

Implementasi JasperReports pada Sistem Informasi Manajemen

(Studi Kasus: Sistem Informasi Manajemen Penyelesaian Laporan Ombudsman)

Amalia Nur Hasanah¹
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta, Indonesia
17523212@students.uii.ac.id

Hanson Prihantoro Putro²
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta, Indonesia
hanson@uii.ac.id

Abstrak — 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*, sehingga 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 makalah 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*

I. PENDAHULUAN

Ombudsman merupakan sebuah lembaga pemerintahan yang memiliki kewenangan untuk mengawasi pelayanan publik baik yang diselenggarakan oleh penyelenggara negara dan pemerintahan yang diberi tugas menyelenggarakan pelayanan publik tertentu yang sebagian atau seluruh dananya bersumber dari anggaran pendapatan dan belanja negara dan/atau anggaran pendapatan dan belanja daerah [1]. Salah satu peran Ombudsman Republik Indonesia adalah bertindak sebagai perantara bagi masyarakat dan lembaga-lembaga penyelenggara negara. Ombudsman Republik Indonesia melakukan klarifikasi, *monitoring*, atau pemeriksaan terhadap

laporan masyarakat atas dugaan maladministrasi yang dilakukan oleh penyelenggara negara atau instansi lain atau perseorangan dalam pelayanan publik. Untuk mempermudah pengawasan tersebut, Ombudsman Republik

Indonesia didukung oleh sebuah aplikasi berbasis website yaitu Sistem Informasi Manajemen Penyelesaian Laporan (SIMPeL) [2]. SIMPeL digunakan untuk memberikan efektifitas dan efisiensi dalam hal penanganan laporan masyarakat yang masuk ke Ombudsman, baik di tingkat pusat maupun perwakilan.

Dalam proses penanganan laporan masyarakat, terdapat tahapan-tahapan yang mengharuskan dilakukannya pencetakan surat atau dokumen sesuai dengan alur penyelesaian laporan. Oleh karena itu, fitur cetak surat menjadi salah satu fitur penting yang selalu ada di setiap aktivitas dalam alur proses bisnis sistem. Dengan jenis surat yang beragam, serta data tentang laporan masyarakat yang sangat banyak, menghasilkan sebuah keluaran dokumen menjadi tidak mungkin apabila proses pendokumentasian data laporan masyarakat ke dalam bentuk surat dilakukan secara manual. Sehingga digunakanlah *reporting tools* untuk membantu menyelesaikan permasalahan tersebut.

Ada berbagai *reporting tools* yang dikenal hingga saat ini, seperti **Pentaho** [3], **Valentina Reports** [4], **JasperReports** [5], **Telerik Reporting** [6], dan lain sebagainya. Namun, dalam pengembangan aplikasi SIMPeL, pengembang menggunakan **JasperReports** karena lingkungan perusahaan yang sudah familiar dengan *tools* tersebut. Dengan *tool* yang familiar, tentu proses pengembangan pun dapat dilakukan secara lebih efisien dan efektif.

Makalah 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 makalah ini dapat menjadi pembelajaran dan referensi sehingga proses pembuatan *report* pada sebuah sistem informasi menjadi lebih efisien.

II. KAJIAN PUSTAKA

A. SIMPeL (Sistem Informasi Manajemen Penyelesaian Laporan)

SIMPeL adalah aplikasi milik lembaga Ombudsman Republik Indonesia yang dijalankan sejak tahun 2014 guna membantu proses penyelesaian laporan masyarakat [7]. Aplikasi **SIMPeL** merupakan sebuah aplikasi berbasis web yang dapat diakses hanya oleh pegawai Ombudsman Republik Indonesia. Seiring berjalannya waktu, prosedur penyelesaian laporan oleh Ombudsman Republik Indonesia telah mengalami beberapa perubahan. Hal tersebut menuntut **SIMPeL** sebagai aplikasi yang menunjang prosedur tersebut mau tidak mau harus mengalami perubahan. Hingga saat ini, aplikasi **SIMPeL** telah melewati tahap pengembangan versi 3.0.

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 [8].

B. JasperReports

JasperReports merupakan sebuah *library* pembuatan laporan *open-source* milik TIBCO Software Inc. *Tool* tersebut biasanya digunakan pada perangkat lunak untuk menghasilkan suatu keluaran berupa laporan atau surat atau dokumen lain yang memerlukan pengambilan data dari basis data. Pembuatan laporan dengan **JasperReports** menggunakan bahasa pemrograman **Java** dan *tool* tersebut memungkinkan pengambilan data dari berbagai jenis sumber data [5]. **JasperReports** dapat menghasilkan keluaran berupa dokumen atau laporan atau surat dengan format PDF, HTML, XLS, XLSX, RTF, CSV, XML, DOCX, ODT, ODS, dan JPRINT.

Report dibuat dengan menggunakan software **Jaspersoft Studio** yang dapat di-*install* di berbagai macam sistem operasi komputer. **Jaspersoft Studio** digunakan untuk mendesain *report*, menghubungkan *report* dengan sumber data, dan mem-*publish report* yang telah dibuat ke **JasperReports Server**. Sementara itu, **JasperReports Server** merupakan sebuah aplikasi pada *server* yang memungkinkan pengiriman data, dengan men-*deploy report* ke **JasperReports Server**, *report* tersebut akan dapat diakses sesuai dengan keluaran yang diharapkan [9].

C. Implementasi JasperReports pada Pengembangan Sistem Informasi Lain

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

JasperReports juga digunakan untuk mengkompilasi dan menghasilkan *report* berupa file **jasper* atau **jrxml* [10]. Yang kedua adalah pengembangan sistem penjadwalan kuliah pada Program Studi Sistem Informasi UNIKOM. Dalam pengembangan sistem tersebut, **JasperReports** digunakan untuk mencetak hasil *reporting* dalam bentuk PDF [11]. Dari beberapa pengembangan sistem informasi, dapat disimpulkan bahwa **JasperReports** merupakan sebuah *library* yang cukup umum digunakan untuk keperluan *reporting*.

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.

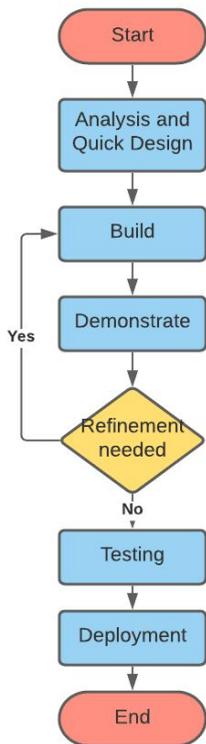
Salah satu penelitian luar membahas tentang implementasi **JasperReports** dalam pengembangan sebuah sistem informasi untuk menghasilkan sebuah *report* [14]. Pada penelitian tersebut, disimpulkan bahwa penggunaan **JasperReports** membantu dalam hal fleksibilitas dalam pembuatan *report*. **JasperReports** 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.

III. METODOLOGI

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 [12]. Hal ini bersesuaian dengan harapan pengguna yang menginginkan proyek dapat selesai dengan cepat. 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 1**.

A. 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).



Gambar 1. Diagram alir metode pengembangan RAD

B. Prototype Cycles

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

1. *Build*: Proses ini mengimplementasikan hasil analisis ke dalam *code* yang mana tahap ini dilakukan oleh *programmer*.
2. *Demonstrate*: Dalam tahap ini, 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*.
3. *Refine*: Proses ini memperbaiki hasil dari *build* apabila terdapat hal-hal yang tidak sesuai dengan keinginan calon pengguna.

C. 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*. Aplikasi yang telah melewati tahap *testing* dan bebas dari *bug* berarti aplikasi tersebut siap untuk di-*deploy*.

D. Deployment

Deployment dilakukan apabila aplikasi yang dikembangkan telah sesuai dengan rencana kebutuhan pengguna dan bebas dari *bug*. Pada tahap ini dilakukan instalasi aplikasi di *Server Production* dan jaringan infrastruktur yang ada [13].

IV. HASIL DAN PEMBAHASAN

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, HTML, XLS, XLSX, RTF, CSV, XML, DOCX, ODT, ODS, dan JPRINT. Namun, khusus untuk surat pada aplikasi **SIMPeL**, keluaran yang dihasilkan adalah dokumen dengan ekstensi .PDF dan .DOCX.

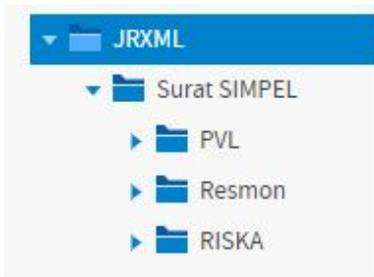
A. Analysis and quick design

Aplikasi **SIMPeL** mulai dikerjakan sejak bulan Oktober 2019 dengan lama proyek 6 bulan ditambah masa *maintenance* hingga bulan Desember 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 *client* dengan total 57 surat untuk modul Penerimaan dan Verifikasi Laporan (PVL), 37 surat untuk modul Pemeriksaan (Riksa), dan 56 surat untuk modul Resolusi dan *Monitoring* (Resmon).

B. Prototype Cycles

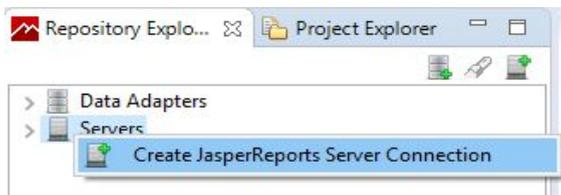
1. Konfigurasi Jasperreports Server dan Basis Data pada Jaspersoft Studio

Proses pengembangan fitur cetak surat yang mengimplementasikan **JasperReports** pada aplikasi **SIMPeL** diawali dengan instalasi **JasperReports Server** di *server development* aplikasi **SIMPeL**. **JasperReports Server** dapat diakses setelah komputer terhubung dengan *port SIMPeL* dan melakukan *log-in* dengan mengakses `http://<host>:<port>/jasperserver[-pro]/`. Semua surat untuk aplikasi **SIMPeL** harus tersimpan di dalam sebuah folder pada **JasperReports Server** seperti pada **Gambar 2**. 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.

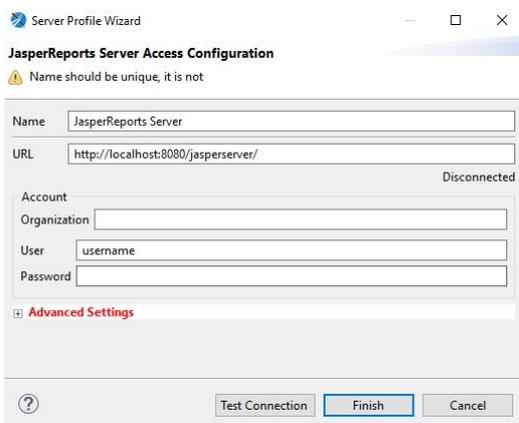


Gambar 2. Folder yang menyimpan surat SIMPEL di JasperReports Server

Setelah JasperReports Server terpasang di aplikasi SIMPEL, aktivitas selanjutnya adalah membuat surat dengan software Jaspersoft Studio, yang mana pengembang menggunakan versi 6.10.0. Sebelum memasuki proses mendesain, hal pertama yang harus dilakukan saat membuka software 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 3 dan melakukan konfigurasi akses dengan JasperReports Server seperti pada Gambar 4. Proses menghubungkan software dengan JasperReports Server bertujuan untuk mengambil repository dari JasperReports Server sehingga Jaspersoft Studio dan JasperReports Server langsung terintegrasi.

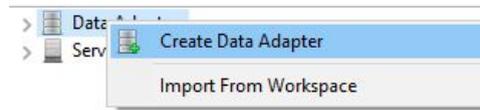


Gambar 3. Menu pembuatan koneksi JasperReports Server

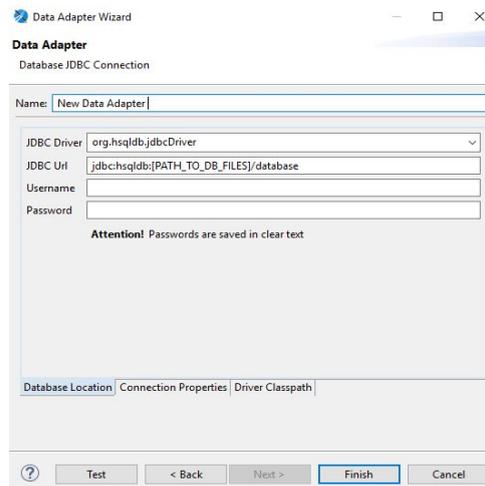


Gambar 4. Konfigurasi akses terhadap JasperReports Server

Setiap surat pada aplikasi SIMPEL berisi data tentang laporan masyarakat. Data tersebut tersimpan di dalam basis data SIMPEL milik Ombudsman Republik Indonesia, sehingga untuk menampilkan surat dengan data yang sesuai, juga diperlukan koneksi antara Jaspersoft Studio dengan basis data SIMPEL. Aktivitas penghubungan antara basis data dan Jaspersoft Studio dilakukan dengan menambah JDBC Data Adapter seperti yang ditunjukkan pada Gambar 5, kemudian memasukkan informasi mengenai basis data SIMPEL seperti pada Gambar 6. Data adapter digunakan untuk mendukung pertukaran data antara data source dan dataset. Sementara JDBC dipilih karena jenis adapter tersebut memungkinkan penggunaan basis data yang diakses melalui JDBC driver, dalam hal ini merupakan PostgreSQL.



Gambar 5. Penambahan data adapter



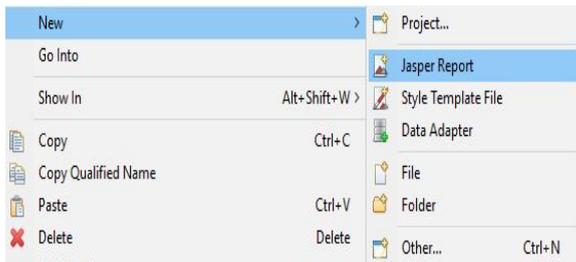
Gambar 6. Konfigurasi database SIMPEL pada JDBC data adapter

2. Proses Desain Surat dengan Jaspersoft Studio

Setelah terhubung dengan JasperReports Server dan basis data SIMPEL, tahapan pembuatan surat SIMPEL selanjutnya adalah membuat template atau desain surat sesuai template yang telah ditentukan oleh Ombudsman Republik Indonesia. Template surat dibuat dengan menambahkan file report baru di project explorer Jaspersoft Studio seperti ditunjukkan pada Gambar 7, termasuk memilih 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*.



Gambar 7. 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*.

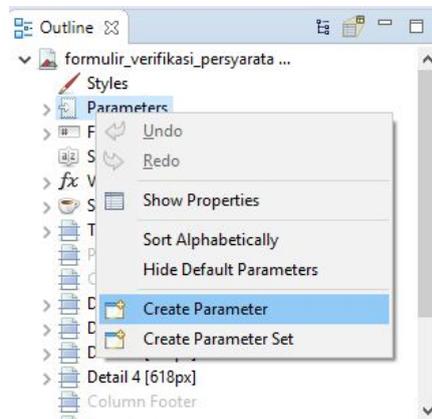
Salah satu surat yang dibuat pada aplikasi **SIMPeL** adalah surat “Usulan Konsiliasi” seperti yang ditunjukkan pada **Gambar 8**. Pada gambar tersebut surat terbagi menjadi dua bagian, yaitu *title* yang berisi kop surat dan *detail* yang memuat isi surat. Pada kop surat terdapat logo yang menggunakan elemen *image* dan diambil dari *repository JasperReports Server*.



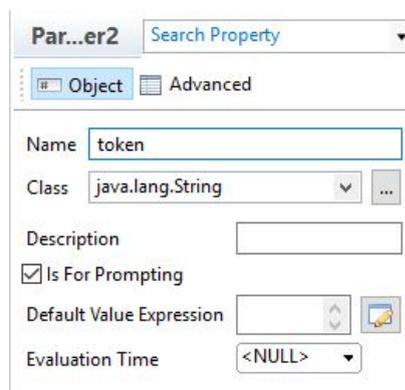
Gambar 8. Hasil desain layout surat “Usulan Konsiliasi”

3. Pengambilan Data dari Basis Data SIMPeL

Data laporan masyarakat di dalam basis data **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 basis data 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 memberi nama parameter seperti yang ditunjukkan pada **Gambar 9** dan **Gambar 10**.



Gambar 9. Pembuatan parameter baru untuk token surat



Gambar 10. Konfigurasi parameter dengan nama parameter “token”

Setelah parameter yang berfungsi untuk menyimpan token dibuat, tahap selanjutnya adalah penulisan *query* pada *Dataset and Query Dialog*. *Query* surat pada 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 terdapat pada **Gambar 11**. Pada *Dataset and Query Dialog*, terdapat pilihan *data adapter* yang dapat digunakan. Karena itu, dipilihlah *data adapter* yang sebelumnya telah dibuat dan terhubung dengan basis data **SIMPeL**.



Gambar 15. 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 16**.

Body/Jasper Path

```
/JRXML/Surat_SIMPEL/Resmon/agenda_pra_resolusi
```

Gambar 16. Pendefinisian lokasi *path* surat dari **JasperReports Server**

C. Testing

Aktivitas pengujian terhadap surat **SIMPeL** dilakukan oleh *tester* secara manual melalui *development server* sebelum di-*deploy*. Pengujian dilakukan dengan memeriksa apakah *layout* surat yang telah dibuat menggunakan **JasperReports** telah sesuai dengan *template* yang ditentukan oleh *client*. 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**.

D. Deployment

Pendaftaran surat pada menu “Template Surat” merupakan tahapan terakhir dari proses pembuatan surat **SIMPeL**. Ketika aplikasi **SIMPeL** di-*deploy* pada *production server*, **JasperReports Server** yang menyimpan surat juga di-*install* pada *production server* aplikasi **SIMPeL**. Kemudian setelah aplikasi **SIMPeL** *live* dan memasuki awal masa *maintenance*, 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, selama masa *maintenance*, semua masalah pada umpan balik dapat diatasi dengan memperbarui surat menggunakan **JasperReports** kembali hingga akhirnya proses pembuatan surat dengan **JasperReports** dapat selesai dalam 6 bulan waktu proyek ditambah 4 bulan pertama masa *maintenance*.

A. 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 tahap *maintenance*. Dengan implementasi **JasperReports**, pembuatan surat dengan jumlah 150 dapat terealisasi dalam waktu 10 bulan sejak proyek **SIMPeL** dimulai hingga memasuki masa *maintenance*.

B. 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:

1. Explorasi lebih jauh mengenai implementasi semua fitur yang ada pada **JasperReports**.
2. Penelitian lebih lanjut yang membahas perbandingan antara **JasperReports** dengan *reporting tools* lain.

VI. REFERENSI

- [1] Pasal 1 Undang-Undang Republik Indonesia Nomor 37 Tahun 2008 Tentang Ombudsman Republik Indonesia.
- [2] “Welcome Home | Simpel.” <https://simpler.ombudsman.go.id/dashboard> (accessed Nov. 24, 2020).
- [3] “Pentaho.” <https://community.hitachivantara.com/s/topic/0TO1J0000017kUqWAI/pentaho> (accessed Nov. 24, 2020).
- [4] “Overview.” <https://www.valentina-db.com/en/valentina-reports-adk-overview> (accessed Nov. 24, 2020).
- [5] “JasperReports® Library | Jaspersoft Community.” <https://community.jaspersoft.com/project/jasperreports-library> (accessed Nov. 19, 2020).
- [6]. “NET Reporting - Core, Blazor, WPF, Angular | Telerik Reporting.” <https://www.telerik.com/products/reporting.aspx> (accessed Nov. 24, 2020).
- [7] “Hasil Studi Evaluasi Kinerja Ombudsman Republik Indonesia (ORI) Periode 2011-2016” <http://pattiro.org/wp-content/uploads/2016/03/Buku-Hasil-Studi-Evaluasi-Kinerja-ORI-2011-2016.pdf> (accessed Nov. 10, 2020)
- [8] Nancy Tindige, Johny Revo Tampi, and Very Londa,

- “Kinerja Ombudsman RI Perwakilan Provinsi Sulawesi Utara pada Penyelesaian Laporan Masyarakat”, 2017.
- [9] “What is JasperReports Server? | Jaspersoft Community.”
<https://community.jaspersoft.com/wiki/what-jasperreports-server> (accessed Nov. 19, 2020).
- [10] M. Riestiana and Sukadi, “Sistem Informasi Penggajian Karyawan Pada Commenditaire Vennotschap (CV) RGL Bordir Dan Konveksi Pacitan,” *IJNS – Indones. J. Netw. Secur.*, vol. 9330, no. 4, pp. 1–7, 2014, [Online]. Available: <http://ijns.org/journal/index.php/ijns/article/view/406>.
- [11] Edison Marpaung, “Sistem Penjadwalan Kuliah pada Program Studi Sistem Informasi UNIKOM”, 2013.
- [12] P. Beynon-Davies, C. Came, H. Mackay, and D. Tudhope, “Rapid application development (RAD): An empirical review,” *Eur. J. Inf. Syst.*, vol. 8, no. 3, pp. 211–232, 1999, doi: 10.1057/palgrave.ejis.3000325.
- [13] J. Biesemans, W. Horsten, and T. Deroose, “PAF for Airborne Remote Sensing : Overview of Hardware , Software System and Operations,” *System*, no. September, 2010.
- [14] 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.