

SKRIPSI

«JUDUL BAHASA INDONESIA»



LUCKY SENJAYA DARMAWAN

NPM: 2012730009

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN**

«tahun»

UNDERGRADUATE THESIS

«JUDUL BAHASA INGGRIS»



LUCKY SENJAYA DARMAWAN

NPM: 2012730009

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND
SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
«tahun»**

DAFTAR ISI

DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	1
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 DASAR TEORI	3
2.1 Skripsi	3
2.2 <i>Business Process Management (BPM)</i>	3
2.2.1 <i>Komponen Business Process</i>	3
2.2.2 <i>Siklus Business Process Management</i>	4
2.3 <i>Business Process Model Notation</i>	5
2.3.1 <i>Event</i>	5
2.3.2 <i>Activity</i>	5
2.3.3 <i>Gateway</i>	6
2.3.4 <i>Data</i>	6
2.3.5 <i>Artifact</i>	7
2.3.6 <i>Pools dan Lanes</i>	7
2.4 <i>Business Process Management System (BPMS)</i>	7
2.4.1 <i>Arsitektur BPMS</i>	7
2.5 <i>BPMS Camunda</i>	8
2.5.1 <i>Arsitektur BPMS Camunda</i>	8
2.6 <i>Pengoperasian Camunda</i>	10
2.6.1 <i>Instalasi</i>	10
2.6.2 <i>Mempersiapkan Proyek Java</i>	11
2.6.3 <i>Memodelkan Proses</i>	11
2.6.4 <i>Menjalankan Camunda</i>	11
2.7 <i>JavaMail</i>	11
3 ANALISIS	13
3.1 Analisis BPMN	13
3.2 Analisis Integrasi Sistem Email	13
3.2.1 <i>Integrasi Camunda dengan Sistem Email</i>	13
3.2.2 <i>Mekanisme Integrasi Sistem Email</i>	13

4 PERANCANGAN	15
4.1 Perancangan Sistem	15
4.1.1 Email	15
4.1.2 Algoritma Pengiriman Email	16
4.2 Tugas Desainer	16
DAFTAR REFERENSI	17
A KODE PROGRAM	19
B HASIL EKSPERIMEN	21

DAFTAR GAMBAR

2.1	Komponen BPM	3
2.2	Siklus BPM	4
2.3	Notasi <i>Event</i>	5
2.4	Notasi <i>Task</i>	6
2.5	Notasi <i>Gateway</i>	6
2.6	Notasi <i>Data</i>	6
2.7	Notasi <i>Artifact</i>	7
2.8	Notasi <i>Lanes dan Pools</i>	7
2.9	Arsitektur BPMS	8
2.10	Arsitektur BPMS Camunda	8
2.11	Camunda Modeler	9
2.12	Camunda Tasklist	9
2.13	Camunda Cockpit	10
2.14	Camunda Admin	10
B.1	Hasil 1	21
B.2	Hasil 2	21
B.3	Hasil 3	21
B.4	Hasil 4	21

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Workflow merupakan pemodelan proses bisnis yang dapat digambarkan sebagai *flow map* atau BPMN (*Business Process Model and Notation*). *Workflow* ini dapat diotomasi menggunakan BPMS (*Business Process Management System*), seperti Camunda. Agar eksekusi *workflow* lebih alamiah dengan model komunikasi organisasi saat ini, maka *event* dapat dipropagasi dan diintegrasikan dengan sistem email.

Dalam skripsi ini, akan dibuat suatu integrasi antara *user task* dan sistem email. *User task* adalah suatu tugas yang perlu dilakukan oleh pengguna. Ketika ada suatu *user task*, sistem email akan mengirimkan email ke pengguna yang akan mengerjakan task tersebut. Email tersebut akan berisi tautan yang mengarah ke tugas yang perlu dikerjakan tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dipaparkan sebelumnya, maka rumusan masalah dalam skripsi ini adalah sebagai berikut :

1. Bagaimana memodelkan *workflow* dengan BPMN?
2. Event-event *workflow* apa saja yang dapat dipropagasi ke sistem email?
3. Bagaimana mekanisme propagasi dan integrasi *workflow* dengan sistem email?

1.3 Tujuan

Berdasarkan rumusan masalah yang dipaparkan sebelumnya, tujuan dari penelitian ini adalah :

1. Memodelkan *workflow* dengan BPMN.
2. Mengidentifikasi event-event *workflow* yang dapat dipropagasi ke sistem email.
3. Menentukan mekanisme propagasi dan mengintegrasikan *workflow* dengan sistem email.

1.4 Batasan Masalah

1. Pemodelan BPMN menggunakan versi 2.0 dan menggunakan editor Camunda Modeler versi 1.7.2.
2. Perangkat lunak BPMS Camunda yang digunakan merupakan versi 7.6.0 dan berjalan pada tomcat versi 8.0.24.

1.5 Metodologi

Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Melakukan studi mengenai proses bisnis, *workflow*, *Business Process Model and Notation (BPMN)*, *Business Process Management System (BPMS)*, dan sistem e-mail.
2. Memodelkan proses bisnis tertentu menggunakan BPMN.
3. Mengidentifikasi *event-event* dari *workflow* yang dapat diintegrasikan dengan sistem email.
4. Merancang integrasi sistem email.
5. Mengimplementasikan sistem email ke BPMS.
6. Melakukan pengujian fungsionalitas.

1.6 Sistematika Pembahasan

1. Bab 1 Pendahuluan, berisi latar belakang masalah, rumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, dan sistematika penulisan.
2. Bab 2 Dasar Teori, berisi dasar teori yang mencakup *Business Process Management*, *Business Process Model and Notation (BPMN)*, *Business Process Management System (BPMS)*, dan sistem e-mail.
3. Bab 3 Analisis, Berisi analisis BPMN dengan menggunakan contoh kasus dan analisis integrasi sistem email.
4. Bab 4 Perancangan, Berisi rancangan algoritma yang akan dibuat dan format email, .
5. Bab 5 Implementasi, dan Pengujian Berisi implementasi dari program yang dibuat dan pengujian aplikasi berdasarkan contoh kasus pada bab tiga.
6. Bab 6 Penutup, Berisi kesimpulan dan saran-saran untuk pengembangan selanjutnya.

BAB 2

DASAR TEORI

2.1 Skripsi

Bab dua ini berisi dasar-dasar teori yang terkait dengan BPM, BPMN, BPMS, dan sistem email

2.2 *Business Process Management* (BPM)

Business Process adalah kumpulan dari *event*/kejadian, *activity*/kegiatan, dan *decision point*/keputusan serta melibatkan sejumlah aktor dan objek yang bertujuan untuk menghasilkan nilai dalam bentuk produk/jasa yang berguna bagi konsumen. Dari definisi proses bisnis, *Business Process Management* dapat didefinisikan sebagai kumpulan metode, teknik, dan alat untuk menemukan, menganalisa, mendesain kembali, menjalankan, dan mengawasi proses bisnis.

2.2.1 *Komponen Business Process*

Business Process Management memiliki komponen-komponen sebagai berikut :

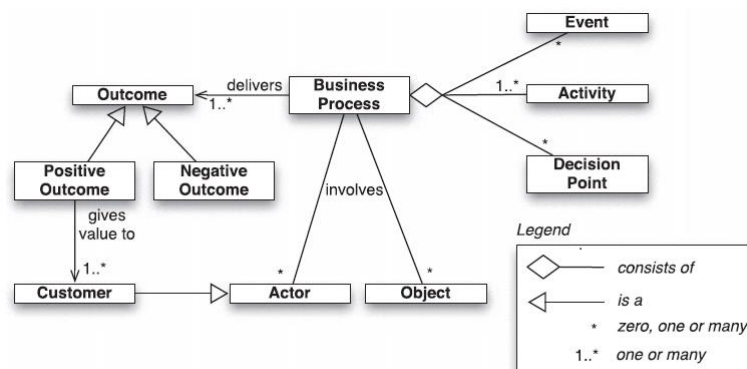


Fig. 1.1 Ingredients of a business process

Gambar 2.1: Komponen BPM

Event

Event adalah kejadian yang terjadi saat proses bisnis berjalan.

Activity

Activity adalah kumpulan kegiatan yang dapat dikerjakan. Ketika suatu *Activity*

berupa sebuah kegiatan yang sederhana, *activity* disebut dengan *task*.

Decision Point

Decision point adalah keputusan yang mempengaruhi proses selanjutnya.

Actor

Actor berupa individu, organisasi, maupun sistem yang mempengaruhi proses bisnis.

Object

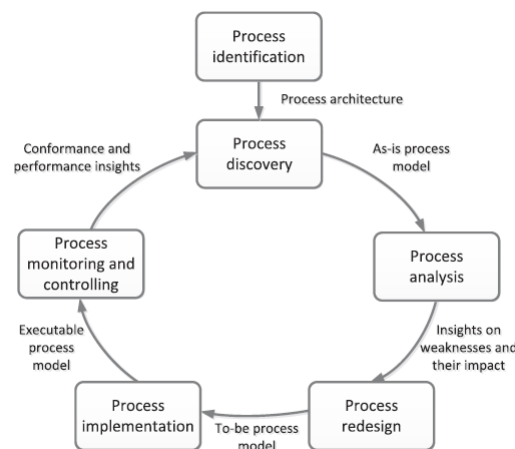
Object dapat berupa objek fisik (peralatan, bahan baku, produk, dokumen) maupun non fisik (dokumen elektronik, basis data elektronik).

Positive/Negative Outcome

Hasil dari bisnis proses dapat menghasilkan nilai bagi konsumen (positif) atau tidak menghasilkan nilai (negatif).

2.2.2 Siklus *Business Process Management*

Suatu proses bisnis tidak selalu berjalan dengan baik. Banyak hal yang tidak diantisipasi sebelumnya dapat mengganggu proses bisnis. Untuk menjaga kualitas dari sebuah proses bisnis diperlukan pengawasan dan kontrol pada suatu fase tertentu serta perbaikan apabila diperlukan. Maka dari itu, suatu bisnis proses dapat dilihat sebagai suatu siklus yang terus menerus meningkatkan kualitasnya. Siklus dalam proses bisnis berupa :



Gambar 2.2: Siklus BPM

Process Identification

Pada fase ini, suatu masalah bisnis ditemukan, kemudian proses-proses yang berhubungan dengan masalah bisnis tersebut diidentifikasi, dibatasi, dan dihubungkan satu sama lain. Proses ini terbagi menjadi dua tahap, yaitu *designation* dan *evaluation*. Tahap *designation* bertujuan untuk mengenali proses-proses yang ada dan hubungan antar proses tersebut. Sedangkan tahap *evaluation* memprioritaskan proses-proses yang menghasilkan nilai dan mempertimbangkan proses yang memiliki risiko atau tidak menghasilkan nilai. Fase ini menghasilkan arsitektur dari proses bisnis yang merepresentasikan proses bisnis dan relasi-relasinya.

Process Discovery

Setiap proses yang relevan dengan masalah bisnis didokumentasikan, umumnya dalam bentuk model proses. Fase ini menghasilkan *as-is process model*

Process Analysis

Pada fase ini, masalah pada model proses diidentifikasi, didokumentasikan, dan diukur kinerjanya dengan ukuran yang telah ditetapkan. Hasil dari fase ini adalah kumpulan masalah pada proses model.

Process Redesign

Tujuan dari fase ini adalah membuat perubahan pada proses yang dapat mengatasi berbagai kumpulan masalah yang telah diidentifikasi pada fase sebelumnya. Proses ini menghasilkan *to-be process model*.

Process Implementation

Pada fase ini, model proses diimplementasikan untuk dieksekusi menggunakan *Business Process Management System*.

Process Monitoring and Controlling

Setelah proses bisnis berjalan pada BPMS, berbagai data yang relevan dikumpulkan dan dianalisa untuk menentukan kualitas dari proses. Apabila terdapat masalah baru yang ditemukan, maka proses diulangi.

2.3 Business Process Model Notation

Business Process Model Notation (BPMN) adalah notasi grafis yang menggambarkan langkah-langkah dalam proses bisnis. Notasi-notasi tersebut terdiri dari :

2.3.1 Event

Event merupakan kejadian yang terjadi pada proses bisnis yang dilambangkan dengan bentuk lingkaran. Notasi event secara umum terbagi menjadi tiga, yaitu *start event*, *intermediate event*, dan *end event*. *Start event* menunjukkan dimulainya proses, *intermediate event* dapat muncul ketika proses berjalan, sedangkan *end event* menunjukkan berakhirnya proses.

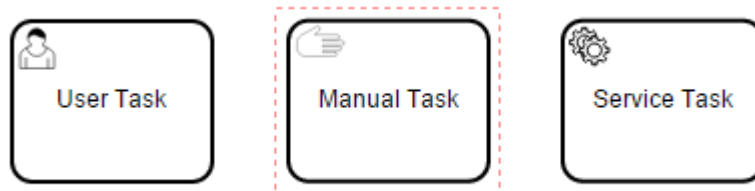


Gambar 2.3: Notasi *Event*

2.3.2 Activity

Activity merupakan kumpulan kegiatan yang dapat dikerjakan. Sebuah *task* merupakan bagian dari *Activity* yang tidak dapat dipecah lagi. Beberapa contoh *Task* adalah :

1. *User Task*, yaitu pekerjaan yang perlu dilakukan oleh manusia melalui sistem.
2. *Manual Task*, yaitu pekerjaan yang dilakukan manusia tanpa melalui sistem.
3. *Service Task*, yaitu pekerjaan yang dilakukan oleh sistem dengan memanggil kode Java.



Gambar 2.4: Notasi *Task*

2.3.3 Gateway

Gateway merupakan simbol yang menentukan percabangan dan penggabungan jalur dalam proses. Gateway dilambangkan dengan belah ketupat. Beberapa macam adalah :

- *Exclusive Gateway* (XOR) berarti memilih salah satu dari cabang yang ada.
- *Inclusive Gateway* berarti memilih satu, beberapa, atau seluruh cabang yang ada.
- *Parallel Gateway* berarti mengerjakan proses pada seluruh cabang yang ada.
- *Event Based* berarti mengerjakan proses setelah suatu *event* selesai.



Gambar 2.5: Notasi *Gateway*

2.3.4 Data

Data Object melambangkan informasi yang berjalan dalam proses seperti dokumen, e-mail, atau surat. Sedangkan *Data Store* merupakan tempat proses membaca atau menyimpan data seperti basis data atau rak.



Gambar 2.6: Notasi *Data*

2.3.5 Artifact

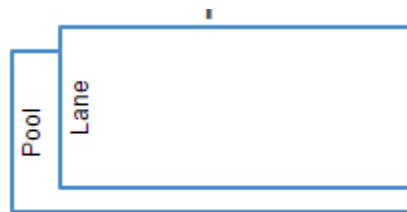
Artifact tidak mempengaruhi jalannya proses, tetapi hanya sebagai informasi tambahan agar proses lebih mudah dimengerti. Terdapat dua jenis, yaitu *Text Annotation* dan *Group*



Gambar 2.7: Notasi *Artifact*

2.3.6 Pools dan Lanes

Lanes digunakan untuk memberikan kumpulan *tasks* kepada yang bertanggung jawab untuk mengerjakannya. Sedangkan *Pools* merupakan kumpulan dari *Lanes*.



Gambar 2.8: Notasi *Lanes dan Pools*

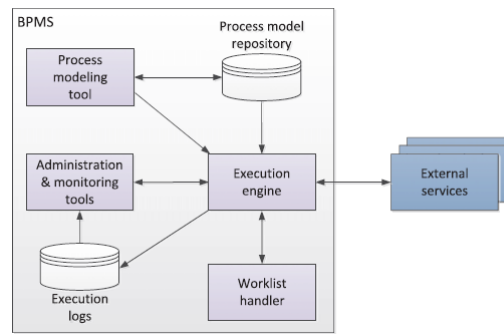
2.4 Business Process Management System (BPMS)

Business Process Management System (BPMS) adalah sistem yang mengkoordinasikan otomatisasi proses bisnis. Tujuan dari BPMS adalah menyelesaikan proses pada waktu yang ditentukan dan menggunakan sumber daya yang tepat.

2.4.1 Arsitektur BPMS

Komponen-komponen BPMS beserta hubungannya yang ditunjukkan pada Gambar 2.9 terdiri dari :

- *Execution Engine*, menyediakan beberapa fungsi seperti mengeksekusi proses, mendistribusikan *task*, mengambil dan menyimpan data yang diperlukan.
- *Process Modeling Tool*, *tool* untuk membuat model proses.
- *Worklist Handler*, *tool* untuk mendistribusikan pekerjaan.
- *Administration dan Monitoring Tool tools* untuk administrasi dan memonitor proses.



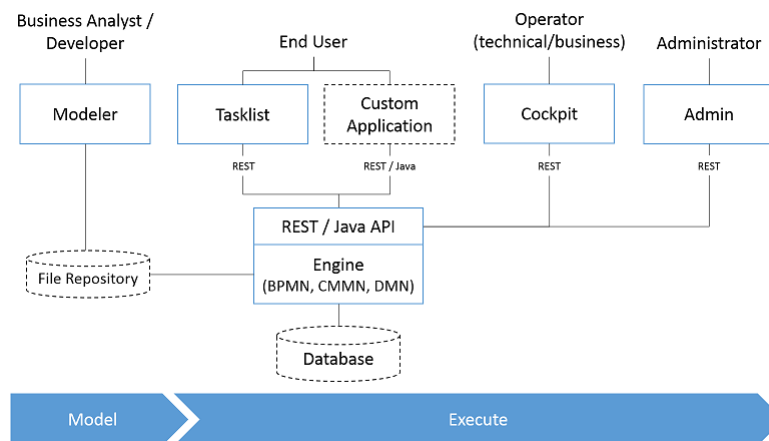
Gambar 2.9: Arsitektur BPMS

2.5 BPMS Camunda

Camunda adalah *framework* BPMS berbasis Java yang mendukung *workflow* BPMN dan otomatisasi proses bisnis.

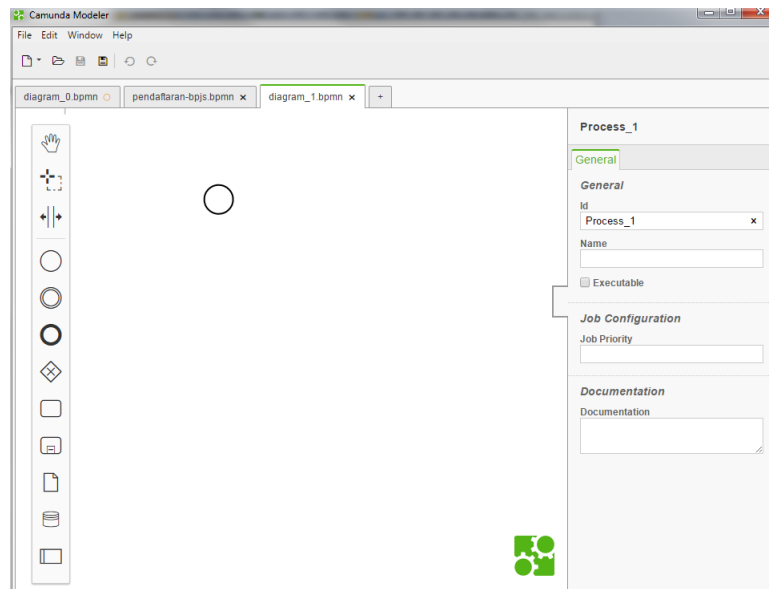
2.5.1 Arsitektur BPMS Camunda

Komponen-komponen pada BPMS Camunda adalah sebagai berikut :



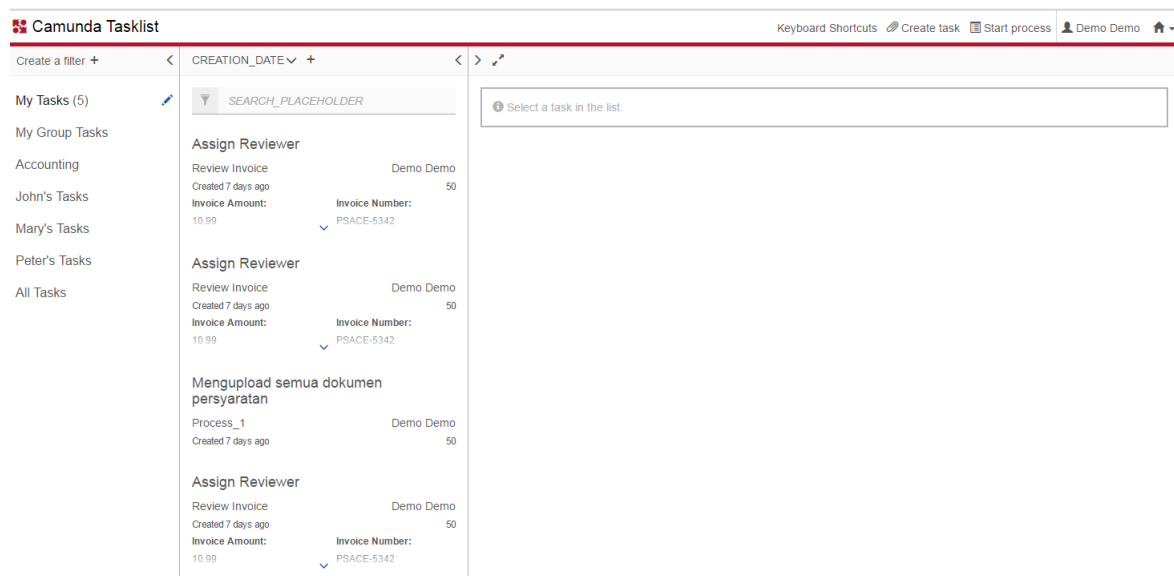
Gambar 2.10: Arsitektur BPMS Camunda

- *Modeler*, *tool* untuk membuat diagram BPMN yang dapat dieksekusi. Camunda *Modeler* menyediakan berbagai notasi yang diperlukan untuk membuat diagram BPMN. Terdapat pula beberapa pengaturan yang dapat dimasukkan ke dalam notasi.



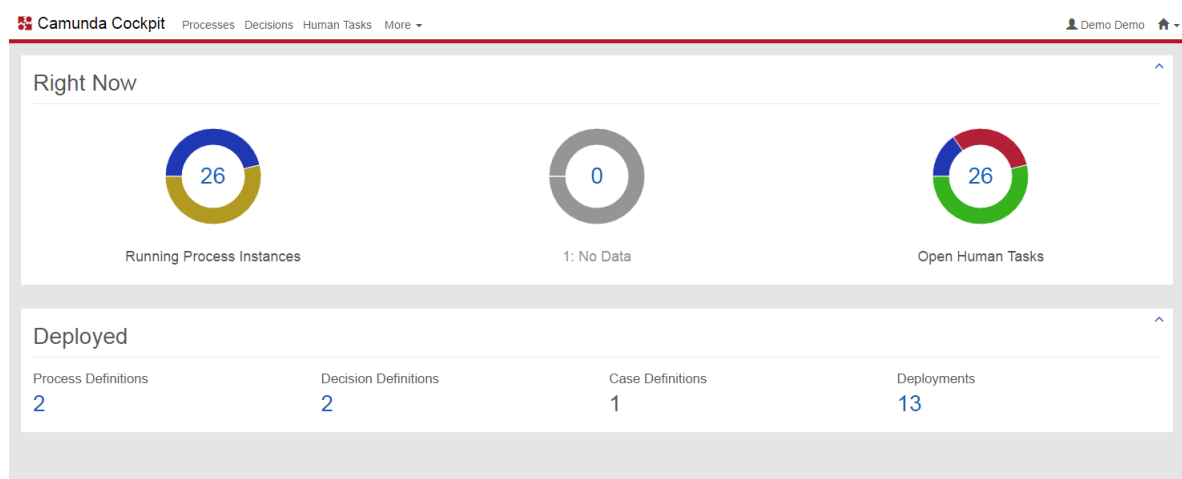
Gambar 2.11: Camunda Modeler

- *Tasklist*, tempat pengguna mengakses dan mengerjakan tugas. Tugas yang dikerjakan mengikuti alur model proses (BPMN) yang telah dibuat.



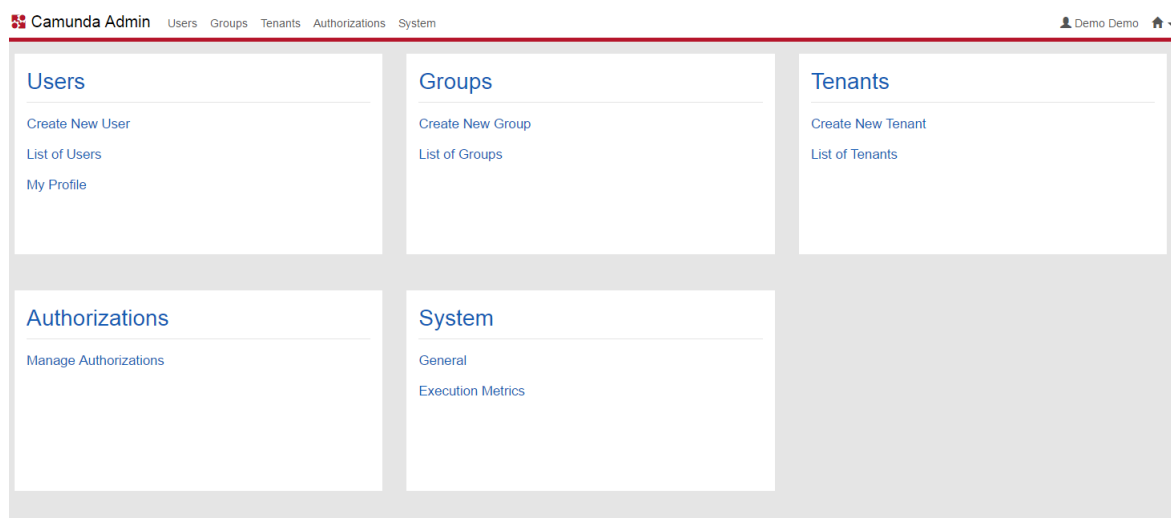
Gambar 2.12: Camunda Tasklist

- *Cockpit*, memeriksa proses yang sedang berjalan maupun proses yang sudah selesai.



Gambar 2.13: Camunda Cockpit

- *Admin*, memiliki tugas untuk mengatur, mengelompokkan, dan memberi izin kepada pengguna untuk melakukan tugas.



Gambar 2.14: Camunda Admin

- *Custom Application*, aplikasi lain yang diintegrasikan dengan Camunda menggunakan Java atau REST API.

2.6 Pengoperasian Camunda

2.6.1 Instalasi

Untuk menjalankan Camunda, diperlukan beberapa *tool*, yaitu :

- Java JDK 1.7+.
- Apache Maven atau Maven yang sudah terpasang di Eclipse.
- Web browser.
- Camunda BPM Platform

- Camunda Modeler

2.6.2 Mempersiapkan Proyek Java

Membuat Proyek Maven di Eclipse.

1. Pilih File / New / Other / Maven / Maven Project kemudian pilih *Next*.
2. Pilih Create a simple project (skip archetype selection) kemudian pilih *next*.
3. Pilih Packaging : war, kemudian pilih Finish.

Tambahkan *Camunda Maven Dependencies*.

Tambahkan sebuah kelas Process Application.

Tambahkan *Deployment Descriptor* di META-INF/processes.xml.

2.6.3 Memodelkan Proses

1. Membuat file BPMN baru dengan File / New File / BPMN Diagram.
2. Memodelkan proses.
3. Menambahkan Form HTML pada User Task.
4. Menambahkan Service Task.

2.6.4 Menjalankan Camunda

1. Klik kanan pom.xml dan pilih Run As / Maven Install. Langkah ini akan menghasilkan file WAR di folder target.
2. *Copy paste* file WAR ke CAMUNDA_HOME / server / apache-tomcat / webapps folder.

2.7 JavaMail

JavaMail adalah Java API yang digunakan untuk mengirim dan menerima email melalui SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3), dan IMAP (Internet Message Access Protocol).

BAB 3

ANALISIS

Bab ini berisi.

3.1 Analisis BPMN

3.2 Analisis Integrasi Sistem Email

3.2.1 Integrasi Camunda dengan Sistem Email

Integrasi Camunda dengan sistem email pada skripsi ini bertujuan untuk memberi tahu pengguna Camunda apabila ada *tasks* yang perlu dikerjakan oleh pengguna. Ketika pengguna menerima email mengenai *tasks* yang perlu dikerjakan, pengguna dapat langsung mengerjakannya.

Camunda memiliki berbagai jenis *tasks* seperti *user tasks*, *manual tasks*, *service task*, dan lainnya. Karena proses integrasi email dengan Camunda melibatkan pengguna (pengguna menerima pemberitahuan pekerjaannya melalui email), *task* yang akan diintegrasikan dengan sistem email adalah *user tasks*.

3.2.2 Mekanisme Integrasi Sistem Email

User tasks memiliki atribut *Task Listener* yang dapat mengeksekusi perintah. *Task Listener* memiliki dua atribut, yaitu *Event Type* dan *Listener Type*. Terdapat empat pilihan dari *Event Type*, yaitu *create*, *assignment*, *complete*, *delete*.

- Create, perintah dieksekusi ketika *task* telah dibuat dan siap untuk dikerjakan.
- Assignment, perintah dieksekusi ketika pengguna yang akan mengerjakan *task* sudah ditentukan.
- Complete, perintah dieksekusi ketika *task* sudah dikerjakan dan sebelum *task* dihapus.
- Delete, perintah dieksekusi setelah *task* dihapus.

Untuk mengintegrasikan *user tasks* dengan email, *event type* yang dapat digunakan adalah *assignment*. *Event complete* dan *delete* tidak dapat digunakan untuk memberi tahu pengguna karena setelah *task* selesai dan dihapus, alamat email untuk *Task* selanjutnya belum didapatkan sementara *event* sudah selesai dipanggil.

Apabila menggunakan *event create*, *task* harus memiliki pemiliknya masing-masing ketika BPMN dibuat. Bila pemilik *task* belum ditentukan, email tidak akan terkirim, karena

event create sudah selesai dipanggil sebelum *task* memiliki pemilik. Sedangkan pada *event assignment*, pengiriman email dilakukan setelah *task* didelegasikan ke masing-masing user.

BAB 4

PERANCANGAN

Untuk mempropagasi email, diperlukan perancangan sistem dan peran yang harus dilakukan oleh desainer.

4.1 Perancangan Sistem

Berdasarkan analisis di bab sebelumnya, maka untuk mempropagasi email diperlukan beberapa persyaratan, yaitu :

1. Model proses menggunakan BPMN yang sudah dilengkapi form HTML untuk *user task*, implementasi untuk *service task* dan atribut lain yang diperlukan.
2. Kumpulan *user/group* yang akan mengerjakan tugas.
3. Alamat email yang merepresentasikan sistem.
4. Algoritma untuk mengirim email.
5. Business Process Management System (BPMS), yaitu tools untuk mengotomasi jalannya proses.

4.1.1 Email

Alamat email yang digunakan untuk merepresentasikan sistem berbasis Gmail SMTP. Gmail SMTP yang akan digunakan memiliki konfigurasi sebagai berikut :

- Alamat server = smtp.gmail.com.
- Port = 587.
- Username Gmail.
- Password Gmail.

Email yang akan dikirimkan ke pengguna memiliki format :

1. Subjek :
2. Nama pengguna.
3. Nama *task*.
4. Link ke *task*, yaitu [http://localhost/camunda/app/tasklist/default/#/?task=\(id task\)](http://localhost/camunda/app/tasklist/default/#/?task=(id task)).

4.1.2 Algoritma Pengiriman Email

Berikut adalah algoritma untuk mengirimkan email.

1. Mendapatkan id dari *task*.
2. Mendapatkan email pengguna yang akan mengerjakan *task*.
3. Membangkitkan subjek dan isi email yang berisi tautan ke task yang akan dikerjakan. Tautan didapatkan dari id *task*.
4. Menghubungkan kode ke email dengan mengisi *username* dan *password*
5. Mengirim email.

4.2 Tugas Desainer

Berdasarkan perancangan sistem di atas, seorang desainer model proses memiliki beberapa tugas, yaitu :

1. Membuat alamat email yang merepresentasikan sistem.
2. Merancang model proses.
3. Menambahkan form HTML pada user task, implementasi service task, task listener untuk propagasi email, dan berbagai atribut lainnya sesuai kebutuhan.
4. Menambahkan username, password, dan host email pada kode task listener yang berhubungan dengan propagasi email.
5. Menambahkan user/group yang akan mengerjakan pada Camunda Admin.
6. Mendelegasikan task kepada user/group yang akan mengerjakan.
7. Menjalankan dan memulai proses.

DAFTAR REFERENSI

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyFurSet.java

```

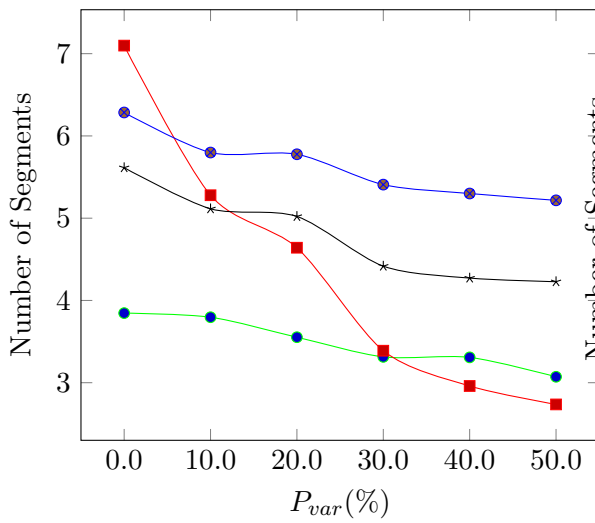
1|
2| import java.util.ArrayList;
3| import java.util.Collections;
4| import java.util.HashSet;
5|
6| /**
7|  *
8|  * @author Lionov
9|  */
10|
11| //class for set of vertices close to furthest edge
12| public class MyFurSet {
13|     protected int id; //id of the set
14|     protected MyEdge FurthestEdge; //the furthest edge
15|     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
16|     protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each
17|         trajectory
18|     protected ArrayList<Integer> closeID; //store the ID of all vertices
19|     protected ArrayList<Double> closeDist; //store the distance of all vertices
20|     protected int totaltrj; //total trajectories in the set
21|
22|     /**
23|      * Constructor
24|      * @param id : id of the set
25|      * @param totaltrj : total number of trajectories in the set
26|      * @param FurthestEdge : the furthest edge
27|      */
28|     public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
29|         this.id = id;
30|         this.totaltrj = totaltrj;
31|         this.FurthestEdge = FurthestEdge;
32|         set = new HashSet<MyVertex>();
33|         ordered = new ArrayList<ArrayList<Integer>>();
34|         for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
35|         closeID = new ArrayList<Integer>(totaltrj);
36|         closeDist = new ArrayList<Double>(totaltrj);
37|         for (int i = 0;i <totaltrj;i++) {
38|             closeID.add(-1);
39|             closeDist.add(Double.MAX_VALUE);
40|         }
41|     }
42|
43|     /**
44|      * set a vertex into the set
45|      * @param v : vertex to be added to the set
46|      */
47|     public void add(MyVertex v) {
48|         set.add(v);
49|     }
50|
51|     /**
52|      * check whether vertex v is a member of the set
53|      * @param v : vertex to be checked
54|      * @return true if v is a member of the set, false otherwise
55|      */
56|     public boolean contains(MyVertex v) {
57|         return this.set.contains(v);
58|     }
59| }

```

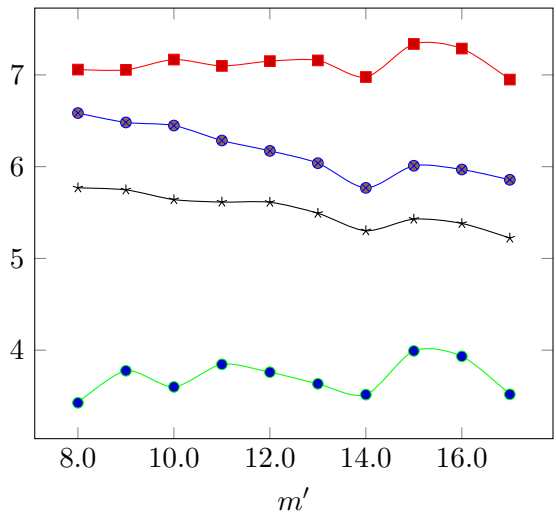

LAMPIRAN B

HASIL EKSPERIMEN

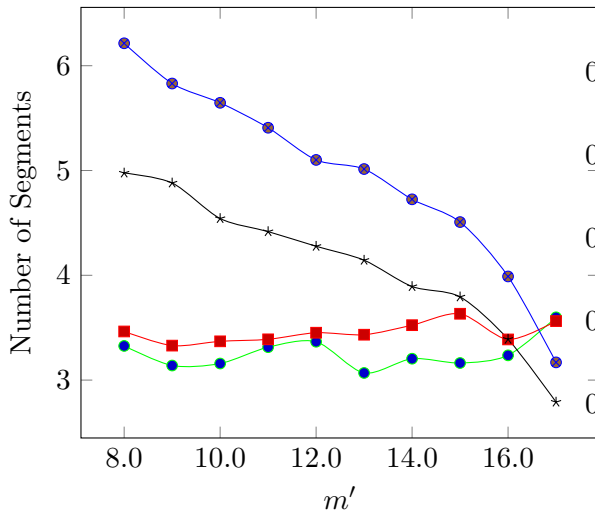
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



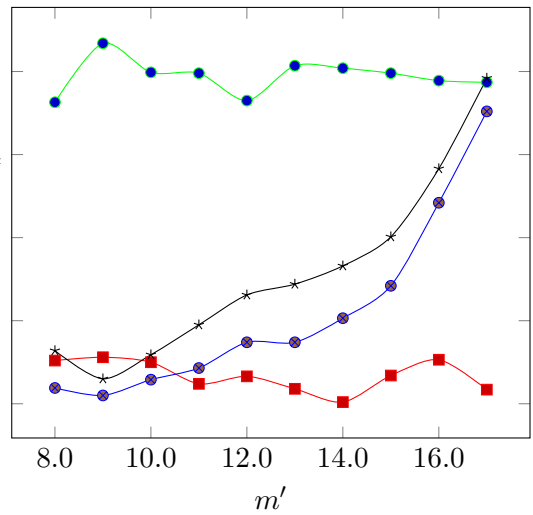
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4