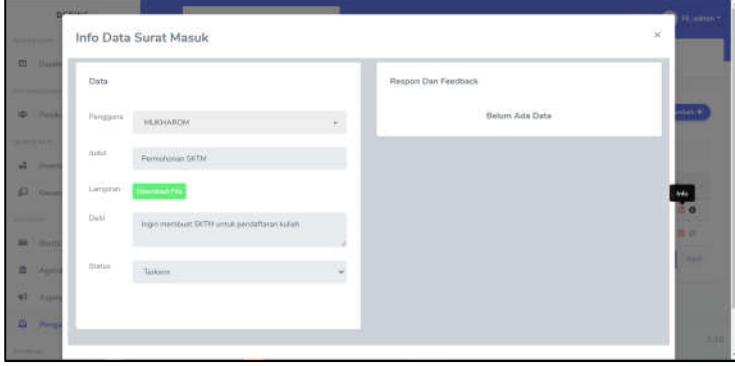
Keterangan kode program sebagai berikut:

(Lampiran 22 Kode Fitur Manajemen Pengajuan Surat)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 41 Pengujian Kompatibilitas Fitur Manajemen Pengajuan Surat

Halaman Aplikasi	Website
Manajemen Pengajuan Surat	
Detail Surat	 <p>Keterangan: Tampilan halaman manajemen pengajuan surat pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 42 Pengujian Fungsionalitas Fitur Manajemen Pengajuan Surat

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data pengajuan surat	Administrator dapat melihat data pengajuan surat	Administrator dapat mengklik menu	Manampilkan data pengajuan surat	✓

	pengajuan surat	pengajuan surat pada sidebar		
Melihat detail data pengajuan surat	Administrator dapat melihat detail data pengajuan surat	Administrator dapat mengklik tombol detail	Detail data pengajuan surat berhasil ditampilkan	✓
Mencari data pengajuan surat	Administrator dapat mencari data pengajuan surat	Administrator dapat mencari data pengajuan surat dengan menuliskan nama pada kotak pencarian	Data pengajuan surat yang dicari berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.42 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen pengajuan surat seperti menampilkan data pengajuan surat, menghapus data pengajuan surat, melihat detail data pengajuan surat dan mencari data pengajuan surat.

j. Fitur Manajemen UMKM

Berikut hasil pembuatan dan pengujian fitur manajemen UMKM:

1) Kode Program

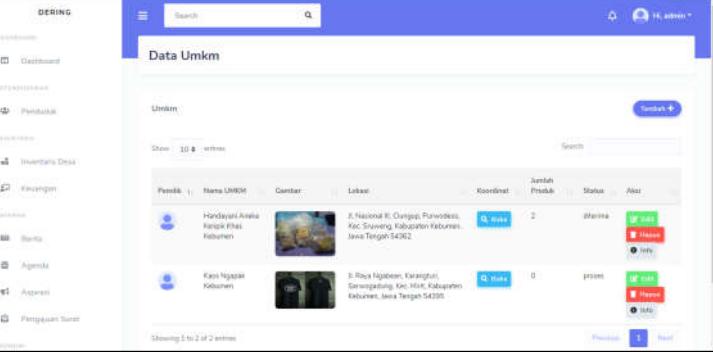
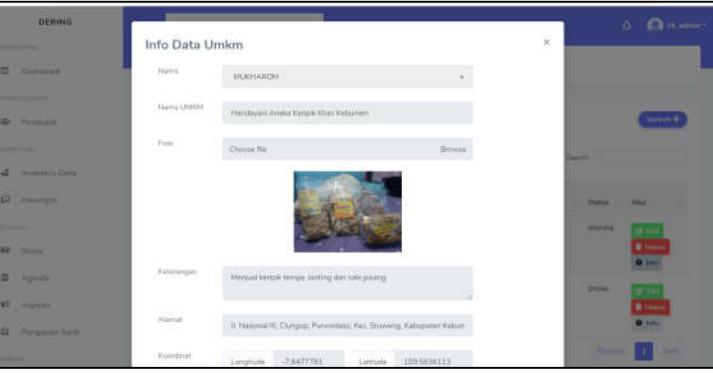
Keterangan kode program sebagai berikut:

(Lampiran 23 Kode Fitur Manajemen UMKM)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 43 Pengujian Kompatibilitas Fitur Manajemen UMKM

Halaman Aplikasi	Website
Manajemen UMKM	
Detail UMKM	
Keterangan: Tampilan halaman manajemen UMKM pada website sesuai dengan desain yang direncanakan.	

Tabel 3. 44 Pengujian Fungsionalitas Fitur Manajemen UMKM

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data UMKM	Administrator dapat melihat data UMKM	Administrator dapat mengklik menu UMKM pada sidebar	Manampilkan data UMKM	✓
Menghapus data UMKM	Administrator dapat	Administrator dapat	Data UMKM berhasil dihapus	✓

	menghapus data UMKM	mengklik tombol hapus		
Melihat detail data UMKM	Administrator dapat melihat detail data UMKM	Administrator dapat mengklik tombol detail	Detail data UMKM berhasil ditampilkan	✓
Mencari data UMKM	Administrator dapat mencari data UMKM	Administrator dapat mencari data UMKM dengan menuliskan nama UMKM pada kotak pencarian	Data UMKM yang dicari berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.44 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen UMKM seperti menampilkan data UMKM, menambahkan data UMKM, mengedit data UMKM, menghapus data UMKM, melihat detail data UMKM dan mencari data UMKM.

k. Fitur Manajemen Produk UMKM

Berikut hasil pembuatan dan pengujian fitur manajemen produk UMKM:

1) Kode Program

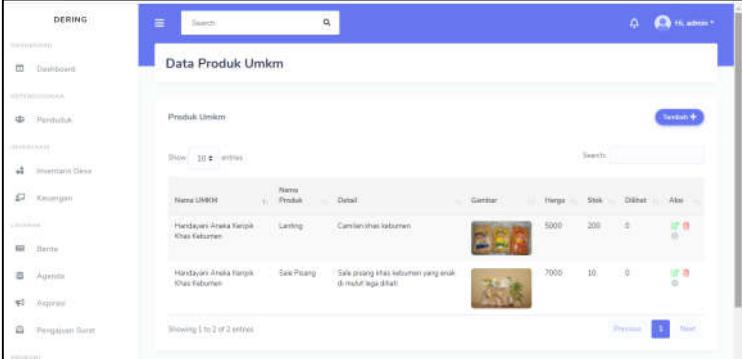
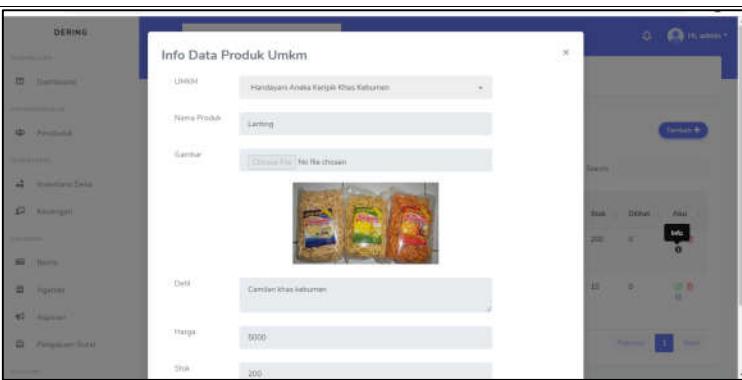
Keterangan kode program sebagai berikut:

(Lampiran 24 Kode Fitur Manajemen Produk UMKM)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 45 Pengujian Kompatibilitas Fitur Manajemen Produk UMKM

Halaman Aplikasi	Website
Manajemen Produk UMKM	
Detail Produk UMKM	
<p>Keterangan: Tampilan halaman manajemen produk UMKM pada website sesuai dengan desain yang direncanakan.</p>	

Tabel 3. 46 Pengujian Fungsionalitas Fitur Manajemen Produk UMKM

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data produk UMKM	Administrator dapat melihat data produk UMKM	Administrator dapat mengklik menu produk UMKM pada sidebar	Manampilkan data produk UMKM	√
Menghapus data produk UMKM	Administrator dapat menghapus	Administrator dapat	Data produk UMKM	√

	data produk UMKM	mengklik tombol hapus	berhasil dihapus	
Melihat detail data produk UMKM	Administrator dapat melihat detail data produk UMKM	Administrator dapat mengklik tombol detail	Detail data produk UMKM berhasil ditampilkan	✓
Mencari data produk UMKM	Administrator dapat mencari data produk UMKM	Administrator dapat mencari data produk UMKM dengan menuliskan nama UMKM pada kotak pencarian	Data produk UMKM yang dicari berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.46 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen produk UMKM seperti menampilkan data produk UMKM, menambahkan data produk UMKM, mengedit data produk UMKM, menghapus data produk UMKM, melihat detail data produk UMKM dan mencari data produk UMKM.

1. Fitur Manajemen Keahlian

Berikut hasil pembuatan dan pengujian fitur manajemen keahlian:

1) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 25 Kode Fitur Manajemen Keahlian)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 47 Pengujian Kompatibilitas Fitur Manajemen Keahlian

Halaman Aplikasi	Website
Manajemen Keahlian	
Detail Keahlian	
Keterangan: Tampilan halaman manajemen keahlian pada website sesuai dengan desain yang direncanakan.	

Tabel 3. 48 Pengujian Fungsionalitas Fitur Manajemen Keahlian

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data keahlian	Administrator dapat melihat data keahlian	Administrator dapat mengklik menu keahlian pada sidebar	Manampilkan data keahlian	✓
Menghapus data keahlian	Administrator dapat	Administrator dapat	Data keahlian berhasil dihapus	✓

	menghapus data keahlian	mengklik tombol hapus		
Melihat detail data keahlian	Administrator dapat melihat detail data keahlian	Administrator dapat mengklik tombol detail	Detail data keahlian berhasil ditampilkan	✓
Mencari data keahlian	Administrator dapat mencari data keahlian	Administrator dapat mencari data keahlian dengan menuliskan nama pada kotak pencarian	Data keahlian yang dicari berhasil ditampilkan	✓

Dari hasil pengujian Tabel 3.48 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen keahlian seperti menampilkan data keahlian, menambahkan data keahlian, mengedit data keahlian, menghapus data keahlian, melihat detail data keahlian dan mencari data keahlian.

m. Fitur Manajemen Profil Desa

Berikut hasil pembuatan dan pengujian fitur manajemen profil desa:

1) Kode Program

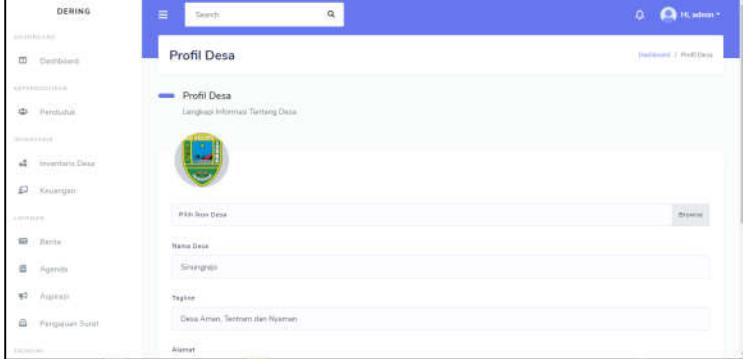
Keterangan kode program sebagai berikut:

(Lampiran 26 Kode Fitur Profil Desa)

2) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* website berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 49 Pengujian Kompatibilitas Fitur Manajemen Profil Desa

Halaman Aplikasi	Website
Manajemen Profil Desa	

Keterangan: Tampilan halaman manajemen profil desa pada website sesuai dengan desain yang direncanakan.

Tabel 3. 50 Pengujian Fungsionalitas Fitur Manajemen Profil Desa

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan data profil desa	Administrator dapat melihat data profil desa	Administrator dapat mengklik menu profil desa pada sidebar	Manampilkan data profil desa	✓
Memperbarui data profil desa	Administrator dapat memperbarui data profil desa	Administrator dapat mengisi data profil desa	Data keahlian berhasil diperbarui	✓

Dari hasil pengujian Tabel 3.50 didapatkan hasil bahwa fitur ini telah sesuai dengan pengujian ditandai dengan berjalannya fitur manajemen profil desa seperti menampilkan data profil desa dan memperbarui data profil desa.

3.4.5. Sprint 5: Perancangan Uji Kepuasan Pengguna

Pada tahapan ini dilakukan pengujian sistem yang telah jadi kepada beberapa pengguna. Dalam pengujian ini caranya dengan memberikan beberapa pertanyaan seperti yang ditunjukkan pada Tabel 3.41 untuk mendapatkan hasil perhitungan yang digunakan sebagai acuan untuk mengetahui tingkat kepuasan pengguna dalam menggunakan sistem ini. Perancangan uji kepuasan pengguna ini terdiri dari daftar pertanyaan, indikator penilaian, dan indikator kepuasan pengguna.

Tabel 3. 51. Daftar Pertanyaan

No	Kriteria Pertanyaan
1	Seberapa puas kebermanfaatan aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik <i>Crowdsourcing</i> ?
2	Seberapa puas kebermanfaatan fitur aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik <i>Crowdsourcing</i> ?
3	Seberapa puas kemudahan penggunaan aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik <i>Crowdsourcing</i> ?
4	Seberapa puas tampilan aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik <i>Crowdsourcing</i> ?
5	Seberapa puas pengalaman pengguna aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik <i>Crowdsourcing</i> ?
6	Seberapa puas kebermanfaatan aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik <i>Crowdsourcing</i> ?

Tabel 3. 52. Indikator Penilaian

Predikat	Nilai
Sangat Puas	5

Puas	4
Cukup	3
Tidak Puas	2
Sangat Tidak Puas	1

Tabel 3. 53. Indikator Kepuasan Pengguna

Kategori	Presentasi
Sangat Memuaskan	81% - 100%
Memuaskan	61% - 80%
Cukup Memuaskan	41% - 60%
Tidak Memuaskan	21% - 40%
Sangat Tidak Memuaskan	0 - 20 %

Untuk menghitung tingkat kepuasan pengguna menggunakan rumus sebagai berikut:

$$\text{Presentase Kepuasan Pengguna (\%)} = \frac{\text{Total kepuasan pengguna}}{\text{Total kepuasan maksimum pengguna}} \times 100\%$$

BAB IV ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis dan pembahasan sistem serta aplikasi yang sudah dirancang sebelumnya. Analisis dan pembahasan terdiri dari analisis serta pengujian sistem aplikasi dan kepuasan pengguna.

4.1. Analisis dan Pembahasan Sistem

Bagian ini berisi pembahasan serta analisis digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing* sebagai sebuah sistem yang digunakan untuk meningkatkan kualitas pelayanan pemerintah desa kepada masyarakat desa melalui kemudahan administrasi dan akses informasi serta mempermudah pelaku UMKM dalam mempromosikan produknya dengan menggunakan teknik *crowdsourcing*. Analisis sistem berisi penguraian dari suatu sistem yang utuh kedalam bagian-bagian dengan maksud mengidentifikasi dan mengevaluasi permasalahan yang terjadi.

4.1.1. Analisis Aplikasi Android

Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa ini terdiri dari aplikasi android untuk pengguna dan *website* untuk administrator. Aplikasi android dibangun menggunakan bahasa pemrograman Dart dengan menggunakan *framework* Flutter. Target pengguna dari aplikasi android yaitu penduduk desa dan pelaku UMKM yang tinggal di wilayah desa. Teknik *crowdsourcing* digunakan pada fitur mempromosikan produk UMKM. Ketika pelaku UMKM mengunggah sebuah produk dengan menyertakan lokasi UMKM dan dihubungan dengan *marketplace*, *user* dapat melihat produk tersebut dan dapat dengan mudah memesan melalui aplikasi *marketplace* yang dihubungkan atau dapat langsung menuju lokasi UMKM dengan bantuan Google Maps.

4.1.2. Analisis Website Administrator

Pada bagian *website* administrator dibangun menggunakan bahasa pemrograman PHP dengan *framework* Laravel. Peran *website* administrator pada aplikasi ini adalah untuk manajemen data yang dibutuhkan dalam proses pertukaran data.

Aplikasi ini menggunakan MySQL sebagai tempat penyimpanan datanya atau *databasenya*. Pada *platform android* dan *website* ini saling terhubung dengan menggunakan REST API (*Representational State Transfer Application Programming Interface*) yang memungkinkan antar *platform* untuk mengirimkan data satu sama lain. REST adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan (*scale*) terutama dalam pertukaran dan komunikasi data.

Proses dalam *request API* pada sistem ini yaitu setelah *server* menerima permintaan yang benar dan telah valid, maka *server* akan memproses permintaan menghubungkan dengan *database*. REST API pada aplikasi terdapat beberapa poin berikut.

1) URL *Endpoint*

Pada setiap *request* data dengan API pada aplikasi ini membutuhkan sebuah URL. Permintaan *request* API diakses melalui alamat URL <http://dering.techformacs/api/<route>>.

2) Parameter

Parameter yang dibutuhkan dalam setiap proses. Semua parameter harus valid dalam setiap permintaan, *server* akan memvalidasi dan akan mengembalikan balikan error apabila terjadi kesalahan. Parameter-parameter tersebut adalah *api_key*, *id*, dan *token*, ditunjukkan pada Gambar 4.1.

Params	Authorization	Headers (10)	Body	Pre-request Script	Tests	Settings
Headers (7 hidden)						
KEY	VALUE					
<input checked="" type="checkbox"/> api_key	8bc72dbd2869f4adc744a223cceca3c8357e1400					
<input checked="" type="checkbox"/> id	1					
<input checked="" type="checkbox"/> token	qUqYr6wcNQ					
Key	Value					

Gambar 4.1. Parameter *Request API*

3) HTTP Verb

Setiap *request* yang dilakukan, terdapat sebuah *verbs* atau *method* yang dipakai untuk meningkatkan kejelasan maksud *request* yang diberikan. Dalam aplikasi ini menggunakan *GET* untuk membaca atau mendapatkan data, *POST* untuk membuat data, dan *PUT* untuk memperbarui data.

4) Response

Apabila sistem berhasil memvalidasi permintaan yang masuk, maka sistem akan mengembalikan permintaan yang dibutuhkan. *Response* yang diberikan berupa format JSON. Contoh *response* dari permintaan yang valid ditampilkan pada Gambar 4.2.

```

1 {
2   "meta": {
3     "code": 200,
4     "status": "success",
5     "message": "Login Berhasil"
6   },
7   "data": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNj9.
eyJhdWQiOiISNDU0OWNiYS1iMGRnLTQyNvYtOWJiNC03M2E2NDI0NmI2ZjgjILC3edGkiOjI8MTU20DQxNzA4YmEyOTcwZThiNmYmGNiZjZlMG
I3MWNkOTx200k0MTE5zNMmF1NzUyMGQzZjAyNTE1NTNmDzKNTk0YTA20Th1ZGzNvYIsImIhdC16MTVzMTQ20DI4My45NzcSMTksIm5iZi26
MTYzMjTQ20D14My45NzcBMjcsInV4cCI6MTY2MzAwNDI4My45NTUBNTmInNiYiIj0NTQ4ZmU0M0f1NDQ2ZjM4ZGQyOGJ1YTcwZjR1jhkIi
wic2hvcGVzIijpbXX0.
K27MP84Kaw2SwNnfv2XhPeqvudoBPBPk120A2dBO5azMNE8gDd6POGSVJa6ySk4vw1rQAzAI-1lLUhug9i8q3mNEp6hZF2-IUFX6Ceczh184u
Qe6YxY4Vn1soHmR2dHnSE9EbG3sHCxVv7f9elA6SsLax6l7_ZXnV_Y37oV20hCSzBpxXnAHZheCvNb1iM2j1MgtkIN3Cp1uv8c9dCwq
-BkSeMah-17PqZw7hgzYt1Af_p5XyVGued1Qnk99hdAk9ca@01H7kDvjiELU-1ktb4840KwOmYcZVuUcQuEpvyTDvk2@tM2NPkTVQn4P2w
9dHuD98LxZiuAOUB66k2aD2lKe9g8po1Dy00WuqfQ11h51668I_jMk0XlfvKNS1X15y6sg59zRdxkmrgkCdxJVFae@xrbi6Yre6xCyEQd8F0M
-VtqbKhtwMc25IRvHOP5PvRj0LtzP26n8INs075LfIn1fmKHsY3K482ncP6RRR876xZj03s12LygNt64gvjTyOrtf81Xir68IEXH0_dp4_oKN
ayyVTzH2HTQoQV6FBRoob1g8yEJpdydVEoV8SU61xfND0pJQ8vUbh7X1zb98qzT7FIf3Ghs5PevUDvJRAGHJLGw3gjnQzK0fxwzIc-8cC
SaM13FANF6mPGmSzuCQAK4A"

```

Gambar 4.2. Contoh *Response* API

4.2. Pembahasan Sistem

Pada pembahasan sistem ini dilakukan pengujian terhadap aplikasi dan rekapitulasi mengenai beberapa pengujian pada sistem yang dilakukan pada tahapan kegiatan pelaksanaan.

4.2.1. Rekapitulasi Uji Tampilan dan Fungsionalitas Aplikasi Android

Pengujian tampilan dan fungsionalitas aplikasi android dilakukan dengan menggunakan tiga buah *smartphone* android yang memiliki spesifikasi yang berbeda-beda seperti yang terlihat pada Tabel 4.1.

Tabel 4. 1. Tabel Spesifikasi *Smartphone* Penguji Aplikasi Android

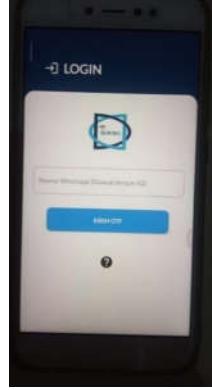
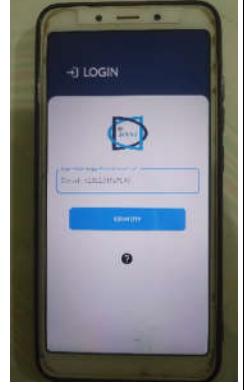
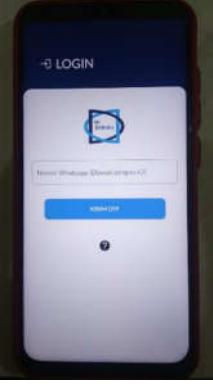
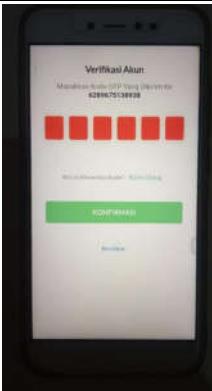
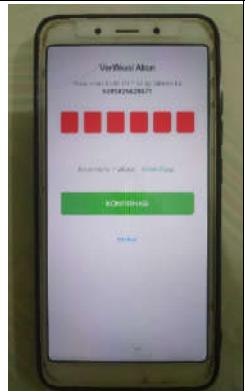
Tipe <i>Smartphone</i>	Resolusi Layar	Versi Android
------------------------	----------------	---------------

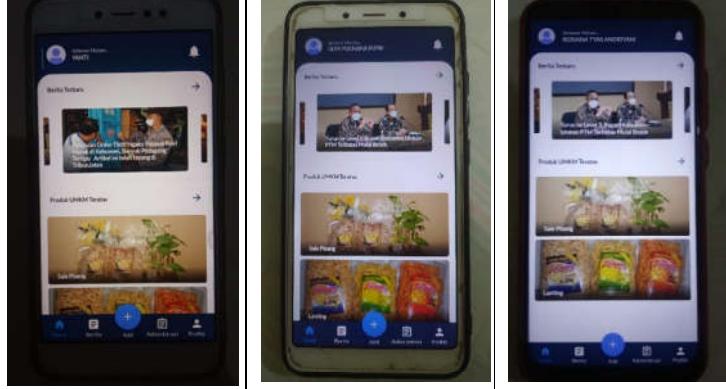
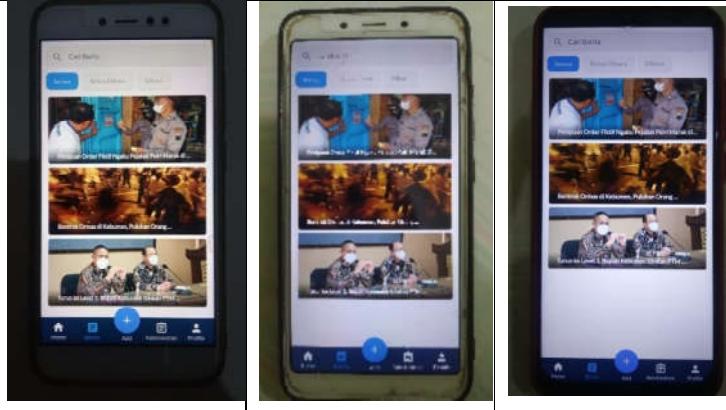
Xiaomi Redmi Note 5A	720 x 1280 pixels	7.1 (Android Nougat)
Xiaomi Redmi 6A	720 x 14400 pixels	8.1 (Android Oreo)
Vivo Y17	720 x 1544 pixels	9.0 (Android Pie)

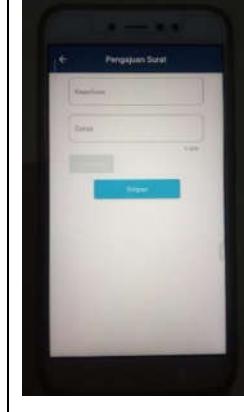
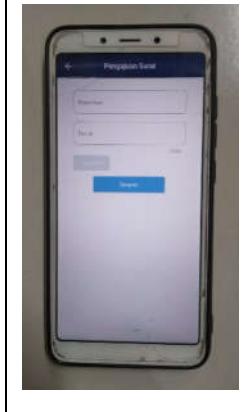
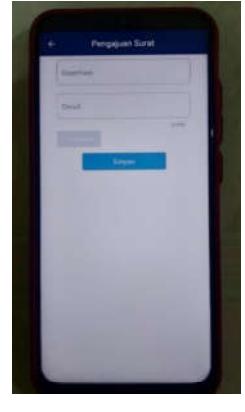
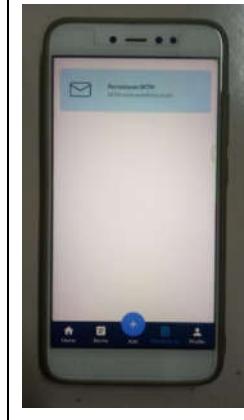
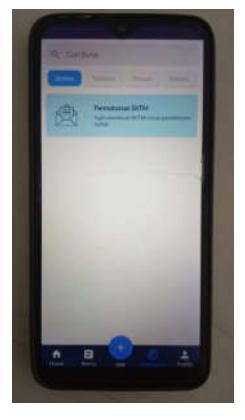
1) Hasil Rekap Uji Tampilan Aplikasi Android

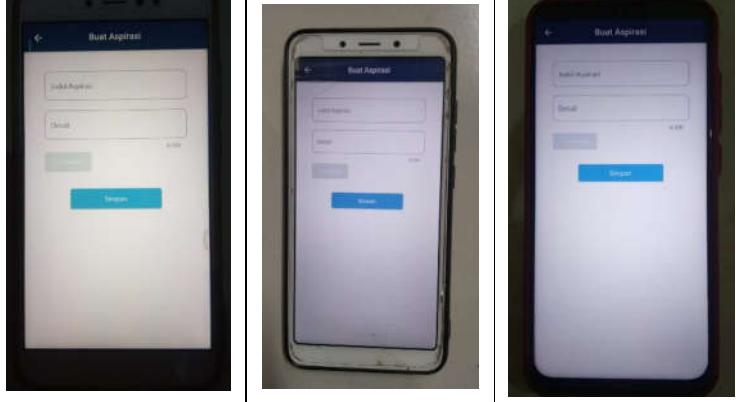
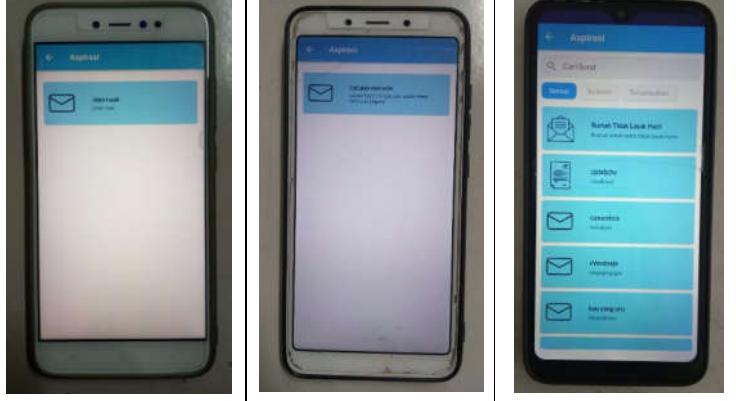
Berikut Tabel 4.2 merupakan hasil rekap pengujian tampilan aplikasi android pada beberapa perangkat android sesuai dengan yang tercantum pada Tabel 4.2.

Tabel 4. 2. Rekap Pengujian Tampilan Aplikasi Android

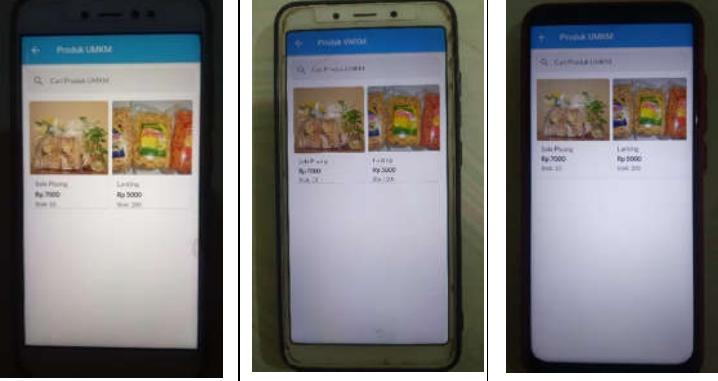
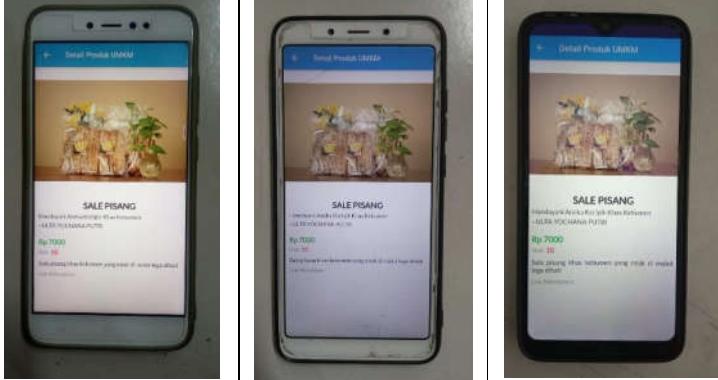
Pengujian	Smartphone Penguinji		
	Xiaomi Redmi Note 5A	Xiaomi Redmi 6A	Vivo Y17
Halaman <i>Login</i>			
Keterangan : Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.			
Halaman <i>Verifikasi</i>			

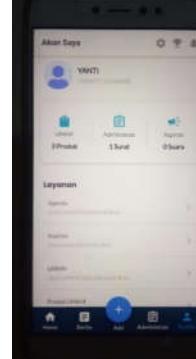
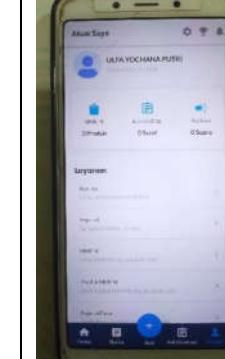
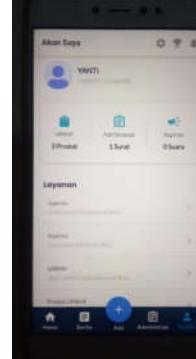
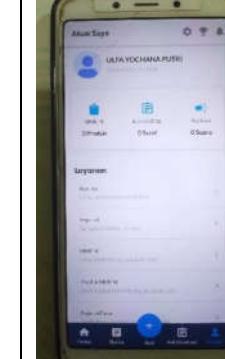
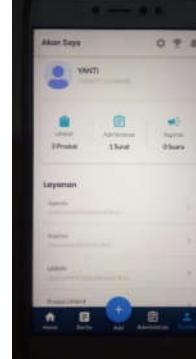
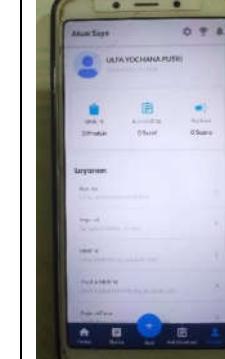
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>
Halaman Beranda	
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>
Halaman Berita	
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>

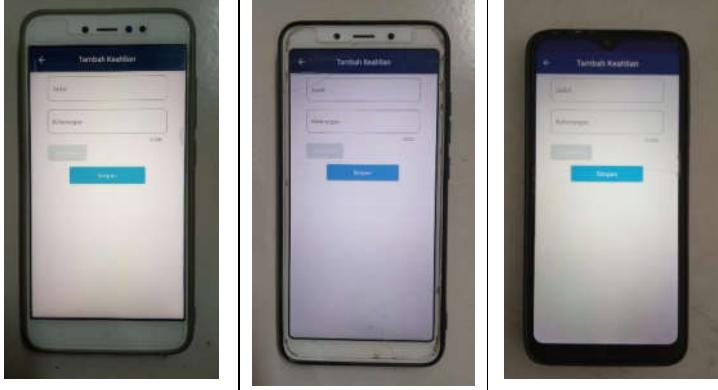
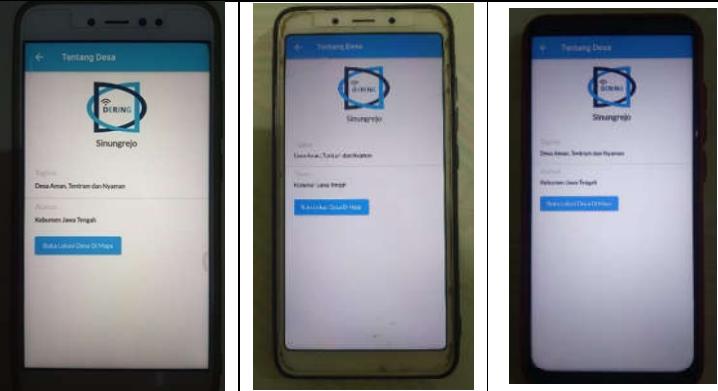
Halaman Detail Berita			
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>		
Halaman Surat			
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>		
Halaman Daftar Surat			

	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>
Halaman Aspirasi	
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>
Halaman Daftar Aspirasi	
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>

Halaman Agenda			
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>		
Halaman UMKM			
	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>		
Halaman Detail UMKM			

	Keterangan : Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.
Halaman Produk UMKM	
Halaman Detail Produk UMKM	<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>  <p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>

Halaman Profil			
<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>			
Halaman Ubah Nomor WhatsApp			
<p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>			
Halaman Keahlian			
<p>Keterangan :</p>			

	Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.
Halaman Tambah Keahlian	 <p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>
Halaman Informasi Desa	 <p>Keterangan :</p> <p>Pengujian tampilan berjalan dengan baik pada tiap perangkat android dan telah sesuai dengan tampilan yang direncanakan.</p>

2) Hasil Rekap Pengujian Fungsionalitas Aplikasi Android

1. Pengujian *Login User*

Tujuan : Pengguna dapat menuju halaman verifikasi

Cara Uji : Pengguna harus memasukan akun yang sudah terdaftar

sebelumnya dengan memasukan nomor WhatsApp.

2. Pengujian Verifikasi

Tujuan : Pengguna dapat menuju halaman beranda.

Cara Uji : Pengguna memasukan kode verifikasi yang telah dikirimkan melalui nomor WhatsApp yang diinputkan.

3. Pengujian Beranda

Tujuan : Pengguna dapat mengakses halaman beranda

Cara Uji : Pengguna memilih fitur yang ada pada halaman beranda.

4. Pengujian Berita

Tujuan : Pengguna dapat melihat daftar berita untuk mengetahui berita terbaru

Cara Uji : Pengguna memilih menu berita.

5. Pengujian Detail Berita

Tujuan : Pengguna dapat melihat detail berita untuk mengetahui berita terbaru

Cara Uji : Pengguna memilih berita yang terdapat pada halaman

6. Pengujian Surat

Tujuan : Pengguna dapat mengajukan surat

Cara Uji : Pengguna mengisi data pengajuan surat kemudian menyimpannya.

7. Pengujian Daftar Surat

Tujuan : Pengguna dapat melihat daftar surat

Cara Uji : Pengguna memilih daftar surat

8. Pengujian Aspirasi

Tujuan : Pengguna dapat mengajukan aspirasi

Cara Uji : Pengguna mengisi data pengajuan aspirasi

9. Pengujian Daftar Aspirasi

Tujuan : Pengguna dapat melihat daftar aspirasi surat

Cara Uji : Pengguna memilih daftar aspirasi

10. Pengujian Agenda

Tujuan : Pengguna dapat melihat agenda

Cara Uji : Pengguna memilih agenda yang terdapat pada halaman

11. Pengujian UMKM

Tujuan : Pengguna dapat menggunakan fitur UMKM

Cara Uji : Pengguna memilih menu UMKM

12. Pengujian Detail UMKM

Tujuan : Pengguna dapat melihat detail UMKM.

Cara Uji : Pengguna memilih UMKM.

13. Pengujian Produk UMKM

Tujuan : Pengguna dapat melihat produk UMKM desa

Cara Uji : Pengguna memilih menu produk UMKM

14. Pengujian Detail Produk UMKM

Tujuan : Pengguna dapat melihat detail dari produk UMKM yang dipilih

Cara Uji : Pengguna memilih produk UMKM.

15. Pengujian Profil

Tujuan : Pengguna dapat melihat profil dan menu-menu yang dapat diakses oleh pengguna

Cara Uji : Pengguna memilih menu profil.

16. Pengujian Ubah Nomor WhatsApp

Tujuan : Pengguna dapat mengubah nomor WhatsApp lama dengan nomor WhatsApp yang baru.

Cara Uji : Pengguna memasukan nomor WhatsApp baru beserta kode verifikasi yang dikirim.

17. Pengujian Keahlian

Tujuan : Pengguna dapat melihat keahlian yang dimiliki.

Cara Uji : Pengguna memilih menu keahlian.

18. Pengujian Tambah Keahlian

Tujuan : Pengguna dapat menambahkan data keahlian

Cara Uji : Pengguna mengisi data keahlian kemudian menyimpannya

19. Pengujian Informasi Desa

Tujuan : Pengguna dapat melihat informasi desa.

Cara Uji : Pengguna memilih menu tentang desa.

4.2.2. Rekapitulasi Uji Tampilan dan Fungsionalitas Website Administrator

Pengujian tampilan dan fungsionalitas *website* administrator dilakukan dengan membuka tampilan *website* melalui Google Chrome dan Mozilla Firefox.

1) Hasil Rekap Uji Tampilan Website Administrator

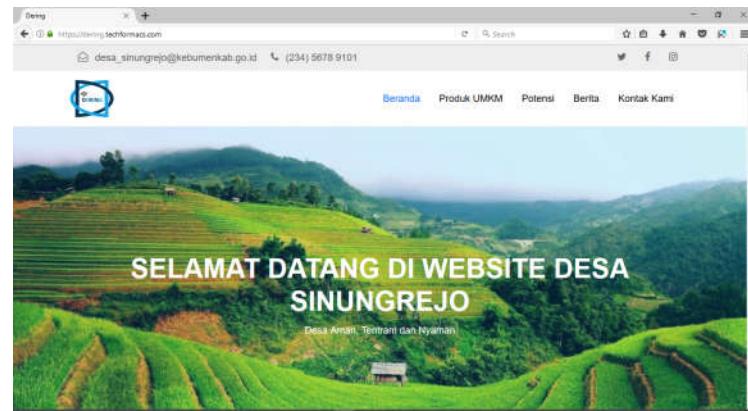
Hasil rekap tampilan *website* administrator adalah sebagai berikut.

1. Halaman Landing Page

Pengujian tampilan *landing page* berjalan dengan cukup baik pada tiap browsernya, halaman *landing page* terlihat sama baik di Google Chrome maupun Mozilla Firefox seperti yang ditunjukkan pada Gambar 4.3 dan Gambar 4.4.



Gambar 4.3. Halaman *Landing Page* pada Google Chrome

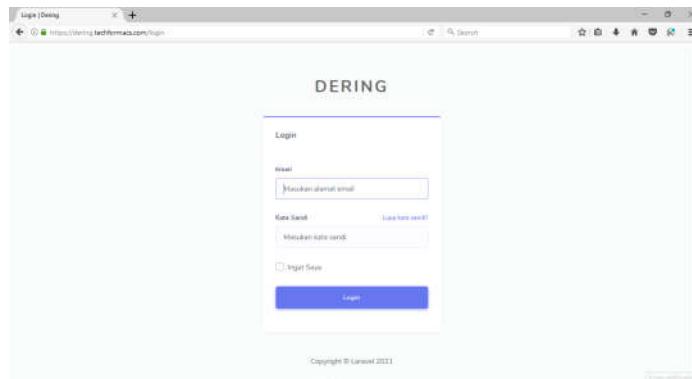


Gambar 4.4. Halaman *Landing Page* pada Mozilla Firefox

2. Halaman *Login* Administrator

Tampilan *login* administrator pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.5 dan Gambar 4.6.

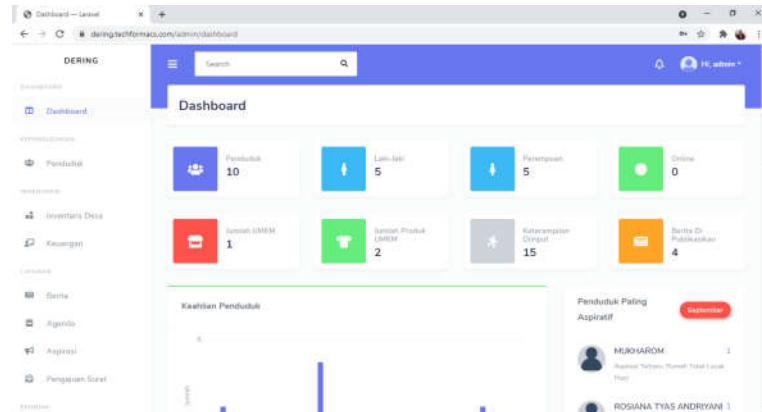
Gambar 4.5. Halaman *Login* pada Google Chrome



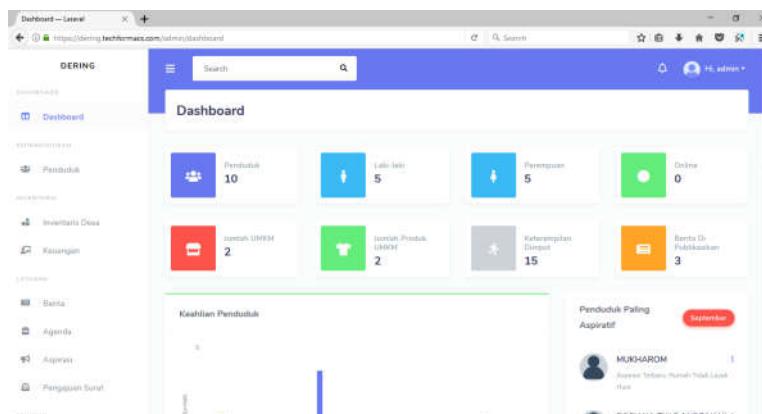
Gambar 4.6. Halaman *Login* pada Mozilla Firefox

3. Halaman *Dashboard*

Tampilan *dashboard* pada tiap browser berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.7 dan Gambar 4.8.



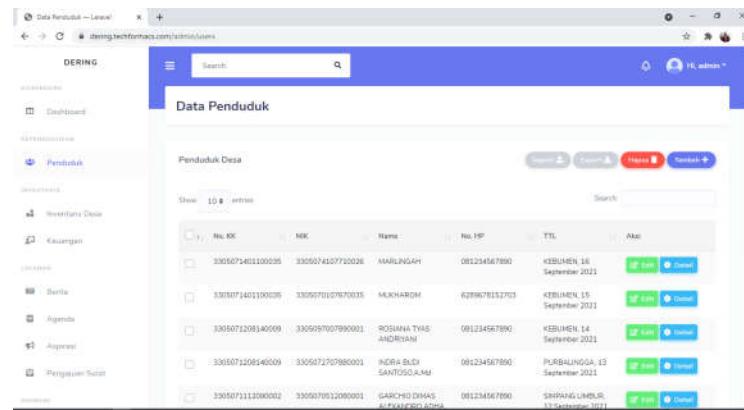
Gambar 4.7. Halaman *Dashboard* pada Google Chrome



Gambar 4.8. Halaman *Dashboard* pada Mozilla Firefox

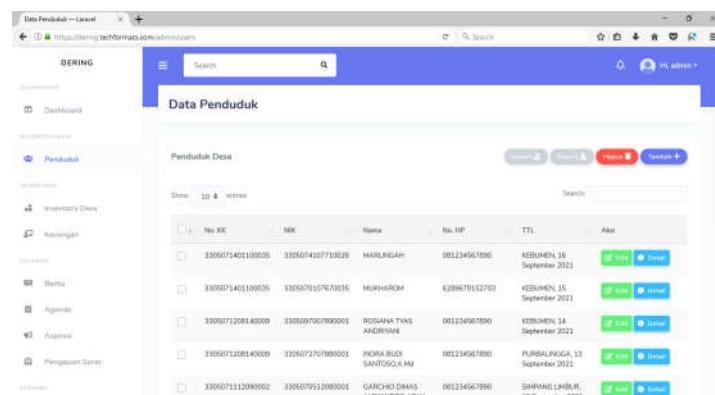
4. Halaman Manajemen Penduduk

Tampilan manajemen penduduk pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.9 dan Gambar 4.10.



	No. KK	NIK	Nama	No. HP	TTL	Aksi
<input type="checkbox"/>	3305075402100035	3305074107710036	MARLINGAH	081234567890	KEBUMEN, 16 September 2011	
<input type="checkbox"/>	3305071401100035	3305070107870035	MUHAMAD	6289678152703	KEBUMEN, 15 September 2011	
<input type="checkbox"/>	3305071208140009	3305097007990001	ROSHANA TYAS ANDRIYANI	081234567890	KEBUMEN, 14 September 2012	
<input type="checkbox"/>	3305071208140009	3305072707980001	INDRA BUDI SANTOSO, A.M.	081234567890	PURBALINGGA, 13 September 2011	
<input type="checkbox"/>	3305071112090002	3305070812080001	GARGHO DIHAS ALFAZAH	081234567890	SIMPANG LIMBIR, 17 September 2011	

Gambar 4.9. Halaman Manajemen Penduduk pada Google Chrome



	No. KK	NIK	Nama	No. HP	TTL	Aksi
<input type="checkbox"/>	3305071401100035	3305074107710036	MARLINGAH	081234567890	KEBUMEN, 16 September 2011	
<input type="checkbox"/>	3305071401100035	3305070107870035	MUHAMAD	6289678152703	KEBUMEN, 15 September 2011	
<input type="checkbox"/>	3305071208140009	3305097007990001	ROSHANA TYAS ANDRIYANI	081234567890	KEBUMEN, 14 September 2012	
<input type="checkbox"/>	3305071208140009	3305072707980001	INDRA BUDI SANTOSO, A.M.	081234567890	PURBALINGGA, 13 September 2011	
<input type="checkbox"/>	3305071112090002	3305070812080001	GARGHO DIHAS ALFAZAH	081234567890	SIMPANG LIMBIR, 17 September 2011	

Gambar 4.10. Halaman Manajemen Penduduk pada Mozilla Firefox

5. Halaman Tambah Data Penduduk

Tampilan tambah data penduduk pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.11 dan Gambar 4.12.

The screenshot shows a web browser window for 'DERRING'. The main content is a 'Tambah Data' (Add Data) form for 'Penduduk' (Resident). The form has a circular placeholder image for 'Foto Avatar'. Fields include 'Email' (with placeholder 'Email'), 'No. HP' (with placeholder '081234567890'), 'Nama Lengkap' (Full Name), 'Tanggal Lahir' (Date of Birth), and 'Ketika Nikah' (Date of Marriage). There are also fields for 'NIK' and 'NK'.

Gambar 4.11. Halaman Tambah Data Penduduk pada Google Chrome

This screenshot is identical to the one above, showing the 'Tambah Data Penduduk' form in Mozilla Firefox. The fields and overall layout are the same, including the placeholder images and text inputs.

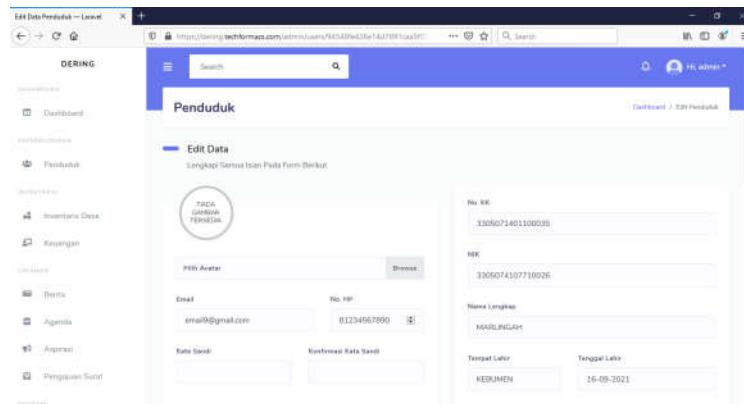
Gambar 4.12. Halaman Tambah Data Penduduk pada Mozilla Firefox

6. Halaman Edit Data Penduduk

Tampilan edit data penduduk pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.13 dan Gambar 4.14.

The screenshot shows the 'Edit Data' form for 'Penduduk'. It displays the following data:
 - NIK: 330807140110001
 - NK: 3308074107710006
 - Nama Lengkap: MARJUNAH
 - Tempat Lahir: KEBUMEN
 - Tanggal Lahir: 16-08-2021
 - Email: wrwrt@gmail.com
 - No. HP: 081234567890
 - Kata Sandi: (empty)
 - Konfirmasi Kata Sandi: (empty)

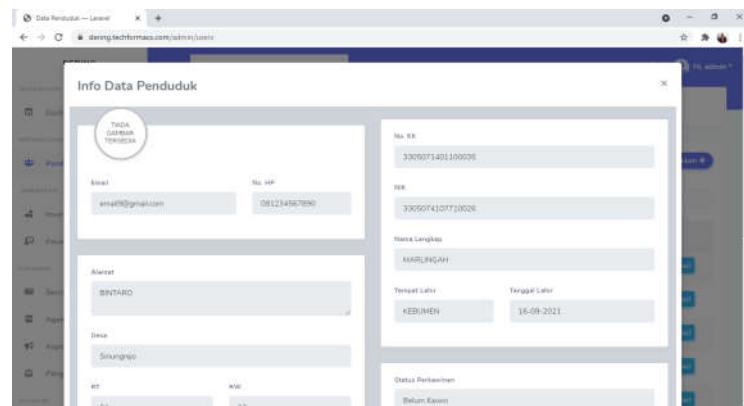
Gambar 4.13. Halaman Edit Data Penduduk pada Google Chrome



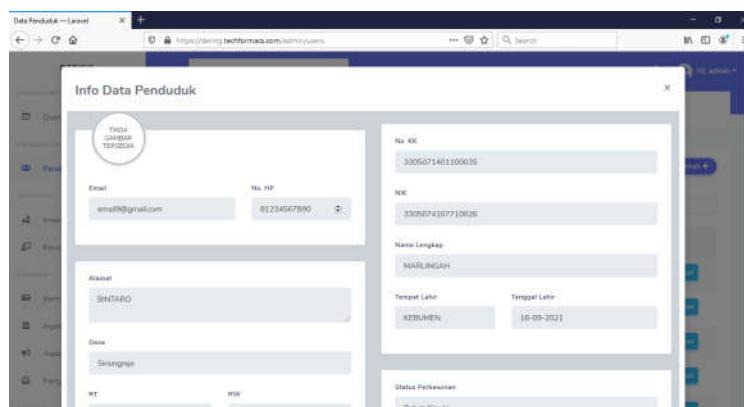
Gambar 4.14. Halaman Edit Data Penduduk pada Mozilla Firefox

7. Halaman Detail Data Penduduk

Tampilan detail data penduduk pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.15 dan Gambar 4.16.



Gambar 4.15. Halaman Detail Data Penduduk pada Google Chrome



Gambar 4.16. Halaman Detail Data Penduduk pada Mozilla Firefox

8. Halaman Manajemen Inventaris Desa

Tampilan manajemen inventaris desa pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.17 dan Gambar 4.18.

Type	Nama	Ukuran	Jumlah	Status	Aksi
Alat	Gitar	-	50	Baik	
Alat	Piring	-	100	Baik	
Alat Kebutuhan	Sepu	-	5	Baik	

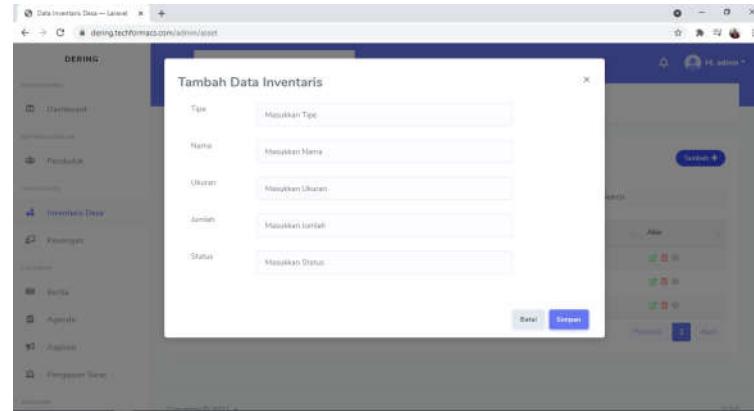
Gambar 4.17. Halaman Manajemen Inventaris pada Google Chrome

Type	Nama	Ukuran	Jumlah	Status	Aksi
Alat	Gitar	-	50	Baik	
Alat	Piring	-	100	Baik	
Alat Kebutuhan	Sepu	-	5	Baik	

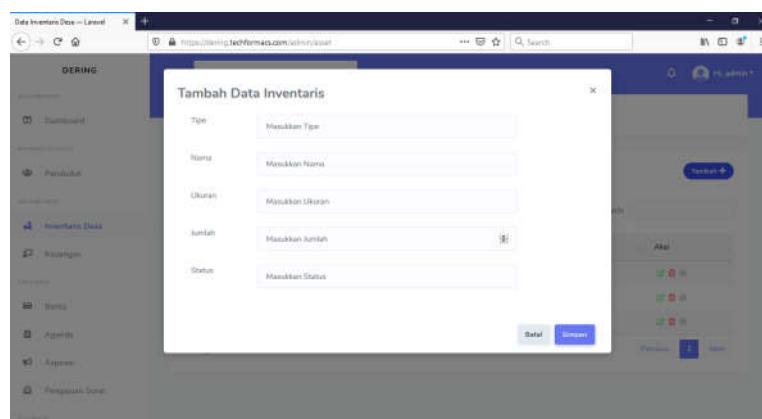
Gambar 4.18. Halaman Manajemen Inventaris pada Mozilla Firefox

9. Halaman Tambah Data Inventaris

Tampilan detail data inventaris pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.19 dan Gambar 4.20.



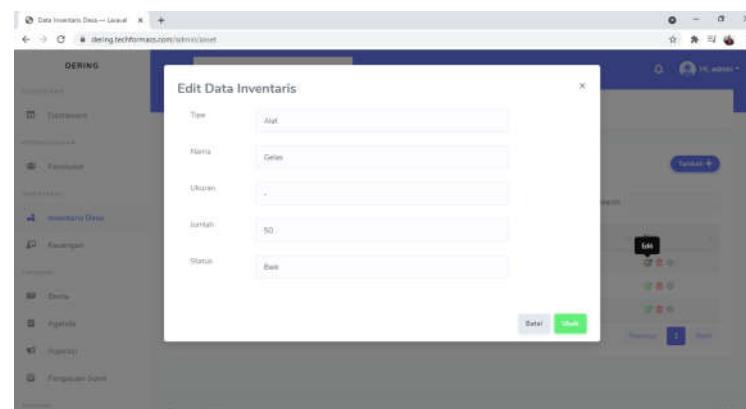
Gambar 4.19. Halaman Manajemen Inventaris pada Google Chrome



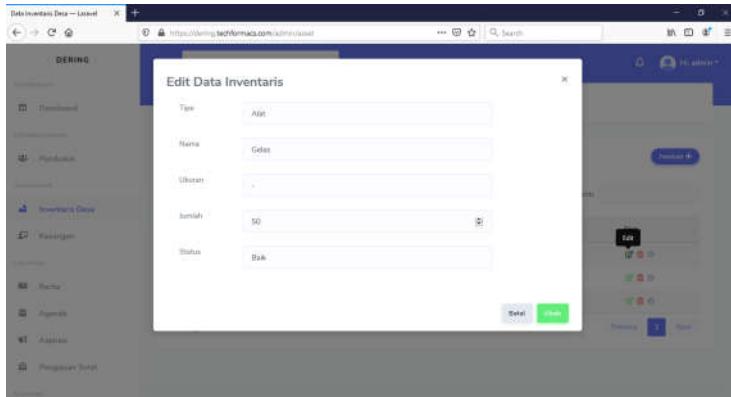
Gambar 4.20. Halaman Manajemen Inventaris pada Mozilla Firefox

10. Halaman Edit Data Inventaris

Tampilan edit data inventaris pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.21 dan Gambar 4.22.



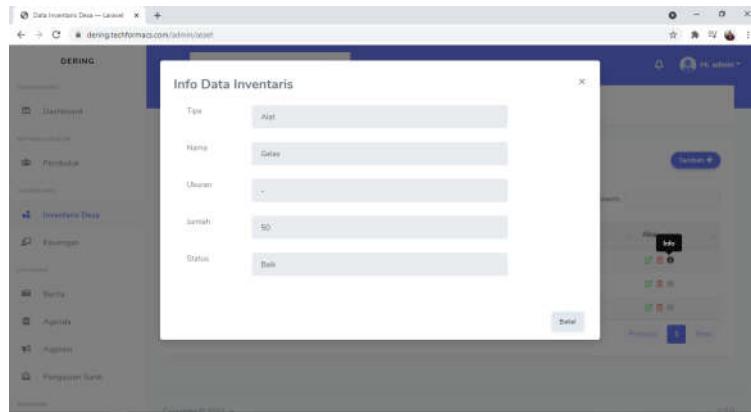
Gambar 4.21. Halaman Edit Data Inventaris pada Google Chrome



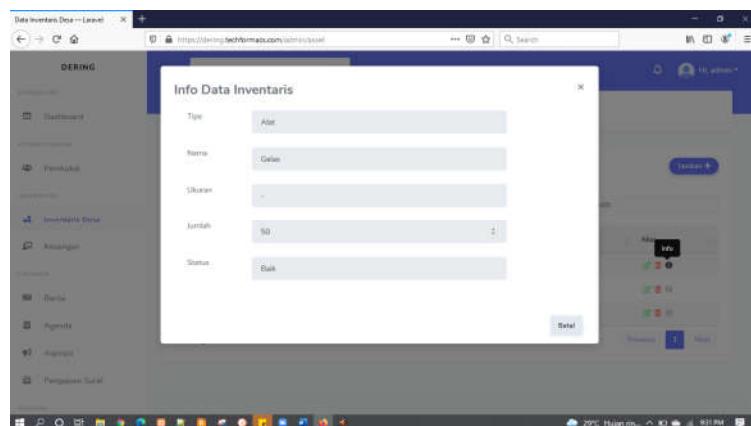
Gambar 4.22. Halaman Edit Data Inventaris pada Mozilla Firefox

11. Halaman Detail Data Inventaris

Tampilan detail data inventaris pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.23 dan Gambar 4.24.



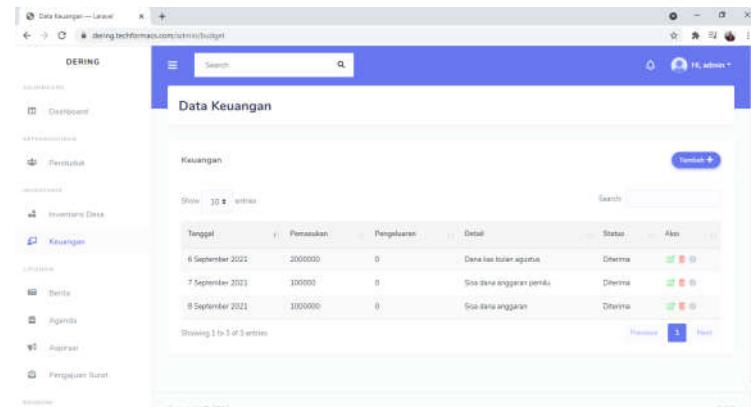
Gambar 4.23. Halaman Detail Data Inventaris pada Google Chrome



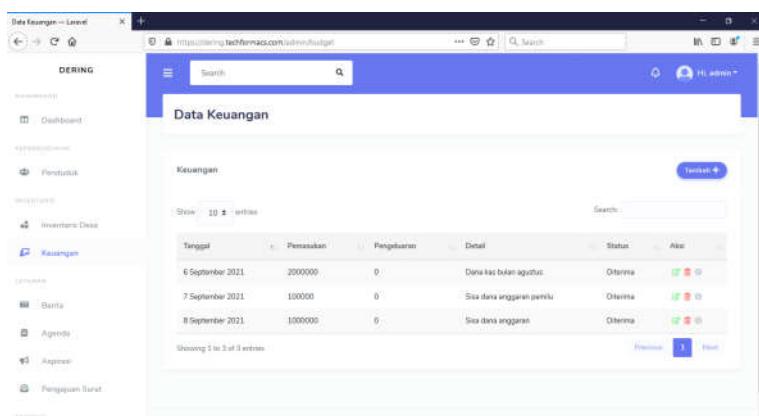
Gambar 4.24. Halaman Detail Data Inventaris pada Mozilla Firefox

12. Halaman Manajemen Data Keuangan

Tampilan manajemen data keuangan pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.25 dan Gambar 4.26.



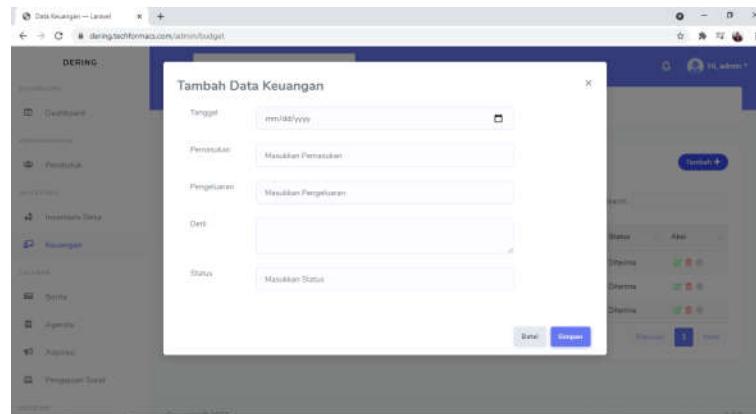
Gambar 4.25. Halaman Manajemen Keuangan pada Google Chrome



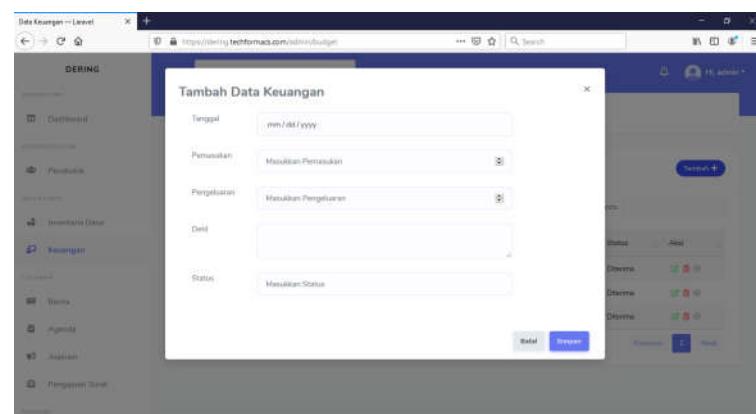
Gambar 4.26. Halaman Manajemen Keuangan pada Mozilla Firefox

13. Halaman Tambah Data Keuangan

Tampilan tambah data keuangan pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.27 dan Gambar 4.28.



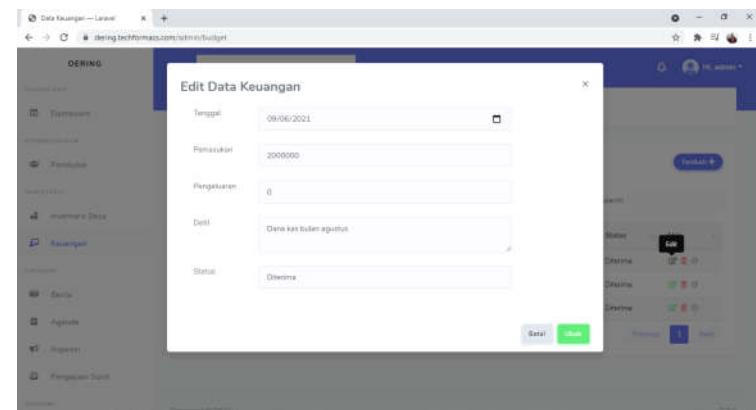
Gambar 4.27. Halaman Tambah Data Keuangan pada Google Chrome



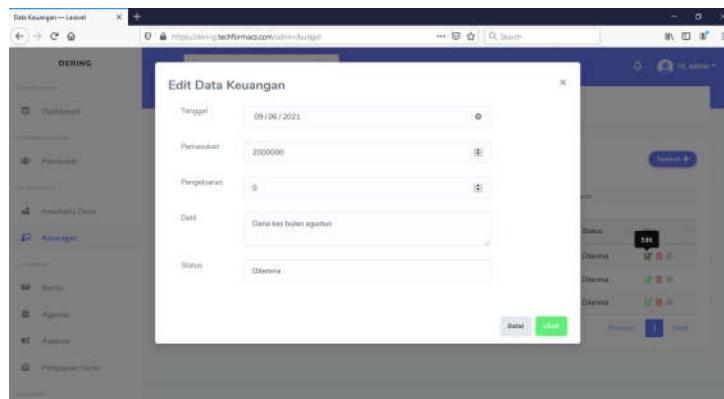
Gambar 4.28. Halaman Tambah Data Keuangan pada Mozilla Firefox

14. Halaman Edit Data Keuangan

Tampilan edit data keuangan pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.29 dan Gambar 4.30.



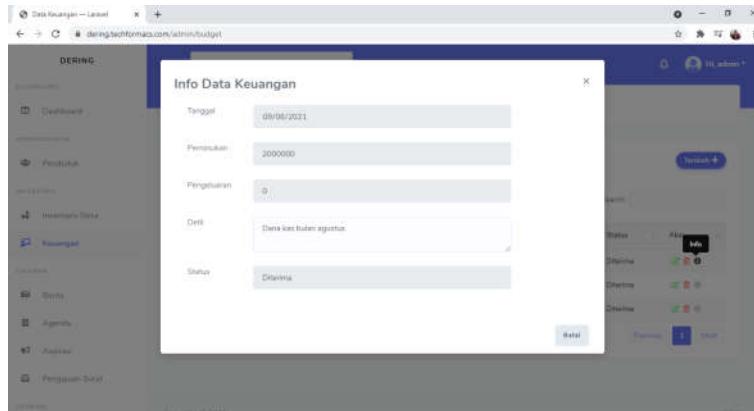
Gambar 4.29. Halaman Edit Data Keuangan pada Google Chrome



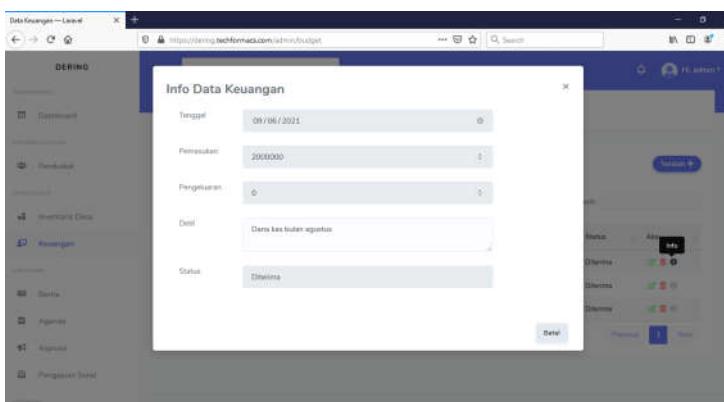
Gambar 4.30. Halaman Edit Data Keuangan pada Mozilla Firefox

15. Halaman Detail Data Keuangan

Tampilan detail data keuangan pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.31 dan Gambar 4.32.



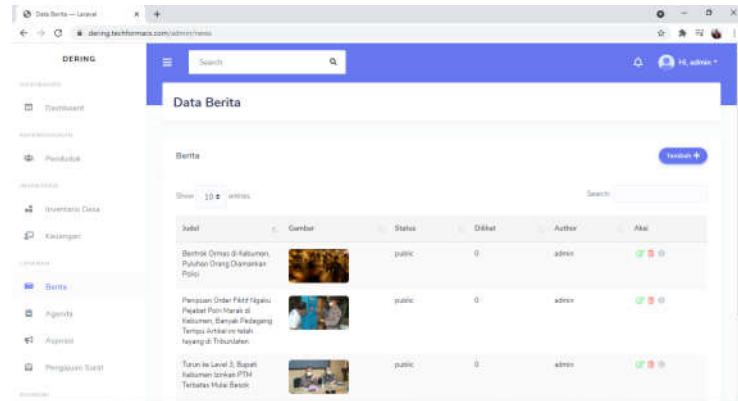
Gambar 4.31. Halaman Detail Data Keuangan pada Google Chrome



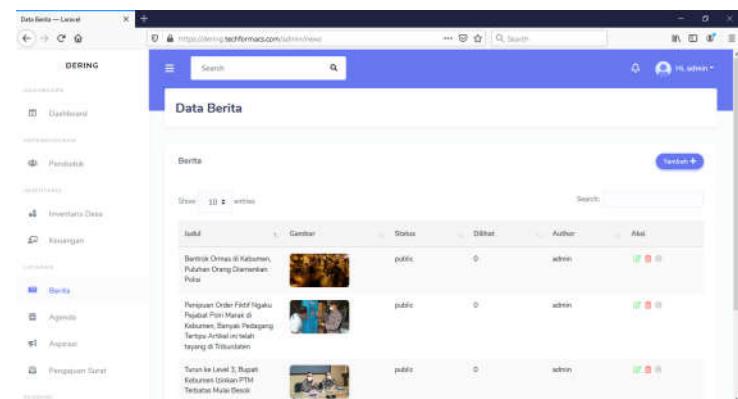
Gambar 4.32. Halaman Detail Data Keuangan pada Mozilla Firefox

16. Halaman Manajemen Berita

Tampilan manajemen berita pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.33 dan Gambar 4.34.



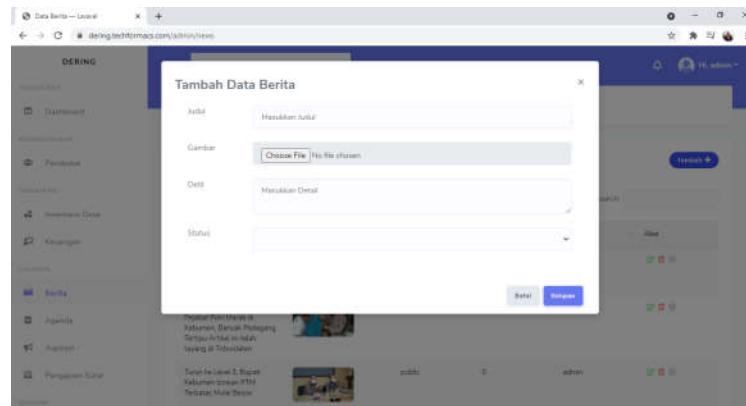
Gambar 4.33. Halaman Manajemen Berita pada Google Chrome



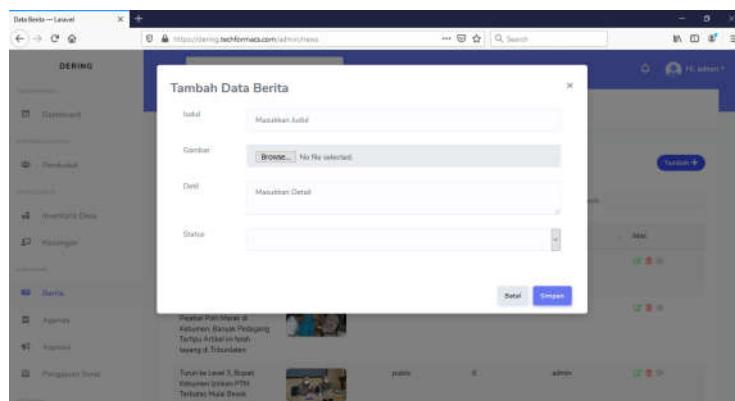
Gambar 4.34. Halaman Manajemen Berita pada Mozilla Firefox

17. Halaman Tambah Data Berita

Tampilan tambah data berita pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.35 dan Gambar 4.36.



Gambar 4.35. Halaman Tambah Data Berita pada Google Chrome



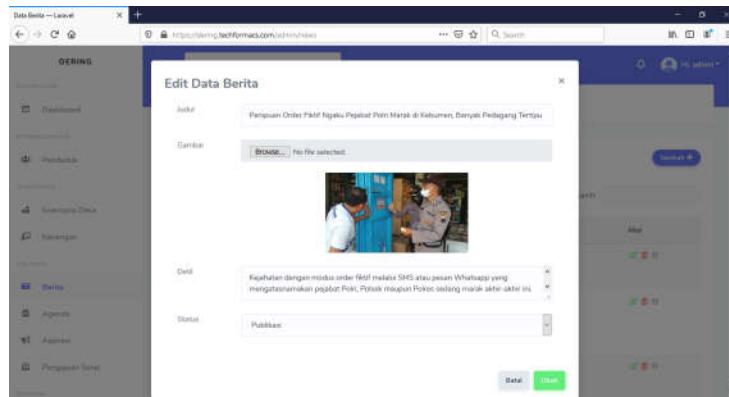
Gambar 4.36. Halaman Tambah Data Berita pada Mozilla Firefox

18. Halaman Edit Data Berita

Tampilan edit data berita pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.37 dan Gambar 4.38.



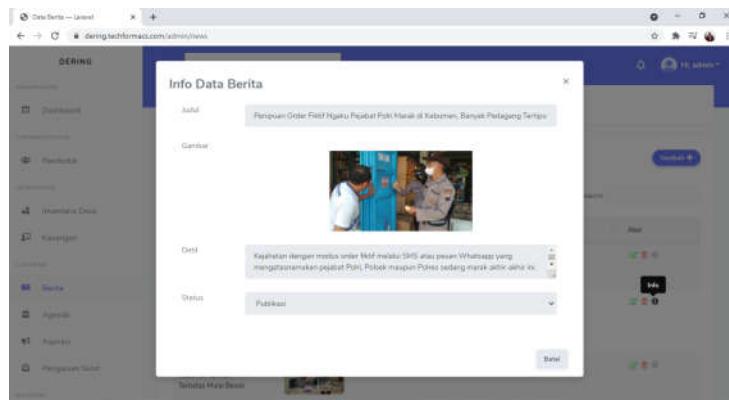
Gambar 4.37. Halaman Edit Data Berita pada Google Chrome



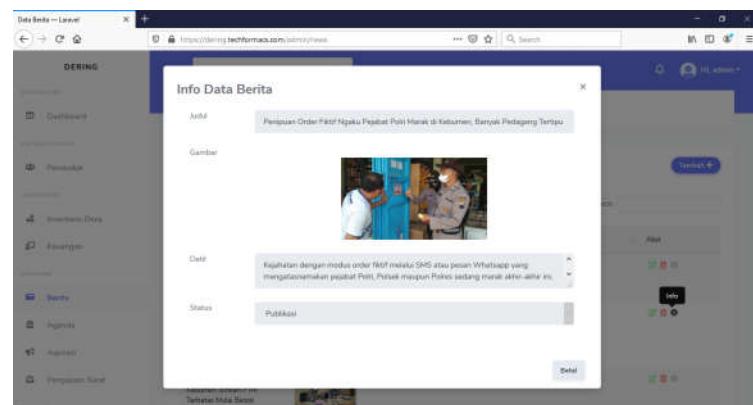
Gambar 4.38. Halaman Edit Data Berita pada Mozilla Firefox

19. Halaman Detail Berita

Tampilan detail berita pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.39 dan Gambar 4.40.



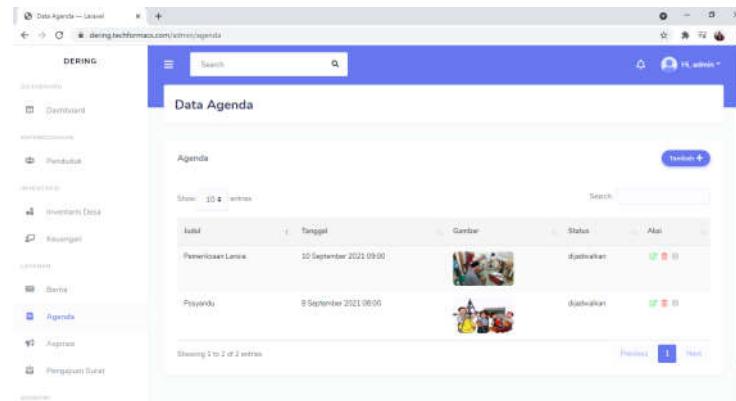
Gambar 4.39. Halaman Detail Berita pada Google Chrome



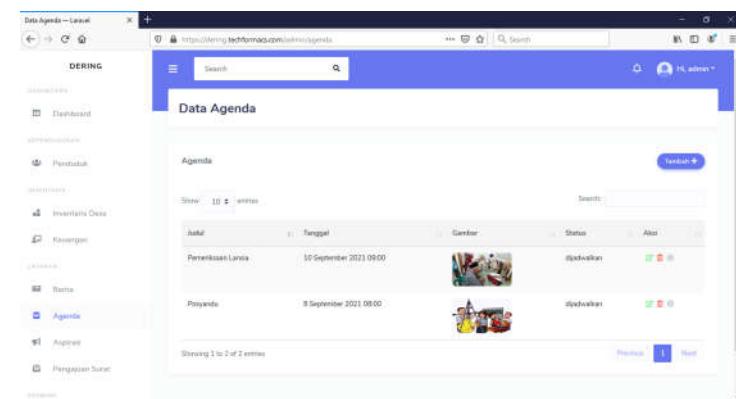
Gambar 4.40. Halaman Detail Berita pada Mozilla Firefox

20. Halaman Manajemen Agenda

Tampilan manajemen agenda pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.41 dan Gambar 4.42.



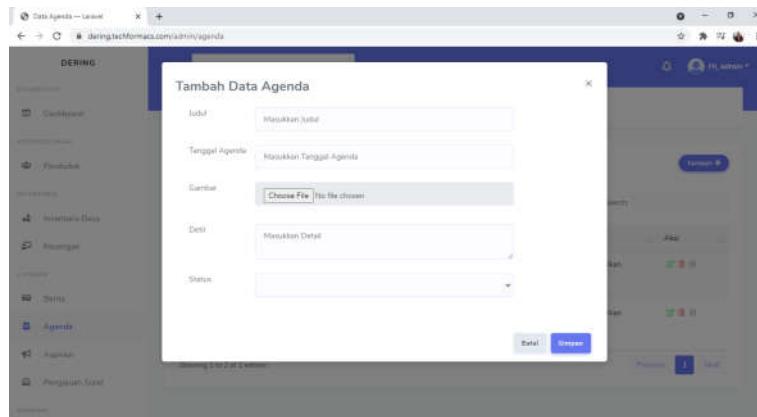
Gambar 4.41. Halaman Manajemen Agenda pada Google Chrome



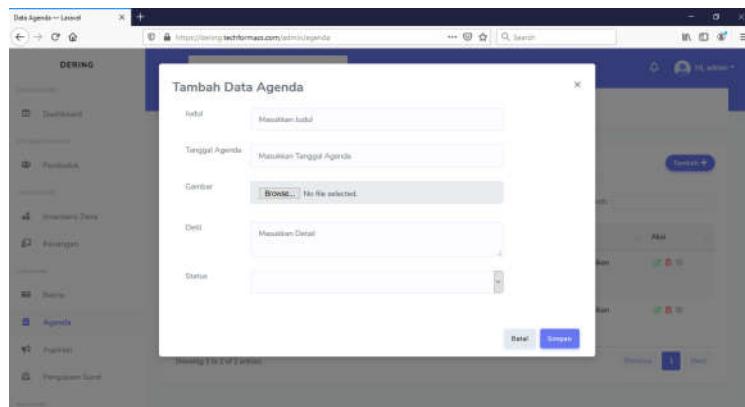
Gambar 4.42. Halaman Manajemen Agenda pada Mozilla Firefox

21. Halaman Tambah Data Agenda

Tampilan tambah data agenda pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.43 dan Gambar 4.44.



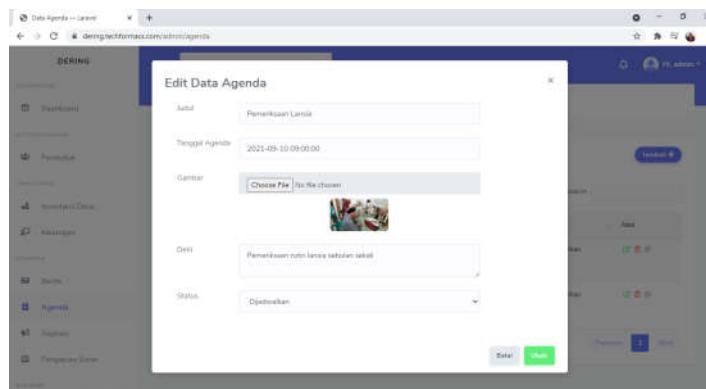
Gambar 4.43. Halaman Tambah Data Agenda pada Google Chrome



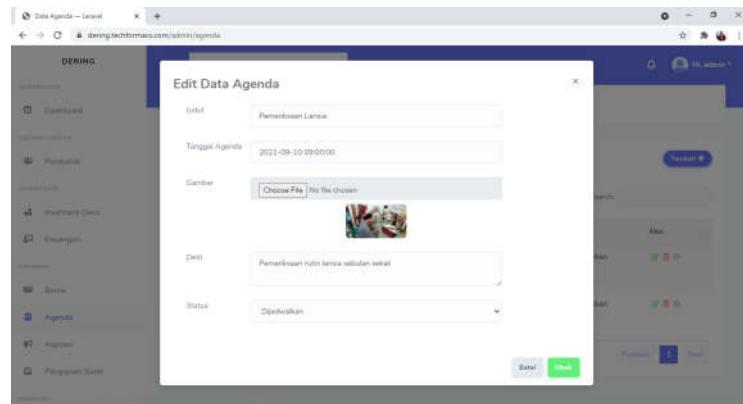
Gambar 4.44. Halaman Tambah Data Agenda pada Mozilla Firefox

22. Halaman Edit Data Agenda

Tampilan edit data agenda pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.45 dan Gambar 4.46.



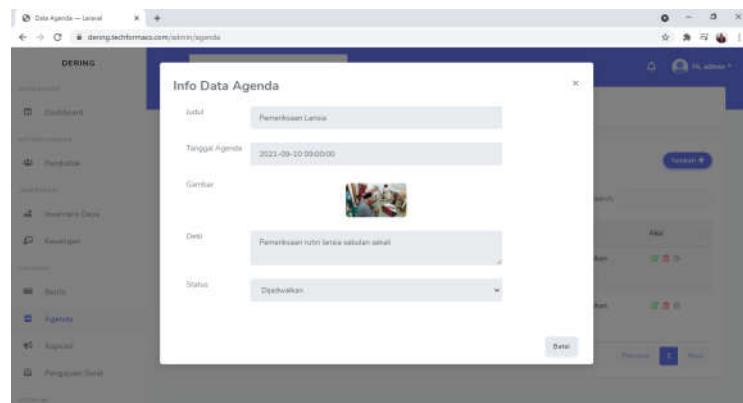
Gambar 4.45. Halaman Edit Data Agenda pada Google Chrome



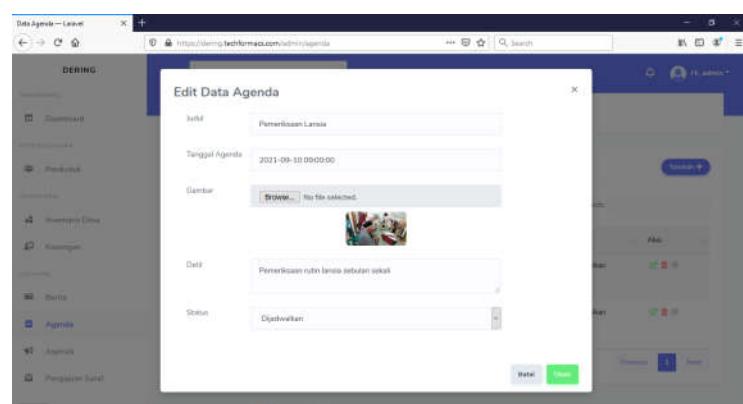
Gambar 4.46. Halaman Tambah Data Agenda pada Mozilla Firefox

23. Halaman Detail Agenda

Tampilan detail agenda pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.47 dan Gambar 4.48.



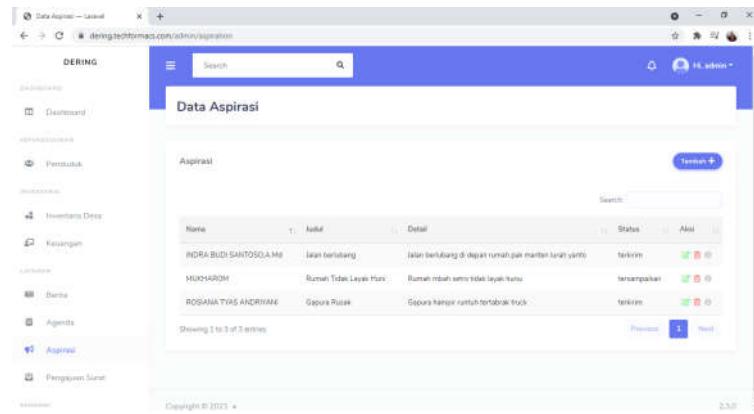
Gambar 4.47. Halaman Detail Agenda pada Google Chrome



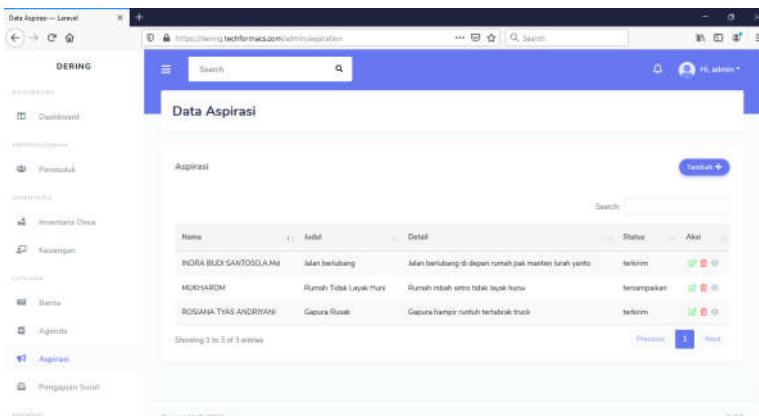
Gambar 4.48. Halaman Detail Agenda pada Mozilla Firefox

24. Halaman Manajemen Aspirasi

Tampilan manajemen aspirasi pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.49 dan Gambar 4.50.



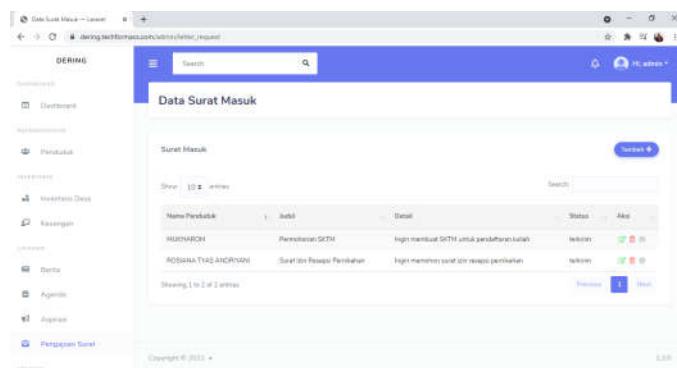
Gambar 4.49. Halaman Manajemen Aspirasi pada Google Chrome



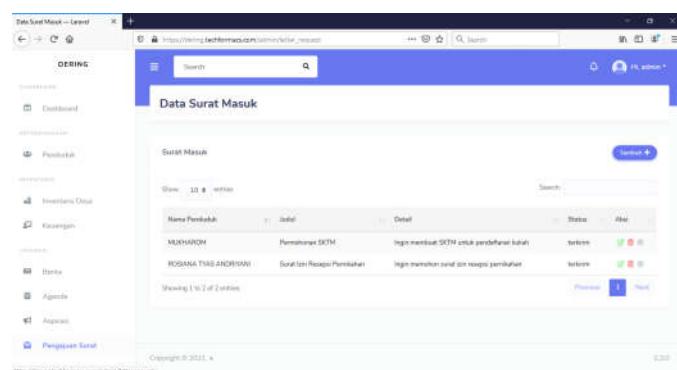
Gambar 4.50. Halaman Manajemen Aspirasi pada Mozilla Firefox

25. Halaman Manajemen Pengajuan Surat

Tampilan manajemen pengajuan surat pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.51 dan Gambar 4.52.



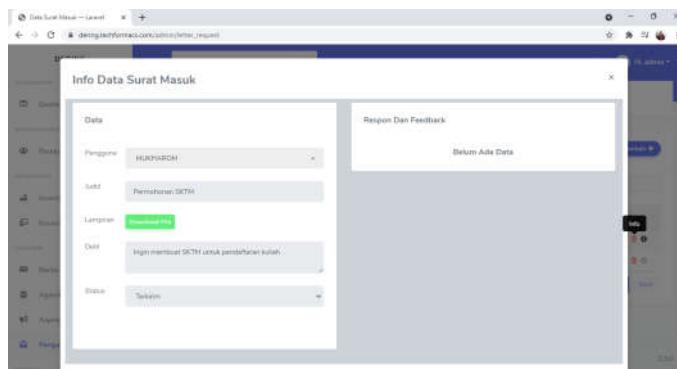
Gambar 4.51. Halaman Manajemen Surat pada Google Chrome



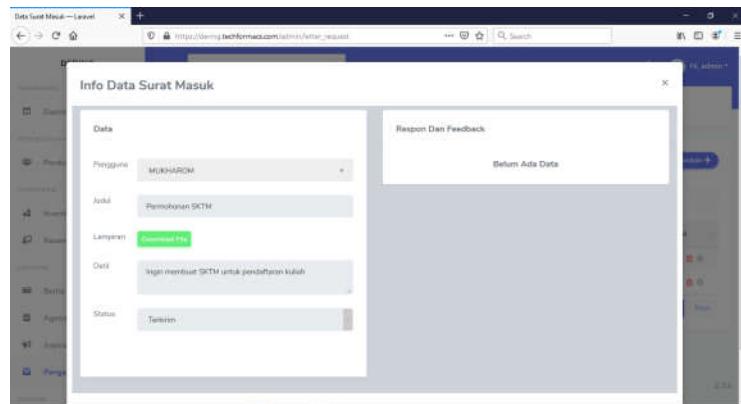
Gambar 4.52. Halaman Manajemen Surat pada Mozilla Firefox

26. Halaman Detail Surat

Tampilan detail surat pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.53 dan Gambar 4.54.



Gambar 4.53. Halaman Detail Surat pada Google Chrome



Gambar 4.54. Halaman Detail Surat pada Mozilla Firefox

27. Halaman Manajemen UMKM

Tampilan manajemen UMKM pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.55 dan Gambar 4.56.

Penulis	Nama UMKM	Gender	Lokasi	Koordinat	Jumlah Produk	Status	Aksi
	Handoyo Aneka Keripik Khas Kulon		Jl. Nasional II, Cenglok, Purwodadi, Kec. Sreng, Kabupaten Kulonprogo, Jawa Tengah 54362		2	diterima	
	Kaos Ngapak Kulon		Jl. Raya Ngapak, Karangturi, Sanogadung, Kec. Mert, Kabupaten Kulonprogo, Jawa Tengah 54395		0	proses	

Gambar 4.55. Halaman Manajemen UMKM pada Google Chrome

Penulis	Nama UMKM	Gender	Lokasi	Koordinat	Jumlah Produk	Status	Aksi
	Handoyo Aneka Keripik Khas Kulon		Jl. Nasional II, Cenglok, Purwodadi, Kec. Sreng, Kabupaten Kulonprogo, Jawa Tengah 54362		2	diterima	
	Kaos Ngapak Kulon		Jl. Raya Ngapak, Karangturi, Sanogadung, Kec. Mert, Kabupaten Kulonprogo, Jawa Tengah 54395		0	proses	

Gambar 4.56. Halaman Manajemen UMKM pada Mozilla Firefox

28. Halaman Manajemen Produk UMKM

Tampilan manajemen produk UMKM pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.57 dan Gambar 4.58.

Nama UMKM	Jenis Produk	Detail	Gambar	Harga	Stock	Diketahui	Aksi
Handycare Aneka Kuek Khas Keluasan	Lanting	Camilan khas keluasan		5000	200	0	
Handycare Aneka Kuek Khas Keluasan	Sale Pisang	Sale pisang atau keluasan yang enak & mudah bagi ibu		7000	10	0	

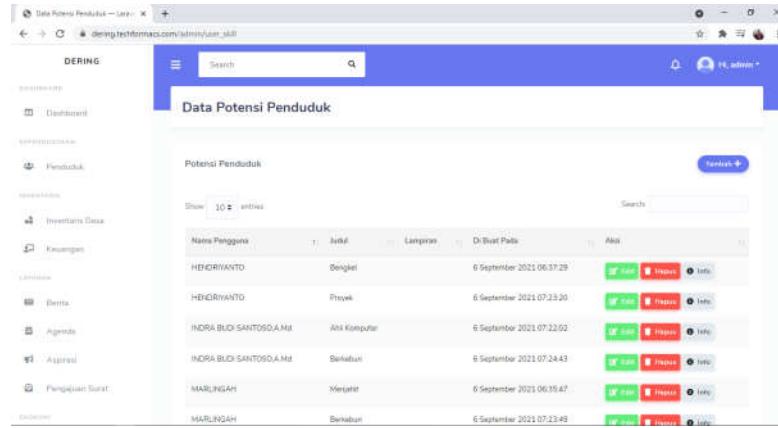
Gambar 4.57. Halaman Manajemen Produk pada Google Chrome

Nama UMKM	Jenis Produk	Detail	Gambar	Harga	Stock	Diketahui	Aksi
Handycare Aneka Kuek Khas Keluasan	Lanting	Camilan khas keluasan		5000	200	0	
Handycare Aneka Kuek Khas Keluasan	Sale Pisang	Sale pisang atau keluasan yang enak & mudah bagi ibu		7000	10	0	

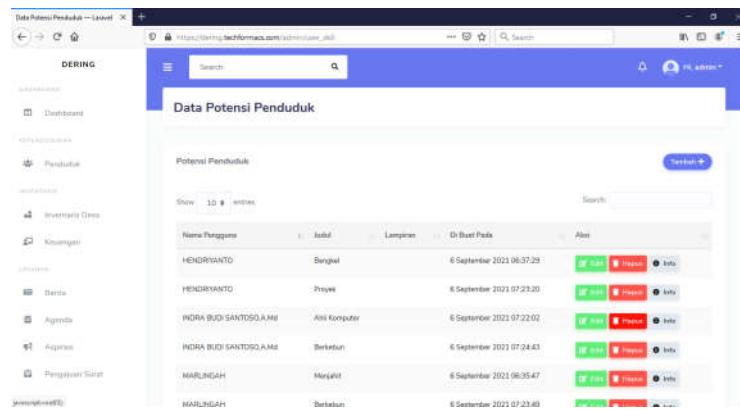
Gambar 4.58. Halaman Manajemen Produk pada Mozilla Firefox

29. Halaman Manajemen Keahlian

Tampilan manajamen keahlian pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.59 dan Gambar 4.60.



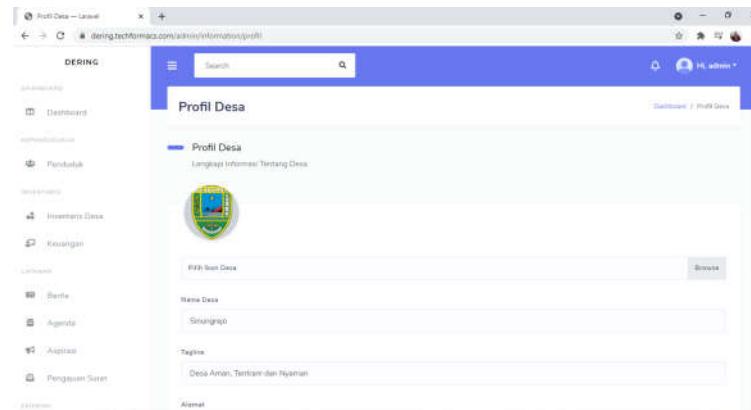
Gambar 4.59. Halaman Manajemen Keahlian pada Google Chrome



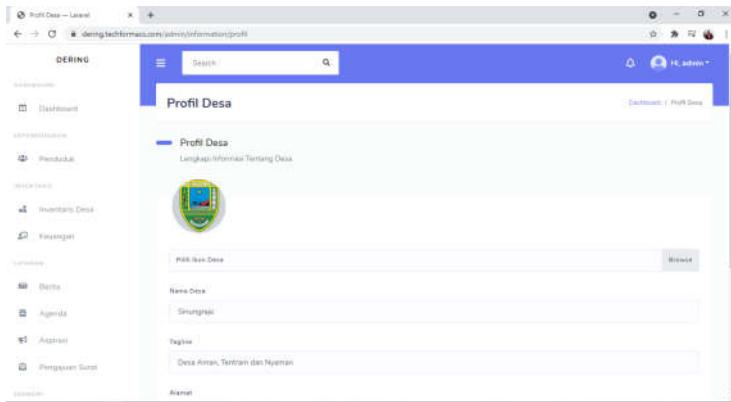
Gambar 4.60. Halaman Manajemen Keahlian pada Mozilla Firefox

30. Halaman Manajemen Profil

Tampilan manajemen profil pada tiap *browser* berjalan dengan baik seperti yang ditunjukkan pada Gambar 4.61 dan Gambar 4.62.



Gambar 4.61. Halaman Manajemen Profil pada Google Chrome



Gambar 4.62. Halaman Manajemen Profil pada Mozilla Firefox

2) Hasil Rekapitulasi Uji Fungsionalitas Website Administrator

1. Pengujian *Landing Page*

- Tujuan : Masyarakat luas dapat mengakses halaman *landing page*.
 Cara Uji : Masyarakat mengunjungi alamat website yang diarahkan pada halaman *landing page*.

2. Pengujian *Login* Administrator

- Tujuan : Administrator dapat mengakses semua halaman.
 Cara Uji : Administrator memasukan *email* dan *password* pada halaman *login*.

3. Pengujian Dashboard

- Tujuan : Administrator dapat mengetahui kalkulasi jumlah data yang masuk.
 Cara Uji : Administrator berhasil memasukan *email* dan *password* pada halaman *login*.

4. Pengujian Manajemen Penduduk

- Tujuan : Administrator dapat mengolah data penduduk.
 Cara Uji : Administrator melakukan *import* data, tambah data, edit data, melihat detail data, cari data dan hapus data penduduk.

5. Pengujian Manajemen Inventaris

Tujuan : Administrator dapat mengolah data inventaris.

Cara Uji : Administrator melakukan tambah data, edit data, melihat detail data, cari data dan hapus data inventaris.

6. Pengujian Manajemen Keuangan

Tujuan : Administrator dapat mengolah data keuangan.

Cara Uji : Administrator melakukan tambah data, edit data, melihat detail data, cari data dan hapus data keuangan.

7. Pengujian Manajemen Berita

Tujuan : Administrator dapat mengolah berita.

Cara Uji : Administrator melakukan tambah berita, edit berita, melihat detail berita, cari berita dan hapus berita berita.

8. Pengujian Manajemen Agenda

Tujuan : Administrator dapat mengolah agenda.

Cara Uji : Administrator melakukan tambah agenda, edit agenda, melihat detail agenda, cari agenda dan hapus berita agenda.

9. Pengujian Manajemen Aspirasi

Tujuan : Administrator dapat mengelola aspirasi

Cara Uji : Administrator mengubah status aspirasi.

10. Pengujian Manajemen Pengajuan Surat

Tujuan : Administrator dapat mengelola surat

Cara Uji : Administrator mengubah status surat dan memberikan *feedback*.

11. Pengujian Manajemen UMKM

Tujuan : Administrator dapat mengelola UMKM

Cara Uji : Administrator melakukan verifikasi terhadap UMKM yang diajukan *user*.

12. Pengujian Manajemen Produk UMKM

Tujuan : Administrator dapat mengelola produk UMKM
 Cara Uji : Administrator memantau produk UMKM yang ditambahkan *user*.

13. Pengujian Manajemen Keahlian

Tujuan : Administrator dapat mengelola keahlian masyarakat desa.
 Cara Uji : Administrator memantau keahlian yang ditambahkan *user*.

14. Pengujian Manajemen Profil

Tujuan : Administrator dapat melihat dan memperbarui data informasi desa.
 Cara Uji : Administrator mengisi data informasi desa pada menu profil.

4.2.3. Pengujian Tingkat Kepuasan Pengguna

Pengujian ini dilakukan pada tanggal 01 sampai 09 September 2021 pada pemerintah desa, pelaku UMKM dan penduduk desa. Berikut adalah pertanyaan yang diajukan kepada 10 responden yang berbeda.

- 1) Seberapa puas kebermanfaatan aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing*?
- 2) Seberapa puas kebermanfaatan fitur aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing*?
- 3) Seberapa puas kemudahan penggunaan aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing*?
- 4) Seberapa puas tampilan aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing*?

- 5) Seberapa puas pengalaman pengguna aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing*?
- 6) Seberapa puas kebermanfaatan aplikasi Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing* dalam menciptakan masyarakat digital?

Berikut adalah hasil penilaian responden mengenai Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing* dari 10 responden yang berbeda, seperti yang ditunjukkan pada Tabel 4.5.

Tabel 4. 5 Hasil Kuisioner Responden

No.	Nama Responden	Keterangan	Pertanyaan						Total Nilai
			1	2	3	4	5	6	
1.	Sutadi	Perangkat Desa	4	4	5	4	5	5	27
2.	Suripto	Perangkat Desa	4	4	4	4	4	4	24
3.	Nur Muhammad	Pelaku UMKM	5	4	4	5	5	5	28
4.	Nor Rahma	Pelaku UMKM	4	5	5	4	4	5	27
5.	Nur Azizah	Pelaku UMKM	5	4	5	5	5	5	29
6.	Arifah Nur Hidayah	Penduduk Desa	4	5	5	4	5	5	28
7.	Syafiq Fajrun	Penduduk Desa	4	4	4	4	4	5	25
8.	Heri Kurniawan	Penduduk Desa	5	4	4	4	4	4	26
9.	Grevi Finggun	Penduduk Desa	4	5	5	4	4	4	26

10.	Ika Bekti	Penduduk Desa	5	4	4	4	4	25
Total								265

Jumlah nilai maksimal untuk indikator penilaian = $5 \times 6 = 30$

Jumlah responden yang mengisi kuesioner = 10

Jumlah kepuasan maksimal = $10 \times 30 = 300$

$$\begin{aligned} \text{Skoring Kepuasan} &= \frac{\text{Total Kepuasan Responden}}{\text{Total Kepuasan Maksimum Responden}} \times 100\% \\ &= \frac{265}{300} \times 100\% = 88,3\% \end{aligned}$$

Dari hasil jajak pendapat pada Tabel 4.5 didapatkan hasil bahwa tingkat kepuasan pengguna adalah 88,3%. Dari hasil perhitungan tingkat kepuasan pengguna tersebut berdasarkan Tabel 3.52 didapatkan indikator penilaian sangat memuaskan.

Penelitian dengan judul “Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa Dengan Teknik *Crowdsourcing*” memiliki kesamaan fitur dengan semua penelitian yang sudah dilakukan sebelumnya yaitu fitur layanan administrasi dan informasi (Imaniawan dan Wati, 2017), (Rachman dan Noviyanto, 2017), (Muntofiah dan Fitrian, 2017), (Syaharuddin 2017), (Rozi dan Listiawan, 2017), (Usmanto et al. 2018), (Pratiwi dan Muslihudin, 2018), (Samsudin dan Muslihudin, 2018), (Susilowati dan Chandra, 2018), (Setyowati, Anton dan Radiyah, 2019), (Soraya 2019), (Jimi 2019), (Putra, Hidayatullah dan Juniarta, 2019), (Nurmalasari dan Sujadi, 2019), (Kurniawan, Chabibi dan Dewi, 2020), (Ritnawati, Suppa dan Muhallim, 2020), (Alda 2020), (Winanjar dan Susanti, 2021), (Purnawan 2021), (Al Hasri dan Sudarmilah, 2021), penerapan website desa (Imaniawan dan Wati, 2017), (Rachman dan Noviyanto, 2017), (Muntofiah dan Fitrian, 2017), (Syaharuddin 2017), (Rozi dan Listiawan, 2017), (Usmanto et al. 2018), (Pratiwi dan Muslihudin, 2018), (Samsudin dan Muslihudin, 2018), (Susilowati dan Chandra, 2018), (Setyowati, Anton dan Radiyah, 2019), (Soraya 2019), (Jimi 2019), (Putra, Hidayatullah dan Juniarta, 2019), (Kurniawan, Chabibi dan Dewi, 2020), (Winanjar dan Susanti, 2021), (Purnawan 2021), (Al Hasri dan Sudarmilah, 2021), menggunakan *platform* android (Syaharuddin 2017), (Usmanto

et al. 2018), (Pratiwi dan Muslihudin, 2018), (Nurmalasari dan Sujadi, 2019), (Ritnawati, Suppa dan Muhallim, 2020), (Alda 2020). Sistem ini telah menggabungkan beberapa fitur tersebut dan menambahkan inovasi fitur menampilkan produk UMKM desa yang dihubungkan aplikasi *marketplace* dengan memanfaatkan teknik *crowdsourcing*. Selain itu, fitur pengaduan dan aspirasi kepada pemerintah desa akan ditambahkan dalam aplikasi yang akan dibuat. Beberapa kelebihan lainnya yang dimiliki aplikasi digitalisasi desa ini yaitu menyediakan fitur untuk manajemen kependudukan, mengolah dana desa, surat menyurat dan berita desa.

BAB V KESIMPULAN

Dari hasil pembuatan Tugas Akhir Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik *Crowdsourcing*, didapatkan kesimpulan sebagai berikut:

1. Aplikasi berhasil digunakan sebagai pengelolaan data pemerintah desa untuk melayani proses administrasi masyarakat desa secara *online*.
2. Aplikasi berhasil digunakan sebagai media penyampaian secara *online* terhadap permasalahan/ide masyarakat.
3. Aplikasi berhasil digunakan untuk mempromosikan dan menjual produk UMKM secara *online*.
4. Berdasarkan hasil kuesioner tingkat kepuasan pengguna, diperoleh persentase 88,3%. Hasil tersebut menyatakan bahwa pengguna sangat puas dengan digitalisasi pelayanan administrasi dan ketahanan ekonomi masyarakat desa dengan teknik *crowdsourcing*, yang berarti bahwa sistem ini telah dibuat sesuai dengan kebutuhan dan harapan pengguna.

DAFTAR PUSTAKA

- Alda, Muhamad. 2020. "Sistem Informasi Pengolahan Data Kependudukan Pada Kantor Desa Sampean Berbasis Android." *Jurnal Media Informatika Budidarma* 4(1):1. doi: 10.30865/mib.v4i1.1716.
- Al Hasri, Vicky Muhammad, and Endah Sudarmilah. 2021. "Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis Website Kelurahan Banaran." 20(2):249–60. doi: 10.30812/matrik.v20i2.1056.
- Imaniawan, Fabriyan Fandi Dwi, and Fanny Fatma Wati. 2017. "Sistem Informasi Administrasi Kependudukan Berbasis Web Pada Desa Bogangin Sumpiuh." *Indonesian Journal on Networking and Security* 7(3):1–9.
- Jimi, Asmara. 2019. "Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala)." *Jurnal Pendidikan Teknologi Informasi (JUKANTI)* 2(1):1–7. doi: 10.37792/jukanti.v2i1.17.
- Kurniawan, Agung, Muhammad Chabibi, and Renny Sari Dewi. 2020. "Pengembangan Sistem Informasi Pelayanan Desa Berbasis Web Dengan Metode Prototyping Pada Desa Leran." *JURIKOM (Jurnal Riset Komputer)* 7(1):114. doi: 10.30865/jurikom.v7i1.1863.
- Muntofiah, Titin, and Yuri Fitrian. 2017. *E-Government Pada Pekon Srimulyo Kalirejo Kabupaten Lampung Tengah*.
- NurmalaSari, Yuli, and Harun Sujadi. 2019. "Rancang Bangun Sistem Administrasi Desa Berbasis Android Di Kantor Kepala Desa Keramat Jaya Kabupaten Majalengka." 130–37.
- Paryanta, Sutariyani, and Desi Susilowati. 2017. "Sistem Informasi Administrasi Kependudukan Berbasis Web Desa Sawahan." *IJSE - Indonesian Journal on Software Engineering* 3(2):77–81.
- Pratiwi, Eka, and Muhamad Muslihudin. 2018. "Implementasi E-Goverment Sebagai Upaya Peningkatan Potensi Desa Di Desa Bumirejo Menggunakan Web Mobile." *Technology Acceptance Model* 9(1):22–29.
- Purnawan, I. Putu Abdi Purnawan. 2021. "Rancang Bangun Sistem Informasi Manajemen Kependudukan Desa Adat Terintegrasi Berbasis Web." (April). doi: 10.13140/RG.2.2.28344.06405.
- Putra, Indra Jaya, Muhammad Hidayatullah, and Aswin Tri Juniarta. 2019. "Aplikasi Pelayanan Administrasi Desa Karang Dima Berbasis Web." 1(1):29–38.
- Rachman, Eviana Septiana, and Beni Noviyanto. 2017. "Pemanfaatan E-

- Government Pada Desa Wonokarto Untuk Meningkatkan Akurasi Dan Informasi Potensi Desa.” *Technology Acceptance Model* 8(1):45–50.
- Rahmawati, Annisa Dwi, and Azizah Fatmawati. 2020. “Sistem Administrasi Desa Mendirikan Kecamatan Ngrambe Kabupaten Ngawi Berbasis Web.” *Emitor: Jurnal Teknik Elektro* 20(02):149–55. doi: 10.23917/emitor.v20i02.9893.
- Ritnawati, Ritnawati, Rinto Suppa, and Muhlis Muhallim. 2020. “Sistem Informasi Pelayanan Masyarakat Berbasis Android Pada Kantor Desa Kaliba Mamase.” *SPEKTA (Jurnal Pengabdian Kepada Masyarakat : Teknologi Dan Aplikasi)* 1(2):95. doi: 10.12928/spekta.v1i2.2860.
- Rohaeti, Eti, and Dadan Zaliluddin. 2018. “Perancangan Sistem Informasi Manajemen Administrasi Desa Berbasis Web.” *Seminar Nasional Rekayasa Teknologi Informasi* 2(November):1–7.
- Rozi, Fahrur, and Tomi Listiawan. 2017. “Pengembangan Website Dan Sistem Informasi Desa Di Kabupaten Tulungagung.” *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)* 2(2):107–12. doi: 10.29100/jipi.v2i2.366.
- Russell J.T. Dyer. 2008. *MySQL in a Nutshell*. Second Esi.
- Samsudin, Imam, and Muhamad Muslihudin. 2018. “Implementasi Web Government Dalam Meningkatkan Potensi Produk Unggulan Desa Berbasis Android.” *Jtksi* 1(2):10–16.
- Setyowati, Dheska Ranita, Anton Anton, and Ummu Radiyah. 2019. “Implementasi Pelayanan Administrasi Kependudukan Desa Kalimanah Wetan Purbalingga Berbasis Online.”
- Sihombing, Volvo. 2018. “Aplikasi Simade (Sistem Informasi Manajemen Desa) Dalam Meningkatkan Pelayanan Administrasi Di Kepenghuluan Bakti Makmur Kecamatan Bagan Sinembah Kab. Rokan Hilir Riau.” *Sistemasi* 7(3):292. doi: 10.32520/stmsi.v7i3.384.
- Sommerville, Ian. 2011. *Software Engineering (9th Edition)*. 9th ed. USA: Pearson Education.
- Soraya, Tania. 2019. “Inovasi Pelayanan Administrasi Kependudukan Secara Online Di Kabupaten Pati.” *Astuti, Puji* 1–10.
- Susilowati, Memé, and Dicky Chandra. 2018. “Sistem Informasi Administrasi Desa Di Kantor Desa Sumbersekar.” *KURAWAL : Jurnal Teknologi, Informasi, Dan Industri* I(1):40–44.
- Syaharuddin. 2017. “Aplikasi Sistem Informasi Desa Sebagai Teknologi Tepat Guna Untuk Pendataan Penduduk Dan Potensi Desa.” *Jurnal Masyarakat Mandiri* 1(1):4–11.
- Syukron, Akhmad. 2019. “Perancangan Sistem Informasi Administrasi

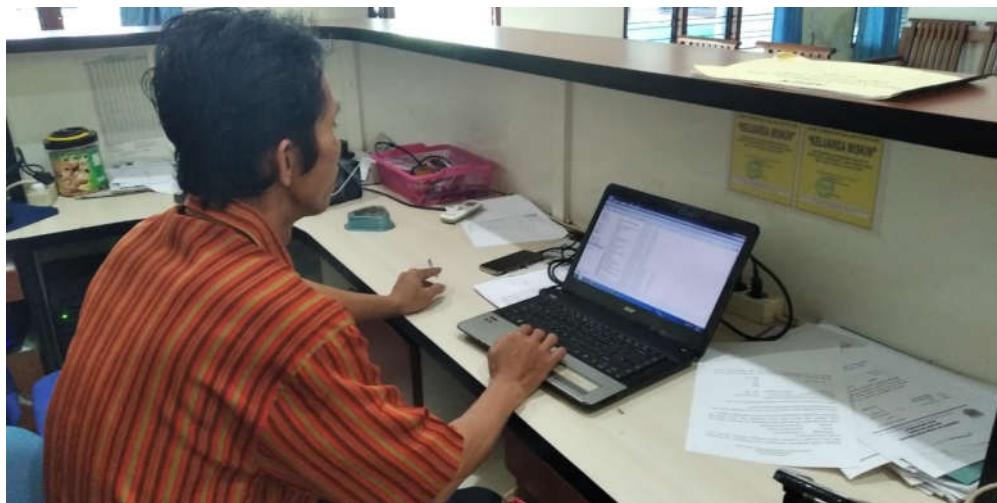
- Kependudukan Desa Berbasis Website Pada Desa Winong.” *Bianglala Informatika* 7(1):16–21. doi: 10.31294/bi.v7i1.5790.
- Telaumbanua, Dalinama. 2020. “Urgensi Pembentukan Aturan Terkait Pencegahan Covid-19 Di Indonesia.” *QALAMUNA: Jurnal Pendidikan, Sosial, Dan Agama* 12(01):59–70. doi: 10.37680/qalamuna.v12i01.290.
- Usmanto, Budi, Riswandha Immawan, Fauzi, Keni Puspita Sari, and M. Isam Mahdi. 2018. “Implementasi Web Mobile Sebagai Media Informasi Pemberdayaan Masyarakat Di Desa Pirngadi.” *Jurnal Keteknikan Dan Sains* 1(1):32–40.
- Winanjar, Jajang, and Deffy Susanti. 2021. “Rancangan Bangunan Sistem Informasi Administrasi Desa Berbasis Web Menggunakan PHP Dan MySQL.” *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)* 97–105.

LAMPIRAN

Lampiran 1 Dokumentasi



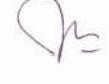
Desa Penelitian



Uji Coba Sistem

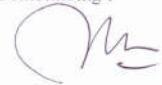
Lampiran 2 Lembar Kontrol Bimbingan Tugas Akhir

	LEMBAR KONTROL BIMBINGAN TUGAS AKHIR/SKRIPSI	No. PM	7.5.18/L1
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	2 / 2

No	Tanggal	Uraian	Tandatangan Pembimbing
			Pembimbing 1 Pembimbing 2 Pembimbing 3
1	18 Maret 2021	- Pemaparan konsep Tugas Akhir - Revisi naskah proposal Tugas Akhir	   
2	29 Maret 2021	- Alur kerja sistem - Pemberian ide nama sistem	  
3	28 Juni 2021	- Mengurangi fitur sistem - Fokus pada dana desa, UMKM dihubungkan markerplace, aspirasi masyarakat dan surat menyurat	  
4	13 Juli 2021	- Menampilkan progress - Revisi halaman dashboard	 
5	10 Agustus 2021	- Demo aplikasi website dan android - Menambahkan OTP melalui WhatsApp	 
6	3 September 2021	- Demo sistem - Revisi tampilan chart	 

8	26 Agustus 2021	Acc laporan BAB 1	
9	27 Agustus 2021	Bimbingan laporan BAB 2	
10	30 Agustus 2021	Acc laporan BAB 2	
11	2 September 2021	Bimbingan laporan BAB 3	
12	6 September 2021	Acc laporan BAB 3	
13	8 September 2021	Bimbingan laporan BAB 4	
14	9 September 2021	Acc laporan BAB 4	
15	10 September 2021	Bimbingan laporan BAB 5	
16	13 September 2021	Acc laporan BAB 5	

Semarang, 13 September 2021
Pembimbing I



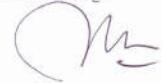
Mardiyono, S.Kom., M.Sc.
NIP. 197403112000121001

	LEMBAR KONTROL BIMBINGAN TUGAS AKHIR/SKRIPSI	No. PM	7.5.18/L1
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	2 / 2

No	Tanggal	Uraian	Tandatangan Pembimbing
1	18 Maret 2021	- Pemaparan konsep Tugas Akhir - Revisi naskah proposal Tugas Akhir	     
2	29 Maret 2021	- Alur kerja sistem	    
3	28 Juni 2021	- Pemberian ide nama sistem - Mengurangi fitur sistem - Fokus pada dana desa, UMKM dihubungkan markerplace, aspirasi masyarakat dan surat menyurat	    
4	13 Juli 2021	- Menampilkan progress	
5	10 Agustus 2021	- Revisi halaman dashboard - Demo aplikasi website dan android - Menambahkan OTP melalui WhatsApp	   
6	3 September 2021	- Demo sistem	

8	26 Agustus 2021	Acc laporan BAB 1	
9	27 Agustus 2021	Bimbingan laporan BAB 2	
10	30 Agustus 2021	Acc laporan BAB 2	
11	2 September 2021	Bimbingan laporan BAB 3	
12	6 September 2021	Acc laporan BAB 3	
13	8 September 2021	Bimbingan laporan BAB 4	
14	9 September 2021	Acc laporan BAB 4	
15	10 September 2021	Bimbingan laporan BAB 5	
16	13 September 2021	Acc laporan BAB 5	

Semarang, 13 September 2021
Pembimbing I



Mardiyono, S.Kom., M.Sc.
NIP. 197403112000121001

	LEMBAR KONTROL BIMBINGAN TUGAS AKHIR/SKRIPSI	No. PM	7.5.18/L1
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	1 / 2

No	Tanggal	Uraian	Tandatangan Pembimbing
1	5 April 2021	- Pemaparan konsep tugas akhir	/
2	25 April 2021	- Bimbingan proposal tugas akhir dan KMIPN	/
3	29 April 2021	- Bimbingan revisi proposal tugas akhir dan KMIPN	/
4	21 Juni 2021	- Bimbingan progress penggerjaan sistem	/
5	15 Agustus 2021	- Bimbingan laporan BAB 1 dan 2	/
		- Revisi perumusan masalah	/
6	2 September 2021	- Demo sistem	/
		- Revisi sistem	/
7	6 September 2021	- Bimbingan BAB 3, 4 dan 5	/
8	14 September 2021	- Acc sistem dan laporan	/

Semarang, 14 September 2021
Pembimbing II



Dr. Ir. Kurnianingsih, S.T., M.T.
NIP. 197904262003122002

	LEMBAR KONTROL BIMBINGAN TUGAS AKHIR/SKRIPSI	No. PM	7.5.18/L1
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	1 / 2

No	Tanggal	Uraian	Tandatangan Pembimbing
			Pembimbing I
1	5 April 2021	- Pemaparan konsep tugas akhir	/
2	25 April 2021	- Bimbingan proposal tugas akhir dan KMIPN	/
3	29 April 2021	- Bimbingan revisi proposal tugas akhir dan KMIPN	/
4	21 Juni 2021	- Bimbingan progress penggeraan sistem	/
5	15 Agustus 2021	- Bimbingan laporan BAB 1 dan 2	/
		- Revisi perumusan masalah	/
6	2 September 2021	- Demo sistem	/
		- Revisi sistem	/
7	6 September 2021	- Bimbingan BAB 3, 4 dan 5	/
8	14 September 2021	- Acc sistem dan laporan	/

Semarang, 14 September 2021
Pembimbing II

Dr. Ir. Kurnianingsih, S.T., M.T.
NIP. 197904262003122002

Lampiran 3 Form Revisi Tugas Akhir

	LEMBAR REVISI TUGAS AKHIR	No. PM 7.5.21/L3 Revisi 2 Tanggal 1 Juli 2010 Halaman 2 / 2
---	----------------------------------	--

No	Uraian	Tandatangan Penguji
1.	Revisi lihat di naskah sesuai halaman yang dilipat	

Semarang, 16 September 2021

Penguji I,



Tri Raharjo Yudantoro, S.Kom., M.Kom.

NIP. 196810252000121001

	LEMBAR REVISI TUGAS AKHIR	No. PM	7.5.21/L3
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	2 / 2

No	Uraian	Tandatangan Penguji
1.	Pendahuluan data dukung kunjungan dan jumlah penduduk	
2.	Hilangkan keuangan dalam laporan	
3.	Perluas publikasi dengan menampilkan pada website (seluruh produk per kategori)	
4.	Tambahkan ERD	

Semarang, 16 September 2021

Penguji II,



Liliket Triyono, S.T., M.Kom.

NIP 198404202015041003

	LEMBAR REVISI TUGAS AKHIR	No. PM	7.5.21/L3
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	2 / 2

No	Uraian	Tandatangan Penguji
1.	Naskah : lihat naskah Pak Handoko	
2.	Aplikasi :	
	Penduduk (Aktor → RT, RW, Kasi, Kades, Sekdes)	
	Surat menyurat (Disesuaikan dengan kondisi riil → jenis surat dan syarat yang harus diupload)	
	Aspirasi (Didasarkan dari aspirasi RT, RW untuk bahan rembug desa)	
	UMKM (Menjual potensi yang dihasilkan desa ke khalayak umum)	

Semarang, 16 September 2021

Penguji III,



Slamet Handoko, S.Kom., M.Kom.

NIP 197501302001121001

Lampiran 4 Surat Keterangan Selesai Revisi Tugas Akhir

176

Lampiran 4 Surat Keterangan Selesai Revisi Tugas Akhir

	REVISI TUGAS AKHIR	No. FPM	7.5.21/L3
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	1 / 1

Yang bertanda tangan di bawah ini, Pengaji I/II/III menerangkan bahwa:

No.	Nama	NIM	Kelas
1	Dwi Hapsari	3.34.18.0.08	IK-3A
2	Muhammad Irhamul Umam	3.34.18.0.16	IK-3A

Dengan judul Tugas Akhir:

**Digitalisasi Pelayanan Administrasi dan Ketahanan Ekonomi Masyarakat Desa dengan Teknik
*Crowdsourcing***

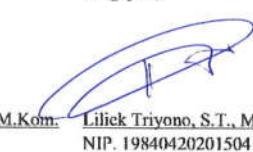
Benar-benar telah melaksanakan revisi tugas akhir. Surat keterangan ini dapat dipergunakan sebagaimana mestinya.

Semarang, 16 September 2021

Pengaji I,

Tri Raharjo Yudantoro S.Kom., M.Kom.
NIP. 196810252000121001

Pengaji II,

Liliek Triyono, S.T., M. Kom.
NIP. 198404202015041003

Pengaji III,

Slamet Handoko, S.Kom., M.Kom.
NIP 197501302001121001

Lampiran 5 Kode Main File

```

Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
  await Firebase.initializeApp();
  final SharedPreferences = await SharedPreferences.getInstance();
  await SharedPreferences.setString('notification', message.data['content']);
  print("Handling a background message: ${message.messageId}");
  print(message.data['content']);
}

AndroidNotificationChannel? channel;

FlutterLocalNotificationsPlugin? flutterLocalNotificationsPlugin;
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  di.init();
  await di.sl.allReady();
  fcmInit();
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
  channel = const AndroidNotificationChannel(
    'high_importance_channel', // id
    'High Importance Notifications', // title
    'This channel is used for important notifications.', // description
    importance: Importance.high,
  );
  flutterLocalNotificationsPlugin = FlutterLocalNotificationsPlugin();
  await flutterLocalNotificationsPlugin!
    .resolvePlatformSpecificImplementation<
      AndroidFlutterLocalNotificationsPlugin>()
    ?.createNotificationChannel(channel!);
  FirebaseMessaging.onMessage.listen((RemoteMessage message) async{
    RemoteNotification? notification = message.notification;
    AndroidNotification? android = message.notification?.android;
    if (notification != null && android != null) {
      flutterLocalNotificationsPlugin!.show(
        notification.hashCode,
        notification.title,
        notification.body,
        NotificationDetails(
          android: AndroidNotificationDetails(
            channel!.id,
            channel!.name,
            channel!.description,
            icon: '@mipmap/ic_launcher',
            styleInformation: BigTextStyleInformation(),
          ),
        ));
    }
  });
  runApp(MyApp());
}

BuildContext? testContext;
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final textTheme = Theme.of(context).textTheme;

```

```

return MultiBlocProvider(
  providers: [
    BlocProvider<AuthBloc>(
      create: (BuildContext context) => di.sl<AuthBloc>(),
    ),
    BlocProvider<AspirationBloc>(
      create: (BuildContext context) => di.sl<AspirationBloc>(),
    ),
    BlocProvider<LetterRequestBloc>(
      create: (BuildContext context) => di.sl<LetterRequestBloc>(),
    ),
    BlocProvider<UmkmBloc>(
      create: (BuildContext context) => di.sl<UmkmBloc>(),
    ),
    BlocProvider<UmkmProductBloc>(
      create: (BuildContext context) => di.sl<UmkmProductBloc>(),
    ),
    BlocProvider<HomeBloc>(
      create: (BuildContext context) => di.sl<HomeBloc>(),
    ),
    BlocProvider<UserSkillBloc>(
      create: (BuildContext context) => di.sl<UserSkillBloc>(),
    ),
    BlocProvider<InformationBloc>(
      create: (BuildContext context) => di.sl<InformationBloc>(),
    ),
    BlocProvider<NotificationBloc>(
      create: (BuildContext context) => di.sl<NotificationBloc>(),
    ),
  ],
  child: MaterialApp(
    title: 'Dering Login',
    debugShowCheckedModeBanner: false,
    theme: ThemeData(
      textTheme: GoogleFonts.latoTextTheme(textTheme).copyWith(
        bodyText1: GoogleFonts.openSans(textStyle: textTheme.bodyText1),
      ),
      primaryColor: HexColor('#18A4E0'),
      scaffoldBackgroundColor: Colors.white,
      visualDensity: VisualDensity.adaptivePlatformDensity),
    home: SplashScreen(),
    builder: EasyLoading.init(),
    onGenerateRoute: RouteGenerator.generateRoute,
  );
}
}
}

```

Lampiran 6 Kode Fitur *Splash Screen*

```

class SplashScreen extends StatefulWidget {
  const SplashScreen({Key? key}) : super(key: key);

  @override
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  bool _visible = false;
  void sharedPrefs() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.reload();
  }
  @override
  void initState() {
    super.initState();
    sharedPrefs();
    Future.delayed(Duration(seconds: 5));
  }
}

```

```

        setState(() {
            _visible = true;
        });
    }
    @override
    Widget build(BuildContext context) {
        return BlocBuilder<AuthBloc, AuthState>(
            builder: (context, state) {
                if (state is AuthInitial) {
                    BlocProvider.of<AuthBloc>(context).add(AuthCheck());
                }
                if (state is UserUnauthenticated) {
                    Future.delayed(
                        Duration(seconds: 2),
                        () => Navigator.of(context)
                            .pushNamedAndRemoveUntil('/login-otp', (route) => false));
                }
                if (state is UserAuthenticated) {
                    Future.delayed(
                        Duration(seconds: 2),
                        () => Navigator.of(context)
                            .pushNamedAndRemoveUntil('/home', (route) => false));
                }
                if (state is AuthenticationFail) {
                    Future.delayed(
                        Duration(seconds: 2),
                        () => Navigator.of(context)
                            .pushNamedAndRemoveUntil('/login-otp', (route) => false));
                }
            View().init(context);
            return Scaffold(
                body: Container(
                    color: Colors.white,
                    child: Stack(
                        children: [
                            Positioned(
                                top: 0,
                                left: 0,
                                child: Container(
                                    width: MediaQuery.of(context).size.width,
                                    height: MediaQuery.of(context).size.height,
                                    child: Column(
                                        mainAxisAlignment: MainAxisAlignment.center,
                                        children: [
                                            DefaultTextStyle(
                                                style: TextStyle(
                                                    fontWeight: FontWeight.bold,
                                                    fontSize: 25,
                                                    color: Colors.grey.shade800),
                                                child: AnimatedTextKit(
                                                    isRepeatingAnimation: true,
                                                    repeatForever: true,
                                                    animatedTexts: [
                                                        FadeAnimatedText(
                                                            'DESA DARING',
                                                            duration: Duration(milliseconds: 2000),
                                                            fadeInEnd: 0.5,
                                                        ),
                                                    ],
                                                ),
                                            ),
                                        ],
                                    ),
                                ),
                            ),
                        ],
                    ),
                ),
            );
        );
    }
}

```

```
        fadeOutBegin: 0.6
    ),
    ],
),
],
)),
),
),
Positioned(
    bottom: 0.0,
    child: Container(
        width: MediaQuery.of(context).size.width,
        margin: EdgeInsets.only(bottom: 40),
        alignment: Alignment.center,
        child: Text('1.0')),
),
],
),
));
},
);
}
}
```

Lampiran 7 Kode Fitur *Login*

```
File login_otp.dart
class LoginOtp extends StatefulWidget {
  LoginOtp({Key? key}) : super(key: key);

  @override
  LoginOtpState createState() => LoginOtpState();
}

class LoginOtpState extends State<LoginOtp> {
  final nomorHpController = TextEditingController();
  bool _validate = false;

  @override
  void dispose() {
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    View().init(context);
    Size size = MediaQuery.of(context).size;
    return Scaffold(
      backgroundColor: HexColor("#043353"),
      body: SafeArea(
        child: BlocListener<AuthBloc, AuthState>(
          listener: (context, state) {
            if (state is AuthLoading) {
              EasyLoading.dismiss();
              EasyLoading.show(status: 'Proses Login...!');
            }
            if (state is OtpSent) {
              EasyLoading.dismiss();
              pushNewScreen(context,
                screen: InputOtp(
                  noHp: nomorHpController.text,
                ),
              );
            }
          },
        ),
      ),
    );
  }
}
```

```

    ));
}
if(state is AuthenticationFail) {
    EasyLoading.dismiss();
    ScaffoldMessenger.of(context)
        .showSnackBar(SnackBar(content: Text(state.message)));
}
if(state is OtpSent) {
    pushNewScreen(context,
        screen: InputOtp(
            noHp: nomorHpController.text,
        )));
}
},
child: Background(
    title: "LOGIN",
    subtitle: "Selamat Datang...",
    child: BlocBuilder<AuthBloc, AuthState>(
        builder: (context, state) {
            return Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                    SizedBox(height: 10),
                    Container(
                        alignment: Alignment.center,
                        margin: EdgeInsets.symmetric(horizontal: 10),
                        child: TextField(
                            autocorrect: true,
                            autofocus: false,
                            controller: nomorHpController,
                            enabled: true,
                            keyboardType: TextInputType.number,
                            decoration: InputDecoration(
                                hintText: "Contoh: 6281234567890",
                                labelText: 'Nomor Whatsapp (Diawali dengan 62)',
                                border: OutlineInputBorder(
                                    borderRadius:
                                        BorderRadius.all(Radius.circular(10))),
                                errorText: _validate
                                    ? 'Nomor Whatsapp Harus Diisi'
                                    : null)),
                ),
                    SizedBox(height: size.height * 0.03),
                    Container(
                        alignment: Alignment.center,
                        margin:
                            EdgeInsets.symmetric(horizontal: 40, vertical: 10),
                        child: ElevatedButton(
                            onPressed: () async {
                                setState(() {
                                    nomorHpController.text.isEmpty
                                        ? _validate = true
                                        : _validate = false;
                                });
                                if(_validate == false) {
                                    BlocProvider.of<AuthBloc>(context).add(
                                        SignInNoHpEvent(
                                            noHp: nomorHpController.text));
                                }
                            },
                        ),
                    ),
                ],
            );
        },
    ),
);
}

```

File auth_bloc.dart

```
if(event is SignInNoHpEvent) {  
    yield AuthLoading();  
    final result = await authRepository.sendOtp(data: {'no_hp': event.noHp});  
    yield* result.fold((failure) async* {  
        if(failure is ServerFailure) {  
            yield AuthenticationFail(message: ServerFailure.MESSAGE);  
        }  
        if(failure is UserLoginNoHpFailure) {  
            yield AuthenticationFail(message: UserLoginNoHpFailure.MESSAGE);  
        }  
    });  
}
```

```
        },
        (userInfo) async* {
            yield OtpSent(noHp: event.noHp);
        });
    }
}
```

File auth_repository.dart

```
Future<Either<Failure, String>> sendOtp({required data}) async {
  if (await networkInfo.isConnected) {
    try {
      final _token = await AppStorage.getToken('token');
      var response = await http.post(Uri.parse(Constants.SEND_OTP_URL),
          headers: Constants.HEADERS(_token ?? ""),
          body: jsonEncode(data));
      if (response.statusCode == 200) {
        if (jsonDecode(response.body)['success'] == true) {
          return Right("Kode OTP Telah Dikirimkan");
        } else {
          return Left(UserLoginNoHpFailure());
        }
      } else {
        return Left(UserLoginNoHpFailure());
      }
    } catch (e) {
      return Left(UserLoginNoHpFailure());
    }
  }
  return Left(ServerFailure());
}
```

Lampiran 8 Kode Validasi OTP

```
File input_otp.dart

class InputOtp extends StatefulWidget {
    final noHp;
    const InputOtp({Key? key, this.noHp}) : super(key: key);

    @override
    _InputOtpState createState() => _InputOtpState();
}

class _InputOtpState extends State<InputOtp> {
    TextEditingController otpController = TextEditingController();
    StreamController<ErrorAnimationType>? _errorController;

    bool hasError = false;
    String currentText = "";
    final formKey = GlobalKey<FormState>();

    @override
    void initState() {
        _errorController = StreamController<ErrorAnimationType>();
        super.initState();
    }
}
```

```

@Override
void dispose() {
    _errorController!.close();
    super.dispose();
}

snackBar(String? message) {
    return ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text(message!),
            duration: Duration(seconds: 2),
        ),
    );
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: HexColor('#FFFFFF'),
        body: BlocListener<AuthBloc, AuthState>(
            listener: (context, state) {
                if (state is AuthenticationFail) {
                    EasyLoading.dismiss();
                    ScaffoldMessenger.of(context)
                        .showSnackBar(SnackBar(content: Text(state.message)));
                }
                if (state is AuthLoading) {
                    EasyLoading.dismiss();
                    EasyLoading.show(status: 'Proses Login...');
                }
                if (state is LoginSuccess) {
                    EasyLoading.dismiss();
                    Navigator.of(context).pushNamedAndRemoveUntil('/home', (route) => false);
                }
            },
            child: GestureDetector(
                onTap: () {},
                child: Container(
                    height: MediaQuery.of(context).size.height,
                    width: MediaQuery.of(context).size.width,
                    child: ListView(
                        children: <Widget>[
                            SizedBox(height: 30),
                            Padding(
                                padding: const EdgeInsets.symmetric(vertical: 8.0),
                                child: Text(
                                    'Verifikasi Akun',
                                    style: TextStyle(fontWeight: FontWeight.bold, fontSize: 22),
                                    textAlign: TextAlign.center,
                                ),
                            ),
                            Padding(
                                padding: const EdgeInsets.symmetric(horizontal: 30.0, vertical: 8),
                                child: RichText(
                                    text: TextSpan(
                                        text: "Masukkan Kode OTP Yang Dikirim Ke ",

```



```

    });
},
beforeTextPaste: (text) {
  print("Allowing to paste ");
  return true;
},
),
),
),
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 30.0),
  child: Text(
    hasError ? "*Mohon Masukkan Kode OTP Ke Dalam Isian" : "",
    style: TextStyle(
      color: Colors.red,
      fontSize: 12,
      fontWeight: FontWeight.w400,
    ),
  ),
),
SizedBox(
  height: 20,
),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Text(
      "Belum Menerima Kode? ",
      style: TextStyle(color: Colors.black54, fontSize: 15),
    ),
    TextButton(
      onPressed: () => snackBar("Kode Telah Dikirim!!"),
      child: Text(
        "Kirim Ulang",
        style: TextStyle(
          color: Color(0xFF91D3B3),
          fontWeight: FontWeight.bold,
          fontSize: 16,
        ),
      )));
],
),
SizedBox(
  height: 14,
),
Container(
  margin: const EdgeInsets.symmetric(
    vertical: 16.0, horizontal: 30),
  child: ButtonTheme(
    height: 50,
    child: TextButton(
      onPressed: () async {
        formKey.currentState!.validate();
        if (currentText.length != 6) {
          _errorController!.add(ErrorAnimationType
            .shake); // Triggering error shake animation
          setState(() => hasError = true);
        } else {
          setState(

```

```
        () {
            hasError = false;
        },
    );
    BlocProvider.of<AuthBloc>(context).add(
        SignInOtpEvent(noHp: widget.noHp, otp: currentText));
}
),
child: Center(
    child: Text(
        "KONFIRMASI".toUpperCase(),
        style: TextStyle(
            color: Colors.white,
            fontSize: 18,
            fontWeight: FontWeight.bold),
    )));
),
),
),
),
decoration: BoxDecoration(
    color: Colors.green.shade300,
    borderRadius: BorderRadius.circular(5),
),
),
),
),
SizedBox(
    height: 16,
),
Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
        Flexible(
            child: TextButton(
                child: Text("Bersihkan"),
                onPressed: () {
                    otpController.clear();
                },
            )));
    ],
),
),
),
),
),
);
}
}
```

```
File auth_bloc.dart
if (event is SignInOtpEvent) {
  yield AuthLoading();
  final data = {'no_hp': event.noHp, 'otp_code': event.otp};
  final result = await authRepository.validateOtp(data: data);
  yield* result.fold((failure) async* {
    if (failure is ServerFailure) {
      yield AuthenticationFail(message: ServerFailure.MESSAGE);
      print('inner');
    }
    if (failure is UserLoginOtpFailure) {
```

```

        yield AuthenticationFail(message: UserLoginOtpFailure.MESSAGE);
        print('gaagal');
    }
}, (message) async* {
    yield LoginSuccess(message: message);
    print('success');
});
}

File auth_repository
Future<Either<Failure, String>> validateOtp({required data}) async {
    print('start');
    if (await networkInfo.isConnected) {
        print('koneksi');
        try {
            final _token = await AppStorage.getToken('token');
            print('cek token');
            print(data);
            var response = await http.post(Uri.parse(Constants.VALIDATE_OTP_URL),
                headers: Constants.HEADERS(_token ?? ""), body: jsonEncode(data));
            print('post form');
            if (response.statusCode == 200) {
                if (jsonDecode(response.body)['meta']['code'] == 200) {
                    AppStorage.setToken('token', jsonDecode(response.body)['data']);
                    print('yoooo');
                    return Right("Login Berhasil");
                } else {
                    print(jsonDecode(response.body));
                    return Left(UserLoginOtpFailure());
                }
            } else {
                print('nool');
                return Left(UserLoginOtpFailure());
            }
        } catch (e) {
            print(e);
            return Left(UserLoginOtpFailure());
        }
    } else {
        print('gak konek');
        return Left(ServerFailure());
    }
}
}

```

Lampiran 9 Kode Fitur Beranda

```

File home.dart
class HomeScreen extends StatefulWidget {
    final BuildContext? menuScreenContext;
    final Function? onScreenHideButtonPressed;
    final bool? hideStatus;
    const HomeScreen({
        Key? key,
        this.menuScreenContext,
        this.onScreenHideButtonPressed,
        this.hideStatus = false,
    }) : super(key: key);
}

```

```

@Override
    _HomeScreenState createState() => _HomeScreenState();
}
class _HomeScreenState extends State<HomeScreen> {
    final TextEditingController _inputController = TextEditingController();
    final CarouselController _controller = CarouselController();
    int _current = 0;
    List<Widget>? myList = [];
    late HomeBloc _homeBloc;

    List<News> news = [];
    List<UmkmProduct> umkmProduct = [];
    User? user;

    String greeting = "Selamat";
    late FirebaseMessaging messaging;

    void _fetchHome() {
        news.clear();
        umkmProduct.clear();
        _homeBloc.add(GetLatest());
        var hour = DateTime.now().hour;
        if (hour <= 10) {
            greeting = "Selamat Pagi...";
        } else if (hour < 15) {
            greeting = "Selamat Siang...";
        } else if (hour < 18) {
            greeting = "Selamat Sore...";
        } else {
            greeting = "Selamat Malam...";
        }
    }
}

@Override
void initState() {
    _homeBloc = HomeBloc(
        newsRepository: NewsRepository(networkInfo: sl()),
        umkmProductRepository: UmkmProductRepository(networkInfo: sl()),
        authRepository: AuthRepository(networkInfo: sl()));
    _fetchHome();
    super.initState();
}

@Override
Widget build(BuildContext context) {
    // Size size = MediaQuery.of(context).size;
    return BlocConsumer<HomeBloc, HomeState>(
        bloc: _homeBloc,
        listener: (context, state) {
            if (state is ShowMessage) {
                showDialog(
                    context: context,
                    builder: (context) {
                        return AlertDialog(
                            content: Text(state.message),
                        );
                    });
            }
        } else if (state is HomeLoaded) {
}

```

```

print('loaded');
news.clear();
umkmProduct.clear();
news.addAll(state.news);
umkmProduct.addAll(state.umkmProduct);
user = state.user;
}
},
builder: (context, state) {
return RefreshIndicator(
onRefresh: () async {
_fetchHome();
},
child: Container(
child: SizedBox(
height: MediaQuery.of(context).size.height,
width: MediaQuery.of(context).size.width,
child: Scaffold(
backgroundColor: HexColor('#043353'),
body: SafeArea(
child: Column(children: <Widget>[
Container(
margin: EdgeInsets.fromLTRB(20, 20, 20, 20),
child: Row(
children: [
Expanded(
flex: 1,
child: Align(
alignment: Alignment.centerLeft,
child: state is HomeLoading
? ShimmerWidget.circular(
width: 40, height: 40)
: ClipRRect(
borderRadius: BorderRadius.circular(45.0),
child: Image.asset(
"assets/images/avatar.png",
width: 40,
height: 40),
),
),
),
Expanded(
flex: 4,
child: Column(
children: [
Align(
alignment: Alignment.centerLeft,
child: state is HomeLoading
? ShimmerWidget.rectangular(
width: 100, height: 7)
: Text(
greeting,
style: TextStyle(
fontSize: 12,
color: HexColor("#FFFFFF")),
)),
),
state is HomeLoading
? SizedBox(

```

```

        height: 5,
    )
: SizedBox(
    height: 1,
),
Align(
alignment: Alignment.centerLeft,
child: state is HomeLoading
? ShimmerWidget.rectangular(
    width: 150, height: 10)
: Text(
    user != null
    ? user!.userDetail!.namaLengkap!
    : 'Pengguna',
style: TextStyle(
    fontSize: 14,
    fontWeight: FontWeight.bold,
    color: HexColor("#FFFFFF")),
),
),
],
),
),
),
Expanded(
flex: 1,
child: Align(
alignment: Alignment.centerRight,
child: GestureDetector(
onTap: ()=> {
    pushNewScreen(context,
        screen: NotificationIndex(),
        withNavBar: false)
},
child: Icon(
Icons.notifications,
color: HexColor("#FFFFFF"),
size: 30,
),
),
),
),
),
],
),
),
),
Expanded(
child: Container(
margin: EdgeInsets.only(bottom: 5),
decoration: BoxDecoration(
color: HexColor("#FFFFFF"),
borderRadius: BorderRadius.only(
topLeft: Radius.circular(30.0),
topRight: Radius.circular(30.0))),
child: Column(
children: [
Container(
margin: EdgeInsets.symmetric(
vertical: 20, horizontal: 20),
child: Row(

```

```

mainAxisAlignment: MainAxisAlignment.spaceBetween,
crossAxisAlignment: CrossAxisAlignment.center,
children: [
  Text(
    "Berita Terbaru",
    style: TextStyle(
      color: Colors.black,
      fontWeight: FontWeight.bold),
  ),
  GestureDetector(
    onTap: () => {
      pushNewScreen(
        context,
        screen: Home(indextab: 1),
        withNavBar:
          false, // OPTIONAL VALUE. True by default.
        pageTransitionAnimation:
          PageTransitionAnimation.cupertino,
    )
  },
  child: Icon(
    Icons.arrow_forward,
    color: HexColor("#043353"),
  ),
),
],
),
),
),
Container(
height: 150,
margin: EdgeInsets.only(left: 10, right: 10),
child: state is HomeLoading
? _latestNewsLoading(state)
: _createNewsList(),
),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: mylist!.asMap().entries.map((entry) {
  return Container(
    width: 12.0,
    height: 12.0,
    margin: EdgeInsets.symmetric(
      vertical: 8.0, horizontal: 4.0),
    decoration: BoxDecoration(
      shape: BoxShape.circle,
      color: (Theme.of(context).brightness ==
        Brightness.dark
        ? HexColor("#E2E2E2")
        : HexColor("#1C80FB"))
      .withOpacity(
        _current == entry.key ? 0.9 : 0.2),
    );
  }).toList(),
),
Divider(),
Container(
margin: EdgeInsets.symmetric(
  vertical: 20, horizontal: 20),

```

```
child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    crossAxisAlignment: CrossAxisAlignment.center,
    children: [
        Text(
            "Produk UMKM Teratas",
            style: TextStyle(
                color: Colors.black,
                fontWeight: FontWeight.bold),
        ),
        GestureDetector(
            onTap: () => {
                pushNewScreen(context, screen: UmkmIndex());
            },
            child: Icon(
                Icons.arrow_forward,
                color: HexColor("#043353"),
            ),
        ),
        ],
    ),
),
Expanded(
    child: Container(
        height: 150,
        margin: EdgeInsets.only(
            left: 10, right: 10, bottom: 50),
        child: state is HomeLoading
            ? _topProductLoading(state)
            : _createUmkmProductList(),
    ),
),
],
),
),
),
),
),
),
),
);
},
});
```

```
Widget _latestNewsLoading(state) {
  if (state is HomeLoading) {
    return Container(
      margin: EdgeInsets.all(5.0),
      child: ClipRRect(
        borderRadius: BorderRadius.all(Radius.circular(5.0)),
        child: InkWell(
          onTap: () {},
          child: Stack(
            children: <Widget>[
              ShimmerWidget.rectangular(width: 1000, height: 150),
            ],
          ),
        )));
  }
}
```

```

    );
}
return Container(child: Text('data'));
}

Widget _topProductLoading(state) {
if (state is HomeLoading) {
    return Container(
        height: 150,
        child: new Card(
            child: InkWell(
                onTap: () {},
                child: Container(
                    child: ClipRRect(
                        borderRadius: BorderRadius.all(Radius.circular(5.0)),
                        child: Stack(children: <Widget>[
                            ShimmerWidget.rectangular(width: 1000, height: 150),
                        ]))),),
        );
}
return Container(child: Text('data'));
}

Widget _createNewsList() {
if (news.isNotEmpty) {
    return CarouselSlider(
        items: news
            .map((e) => Container(
                margin: EdgeInsets.all(5.0),
                child: ClipRRect(
                    borderRadius: BorderRadius.all(Radius.circular(5.0)),
                    child: InkWell(
                        onTap: () {
                            pushNewScreen(context,
                                screen: NewsDetail(news: e), withNavBar: false);
                        },
                        child: Stack(
                            children: <Widget>[
                                Image.network(
                                    e.file == null
                                        ? Constants.STORAGE_URL + 'no-image.png'
                                        : Constants.STORAGE_URL + e.file!.path,
                                    fit: BoxFit.cover,
                                    width: 1000.0),
                                Positioned(
                                    bottom: 0.0,
                                    left: 0.0,
                                    right: 0.0,
                                    child: Container(
                                        decoration: BoxDecoration(
                                            gradient: LinearGradient(
                                                colors: [
                                                    Color.fromARGB(200, 0, 0, 0),
                                                    Color.fromARGB(0, 0, 0, 0)
                                                ],
                                                begin: Alignment.bottomCenter,
                                                end: Alignment.topCenter,
                                            )),

```



```
File home_bloc.dart
Stream<HomeState> mapEventToState(
    HomeEvent event,
) async* {
    if (event is GetLatest) {
        yield* _getAllLatest();
    }
}
```

```

Stream<HomeState> _getAllLatest() async*
{
  try {
    yield HomeLoading();
    User? user;
    List<News> news = await newsRepository.latestNews();
    List<UmkmProduct> umkmProduct = await umkmProductRepository.topUmkmProduct();
    Either<Failure, User> response = await authRepository.showUser();
    yield* response.fold((failure) async* {
      }, (u) async* {
        yield HomeLoaded(news: news, umkmProduct: umkmProduct, user: u);
      });
    } catch (e) {
      yield ShowMessage(message: e.toString());
    }
}

File news_repository.dart
Future<List<News>> latestNews() async {
  List<News> news = [];
  final _token = await AppStorage.getToken('token');
  final response = await http.get(
    Uri.parse(Constants.NEWS_LATEST_URL),
    headers: Constants.HEADERS(_token ?? "")
  );
  if(response.statusCode == 200){
    List<dynamic> jsonResponse = json.decode(response.body)['data'];
    return jsonResponse.map((e) => new News.fromJson(e)).toList();
  }
  return news;
}

File umkm_product_repository.dart
Future<List<UmkmProduct>> topUmkmProduct() async{
  List<UmkmProduct> umkmProduct = [];
  final _token = await AppStorage.getToken('token');
  final response = await http.get(
    Uri.parse(Constants.UMKM_PRODUCT_LATEST_URL),
    headers: Constants.HEADERS(_token ?? "")
  );
  if(response.statusCode == 200){
    List<dynamic> jsonResponse = json.decode(response.body)['data'];
    return jsonResponse.map((e) => new UmkmProduct.fromJson(e)).toList();
  }
  return umkmProduct;
}

File auth_repository.dart
Future<Either<Failure, User>> showUser() async {
  if (await networkInfo.isConnected) {
    try {
      final _token = await AppStorage.getToken('token');
      final response = await http.get(Uri.parse(Constants.USER_GET_URL),
        headers: Constants.HEADERS(_token ?? ""));
      if (response.statusCode == 200) {
        return Right(User.fromJson(jsonDecode(response.body)));
      } else {
        return Left(UserNotFound());
      }
    }
  }
}

```

```
        }
    } catch (e) {
        return Left(UserNotFound());
    }
} else {
    return Left(ServerFailure());
}
}
```

Lampiran 10 Kode Fitur Berita

```
File news_index.dart
class NewsIndex extends StatefulWidget {
  final BuildContext? menuScreenContext;
  final Function? onScreenHideButtonPressed;
  final bool? hideStatus;
  const NewsIndex(
    {Key? key,
    this.menuScreenContext,
    this.onScreenHideButtonPressed,
    this.hideStatus = false})
    : super(key: key);

  @override
  _NewsIndexState createState() => _NewsIndexState();
}

class _NewsIndexState extends State<NewsIndex> {
List<News> news = [];
late NewsBloc _newsBloc;
bool semua = true;
bool dibaca = false;
bool belumDibaca = false;

void _fetchNews() {
  news.clear();
  _newsBloc.add(GetAll());
}

@override
void initState() {
  _newsBloc = NewsBloc(newsRepository: NewsRepository(networkInfo: sl()));
  _fetchNews();
  super.initState();
}

@override
Widget build(BuildContext context) {
  Size size = MediaQuery.of(context).size;
  return BlocConsumer<NewsBloc, NewsState>(
    bloc: _newsBloc,
    listener: (context, state) {
      if (state is ShowMessage) {
        showDialog(
          context: context,
          builder: (context) {
            return AlertDialog(

```

```

        content: Text(state.message),
    );
})
} else if (state is NewsLoaded) {
    print('loaded');
    news.clear();
    news.addAll(state.news);
}
},
builder: (context, state) {
    return Container(
        margin: const EdgeInsets.only(bottom: 40),
        child: SizedBox(
            width: size.width,
            height: size.height,
            child: Scaffold(
                backgroundColor: HexColor('#FFFFFF'),
                body: SafeArea(
                    child: Padding(
                        padding: const EdgeInsets.only(top: 10, left: 10, right: 10),
                        child: Column(
                            children: <Widget>[
                                Expanded(
                                    child: Container(
                                        margin: const EdgeInsets.symmetric(vertical: 10),
                                        child: RefreshIndicator(
                                            onRefresh: () async {
                                                _fetchNews();
                                            },
                                            child: state is NewsLoading
                                                ? _isLoading(state)
                                                : _createNewsList(),
                                        ),
                                    ),
                                ),
                            ],
                        ))),
    );
},
);
}

Widget _createNewsList() {
    if (news.isNotEmpty) {
        return ListView.builder(
            scrollDirection: Axis.vertical,
            itemCount: news.length,
            itemBuilder: (context, index) {
                return Container(
                    height: 270,
                    child: new Card(
                        child: InkWell(
                            onTap: () {
                                _displayDetail(news[index]);
                            },
                            child: Container(
                                child: Column(

```



```

    });
}
return Container(child: Text('data'));
}

void _displayDetail(News news) {
  pushNewScreen(context, screen: NewsDetail(news: news), withNavBar: false);
}

File news_bloc.dart
@Override
Stream<NewsState> mapEventToState(
  NewsEvent event,
) async* {
  if (event is GetAll) {
    yield* _getAllNews();
  }
}

Stream<NewsState> _getAllNews() async* {
  try {
    yield NewsLoading();
    List<News> news = await newsRepository.fetchNews();
    yield NewsLoaded(news: news);
  } catch (e) {
    print(e);
    yield ShowMessage(message: e.toString());
  }
}

File news_repository.dart
Future<List<News>> fetchNews() async {
  List<News> news = [];
  final _token = await AppStorage.getToken('token');
  print(_token);
  final response = await http.get(
    Uri.parse(Constants.NEWS_INDEX_URL),
    headers: Constants.HEADERS(_token ?? "")
  );
  if(response.statusCode == 200){
    List<dynamic> jsonResponse = json.decode(response.body)['data'];
    return jsonResponse.map((e) => new News.fromJson(e)).toList();
  }
  return news;
}

```

Lampiran 11 Kode Fitur Detail Berita

```

class NewsDetail extends StatefulWidget {
  final News news;
  const NewsDetail({Key? key, required this.news}) : super(key: key);

  @override
  _NewsDetailState createState() => _NewsDetailState();
}

```

```

class _NewsDetailState extends State<NewsDetail> {
  @override
  Widget build(BuildContext context) {
    // Size size = MediaQuery.of(context).size;
    return Scaffold(
      appBar: AppBar(
        backgroundColor: HexColor('#043353'),
        centerTitle: true,
        title: Text('Detail Berita'),
      ),
      backgroundColor: HexColor('#FFFFFF'),
      body: Container(
        padding: const EdgeInsets.symmetric(horizontal: 10),
        child: SingleChildScrollView(
          child: Column(
            mainAxisSize: MainAxisSize.min,
            children: <Widget>[
              SizedBox(
                height: 10,
              ),
              Flexible(
                child: SizedBox(
                  height: 200,
                  width: double.infinity,
                  child: Card(
                    margin: EdgeInsets.all(10),
                    child: Padding(
                      padding: const EdgeInsets.all(10.0),
                      child: Image.network(widget.news.file == null ? "" : Constants.STORAGE_URL+widget.news.file!.path,
                        fit: BoxFit.fitHeight),
                    ),
                  ),
                ),
              )),
              Flexible(
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    SizedBox(
                      width: double.infinity,
                      child: Padding(
                        padding: const EdgeInsets.all(10.0),
                        child: Text(
                          widget.news.title,
                          textAlign: TextAlign.center,
                          style: TextStyle(
                            fontWeight: FontWeight.bold, fontSize: 18),
                        ),
                      ),
                    ),
                    SizedBox(
                      child: Padding(
                        padding: const EdgeInsets.symmetric(horizontal: 10.0),
                        child: Text(
                          widget.news.localTime,
                          textAlign: TextAlign.start,
                          style: TextStyle(
                            color: Colors.grey.shade600,
                            decoration: TextDecoration.underline),
                        ),
                      ),
                    ),
                  ],
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}

```

Lampiran 12 Kode Fitur Administrasi Surat

```
File letter_index.dart
class LetterIndex extends StatefulWidget {
    final BuildContext? menuScreenContext;
    final Function? onScreenHideButtonPressed;
    final bool? hideStatus;
    const LetterIndex(
        {Key? key,
        this.menuScreenContext,
        this.onScreenHideButtonPressed,
        this.hideStatus = false})
        : super(key: key);

    @override
    _LetterIndexState createState() => _LetterIndexState();
}

class _LetterIndexState extends State<LetterIndex> {
    bool semua = true;
    bool terkirim = false;
    bool proses = false;
    bool selesai = false;
    List<LetterRequest> letterRequest = [];
    late LetterRequestBloc letterRequestBloc;

    void _fetchLetterRequest() async {
```

```

        letterRequest.clear();
        letterRequestBloc.add(GetAll());
    }

    @override
    void initState() {
        letterRequestBloc = LetterRequestBloc(LetterRequestRepository(networkInfo: sl()));
        _fetchLetterRequest();
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        Size size = MediaQuery.of(context).size;
        return BlocConsumer<LetterRequestBloc, LetterRequestState>(
            bloc: letterRequestBloc,
            listener: (context, state) {
                if (state is ShowMessage) {
                    showDialog(
                        context: context,
                        builder: (context) {
                            return AlertDialog(
                                content: Text(state.message),
                            );
                        });
                } else if (state is LetterRequestLoaded) {
                    print('loaded');
                    letterRequest.clear();
                    letterRequest.addAll(state.letterRequest);
                }
            },
            builder: (context, state) {
                return Container(
                    // margin: const EdgeInsets.only(bottom: 40),
                    child: SizedBox(
                        width: size.width,
                        height: size.height,
                        child: Scaffold(
                            backgroundColor: HexColor('#FFFFFF'),
                            body: SafeArea(
                                child: Padding(
                                    padding: const EdgeInsets.only(top: 10, left: 10, right: 10),
                                    child: Column(
                                        children: [
                                            Expanded(
                                                child: Container(
                                                    margin: const EdgeInsets.symmetric(vertical: 10),
                                                    child: RefreshIndicator(
                                                        onRefresh: () async {
                                                            _fetchLetterRequest();
                                                        },
                                                        child: state is LetterRequestLoading
                                                            ? _isLoading(state)
                                                            : _buildLetterRequestList(state)),
                                                ),
                                            ),
                                        ],
                                    ),
                                ),
                            ),
                        ),
                    );
                );
            }
        );
    }
}

```

```

        ),
        ),
        ),
        ),
        );
    },
);
}

Widget _isLoading(state) {
if(state is LetterRequestLoading){
    return ListView.builder(
        itemCount: 6,
        itemBuilder: (context, index){
            return Container(
                height: 100,
                child: new Card(
                    child: Container(
                        height: 100,
                        decoration: BoxDecoration(
                            color: AppColors.white,
                            borderRadius: BorderRadius.all(Radius.circular(5))),
                    child: Row(children: [
                        Expanded(
                            flex: 1,
                            child: Container(
                                margin: EdgeInsets.all(10),
                                child: ShimmerWidget.circular(width: 50, height: 50)
                            ),
                        ),
                        VerticalDivider(),
                        Expanded(
                            flex: 3,
                            child: Container(
                                margin: EdgeInsets.all(10),
                                child: Column(
                                    mainAxisAlignment: MainAxisAlignment.center,
                                    crossAxisAlignment: CrossAxisAlignment.start,
                                    children: [
                                        ShimmerWidget.rectangular(width: 300, height: 20),
                                        SizedBox(height: 5,),
                                        ShimmerWidget.rectangular(width: 300, height: 10),
                                    ],
                                )));
                    ]));
            );
        });
    return Container(child: Text('data'));
}

Widget _buildLetterRequestList(state){
    return ListView.builder(
        scrollDirection: Axis.vertical,
        itemCount: letterRequest.length,
        itemBuilder: (context, index) {
            if(index == letterRequest.length-1){

```

```

return Column(
  children: [
    _cardListLetter(index),
    SizedBox(height: 50,),
  ],
);
} else{
  return _cardListLetter(index);
}
});

Widget _cardListLetter(index)
{
  return Container(
    height: 100,
    child: new Card(
      child: InkWell(
        onTap: () {
          pushNewScreen(context, screen: LetterRequestDetail(letterRequest: letterRequest[index]), withNavBar: false);
        },
        child: Container(
          height: 100,
          decoration: BoxDecoration(
            color: Colors.lightBlue.shade100,
            borderRadius: BorderRadius.all(Radius.circular(5))),
          child: Row(children: [
            Expanded(
              flex: 1,
              child: Container(
                margin: EdgeInsets.fromLTRB(20, 10, 10, 10),
                child: Center(
                  child: letterRequest[index].status == 'terkirim'
                    ? Image.asset(
                      "assets/images/letter_sent.png")
                    : (letterRequest[index].status == 'proses'
                      ? Image.asset(
                        "assets/images/letter_proses.png")
                      : Image.asset(
                        "assets/images/letter_done.png"))),
            ),
          )),
          VerticalDivider(),
        ],
      ),
    ),
  );
}

Expanded(
  flex: 3,
  child: Container(
    margin: EdgeInsets.all(10),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          letterRequest[index].title,
          style: TextStyle(
            fontSize: 14,
            fontWeight: FontWeight.bold),
          overflow: TextOverflow.ellipsis,
          softWrap: false,
        ),
      ],
    ),
  ),
);
}

```

```

        ),
        SizedBox(height: 5,),
        Text(letterRequest[index].detail,      softWrap:      true,      overflow:
TextOverflow.fade, style: TextStyle(fontSize: 12),
        ],
        )))
    ],
);
}
}

File letter_request_bloc.dart
@Override
Stream<LetterRequestState> mapEventToState(
    LetterRequestEvent event,
) async* {
    if (event is GetAll) {
        yield* _getAllLetterRequest();
    } else if (event is LetterRequestStore) {
        String? userToken = await AppStorage.getToken('user');
        User user = User.fromJson(jsonDecode(userToken!));
        var dataStore = {
            'title': event.title,
            'detail': event.detail,
            'file': event.file,
            'user_id': user.id
        };
        yield* _store(dataStore);
    }
}
Stream<LetterRequestState> _getAllLetterRequest() async* {
    try {
        yield LetterRequestLoading();
        List<LetterRequest> letterRequests = await letterRequestRepository.fetchLetterRequest();
        yield LetterRequestLoaded(letterRequests);
    } catch (e) {
        yield ShowMessage(message: e.toString());
    }
}

File letter_request_repository.dart
Future<List<LetterRequest>> fetchLetterRequest() async {
    List<LetterRequest> letterRequest = [];
    final _token = await AppStorage.getToken('token');
    final response = await http.get(
        Uri.parse(Constants.LETTER_REQUEST_INDEX_URL),
        headers: Constants.HEADERS(_token ?? "")
    );
    if(response.statusCode == 200){
        List<dynamic> jsonResponse = json.decode(response.body)['data'];
        return jsonResponse.map((e) => new LetterRequest.fromJson(e)).toList();
    }
    return letterRequest;
}

```

Lampiran 13 Kode Fitur Detail Surat

```

class LetterRequestDetail extends StatefulWidget {
  final LetterRequest letterRequest;
  const LetterRequestDetail({Key? key, required this.letterRequest})
      : super(key: key);

  @override
  _LetterRequestDetailState createState() => _LetterRequestDetailState();
}

class _LetterRequestDetailState extends State<LetterRequestDetail> {
  TextEditingController titleController = TextEditingController();
  TextEditingController detailController = TextEditingController();
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  String? _fileName;
  List<PlatformFile>? _paths;
  String? _directoryPath;
  String? _extension;
  bool _loadingPath = false;
  bool _multiPick = false;
  FileType _pickingType = FileType.any;
  TextEditingController _controller = TextEditingController();

  @override
  void initState() {
    _controller.addListener(() => _extension = _controller.text);
    titleController.text = widget.letterRequest.title;
    detailController.text = widget.letterRequest.detail;
    super.initState();
  }

  void _openFileExplorer() async {
    setState(() => _loadingPath = true);
    try {
      _directoryPath = null;
      _paths = (await FilePicker.platform.pickFiles(
        type: _pickingType,
        allowMultiple: _multiPick,
        onFileLoading: (FilePickerStatus status) => print(status),
        allowedExtensions: (_extension?.isNotEmpty ?? false)
            ? _extension?.replaceAll(' ', '').split(',')
            : null,
      ))
          ?.files;
    } on PlatformException catch (e) {
      print("Unsupported operation" + e.toString());
    } catch (ex) {
      print(ex);
    }
    if (!mounted) return;
    setState(() {
      _loadingPath = false;
      _fileName =
          _paths != null ? _paths!.map((e) => e.name).toString() : "...";
    });
  }

  void _clearCachedFiles() {
    FilePicker.platform.clearTemporaryFiles().then((result) {

```

```

ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
        backgroundColor: result! ? Colors.green : Colors.red,
        content: Text(
            (result ? 'File Berhasil Dihapus.' : 'Gagal Menghapus File')),
    ),
);
_paths!.clear();
_directoryPath = null;
});
}

@Override
Widget build(BuildContext ctx) {
    Size size = MediaQuery.of(ctx).size;
    return Scaffold(
        appBar: AppBar(
            backgroundColor: AppColors.darkBlue,
            centerTitle: true,
            title: Text('Pengajuan Surat'),
            floatingActionButton: widget.letterRequest.status == 'terkirim'
                ? FloatingActionButton(
                    child: Icon(Icons.check),
                    onPressed: () async {
                        BlocProvider.of<LetterRequestBloc>(context).add(
                            LetterRequestUpdate(
                                id: widget.letterRequest.id,
                                title: titleController.text,
                                detail: detailController.text,
                                file: _paths != null ? _paths!.first.path : null));
                    },
                )
                : null,
        body: BlocListener<LetterRequestBloc, LetterRequestState>(
            listener: (context, state) {
                if (state is LetterRequestLoading) {
                    EasyLoading.show(status: "Mengirim Surat");
                } else if (state is LetterRequestStoreSuccess) {
                    EasyLoading.dismiss();
                    ScaffoldMessenger.of(context).showSnackBar(
                        SnackBar(content: Text('Surat Pengajuan Berhasil Dikirim')));
                    Future.delayed(
                        Duration(seconds: 2),
                        () => Navigator.of(context)
                            .pushNamedAndRemoveUntil('/home', (route) => false));
                } else if (state is ShowMessage) {
                    EasyLoading.dismiss();
                    ScaffoldMessenger.of(context)
                        .showSnackBar(SnackBar(content: Text(state.message)));
                }
            },
        child: SingleChildScrollView(
            child: Center(
                child: Column(
                    mainAxisSize: MainAxisSize.min,
                    children: <Widget>[
                        SizedBox(
                            height: 30,

```

```
),
BlocBuilder<LetterRequestBloc, LetterRequestState>(
  builder: (context, state) {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Container(
          alignment: Alignment.center,
          margin: EdgeInsets.symmetric(horizontal: 40),
          child: TextField(
            controller: titleController,
            decoration: InputDecoration(
              enabled: widget.letterRequest.status == 'terkirim' ? true : false,
              labelText: "Keperluan",
              border: OutlineInputBorder(
                borderRadius:
                  BorderRadius.all(Radius.circular(10))),
            ),
          ),
        ),
        SizedBox(
          height: 20,
        ),
        Container(
          alignment: Alignment.center,
          margin: EdgeInsets.symmetric(horizontal: 40),
          child: TextField(
            controller: detailController,
            enabled: widget.letterRequest.status == 'terkirim' ? true : false,
            decoration: InputDecoration(
              labelText: "Detail",
              border: OutlineInputBorder(
                borderRadius:
                  BorderRadius.all(Radius.circular(10))),
            ),
            maxLength: 200,
            maxLines: null,
            keyboardType: TextInputType.multiline,
          ),
        ),
        Container(
          margin: EdgeInsets.symmetric(horizontal: 40),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              Container(
                child: Text(
                  'Status: ${widget.letterRequest.status.toUpperCase()}'),
              ),
              if (_paths == null)
                ElevatedButton(
                  onPressed: () async {
                    var url = STORAGE_URL +
                      widget.letterRequest.file!.path;
                    await canLaunch(url)
                      ? await launch(url)
                      : throw 'Could not launch $url';
                },
            ),
          ],
        ),
      ],
    );
  }
);
```

```

        child: Text("Download Lampiran")),
    ],
),
),
),
if (_paths != null && _paths!.isNotEmpty)
Container(
    margin: EdgeInsets.symmetric(horizontal: 40),
    child: ElevatedButton(
        onPressed: () async {
            _clearCachedFiles();
        },
        child: Text('- Hapus'),
        style: ButtonStyle(
            backgroundColor:
                MaterialStateProperty.all<Color>(
                    Colors.red)),
    )), if (_paths == null && widget.letterRequest.status == 'terkirim')
Container(
    width: size.width,
    height: 50,
    margin: EdgeInsets.symmetric(horizontal: 40),
    child: ElevatedButton(
        onPressed: () async {
            _openFileExplorer();
        },
        child: Text('Ubah Lampiran'),
        style: ButtonStyle(
            backgroundColor:
                MaterialStateProperty.all<Color>(
                    AppColors.lightblue)),
    )), Container(
    alignment: Alignment.center,
    margin: EdgeInsets.symmetric(horizontal: 20),
    child: Builder(
        builder: (BuildContext context) => _loadingPath
            ? Padding(
                padding:
                    const EdgeInsets.only(bottom: 10.0),
                child: const CircularProgressIndicator(),
            )
            : _directoryPath != null
            ? ListTile(
                title: const Text('Direcory path'),
                subtitle: Text(_directoryPath!),
            )
            : _paths != null
            ? Container(
                padding: const EdgeInsets.only(
                    bottom: 30.0),
                height: MediaQuery.of(context)
                    .size
                    .height *
                    0.50,
                child: Scrollbar(
                    child: ListView.separated(

```

Lampiran 14 Kode Fitur Profil

```

class ProfileIndex extends StatefulWidget {
  final BuildContext? menuScreenContext;
  final Function? onScreenHideButtonPressed;
  final bool? hideStatus;
  const ProfileIndex(
    {Key? key,
    this.menuScreenContext,
    this.onScreenHideButtonPressed,
    this.hideStatus = false})
    : super(key: key);
  @override
  _ProfileIndexState createState() => _ProfileIndexState();
}

class _ProfileIndexState extends State<ProfileIndex> {
  late ProfileBloc _profileBloc;

  User? user;
  List<Umkm> umkm = [];
  List<LetterRequest> letterRequest = [];
  List<Aspiration> aspiration = [];

  @override
  void initState() {
    _profileBloc = ProfileBloc(
      umkmRepository: UmkmRepository(networkInfo: sl()),
      authRepository: AuthRepository(networkInfo: sl()),
      letterRequestRepository: LetterRequestRepository(networkInfo: sl()),
      aspirationRepository: AspirationRepository(networkInfo: sl()));
    _fetchProfile();
    super.initState();
  }

  void _fetchProfile() async {
    _profileBloc.add(GetDetails());
  }

  @override
  Widget build(BuildContext context) {
    Size size = MediaQuery.of(context).size;
    return BlocConsumer<ProfileBloc, ProfileState>(
      bloc: _profileBloc,
      listener: (context, state) {
        if (state is ShowMessage) {
          showDialog(
            context: context,
            builder: (context) {
              return AlertDialog(
                content: Text(state.message),
              );
            });
        } else if (state is ProfileLoaded) {
          umkm.clear();
          letterRequest.clear();
          aspiration.clear();
        }
        umkm.addAll(state.umkm);
        letterRequest.addAll(state.letterRequest);
      },
    );
  }
}

```

```
aspiration.addAll(state.aspiration);
user = state.user;
print(state.user);
},
builder: (context, state) {
return Scaffold(
appBar: AppBar(
backgroundColor: AppColors.whiteOri,
title: Text(
'Akun Saya',
style: TextStyle(
color: Colors.grey.shade800, fontWeight: FontWeight.w800),
),
actions: [
GestureDetector(
onTap: () {
pushNewScreen(context,
screen: PengaturanIndex(), withNavBar: false);
},
child: Icon(Icons.settings, color: Colors.grey, size: 25),
),
SizedBox(
width: 10,
),
SizedBox(
width: 10,
),
GestureDetector(
onTap: () {
pushNewScreen(context,
screen: NotificationIndex(), withNavBar: false);
},
child: Icon(Icons.notifications, color: Colors.grey, size: 25),
),
SizedBox(
width: 10,
),
],
),
body: SafeArea(
child: RefreshIndicator(
onRefresh: () async {
_fetchProfile();
},
child: Container(
width: size.width,
height: size.height,
child: SingleChildScrollView(
child: Column(
children: [
Container(
margin:
EdgeInsets.symmetric(horizontal: 20, vertical: 10),
child: Row(
children: [
Expanded(
flex: 1,
```

```

child: state is ProfileLoading
    ? ShimmerWidget.circular(
        width: 60, height: 60)
    : ClipRRect(
        borderRadius: BorderRadius.circular(40),
        child: Image.asset(
            'assets/images/avatar.png',
            width: 60,
            height: 60,
        )));
SizedBox(
    width: 10,
),
Expanded(
    flex: 4,
    child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            state is ProfileLoading
                ? ShimmerWidget.rectangular(
                    width: 150, height: 15)
                : Text(
                    user != null
                        ? user!.userDetail!.namaLengkap!
                        : 'Pengguna',
                    style: TextStyle(
                        color: Colors.grey.shade900,
                        fontWeight: FontWeight.w600,
                        fontSize: 18)),
            SizedBox(
                height: 5,
),
            state is ProfileLoading
                ? ShimmerWidget.rectangular(
                    width: 200, height: 10)
                : Text(
                    user != null
                        ? user!.userDetail!.nik!
                        : 'Nomor NIK',
                    style: TextStyle(
                        color: Colors.grey.shade400,
                        fontSize: 14)),
        ],
),
),
],
),
),
),
),
),
Container(
    height: 140,
    margin: EdgeInsets.all(10),
    decoration: BoxDecoration(
        borderRadius: BorderRadius.all(Radius.circular(10)),
        boxShadow: [
            new BoxShadow(
                color: Colors.grey.shade100, blurRadius: 4)
        ]),
    child: Card(

```

```

child: Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    Expanded(
      flex: 1,
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Icon(Icons.shopping_bag,
            size: 30, color: HexColor('#18A4E0')),
          SizedBox(
            height: 5,
          ),
          Text('UMKM',
            style: TextStyle(
              color: Colors.grey.shade800,
              fontSize: 14)),
          SizedBox(
            height: 10,
          ),
        ],
        state is ProfileLoading
        ? ShimmerWidget.rectangular(
          width: 60, height: 15)
        : Text(
          "${umkm.length.toString()} Produk",
          style: TextStyle(
            color: Colors.black,
            fontSize: 16,
            fontWeight: FontWeight.bold))
      ],
    ),
  ],
),
Container(
  margin: const EdgeInsets.symmetric(vertical: 40),
  child: VerticalDivider(
    color: Colors.grey.shade400,
  ),
),
Expanded(
  flex: 1,
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Icon(Icons.assignment_outlined,
        size: 30, color: HexColor('#18A4E0')),
      SizedBox(
        height: 5,
      ),
      Text('Administrasi',
        style: TextStyle(
          color: Colors.grey.shade800,
          fontSize: 14)),
      SizedBox(
        height: 10,
      ),
      state is ProfileLoading
      ? ShimmerWidget.rectangular(

```

```

        width: 60, height: 15)
    : Text(
        "${letterRequest.length.toString()} Surat",
        style: TextStyle(
            color: Colors.black,
            fontSize: 16,
            fontWeight: FontWeight.bold))
    ],
    )),
Container(
margin:
    const EdgeInsets.symmetric(vertical: 40),
child: VerticalDivider(
    color: Colors.grey.shade400,
),
),
Expanded(
flex: 1,
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: <Widget>[
    Icon(Icons.campaign,
        size: 30, color: HexColor('#18A4E0')),
    SizedBox(
        height: 5,
    ),
    Text('Aspirasi',
        style: TextStyle(
            color: Colors.grey.shade800,
            fontSize: 14)),
    SizedBox(
        height: 10,
    ),
],
state is ProfileLoading
    ? ShimmerWidget.rectangular(
        width: 60, height: 15)
    : Text(
        "${aspiration.length.toString()} Suara",
        style: TextStyle(
            color: Colors.black,
            fontSize: 16,
            fontWeight: FontWeight.bold))
],
),
],
),
),
),
),
Container(
margin: const EdgeInsets.only(left: 20, top: 20),
child: Column(
children: [
Column(
children: [
Container(
margin: const EdgeInsets.only(bottom: 10),
child: Align(
    alignment: Alignment.centerLeft,

```

```

        child: Text('Layanan',
            style: GoogleFonts.poppins(
                fontWeight: FontWeight.bold,
                fontSize: 18)),
        ),
        ),
        InkWell(
        onTap: () {
            pushNewScreen(context,
                screen: AgendaIndex(),
                withNavBar: false);
        },
        child: Padding(
            padding: const EdgeInsets.all(10.0),
            child: Row(
                mainAxisAlignment:
                    MainAxisAlignment.spaceBetween,
                children: <Widget>[
                    Column(
                        crossAxisAlignment:
                            CrossAxisAlignment.start,
                        children: [
                            Text(
                                'Agenda',
                                style: TextStyle(
                                    color: Colors.grey.shade800,
                                ),
                            ),
                            SizedBox(
                                height: 5,
                            ),
                            Text(
                                'Lihat Jadwal kegiatan di desa',
                                style: TextStyle(
                                    color:
                                        Colors.grey.shade400,
                                ),
                            ),
                        ],
                    ),
                    Container(
                        margin:
                            EdgeInsets.only(right: 10),
                        child: Icon(
                            Icons.navigate_next,
                            color: Colors.grey.shade400,
                        )),
                ],
            ),
        ),
        Divider(
            color: Colors.grey.shade400,
        ),
        InkWell(
        onTap: () {
            // Navigator.of(context).pushNamed('/aspiration-index');
            pushNewScreen(context,
                screen: AspirationIndex(),
        ),
    
```



```

'Lihat Produk UMKM yang ada pada desa',
style: TextStyle(
  color:
    Colors.grey.shade400),
),
],
),
Container(
margin:
  EdgeInsets.only(right: 10),
child: Icon(
  Icons.navigate_next,
  color: Colors.grey.shade400,
)),
],
),
),
),
),
Divider(
color: Colors.grey.shade400,
),
),
InkWell(
onTap: () {
  pushNewScreen(context,
    screen: UserSkillIndex(),
    withNavBar: false);
},
child: Padding(
padding: const EdgeInsets.all(10.0),
child: Row(
mainAxisAlignment:
  MainAxisAlignment.spaceBetween,
children: <Widget>[
  Column(
crossAxisAlignment:
  CrossAxisAlignment.start,
children: [
  Text(
    'Potensi Desa',
    style: TextStyle(
      color:
        Colors.grey.shade800),
  ),
  SizedBox(
    height: 5,
  ),
  Text(
    'Tambahkan Keahlian yang anda miliki',
    style: TextStyle(
      color:
        Colors.grey.shade400),
  ),
],
),
Container(
margin:
  EdgeInsets.only(right: 10),
child: Icon(

```

```

        Icons.navigate_next,
        color: Colors.grey.shade400,
    )),
],
),
),
),
),
Divider(
    color: Colors.grey.shade400,
),
InkWell(
    onTap: () {
        pushNewScreen(context,
            screen: ProfileDesa(),
            withNavBar: false);
    },
),
child: Padding(
    padding: const EdgeInsets.all(10.0),
),
child: Row(
    mainAxisAlignment:
        MainAxisAlignment.spaceBetween,
),
children: <Widget>[
    Column(
        crossAxisAlignment:
            CrossAxisAlignment.start,
    ),
    Text(
        'Tentang Desa',
        style: TextStyle(
            color:
                Colors.grey.shade800),
    ),
    SizedBox(
        height: 5,
    ),
),
Text(
    'Informasi Terkait desa',
    style: TextStyle(
        color:
            Colors.grey.shade400),
),
],
),
),
Container(
    margin:
        EdgeInsets.only(right: 10),
),
child: Icon(
    Icons.navigate_next,
    color: Colors.grey.shade400,
)),
],
),
),
),
),
Divider(
    color: Colors.grey.shade400,
),
),
SizedBox(

```

```
height: 80,
),
],
),
),
),
),
),
),
),
),
),
);
},
);
}
}

File profile_bloc.dart
Stream<ProfileState> mapEventToState(
  ProfileEvent event,
) async* {
  if (event is GetDetails) {
    yield* _getDetails();
  }
}

Stream<ProfileState> _getDetails() async* {
  try {
    yield ProfileLoading();
    Either<Failure, User> response = await authRepository.showUser();
    yield* response.fold((failure) async*{
      print(failure);
    }, (u) async*{
      User user = u;
      List<Umkm> umkm = await umkmRepository.fetchUmkms();
      List<LetterRequest> letterRequest =
        await letterRequestRepository.fetchLetterRequest();
      List<Aspiration> aspiration =
        await aspirationRepository.fetchAspiration();
      yield ProfileLoaded(user: user, umkm: umkm, letterRequest: letterRequest, aspiration: aspiration);
    });
  } catch (e) {
    print(e);
    yield ShowMessage(message: e.toString());
  }
}
```

Lampiran 15 Kode Fitur UMKM

```
class UmkmIndex extends StatefulWidget {  
  final BuildContext? menuScreenContext;  
  final Function? onScreenHideButtonPressed;  
  final bool? hideStatus;  
  const UmkmIndex(  
    {Key? key,
```

```

this.menuScreenContext,
this.onScreenHideButtonPressed,
this.hideStatus = false;
)
: super(key: key);

@Override
_UmkmIndexState createState() => _UmkmIndexState();
}

class _UmkmIndexState extends State<UmkmIndex> {
List<Umkm> umkm = [];
late UmkmBloc umkmBloc;

void _fetchUmkm() async {
umkm.clear();
umkmBloc.add(GetAll());
}

@Override
void initState() {
umkmBloc = UmkmBloc(UmkmRepository(networkInfo: sl()));
_fetchUmkm();
super.initState();
}

@Override
Widget build(BuildContext context) {
Size size = MediaQuery.of(context).size;
return BlocConsumer<UmkmBloc, UmkmState>(
bloc: umkmBloc,
listener: (context, state) {
if (state is ShowMessage) {
showDialog(
context: context,
builder: (context) {
return AlertDialog(
content: Text(state.message),
);
});
} else if (state is UmkmLoaded) {
print('loaded');
umkm.clear();
umkm.addAll(state.umkm);
} else if (state is UmkmStoreSuccess) {
_fetchUmkm();
}
},
builder: (context, state) {
return Container(
child: SizedBox(
width: size.width,
height: size.height,
child: Scaffold(
appBar: AppBar(
title: Text('UMKM'),
),
backgroundColor: HexColor('#FFFFFF'),
body: SafeArea(

```

```
child: Padding(  
padding: const EdgeInsets.only(top: 10, left: 10, right: 10),  
child: Column(  
children: [  
Expanded(  
child: Container(  
margin: const EdgeInsets.symmetric(vertical: 10),  
child: RefreshIndicator(  
onRefresh: () async {  
_fetchUmkm();  
},  
child: state is UmkmLoading  
? _isLoading(state)  
: _buildUmkmList(state)),  
),  
),  
SizedBox(  
height: 20,  
)  
],  
),  
,  
,  
,  
,  
,  
,  
);  
},  
);  
}  
  
Widget _isLoading(state) {  
if (state is UmkmLoading) {  
return ListView.builder(  
scrollDirection: Axis.vertical,  
itemCount: 5,  
itemBuilder: (context, index) {  
return Container(  
height: 150,  
child: new Card(  
child: InkWell(  
onTap: () {},  
child: Container(  
child: ClipRRect(  
borderRadius:  
BorderRadius.all(Radius.circular(5.0)),  
child: Stack(children: <Widget>[  
// Image.network(item, fit: BoxFit.cover, width: 1000.0),  
ShimmerWidget.rectangular(width: 1000, height: 150),  
Positioned(  
bottom: 0.0,  
left: 0.0,  
right: 0.0,  
child: Container(  
padding: EdgeInsets.symmetric(  
vertical: 10.0, horizontal: 20.0),  
child:  
ShimmerWidget.rectangular(width: 100, height: 20),  
),
```

```

),
Positioned(
  top: 0.0,
  left: 0.0,
  right: 0.0,
  child: Container(
    padding: EdgeInsets.symmetric(
      vertical: 10.0, horizontal: 20.0),
    child: ShimmerWidget.rectangular(width: 100, height: 20),
  ),
)
))))),
);
});
}
return Container(child: Text('data'));
}

Widget _buildUmkmList(state) {
  return ListView.builder(
    scrollDirection: Axis.vertical,
    itemCount: umkm.length,
    itemBuilder: (context, index) {
      return Container(
        height: 150,
        child: new Card(
          child: InkWell(
            onTap: () {
              pushNewScreen(context, screen: UmkmDetail(umkmId: umkm[index].id,));
            },
            child: Container(
              child: ClipRRect(
                borderRadius:
                  BorderRadius.all(Radius.circular(5.0)),
                child: Stack(children: <Widget>[
                  // Image.network(item, fit: BoxFit.cover, width: 1000.0),
                  Image.network(
                    umkm[index].file != null
                      ? STORAGE_URL + umkm[index].file!.path
                      : STORAGE_URL + 'no-image.png',
                    fit: BoxFit.cover,
                    width: 1000.0),
                  Positioned(
                    bottom: 0.0,
                    left: 0.0,
                    right: 0.0,
                    child: Container(
                      decoration: BoxDecoration(
                        gradient: LinearGradient(
                          colors: [
                            Color.fromRGBO(200, 0, 0, 0),
                            Color.fromRGBO(0, 0, 0, 0)
                          ],
                          begin: Alignment.bottomCenter,
                          end: Alignment.topCenter,
                        ),
                      ),
                    ),
                  ),
                ],
                padding: EdgeInsets.symmetric(

```

```

        vertical: 10.0, horizontal: 20.0),
      child: Text(
        umkm[index].name,
        style: TextStyle(
          color: Colors.white,
          fontSize: 16.0,
          fontWeight: FontWeight.bold,
        ),
        softWrap: true,
        overflow: TextOverflow.clip,
      ),
    ),
  ),
  Positioned(
    top: 0.0,
    left: 0.0,
    right: 0.0,
    child: Container(
      decoration: BoxDecoration(
        gradient: LinearGradient(
          colors: [
            Color.fromRGBO(0, 0, 0, 0),
            Color.fromRGBO(200, 0, 0, 0)
          ],
          begin: Alignment.bottomCenter,
          end: Alignment.topCenter,
        ),
      ),
      padding: EdgeInsets.symmetric(
        vertical: 10.0, horizontal: 20.0),
      child: Text(
        "Oleh: " +
        umkm[index]
          .user
          .userDetail!
          .namaLengkap!,
        style: TextStyle(
          color: Colors.white,
          fontSize: 12.0,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  )));
}
);
}
);

File umkm_bloc.dart
@Override
Stream<UmkmState> mapEventToState(
  UmkmEvent event,
) async* {
  if (event is GetAll) {
    yield* _getAllUmkm();
  }
}

```

```

}

Stream<UmkmState> _getAllUmkm() async* {
try {
yield UmkmLoading();
List<Umkm> umkm = await umkmRepository.fetchUmkms();
yield UmkmLoaded(umkm);
} catch (e) {
print(e);
yield ShowMessage(message: e.toString());
}
}

File umkm_repository.dart
Future<List<Umkm>> fetchUmkms() async {
List<Umkm> umkm = [];
final _token = await AppStorage.getToken('token');
final response = await http.get(Uri.parse(Constants.UMKM_INDEX_URL),
headers: Constants.HEADERS(_token ?? ""));
if (response.statusCode == 200) {
List<dynamic> jsonResponse = json.decode(response.body)['data'];
return jsonResponse.map((e) => new Umkm.fromJson(e)).toList();
}
return umkm;
}
}

```

Lampiran 16 Kode Fitur Produk UMKM

```

class UmkmProductIndex extends StatefulWidget {
final BuildContext? menuScreenContext;
final Function? onScreenHideButtonPressed;
final bool? hideStatus;
const UmkmProductIndex(
{Key? key,
this.menuScreenContext,
this.onScreenHideButtonPressed,
this.hideStatus = false})
: super(key: key);

@Override
_UmkmProductIndexState createState() => _UmkmProductIndexState();
}

class _UmkmProductIndexState extends State<UmkmProductIndex> {
List<UmkmProduct> umkmProduct = [];
late UmkmProductBloc umkmProductBloc;

void _fetchUmkmProduct() async {
umkmProduct.clear();
umkmProductBloc.add(GetAll());
}

@Override
void initState() {
umkmProductBloc = UmkmProductBloc(
umkmProductRepository: UmkmProductRepository(networkInfo: sl()));
fetchUmkmProduct();
}
}

```

```

super.initState();
}

@Override
Widget build(BuildContext context) {
Size size = MediaQuery.of(context).size;
return BlocConsumer<UmkmProductBloc, UmkmProductState>(
  bloc: umkmProductBloc,
  listener: (context, state) {
    if (state is ShowMessage) {
      showDialog(
        context: context,
        builder: (context) {
          return AlertDialog(
            content: Text(state.message),
          );
        });
    } else if (state is UmkmProductLoaded) {
      print('loaded');
      umkmProduct.clear();
      umkmProduct.addAll(state.umkmProduct);
    }
  },
  builder: (context, state) {
    return Container(
      child: SizedBox(
        width: size.width,
        height: size.height,
        child: Scaffold(
          appBar: AppBar(
            title: Text('Produk UMKM'),
          ),
          backgroundColor: HexColor('#FFFFFF'),
          body: SafeArea(
            child: Padding(
              padding: const EdgeInsets.only(top: 10, left: 10, right: 10),
              child: Column(
                children: [
                  Container(
                    height: 50,
                    child: TextField(
                      decoration: InputDecoration(
                        contentPadding:
                          const EdgeInsets.symmetric(vertical: 10),
                        filled: true,
                        fillColor: Colors.grey[200],
                        prefixIcon: Icon(Icons.search),
                        border: OutlineInputBorder(
                          borderRadius:
                            BorderRadius.all(Radius.circular(10.0)),
                          borderSide: BorderSide.none),
                        hintText: 'Cari Produk UMKM'),
                  ),
                  ),
                  Expanded(
                    child: Container(
                      margin: const EdgeInsets.symmetric(vertical: 10),
                      child: RefreshIndicator(

```

```

        onRefresh: () async {
            _fetchUmkmProduct();
        },
        child: state is UmkmProductLoading
            ? _isLoading(state)
            : _buildUmkmProductList(state, context),
    ),
),
SizedBox(
    height: 20,
)
],
),
),
),
),
),
),
),
);
},
);
}
}

Widget _isLoading(state) {
if (state is UmkmProductLoading) {
return GridView.builder(
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
crossAxisCount: 2, childAspectRatio: 0.75),
itemCount: 6,
shrinkWrap: true,
itemBuilder: (context, index) {
return Container(
child: Card(
semanticContainer: true,
child: InkWell(
onTap: () {}),
child: Container(
color: Colors.white,
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
ClipRRect(
borderRadius: BorderRadius.only(
topLeft: Radius.circular(5.0),
topRight: Radius.circular(5.0),
),
child: ShimmerWidget.rectangular(
width:
MediaQuery.of(context).size.width * 0.5,
height: 150)),
Container(
padding: EdgeInsets.fromLTRB(10, 10, 10, 5),
child: ShimmerWidget.rectangular(
width: 150, height: 15)),
Container(
padding: EdgeInsets.symmetric(horizontal: 10),
child: ShimmerWidget.rectangular(
width: 70, height: 10)),
SizedBox(

```

```

        height: 5,
    ),
    Container(
        padding: EdgeInsets.symmetric(horizontal: 10),
        child: ShimmerWidget.rectangular(
            width: 60, height: 10),
    ],
),
),
),
),
),
);
});
}
return Container(child: Text('data'));
}

Widget _buildUmkmProductList(state, context) {
Size size = MediaQuery.of(context).size;
return GridView.builder(
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
crossAxisCount: 2,
childAspectRatio: (size.width / 3.0) / (size.height / 3.8),
itemCount: umkmProduct.length,
shrinkWrap: true,
itemBuilder: (context, index) {
return Container(
child: Card(
semanticContainer: true,
child: InkWell(
onTap: () {
pushNewScreen(context,
screen: UmkmProductDetail(
umkmProductId: umkmProduct[index].id),
withNavBar: false);
},
child: Container(
color: Colors.white,
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
ClipRRect(
borderRadius: BorderRadius.only(
topLeft: Radius.circular(5.0),
topRight: Radius.circular(5.0),
),
child: Image.network(
umkmProduct[index].file != null
? STORAGE_URL + umkmProduct[index].file!.path
: STORAGE_URL + 'no-image.png',
fit: BoxFit.cover,
height: 150,
)),
Container(
padding: EdgeInsets.fromLTRB(10, 10, 10, 5),
child: Text(
umkmProduct[index].name,
style: TextStyle(
fontSize: 16,

```

```

),
overflow: TextOverflow.ellipsis,
softWrap: true,
),
),
Container(
padding: EdgeInsets.symmetric(horizontal: 10),
child: Text(
"Rp ${umkmProduct[index].price}",
style: TextStyle(
fontSize: 16, fontWeight: FontWeight.bold),
overflow: TextOverflow.ellipsis,
softWrap: true,
),
),
SizedBox(
height: 5,
),
Container(
padding: EdgeInsets.symmetric(horizontal: 10),
child: Text(
"Stok: ${umkmProduct[index].stock}",
style: TextStyle(fontSize: 14),
overflow: TextOverflow.ellipsis,
softWrap: true,
),
),
),
],
),
)),
),
),
);
});
}
}
}

File umkm_product_bloc.dart
@Override
Stream<UmkmProductState> mapEventToState(
UmkmProductEvent event,
) async* {
if (event is GetAll) {
yield* _getAllUmkmProduct();
}
}
Stream<UmkmProductState> _getAllUmkmProduct() async* {
try {
yield UmkmProductLoading();
List<UmkmProduct> umkmProduct = await
umkmProductRepository.fetchUmkmProducts();
yield UmkmProductLoaded(umkmProduct);
} catch (e) {
print(e);
yield ShowMessage(message: e.toString());
}
}
}

File umkm_product_repository

```

```
Future<List<UmkmProduct>> fetchUmkmProducts() async{
  List<UmkmProduct> umkmProduct = [];
  final _token = await AppStorage.getToken('token');
  final response = await http.get(
    Uri.parse(Constants.UMKM_PRODUCT_INDEX_URL),
    headers: Constants.HEADERS(_token ?? "")
  );
  if(response.statusCode == 200){
    List<dynamic> jsonResponse = json.decode(response.body)['data'];
    return jsonResponse.map((e) => new UmkmProduct.fromJson(e)).toList();
  }
  return umkmProduct;
}
```

Lampiran 17 Kode Fitur Keahlian Penduduk

```
class UserSKillIndex extends StatefulWidget {
  final BuildContext? menuScreenContext;
  final Function? onScreenHideButtonPressed;
  final bool? hideStatus;
  const UserSKillIndex(
    {Key? key,
    this.menuScreenContext,
    this.onScreenHideButtonPressed,
    this.hideStatus = false})
    : super(key: key);

  @override
  _UserSKillIndexState createState() => _UserSKillIndexState();
}

class _UserSKillIndexState extends State<UserSKillIndex> {
  List<UserSkill> userSkill = [];
  late UserSkillBloc userSkillBloc;

  void _fetchUserSkill() async {
    userSkill.clear();
    userSkillBloc.add(GetAll());
  }

  @override
  void initState() {
    userSkillBloc = UserSkillBloc(
      userSkillRepository: UserSkillRepository(networkInfo: sl()));
    _fetchUserSkill();
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    Size size = MediaQuery.of(context).size;
    return BlocConsumer<UserSkillBloc, UserSkillState>(
      bloc: userSkillBloc,
      listener: (context, state) {
        if (state is ShowMessage) {
```



```

Widget _isLoading(state) {
  if (state is UserSkillLoading) {
    return ListView.builder(
      scrollDirection: Axis.vertical,
      itemCount: 10,
      itemBuilder: (context, index) {
        return Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Container(
              height: 60,
              child: InkWell(
                onTap: () {},
                child: Container(
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.all(Radius.circular(5))),
                  child: Container(
                    margin: EdgeInsets.symmetric(horizontal: 20, vertical: 10),
                    child: Column(
                      mainAxisAlignment: MainAxisAlignment.center,
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        ShimmerWidget.rectangular(width: MediaQuery.of(context).size.width*0.8,
height: 15),
                        SizedBox(
                          height: 5,
                        ),
                        ShimmerWidget.rectangular(width: MediaQuery.of(context).size.width*0.8,
height: 10)
                      ],
                    )));
            ),
            Divider(
              thickness: 2,
              color: Colors.grey.shade400,
            ),
          ],
        );
      });
    }
    return Container(child: Text('data'));
  }
}

Widget _buildUserSkillList(state) {
  return ListView.builder(
    scrollDirection: Axis.vertical,
    itemCount: userSkill.length,
    itemBuilder: (context, index) {
      return Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Container(
            height: 60,
            child: InkWell(
              onTap: () {},
              child: Container(
                decoration: BoxDecoration(

```

```
borderRadius: BorderRadius.all(Radius.circular(5))),  
child: Container(  
    margin: EdgeInsets.symmetric(horizontal: 20, vertical: 10),  
    child: Column(  
        mainAxisSize: MainAxisSize.center,  
        crossAxisAlignment: CrossAxisAlignment.start,  
        children: [  
            Text(  
                userSkill[index].title,  
                style: TextStyle(  
                    fontSize: 14,  
                    fontWeight: FontWeight.bold),  
                overflow: TextOverflow.ellipsis,  
                softWrap: true,  
            ),  
            SizedBox(  
                height: 5,  
            ),  
            Text(  
                userSkill[index].detail,  
                style: TextStyle(  
                    fontSize: 12,  
                    color: Colors.grey.shade400),  
                overflow: TextOverflow.ellipsis,  
                softWrap: true,  
            ),  
        ],  
    )),  
    Divider(  
        thickness: 2,  
        color: Colors.grey.shade400,  
    ),  
],  
);  
});  
}  
}
```

```
File user_skill_bloc.dart
Stream<UserSkillState> mapEventToState(
    UserSkillEvent event,
) async* {
    if (event is GetAll) {
        yield* _getAllUserSkill();
    }
}
Stream<UserSkillState> _getAllUserSkill() async* {
    try {
        yield UserSkillLoading();
        List<UserSkill> userSkill = await userSkillRepository.fetchUserSkill();
        yield UserSkillLoaded(userSkill: userSkill);
    } catch (e) {
        print(e);
        yield ShowMessage(message: e.toString());
    }
}
```

File user skill repository.dart

```
Future<List<UserSkill>> fetchUserSkill() async {
    List<UserSkill> userSkill = [];
    final _token = await AppStorage.getToken('token');
    final response = await http.get(Uri.parse(Constants.USER_SKILL_INDEX_URL),
        headers: Constants.HEADERS(_token ?? ""));
    if (response.statusCode == 200) {
        List<dynamic> jsonResponse = json.decode(response.body)['data'];
        return jsonResponse.map((e) => new UserSkill.fromJson(e)).toList();
    }
    return userSkill;
}
```

Lampiran 18 Kode Fitur Login Administrator

```
File AuthController.php

public function loginView(){
    return view('auth.login');
}

public function login(Request $request){
    $this->validate($request,[
        'email'=>['required', 'email'],
        'password'=>['required','string'],
    ]);
    $admin = Admin::whereEmail($request->email)->first();
    if($admin){
        if(Hash::check($request->password,$admin->password)){
            if(Auth::guard('web')->attempt(['email' => $request->email, 'password' => $request->password], $request->remember))
            {
                $request->session()->regenerate();
                return redirect()->intended(route('admin.dashboard'));
            }
            return redirect()->back()->withErrors([
                'email' => 'Email tidak cocok',
                'password' => 'Password tidak cocok.'
            ])->withInput(['email' => $request->email]);
        }
        return redirect()->back()->withErrors(['password' => 'Password Salah.'])->withInput(['email' => $request->email]);
    }
    return redirect()->back()->withErrors(['email' => 'Email belum terdaftar pada sistem.'])->withInput(['email' => $request->email]);
}
```

Lampiran 19 Kode Fitur Dashboard Administrator

```
File HomeController.dart

public function dashboardView(){
    //
    $n_penduduk = User::count();
    $n_user_gender = Gender::withCount('user_detail')->get();
    $n_user_active = User::where('email_verified_at', '!=', null)->count();
    //
```

```

$aspiration = Aspiration::all();
if(count($aspiration) > 0){
    $top_aspirative_user = User::with('aspiration','file','user_detail')-
>withCount('aspiration')->get()->take(5);
} else{
    $top_aspirative_user = [];
}
//
$n_umkm = Umkm::count();
$n_product = UmkmProduct::count();
$n_news = News::count();
$n_skill = UserSkill::count();
//
$top_umkm = [];
$month = [now()->subMonth(), now()];
foreach($month as $value){
    $umkm = Umkm::with('file')->withCount(['umkm_product' => function($query)
use($value){
    $query->whereMonth('created_at', $value->isoFormat('M'));
}])->orderBy('umkm_product_count')->get()->take(3);
    $top_umkm[$value->isoFormat('MMMM')] = $umkm;
}
//
$top_product = UmkmProduct::with('file')->withCount('umkm_product_viewer')-
>orderBy('umkm_product_viewer_count','desc')->get()->take(1);
$latest_letter = LetterRequest::with('user.user_detail')->latest()->get()->take(5);
$count_aspiration = Aspiration::count();
$latest_aspiration = Aspiration::with('user.user_detail')->latest()->get()->take(3);
//
$user_skill = UserSkill::orderBy('title','asc')->groupBy('title')->take(10)->get()-
>pluck('title');
$user_skill_count = UserSkill::select(DB::raw('count(*) as jumlah'))->orderBy('title','asc')-
>groupBy('title')->take(10)->get()->pluck('jumlah');

return view('admin.dashboard.index', compact(
    'n_penduduk',
    'n_user_gender',
    'n_user_active',
    'n_umkm',
    'n_product',
    'n_news',
    'n_skill',
    'top_aspirative_user',
    'top_umkm',
    'top_product',
    'latest_letter',
    'count_aspiration',
    'latest_aspiration',
    'user_skill',
    'user_skill_count'
));
}

```

Lampiran 20 Kode Fitur Manajemen Penduduk

File UserController.php

```

public function index()
{
    return view('admin.penduduk.index');
}
public function create()
{
    $data = null;
    $status_perkawinans = StatusPerkawinan::all();
    $genders = Gender::all();
    $educations = Education::all();
    $religions = Religion::all();
    $professions = Profession::all();
    $endpoint = (object)[
        'mode' => 'create',
        'method' => 'POST',
        'url' => route('admin.users.store')
    ];
    return view('admin.penduduk.create_edit', compact('data','status_perkawinans', 'genders', 'educations', 'religions', 'professions', 'endpoint'));
}
public function store(UserRequest $request)
{
    //save user
    $inputUser = $request->only(
        'email',
        'no_hp',
        'password'
    );
    // check password
    $request->validate([
        'password' => ['string','confirmed', 'min:8'],
    ]);
    if($request->hasFile('avatar')) {
        // save file
        $file = FileHelper::store($request->file('avatar'),'uploads/images/users');
        //push to array input user detail
        $inputUser['avatar'] = $file->id;
    }
    $inputUser['email_verified_at'] = now();

    $inputUserDetail = $request->user_detail;
    $inputUserDetail['tanggal_lahir'] = (new Carbon($inputUserDetail['tanggal_lahir']))->format('Y-m-d');
    if(Profession::find($inputUserDetail['profession_id']) == null){
        $last_kode = (int)Profession::orderBy('kode', 'desc')->first()->kode;
        $new_profession = Profession::create([
            'kode' => $last_kode,
            'name' => $inputUserDetail['profession_id']
        ]);
        unset($inputUserDetail['profession_id']);
        $inputUserDetail['profession_id'] = $new_profession->id;
    }
    try {
        $user = User::create($inputUser);
        //save user detail
        $user->user_detail()->create($inputUserDetail);
        $user_data = $user->load('user_detail.gender', 'user_detail.education', 'user_detail.religion', 'user_detail.profession', 'user_detail.status_perkawinan', 'file');
    }
}

```

```

        return ResponseFormatter::success($user_data, 'Berhasil Menyimpan Data');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage());
    }
}

public function edit(User $user)
{
    $data = $user->load('user_detail', 'file');
    $status_perkawinans = StatusPerkawinan::all();
    $genders = Gender::all();
    $educations = Education::all();
    $religions = Religion::all();
    $professions = Profession::all();
    $endpoint = (object)[
        'mode' => 'edit',
        'method' => 'POST',
        'url' => route('admin.users.update', $user->id)
    ];
    return view('admin.penduduk.create_edit', compact('data', 'status_perkawinans', 'genders', 'educations', 'religions', 'professions', 'endpoint'));
}

public function update(UserRequest $request, User $user)
{
    //save user
    $inputUser = $request->only(
        'email',
        'no_hp',
    );
    if ($request->has('password')) {
        $inputUser['password'] = $request->password;
    }
    if ($request->hasFile('avatar')) {
        //delete file
        if ($user->file != null) {
            FileHelper::delete($user->file->path);
        }
        // save file
        $file = FileHelper::store($request->file('avatar'), 'uploads/images/users');
        //push to array input user detail
        $inputUser['avatar'] = $file->id;
    }
    $inputUserDetail = $request->user_detail;
    $inputUserDetail['tanggal_lahir'] = (new Carbon($inputUserDetail['tanggal_lahir']))->format('Y-m-d');
    if (!Profession::find($inputUserDetail['profession_id'])) {
        $last_kode = (int)Profession::orderBy('kode', 'desc')->first()->kode;
        $new_profession_id = Profession::create([
            'kode' => $last_kode,
            'name' => $inputUserDetail['profession_id']
        ])->id;
        unset($inputUserDetail['profession_id']);
        $userDetail['profession_id'] = $new_profession_id;
    }
    try {
        $user->update($inputUser);
        //save user detail
    }
}

```

```

    $user->user_detail()->update($inputUserDetail);
    $user_data = $user->load('user_detail.gender', 'user_detail.education',
    'user_detail.religion', 'user_detail.profession', 'user_detail.status_perkawinan', 'file');
    return ResponseFormatter::success($user_data, 'Berhasil Menyimpan Data');
} catch (Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage());
}
}

public function destroy(User $user)
{
try {
    if($user->file != null){
        FileHelper::delete($user->file->path);
    }
    if($user->user_detail != null){
        $user->user_detail->delete();
    }
    $user->delete();
    return ResponseFormatter::success(null, 'Sukses menghapus data');
} catch (Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
}

public function deleteMultiple(Request $request){
$request->validate(['id' => ['required']]);
$data_id = json_decode($request->id, true);
try {
foreach($data_id as $id){
    $user = User::find($id);
    if($user->file != null){
        FileHelper::delete($user->file->path);
    }
    if($user->user_detail != null){
        $user->user_detail->delete();
    }
    $user->delete();
}
return ResponseFormatter::success(null, 'Sukses menghapus data');
} catch (Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
}

public function userData()
{
try {
    $users = User::with('user_detail.gender', 'user_detail.education', 'user_detail.religion',
    'file', 'user_detail.status_perkawinan', 'user_detail.profession')->get();
    return ResponseFormatter::success($users, 'Data Penduduk Berhasil Didapat');
} catch (Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
}

public function export(){
$file_name = 'UserExport'.Carbon::now()->format('Ymd');
return Excel::download(new UsersExport, $file_name.'.xlsx');
}
}

```

```

public function import(Request $request){
    $request->validate([
        'file' => ['file','mimes:xls,xlsx']
    ]);
    try {
        Excel::import(new UsersImport, $request->file('file'));
        return ResponseFormatter::success(null, 'Berhasil Import Data');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function import_template(){
    return Excel::download(new UserTemplateImport, 'template_users_import.xlsx');
}

```

Lampiran 21 Kode Fitur Manajemen Inventaris Desa

```

public function index()
{
    //
    return view('admin.asset.index');

}
public function store(AssetRequest $request)
{
    $input = $request->only('type','name','amount','size','status');
    try {
        $asset = Asset::create($input);
        return ResponseFormatter::success($asset,'Data Inventaris Berhasil Ditambahkan');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function update(AssetRequest $request, Asset $asset)
{
    $input = $request->only('type','name','amount','size','status');
    try {
        $asset->update($input);
        return ResponseFormatter::success($asset,'Data Inventaris Berhasil Diubah');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function destroy(Asset $asset)
{
    try {
        $asset->delete();
        return ResponseFormatter::success(null, 'Berhasil Menghapus Data Inventaris');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function assetData(){
    $data = Asset::all();
    return ResponseFormatter::success($data, 'Berhasil Mendapatkan Data Inventaris');
}

```

```
 }
```

Lampiran 22 Kode Fitur Manajemen Keuangan

```
public function index()
{
    return view('admin.budget.index');
}
public function store(BudgetRequest $request)
{
    $input = $request->only('date','income','spending','detail','status');
    try {
        $budget = Budget::create($input);
        return ResponseFormatter::success($budget, "Berhasil Menyimpan Data");
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}

public function update(BudgetRequest $request, Budget $budget)
{
    $input = $request->only('date','income','spending','detail','status');
    try {
        $budget->update($input);
        return ResponseFormatter::success($budget, 'Berhasil Mengubah Data Keuangan');

    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function destroy(Budget $budget)
{
    //
    try {
        $budget->delete();
        ResponseFormatter::success(null, "Berhasil Menghapus Data");
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}

public function budgetData(){
    $data = Budget::all();
    return ([
        'success' => true,
        'data' => $data,
        'message' => 'Sukses mendapatkan data'
    ]);
}
```

Lampiran 23 Kode Fitur Manajemen Berita

```
public function index()
{
    return view('admin.news.index');
}
```

```

public function store(NewsRequest $request)
{
    $input = $request->except('image');
    if($request->hasFile('image')){
        $path = $request->file('image')->store('uploads/images/news','public');
        $file = File::create([
            'path' => $path
        ]);
        $input['image'] = File::where('path', $path)->first()->id;
    }
    $input['admin_id'] = auth('web')->user()->id;
    try {
        $news = News::create($input);
        return ResponseFormatter::success($news, 'Berhasil Menyimpan Berita');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}

public function update(NewsRequest $request, $id)
{
    $news = News::find($id);
    $input = $request->except('image');
    if($request->hasFile('image')){
        FileHelper::delete($news->file->path);
        $file = FileHelper::store($request->file('image'), 'uploads/images/news');
        $input['image'] = $file->id;
    }
    $input['admin_id'] = auth('web')->user()->id;
    try {
        $news = $news->update($input);
        return ResponseFormatter::success($news, 'Berhasil Mengubah Berita');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function destroy(News $news)
{
    try {
        FileHelper::delete($news->file?->path);
        $news->delete();
        return ResponseFormatter::success(null, 'Berhasil Menghapus Berita');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}

public function newsData()
{
    try {
        $news = News::with('file','admin')->withCount('news_viewer')->withCount('comment')-
>get();
        return ResponseFormatter::success($news, 'Berhasil Mendapatkan Berita');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}

```

Lampiran 24 Kode Fitur Manajemen Agenda

```

public function index()
{
    return view('admin.agenda.index');
}
public function store(AgendaRequest $request)
{
    //
    $input = $request->except('image');
    if($request->hasFile('image')){
        $request->validate([
            'image' => ['required','mimetypes:image/*', 'max:5000']
        ]);
        $file = FileHelper::store($request->file('image'), 'uploads/images/agendas');
        $input['image'] = $file->id;
    }
    $input['admin_id'] = auth('web')->user()->id;
    try {
        $agenda = Agenda::create($input);
        return ResponseFormatter::success($agenda, 'Berhasil Menyimpan Agenda');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function update(AgendaRequest $request, Agenda $agenda)
{
    $input = $request->except('image');
    if($request->hasFile('image')){
        $request->validate([
            'image' => ['required','mimetypes:image/*', 'max:5000']
        ]);
        FileHelper::delete($agenda->file->path);
        $file = FileHelper::store($request->file('image'), 'uploads/images/agendas');
        $input['image'] = $file->id;
    }
    $input['admin_id'] = auth('web')->user()->id;
    try {
        $agenda = $agenda->update($input);
        return ResponseFormatter::success($agenda, 'Berhasil Mengubah Agenda');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function destroy(Agenda $agenda)
{
    //
    try {
        $agenda->delete();
        FileHelper::delete($agenda->file->path);
        return ResponseFormatter::success(null, 'Berhasil Menghapus Agenda');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function agendaData()
{
    try {

```

```
$agendas = Agenda::with('file', 'admin')->get();
return ResponseFormatter::success($agendas, 'Berhasil Mendapatkan Data Agenda');
} catch (\Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
```

Lampiran 25 Kode Fitur Manajemen Aspirasi

```

public function index()
{
    return view('admin.aspiration.index');
}

public function store(AspirationRequest $request)
{
    //
    $input = $request->except('attachment');
    if($request->hasFile('attachment')){
        $request->validate([
            'attachment' => ['max:5000', 'mimes:png,jpg,pdf,xlsx,docx,rar,zip'],
        ]);
        $file = FileHelper::store($request->file('attachment'), 'uploads/documents/aspirations');
        $input['attachment'] = $file->id;
    }
    try {
        $aspiration = Aspiration::create($input);
        return ResponseFormatter::success($aspiration, 'Berhasil Menyimpan Aspirasi');
    } catch (Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function update(AspirationRequest $request, Aspiration $aspiration)
{
    //
    $input = $request->except('attachment');
    if($request->hasFile('attachment')){
        $request->validate([
            'attachment' => ['max:5000', 'mimes:png,jpg,pdf,xlsx,docx,rar,zip'],
        ]);
        FileHelper::delete($aspiration->file?->path);
        $file = FileHelper::store($request->file('attachment'), 'uploads/documents/aspirations');
        $input['attachment'] = $file->id;
    }
    try {
        $aspiration = $aspiration->update($input);
        return ResponseFormatter::success($aspiration, 'Berhasil Mengubah Aspirasi');
    } catch (Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function destroy(Aspiration $aspiration)
{
    //
    try {
        $aspiration->delete();
        FileHelper::delete($aspiration->file->path);
    }
}

```

```

        return ResponseFormatter::success($aspiration, 'Berhasil Menghapus Aspirasi');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function aspirationData()
{
    //
    try {
        $data = (object)[
            'users' => User::with('user_detail')->get(),
            'data' => Aspiration::with('file','user.user_detail')->get()
        ];
        return ResponseFormatter::success($data, 'Berhasil Mendapatkan Data Aspirasi');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function attachmentDownload(Aspiration $aspiration){
    return Storage::download('public/'.$aspiration->file->path);
}

```

Lampiran 26 Kode Fitur Manajemen Pengajuan Surat

```

public function index()
{
    $user = User::with('user_detail')->get();
    return view('admin.letter_request.index',[ 'user' => $user ]);

}
public function store(RequestsLetterRequest $request)
{
    //
    $input = $request->except('file_id','letter_response', 'user');
    if($request->hasFile('file_id')){
        $file = FileHelper::store($request->file('file_id'), 'uploads/documents/requests');
        $input['file_id'] = $file->id;
    }
    try {
        $letter_request = LetterRequest::create($input);
        return ResponseFormatter::success($letter_request, 'Berhasil Menyimpan Data Surat
Permintaan');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function update(RequestsLetterRequest $request, LetterRequest $letter_request)
{
    //
    $input = $request->except('file_id','letter_response', 'user');
    if($request->hasFile('file_id')){
        $request->validate([
            'file_id' => ['required', 'max:5000'],
        ]);
        FileHelper::delete($letter_request->file->path);
        $file = FileHelper::store($request->file('file_id'), 'uploads/documents/requests');
        $input['file_id'] = $file->id;
    }
}

```

```

    }
    try {
        $letter_request->update($input);
        return ResponseFormatter::success($letter_request, 'Berhasil Mengubah Data Surat Permintaan');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function destroy(LetterRequest $letter_request)
{
    try {
        FileHelper::delete($letter_request->file->path);
        $letter_request->delete();
        return ResponseFormatter::success(null, 'Berhasil Menghapus Data Surat Permintaan');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function letterRequestData()
{
    try {
        $data
        = LetterRequest::with('file','user.user_detail','letter_response.admin','letter_response.letter_feedback')->get();
        return ResponseFormatter::success($data, 'Berhasil Mendapatkan Data Surat Permintaan');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}
public function fileDownload(LetterRequest $letter_request){
    return Storage::download('public/'.$letter_request->file->path);
}

```

Lampiran 27 Kode Fitur Manajemen UMKM

```

public function index()
{
    //
    $users = User::with('user_detail')->get();
    return view('admin.umkm.index', compact('users'));
}
public function store(UmkmRequest $request)
{
    $input = $request->except('image','long', 'lat', 'shopee', 'tokopedia', 'lazada', 'bukalapak');
    if ($request->hasFile('image')) {
        $request->validate([
            'image' => ['max:5000','mimetypes:image/*']
        ]);
        $file = FileHelper::store($request->file('image'), 'uploads/images/umkm');
        $input['image'] = $file->id;
    }
    $input['coordinat'] = json_encode((object)[
        'long' => $request->long,
        'lat' => $request->lat,
    ]);
}

```

```

$input['marketplace_url'] = json_encode((object)[
    'shopee' => $request->shopee,
    'tokopedia' => $request->tokopedia,
    'lazada' => $request->lazada,
    'bukalapak' => $request->bukalapak,
]);
try {
    $umkm = Umkm::create($input);
    return ResponseFormatter::success($umkm, 'Berhasil Menyimpan Data Umkm');
} catch (\Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
}

public function update(UmkmRequest $request, Umkm $umkm)
{
    $input = $request->except('image','long', 'lat', 'shopee', 'tokopedia', 'lazada', 'bukalapak');
    if ($request->hasFile('image')) {
        $request->validate([
            'image' => ['max:5000','mimetypes:image/*']
        ]);
        FileHelper::delete($umkm->file->path);
        $file = FileHelper::store($request->file('image'), 'uploads/images/umkm');
        $input['image'] = $file->id;
    }
    $input['coordinat'] = json_encode((object)[
        'long' => $request->long,
        'lat' => $request->lat,
    ]);
    $input['marketplace_url'] = json_encode((object)[
        'shopee' => $request->shopee,
        'tokopedia' => $request->tokopedia,
        'lazada' => $request->lazada,
        'bukalapak' => $request->bukalapak,
    ]);
    try {
        $umkm_update = $umkm->update($input);
        $user = User::find($input['user_id']);
        $content = (object)[
            'type' => 'umkm_notification',
            'title' => 'Perubahan Status UMKM',
            'image' => null,
            'body' => "Status UMKM ". $input['name']. ' Anda Diubah Menjadi ' .
strtoupper($input['status']),
        ];
        $data = (object)[
            'token' => $user->fcm_token,
            'type' => $content->type,
            'title' => $content->title,
            'body' => $content->body,
            'image' => $content->image,
            'status' => $input['status'],
            'detail' => 'true',
            'content' => $umkm,
        ];
        FcmHelper::push($user, $data);
        return ResponseFormatter::success($umkm, 'Berhasil Mengubah Data Umkm');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}

```

```

        }
    }

    public function destroy(Umkm $umkm)
    {
        try {
            FileHelper::delete($umkm->file->path);
            $umkm->delete();
            return ResponseFormatter::success(null, 'Berhasil Menghapus Data Umkm');
        } catch (\Throwable $th) {
            return ResponseFormatter::error(null, $th->getMessage(), 400);
        }
    }

    public function umkmData()
    {
        try {
            $umkms = Umkm::with('file', 'user.user_detail')->withCount('umkm_product')->get();
            return ResponseFormatter::success($umkms, 'Berhasil Mendapatkan Data Umkm');
        } catch (\Throwable $th) {
            return ResponseFormatter::error(null, $th->getMessage(), 400);
        }
    }
}

```

Lampiran 28 Kode Fitur Manajemen Produk UMKM

```

public function index()
{
    //
    $umkms = Umkm::all();
    return view('admin.umkm_product.index', compact('umkms'));
}

public function store(UmkmProductRequest $request)
{
    $input = $request->validated();
    if($request->hasFile('image')){
        $request->validate([
            'image' => ['max:5000','mimetypes:image/*']
        ]);
        $file = FileHelper::store($request->file('image'), 'uploads/images/umkm_products');
        $input['image'] = $file->id;
    }
    try {
        $umkm_product = UmkmProduct::create($input);
        return ResponseFormatter::success($umkm_product, 'Berhasil Menyimpan Data Produk UMKM');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}

public function update(UmkmProductRequest $request, UmkmProduct $umkm_product)
{
    $input = $request->validated();
    if($request->hasFile('image')){
        $request->validate([
            'image' => ['max:5000','mimetypes:image/*']
        ]);
        FileHelper::delete($umkm_product->file->path);
    }
}

```

```

    $file = FileHelper::store($request->file('image'), 'uploads/images/umkm_products');
    $input['image'] = $file->id;
}
try {
    $umkm_product->update($input);
    return ResponseFormatter::success($umkm_product, 'Berhasil Mengubah Data Produk Umkm');
} catch (\Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
public function destroy(UmkmProduct $umkm_product)
{
try {
    FileHelper::delete($umkm_product->file->path);
    $umkm_product->delete();
    return ResponseFormatter::success(null, 'Berhasil Menghapus Data Produk Umkm');
} catch (\Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
}

public function umkmProductData(){
try {
    $umkm_products = UmkmProduct::with('umkm', 'file')->withCount('umkm_product_viewer')->get();
    return ResponseFormatter::success($umkm_products, 'Berhasil Mendapatkan Data Produk Umkm');
} catch (\Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
}
}

```

Lampiran 29 Kode Fitur Manajemen Keahlian

```

public function index()
{
    //
    $users = User::with('user_detail')->get();
    return view('admin.user_skill.index', compact('users'));
}
public function store(UserSkillRequest $request)
{
    //
    $input = $request->validated();
    if ($request->hasFile('attachment')) {
        $request->validate([
            'attachment' => ['max:5000', 'mimes:png,jpg,jpeg,rar,zip,doc,docx,xls,xlsx, pdf']
        ]);
        $file = FileHelper::store($request->file('attachment'), 'uploads/images/user_skill');
        $input['attachment'] = $file->id;
    }
    try {
        $userSkill = UserSkill::create($input);
        return ResponseFormatter::success($userSkill, 'Data Skill User Berhasil Disimpan');
    } catch (\Throwable $th) {
        return ResponseFormatter::error(null, $th->getMessage(), 400);
    }
}

```

```

        }
    }

    public function update(UserSkillRequest $request, UserSkill $userSkill)
    {
        $input = $request->validated();
        if ($request->hasFile('attachment')) {
            $request->validate([
                'attachment' => ['max:5000', 'mimes:png,jpg,jpeg,rar,zip,doc,docx,xls,xlsx,pdf']
            ]);
            FileHelper::delete($userSkill->file?->path);
            $file = FileHelper::store($request->file('attachment'), 'uploads/images/user_skill');
            $input['attachment'] = $file->id;
        }
        try {
            $userSkill->update($input);
            return ResponseFormatter::success($userSkill, 'Data Skill User Berhasil Diubah');
        } catch (\Throwable $th) {
            return ResponseFormatter::error(null, $th->getMessage(), 400);
        }
    }

    public function destroy(UserSkill $userSkill)
    {
        try {
            FileHelper::delete(File::find($userSkill->file?->path));
            $userSkill->delete();
            return ResponseFormatter::success(null, 'Data Skill User Berhasil Dihapus');
        } catch (\Throwable $th) {
            return ResponseFormatter::error(null, $th->getMessage(), 400);
        }
    }

    public function userSkillData(){
        try {
            $userSkills = UserSkill::with('user.user_detail','file')->get();
            return ResponseFormatter::success($userSkills, 'Data Skill User Berhasil Didapatkan');
        } catch (\Throwable $th) {
            return ResponseFormatter::error(null, $th->getMessage(), 400);
        }
    }

    public function downloadLampiran(UserSkill $user_skill){
        return Storage::download('public/'.$user_skill->file->path);
    }
}

```

Lampiran 30 Kode Fitur Profil Desa

```

public function informationProfil()
{
    $data = Information::with('file')->first();
    return view('admin.profile.index', compact('data'));
}

public function store(Request $request)
{
    $request->validate([
        'nama_desa' => ['required','string'],
        'tagline' => ['required', 'string'],
        'alamat' => ['required', 'string'],
        'lat' => ['nullable', 'string'],
    ]);
}

```

```
'long' => ['nullable', 'string']
]);
$input = $request->only('nama_desa','alamat','tagline','lat','long');
if($request->hasFile('file_id')){
    $request->validate([
        'file_id' => ['max:5000','mimetypes:image/*']
    ]);
    $file = FileHelper::store($request->file('file_id'),'uploads/images/information');
    $input['file_id'] = $file->id;
}
try {
    $information = Information::updateOrCreate(['id' => $request->id], $input);
    return ResponseFormatter::success($information, 'Sukses Menyimpan Informasi');
} catch (\Throwable $th) {
    return ResponseFormatter::error(null, $th->getMessage(), 400);
}
```