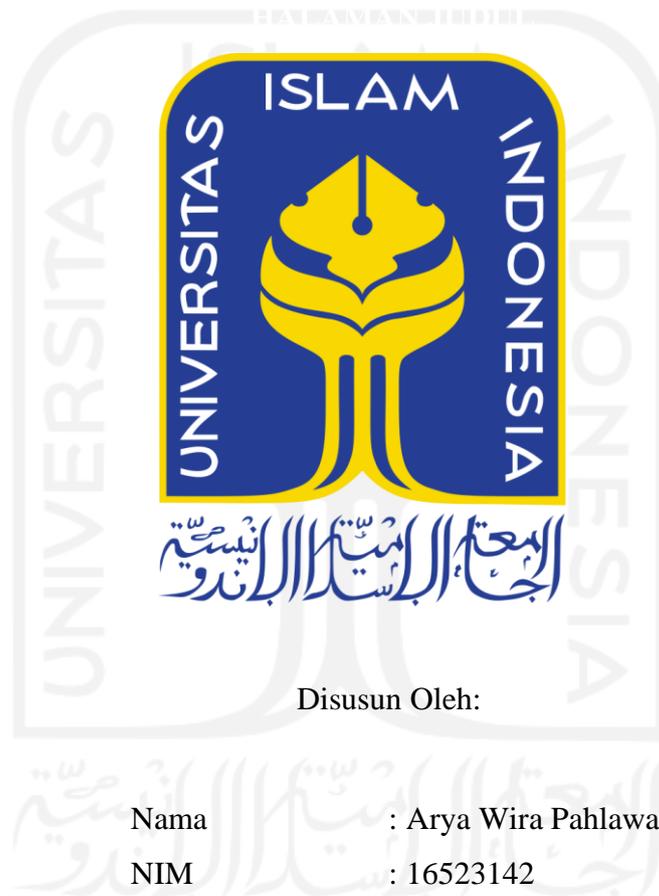


SKRIPSI

METODE PENYIMPANAN DATA PADA JOGET WORKFLOW (STUDI KASUS: WEBSITE ELECTRONIC HUMAN RESOURCE)



Disusun Oleh:

Nama : Arya Wira Pahlawan

NIM : 16523142

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

2020

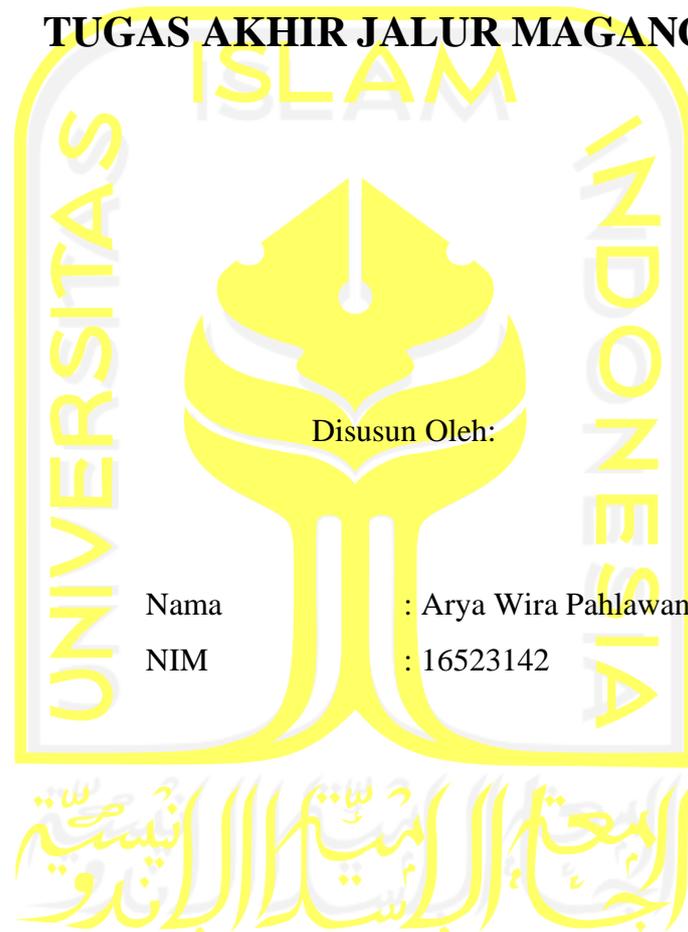
HALAMAN PENGESAHAN DOSEN PEMBIMBING

METODE PENYIMPANAN DATA PADA JOGET

WORKFLOW (STUDI KASUS: WEBSITE

ELECTRONIC HUMAN RESOURCE)

TUGAS AKHIR JALUR MAGANG



Yogyakarta, 10 Desember 2020

Pembimbing,

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

HALAMAN PENGESAHAN DOSEN PENGUJI**METODE PENYIMPANAN DATA PADA JOGET
WORKFLOW (STUDI KASUS: WEBSITE
ELECTRONIC HUMAN RESOURCE)****TUGAS AKHIR JALUR MAGANG**

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

Yogyakarta, 11 Januari 2021

Tim Penguji

Ketua Penguji

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

Anggota 1

Hendrik, S.T., M.Eng.

Anggota 2

Moh. Idris, S.Kom., 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 : Arya Wira Pahlawan
NIM : 16523142

Tugas akhir dengan judul:

METODE PENYIMPANAN DATA PADA JOGET WORKFLOW (STUDI KASUS: WEBSITE ELECTRONIC HUMAN RESOURCE)

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, 10 Desember 2020



(Arya Wira Pahlawan)

HALAMAN PERSEMBAHAN

Skripsi ini saya persembahkan kepada kedua orang tua yang saya cintai, Bapak Nugroho dan Ibu Atik Hidayati. Serta keempat saudara saya Mas Yudha, Mas Akbar, Uma dan Panca.

Skripsi ini juga saya persembahkan kepada keluarga Hexadecima 16 dan teman-teman yang tak bisa disebutkan satu persatu. Serta tenaga kesehatan yang telah berjuang di garis terdepan selama pandemi Covid-19 berlangsung.



HALAMAN MOTO

“Mahkota seseorang adalah akalnya, derajat seseorang adalah agamanya, sedangkan kehormatan seseorang adalah budi pekertinya.”

(Umar bin Khattab)

“Sebaik-baiknya manusia adalah yang paling bermanfaat bagi orang lain”

(HR Ahmad dan Thabrani).

“I’m only brave when I have to be.

Being brave doesn’t mean you go looking for trouble.”

(Mufasa)

“Live as if you were to die tomorrow. Learn as if you were to live forever.”

(Mahatma Gandhi)

UNIVERSITAS ISLAM INDONESIA
البعثة الإسلامية الأندلسية

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah, pertama penulis haturkan puji dan syukur kepada Allah SWT yang telah melimpahkan rahmat dan taufiq serta hidayat-Nya karena atas pertolongan-Nya saya dapat menyelesaikan skripsi ini dengan baik. Tak lupa sholawat dan salam penulis ucapkan kepada Nabi Muhammad SAW, yang telah membawa kita dari zaman jahiliyah menuju zaman yang terang benderang seperti saat ini.

Adapun skripsi ini disusun untuk memenuhi salah satu syarat kelulusan saya di Program Sarjana Informatika Universitas Islam Indonesia. Dalam proses penyusunan skripsi ini saya mendapatkan banyak sekali bantuan, bimbingan, serta dukungan dari berbagai pihak, sehingga dalam kesempatan ini saya juga bermaksud menyampaikan rasa terima kasih kepada:

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D. selaku Rektor Universitas Islam Indonesia.
2. Bapak Hari Purnomo, Prof., Dr., Ir., M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Program Studi Informatika Program Sarjana.
4. Bapak Hanson Prihantoro Putro, S.T., M.T. selaku dosen pembimbing yang membimbing dalam penyusunan laporan akhir ini hingga selesai.
5. Bapak Nugroho dan Ibu Atik Hidayati, selaku kedua orang tua saya yang senantiasa memberi dukungan, semangat, dan doa kepada saya.
6. Kakak Adi Yudha Permana, Kakak Akbar Juang Saputra, Adik Rukma Hermawan dan Adik Panca Nusa Bakti yang selalu mendoakan dan memberikan dukungan kepada saya.
7. Bapak Wisnu Manupraba, dan rekan-rekan PT Javan Cipta Solusi yang telah memberikan kesempatan magang dan membimbing saya selama pelaksanaan magang.
8. Seluruh dosen-dosen Program Sarjana Informatika yang telah memberikan ilmu dan arahan selama menjalani pendidikan ini.
9. Seluruh rekan-rekan Informatika 2016, yang telah membantu, menyemangati dan mengingatkan satu sama lain.

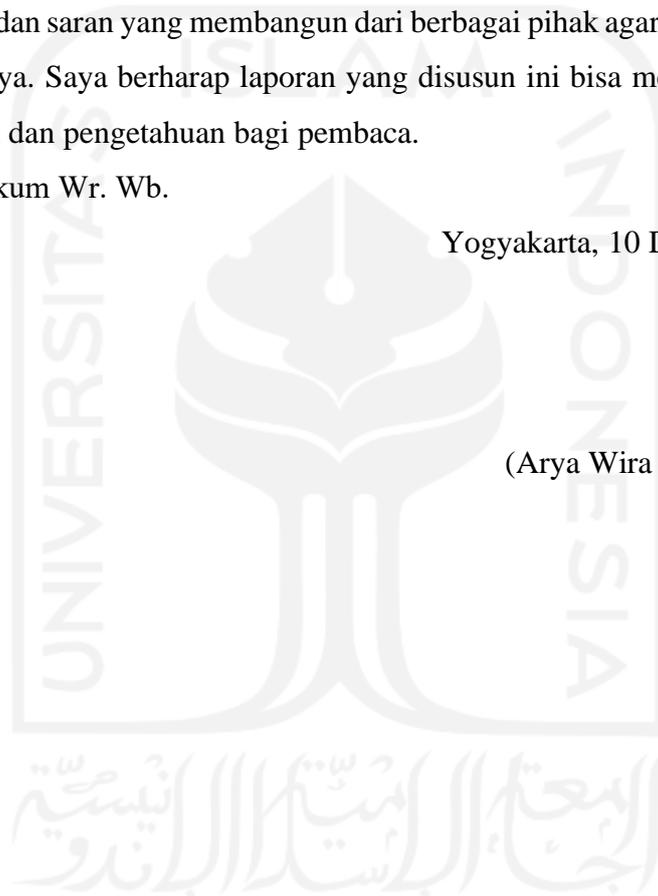
10. Owen Prasimsha, Muhammad Rizky, Elang Sartika, Saputro Andi, Mahisa Agni dan Puspita Dewi rekan seperjuangan magang yang telah mewarnai pelaksanaan magang saya.
11. Seluruh sahabat, rekan dan semua pihak yang tidak dapat saya sebutkan satu persatu atas dukungan dan doanya.

Saya menyadari bahwa laporan tugas akhir ini masih memiliki banyak kekurangan oleh karena itu diperlukan kritik dan saran yang membangun dari berbagai pihak agar laporan ini dapat menjadi lebih baik kedepannya. Saya berharap laporan yang disusun ini bisa memberikan manfaat untuk menambah wawasan dan pengetahuan bagi pembaca.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 10 Desember 2020

(Arya Wira Pahlawan)



SARI

Program Sarjana Studi Informatika UII menggunakan kurikulum 2016 dalam kegiatan pembelajaran. Perbedaan kurikulum 2016 dengan kurikulum-kurikulum sebelumnya terdapat pada tahun keempat yaitu pemilihan jalur yang diminati oleh mahasiswa. Jalur Magang merupakan salah satu dari lima jalur yang disediakan. Magang dapat dilakukan di perusahaan mitra Informatika UII, salah satunya adalah PT Javan Cipta Solusi (Javan). Salah satu proyek yang dikerjakan selama pelaksanaan magang di Javan adalah proyek Electronic Human Resource. 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. Sistem akan melakukan pengolahan data dengan berbagai macam skema perubahan data pada *form*. Hal tersebut akan membuat proses penyimpanan data akan sulit untuk ditangani. Maka dari itu, pengembangan proyek ini memerlukan pembuatan *form* dengan mekanisme penyimpanan data yang efektif. Solusinya adalah penyimpanan data pada *platform* bernama Joget Workflow. Proses penyimpanan data yang bervariasi sesuai proses bisnis dapat ditangani oleh *platform* ini. Joget Workflow mempunyai fitur pengembangan sistem berbasis proses bisnis dengan berbagai macam metode penyimpanan data. Pengembangan EHR akan menerapkan tiga metode penyimpanan pada Joget Workflow. Penerapan metode ini akan melalui beberapa tahapan, mulai dari tahap analisis hingga tahap pengujian. Setelah dilakukan pengembangan dapat disimpulkan bahwa penerapan *form* menggunakan metode penyimpanan data pada Joget Workflow cocok digunakan dalam pengembangan sistem dengan fungsionalitas *form* terutama penyimpanan data yang bervariasi.

Kata kunci: Electronic Human Resource, penyimpanan data, Joget Workflow.

GLOSARIUM

EHR	Electronic Human Resource merupakan sistem manajemen data pegawai pada sebuah perusahaan
Skema	suatu rencana atau struktur yang memuat gambaran umum tentang bagaimana tujuan dapat dicapai
Workflow	suatu proses kerja atau bisnis yang sistematis dimana dokumen atau informasi yang dibuat, dialirkan dari satu pihak ke pihak yang lain untuk tindakan lanjutan menurut suatu aturan atau prosedur tertentu yang telah disepakati bersama dalam sebuah organisasi
Korporasi	perusahaan atau badan usaha yang sangat besar atau beberapa perusahaan yang dikelola dan dijalankan sebagai satu perusahaan besar
Proses Bisnis	suatu kumpulan aktivitas atau pekerjaan terstruktur yang saling terkait untuk menyelesaikan suatu masalah tertentu atau yang menghasilkan produk atau layanan (demi meraih tujuan tertentu)
Platform	kombinasi antara sebuah arsitektur perangkat keras dengan sebuah kerangka kerja perangkat lunak (termasuk kerangka kerja aplikasi)
Query	bahasa komputer yang digunakan untuk melakukan permintaan terhadap basis data dan sistem informasi
Open Source	sistem pengembangan yang tidak dikoordinasi oleh suatu individu/lembaga pusat, tetapi oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber
Form	halaman sistem yang digunakan untuk menerima masukan dari pengguna
Debugging	sebuah metode yang dilakukan oleh para pemrogram dan pengembang perangkat lunak untuk menganalisis alur kerja program, mencari dan mengurangi bug, atau kerusakan di dalam sebuah program komputer atau perangkat keras sehingga perangkat tersebut bekerja sesuai dengan harapan
Bug	suatu cacat desain pada perangkat keras atau perangkat lunak yang mengakibatkan terjadinya galat pada peralatan atau program sehingga tidak berfungsi sebagaimana mestinya

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	2
1.4 Manfaat	3
1.5 Sistematika Penulisan	3
BAB II KAJIAN PUSTAKA	4
2.1 Sistem Informasi Sumber Daya Manusia	4
2.2 Platform/Framework Aplikasi Web	4
2.3 Joget Workflow	6
2.4 Query SQL	12
2.5 Apache Tomcat	13
BAB III PELAKSANAAN MAGANG	14
3.1 Electronic Human Resource	14
3.1.1 Dashboard	15
3.1.2 Pengurusan Staf Baru	16
3.1.3 Pengurusan Organisasi	16
3.1.4 Pengurusan Pegawai	17
3.1.5 Profil Pegawai	18
3.2 Penyimpanan Data pada Joget Workflow	20
3.2.1 Analisis Kebutuhan Sistem	20
3.2.2 Penentuan Metode Penyimpanan Data	23
3.2.3 Persiapan Joget Workflow	24
3.2.4 Pembuatan Form	27
3.2.5 Implementasi Penyimpanan Data	30
3.2.6 Pengujian	38
BAB IV REFLEKSI PELAKSANAAN MAGANG	40
4.1 Teknis	40
4.1.1 Penyimpanan Data pada Joget Workflow	40
4.1.2 Perbandingan Solusi dengan Alternatif Lain	42
4.2 Non Teknis	43
4.2.1 Manfaat Magang	43
4.2.2 Hambatan dan Tantangan Magang	44

4.2.3 Kontribusi Selama Magang	45
BAB V KESIMPULAN DAN SARAN	46
5.1 Kesimpulan	46
5.2 Saran	46
DAFTAR PUSTAKA	47



DAFTAR TABEL

Tabel 2.1 Contoh <i>Query</i> SQL.....	9
Tabel 2.2 Perbandingan Metode	12
Tabel 3.1 Fitur EHR.....	15
Tabel 3.2 Analisis Kebutuhan Sistem	21
Tabel 3.3 Pencocokan Karakteristik <i>Form</i> dan Metode.....	23
Tabel 3.4 Metode Penyimpanan Data	24
Tabel 3.5 Perbandingan Nama Tabel dan Kolom.....	37
Tabel 3.6 Hasil Pengujian.....	38
Tabel 4.1 Perbandingan antar <i>Framework/Platform</i>	42



DAFTAR GAMBAR

Gambar 2.1 Halaman Process Builder	7
Gambar 2.2 Pengaturan Bean Shell Form Binder.....	8
Gambar 2.3 Pengaturan JDBC Binder	9
Gambar 2.4 Pengaturan Workflow Form Binder.....	10
Gambar 2.5 Pengaturan Directory Form Binder.....	11
Gambar 3.1 Halaman <i>Dashboard</i>	16
Gambar 3.2 Halaman Susunan Laporan Diri	16
Gambar 3.3 Halaman Pengurusan Organisasi	17
Gambar 3.4 Halaman Pengurusan Pegawai.....	18
Gambar 3.5 Halaman Profil.....	19
Gambar 3.6 Perintah <i>Create Database MySQL</i>	25
Gambar 3.7 Halaman Konfigurasi <i>Database</i>	26
Gambar 3.8 Halaman Awal <i>Platform</i> Joget Workflow.....	26
Gambar 3.9 Tombol Mode Pengembang	27
Gambar 3.10 Tombol <i>Form & UI</i>	27
Gambar 3.11 Tombol <i>Create New Form</i>	28
Gambar 3.12 Detail <i>Create New Form</i>	28
Gambar 3.13 Desain <i>Form</i> Joget	29
Gambar 3.14 Pratinjau <i>Form</i> Joget.....	29
Gambar 3.15 Pengaturan <i>Form Builder</i>	30
Gambar 3.16 Pemilihan <i>Field</i> pada <i>Form</i> Informasi Pribadi	31
Gambar 3.17 <i>Query Insert</i> dan <i>Select</i> pada Informasi Pribadi	31
Gambar 3.18 Relasi Tabel pada <i>Form</i> Profil Pribadi.....	32
Gambar 3.19 <i>Query Update</i> dan Pengecekan Data pada Informasi Pribadi	33
Gambar 3.20 Halaman <i>Form</i> Informasi Pribadi	34
Gambar 3.21 <i>Query</i> Penyimpanan Data <i>Form</i> Tambah Jabatan	35
Gambar 3.22 Halaman <i>Form</i> Tambah Jabatan	36
Gambar 3.23 Penyimpanan Data dengan Workflow Form Binder.....	36
Gambar 3.24 Halaman <i>Form</i> Penukaran secara Administrasi	37
Gambar 3.25 Halaman <i>Debugging Log</i> Tomcat.....	38

Gambar 4.1 Sintaks *Update* PHP Mata Kuliah PABW41
Gambar 4.2 *Query* Mata Kuliah PSI.....41



BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini, Program Studi Informatika UII terutama program sarjana menggunakan kurikulum 2016 dalam kegiatan pembelajaran. Kurikulum 2016 mempunyai perbedaan dibandingkan dengan kurikulum-kurikulum sebelumnya. Salah satu hal yang membedakan adalah pada tahun keempat perkuliahan mahasiswa dapat memilih jalur yang diminati. Jalur magang merupakan salah satu jalur yang disediakan oleh pihak program studi. Pelaksanaan magang dilakukan minimal selama enam bulan. Berbekal ilmu pengetahuan selama menjalani perkuliahan, mahasiswa akan diterjunkan ke dalam proyek. Mahasiswa dapat mengajukan diri ke perusahaan magang melalui prosedur yang telah ditetapkan.

Pihak Program Studi Informatika mempunyai beberapa mitra kerja yang dapat dijadikan tempat pelaksanaan magang. Salah satu mitra yang bekerja sama adalah PT Javan Cipta Solusi (Javan). Javan merupakan *software house* yang didirikan pada tahun 2008 dan kantor cabangnya terletak di Sleman. Layanan yang disediakan oleh Javan untuk optimalisasi bisnis adalah *Product Development, Web and Mobile App Development, Data Analytics dan Business Process Optimisation Consultant*. Banyak klien yang berminat dengan layanan tersebut dan mempercayakan pengerjaan sistem perangkat lunak kepada Javan. Salah satu klien Javan adalah 3FResource dari Malaysia dengan proyek Electronic Human Resource (EHR). EHR merupakan sebuah sistem informasi sumber daya manusia atau *Human Resources Information System (HRIS)* yang digunakan untuk mengelola data di dalam sebuah perusahaan koorporasi.

Sistem EHR akan mengelola data termasuk data sumber daya manusia. Sumber daya manusia (SDM) terutama pegawai sangat penting dalam sebuah perusahaan karena pegawai merupakan orang yang menggunakan dan mengelola sumber daya lainnya untuk menjalankan perusahaan (Franky & Devitra, 2016). Banyaknya jumlah informasi yang harus dikelola berhubungan dengan SDM akan memunculkan permasalahan dengan tingkat kompleksitas tinggi (Muhammad & Niki, 2018). Seiring berjalannya waktu, informasi data pegawai tentu akan mengalami perubahan sehingga fungsionalitas proses bisnis menggunakan *form* akan menjadi fitur utama sistem EHR. Salah satu fungsi *form* adalah melakukan penyimpanan data sesuai proses bisnis. Sistem EHR dikembangkan menggunakan *platform* Joget Workflow. Joget Workflow

merupakan *platform* pengembangan sistem *open-source* dan *low-code* yang terkini untuk membangun sebuah sistem web *enterprise* dan otomasi proses bisnis yang dapat digunakan di *cloud* (Joget Workflow, 2019). Berbagai macam metode penyimpanan data telah disediakan oleh Joget Workflow tapi dalam pengembangan EHR akan memanfaatkan tiga metode penyimpanan data. Pada metode tertentu akan diperlukan penulisan *query* seperti *query* SQL atau Java. Penerapan metode penyimpanan data pada Joget Workflow dengan menggunakan *query* yang sesuai diharapkan dapat membuat proses penyimpanan data pada sistem berfungsi dengan optimal.

1.2 Ruang Lingkup Magang

Penulis melaksanakan magang di PT. Javan Cipta Solusi atau biasa disebut Javan, selama enam bulan, mulai dari bulan Oktober 2019 hingga Maret 2020. Javan merupakan salah satu mitra kerja dari Program Studi Informatika UII. Perusahaan ini bergerak di bidang teknologi informasi khususnya dalam pengembangan perangkat lunak (*software house development*). Selama pelaksanaan penulis telah terlibat dalam beberapa proyek pengembangan perangkat lunak. Adapun aktivitas yang dilakukan selama magang sebagai berikut:

- a. Proyek Electronic Human Resource (EHR). Proyek ini merupakan proyek pertama penulis dalam pelaksanaan magang. Saat penulis mulai bergabung, proyek EHR telah berjalan selama kurang lebih 1 bulan. Pengembangan perangkat lunak ini menggunakan *tool* yang bernama Joget Workflow. Aktivitas yang dikerjakan adalah membuat berbagai *form*, mendesain *query*, mengatur hak akses pengguna, dan membuat eReport dengan JasperReport.
- b. Proyek Acapela. Penulis terlibat dalam pengujian dan perbaikan *bug* kecil selama pengembangan.
- c. Proyek Sempel Ombudsman. Penulis ikut terlibat dalam melakukan pembuatan eReport surat dengan JasperReport.
- d. Proyek Pengembangan dan Pembinaan SDM PBJ. Penulis terlibat dalam melakukan perbaikan *bug* kecil dan pembuatan eReport dengan JasperReport.

1.3 Tujuan

Tujuan dari magang ini yaitu:

1. Mengembangkan sistem dengan fungsionalitas proses bisnis pada *form* terutama penyimpanan data menggunakan Joget Workflow.
2. Mengetahui bagaimana melakukan penyimpanan data dengan metode pada Joget Workflow.
3. Mengetahui bagaimana membuat *query* penyimpanan data pada Joget Workflow sesuai metode yang digunakan.

1.4 Manfaat

Manfaat dari magang ini yaitu:

1. Mempermudah pengembangan sistem dengan fungsionalitas proses bisnis pada *form*.
2. Memberikan pengetahuan tentang bagaimana melakukan penyimpanan data dan membuat *query* pada Joget Workflow.

1.5 Sistematika Penulisan

Sistematika penulisan berisikan urutan penulisan tugas akhir ini:

1. BAB I Pendahuluan

Bab ini membahas mengenai latar belakang penulisan laporan tugas akhir, ruang lingkup aktivitas magang, tujuan dan manfaat.

2. BAB II Kajian Pustaka

Bab ini membahas mengenai kajian pustaka yang berkaitan dengan HRIS dan Joget Workflow yang digunakan dalam proyek magang dengan beberapa ringkasan penelitian sebelumnya.

3. BAB III Pelaksanaan Magang

Bab ini membahas aktivitas yang telah dilakukan saat pelaksanaan magang, penjelasan proyek apa yang dikerjakan dan pembahasan mengenai metode penyimpanan data pada Joget Workflow.

4. BAB IV Refleksi Pelaksanaan Magang

Bab ini membahas perbandingan implementasi proyek dengan teori yang ada.

5. BAB V Kesimpulan dan Saran

Bab ini menyampaikan kesimpulan dari implementasi proyek dan saran terhadap pengembangan selanjutnya.

BAB II

KAJIAN PUSTAKA

2.1 Sistem Informasi Sumber Daya Manusia

Sistem Informasi Sumber Daya Manusia atau *Human Resources Information System (HRIS)* dapat didefinisikan sebagai suatu sistem terintegrasi yang digunakan untuk mengumpulkan, menyimpan dan menganalisa informasi tentang sumber daya manusia dalam sebuah organisasi. HRIS terdiri dari basis data, 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 (Ambo & Ghufron, 2015).

Penggunaan HRIS disebutkan dapat memperlancar proses komunikasi dan dapat menghemat biaya serta dapat merubah paradigma dalam organisasi yang semula tradisional menjadi terfokus pada strategi perusahaan. Ketidaktahuan organisasi dalam menyadari manfaat penggunaan HRIS menjadi akar permasalahan ini. Untuk itu dibuat percobaan pada suatu perusahaan di Bangladesh dengan menerapkan sistem HRIS. Pengembangan HRIS yang digunakan sesuai dengan cakupan fungsi dan biaya pada perusahaan tersebut. Secara bertahap penggunaan HRIS menjadi populer dan diterima di Bangladesh. Perusahaan menganggap HRIS sebagai investasi daripada tanggungan biaya karena diketahui bermanfaat dalam pengelolaan data di dalam perusahaan (Jahan, 2014).

Sementara itu, (Shiri, 2012) pada penelitiannya menjelaskan bahwa penggunaan HRIS pada sebuah organisasi juga 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.

2.2 Platform/Framework Aplikasi Web

Framework aplikasi web adalah jenis *framework* yang dirancang khusus untuk membantu pengembang dalam membangun sebuah aplikasi web. *Framework* ini biasanya menyediakan fungsionalitas umum yang dimiliki sebagian besar aplikasi web, seperti: manajemen pengguna, persistensi data, dan kerangka sistem. Dengan menggunakan *framework* yang sesuai, pengembang dapat menghemat waktu untuk membangun situs web (DocForge, 2015). Terdapat dua *framework* aplikasi web yang sering digunakan, yaitu:

1. CodeIgniter

Codeigniter adalah sebuah aplikasi *open-source* yang berupa kerangka kerja atau *framework* untuk membangun *website* menggunakan bahasa pemrograman PHP. Tujuannya memungkinkan pengembangan proyek yang lebih cepat daripada penulisan kode dasar atau kode terstruktur, dengan menyediakan banyak *library* yang biasanya digunakan dalam pengerjaan. Antarmuka yang sederhana dan struktur logika untuk mengakses *library* ini membuat Codeigniter mudah digunakan dan dipelajari. Codeigniter ditulis atau dibuat oleh Ellis Lab dan dirilis pertama kali pada 28 Februari 2006 (Sulistiono, 2018). *Framework* ini mempunyai beberapa kelebihan, yaitu (Destiningrum & Adrian, 2017):

- Gratis (*Open-Source*), *framework* Codeigniter memiliki lisensi dibawah Apache/BSD *open-source* sehingga bersifat bebas atau gratis.
- Berukuran kecil, ukuran yang kecil dalam eksekusi maupun penyimpanannya merupakan keunggulan tersendiri jika dibandingkan *framework* lain.
- Menggunakan konsep MVC (*Model, View, Controller*).

2. Laravel

Laravel merupakan *framework* PHP yang rilis di bawah lisensi MIT dan dibangun dengan konsep MVC (*Model, View, Controller*). Laravel dirancang untuk meningkatkan kualitas perangkat lunak, memudahkan proses autentikasi, memudahkan *routing*, memudahkan akses, dan menambah kekuatan dalam *framework* web (Parkar, Shinde, Gadade, & Shinde, 2015). Laravel menyediakan beberapa fitur, yaitu (Aminudin, 2015):

- *Bundles*, yaitu fitur pengemasan modular.
- *Eloquent ORM*, metode untuk mengatasi hubungan objek pada *database*.
- *Application Logic*, logika yang terdapat pada bagian *controller*.
- *Reverse Routing*, mendefinisikan relasi antara *route* dan *link*.
- *Restful Controllers*, memisahkan logika dalam melayani HTTP GET and POST.
- *Class Auto Loading*, menyediakan *loading* otomatis untuk PHP *class*.
- *View Composer*, adalah kumpulan kode yang dieksekusi ketika *view* sedang berjalan.
- *Migration*, menyediakan kontrol untuk mengintegrasikan data dari *database*.
- *Unit Testing*, pengujian untuk mendeteksi dan mencegah *bug*.
- *Automatic Pagination*, menyederhanakan tugas dari penerapan halaman.

Selain *framework*, terdapat *platform* yang dapat digunakan dalam pengembangan aplikasi terutama aplikasi web. Ruang lingkup yang ditangani *platform* lebih luas karena *platform* merupakan kombinasi antara perangkat keras dan perangkat lunak. Contoh *platform* yang tersebut, yaitu:

1. Outsystem

Outsystem merupakan *platform* yang menyediakan solusi pengembangan aplikasi secara cepat. Teknologi Outsystem sangat fleksibel karena dapat melakukan *built, integrate, deploy, dan manage* untuk berbagai perangkat. Outsystem mempunyai beberapa komponen diantaranya (OutSystems, 2020):

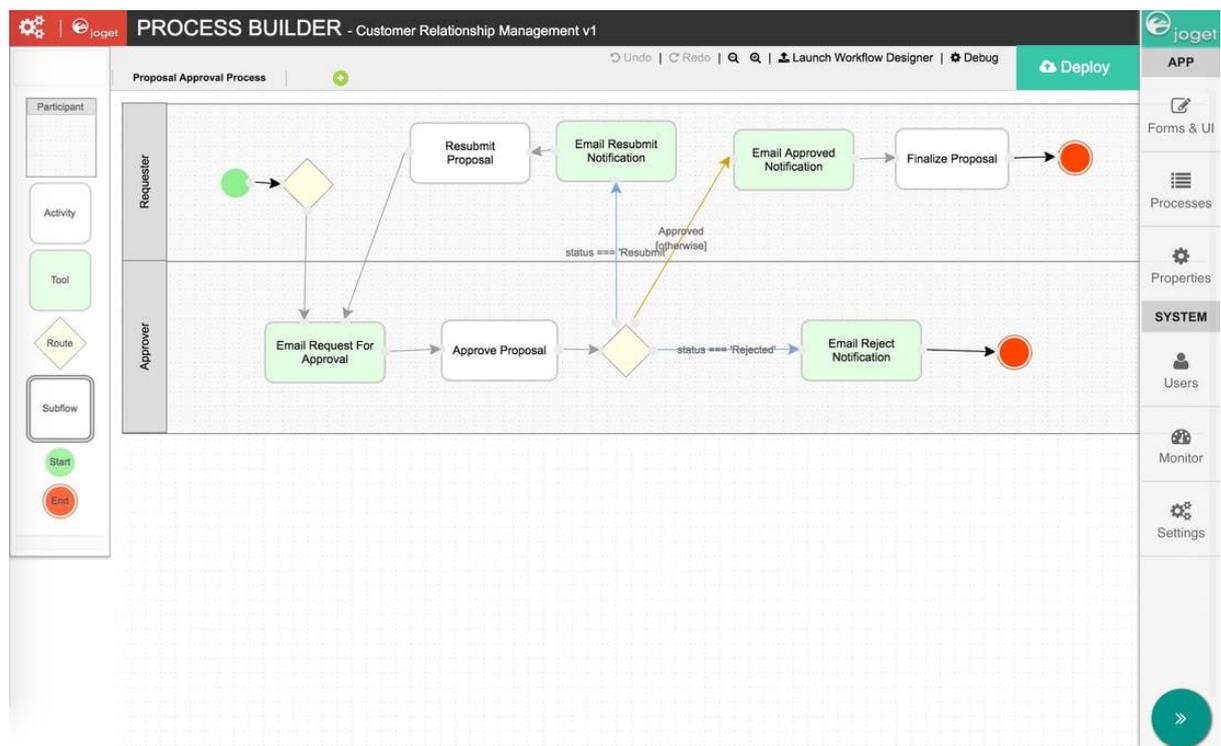
- Platform Server, komponen untuk mengatur semua aktivitas *runtime, deployment*, dan manajemen pada semua aplikasi.
- LifeTime, konsol yang terpusat untuk mengelola infrastruktur, *environment*, aplikasi, IT users, dan *security*.
- Service Center, konsol administrasi untuk *environment* dari infrastruktur
- Service Studio, alat pengembangan visual untuk membuat, merubah, dan memasang aplikasi.
- Integration Studio, alat pengembangan untuk membuat koneksi antara aplikasi yang dibuat dengan Outsystem dengan sistem *enterprise* lain.

2. Joget Workflow

Platform Joget Workflow akan dijelaskan pada **Subbab 2.3**.

2.3 Joget Workflow

Joget Workflow merupakan *platform* pengembangan sistem *open-source* dan *low-code* yang terkini untuk membangun sebuah sistem web *enterprise* dan otomasi proses bisnis yang dapat digunakan di *cloud* (Joget Workflow, 2019). Proses pengembangan dapat dilakukan melalui web *browser* di tautan *localhost* dengan masuk ke dalam mode pengembang Joget Workflow. *Platform* ini bersifat *low-code* karena *platform* menyediakan otomasi proses bisnis dengan Workflow. Workflow yang dibangun akan memudahkan dalam mengatasi proses bisnis karena Workflow ini dapat membangun alur proses dengan *form* secara *drag&drop* di halaman Process Builder. **Gambar 2.1** memperlihatkan halaman pembuatan *workflow* dengan Process Builder.

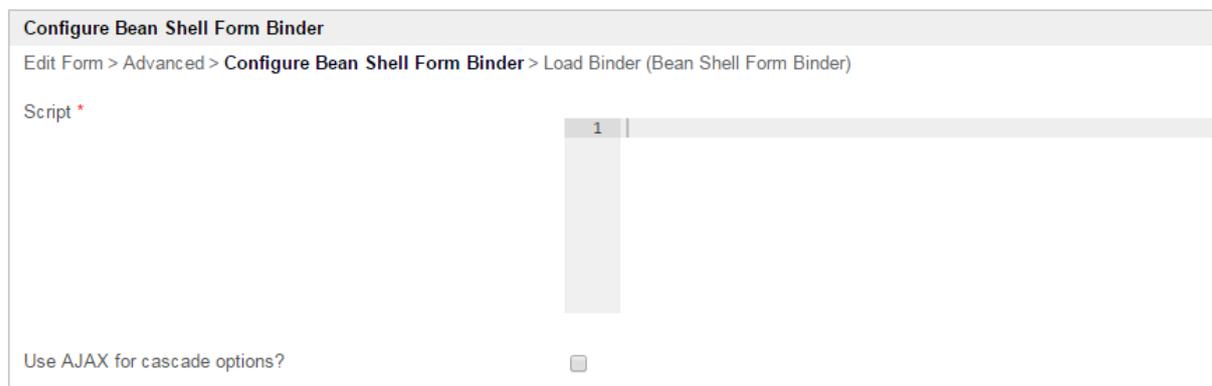


Gambar 2.1 Halaman Process Builder (Joget Workflow, 2019)

Penyimpanan data pada Joget Workflow akan disimpan di *database* yang sama, dalam hal ini menggunakan MySQL. Fungsionalitas *form* terutama penyimpanan data pada Joget Workflow mempunyai empat metode antara lain:

1. Bean Shell Form Binder

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 (Administrator & Justin, 2018). Pengaturan *form binder* dapat dilihat pada **Gambar 2.2**



Gambar 2.2 Pengaturan Bean Shell Form Binder

Berbagai macam fungsi pada Java dapat dilakukan pada metode ini, fungsi tersebut juga dapat dikombinasikan dengan *query SQL*. Dalam pemanfaatan metode Bean Shell Form memerlukan adanya tabel pada *database* terlebih dahulu. Penyimpanan data dapat dilakukan pada tabel yang mempunyai relasi dengan satu tabel atau lebih. Selain digunakan sebagai penyimpanan data, Bean Shell juga dapat digunakan sebagai berikut:

- Form Load Binder: untuk menampilkan data *form*
- Form Options Binder: untuk menampilkan *form* dengan *field select box*
- Form Validator: untuk validasi *form*
- Form Multi Row Load Binder: untuk menampilkan data *form* dalam bentuk *grid*
- Form Multi Row Store Binder: untuk menyimpan data *form* dalam bentuk *grid*
- Form Multi Row Validator: untuk validasi *form* dengan bentuk *grid*
- Form Permission: untuk membuat hak akses pada *form*
- Form Post Submission Processing Tool: untuk mengirim *email*

2. JDBC Binder

JDBC Binder adalah metode yang digunakan untuk menyimpan atau memuat data dari Joget maupun *database* khusus melalui *query SQL* (Justin & Andrew, 2019). Dalam penyimpanan data menggunakan *query SQL* seperti *select*, *insert*, *update*, dan *delete*. Selain itu, metode ini memerlukan adanya tabel *database* terlebih dahulu agar fungsi penyimpanan dapat digunakan. Hal ini karena JDBC Binder membutuhkan sumber data (*datasource*) untuk menjalankan fungsi *query SQL*. Pengaturan JDBC *form binder* dapat dilihat pada **Gambar 2.3**.

Configure JDBC Binder

Edit Form > Advanced > Load Binder (JDBC Binder) > **Configure JDBC Binder**

Datasource

Custom JDBC Driver *
Eg. com.mysql.jdbc.Driver (MySQL), oracle.jdbc.driver.OracleDriver (Oracle), com.microsoft.sqlserver.jdbc.SQLServerDriver (Microsoft SQL Server)

Custom JDBC URL *

Custom JDBC Username *

Custom JDBC Password

SQL SELECT Query *
Used to decide an insert or update operation. Use syntax like {field_id} in query to inject submitted form data.

SQL INSERT Query *
Use syntax like {field_id} in query to inject submitted form data.

SQL UPDATE Query *
Use syntax like {field_id} in query to inject submitted form data.

SQL DELETE Query *
Used to delete deleted form data in Grid element. Use syntax like {id} in query to inject form data primary key.

Gambar 2.3 Pengaturan JDBC Binder

Pada bagian *datasource* dapat diatur sesuai dengan *database* yang telah disiapkan. Setiap perintah SQL dilakukan pada masing-masing *text editor* yang telah disediakan sehingga pemanfaatan fungsi hanya terbatas pada *query* SQL. Contoh penggunaan *query* SQL pada JDBC Binder dapat dilihat pada **Tabel 2.1**.

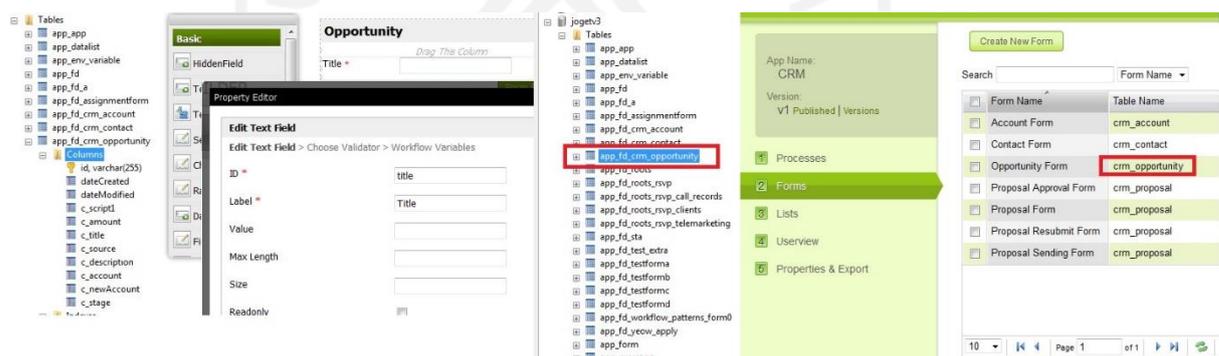
Tabel 2.1 Contoh Query SQL

Jenis Query	Penggunaan
<i>SELECT Query</i>	<pre>SELECT * FROM app_fd_table WHERE id = '#requestParam.id#'</pre> <p>Sintaks '#requestParam.id#' merupakan variabel hash untuk mendapatkan <i>primary key</i> secara langsung.</p>
<i>INSERT Query</i>	<pre>INSERT app_fd_table (id, c_name, c_emailaddress, c_contacttel) VALUES ({id}, {name}, {emailAddress}, {contactTel})</pre> <p>Sintaks seperti {field_id} digunakan di dalam <i>query</i> untuk memasukkan data <i>form</i> yang telah di kirim.</p>

Jenis Query	Penggunaan
<i>UPDATE Query</i>	<pre>UPDATE app_fd_table SET id = {id}, c_name = {name}, c_emailaddress = {emailAddress}, c_contacttel = {contactTel} WHERE id = {id}</pre> <p>Sintaks seperti {field_id} digunakan di dalam <i>query</i> untuk memasukkan data <i>form</i> yang telah di kirim.</p>
<i>DELETE Query</i>	<pre>DELETE FROM app_fd_table WHERE id = {id}</pre> <p>Digunakan untuk menghapus rekaman data dalam elemen <i>Datalist/Grid</i>. Gunakan sintaks seperti tanda kurung kurawal {id} di dalam <i>query</i> untuk memasukkan <i>primary key</i> dari data <i>form</i>.</p>

3. Workflow Form Binder

Workflow Form Binder adalah *form* untuk proses penyimpanan maupun pemuatan data yang ditangani oleh Form Builder milik Joget (Administrator & Justin, 2017). *Form* ini akan menyimpan data secara otomatis tanpa perlu ada tabel pada *database* terlebih dahulu. Hal ini karena tabel dan kolom di *database* akan dibuat oleh Workflow Form sesuai dengan pengaturan *form*. Pengaturan Workflow Form Binder dapat dilihat pada **Gambar 2.4**.



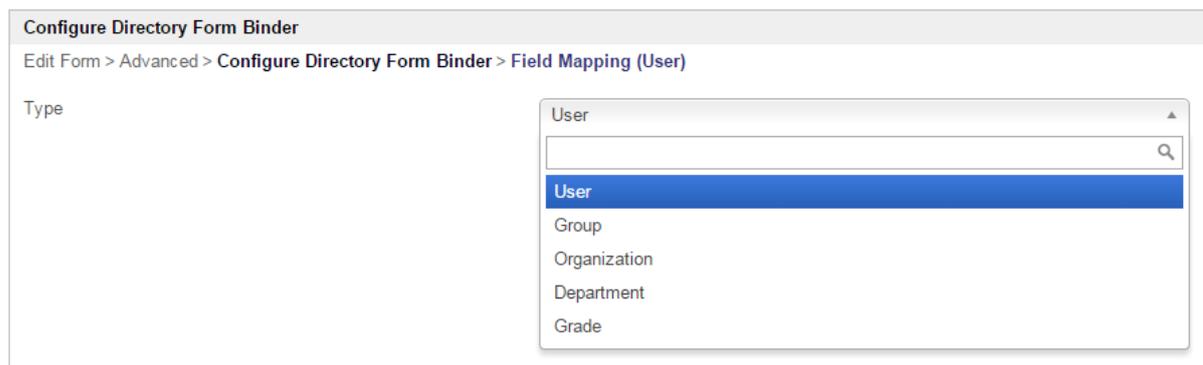
Gambar 2.4 Pengaturan Workflow Form Binder

Bagian nama *id* pada *field* akan menjadi nama kolom pada *database*. Nama tabel pada *database* akan diambil dari nama tabel *form*. Contoh perubahan nama di *form* Joget dengan nama di *database* yaitu, nama *field* *title* akan menjadi *c_title* sedangkan nama tabel *crm_proposal* menjadi *app_fd_crm_proposal*. Hal ini membuat *form* dapat digunakan tanpa ada tabel serta kolom

pada *database* terlebih dahulu. Selain itu penyimpanan data pada *form* yang menggunakan metode Workflow Form Binder tidak bisa ditambah dengan *query* pendukung lain sehingga metode penyimpanan ini mempunyai sifat yang statis.

4. Directory Form Binder

Directory Form Binder adalah metode penyimpanan pada Joget Workflow yang memungkinkan data pada *form* untuk berinteraksi dengan *directory manager* (Administrator & Justin, 2017). *Directory manager* ini menampung tabel *database* yang terdaftar pada Joget Workflow. *Field type* pada halaman pengaturan akan menampilkan tabel yang dapat digunakan. Di dalam tabel tersebut mempunyai *field* yang tersedia seperti id, nama dan lain-lain. Pengaturan Directory Form Binder dapat dilihat pada **Gambar 2.5**.



Gambar 2.5 Pengaturan Directory Form Binder

Dari keempat metode tersebut, di dalam pengembangan EHR saat magang hanya menggunakan tiga metode saja yaitu: Bean Shell Form Binder, JDBC Binder, dan Workflow Form Binder. Hal ini karena tiga metode tersebut dapat menangani skema penyimpanan data pada sistem EHR. Ketiga metode ini mempunyai cara penyimpanan data dengan *query*, kebutuhan tabel *database* dan karakteristik yang berbeda. **Tabel 2.2** memperlihatkan perbandingan antara ketiga metode pada Joget Workflow.

Tabel 2.2 Perbandingan Metode

No	Metode	Karakteristik			Keterangan
		Query	Tabel Database	Relasi Tabel	
1	Bean Shell Form Binder	Java	Perlu	Bisa	Pemanfaatan Fungsi Java
2	JDBC Binder	SQL	Perlu	Bisa	Pemanfaatan Fungsi SQL
3	Workflow Form Binder	<i>Non query</i>	Tidak perlu	Tidak	Penyimpanan statis

Selain penyimpanan data, fitur lain Joget Workflow adalah memfalsifikasi pembuatan sistem dengan proses bisnis yang ada. Hal ini membantu dalam menjalankan kegiatan operasional UPT Layanan Internasional Universitas Andalas. 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 sehingga hal tersebut dapat menimbulkan kesulitan dan/atau kelambatan dalam pemrosesan, penyimpanan, dan pengelolaan data. Penggunaan otomasi proses bisnis Joget Workflow dapat memudahkan dan meringankan proses bisnis serta meningkatkan efisiensi kinerja di UPT Layanan Internasional Universitas Andalas (David, 2017).

2.4 Query SQL

SQL atau *Structured Query Language* pada dasarnya adalah bahasa komputer standar yang ditetapkan untuk mengakses dan memanipulasi sistem pada *database*. Sebuah *database* berisi satu tabel atau lebih dan memiliki nama yang berbeda untuk masing-masing tabel. Masing-masing tabel memiliki satu kolom (*field*) atau lebih dan memiliki baris (*record*). *Query* digunakan untuk mengakses dan mengolah *database* (Utami & Sukrisno, 2008). SQL terdiri dari 5 bagian utama, yaitu:

1. *Retrieving data*: perintah untuk menampilkan data dari *database* (*SELECT*).
2. *Data Definition Language* (DDL): bahasa yang digunakan untuk membuat atau menghapus tabel atau *database* itu sendiri (*CREATE, DROP, ALTER*).
3. *Data Manipulation Language* (DML): merupakan bahasa untuk memanipulasi/mengubah isi tabel (*INSERT, DELETE, UPDATE*).

4. *Data Control Language (DCL)*: bahasa yang berhubungan dengan pengendalian akses ke *database (GRANT, REVOKE)*.
5. *Data Transaction Language (DTL)*: bahasa yang digunakan untuk mengelola transaksi *database (COMMIT, ROLLBACK)*.

Query SQL digunakan sebagai *query* pembantu dalam proses penyimpanan data pada Joget Workflow. *Query* ini digunakan dalam metode JDBC Binder dan Bean Shell Form Binder. Penggunaan SQL pada Bean Shell Form Binder dilakukan pada proses pemilihan data sebagai tujuan penyimpanan atau data pembanding. Beberapa *query SQL* yang umum digunakan pada pengembangan ini yaitu: *select, insert, update* dan *delete*.

2.5 Apache Tomcat

Apache Tomcat merupakan *web service open-source* berbasis Java yang digunakan untuk menjalankan *servlet* dan JavaServer Pages (JSP). Tomcat dibuat dibawah sub proyek Apache-Jakarta tetapi sekarang dipegang oleh beberapa proyek Apache secara terpisah dan didukung oleh sekelompok grup dari komunitas Java. Apache Tomcat sangat stabil dan mempunyai semua fitur wadah aplikasi web dibawah *open-source* Apache License. Beberapa fitur Apache Tomcat yang disediakan antara lain: keamanan, integrasi Apache *web server*, keseimbangan dalam memuat data, dan pemasangan Tomcat *server* di aplikasi Java. Sampai sekarang Apache mempunyai beberapa versi, antara lain: 5.5X, 6.0X, dan 7.0X (Vukotic & Goodwill, 2011). Apache Tomcat akan digunakan sebagai *web services* pada *platform* Joget Workflow. Penggunaan Apache akan membantu proses pemantauan dan pangujian pada sistem yang dibangun dengan tampilan *log*.

BAB III

PELAKSANAAN MAGANG

3.1 Electronic Human Resource

Electronic Human Resource atau biasa disebut EHR merupakan proyek pertama yang didapatkan saat magang di Javan. EHR adalah sebuah *website* yang digunakan untuk membantu pengolahan data pegawai dalam suatu perusahaan. Klien proyek ini dari sebuah perusahaan korporasi di Malaysia. Pengembangan sistem EHR dikerjakan bersama dengan tim yang terdiri dari analis sistem, *programmer* dan manajer proyek. Proses pengerjaan dimulai dengan pengumpulan data dengan berdiskusi dengan klien. Terpilihnya Joget Workflow sebagai *platform* pengembangan sistem ini menjadi salah satu hasil diskusi tersebut karena *platform* ini menyediakan pembuatan *form* berdasarkan *workflow* dengan metode penyimpanan data yang beranekaragam. Pengembangan EHR ini menggunakan *platform* Joget Workflow V.6, Java Runtime Environment (JRE) atau Java Development Kit (JDK) V.8, *web services* Apache Tomcat V.8.5.41, *database* MySQL V.5.5 dan dikembangkan pada sistem operasi Linux Ubuntu.

Setelah itu, analisis sistem akan menganalisis data yang telah diperoleh. Hasil analisis menjadi kumpulan *task* yang diberikan kepada *programmer*. Pembuatan *form* menjadi *task* utama dalam pengembangan sistem EHR. Berbagai macam *form* yang berbeda akan dibuat. *Form* tersebut akan mempunyai fungsi untuk menampilkan, mengubah dan menyimpan data. Selain pembuatan *form* terdapat beberapa *task* lain seperti: pembuatan *advance search*, hak akses pengguna, eReport, dan otomasi *update*.

Pada tahap selanjutnya yaitu implementasi yang dilakukan oleh *programmer*. Di sini, *programmer* perlu mengetahui detail *task* secara rinci. Hal ini untuk mempermudah *programmer* dalam memahami data yang akan diolah dan ditampilkan oleh sistem. Sebagai *programmer*, didapatkan beberapa *task* yang mencakup empat dari lima fitur yang ada. Selain itu, fitur pada sistem EHR dapat digunakan oleh empat aktor admin yaitu:

- Admin Joget, pengelola keseluruhan sistem EHR.
- Admin Unit Perkhidmatan, pengelola bagian mengenai layanan pengesahan.
- Admin Unit HRMIS, pengelola bagian mengenai manajemen SDM.
- Admin Unit Prestasi, pengelola bagian mengenai capaian pegawai.

Keempat aktor tersebut mempunyai hak akses yang berbeda sesuai dengan peranannya. Hak akses akan memberi wewenang pada pengguna untuk menggunakan fitur yang disediakan oleh sistem. **Tabel 3.1** memperlihatkan fitur yang ada serta aktor yang dapat mengaksesnya.

Tabel 3.1 Fitur EHR

No.	Fitur	Form	Admin Joget	Admin Unit Perkhidmatan	Admin Unit HRMIS	Admin Unit Prestasi
1.	<i>Dashboard</i>	Pengesahan Profil	✓	✓	-	-
2.	Pengurusan Staf Baru	Tukar Penempatan	✓	✓	-	-
		Pengesahan Lapori Diri	✓	✓	-	-
3.	Pengurusan Organisasi	Tambah Kelompok Kerja	✓	-	✓	-
		Ubah Kelompok Kerja	✓	-	✓	-
4.	Pengurusan Pegawai	Kenaikan Pangkat	✓	-	✓	✓
		Memangku	✓	-	✓	✓
		Penukaran Penempatan secara Administrasi	✓	-	✓	-
5.	Profil	Informasi Pribadi	✓	✓	✓	✓
		Informasi Alamat	✓	✓	✓	✓
		Informasi Pendidikan	✓	✓	✓	✓
		Informasi Pasangan	✓	✓	✓	✓
		Informasi Anak	✓	✓	✓	✓
		Informasi Darurat	✓	✓	✓	✓
		Informasi Keuangan	✓	✓	✓	✓

3.1.1 *Dashboard*

Dashboard EHR merupakan halaman awal yang akan tampil setelah pengguna masuk ke dalam sistem EHR. Halaman ini akan menampilkan tabel yang berisi susunan permohonan untuk melakukan ubah profil pegawai. Pengguna yang dapat mengakses halaman *dashboard* hanya admin. Halaman *dashboard* dapat dilihat pada **Gambar 3.1**.

The screenshot shows the 'Senarai Permohonan Untuk Tindakan' (List of Requests for Action) section in the E HUMAN RESOURCE system. The table lists 9 requests with columns for NO, NO PERMOHONAN, NAMA PEMOHON, PENEMPATAN, TARIKH PERMOHONAN, KATEGORI MAKLUMAT, and STATUS.

NO	NO PERMOHONAN	NAMA PEMOHON	PENEMPATAN	TARIKH PERMOHONAN	KATEGORI MAKLUMAT	STATUS
1	PROFIL181119-0000010	HARMAWIAH BINTI SAIDI	UNIT PEMBAYARAN DAN GAJI	18/11/2019	ANAK	HANTAR KE HR
2	PROFIL181119-0000013	HARMAWIAH BINTI SAIDI	UNIT PEMBAYARAN DAN GAJI	18/11/2019	PERIBADI	HANTAR KE HR
3	PROFIL201119-0000020	SALIZA BINTI SAMSUDIN	UNIT PERKHIDMATAN	20/11/2019	PERIBADI	HANTAR KE HR
4	PROFIL201119-0000022	SALIZA BINTI SAMSUDIN	UNIT PERKHIDMATAN	20/11/2019	PENDIDIKAN	HANTAR KE HR
5	PROFIL201119-0000024	SALIZA BINTI SAMSUDIN	UNIT PERKHIDMATAN	20/11/2019	ANAK	HANTAR KE HR
6	PROFIL201119-0000025	Admin		20/11/2019	ALAMAT	HANTAR KE HR
7	PROFIL211119-0000032	Admin		21/11/2019	PENDIDIKAN	HANTAR KE HR
8	PROFIL261119-0000036	NIK SHUKRI BIN NIK SOH	JABATAN KEWANGAN	26/11/2019	ALAMAT	HANTAR KE HR
9	PROFIL261119-0000043	SAIFUL NIZAM BIN MUKHTAR	UNIT PENGURUSAN PRESTASI	26/11/2019	ANAK	HANTAR KE HR

Gambar 3.1 Halaman *Dashboard*

3.1.2 Pengurusan Staf Baru

Fitur Pengurusan Staf Baru (atau Senarai Laporan Diri) menampilkan data pegawai baru yang berhasil lolos dalam tahap perekrutan perusahaan. Terdapat dua tindakan yang dapat dilakukan oleh admin untuk memasukkan data pegawai baru tersebut yaitu Tukar Penempatan dan Pengesahan Laporan Diri. Tampilan fitur ini dapat dilihat pada **Gambar 3.2**.

Senarai Laporan Diri

The screenshot shows the 'Senarai Laporan Diri' (Self-Report List) page. The table displays one record for a staff member named Arya Wira Pahlawan, with columns for TARIKH LAPOR DIRI, STAF PPJ (TETAP), NAMA, NO KAD PENGENALAN, KATEGORI, JAWATAN WARAN, JAWATAN HAKIKI, PENEMPATAN OPERASI, TINDAKAN, and TINDAKAN.

TARIKH LAPOR DIRI	STAF PPJ (TETAP)	NAMA	NO KAD PENGENALAN	KATEGORI	JAWATAN WARAN	JAWATAN HAKIKI	PENEMPATAN OPERASI	TINDAKAN	TINDAKAN
01/04/2020	Tidak	Arya Wira Pahlawan	801212177021	Kontrak	P P Kshtn Psktrm/P Kshtn Psktrm U32	Pen. Peg. Kesihatan Persektm U32	UNIT KAWALAN KACAUGANGGU DAN PERSEKITARAN	Tukar Penempatan	Pengesahan Laporan Diri

One item found.
[CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Gambar 3.2 Halaman Susunan Laporan Diri

3.1.3 Pengurusan Organisasi

Pengurusan Organisasi dapat diakses oleh admin HR untuk mengelola data struktur operasi yang ada pada perusahaan. Pada perusahaan ini terdapat lima struktur utama dengan berbagai kelompok kerja di dalamnya, yaitu:

- Jabatan: kelompok kerja teratas di dalam perusahaan
- Bahagian: kelompok kerja di bawah jabatan
- Seksyen: kelompok kerja di bawah bahagian
- Unit: kelompok kerja di bawah seksyen
- Subunit: kelompok kerja di bawah unit

Setiap struktur akan menampilkan data tabel kelompok kerja yang berisi nama kelompok dan nama ketua kelompok. Nama kelompok kerja berisi tautan untuk mengakses data kelompok di bawah struktur yang ada. Tampilan pengurusan organisasi dapat dilihat pada **Gambar 3.3**.

Home > / View Organisasi

Maklumat	
Organisasi	Carta Organisasi Perbadanan
Nama	PEJABAT PRESIDEN
Ketua	DATUK AMINUDDIN

Senarai Jabatan	
Tambah	
Nama	Ketua
1 PEJABAT PRESIDEN	DATUK AMINUDDIN ✎
2 JABATAN KEWANGAN	NIK SHUKRI ✎

Gambar 3.3 Halaman Pengurusan Organisasi

3.1.4 Pengurusan Pegawai

Pengurusan Pegawai menampilkan tabel kumpulan pegawai yang terdaftar di dalam perusahaan. Admin dapat memilih pegawai yang diinginkan untuk mengubah data kepegawaian pegawai tersebut. Tampilan pengurusan pegawai dapat dilihat pada **Gambar 3.4**.

Home > / Pengurusan Carta Operasi > / Pengurusan Pegawai

10 ▾ Arya Show

NAMA	EMAIL	NO STAF	JAWATAN WARAN	JAWATAN HAKIKI	GELARAN JAWATAN	PENEMPATAN
Arya Pahlawan	aryap@gmail.com	00000142	Pembantu Tadbir (P/O) N17/22	Pembantu Tadbir (P/O) N22	PEMBANTU TADBIR (PERKERANIAN/OPERASI) N22	UNIT PENTADBIRAN
Wira Arya	wiraarya@gmail.com	00000523	Pegawai Tadbir N41/44	Pegawai Tadbir N41	PEGAWAI TADBIR N41	UNIT KHIDMAT PELANGGAN

2 items found, displaying all items.
[CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Gambar 3.4 Halaman Pengurusan Pegawai

Data kepegawaian dapat diubah pada detail pengurusan pegawai. Halaman ini dapat diakses dengan menekan nama pegawai yang diinginkan. Detail pengurusan pegawai akan menampilkan beberapa tombol *form* sesuai dengan data kepegawaian. Berikut beberapa *form* yang terdapat pada detail pengurusan pegawai:

- Kenaikan Pangkat: halaman *form* yang digunakan untuk mengelola pangkat dan capaian pegawai.
- Memangku: halaman *form* yang digunakan untuk mengelola kelompok kerja yang ditangani pegawai.
- Penukaran Penempatan secara Administrasi: halaman *form* untuk melakukan penukaran penempatan antar pegawai sesuai dengan ketentuan.

3.1.5 Profil Pegawai

Profil pegawai (atau Maklumat Pegawai) akan menampilkan informasi dari pegawai sesuai dengan pengguna yang masuk. Terdapat tujuh informasi yang dimuat pada halaman ini, yaitu:

- Informasi pribadi, yaitu informasi mengenai nama dan status kepegawaian.
- Informasi alamat, yaitu informasi mengenai alamat tempat tinggal dari pegawai.
- Informasi pasangan, yaitu informasi mengenai pasangan dari pegawai.
- Informasi anak, yaitu informasi mengenai anak yang dimiliki pegawai.
- Informasi pendidikan, yaitu informasi riwayat pendidikan yang telah ditempuh pegawai.

- Informasi keuangan, yaitu informasi mengenai data rekening yang dimiliki.
- Informasi darurat, yaitu informasi mengenai orang yang dihubungi selain anggota keluarga utama apabila hal mendesak yang terjadi pada pegawai.

Pegawai dapat melakukan perubahan data sesuai yang diinginkan. Perubahan data yang dilakukan akan dikirim kepada admin terkait untuk melakukan persetujuan perubahan tersebut. Apabila tidak disetujui, pemberitahuan akan dikirim melalui *email* pegawai. Halaman ini dibuat menggunakan *form grid* Joget Workflow. Tampilan halaman profil pegawai dapat dilihat pada **Gambar 3.5**.

The screenshot shows the 'E HUMAN RESOURCE' system interface. The page title is 'Maklumat Pegawai' (Employee Information). It features a profile picture of a man in a batik shirt on the left. To the right, a list of employee details is displayed:

No Staf	16523142
Nama Staf	Arya Wira Pahlawan
No. KP	16523142
Gelaran Jawatan	MAHASISWA
Singkatan Jawatan	MHS
Gred	AWP142
Jabatan	JABATAN 1
Bahagian	BAHAGIAN 2
Seksyen	SEKSYEN 3
Unit	UNIT 4
Sub Unit	SUB UNIT 5

Below this is the 'Maklumat Peribadi' (Personal Information) section, which includes a 'Kemaskini' (Update) button. The details are as follows:

Gelaran	Awp	Status Berkahwin	Bujang
Alamat Email	16523142@students.uui.ac.id	Agama	Islam

Gambar 3.5 Halaman Profil

Dengan menggunakan Joget Workflow, aplikasi EHR ini juga dirancang menjadi 4 *workflow* yaitu:

- Proses Ubah Profil, proses ini meliputi perubahan data pegawai oleh pengguna dan perubahan tersebut akan memerlukan persetujuan dari pihak admin.
- Proses Kenaikan Pangkat, proses ini digunakan pada perubahan data kepangkatan pegawai.

- Proses Memangku, proses ini digunakan pada perubahan data kelompok kerja yang ditanggung pegawai.
- Proses Penukaran Penempatan, proses ini digunakan pada perubahan data penempatan pegawai oleh admin.

3.2 Penyimpanan Data pada Joget Workflow

Pada pengembangan aplikasi EHR, khususnya dalam pembuatan *form* terutama penyimpanan data menggunakan metode pada Joget Workflow, dilakukan beberapa tahapan. Tahapan tersebut sebagai berikut:

1. Analisis, yaitu tahap pertama untuk menentukan kebutuhan *form* yang diperlukan oleh sistem.
2. Penentuan Metode, yaitu tahap kedua untuk menentukan metode penyimpanan data pada *form* yang akan dikembangkan.
3. Persiapan Joget Workflow, yaitu tahap ketiga untuk melakukan pemasangan dan konfigurasi *platform* Joget Workflow agar dapat digunakan dalam proses pengembangan sistem.
4. Pembuatan *Form*, yaitu tahap untuk membuat *form* yang akan berjalan pada sistem.
5. Implementasi Penyimpanan Data, yaitu tahap keempat untuk mengimplementasikan metode penyimpanan data ke dalam *form* yang telah dibuat.
6. Pengujian, yaitu tahap terakhir untuk memastikan hasil dari implementasi telah sesuai dengan kebutuhan sistem tanpa terdapat *bug* maupun *error*.

3.2.1 Analisis Kebutuhan Sistem

Proses analisis dimulai dengan melakukan diskusi antara tim pengembang dan klien. Diskusi ini dilakukan untuk memperoleh informasi yang dibutuhkan dalam pengembangan sistem. Informasi yang didapat akan diolah oleh tim untuk menentukan kebutuhan sistem sesuai dengan keinginan klien. Proses analisis akan memperoleh hasil berupa fitur pada sistem dan kumpulan *task* yang perlu untuk dikerjakan. Sistem EHR mempunyai beberapa fitur *form* yang di antaranya terdapat pada halaman: Profil, Pengurusan Staf Baru, Pengurusan Organisasi, dan Pengurusan Pegawai.

- a. Fitur Profil, fitur ini digunakan untuk menampilkan data profil pegawai yang telah masuk. Pada halaman ini pegawai dapat melakukan perubahan data profilnya. Profil diri yang ingin diubah perlu disetujui dari pihak pengelola agar data profil dapat berubah.
- b. Fitur Pengurusan Staf Baru, fitur ini akan menampilkan data profil pegawai baru. Pengelola akan melakukan konfirmasi akhir kepada data tersebut agar data pegawai baru ini terdaftar pada susunan pegawai yang ada.
- c. Fitur Pengurusan Organisasi, fitur ini digunakan untuk mengolah data kelompok kerja yang ada. Setiap kelompok kerja akan mempunyai informasi terkait nama ketua dan nama anggota.
- d. Fitur Pengurusan Pegawai, fitur ini akan 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 Pengurusan Pegawai disimpan ke dalam tabel sementara terlebih dahulu. Penyimpanan lanjut ke data pegawai sebenarnya akan dilakukan secara otomatis menggunakan Cron Laravel.

Setiap *form* akan menampilkan atau menyimpan bentuk data yang berbeda-beda. Bentuk data dari *database* dapat berbentuk numerik, *date/time*, dan *string*. Selain bentuk, data yang diolah akan mempunyai format sesuai kebutuhan klien seperti format dd-mm-yyyy atau mm-dd-yyy untuk bentuk data tanggal. Sehingga analis perlu menyusun skema penyimpanan data pada *form* sesuai dengan hasil yang diinginkan. Skema pada *form* akan ditangani fungsi pemrograman agar penyimpanan dapat berjalan. Fungsi yang dipakai akan menyesuaikan dengan skema yang dibuat oleh analis berdasarkan dokumentasi yang telah disediakan oleh Joget Workflow. Selain itu, relasi data pada *form* juga akan diperhitungkan dalam menyimpan data karena terdapat metode yang tidak dapat menangani data yang mempunyai relasi dengan tabel lain. **Tabel 3.2** memperlihatkan pembagian empat fitur aplikasi (kecuali fitur Dashboard) beserta karakteristik *form*-nya secara terperinci.

Tabel 3.2 Analisis Kebutuhan Sistem

No.	Fitur	Nama <i>Form</i>	Skema <i>Form</i>	Fungsi	Relasi Tabel
1.	Profil	Informasi Pribadi	Terdapat kondisi untuk melakukan pengecekan nilai data <i>field</i> pada <i>form</i>	<i>while</i> dan <i>if</i>	Ada

No.	Fitur	Nama <i>Form</i>	Skema <i>Form</i>	Fungsi	Relasi Tabel
			dengan data pada <i>database</i> .		
		Informasi Pendidikan	Terdapat kondisi untuk melakukan pengecekan nilai data <i>field</i> pada <i>form</i> dengan data pada <i>database</i> .	<i>while</i> dan <i>if</i>	Ada
2.	Pengurusan Staf Baru	Tukar Penempatan	Terdapat kondisi untuk melakukan pengecekan nilai data <i>field</i> pada <i>form</i> dengan data pada <i>database</i> .	<i>while</i> dan <i>if</i>	Ada
		Pengesahan Lapori Diri	Terdapat kondisi untuk melakukan pengecekan nilai data <i>field</i> pada <i>form</i> dengan data pada <i>database</i> .	<i>while</i> dan <i>if</i>	Ada
3.	Pengurusan Organisasi	Tambah Jabatan	Terdapat kondisi untuk melakukan perubahan format nilai yang disimpan.	<i>case when</i>	Ada
		Ubah Jabatan	Terdapat kondisi untuk melakukan perubahan format nilai yang disimpan.	<i>case when</i>	Ada
4.	Pengurusan Pegawai	Kenaikan Pangkat	Penyimpanan data sementara ke dalam satu tabel.	Tidak ada	Tidak ada
		Memangku	Penyimpanan data sementara ke dalam satu tabel.	Tidak ada	Tidak ada
		Penukaran Penempatan secara Administrasi	Penyimpanan data sementara ke dalam satu tabel.	Tidak ada	Tidak ada

Form pada fitur Profil dan Pengurusan Staf Baru mempunyai skema penyimpanan yang sama yaitu terdapat kondisi untuk melakukan pengecekan nilai data *field* pada *form* dengan data pada *database*. Pengecekan nilai ini digunakan untuk memastikan bahwa nilai data pada *field* yang dipilih tidak sama dengan data yang sudah ada di dalam *database*. Kondisi pengecekan dapat menggunakan fungsi *while* dan *if* pada bahasa pemrograman Java. Selain skema dan fungsi, kebutuhan data pada kedua fitur tersebut juga menggunakan tabel *database* yang berelasi dengan tabel lain. Selanjutnya, fitur Pengurusan Organisasi mempunyai skema penyimpanan berupa

adanya kondisi untuk melakukan perubahan format nilai data yang akan disimpan. Kondisi ini dapat dilakukan menggunakan fungsi *case when* pada *query* SQL. Fitur Pengurusan Organisasi juga membutuhkan tabel yang berelasi dengan tabel lain. Terakhir, fitur Pengurusan Pegawai mempunyai skema penyimpanan berupa penyimpanan data sementara ke dalam satu tabel. Penyimpanan ini tidak memerlukan fungsi atau *query* tambahan karena dapat diolah secara otomatis. Fitur Pegawai hanya melakukan penyimpanan data pada satu tabel sehingga tabel yang diperlukan tidak membutuhkan relasi dengan tabel lain.

3.2.2 Penentuan Metode Penyimpanan Data

Setelah mengetahui karakteristik *form* pada fitur yang akan dibangun, maka tahap berikutnya adalah melakukan penentuan metode penyimpanan data yang akan digunakan di tiap *form*. Penentuan ini berdasarkan pada karakteristik *form* yang ditemui dan perlu mencocokkan karakteristik *form* dengan karakteristik metode penyimpanan data pada Joget Workflow. Dari seluruh fitur yang akan dibangun, terdapat tiga fungsi yang berbeda dengan jenis *query* yang berbeda juga dalam pembuatannya. **Tabel 3.3** memperlihatkan pencocokan karakteristik *form* dengan karakteristik metode penyimpanan pada Joget Workflow.

Tabel 3.3 Pencocokan Karakteristik *Form* dan Metode

No.	Fungsi	Relasi Tabel	Query	Metode Penyimpanan
1.	<i>while</i> dan <i>if</i>	Ada	Java	Bean Shell Form Binder
2.	<i>case when</i>	Ada	SQL	JDBC Binder
3.	Tidak ada	Tidak ada	<i>Non query</i>	Workflow Form Binder

Form yang membutuhkan fungsi *while* dan *if* pada Java serta membutuhkan tabel yang berelasi dapat menggunakan metode penyimpanan Bean Shell Form Binder. Bean Shell Form Binder dapat menangani penyimpanan data dengan implementasi Java sehingga fungsi *while* dan *if* dapat dibuat. Selanjutnya, *form* dengan fungsi *case when* pada SQL dapat dilakukan menggunakan metode JDBC Binder. Metode ini dapat menangani *query* SQL sehingga kebutuhan *form* dengan tabel yang berelasi juga bisa dibuat. Terakhir, *form* yang tidak membutuhkan fungsi dan hanya dilakukan dalam satu tabel tanpa relasi dapat menggunakan metode Workflow Form Binder. Workflow akan otomatis melakukan penyimpanan data secara

no code sehingga pembuatan *form* akan lebih cepat. Setelah dilakukan pencocokan pada **Tabel 3.3**, penentuan metode penyimpanan data pada *form* dapat dilihat pada **Tabel 3.4**.

Tabel 3.4 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
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 3.4 memperlihatkan empat *form* dengan metode Bean Shell Form Binder. Hal ini karena *form-form* tersebut memiliki skema penyimpanan data yang membutuhkan fungsi pada Java seperti fungsi percabangan *if* dan perulangan *while* sebagai kondisi pengecekan data. Selain itu, empat *form* itu juga mempunyai tabel yang berelasi dengan tabel lain. Selanjutnya metode JDBC Binder digunakan pada dua *form* dengan skema yang membutuhkan fungsi *case when* dan mempunyai tabel yang berelasi. Kebutuhan fungsi dan relasi tabel ini dapat dilakukan pada *query* SQL. Terakhir, tiga *form* akan menggunakan metode Workflow Form Binder. Metode ini digunakan pada *form-form* tersebut karena skema penyimpanan yang ada hanya melakukan penyimpanan data sementara pada satu tabel. Workflow akan menangani secara otomatis penyimpanan tersebut tanpa perlu *query* maupun tabelnya terlebih dahulu.

3.2.3 Persiapan Joget Workflow

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

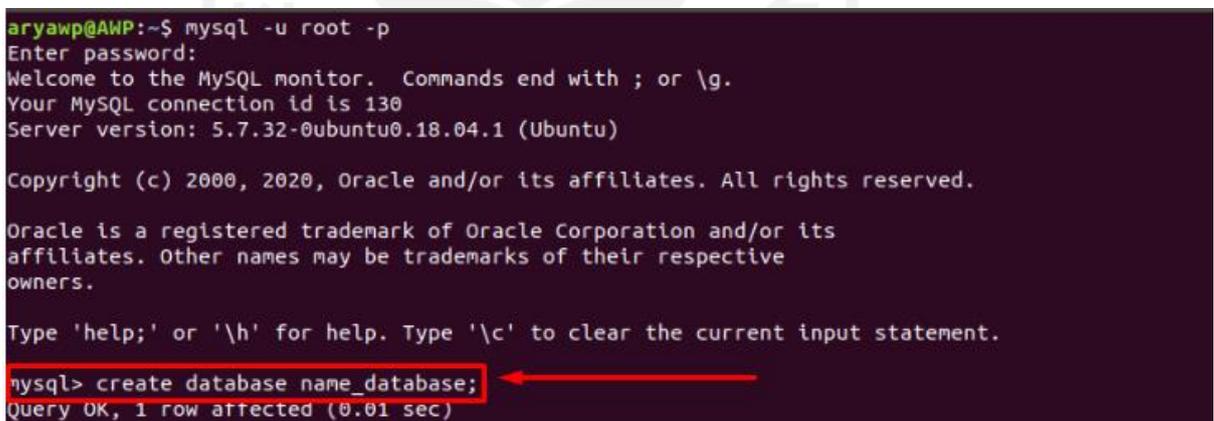
1. Pemasangan Joget Workflow

Spesifikasi dalam pemasangan Joget Workflow membutuhkan Java Runtime Environment (JRE) atau Java Development Kit (JDK) versi 8 dan MySQL versi 5.5

(Administrator & Andrew, 2019). Pemasangan dilakukan dengan melakukan pengunduhan berkas Joget Workflow dan menjalankan berkas tersebut. Joget Workflow dapat digunakan setelah menyalakan Apache Tomcat sebagai *web services*. *Web services* ini dapat berjalan dengan melakukan perintah `./tomcat8.sh start` di terminal. Saat pertama kali memasang Joget Workflow, tautan *localhost* akan menampilkan halaman konfigurasi *database* terlebih dahulu sebelum halaman awal *platform* dapat diakses.

2. Pembuatan *Database* MySQL

Sebelum melakukan pengembangan sistem pada *platform* Joget Workflow diperlukan *database* MySQL terlebih dahulu sebagai wadah untuk penampung dan pengolahan data. Perintah untuk membuat *database* baru dapat dilihat pada **Gambar 3.6**. Pembuatan *database* ini dilakukan di MySQL *Server*.



```

aryawp@AWP:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 130
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database name_database;
Query OK, 1 row affected (0.01 sec)

```

Gambar 3.6 Perintah *Create Database* MySQL

3. Konfigurasi *Database* MySQL

Konfigurasi *database* dilakukan untuk menyambungkan antara aplikasi Joget dengan *database* yang akan digunakan. Proses konfigurasi akan memakan waktu beberapa saat untuk memasukkan data *default* Joget ke dalam *database* agar Joget Workflow dapat berjalan. Contoh konfigurasi *database* MySQL dapat dilihat pada **Gambar 3.7**.

JOGET DX SETUP

DATABASE SETUP

No database configuration was detected, so please configure your database settings below. Please ensure that the database server is installed and running first. [More Information](#)

Database Type: MySQL

Database Host: localhost

Database Port: 3306

Database Name: ppi_joget

Database User: root

Database Password:

Include Sample Apps:

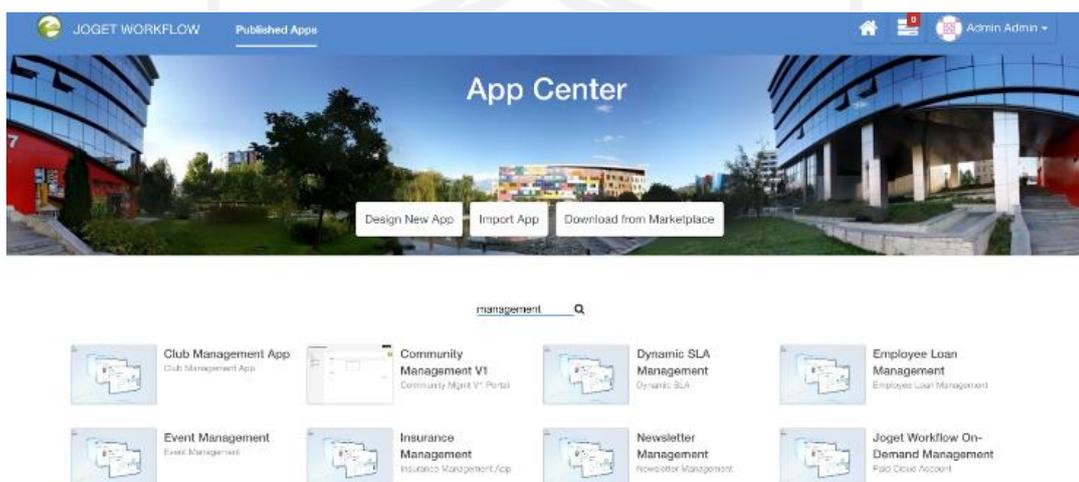
Include Sample Users:

[Save](#)

Gambar 3.7 Halaman Konfigurasi *Database*

4. Akses Joget Workflow

Setelah konfigurasi *database* telah selesai, Joget Workflow telah siap digunakan untuk pengembangan sistem. Halaman awal Joget Workflow dapat diakses melalui tautan *localhost*. **Gambar 3.8** memperlihatkan tampilan halaman awal dari *platform* Joget Workflow.



Gambar 3.8 Halaman Awal *Platform* Joget Workflow

3.2.4 Pembuatan Form

Setelah Joget Workflow dapat berjalan, tahap selanjutnya adalah membuat *form*. *Form* pada Joget Workflow dapat dibangun melalui di dalam mode pengembang *platform* ini. Pembuatan *form* di sini dilakukan dengan cara sebagai berikut:

- Pertama, tekan tombol konfigurasi Joget Workflow untuk masuk ke dalam mode pengembangan. Tombol ini terletak di bagian bawah kanan antarmuka Joget yang dapat dilihat pada **Gambar 3.9**.

Senarai Tindakan Pengesahan

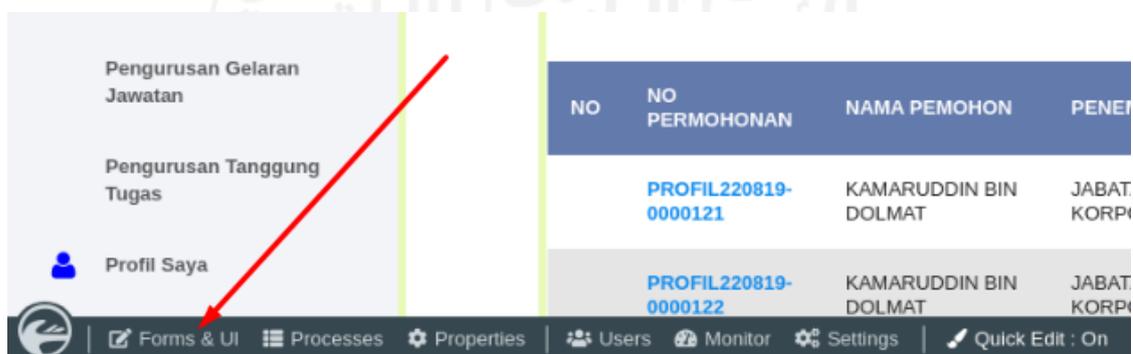
10 ▼ No Permohonan Show

NO	NO PERMOHONAN	NAMA PEMOHON	PENEMPATAN	TARIKH PERMOHONAN	KATEGORI MAKLUMAT	STATUS
	PROFIL220819-0000121	KAMARUDDIN BIN DOLMAT	JABATAN PERKHIDMATAN KORPORAT	26/10/2019	PASANGAN	HANTAR KE HR
	PROFIL220819-0000122	KAMARUDDIN BIN DOLMAT	JABATAN PERKHIDMATAN KORPORAT	26/10/2019	PERIBADI	HANTAR KE HR
	PROFIL220819-0000123	HARMAWIAH BINTI SAIDI	UNIT PEMBAYARAN DAN GAJI	26/10/2019	PERIBADI	HANTAR KE HR



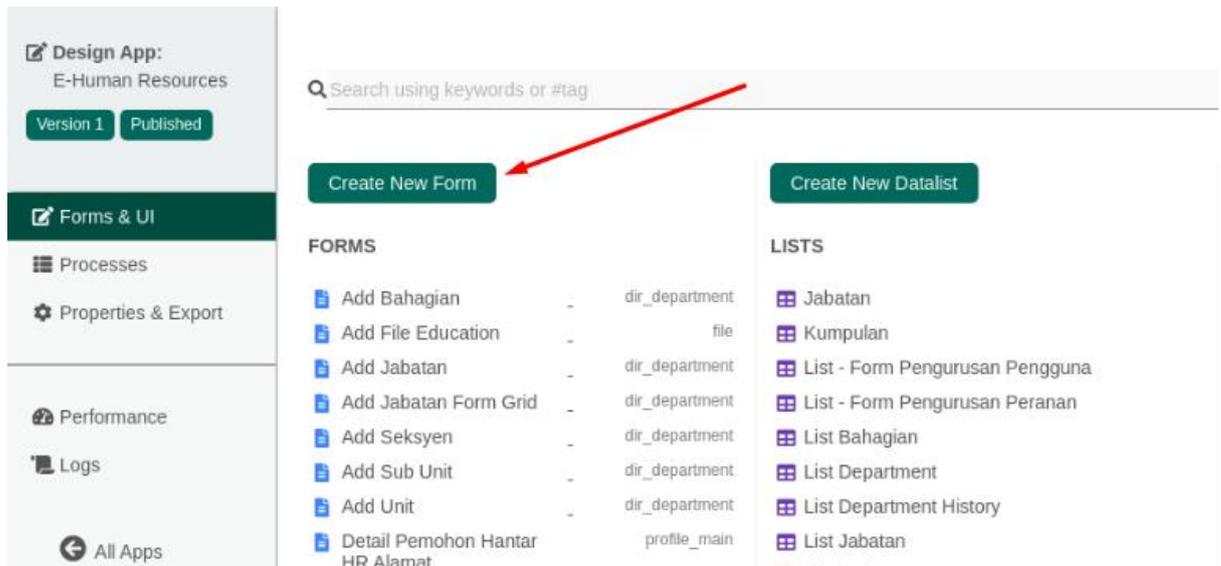
Gambar 3.9 Tombol Mode Pengembang

- Kedua, saat masuk ke dalam mode pengembang pilih tombol *Form & UI*. **Gambar 3.10** memperlihatkan letak tombol *Form & UI*.



Gambar 3.10 Tombol *Form & UI*

- Ketiga, tekan *Create New Form* untuk membuat *form* baru. Letak tombol ini dapat dilihat pada **Gambar 3.11**.



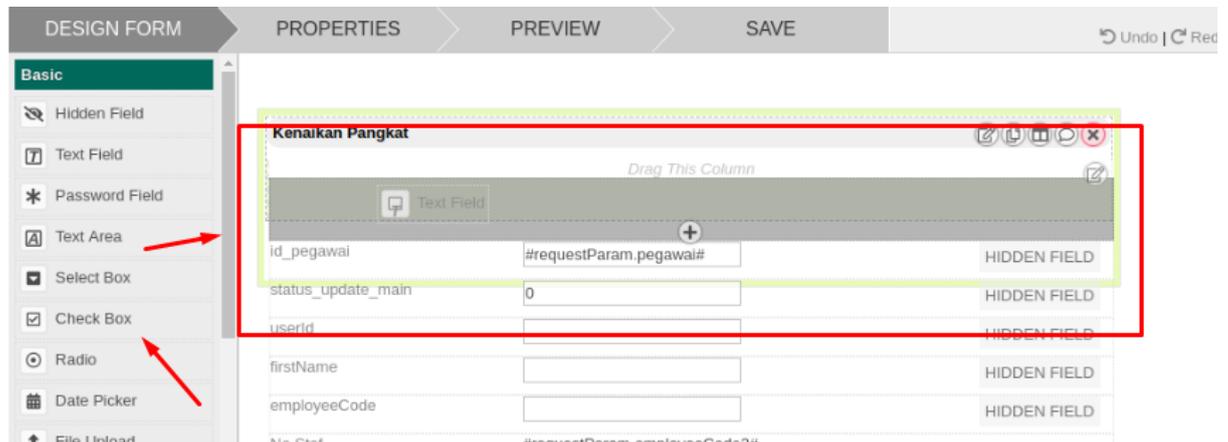
Gambar 3.11 Tombol *Create New Form*

- Keempat, pengisian detail *form* sesuai kebutuhan. Detail *form* yang perlu diisi berupa *id* dan nama *form*, serta nama tabel data. **Gambar 3.12** memperlihatkan detail *form* pada *Create New Form*.

Gambar 3.12 Detail *Create New Form*

- Kelima, merancang *form* yang akan digunakan. Rancangan *form* dapat ditambahkan *field form* dengan cara *drag & drop*. Jenis *field form* dapat dipilih pada bagian kiri

antarmuka, kemudian letakkan *field* yang dipilih ke dalam *form section*. **Gambar 3.13** memperlihatkan desain *form* pada Joget Workflow.



Gambar 3.13 Desain *Form* Joget

- Terakhir, *form* yang sudah dibuat dapat ditinjau hasilnya dengan menekan tombol *preview*. Pratinjau *form* dapat dilihat pada **Gambar 3.14**.

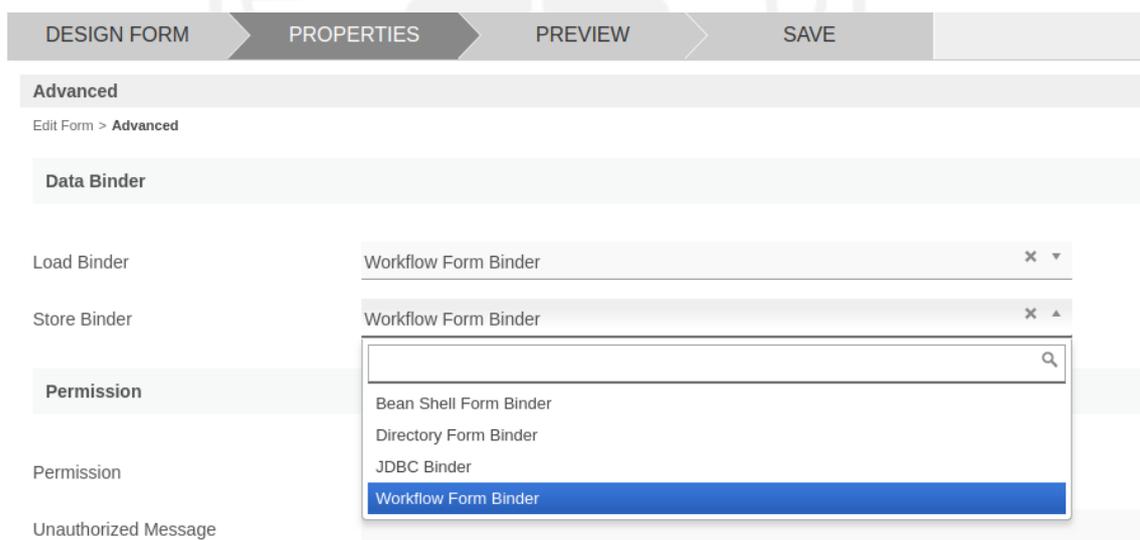
Gambar 3.14 Pratinjau *Form* Joget

Pembuatan *form* pada Joget Workflow lebih cepat daripada menggunakan fitur pada *framework* seperti Laravel dan CodeIgniter karena di dalam halaman mode pengembang tersedia pembuatan *form* dengan cara *drag & drop* sehingga minim menggunakan sintaks pemrograman.

Tampilan pada *form* yang dibuat akan langsung menampilkan desain yang ada tanpa perlu membuat *layout* desain terlebih dahulu. Begitu pula, fungsi *button* dan *route* yang ada pada *form* juga dapat langsung berjalan.

3.2.5 Implementasi Penyimpanan Data

Tahap selanjutnya adalah mengimplementasikan penyimpanan data ke dalam tiap *form* yang dibuat. Setiap *form* yang dibuat di Joget Workflow mempunyai pengaturan. Salah satu di antaranya digunakan untuk memilih metode penyimpanan data pada *form*. Pengaturan penyimpanan data untuk memilih metode yang akan digunakan *form* pada Joget Workflow terdapat di *Properties Form Builder*. Pengaturan tersebut dapat dilihat pada **Gambar 3.15**.



Gambar 3.15 Pengaturan *Form Builder*

Setelah itu, pada beberapa metode diperlukan *query* pendukung agar penyimpanan data dapat berjalan sesuai keinginan. Bean Shell Form Binder akan menggunakan bahasa pemrograman Java dan JDBC Binder menggunakan *query* SQL. Sementara, Workflow Form Binder tidak membutuhkan *query* tambahan maupun tabel *database* terlebih dahulu karena akan diproses otomatis oleh sistem.

3.2.5.1 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 menggunakan *query select*. Pemilihan disesuaikan dengan *field* pada *form* yang ada. **Gambar 3.16** memperlihatkan cara untuk memilih data dari *field* pada *form*.

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

Gambar 3.16 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 3.17**.

```
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";

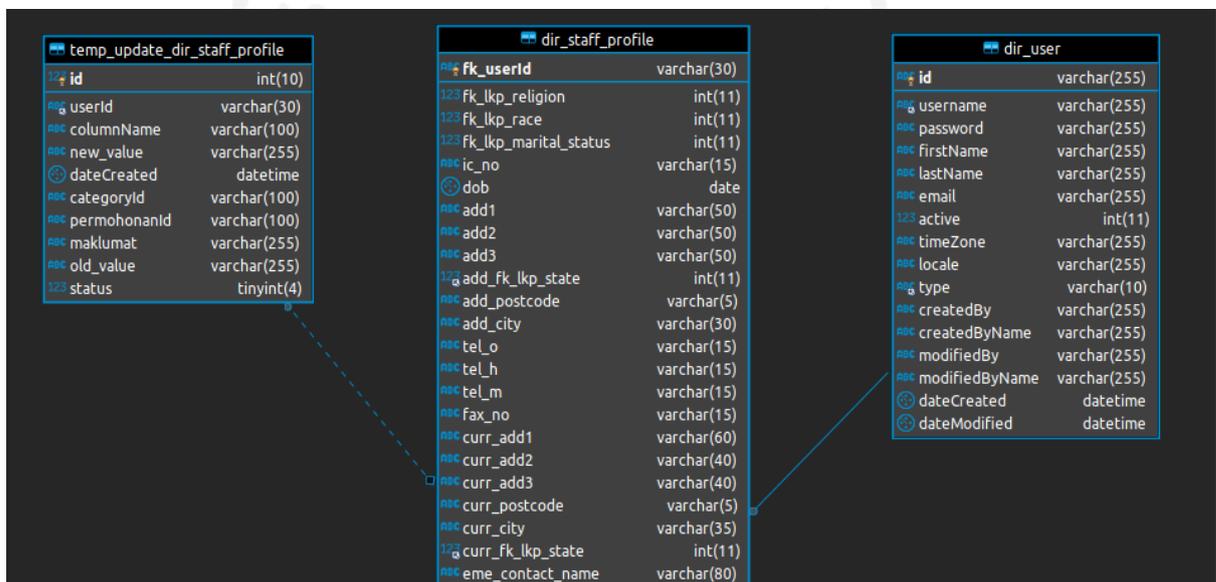
PreparedStatement istmt = con.prepareStatement(insertQuery);

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 3.17 *Query Insert* dan *Select* pada Informasi Pribadi

Data 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 ingin mengubah data. Hal ini untuk mengecek antara nilai data baru dan data lama agar menghindari kesamaan data. **Gambar 3.18** memperlihatkan relasi tabel di *database* yang digunakan pada *form* Profil Pribadi.



Gambar 3.18 Relasi Tabel pada *Form* Profil Pribadi

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 3.19** 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 3.19 *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 3.17**. 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 data terbaru. Kemudian variabel *istmt* dieksekusi fungsi *executeUpdate* untuk melakukan proses perubahan data ke dalam *database*.

Penyimpanan data Bean Shell Form Binder ini dilakukan pada *form* Informasi Pribadi. Tampilan dari *form* tersebut dapat dilihat pada **Gambar 3.20**.

KEMASKINI PROFILE PERIBADI

Maklumat Peribadi	
Gelaran	admin
Alamat Email	16523142@students.uui.ac.id
Tarikh Lahir	30 Januari 1997
Negeri Lahir	Indonesia
Warganegara	Indonesia
Jantina	Laki-laki
Status	Aktif
Agama	Islam
Bangsa	Indonesia
No. Telefon Bimbit	081763728127
No Sambungan	57890

Gambar 3.20 Halaman *Form* Informasi Pribadi

3.2.5.2 JDBC Binder

Metode yang kedua adalah JDBC Binder. Langkah yang untuk membuat *query* penyimpanan di sini 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 berjumlah sama. *Query* tersebut dapat dilihat pada **Gambar 3.21**.

```

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 3.21 Query Penyimpanan Data Form Tambah Jabatan

Query insert akan memilih tujuan kolom tabel yang akan disimpan ke tabel `dir_department`. *Query select* digunakan untuk menentukan data yang akan dimasukkan pada *query insert*. Pada pemilihan *field query select*, terdapat nama *field* yang menggunakan simbol bracket (`{}`). Nilai data *field* tersebut akan mengambil dari nilai field secara *realtime*. Selanjutnya, terdapat fungsi *case when* untuk menyimpan data sesuai dengan format yang diinginkan. Terakhir, pengambilan asal data *select* juga memanfaatkan *sub query* untuk memperoleh tabel yang tepat.

Metode JDBC Binder ini digunakan pada *form* Tambah Jabatan. **Gambar 3.22** memperlihatkan tampilan *form* Tambah Jabatan.

Home > / Add Jabatan

Tambah Jabatan

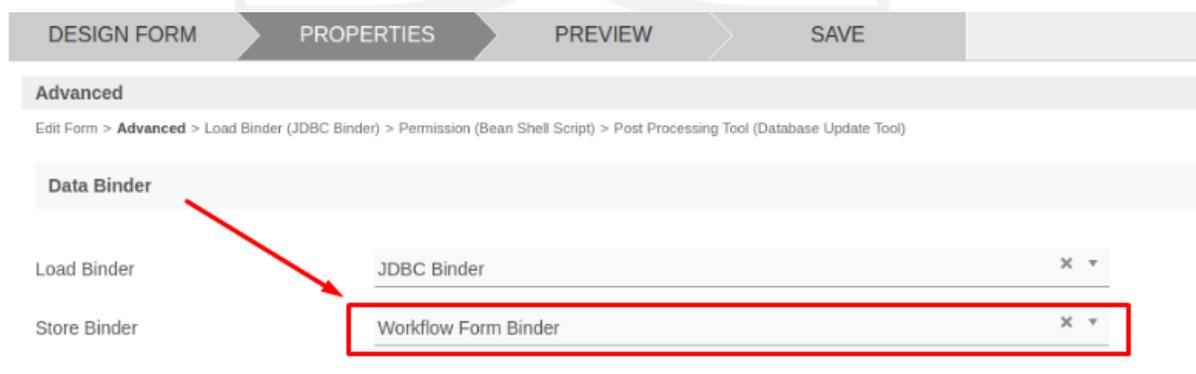
ID	J012
Nama	• Keuangan
Keterangan	<hr/>
Ketua	<hr/>
Susunan	<hr/>

Kembali
Simpan

Gambar 3.22 Halaman *Form* Tambah Jabatan

3.2.5.3 Workflow Form Binder

Metode Workflow Form Binder termasuk metode dengan cara penggunaan yang paling mudah di antara metode lain di *platform* Joget Workflow. *Form* yang sudah dibuat hanya perlu diubah cara penyimpanan datanya menjadi Workflow Form Binder dan tidak membutuhkan adanya tabel penyimpanan data terlebih dahulu. **Gambar 3.23** akan memperlihatkan pengaturan form dengan metode Workflow Form Binder.



Gambar 3.23 Penyimpanan Data dengan Workflow Form Binder

Proses penyimpanan ini akan ditangani oleh Joget sehingga pengembang tidak perlu menambahkan *query* pendukung. Data yang disimpan akan mempunyai nama tabel dan kolom yang sesuai dengan detail *form*. Akan tetapi penamaan pada *database* akan memiliki sedikit

perbedaan dengan nama di *form*. **Tabel 3.5** akan memperlihatkan contoh perbandingan antara nama di *form* dengan nama di *database*.

Tabel 3.5 Perbandingan Nama Tabel dan Kolom

Jenis	Nama di <i>Form</i>	Nama di <i>Database</i>
Tabel	staff_move_dept	app_fd_staff_move_dept
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	

Metode Workflow Form Binder ini digunakan pada *form* Penukaran secara Administrasi. **Gambar 3.24** memperlihatkan tampilan *form* Penukaran secara Administrasi.

Kemaskini Penukaran Secara Pentadbiran

Pegawai Arya Wira Pahlawan

Jawatan Hakiki Mahasiswa

Penempatan * Jurusan Informatika

Ketua Ya Tidak

Kumpulan Staf * Programmer

Tarikh Berkuatkuasa * 01-11-2020

Makluman: Setiap proses penukaran mengambil masa 5 minit

[Kembali](#) [Simpan](#)

Gambar 3.24 Halaman *Form* Penukaran secara Administrasi

3.2.6 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 melalui *debugging* data. *Debugging* adalah sebuah metode yang dilakukan oleh para pemrogram dan pengembang perangkat lunak untuk menganalisis alur kerja program, mencari dan mengurangi *bug*, atau kerusakan di dalam sebuah program komputer atau perangkat keras sehingga perangkat tersebut bekerja sesuai dengan harapan. *Debugging* pada Joget Workflow menggunakan Log Tomcat 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 3.25**. Pengujian akan dilakukan secara manual oleh pengembang dengan menambahkan atau mengubah data *field* pada setiap *form* yang telah dibuat. Setelah itu hasil pengujian dapat dilihat pada **Tabel 3.6**.

```
File Edit View Search Terminal Help
aryawp@AMP:~/Downloads/joget/joget-enterprise-linux-7.0.10$ tail -f apache-tomcat-8.5.41/logs/catalina.out
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
at java.lang.Thread.run(Thread.java:748)
Caused by: java.sql.SQLException: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right s
yntax to use near '(order by a.dateCreated asc) as no, a.id, a.apply_no, b.firstName, d.name, a.use' at line 1
at org.mariadb.jdbc.internal.protocol.AbstractQueryProtocol.readErrorPacket(AbstractQueryProtocol.java:1599)
at org.mariadb.jdbc.internal.protocol.AbstractQueryProtocol.readPacket(AbstractQueryProtocol.java:1461)
at org.mariadb.jdbc.internal.protocol.AbstractQueryProtocol.getResult(AbstractQueryProtocol.java:1424)
at org.mariadb.jdbc.internal.protocol.AbstractQueryProtocol.executeQuery(AbstractQueryProtocol.java:240)
at org.mariadb.jdbc.ClientSidePreparedStatement.executeInternalAroundBody6(ClientSidePreparedStatement.java:216)
... 188 more
ERROR 16 Nov 2020 15:13:05 org.joget.plugin.enterprise.JdbcDataListBinder - java.sql.SQLException: (conn=308) Table 'ppj_joget.newstaff_ma
in' doesn't exist
java.sql.SQLException: (conn=308) Table 'ppj_joget.newstaff_main' doesn't exist
at org.mariadb.jdbc.internal.util.exceptions.ExceptionMapper.get(ExceptionMapper.java:243)
at org.mariadb.jdbc.internal.util.exceptions.ExceptionMapper.getException(ExceptionMapper.java:164)
at org.mariadb.jdbc.MariaDbStatement.executeExceptionEpilogue_aroundBody2(MariaDbStatement.java:244)
at org.mariadb.jdbc.MariaDbStatement.executeExceptionEpilogue_aroundBody3$Advice(MariaDbStatement.java:127)
at org.mariadb.jdbc.MariaDbStatement.executeExceptionEpilogue(MariaDbStatement.java:1)
at org.mariadb.jdbc.ClientSidePreparedStatement.executeInternalAroundBody6(ClientSidePreparedStatement.java:225)
at org.mariadb.jdbc.ClientSidePreparedStatement.executeInternalAroundBody7$Advice(ClientSidePreparedStatement.java:127)
```

Gambar 3.25 Halaman *Debugging Log* Tomcat

Tabel 3.6 Hasil Pengujian

No.	Nama Form	Skenario Pengujian	Berhasil/Gagal
1.	Informasi Pribadi	Mengubah data <i>field</i> nomor telepon bimbit dan nomor sambungan	Berhasil
2.	Informasi Pendidikan	Menambah baris jenjang pendidikan sarjana	Berhasil

No.	Nama <i>Form</i>	Skenario Pengujian	Berhasil/Gagal
3.	Tukar Penempatan	Mengubah data <i>field</i> penempatan operasi	Berhasil
4.	Pengesahan Lapor Diri	Mengubah data <i>field</i> kedatangan lapor diri dan memilih <i>field</i> id <i>username</i>	Berhasil
5.	Tambah Jabatan	Menambah jenis data jabatan baru	Berhasil
6.	Ubah Jabatan	Mengubah data jabatan lama pada <i>field</i> nama, ketua dan susunan	Berhasil
7.	Kenaikan Pangkat	Mengubah data <i>field</i> jawatan hakiki dan gelaran jawatan	Berhasil
8.	Memangku	Mengubah data <i>field</i> penempatan dan jabatan	Berhasil
9.	Penukaran Penempatan secara Administrasi	Mengubah data <i>field</i> penempatan dan kumpulan staf	Berhasil

Pengujian dilakukan pada kesembilan *form* yang ada pada sistem EHR. Pengujian ini menggunakan skenario yang sudah direncanakan. Pengujian dilakukan setiap kali secara mandiri oleh pengembang sebelum pengembang melaporkan *task* pembuatan *form* tersebut. Selain pengujian mandiri, analis dan *tester* akan melakukan pengujian saat *task* berstatus *ready to test*. Pada masa UAT (*User Acceptance Test*), pengujian akan dilakukan bersama antara tim pengembang dan klien. Setelah dilakukan beberapa kali tahap pengujian, disimpulkan bahwa penyimpanan data menggunakan Joget Workflow dapat berfungsi dengan baik pada pengembangan sistem EHR.

BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Teknis

Dari sisi teknis atau pengembangan sistem terutama pemanfaatan Joget Workflow saat magang, telah dihasilkan pembahasan sebagai berikut:

4.1.1 Penyimpanan Data pada Joget Workflow

Implementasi metode penyimpanan data merupakan salah satu hal yang paling digeluti selama menjalani proses magang. Implementasi ini memanfaatkan tiga metode yang digunakan dalam pembuatan *form* pada sistem Electronic Human Resource (EHR). Pengembangan menggunakan *platform* Joget Workflow yang minim menggunakan sintaks pemrograman tetapi *platform* ini juga menyediakan berbagai bahasa pemrograman apabila dibutuhkan. Bahasa pemrograman yang dapat digunakan, yaitu Java dan SQL. Penyimpanan data pada Joget Workflow dapat menangani proses penyimpanan data pada *form* EHR. Metode-metode tersebut berhasil diimplementasikan pada setiap *form* EHR. Selain itu, penggunaan metode penyimpanan data pada Joget Workflow juga mempermudah proses penyimpanan data pada *form* EHR karena metode yang digunakan mempunyai karakteristik yang sesuai dengan skema penyimpanan pada *form*.

Platform Joget Workflow belum digunakan dalam pengembangan sistem di kalangan mahasiswa Informatika UII. Selama pembelajaran kuliah, mahasiswa mendapatkan materi mengenai penyimpanan data menggunakan sintaks PHP, Java dan *query insert* SQL. Pembelajaran pembuatan sintaks ini menggunakan bahasa pemrograman *native* tanpa *framework* pendukung. Materi dasar mengenai penyimpanan data menggunakan sintaks PHP didapat pada mata kuliah Pengembangan Aplikasi Berbasis Web (PABW) pada Kurikulum 2016 selama empat pertemuan. Pembahasan mengenai sintaks PHP yang berhubungan dengan *database* ini dijelaskan cara menyambungkan *form* sintaks PHP dengan *database* yang akan digunakan. Contoh sintaks untuk menyimpan data menggunakan PHP dipaparkan pada **Gambar 4.1**. Sementara materi pembuatan *query* SQL didapat pada mata kuliah Pengembangan Sistem Informasi (PSI) pada Kurikulum 2016 selama enam pertemuan. Pembelajaran ini dimulai dari pengenalan dasar SQL, pembuatan *query* dasar SQL hingga implementasi pada contoh kasus. *Query* dasar yang diperoleh antara lain: *select*, *insert*, *update*, *delete* dan klausa *join*. **Gambar 4.2** memperlihatkan contoh implementasi yang diterapkan pada studi kasus yang ada.

```

updatedepartemen.php x
<?php
//mengakses file koneksi.php
include "conf/koneksi.php";

if (isset($_POST['submit'])) {
    $kode = $_POST['id'];
    $nama = $_POST['nama'];
    $manajer = $_POST['manajer'];
    $tanggal = $_POST['tanggal'];

    $sql = "Update departemen set nama_departemen = '$nama', id_manajer = $manajer,
    tanggal_mulai_manajer='$tanggal' where kode_departemen='$kode'";
    $hasil = $koneksi->query($sql);
    if ($hasil === TRUE) {
        header('Location: tampildatadepartemen.php');
    } else {
        echo "Update gagal";
    }
} else {
    print "gagal";
}
$koneksi->close(); // menutup koneksi
?>

```

Gambar 4.1 Sintaks *Update* PHP Mata Kuliah PABW

```

//query SELECT
SELECT * FROM jurusan;

//query INSERT
INSERT INTO jurusan VALUES (1, 'Teknik Informatika');

//query UPDATE
UPDATE jurusan
SET nama = 'Teknik Mesin'
WHERE nama = 'Teknik Kimia';

//query DELETE
DELETE FROM jurusan
WHERE nama = 'Teknik Informatika';

//query SELECT menggunakan JOIN
SELECT department_id, department_name, location_id, city
FROM departments
NATURAL JOIN locations; |

```

Gambar 4.2 *Query* Mata Kuliah PSI

Kedua materi di atas masih diterapkan pada model kasus yang sederhana dan tanpa menggunakan *framework*. Pada proses magang ditemui bahwa penerapan materi tersebut lebih kompleks, salah satunya adalah pemanfaatan *framework* dalam pengembangan sistem informasi

seperti Laravel. Selain *framework* Laravel, terdapat *tool* lain yang dapat dimanfaatkan dalam pengembangan sistem yaitu Joget Workflow. Implementasi *query* juga membutuhkan klausa lanjutan seperti *sub query*, *join* lebih dari satu tabel, dan *case when*.

Selama pengembangan sistem EHR menggunakan Joget Workflow, tim pengembang juga menemui beberapa kendala dalam penerapan sebuah fitur. Salah satu di antaranya adalah pembuatan notifikasi *pop up* setelah melakukan pengisian *form*. Hal ini karena pada Joget Workflow sulit untuk melakukan desain secara manual pada bagian antarmuka tertentu sehingga implementasi notifikasi *pop up* menggunakan *javascript* tidak terealisasi. Selain itu, pemanfaatan *query SQL* pada *form* belum dibuat di dalam *stored procedure*. Pemanfaatan *query SQL* di dalam *stored procedure* akan meringkas penulisan *query* pada *platform* karena *query* akan disimpan di dalam *database*. *Query* tersebut akan dijalankan dari *database* sehingga pemanfaatan ini akan mengurangi *traffic* antara sistem dan server *database*. Kendala terakhir, ditemui pada tahap pengujian *form* yang dilakukan secara manual dengan memantau *Log Tomcat*. Metode pengujian tersebut akan memakan banyak waktu karena *form* yang diuji berjumlah banyak dengan skenario pengujian yang bervariasi. Maka dari itu, pengujian perlu diubah dari manual menjadi secara otomatis menggunakan bantuan *tools automated testing*. Pemasangan *automated testing* pada sistem akan meringkas waktu pengujian yang ada.

4.1.2 Perbandingan Solusi dengan Alternatif Lain

Fungsionalitas *form* terutama pada proses penyimpanan dapat menggunakan metode yang disediakan pada Joget Workflow. Selain metode pada Joget Workflow terdapat alternatif lain dengan memanfaatkan fungsionalitas *form* pada *framework* CodeIgniter dan Laravel. Kedua *framework* ini merupakan *framework* yang populer digunakan dalam pengembangan web. Setiap *framework/platform* mempunyai karakteristiknya masing-masing dalam pengembangan aplikasi web. **Tabel 4.1** akan memperlihatkan perbandingan antara *framework/platform* tersebut.

Tabel 4.1 Perbandingan antar *Framework/Platform*

No.	<i>Framework/Platform</i>		
	Laravel	CodeIgniter	Joget Workflow
1.	MVC (<i>Model, View, Controller</i>) dengan artisan	MVC (<i>Model, View, Controller</i>)	Membangun aplikasi secara <i>build full-fledged apps</i>

No.	Framework/Platform		
	Laravel	CodeIgniter	Joget Workflow
2.	Menyediakan berbagai modul bawaan	Tidak menyediakan modul	<i>Drag & drop forms, lists, UI</i>
3.	Mendukung HTTPS	Memerlukan <i>URL Helper</i>	Otomasi proses dengan <i>Workflow</i>
4.	Menyediakan fitur autentikasi kelas	<i>Explicit/Implicit Routing</i>	Perluasan dengan <i>plugin</i> arsitektur
5.	Mempunyai <i>unit testing</i>	Memiliki keamanan yang handal seperti <i>xss filtering</i> dan <i>session encryption</i>	Integrasi dengan sistem eksternal
6.	Eloquent Object Relational Mapping (ORM)	<i>Open-Source</i>	<i>Enterprise Form Features</i>

Platform Joget Workflow mempunyai kelebihan dalam membuat sistem yang membutuhkan permodelan proses bisnis. Sistem dapat dibangun secara menyeluruh dan terintegrasi. Pembuatan *form* atau *user interface* dilakukan dengan fitur *drag & drop* sehingga proses pembuatan akan lebih cepat. Pada Joget Workflow menyediakan fitur *workflow* yaitu fitur untuk melakukan otomasi proses bisnis. Workflow ini dapat diterapkan pada *form* sesuai proses bisnis yang berjalan. Joget Workflow juga menyediakan berbagai macam metode penyimpanan yang dapat digunakan pada *form*. Berbeda dengan *framework* CodeIgniter maupun Laravel, metode-metode pada Joget Workflow mempunyai karakteristik yang berbeda sehingga penggunaan metode dapat disesuaikan dengan karakteristik *form* terutama skema penyimpanan yang ditemui.

4.2 Non Teknis

Selama pelaksanaan magang telah diperoleh pengalaman terkait hal non teknis, di antaranya yaitu:

4.2.1 Manfaat Magang

Selama magang kurang lebih selama enam bulan di Javan didapat banyak manfaat. Manfaat yang paling dirasakan adalah mendapatkan kesempatan untuk bergabung ke dalam sebuah tim pengembang yang bergerak langsung di dunia kerja. Kompetisi di dunia kerja terutama pada bidang IT sangat dirasakan. Tiap individu harus mempunyai kemampuan dan ketahanan untuk dapat bertahan dalam kompetisi ini. Untuk memenuhi hal tersebut, perlu tergerak untuk aktif selama proses magang ini.

Saat magang, pembelajaran dalam penggunaan *framework* Laravel dan Git akan melalui contoh kasus yang nyata. Untuk mempercepat kinerja pegawai, Javan menyediakan sebuah *package* bernama Laravolt. Selain menawarkan kemudahan dalam pembuatan fitur, penggunaan Laravolt juga memperkuat pemahaman mengenai model MVC. Dunia IT sangatlah luas, banyak *platform* yang masih belum dikenal di Indonesia seperti Joget Workflow. Penggunaan Joget Workflow ini dipelajari selama dua bulan. Di sana, banyak hal baru yang dipelajari terutama penulisan *query*. Selain belajar mengenai hal berbaur teknis, dipelajari juga mengenai dokumentasi proyek. Dokumentasi proyek meliputi berbagai dokumen yang mempunyai fungsinya masing-masing seperti dokumentasi analisis, dokumentasi kebutuhan sistem, dokumentasi pengujian dan lain-lain. Agar pengembangan aplikasi berjalan dengan lancar maka kemampuan membaca dokumentasi sangat diperlukan. Hal ini diperlukan untuk menghindari dan mengurangi resiko kesalahan selama pengerjaan. Pihak Javan juga menyediakan wadah untuk menambah wawasan mengenai teknologi baru dalam kegiatan “Jumat berkualitas”. Hal yang dipelajari dari kegiatan ini adalah implementasi *automated testing*, membuat *progressive web apps* dan pelatihan *tools* BPMN bernama Carrot.

Magang di Javan merupakan pengalaman magang pertama di dunia IT. Selama magang terdapat orang-orang baru yang mempunyai buah pikiran yang berbeda. Dari sana didapatkan pembelajaran mengenai kehidupan terutama dalam bidang pekerjaan. Semua yang dilakukan memerlukan proses karena tidak ada kesuksesan yang didapat secara instan. Bekerja sama, beradaptasi, belajar dan berbagi merupakan modal untuk meningkatkan kualitas diri. Selain itu didapatkan juga manajemen waktu secara seimbang antara magang, kuliah dan keluarga. Hal ini sangat penting karena manajemen waktu akan berdampak baik seperti dapat menikmati waktu yang ada secara maksimal dan tidak membuang waktu secara sia-sia.

4.2.2 Hambatan dan Tantangan Magang

Di samping memperoleh manfaat dari aktivitas magang ini, adapun ditemui hambatan dan tantangan. Hambatan yang ditemui berasal dari diri sendiri yaitu rasa malas. Rasa malas ini tidak hanya membuat waktu terbuang sia-sia tetapi juga akan merusak rencana yang telah disusun dalam kurun waktu tertentu. Selain rasa malas, dirasakan juga pikiran mudah jenuh apabila melakukan rutinitas yang sama setiap harinya. Kegiatan magang diharuskan rutin untuk berangkat pagi dan

pulang sore pada hari kerja. Jam kerja kantor kurang lebih delapan jam, belum termasuk jam lembur. Hal ini kadang membuat semangat dan fokus menurun.

Sejak mulai magang, terdapat berbagai macam tantangan yang ditemui. Tantangan awal yang harus dihadapi adalah mempelajari *framework* Laravel dan Git. Di dalam Laravel terdapat banyak *package* yang perlu dikenal termasuk Laravolt. Pada tiap proyek tidak pasti menggunakan *platform* atau *framework* yang sama, sehingga kemampuan beradaptasi menjadi penting dalam kasus ini. *Platform* asing yang ditemui adalah Joget Workflow. Berbagai macam bahasa pemrograman bisa digunakan dalam *platform* ini, antara lain: Java dan SQL. Hal ini membuat tantangan muncul untuk memanfaatkan semua bahasa pemrograman tadi di dalam satu *platform*.

4.2.3 Kontribusi Selama Magang

Magang sebagai seorang *junior programmer* telah memberikan kontribusi yang cukup selama magang di Javan. Walau begitu, pekerjaan yang diterima akan dikerjakan secara maksimal. Apabila ditemui pekerjaan yang sulit, maka pekerjaan itu perlu didiskusikan terlebih dahulu dengan mentor. Hal tersebut dilakukan agar pekerjaan dapat selesai dengan hasil yang baik, tepat waktu dan terhindar dari *blocking* proyek. Sejak bergabung di dalam proyek EHR, telah dikerjakan kurang lebih seratus *task*. Selain mengerjakan *task*, juga memberikan usulan dalam pembuatan fitur. Pembuatan fitur unggah dokumen menggunakan *grid form* merupakan salah satu usulan yang disampaikan. Usulan tersebut disampaikan agar mempermudah dan mempercepat alur eksekusi. Apabila pekerjaan telah selesai, diluangkan waktu untuk menguji hasil pekerjaan terlebih dahulu. Hal ini dilakukan untuk membantu pekerjaan *tester* karena sesuatu yang dibuat perlu dicicipi terlebih dahulu sebelum disajikan kepada orang lain. Disamping itu, juga ikut meramaikan kegiatan pelatihan yang diadakan oleh pihak Javan untuk menambah kemampuan di luar jam pekerjaan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Selama pengerjaan Sistem Electronic Human Resource atau EHR, implementasi metode pada Joget Workflow dapat mempermudah pengembangan sistem dengan fungsionalitas proses bisnis pada penyimpanan data *form*. Hal ini juga menjadi catatan dari para pengembang karena berbagai macam *form* dengan karakteristiknya dapat ditangani oleh metode yang sesuai pada Joget Workflow. Karakteristik dengan skema yang kompleks seperti penggunaan kondisi percabangan *if* atau perulangan *while* pada Java dapat menggunakan metode Bean Shell Form Binder. Skema dengan fungsi *query* SQL dapat menggunakan JDBC Binder. Sedangkan untuk skema statis yang hanya membutuhkan penyimpanan data sementara pada satu tabel tanpa *query* dapat menggunakan Workflow Form Binder. Hal ini dapat membantu sistem dalam menangani penyimpanan data pada *form* sesuai dengan proses bisnis. Selain itu pada setiap *form* telah ditambah proses validasi agar data yang disimpan sesuai dengan format yang diinginkan.

5.2 Saran

Implementasi penyimpanan data pada Joget Workflow yang dilakukan dalam laporan ini masih mempunyai banyak kekurangan. Untuk mengoptimasi proses pengembangan selanjutnya, maka perlu dilakukan beberapa saran sebagai berikut:

- Perlu ditambah notifikasi (*pop up*) pemberitahuan data berhasil disimpan di setiap *form*.
- *Query* SQL yang rumit perlu dipecah ke dalam beberapa *view tables* atau *stored procedures*.
- Perlu ditambah pengujian secara otomatis (*automated testing*) karena setiap *form* mempunyai *field* yang tidak sedikit.

DAFTAR PUSTAKA

- Administrator, & Andrew. (2019, November 28). *Installation on Linux - Knowledge Base for v6 - Joget / Community*. Dipetik April 30, 2020, dari <https://dev.joget.org/community/display/KBv6/Installation+on+Linux>
- Administrator, & Justin. (2017, Agustus 23). *Directory Form Binder - Knowledge Base for v6 - Joget / Community*. Dipetik April 20, 2020, dari [dev.joget.org: https://dev.joget.org/community/display/KBv6/Directory+Form+Binder](https://dev.joget.org/community/display/KBv6/Directory+Form+Binder)
- Administrator, & Justin. (2017, Agustus 23). *Workflow Form Binder - Knowledge Base for v6 - Joget / Community*. Dipetik April 24, 2020, dari <https://dev.joget.org/community/display/KBv6/Workflow+Form+Binder>
- Administrator, & Justin. (2018, Januari 15). *Bean Shell Form Binder - Knowledge Base for v6 - Joget / Community*. Dipetik April 24, 2020, dari <https://dev.joget.org/community/display/KBv6/Bean+Shell+Form+Binder>
- Ambo, S. N., & Ghufron, M. (2015). Rancang Bangun Aplikasi Human Resource Information System (HRIS) Menggunakan Metode Model View Controller (MVC). *Seminar Nasional Sains dan Teknologi* .
- Aminudin, A. (2015). *Cara Efektif Belajar Framework LARAVEL*. Yogyakarta: Lokomedia.
- David, K.-I. O. (2017). Implementasi Joget Workflow Pada Proses Bisnis Unit Pelaksana Teknis (UPT) Layanan Internasional Universitas Andalas. 128-129.
- Destiningrum, M., & Adrian, Q. J. (2017). Sistem Informasi Penjadwalan Dokter Berbasis Web dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre). *TEKNOINFO*, 30-37.
- DocForge. (2015, Juli 23). *Web application framework - DocForge*. Dipetik Mei 5, 2020, dari https://web.archive.org/web/20150723163302/http://docforge.com:80/wiki/Web_application_framework
- Franky, & Devitra, J. (2016). Analisis dan Perancangan Sistem Informasi. *Jurnal Manajemen Sistem Informasi*, 1(2), 185-190.
- Hugo. (2019, Juli 16). *Joget Workflow v6 Knowledge Base - Knowledge Base for v6 - Joget / Community*. Dipetik April 20, 2020, dari <https://dev.joget.org/community/display/KBv6/Joget+Workflow+v6+Knowledge+Base>

- Jahan, S. S. (2014). Human Resource Information System (HRIS): A Theoretical Perspective. *Journal of Human Resource and Sustainability*, 2, 33-39.
- Joget Workflow. (2019). *Joget Workflow - Joget: Open Source Workflow and Low Code Platform for Digital Transformation*. Dipetik April 23, 2020, dari <https://www.joget.com/joget-workflow.html>
- Justin. (2019, Juli 2). *Hash Variable - Knowledge Base for v6 - Joget | Community*. Dipetik April 21, 2020, dari <https://dev.joget.org/community/display/KBv6/Hash+Variable>
- Justin, & Andrew. (2019, Oktober 29). *JDBC Form Binder - Knowledge Base for v6 - Joget | Community*. Dipetik April 24, 2020, dari <https://dev.joget.org/community/display/KBv6/JDBC+Form+Binder>
- Muhammad, H. F., & Niki, P. S. (2018). Pengembangan Human Resource Information System (HRIS). *JUPITER (Jurnal Pendidikan Teknik Elektro)*, 03(02), 1-12.
- OutSystems. (2020, November 2020). *Setting Up OutSystems*. Dipetik Desember 15, 2020, dari https://success.outsystems.com/Documentation/11/Setting_Up_OutSystems
- Parkar, V. V., Shinde, P. P., Gadade, S. C., & Shinde, P. M. (2015). Utilization of Laravel Framework for Development of Web Based. *IOSR Journal of Computer Engineering*, 36-41.
- Shiri, S. (2012). Effectiveness of Human Resource Information System on HR Functions of the Organization: A Cross Sectional Study. *US-China Education Review, A* 9, 830-839.
- Sulistiono, H. (2018). *Coding Mudah dengan Codeigniter, JQuery, Bootstrap dan Datatable*. Jakarta: Elex Media Komputindo.
- Utami, E., & Sukrisno. (2008). *Mengoptimalkan Query pada Microsoft SQL Server*. Yogyakarta: Penerbit Andi.
- Vukotic, A., & Goodwill, J. (2011). *Apache Tomcat 7*. New York: Springer Science+Business Media.