

Implementasi JasperReports pada Sistem Informasi Manajemen

by John Doe

Submission date: 24-Nov-2020 07:31PM (UTC+0700)

Submission ID: 1456015028

File name: lementasi_JasperReports_pada_Sistem_Informasi_Manajemen.docx.pdf (587.34K)

Word count: 3254

Character count: 21503

Implementasi JasperReports pada Sistem Informasi Manajemen

(Studi Kasus: Sistem Informasi Manajemen Penyelesaian Laporan Ombudsman)

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 pengembanganpun terbilang lama mengingat banyaknya pekerjaan lain yang dimiliki oleh *programmer*. Terlebih, jumlah surat yang harus dibuat lebih banyak daripada versi sebelumnya. Perlu digunakan *reporting tools* demi memudahkan proses pengembangan fitur cetak surat. *Reporting tools* yang digunakan adalah JasperReports. JasperReports merupakan sebuah *tool* pembuatan laporan *open-source* yang menggunakan bahasa pemrograman Java. Hasil dari penggunaan teknologi ini adalah lebih cepatnya pengembangan fitur cetak surat. 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*

5

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] sehingga tercipta 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 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 *commendaire 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*.

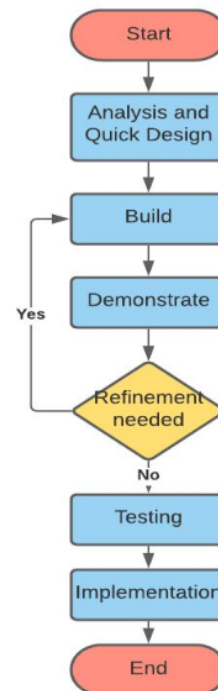
III. METODOLOGI

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**, yaitu:

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).



Gambar 1. Diagram alir metode pengembangan RAD

2. Prototype Cycles

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

- **Build:** Proses ini mengimplementasikan hasil analisis ke dalam *code* yang mana tahap ini dilakukan oleh *programmer*.
- **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*.

- *Refine*: Proses ini memperbaiki setelah mendapat umpan balik apabila terdapat hal-hal yang tidak sesuai dengan keinginan calon pengguna.

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

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 *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 Server** menyediakan **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.

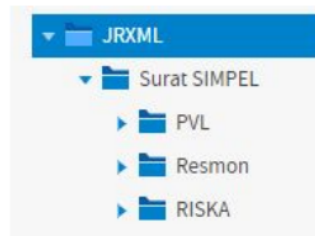
A. *Analysis and quick design*

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*

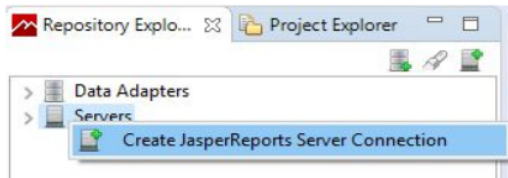
Proses pengembangan fitur cetak surat yang mengimplementasikan **JasperReports** pada aplikasi **SIMPeL** diawali dengan instalasi **JasperReports Server** di *server development* aplikasi **SIMPeL**. **JasperReports Server** dapat diakses setelah komputer terhubung dengan *port SIMPeL* dan melakukan *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.



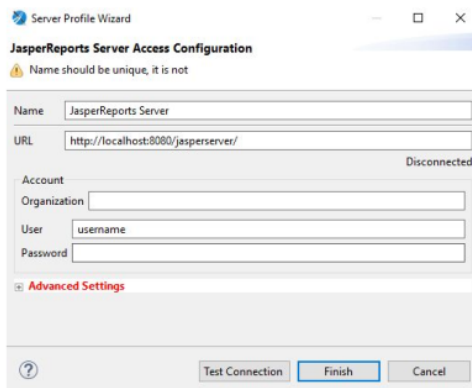
Gambar 2. Folder yang menyimpan surat **SIMPeL** di **JasperReports Server**

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**. 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. 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.

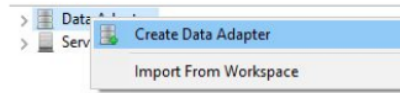


Gambar 3. Menu pembuatan koneksi **JasperReports Server**

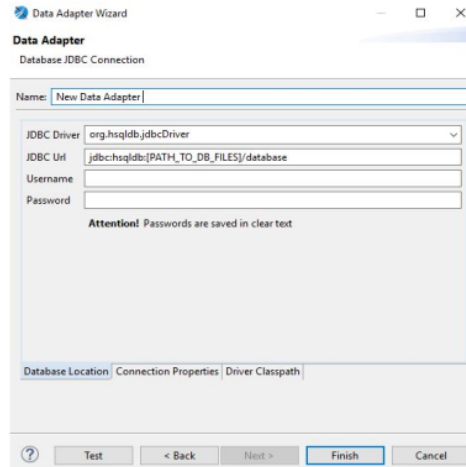


Gambar 4. Konfigurasi akses terhadap **JasperReports Server**

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**. Menghubungkan basis data 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

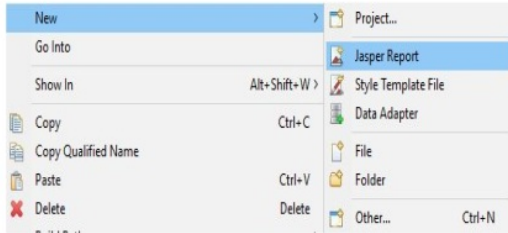


Gambar 6. Konfigurasi database **SIMPeL** pada **JDBC Data Adapter**

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 meng-*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*.

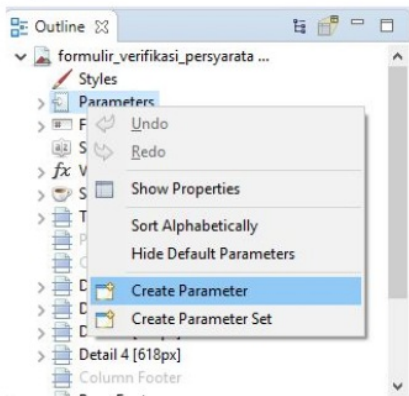


Gambar 7. Penambahan layout surat baru

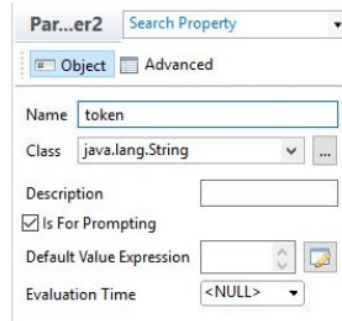
Selanjutnya, proses mendesain surat SIMPeL dilakukan dengan melakukan *drag and drop* secara langsung elemen yang sudah disediakan oleh **Jaspersoft Studio**. Selain melakukan *drag and drop*, tersedia juga opsi lain untuk mengedit dengan *source JRXML*. Beberapa elemen yang paling sering digunakan dalam 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*.

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 8 dan Gambar 9.

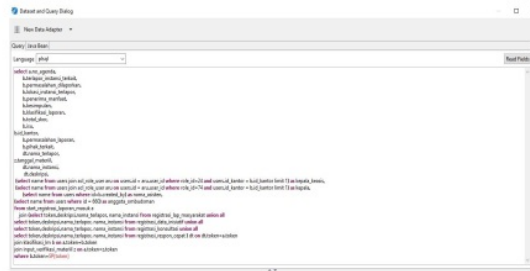


Gambar 8. Pembuatan parameter baru untuk token surat



Gambar 9. 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 10. 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 10. Query untuk salah satu surat SIMPeL

Query yang ditulis kemudian dibaca dengan memilih tombol "Read Fields". 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 11. *Query* yang ditulis untuk setiap surat adalah berbeda sesuai kebutuhan yang diperlukan untuk masing-masing surat. Sehingga pada aplikasi SIMPeL diperlukan kurang lebih 150 *query*.

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, hal tersebut terjadi karena fitur-fitur tersebut tidak digunakan untuk pembuatan surat **SIMPeL**. Oleh karena itu, beberapa pekerjaan lanjutan yang diharapkan:

- Explorasi lebih jauh mengenai implementasi semua fitur yang ada pada **JasperReports**
- 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] مطالعه قراملکی and بطر ش. یامچلو، خ. ایلی شناخت وضعیت موجود مسئولیت اجتماعی دانشگاه مینتی بر الگوی والیس بر اساس دیدگاه اعضای هیئت علمی (مورد: دانشگاه تهران). 1396.
- [8] A. Pérez *et al.*, "No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title," *BMC Public Health*, vol. 5, no. 1, pp. 1–8, 2017, [Online]. Available: <https://ejournal.poltektegal.ac.id/index.php/siklus/article/view/298%0Ahttp://repositorio.unan.edu.ni/2986/1/5624.pdf%0Ahttp://dx.doi.org/10.1016/j.jana.2015.10.005%0Ahttp://www.biomedcentral.com/1471-2458/12>

- [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 Vennontschap (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.

Implementasi JasperReports pada Sistem Informasi Manajemen

ORIGINALITY REPORT

8%

SIMILARITY INDEX

7%

INTERNET SOURCES

4%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1	pewarta-indonesia.com Internet Source	1%
2	Samaneh Sanaeipoor, Khashayar Hojjati Emami. "Smart [AR] Mini-Application: Engaging Citizens in Digital Placemaking Approach", 2020 4th International Conference on Smart City, Internet of Things and Applications (SCIOT), 2020 Publication	1%
3	docplayer.net Internet Source	1%
4	repository.radenintan.ac.id Internet Source	1%
5	hardyee.blogspot.com Internet Source	1%
6	id.scribd.com Internet Source	<1%
7	upcommons.upc.edu Internet Source	<1%

8	jurnal.batan.go.id Internet Source	<1%
9	www.guru99.com Internet Source	<1%
10	Nur Ariesanto Ramdhan, Dimas Wahyudi. "Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Berbasis WEB Di SMP Negeri 1 Wanasari Brebes", Jurnal Ilmiah Intech : Information Technology Journal of UMUS, 2019 Publication	<1%
11	id.123dok.com Internet Source	<1%
12	Michihito Ando, Masato Furuichi. "The impact of COVID-19 employment shocks on suicide and poverty alleviation programs: An early-stage investigation", Cold Spring Harbor Laboratory, 2020 Publication	<1%
13	repository.javeriana.edu.co Internet Source	<1%
14	ejournal.upi.edu Internet Source	<1%
15	mahasiswa.dinus.ac.id Internet Source	<1%
16	jtiulm.ti.ft.ulm.ac.id Internet Source	<1%

<1%

17 es.scribd.com
Internet Source

<1%

18 www.slideshare.net
Internet Source

<1%

19 mutiarahombing.blogspot.com
Internet Source

<1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On