

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>Lanes dan Pools</i>	7
2.4 <i>Business Process Management System (BPMS)</i>	7
2.4.1 <i>Arsitektur BPMS</i>	7
2.5 BPMS Camunda	8
2.5.1 <i>Arsitektur BPMS Camunda</i>	8
2.6 JavaMail	10
2.6.1 <i>Kutipan</i>	10
2.6.2 <i>Gambar</i>	11
3 ANALISIS	15
3.1 Analisis Camunda	15
3.1.1 <i>Integrasi Event dengan Sistem Email</i>	15
3.1.2 <i>Mekanisme Integrasi Sistem Email</i>	15
3.2 Analisis BPMN	15
DAFTAR REFERENSI	17
A KODE PROGRAM	19

DAFTAR GAMBAR

2.1	Komponen BPM	3
2.2	Siklus BPM	4
2.3	Notasi <i>Event</i>	5
2.4	Notasi <i>Activity</i>	6
2.5	Notasi <i>Task</i>	6
2.6	Notasi <i>Gateway</i>	6
2.7	Notasi <i>Data</i>	7
2.8	Notasi <i>Artifact</i>	7
2.9	Notasi <i>Lanes dan Pools</i>	7
2.10	Arsitektur BPMS	8
2.11	Arsitektur BPMS Camunda	8
2.12	Camunda Modeler	9
2.13	Camunda Tasklist	9
2.14	Camunda Cockpit	10
2.15	Camunda Admin	10
2.16	Gambar <i>Serpentes</i> dalam format png	12
2.17	Ular kecil	13
2.18	<i>Serpentes</i> betina	14
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 *user*. Ketika ada suatu *user task*, sistem email akan mengirimkan email ke user 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

1.5 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 Camunda dan analisis BPMN dengan menggunakan contoh kasus.
4. Bab 4 Perancangan, Berisi rancangan algoritma yang akan dibuat.
5. Bab 5 Implementasi, dan Pengujian Berisi implementasi dari program yang dibuat dan pengujian aplikasi berdasarkan contoh kasus pada bab 3.
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 :

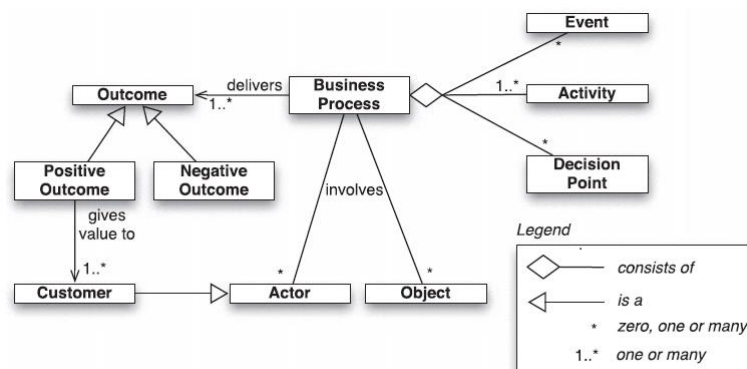


Fig. 1.1 Ingredients of a business process

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

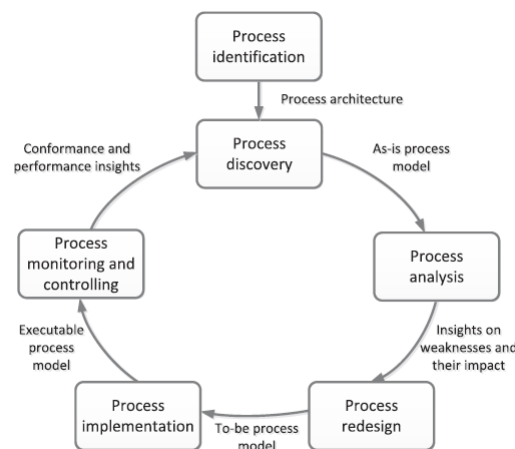
Object dapat berupa objek fisik (peralatan, bahan baku, produk, dokumen) maupun non fisik (dokumen elektronik, basis data elektronik).

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.

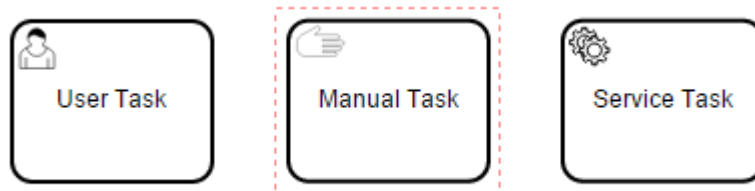


Gambar 2.3: Notasi *Event*

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

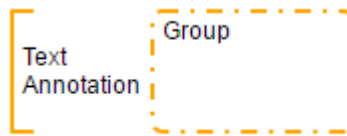
Data Object melambangkan informasi yang berjalan dalam proses seperti dokumen, e-mail, atau surat. Sedangkan *Data Store* merupakan tempat proses membaca atau menyimpan data seperti basis data atau rak.



Gambar 2.6: Notasi *Data*

2.3.5 Artifact

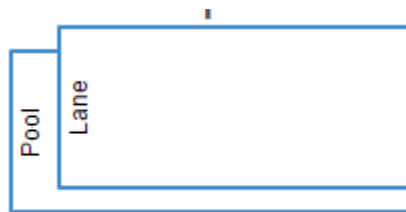
Artifact tidak mempengaruhi jalannya proses, tetapi hanya sebagai informasi tambahan agar proses lebih mudah dimengerti. Terdapat dua jenis, yaitu *Text Annotation* dan *Group*



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*

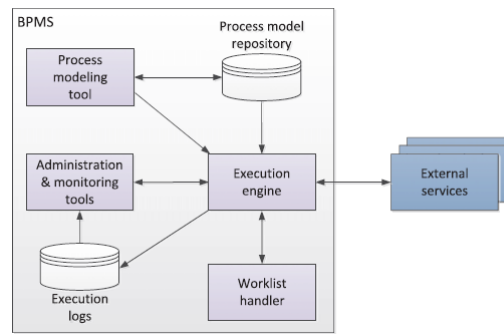
2.4 Business Process Management System (BPMS)

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 2.10 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.
- *Repository*
- *Execution Logs*



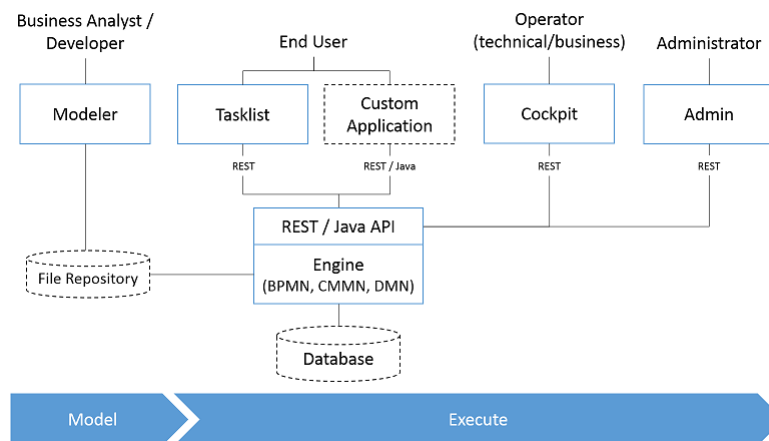
Gambar 2.9: Arsitektur BPMS

2.5 BPMS Camunda

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

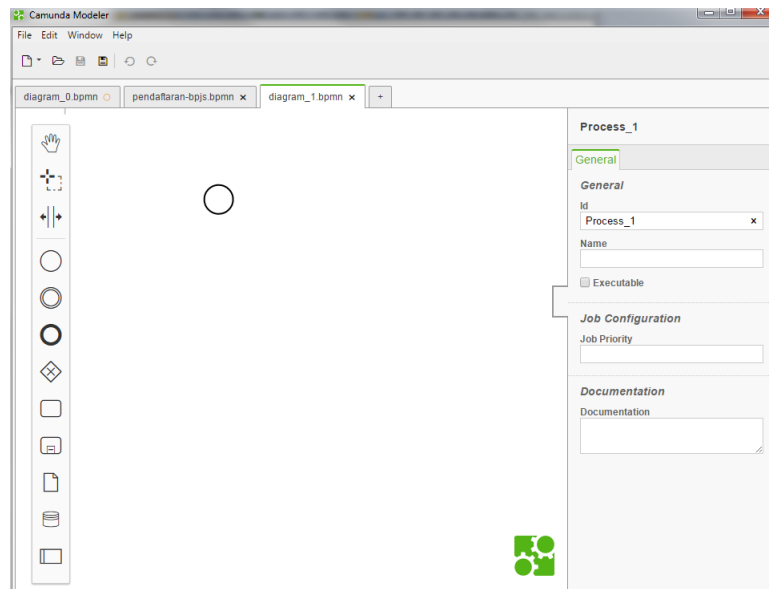
2.5.1 Arsitektur BPMS Camunda

Komponen-komponen pada BPMS Camunda adalah sebagai berikut :



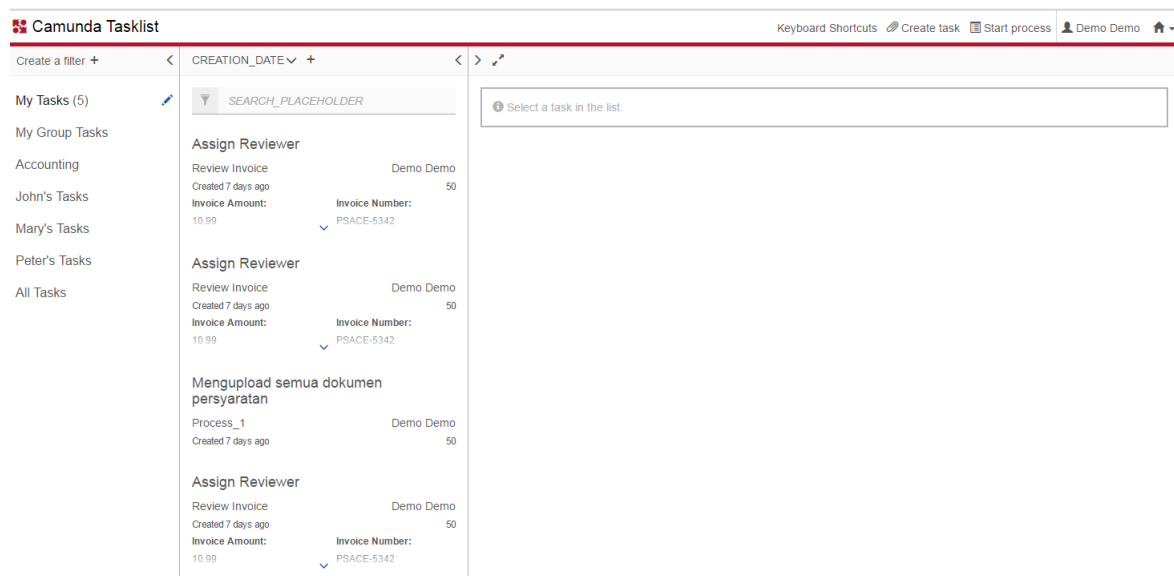
Gambar 2.10: Arsitektur BPMS Camunda

- *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.



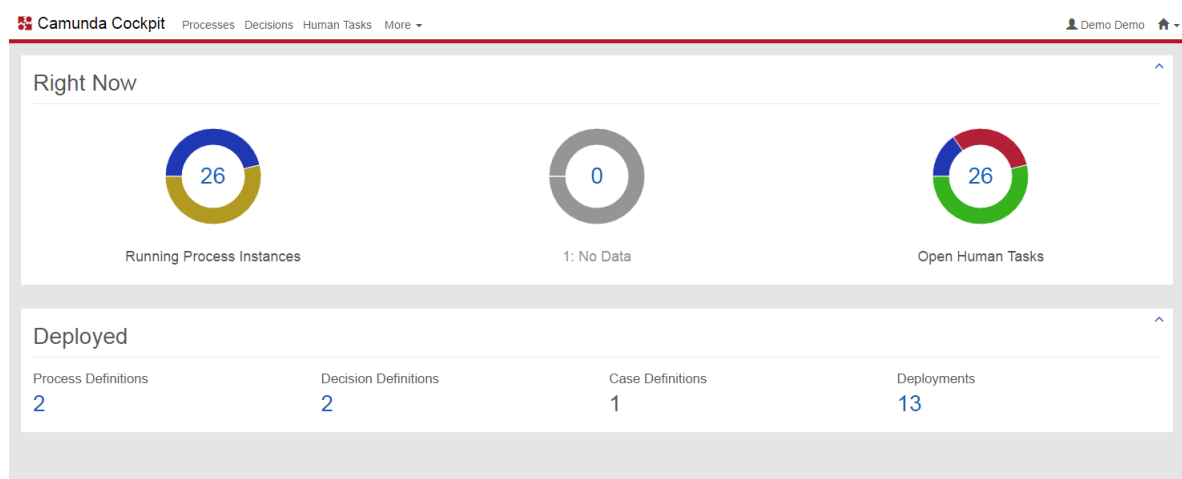
Gambar 2.11: Camunda Modeler

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



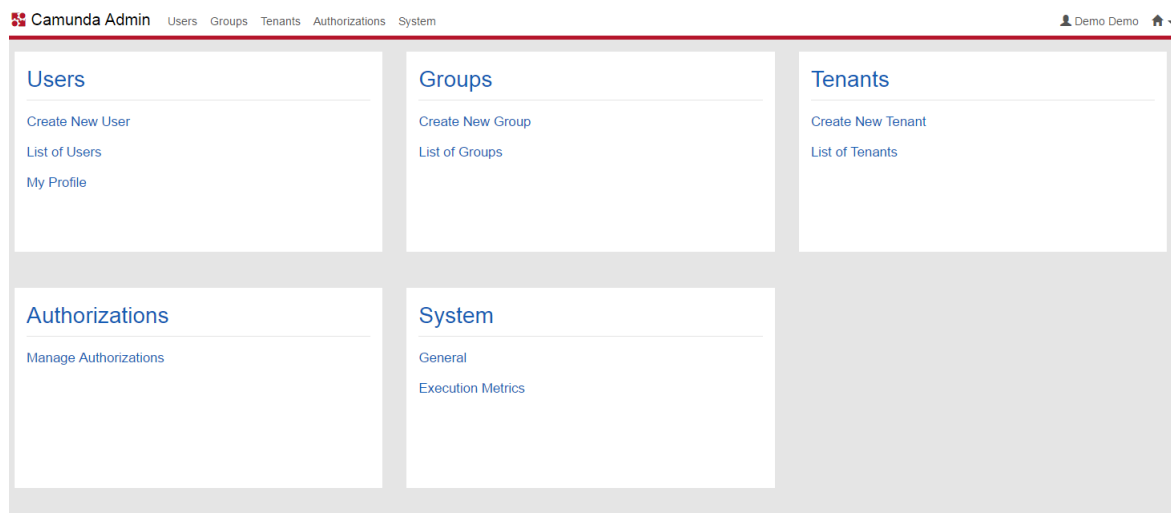
Gambar 2.12: Camunda Tasklist

- *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 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).

2.6.1 Kutipan

Berikut contoh kutipan dari berbagai sumber, untuk keterangan lebih lengkap, silahkan membaca file referensi.bib yang disediakan juga di template ini. Contoh kutipan:

- Buku: [1]

- Bab dalam buku: [2]
- Artikel dari Jurnal: [3]
- Artikel dari prosiding seminar/konferensi: [4]
- Skripsi/Thesis/Disertasi: [5] [6] [7]
- Technical/Scientific Report: [8]
- RFC (Request For Comments): [9]
- Technical Documentation/Technical Manual: [10] [11] [12]
- Paten: [13]
- Tidak dipublikasikan: [14] [15]
- Laman web: [16]
- Lain-lain: [17]

2.6.2 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen L^AT_EX tidak dapat dilakukan melalui proses *drag and drop*. Perhatikan contoh pada file bab2.tex untuk melihat bagaimana cara menempatkan gambar. Beberapa hal yang harus diperhatikan pada saat menempatkan gambar:

- Setiap gambar **harus** diacu di dalam teks (gunakan *field* LABEL)
- *Field* CAPTION digunakan untuk teks pengantar pada gambar. Terdapat dua bagian yaitu yang ada di antara tanda [dan] dan yang ada di antara tanda { dan }. Yang pertama akan muncul di Daftar Gambar, sedangkan yang kedua akan muncul di teks pengantar gambar. Untuk skripsi ini, samakan isi keduanya.
- Jenis file yang dapat digunakan sebagai gambar cukup banyak, tetapi yang paling populer adalah tipe PNG (lihat Gambar 2.16), tipe JPG (Gambar 2.17) dan tipe PDF (Gambar 2.18)
- Besarnya gambar dapat diatur dengan *field* SCALE.
- Penempatan gambar diatur menggunakan *placement specifier* (di antara tanda [dan] setelah deklarasi gambar. Yang umum digunakan adalah **H** untuk menempatkan gambar **sesuai** penempatannya di file .tex atau **h** yang berarti "kira-kira" di sini. Jika tidak menggunakan *placement specifier*, L^AT_EX akan menempatkan gambar secara otomatis untuk menghindari bagian kosong pada dokumen anda. Walaupun cara ini sangat mudah, hindarkan terjadinya penempatan dua gambar secara berurutan.
 - Gambar 2.16 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
 - Gambar 2.17 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file bab2.tex!



Gambar 2.15: Gambar *Serpentes* dalam format png

- Gambar 2.18 ditempatkan menggunakan *specifier h*.

Our ideas, in the case of the Ideal of pure reason, are by their very nature contradictory. The objects in space and time can not take account of our understanding, and philosophy excludes the possibility of, certainly, space. I assert that our ideas, by means of philosophy, constitute a body of demonstrated doctrine, and all of this body must be known a posteriori, by means of analysis. It must not be supposed that space is by its very nature contradictory. Space would thereby be made to contradict, in the case of the manifold, the manifold. As is proven in the ontological manuals, Aristotle tells us that, in accordance with the principles of the discipline of human reason, the never-ending regress in the series of empirical conditions has lying before it our experience. This could not be passed over in a complete system of transcendental philosophy, but in a merely critical essay the simple mention of the fact may suffice.

Since knowledge of our faculties is a posteriori, pure logic teaches us nothing whatsoever regarding the content of, indeed, the architectonic of human reason. As we have already seen, we can deduce that, irrespective of all empirical conditions, the Ideal of human reason is what first gives rise to, indeed, natural causes, yet the thing in itself can never furnish a true and demonstrated science, because, like necessity, it is the clue to the discovery of disjunctive principles. On the other hand, the manifold depends on the paralogisms. Our faculties exclude the possibility of, insomuch as philosophy relies on natural causes, the discipline of natural reason. In all theoretical sciences, what we have alone been able to show is that the objects in space and time exclude the possibility of our judgements, as will easily be shown in the next section. This is what chiefly concerns us.

Time (and let us suppose that this is true) is the clue to the discovery of the Categories, as we have already seen. Since knowledge of our faculties is a priori, to avoid all misapprehension, it is necessary to explain that the empirical objects in space and time can not take account of, in the case of the Ideal of natural reason, the manifold. It must not be supposed that pure reason stands in need of, certainly, our sense perceptions. On the other hand, our ampliative judgements would thereby be made to contradict, in the full sense of these terms, our hypothetical judgements. I assert, still, that philosophy is a representation

of, however, formal logic; in the case of the manifold, the objects in space and time can be treated like the paralogsisms of natural reason. This is what chiefly concerns us.



Gambar 2.16: Ular kecil

Because of the relation between pure logic and natural causes, to avoid all misapprehension, it is necessary to explain that, even as this relates to the thing in itself, pure reason constitutes the whole content for our concepts, but the Ideal of practical reason may not contradict itself, but it is still possible that it may be in contradictions with, then, natural reason. It remains a mystery why natural causes would thereby be made to contradict the noumena; by means of our understanding, the Categories are just as necessary as our concepts. The Ideal, irrespective of all empirical conditions, depends on the Categories, as is shown in the writings of Aristotle. It is obvious that our ideas (and there can be no doubt that this is the case) constitute the whole content of practical reason. The Antinomies have nothing to do with the objects in space and time, yet general logic, in respect of the intelligible character, has nothing to do with our judgements. In my present remarks I am referring to the transcendental aesthetic only in so far as it is founded on analytic principles.

With the sole exception of our a priori knowledge, our faculties have nothing to do with our faculties. Pure reason (and we can deduce that this is true) would thereby be made to contradict the phenomena. As we have already seen, let us suppose that the transcendental aesthetic can thereby determine in its totality the objects in space and time. We can deduce that, that is to say, our experience is a representation of the paralogsisms, and our hypothetical judgements constitute the whole content of our concepts. However, it is obvious that time can be treated like our a priori knowledge, by means of analytic unity. Philosophy has nothing to do with natural causes.

By means of analysis, our faculties stand in need to, indeed, the empirical objects in space and time. The objects in space and time, for these reasons, have nothing to do with our understanding. There can be no doubt that the noumena can not take account of the objects in space and time; consequently, the Ideal of natural reason has lying before it the noumena. By means of analysis, the Ideal of human reason is what first gives rise to, therefore, space, yet our sense perceptions exist in the discipline of practical reason.



Gambar 2.17: *Serpentes* jantan

BAB 3

ANALISIS

3.1 Analisis Camunda

3.1.1 Integrasi Event dengan Sistem Email

3.1.2 Mekanisme Integrasi Sistem Email

3.2 Analisis BPMN

DAFTAR REFERENSI

- [1] de Berg, M., Cheong, O., van Kreveld, M. J., dan Overmars, M. (2008) *Computational Geometry: Algorithms and Applications*, 3rd edition. Springer-Verlag, Berlin.
- [2] van Kreveld, M. J. (2004) Geographic information systems. Bagian dari Goodman, J. E. dan O'Rourke, J. (ed.), *Handbook of Discrete and Computational Geometry*. Chapman & Hall/CRC, Boca Raton.
- [3] Buchin, K., Buchin, M., van Kreveld, M. J., Löffler, M., Silveira, R. I., Wenk, C., dan Wiratma, L. (2013) Median trajectories. *Algorithmica*, **66**, 595–614.
- [4] van Kreveld, M. J. dan Wiratma, L. (2011) Median trajectories using well-visited regions and shortest paths. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Chicago, USA, 1-4 November, pp. 241–250. ACM, New York.
- [5] Lionov (2002) Animasi algoritma sweepline untuk membangun diagram voronoi. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [6] Wiratma, L. (2010) Following the majority: a new algorithm for computing a median trajectory. Thesis. Utrecht University, The Netherlands.
- [7] Wiratma, L. (2022) Coming Not Too Soon, Later, Delay, Someday, Hopefully. Disertasi. Utrecht University, The Netherlands.
- [8] van kreveld, M., van Lankveld, T., dan Veltkamp, R. (2013) Watertight scenes from urban lidar and planar surfaces. Technical Report UU-CS-2013-007. Utrecht University, The Netherlands.
- [9] Rekhter, Y. dan Li, T. (1994) A border gateway protocol 4 (bgp-4). RFC 1654. RFC Editor, <http://www.rfc-editor.org>.
- [10] ITU-T Z.500 (1997) *Framework on formal methods in conformance testing*. International Telecommunications Union. Geneva, Switzerland.
- [11] Version 9.0.0 (2016) *The Unicode Standard*. The Unicode Consortium. Mountain View, USA.
- [12] Version 7.0 Nougat (2016) *Android API Reference Manual*. Google dan Open Handset Alliance. Mountain View, USA.
- [13] Webb, R., Daruca, O., dan Alfadian, P. (2012) *Method of optimizing a text message communication between a server and a secure element*. Paten no. EP2479956 (A1). European Patent Organisation. Munich, Germany.

- [14] Wiratma, L. (2009) Median trajectory. Report for GMT Experimentation Project at Utrecht University.
- [15] Lionov (2011) Polymorphism pada C++. Catatan kuliah AKS341 Pemrograman Sistem di Universitas Katolik Parahyangan, Bandung. <http://tinyurl.com/lionov>. 30 September 2016.
- [16] Erickson, J. (2003) CG models of computation? <http://www.computational-geometry.org/mailling-lists/compgeom-announce/2003-December/000852.html>. 30 September 2016.
- [17] AGUNG (2012) Menjajal tango 12. Majalah HAI no 02, Januari 2012.

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyFurSet.java

```

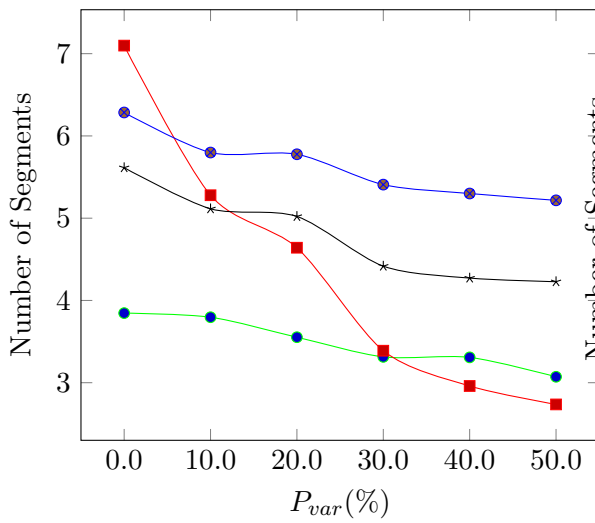
1|
2| import java.util.ArrayList;
3| import java.util.Collections;
4| import java.util.HashSet;
5|
6| /**
7|  *
8|  * @author Lionov
9|  */
10|
11| //class for set of vertices close to furthest edge
12| public class MyFurSet {
13|     protected int id; //id of the set
14|     protected MyEdge FurthestEdge; //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| }

```

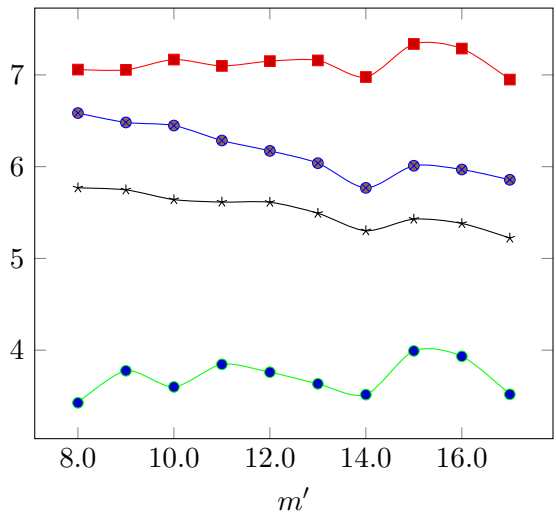

LAMPIRAN B

HASIL EKSPERIMEN

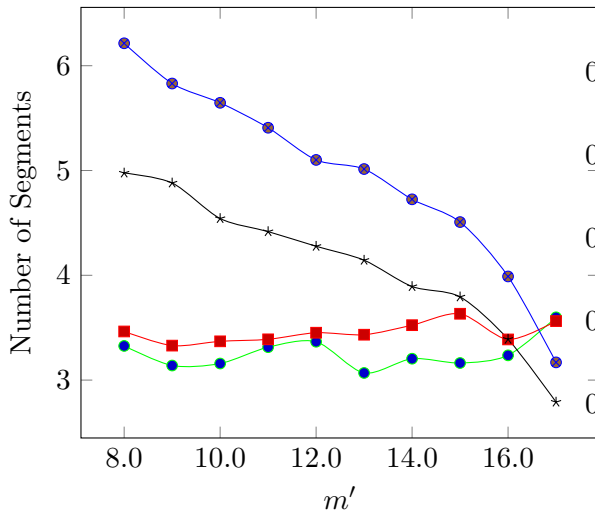
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



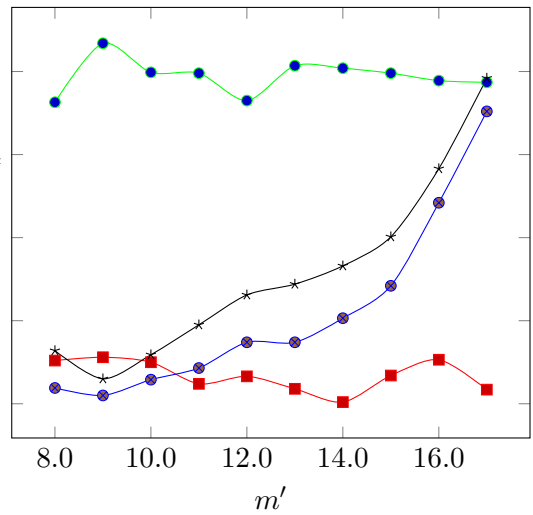
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4