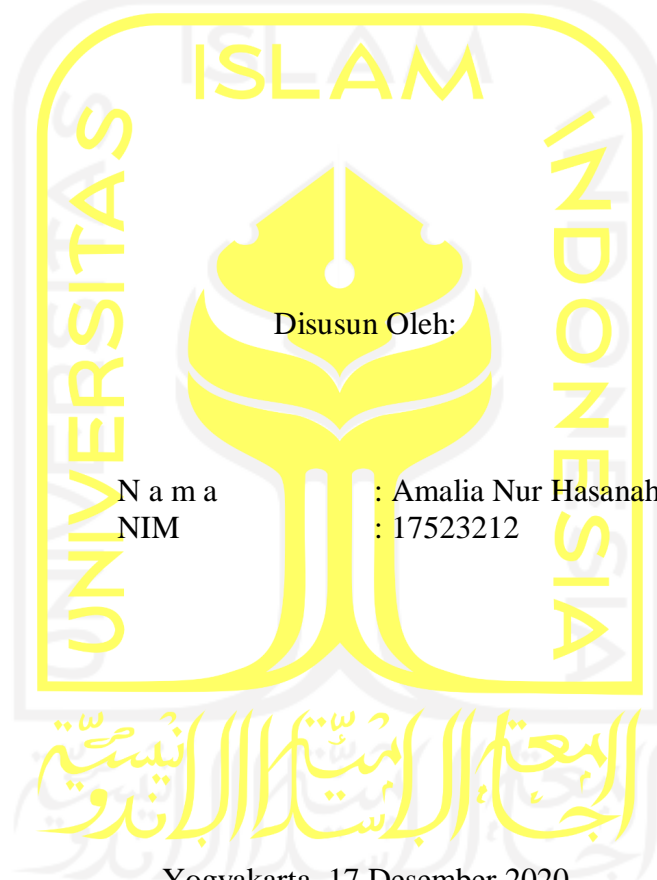


HALAMAN PENGESAHAN DOSEN PEMBIMBING

**IMPLEMENTASI JASPERREPORTS PADA SISTEM
INFORMASI MANAJEMEN (STUDI KASUS:
SISTEM INFORMASI MANAJEMEN
PENYELESAIAN LAPORAN)**

TUGAS AKHIR JALUR MAGANG



Yogyakarta, 17 Desember 2020

Pembimbing,

(Hanson Prihantoro Putro, S.T., M.T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**IMPLEMENTASI JASPERREPORTS PADA SISTEM
INFORMASI MANAJEMEN (STUDI KASUS:
SISTEM INFORMASI MANAJEMEN
PENYELESAIAN LAPORAN)**

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 14 Januari 2021

Tim Penguji

Ketua Penguji

Hanson Prihantoro Putro, S.T., M.T.

Anggota 1

Arrie Kurniawardhani, S.Si., M.Kom.

Anggota 2

Kholid Haryono, S.T., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

SKRIPSI

**IMPLEMENTASI JASPERREPORTS PADA SISTEM
INFORMASI MANAJEMEN (STUDI KASUS:
SISTEM INFORMASI MANAJEMEN
PENYELESAIAN LAPORAN)**



Disusun Oleh:

Nama : Amalia Nur Hasanah
NIM : 17523212

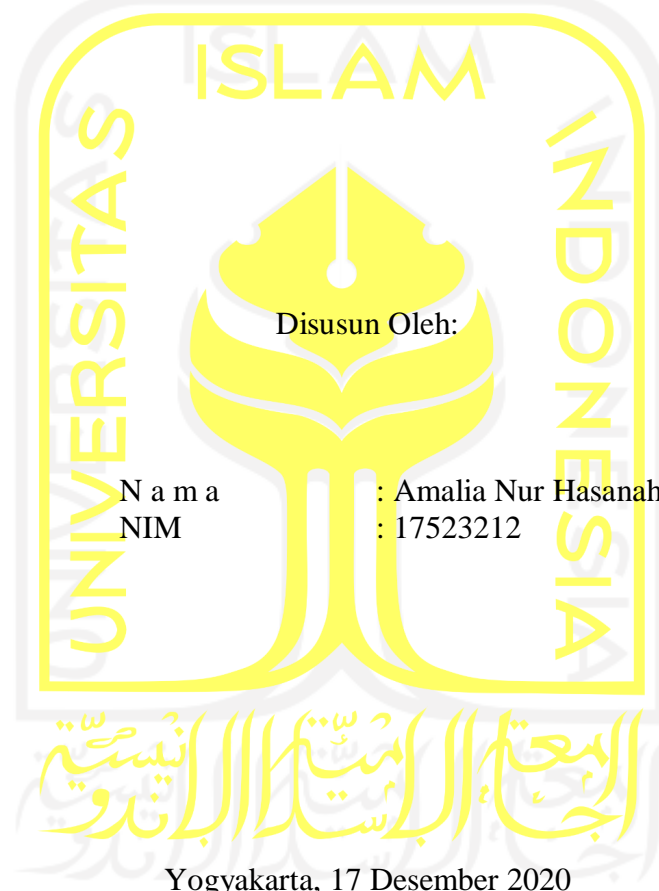
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**IMPLEMENTASI JASPERREPORTS PADA SISTEM
INFORMASI MANAJEMEN (STUDI KASUS:
SISTEM INFORMASI MANAJEMEN
PENYELESAIAN LAPORAN)**

TUGAS AKHIR JALUR MAGANG



Yogyakarta, 17 Desember 2020

Pembimbing,

(Hanson Prihantoro Putro, S.T., M.T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**IMPLEMENTASI JASPERREPORTS PADA SISTEM
INFORMASI MANAJEMEN (STUDI KASUS:
SISTEM INFORMASI MANAJEMEN
PENYELESAIAN LAPORAN)**

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 17 Desember 2020

Tim Penguji

Ketua Penguji

Hanson Prihantoro Putro, S.T., M.T. _____

Anggota 1

Arrie Kurniawardhani, S.Si., M.Kom. _____

Anggota 2

Kholid Haryono, S.T., M.Kom. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Amalia Nur Hasanah

NIM : 17523212


Tugas akhir dengan judul:

**IMPLEMENTASI JASPERREPORTS PADA SISTEM
INFORMASI MANAJEMEN (STUDI KASUS:
SISTEM INFORMASI MANAJEMEN
PENYELESAIAN LAPORAN)**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 17 Desember 2020



(Amalia Nur Hasanah)

HALAMAN PERSEMBAHAN

Laporan Tugas Akhir ini saya persembahkan untuk kedua orang tua (Hambali dan Jumiati), kakak dan adik saya (Ridho Muhari dan Ragil Nur Rasyid).



HALAMAN MOTO

“Life is not fair, get used to it”,

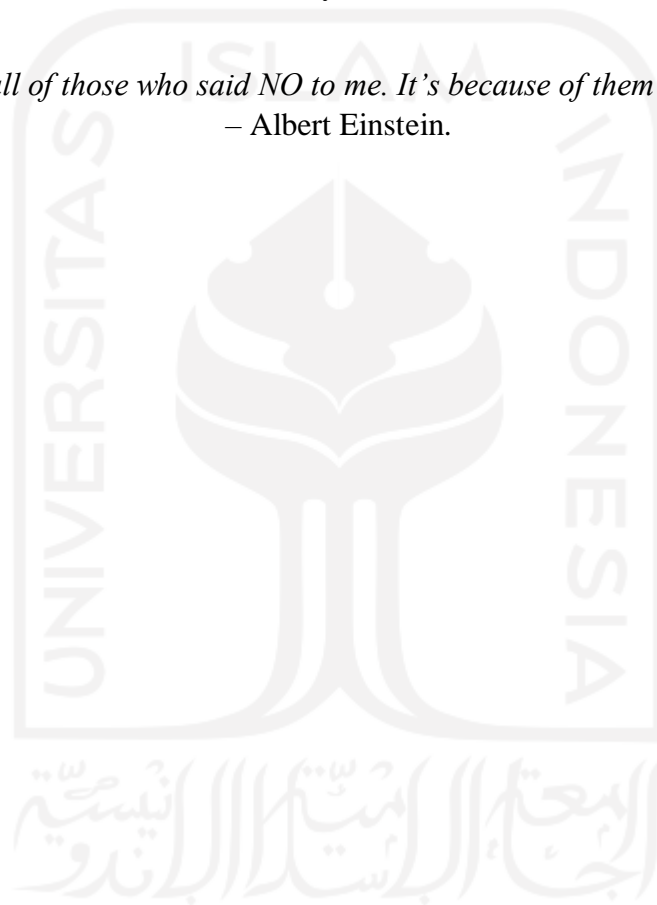
- Bill Gates.

“I refuse to accept other people’s ideas of happiness for me. As if there’s a one size fits all standard for happiness.”

– Kanye West.

“I am thankful for all of those who said NO to me. It’s because of them I’m doing it myself.”

– Albert Einstein.



KATA PENGANTAR

Assalamualaikum Wr.Wb.

Pertama-tama, penulis ucapkan puji dan syukur atas kehadiran Allah SWT yang telah memberikan rahmat dan berkah sehingga penulis dapat menyelesaikan Laporan Tugas Akhir dengan berjudul **“IMPLEMENTASI JASPERREPORTS PADA PENGEMBANGAN SISTEM INFORMASI MANAJEMEN (STUDI KASUS: SISTEM INFORMASI MANAJEMEN PENYELESAIAN LAPORAN)”** ini. Kemudian, shalawat serta salam senantiasa tercurah kepada bimbingan kita Nabi Besar Muhammad SAW yang telah menuntun kita menuju jalan yang terang benderang.

Laporan Tugas Akhir ini disusun untuk memenuhi kewajiban penulis dalam memperoleh gelar sarjana pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.

Dalam proses penyusunan Laporan Tugas Akhir ini, banyak sekali pihak yang mendukung penulis hingga proses ini pun menjadi mungkin untuk diselesaikan tepat waktu. Oleh karena itu, dalam kesempatan ini izinkan penulis untuk menyampaikan terima kasih kepada:

1. Bapak Hambali dan Ibu Jumiati selaku orang tua penulis yang tak pernah berhenti memberi dukungan dan doa atas segala sesuatu yang ada dalam hidup penulis sejak kecil hingga dewasa saat ini. Terima kasih juga kepada Ragil Nur Rasyid, adik penulis yang secara tidak langsung selalu membantu saya untuk menghadapi segala tekanan dalam hidup, juga Ridho Muhari sebagai kakak penulis yang memberikan inspirasi agar saya dapat terus maju.
2. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Program Studi Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Hanson Prihantoro Putro, S.T., M.T., selaku dosen pembimbing yang selalu membantu saya ketika mengalami kesulitan saat mengerjakan tugas akhir.
4. Bapak Wisnu, Bapak Bayu, Mba Erikha, Mba Maysha, Mba Nurul, dan karyawan lain yang ada di PT. Javan Cipta Solusi dan telah memberikan kesempatan kepada penulis untuk dapat magang di PT Javan Cipta Solusi, juga atas ilmu dan pelajaran serta bantuan yang diberikan selama proses magang.
5. Seluruh dosen di Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, yang telah mengajarkan saya banyak hal sejak semester 1 hingga sekarang.

6. Teman-teman yang selalu menjadi *support system* dan menemani ketika suka dan duka (Fika, Dewi, Adel, Nauli, Septi, Imeh, Dede, Aisy). Ack, yang telah mengajarkan banyak hal tentang hidup dan selalu siap mendengarkan penulis.
7. Semua pihak yang telah membantu penulis dalam hal apapun, semoga hal baik selalu menyertai kalian.
8. Penulis yang telah berusaha dan tidak menyerah.

Dengan adanya Laporan Tugas Akhir ini, penulis berharap dapat menjadi manfaat, tidak hanya bagi penulis, namun semua orang yang membaca dan terlibat dalam proses penyusunannya. Semoga Laporan Tugas Akhir ini dapat menambah pengetahuan pembaca. Penulis memohon maaf atas segala kekurangan, sesungguhnya kesempurnaan hanya milik Allah SWT.

Wassalamualaikum Wr.Wb.

Yogyakarta, 17 Desember 2020

(Amalia Nur Hasanah)

SARI

SIMPeL merupakan sebuah aplikasi berbasis *website* yang dikembangkan guna mendukung proses kerja lembaga Ombudsman Republik Indonesia dalam memproses laporan masyarakat. Salah satu fitur penting dari aplikasi SIMPeL versi terbaru adalah cetak surat. Fitur tersebut digunakan oleh pengguna untuk menghasilkan keluaran berupa dokumen siap cetak yang berisi data tentang laporan masyarakat tertentu. Awalnya, fitur cetak surat dikembangkan dengan menulis *source* HTML secara *hardcode* yang mana pekerjaan tersebut hanya dapat dilakukan oleh *programmer*. Proses pengembangan pun terbilang lama mengingat banyaknya pekerjaan lain yang dimiliki oleh *programmer*. Terlebih, jumlah surat yang harus dibuat lebih banyak daripada versi sebelumnya. *Reporting tools* diperlukan untuk memudahkan proses pengembangan fitur cetak surat tersebut. Dalam pengembangan aplikasi SIMPeL, *reporting tools* yang digunakan adalah *JasperReports*. *JasperReports* merupakan sebuah *tool* pembuatan laporan *open-source* yang menggunakan bahasa pemrograman Java. Hasil dari didapatkan dari implementasi *JasperReports* adalah kemudahan pembuatan surat sehingga proses pembuatan fitur cetak surat menjadi lebih cepat. Dengan kemudahan yang diberikan *JasperReports*, proses pembuatan desain surat untuk aplikasi SIMPeL berhasil dikerjakan oleh lebih banyak orang, bahkan untuk mereka yang tidak terlalu familiar dengan *coding*. Pada penelitian ini akan dibahas tentang bagaimana penggunaan *JasperReports* dalam proses pengembangan aplikasi SIMPeL dengan harapan dapat menjadi pembelajaran dan referensi sehingga proses pembuatan laporan atau surat menjadi lebih efisien.

Kata kunci: *JasperReports*, SIMPeL, *Jaspersoft Studio*, *JasperReports Server*

GLOSARIUM

<i>Change request</i>	dokumen yang berisi permintaan penyesuaian sistem
<i>Hardcode</i>	penulisan kode secara statik dimana nilainya tidak dapat diubah
<i>Reporting tools</i>	sebuah <i>tools</i> untuk membuat <i>report</i> dari sumber data
Ombudsman	lembaga negara yang mempunyai kewenangan mengawasi penyelenggaraan pelayanan publik



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Ruang Lingkup Magang	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Sistematika Penulisan	3
BAB II KAJIAN PUSTAKA	5
2.1 Sistem Informasi Manajemen Penyelesaian Laporan (SIMPeL)	5
2.1.1 Modul Penerimaan dan Verifikasi Laporan	5
2.1.2 Modul Pemeriksaan (Riksa)	6
2.1.3 Modul Resolusi dan Monitoring (Resmon)	6
2.2 <i>JasperReports</i>	6
2.2.1 <i>Jaspersoft Studio</i>	7
2.2.2 <i>Jasperreports Server</i>	8
2.3 Implementasi <i>JasperReports</i> pada Pengembangan Sistem Informasi Lain	8
BAB III METODOLOGI MAGANG	10
3.1 Manajemen Proyek	10
3.1.1 Tahap Persiapan	10
3.1.2 <i>Requirement Analysis</i>	10
3.1.3 <i>Build Application/ Prototype</i>	11
3.1.4 Pengujian	13
3.1.5 Implementasi	14
3.1.6 Tinjauan Pasca Implementasi	14

3.1.7	<i>Transfer Knowledge</i>	14
3.1.8	Serah Terima Pekerjaan	15
3.1.9	Garansi	15
3.2	Metode Pengembangan Perangkat Lunak SIMPeL	16
3.2.1	<i>Analysis and Quick Design</i>	16
3.2.2	<i>Prototype Cycles</i>	17
3.2.3	<i>Testing</i>	18
3.2.4	<i>Deployment</i>	18
BAB IV HASIL MAGANG		19
4.1	<i>Analysis and Quick Design</i>	22
4.2	<i>Prototype Cycles</i>	24
4.2.1	Konfigurasi <i>Jasperreports Server</i> pada <i>Jaspersoft Studio</i>	24
4.2.2	Konfigurasi <i>Database SIMPeL</i> pada <i>Jaspersoft Studio</i>	25
4.2.3	Proses Desain Surat pada <i>Jaspersoft Studio</i>	26
4.2.4	Pengambilan Data dari <i>Database SIMPeL</i>	28
4.2.5	Proses <i>Publish</i> Surat ke <i>JasperReports Server</i>	32
4.3	<i>Testing</i>	33
4.4	<i>Deployment</i>	33
BAB V REFLEKSI PELAKSANAAN MAGANG		35
5.1	Teknis	35
5.2	Non-Teknis	36
5.2.1	<i>Time Management</i>	36
5.2.2	Komunikasi	36
5.2.3	<i>Knowledge</i>	36
BAB VI KESIMPULAN DAN SARAN		38
6.1	Kesimpulan	38
6.2	Saran	38
DAFTAR PUSTAKA		39
LAMPIRAN		41

DAFTAR TABEL

Tabel 3.1 Spesifikasi teknis aplikasi SIMPeL

10



DAFTAR GAMBAR

Gambar 2.1 Tampilan <i>Jaspersoft Studio</i>	7
Gambar 2.2 Tampilan <i>Jasperreports Server</i>	8
Gambar 3.1 Fitur label pada <i>ActiveCollab</i> yang menunjukkan status pengerjaan <i>task</i>	12
Gambar 3.2 <i>Repository Gitlab</i> pada proses pengembangan SIMPeL	12
Gambar 3.3 Contoh <i>scenario test</i> modul PVL SIMPeL	14
Gambar 3.4 Alur proses metode <i>Rapid Application Development (RAD)</i>	16
Gambar 4.1 Fitur cetak surat pada aplikasi SIMPeL	19
Gambar 4.2 Contoh elemen <i>text field</i> dengan logika percabangan yang rumit	21
Gambar 4.3 Contoh elemen <i>text field</i> tanpa logika percabangan	22
Gambar 4.4 Desain “Surat Pemberitahuan Dimulainya Pemeriksaan” dari Ombudsman Republik Indonesia	23
Gambar 4.5 Folder yang menyimpan surat SIMPeL di <i>JasperReports Server</i>	24
Gambar 4.6 Menu pembuatan koneksi <i>JasperReports Server</i>	25
Gambar 4.7 Konfigurasi akses terhadap <i>JasperReports Server</i>	25
Gambar 4.8 Penambahan <i>data adapter</i>	26
Gambar 4.9 Konfigurasi <i>database</i> SIMPeL pada <i>JDBC data adapter</i>	26
Gambar 4.10 Penambahan <i>layout</i> surat baru	27
Gambar 4.11 Desain <i>layout</i> “Surat Pemberitahuan Dimulainya Pemeriksaan”	28
Gambar 4.12 Menu penambahan parameter	29
Gambar 4.13 Konfigurasi parameter baru	29
Gambar 4.14 <i>Query</i> “Surat Pemberitahuan Dimulainya Pemeriksaan”	30
Gambar 4.15 Hasil pembacaan <i>field</i> dari <i>query</i>	30
Gambar 4.16 <i>Expression Editor</i> pada salah satu elemen <i>text field</i> “Surat Pemberitahuan Dimulainya Pemeriksaan”	31
Gambar 4.17 Salah satu bentuk logika <i>if-else</i> pada surat SIMPeL	31
Gambar 4.18 Hasil <i>preview</i> surat “Usulan Konsiliasi” yang telah didesain	32
Gambar 4.19 Menu <i>publish</i> ke <i>JasperReports Server</i>	33
Gambar 4.20 Pendefinisian lokasi <i>path</i> surat dari <i>JasperReports Server</i>	33

BAB I PENDAHULUAN

1.1 Latar belakang

Aktivitas magang dilakukan oleh penulis di PT Javan Cipta Solusi selama 6 bulan. Selama periode waktu tersebut, penulis terlibat di beberapa proyek, di antaranya adalah : Portal PPSDM, SIMPeL, dan APIK. Dari beberapa proyek yang pernah dikerjakan, SIMPeL merupakan proyek yang paling lama dikerjakan oleh penulis. Beberapa pekerjaan yang dilakukan terkait aplikasi SIMPeL adalah memperbaiki *issue coding* dan yang paling banyak adalah membuat surat dengan *JasperReports*. Lamanya waktu yang diluangkan untuk terlibat dalam pembuatan surat SIMPeL membuat penulis memilih untuk membahas mengenai implementasi *JasperReports* pada aplikasi SIMPeL.

Aplikasi SIMPeL adalah sebuah sistem informasi yang digunakan oleh Ombudsman Republik Indonesia guna mendukung prosedur penyelesaian laporan masyarakat. Alur penyelesaian laporan oleh Ombudsman Republik Indonesia tidak selalu sama dari tahun ke tahun. Oleh karena itu, aplikasi SIMPeL juga turut mengalami perubahan. Hingga pada tahun 2019, SIMPeL versi 3.0 berada dalam tahap pengembangan oleh PT Javan Cipta Solusi. Salah satu fitur penting pada aplikasi SIMPeL adalah fitur cetak surat yang ada pada hampir semua proses bisnis dalam alur penyelesaian laporan masyarakat. Fitur ini digunakan oleh pengguna untuk menghasilkan keluaran dokumen surat yang berisi data tentang laporan masyarakat dalam bentuk PDF atau DOCX.

Dalam pengembangan aplikasi SIMPeL khususnya sebagai penunjang dalam fitur cetak surat, digunakanlah teknologi *JasperReports*. *JasperReports* merupakan sebuah *reporting tool open-source* dengan bahasa pemrograman Java yang memungkinkan keluaran dokumen dalam berbagai format, seperti PDF, HTML, XLS, XLSX, RTF, CSV, XML, DOCX, ODT, ODS, dan JPRINT. Pada versi-versi sebelumnya, fitur cetak surat pada aplikasi SIMPeL hanya perlu mencetak surat dengan jumlah sedikit (maksimal 4 surat), yang mana jumlahnya sangat jauh berbeda dibandingkan dengan jumlah surat pada versi 3.0.

Sebelumnya, surat aplikasi SIMPeL dibuat dengan menulis *source* HTML secara manual pada *code* aplikasi SIMPeL. Cara tersebut sangat tidak mungkin diterapkan pada surat yang berjumlah ratusan. Selain itu, cara tersebut hanya dapat dilakukan oleh *programmer* sehingga waktu yang dibutuhkan akan sangat lama mengingat banyaknya pekerjaan lain yang

dimiliki oleh *programmer*. Kemudian, pengembang memutuskan untuk memilih *JasperReports* sebagai *reporting tools* dalam pengembangan aplikasi SIMPeL agar waktu pengerjaan surat aplikasi SIMPeL menjadi lebih efisien. *JasperReports* memungkinkan pembuatan *report* dengan mudah hanya dengan melakukan *drag and drop* elemen-elemen yang tersedia. Selain itu, *JasperReports* dipilih juga karena sifatnya yang *open-source* dan gratis. Kemudahan penggunaan *JasperReports* memungkinkan proses pembuatan surat tidak hanya dapat dikerjakan oleh *programmer*, namun juga analis yang lebih tahu tentang aplikasi secara keseluruhan sehingga apabila terdapat masalah pada surat, masalah tersebut dapat segera ditangani.

Laporan tugas akhir ini akan fokus membahas tentang implementasi *JasperReports* dalam proses pengembangan aplikasi SIMPeL, khususnya pada fitur cetak surat. Pembahasan dimulai dari tahap awal pengembangan, hingga aplikasi sampai ke tangan pengguna dan melalui tahap evaluasi juga perbaikan. Diharapkan laporan tugas akhir ini dapat menjadi pembelajaran dan referensi sehingga proses pembuatan report pada sebuah sistem informasi menjadi lebih efisien.

1.2 Ruang Lingkup Magang

Aktivitas magang dilaksanakan di PT. Javan Cipta Solusi selama 6 bulan sejak bulan Maret hingga Agustus tahun 2020. Sementara itu, proyek SIMPeL sendiri telah dimulai sejak bulan Oktober tahun 2019 dengan durasi proyek selama 6 bulan ditambah masa garansi hingga akhir tahun 2021. Kemudian, aktivitas yang dilakukan selama magang, khususnya pada fitur cetak surat aplikasi SIMPeL adalah sebagai berikut:

- a. Desain *layout* surat *.jrxml* mengikuti *template* yang telah ditentukan oleh Ombudsman Republik Indonesia dengan *Jaspersoft Studio*
- b. Pembuatan dan penghubungan *query* ke setiap surat aplikasi SIMPeL dengan *Jaspersoft Studio*
- c. Penghubungan surat SIMPeL dengan *data-source* pada elemen *text field* menggunakan bahasa pemrograman Java
- d. *Publishing* hasil pembuatan surat SIMPeL ke *Jasperreports Server* yang telah terpasang di aplikasi SIMPeL
- e. Pendaftaran surat yang telah di-*publish* pada aplikasi SIMPeL sesuai dengan proses bisnisnya

- f. Pengujian terhadap surat yang telah dibuat, baik sebelum atau setelah surat di-*publish* ke *Jasperreports Server* dan didaftarkan pada aplikasi SIMPeL.

1.3 Tujuan

Tujuan dari laporan tugas akhir ini adalah untuk mengimplementasikan *JasperReports* pada pengembangan Sistem Informasi Manajemen Penyelesaian Laporan (SIMPeL). Tujuan ini perlu dicapai untuk mengetahui proses dan tahapan implementasi *JasperReports* pada pengembangan sebuah sistem informasi. Selain itu, dengan adanya pekerjaan ini, kesimpulan tentang keefektifan implementasi *JasperReports* didapatkan dan dijadikan referensi untuk pengembangan sistem informasi lainnya.

1.4 Manfaat

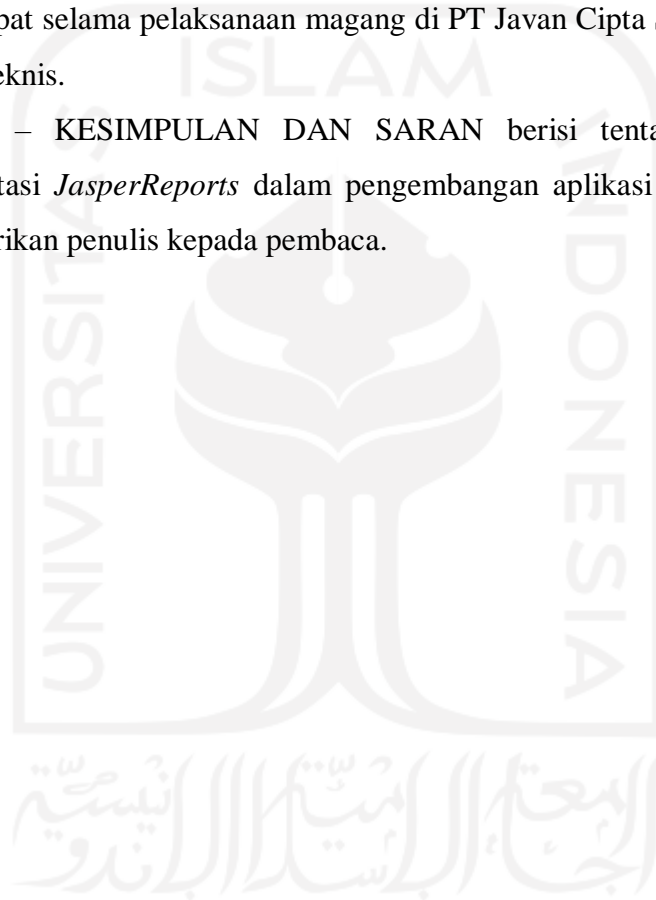
Manfaat yang diperoleh dari implementasi *JasperReports* dalam pengembangan aplikasi SIMPeL adalah efisiensi waktu dalam pembuatan surat untuk aplikasi SIMPeL. Dengan menggunakan sebuah *reporting tool*, yaitu *JasperReports*, proses pembuatan surat tidak hanya dapat dikerjakan oleh *programmer*, namun juga dibantu oleh analis serta sumber daya manusia lain yang berada di dalam lingkungan pengembang. Dengan demikian, surat-surat untuk melengkapi fitur cetak surat aplikasi SIMPeL dapat diselesaikan dalam waktu yang singkat.

1.5 Sistematika Penulisan

Untuk menguraikan topik yang akan dibahas, laporan ini menggunakan sistematika penulisan sebagai berikut:

- a. BAB I – PENDAHULUAN menguraikan mengenai latar belakang aplikasi SIMPeL dan implementasi *JasperReports* dalam proses pengembangannya, ruang lingkup magang, tujuan implementasi *JasperReports*, manfaat implementasi *JasperReports*, dan sistematika penulisan.
- b. BAB II – DASAR TEORI menguraikan beberapa teori yang mendukung proses pengembangan aplikasi SIMPeL dengan mengimplementasikan *JasperReports*. Beberapa hal yang dibahas adalah Sistem Informasi Manajemen Penyelesaian Laporan (SIMPeL), *JasperReports*, dan beberapa pembahasan mengenai penerapan *JasperReports* pada pengembangan sistem informasi lain.

- c. BAB III – METODOLOGI MAGANG terdiri dari 2 sub bab, yaitu: manajemen proyek dan metode pengembangan aplikasi SIMPeL. Manajemen proyek membahas tentang bagaimana proyek SIMPeL dikelola untuk mencapai tujuan proyek. Metode pengembangan membahas tentang bagaimana aplikasi SIMPeL dikembangkan berdasarkan metode pengembangan perangkat lunak yang diterapkan.
- d. BAB IV – HASIL MAGANG membahas mengenai proses implementasi *JasperReports* pada pengembangan aplikasi SIMPeL
- e. BAB V – REFLEKSI PELAKSANAAN MAGANG menguraikan mengenai refleksi yang didapat selama pelaksanaan magang di PT Javan Cipta Solusi dari segi teknis dan non-teknis.
- f. BAB VI – KESIMPULAN DAN SARAN berisi tentang kesimpulan dari implementasi *JasperReports* dalam pengembangan aplikasi SIMPeL, serta saran yang diberikan penulis kepada pembaca.



BAB II

KAJIAN PUSTAKA

2.1 Sistem Informasi Manajemen Penyelesaian Laporan (SIMPeL)

SIMPeL adalah sebuah aplikasi berbasis *website* milik lembaga Ombudsman Republik Indonesia yang dijalankan sejak tahun 2014 guna membantu proses penyelesaian laporan masyarakat (Ombudsman Republik Indonesia, 2016). Dari masa ke masa, prosedur penyelesaian laporan oleh Ombudsman Republik Indonesia tidak lepas dari perubahan-perubahan. Hal tersebut menuntut SIMPeL sebagai aplikasi yang menunjang prosedur tersebut mau tidak mau harus turut mengalami perubahan. Hingga saat ini, aplikasi SIMPeL telah memasuki versi 3.0 yang dikembangkan oleh PT Javan Cipta Solusi.

Aplikasi SIMPeL dapat diakses hanya oleh pegawai Ombudsman Republik Indonesia. Aplikasi SIMPeL digunakan untuk merekam, menyajikan data, serta memonitor penyelesaian laporan/pengaduan masyarakat oleh internal Ombudsman Republik Indonesia maupun instansi penyelenggara pelayanan publik. Setiap hal dalam penanganan laporan masyarakat harus dimasukkan ke dalam SIMPeL, mulai dari hasil kajian hukum, hasil klarifikasi dengan terlapor, hasil investigasi lapangan, tambahan data dari pelapor, sampai kepada berita acara penutupan laporan (Tindige, Tampi, & Londa, 2017). Untuk menunjang prosedur tersebut, aplikasi SIMPeL terdiri dari 3 modul utama, yaitu modul Pemeriksaan dan Verifikasi Laporan (PVL), modul Pemeriksaan (Riksa), dan modul Resolusi dan *Monitoring* (Resmon).

2.1.1 Modul Penerimaan dan Verifikasi Laporan

Dari keseluruhan alur dalam prosedur penyelesaian laporan oleh Ombudsman Republik Indonesia, tim Penerimaan dan Verifikasi Laporan (PVL) adalah tim yang pertama kali menangani laporan masyarakat. Tim PVL bertugas menerima pengaduan dari masyarakat termasuk melakukan konsultasi dengan masyarakat tentang laporan tersebut (Nugraha, 2019). Selain menerima dan melakukan konsultasi, Tim PVL juga melakukan pencatatan dan verifikasi laporan. Data dari pencatatan dan kelengkapan dokumen mengenai laporan inilah yang kemudian dimasukkan oleh tim PVL ke dalam modul PVL aplikasi SIMPeL. Alur proses bisnis pada modul PVL dimulai dari memilih tipe laporan dan memasukkan data mengenai laporan. Pada modul ini, terdapat proses *input* verifikasi formil, *input* verifikasi materil,

klasifikasi laporan masyarakat dan pleno (PT Javan Cipta Solusi, 2019). Alur lengkap proses bisnis modul PVL dapat dilihat pada lampiran a.

2.1.2 Modul Pemeriksaan (Riksa)

Laporan masyarakat yang telah diterima dan diverifikasi oleh tim PVL kemudian ditindaklanjuti oleh anggota keasistenan substansi Ombudsman Republik Indonesia. Anggota keasistenan substansi melakukan pemeriksaan substansi atas laporan, menindaklanjuti laporan, melakukan koordinasi dengan lembaga terkait, dan melakukan pencegahan maladministrasi dalam penyelenggaraan pelayanan publik. Hasil dari pemeriksaan ini kemudian dimasukkan ke dalam modul Riksa aplikasi SIMPeL. Pada modul ini, terdapat proses disposisi tim riksa, pemberitahuan mulai pemeriksaan, laporan hasil pemeriksaan dokumen, tindak lanjut, laporan akhir hasil pemeriksaan, *monitoring* korektif, pleno pemeriksaan, pelimpahan resmon, penyerahan tim resmon, dan klasifikasi laporan masyarakat riksa (PT Javan Cipta Solusi, 2019). Alur mengenai proses bisnis modul Riksa dapat dilihat pada lampiran b.

2.1.3 Modul Resolusi dan *Monitoring* (Resmon)

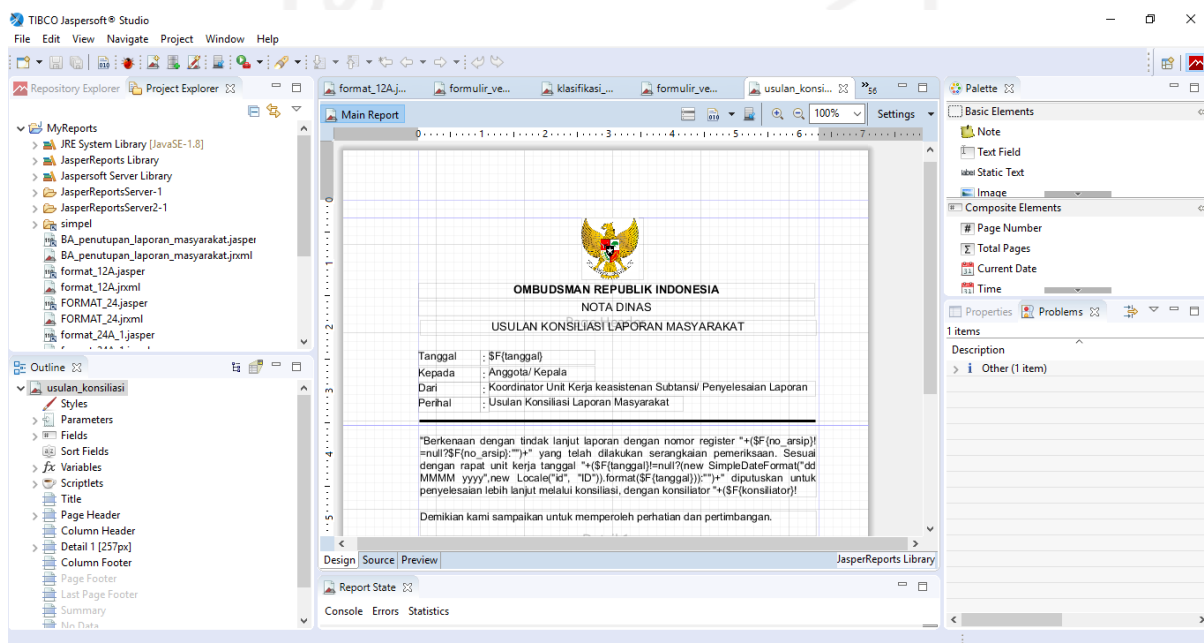
Modul Resmon merupakan modul yang menunjang kinerja tim Resolusi dan *Monitoring* (Resmon) Ombudsman Republik Indonesia. Tim Resmon melakukan resolusi melalui mekanisme mediasi, ajudikasi khusus ganti rugi, serta melakukan *monitoring* kesepakatan mediasi dan rekomendasi. Pada modul Resmon, prosedur penyelesaian laporan terdiri dari beberapa proses, yaitu disposisi tim resmon, telaah, pra resolusi, mediasi, rekomendasi, dan ajudikasi (PT Javan Cipta Solusi, 2019). Alur mengenai modul tersebut dapat dilihat pada lampiran c.

2.2 JasperReports

JasperReports merupakan sebuah *library* pembuatan laporan *open-source* yang dikembangkan oleh TIBCO Software Inc. *JasperReports* sebagai sebuah *Reporting tools* biasanya digunakan untuk menghasilkan suatu keluaran berupa laporan atau surat atau dokumen lain yang memerlukan pengambilan data dari *database*. *JasperReports* dapat menghasilkan keluaran berupa dokumen atau laporan atau surat dengan format PDF, HTML, XLS, XLSX, RTF, CSV, XML, DOCX, ODT, ODS, dan JPRINT (*JasperReports® Library / Jaspersoft Community*, 2020.).

2.2.1 Jaspersoft Studio

Jaspersoft Studio merupakan sebuah perangkat lunak berbasis aplikasi desktop untuk membuat *report* pada lingkup *JasperReports*. Tampilan *Jaspersoft Studio* dapat dilihat pada **Gambar 2.1**. *Software* tersebut digunakan untuk pembuatan desain *report*, membangun *query*, serta menulis *complex expression* dalam bahasa pemrograman Java untuk ditampilkan pada *report* yang dibuat. Dalam penggunaannya, *Jaspersoft studio* menyediakan berbagai macam elemen visual yang mendukung tampilan *report*, antara lain: *static text*, *text field*, *table*, *chart*, *image*, dan lain sebagainya. Pembuatan *report* dengan perangkat lunak ini juga memungkinkan dilakukannya *editing report* melalui kode *jrxml* yang otomatis dihasilkan pada saat *layout report* dibuat.

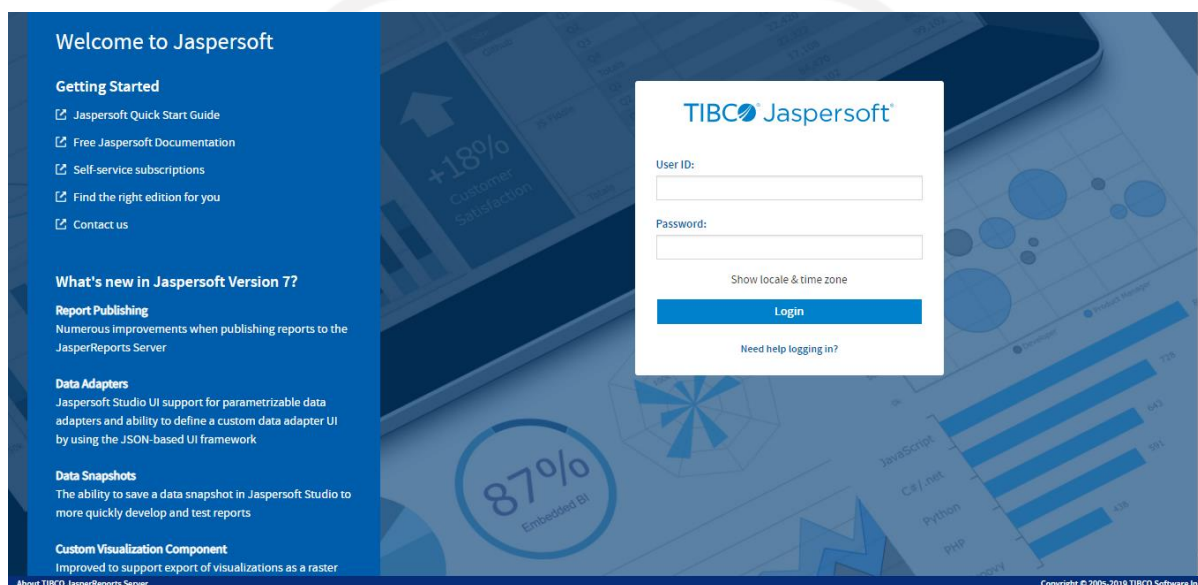


Gambar 2.1 Tampilan *Jaspersoft Studio*

Jaspersoft Studio juga digunakan untuk melakukan *publish report* apabila telah terintegrasi dengan *Jasperreports Server*, sehingga *report* yang telah didesain dapat digunakan pada sistem yang ingin dibangun. Selain itu, *Jaspersoft Studio* memungkinkan akses terhadap berbagai macam *data-source* seperti *big data*, CSV, Hibernate, *Jaspersoft Domain*, JavaBeans, JDBC, JSON, NoSQL, TIBCO Spotfire® Information Links, XML, atau *data-source* lain (*Jaspersoft® Studio | Jaspersoft Community*, 2020).

2.2.2 Jasperreports Server

JasperReports Server merupakan sebuah *reporting server* dalam lingkup *JasperReports* yang memungkinkan penyematannya di dalam sebuah sistem atau perangkat lunak (*JasperReports® Server* / *Jaspersoft Community*, 2020). Untuk menggunakannya, diperlukan log-in seperti ditunjukkan pada **Gambar 2.2**. Dalam penggunaannya, *Jasperreports Server* harus terintegrasi dengan *editing software*, seperti misalnya *Jaspersoft Studio* atau *iReport Designer* untuk dapat mengirimkan hasil pembuatan *report* ke sistem atau perangkat lunak yang dikembangkan.



Gambar 2.2 Tampilan *Jasperreports Server*

2.3 Implementasi *JasperReports* pada Pengembangan Sistem Informasi Lain

Salah satu penelitian luar membahas tentang implementasi *JasperReports* dalam pengembangan sebuah sistem informasi untuk menghasilkan sebuah *report* (Sapre et al., 2012). Pada penelitian tersebut, disimpulkan bahwa penggunaan *JasperReports* membantu dalam hal fleksibilitas dalam pembuatan *report*. *JasperReports* memisahkan antara *template report* dan data *report*. Hal tersebut memungkinkan perubahan *template report* tanpa memengaruhi data *report*, sehingga sangat bermanfaat untuk memenuhi kebutuhan bisnis dan proyek dengan cepat dan terukur.

Pada pengembangan sebuah sistem informasi penggajian karyawan pada *commendaire vennontschap* (CV) RGL bordir dan konveksi Pacitan, digunakan *software iReport Designer* untuk membuat perancangan laporan secara visual (Riestiana & Sukadi, 2014). Kemudian,

JasperReports juga digunakan untuk mengkompilasi dan menghasilkan *report* berupa file **jasper* atau **jrxml*. Yang kedua adalah pengembangan sistem penjadwalan kuliah pada Program Studi Sistem Informasi UNIKOM (Marpaung, 2013). Dalam pengembangan sistem tersebut, *JasperReports* digunakan untuk mencetak hasil *reporting* dalam bentuk PDF.

JasperReports sering digunakan dalam mengembangkan sebuah sistem informasi, tetapi tidak ada penelitian lokal yang menjadikan implementasi *JasperReports* sebagai kajian utama penelitian tersebut. Penelitian hanya menyebutkan bahwa *JasperReports* diimplementasikan untuk menunjang fitur *reporting* pada sistem informasi yang dikembangkan. Penjelasan lebih lanjut mengenai bagaimana implementasi dan hasil dari implementasi tidak dibahas, baik pada bagian pembahasan maupun hasil. Oleh karena itu, pada penelitian ini akan dilakukan pembahasan mengenai bagaimana implementasi dari *JasperReports* pada pengembangan sebuah sistem informasi, beserta hasil yang didapatkan dari implementasi tersebut.



BAB III METODOLOGI MAGANG

3.1 Manajemen Proyek

Proses pengembangan aplikasi SIMPeL melalui beberapa tahapan, yaitu: persiapan, *requirement analysis*, *build application/prototype*, pengujian, implementasi, tinjauan pasca implementasi, serah terima, *transfer knowledge*, serah terima pekerjaan, dan yang terakhir adalah masa garansi. Tahapan tersebut dilalui agar proses pengembangan perangkat lunak dapat berjalan dengan terorganisir sejak awal proyek diterima hingga diserahkan secara keseluruhan kepada klien.

3.1.1 Tahap Persiapan

Pengembangan aplikasi SIMPeL dimulai dengan melakukan koordinasi bersama perwakilan Ombudsman Republik Indonesia selaku klien, melakukan mobilisasi personil, juga membuat dokumen-dokumen administratif yang diperlukan selama proses pengembangan.

3.1.2 Requirement Analysis

Requirement Analysis merupakan tahap di mana kebutuhan-kebutuhan untuk aplikasi SIMPeL didefinisikan. Pada tahap ini, dilakukan pengumpulan dan analisis spesifikasi kebutuhan bisnis dan sistem informasi secara rinci. Analisis juga dilakukan dengan calon pengguna untuk mendapatkan spesifikasi mengenai sistem yang diharapkan. Beberapa hal yang didefinisikan spesifikasinya pada tahap *requirement analysis* adalah *hardware*, *software* dan *brainware* yang akan terhubung dengan sistem yang diinginkan. Hasil dari *requirement analysis* terkait dengan spesifikasi teknis aplikasi SIMPeL dapat dilihat pada **Tabel 3.1**. Kemudian, pada tahap *requirement analysis* juga dilakukan perancangan database, arsitektur aplikasi, arsitektur API, dan desain tampilan antar muka aplikasi.

Tabel 3.1 Spesifikasi teknis aplikasi SIMPeL

Aspek	Spesifikasi
Database	PostgreSQL v10.1 / MySQL v8.0 dan menggunakan middleware berbentuk API

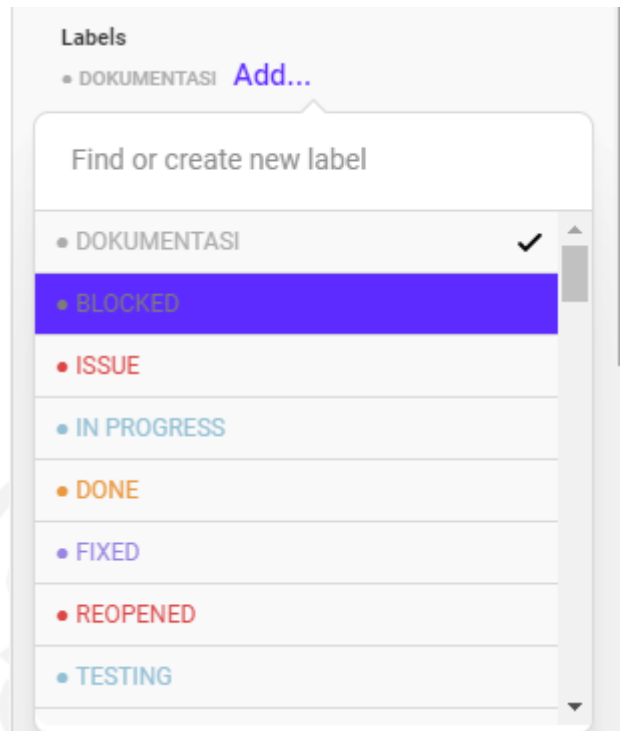
Sistem	<ul style="list-style-type: none"> • Berbasis website • Bahasa pemrograman PHP minimal versi 7.0 • Web framework laravel versi 5.8 • CSS framework yang mengadopsi responsive design • Memiliki fungsi export data ke Microsoft Excel, Microsoft Word dan PDF
Jaringan	Mendukung tipe jaringan client-server
Keamanan	<ul style="list-style-type: none"> • Autentifikasi pengguna • Tingkatan pengguna • Backup/restore database • User log

3.1.3 Build Application/ Prototype

Pada tahap *build application/prototype*, aplikasi SIMPeL mulai dikembangkan oleh *programmer* sesuai dengan hasil perancangan dan spesifikasi yang telah dianalisis sebelumnya. Pada tahap ini, *programmer* akan berada di bawah supervisi analis, menuliskan *code* sesuai dengan skenario dan antarmuka yang telah dirancang. Untuk menjamin penulisan kode perangkat lunak bisa selesai tepat waktu dan sesuai dengan kebutuhan pengguna, pengembang secara berkala berkomunikasi dengan pengguna untuk kemudian diimplementasikan ke dalam *code*. Setiap harinya, terdapat target fungsi atau fitur apa saja yang harus diselesaikan oleh seorang *programmer*. Dalam mendukung tahapan *build application/prototype*, pengembang menggunakan beberapa *tools* berikut ini:

ActiveCollab

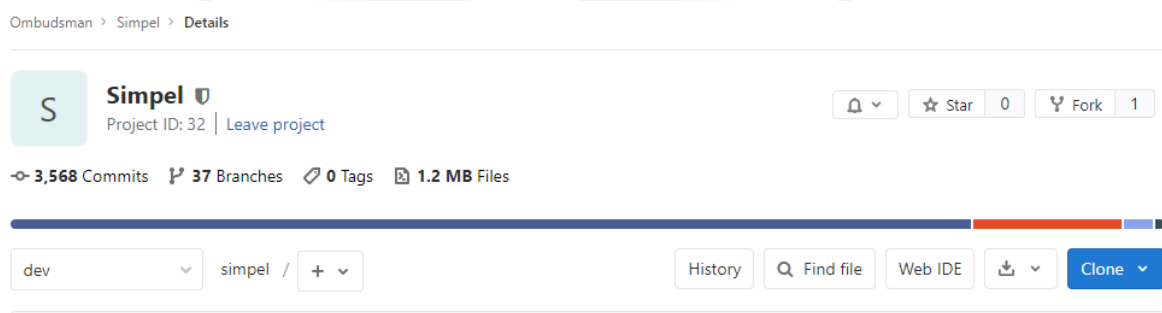
ActiveCollab merupakan sebuah *software* untuk manajemen proyek yang dapat diakses secara *online*. *ActiveCollab* dapat menampilkan daftar *task* yang harus dikerjakan oleh pengembang dalam proses pengembangan aplikasi SIMPeL. Selain itu, *ActiveCollab* juga menyediakan fitur pemberian *label* pada *task* yang memungkinkan dilakukannya *tracking* terhadap status pengerjaan *task* yang dimiliki oleh setiap personil pengembang. Bentuk fitur label pada *ActiveCollab* dapat dilihat pada **Gambar 3.1**. Dengan adanya *label* pada *ActiveCollab*, *programmer* dapat melihat daftar *task* yang belum dan harus dikerjakan, kemudian *tester* dapat menguji *task* yang sudah dikerjakan oleh *programmer*. Dalam pengembangan aplikasi SIMPeL, *ActiveCollab* tidak hanya diakses oleh pengembang, namun juga perwakilan dari klien untuk memantau proses pengembangan aplikasi SIMPeL.



Gambar 3.1 Fitur label pada *ActiveCollab* yang menunjukkan status pengerjaan *task*

Git

Git adalah aplikasi yang memiliki fungsi untuk melakukan *source versioning* dan memudahkan kolaborasi antar *programmer* dalam menyatukan kode aplikasi. Perwakilan dari klien juga mendapatkan akses *git* sehingga dapat menyimpan kode aplikasi. Pengembangan SIMPeL menggunakan *Gitlab* untuk mendukung hal tersebut, seperti yang ditunjukkan pada **Gambar 3.2**.



Gambar 3.2 *Repository Gitlab* pada proses pengembangan SIMPeL

Mattermost

Mattermost merupakan sebuah aplikasi *chatting* untuk melakukan komunikasi secara *real-time*. Semua personil pengembang dapat berkomunikasi atau melakukan koordinasi

melalui aplikasi tersebut. Dengan penggunaan *mattermost*, diharapkan pengembangan aplikasi SIMPeL dapat berjalan dengan baik karena adanya kelancaran komunikasi antar pengembang.

3.1.4 Pengujian

Pengujian merupakan tahap dimana aplikasi yang telah dibangun diuji dengan beberapa *test case* untuk memastikan bahwa tidak ada *bug* atau *error* pada aplikasi. Target dari pengujian aplikasi SIMPeL adalah untuk menunjukkan bahwa aplikasi yang telah dibangun layak dan memenuhi persyaratan. Target kedua adalah untuk menemukan kekurangan pada aplikasi, bisa berupa *bug* atau *error* atau terdapat fitur yang belum sesuai dengan keinginan klien. Dalam pengembangan aplikasi SIMPeL, digunakan 3 jenis metode pengujian, yaitu:

Unit Testing

Unit testing merupakan jenis pengujian dimana *programmer* menulis *test case* tertentu pada kode sebuah kelas atau fungsi tertentu. Kemudian, kode berisi *test case* yang telah dibuat akan dijalankan secara otomatis untuk memeriksa apakah kode mengenai kelas dan fungsi pada aplikasi SIMPeL lolos dari pengujian *unit testing*.

System Integration Test (SIT)

SIT merupakan jenis pengujian yang dilakukan dengan menggunakan lingkungan teknis di Ombudsman untuk memastikan semua fitur yang telah dikembangkan dapat bekerja sesuai harapan dan tidak menimbulkan konflik terhadap aplikasi maupun sistem lain yang telah ada.

User Acceptance Test (UAT)

User Acceptance Test merupakan proses pengujian yang melibatkan pengguna dan *stakeholder* aplikasi untuk menguji dan memverifikasi bahwa perangkat lunak yang dibangun telah memenuhi kebutuhan dan menghasilkan keluaran yang sesuai dengan harapan. Pengembang menyiapkan skenario pengujian berupa langkah-langkah berupa contoh kasus dengan *spreadsheet*. Langkah-langkah tersebut harus dijalankan pada aplikasi SIMPeL dan memeriksa apakah keluaran yang didapat sama dengan yang diharapkan. Contoh UAT aplikasi SIMPeL dapat dilihat pada **Gambar 3.3**.

No Case	Objektif	Type Case	Langkah Pengujian / Input	Hasil Yang Diharapkan	Hasil Aktual	Status by Javan	Status by User	Status [dd/mm/yyyy]	Catatan User (14-17 April)
TS-PVL001	TC-PVL001	Login	Normal Case	1. Akses ke halaman ombudsman.go.id/auth/login 2. Klik field "Email/Username" 3. Input email/username yang valid 4. Klik field "Password" 5. Input password yang valid (seoret) 6. Klik tombol "Login"	Berhasil masuk ke dalam sistem	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
TS-PVL002	TC-PVL002	List Registrasi Laporan	Normal Case	1. Login 2. Klik menu "Registrasi Laporan"	- Tampil list laporan yang sudah ada	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	TC-PVL003	Registrasi Laporan	Normal Case	1. Login 2. Klik menu "Registrasi Laporan" 3. Klik tombol "Beriinisi" 4. Pilih Tipe Laporan 5. Pilih Cara Penyempulan 6. Klik tombol "Simpan"	- Tampil form Laporan sesuai dengan Tipe Laporan yang dipilih - Tanggal terisi hari ini	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	TC-PVL004	Laporan Masyarakat	Normal Case	1. Pilih Tipe Laporan "Laporan Masyarakat" 2. Pilih Cara Penyempulan 3. Klik tombol "Simpan" 4. Input data laporan 4.1 Is Substansi 4.2 Is Pihak Pemasttashan 4.3 Is Perihal 5. Is Data Informasi Pelapor 6. Is Data Informasi Terlapor 7. Klik tombol "Simpan sebagai Draft" (jika belum yakin dan memungkinkan akan melubun kembali datanya) 8. Klik tombol "Simpan" yakin akan dilanjutkan ke process selanjutnya	- Dapat disimpan - Tampil form Input Laporan Masyarakat - Berhasil menyimpan data laporan masyarakat - Laporan ke list Verifikasi Form	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	TC-PVL005	Respon Cepat	Normal Case	1. Pilih Tipe Laporan "Respon Cepat" 2. Pilih Cara Penyempulan 3. Klik tombol "Simpan" 4. Input data laporan 4.1 Pemberi Disposisi 4.2 Alasan Respon Cepat 4.3 Tanggal Disposisi 5. Is data laporan 6. Is Data Informasi Pelapor 7. Is Data Informasi Terlapor 8. Klik tombol "Simpan sebagai Draft" (jika belum yakin dan kemu 9. Klik tombol "Simpan" yakin akan dilanjutkan ke process selanjutnya	- Tampil form Input Respon Cepat - Data berhasil tersimpan - Laporan ke list Verifikasi Form	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	TC-PVL006	Insiafat	Normal Case	1. Klik Tipe Laporan "Insiafat" 2. Pilih Cara Penyempulan 3. Klik tombol "Simpan" 4. Input data laporan 4.1 Is Substansi 4.2 Is Pihak Pemasttashan 4.3 Is Perihal 5. Is Data Informasi Pelapor 6. Is Data Informasi Terlapor 7. Klik tombol "Simpan sebagai Draft" (jika belum yakin dan kemu	- Tampil form Input Insiafat - Data berhasil tersimpan - List ke Disposisi di RKSA	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Gambar 3.3 Contoh *scenario test* modul PVL SIMPeL

3.1.5 Implementasi

Implementasi sistem dapat dilakukan apabila sistem yang dikembangkan telah sesuai dengan rencana kebutuhan pengguna dan bebas dari *bug* atau *error*. Tahapan implementasi merupakan proses penerapan sistem informasi yang telah dikembangkan pada lingkungan operasional. Pada tahap ini dilakukan instalasi aplikasi SIMPeL di *production server* dan jaringan infrastruktur yang ada.

3.1.6 Tinjauan Pasca Implementasi

Tahap ini merupakan proses evaluasi yang dilaksanakan sebagai bahan pembelajaran untuk pengembangan sistem informasi selanjutnya. Pada tahapan ini, pengembang melakukan *monitoring* terhadap aktivitas *server* dengan adanya modul yang baru, yang apabila ditemukan ketidakwajaran, maka akan langsung ditindaklanjuti. Kegiatan lain adalah mengumpulkan umpan balik dari pengguna tentang modul yang telah dibuat. Umpan balik yang sifatnya mendesak akan langsung ditindaklanjuti, sedangkan umpan balik lain akan dianalisa terlebih dahulu untuk menindaklanjutinya.

3.1.7 Transfer Knowledge

Setelah kegiatan pengembangan selesai, pihak pengembang menyelenggarakan pertemuan dan pembahasan dalam rangka alih pengetahuan tentang sistem informasi yang dikembangkan kepada klien dan calon pengguna, termasuk tim teknis di Ombudsman Republik Indonesia. Pada kegiatan ini, pengguna akan dilatih menggunakan *web* untuk latihan, sehingga diharapkan

aplikasi bisa dimanfaatkan untuk meningkatkan kinerja di lingkungan instansi. Teknis dalam kegiatan pelatihan dan informasi adalah sebagai berikut:

- a. Melakukan pelatihan cara instalasi dan penggunaan kepada pengguna;
- b. Melakukan pelatihan operasional kepada staf sesuai dengan modul aplikasi.

Tujuan dari *transfer knowledge* adalah pengguna dapat memahami dan mempraktekkan tata cara pengoperasian dari aplikasi SIMPeL yang telah dikembangkan. Kegiatan ini harus dilaksanakan untuk memastikan bahwa pengguna aplikasi dan pihak-pihak yang terkait, seperti *administrator* dan *support helpdesk* memahami cara kerja aplikasi dan hal-hal apa saja yang harus dilakukan terkait dengan pemeliharaan dan penanggulangan jika terjadi kesalahan.

Proses pelatihan dilakukan dengan cara *workshop*. Pemateri dari pelaksana kegiatan akan memaparkan dan menjelaskan bagaimana cara menggunakan aplikasi kepada peserta pelatihan. Proses alih pengetahuan dilakukan dengan cara pembuatan dokumen yang berisi rancangan, implementasi dan manual penggunaan aplikasi.

3.1.8 Serah Terima Pekerjaan

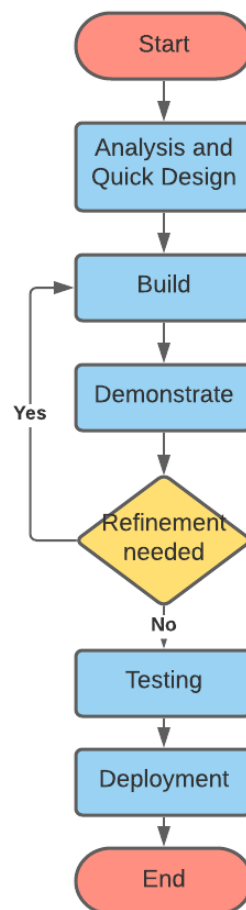
Serah terima pekerjaan dilakukan setelah aplikasi berjalan di lingkungan *production* dengan baik. Serah terima dibuktikan dengan adanya berita acara yang ditandatangani oleh kedua belah pihak antara pengembang dan klien.

3.1.9 Garansi

Pada tahapan kegiatan ini, pembuatan sistem sudah final dan dilakukan pendampingan. Garansi dan pemeliharaan aplikasi SIMPeL adalah sejak waktu proyek selesai hingga bulan Desember tahun 2021. Dengan adanya tahapan ini, jika terdapat masalah atau ditemukan keadaan yang tidak sesuai dengan rencana kebutuhan yang telah disusun, masalah tersebut dapat segera diatasi. Penyedia akan melakukan perbaikan aplikasi apabila ditemukan *bug* atau error yang masih dalam ruang lingkup pekerjaan di awal *requirement*. Namun, apabila terdapat perubahan ruang lingkup atau proses bisnis di luar dari fitur yang telah disepakati pada awal kontrak, maka harus dilakukan *change request* sehingga penyedia akan mengeluarkan penawaran baru.

3.2 Metode Pengembangan Perangkat Lunak SIMPeL yang Mengimplementasikan JasperReports

Secara keseluruhan, SIMPeL dikembangkan dengan metode Rapid Application Development (RAD). Sebuah proyek yang menerapkan metode RAD umumnya berskala relatif kecil dan berdurasi pendek, yaitu 2 sampai 6 bulan (Beynon-Davies et al., 1999). Hal ini bersesuaian dengan harapan pengguna yang menginginkan proyek dapat selesai dalam waktu 6 bulan. Metode RAD memiliki kemiripan dengan metode *Waterfall*, hanya saja waktu yang dibutuhkan untuk pengerjaan proyek lebih singkat. Secara garis besar, metode RAD memiliki 4 tahapan dalam proses pengembangannya, seperti yang diperlihatkan pada **Gambar 3.4**.



Gambar 3.4 Alur proses metode *Rapid Application Development* (RAD)

3.2.1 *Analysis and Quick Design*

Analisis merupakan proses identifikasi tentang tujuan sistem dan kebutuhan sistem dengan mengumpulkan fakta-fakta yang diperlukan. Selain analisis, dilakukan juga pembuatan rancangan sistem (API, basis data, dan antarmuka sistem). Pada tahapan ini juga, diputuskan

untuk menggunakan *JasperReports* sebagai *reporting tools* untuk proses pembuatan surat SIMPeL. *JasperReports* telah lama digunakan di lingkungan perusahaan pengembang. Alasan utama digunakannya *JasperReports* adalah sifatnya yang *open-source* dan gratis. Selain itu, sebuah penelitian membahas mengenai perbandingan *JasperReports* dengan *reporting engine* lain seperti *The Business Intelligence and Reporting Tools* (BIRT) dan *Crystal Report* dari aspek komparabilitas, performa, dan kegunaan. Dari penelitian tersebut disimpulkan bahwa *JasperReports* unggul dalam ketiga aspek yang diuji, sementara kedua *reporting engine* lain memiliki beberapa kelemahan, seperti BIRT yang kurang dalam aspek performa dan *Crystal Report* yang kurang dalam aspek kegunaan dan komparabilitas. Kemudahan pembuatan *report* dengan *layout* yang ada pada *JasperReports* memungkinkan *tools* tersebut dapat digunakan oleh orang awam (Adhi et al., 2019).

3.2.2 *Prototype Cycles*

Tahapan selanjutnya adalah pengembangan dan pengimplementasian hasil analisis yang diperoleh dari tahapan sebelumnya. *Prototype Cycles* terdiri dari 3 sub tahapan, yaitu:

- a. *Build*: Proses ini mengimplementasikan hasil analisis ke dalam *code* yang mana tahap ini dikerjakan oleh *programmer*. Dalam implementasi *JasperReports* pada aplikasi SIMPeL, sub tahapan *build* adalah proses pembuatan surat, mulai dari proses instalasi *JasperReports Server* pada development server aplikasi SIMPeL, penggunaan *Jaspersoft Studio* dalam hal konfigurasi *JasperReports Server* dan *database* SIMPeL, proses desain surat, pengambilan data dari *database*, serta proses *publish* surat ke *JasperReports Server*.
- b. *Demonstrate*: Hasil dari *build* ditunjukkan kepada calon pengguna untuk mendapat umpan balik tentang *prototype* yang telah dibuat. Apabila calon pengguna puas dengan hasil dari tahap *build* maka proses langsung dilanjutkan ke tahap *testing*. Untuk pembuatan surat, peran calon pengguna pada sub tahapan dilakukan oleh analis.
- c. *Refine*: Proses ini memperbaiki hasil dari *build* apabila terdapat hal-hal yang tidak sesuai dengan penilaian analis.

Pada tahapan *Prototype Cycles* dalam proses pengembangan SIMPeL dengan *JasperReports*, diawali dengan pembuatan desain dan *query* surat hingga *publishing* surat tersebut ke *JasperReports Server*. Kegiatan tersebut dilakukan oleh sekitar 3 hingga 5 *programmer*. Setiap *programmer* diberikan *task* untuk mengerjakan surat yang berbeda. Ketika 1 surat selesai dibuat dan di-*publish*, surat tersebut akan diperiksa oleh analis dan ini termasuk

dalam sub tahapan *demonstrate*. Kemudian, apabila surat yang diperiksa oleh analis ternyata belum sesuai dengan *requirement*, maka surat tersebut akan kembali ke proses *build* untuk diperbaiki. Siklus tersebut terus berlangsung hingga surat benar-benar telah sesuai dengan *requirement* dan rata-rata untuk 1 surat berlangsung sebanyak 3 hingga 4 kali.

3.2.3 Testing

Tahap ini adalah tahap dilakukannya pengujian terhadap aplikasi yang dikembangkan guna memastikan aplikasi telah sesuai dengan kebutuhan. *Testing* biasanya dilakukan secara manual atau dengan *automated testing*. Pengujian secara manual dilakukan oleh tester terhadap aplikasi secara manual tanpa menggunakan suatu *tool*. Sedangkan *automated testing* dilakukan dengan *scenario test* yang berbentuk kode. Aplikasi yang telah melewati tahap *testing* dan bebas dari *bug* berarti aplikasi tersebut siap untuk di-*deploy*. Surat yang telah berhasil dibuat, diuji melalui *development web* aplikasi SIMPeL dengan melihat satu per satu kesesuaian surat dengan *requirement*.

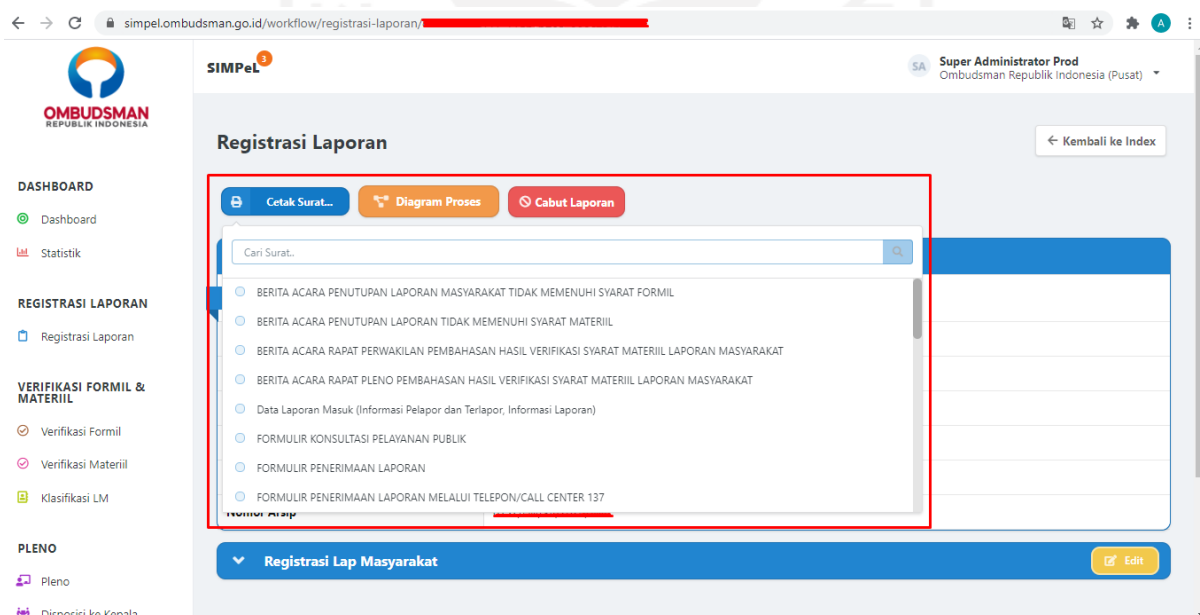
3.2.4 Deployment

Deployment dilakukan apabila aplikasi yang dikembangkan telah sesuai dengan rencana kebutuhan pengguna dan bebas dari *bug*. Pada tahap ini dilakukan instalasi aplikasi di *production server* dan jaringan infrastruktur yang ada (Biesemans et al., 2010). Ketika proses ini dilakukan, JasperReports yang telah berisi semua surat yang siap digunakan juga di-*install* di *production server* aplikasi SIMPeL.

BAB IV

HASIL MAGANG

Teknologi *JasperReports* yang diimplementasikan dalam aplikasi SIMPeL terdapat pada fitur cetak surat. Fitur cetak surat sendiri merupakan sebuah fitur penting dalam aplikasi SIMPeL yang dapat dijumpai pada setiap tahapan dalam alur penyelesaian laporan masyarakat. Dalam implementasinya, *JasperReports* menyediakan *JasperReports Server* untuk mengakses *report* yang telah dibuat dengan melakukan komunikasi melalui REST API. REST API tersebut memungkinkan keluaran dokumen berupa PDF dan DOCX pada aplikasi SIMPeL. *JasperReports* bekerja dari sisi *backend* aplikasi SIMPeL. Pembuatan surat oleh pengembang lebih ke sisi desain surat. Data yang telah di-*input* oleh pengguna pada aplikasi SIMPeL nantinya dapat ditampilkan pada surat yang ingin dicetak oleh pengguna. Pengguna dapat mencetak surat pada setiap tahapan dalam alur proses penyelesaian laporan seperti yang ditunjukkan pada **Gambar 4.1**. Surat yang dipilih oleh pengguna akan muncul dalam bentuk PDF pada *browser*.

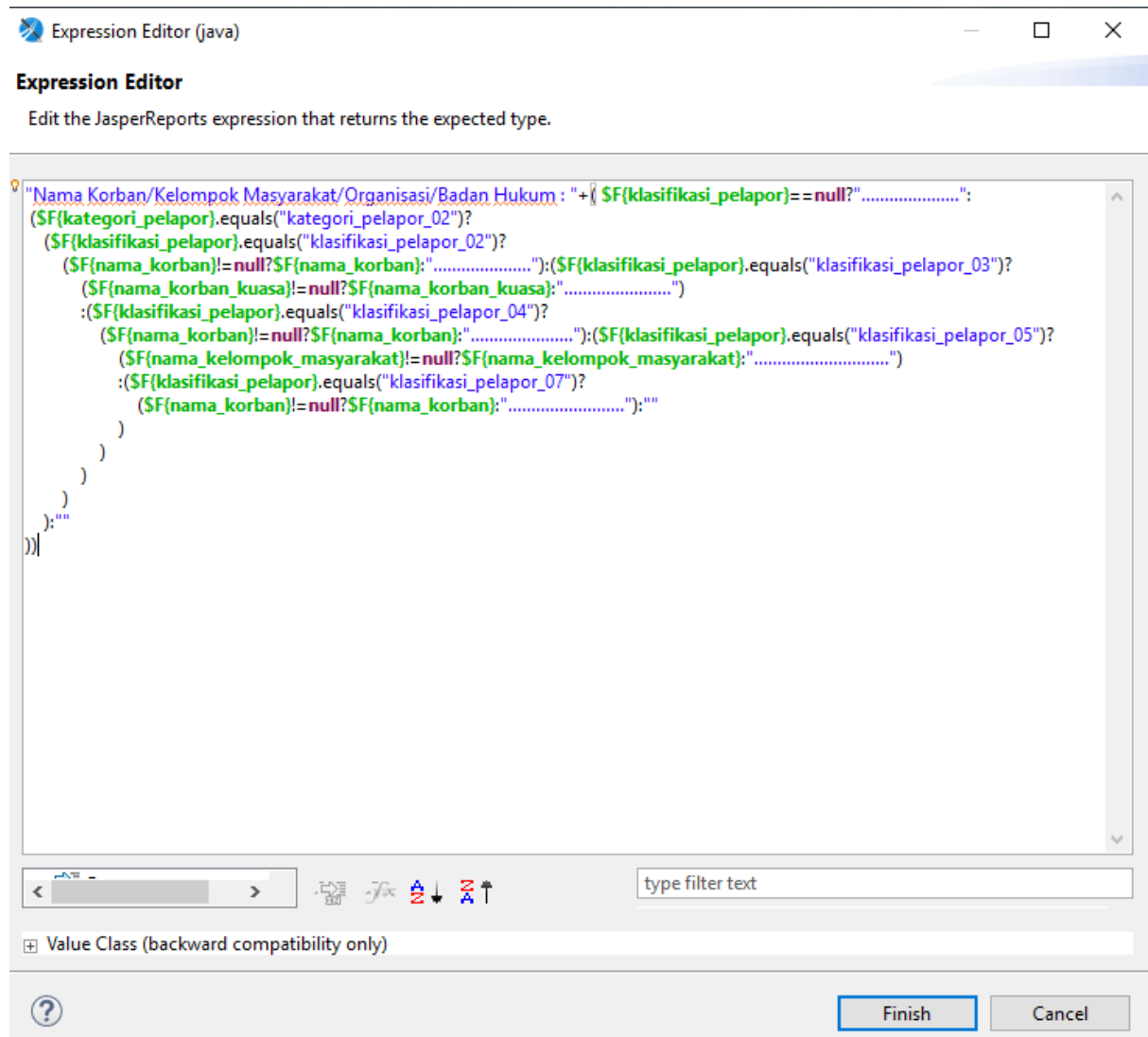


Gambar 4.1 Fitur cetak surat pada aplikasi SIMPeL

Dalam proses pembuatan surat untuk aplikasi SIMPeL, antar anggota tim pengembang selalu melakukan koordinasi. Salah satu *tool* yang digunakan untuk mendukung proses tersebut adalah *ActiveCollab* yang merupakan sebuah *software* untuk manajemen proyek. Pada proses awal, analis mendefinisikan daftar *task* untuk ditujukan pada setiap anggota tim melalui

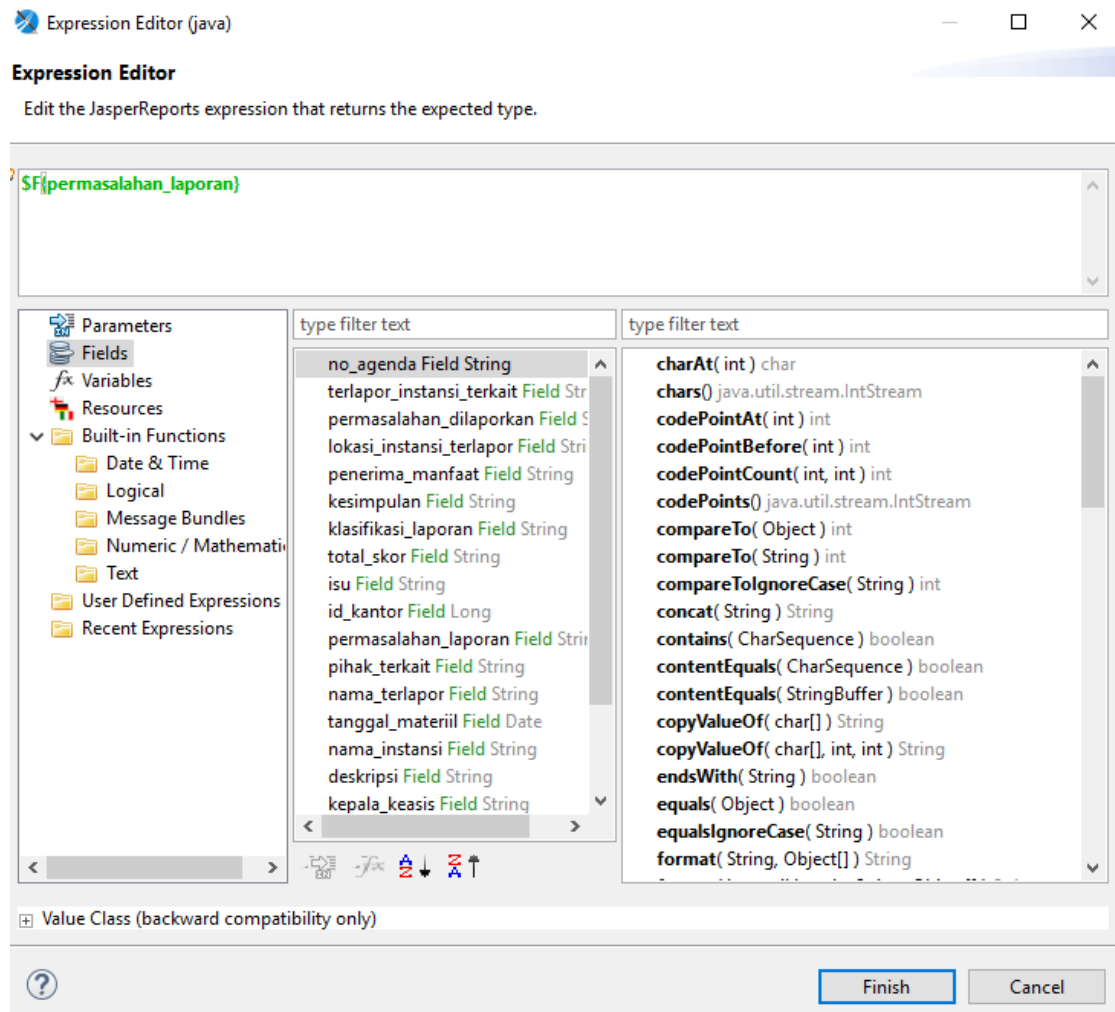
ActiveCollab. *Programmer* akan membuat surat berdasarkan daftar *task* yang diberikan oleh analis kepada *programmer* tersebut. Satu *task* biasanya berisi satu surat dan dikerjakan oleh satu orang. Ketika surat untuk *task* tersebut selesai dibuat, *programmer* akan menandai *task* tersebut dengan label “Ready to Test” agar hasilnya dapat diperiksa oleh analis. Apabila surat telah sesuai dengan *requirement*, status *task* akan diubah menjadi “Done”, sedangkan jika diperlukan perbaikan, *task* tersebut akan kembali di-*assign* kepada *programmer*. Selain itu, *Telegram* dan *Google Meet* juga digunakan untuk melakukan diskusi apabila ditemui masalah dalam mengerjakan sebuah *task*.

Dari pembuatan beberapa surat yang dikerjakan oleh penulis, surat-surat tersebut dapat dikategorikan menjadi 2 macam berdasarkan kompleksitas pada elemen *text field*-nya. Kategori pertama adalah surat yang elemen *text field* nya tidak memerlukan banyak logika percabangan, dan yang kedua adalah surat yang memerlukan banyak logika percabangan. Salah satu contoh surat yang memerlukan banyak logika percabangan adalah surat Verifikasi Formil. Pada surat tersebut, terdapat *text field* yang memuat nama korban pada laporan masyarakat. Korban pada laporan masyarakat memiliki banyak kategori dan klasifikasi, karena itu pada *text field* tersebut harus digunakan logika percabangan yang rumit seperti ditunjukkan pada **Gambar 4.2**. Pada gambar tersebut, \$F mendefinisikan *field* setelah pengambilan *query*.



Gambar 4.2 Contoh elemen *text field* dengan logika percabangan yang rumit

Kategori berikutnya adalah surat yang elemen *text field*-nya tidak memerlukan logika percabangan seperti pada surat Klasifikasi Laporan. Pada surat tersebut terdapat *text field* yang berisi deskripsi mengenai masalah yang dilaporkan. Untuk kasus tersebut, tidak ada kategori dan hanya ada satu data yang pasti. Oleh karena itu tidak diperlukan logika percabangan pada elemen *text field*-nya seperti yang ditunjukkan pada **Gambar 4.3**.



Gambar 4.3 Contoh elemen *text field* tanpa logika percabangan

Selanjutnya, jika pengerjaan surat aplikasi SIMPeL dibagi berdasarkan metode pengembangan perangkat lunak yang digunakan pada pengembangan aplikasi SIMPeL, yaitu *Rapid Application Development* (RAD), maka tahapan-tahapan dalam pembuatan surat aplikasi SIMPeL adalah sebagai berikut:

4.1 *textAnalysis and Quick Design*

Aplikasi SIMPeL mulai dikerjakan sejak bulan Oktober 2019 dengan lama proyek 6 bulan ditambah masa garansi hingga bulan Desember tahun 2021. Ketika pengembangan aplikasi SIMPeL berada di tahap awal, yaitu *analysis and quick design*, pengembang melakukan identifikasi mengenai surat secara keseluruhan, khususnya desain. Desain surat telah ditentukan oleh Ombudsman Republik Indonesia selaku klien dengan jumlah 57 surat untuk modul Penerimaan dan Verifikasi Laporan (PVL), 37 surat untuk modul Pemeriksaan (Riksa), dan 56 surat untuk modul Resolusi dan *Monitoring* (Resmon), sehingga totalnya adalah 150

surat. Selama kegiatan magang, jumlah dan jenis surat yang *layout*-nya didesain oleh penulis adalah sekitar 30 hingga 40, sementara untuk *query* yang dibangun dari awal adalah untuk 2 surat. Terdapat 20-an surat yang diperbaiki, baik dari sisi *layout* yang kurang tepat dan atau perbaikan *query*. Contoh desain “Surat Pemberitahuan Dimulainya Pemeriksaan” untuk modul PVL yang disediakan oleh Ombudsman Republik Indonesia dapat dilihat pada **Gambar 4.4**.

CONTOH FORMAT 1
 SURAT PEMBERITAHUAN
 DIMULAI PEMERIKSAAN



OMBUDSMAN REPUBLIK INDONESIA
 Disesuaikan alamat kantor

Tgl...Bln...Th...

Nomor :
 Lampiran : +
 Perihal : Dimulainya Pemeriksaan

Yth.
 Di

Dengan Hormat,

Bersama ini kami beritahukan bahwa Ombudsman Republik Indonesia/ Ombudsman Republik Indonesia Perwakilan Provinsi* ... telah menerima Laporan Saudara, terdaftar dalam nomor register: ... (nomor register), ... (tanggal register). Saat ini laporan dimaksud telah masuk tahap pemeriksaan substantif oleh Tim ... (Tim substansi di Jakarta atau Tim Pemeriksaan di Perwakilan).

Demikian atas perhatian Saudara, kami ucapkan terima kasih.

KOORDINATOR UNIT KERJA PEMERIKSAAN



NAMA LENGKAP

Gambar 4.4 Desain “Surat Pemberitahuan Dimulainya Pemeriksaan” dari Ombudsman Republik Indonesia

4.2 Prototype Cycles

4.2.1 Konfigurasi *Jasperreports Server* pada *Jaspersoft Studio*

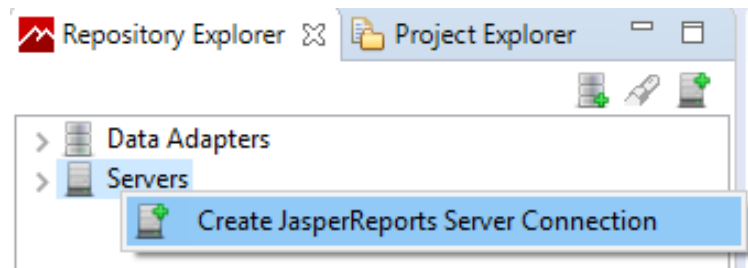
Proses pengembangan fitur cetak surat yang mengimplementasikan *JasperReports* pada aplikasi SIMPeL diawali dengan instalasi *JasperReports Server* pada development server aplikasi SIMPeL. *JasperReports Server* dapat diakses setelah komputer terhubung dengan *port* SIMPeL dan melakukan *log-in* dengan mengakses `http://<host>:<port>/jasperserver[-pro]/` pada *browser*. Semua surat untuk aplikasi SIMPeL harus tersimpan di dalam sebuah folder pada *JasperReports Server* seperti pada **Gambar 4.5**. Folder untuk menyimpan surat dapat dibuat dengan menambahkan folder baru di bawah folder JRXML. Dalam proses pengembangan SIMPeL, *JasperReports Server* paling sering diakses untuk melihat *preview* surat atau mengedit properti surat, seperti penamaan surat, juga untuk mengambil *path* surat tertentu dan menyalinnya ke aplikasi SIMPeL untuk nantinya surat tersebut didaftarkan pada aplikasi dan dapat ditampilkan saat menggunakan fitur cetak surat.



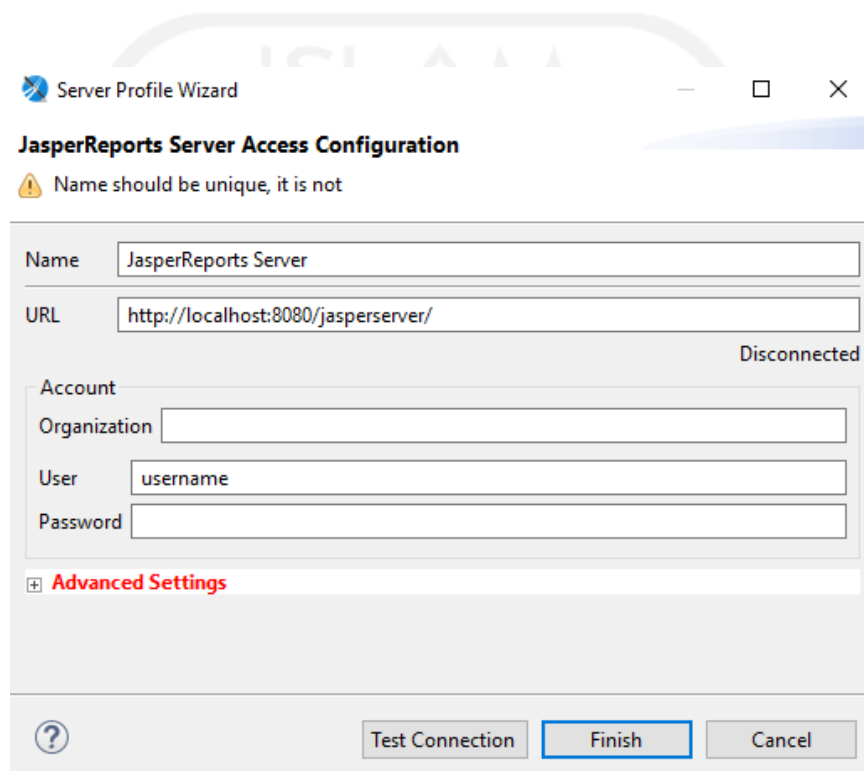
Gambar 4.5 Folder yang menyimpan surat SIMPeL di *JasperReports Server*

Setelah *JasperReports Server* terpasang di aplikasi SIMPeL, aktivitas selanjutnya adalah membuat surat dengan *Jaspersoft Studio*, yang mana pengembang menggunakan versi 6.10.0. *Jaspersoft Studio* dapat dipasang secara gratis pada berbagai macam sistem operasi komputer. Sebelum memasuki proses mendesain, hal pertama yang harus dilakukan saat membuka *Jaspersoft Studio* adalah menghubungkan *software* tersebut dengan *JasperReports Server* yang telah terpasang di *development server* aplikasi SIMPeL. Proses menghubungkan *Jaspersoft Studio* dengan *JasperReports Server* dilakukan dengan memilih menu “Create JasperReports Server Connection” seperti yang ditunjukkan pada **Gambar 4.6** dan melakukan konfigurasi akses dengan *JasperReports Server* seperti pada **Gambar 4.7**. Proses penghubungan ini

bertujuan untuk mengintegrasikan keduanya sehingga lokasi *repository* dari *JasperReports Server* dapat terbaca pada *Jaspersoft Studio*.



Gambar 4.6 Menu pembuatan koneksi *JasperReports Server*

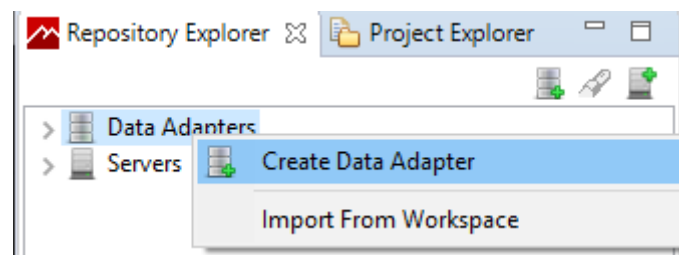


Gambar 4.7 Konfigurasi akses terhadap *JasperReports Server*

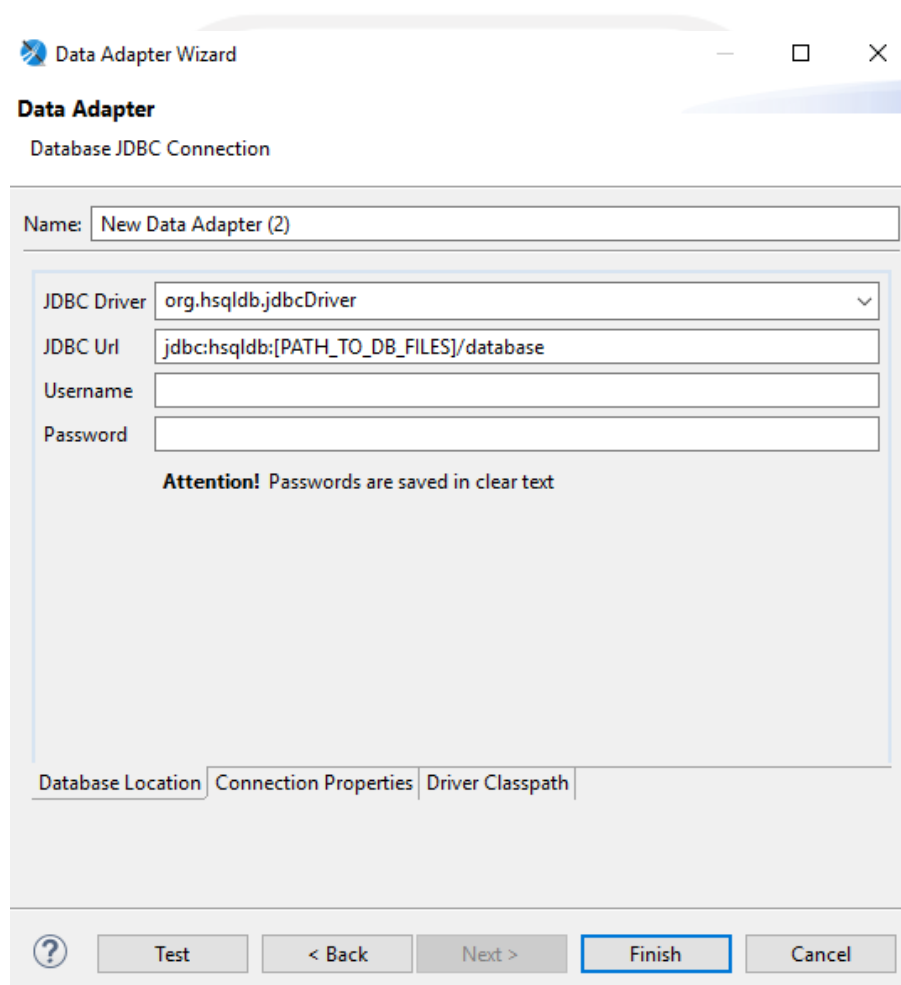
4.2.2 Konfigurasi *Database SIMPeL* pada *Jaspersoft Studio*

Setiap surat pada aplikasi SIMPeL berisi data tentang laporan masyarakat. Data tersebut tersimpan di dalam *database* SIMPeL milik Ombudsman Republik Indonesia, sehingga untuk menampilkan surat dengan data yang sesuai, juga diperlukan koneksi antara *Jaspersoft Studio* dengan *database* SIMPeL. Aktivitas penghubungan antara *database* dan *Jaspersoft Studio* dilakukan dengan menambahkan *JDBC Data Adapter* seperti yang ditunjukkan pada **Gambar 4.8**, kemudian memasukkan informasi mengenai *database* SIMPeL seperti pada **Gambar 4.9**. *Data Adapter* digunakan untuk mendukung pertukaran data antara *data-source* dan *dataset*.

Sementara JDBC dipilih karena jenis adapter tersebut memungkinkan penggunaan *database* yang diakses melalui *JDBC driver*, dalam hal ini merupakan PostgreSQL.



Gambar 4.8 Penambahan *data adapter*

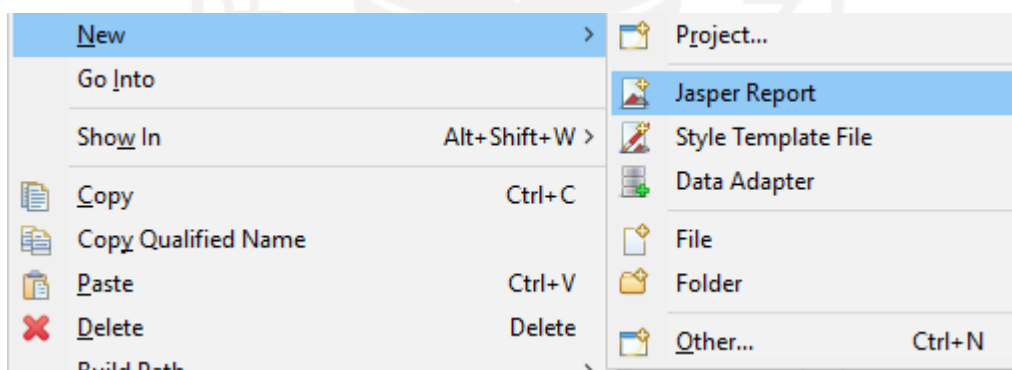


Gambar 4.9 Konfigurasi *database* SIMPeL pada JDBC *data adapter*

4.2.3 Proses Desain Surat pada *Jaspersoft Studio*

Setelah terhubung dengan *JasperReports Server* dan *database* SIMPeL, tahapan pembuatan surat SIMPeL selanjutnya adalah membuat *layout* surat sesuai desain yang telah ditentukan oleh Ombudsman Republik Indonesia. *Layout* surat dibuat dengan menambahkan *file report* baru pada *project explorer Jaspersoft Studio* seperti ditunjukkan pada **Gambar 4.10**,

termasuk memilih ukuran *layout* surat yang akan dibuat dan memberi nama surat. Aktivitas tersebut secara otomatis men-*generate file* dengan ekstensi JRXML yang merupakan ekstensi untuk *file layout report*. Surat SIMPeL menggunakan ukuran *layout* A4, baik *landscape* maupun *portrait* sesuai dengan bentuk yang telah ditentukan oleh Ombudsman Republik Indonesia. *Layout* yang ter-*generate* secara otomatis terdiri dari beberapa bagian, yaitu: *title*, *page header*, *column header*, *detail*, *column footer*, *page footer*, dan *summary*. Tidak semua bagian digunakan dalam surat aplikasi SIMPeL. Bagian yang tidak digunakan dapat dihapus dan apabila nantinya dibutuhkan, bagian tersebut dapat ditambahkan kembali. Bagian yang paling sering digunakan pada surat aplikasi SIMPeL adalah *title* dan *detail*. Bagian yang masuk pada bagian *title* hanya akan muncul di halaman pertama sebanyak satu kali, karena itu bagian *title* akan diisi dengan kop surat, sedangkan isi surat akan diletakkan di dalam bagian *detail*. Hasil desain “Surat Pemberitahuan Dimulainya Pemeriksaan” dapat dilihat pada **Gambar 4.11**. Pada gambar tersebut, surat terbagi menjadi dua bagian, yaitu *title* yang berisi kop surat dan *detail* yang memuat isi surat. Pada kop surat terdapat logo yang menggunakan elemen *image* dan diambil dari *repository JasperReports Server*.



Gambar 4.10 Penambahan *layout* surat baru

Selanjutnya, proses mendesain surat SIMPeL dilakukan dengan melakukan *drag and drop* secara langsung elemen yang sudah disediakan oleh *Jaspersoft Studio*. Selain melakukan *drag and drop*, tersedia juga opsi lain untuk mengedit dengan *source JRXML*. Beberapa elemen yang paling sering digunakan dalam surat SIMPeL adalah *text field*, *static text*, *image*, *frame*, *break*, *rectangle*, dan *line*. Namun, dari beberapa elemen tersebut, yang berhubungan langsung dengan *data-source* adalah *text field* dan *image*. Surat pada aplikasi SIMPeL menggunakan elemen *image* hanya untuk meletakkan logo pada kop surat. Logo diambil dari *repository JasperReports Server*.


OMBUDSMAN REPUBLIK INDONESIA
SF(nama_kantor)
SF(deskripsi_kantor)
Website : www.ombudsman.go.id

no. arsip/absip dinas, dd
MMMM yyyy' row Local

Nomor :
Sifat :
Lampiran :
Hal : Dimulainya Pemeriksaan

th. SF(nama_pelapor)
di SF(alamat_surat)

Dengan Hormat,

Bersama ini kami beritahukan bahwa Ombudsman Republik Indonesia "*(SF id_kantor)=1?Pusat*SF(nama_kantor)*" telah menerima Laporan Saudara, terdaftar dalam nomor register: "*(SF(no_arsip)*"*(SF(tanggal_diterima)*". Saat ini laporan dimaksud telah masuk tahap pemeriksaan substantif oleh "*(SF(id_kantor)=1?Tim subdansi di Jakarta.*Tim Pemeriksaan di Perwakilan.)"

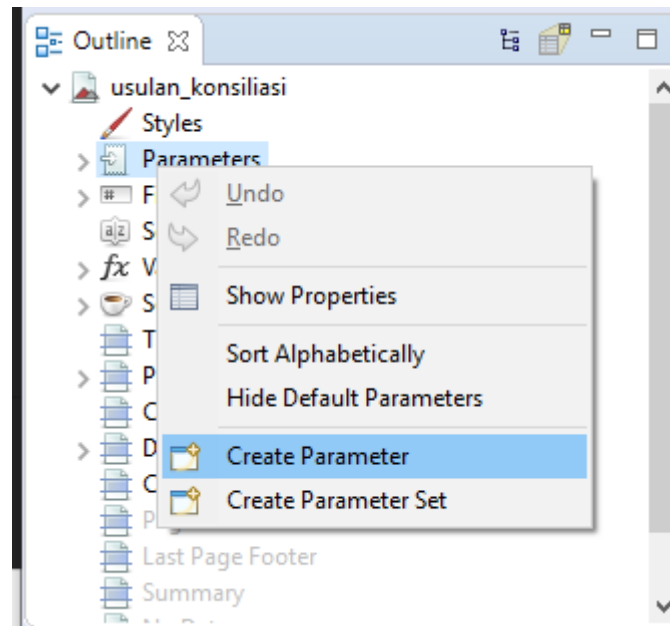
Demikian atas perhatian Saudara, kami ucapkan terima kasih

KOORDINATOR UNIT
KERJA PEMERIKSAAN

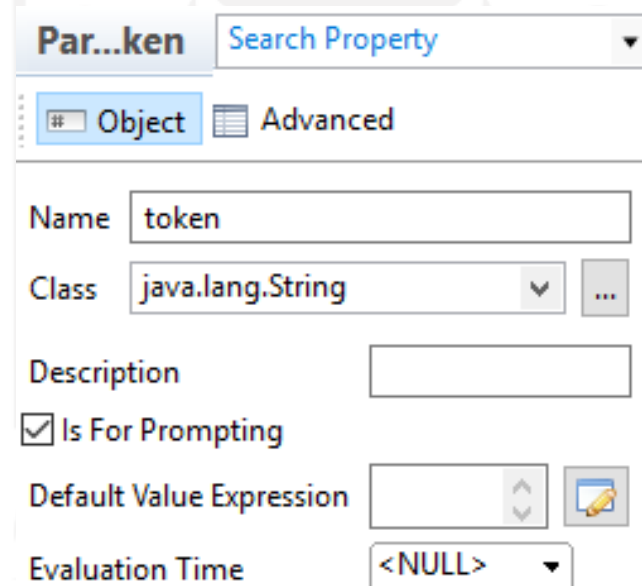
Gambar 4.11 Desain *layout* “Surat Pemberitahuan Dimulainya Pemeriksaan

4.2.4 Pengambilan Data dari *Database* SIMPeL

Data laporan masyarakat di dalam *database* SIMPeL memiliki token untuk membedakan laporan dalam suatu proses bisnis dengan laporan dalam suatu proses bisnis lain. Token otomatis ter-*generate* apabila ada laporan masyarakat baru yang masuk. Token tersebutlah yang mengidentifikasi data tertentu untuk ditampilkan pada surat. Pengambilan data dari *database* sendiri mengharuskan penulisan *query* secara manual. Sebelum itu, surat yang telah dibuat desainnya membutuhkan parameter yang nantinya akan diisi oleh token. Parameter ditambahkan dengan memilih “*Create Parameter*” dan kemudian diberi nama seperti yang ditunjukkan pada **Gambar 4.12** dan **Gambar 4.13**.



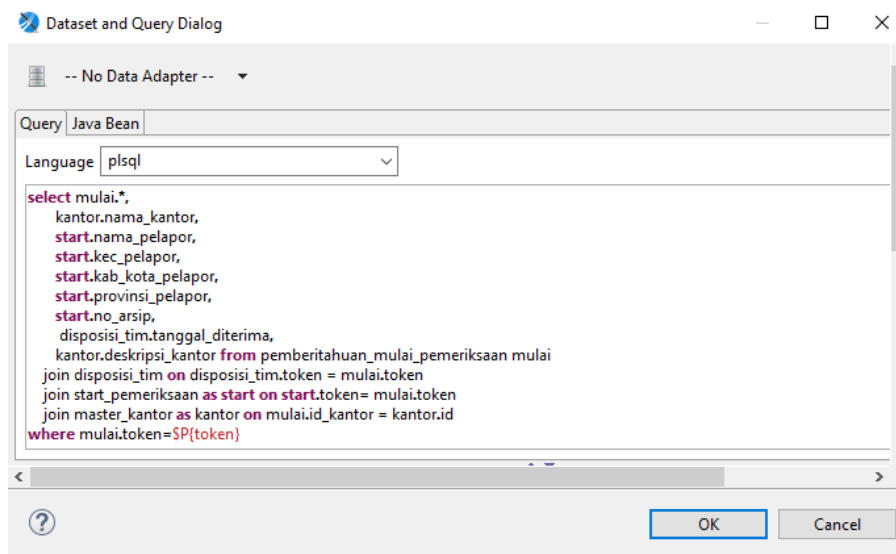
Gambar 4.12 Menu penambahan parameter



Gambar 4.13 Konfigurasi parameter baru

Setelah parameter yang berfungsi untuk menyimpan token dibuat, tahap selanjutnya adalah penulisan *query* pada *Dataset and Query Dialog*. *Query* untuk surat aplikasi SIMPeL dibuat oleh analis dan *programmer*. *Query* ditulis berdasarkan data yang diperlukan oleh surat dengan mencantumkan token sebagai parameternya. Salah satu contoh *query* yang digunakan untuk pengambilan data “Surat Pemberitahuan Dimulainya Pemeriksaan” terdapat pada **Gambar 4.14**. Pada *Dataset and Query Dialog*, terdapat pilihan *data adapter* yang dapat digunakan.

Karena itu, dipilihlah JDBC *data adapter* yang sebelumnya telah dibuat dan dihubungkan dengan *database* SIMPeL.



Gambar 4.14 Query “Surat Pemberitahuan Dimulainya Pemeriksaan”

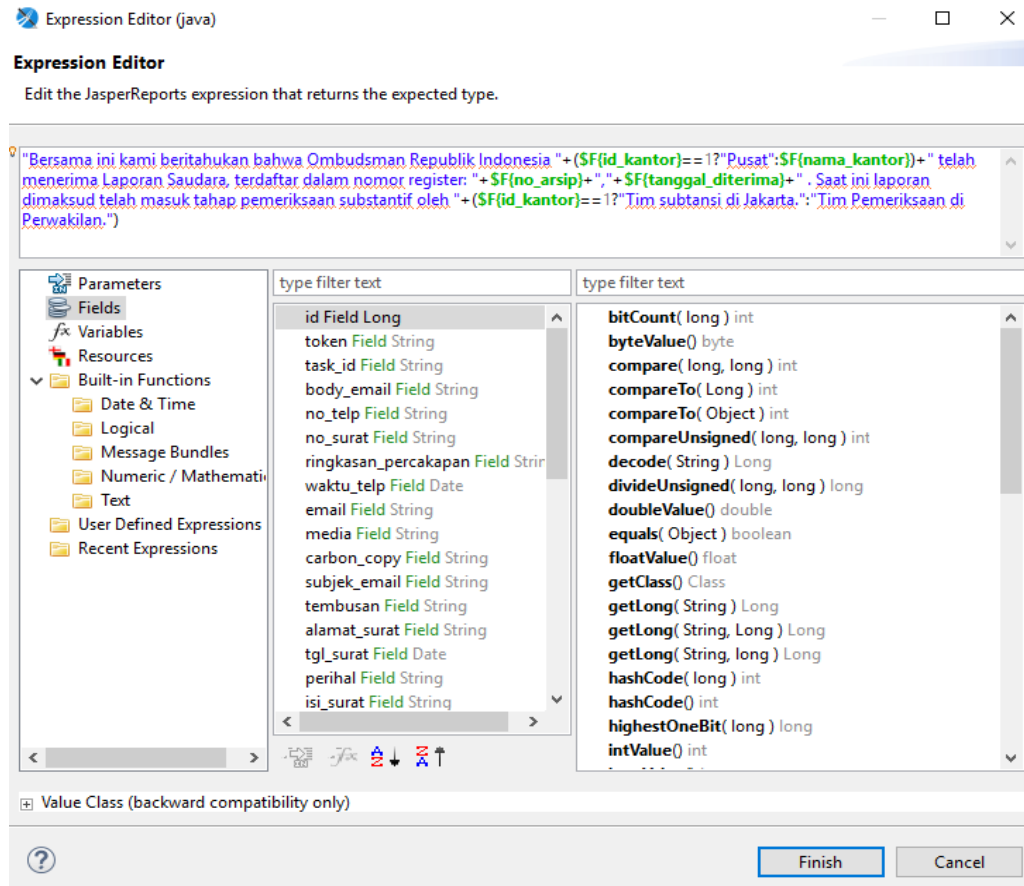
Query yang ditulis kemudian dibaca dengan memilih tombol “*Read Fields*” pada *Dataset and Query Dialog*. Tombol tersebut secara otomatis menjalankan *query* sehingga semua data yang telah di-*select* pada *query* akan ditampilkan ke dalam daftar *field* seperti yang ditunjukkan pada **Gambar 4.15**. Query yang ditulis untuk setiap surat adalah berbeda sesuai kebutuhan yang diperlukan untuk masing-masing surat.

Field Name	Class Type	Description	Column name	Column label	Column index	Time zone	Properties
id	java.lang.Long						2 Properties
token	java.lang.String						2 Properties
task_id	java.lang.String						2 Properties
body_email	java.lang.String						2 Properties
no_telp	java.lang.String						2 Properties
no_surat	java.lang.String						2 Properties
ringkasan_percakapan	java.lang.String						2 Properties
waktu_telp	java.sql.Date						2 Properties
email	java.lang.String						2 Properties
media	java.lang.String						2 Properties
carbon_copy	java.lang.String						2 Properties
subjek_email	java.lang.String						2 Properties
tembusan	java.lang.String						2 Properties
alamat_surat	java.lang.String						2 Properties
tgl_surat	java.sql.Date						2 Properties
perihal	java.lang.String						2 Properties
isi_surat	java.lang.String						2 Properties
signature_email	java.lang.String						2 Properties null
created_by	java.lang.Long						2 Properties
updated_by	java.lang.Long						2 Properties
created_at	java.sql.Timestamp						2 Properties
updated_at	java.sql.Timestamp						2 Properties
no_agenda	java.lang.String						2 Properties
lampiran	java.lang.String						2 Properties
nama_pelapor	java.lang.String						2 Properties
tgl_email	java.sql.Date						2 Properties

Gambar 4.15 Hasil pembacaan *field* dari *query*

Field hasil pengambilan data dengan *query* yang telah terbaca akan digunakan pada elemen *text field* yang telah di-*drag and drop* pada *layout* surat melalui *Expression Editor* dengan melakukan *double-click* pada *text field*. Ketika surat dicetak, elemen *text field* tersebut akan otomatis terisi dengan data bertipe *string* yang berasal dari *field* hasil *query* sesuai dengan token surat yang ingin dicetak. *Expression Editor* seperti yang ditunjukkan pada **Gambar 4.16**

menggunakan bahasa pemrograman Java. Terdapat beberapa kasus di mana surat pada aplikasi SIMPeL mengharuskan adanya logika percabangan, sebagai contoh adalah salah satu *text field* pada “Formulir Verifikasi Syarat Formil Laporan” seperti pada **Gambar 4.17**.



Gambar 4.16 *Expression Editor* pada salah satu elemen *text field* “Surat Pemberitahuan Dimulainya Pemeriksaan”

```
(SF{kategori_pelapor}.equals("kategori_pelapor_02")?
(SF{klasifikasi_pelapor}.equals("klasifikasi_pelapor_03")?
(SF{fc_identitas_pemberi_kuasa}==true?
"\u2714:")
):(SF{klasifikasi_pelapor}.equals("klasifikasi_pelapor_02")?
(SF{fc_identitas_pemberi_kuasa}==true?
"\u2714:")
:(SF{klasifikasi_pelapor}.equals("klasifikasi_pelapor_04")?
(SF{fc_identitas_pemberi_kuasa}==true?
"\u2714:")
:(SF{klasifikasi_pelapor}.equals("klasifikasi_pelapor_07")?
(SF{fc_identitas_pemberi_kuasa}==true?
"\u2714:")
)
)
)
):")
```

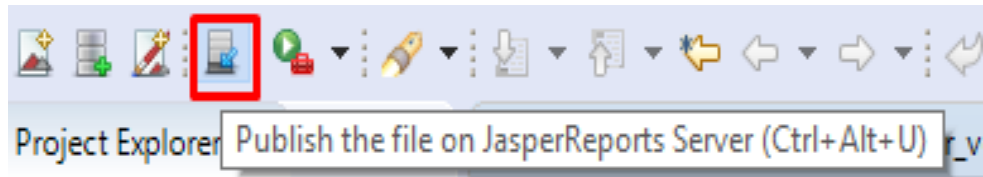
Gambar 4.17 Salah satu bentuk logika *if-else* pada surat SIMPeL

4.2.5 Proses Publish Surat ke JasperReports Server

Sebelum proses *publish*, kesesuaian surat yang telah didesain biasanya diperiksa terlebih dahulu dengan memilih menu “*Preview*”. Ketika melihat *preview* surat pada *Jaspersoft Studio*, diperlukan token untuk pengambilan data dari *database*. Contoh tampilan *preview* “Usulan Konsiliasi” dapat dilihat pada Gambar 4.18. Beberapa data pada gambar tersebut ditampilkan “null” karena token yang dimasukkan pada saat melakukan *preview* adalah “0”. Selanjutnya, apabila surat telah sesuai dengan harapan, dilakukanlah proses *publish* ke *JasperReports Server* dengan memilih menu seperti ditunjukkan pada Gambar 4.19. Pada saat proses *publishing*, server yang dipilih adalah *Jasperreports Server* yang telah terpasang di aplikasi SIMPeL, yang sebelumnya telah dihubungkan dengan *Jaspersoft Studio*.

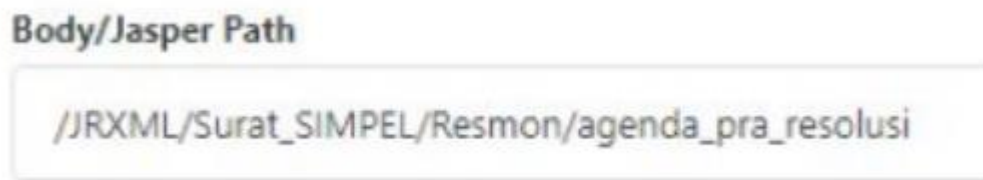


Gambar 4.18 Hasil *preview* surat “Usulan Konsiliasi” yang telah didesain



Gambar 4.19 Menu *publish* ke *JasperReports Server*

Berhasil mem-*publish* surat ke *Jasperreports Server* bukan berarti surat tersebut telah masuk dan siap digunakan di aplikasi SIMPeL. Surat tersebut harus didaftarkan pada aplikasi SIMPeL terlebih dahulu. Pendaftaran surat dilakukan oleh pengguna yang memiliki *role* sebagai *administrator* dengan memilih menu “Template Surat” pada aplikasi SIMPeL. Pada menu tersebut, terdapat fitur tambah surat dan di fitur tersebutlah dimasukkan informasi mengenai nama surat yang baru, pemilihan proses bisnis tempat surat dapat dicetak, juga mendefinisikan lokasi *path* surat tersebut di *JasperReports Server* seperti yang ditunjukkan pada **Gambar 4.20**.



Gambar 4.20 Pendefinisian lokasi *path* surat dari *JasperReports Server*

4.3 *Testing*

Aktivitas pengujian terhadap surat SIMPeL dilakukan oleh *tester* dan analisis secara manual melalui *development server* sebelum di-*deploy*. Pengujian dilakukan oleh 3 orang dengan memeriksa apakah *layout* surat yang telah dibuat menggunakan *JasperReports* telah sesuai dengan template yang ditentukan oleh klien. Surat diperiksa satu per satu melalui *development server* SIMPeL sesuai dengan daftar surat yang ada. Pengujian ini berlangsung selama kurang lebih 7 hari. Ketika pengujian, terdapat beberapa kasus terjadinya *overlap* karena penggunaan elemen yang belum diatur menjadi *float*. Namun, kesalahan-kesalahan tersebut segera diatasi dengan mengatur elemen menjadi *float* dengan *Jaspersoft Studio*. Hasilnya, hampir semua surat pernah mengalami perbaikan walaupun hanya memerlukan perbaikan skala kecil.

4.4 *Deployment*

Pendaftaran surat pada menu “Template Surat” merupakan tahapan terakhir dari proses pembuatan surat SIMPeL. Ketika aplikasi SIMPeL di-*deploy* pada *production server* milik

Ombudsman Republik Indonesia, *JasperReports Server* yang menyimpan surat juga di-*install* pada *production server* aplikasi SIMPeL. Kegiatan *deployment* dilaksanakan oleh *programmer* senior di PT Javan Cipta Solusi. Setelah aplikasi di-*deploy* dan mulai *live*, pengembang menyediakan *spreadsheet* untuk dapat digunakan oleh pengguna dalam menyampaikan masalah-masalah yang dialami ketika menggunakan aplikasi. Terdapat beberapa umpan balik yang menyatakan bahwa ada beberapa hal dari surat yang perlu diperbarui. Jika dirangkum, hal tersebut terbagi menjadi 2, yaitu masalah *layout* yang kurang rapi dan beberapa permintaan *update* mengenai logika *if-else* yang terdapat di dalam *text field*. Namun, beberapa masalah yang timbul setelah aplikasi SIMPeL di-*deploy* segera diatasi dengan mengedit surat kembali berdasarkan *list* aduan pada *spreadsheet*.



BAB V

REFLEKSI PELAKSANAAN MAGANG

4.3 Teknis

Untuk dapat menggunakan *JasperReports* ketika proses pengembangan SIMPeL, tidak diperlukan waktu lama untuk mempelajari fitur-fitur yang sering digunakan untuk pengembangan SIMPeL. Salah satu manfaat yang diperoleh adalah kemudahan karena penggunaannya hanya perlu melakukan *drag and drop* elemen yang dibutuhkan tanpa perlu menulis kode. Penggunaan *JasperReports* dapat dipelajari melalui dokumentasi yang tersedia pada *website JasperReports*. Selain itu, para *programmer* senior di PT Javan Cipta Solusi juga membantu penulis untuk beradaptasi dengan *JasperReports*. Namun, kemampuan penggunaan *JasperReports* penulis pun hanya terbatas kepada fitur-fitur yang digunakan pada aplikasi SIMPeL saja karena waktu yang terbatas. Sebagai contoh adalah penggunaan elemen-elemen selain *text field*, *static text*, *image*, *frame*, *break*, *rectangle*, dan *line* yang tidak pernah digunakan untuk pengembangan aplikasi SIMPeL.

Kesulitan yang dialami penulis selama menggunakan *JasperReports* untuk membuat surat di aplikasi SIMPeL adalah kurangnya jam terbang sehingga beberapa kali mengalami kendala saat menentukan elemen untuk mendesain surat. Kemudian, pernah terjadi suatu kasus di mana semua surat yang telah jadi harus diubah parameternya. Parameter yang awalnya merupakan PID harus diubah menjadi token. Perubahan tersebut tidak hanya harus dilakukan pada nama parameter, namun juga *query* surat. Perubahan ini harus dilakukan secara manual satu per satu pada 150 surat, sehingga cukup menguras waktu dalam prosesnya. Selain itu, beberapa *query* surat yang dibuat cukup rumit dan tidak menggunakan *view* sehingga waktu yang diperlukan untuk proses pengambilan data tidak terlalu cepat.

Ketika aplikasi SIMPeL telah digunakan oleh pengguna dan memasuki masa garansi, beberapa umpan balik dari pengguna mengenai kekurangan pada surat juga telah diatasi diatasi dengan memperbarui surat menggunakan *JasperReports* kembali. Hingga akhirnya proses pembuatan surat dengan *JasperReports* dapat selesai dengan mudah dan cepat yaitu 6 bulan waktu proyek ditambah 4 bulan pertama masa *maintenance*.

Selain itu, terdapat proyek lain yang dikerjakan oleh penulis selama magang, salah satunya adalah Portal PPSDM. Portal PPSDM merupakan sebuah aplikasi milik Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah yang mengintegrasikan antara fungsi sistem

informasi sumber daya manusia (HRIS), Pelatihan dan sertifikasi profesi, serta layanan pengembangan sumber daya manusia yang berkaitan dengan pengadaan (Pejabat Fungsional PPBJ dan PPK). Dari pengalaman mengerjakan proyek tersebut, salah satu pengetahuan baru terkait PT Javan Cipta Solusi adalah penggunaan *package laravolt*. *Laravolt* merupakan sebuah *package* yang dapat digunakan pada *framework laravel*. Penggunaan *package* ini banyak diterapkan pada beberapa sistem yang dikembangkan oleh PT Javan Cipta Solusi untuk fitur otentifikasi, manajemen pengguna, CRUD *generator*, dan masih banyak lagi.

4.4 Non-Teknis

4.4.1 *Time Management*

Pada saat magang, penulis tidak hanya bergabung di dalam satu proyek, karena itu penulis harus berpindah-pindah dari satu proyek ke proyek lain dalam satu hari. Keadaan ini menuntut penulis untuk mampu mengatur waktu dengan baik. Sebagai contoh, pada hari Senin, mulai dari jam 8 sampai dengan jam 12 penulis akan mengalokasikan waktu untuk mengerjakan *task* pada proyek PPSDM, kemudian di jam lain waktu akan dialokasikan untuk mengerjakan *task* pada proyek SIMPEL.

4.4.2 **Komunikasi**

Bekerja di sebuah perusahaan adalah tentang bekerja di dalam tim. Setiap proyek pun memiliki tim kecil tersendiri. Ketika mengalami kendala atau mendapatkan *task* yang kurang jelas, komunikasi adalah hal yang sangat diperlukan. Keadaan tersebut menuntut penulis untuk berani bertanya baik secara tertulis maupun lisan. Untuk bentuk tertulis, penulis biasanya bertanya langsung di *chatting group* suatu proyek. Sementara itu, contoh dari komunikasi secara lisan terkait kendala adalah ketika melakukan *conference call*. Penulis menjadi terbiasa untuk berani menyampaikan kendala- kendala dengan *supervisor* atau analis yang memberi *task*, lalu meminta saran untuk menyelesaikan kendala tersebut.

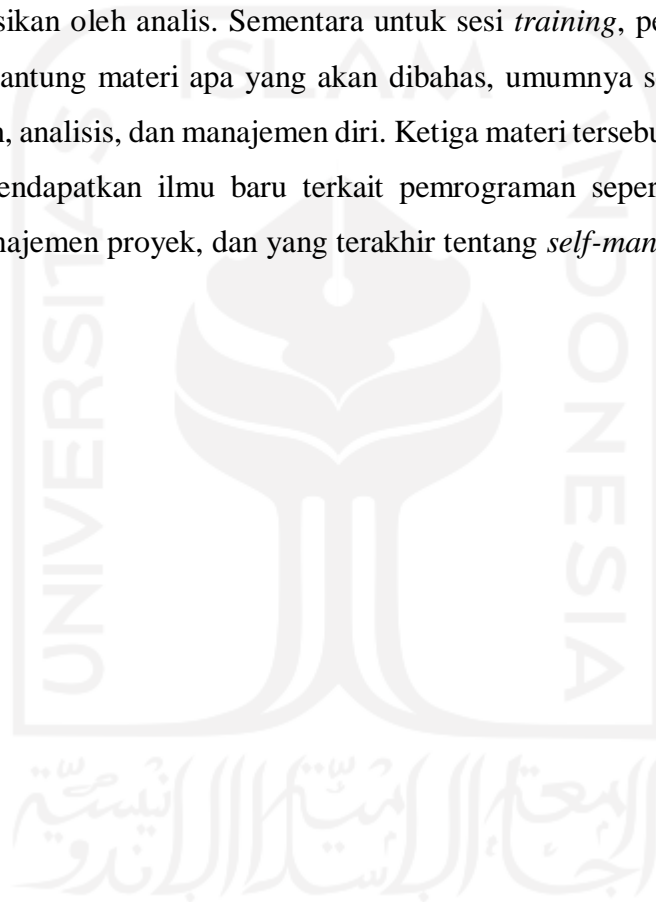
4.4.3 *Knowledge*

PT Javan Cipta Solusi mewajibkan setiap karyawannya, termasuk magang, untuk menulis satu artikel setiap bulan untuk nantinya artikel tersebut akan di-*publish* di blog perusahaan. Dengan adanya kewajiban ini, mau tidak mau penulis harus lebih banyak membaca artikel-artikel tentang teknologi agar mendapat inspirasi dan dapat menulis dengan baik.

Kemudian, setelah mengikuti program magang, penulis menyadari ada banyak sekali hal yang masih perlu dipelajari yang mana tidak didapatkan di bangku perkuliahan. Sebagai contoh

adalah penggunaan *git* untuk manajemen versi *source code*, juga beberapa *tools* lain seperti *workflow engine*, *reporting tools*, dan *software project management* seperti *ActiveCollab*.

Selain itu, PT Javan Cipta Solusi rutin mengadakan *review project* dan *training* setiap pekan, yaitu pada hari Jum'at. Kegiatan tersebut membuat penulis berkesempatan untuk mendapat ilmu lebih selain pengalaman bekerja. *Review project* dilakukan dalam lingkungan perusahaan dengan tujuan untuk menemukan *bug* sebelum aplikasi sampai ke pengguna. *Review project* umumnya dilakukan oleh analis dengan menjelaskan fitur baru apa yang telah dikerjakan di pekan tersebut, kemudian para penonton akan memberi kritik dan saran terhadap apa yang dipresentasikan oleh analis. Sementara untuk sesi *training*, pembicara setiap pekan adalah berbeda, tergantung materi apa yang akan dibahas, umumnya seputar masalah teknis seperti pemrograman, analisis, dan manajemen diri. Ketiga materi tersebut pernah penulis ikuti, sehingga penulis mendapatkan ilmu baru terkait pemrograman seperti *Automated Testing* dengan *Laravel*, manajemen proyek, dan yang terakhir tentang *self-management*.



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

JasperReports berhasil diimplementasikan dalam pengembangan fitur cetak surat pada aplikasi SIMPeL dengan melalui berbagai tahapan, mulai dari analisis dan identifikasi terhadap kebutuhan surat, pembuatan hingga publish surat dengan *Jaspersoft Studio*, pengujian sebelum dan setelah aplikasi di-*deploy*, juga proses *deployment* surat ke *server production* SIMPeL. Implementasi *JasperReports* dikerjakan oleh analis dan beberapa *programmer* sejak awal pengembangan hingga aplikasi SIMPeL *launching* dan memasuki masa garansi. Dengan implementasi *JasperReports*, pembuatan surat dengan jumlah 150 dapat terealisasi dalam waktu 10 bulan sejak proyek SIMPeL dimulai hingga memasuki masa garansi.

6.2 Saran

Selama proses pengembangan, disadari bahwa masih ada banyak fitur pada *JasperReports* yang belum terekplorasi. Fitur-fitur tersebut tidak digunakan karena fitur-fitur tersebut memang tidak diperlukan dalam proses pembuatan surat SIMPeL. Oleh karena itu, beberapa pekerjaan lanjutan yang diharapkan adalah adanya eksplorasi lebih jauh mengenai implementasi semua fitur yang ada pada *JasperReports* dan penelitian lebih lanjut yang membahas perbandingan antara *JasperReports* dengan *reporting tools* lain. Yang kedua adalah saran untuk menggunakan *view* pada *query* surat sehingga dapat mempercepat proses pengambilan data.

DAFTAR PUSTAKA

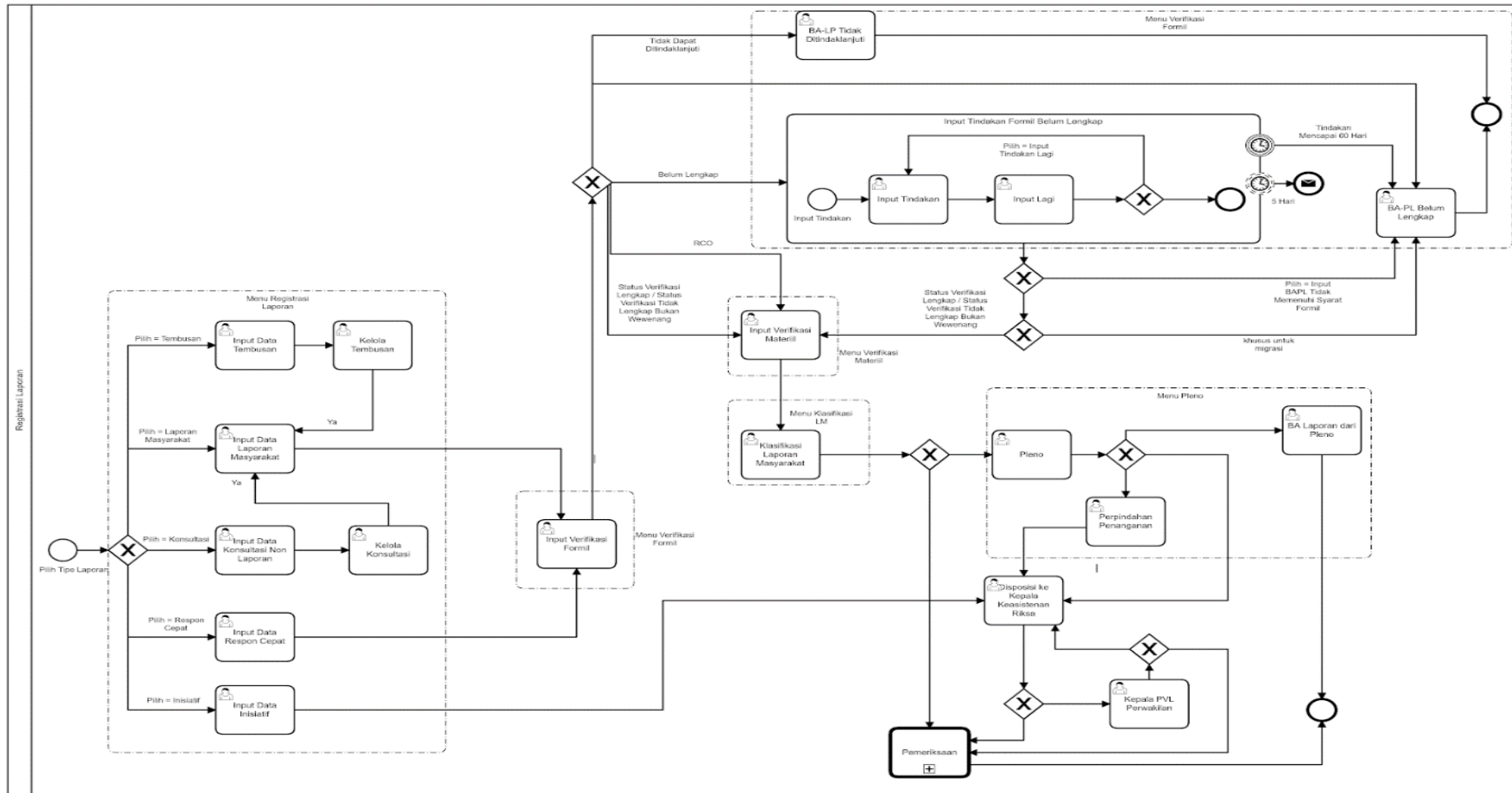
- Pasal 1 Undang-Undang Republik Indonesia Nomor 37 Tahun 2008 Tentang Ombudsman Republik Indonesia.
- Ombudsman Republik Indonesia. (2016). Hasil Studi Evaluasi Kinerja Ombudsman Republik Indonesia (ORI) Periode 2011-2016. Retrieved November 10, 2020, from <http://pattiro.org/wp-content/uploads/2016/03/Buku-Hasil-Studi-Evaluasi-Kinerja-ORI-2011-2016.pdf>
- Javan Cipta Solusi. (2019). Laporan Awal Pengembangan Aplikasi Sistem Informasi Manajemen Penyelesaian Laporan (SIMPEL V.3).
- Tindige, Nancy, Tampi, Johny Revo, & Londa, Very (2017). Kinerja Ombudsman RI Perwakilan Provinsi Sulawesi Utara pada Penyelesaian Laporan Masyarakat.
- Marpaung, Edison (2013). Sistem Penjadwalan Kuliah pada Program Studi Sistem Informasi UNIKOM.
- Beynon-Davies, P., Came, C., Mackay, H., & Tudhope, D. (1999). Rapid application development (Rad): An empirical review. *European Journal of Information Systems*, 8(3), 211–232. <https://doi.org/10.1057/palgrave.ejis.3000325>
- Biesemans, J., Horsten, W., & Deroose, T. (2010). PAF for Airborne Remote Sensing: Overview of Hardware , Software System and Operations. *System, September*.
- JasperReports® Server* | *Jaspersoft Community*. (n.d.). Retrieved December 9, 2020, from <https://community.jaspersoft.com/project/jasperreports-server>
- Jaspersoft® Studio* | *Jaspersoft Community*. (n.d.). Retrieved December 9, 2020, from <https://community.jaspersoft.com/project/jaspersoft-studio>
- Adhi, B. P., Prasetya, D. N., & Widodo, W. (2019). Performance comparison of reporting engine birt, jasper report, and crystal report on the process business intelligence. *IOP Conference Series: Materials Science and Engineering*, 508(1). <https://doi.org/10.1088/1757-899X/508/1/012129>
- JasperReports® Library* | *Jaspersoft Community*. (n.d.). Retrieved December 16, 2020, from <https://community.jaspersoft.com/project/jasperreports-library>
- Nugraha, Bobby Pratama Widya. (2019). Sistem dan Prosedur Pelayanan pada Kantor Ombudsman Republik Indonesia Perwakilan DIY. Retrieved December 9, 2020, from <https://edoc.uui.ac.id/bitstream/handle/123456789/18136/14211063.pdf?sequence=11&isAllowed=y>

- Riestiana, M., & Sukadi. (2014). Sistem Informasi Penggajian Karyawan Pada Commenditaire Vennontschap (CV) RGL Bordir Dan Konveksi Pacitan. *IJNS – Indonesian Journal on Networking and Security*, 9330(4), 1–7.
<http://ijns.org/journal/index.php/ijns/article/view/406>
- Sapre, B. S., Thakare, R. V., & Kakade, S. V. (2012). Design and Application of the Hibernate Persistence Layer Data Report System using JasperReports. *International Journal of Engineering and Innovative Technology (IJEIT)*, 1(5), 2277–3754.

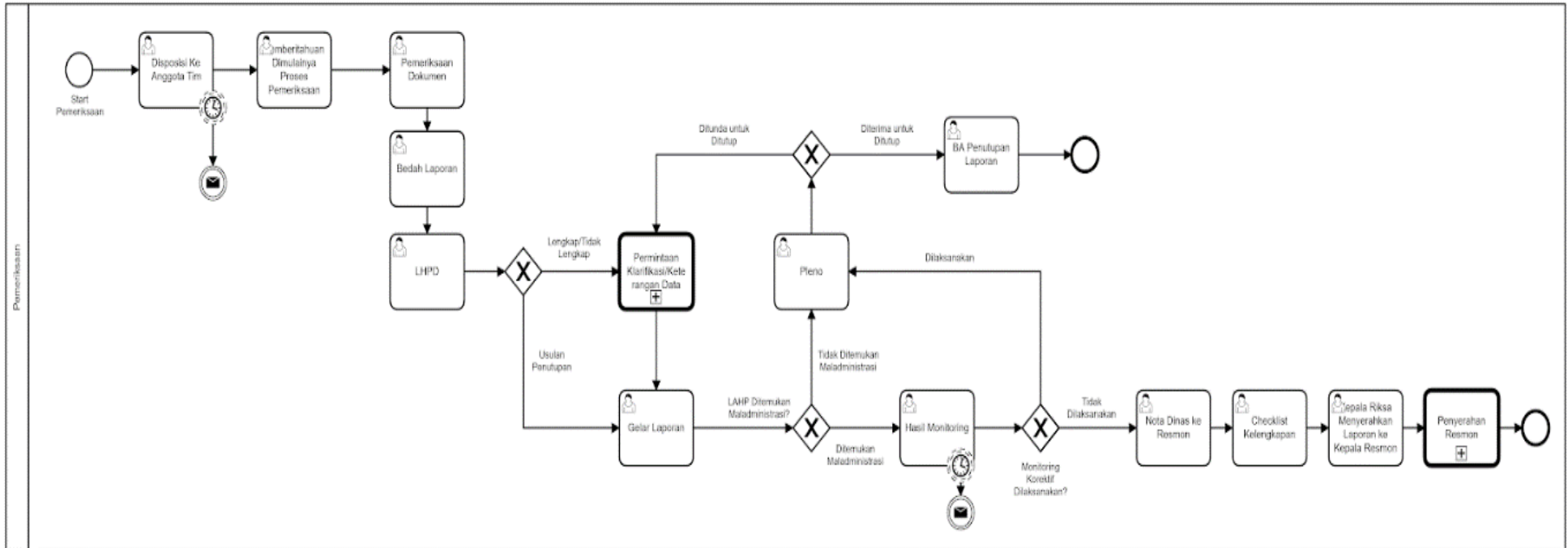


LAMPIRAN

a. Alur proses bisnis modul Penerimaan dan Verifikasi Laporan SIMPeL (



b. Alur proses bisnis modul Pemeriksaan SIMPeL



c. Alur proses bisnis modul Resolusi dan *Monitoring* SIMPeL

