

Metode Penyimpanan Data pada Joget Workflow (Studi Kasus: Website Electronic Human Resource)

Arya Wira Pahlawan
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
16523142@students.uii.ac.id

Hanson Prihantoro Putro
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
hanson@uui.ac.id

Abstrak— Electronic Human Resource atau biasa disingkat dengan EHR merupakan sebuah *website* yang digunakan untuk membantu pengolahan data sumber daya manusia (SDM) terutama data pegawai dalam suatu perusahaan. EHR akan mengelola berbagai data pegawai mulai dari data pribadi sampai data pekerjaannya. Seiring berjalannya waktu, data pegawai tersebut akan mengalami perubahan. Setiap proses perubahan data akan melalui proses bisnis yang berlaku dalam perusahaan tersebut. Sistem akan mengelola banyak data dengan berbagai macam proses perubahan data sehingga akan membuat proses penyimpanan data sulit untuk ditangani. Maka dari itu, pengembangan proyek ini memerlukan mekanisme penyimpanan data yang efektif, solusinya adalah penyimpanan data pada *platform* bernama Joget Workflow. Proses penyimpanan data yang banyak sesuai proses bisnis dapat ditangani oleh *platform* ini. Joget Workflow mempunyai fitur pengembangan sistem berbasis proses bisnis dengan berbagai macam metode penyimpanan data. Setiap metode tersebut dapat digunakan sesuai dengan skema yang ada sehingga akan mempermudah proses penyimpanan data. Setelah dilakukan pengembangan dapat disimpulkan bahwa penerapan metode penyimpanan data pada Joget Workflow cocok untuk digunakan dalam pengembangan sistem dengan skema penyimpanan data yang bervariasi.

Keywords— *Electronic Human Resource; penyimpanan data; Joget Workflow.*

I. PENDAHULUAN

Sebuah organisasi terdiri dari berbagai sumber daya yang salah satunya adalah sumber daya manusia (SDM). Sumber daya manusia (SDM) sangat penting dalam sebuah organisasi karena pegawai merupakan orang yang menggunakan dan mengelola sumber daya lainnya untuk menjalankan perusahaan [1]. Kumpulan pegawai tersebut akan membentuk sebuah struktur organisasi yang terdiri dari berbagai macam unit kerja. Unit kerja pegawai dibagi berdasarkan fungsi dan pembagian kerja. Banyaknya jumlah informasi yang harus dikelola berhubungan dengan SDM sehingga memunculkan permasalahan dengan tingkat kompleksitas tinggi [2]. Seiring berjalannya waktu, informasi data pegawai tentu akan mengalami perubahan. Perubahan data yang tidak ditangani dengan baik akan riskan terjadi kesalahan informasi. Sementara itu, beberapa proses penyimpanan data mempunyai skema penyimpanan yang kompleks sehingga proses tersebut akan sulit untuk ditangani. Untuk itu perlu dibangun sebuah *Human Resource Information System* (HRIS). HRIS yang dibangun bernama Electronic Human Resource atau disingkat EHR. Sistem ini akan digunakan untuk mengatur data pegawai sesuai dengan proses bisnis yang ada pada sebuah perusahaan korporasi.. Dengan adanya sistem EHR maka

dapat diketahui data informasi terkini dari tiap individu pegawai dalam perusahaan.

Pengolahan data akan menjadi fitur utama sistem. Salah satu di dalamnya adalah penyimpanan data sesuai proses bisnis. Data yang disimpan akan melalui skema penyimpanan yang bervariasi seperti pengecekan format data atau perbandingan nilai data. Untuk itu, sistem EHR dikembangkan menggunakan *platform* Joget Workflow sebagai *tools* utama. Berbagai macam metode penyimpanan data telah disediakan oleh Joget Workflow. Setiap metode mempunyai fungsinya masing-masing sesuai dengan skema penyimpanan data yang ditemui. Dalam penerapan beberapa metode akan membutuhkan *query* pendukung pada skema tertentu seperti *query* SQL atau Java. Selain menggunakan Joget Workflow, untuk penampungan data menggunakan DBMS MySQL. Kedua *tools* tersebut akan saling terintegrasi satu sama lain sehingga metode penyimpanan data yang digunakan dapat berfungsi dengan baik.

Berdasarkan uraian di atas, makalah ini bertujuan untuk mengevaluasi penerapan metode penyimpanan data pada Joget Workflow dalam pengembangan sistem EHR. Penerapan ini diharapkan dapat membuat proses pengolahan terutama penyimpanan data pada sistem berfungsi dengan optimal sesuai skema yang ditemui.

II. KAJIAN PUSTAKA

Human Resource Information System (HRIS) dapat didefinisikan sebagai suatu sistem terintegrasi yang digunakan untuk mengumpulkan, menyimpan dan menganalisa informasi tentang sumber daya manusia dalam sebuah organisasi yang terdiri dari database, komputer aplikasi, perangkat keras dan perangkat lunak yang diperlukan untuk mengumpulkan, merekam, menyimpan, mengelola, memberikan, menyajikan dan memanipulasi data untuk fungsi – fungsi sumber daya manusia [3].

Sementara itu, pada penelitian berjudul *Effectiveness of Human Resource Information System on HR Functions of the Organization: A Cross Sectional Study* menjelaskan bahwa penggunaan HRIS pada sebuah organisasi memberikan hasil kerja yang lebih efektif dan lebih cepat. Di sisi lain, HRIS juga membawa perbaikan pada fungsi sumber daya manusia secara keseluruhan, bahkan telah membantu dalam menyelaraskan praktik sumber daya manusia dengan strategi organisasi, sehingga meningkatkan efisiensi fungsi SDM [4].

Pengembangan sistem Electronic Human Resource ini menggunakan *platform* Joget Workflow. Joget Workflow

merupakan *platform* pengembangan sistem *open-source* dan *no-code* yang terkini untuk membangun sebuah sistem web *enterprise* dan otomasi proses bisnis yang dapat digunakan di *cloud* [5]. Penyimpanan data pada Joget Workflow menggunakan beberapa metode antara lain:

- Bean Shell Form Binder adalah metode penyimpanan yang menggunakan implementasi dari Bean Shell. Bean Shell adalah kumpulan fungsi Java yang dapat dimasukkan ke dalam fitur penulisan bahasa dan ditulis ke dalam bahasa pemrograman Java [6]. Penggunaan bahasa Java pada metode ini membuat implementasi penyimpanan data lebih fleksibel. Sehingga selain fungsi untuk menyimpan data, fungsi lain dapat dibuat seperti pengecekan kondisi menggunakan fungsi *if* atau *while*.
- JDBC Binder adalah metode yang digunakan untuk menyimpan atau memuat data dari Joget maupun *database* khusus melalui *query* SQL [7]. Dalam penyimpanan data hanya perlu menulis *query* SQL untuk *select*, *insert*, *update*, dan *delete*. Penggunaan *query* SQL dapat ditambah dengan fungsi yang disediakan oleh SQL.
- Workflow Form Binder adalah *form* untuk proses penyimpanan maupun pemuatan data yang ditangani oleh Form Builder milik Joget. *Form* ini akan menyimpan data secara otomatis tanpa perlu ada tabelnya terlebih dahulu. Tabel dan kolom akan dibuat oleh Joget dengan penamaan yang disesuaikan dengan pengaturan *form* [8]. Penggunaan metode ini praktis ketika ditemui penyimpanan data pada satu tabel tetapi metode ini tidak dapat digunakan ketika penyimpanan data dilakukan pada lebih dari satu tabel atau *join* tabel. Penyimpanan juga tidak bisa berubah-ubah karena metode ini tidak bisa ditambah dengan *query* pendukung.

Selain penyimpanan data, fitur lain Joget Workflow adalah memfasilitasi pembuatan sistem dengan proses bisnis yang ada. Hal ini membantu dalam menjalankan kegiatan operasional UPT Layanan Internasional Universitas Andalas [9]. UPT Layanan Internasional Universitas Andalas yang awalnya masih melakukan pengarsipan berkas secara manual dengan proses bisnis yang panjang dan belum menggunakan teknologi/sistem informasi yang sesuai dengan kebutuhan. Hal tersebut menimbulkan kesulitan dan/atau kelambatan dalam pemrosesan, penyimpanan, dan pengelolaan data. Maka penggunaan otomasi proses bisnis Joget Workflow memudahkan dan meringankan proses bisnis serta meningkatkan efisiensi kinerja di UPT Layanan Internasional Universitas Andalas.

III. METODOLOGI

Pada pengembangan sistem EHR ini dikerjakan bersama dalam sebuah tim yang terdiri dari analis sistem, programmer dan proyek manajer. Pengembangan sistem, khususnya dalam penyimpanan data menggunakan metode pada Joget Workflow, dilakukan beberapa tahapan. Tahapan tersebut sebagai berikut:

- Analisis, yaitu tahap pertama dengan melakukan pengumpulan informasi dengan berdiskusi dengan klien. Informasi yang didapat akan menentukan kebutuhan sistem berkaitan dengan fitur *form* yang akan dibuat dan skema penyimpanan datanya.

- Penentuan Metode, yaitu tahap kedua untuk menentukan metode penyimpanan data pada *form* yang akan dikembangkan. Penentuan ini berdasarkan pada skema penyimpanan data yang ditemui.
- Persiapan Joget Workflow, yaitu tahap ketiga untuk melakukan pemasangan dan konfigurasi *platform* Joget Workflow agar dapat digunakan dalam proses pengembangan sistem dan pembuatan *form*.
- Implementasi Penyimpanan Data, yaitu tahap keempat untuk mengimplementasikan metode penyimpanan data ke dalam *form* yang telah dibuat
- Pengujian, yaitu tahap terakhir untuk memastikan hasil dari implementasi telah sesuai dengan kebutuhan sistem tanpa terdapat *bug* maupun *error*. Pada tahap ini *form* yang telah dibuat akan melalui proses *debugging* menggunakan Tomcat *log* dengan pengisian data secara manual.

IV. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan Sistem

Sistem EHR mempunyai beberapa fitur *form* di antaranya yaitu Profil Saya, Pengurusan Staf Baru, Pengurusan Organisasi, dan Pengurusan Pegawai. Fitur Profil Saya digunakan untuk menampilkan data profil pegawai yang telah masuk. Pada halaman ini pegawai dapat melakukan perubahan data profilnya. Profil diri yang diubah perlu disetujui dari pihak pengelola agar data profil dapat berubah. Fitur kedua adalah Pengurusan Staf Baru dimana data profil pegawai baru akan ditampilkan. Pengelola akan melakukan konfirmasi akhir kepada data tersebut agar data pegawai baru ini terdaftar pada susunan pegawai yang ada. Fitur ketiga adalah Pengurusan Organisasi, fitur ini digunakan untuk mengolah data kelompok kerja yang ada. Fitur Pengurusan Pegawai dimana fitur ini mempunyai beberapa *form* berbeda untuk mengubah data pegawai yang berkaitan dengan informasi kelompok kerjanya. *Form-form* ini antara lain: Kenaikan Pangkat, Memangku, dan Penukaran Penempatan secara Administrasi. Perubahan data pada fitur Pengurusan Pegawai disimpan ke dalam tabel sementara terlebih dahulu. Penyimpanan lanjut ke data pegawai akan dilakukan secara otomatis oleh *cron* Laravel.

Setiap *form* tadi akan menampilkan data yang berbeda-beda sehingga skema penyimpanan pada tiap *form* juga akan berbeda. *Form* pada fitur Profil Saya dan Pengurusan Staf baru akan menemui skema penyimpanan dengan kondisi pengecekan dengan *while* dan *if*. Selanjutnya, *form* pada fitur Pengurusan Organisasi menggunakan skema penyimpanan dengan kondisi *case when* untuk merubah nilai data. Sementara fitur Pengurusan Pegawai mempunyai skema penyimpanan dengan satu tabel secara sementara. Tabel 1. memperlihatkan pembagian fitur *form* dan skema penyimpanan data secara terperinci.

TABLE I. ANALISIS KEBUTUHAN FORM SISTEM

Fitur	Nama Form	Skema Penyimpanan Data
Profil Saya	Informasi Pribadi	Terdapat kondisi pengecekan dengan <i>while</i> dan <i>if</i>
	Informasi Pendidikan	Terdapat kondisi pengecekan dengan <i>while</i> dan <i>if</i>

Fitur	Nama Form	Skema Penyimpanan Data
Pengurusan Staf Baru	Tukar Penempatan	Terdapat kondisi pertukaran nilai data lama dan baru
	Pengesahan Laporan Diri	Terdapat kondisi pengecekan dengan <i>while</i> dan <i>if</i>
Pengurusan Organisasi	Tambah Jabatan	Terdapat kondisi perubahan nilai yang disimpan dengan <i>case when</i>
	Ubah Jabatan	Terdapat kondisi perubahan nilai yang disimpan dengan <i>case when</i>
Pengurusan Pegawai	Kenaikan Pangkat	Penyimpanan data sementara ke dalam satu tabel
	Memangku	Penyimpanan data sementara ke dalam satu tabel
	Penukaran Penempatan secara Administrasi	Penyimpanan data sementara ke dalam satu tabel

B. Penentuan Metode Penyimpanan Data

Setelah mengetahui skema penyimpanan yang digunakan, maka tahap berikutnya adalah melakukan penentuan metode penyimpanan data yang akan digunakan di tiap *form*. Penentuan ini berdasarkan pada tingkat skema yang ditemui. Tingkat skema penyimpanan akan dicocokkan dengan metode penyimpanan yang ada. Setiap metode pada Joget Workflow mempunyai implementasi *query* dan karakteristiknya masing-masing. Tabel II memperlihatkan perbandingan *query* dan karakteristik pada metode penyimpanan data Joget Workflow.

TABLE II. PERBANDINGAN METODE

No	Metode	Query	Karakteristik
1	Bean Shell Form Binder	Java	Fleksibel dalam pemanfaatan fungsi
2	JDBC Binder	SQL	Fungsi terbatas dalam bentuk SQL
3	Workflow Form Binder	-	Penyimpanan statis

Bean Shell Form Binder dapat menangani penyimpanan data dengan skema yang kompleks karena penggunaan Java membuat fungsi yang dibuat lebih fleksibel dan JDBC Binder digunakan untuk menangani penyimpanan data dengan skema yang sederhana. Skema dapat dikatakan kompleks apabila skema tersebut mempunyai kondisi percabangan *if* atau perulangan *while* sedangkan skema yang sederhana hanya membutuhkan *query* dan fungsi dasar SQL seperti *select*, *insert*, *update*, dan *delete*. Metode Workflow Form Binder digunakan pada *form* dengan skema tanpa fungsi *query* karena penyimpanan bersifat statis atau tetap. Penentuan metode dapat dilihat pada Tabel III.

TABLE III. METODE PENYIMPANAN DATA

No	Nama Form	Metode
1	Informasi Pribadi	Bean Shell Form Binder
2	Informasi Pendidikan	Bean Shell Form Binder
3	Tukar Penempatan	Bean Shell Form Binder

No	Nama Form	Metode
4	Pengesahan Laporan Diri	Bean Shell Form Binder
5	Tambah Jabatan	JDBC Binder
6	Ubah Jabatan	JDBC Binder
7	Kenaikan Pangkat	Workflow Form Binder
8	Memangku	Workflow Form Binder
9	Penukaran Penempatan secara Administrasi	Workflow Form Binder

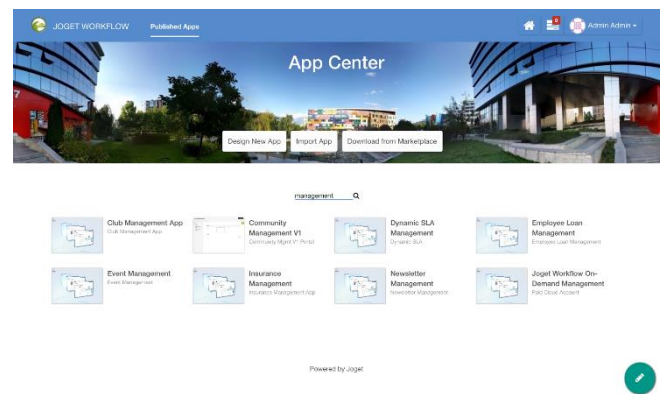
Tabel II memperlihatkan empat *form* dengan metode Bean Shell Form Binder. Hal ini karena *form-form* tersebut memiliki skema penyimpanan data yang kompleks seperti fungsi percabangan *if* dan perulangan *while* untuk pengecekan data. Selanjutnya metode JDBC Binder digunakan pada dua *form* dengan skema yang membutuhkan fungsi *case when*. Fungsi ini dapat dilakukan dalam *query select* SQL pada JDBC Binder. Terakhir, *form* yang menggunakan metode Workflow Form Binder. Metode ini digunakan pada *form* tersebut karena skema penyimpanan yang ada hanya melakukan penyimpanan data sementara pada satu tabel tanpa *query* pendukung.

C. Persiapan Joget Workflow

Beberapa persiapan perlu dilakukan agar Joget Workflow dapat berjalan. Berikut langkah untuk mempersiapkan Joget Workflow:

1) Pemasangan Joget Workflow

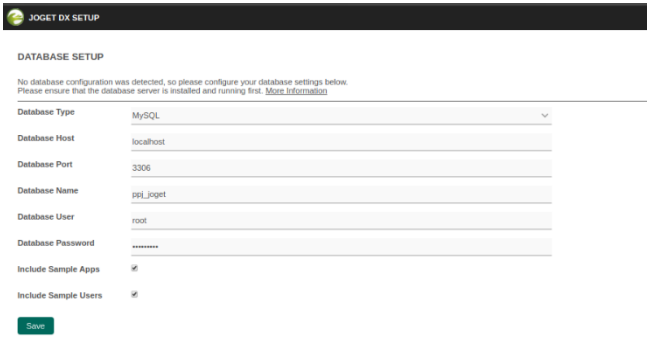
Spesifikasi dalam pemasangan Joget Workflow membutuhkan Java Runtime Environment (JRE) atau Java Development Kit (JDK) versi 8 dan MySQL versi 5.5. Pemasangan dilakukan dengan melakukan pengunduhan *file* Joget Workflow dan menjalankan *file* tersebut. Joget Workflow yang berhasil terpasang dapat diakses dengan tautan *localhost*. Gambar 1 memperlihatkan tampilan awal dari platform Joget Workflow.



Gambar 1. Halaman awal Joget Workflow

2) Konfigurasi Database MySQL

Konfigurasi *database* dilakukan untuk menyambungkan antara aplikasi Joget dengan *database* yang digunakan. Contoh konfigurasi *database* MySQL dapat dilihat pada Gambar 2.

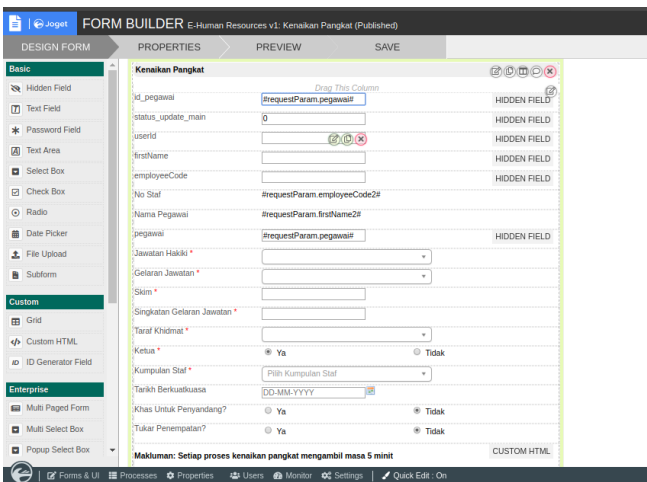


Gambar 2. Halaman konfigurasi *database* MySQL

3) Pembuatan Form

Setelah melakukan konfigurasi *database*, langkah selanjutnya adalah membuat *form*. Gambar 3 menampilkan halaman mode pengembang untuk membuat *form* pada Joget Workflow. Pembuatan *form* dilakukan dengan cara sebagai berikut:

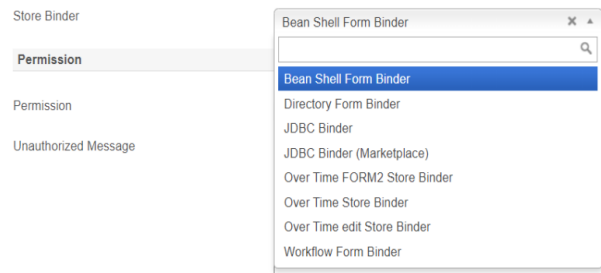
- Tekan tombol konfigurasi Joget Workflow untuk masuk ke dalam mode pengembangan
- Pilih *Form & UI*
- Tekan *Create New Form* untuk membuat *form* baru
- Isi detail *form* sesuai kebutuhan lalu tekan tombol *save*
- Pemilihan *field* yang digunakan dapat dengan menggeser pilihan *field* di bagian kiri *form builder*



Gambar 3. Halaman pembuatan *form*

D. Implementasi Penyimpanan Data

Tahap selanjutnya adalah mengimplementasikan *penyimpanan data* ke dalam tiap *form* yang dibuat. Setiap *form* yang telah dibuat di Joget Workflow mempunyai pengaturan. Salah satu diantaranya digunakan untuk memilih metode penyimpanan data pada *form*. Pengaturan penyimpanan data untuk memilih metode yang akan digunakan pada *form* Joget Workflow terdapat di *Properties Form Builder*. Pengaturan tersebut dapat dilihat pada Gambar 4.



Gambar 4. Pengaturan metode penyimpanan data di *properties form* Joget Workflow

Setelah itu untuk memperoleh penyimpanan data yang diinginkan, maka perlu ditambahkan *query* pendukung sesuai dengan metode yang digunakan. Bean Shell Form Binder menggunakan bahasa pemrograman Java dan JDBC Binder menggunakan *query* SQL. Sementara Workflow Form Binder tidak menggunakan *query* tambahan.

1) Bean Shell Form Binder

Metode Bean Shell Form Binder menggunakan *query* dengan bahasa pemrograman Java. Fungsi yang diperlukan untuk melakukan penyimpanan data adalah *insert* dan *update*. Sebelum itu data dari *form* perlu dipilih terlebih dahulu. Pemilihan disesuaikan dengan *field form* yang ada. Gambar 5. memperlihatkan cara untuk memilih data dari *field form*.

```
newRow.put("new_tel_m", originalRow.getProperty("new_tel_m"));
newRow.put("new_tel_ext", originalRow.getProperty("new_tel_ext"));
```

Gambar 5. Pemilihan *field* pada *form* Informasi Pribadi

Field pada *form* yang dapat diubah adalah *field* *new_tel_m* dan *new_tel_ext*. Fungsi *getProperty* digunakan untuk mendapatkan nilai data yang baru pada kedua *field* tersebut. Selanjutnya dibuat *query* untuk *insert* data. Karena penyimpanan ini mempunyai kondisi untuk membandingkan dengan data yang telah ada maka perlu membuat *query select* data pembanding. *Query insert* dan *select* dapat dilihat pada Gambar 6.

```
String insertQuery = "insert into temp_update_dir_staff_profile(
    userId, columnName,
    new_value, categoryId,
    maklumat, old_value, permohonanId)"
    +" select ?,?,?, 'Peribadi', ?, ?,
    concat('PROFIL',DATE_FORMAT(NOW(), '%d%m%y'),'-',
    lpad(a.jumlah+1,7,0)) from (select count(1) as jumlah
    from profile_main) a";

PreparedStatement istmt = con.prepareStatement(insertQuery);

String selectQuery = "select b.tel_m, b.tel_ext"
    +" from dir_user a"
    +" left join dir_staff_profile b on a.id = b.fk_userId"
    +" where a.id = ?";

PreparedStatement sstmt = con.prepareStatement(selectQuery);
```

Gambar 6. *Query insert* dan *select* pada Informasi Pribadi

Data dari *form* akan disimpan ke dalam tabel *temp_update_dir_staff_profile* dengan beberapa kolom data yang dipilih. Nilai data baru akan disimpan di dalam kolom *new_value*. Setiap *field* pada *form* yang diubah akan disimpan dalam baris yang berbeda. *Query select* pembanding akan memilih data dari kolom *tel_m* dan *tel_ext* pada tabel *dir_staff_profile*. Tabel *dir_staff_profile* akan *join* dengan

tabel `dir_user` untuk mendapatkan `id` pegawai yang melakukan perubahan data.

Setelah `query insert` dan `select` telah siap maka dibuat `query update` untuk memasukkan data ke tabel utama. Pengecekan kondisi akan dilakukan pada setiap `row` yang telah dipilih. Gambar 7 memperlihatkan `query` secara lebih rinci.

```
String user = "#currentUser.id#";
sstmt.setString(1, user);
ResultSet rs = sstmt.executeQuery();
while (rs.next()) {
    if (!row.getProperty("new_tel_m")
        .equalsIgnoreCase(rs.getString("tel_m"))) {
        istmt.setString(1, user);
        istmt.setString(2, "tel_m");
        istmt.setString(3, row.getProperty("new_tel_m"));
        istmt.setString(4, "No Telefon Bimbit");
        istmt.setString(5, rs.getString("tel_m"));
        istmt.executeUpdate();
    }
    if (!row.getProperty("new_tel_ext")
        .equalsIgnoreCase(rs.getString("tel_ext"))) {
        istmt.setString(1, user);
        istmt.setString(2, "tel_ext");
        istmt.setString(3, row.getProperty("new_tel_ext"));
        istmt.setString(4, "No Sambungan");
        istmt.setString(5, rs.getString("tel_ext"));
        istmt.executeUpdate();
    }
}
```

Gambar 7. `Query update` dan pengecekan data pada Informasi Pribadi

Kondisi percabangan `if` diletakkan di dalam kondisi perulangan `while`. Variabel `rs` digunakan sebagai `statement` eksekusi `query` hasil dari `select` pada Gambar 6. Percabangan `if` di setiap `field` pada `form` menggunakan fungsi `equalsIgnoreCase` untuk membandingkan nilai data baru dan lama. Apabila berhasil melewati kondisi `if`, maka variabel `istmt` yang merupakan `query insert` akan mendapatkan nilai terbaru untuk dieksekusi `update` menggunakan fungsi `executeUpdate`.

Penyimpanan data Bean Shell Form Binder ini dilakukan pada `form` Informasi Pribadi. Tampilan dari `form` tersebut dapat dilihat pada Gambar 8.

Gambar 8. Halaman `form` Informasi Pribadi

2) JDBC Binder

Metode yang kedua adalah JDBC Binder. Langkah yang untuk membuat `query` penyimpanan disini hampir sama

dengan `query` di `SQL`. Perbedaannya hanya pada nilai data yang diambil karena `platform` Joget Workflow menyediakan fungsi untuk menyimpan data `realtime` bernama `hash variable`.

Penyimpanan data ini memerlukan `query` dengan fungsi `insert` dan `update`. `Query insert` akan memilih tabel dan kolom yang digunakan untuk menyimpan data. Setelah itu terdapat `query select` untuk memilih data yang ingin disimpan. Jumlah kolom data yang dipilih pada `query insert` dan `select` harus sama. `Query` tersebut dapat dilihat pada Gambar 9.

```
INSERT dir_department(
    id, name, description,
    hod, organizationId,
    parentId, createdBy,
    createdByName, dateCreated,
    org_order)
select
    case when (a.id is null or id = '')
        then 'J001' else id end as id,
    {name}, {description} as description,
    case when {hod} = '' then null
        else {hod} end,
    'ppj_cartaOrganisasi', {parentId},
    '#currentUser.id#' as createdBy,
    '#currentUser.firstName#' as createdByName,
    now() as dateCreated, {org_order} as org_order
from (
    select (
        select concat(
            'J', lpad(substr(dir_department.id,3,3) + 1,3,0)
        ) as id from dir_department
        where id like 'J%' and organizationId = 'ppj_cartaOrganisasi'
        order by dir_department.id desc limit 1) id) a;
```

Gambar 9. `Query` penyimpanan data pada Tambah Jabatan

Pada Gambar 9 `query insert` akan memilih kolom pada tabel `dir_department` sebagai tempat penyimpanan data. Kolom yang dipilih harus sesuai dengan `field` pada `form`. `Query select` digunakan untuk menentukan data yang akan dimasukkan pada `query insert`. Terdapat fungsi `case when` untuk menyimpan data sesuai dengan format yang diinginkan. Pengambilan data `select` juga memanfaatkan `subquery` dengan fungsi `where` untuk memperoleh data dari tabel yang tepat.

Metode JDBC Binder ini digunakan pada `form` Tambah Jabatan. Gambar 10 memperlihatkan tampilan `form` Tambah Jabatan.

Gambar 10. Halaman `form` Tambah Jabatan

3) Workflow Form Binder

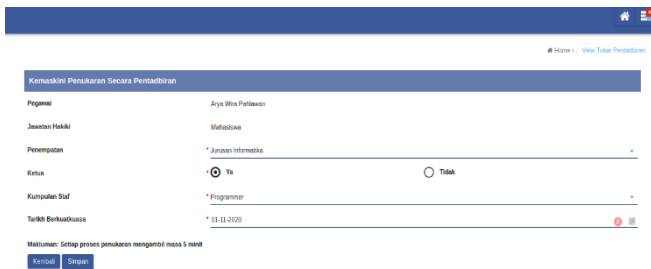
Metode Workflow Form Binder termasuk metode dengan cara penggunaan yang paling mudah diantara metode lain di `platform` Joget Workflow. `Form` yang sudah dibuat hanya perlu diubah cara penyimpanan data -nya menjadi Workflow Form Binder. Proses penyimpanan akan ditangani oleh Joget tanpa memerlukan `query` pendukung. Data yang disimpan akan mempunyai nama tabel dan kolom yang sesuai dengan

form-nya. Akan tetapi penamaan akan diubah sedikit pada bagian awal. Tabel III akan memperlihatkan contoh perbandingan antara nama di *form* dengan nama di *database*.

TABLE IV. PERBANDINGAN NAMA TABEL DAN KOLOM

Jenis	Nama di Form	Nama di Database
Tabel	temp_pendtabiran	app_fd_ temp_pendtabiran
Kolom	positionId	c_positionId
	employeeCode	c_employeeCode
	from_departmentId	c_from_departmentId
	userId	c_userId
	employeeId	c_employeeId
	jobId	c_jobId
	firstName	c_firstName
	jobName	c_jobName
	to_departmentId	c_to_departmentId
	Hod	c_hod
	groupId	c_groupId
	startDate	c_startDate
	status_update_sap	c_status_update_sap
	status_update_main	c_status_update_main

Metode Workflow Form Binder ini digunakan pada *form* Penukaran secara Administrasi. Gambar 11 memperlihatkan tampilan *form* Penukaran secara Administrasi.



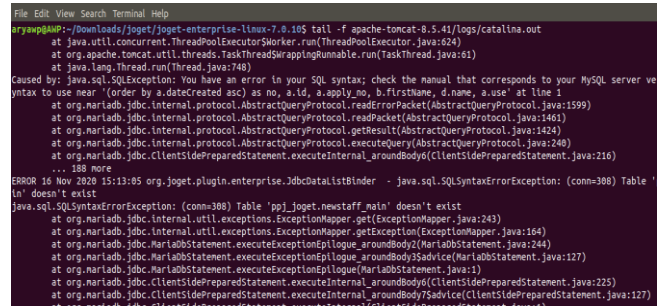
Gambar 11. Halaman *form* Penukaran secara Administrasi

E. Pengujian

Penyimpanan data yang telah dibuat akan melalui proses pengujian terlebih dahulu. Hal ini untuk mengetahui data yang disimpan telah berhasil atau tidak. Pengujian dilakukan dengan melakukan penambahan atau perubahan data *field form* secara manual. Selain itu, pengujian dipantau dengan cara *debugging*. *Debugging* Joget Workflow menggunakan Tomcat *log* dengan menjalankan perintah berikut di terminal.

```
tail -f apache-tomcat-8.5.41/logs/catalina.out
```

Terminal akan menampilkan *log* yang berjalan pada sistem EHR termasuk proses penyimpanan data. Halaman *debugging* ini akan ditampilkan pada Gambar 12. Setelah itu hasil pengujian dapat dilihat pada Tabel V.



Gambar 12. Halaman *debugging* Tomcat *log*

TABLE V. HASIL PENGUJIAN

No	Nama <i>Form</i>	Berhasil/Gagal
1	Informasi Peribadi	Berhasil
2	Informasi Pendidikan	Berhasil
3	Tukar Penempatan	Berhasil
4	Pengesahan Laporan Diri	Berhasil
5	Tambah Jabatan	Berhasil
6	Ubah Jabatan	Berhasil
7	Kenaikan Pangkat	Berhasil
8	Memangku	Berhasil
9	Penukaran Penempatan secara Administrasi	Berhasil

F. Pembahasan

Berdasarkan pemaparan sebelumnya, dapat diketahui bahwa Joget Workflow menyediakan berbagai macam metode untuk melakukan penyimpanan data. Hal yang perlu diperhatikan dalam pemilihan metode ini adalah seberapa kompleks skema yang akan digunakan saat proses penyimpanan data. Skema yang kompleks terdapat kondisi *if* atau perulangan *while*. Skema tersebut akan menggunakan metode Bean Shell Form Binder, dan JDBC Form Binder digunakan untuk skema yang sederhana seperti pemanfaatan *query* dasar SQL. Sementara penyimpanan data tanpa adanya skema akan menggunakan Workflow Form Binder. Pemanfaatan penyimpanan data pada Joget Workflow ini mempermudah proses penyimpanan data sesuai proses bisnis terutama dalam hal pengolahan data. Hal ini karena metode yang digunakan sesuai dengan skema yang ada. Hasil pengujian memperlihatkan bahwa penerapan metode penyimpanan data pada Joget Workflow berhasil dilakukan dalam pengembangan sistem Electronics Human Resource (EHR).

V. KESIMPULAN DAN SARAN

Penyimpanan data dalam Electronics Human Resource (EHR) dapat dikembangkan menggunakan metode yang disediakan oleh *platform* Joget Workflow. Berbagai macam skema penyimpanan yang bervariasi dapat ditangani dengan metode pada Joget Workflow. Skema yang kompleks seperti penggunaan kondisi percabangan *if* atau perulangan *while* dapat menggunakan metode Bean Shell Form Binder. Skema dengan *query* SQL dapat menggunakan JDBC Form Binder, sedangkan untuk skema sederhana yang hanya butuh

penyimpanan data sementara pada satu tabel dapat menggunakan Workflow Form Binder.

Berdasarkan hasil makalah ini, pengembangan aplikasi EHR difokuskan pada *form* penyimpanan data. Pada pengembangan selanjutnya disarankan untuk lebih memaksimalkan pembuatan *form* dengan fitur *workflow* pada Joget Workflow. Hal itu perlu, karena fitur ini digunakan untuk membuat sekumpulan *form* dalam *workflow* yang sama sehingga alur pengiriman data pada proses bisnis akan dapat diketahui dengan baik. Selain itu dalam melakukan *query* khususnya SQL dapat memanfaatkan *stored procedure*. Pemanfaatan *stored procedure* dalam menampung *query* akan mengurangi *traffic* data antara aplikasi dan *database*.

VI. REFERENCES

- [1] Franky and J. Devitra, "Analisis dan Perancangan Sistem Informasi," *Jurnal Manajemen Sistem Informasi*, vol. 1, no. 2, pp. 185-190, 2016.
- [2] H. F. Muhammad and P. S. Niki, "Pengembangan Human Resource Information System (HRIS)," *JUPITER (Jurnal Pendidikan Teknik Elektro)*, vol. 03, no. 02, pp. 1-12, 2018.
- [3] S. N. Ambo and M. Ghufron, "Rancang Bangun Aplikasi Human Resource Information System(HRIS) Menggunakan Metode Model View Controller(MVC)," *Seminar Nasional Sains dan Teknologi*, 2015.
- [4] S. Shiri, "Effectiveness of Human Resource Information System on HR Functions of the Organization: A Cross Sectional Study," *US-China Education Review*, vol. A 9, pp. 830-839, 2012.
- [5] Joget Workflow, "Joget Workflow - Joget: Open Source Workflow and Low Code Platform for Digital Transformation," 2019. [Online]. Available: <https://www.joget.com/joget-workflow.html>. [Accessed 23 April 2020].
- [6] Administrator and Justin, "Bean Shell Form Binder - Knowledge Base for v6 - Joget | COMMUNITY," 15 Januari 2018. [Online]. Available: <https://dev.joget.org/community/display/KBv6/Bean+Shell+Form+Binder>. [Accessed 24 April 2020].
- [7] Justin and Andrew, "JDBC Form Binder - Knowledge Base for v6 - Joget | COMMUNITY," 29 Oktober 2019. [Online]. Available: <https://dev.joget.org/community/display/KBv6/JDBC+Form+Binder>. [Accessed 24 April 2020].
- [8] Administrator and Justin, "Workflow Form Binder - Knowledge Base for v6 - Joget | COMMUNITY," 23 Agustus 2017. [Online]. Available: <https://dev.joget.org/community/display/KBv6/Workflow+Form+Binder>. [Accessed 24 April 2020].
- [9] K.-I. O. David, "Implementasi Joget Workflow Pada Proses Bisnis Unit Pelaksana Teknis (UPT) Layanan Internasional Universitas Andalas," pp. 128-129, 2017.