



SKRIPSI

RANCANG BANGUN SISTEM INFORMASI E-OFFICE BERBASIS DESKTOP MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT PADA CV BELIDA

RANDY DWI SAPUTRA
NPM 20082010100

DOSEN PEMBIMBING

Agung Brastama Putra, S.Kom., M.Kom.
Rizka Hadiwiyanti, S.Kom., M.Kom.

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAWA TIMUR
FAKULTAS ILMU KOMPUTER
PROGRAM STUDI SISTEM INFORMASI
SURABAYA
2025**



SKRIPSI

**RANCANG BANGUN SISTEM INFORMASI E-
OFFICE BERBASIS DESKTOP
MENGUNAKAN METODE RAPID
APPLICATION DEVELOPMENT PADA CV
BELIDA**

Randy Dwi Saputra
NPM 20082010100

DOSEN PEMBIMBING

Agung Brastama Putra, S.Kom., M.Kom.
Rizka Hadiwiyanti, S.Kom., M.Kom.

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAWA TIMUR
FAKULTAS ILMU KOMPUTER
PROGRAM STUDI SISTEM INFORMASI
SURABAYA
2025**

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM INFORMASI E-OFFICE BERBASIS DESKTOP MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT PADA CV BELIDA

Oleh:

RANDY DWI SAPUTRA

NPM. 20082010100

Telah dipertahankan dihadapan dan diterima oleh Tim Penguji Skripsi
Prodi Sistem Informasi Fakultas Ilmu Komputer Universitas
Pembangunan Nasional Veteran Jawa Timur Pada tanggal 10 Juni
2025

Menyetujui

Agung Brastama Putra, S.Kom., M.Kom.

NIP. 19851124 2021211 003

(Pembimbing I)

Rizka Hadiwiyanti, S.Kom., M.Kom.

NIP. 19860727 2018032 001

(Pembimbing II)

Eka Dyar Wahyuni, S.Kom., M.Kom.

NIP. 19841201 2021212 005

(Ketua Penguji)

Reisa Permatasari, S.T., M.Kom.

NIP. 19920514 2022032 007

(Anggota Penguji II)

Tri Luhur Indayanti Sugata, M.IM.

NIP. 19920616 2024062 001

(Anggota Penguji III)

Mengetahui,

Dekan Fakultas Ilmu Komputer

Prof. Dr. Ir. Novirina Hendrasarie, MT.

NIP. 19681126 199403 2 001

Halaman ini sengaja dikosongkan

LEMBAR PERSETUJUAN

**RANCANG BANGUN SISTEM INFORMASI E-OFFICE
BERBASIS DESKTOP MENGGUNAKAN METODE RAPID
APPLICATION DEVELOPMENT PADA CV BELIDA**

Oleh:

RANDY DWI SAPUTRA

NPM. 20082010100

Menyetujui,

**Koordinator Program Studi Sistem Informasi
Fakultas Ilmu Komputer**



Agung Brastama Putra, S.Kom, M.Kom.
NIP. 19851124 2021211 003

Halaman ini sengaja dikosongkan

SURAT PERNYATAAN BEBAS PLAGIASI

Yang bertandatangan di bawah ini:

Nama Mahasiswa : Randy Dwi Saputra
Program Studi : Sistem Informasi
NPM : 20082010100
Program : Sarjana (S1)
Program Studi : Sistem Informasi
Fakultas : Ilmu Komputer

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu Lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/Lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan secara lengkap dalam daftar pustaka.

Dan saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi. Apabila dikemudian hari ditemukan indikasi plagiat pada Skripsi ini, saya bersedia menerima sanksi sesuai dengan peraturan perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya tanpa ada paksaan dari siapapun juga dan untuk dipergunakan sebagaimana mestinya.

Surabaya, Juni 2025
Yang Membuat Pernyataan,



Randy Dwi Saputra
NPM. 20082010100

Halaman ini sengaja dikosongkan

ABSTRAK

Nama Mahasiswa / NPM : Randy Dwi Saputra / 20082010100
Judul Skripsi : Rancang Bangun Sistem Informasi E-Office
Berbasis Desktop Menggunakan Metode Rapid
Application Development pada CV Belida
Dosen Pembimbing : 1. Agung Brastama Putra, S.Kom., M.Kom.
2. Rizka Hadiwiyanti, S.Kom., M.Kom.

Proses pengelolaan surat dan agenda pada CV Belida masih dilakukan secara manual menggunakan media fisik, sehingga rentan terhadap risiko kehilangan, kerusakan dokumen, serta keterlambatan dalam penyampaian informasi. Permasalahan ini menghambat efisiensi kerja dan akurasi pencatatan arsip surat maupun agenda internal. Untuk mengatasi hal tersebut, skripsi ini membangun Sistem Informasi *E-Office* berbasis *desktop* yang bertujuan mempermudah pengelolaan dokumen dan pencatatan agenda secara terstruktur dan terkomputerisasi. Pengembangan sistem menggunakan metode *Rapid Application Development* (RAD) karena mampu mempercepat proses pembuatan aplikasi dengan pendekatan iteratif dan *prototyping* fungsional diuji dan disempurnakan secara berulang berdasarkan umpan balik pengguna untuk memastikan kesesuaian fitur. Pengujian sistem dilakukan menggunakan metode *Black Box Testing* untuk memastikan fungsi-fungsi berjalan sesuai kebutuhan pengguna. Hasil dari skripsi ini adalah aplikasi *e-office desktop* yang memiliki lima fitur utama, yaitu pengelolaan surat masuk, surat keluar, disposisi surat, agenda, dan memo yang sesuai dengan kebutuhan administrasi di CV Belida.

Kata kunci : Sistem Informasi, *E-Office*, *Rapid Application Development*, CV Belida

Halaman ini sengaja dikosongkan

ABSTRACT

Nama Mahasiswa / NPM : Randy Dwi Saputra / 20082010100
Judul Skripsi : Design and Development of a Desktop-Based E-Office Information System Using the Rapid Application Development Method at CV Belida
Dosen Pembimbing : 1. Agung Brastama Putra, S.Kom., M.Kom.
2. Rizka Hadiwiyanti, S.Kom., M.Kom.

ABSTRACT

The document and agenda management process at CV Belida is still conducted manually using physical media, making it vulnerable to risks such as loss, document damage, and delays in information delivery. These problems hinder work efficiency and the accuracy of mail archives and internal agenda records. To address these issues, this thesis develops a desktop-based E-Office Information System aimed at streamlining document and agenda management in a structured, computerized manner. The system was developed using the Rapid Application Development (RAD) method, which accelerates the application development process through an iterative approach; functional prototypes are repeatedly tested and refined based on user feedback to ensure feature suitability. System testing was performed using the Black Box Testing method to ensure that all functions run according to user requirements. The result of this research is a desktop e-office application with five main features: incoming mail management, outgoing mail management, mail disposition, agenda, and memo management, tailored to the administrative needs at CV Belida.

Keywords: Information System, E-Office, Rapid Application Development, CV Belida

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Dengan hati yang dipenuhi rasa syukur dan kerendahan hati, penulis mengucapkan segala puji dan terima kasih kepada Allah SWT, atas limpahan rahmat, kekuatan, dan ketenangan batin yang telah diberikan hingga akhirnya penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Informasi E-Office Berbasis Desktop menggunakan Metode Rapid Application Development pada CV Belida”. Skripsi ini disusun sebagai salah satu bentuk tanggung jawab akademik dalam menyelesaikan studi pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur.

Perjalanan dalam menyusun karya ini bukan sekadar menyusun kata dan data, melainkan juga perjalanan emosi, pikiran, serta pertumbuhan pribadi. Di setiap halaman yang ditulis, terdapat doa, dukungan, dan ketulusan banyak pihak yang tak ternilai. Maka dari itu, penulis dengan tulus ingin menyampaikan penghargaan dan rasa terima kasih yang mendalam kepada:

1. Orang tua dan keluarga penulis yang dengan penuh cinta dan doa tanpa henti telah menjadi sumber kekuatan utama di setiap langkah. Terima kasih atas keikhlasan, pengorbanan, dan menjadi tempat pulang terbaik dalam segala keadaan.
2. Bapak Agung Brastama Putra, S.Kom., M.Kom. selaku dosen pembimbing I, yang telah membimbing dengan sabar dan penuh dedikasi, memberikan arahan yang bermakna, serta membuka wawasan penulis dalam menyusun skripsi ini.
3. Ibu Rizka Hadiwiyanti, S.Kom., M.Kom. selaku dosen pembimbing II, yang telah memberikan banyak masukan, dorongan, serta koreksi yang membangun selama proses penulisan.
4. Ibu Eka Dyar Wahyuni, S.Kom., M.Kom. selaku dosen wali yang selalu memberi motivasi.
5. Seluruh dosen Program Studi Sistem Informasi, yang dengan tulus berbagi ilmu dan pengalaman hidup, telah membentuk karakter dan pengetahuan penulis selama menempuh studi.

6. Teman-teman seperkuliah, terima kasih atas tawa yang tulus, candaan yang menyegarkan, serta semangat yang selalu terasa nyata dalam perjalanan ini.
7. Teman-teman seperbimbingan, atas kebersamaan, diskusi dan pertukaran ide yang membantu proses berpikir penulis.
8. Rekan-rekan seperjuangan Sistem Informasi angkatan 2020, yang telah menemani perjalanan akademik ini dengan rasa saling dukung dan kolaborasi yang tak tergantikan.
9. Dan kepada semua pihak yang tidak dapat disebutkan satu per satu, yang telah hadir dalam bentuk doa, bantuan, dukungan, maupun senyuman kecil yang memberi kekuatan luar biasa di balik layar.

Penulis menyadari bahwa karya ini jauh dari kata sempurna. Oleh sebab itu, saran dan kritik yang membangun sangat diharapkan, demi peningkatan kualitas karya tulis ini di masa mendatang. Semoga skripsi ini dapat memberikan manfaat, baik secara akademis maupun sebagai bahan refleksi atas pentingnya kepercayaan, kemudahan, dan kualitas dalam pengembangan aplikasi layanan publik digital di Indonesia.

Surabaya, Juni 2025

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	iii
LEMBAR PERSETUJUAN	v
SURAT PERNYATAAN BEBAS PLAGIASI.....	vii
ABSTRAK	ix
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Skripsi	3
1.5 Manfaat Skripsi	3
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 Landasan Teori.....	8
2.2.1 CV Belida.....	8
2.2.2 <i>E-Office</i>	9
2.2.3 Sistem Informasi	10
2.2.4 Pemrograman <i>Desktop</i>	10
2.2.5 Basis data	11
2.2.6 MySQL.....	11
2.2.7 <i>Conceptual Data Model</i>	11
2.2.8 <i>Physical Data Model</i>	12
2.2.9 <i>Entity Relationship Diagram</i>	13
2.2.10 <i>Context Diagram</i>	14
2.2.11 <i>Data Flow Diagram</i>	15
2.2.12 <i>Black Box Testing</i>	16

2.2.13 Metode <i>Rapid Application Development</i>	16
BAB III METODOLOGI PENELITIAN.....	19
3.1 Metode Penelitian.....	19
3.1.1 Identifikasi Masalah	19
3.1.2 Studi Literatur.....	19
3.1.3 Pembuatan Sistem	20
BAB IV HASIL DAN PEMBAHASAN	25
4.1 Pembuatan Sistem	25
4.1.1 <i>Requirements Planning</i>	25
4.1.2 <i>User Design</i>	41
4.1.2.2 <i>Testing dan Refine</i>	55
4.1.3 <i>Construction</i>	59
4.1.4 <i>Cutover</i>	103
BAB V PENUTUP	118
5.1. Kesimpulan.....	119
5.2. Saran	120
DAFTAR PUSTAKA	121
LAMPIRAN	125

DAFTAR GAMBAR

Gambar 2.1 CV Belida	8
Gambar 2.2 Struktur CV Belida	9
Gambar 2.3 Metode <i>Rapid Application Development</i>	16
Gambar 3.1 Alur Skripsi	19
Gambar 3.2 Alur Metode <i>Rapid Application Development</i>	20
Gambar 4.1 <i>DocsFlow</i> CV Belida	26
Gambar 4.2 <i>DocsFlow</i> CV Belida	27
Gambar 4.3 <i>Sysflow</i> Proses Bisnis Sistem Surat.....	33
Gambar 4.4 <i>Sysflow</i> Proses Bisnis Sistem Surat.....	34
Gambar 4.5 <i>Sysflow</i> Alur Agenda dan Memo.....	35
Gambar 4.6 <i>Context Diagram E-Office</i>	36
Gambar 4.7 <i>Data Flow Diagram Level 1</i>	37
Gambar 4.8 <i>Conceptual Data Model E-Office</i>	38
Gambar 4.9 <i>Physical Data Model E-Office</i>	39
Gambar 4.10 <i>Entity Relationship Diagram E-Office</i>	40
Gambar 4.11 <i>Design</i> Tampilan Login.....	42
Gambar 4.12 <i>Design</i> Tampilan Halaman Surat Masuk	42
Gambar 4.13 <i>Design</i> Tampilan Halaman Tambah Surat Masuk	43
Gambar 4.14 <i>Design</i> Tampilan Halaman Tambah Disposisi.....	43
Gambar 4.15 <i>Design</i> Tampilan Surat Keluar	44
Gambar 4.16 <i>Design</i> Tampilan Halaman Tambah Surat Keluar	44
Gambar 4.17 <i>Design</i> Tampilan Disposisi Surat.....	45
Gambar 4.18 <i>Design</i> Tampilan Halaman Tambah Update Disposisi	45
Gambar 4.19 <i>Design</i> Tampilan Agenda.....	46
Gambar 4.20 <i>Design</i> Tampilan Halaman Tambah Agenda	46
Gambar 4.21 <i>Design</i> Tampilan Halaman Tambah Anggota Agenda	47
Gambar 4.22 <i>Design</i> Tampilan Memo	47
Gambar 4.23 <i>Design</i> Tampilan Halaman Tambah Memo	48
Gambar 4.24 <i>Design</i> Tampilan Data Karyawan	48
Gambar 4.25 <i>Design</i> Tampilan Halaman Tambah Data Karyawan.....	49

Gambar 4.26 Design Tampilan Halaman Log	49
Gambar 4.27 Design Tampilan Login	50
Gambar 4.28 Design Tampilan Disposisi Surat	51
Gambar 4.29 Design Tampilan Halaman Tambah Update Disposisi.....	51
Gambar 4.30 Design Tampilan Agenda	52
Gambar 4.31 Design Tampilan Halaman Tambah Agenda.....	52
Gambar 4.32 Design Tampilan Halaman Tambah Anggota Agenda.....	53
Gambar 4.33 Design Tampilan Memo	53
Gambar 4.34 Design Tampilan Halaman Tambah Memo.....	54
Gambar 4.35 Design Tampilan Data Karyawan.....	54
Gambar 4.36 Design Tampilan Halaman Tambah Data Karyawan	55
Gambar 4.37 Design Tampilan Tambah Surat Masuk Input Manual.....	56
Gambar 4.38 Design Tampilan Tambah Surat Keluar Input Manual.....	56
Gambar 4.39 Design Tampilan Tambah Surat Masuk Input Otomatis	57
Gambar 4.40 Design Tampilan Tambah Surat Keluar Input Otomatis	58
Gambar 4.41 Database Construction	59
Gambar 4.42 Potongan Kode Sistem Login	61
Gambar 4.43 Halaman Login	61
Gambar 4.44 Potongan Kode Sistem Data Karyawan.....	63
Gambar 4.45 Halaman Data Karyawan	64
Gambar 4.46 Potongan Kode Sistem Tambah Data Karyawan.....	65
Gambar 4.47 Halaman Tambah Karyawan	65
Gambar 4.48 Potongan Kode Surat Masuk	66
Gambar 4.49 Halaman Surat Masuk	66
Gambar 4.50 Potongan Kode Halaman Tambah Surat Masuk.....	68
Gambar 4.51 Halaman Tambah Surat Masuk	68
Gambar 4.52 Potongan Kode Sistem Tambah Disposisi.....	69
Gambar 4.53 Halaman Tambah Disposisi	69
Gambar 4.54 Potongan Kode Kirim Notifikasi Whatsapp	70
Gambar 4.55 Hasil Notifikasi <i>Whatsapp</i>	71
Gambar 4.56 Potongan Kode Sistem Halaman Disposisi	72
Gambar 4.57 Halaman Disposisi	73

Gambar 4.58 Potongan Kode Tindak Lanjut Disposisi	73
Gambar 4.59 Halaman Tindak Lanjut Disposisi.....	74
Gambar 4.60 Potongan Kode Sistem Surat Keluar	75
Gambar 4.61 Halaman Surat Keluar	75
Gambar 4.62 Potongan Kode Tambah Surat Keluar.....	77
Gambar 4.63 Halaman Tambah Surat Keluar	77
Gambar 4.64 Potongan Kode Sistem Halaman Agenda	79
Gambar 4.65 Halaman Agenda	79
Gambar 4.66 Potongan Kode Tambah Agenda.....	80
Gambar 4.67 Halaman Tambah Agenda.....	81
Gambar 4.68 Potongan Kode Tambah Anggota Agenda.....	82
Gambar 4.69 Halaman Tambah Agenda.....	82
Gambar 4.70 Potongan Kode Sistem Halaman Memo	83
Gambar 4.71 Halaman Memo	84
Gambar 4.72 Potongan Kode Tambah Memo.....	85
Gambar 4.73 Halaman Tambah Memo	85
Gambar 4.74 Potongan Kode Tambah Memo.....	86
Gambar 4.75 Halaman Tambah Memo	87
Gambar 4.76 Potongan Kode Sistem Login.....	88
Gambar 4.77 Halaman Login	89
Gambar 4.78 Potongan Kode Sistem Data Karyawan	91
Gambar 4.79 Halaman Tambah Karyawan.....	91
Gambar 4.80 Potongan Kode Sistem Tambah Data Karyawan	92
Gambar 4.81 Halaman Ubah Data Karyawan.....	93
Gambar 4.82 Potongan Kode Sistem Halaman Disposisi.....	95
Gambar 4.83 Halaman Disposisi.....	95
Gambar 4.84 Potongan Kode Tindak Lanjut Disposisi	96
Gambar 4.85 Halaman Tindak Lanjut Disposisi.....	96
Gambar 4.86 Potongan Kode Sistem Halaman Agenda	98
Gambar 4.87 Halaman Agenda	98
Gambar 4.88 Potongan Kode Lihat Agenda	99
Gambar 4.89 Halaman Lihat Agenda.....	99

Gambar 4.90 Potongan Kode Lihat Anggota Agenda.....	100
Gambar 4.91 Halaman Lihat Anggota Agenda	100
Gambar 4.92 Potongan Kode Sistem Halaman Memo.....	101
Gambar 4.93 Halaman Memo	102
Gambar 4.94 Potongan Kode Tambah Memo	102
Gambar 4.95 Halaman Tambah Memo	103

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	5
Tabel 2.2 Simbol pada Conceptual Data Model	12
Tabel 2.3 Simbol pada <i>Physical Data Model</i>	13
Tabel 2.4 Komponen Utama Entity Relationship Diagram (ERD).....	14
Tabel 2.5 Context Diagram	15
Tabel 4.1 Kebutuhan Fungsionalitas.....	28
Tabel 4.2 Kebutuhan Non Fungsionalitas	31
Tabel 4.3 Iterasi 1: Umpan Balik Pengguna terhadap Penomoran Manual	57
Tabel 4.4 Iterasi 2: Penyesuaian Sistem dan Hasil Validasi	58
Tabel 4.5 Pengujian Sistem Login	104
Tabel 4.6 Pengujian Sistem Data Karyawan.....	105
Tabel 4.7 Pengujian Sistem Tambah Data Karyawan.....	105
Tabel 4.8 Pengujian Sistem Surat Masuk	106
Tabel 4.9 Pengujian Sistem Tambah Surat Masuk	106
Tabel 4.10 Pengujian Sistem Surat Keluar	107
Tabel 4.11 Pengujian Sistem Tambah Surat Keluar	107
Tabel 4.12 Pengujian Sistem Disposisi.....	108
Tabel 4.13 Pengujian Sistem Tambah Disposisi.....	108
Tabel 4.14 Pengujian Sistem Tambah Tindak lanjut	109
Tabel 4.15 Pengujian Sistem Agenda	109
Tabel 4.16 Pengujian Sistem Tambah Agenda	110
Tabel 4.17 Pengujian Sistem Tambah Anggota Agenda	110
Tabel 4.18 Pengujian Sistem Memo	111
Tabel 4.19 Pengujian Sistem Tambah Memo	112
Tabel 4.20 Pengujian Sistem Login	113
Tabel 4.21 Pengujian Sistem Disposisi.....	113
Tabel 4.22 Pengujian Sistem Tambah Tindak lanjut	114
Tabel 4.23 Pengujian Sistem Agenda	114
Tabel 4.24 Pengujian Sistem Lihat Agenda.....	115
Tabel 4.25 Pengujian Sistem Lihat Anggota Agenda	115
Tabel 4.26 Pengujian Sistem Memo	116

Tabel 4.27 Pengujian Sistem Tambah Memo.....	116
--	-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring berjalannya waktu, perkembangan teknologi khususnya teknologi informasi tergolong sangat pesat. Banyak sekali bentuk perubahan yang dapat dirasakan dengan adanya kemajuan teknologi informasi. Pada era industri 4.0 seperti saat ini, banyak perusahaan yang memanfaatkan kemajuan teknologi untuk mendukung berbagai kegiatan mereka. Mulai dari produksi, penyimpanan, hingga pembayaran semuanya sudah banyak menggunakan kemudahan dari teknologi. Dengan begitu, perkembangan teknologi terutama di bidang pengembangan sistem berbasis *desktop* telah mengubah komputer dari sekadar alat komputasi menjadi *platform* untuk berbagai aplikasi dan layanan yang mempermudah aktivitas sehari-hari. Kehadiran teknologi tersebut dapat membantu berbagai pihak karena lebih praktis dan membantu memberikan kemudahan dalam melakukan kegiatan mereka. Oleh karena itu, kemajuan teknologi saat ini diakui memberikan faktor kemudahan dan kenyamanan bagi para penggunanya [1].

Sistem informasi menjadi sebuah inovasi teknologi yang bermanfaat bagi sebuah organisasi maupun perusahaan. Dalam ranah bisnis, penerapan sistem informasi dapat mendukung perusahaan untuk meningkatkan tingkat produktivitas, mengoptimalkan manajemen rantai pasokan, meningkatkan kepuasan pelanggan, dan mengenali peluang-peluang pasar. Selain itu, sistem informasi juga memiliki peran yang cukup penting dalam pengelolaan pengetahuan dan mendukung proses inovasi di dalam organisasi. Dengan adanya sistem informasi dan teknologi, pekerjaan manusia menjadi lebih sederhana dan produktif, sehingga banyak perusahaan yang memanfaatkan dan juga mengadopsi sistem informasi. Sistem informasi dalam era modern tidak hanya terbatas pada sektor bisnis. Di sektor kesehatan, pendidikan, pemerintahan, dan berbagai bidang lainnya, peran penting sistem informasi terlihat dalam peningkatan layanan, proses pengambilan keputusan, dan efisiensi operasional [2]. Dengan memperhatikan beberapa faktor, pengelolaan sistem informasi dan teknologi informasi dapat meningkatkan kinerja dan memberikan banyak keuntungan dalam pengolahan data dan juga menghasilkan informasi yang akurat, efektif, dan efisien.[3] . Walaupun sistem informasi memiliki peran penting untuk mendukung

proses bisnis dalam perusahaan, masih ada perusahaan yang masih melakukan proses bisnisnya secara manual.

CV Belida merupakan sebuah perusahaan yang bergerak pada bidang pengolahan ikan berupa fillet ikan patin. CV Belida sendiri berdiri pada tahun 2018 yang terletak pada di Desa Bono Kecamatan Boyolangu, Kabupaten Tulungagung. Berdasarkan identifikasi dan observasi terdapat permasalahan ketika melakukan semua alur proses bisnis terutama pada proses pembuatan dan penyimpanan laporan surat yang dimana masih dilakukan secara manual sehingga dapat memakan waktu lama, kemudian dalam komunikasi internal seperti pembuatan surat yang masih menggunakan dokumen fisik. Dengan begitu, sering terjadi kesalahan pada saat melakukan pendataan dan juga penataan data karena akses dan penataan data masih harus manual dengan mengolah laporan satu per satu yang tentunya membutuhkan waktu lama dalam melakukan pencarian laporan tersebut.

Untuk mengatasi hal tersebut perlu adanya pemanfaatan sistem informasi yang bertujuan untuk memudahkan karyawan pada CV Belida dalam menangani semua alur proses bisnis mengenai laporan yang nantinya dapat mempermudah karyawan atau pengguna dalam melakukan proses bisnisnya dengan sistem tidak lagi dilakukan secara manual. Diharapkan dengan adanya sistem informasi *desktop* ini, perusahaan dan karyawan dapat menjalankan proses bisnis lebih efektif dan efisien serta dapat memberikan informasi secara menyeluruh mengenai laporan data dan dapat melakukan pekerjaan dengan lebih optimal.

Berdasarkan penelitian terdahulu yang berjudul Rancang Bangun Sistem Informasi Akuntansi Penjualan Dengan Model Waterfall Berbasis Java Desktop [4], Rancang Bangun Sistem Informasi E-Office Pada Tata Usaha Fakultas Teknik Universitas Negeri Padang[5], dan Metode Waterfall Dalam Sistem Informasi Penyewaan Mobil Berbasis Desktop [6] terdapat beberapa pembaharuan fitur pada Sistem Informasi desktop ini yang nantinya dapat digunakan oleh admin, ketua divisi, serta penanggung jawab operasional dan keuangan. pembaharuan pada fitur ini nantinya pegawai akan bisa mengakses data sesuai kapasitasnya, fitur pembuatan surat masuk, fitur disposisi surat, pembuatan surat keluar, fitur pencatatan agenda, fitur pencatatan memo, fitur terintegrasi dengan *whatsapp* untuk keperluan komunikasi internal, sehingga pegawai pada CV Belida tidak perlu mengakses data secara manual

yang membutuhkan waktu lama, cukup menggunakan aplikasi ini berdasarkan kategori divisi pegawai yang nantinya dapat membantu jalannya proses bisnis ini.

Tujuan adanya Sistem Informasi *e-office* pada CV Belida nantinya, dapat mendukung komunikasi internal, dan proses bisnis laporan surat, sehingga menjadi lebih sederhana serta lebih menghemat waktu dan tenaga para karyawan CV Belida.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, dapat dirumuskan masalah yang diangkat dalam penulisan skripsi ini yaitu "Bagaimana merancang dan membangun sistem informasi berbasis *desktop* untuk pengelolaan data surat masuk, surat keluar, disposisi surat, agenda internal dan komunikasi internal yang nantinya dapat mempermudah proses bisnis serta komunikasi internal pada CV Belida?".

1.3 Batasan Masalah

Adapun batasan masalah yang sesuai dengan perumusan masalah tersebut adalah sebagai berikut:

1. Sistem informasi ini dapat digunakan oleh *admin*, penanggung jawab operasional dan keuangan, ketua produksi, ketua aset dan ketua logistik.
2. Pada tahap perancangan ini, sistem baik *client* atau *server* berjalan di lingkungan *localhost* dengan *tools development*.
3. Pengelolaan data surat masuk, surat keluar, disposisi surat, agenda internal dan komunikasi internal yang nantinya dapat mempermudah proses bisnis serta komunikasi internal pada CV Belida.

1.4 Tujuan Skripsi

Berdasarkan uraian rumusan masalah yang telah dijelaskan, maka tujuan dari penelitian ini ialah untuk merancang dan membangun sistem informasi berbasis *desktop* untuk pengelolaan data surat masuk, surat keluar, disposisi surat, agenda dan komunikasi internal yang nantinya dapat mempermudah proses bisnis serta komunikasi internal pada CV Belida.

1.5 Manfaat Skripsi

Manfaat yang dapat diambil dari skripsi ini ialah sebagai berikut:

1. Sebagai solusi bagi CV Belida dalam mempermudah pengelolaan dokumen surat, disposisi, agenda, dan memo secara terkomputerisasi, sehingga proses administrasi menjadi lebih terstruktur dan terdokumentasi dengan baik.

2. Sebagai referensi untuk peneliti lain dalam melakukan penelitian selanjutnya

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Skripsi ini mengacu pada penelitian terdahulu sebelum skripsi ini. Adapun penelitian terdahulu yang dijadikan acuan adalah penelitian yang memiliki topik yang linier dan mirip dengan skripsi ini yaitu tentang rancang bangun sistem informasi *e-office*. Adapun penelitian terdahulu yang dijadikan acuan akan dijelaskan dalam bentuk tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

Jurnal dan Peneliti	Tujuan	Teknis	Kaitan dengan Penelitian
Rancang Bangun Sistem Informasi Akuntansi Penjualan Dengan Model Waterfall Berbasis Java Desktop [7].	Penelitian ini mengembangkan sistem informasi penjualan menggunakan metode waterfall. Bertujuan meningkatkan efisiensi dan akurasi dalam proses transaksi penjualan, serta membantu pemilik usaha dalam pengambilan keputusan.	- Analisa kebutuhan menggunakan pemodelan Unified Modelling Language (UML) dengan use case, activity diagram, sequence diagram, dan Deployment diagram. - Pada penelitian ini menggunakan Metode Waterfall. - Pengujian atau testing coding program dilakukan dengan metode black box testing	Penelitian terdahulu ini menggunakan metode waterfall. Metode pengembangan ini diimplementasikan menggunakan bahasa pemrograman Java dan MySQL, serta diuji menggunakan black box testing untuk memastikan kebenaran program.

Jurnal dan Peneliti	Tujuan	Teknis	Kaitan dengan Penelitian
Rancang Bangun Sistem Informasi E-Office Pada Tata Usaha Fakultas Teknik Universitas Negeri Padang [5]	Penelitian ini bertujuan menghasilkan sebuah sistem informasi e-Office yang didalamnya terdapat beberapa pelayanan seperti pelayanan kepegawaian dan kemahasiswaan.	- Pada penelitian ini melakukan tahapan observasi dan wawancara. - Pada Penelitian ini Melakukan desain sistem khususnya database dengan membuat ERD dan relasi antar tabel. - Melakukan implementasi dengan coding dan integrasi dengan database	Pada penelitian terdahulu menggunakan pemodelan UML sebagai desain arsitektur perangkat lunak sehingga dapat dijadikan acuan pada penelitian ini dalam pemodelan UML khususnya database dengan membuat ERD.
Metode Waterfall Dalam Sistem Informasi Penyewaan Mobil Berbasis Desktop [6]	Tujuan Penelitian ini bertujuan untuk mengembangkan sistem informasi penyewaan mobil berbasis desktop menggunakan metode Waterfall. Tujuan penelitian ini adalah untuk meningkatkan kualitas layanan, mempercepat	- Crystal report digunakan untuk pembuatan laporan dalam aplikasi tersebut - Analisa perancangan SWOT untuk mengetahui dan mengevaluasi kekuatan, kelemahan, peluang dan ancaman dari spekulasi bisnis.	Pada penelitian terdahulu menggunakan tools VB dan microsoft acces sebagai sebagai pembuatan sistem informasi penyewaan mobil berbasis desktop sehingga dapat dijadikan acuan pada penelitian ini dalam implementasi dengan tools VB

Jurnal dan Peneliti	Tujuan	Teknis	Kaitan dengan Penelitian
	proses transaksi, dan mengurangi penggunaan kertas dalam proses bisnis penyewaan mobil. Metode penelitian yang digunakan meliputi, observasi, wawancara, dan studi pustaka.	- Menggunakan metode Waterfall dengan Desain Sistem Informasi yang mencakup database, spesifikasi file, use case diagram, activity diagram, dan skema database diagram.	dan penelitian ini fokus pada pencatatan proses bisnis produksi, logistik, aset, dan keuangan dapat menjadi lebih sederhana serta lebih menghemat waktu dan tenaga para karyawan pada CV Belida

Penelitian terdahulu yang berjudul Rancang Bangun Sistem Informasi Akuntansi Penjualan Dengan Model Waterfall Berbasis Java Desktop [4], Rancang Bangun Sistem Informasi E-Office Pada Tata Usaha Fakultas Teknik Universitas Negeri Padang [5], dan Metode Waterfall Dalam Sistem Informasi Penyewaan Mobil Berbasis Desktop [6] terdapat beberapa pembaharuan pada fitur ini nantinya pegawai akan bisa mengakses data sesuai kapasitasnya, fitur surat masuk, fitur disposisi surat, fitur surat keluar, fitur pencatatan agenda, fitur pencatatan memo, dan terintegrasi dengan *whatsapp* perusahaan kedalam aplikasi, jadi pegawai pada CV Belida tidak perlu mengakses data secara manual yang membutuhkan waktu lama, cukup menggunakan aplikasi ini berdasarkan kategori role pegawai yang nantinya dapat membantu jalanya proses bisnis ini.

2.2 Landasan Teori

2.2.1 CV Belida



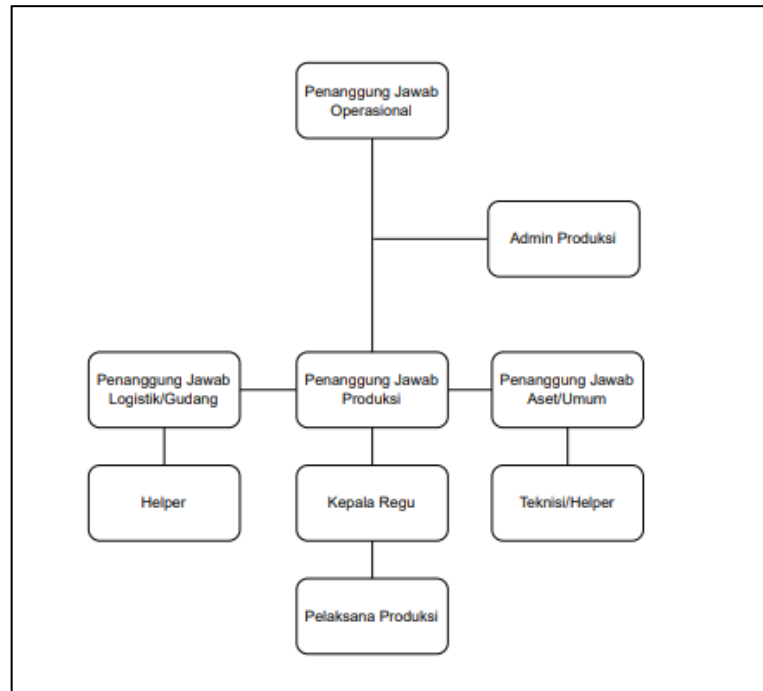
Gambar 2.1 CV Belida

CV Belida didirikan pada tahun 2018 yang bertempat di kota Tulungagung. CV Belida merupakan Perusahaan yang terletak di Desa Bono bergerak pada bidang pengolahan ikan berupa fillet ikan patin. CV Belida merupakan supplier hasil perikanan di Indonesia dengan produk utama dari CV Belida adalah fillet ikan patin. Pada saat ini CV Belida telah berdiri kurang lebih selama 6 tahun.

CV Belida sendiri saat ini memiliki beberapa produk ikan segar seperti ikan patin, lele, gurami, dan nila. CV Belida juga memiliki sebuah produk makanan beku yang sudah dikelola menjadi produk ikan tanpa kulit, fillet, dan ikan utuh. CV Belida memiliki standar produksi yaitu *freshness, nature, dan healthy*. *Freshness* sendiri mengutamakan kesegaran ikan dimana yang akan mempengaruhi selama pengolahan dan pembekuan. CV Belida menerima ikan sebagai bahan baku dalam kondisi hidup sehingga harus dijaga kesegarannya selama pengolahan. Selanjutnya pada CV Belida memiliki standar produksi *nature*, Keaslian dan kemurnian daging ikan merupakan faktor penting dalam teknologi pengolahan ikan. CV Belida mengolah ikan dengan cara dibersihkan, dipotong, dan dibekukan, tanpa menambahkan bahan dan atau pewarna *non food grade*, sehingga tekstur produk harus tetap terjaga kealamiannya. CV Belida juga memiliki standar produksi *healthy*, Dengan menggunakan bahan baku yang segar, kemudian diolah dengan cara dan pengerjaan yang baik, didukung dengan fasilitas produksi yang handal dan lingkungan yang bersih, maka akan dihasilkan produk ikan beku yang aman dan sehat.

CV Belida memiliki prinsip pendistribusian penjualan yaitu dari *Regular Order, Custom Order, Maklon/Tooling, Joint Operation*. Jadi sistem penjualan yang dilakukan CV Belida pada saat ini yaitu menggunakan cara sesuai dari permintaan

pihak konsumen yang dikomunikasikan dengan cara menunggu pesanan diterima pada satu hari sebelum hari pelaksanaan pendistribusian. Pada saat ini proses produksi dan pengantaran produk yang masih manual. Hal tersebut dilakukan karena bergantung pada permintaan pesanan yang masuk pada CV Belida. Gambar 2.2 berikut merupakan struktur dari CV Belida.



Gambar 2.2 Struktur CV Belida

2.2.2 E-Office

Aplikasi, berasal dari kata *application*, memiliki arti dasar sebagai penerapan atau penggunaan dalam konteks tertentu. Secara istilah, aplikasi merujuk pada sebuah program yang dirancang untuk melaksanakan fungsi spesifik yang memenuhi kebutuhan pengguna atau mendukung program lainnya. Program ini dikembangkan dengan struktur yang memungkinkan pengguna mengakses dan memanfaatkan fitur-fiturnya sesuai dengan tujuan yang telah ditentukan. *E-Office* adalah istilah yang diciptakan untuk menutupi meningkatnya penggunaan komputer berbasis teknologi informasi untuk pekerjaan kantor. Seiring perkembangan zaman serta dengan kebijakan penerapan reformasi birokrasi di Indonesia yang mewajibkan setiap proses bisnis/kegiatan kantor pemerintahan daerah harus berjalan dengan efisien dan efektif yang menyebabkan terjadinya transformasi proses bisnis dari sistem konvensional atau manual ke sistem otomatisasi. Proses bisnis yang dilakukan secara manual menggambarkan penyimpanan sebuah dokumen data, misalnya data surat masuk dan

keluar yang masih berupa *hardcopy*, dan diakses secara manual ketika membutuhkannya. *E-Office* mengintegrasikan berbagai alat Teknologi Informasi dan Komunikasi seperti aplikasi perangkat lunak, sistem manajemen dokumen, komunikasi *online*, kolaborasi tim, dan sistem manajemen konten untuk menciptakan lingkungan kerja yang lebih efisien dan terorganisir.[8].

2.2.3 Sistem Informasi

Sistem informasi adalah suatu sistem yang menggabungkan antara aktivitas manusia dan penggunaan teknologi informasi yang dapat mempermudah dalam melakukan kegiatan operasional. Menurut Haposan sistem informasi adalah menerjemahkan desain ke dalam kode program, menggunakan kembali kelas yang sudah tersedia di perpustakaan objek perangkat lunak yang dapat digunakan kembali, dan menambahkan yang baru selama fase desain objek. *Implementasi* juga dapat melibatkan pembuatan *database* yang berorientasi pada objek. Sistem yang dihasilkan harus diuji secara menyeluruh.[9]

Sedangkan menurut Husein dan Wibowo (2006) dalam buku “Sistem Informasi Agenda”, sistem informasi adalah seperangkat komponen yang saling berhubungan yang berfungsi untuk mengumpulkan, memproses, menyimpan dan mendistribusikan informasi untuk mendukung pembuatan keputusan dan pengawasan dalam suatu organisasi.

2.2.4 Pemrograman Desktop

Pemrograman *Desktop* adalah pemrograman yang dilakukan dengan memanipulasi elemen-elemen visual yang dilakukan pada sebuah PC Tunggal yang pengoperasiannya tidak bergantung pada PC lain dalam jaringan maupun *web*. Menurut Konixbam (2009) *Desktop Based Application* adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi internet di suatu komputer otonom dengan *operating system* atau *platform* tertentu. Aplikasi *Desktop* difokuskan kepada aplikasi yang lebih independen. Tentu Tujuannya adalah untuk mempermudah para pengguna aplikasi *desktop* dalam hal memodifikasi pengaturan aplikasi sehingga efektifitas, efisiensi waktu, dana, dan tenaga dapat lebih ditekankan semaksimal mungkin.[10].

2.2.5 Basis data

Basis data atau *database* adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari *basis data*. *Basis data* atau *database* ini juga dapat mempresentasikan pandangan yang berbeda kepada para pengguna, *programmer*, dan administratornya.[11]

Basis Data merupakan perangkat lunak yang dirancang untuk mengelola dan memanggil kueri (*query*) basis data. DBMS adalah perangkat lunak (software) yang berfungsi untuk mengelola *database*, mulai dari membuat *database* itu sendiri, sampai dengan proses-proses yang berlaku dalam database tersebut, baik berupa *entry*, *edit*, hapus *query* terhadap data, membuat laporan dan lain sebagainya secara efektif dan efisien. Salah satu jenis DBMS yang sangat terkenal saat ini adalah *relational* DBMS (RDBMS) yang merepresentasikan data dalam bentuk tabel-tabel yang saling berhubungan. Sebuah tabel disusun dalam bentuk baris (*record*) dan kolom (*field*). Banyak sekali berkembang perangkat lunak RDBMS ini, misalnya *mysql*, *oracle*, *sybase*, *dbase*, *ms.sql*, *microsoft access*. [12].

2.2.6 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. (Edy Winarno et al., 2014).

2.2.7 Conceptual Data Model

Conceptual data model (CDM) merupakan model yang merepresentasikan tabel yang merupakan entitas yang berisi atribut. Setiap entitas memiliki satu primary key yang bersifat unik (nilainya tidak sama dengan nilai yang lainnya) dan setiap entitas berhubungan dengan entitas lain yang disebut *relationships* [14].

Menurut Whitten (2004), *conceptual data model* (CDM) adalah model data yang menggunakan beberapa notasi untuk menggambarkan data dalam konteks entitas dan hubungan yang dideskripsikan oleh data tersebut.

Dalam pembuatan *conceptual data model* (CDM) umumnya memiliki dua simbol yaitu entitas dan relasi. Penjelasan kedua simbol tersebut dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol pada Conceptual Data Model

No	Simbol	Keterangan
1	Entitas/Tabel	Entitas atau tabel yang menyimpan data dalam basis data.
	Nama Tabel	
2	Relasi 1..* Nama Relasi 1..*	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i> .

Berikut langkah-langkah yang dibutuhkan dalam perancangan *Conceptual Data Model* (CDM):

1. Pahami terlebih dahulu inti permasalahan dari studi kasus yang dipakai.
2. Tentukan *entity* apa saja yang terlibat.
3. Tentukan atribut-atribut data untuk setiap entitas beserta tipe datanya.
4. Tentukan hubungan antara setiap entitas beserta kardinalitasnya.
5. Modelkan *entitiy* dan *relationship*.
6. Cek ketepatan model.
7. Perbaiki setiap *error* dan *warning*.

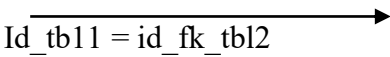
2.2.8 Physical Data Model

PDM atau Physical Data Model merupakan model yang menggunakan sejumlah tabel untuk memberikan gambaran data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom dimana setiap kolom memiliki nama yang unik (Oktafiani, 2018). PDM dan CDM memiliki kemiripan dari segi entitas,

atribut dan relationships. Perbedaannya yaitu pada CDM tidak terdapat *foreign key*, sedangkan pada PDM terdapat *foreign key* dimana satu tabel bergantung pada tabel lainnya. *foreign key* yaitu *primary key* yang berada pada tabel yang memiliki hubungan dengan tabel tersebut (Sumadya et al., 2016).

Pada pembuatan *Physical Data Model* (PDM) biasanya memiliki dua simbol yaitu tabel dan relasi. Berikut pada halaman setelah ini pada Tabel 2.3 merupakan penjelasan simbol.

Tabel 2.3 Simbol pada *Physical Data Model*

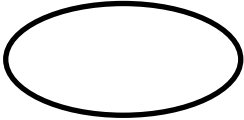
No	Simbol	Keterangan
1	Tabel	Tabel digunakan untuk menyimpan atribut, tipe data, dan tipe primary.
	Nama Tabel	
2	Relasi 	Relasi antar tabel yang terdiri dari persamaan antara <i>primary key</i> tabel yang diacu dengan kunci yang menjadi referensi acuan pada tabel lain

2.2.9 Entity Relationship Diagram

ERD adalah model konseptual yang digunakan untuk mendefinisikan struktur data dalam suatu sistem informasi dengan menunjukkan entitas, atribut, dan hubungan antar entitas dalam suatu domain tertentu.[15]

ERD merupakan bagian dari *Entity-Relationship Model* (ERM) yang bertujuan untuk mempermudah analisis dan perancangan basis data sebelum diimplementasikan ke dalam sistem manajemen basis data (DBMS). Model ini digunakan untuk merepresentasikan informasi yang ada dalam suatu sistem secara lebih jelas melalui diagram grafis. Berikut merupakan simbol-simbol dari komponen utama *ERD*.

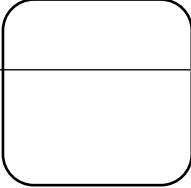
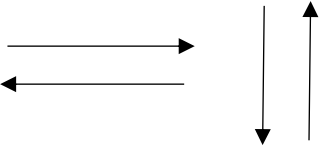
Tabel 2.4 Komponen Utama Entity Relationship Diagram (ERD)

Komponen	Simbol dalam ERD	Penjelasan
Entitas (<i>Entity</i>)		Objek atau konsep yang memiliki data dalam sistem. Bisa berupa entitas kuat atau entitas lemah.
Atribut (<i>Attribute</i>)		Karakteristik atau properti yang dimiliki oleh suatu entitas. Bisa berupa atribut sederhana, komposit, <i>multivalued</i> , atau derivatif.
Hubungan (<i>Relationship</i>)		Menunjukkan bagaimana entitas-entitas berinteraksi satu sama lain dalam sistem.
Kardinalitas (<i>Cardinality</i>)	(1:1), (1:M), (M:N)	Menunjukkan jumlah maksimum entitas yang dapat berhubungan dengan entitas lainnya.
<i>Foreign Key</i> (FK)	(Hubungan antar entitas)	Atribut yang menjadi referensi ke <i>primary key</i> tabel lain, digunakan untuk membentuk relasi.
<i>Primary Key</i> (PK)	(Garis bawah pada atribut utama)	Atribut unik yang membedakan setiap entitas dalam tabel.

2.2.10 Context Diagram

Context Diagram adalah gambaran umum tentang suatu sistem yang memperlihatkan batasan (*boundary*) sistem, adanya interaksi antara eksternal *entity* dengan suatu sistem dan informasi secara umum yang mengalir diantaranya *entity* dan sistem. Simbol-simbol yang digunakan didalam *Context Diagram* hampir sama dengan simbol-simbol DFD, hanya saja pada *Context Diagram* tidak terdapat simbol file.[16]. Berikut merupakan simbol-simbol dari *Context Diagram* :

Tabel 2. 5 Context Diagram

No	Gambar	Keterangan
1		Entity Eksternal Merupakan sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada diluar lingkungan luarnya yang akan memberikan input atau menerima output sistem.
2		Proses kegiatan yang dilakukan oleh mesin atau komputer dari suatu arus daya yang masuk ke dalam proses untuk dihasilkan arus data yang masuk ke dalam proses daya yang akan keluar dari proses.
3		Aliran data/Arus data mengalir diantara proses, simpanan data dan entity. Arus data ini meunjukkan arus data dari yang masuk ke dalam proses sistem.

2.2.11 Data Flow Diagram

Data Flow Diagram (DFD) diagram alir data adalah alat yang meng-gambarkan aliran data melalui sistem dan atau pengolahan yang dilakukan oleh sistem tersebut. DFD dapat melihatkan proses/kejadian (*event*) yang dilakukan oleh suatu sistem dari organisasi terhadap data yang masuk ke dalam sistem ataupun terhadap data yang keluar dari sistem, dan pada akhirnya dapat dilihat data tersebut disimpan. *Data flow diagram* dimulai dari diagram konteks yaitu diagram yang berisi gambaran umum dari sistem. DFD dipopulerkan Oleh DeMacro & Yordan (1979) dan Gane & Sarson (1979) dengan menggunakan pendekatan Metode Analisis Sistem Terstruktur. DFD ini merupakan model proses. Model proses merupakan teknik untuk mengorganisasikan dan mendokumentasikan struktur dan alir data di dalam sistem.

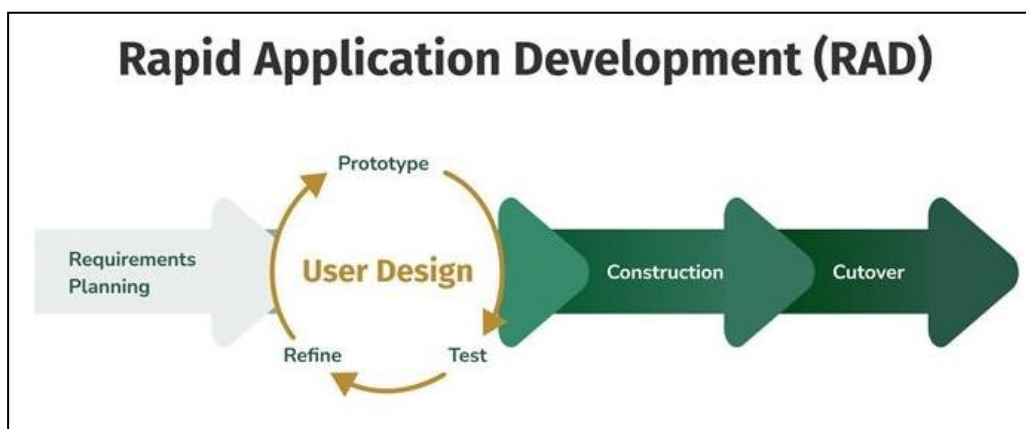
2.2.12 Black Box Testing

Metode *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pada *black box testing* dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Black-box Testing* ini lebih menguji ke tampilan luar (*interface*) dari suatu aplikasi agar mudah digunakan oleh *customer*. Pengujian ini tidak melihat dan menguji *source code program*. *Black-box testing* bekerja dengan mengabaikan struktur *control* sehingga perhatiannya hanya terfokus pada informasi *domain*. [17].

2.2.13 Metode Rapid Application Development

Metode *Rapid Application Development (RAD)* merupakan metode pengembangan perangkat lunak yang menekankan kecepatan dalam pembuatan sistem melalui penggunaan prototipe, iterasi cepat, keterlibatan langsung pengguna, dan alat bantu pengembangan yang canggih. Model ini melakukan menggunakan pendekatan iteratif dan inkremental, yang berarti pengembangan dilakukan dalam siklus berulang dengan perbaikan bertahap. *Rapid Application Development (RAD)* disebut demikian karena metodologi ini dirancang untuk mengembangkan aplikasi dengan lebih cepat dibandingkan metode tradisional seperti *Waterfall*. *RAD* mempercepat pengembangan aplikasi dengan iterasi cepat, tanpa terjebak dalam dokumentasi yang panjang seperti *Waterfall*. [18]

Menurut referensi James Martin fase – fase dalam *RAD Model* :



Gambar 2. 3 Metode *Rapid Application Development*

1) *Requirements Planning*

Tahapan awal ini merupakan tahap di mana pengguna, pengembang, dan stakeholder berdiskusi untuk menentukan kebutuhan sistem.

2) *User Design*

Tahapan kedua ini merupakan proses lanjutan dari tahapan *Requirements Planning* yang akan mulai melakukan pembuatan desain awal dan prototipe sistem. Melakukan *testing dan refine* dari *feedback* yang didapatkan oleh penanggungjawab sehingga menghasilkan *wireframe* yang nantinya sebagai acuan pada *construction* agar menu, fitur, inputan tertata secara jelas yang nantinya akan di implementasikan pada Sistem Informasi *E-Office* Berbasis *Desktop*.

3) *Construction*

Tahap *Construction* ini merupakan tahapan proses pengkodean yang dimana dimulai berdasarkan desain dan prototipe yang telah dibuat dan diperbaiki pada fase *user design*.

4) *Cutover*

Cutover merupakan proses yang dilakukan ketika sistem siap digunakan oleh pengguna setelah semua fitur utama selesai. Proses ini melibatkan pengujian akhir, pelatihan pengguna, serta implementasi ke lingkungan produksi.

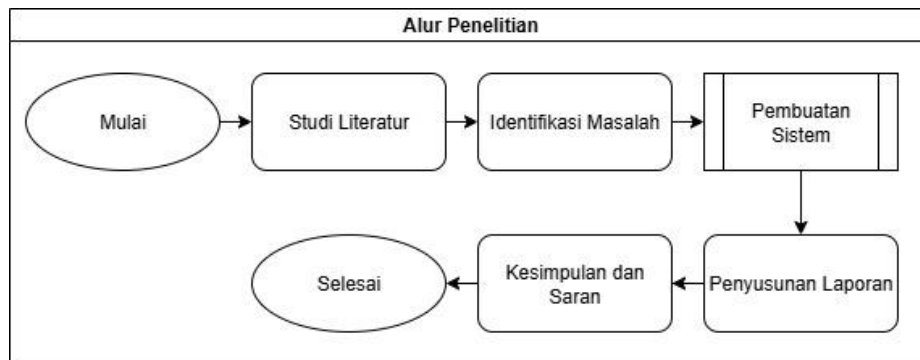
Halaman ini sengaja dikosongkan

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Bagian ini berisi metodologi yang dilakukan dalam menyusun skripsi ini. Pada alur skripsi ini akan menjelaskan tahapan skripsi mulai dari tahap awal hingga tahap akhir. Alur skripsi dapat dilihat pada gambar 3.1.



Gambar 3.1 Alur Skripsi

3.1.1 Identifikasi Masalah

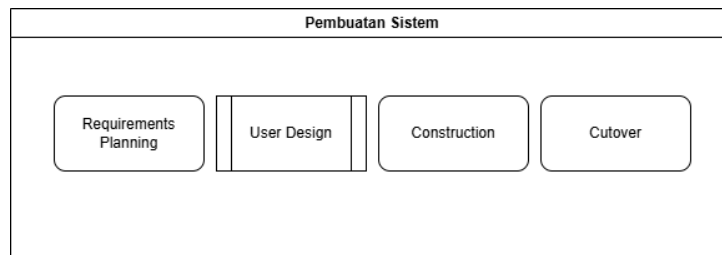
Tahapan identifikasi masalah dalam skripsi ini dimulai dengan pemahaman terhadap permasalahan pada pengelolaan data, pemborosan kertas, waktu dalam pencarian dokumen dan komunikasi internal di CV Belida. Pemahaman ini bertujuan untuk merinci permasalahan secara komprehensif dan menjadi dasar bagi pengembangan solusi pada CV Belida agar sistem terkomputerisasi.

3.1.2 Studi Literatur

Tahap studi literatur yang menjadi bagian integral dari pengembangan sistem ini, melibatkan penelusuran dan analisis informasi sebelumnya, seperti artikel, jurnal, riset terkait dan sumber informasi relevan lainnya. Teknologi yang telah ada, solusi-solusi yang mungkin telah diimplementasikan oleh peneliti dalam topik *e-office* yang dipahami melalui studi literatur ini. Studi pustaka yang mencakup pengumpulan dasar teori dari berbagai sumber artikel, jurnal, dan buku, serta pembelajaran dari penelitian sebelumnya dilakukan untuk mendapatkan wawasan sebagai dasar dan panduan dalam skripsi ini.

3.1.3 Pembuatan Sistem

Pada Tahapan ini, akan dibahas langkah-langkah yang akan diambil dalam pengembangan sistem informasi *e-office* dengan menggunakan metode *Rapid Application Development (RAD)*. Metode ini merupakan suatu teknik pengembangan sistem yang menunjukkan tahapan dari fase ke fase secara berurutan yang menekankan kecepatan, fleksibilitas, dan pengembangan berbasis prototipe.



Gambar 3. 2 Alur Metode Rapid Application Development

3.1.3.1 Requirements Planning

Tahap pertama pada skripsi ini yaitu tahap *Requirements Planning* yang dimana tahapan ini memastikan bahwa sistem yang sedang dikembangkan sesuai dengan harapan dan kebutuhan pengguna. Hal ini sangat penting dalam RAD, di mana keterlibatan pengguna sangat tinggi, dan sistem dikembangkan secara berulang[19]. Oleh karena itu untuk mendapatkan fakta tersebut perlu dilakukan studi literatur dan wawancara dengan beberapa pihak yang terkait pada bidang sistem informasi sesuai studi kasus di CV Belida. Pihak yang bersangkutan dalam skripsi ini adalah Penanggung Jawab Operasional Pada lingkup CV Belida.

3.1.3.1.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan tujuan untuk mendapatkan informasi mengenai bagaimana sistem akan dibuat. Untuk membuat sebuah sistem maka perlu diketahui terlebih dahulu bagaimana alur proses bisnis yang selama ini berjalan pada CV Belida. Metode pengumpulan informasi mengenai kebutuhan perusahaan dilakukan dengan wawancara untuk mendapatkan informasi mengenai kebutuhan sistem informasi *e-office*. Wawancara dilakukan dengan penanggung jawab operasional CV Belida.

3.1.3.1.2 Context Diagram

Context Diagram berfungsi sebagai gambaran visual mengenai interaksi antara berbagai komponen dan *stakeholders* yang terlibat dalam proses pembuatan perangkat lunak. Diagram ini menampilkan hubungan serta aliran informasi, yang sangat krusial untuk memahami lingkungan dan kebutuhan sistem[20]. Diagram ini sangat krusial untuk memastikan bahwa setiap aspek pada *software* dipertimbangkan selama proses pengembangan.

3.1.3.1.3 Data Flow Diagram Level 1

Data Flow Diagram level 1 merupakan instrumen penting dalam pengembangan *software*, yang menawarkan perspektif luas tentang proses sistem dan pergerakan data. Diagram ini menunjukkan aliran data di antara proses, repository data, dan entitas eksternal, sehingga memungkinkan *stakeholders* kepentingan untuk memahami operasi sistem tanpa harus masuk ke dalam implementasi secara spesifik[21]. Diagram ini sangat penting untuk mengenali batas-batas sistem dan memastikan bahwa semua elemen terhubung secara runtut.

3.1.3.1.4 Conceptual Data Model E-Office (CDM)

Conceptual Data Model E-Office menggambarkan tentang data yang diperlukan secara menyeluruh dalam pembuatan Sistem Informasi sebelum melakukan tahap rancangan teknis.

3.1.3.1.5 Physical Data Model E-Office (PDM)

Physical Data Model E-Office bertujuan untuk menggambarkan desain basis data secara teknis dari Sistem Informasi *E-Office*. Ilustrasi tersebut berupa desain tabel basis data beserta *foreign key* (fk) dari setiap tabel yang terhubung antara entitas.

3.1.3.1.6 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) sangat penting dalam pembuatan *software* karena menyediakan kerangka kerja konseptual tingkat tinggi untuk desain database. ERD secara visual merepresentasikan entitas, atribut, dan hubungan di antara mereka, sehingga memudahkan pemahaman yang jelas tentang kebutuhan data. Proses pembuatan ERD melibatkan beberapa langkah, termasuk ekstraksi kebutuhan, identifikasi entitas, dan konstruksi diagram [22].

3.1.3.2 *User Design (Iteratif)*

Setelah dilakukan tahapan *Requirements planning* dan didapatkan hasil wawancara untuk kebutuhan sistem, selanjutnya adalah tahap *User Design*. Tahap ini menekankan pendekatan yang berpusat pada pengguna melalui iteratif dari pembuatan *prototype*, *test*, dan *refine*. Metodologi ini memungkinkan penyesuaian cepat berdasarkan *feedback* pengguna, memastikan bahwa produk akhir selaras dengan kebutuhan dan harapan pengguna[23]. *User Design* berfokus pada perancangan dalam pembuatan *software* Sistem Informasi *E-Office* Berbasis *Desktop* pada alur proses bisnis termasuk rencana yang nantinya diaplikasikan ke dalam sistem.

3.1.3.2.1 *Design*

Tahap *design* ini merupakan tahapan desain *wireframe*. Desain *wireframe* adalah aspek fundamental dalam pengembangan perangkat lunak, yang berfungsi sebagai *blue-print* visual yang menguraikan struktur dan fungsionalitas *User Interface* (UI). Proses ini sangat penting selama fase desain, karena membantu menerjemahkan kebutuhan pengguna ke dalam desain UI yang efektif, sehingga meningkatkan *User Experience* (UX) dan memfasilitasi komunikasi di antara para *stakeholders*[24]. Desain *wireframe* ini nantinya akan di implementasikan pada Sistem Informasi *E-Office* Berbasis *Desktop*.

3.1.3.3 *Construction*

Tahap ketiga yang bisa disebut tahap *construction* merupakan tahap inti dari pembuatan sistem. Tahap *construction* berfokus pada pengembangan aktual, di mana sistem dibangun menggunakan umpan balik dari siklus berulang di tahap *user design* [25]. Proses ini akan menjelaskan bagaimana sistem dibangun mulai dari pemilihan *database* (basis data) yang akan digunakan, bahasa pemrograman yang akan dipakai sampai pada tahap *coding* (pengkodean) dan sampai juga pada tahap pengujian sistem seiring tahap *development*.

Skripsi sistem informasi ini dibangun berbasis *desktop* dengan menggunakan pemrograman terstruktur sebagai bahasa pemrograman C# *.Net Framework* untuk membangun aplikasi yang mudah dikembangkan dan mudah juga untuk menemukan bagian yang salah ketika nantinya pada pengujian sistem. Kemudian basis data yang digunakan pada bagian *back-end* Sistem Informasi *E-Office* ini adalah basis data MySQL.

3.1.3.4 *Cutover*

Tahapan *cutover* dalam *Rapid Application Development* (RAD) sangat penting untuk transisi dari pengembangan ke penerapan. Tahap ini, sebagaimana diuraikan oleh (Martin, 1991 disitasimu), merupakan langkah terakhir dalam proses RAD, setelah perencanaan kebutuhan, desain pengguna, dan konstruksi. Tahap *cutover* memastikan bahwa aplikasi yang dikembangkan diintegrasikan secara efektif ke dalam lingkungan operasional. Proses ini akan melibatkan pengujian akhir, pelatihan pengguna, dan implementasi.

Pada sistem ini diperlukan metode pengujian untuk dilakukan *testing*. Ada dua pendekatan metode dalam tahapan *testing* yaitu metode *black box testing* dan *white box testing*. Pada skripsi ini metode yang digunakan adalah metode *black box testing* dikarenakan pengujian yang dilakukan berfokus pada detail aplikasinya, seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi dan kesesuaian alur fungsi maupun proses bisnis.

Halaman ini sengaja dikosongkan

BAB IV

HASIL DAN PEMBAHASAN

Bab IV memuat informasi mengenai *output* dari setiap fase dalam metodologi skripsi, serta pembahasan mengenai kesuksesan pengembangan sistem dan evaluasi atas proses yang dilakukan selama skripsi ini dibuat. Beberapa dari evaluasi tersebut mencakup analisis, desain sistem, desain basis data, proses pembuatan program, dan pengujian program.

4.1 Pembuatan Sistem

Tahap *planning* yang dimana tahapan ini membutuhkan fakta-fakta yang mendukung proses pengerjaan skripsi. Oleh karena itu, untuk mendapatkan fakta tersebut perlu dilakukan studi literatur dan wawancara dengan beberapa pihak yang terkait pada bidang sistem informasi sesuai studi kasus di CV Belida. Pihak yang bersangkutan dalam skripsi ini adalah Petugas Pada lingkup CV Belida. Wawancara dilakukan bersama Ir. Seno Yunarwanto (Penanggung Jawab Operasional dan Keuangan). Dari hasil wawancara dengan narasumber didapatkan informasi untuk sistem pengarsipan pada CV Belida ini masih manual menggunakan *map ordner* jadi setiap dokumen dimasukkan ke dalam map sesuai dengan jenis surat masing-masing, lalu ditaruh pada lemari dokumen kantor.

4.1.1 Requirements Planning

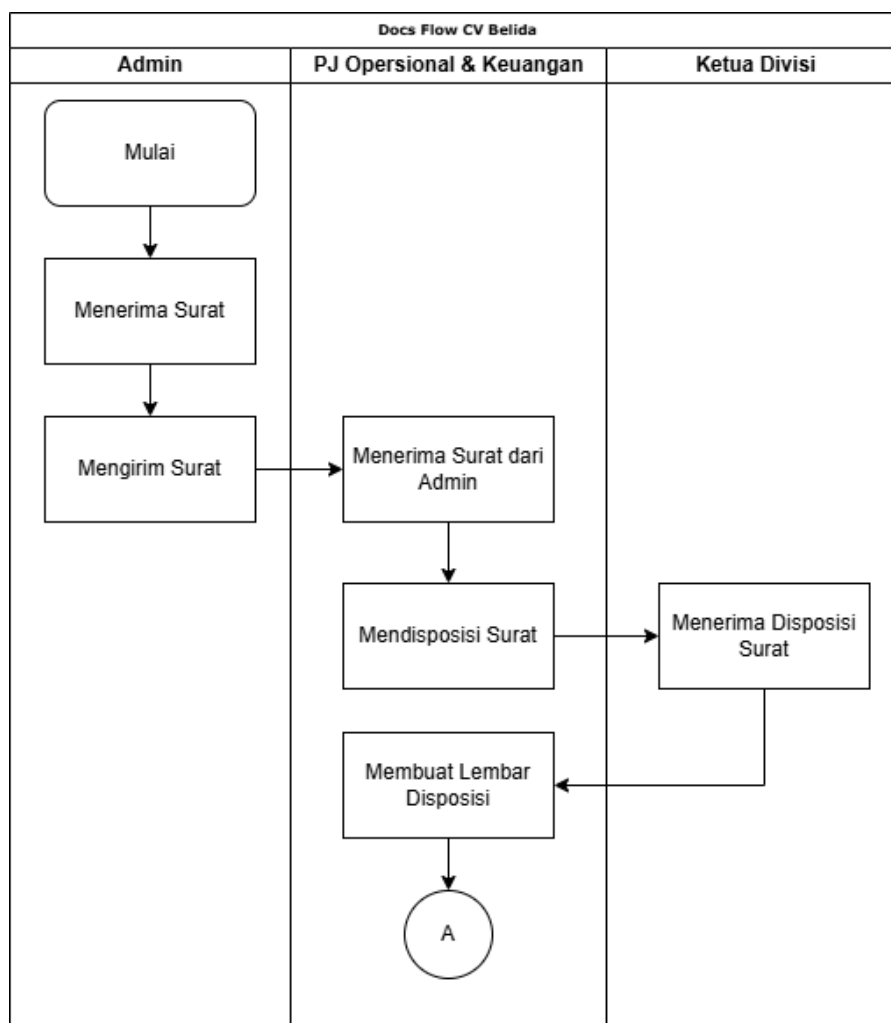
Tahap pertama pada skripsi ini yaitu tahap *Requirements Planning* yang dimana tahapan ini membutuhkan fakta-fakta yang mendukung proses pengerjaan skripsi. Oleh karena itu, untuk mendapatkan fakta tersebut perlu dilakukan studi literatur dan wawancara dengan beberapa pihak yang terkait pada bidang sistem informasi sesuai studi kasus di CV Belida. Pihak yang bersangkutan dalam skripsi ini adalah penanggung jawab operasional pada lingkup CV Belida.

4.1.1.1 Analisis Kebutuhan

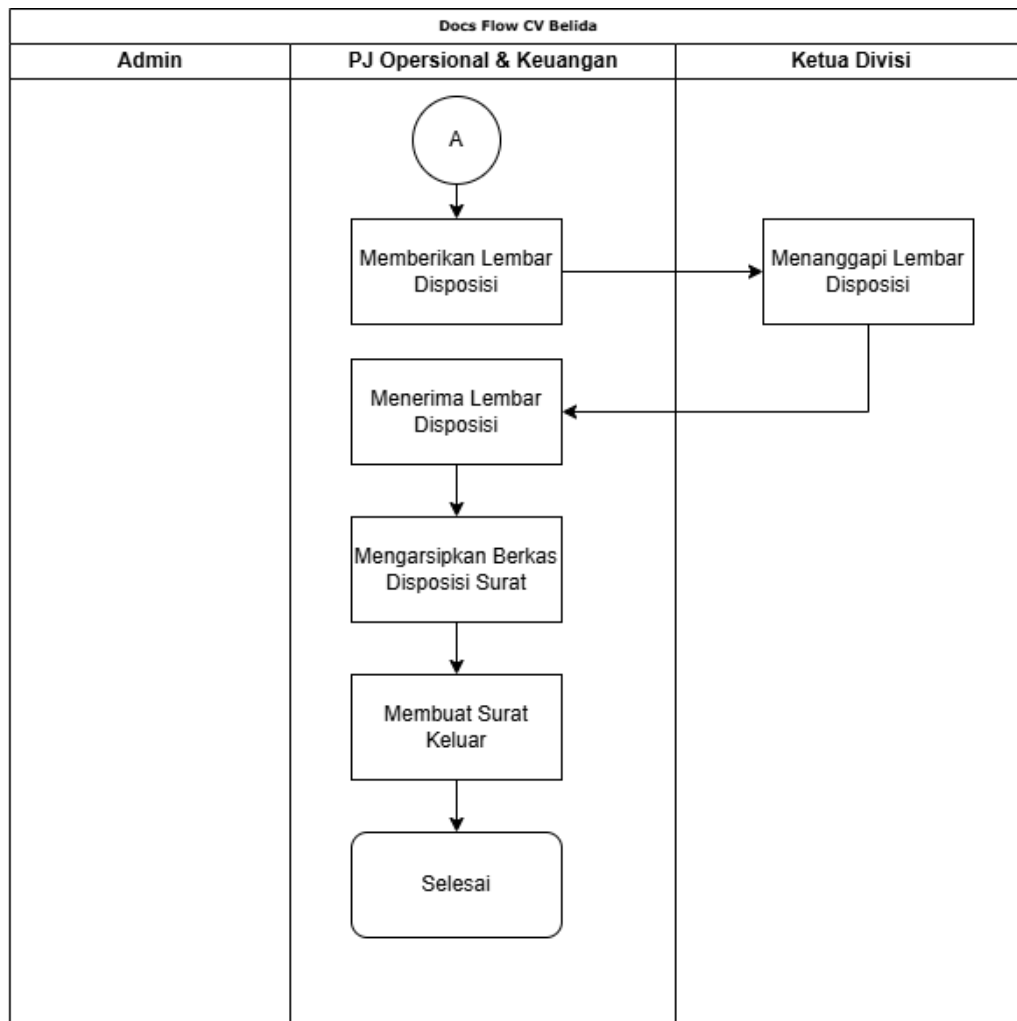
Analisis kebutuhan dilakukan dengan tujuan untuk mendapatkan informasi mengenai bagaimana sistem akan dibuat. Untuk membuat sebuah sistem maka perlu diketahui terlebih dahulu bagaimana alur proses bisnis yang selama ini berjalan pada CV Belida. Metode pengumpulan informasi mengenai kebutuhan perusahaan dilakukan dengan wawancara untuk mendapatkan informasi mengenai kebutuhan

sistem informasi *e-office*. Wawancara dilakukan dengan penanggung jawab operasional CV Belida.

Berdasarkan analisis kebutuhan, didapatkan fakta bahwa alur proses bisnis pada CV Belida diawali dengan pembuatan laporan surat masuk oleh pegawai CV Belida yang bertugas sebagai *admin*. Laporan tersebut kemudian disimpan kedalam penyimpanan laporan, kemudian diteruskan kepada penerima surat. Pegawai CV Belida yang menerima surat, kemudian menindak lanjuti keperluan atau isi surat serta membuat umpan balik berupa surat keluar kepada pengirim surat melalui admin. Kemudian, pihak *admin* akan mengirim laporan umpan balik tersebut untuk bukti bahwa surat sudah ditindak lanjuti. Berikut adalah *DocsFlow* CV Belida dapat dilihat pada Gambar 4.1 dan 4.2.



Gambar 4.1 *DocsFlow* CV Belida



Gambar 4.2 DocsFlow CV Belida

4.1.1.1.1 Analisis Kebutuhan Fungsional

Berdasarkan hasil wawancara dengan Penanggung Jawab Operasional CV Belida serta analisis terhadap alur proses bisnis yang ada, kebutuhan fungsional sistem informasi e-office ini dapat diidentifikasi. Kebutuhan ini dirancang untuk mengatasi kelemahan proses manual dan menyediakan solusi digital yang terstruktur.

Sistem ini akan digunakan oleh beberapa tingkatan pengguna dengan hak akses yang berbeda, sesuai dengan struktur organisasi dan tanggung jawabnya di CV Belida. Pengguna sistem dibagi menjadi 3, yaitu *admin*, penanggung jawab operasional dan keuangan, serta ketua divisi.

Berdasarkan hasil analisis terhadap proses bisnis dan tujuan pengembangan sistem di CV Belida, kebutuhan fungsional untuk sistem e-office ini dapat dirinci sebagai berikut. Sistem dirancang untuk dapat digunakan oleh seluruh pengguna yang

berwenang dengan fitur yang komprehensif untuk mengelola administrasi perkantoran secara digital. Sistem harus dapat melakukan fungsionalitas berikut:

Tabel 4. 1 Kebutuhan Fungsionalitas

Fungsionalitas	Deskripsi
Autentikasi Pengguna	Sistem mampu memvalidasi pengguna melalui proses login menggunakan username dan password, serta menyediakan fungsi logout yang aman.
Pengelolaan Data Induk Karyawan	Sistem menyediakan fitur bagi <i>admin</i> untuk mengelola data pengguna lain, meliputi menambah, mengubah, dan menonaktifkan akun karyawan.
Pencatatan Surat Masuk	Sistem dapat digunakan untuk mencatat data surat masuk baru, termasuk mengisi detail seperti nomor surat, pengirim, perihal, dan tanggal surat
Manajemen Lampiran Surat Masuk	Pengguna dapat mengunggah (upload) dan melampirkan file dokumen berupa pdf atau gambar pada setiap data surat masuk sebagai arsip digital.

Fungsionalitas	Deskripsi
Pembuatan Disposisi Surat	Sistem memungkinkan pengguna untuk membuat lembar disposisi dari sebuah surat masuk yang ditujukan kepada satu atau beberapa pengguna lain di dalam sistem untuk ditindaklanjuti.
Notifikasi Disposisi via <i>WhatsApp</i>	Setelah disposisi dibuat, sistem secara otomatis mengirimkan notifikasi pesan melalui <i>WhatsApp</i> kepada pengguna yang dituju, berisi informasi perihal surat dan instruksi disposisi
Pencatatan Surat Keluar	Sistem menyediakan fungsi untuk membuat dan mencatat data surat keluar, baik sebagai surat baru maupun sebagai balasan dari surat masuk yang sudah ada.
Cetak Surat Keluar dengan Gabungan Lampiran	Sistem memiliki kemampuan untuk mencetak surat keluar beserta lampirannya menjadi satu dokumen PDF tunggal.
Pengelolaan Agenda Internal	Pengguna dapat membuat, mengubah, dan menghapus jadwal agenda atau rapat. Fungsi ini mencakup penentuan nama agenda, waktu pelaksanaan, dan deskripsi kegiatan.

Fungsionalitas	Deskripsi
Manajemen Peserta Agenda & Notifikasi	Sistem memungkinkan pembuat agenda untuk menambahkan peserta dari daftar karyawan dan secara otomatis mengirimkan undangan beserta detail agenda melalui pesan <i>WhatsApp</i> .
Pengelolaan Memo Internal	Pengguna dapat membuat memo internal yang ditujukan kepada pengguna lain, serta dapat melampirkan file pendukung pada memo tersebut.
Pencarian dan Filter Data Dinamis	Sistem menyediakan fungsi pencarian dan <i>filter</i> pada setiap modul utama (Surat Masuk, Surat Keluar, Agenda, Memo) untuk memudahkan pengguna menemukan data secara cepat berdasarkan kata kunci, tanggal, atau kategori.
Tindak Lanjut Disposisi	Pengguna yang menerima disposisi dapat memberikan laporan tindak lanjut dan mengubah status disposisi tersebut (misalnya: "Selesai Dikerjakan" atau "Ditolak").
Manajemen File Terpusat	Pengguna dapat melihat dan mengunduh kembali semua file yang pernah dilampirkan, baik pada surat masuk, surat keluar, maupun memo.

Fungsionalitas	Deskripsi
Manajemen Profil Pengguna	Setiap pengguna dapat melihat dan mengubah data profil pribadinya, seperti nama lengkap, kontak, dan kata sandi.
Manajemen Log Histori Aktivitas Pengguna	Aplikasi dapat mencatat semua aktivitas yang dilakukan oleh pengguna, sehingga dapat digunakan untuk pelacakan aktivitas yang terjadi di aplikasi.

4.1.1.1.2 Analisis Kebutuhan Non Fungsional

Selain kebutuhan fungsional yang mendefinisikan fitur-fitur sistem, pengembangan Sistem Informasi E-Office ini juga harus memenuhi sejumlah kriteria kualitas dan batasan operasional. Kebutuhan non-fungsional ini mendefinisikan atribut kualitas dan batasan pada sistem, yang memastikan bahwa sistem tidak hanya bekerja sesuai fungsi, tetapi juga andal, aman, dan efisien untuk digunakan.

Tabel 4. 2 Kebutuhan Non Fungsionalitas

Non Fungsionalitas	Deskripsi
Keamanan	Sistem wajib memiliki mekanisme login yang aman untuk memastikan hanya pengguna terdaftar yang dapat mengakses data di dalamnya serta menerapkan pembatasan hak akses sesuai dengan peran pengguna.

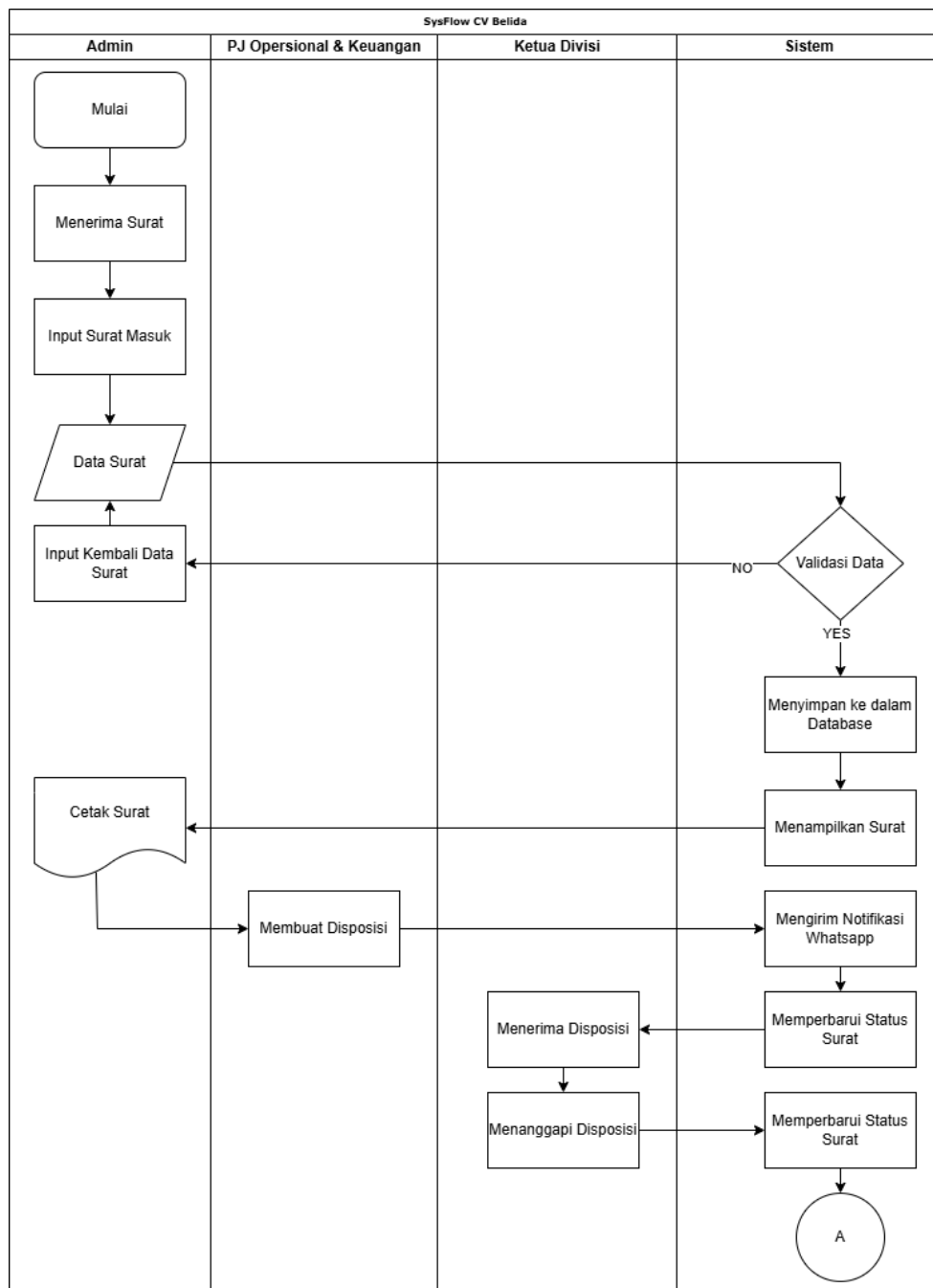
Non Fungsionalitas	Deskripsi
Kehandalan	Sistem harus mampu menangani kesalahan input dari pengguna (misalnya, format isian yang tidak valid) dengan menampilkan pesan yang informatif tanpa menghentikan aplikasi.
Kemudahan Penggunaan	Alur proses untuk tugas-tugas umum, seperti membuat disposisi, harus dirancang sesingkat dan sejasas mungkin untuk mempercepat pekerjaan administratif
Batasan Operasional	Aplikasi dapat berjalan dalam kondisi luring (<i>offline</i>). Koneksi internet hanya dibutuhkan secara spesifik untuk fitur yang memang memerlukan koneksi internet.

4.1.1.2 Alur Proses Bisnis Sistem Surat

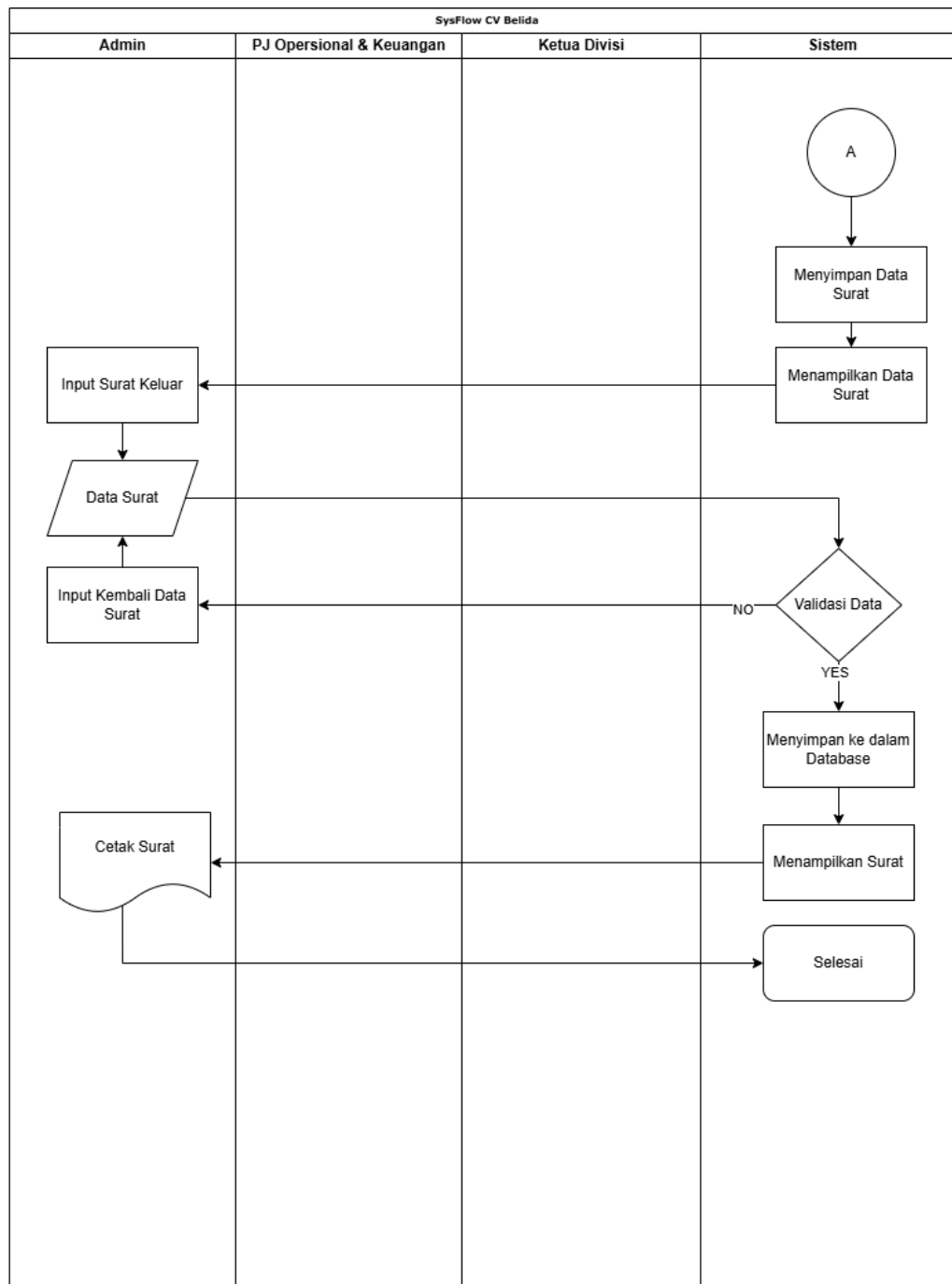
Proses bisnis dimulai dengan admin yang menerima surat laporan, lalu menginput data surat ke dalam sistem. Setelah itu, admin meneruskan surat kepada pegawai yang menerima surat sesuai isi surat. Surat tersebut kemudian dapat diakses oleh pegawai yang menerima, serta akan menanggapi surat tersebut dan membuat disposisi kepada pegawai yang menindak lanjuti isi surat. Setelah di tindak lanjuti admin dapat membuat surat keluar sebagai umpan balik dari surat masuk tersebut. Sistem juga mendukung pencetakan lembar surat dari laporan yang dibuat.

Dalam proses bisnis ini, hanya pegawai yang berposisi sebagai *admin* saja yang memiliki kemampuan untuk membuat surat masuk melalui sistem sesuai kebutuhan mereka. Surat tersebut dapat dikirimkan kepada pihak yang dituju dan pengajuannya dapat mencetak lembar surat sesuai dengan format yang telah ditetapkan oleh sistem.

Saat pembuatan laporan dan surat, sistem mampu menampilkan isi surat. Berikut pada Gambar 4.3 dan 4.4 merupakan *sysflow* proses bisnis sistem surat.



Gambar 4.3 Sysflow Proses Bisnis Sistem Surat

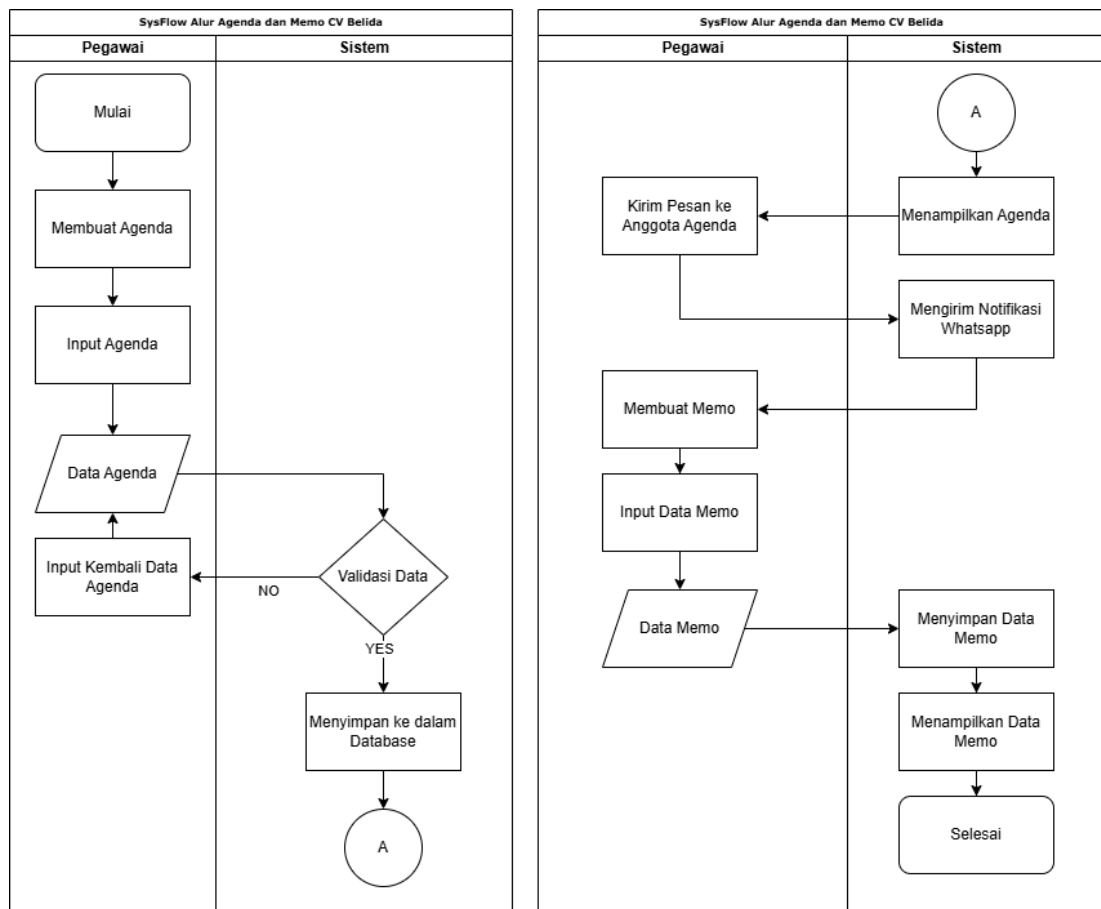


Gambar 4. 4 Sysflow Proses Bisnis Sistem Surat

4.1.1.3 Alur Proses Bisnis Sistem Agenda dan Memo

Proses bisnis sistem agenda dan memo dimulai dengan pegawai yang membuat agenda beserta anggota agenda. Setelah itu, pegawai menyimpan data agenda. Pegawai juga dapat melakukan pengiriman pesan kepada anggota agenda melalui sistem. Pesan dikirim menggunakan platform *whatsapp* dan terkirim secara otomatis. Setelah agenda tersimpan didalam sistem, pegawai dapat membuat *memo* sebagai pencatatan terhadap

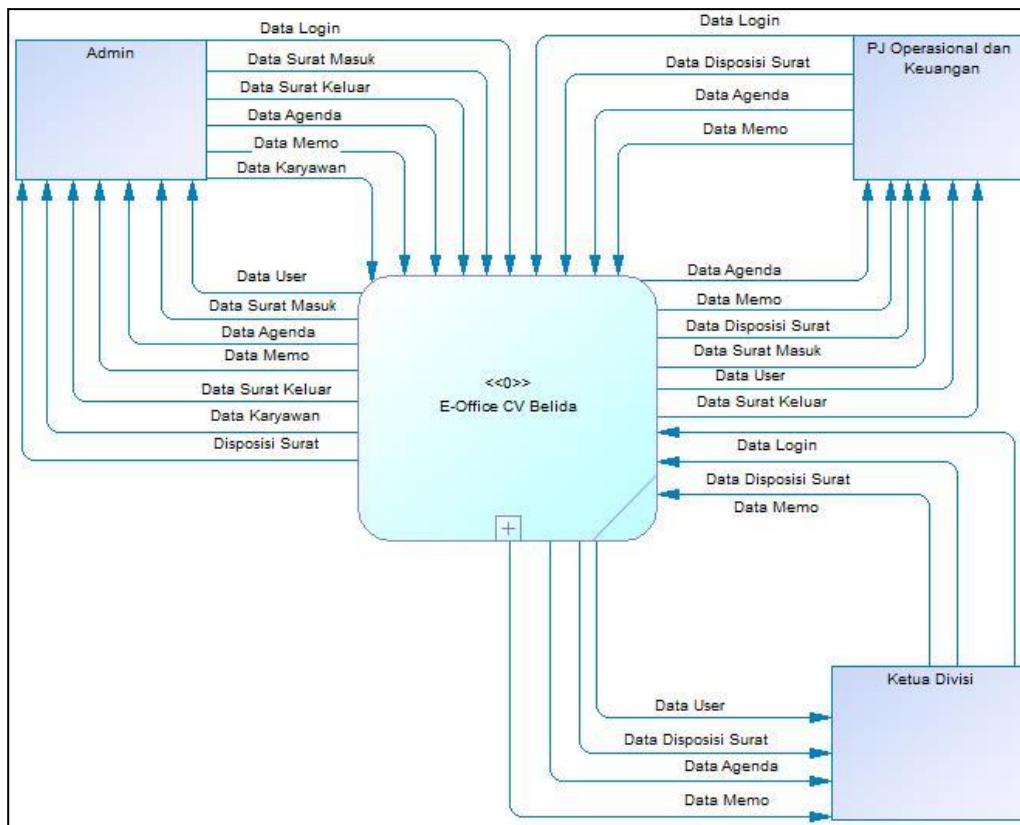
agenda yang telah dibuat. Pembuatan *memo* agenda hanya dapat dibuat oleh pegawai. Berikut pada gambar 4.5 merupakan *sysflow* alur agenda dan memo.



Gambar 4.5 Sysflow Alur Agenda dan Memo

4.1.1.4 Context Diagram

Context Diagram merupakan tahap awal dari perancangan *data flow diagram*. Pada *context diagram* ini digambarkan dengan adanya aliran data dari entitas-entitas ke sistem, dan sebaliknya dari sistem ke entitas-entitas. Untuk lebih jelasnya *context diagram* tersebut dapat dilihat pada Gambar 4.6 sebagai berikut.



Gambar 4.6 Context Diagram E-Office

Pada *context diagram* tersebut, terlihat bahwa sistem *E-Office* CV Belida melibatkan tiga peran utama pengguna, yaitu *admin*, penanggung jawab operasional, ketua divisi, yang masing-masing terdiri dari beberapa jenis pengguna dengan tanggung jawab dan akses yang berbeda.

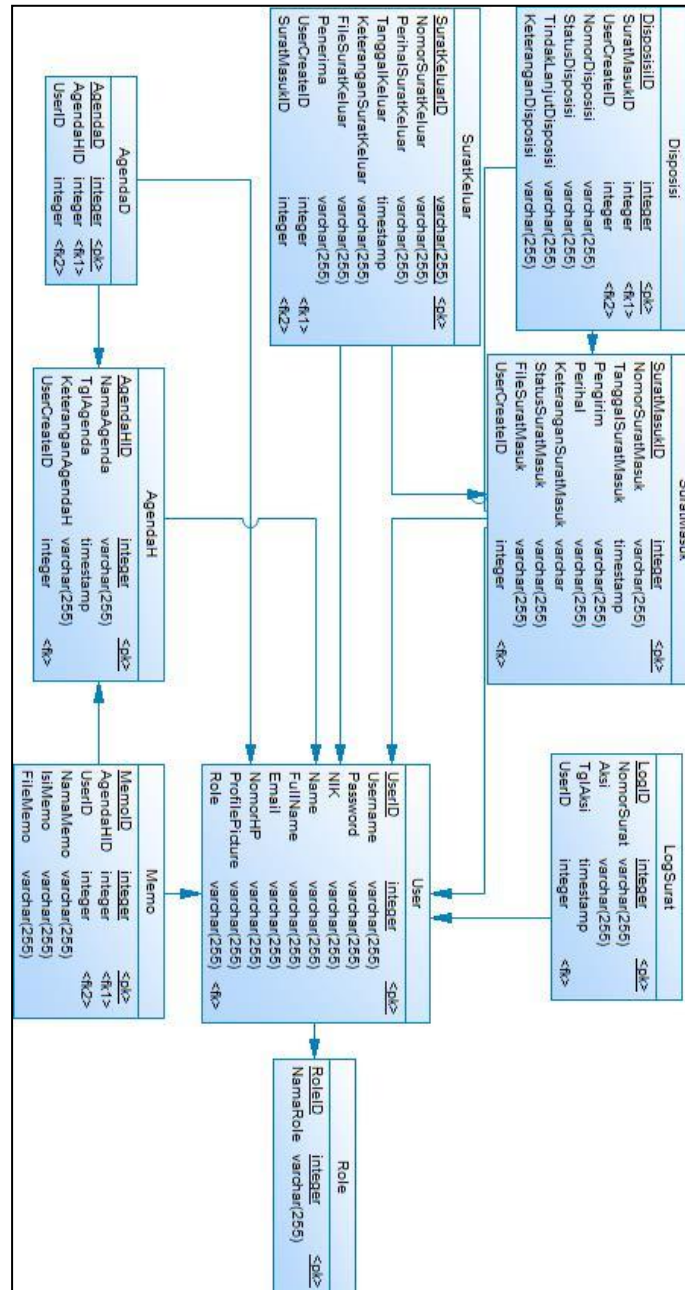
- *Admin* mencakup pengguna:
 - a. *Admin* yang ditugaskan, bertanggung jawab atas pengelolaan data surat masuk, surat keluar, karyawan, agenda, dan memo secara menyeluruh.
 - b. Penanggung Jawab Operasional dan Keuangan, memiliki peran administratif yang fokus pada proses disposisi surat, serta pengelolaan agenda dan memo.
- Karyawan mencakup:
 - a. Ketua Divisi, yang memiliki akses terhadap fitur pembuatan surat keluar, disposisi surat, serta pengelolaan agenda dan memo dalam ruang lingkup divisinya masing-masing.

Setiap pengguna tersebut memiliki hak akses terhadap entitas data tertentu dalam sistem, yang disesuaikan dengan fungsi dan tugas masing-masing. Identifikasi



4.1.1.7 Physical Data Model (PDM)

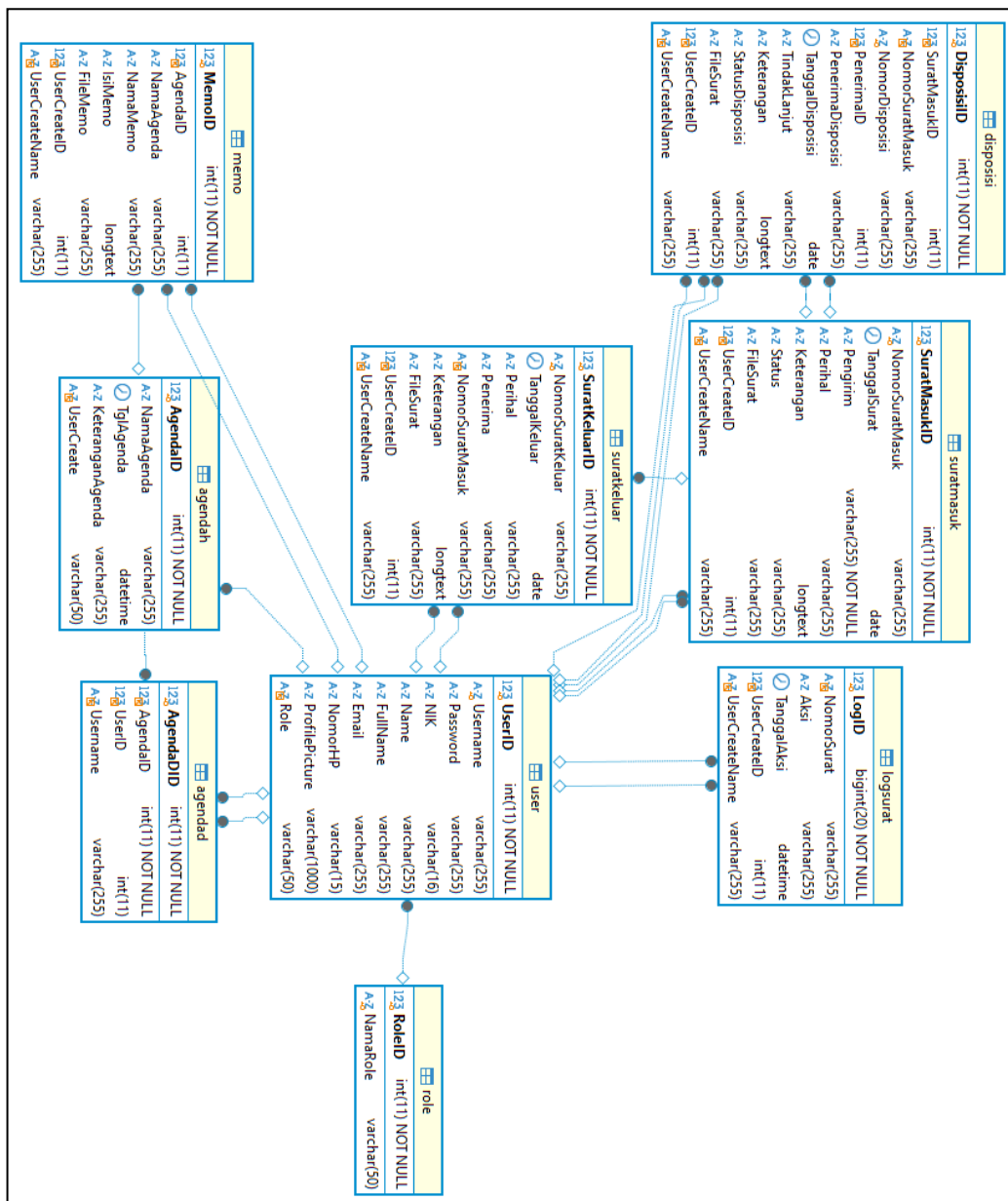
Physical Data Model atau PDM merupakan sebuah tahapan setelah merancang *conceptual data model* (CDM). PDM berisi gambaran detail dari struktur *database* pada sistem informasi *E-Office* berbasis *desktop* pada CV Belida. PDM menggambarkan relasi antara entitas dan tabel dengan *foreign key* (FK). *Physical Data Model* (PDM) dapat dilihat pada Gambar 4.9.



Gambar 4.9 Physical Data Model E-Office

4.1.1.8 Entity Relationship Diagram (ERD)

Entity relationship diagram dapat dijelaskan bahwa masing-masing entitas memiliki atribut-atribut pendukung yang dibutuhkan dalam membuat sebuah rancangan basis data. Antara entitas tersebut dapat berelasi dengan entitas lain dalam satu-kesatuan. Hubungan (relasi) antar entitas tersebut terdapat relasi kardinalitas. Kardinalitas merupakan relasi yang menunjukkan jumlah maksimal entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. *Entity Relationship Diagram E-Office* dapat dilihat pada Gambar 4.10.



Gambar 4.10 Entity Relationship Diagram E-Office

4.1.2 User Design

Setelah dilakukan tahapan *Requirements planning* dan didapatkan hasil wawancara untuk kebutuhan sistem, selanjutnya adalah tahap *User Design* dimana akan dilakukan pembuatan desain *wireframe* antarmuka serta desain arsitektur aliran data yang selanjutnya akan menghasilkan *feedback* atau umpan balik dari pengguna yaitu CV Belida serta melakukan perbaikan yang berhubungan dengan keinginan pengguna dan tahapan ini berfokus pada perancangan dalam pembuatan *software* Sistem Informasi *E-Office* berbasis *Desktop* pada alur proses bisnis termasuk rencana yang nantinya diaplikasikan ke dalam sistem.

4.1.2.1 Design

Tahap *design* ini merupakan tahapan desain *wireframe* sebelum masuk pada implementasi atau *development*. Pada tahapan ini akan menghasilkan *wireframe* yang nantinya dapat digunakan sebagai acuan dalam *development* agar menu, fitur, inputan tertata secara jelas. Desain *wireframe* ini nantinya akan di implementasikan pada Sistem Informasi *E-Office* Berbasis *Desktop*.

4.1.2.1.1 Design Sistem User Admin

Desain sistem untuk *user admin* dirancang untuk memberikan kemudahan dalam mengelola seluruh data dan fitur utama pada Sistem Informasi E-Office. Desain *wireframe* pada bagian ini mencakup tampilan menu surat masuk, surat keluar, disposisi, agenda, memo, dan data karyawan. Tampilan dirancang dengan urutan dan struktur menu yang jelas agar admin dapat memantau dan mengelola data secara terorganisir dan tepat sasaran. Desain ini juga memperhatikan aspek keterpahaman pengguna (*usability*) serta konsistensi antar antarmuka untuk mendukung proses implementasi sistem tanpa membingungkan pengguna.

c. Halaman Tambah Surat Masuk

The screenshot shows the 'E-Office CV Belida' interface. At the top, there's a header with 'Surat Masuk' and a search bar. Below the header, a sidebar on the left contains a menu with 'Surat Masuk' selected. The main content area is titled 'Tambah Surat Masuk' and contains the following fields: 'No Surat' (text input), 'Tanggal' (date picker), 'Pengirim' (text input), 'Perihal' (text input), 'File' (with an 'Upload' button), and 'Keterangan' (text area). At the bottom of the form are 'Simpan' (Save) and 'Print' buttons.

Gambar 4.13 Design Tampilan Halaman Tambah Surat Masuk

Gambar 4.13 merupakan desain halaman tambah surat masuk untuk *admin* terdapat kolom nomor surat, divisi, tanggal, pengirim, perihal, *file*, dan keterangan.

d. Halaman Tambah Disposisi

The screenshot shows the 'E-Office CV Belida' interface. At the top, there's a header with 'Surat Masuk' and a search bar. Below the header, a sidebar on the left contains a menu with 'Surat Masuk' selected. The main content area is titled 'Tambah Surat Disposisi' and contains the following fields: 'No Surat' (with a dropdown menu labeled 'Nomor Surat Masuk'), 'Tanggal' (date picker), 'Pilih Petugas' (dropdown menu labeled 'List User'), 'Perihal' (text input), and 'Keterangan' (text area). At the bottom of the form is a 'Simpan' (Save) button.

Gambar 4.14 Design Tampilan Halaman Tambah Disposisi

Gambar 4.14 merupakan desain halaman tambah disposisi untuk *admin* terdapat kolom nomor surat, tanggal, pilih petugas, perihal, dan keterangan.

e. Halaman Surat Keluar

Gambar 4.15 Design Tampilan Surat Keluar

Gambar 4.15 merupakan desain halaman surat keluar untuk *admin* terdapat kolom nomor surat, divisi, tanggal, penerima, perihal, keterangan, dan *action*.

f. Halaman Tambah Surat Keluar

Gambar 4.16 Design Tampilan Halaman Tambah Surat Keluar

Gambar 4.16 merupakan desain halaman tambah surat keluar untuk *admin* terdapat kolom nomor surat, divisi, tanggal, penerima, perihal, surat masuk, *file*, dan keterangan.

g. Halaman Disposisi Surat

[illegible]

Gambar 4.17 *Design* Tampilan Disposisi Surat

Gambar 4.17 merupakan desain halaman disposisi untuk *admin* terdapat kolom nomor masuk, penerima, tanggal, tindak lanjut, keterangan, status, dan *action*.

h. Halaman *Update* Disposisi

E-Office CV Belida

...

CV Belida

Surat Masuk

Search....

Tambah

Profile

Halo "Nama"

Jabatan

Surat Masuk

Surat Keluar

Disposisi

Agenda

Memo

Data Karyawan

Tindak Lanjut

No Surat

Status

Selesai/Tindak Lanjut/Ditolak

Tindak Lanjut

Simpan

Gambar 4.18 *Design* Tampilan Halaman Tambah Update Disposisi

Gambar 4.18 merupakan desain halaman *update* disposisi untuk *admin* terdapat kolom nomor surat, status, dan tindak lanjut.

i. Halaman Agenda

The design of the Agenda page for an admin user includes a header 'E-Office CV Belida' and a sidebar with navigation links: Surat Masuk, Surat Keluar, Disposisi, Agenda, Memo, and Data Karyawan. The main content area features a search bar, a calendar for February 2025, and a table with columns: Tanggal, Agenda, Waktu, User Create, Keterangan, and Action. A 'Tambah' button is located next to the calendar.

Gambar 4.19 Design Tampilan Agenda

Gambar 4.19 merupakan desain halaman agenda untuk *admin* terdapat kolom tanggal, agenda, waktu, *usercreate*, keterangan, dan *action*.

j. Halaman Tambah Agenda

The design of the Add Agenda page for an admin user includes a header 'E-Office CV Belida' and a sidebar with navigation links: Surat Masuk, Surat Keluar, Disposisi, Agenda, Memo, and Data Karyawan. The main content area features a search bar, a 'Tambah' button, and a form titled 'Tambah Agenda' with fields for Tanggal, Agenda, Waktu, Anggota, and Keterangan. The form also includes 'Show', 'Kirim Whatsapp', and 'Simpan' buttons.

Gambar 4.20 Design Tampilan Halaman Tambah Agenda

Gambar 4.20 merupakan desain halaman tambah agenda untuk *admin* terdapat kolom tanggal, agenda, waktu, anggota, dan keterangan.

m. Halaman Tambah Memo

...

CV Belida

Surat Masuk

Search...

Tambah

Profile

Halo "Nama"

Jabatan

Surat Masuk

Surat Keluar

Disposisi

Agenda

Memo

Data Karyawan

Tambah Memo

Memo

Agenda

File

Keterangan

Simpan

Gambar 4.23 *Design* Tampilan Halaman Tambah Memo

Gambar 4.23 merupakan desain halaman tambah memo untuk *admin* terdapat kolom memo, agenda, *file*, dan keterangan.

n. Halaman Data Karyawan

[illegible]

Gambar 4.24 *Design* Tampilan Data Karyawan

Gambar 4.24 merupakan desain halaman data karyawan untuk *admin* terdapat kolom *username*, NIK, nama, nama lengkap, email, status, dan tindak lanjut.

o. Halaman Tambah Data Karyawan



CV Belida

Surat Masuk

Search....

Halo "Nama"
Jabatan

Profile

Tambah

Surat Masuk

Surat Keluar

Disposisi

Agenda

Memo

Data Karyawan

Tambah Data Karyawan

Username

Password

NIK

Name

Fullname

Email

Nomor HP

Role

Foto

Simpan

Gambar 4.25 *Design* Tampilan Halaman Tambah Data Karyawan

Gambar 4.25 merupakan desain halaman tambah data karyawan untuk *admin* terdapat kolom *username*, *password*, NIK, nama, *fullname*, email, nomor hp, *role*, dan foto.

k. Halaman Log

[illegible]

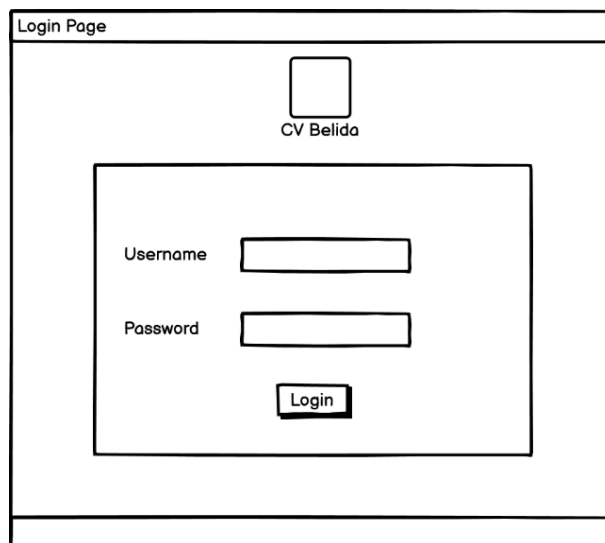
Gambar 4. 26 Design Tampilan Halaman Log

Gambar 4.26 merupakan desain halaman log untuk *admin* terdapat kolom tanggal dan waktu, aksi, nomor terkait, dan pengguna.

4.1.2.1.2 Design Sistem User Karyawan

Desain sistem untuk *user* karyawan dirancang untuk memberikan kemudahan dalam mengelola data dan fitur disposisi, agenda, memo, data *user* pada Sistem Informasi E-Office. Desain *wireframe* pada bagian ini mencakup tampilan menu disposisi, agenda, memo, dan data karyawan. Tampilan dirancang dengan urutan dan struktur menu yang jelas agar admin dapat memantau dan mengelola data secara terorganisir dan tepat sasaran. Desain ini juga memperhatikan aspek keterpahaman pengguna (*usability*) serta konsistensi antar antarmuka untuk mendukung proses implementasi sistem tanpa membingungkan pengguna.

a. Halaman *Login*



Gambar 4. 27 Design Tampilan Login

Gambar 4.27 merupakan desain *wireframe* halaman login untuk karyawan terdapat kolom pengisian *username* dan *password*.

b. Halaman Disposisi Surat

[illegible]

Gambar 4. 28 Design Tampilan Disposisi Surat

Gambar 4.28 merupakan desain halaman disposisi untuk karyawan terdapat kolom nomor masuk, penerima, tanggal, tindak lanjut, keterangan, status, dan *action*.

c. Halaman *Update* Disposisi

E-Office CV Belida

...

CV Belida

Surat Masuk

Search....

Tambah

Profile

Halo "Nama"

Jabatan

Surat Masuk

Surat Keluar

Disposisi

Agenda

Memo

Data Karyawan

Tindak Lanjut

No Surat

Status

Selesai/Tindak Lanjut/Ditolak

Tindak Lanjut

Simpan

Gambar 4. 29 Design Tampilan Halaman Tambah Update Disposisi

Gambar 4.29 merupakan desain halaman *update* disposisi untuk karyawan terdapat kolom nomor surat, status, dan tindak lanjut.

d. Halaman Agenda

Gambar 4. 30 Design Tampilan Agenda

Gambar 4.30 merupakan desain halaman agenda untuk karyawan terdapat kolom tanggal, agenda, waktu, *usercreate*, keterangan, dan action.

e. Halaman Tambah Agenda

Gambar 4. 31 Design Tampilan Halaman Tambah Agenda

Gambar 4.31 merupakan desain halaman tambah agenda untuk karyawan terdapat kolom tanggal, agenda, waktu, anggota, dan keterangan.

f. Halaman Tambah Anggota Agenda

E-Office CV Belida

Surat Masuk

CV Belida

Search....

Tambah

Profile

Halo "Nama" Jabatan

Surat Masuk

Surat Keluar

Disposisi

Agenda

Memo

Data Karyawan

Tambah Anggota Agenda

Action	Nama
☐	Nama User
☒	Nama User
☐	Nama User

Simpan

Gambar 4. 32 Design Tampilan Halaman Tambah Anggota Agenda

Gambar 4.32 merupakan desain halaman tambah anggota agenda untuk karyawan terdapat kolom *action*, dan nama.

g. Halaman *Memo*

E-Office CV Belida

Memo

CV Belida

Search....

Tambah

Profile

Halo "Nama" Jabatan

Surat Masuk

Surat Keluar

Disposisi

Agenda

Memo

Data Karyawan

No	Nama Memo	Agenda	Isi Memo	File Memo	Action

Gambar 4. 33 Design Tampilan Memo

Gambar 4.33 merupakan desain halaman memo untuk karyawan terdapat kolom nomor nama memo, agenda, isi memo, *file* memo, dan *action*.

j. Halaman Ubah Data Karyawan

The screenshot shows a web application interface for managing employee data. The main form, titled 'Tambah Data Karyawan', includes the following fields and controls:

- Username: Text input field
- Password: Text input field
- NIK: Text input field
- Name: Text input field
- Fullname: Text input field
- Email: Text input field
- Nomor HP: Text input field
- Role: Dropdown menu with 'List Role' as the selected option
- Foto: Text input field with an 'Upload' button
- Simpan: Save button

Gambar 4. 36 Design Tampilan Halaman Tambah Data Karyawan

Gambar 4.36 merupakan desain halaman ubah data karyawan untuk karyawan terdapat kolom *username*, *password*, NIK, nama, *fullname*, email, nomor hp, *role*, dan foto.

4.1.2.2 Testing dan Refine

Testing dan refine dilakukan secara iteratif terhadap fitur penomoran surat. Tujuan dari tahap ini adalah untuk mengidentifikasi kelemahan dari desain awal, mendapatkan umpan balik dari pengguna, serta menyempurnakan fitur agar sesuai dengan kebutuhan operasional.

a. Iterasi 1: Desain Awal dengan Input Manual pada Surat Masuk dan Keluar

Pada prototipe awal, fitur penomoran surat dirancang sepenuhnya manual. Pengguna (admin) diwajibkan untuk mengetik sendiri nomor surat secara lengkap pada sebuah *textbox* standar. Pendekatan ini awalnya dipilih untuk memberikan fleksibilitas, namun setelah diuji coba oleh pengguna, ditemukan beberapa kelemahan signifikan.

Umpan balik utama dari pengguna adalah bahwa input manual sangat rentan terhadap *human error*, seperti kesalahan ketik, duplikasi nomor, dan inkonsistensi format penulisan, terutama pada kode divisi. Pengguna menginginkan sebuah sistem yang lebih terstruktur dan dapat meminimalkan potensi kesalahan tersebut.

E-Office CV Belida

...

CV Belida

Surat Masuk

Search....

Tambah

Halo "Nama"

Jabatan

Profile

Surat Masuk

Surat Keluar

Disposisi

Agenda

Memo

Data Karyawan

Tambah Surat Masuk

No Surat

Tanggal

/ /

Pengirim

Perihal

File

Upload

Keterangan

Simpan

Print

Gambar 4. 37 Design Tampilan Tambah Surat Masuk Input Manual

E-Office CV Belida

...

CV Belida

Surat Masuk

Search....

Tambah

Halo "Nama"

Jabatan

Profile

Surat Masuk

Surat Keluar

Disposisi

Agenda

Memo

Data Karyawan

Tambah Surat Keluar

No Surat

Tanggal

/ /

Penerima

Perihal

Surat Masuk

Nomor Surat Masuk

File

Upload

Keterangan

Simpan

Print

Gambar 4. 38 Design Tampilan Tambah Surat Keluar Input Manual

Tabel 4. 3 Iterasi 1: Umpan Balik Pengguna terhadap Penomoran Manual

Deskripsi Fitur	Feedback
Pada prototipe awal, input nomor surat dilakukan secara manual oleh pengguna setiap kali menambahkan surat masuk atau surat keluar.	Pengguna menyatakan bahwa input manual berpotensi menimbulkan duplikasi nomor surat. Selain itu, pengguna menginginkan agar sistem dapat secara otomatis menghasilkan nomor surat berdasarkan pola tertentu, misalnya: SM/001/PD/05/2025.

b. Iterasi 2: Penyempurnaan dengan Penomoran Otomatis dan Opsi Divisi

Berdasarkan umpan balik dari iterasi pertama, dilakukan perbaikan signifikan pada fitur penomoran surat. *Textbox* manual digantikan dengan sistem penomoran semi-otomatis yang lebih cerdas.

The screenshot displays the 'E-Office CV Belida' web application. At the top, there's a header with a search bar for 'Surat Masuk' and a user profile section showing 'Halo "Nama" Jabatan' and a 'Tambah Profile' button. A sidebar on the left contains navigation links: 'Surat Masuk', 'Surat Keluar', 'Disposisi', 'Agenda', 'Memo', and 'Data Karyawan'. The main content area features a 'Tambah Surat Keluar' form. This form contains several input fields: 'No Surat', 'Tanggal' (with a calendar icon), 'Penerima', 'Perihal', 'Surat Masuk' (a dropdown menu currently showing 'Nomor Surat Masuk'), and 'File' (with an 'Upload' button). There is also a 'Divisi' dropdown menu. A large text area for 'Keterangan' is at the bottom of the form. At the very bottom of the form, there are 'Simpan' and 'Print' buttons.

Gambar 4. 39 Design Tampilan Tambah Surat Masuk Input Otomatis

Gambar 4. 40 Design Tampilan Tambah Surat Keluar Input Otomatis

Tabel 4. 4 Iterasi 2: Penyesuaian Sistem dan Hasil Validasi

Deskripsi Fitur	Feedback	Hasil
Sistem diperbarui untuk menghasilkan nomor surat secara otomatis. Pola penomoran mengikuti format: Nomor Urut/Bidang/Bulan/Tahun. Nomor surat otomatis di-<i>generate</i> ketika pengguna memilih kategori dan tanggal surat.	Pengguna menyatakan fitur sudah sesuai kebutuhan dan mengurangi risiko kesalahan input. Disarankan agar tetap ada opsi <i>override</i> manual bila dibutuhkan (ditambahkan sebagai catatan pengembangan lanjutan).	Fitur dianggap final dan siap dilanjutkan ke tahap <i>Construction</i>.

Melalui dua iterasi tersebut, sistem disesuaikan dengan kebutuhan pengguna untuk meminimalkan kesalahan input dan meningkatkan konsistensi data. Tahapan ini merupakan bagian dari metode Rapid Application Development (RAD) yang

menekankan iterasi dan keterlibatan pengguna secara aktif dalam proses pengembangan.

4.1.3 Construction

Tahap ketiga yang bisa disebut tahap *Construction* merupakan tahap inti dari pembuatan sistem. Ditahapan inilah yang secara nyata menunjukkan proses pengerjaan *software* yang akan memaksimalkan penggunaan *hardware*. Proses ini akan menjelaskan bagaimana sistem dibangun mulai dari pemilihan *database (basis data)* yang akan digunakan, Bahasa pemrograman yang akan dipakai sampai pada tahap *coding* (pengkodean) dan sampai juga pada tahap pengujian sistem seiring tahap *development*.

Pembuatan sistem informasi ini dibangun berbasis *desktop* dengan menggunakan pemrograman terstruktur sebagai Bahasa pemrograman C# *.Net Framework* untuk membangun aplikasi yang mudah dikembangkan dan mudah juga untuk menemukan bagian yang salah ketika nantinya pada pengujian sistem. Kemudian basis data yang digunakan pada bagian *back-end* Sistem Informasi E-Office ini adalah basis data MySQL.

4.1.3.1 Database Construction

Pada tahap *Development Database*, dirancang dengan mengacu pada perancangan basis data yang telah dibuat pada tahap *system design*. Pada *development database* sistem informasi *e-office* berbasis *desktop* pada CV Belida memiliki 9 tabel, yaitu agendah, agendad, user, role, disposisi, logsurat, memo, suratmasuk, suratkeluar. Di bawah ini merupakan *Development Database* dari system *e-office* Gambar 4.41 berikut.

Name	Rows	Size ^	Created	Updated	Engine	Comment	Type
agendah	3	32,0 KiB	2024-12-01 15:45:04		InnoDB		Table
role	2	32,0 KiB	2024-11-29 20:17:33		InnoDB		Table
user	11	48,0 KiB	2024-12-01 15:44:46		InnoDB		Table
agendad	13	64,0 KiB	2024-12-03 19:42:10		InnoDB		Table
disposisi	0	64,0 KiB	2024-12-01 15:59:52		InnoDB		Table
logsurat	0	64,0 KiB	2024-12-01 16:01:08		InnoDB		Table
memo	1	64,0 KiB	2025-02-16 22:46:51		InnoDB		Table
suratmasuk	0	64,0 KiB	2024-12-01 16:03:47		InnoDB		Table
suratkeluar	0	80,0 KiB	2024-12-01 16:03:09		InnoDB		Table

Gambar 4. 41 Database Construction

4.1.3.2 Application Construction

Pada tahap *Application Construction*, akan memulai melakukan pembuatan sistem dengan mengacu pada perancangan *wireframe* yang telah dibuat pada tahap *user design*. Pada *application construction* sistem informasi *e-office* berbasis *desktop* pada CV Belida memiliki 5 fitur utama yang mengacu pada alur proses bisnis sistem. Pada tahap ini akan menjabarkan bagaimana cara kerja dari sistem informasi *e-office* berbasis *desktop* pada CV Belida. Alur kerja sistem akan terdiri 2 *role* yang berfungsi untuk membatasi hak akses fitur tiap *user*.

4.1.3.2.1 Sistem User Admin

User Admin sendiri memiliki semua hak akses fitur sistem, yang berfungsi sebagai pengatur dan pengolah data di sistem. Hak akses *admin* sendiri akan diberikan kepada penanggung jawab operasional dan keuangan, serta admin yang berikan tugas.

4.1.3.2.1.1 Halaman Login

Gambar 4.42 ini merupakan potongan kode program yang berfungsi untuk proses login memeriksa kecocokan *username* dan *password users* dengan yang ada dalam *database*.

```
public LoginViewModel()
{
    userRepository = new UserRepository();
    LoginCommand = new ViewModelCommand(ExecuteLoginCommand,
CanExecuteLoginCommand);
    RecoverPasswordCommand = new ViewModelCommand(p =>
ExecuteRecoverPassCommand("", ""));
}

private bool CanExecuteLoginCommand(object obj)
{
    bool validData;
    if (string.IsNullOrEmpty(Username) || Username.Length < 3 ||
        Password == null || Password.Length < 3)
        validData = false;
    else
```

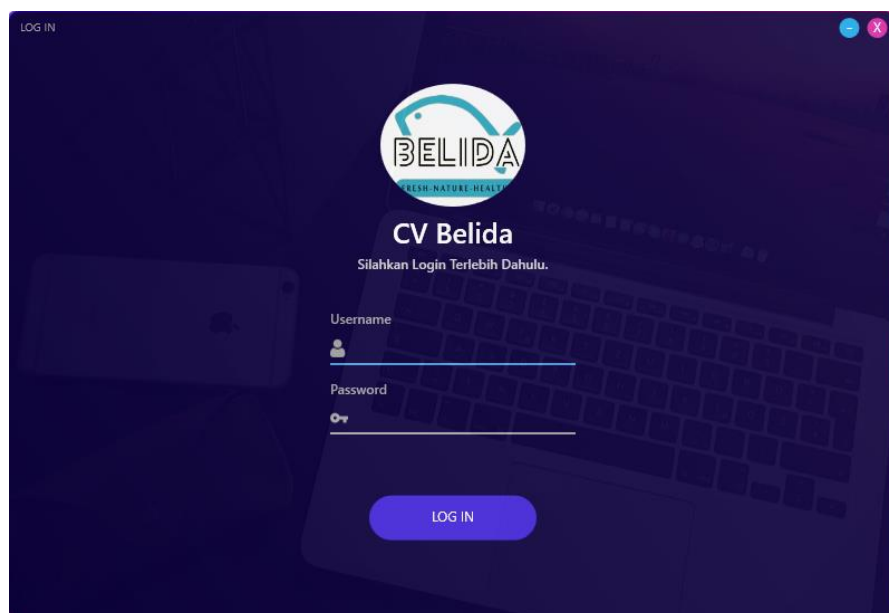
```

        validData = true;
        return validData;
    }

    private void ExecuteLoginCommand(object obj)
    {
        var    isValidUser    =    userRepository.AuthenticateUser(new
NetworkCredential(Username, Password));
        if (isValidUser)
        {
            Thread.CurrentPrincipal = new GenericPrincipal(
                new GenericIdentity(Username), null);
            IsViewVisible = false;
        }
        else
        {
            ErrorMessage = "* Invalid username or password";
        }
    }
}

```

Gambar 4. 42 Potongan Kode Sistem Login



Gambar 4. 43 Halaman Login

Gambar 4.43 merupakan tampilan dari halaman form login untuk users. Terdapat kolom pengisian username dan *password*, serta tombol login.

4.1.3.2.1.2 Halaman Data Karyawan

Gambar 4.44 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data profilnya. Kode ini memungkinkan pengguna untuk mengubah data profilnya, seperti nama lengkap dan *role*. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public DataKaryawanViewModel()
{
    _userRepository = new UserRepository();

    string userRole = MainViewModel.CurrentUserStatic?.Role;
    int userId = MainViewModel.CurrentUserStatic?.UserID ?? 0;

    if (IsAdmin())
    {
        var users = _userRepository.GetByAll();
        foreach (var user in users)
        {
            if (!string.IsNullOrEmpty(user.ProfilePicture))
            {
                user.ProfilePicture =
                System.IO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
                "Images", user.ProfilePicture);
            }
        }
        Items = new ObservableCollection<UserModel>(users);
    }
    else
    {
        var user = _userRepository.GetById(userId);
        Items = new ObservableCollection<UserModel>();
    }
}
```

```

        if (user != null)
        {
            if (!string.IsNullOrEmpty(user.ProfilePicture))
            {
                user.ProfilePicture =
System.IO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Images", user.ProfilePicture);
            }
            Items.Add(user);
        }
    }

    FilteredItems = new ObservableCollection<UserModel>(Items);
    Message = "Data Karyawan View";

    ShowTambahKaryawanCommand = new
ViewModelCommand(ExecuteShowTambahKaryawan);
    ShowPrintCommand = new ViewModelCommand(ExecuteShowPrint);
    DeleteUserCommand = new ViewModelCommand(ExecuteDeleteUser);
    EditUserCommand = new ViewModelCommand(ExecuteEditUser);
}

```

Gambar 4. 44 Potongan Kode Sistem Data Karyawan

ID	Username	NIK	Nama	Nama Lengkap	Email	No HP	Foto	Role	Action
1	admin	12345678	Admin	Randy Dwi Saputra	admin@gmail.com	6285856440997		Admin	Edit Hapus
2	pi1	3578020903020001	Abiyoga (Penanggung	Abiyoga Dwi Permana	dwisahputarandy@gmail.com	6285856440997		Admin	Edit Hapus
3	pi2	12345666	Abiyoga Iphone (Penan	Abiyoga Dwi Permana	ipenanggungjawab2@gmail.com	628134871384701		Admin	Edit Hapus
4	aset	123456789	Hilman Habib (Ketua D	Hilman Habib (Ketua D	runday@gmail.com	6282143315514		Karyawan	Edit Hapus
5	produksi	test	Adam (Ketua Divisi Pro	Adam (Ketua Divisi Pro	test	6281233895599		Karyawan	Edit Hapus
7	logistik	12345678	Logistik	Ketua Divisi Logistik	logistik@email.com	6285856440997		Karyawan	Edit Hapus

Gambar 4. 45 Halaman Data Karyawan

Gambar 4.45 merupakan hasil dari implementasi kode program halaman edit profile. Terdapat kolom username, nama lengkap, NIK, nama, nama lengkap, email, nomor hp, foto profil, role, ttd dan tombol simpan *profile*.

4.1.3.2.1.3 Halaman Tambah Data Karyawan

Gambar 4.46 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data profilnya. Kode ini memungkinkan pengguna untuk mengubah data profilnya, seperti nama lengkap dan *role*. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```

public TambahKaryawanViewModel()
{
    var userRepository = new UserRepository();
    var roles = userRepository.GetAllRoles();
    Roles = new ObservableCollection<RoleModel>(roles);
    SelectedRole = Roles.FirstOrDefault();
    ShowProfileCommand = new
ViewModelCommand(ExecuteShowProfile);
    AddUserCommand = new ViewModelCommand(ExecuteAddUser);
    ButtonContent = "Tambah";
    if (!Directory.Exists(_imageDirectory))
        Directory.CreateDirectory(_imageDirectory);}

```

Gambar 4. 46 Potongan Kode Sistem Tambah Data Karyawan

Tambah Karyawan

Username admin

Password

NIK 12345678

Name Admin

FullName Randy Dwi Saputra

Email admin@gmail.com

No HP 6285856440997

Profile Upload profil.jpg

Role Admin

Tambah

Gambar 4. 47 Halaman Tambah Karyawan

Gambar 4.47 merupakan hasil dari implementasi kode program halaman edit profile. Terdapat kolom username, nama lengkap, NIK, nama, nama lengkap, email, nomor hp, foto profil, role, ttd dan tombol simpan *profile*.

4.1.3.2.1.4 Halaman Surat Masuk

Gambar 4.48 berisi potongan kode program halaman surat masuk. Kode program tersebut berfungsi untuk menampilkan data dalam bentuk tabel, pengguna dapat dengan mudah melihat daftar surat masuk dan melakukan tindakan yang diperlukan, seperti mengedit atau menghapus data.

```
public SuratMasukViewModel()
{
    _suratMasukRepository = new SuratMasukRepository();
    _disposisiRepository = new DisposisiRepository();
}
```

```

Items = new
ObservableCollection<SuratMasukModel>(_suratMasukRepository.GetAllSuratM
asuk());

FilteredItems = new ObservableCollection<SuratMasukModel>(Items);
Message = "Data Surat Masuk View";
ShowAddSuratMasukCommand = new
ViewModelCommand(ExecuteShowAddSuratMasuk);
ShowEditSuratMasukCommand = new
ViewModelCommand(ExecuteShowEditSuratMasuk,
CanExecuteEditSuratMasuk);
DeleteSuratMasukCommand = new
ViewModelCommand(ExecuteDeleteSuratMasuk);
ShowDisposisiCommand = new
ViewModelCommand(ExecuteShowDisposisi);
}

```

Gambar 4. 48 Potongan Kode Surat Masuk

Nomor Surat	Tanggal Surat	Pengirim	Perihal	Keterangan	Status	Action
SM/00/PO/06/25	31 Mei 2025	PT Mekar Jaya	Surat Pengaduan	Surat Pengaduan dari Mekar Jaya	Disposisi	Disposisi Edit Hapus
SM/00/BLD/06/25	10 Juni 2025	PT Mekar	Surat Permohonan R	Telah di terima surat akan ditujuk	Diterima	Disposisi Edit Hapus

Gambar 4. 49 Halaman Surat Masuk

Gambar 4.89 merupakan tampilan halaman surat masuk. Halaman tersebut berisi judul halaman surat masuk, tombol tambah surat, kolom pencarian, dan data surat masuk yang ada pada *database* beserta aksi disposisi, *edit*, dan hapus.

4.1.3.2.1.5 Halaman Tambah Surat Masuk

Gambar 4.50 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data surat masuk. Kode ini memungkinkan pengguna untuk mengubah, mencetak surat masuk. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public AddSuratMasukViewModel(ISuratMasukRepository
suratMasukRepository, SuratMasukModel selectedSurat)
{
    _suratMasukRepository = suratMasukRepository;
    SuratMasuk = selectedSurat ?? new SuratMasukModel();
    SelectedFilePath = selectedSurat?.FileSurat;
    _logSuratRepository = new LogSuratRepository();
    IsEditMode = true;
    if (!string.IsNullOrEmpty(selectedSurat?.NomorSuratMasuk))
    {
        var parts = selectedSurat.NomorSuratMasuk.Split('/');
        if (parts.Length >= 4) // Format: SM/{Nomor}/{JenisSurat}/{Tahun}
        {
            NomorSuratMasuk = parts[1];
            SelectedJenisSurat = JenisSuratOptions.FirstOrDefault(o => o.Value
== parts[2]);
        }
        else
        {
            NomorSuratMasuk = selectedSurat.NomorSuratMasuk;
        }
    }
    SaveCommand = new ViewModelCommand(ExecuteSave);
    CancelCommand = new ViewModelCommand(ExecuteCancel);
    UploadFileCommand = new ViewModelCommand(ExecuteUploadFile);
    PrintCommand = new ViewModelCommand(ExecutePrint);
    OpenSelectedFileCommand = new
ViewModelCommand(OpenSelectedFile); }
```

Gambar 4. 50 Potongan Kode Halaman Tambah Surat Masuk

Tambah Surat Masuk

Nomor Surat 002 Produksi

Tanggal Saturday, May 31, 2025 15

Pengirim PT Mekar Jaya

Perihal Surat Pengaduan

File Surat Ganti File Open ChatGPT Image Apr 1

Keterangan Surat Pengaduan dari Mekar Jaya terkait produk

Simpan Print

Gambar 4. 51 Halaman Tambah Surat Masuk

Gambar 4.51 merupakan hasil dari implementasi kode program halaman tambah surat masuk. Terdapat kolom nomor surat, tanggal surat, pengirim, perihal, file, keterangan, serta tombol simpan dan *print*.

4.1.3.2.1.6 Halaman Tambah Disposisi

Gambar 4.52 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data dari surat masuk ke disposisi. Kode ini memungkinkan pengguna untuk menambah disposisi serta memilih pegawai. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public AddDisposisiViewModel(ISuratMasukRepository
suratMasukRepository,
    IDisposisiRepository disposisiRepository)
{
    _suratMasukRepository = suratMasukRepository;
    _disposisiRepository = disposisiRepository;
    _logRepository = new LogSuratRepository();
    Disposisi = new DisposisiModel();
}
```

```

SuratMasuk = new SuratMasukModel();
var nomorWhatsAppList = new List<string>();
var namaList = new List<string>();
var userRepository = new UserRepository();
IsAvailable = true;
var penerima = userRepository.GetByAll();
Penerima = new ObservableCollection<UserModel>(penerima);
SaveCommand = new ViewModelCommand(ExecuteSave);
}

```

Gambar 4. 52 Potongan Kode Sistem Tambah Disposisi

Gambar 4. 53 Halaman Tambah Disposisi

Gambar 4.53 merupakan hasil dari implementasi kode program halaman tambah disposisi. Terdapat kolom nomor disposisi, nomor surat masuk, pilih pegawai, status, keterangan, serta tombol simpan.

```

private async Task<HttpResponseMessage> SendMessageToWhatsApp(string
contact, string message, string countryCode)
{
    var messageData = new Dictionary<string, string>
    {
        { "target", contact },
        { "message", message },
        { "countryCode", countryCode } // Menggunakan kode negara dari
parameter
    };

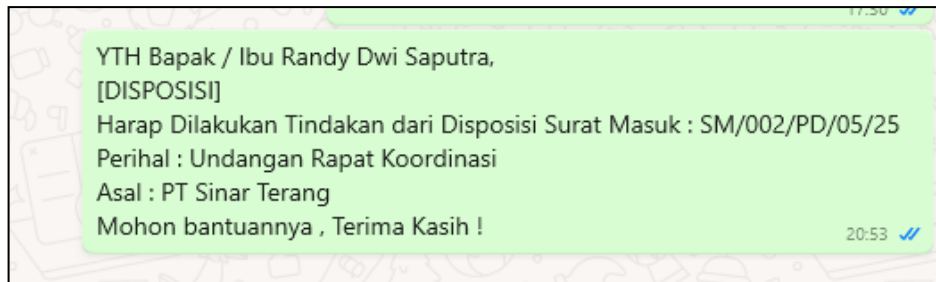
    var content = new FormUrlEncodedContent(messageData);

    try
    {
        // Mengirimkan request ke API Fonnte
        var response = await _httpClient.PostAsync(_apiUrl, content);
        return response;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Gagal mengirim pesan melalui API:
{ex.Message}");
        throw;
    }
}

```

Gambar 4. 54 Potongan Kode Kirim Notifikasi Whatsapp

Gambar 4.54 merupakan potongan kode dari sebuah sistem yang berguna untuk mengirim pesan notifikasi kepada penerima disposisi melalui *whatsapp*.



Gambar 4. 55 Hasil Notifikasi *Whatsapp*

Gambar 4.55 merupakan hasil dari implementasi kode program kirim notifikasi *whatsapp*. Otomatis terkirim kepada nomor pegawai yang dipilih di pembuatan disposisi.

4.1.3.2.1.7 Halaman Disposisi

Gambar 4.56 berisi potongan kode program halaman disposisi. Kode program tersebut berfungsi untuk menampilkan data dalam bentuk tabel, pengguna dapat dengan mudah melihat daftar disposisi dan melakukan tindakan yang diperlukan, seperti tindak lanjut disposisi.

```
public DisposisiViewModel()
{
    _disposisiRepository = new DisposisiRepository();
    int userId = MainViewModel.CurrentUserStatic?.UserID ?? 0;
    if (IsAdmin())
    {
        Items = new
ObservableCollection<DisposisiModel>(_disposisiRepository.GetAllDisposisi());
    }
    else
    {
        var disposisis =
_disposisiRepository.GetAllDisposisiByPenerimaDisposisi(userId);
        Items = new ObservableCollection<DisposisiModel>(disposisis);
    }
    FilteredItems = new ObservableCollection<DisposisiModel>(Items);
    Message = "Data Disposisi View";
}
```

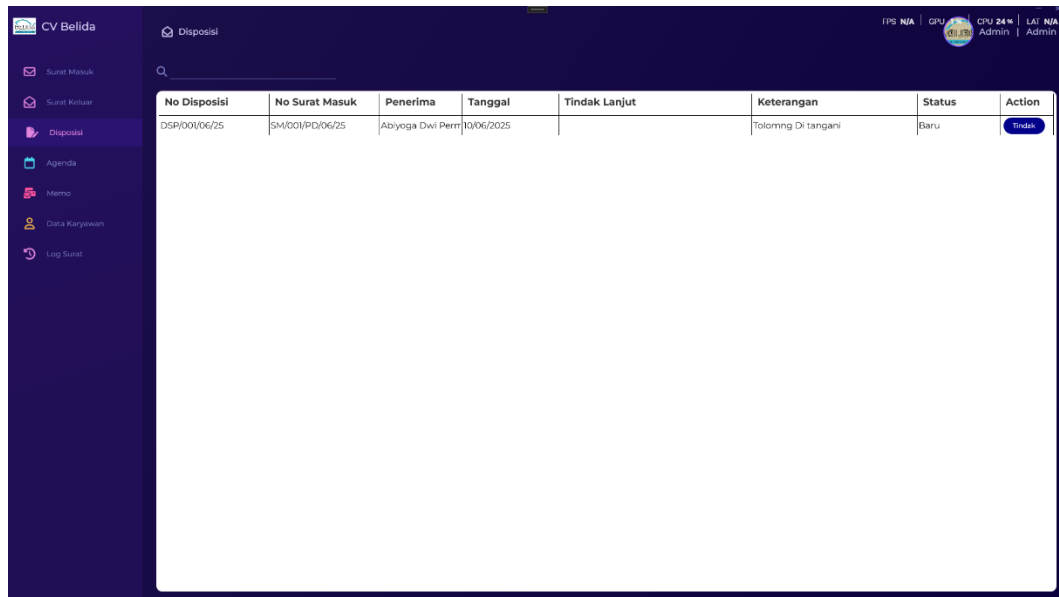


```

        ShowUpdateDisposisiCommand = new
ViewModelCommand(ExecuteUpdateDisposisi);
    }
    public void RefreshData()
    {
        int userId = MainViewModel.CurrentUserStatic?.UserID ?? 0;
        IEnumerable<DisposisiModel> disposisiFromDb;
        if (IsAdmin())
        {
            disposisiFromDb = _disposisiRepository.GetAllDisposisi();
        }
        else
        {
            disposisiFromDb =
            _disposisiRepository.GetAllDisposisiByPenerimaDisposisi(userId);
        }
        Items = new ObservableCollection<DisposisiModel>(disposisiFromDb);
        FilteredItems.Clear();
        foreach (var disposisi in Items)
        {
            FilteredItems.Add(disposisi);
        }
    }
}

```

Gambar 4. 56 Potongan Kode Sistem Halaman Disposisi



Gambar 4. 57 Halaman Disposisi

Gambar 4.57 merupakan tampilan halaman disposisi. Halaman tersebut berisi judul halaman disposisi, kolom pencarian, dan data disposisi yang ada pada *database* beserta aksi tindak lanjut.

4.1.3.2.1.8 Halaman Tindak Lanjut Disposisi

Gambar 4.58 berisi potongan kode program halaman surat masuk. Kode program tersebut berfungsi untuk mengubah data disposisi sesuai hasil disposisi. Pengguna dapat melakukan 2 opsi tindak lanjut atau tolak disposisi.

```
public UpdateDisposisiViewModel(IDisposisiRepository disposisiRepository)
{
    _disposisiRepository = disposisiRepository;
    _logSuratRepository = new LogSuratRepository();
    Disposisi = new DisposisiModel();
    var nomorWhatsAppList = new List<string>();
    var namaList = new List<string>();
    var userRepository = new UserRepository();
    IsAvailable = true;
    var penerima = userRepository.GetByAll();
    SaveCommand = new ViewModelCommand(ExecuteSave);
}
```

Gambar 4. 58 Potongan Kode Tindak Lanjut Disposisi

Tindak lanjut

No Disposisi: DSP/001/05/25

Status: Selesai

Tindak Lanjut: Selesai Di tindak lanjuti

Simpan

Gambar 4. 59 Halaman Tindak Lanjut Disposisi

Gambar 4.59 merupakan tampilan halaman tindak lanjut disposisi. Halaman tersebut kolom status, dan tindak lanjut yang akan disimpan di database.

4.1.3.2.1.9 Halaman Surat Keluar

Gambar 4.60 berisi potongan kode program halaman surat keluar. Kode program tersebut berfungsi untuk menampilkan data dalam bentuk tabel, pengguna dapat dengan mudah melihat daftar surat keluar dan melakukan tindakan yang diperlukan, seperti mengedit atau menghapus data.

```
public SuratKeluarViewModel()
{
    _suratKeluarRepository = new SuratKeluarRepository(); // Pastikan ini bisa
    diinisialisasi tanpa error
    Items = new
    ObservableCollection<SuratKeluarModel>(_suratKeluarRepository.GetAllSuratK
    eluar());
```

```

        FilteredItems = new ObservableCollection<SuratKeluarModel>(Items);
        Message = "Data Surat Masuk View";

        ShowAddSuratKeluarCommand = new
        ViewModelCommand(ExecuteShowAddSuratKeluar);
        ShowEditSuratKeluarCommand = new
        ViewModelCommand(ExecuteShowEditSuratKeluar,
        CanExecuteEditSuratKeluar);
        DeleteSuratKeluarCommand = new
        ViewModelCommand(ExecuteDeleteSuratKeluar);
    }

```

Gambar 4. 60 Potongan Kode Sistem Surat Keluar

Nomor Surat	Tanggal	Penerima	Perihal	No Surat Masuk	Keterangan	Action
SK/001/BLD/06/25	31/05/2025	PT Sukses Abadi Di Tempat	Permohonan Kerja Sama	SM/001/PD/06/25	Dengan hormat,	<button>Edit</button> <button>Hapus</button>
SK/002/PD/07/25	21/07/2025	PT Adi Karya	Permohonan Kerja Sama	SM/001/BLD/06/25	Dengan hormat,	<button>Edit</button> <button>Hapus</button>

Gambar 4. 61 Halaman Surat Keluar

Gambar 4.61 merupakan tampilan halaman surat keluar. Halaman tersebut berisi judul halaman surat keluar, tombol tambah surat, kolom pencarian, dan data surat masuk yang ada pada *database* beserta aksi *edit*, dan *hapus*.

4.1.3.2.1.10 Halaman Tambah Surat Keluar

Gambar 4.62 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data surat keluar. Kode ini memungkinkan pengguna untuk mengubah, mencetak surat masuk. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```

public AddSuratKeluarViewModel(ISuratKeluarRepository
suratKeluarRepository, SuratKeluarModel selectedSurat)
{
    var suratMasukRepository = new SuratMasukRepository();
    var suratMasukOption = suratMasukRepository.GetAllSuratMasuk();
    SuratMasukOption = new
ObservableCollection<SuratMasukModel>(suratMasukOption);
    _suratKeluarRepository = suratKeluarRepository;
    _logSuratRepository = new LogSuratRepository();
    SuratKeluar = selectedSurat ?? new SuratKeluarModel();
    SelectedFilePath = selectedSurat?.FileSurat;
    IsEditMode = true;

    // Perbaiki parsing nomor surat
    if (!string.IsNullOrEmpty(selectedSurat?.NomorSuratKeluar))
    {
        var parts = selectedSurat.NomorSuratKeluar.Split('/');
        if (parts.Length >= 4)
        {
            NomorSuratKeluar = parts[1];
            SelectedJenisSurat = JenisSuratOptions.FirstOrDefault(o => o.Value
== parts[2]);
        }
        else
        {
            NomorSuratKeluar = selectedSurat.NomorSuratKeluar;
        }
    }
    SelectedSuratMasukOption = SuratMasukOption
.FirstOrDefault(s => s.NomorSuratMasuk ==
SuratKeluar.NomorSuratMasuk);
    SaveCommand = new ViewModelCommand(ExecuteSave);
    CancelCommand = new ViewModelCommand(ExecuteCancel);
}

```



```

UploadFileCommand = new ViewModelCommand(ExecuteUploadFile);
PrintCommand = new ViewModelCommand(ExecutePrint);
OpenSelectedFileCommand = new
ViewModelCommand(OpenSelectedFile);
SuratMasukOption.Insert(0, new SuratMasukModel
{
    NomorSuratMasuk = null,
    Perihal = "-- Tidak dipilih --"
});
}

```

Gambar 4. 62 Potongan Kode Tambah Surat Keluar

Tambah Surat Keluar

Nomor Surat Keluar: 002 Umum

Tanggal Keluar: Saturday, May 31, 2025

Perihal: Permohonan Kerja Sama

Penerima: PT Sukses Abadi Di Tempat

Nomor Surat Masuk: SM/002/PD/05/25

File Surat: Ganti File Open 7c88df39-6e51-4222-8

Keterangan:

Dengan hormat,

Sehubungan dengan perkembangan usaha kami dalam bidang pengadaan barang dan jasa, bersama ini kami bermaksud mengajukan permohonan kerja sama dengan PT Sukses Abadi untuk mendukung kelancaran operasional kami dalam proyek-proyek yang sedang dan akan berjalan.

Adapun ruang lingkup kerja sama yang kami tawarkan meliputi:

1. Penyediaan material dan alat berat
2. Konsultasi teknis lapangan
3. Pengiriman logistik proyek

Simpan Print

Gambar 4. 63 Halaman Tambah Surat Keluar

Gambar 4.63 merupakan hasil dari implementasi kode program halaman tambah surat keluar. Terdapat kolom nomor surat, tanggal surat, perihal, file, keterangan, serta tombol simpan dan *print*.

4.1.3.2.1.11 Halaman Agenda

Gambar 4.64 berisi potongan kode program halaman agenda. Kode program tersebut berfungsi untuk menampilkan data dalam bentuk tabel, pengguna dapat dengan mudah melihat daftar agenda dan melakukan tindakan yang diperlukan, seperti mengedit atau menghapus data.

```
public AgendaViewModel()
{
    CurrentUserAccount = MainViewModel.CurrentUserStatic;
    _agendaRepository = new AgendaRepository();

    AgendaH = new
ObservableCollection<AgendaHModel>(_agendaRepository.GetAllAgendaByUse
rname(MainViewModel.CurrentUserStatic.Username));
    FilteredAgendaH = new
ObservableCollection<AgendaHModel>(AgendaH); // Initialize with all agendas
    Message = "Agenda View";
    CurrentUserStatic = CurrentUserAccount;
    CurrentUserStatic = MainViewModel.CurrentUserStatic;
    NomorWA = MainViewModel.CurrentUserStatic.Name;

    CurrentDate = DateTime.Now;
    SelectedDate = CurrentDate; // Set default SelectedDate to CurrentDate

    ShowAddAgendaCommand = new
ViewModelCommand(ExecuteShowAddAgenda);
    EditAgendaCommand = new ViewModelCommand(ExecuteEditAgenda);
    DeleteAgendaCommand = new
ViewModelCommand(ExecuteDeleteAgenda);
```

```

        NextMonthCommand = new ViewModelCommand((obj) =>
NextMonth());

        PreviousMonthCommand = new ViewModelCommand((obj) =>
PreviousMonth());

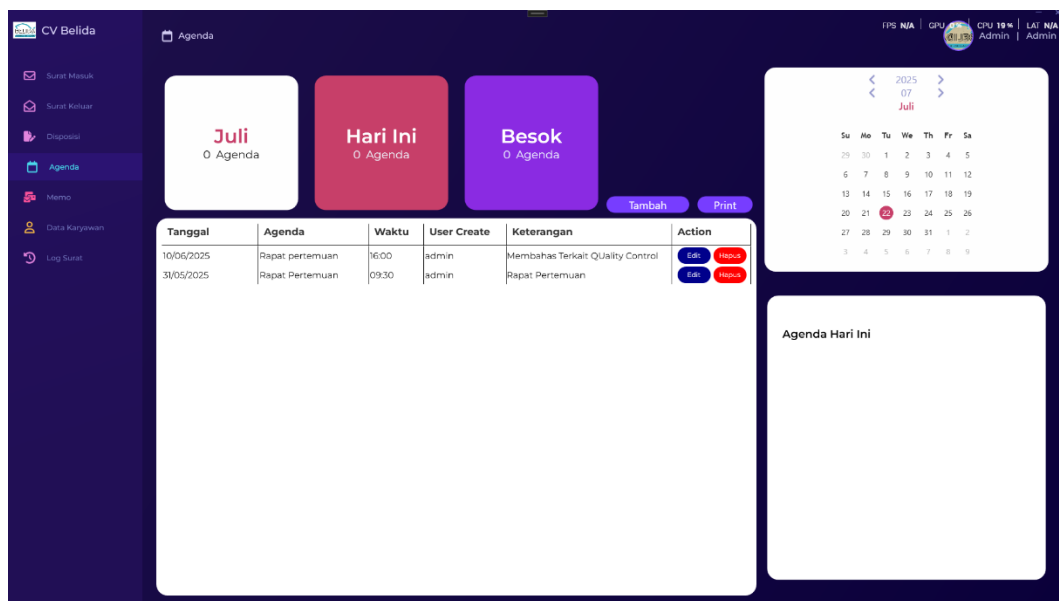
        NextDayCommand = new ViewModelCommand((obj) => NextDay());
        PreviousDayCommand = new ViewModelCommand((obj) =>
PreviousDay());

        NextYearCommand = new ViewModelCommand((obj) => NextYear());
        PreviousYearCommand = new ViewModelCommand((obj) =>
PreviousYear());

    }

```

Gambar 4. 64 Potongan Kode Sistem Halaman Agenda



Gambar 4. 65 Halaman Agenda

Gambar 4.65 merupakan tampilan halaman agenda. Halaman tersebut berisi judul halaman agenda, tombol tambah agenda, kalender pencarian agenda, dan data agenda yang ada pada *database* beserta aksi *edit*, dan print.

4.1.3.2.1.12 Halaman Tambah Agenda

Gambar 4.66 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data agenda. Kode ini memungkinkan pengguna untuk mengubah data agenda, seperti agenda, tanggal, waktu, keterangan, serta anggota

agenda. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public AddAgendaWindowViewModel()
{
    _agendaRepository = new AgendaRepository();
    _logRepository = new LogSuratRepository();
    _anggotaAgendaRepository = new AnggotaAgendaRepository();
    AnggotaAgenda = new ObservableCollection<AnggotaAgendaModel>();
    SaveCommand = new ViewModelCommand(ExecuteSave);
    TglAgenda = DateTime.Today;
    ShowPopupCommand = new ViewModelCommand(ShowPopup);
    ShowNomorWACommand = new ViewModelCommand(ShowNomorWA);
    LoadAnggotaAgenda();
}
private void LoadAnggotaAgenda()
{
    var anggotaAgendaList =
        _anggotaAgendaRepository.GetAnggotaAgendaById(AgendaID);
    AnggotaAgenda.Clear();
    foreach (var anggota in anggotaAgendaList)
    {
        AnggotaAgenda.Add(anggota);
    }
}
```

Gambar 4. 66 Potongan Kode Tambah Agenda

Gambar 4. 67 Halaman Tambah Agenda

Gambar 4.67 merupakan hasil dari implementasi kode program halaman tambah agenda. Terdapat kolom tanggal, agenda, waktu, anggota agenda, keterangan, serta tombol simpan dan *send whatsapp*.

4.1.3.2.1.13 Halaman Tambah Anggota Agenda

Gambar 4.68 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data anggota agenda. Kode ini memungkinkan pengguna untuk mengubah data anggota agenda dari data *users*. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public PopupWindowViewModel(IAnggotaAgendaRepository
anggotaAgendaRepository, int agendaId)
{
    _anggotaAgendaRepository = anggotaAgendaRepository;
    AnggotaAgenda = new ObservableCollection<AnggotaAgendaModel>();
    AgendaID = agendaId;
    LoadAnggotaAgenda();
}
```



```

foreach (var anggota in AnggotaAgenda)
{
    OnPropertyChanged(nameof(AnggotaAgenda)); // Force notify for
debugging
}
SimpanCommand = new ViewModelCommand(SimpanAnggotaAgenda);
}

```

Gambar 4. 68 Potongan Kode Tambah Anggota Agenda

The screenshot shows a window titled 'Anggota' with a close button in the top right. Inside, there's a white rounded rectangle containing a table titled 'List Anggota'. The table has three columns: 'Value', 'Anggota', and 'Terkirim'. There are six rows of data. The second row is selected, indicated by a checkmark in the 'Value' column. Below the table is a blue button labeled 'Simpan'.

Value	Anggota	Terkirim
☐	Admin	0
☒	Abiyoga (Penanggung	2
☐	Abiyoga Iphone (Per	0
☐	Hilman Habib (Ketu	0
☐	Adam (Ketua Divisi	0
☐	Logistik	0

Gambar 4. 69 Halaman Tambah Agenda

Gambar 4.69 merupakan hasil dari implementasi kode program halaman tambah anggota agenda. Terdapat kolom *value*, anggota, terkirim, dan tombol simpan.

4.1.3.2.1.14 Halaman Memo

Gambar 4.70 berisi potongan kode program halaman memo. Kode program tersebut berfungsi untuk menampilkan data dalam bentuk tabel, pengguna dapat

dengan mudah melihat daftar memo dan melakukan tindakan yang diperlukan, seperti mengedit atau menghapus data.

```
public MemoViewModel()
{
    _memoRepository = new MemoRepository();

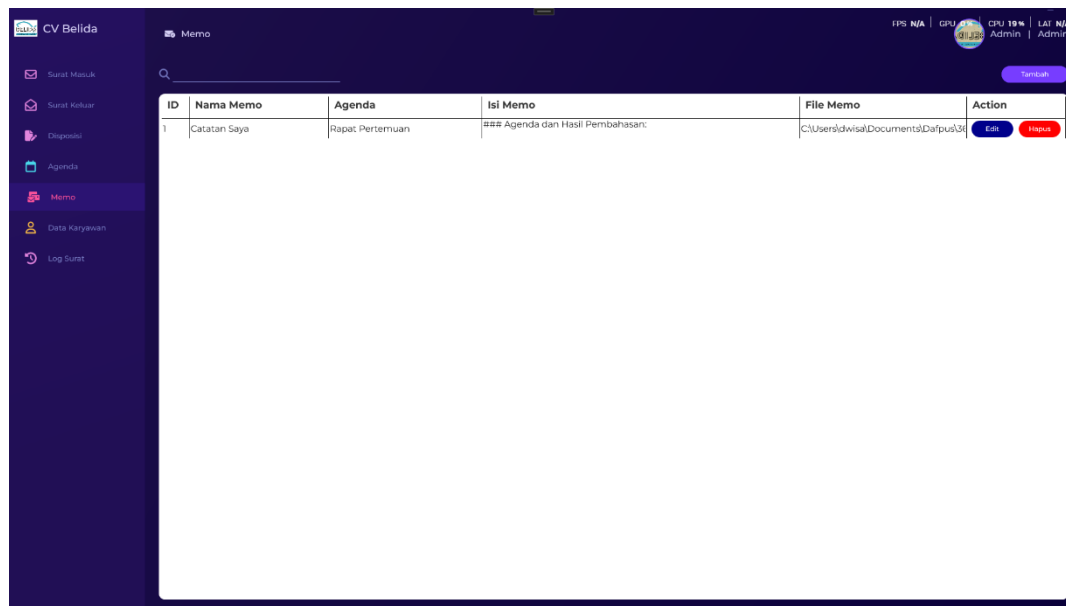
    int userId = MainViewModel.CurrentUserStatic?.UserID ?? 0;

    var memos = _memoRepository.GetAllMemoCreate(userId);
    Items = new ObservableCollection<MemoModel>(memos);

    FilteredItems = new ObservableCollection<MemoModel>(Items);
    Message = "Data Memo View";

    ShowAddMemoCommand = new
    ViewModelCommand(ExecuteShowTambahMemo);
    ShowEditMemoCommand = new
    ViewModelCommand(ExecuteShowEditMemo); // Tambah command edit
    DeleteMemoCommand = new
    ViewModelCommand(ExecuteDeleteMemo);
}
```

Gambar 4. 70 Potongan Kode Sistem Halaman Memo



Gambar 4. 71 Halaman Memo

Gambar 4.71 merupakan tampilan halaman memo. Halaman tersebut berisi judul halaman memo, tombol tambah memo, kolom pencarian, dan data surat masuk yang ada pada *database* beserta aksi *edit*, dan hapus.

4.1.3.2.1.15 Halaman Tambah Memo

Gambar 4.72 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data memo. Kode ini memungkinkan pengguna untuk mengubah data memo, seperti nama memo, agenda, file memo, keterangan. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public AddMemoWindowViewModel()
{
    _memoRepository = new MemoRepository();
    _agendaRepository = new AgendaRepository();
    AgendaList = new
ObservableCollection<AgendaHModel>(_agendaRepository.GetAllAgenda());
    SaveMemoCommand = new ViewModelCommand(SimpanMemo,
CanExecuteSimpan);
    UploadFileCommand = new ViewModelCommand(UploadFile);
}
```

Gambar 4. 72 Potongan Kode Tambah Memo

The screenshot shows a 'Tambah Memo' window with a dark blue background. It features a sidebar on the left with labels for 'Memo', 'Agenda', 'File Memo', and 'Keterangan'. The 'Memo' field contains 'Catatan Saya'. The 'Agenda' field has a calendar icon and a dropdown menu showing 'Rapat Pertemuan'. The 'File Memo' field has a green 'Open' button and the file name '363-1372-2-PB.pdf'. The 'Keterangan' field contains a text area with the following text: '### Agenda dan Hasil Pembahasan: 1. **Evaluasi Kinerja Divisi** - Divisi Operasional melaporkan keterlambatan pengadaan bahan pada minggu ke-2 Mei. - Solusi: Koordinasi lebih awal dengan vendor dilakukan mulai Juni. 2. **Kendala dan Solusi** - Divisi Keuangan mengalami kendala dalam pencairan dana proyek X. - Disarankan percepatan proses approval internal. 3. **Rencana Kerja Bulan Juni** - Masing-masing divisi diminta...'. A blue 'Simpan' button is at the bottom.

Gambar 4. 73 Halaman Tambah Memo

Gambar 4.73 merupakan hasil dari implementasi kode program halaman tambah memo. Terdapat kolom memo, agenda, file memo, keterangan, serta tombol simpan.

4.1.3.2.1.16 Halaman Log

Gambar 4.74 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data log semua aksi yang dilakukan di aplikasi. Kode ini memungkinkan melihat data log, seperti jenis aksi, nama tabel dan download *file* log dalam bentuk excel.

```
public LogSuratViewModel()
{
    _logRepository = new LogSuratRepository();

    // Ambil semua data log dan simpan di koleksi master
    _logs = new
ObservableCollection<LogSuratModel>(_logRepository.GetAllLogSurat());

    // Awalnya, tampilkan semua data log
    FilteredLogs = new ObservableCollection<LogSuratModel>(_logs);
    ExportToExcelCommand = new
ViewModelCommand(ExecuteExportToExcel);
}
```

Gambar 4. 74 Potongan Kode Tambah Memo

Tanggal dan Waktu	Aksi	Nomer Berkait	Pengguna
22-07-2025 23:04:00	Update Surat Keluar	SK/002/PO/07/25	admin
22-07-2025 23:03:46	Update Surat Keluar	SK/002/PO/07/25	admin
21-07-2025 18:47:26	Edit Agenda Rapat Pertemuan	Nama Agenda = Rapat Pertemuan31 Mei 2025	admin
21-07-2025 18:31:44	Edit Agenda Rapat Pertemuan	Nama Agenda = Rapat Pertemuan31 Mei 2025	admin
21-07-2025 17:51:29	Edit Agenda Rapat Pertemuan	Nama Agenda = Rapat Pertemuan31 Mei 2025	admin
21-07-2025 17:47:23	Edit Agenda Rapat pertemuan	Nama Agenda = Rapat pertemuan10 Juni 2025	admin
21-07-2025 17:46:55	Edit Agenda Rapat pertemuan	Nama Agenda = Rapat pertemuan10 Juni 2025	admin
21-07-2025 17:46:46	Edit Agenda Rapat Pertemuan	Nama Agenda = Rapat Pertemuan31 Mei 2025	admin
21-07-2025 17:46:18	Tambah Disposisi	DS/007/07/25	admin
21-07-2025 17:46:18	Update Status Surat Masuk	SM/001/PO/06/25	admin
21-07-2025 17:45:08	Tambah Disposisi	DS/006/07/25	admin
21-07-2025 17:45:08	Update Status Surat Masuk	SM/001/PO/06/25	admin
21-07-2025 17:41:31	Tambah Disposisi	DS/005/07/25	admin
21-07-2025 17:41:31	Update Status Surat Masuk	SM/001/PO/06/25	admin
21-07-2025 17:36:49	Tambah Disposisi	DS/004/07/25	admin
21-07-2025 17:36:48	Update Status Surat Masuk	SM/001/PO/06/25	admin
21-07-2025 17:34:47	Tambah Disposisi	DS/003/07/25	admin
21-07-2025 17:34:47	Update Status Surat Masuk	SM/001/PO/06/25	admin
21-07-2025 17:31:42	Tambah Disposisi	DS/002/07/25	admin
21-07-2025 17:31:41	Update Status Surat Masuk	SM/001/PO/06/25	admin
21-07-2025 17:28:11	Tambah Disposisi	DS/014/07/25	admin
21-07-2025 17:28:11	Update Status Surat Masuk	SM/001/PO/06/25	admin
21-07-2025 17:23:47	Tambah Disposisi	DS/013/07/25	admin
21-07-2025 17:23:47	Update Status Surat Masuk	SM/001/PO/06/25	admin

Gambar 4. 75 Halaman Tambah Memo

Gambar 4.75 merupakan hasil dari implementasi kode program halaman tambah memo. Terdapat kolom memo, agenda, file memo, keterangan, serta tombol simpan.

4.1.3.2.2 Sistem *User Karyawan*

User karyawan sendiri hanya memiliki hak akses fitur sistem, yang berfungsi sebagai penerima data, pengolah data agenda dan memo di sistem. Hak akses karyawan sendiri akan diberikan kepala divisi, serta pegawai yang diberikan tugas.

4.1.3.2.2.1 Halaman Login

Gambar 4.76 ini merupakan potongan kode program yang berfungsi untuk proses login memeriksa kecocokan *username* dan *password users* dengan yang ada dalam *database*.

```
public LoginViewModel()
{
    userRepository = new UserRepository();
    LoginCommand = new ViewModelCommand(ExecuteLoginCommand,
CanExecuteLoginCommand);

    RecoverPasswordCommand = new ViewModelCommand(p =>
ExecuteRecoverPassCommand("", ""));
}

private bool CanExecuteLoginCommand(object obj)
```

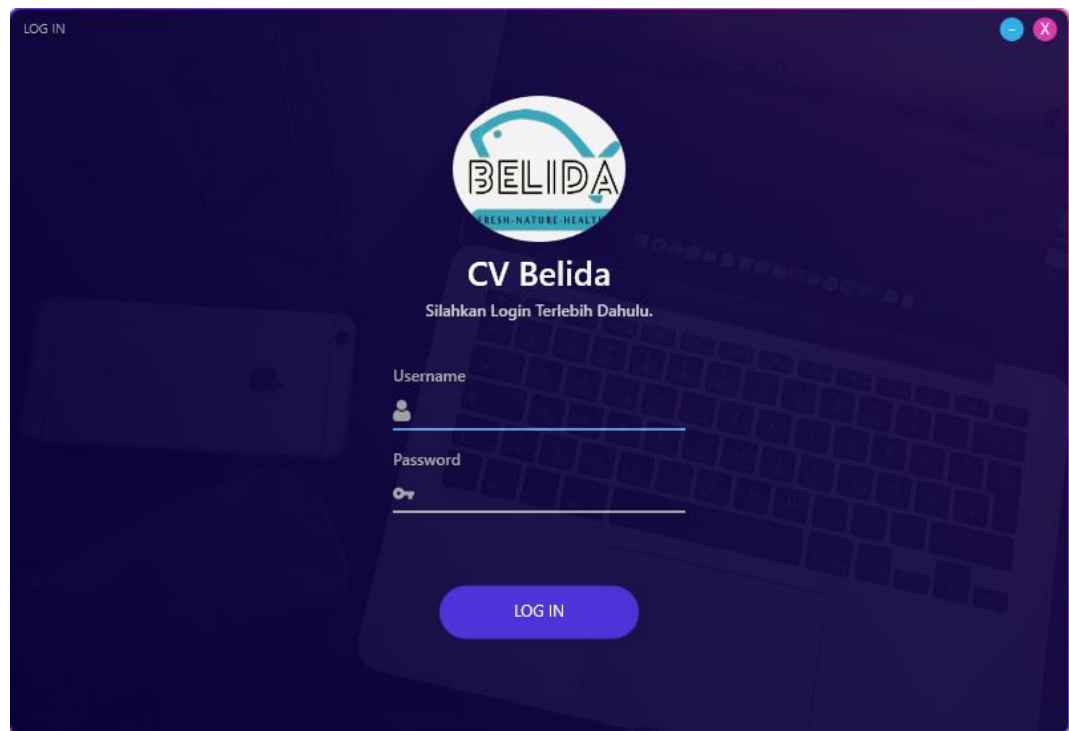
```

{
    bool validData;
    if (string.IsNullOrEmpty(Username) || Username.Length < 3 ||
        Password == null || Password.Length < 3)
        validData = false;
    else
        validData = true;
    return validData;
}

private void ExecuteLoginCommand(object obj)
{
    var isValidUser = userRepository.AuthenticateUser(new
NetworkCredential(Username, Password));
    if (isValidUser)
    {
        Thread.CurrentPrincipal = new GenericPrincipal(
            new GenericIdentity(Username), null);
        IsViewVisible = false;
    }
    else
    {
        ErrorMessage = "* Invalid username or password";
    }
}
}

```

Gambar 4. 76 Potongan Kode Sistem Login



Gambar 4. 77 Halaman Login

Gambar 4.77 merupakan tampilan dari halaman form login untuk users. Terdapat kolom pengisian username dan *password*, serta tombol login.

4.1.3.2.1.2 Halaman Data Karyawan

Gambar 4.78 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data profilnya. Kode ini memungkinkan pengguna untuk mengubah data profilnya, seperti nama lengkap dan *role*. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public DataKaryawanViewModel()
{
    _userRepository = new UserRepository();

    string userRole = MainViewModel.CurrentUserStatic?.Role;
    int userId = MainViewModel.CurrentUserStatic?.UserID ?? 0;

    if (IsAdmin())
    {
        var users = _userRepository.GetByAll();
        foreach (var user in users)
        {
            if (!string.IsNullOrEmpty(user.ProfilePicture))
            {
                user.ProfilePicture =
System.IO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Images", user.ProfilePicture);
            }
        }
        Items = new ObservableCollection<UserModel>(users);
    }
    else
    {
        var user = _userRepository.GetById(userId);
        Items = new ObservableCollection<UserModel>();

        if (user != null)
        {
```

```

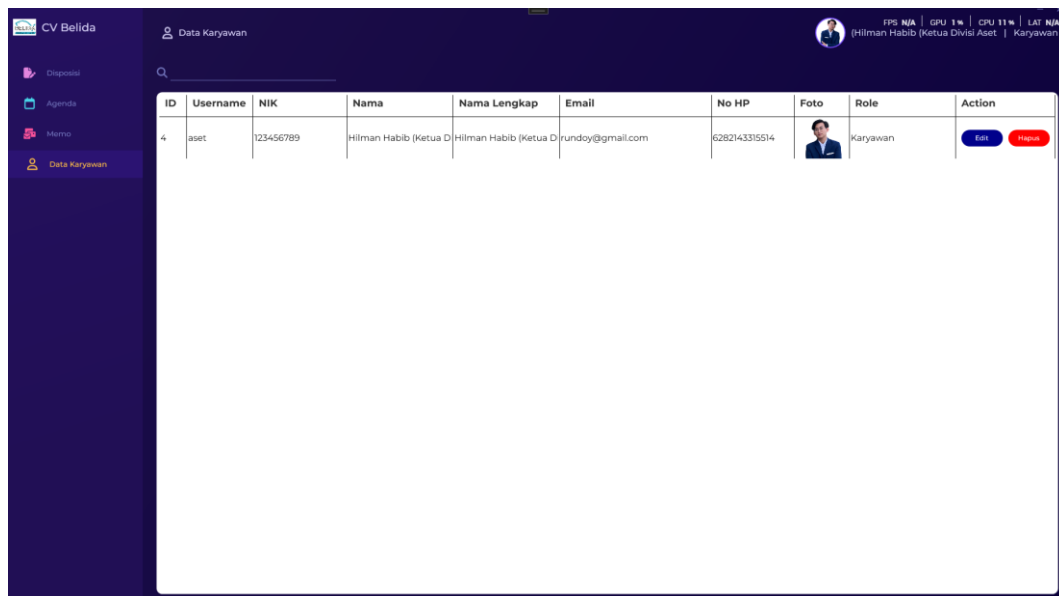
        if (!string.IsNullOrEmpty(user.ProfilePicture))
        {
            user.ProfilePicture =
System.IO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Images", user.ProfilePicture);
        }
        Items.Add(user);
    }
}

FilteredItems = new ObservableCollection<UserModel>(Items);
Message = "Data Karyawan View";

ShowTambahKaryawanCommand =
new
ViewModelCommand(ExecuteShowTambahKaryawan);
ShowPrintCommand = new ViewModelCommand(ExecuteShowPrint);
DeleteUserCommand = new ViewModelCommand(ExecuteDeleteUser);
EditUserCommand = new ViewModelCommand(ExecuteEditUser);
}

```

Gambar 4. 78 Potongan Kode Sistem Data Karyawan



Gambar 4. 79 Halaman Tambah Karyawan

Gambar 4.79 merupakan hasil dari implementasi kode program halaman edit profile. Terdapat kolom username, nama lengkap, NIK, nama, nama lengkap, email, nomor hp, foto profil, role, ttd dan tombol simpan *profile*.

4.1.3.2.1.3 Halaman Ubah Data Karyawan

Gambar 4.80 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data profilnya. Kode ini memungkinkan pengguna untuk mengubah data profilnya, seperti nama lengkap dan *role*. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public TambahKaryawanViewModel()
{
    var userRepository = new UserRepository();
    var roles = userRepository.GetAllRoles();
    Roles = new ObservableCollection<RoleModel>(roles);
    SelectedRole = Roles.FirstOrDefault();
    ShowProfileCommand = new
ViewModelCommand(ExecuteShowProfile);
    AddUserCommand = new ViewModelCommand(ExecuteAddUser);
    ButtonContent = "Tambah";
    if (!Directory.Exists(_imageDirectory))
        Directory.CreateDirectory(_imageDirectory);
}
```

Gambar 4. 80 Potongan Kode Sistem Tambah Data Karyawan

Tambah Karyawan

Username aset

Password

NIK 123456789

Name Hilman Habib (Ketua Divisi Aset)

FullName Hilman Habib (Ketua Divisi Aset)

Email runday@gmail.com

No HP 6282143315514

Profile Upload D:\C# Project\Login-In-WPF-MVVM-C-Sharp-and-SQL-Server-main\WPF-LoginForm

Tambah

Gambar 4. 81 Halaman Ubah Data Karyawan

Gambar 4.81 merupakan hasil dari implementasi kode program halaman edit profile. Terdapat kolom username, nama lengkap, NIK, nama, nama lengkap, email, nomor hp, foto profil, role, ttd dan tombol simpan *profile*.

4.1.3.2.1.7 Halaman Disposisi

Gambar 4.82 berisi potongan kode program halaman disposisi. Kode program tersebut berfungsi untuk menampilkan data dalam bentuk tabel, pengguna dapat dengan mudah melihat daftar disposisi dan melakukan tindakan yang diperlukan, seperti tindak lanjut disposisi.

```
public DisposisiViewModel()
{
    _disposisiRepository = new DisposisiRepository();

    int userId = MainViewModel.CurrentUserStatic?.UserID ?? 0;

    if (IsAdmin())
    {
```

```

        Items = new
ObservableCollection<DisposisiModel>(_disposisiRepository.GetAllDisposisi());
    }
    else
    {
        var disposisis =
_disposisiRepository.GetAllDisposisiByPenerimaDisposisi(userId);
        Items = new ObservableCollection<DisposisiModel>(disposisis);
    }

    FilteredItems = new ObservableCollection<DisposisiModel>(Items);
    Message = "Data Disposisi View";

    ShowUpdateDisposisiCommand = new
ViewModelCommand(ExecuteUpdateDisposisi);
}

public void RefreshData()
{
    int userId = MainViewModel.CurrentUserStatic?.UserID ?? 0;

    IEnumerable<DisposisiModel> disposisiFromDb;

    if (IsAdmin())
    {
        disposisiFromDb = _disposisiRepository.GetAllDisposisi();
    }
    else
    {
        disposisiFromDb =
_disposisiRepository.GetAllDisposisiByPenerimaDisposisi(userId);
    }
    Items = new ObservableCollection<DisposisiModel>(disposisiFromDb);

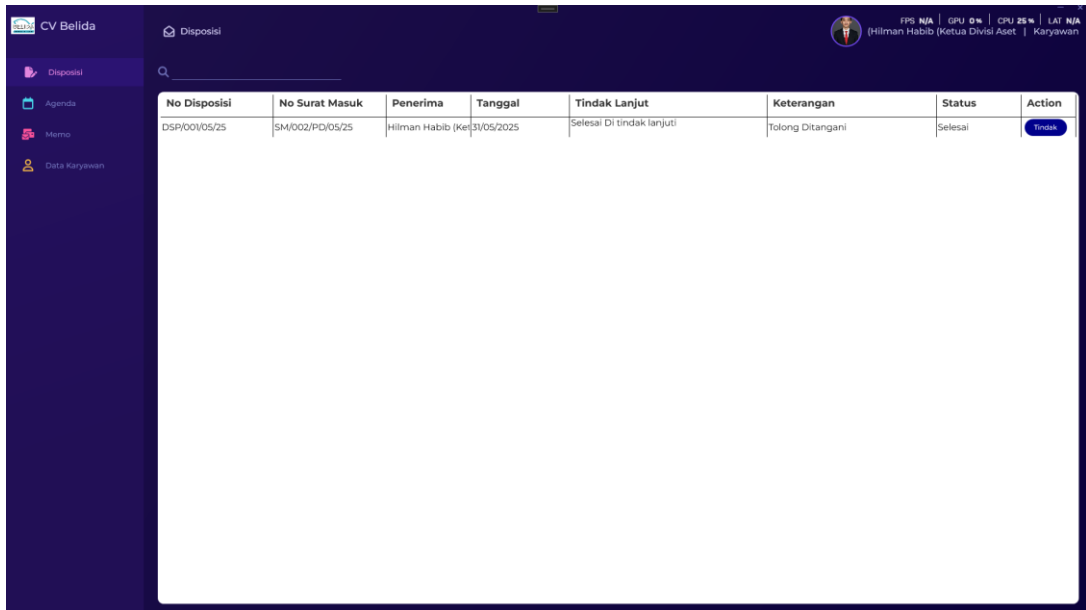
```

```

FilteredItems.Clear();
foreach (var disposisi in Items)
{
    FilteredItems.Add(disposisi);
}
}

```

Gambar 4. 82 Potongan Kode Sistem Halaman Disposisi



No Disposisi	No Surat Masuk	Penerima	Tanggal	Tindak Lanjut	Keterangan	Status	Action
DSP/001/05/25	SM/002/PD/05/25	Hilman Habib (Ket)	31/05/2025	Selesai Di tindak lanjut	Tolong Ditangani	Selesai	Tindak

Gambar 4. 83 Halaman Disposisi

Gambar 4.83 merupakan tampilan halaman disposisi. Halaman tersebut berisi judul halaman disposisi, kolom pencarian, dan data disposisi yang ada pada *database* beserta aksi tindak lanjut.

4.1.3.2.1.8 Halaman Tindak Lanjut Disposisi

Gambar 4.84 berisi potongan kode program halaman surat masuk. Kode program tersebut berfungsi untuk mengubah data disposisi sesuai hasil disposisi. Pengguna dapat melakukan 2 opsi tindak lanjut atau tolak disposisi.

```

public UpdateDisposisiViewModel(IDisposisiRepository disposisiRepository)
{
    _disposisiRepository = disposisiRepository;
    _logSuratRepository = new LogSuratRepository();
    Disposisi = new DisposisiModel();
}

```

```

var nomorWhatsAppList = new List<string>();
var namaList = new List<string>();
var userRepository = new UserRepository();
IsAvaliable = true;
var penerima = userRepository.GetByAll();
SaveCommand = new ViewModelCommand(ExecuteSave);
}

```

Gambar 4. 84 Potongan Kode Tindak Lanjut Disposisi

The screenshot shows a mobile application form titled "Tindak lanjut". The form has a light blue header with the title and a close button. Below the header, there are three input fields: "No Disposisi" with the value "DSP/001/05/25", "Status" with a dropdown menu showing "Selesai", and "Tindak Lanjut" with a text area containing "Selesai Di tindak lanjuti". At the bottom of the form is a blue "Simpan" (Save) button.

Gambar 4. 85 Halaman Tindak Lanjut Disposisi

Gambar 4.85 merupakan tampilan halaman tindak lanjut disposisi. Halaman tersebut kolom status, dan tindak lanjut yang akan disimpan di database.

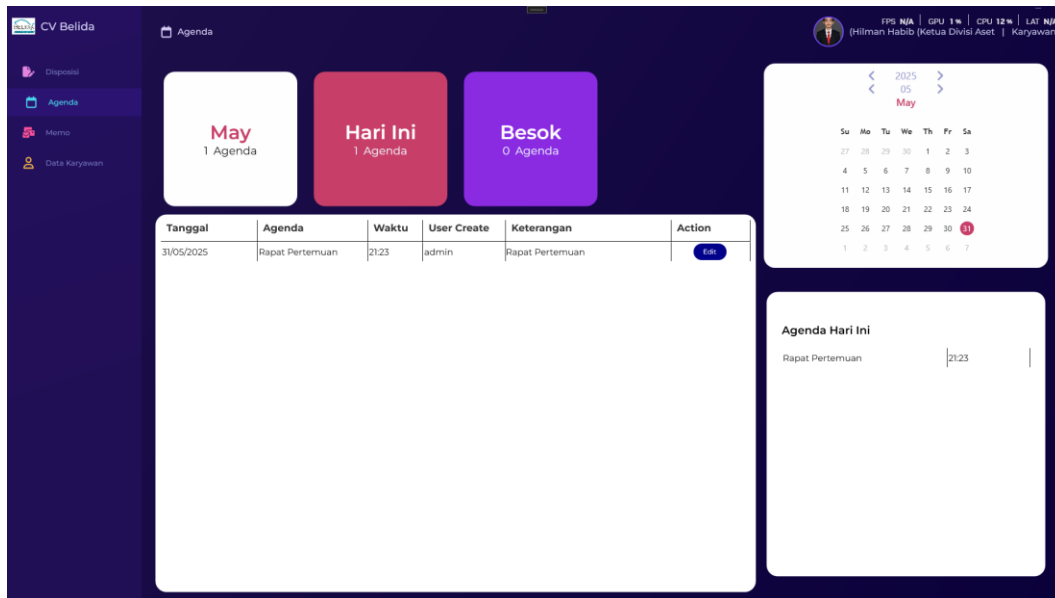
4.1.3.2.1.11 Halaman Agenda

Gambar 4.86 berisi potongan kode program halaman agenda. Kode program tersebut berfungsi untuk menampilkan data dalam bentuk tabel, pengguna dapat dengan mudah melihat daftar agenda.

```
public AgendaViewModel()
{
    CurrentUserAccount = MainViewModel.CurrentUserStatic;
    _agendaRepository = new AgendaRepository();
    AgendaH = new
ObservableCollection<AgendaHModel>(_agendaRepository.GetAllAgendaByUse
rname(MainViewModel.CurrentUserStatic.Username));
    FilteredAgendaH = new
ObservableCollection<AgendaHModel>(AgendaH); // Initialize with all agendas
    Message = "Agenda View";
    CurrentUserStatic = CurrentUserAccount;
    CurrentUserStatic = MainViewModel.CurrentUserStatic;
    NomorWA = MainViewModel.CurrentUserStatic.Name;
    CurrentDate = DateTime.Now;
    SelectedDate = CurrentDate; // Set default SelectedDate to CurrentDate
    ShowAddAgendaCommand = new
ViewModelCommand(ExecuteShowAddAgenda);
    EditAgendaCommand = new ViewModelCommand(ExecuteEditAgenda);
    DeleteAgendaCommand = new
ViewModelCommand(ExecuteDeleteAgenda);
    NextMonthCommand = new ViewModelCommand((obj) =>
NextMonth());
    PreviousMonthCommand = new ViewModelCommand((obj) =>
PreviousMonth());
    NextDayCommand = new ViewModelCommand((obj) => NextDay());
    PreviousDayCommand = new ViewModelCommand((obj) =>
PreviousDay());
    NextYearCommand = new ViewModelCommand((obj) => NextYear());
```

```
PreviousYearCommand = new ViewModelCommand((obj) =>
PreviousYear());}
```

Gambar 4. 86 Potongan Kode Sistem Halaman Agenda



Gambar 4. 87 Halaman Agenda

Gambar 4.7 merupakan tampilan halaman agenda. Halaman tersebut berisi judul halaman agenda, tombol tambah agenda, kalender pencarian agenda, dan data agenda yang ada pada *database* beserta aksi *edit*, dan print.

4.1.3.2.1.12 Halaman Lihat Agenda

Gambar 4.88 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data agenda. Kode ini memungkinkan pengguna untuk melihat data agenda, seperti nama agenda, tanggal, waktu, keterangan, serta anggota agenda.

```
public AddAgendaWindowViewModel()
{
    _agendaRepository = new AgendaRepository();
    _logRepository = new LogSuratRepository();
    _anggotaAgendaRepository = new AnggotaAgendaRepository();
    AnggotaAgenda = new ObservableCollection<AnggotaAgendaModel>();
    SaveCommand = new ViewModelCommand(ExecuteSave);
}
```



```

TglAgenda = DateTime.Today;
ShowPopupCommand = new ViewModelCommand(ShowPopup);
ShowNomorWACommand = new ViewModelCommand(ShowNomorWA);
LoadAnggotaAgenda();
}
private void LoadAnggotaAgenda()
{
    var anggotaAgendaList =
        _anggotaAgendaRepository.GetAnggotaAgendaById(AgendaID);
    AnggotaAgenda.Clear();

    foreach (var anggota in anggotaAgendaList)
    {
        AnggotaAgenda.Add(anggota);
    }
}

```

Gambar 4. 88 Potongan Kode Lihat Agenda

Tambah Agenda

Tanggal: Saturday, May 31, 2025

Agenda: Rapat Pertemuan

Waktu: 21:23

Anggota: Show Send WA

Keterangan: Rapat Pertemuan

Gambar 4. 89 Halaman Lihat Agenda

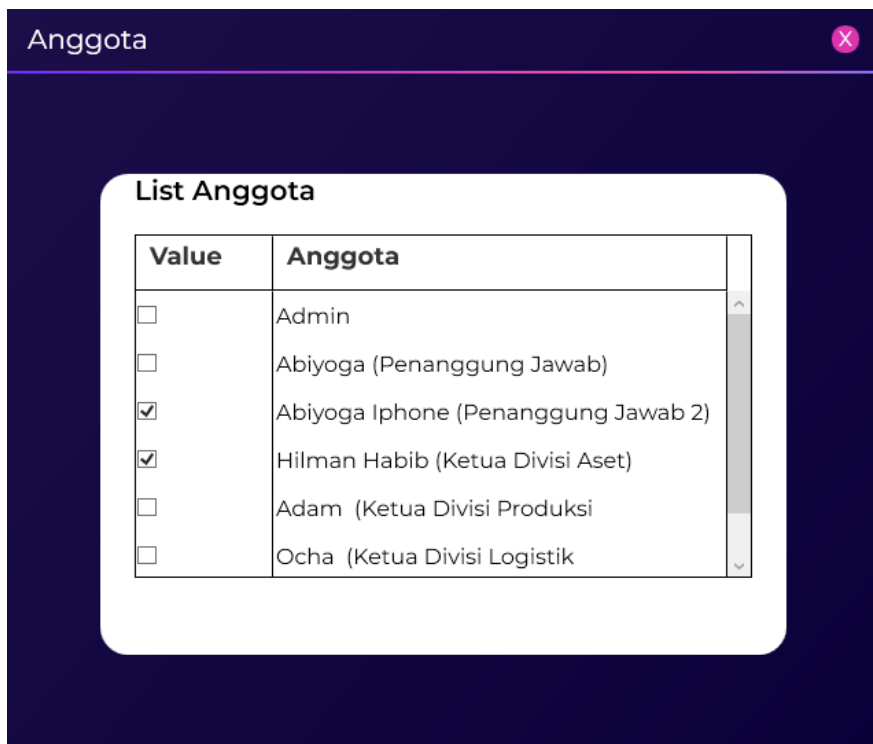
Gambar 4.89 merupakan hasil dari implementasi kode program halaman lihat agenda. Terdapat kolom tanggal, agenda, waktu, anggota agenda, keterangan, dan *send whatsapp*.

4.1.3.2.1.13 Halaman Lihat Anggota Agenda

Gambar 4.90 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna melihat data anggota agenda. Kode ini memungkinkan pengguna untuk.

```
public PopupWindowViewModel(IAnggotaAgendaRepository
anggotaAgendaRepository, int agendaId)
{
    _anggotaAgendaRepository = anggotaAgendaRepository;
    AnggotaAgenda = new ObservableCollection<AnggotaAgendaModel>();
    AgendaID = agendaId;
    LoadAnggotaAgenda();
    foreach (var anggota in AnggotaAgenda)
    {
        // Force notify when Value is updated (debugging purposes)
        OnPropertyChanged(nameof(AnggotaAgenda)); // Force notify for
debugging
    }
    SimpanCommand = new ViewModelCommand(SimpanAnggotaAgenda);
}
```

Gambar 4. 90 Potongan Kode Lihat Anggota Agenda



Gambar 4. 91 Halaman Lihat Anggota Agenda

Gambar 4.90 merupakan hasil dari implementasi kode program halaman lihat anggota agenda. Terdapat kolom *action*, nama *users*.

4.1.3.2.1.14 Halaman Memo

Gambar 4.92 berisi potongan kode program halaman memo. Kode program tersebut berfungsi untuk menampilkan data dalam bentuk tabel, pengguna dapat dengan mudah melihat daftar memo dan melakukan tindakan yang diperlukan, seperti mengedit atau menghapus data.

```
public MemoViewModel()
{
    _memoRepository = new MemoRepository();

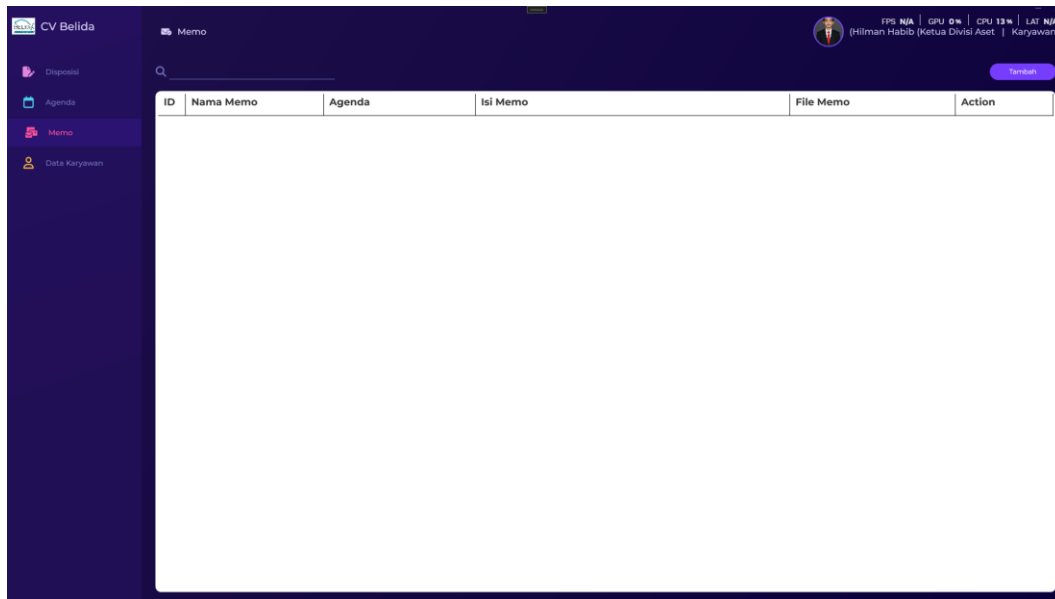
    int userId = MainViewModel.CurrentUserStatic?.UserID ?? 0;

    var memos = _memoRepository.GetAllMemoCreate(userId);
    Items = new ObservableCollection<MemoModel>(memos);

    FilteredItems = new ObservableCollection<MemoModel>(Items);
    Message = "Data Memo View";

    ShowAddMemoCommand = new
    ViewModelCommand(ExecuteShowTambahMemo);
    ShowEditMemoCommand = new
    ViewModelCommand(ExecuteShowEditMemo); // Tambah command edit
    DeleteMemoCommand = new
    ViewModelCommand(ExecuteDeleteMemo);
}
```

Gambar 4. 92 Potongan Kode Sistem Halaman Memo



Gambar 4. 93 Halaman Memo

Gambar 4.93 merupakan tampilan halaman memo. Halaman tersebut berisi judul halaman memo, tombol tambah memo, kolom pencarian, dan data surat masuk yang ada pada *database* beserta aksi *edit*, dan hapus.

4.1.3.2.1.15 Halaman Tambah Memo

Gambar 4.94 merupakan potongan kode dari sebuah sistem yang berguna untuk pengguna menyimpan data memo. Kode ini memungkinkan pengguna untuk mengubah data memo, seperti nama memo, agenda, file memo, keterangan. Proses ini melibatkan pengambilan data dari formulir, validasi data, dan pembaruan data di dalam *database*.

```
public AddMemoWindowViewModel()
{
    _memoRepository = new MemoRepository();
    _agendaRepository = new AgendaRepository();
    AgendaList = new
ObservableCollection<AgendaHModel>(_agendaRepository.GetAllAgenda());
    SaveMemoCommand = new ViewModelCommand(SimpanMemo,
CanExecuteSimpan);
    UploadFileCommand = new ViewModelCommand(UploadFile);
}
```

Gambar 4. 94 Potongan Kode Tambah Memo

Tambah Memo

Memo: Catatan Saya

Agenda: Rapat Pertemuan

File Memo: Open 363-1372-2-PB.pdf

Keterangan:

Agenda dan Hasil Pembahasan:

- 1. Evaluasi Kinerja Divisi**
 - Divisi Operasional melaporkan keterlambatan pengadaan bahan pada minggu ke-2 Mei.
 - Solusi: Koordinasi lebih awal dengan vendor dilakukan mulai Juni.
- 2. Kendala dan Solusi**
 - Divisi Keuangan mengalami kendala dalam pencairan dana proyek X.
 - Disarankan percepatan proses approval internal.
- 3. Rencana Kerja Bulan Juni**
 - Masing-masing divisi diminta...

Simpan

Gambar 4. 95 Halaman Tambah Memo

Gambar 4.95 merupakan hasil dari implementasi kode program halaman tambah memo. Terdapat kolom memo, agenda, file memo, keterangan, serta tombol simpan.

4.1.4 Cutover

Tahap keempat yang bisa disebut tahap *Cutover* merupakan tahap akhir dari pembuatan sistem. Ditahapan inilah yang secara nyata menunjukkan bahwa sistem siap digunakan oleh pengguna setelah semua fitur utama selesai. Proses ini akan melibatkan pengujian akhir, pelatihan pengguna, dan implementasi.

Pada sistem ini diperlukan metode pengujian untuk dilakukan *testing*. Ada dua pendekatan metode dalam tahapan testing yaitu metode black box testing dan white box testing. Pada skripsi ini metode yang digunakan adalah metode black box testing dikarenakan pengujian yang dilakukan berfokus pada detail aplikasinya, seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi dan kesesuaian alur fungsi maupun proses bisnis.

4.1.4.1 Pengujian Sistem

Tahap ini , telah dilakukan pengujian *black box* terhadap sistem informasi *e-office* CV Belida. Pengujian ini bertujuan untuk memverifikasi fungsionalitas, mengevaluasi antarmuka, dan memastikan kesesuaian alur sistem dengan proses bisnis yang telah dirancang, sekaligus untuk mengidentifikasi potensi *bug*.

4.1.4.1.1 Pengujian Sistem Admin

Tahap ini akan menjelaskan hasil dari pengujian sistem *user admin* pada sistem informasi *e-office* pada CV Belida.

4.1.4.1.1.1 Pengujian Sistem Login

Pengujian pada sistem login dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem ini dilihat pada Tabel 4.5.

Tabel 4. 5 Pengujian Sistem Login

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>admin</i> mengisi <i>username</i> dan <i>password</i> yang sesuai di <i>database</i> dan menekan tombol login	Halaman akan berpindah masuk ke halaman <i>dashboard admin</i>	Halaman berpindah ke halaman <i>dashboard admin</i>	Sukses
Alternate Course				
2.	<i>admin</i> salah input <i>username</i> atau <i>password</i>	Menampilkan pesan <i>username</i> atau <i>password</i> salah	Menampilkan pesan <i>username</i> atau <i>password</i> salah	Sukses
3.	<i>Admin</i> input kode <i>sql injection</i> pada kolom <i>password</i>	Menampilkan pesan <i>username</i> atau <i>password</i> salah	Menampilkan pesan <i>username</i> atau <i>password</i> salah	Sukses

4.1.4.1.1.2 Pengujian Sistem Data Karyawan

Pengujian pada sistem Data Karyawam dilakukan dengan menggunakan basic course. Pengujian sistem data karyawan dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Pengujian Sistem Data Karyawan

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>admin</i> menekan menu data karyawan	Akan menampilkan halaman data karyawan	Menampilkan halaman data karyawan	Sukses

4.1.4.1.1.3 Pengujian Sistem Tambah Data Karyawan

Pengujian pada sistem tambah data karyawn dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah data karyawan ini dilihat tabel 4.7.

Tabel 4. 7 Pengujian Sistem Tambah Data Karyawan

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>Admin</i> mengisi data karyawan yang sesuai dan menekan tombol tambah karyawan	Menampilkan pesan tambah karyawan berhasil	Menampilkan pesan tambah karyawan berhasil	Sukses
Alternate Course				
2.	<i>Admin</i> kurang lengkap menginputkan data karyawan	Menampilkan pesan kolom isi karyawan harus diisi	Menampilkan pesan kolom isi karyawan harus diisi	Sukses

4.1.4.1.1.4 Pengujian Sistem Surat Masuk

Pengujian pada sistem surat masuk dilakukan dengan menggunakan basic course. Pengujian sistem surat masuk dapat dilihat pada tabel 4.8.

Tabel 4. 8 Pengujian Sistem Surat Masuk

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>admin</i> menekan menu kotak surat masuk	Akan menampilkan halaman surat masuk	Menampilkan halaman surat masuk	Sukses

4.1.4.1.1.5 Pengujian Sistem Tambah Surat Masuk

Pengujian pada sistem tambah surat masuk dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah surat masuk ini dilihat tabel 4.9.

Tabel 4. 9 Pengujian Sistem Tambah Surat Masuk

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>Admin</i> mengisi data surat masuk yang sesuai dan menekan tombol tambah surat	Menampilkan pesan tambah surat masuk berhasil	Menampilkan pesan tambah surat masuk berhasil	Sukses
Alternate Course				
2.	<i>Admin</i> kurang lengkap menginputkan data surat masuk	Menampilkan pesan kolom isi surat harus diisi	Menampilkan pesan kolom isi surat harus diisi	Sukses

4.1.4.1.1.6 Pengujian Sistem Surat Keluar

Pengujian pada sistem surat keluar dilakukan dengan menggunakan basic course. Pengujian sistem surat keluar dapat dilihat pada tabel 4.10.

Tabel 4. 10 Pengujian Sistem Surat Keluar

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>admin</i> menekan menu kotak surat keluar	Akan menampilkan halaman surat keluar	Menampilkan halaman surat keluar	Sukses

4.1.4.1.1.7 Pengujian Sistem Tambah Surat Keluar

Pengujian pada sistem tambah surat keluar dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah surat keluar ini dilihat tabel 4.11.

Tabel 4. 11 Pengujian Sistem Tambah Surat Keluar

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>Admin</i> mengisi data surat keluar yang sesuai dan menekan tombol tambah surat	Menampilkan pesan tambah surat keluar berhasil	Menampilkan pesan tambah surat keluar berhasil	Sukses
Alternate Course				
2.	<i>Admin</i> kurang lengkap menginputkan data surat keluar	Menampilkan pesan kolom isi surat harus diisi	Menampilkan pesan kolom isi surat harus diisi	Sukses

4.1.4.1.1.8 Pengujian Sistem Disposisi

Pengujian pada sistem disposisi dilakukan dengan menggunakan basic course. Pengujian sistem disposisi dapat dilihat pada Tabel 4.12.

Tabel 4. 12 Pengujian Sistem Disposisi

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>admin</i> menekan menu kotak disposisi	Akan menampilkan halaman disposisi	Menampilkan halaman disposisi	Sukses

4.1.4.1.1.9 Pengujian Sistem Tambah Disposisi

Pengujian pada sistem tambah disposisi dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah disposisi ini dilihat Tabel 4.13.

Tabel 4. 13 Pengujian Sistem Tambah Disposisi

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>Admin</i> mengisi data disposisi yang sesuai dan menekan tombol tambah disposisi	Menampilkan pesan tambah disposisi berhasil	Menampilkan pesan tambah disposisi berhasil	Sukses
Alternate Course				
2.	<i>Admin</i> kurang lengkap menginputkan data disposisi	Menampilkan pesan kolom isi surat harus diisi	Menampilkan pesan kolom isi disposisi harus diisi	Sukses

4.1.4.1.1.10 Pengujian Sistem Tambah Tindak Lanjut

Pengujian pada sistem tambah tindak lanjut dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah tindak lanjut ini dilihat Tabel 4.14.

Tabel 4. 14 Pengujian Sistem Tambah Tindak lanjut

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>Admin</i> mengisi data tindak lanjut disposisi yang sesuai dan menekan tombol tambah surat	Menampilkan pesan tambah tindak lanjut berhasil	Menampilkan pesan tindak lanjut berhasil	Sukses
Alternate Course				
2.	<i>Admin</i> kurang lengkap menginputkan data tindak lanjut	Menampilkan pesan kolom isi tindak lanjut harus diisi	Menampilkan pesan kolom isi tindak lanjut harus diisi	Sukses

4.1.4.1.1.11 Pengujian Sistem Agenda

Pengujian pada sistem agenda dilakukan dengan menggunakan *basic course*. Pengujian sistem agenda dapat dilihat pada Tabel 4.15.

Tabel 4. 15 Pengujian Sistem Agenda

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
2.	<i>admin</i> menekan menu kotak agenda	Akan menampilkan halaman agenda	Menampilkan halaman agenda	Sukses

4.1.4.1.1.12 Pengujian Sistem Tambah Agenda

Pengujian pada sistem tambah agenda dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah agenda ini dilihat Tabel 4.16.

Tabel 4. 16 Pengujian Sistem Tambah Agenda

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
3.	<i>Admin</i> mengisi data agenda yang sesuai dan menekan tombol tambah agenda	Menampilkan pesan tambah agenda berhasil	Menampilkan pesan tambah agenda berhasil	Sukses
Alternate Course				
4.	<i>Admin</i> kurang lengkap menginputkan data agenda	Menampilkan pesan kolom isi agenda harus diisi	Menampilkan pesan kolom isi agenda harus diisi	Sukses

4.1.4.1.1.13 Pengujian Sistem Tambah Anggota Agenda

Pengujian pada sistem tambah anggota agenda dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah anggota agenda ini dilihat Tabel 4.17.

Tabel 4. 17 Pengujian Sistem Tambah Anggota Agenda

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>Admin</i> mengisi data anggota agenda yang sesuai dan menekan tombol	Menampilkan pesan tambah anggota agenda berhasil	Menampilkan pesan tambah anggota agenda berhasil	Sukses

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
	tambah anggota agenda			
Alternate Course				
2.	<i>Admin</i> kurang lengkap menginputkan data anggota agenda	Menampilkan pesan kolom isi anggota agenda harus diisi	Menampilkan pesan kolom isi anggota agenda harus diisi	Sukses

4.1.4.1.14 Pengujian Sistem Memo

Pengujian pada sistem memo dilakukan dengan menggunakan basic course. Pengujian sistem memo dapat dilihat pada Tabel 4.18.

Tabel 4. 18 Pengujian Sistem Memo

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>admin</i> menekan menu kotak memo	Akan menampilkan halaman memo	Menampilkan halaman memo	Sukses

4.1.4.1.15 Pengujian Sistem Tambah Memo

Pengujian pada sistem tambah memo dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah memo ini dilihat tabel 4.19.

Tabel 4. 19 Pengujian Sistem Tambah Memo

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	<i>Admin</i> mengisi data memo yang sesuai dan menekan tombol tambah memo	Menampilkan pesan tambah memo berhasil	Menampilkan pesan tambah memo berhasil	Sukses
Alternate Course				
2.	<i>Admin</i> kurang lengkap menginputkan data memo	Menampilkan pesan kolom isi memo harus diisi	Menampilkan pesan kolom isi memo harus diisi	Sukses

4.1.4.1.2 Pengujian Sistem Karyawan

Tahap ini akan menjelaskan hasil dari pengujian sistem *user* karyawan pada sistem informasi *e-office* pada CV Belida.

4.1.4.1.2.1 Pengujian Sistem Login

Pengujian pada sistem login dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem ini dilihat pada Tabel 4.20.

Tabel 4. 20 Pengujian Sistem Login

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	Karyawan mengisi <i>username</i> dan <i>password</i> yang sesuai di <i>database</i> dan menekan tombol login	Halaman akan berpindah masuk ke halaman <i>dashboard</i> karyawan	Halaman berpindah ke halaman <i>dashboard</i> karyawan	Sukses
Alternate Course				
2.	Karyawan salah input <i>username</i> atau <i>password</i>	Menampilkan pesan <i>username</i> atau <i>password</i> salah	Menampilkan pesan <i>username</i> atau <i>password</i> salah	Sukses

4.1.4.1.2.2 Pengujian Sistem Disposisi

Pengujian pada sistem disposisi dilakukan dengan menggunakan basic course. Pengujian sistem disposisi dapat dilihat pada Tabel 4.21.

Tabel 4. 21 Pengujian Sistem Disposisi

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	Karyawan menekan menu kotak disposisi	Akan menampilkan halaman disposisi	Menampilkan halaman disposisi	Sukses

4.1.4.1.2.3 Pengujian Sistem Tambah Tindak Lanjut

Pengujian pada sistem tambah tindak lanjut dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah tindak lanjut ini dilihat Tabel 4.22.

Tabel 4. 22 Pengujian Sistem Tambah Tindak lanjut

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	Karyawan mengisi data tindak lanjut disposisi yang sesuai dan menekan tombol tambah surat	Menampilkan pesan tambah tindak lanjut berhasil	Menampilkan pesan tindak lanjut berhasil	Sukses
Alternate Course				
2.	Karyawan kurang lengkap menginputkan data tindak lanjut	Menampilkan pesan kolom isi tindak lanjut harus diisi	Menampilkan pesan kolom isi tindak lanjut harus diisi	Sukses

4.1.4.1.2.4 Pengujian Sistem Agenda

Pengujian pada sistem agenda dilakukan dengan menggunakan *basic course*. Pengujian sistem agenda dapat dilihat pada Tabel 4.23.

Tabel 4. 23 Pengujian Sistem Agenda

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	Karyawan menekan menu kotak agenda	Akan menampilkan halaman agenda	Menampilkan halaman agenda	Sukses

4.1.4.1.2.5 Pengujian Sistem Lihat Agenda

Pengujian pada sistem tambah agenda dilakukan dengan menggunakan *basic course*. Detail pengujian sistem tambah agenda ini dilihat Tabel 4.24.

Tabel 4. 24 Pengujian Sistem Lihat Agenda

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	Karyawan melihat data agenda yang sesuai	Akan menampilkan data agenda yang sesuai	Menampilkan halaman lihat agenda	Sukses

4.1.4.1.2.6 Pengujian Sistem Lihat Anggota Agenda

Pengujian pada sistem Lihat anggota agenda dilakukan dengan menggunakan *basic course*. Detail pengujian sistem tambah anggota agenda ini dilihat Tabel 4.25.

Tabel 4. 25 Pengujian Sistem Lihat Anggota Agenda

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	Karyawan melihat data anggota agenda yang sesuai dengan agenda yang dipilih	Menampilkan data tambah anggota agenda.	Menampilkan halaman lihat anggota agenda.	Sukses

4.1.4.1.2.7 Pengujian Sistem Memo

Pengujian pada sistem memo dilakukan dengan menggunakan *basic course*. Pengujian sistem memo dapat dilihat pada Tabel 4.26.

Tabel 4. 26 Pengujian Sistem Memo

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	Karyawan menekan menu kotak memo	Akan menampilkan halaman memo	Menampilkan halaman memo	Sukses

4.1.4.1.2.8 Pengujian Sistem Tambah Memo

Pengujian pada sistem tambah memo dilakukan dengan menggunakan *basic course* dan *alternative course*. Detail pengujian sistem tambah memo ini dilihat Tabel 4.27.

Tabel 4. 27 Pengujian Sistem Tambah Memo

No	Skenario	Hasil yang diharapkan	Hasil Pengujian	Status
Basic Course				
1.	Karyawan mengisi data surat keluar yang sesuai dan menekan tombol tambah memo	Menampilkan pesan tambah memo berhasil	Menampilkan pesan tambah memo berhasil	Sukses
Alternate Course				
2.	Karyawan kurang lengkap menginputkan data memo	Menampilkan pesan kolom isi memo harus diisi	Menampilkan pesan kolom isi memo harus diisi	Sukses

4.1.4.2 Implementasi Sistem

Tahap implementasi merupakan proses penerapan sistem informasi e-office yang telah dikembangkan ke dalam lingkungan operasional CV Belida. Tujuan dari tahap ini adalah untuk memastikan bahwa sistem dapat digunakan sesuai dengan kebutuhan perusahaan dan mendukung proses bisnis yang telah dirancang.

Proses implementasi dimulai dengan pemasangan sistem pada perangkat yang digunakan, baik di sisi server maupun client. Setelah itu dilakukan konfigurasi awal, seperti pembuatan akun pengguna, pengaturan hak akses, serta penyesuaian data awal agar sistem dapat langsung digunakan.

Selama implementasi, sistem mulai dijalankan oleh pengguna dalam kegiatan operasional sehari-hari. Jika terdapat kendala atau kekurangan, dilakukan perbaikan agar sistem dapat berfungsi dengan optimal.

Sebagai bukti bahwa implementasi telah dilakukan, dokumentasi berupa foto-foto hasil penerapan sistem ditampilkan pada bagian lampiran di akhir laporan. Dengan terlaksananya tahap ini, sistem informasi e-office resmi digunakan untuk mendukung proses administrasi dan manajemen dokumen secara digital di lingkungan CV Belida.

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

Bab ini menyajikan kesimpulan dari seluruh hasil skripsi serta memberikan rekomendasi dan saran yang relevan berdasarkan hasil skripsi yang diperoleh.

5.1. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, maka dapat diperoleh kesimpulan bahwa skripsi ini berhasil merancang dan membangun sebuah Sistem Informasi E-Office berbasis *desktop* untuk CV Belida menggunakan metode *Rapid Application Development* (RAD).

Proses pengembangan sistem melalui beberapa tahapan kunci. Tahap *Requirement Planning* diawali dengan wawancara bersama Bapak Ir. Seno Yunarwanto (Penanggung Jawab Operasional dan Keuangan) yang mengonfirmasi bahwa sistem pengarsipan masih manual, sehingga menghasilkan dokumen kebutuhan sistem seperti *context diagram*, DFD Level 1, dan ERD. Selanjutnya, tahap *User Design* menerapkan proses iteratif, di mana wireframe sebagai desain awal diuji kepada pengguna dan disempurnakan berdasarkan umpan balik. Contohnya adalah pada fitur penomoran surat yang berevolusi dari input manual menjadi otomatis dengan dropdown divisi untuk meminimalkan kesalahan.

Tahap *Construction* kemudian mengimplementasikan desain final tersebut menjadi aplikasi nyata dengan *database* yang memiliki sembilan tabel. Pengujian fungsionalitas sistem dilakukan pada tahap *Cutover* menggunakan metode *Black Box Testing* untuk memastikan semua fungsi berjalan sesuai harapan.

Dengan demikian, hasil akhir dari penelitian ini adalah sebuah aplikasi yang fungsional dan teruji, yang mampu memenuhi seluruh kebutuhan fungsional yang telah dirumuskan dalam penelitian. Sistem ini berhasil menyediakan fitur-fitur esensial seperti pengelolaan surat (masuk, keluar, dan disposisi), manajemen agenda, serta pencatatan memo internal. Fungsionalitas tersebut juga didukung oleh kemampuan penting lainnya seperti notifikasi otomatis melalui *WhatsApp* untuk disposisi dan agenda, serta fungsi *filter* di setiap modul untuk pencarian data yang cepat. Secara keseluruhan, sistem ini berhasil menjadi solusi untuk mengatasi permasalahan proses administrasi manual yang sebelumnya tidak efisien..

5.2. Saran

Berdasarkan hasil skripsi yang telah dilakukan, maka saran untuk pengembangan aplikasi selanjutnya sebagai berikut :

1. Sistem Informasi E-Office masih dapat dikembangkan dengan menambahkan fitur cuti karyawan supaya memastikan program memfasilitasi karyawan.
2. Sistem Informasi E-Office masih dapat dikembangkan dengan menambahkan fitur tanda tangan elektronik menggunakan teknologi digital *signature* resmi dari layanan pemerintah.

DAFTAR PUSTAKA

- [1] Jefry Tarantang, Annisa Awwaliyah, Maulidia Astuti, and Meidinah Munawaroh, “PERKEMBANGAN SISTEM PEMBAYARAN DIGITAL PADA ERA REVOLUSI INDUSTRI 4.0 DI INDONESIA,” *Jurnal Al Qardh*, vol. 4, 2019.
- [2] Abdul Aziz and T. Dirgahayu, “Pengembangan Model E-Office dan Purwarupa Intitusi Perguruan Tinggi di Indonesia (Development of E-Office Model and Prototype for Colleges in Indonesia),” *JUITA*, vol. 3, pp. 129–142, 2015.
- [3] H. Nopriandi, I. Kuantan Singingi, and T. Kuantan, “PERANCANGAN SISTEM INFORMASI REGISTRASI MAHASISWA,” *Jurnal Teknologi Dan Open Source*, vol. 1, no. 1, 2018.
- [4] N. Muthia, H. Amalia, A. Puspita, ; Ade, and F. Lestari, “RANCANG BANGUN SISTEM INFORMASI AKUNTANSI PENJUALAN DENGAN MODEL WATERFALL BERBASIS JAVA DESKTOP,” *Jurnal Ilmu Pengetahuan Dan Teknologi Komputer*, vol. 5, 2019, [Online]. Available: www.bsi.ac.idwww.bsi.ac.idwww.bsi.ac.id
- [5] M. A. Avila and D. Kurniadi, “Rancang Bangun Sistem Informasi E-Office Pada Tata Usaha Fakultas Teknik Universitas Negeri Padang,” *Jurnal Vocational Teknik Elektronika dan Informatika*, vol. 9, 2021, [Online]. Available: <http://ejournal.unp.ac.id/index.php/voteknika/index>
- [6] G. Pradini and A. Sudradjat, “Rancang Bangun Sistem Informasi Pengarsipan Surat Kantor Desa Berbasis Web,” *Information Management for Educators and Professionals*, vol. 5, no. 2, pp. 1–10, 2021.
- [7] H. Rohman and S. SHERALINDA, “Pengembangan Sistem Informasi Rawat Jalan dan Pelayanan Persalinan di Klinik Berbasis Web,” *Jurnal Kesehatan Vokasional*, vol. 5, no. 1, p. 53, Feb. 2020, doi: 10.22146/jkesvo.50482.
- [8] H. Damayanti Aprilia and Damayanti, “The Using of E-Office in Efforts to Achieve Company Performance with Excellent Performance Model,” *Advances in Social Science, Educational and Humanities Research*, vol. 606, 2021.
- [9] J. U. Haposan P Simanungkalit, “Konsep Dasar Sistem Informasi.”
- [10] S. Dwi Yulianti, N. Sri Mayangsari Susana Dwi Yulianti, S. Tinggi Teknologi, and N. Sri Mayangsari, “Perancangan dan Implementasi Sistem Pembayaran

- Tiket Berbasis Desktop Pada PT Alma Sarama Wisata Tour,” vol. 13, no. 2, 2016.
- [11] A. Andaru, “PENGERTIAN DATABASE SECARA UMUM”.
 - [12] Arbie, *Manajemen database dengan MYSQL*. Yogyakarta , 2004.
 - [13] Edy Winarno, Ali Zaki, and SmitDev Community, *Pemrograman web berbasis html5, php dan javascript / Edy Winarno, Ali Zaki, SmitDev Community*. Jakarta : Elex Media Komputindo, 2014.
 - [14] H. Ginardi, “Perancangan dan implementasi basis data”.
 - [15] R. Elmasri and S. B. Navathe, “Database Systems SEVENTH EDITION,” 2010.
 - [16] M. M. Hidayat, R. F. Zainal, and A. A. Efendi, “Educational Data Mining for Mapping Student Ability Based on School Location Using Apriori Method Case Study : SMK YPM Sidoarjo,” *JEECS (Journal of Electrical Engineering and Computer Sciences)*, vol. 8, no. 2, pp. 149–156, Dec. 2023, doi: 10.54732/jeees.v8i2.6.
 - [17] B. A. Priyaangga, D. B. Aji, M. Syahroni, N. T. S. Aji, and A. Saifudin, “Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 3, no. 3, p. 150, Aug. 2020, doi: 10.32493/jtsi.v3i3.5343.
 - [18] J. Martin, *Rapid application development*, 3rd ed. New York: New York : Macmillan Pub. Co. ; Toronto : Collier Macmillan Canada ; New York : Maxwell Macmillan International, 1991.
 - [19] F. Musvina, S. Rahmawati, S. Kom, M. Kom, and H. Andrianof, “IMPLEMENTASI METODE RAPID APPLICATION DEVELOPMENT (RAD) DALAM PERANCANGAN SISTEM INFORMASI PERPUSTAKAAN PADA SMPN 22 PADANG,” *JUISIK*, vol. 2, no. 2, 2022, [Online]. Available: <http://journal.sinov.id/index.php/juisik/indexHalamanUTAMAJurnal>:<https://journal.sinov.id/index.php>
 - [20] P. Brézillon, “Elaboration of the Contextual Graphs representation: From a conceptual framework to an operational software,” 2017.
 - [21] I. Polkovnikov, “Unified Control and Data Flow Diagrams Applied to Software Engineering and other Systems,” 2016.

- [22] K. P.G.T.H and S. Sagara, *MERCon 2018 : 4th International Multidisciplinary Engineering Research Conference : May 30-June 1, 2018, Civil Engineering Complex, University of Moratuwa, Sri Lanka*. IEEE, 2018.
- [23] N. H. Wong, “Making User-focused Prototype Using Design Sprint to Test, Design, and Prototype Mobile App Rapidly.”
- [24] D. A. Fatah, A. Mufarroha, and M. A. Husnah, “PERANCANGAN ANTARMUKA PENGGUNA SISTEM INFORMASI AKADEMIK BERBASIS WIREFRAMING WIREFRAMING-BASED ACADEMIC INFORMATION SYSTEM USER INTERFACE DESIGN,” vol. 11, no. 1, 2022.
- [25] S. Aminah and D. Puspita, “MODEL RAPID APPLICATION DEVELOPMENT DALAM PERANCANGAN SISTEM INFORMASI KEPEGAWAIAN.”
- [26] Alan M. Davis, “Software Requirements: A Tutorial.”

Halaman ini sengaja dikosongkan

LAMPIRAN

Lampiran 1 Surat Izin Penelitian

	KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR FAKULTAS ILMU KOMPUTER Jalan Raya Rungkut Madya Gunung Anyar Surabaya 60294 Email : fasikom@upnjatim.ac.id Laman : fasikom.upnjatim.ac.id
Nomor : 774/UN63.7/PJ/2025	Surabaya, 25 Februari 2025
Klasifikasi : B I A S A	
Lampiran : -	
Perihal : <u>Penelitian Tugas Akhir</u>	Kepada :
	Yth. Penanggung Jawab Operasional CV. Belida di Tempat
Dalam rangka menunjang kegiatan Akademik Mahasiswa Fakultas Ilmu Komputer Universitas Pembangunan Nasional "Veteran" Jawa Timur, yang melaksanakan tugas akhir. Sehubungan dengan kegiatan tersebut, maka dengan ini diajukan mahasiswa Fakultas Ilmu Komputer; Nama Lengkap : Randy Dwi Saputra Nomor NPM : 20082010100 Program Studi : Sistem Informasi Keperluan : Pengumpulan data untuk referensi tugas akhir Judul Penelitian : Rancang Bangun Sistem Informasi E-Office berbasis Desktop menggunakan Metode Rapid Application Development pada CV Belida Demikian atas kerja samanya, disampaikan terima kasih.	
	Dekan Fakultas Ilmu Komputer  Prof. Dr. Ir. Novrina Hendrasarie, MT. NIP : 19681126 199403 2 001
Dokumen ini diajukan melalui URL fasikom.link/polam Berkas dikirim melalui ajuan.fasikom@upnjatim.ac.id Arsip dokumen ini tersedia di URL fasikom.link/arsip	

Lampiran 2 Formulir Kuisioner

Hasil Wawancara dengan Penanggung Jawab Opersional CV Belida

Tanggal : Rabu, 12 Juni 2024

Waktu : 19.30 – 21.00

Tempat : Communal Coffee & Eatery

Narasumber : Ir. Seno Yunarwanto (Penanggung Jawab Operasional & Keuangan)

Tujuan : Mendapatkan informasi terkait proses surat menyurat di CV Belida

Hasil :

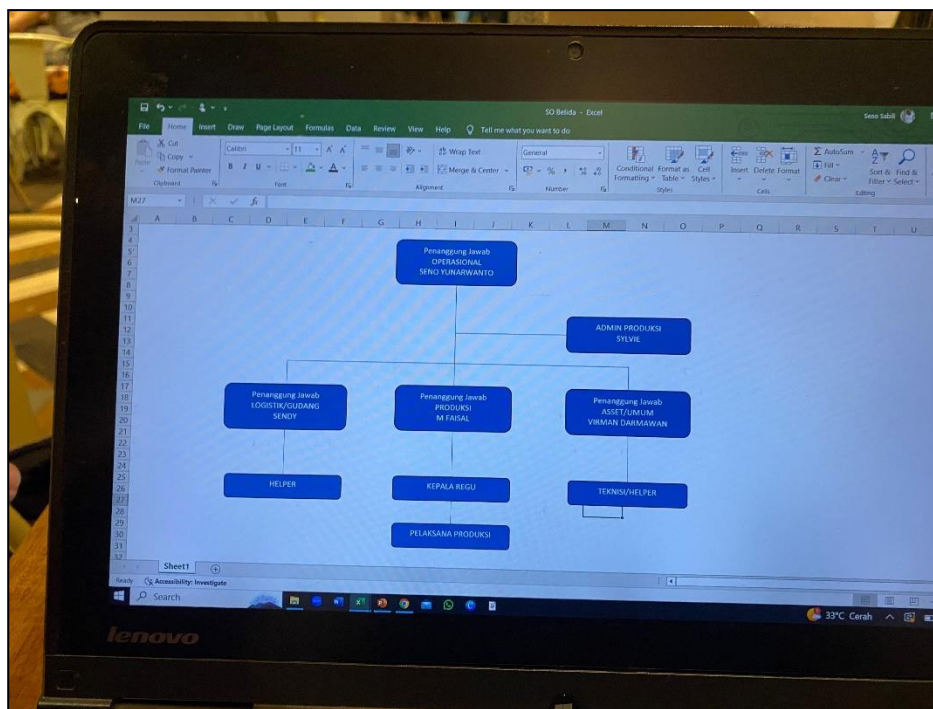
No	Pertanyaan	Jawaban
1.	Bagaimana sistem pengarsipan pada CV Belida saat ini ?	Untuk Sistem pengarsipan pada CV Belida ini masih manual menggunakan map ordner jadi setiap dokumen di masukkan ke dalam map sesuai dengan kategori surat, lalu di taruh pada lemari dokumen kantor.
2.	Terdapat Berapa jumlah Divisi pada CV Belida ? dan apa saja divisinya.	terdapat 3 divisi pada CV Belida yaitu Divisi Aset/Umum, Divisi Produksi, Divisi Logistik/Gudang.
3.	Bagaimana Struktur yang ada pada CV Belida ?	Untuk struktur dari CV Belida sendiri sendiri nantinya berupa lampiran foto.
4.	Siapa yang menerima ketika ada surat masuk dari luar ?	Admin yang diberi tugas menerima surat masuk.
5.	Bagaimana alur surat masuk dari luar ?	Alur surat masuk diterima oleh admin lalu akan diberikan kepada dicatat dan di teruskan ke karyawan terkait, setelah ditangani dapat dibuatkan surat keluar sebagai umpan balik surat masuk.
6.	Bagaimana alur untuk mengadakan agenda di CV Belida?	Karyawan dapat mengadakan agenda dengan cara menghubungi masing-masing anggota agenda secara manual.

7.	Siapa yang membuat surat keluar ?	Pembuat surat keluar yaitu admin yang ditugaskan.
----	-----------------------------------	---

Lampiran 3. Dokumentasi Wawancara



Lampiran 4. Struktur CV Belida



Lampiran 5. Dokumentasi Proses Presentasi Demo Aplikasi

