

***BUSINESS PROCESS MANAGEMENT SYSTEM* UNTUK SISTEM  
INFORMASI PENGINAPAN PADA SYARIAHROOMS  
HOSPITALITY**

**SKRIPSI**

**Oleh :  
ABDUSH SHOMAD BUKHORI  
NIM. 15650054**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

***BUSINESS PROCESS MANAGEMENT SYSTEM* UNTUK SISTEM  
INFORMASI PENGINAPAN PADA SYARIAHROOMS  
HOSPITALITY**

**SKRIPSI**

**Diajukan kepada:  
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:  
ABDUSH SHOMAD BUKHORI  
NIM. 15650054**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**LEMBAR PERSETUJUAN**

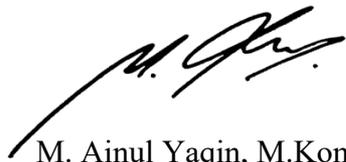
***BUSINESS PROCESS MANAGEMENT SYSTEM* UNTUK SISTEM  
INFORMASI PENGINAPAN PADA SYARIAHROOMS  
HOSPITALITY**

**SKRIPSI**

**Oleh :  
ABDUSH SHOMAD BUKHORI  
NIM. 15650054**

Telah Diperiksa dan Disetujui untuk Diuji  
Pada Tanggal : 6 Agustus 2021

Dosen Pembimbing I



M. Ainul Yaqin, M.Kom  
NIP. 19761013 200604 1 004

Dosen Pembimbing II



Syahiduz Zaman, M.Kom  
NIP. 19710722 201101 1 001

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Fachrul Kurniawan ST., M.MT., IPM  
NIP. 19771020 200912 1 001

**LEMBAR PENGESAHAN**  
***BUSINESS PROCESS MANAGEMENT SYSTEM* UNTUK**  
**SISTEM INFORMASI PENGINAPAN PADA**  
**SYARIAH ROOMS HOSPITALITY**

**SKRIPSI**

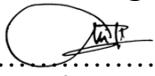
**Oleh :**  
**ABDUSH SHOMAD BUKHORI**  
**NIM. 15650054**

Telah Dipertahankan di Depan Dewan Penguji  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Pada Tanggal 29 Oktober 2021

**Susunan Dewan Penguji**

1. Penguji Utama : Ajib Hanani, M.T  
NIDT. 19840731 201608011 076
2. Ketua Penguji : Fatchurrahman, M.Kom  
NIP. 19700731 200501 1 002
3. Sekretaris Penguji : M. Ainul Yaqin, M.Kom  
NIP. 19761013 200604 1 004
4. Anggota Penguji : Syahiduz Zaman, M.Kom  
NIP. 19700502 200501 1 005

**Tanda Tangan**

(.....)  
(.....)  
(.....)  
(.....)

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Fachrul Kurniawan ST., M.MT., IPM  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Abdush Shomad Bukhori

NIM : 15650054

Fakultas/Jurusan : Sains dan Teknologi/Teknik Infomatika

Judul Skripsi : *Business Process Management System* Untuk Sistem Informasi Penginapan Pada Syariahrooms Hospitality

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar hasil karya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar Pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 1 November 2021  
Yang membuat pernyataan,



Abdush Shomad Bukhori  
NIM. 15650054

**HALAMAN MOTTO**

***“Terbentuklah dari pilihanmu, jangan hanya  
terbentuk dari keadaan.”***

## HALAMAN PERSEMBAHAN

### بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*Dengan menyebut nama Allah Yang Maha Pengasih dan Maha Penyayang.  
Alhamdulillah... puji syukur kehadiran Allah SWT. Atas karunia, hidayah,  
petunjuk dan kemudahan yang Engkau berikan dalam menyelesaikan skripsi ini.  
Sholawat dan salam selalu terlimpahkan keharibaan Rasulullah Muhammad  
SAW.*

#### **Penulis persembahkan karya ini kepada:**

Kedua orang tua penulis tercinta, Bapak Eko Wahono dan Ibu Nur Zulfah yang selalu memberikan semangat dan motivasi dalam menyelesaikan kuliah S1 di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Dosen pembimbing penulis, Bapak M. Ainul Yaqin, M.Kom dan Bapak Syahiduz Zaman, M.Kom yang telah dengan sabar membimbing serta membantu kelancaran penulis dalam menyelesaikan proses penulisan skripsi.

Dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang yang telah mendidik dan memberikan ilmu yang bermanfaat kepada penulis.

Seluruh tim Alfath Tech yang telah memberikan semangat, bantuan dan doa kepada penulis untuk segera menyelesaikan skripsi.

Keluarga besar Teknik Informatika Angkatan 2015 yang telah memberikan semangat dan bantuan dalam proses penulisan.

Semua orang yang penulis sayangi, yang tak bisa penulis sebutkan satu per satu yang selalu memberikan dukungan kepada penulis untuk menyelesaikan pendidikan S1.

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji dan syukur penulis panjatkan ke hadirat Allah subhanahu wa ta'ala yang telah melimpahkan rahmat dan hidayahNya, sehingga penulis bisa menyelesaikan skripsi dengan tepat waktu yang berjudul “*Business Process Management System Untuk Sistem Informasi Penginapan Pada Syariahrooms Hospitality*”. Tujuan dari penyusunan skripsi ini guna memenuhi salah satu syarat untuk bisa menempuh ujian sarjana komputer pada Fakultas Sains dan Teknologi (FSAINTEK) Program Studi Teknik Informatika di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Didalam pengerjaan skripsi ini telah melibatkan banyak pihak yang sangat membantu dalam banyak hal. Oleh sebab itu, disini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada:

1. Prof. Dr. H. M. Zainuddin, M.A. selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan ST., M.MT., IPM, Selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. M. Ainul Yaqin, M.Kom, selaku Dosen Pembimbing I yang telah membimbing dalam penyusunan skripsi ini hingga selesai.
5. Syahiduz Zaman, M.Kom, selaku Dosen Pembimbing II yang telah membimbing dalam penyusunan skripsi ini hingga selesai.

6. Ririen Kusumawati, M.Kom, Selaku Dosen Wali yang senantiasa memberikan banyak motivasi dan saran untuk kebaikan penulis.
7. Tim Syariahrooms.com yang telah bersedia memberikan data.
8. Orang tua tercinta yang telah banyak memberikan doa dan dukungan kepada penulis secara moril maupun materil hingga skripsi ini dapat terselesaikan.
9. Teman-teman interface yang selalu memberikan semangat dan doa kepada penulis.
10. Semua pihak yang telah banyak membantu dalam penyusunan skripsi ini yang tidak bisa penulis sebutkan semuanya.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Malang, 1 November 2021

Penulis

## DAFTAR ISI

|   |      |
|---|------|
| LEMBAR PERSETUJUAN .....  | ii   |
| LEMBAR PENGESAHAN .....   | iii  |
| PERNYATAAN KEASLIAN TULISAN .....                                 | iv   |
| HALAMAN MOTTO .....   | v    |
| HALAMAN PERSEMBAHAN.....  | vi   |
| KATA PENGANTAR.....   | vii  |
| DAFTAR ISI.....   | ix   |
| DAFTAR GAMBAR.....  | xi   |
| DAFTAR TABEL.....   | xiv  |
| ABSTRAK .....   | xv   |
| ABSTRACT .....  | xvi  |
| المخلص .....  | xvii |
| BAB I PENDAHULUAN .....   | 1    |
| 1.1.    Latar Belakang .....                                      | 1    |
| 1.2.    Rumusan Masalah.....                                      | 6    |
| 1.3.    Tujuan Penelitian .....                                   | 6    |
| 1.4.    Manfaat Penelitian .....                                  | 6    |
| 1.5.    Batasan Masalah .....                                     | 6    |
| 1.6.    Sistematika Penulisan .....                               | 7    |
| BAB II KAJIAN PUSTAKA .....                                       | 8    |
| 2.1. <i>Business Process Management (BPM)</i> .....               | 8    |
| 2.2. <i>Business Process Model Notation (BPMN)</i> .....          | 10   |
| 2.3. <i>Service Oriented Architecture (SOA)</i> .....             | 12   |
| 2.4. <i>Business Process Execution Language (BPEL)</i> .....      | 17   |
| 2.5. <i>Business Process Management System (BPM System)</i> ..... | 20   |
| 2.6. <i>McCall's Software Quality Framework</i> .....             | 22   |
| 2.7. <i>Black Box Testing</i> .....                               | 24   |
| 2.8. <i>Advanced Process Model Repository (APROMORE)</i> .....    | 25   |
| 2.9. <i>Workflow Analyzer (Woflan)</i> .....                      | 26   |
| 2.10.   Produk Terkait .....                                      | 26   |
| BAB III METODE PENELITIAN .....                                   | 32   |
| 3.1.    Survei Literatur.....                                     | 32   |
| 3.2.    Merancang <i>BPM System</i> .....                         | 32   |
| 3.2.1.  Arsitektur Data.....                                      | 34   |

|   |                                     |           |
|---|-------------------------------------|-----------|
| 3.2.2.                                  | Arsitektur Aplikasi.....            | 40        |
| 3.4.                                    | Uji Coba Sistem .....               | 48        |
| <b>BAB IV HASIL DAN PEMBAHASAN.....</b> |                                     | <b>52</b> |
| 4.1.                                    | Langkah Pengujian .....             | 52        |
| 4.1.1.                                  | Menentukan Modul.....               | 52        |
| 4.1.2.                                  | Skenario Pengujian .....            | 53        |
| 4.2.                                    | Hasil Pengujian.....                | 54        |
| 4.3.                                    | Pembahasan dan Integrasi Islam..... | 86        |
| <b>BAB V KESIMPULAN DAN SARAN .....</b> |                                     | <b>88</b> |
| 5.1                                     | Kesimpulan .....                    | 88        |
| 5.2                                     | Saran.....                          | 88        |
| <b>DAFTAR PUSTAKA .....</b>             |                                     | <b>89</b> |

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2. 1 Keterhubungan Antar Komponen <i>Web Service</i> .....                           | 13 |
| Gambar 2. 2 Enkapsulasi <i>business process</i> dengan service.....                         | 14 |
| Gambar 2. 3 Enkapsulasi <i>business process</i> dengan service.....                         | 16 |
| Gambar 2. 4 Struktur BPEL .....   | 18 |
| Gambar 2. 5 URI dalam BPEL .....  | 19 |
| Gambar 2. 6 Contoh Kode BPEL.....   | 19 |
| Gambar 2. 7 McCall's Software Quality Framework .....                                       | 23 |
| Gambar 2. 8 <i>Three-Layered SOA in APROMORE</i> .....                                      | 25 |
| Gambar 3. 1 Arsitektur BPMS skala besar .....   | 33 |
| Gambar 3. 2 <i>Entity Relational Diagram BPM System</i> .....                               | 35 |
| Gambar 3. 3 <i>Flowchart web service management</i> .....                                   | 41 |
| Gambar 3. 4 <i>Flowchart tenant management</i> .....  | 42 |
| Gambar 3. 5 <i>Flowchart user management</i> .....  | 43 |
| Gambar 3. 6 <i>Flowchart entry profil perusahaan</i> .....                                  | 44 |
| Gambar 3. 7 <i>Flowchart entry profil admin</i> .....                                       | 45 |
| Gambar 3. 8 <i>Flowchart application management</i> .....                                   | 46 |
| Gambar 3. 9 <i>Flowchart role management</i> .....  | 47 |
| Gambar 3. 10 BPMN Syariahrooms Alur <i>Booking 1</i> .....                                  | 50 |
| Gambar 3. 11 BPMN Syariahrooms Alur <i>Booking 2</i> .....                                  | 50 |
| Gambar 3. 12 BPMN Syariahrooms Alur <i>Booking 3</i> .....                                  | 51 |
| Gambar 3. 13 BPMN Syariahrooms Alur <i>Booking 4</i> .....                                  | 51 |
| Gambar 4. 1 Perbedaan alur pemesanan .....  | 54 |
| Gambar 4. 2 Halaman <i>Dashboard Super Admin</i> .....                                      | 58 |
| Gambar 4. 3 Halaman <i>Dashboard Admin tenant</i> .....                                     | 59 |
| Gambar 4. 4 <i>Alert</i> ketika data <i>invalid</i> .....                                   | 59 |
| Gambar 4. 5 <i>Warning</i> Pada <i>Form Login</i> .....                                     | 59 |
| Gambar 4. 6 Halaman <i>List Web Service</i> .....   | 60 |
| Gambar 4. 7 <i>Warning</i> Pada <i>Form Tambah Web Service</i> .....                        | 60 |
| Gambar 4. 8 <i>Alert</i> Ketika Berhasil Menerima Pendaftaran <i>Tenant</i> .....           | 61 |
| Gambar 4. 9 <i>Alert</i> Ketika Gagal Menerima Pendaftaran <i>Tenant</i> .....              | 61 |
| Gambar 4. 10 <i>Alert</i> Ketika Data Tidak Lengkap Menerima Pendaftaran <i>Tenant</i> . 61 |    |

|   |    |
|---|----|
| Gambar 4. 11 <i>Alert</i> bahwa pendaftaran berhasil.....                     | 61 |
| Gambar 4. 12 Halaman <i>List Tenant</i> .....                                 | 62 |
| Gambar 4. 13 <i>Alert</i> bahwa pendaftaran gagal.....                        | 62 |
| Gambar 4. 14 <i>Warning</i> Pada <i>Form</i> Pendaftaran <i>Tenant</i> .....  | 62 |
| Gambar 4. 15 Halaman <i>List Application</i> .....                            | 63 |
| Gambar 4. 16 <i>Alert</i> bahwa pendaftaran gagal.....                        | 63 |
| Gambar 4. 17 <i>Warning</i> Pada <i>Form</i> Tambah Aplikasi.....             | 63 |
| Gambar 4. 18 Halaman <i>List Role User</i> .....                              | 64 |
| Gambar 4. 19 <i>Warning</i> Pada <i>Form</i> Tambah <i>Role User</i> .....    | 64 |
| Gambar 4. 20 Halaman <i>List User</i> pada Aplikasi.....                      | 65 |
| Gambar 4. 21 <i>Warning</i> Pada <i>Form</i> Tambah <i>User</i> Aplikasi..... | 65 |
| Gambar 4. 24 <i>Warning</i> Pada <i>Form</i> Upload XPDL.....                 | 66 |
| Gambar 4. 22 Halaman <i>Setup Role User</i> dari Rekomendasi Sistem.....      | 66 |
| Gambar 4. 23 <i>Alert</i> bahwa <i>upload</i> XPDL gagal .....                | 66 |
| Gambar 4. 25 Halaman Detail Aplikasi .....                                    | 67 |
| Gambar 4. 26 <i>Warning</i> Pada <i>Form</i> <i>Mapping Web Service</i> ..... | 67 |
| Gambar 4. 27 <i>View List User</i> .....                                      | 68 |
| Gambar 4. 28 <i>Add User</i> .....  | 69 |
| Gambar 4. 29 <i>Delete User</i> .....   | 69 |
| Gambar 4. 30 <i>Edit User</i> .....   | 70 |
| Gambar 4. 31 <i>View List Property</i> .....                                  | 70 |
| Gambar 4. 32 <i>Add Property</i> .....  | 71 |
| Gambar 4. 33 <i>Delete Property</i> .....                                     | 71 |
| Gambar 4. 34 <i>Edit Property</i> .....                                       | 72 |
| Gambar 4. 35 <i>View Finance Report</i> .....                                 | 72 |
| Gambar 4. 36 <i>View List Booking</i> .....                                   | 73 |
| Gambar 4. 37 <i>Cancel Booking</i> .....                                      | 73 |
| Gambar 4. 38 <i>Reschedule Booking</i> .....                                  | 74 |
| Gambar 4. 39 <i>View List Payment</i> .....                                   | 74 |
| Gambar 4. 40 <i>Verify Payment</i> .....                                      | 75 |
| Gambar 4. 41 Alur Proses Pemesanan <i>Save Booking</i> .....                  | 76 |
| Gambar 4. 42 Alur Proses Pemesanan Berdasarkan BPMN 1.....                    | 76 |

|  |    |
|--|----|
| Gambar 4. 43 <i>Add Booking</i> BPMN 1 .....               | 76 |
| Gambar 4. 44 <i>Select Channel</i> BPMN 1 .....            | 77 |
| Gambar 4. 45 <i>Fill Data Guest</i> BPMN 1 .....           | 77 |
| Gambar 4. 46 <i>Fill Data Booking</i> BPMN 1 .....         | 78 |
| Gambar 4. 47 Alur Proses Pemesanan Berdasarkan BPMN 2..... | 78 |
| Gambar 4. 48 <i>Add Booking</i> BPMN 2.....                | 79 |
| Gambar 4. 49 <i>Fill Data Booking</i> BPMN 2 .....         | 79 |
| Gambar 4. 50 <i>Fill Data Guest</i> BPMN 2 .....           | 80 |
| Gambar 4. 51 <i>Select Channel</i> BPMN 2 .....            | 80 |
| Gambar 4. 52 Alur Proses Pemesanan Berdasarkan BPMN 3..... | 81 |
| Gambar 4. 53 <i>Add Booking</i> BPMN 3 .....               | 81 |
| Gambar 4. 54 <i>Fill Data Guest</i> BPMN 3 .....           | 82 |
| Gambar 4. 55 <i>Fill Data Booking</i> BPMN 3 .....         | 82 |
| Gambar 4. 56 <i>Select Channel</i> BPMN 3 .....            | 83 |
| Gambar 4. 57 Alur Proses Pemesanan Berdasarkan BPMN 4..... | 83 |
| Gambar 4. 58 <i>Add Booking</i> BPMN 4.....                | 84 |
| Gambar 4. 59 <i>Fill Data Guest</i> BPMN 4 .....           | 84 |
| Gambar 4. 60 <i>Select Channel</i> BPMN 4 .....            | 85 |
| Gambar 4. 61 <i>Fill Data Booking</i> BPMN 4 .....         | 85 |

## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 2. 1 Perbandingan <i>BPM System</i> Bonita, Camunda dan Bizagi ..... | 28 |
| Tabel 3. 1 Struktur tabel user .....                                       | 36 |
| Tabel 3. 2 Struktur tabel tenant .....                                     | 36 |
| Tabel 3. 3 Struktur tabel web_service.....                                 | 37 |
| Tabel 3. 4 Struktur tabel app .....  | 37 |
| Tabel 3. 5 Struktur tabel version .....                                    | 38 |
| Tabel 3. 6 Struktur tabel activity.....                                    | 39 |
| Tabel 3. 7 Struktur tabel user (pada <i>tenant</i> ).....                  | 39 |
| Tabel 3. 8 Struktur tabel role.....  | 40 |
| Tabel 4. 1 Tabel Modul <i>BPM System</i> .....                             | 52 |
| Tabel 4. 2 Tabel Hasil Pengujian Modul <i>BPM System</i> .....             | 55 |
| Tabel 4. 3 Tabel Hasil <i>Running Application</i> .....                    | 68 |

## ABSTRAK

Bukhori, Abdush Shomad. 2021. *Business Process Management System Untuk Sistem Informasi Penginapan Pada Syariahrooms Hospitality*. Skripsi. Jurusan Teknik Informatika. Fakultas Sains Dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) M. Ainul Yaqin, M.Kom, (II) Syahiduz Zaman, M.Kom

---

**Kata Kunci:** Teknologi Informasi, Sistem Informasi, BPM, BPMN, *BPM System*.

Proses bisnis merupakan hal yang tidak bisa dipisahkan dengan dunia bisnis. Proses bisnis dapat menjadi sebuah pedoman dalam menjalankan sebuah bisnis agar sesuai dengan visi dan misi perusahaan. Salah satu masalah yang sering terjadi adalah pengelolaan data yang masih manual yang menyebabkan tingkat duplikasi data cukup tinggi dan ketidakrapian dalam pengelolaan tempat dokumen perusahaan yang menyebabkan penumpukan kertas-kertas dokumen perusahaan. Pada akhirnya perusahaan dituntut untuk menggunakan sistem informasi guna mengelola data tersebut dengan baik. Namun, permasalahan muncul diantaranya adalah sistem informasi yang belum sesuai dengan kebutuhan perusahaan dimana terdapat lebih dari satu sistem informasi yang digunakan. Biaya dalam membuat sistem informasi tersebut juga terhitung mahal serta dalam pengembangan fitur-fiturnya yang menyesuaikan perubahan proses bisnis yang dijalankan. Dalam hal ini dibutuhkan sebuah sistem informasi yang dapat menyelesaikan permasalahan-permasalahan tersebut. Sehingga biaya bukan lagi menjadi masalah, pengembangan fitur menjadi lebih mudah dan dapat disesuaikan dengan proses bisnis yang berjalan.

Pada penelitian ini dibangunlah sebuah sistem yang dapat menyesuaikan proses bisnis yang dijalankan sebuah perusahaan yang disebut Sistem BPM. Dalam penelitian ini menggunakan pendekatan proses bisnis dalam penggunaan sistem tersebut sehingga sistem bersifat *flexible* dan *reusable*. *Flexible* dalam hal ini adalah sistem dapat dijalankan sesuai dengan proses bisnis yang diinputkan. Sedangkan *reusable* dalam hal ini adalah *web service* yang disediakan oleh sistem dapat digunakan pada proses bisnis lain. Adapun studi kasus pada penelitian ini melibatkan Syariahrooms Hospitality dan penelitian ini berhasil membangun sistem BPM yang bersifat *flexible* dan *reusable*.

## ABSTRACT

Bukhori, Abdush Shomad. 2021. **Business Process Management System for Lodging Information Systems in Syariahrooms Hospitality**. Thesis. Department of *Informatics Engineering*, Faculty of Science and Technology, Islamic State University of Maulana Malik Ibrahim of Malang. Advisers: (I) M. Ainul Yaqin, M.Kom, (II) Syahiduz Zaman, M.Kom

---

**Keyword:** *Information Technology, Information Systems, BPM, BPMN, BPM Systems.*

Business processes are an inseparable part of the business world. Business processes can be a guideline in running a business to match the company's vision and mission. One of the most common problems is manual data management which causes a high level of data duplication and untidiness in the management of the company's document space which causes the accumulation of company documents. In the end, companies are required to use information systems to manage the data properly. However, problems that arise include the information system that is not in accordance with the needs of the company where there is more than one information system used. The cost in making the information system is also quite expensive and in developing its features that adapt to changes in the business processes that are carried out. In this case we need an information system that can solve these problems. So that cost is no longer a problem, feature development becomes easier and can be adapted to current business processes.

In this research, a system is built that can adjust the business processes run by a company called the BPM System. This research uses a business process approach in using the system so that the system is flexible and reusable. Flexible in this case is the system can be run according to the inputted business processes. While reusable in this case is a web service provided by the system that can be used in other business processes. The case study in this study involved Syariahrooms Hospitality and this research succeeded in building a flexible and reusable BPM system.

## المخلص

بخاري ، عبد الصمد. ٢٠٢١ . نظام إدارة عمليات الأعمال لنظام معلومات الإقامة في ضيافة Syariahrooms. أطروحة. الهندسة المعلوماتية. كلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم الإسلامية الحكومية ، مالانج. المشرف: (1) محمد عين اليقين، ماجستير (2) شهيد الزمان، ماجستير

**الكلمات المفتاحية:** تكنولوجيا المعلومات ، نظم المعلومات ، BPM ، BPMN ، نظام BPM.

العمليات التجارية هي جزء لا يتجزأ من عالم الأعمال. يمكن أن تكون العمليات التجارية بمثابة دليل إرشادي في إدارة الأعمال لتناسب مع رؤية الشركة ورسالتها. إحدى المشكلات التي تحدث غالبًا هي الإدارة اليدوية للبيانات التي تتسبب في ارتفاع مستوى تكرار البيانات وعدم التنظيم في إدارة مساحة مستندات الشركة. مما يؤدي إلى تراكم مستندات الشركة. في النهاية ، يتعين على الشركات استخدام أنظمة المعلومات لإدارة البيانات بشكل صحيح. ومع ذلك ، تنشأ مشاكل بما في ذلك نظام المعلومات الذي لا يتوافق مع احتياجات الشركة حيث يوجد أكثر من نظام معلومات واحد مستخدم. تعد تكلفة إنشاء نظام المعلومات أيضًا باهظة الثمن وكذلك في تطوير ميزاته التي تتكيف مع التغييرات في العمليات التجارية التي يتم تنفيذها. في هذه الحالة نحتاج إلى نظام معلومات يمكنه حل هذه المشكلات. بحيث لم تعد هذه التكلفة مشكلة ، يصبح تطوير الميزات أسهل ويمكن تكييفه مع العمليات التجارية الحالية.

في هذا البحث ، تم بناء نظام يمكنه تعديل العمليات التجارية التي تديرها شركة تسمى BPM System. يستخدم هذا البحث نهج العمليات التجارية في استخدام النظام بحيث يكون النظام مرناً وقابل لإعادة الاستخدام. المرونة في هذه الحالة هي أن النظام يمكن تشغيله وفقاً لعمليات الأعمال المدخلة. بينما يمكن إعادة استخدامها في هذه الحالة ، يمكن استخدام خدمة الويب التي يوفرها النظام في عمليات الأعمال الأخرى. تضمنت دراسة الحالة في هذه الدراسة Syariahrooms Hospitality ونجح هذا البحث في بناء نظام BPM مرن وقابل لإعادة الاستخدام.

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Dalam sebuah bisnis yang dijalankan oleh suatu perusahaan tentunya memiliki banyak proses bisnis. Proses bisnis merupakan sekumpulan aktivitas-aktivitas yang dilakukan oleh seorang pelaku bisnis yang meliputi *input*, proses, dan *output* (Ramdhani, 2015). Proses bisnis dibuat agar pelaku bisnis dapat menjalankan suatu bisnis secara terstruktur. Namun, banyaknya proses bisnis memungkinkan proses bisnis tersebut tidak terdokumentasi, terukur dan terkelola dengan baik. Oleh karena itu, diperlukan pengelolaan proses bisnis yang dinamakan dengan Manajemen Proses Bisnis (BPM).

BPM merupakan sebuah pendekatan atau metode yang dapat membantu pelaku bisnis menjadi lebih baik dalam menyelesaikan bisnisnya (Ramdhani, 2015). BPM mengharuskan pelaku bisnis untuk memastikan bahwa proses bisnis yang telah dibuat berjalan dengan baik dan lancar serta dapat ditingkatkan di kemudian hari. Hal itu dikarenakan proses bisnis dapat berubah secara cepat seiring dengan kebutuhan bisnis saat itu. Adanya BPM tersebut dapat menyelaraskan antara tujuan dan kebutuhan bisnis sehingga proses bisnis tetap dalam kondisi baik. Dalam manajemen proses bisnis terdapat 5 tahap (Gricke et al., 2013), antara lain:

- Design

Jika pada tahap ini belum ada proses yang berjalan, maka perancangan proses awal dilakukan pada tahap ini (Gricke et al., 2013). Jika telah terdapat proses yang berjalan, maka akan dilakukan perancangan alternatif berdasarkan tahap diagnosis.

- Diagnosis

Tahap ini dilakukan untuk mendapatkan kekurangan dalam proses yang diteliti (Gricke et al., 2013) yang dapat dijadikan sebagai bahan untuk perancangan alternatif.

- Configuration

Tahap ini berfokus pada spesifikasi detail dari perancangan proses pada tahap design (Gricke et al., 2013).

- Execution

Tahap ini merupakan tahap dimana kegiatan sistem dijalankan dengan mengacu pada perancangan proses yang telah dibuat atau implementasi pada sistem (Sarno et al., 2017).

- Control

Tahap ini adalah tahap terakhir dengan mengawasi berjalannya sistem yang telah dibuat serta mencatat semua *informasi* yang bisa didapatkan (Gricke et al., 2013) sebagai bahan untuk tahap diagnosis.

Pada zaman digital saat ini, hampir semua pekerjaan menggunakan sebuah sistem *informasi*. Pada bidang kesehatan, pendidikan, pariwisata, dan lain-lainnya. Sistem merupakan sebuah kesatuan yang di dalamnya terdapat komponen-komponen yang terhubung dan berfungsi untuk mempermudah penyaluran *informasi*, materi dan energy (Irawan et al., 2017). Tujuan dibuatnya sebuah sistem tidak lain adalah untuk membantu meringankan beban pekerjaan serta menjadi lebih efektif dan efisien. Penelitian ini ditekankan pada pembahasan pengelolaan proses bisnis suatu perusahaan agar dapat membantu suatu perusahaan dalam menjalankan bisnisnya menjadi lebih efektif dan efisien. Karena sebagai hamba Allah harus senantiasa menolong sesama saudaranya. Hal ini sesuai dengan sabda Nabi Muhammad ﷺ yang berbunyi:

عَنْ أَبِي هُرَيْرَةَ رَضِيَ اللَّهُ عَنْهُ عَنِ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ مَنْ نَعَسَ عَنِ مُؤْمِنٍ كُرْبِيَّةً مِنْ كُرْبِ الدُّنْيَا ، نَفَسَ اللَّهُ عَنْهُ كُرْبِيَّةً مِنْ كُرْبِ يَوْمِ الْقِيَامَةِ ، وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ ، يَسِّرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالْآخِرَةِ ، وَمَنْ سَتَرَ مُسْلِمًا ، سَتَرَهُ اللَّهُ فِي الدُّنْيَا وَالْآخِرَةِ ، وَاللَّهُ فِي عَوْنِ الْعَبْدِ مَا كَانَ الْعَبْدُ فِي عَوْنِ أَخِيهِ ، وَمَنْ سَلَكَ طَرِيقًا يَلْتَمِسُ فِيهِ عِلْمًا ، سَهَّلَ اللَّهُ لَهُ بِهِ طَرِيقًا إِلَى الْجَنَّةِ ، وَمَا اجْتَمَعَ قَوْمٌ فِي بَيْتٍ مِنْ بُيُوتِ اللَّهِ يَتْلُونَ كِتَابَ اللَّهِ ، وَيَتَدَارَسُونَ بَيْنَهُمْ ، إِلَّا نَزَلَتْ عَلَيْهِمُ السَّكِينَةُ ، وَعَشِيَتْهُمْ الرَّحْمَةُ ، وَحَقَّتْ لَهُمُ الْمَلَائِكَةُ ، وَذَكَرَهُمُ اللَّهُ فِيمَنْ عِنْدَهُ ، وَمَنْ بَطَأَ بِهِ عَمَلُهُ ، لَمْ يُسْرِعْ بِهِ نَسَبُهُ

Dari Abu Hurairah r.a., Nabi ﷺ bersabda, “Barangsiapa yang melapangkan satu kesusahan dunia dari seorang Mukmin, maka Allah melapangkan darinya satu kesusahan di hari Kiamat. Barangsiapa memudahkan (urusan) orang yang kesulitan (dalam masalah hutang), maka Allah Azza wa Jalla memudahkan baginya (dari kesulitan) di dunia dan akhirat. Barangsiapa menutupi (aib) seorang Muslim, maka Allah akan menutup (aib)-nya di dunia dan akhirat. Allah senantiasa menolong seorang hamba selama hamba tersebut menolong saudaranya. Barangsiapa menempuh jalan untuk menuntut ilmu, maka Allah akan mudahkan baginya jalan menuju Surga. Tidaklah suatu kaum berkumpul di salah satu rumah Allah (masjid) untuk membaca Kitabullah dan mempelajarinya di antara mereka, melainkan ketenteraman akan turun atas mereka, rahmat meliputi mereka, Malaikat mengelilingi mereka, dan Allah menyanjung mereka di tengah para Malaikat yang berada di sisi-Nya. Barangsiapa yang diperlambat oleh amalannya (dalam meraih derajat yang tinggi-red), maka garis keturunannya tidak bisa mempercepatnya.”

Merujuk pada pendapat salah satu *muhaqqiq*, Muhammad Nashiruddin Al-Albani, hadits di atas berisi tentang perintah meringankan beban orang lain dan Allah berjanji akan meringankan beban kita (Mas'adi, 2014). Allah juga memerintahkan kita untuk tidak mempersulit urusan orang lain dan Allah berjanji tidak akan mempersulit kita di akhirat nanti. Allah juga memerintahkan kita untuk menutup aib orang lain dan Allah berjanji akan menutup aib kita. Dalam konteks sains, terdapat kesinambungan pada sebuah sistem dimana sistem tersebut dibuat untuk mempermudah urusan atau pekerjaan seseorang.

Disamping perintah untuk menolong kepada sesama, Rasulullah ﷺ juga mengajak ummatnya untuk memiliki etos kerja yang tinggi. Karena Allah SWT sangat mencintai hamba-Nya yang bekerja secara professional.

عَنْ عَائِشَةَ رَضِيَ اللَّهُ عَنْهَا قَالَتْ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ: إِنَّ اللَّهَ تَعَالَى يُحِبُّ إِذَا عَمِلَ (أَحَدُكُمْ عَمَلًا أَنْ يُثِقَنَهُ) (رواه الطبرني والبيهقي)

Dari Aisyah r.a., sesungguhnya Rasulullah ﷺ bersabda: “Sesungguhnya Allah mencintai seseorang yang apabila bekerja, mengerjakannya secara profesional”. (HR. Thabrani, No: 891, Baihaqi, No: 334).

Jika seseorang itu bekerja secara professional, maka sebuah bisnis yang dibangun akan sesuai dengan tujuan. Bisnis pada dasarnya merupakan kegiatan yang bertujuan untuk mendapatkan keuntungan. Namun terdapat juga pelaku bisnis yang hanya ingin mendapatkan keuntungan sebanyak-banyaknya tanpa berfikir bahwa jalan yang dilakukan itu baik atau tidak. Allah SWT menurunkan ayat tentang beramal saleh, salah satu integrasinya adalah pada etos bekerja secara professional dan baik. Sehingga akan didapatkan keuntungan bukan hanya di dunia, melainkan juga di akhirat kelak.

إِنَّ الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ أُولَٰئِكَ هُمْ خَيْرُ الْبَرِيَّةِ ﴿٧﴾

Artinya : “Sesungguhnya orang-orang yang beriman dan mengerjakan amal saleh, mereka itu adalah sebaik-baik makhluk.”. (QS Al-Bayyinah, 98:7)

*BPM System* atau *Business Process Management System* merupakan sebuah sistem yang memungkinkan pengguna dapat membuat model, desain, merealisasi serta mengelola proses-proses bisnis yang telah dibuat sehingga dapat memantau bisnis tersebut secara fleksibel. *BPM System* juga memiliki fungsi untuk melaporkan segala respon dari proses pembuatan bagan proses bisnis. Sehingga secara umum, *BPM System* adalah sebuah sistem yang dapat mengimplementasikan

BPM. Dengan adanya *BPM System* diharapkan kinerja dari sebuah bisnis sesuai dengan proses bisnis yang dikembangkan maka efisiensi suatu bisnis dapat ditingkatkan dan berjalan dengan lancar. Karena setiap orang yang berada di dalamnya telah mengetahui tugas dan kewajibannya masing-masing.

Syariahrooms merupakan salah satu *startup* yang menjalankan bisnisnya pada bidang akomodasi penginapan. Mereka tentunya memiliki banyak proses bisnis, mulai dari *handling* tamu yang akan menginap, mencatat pesanan, memeriksa pembayaran dari tamu, pengelolaan *check in* dan *check out* tamu, pembersihan kamar setelah tamu *check out*, dan seterusnya. Terlepas dari itu, terdapat beberapa permasalahan yang menunjang harus adanya *BPM System*, antara lain.

1. Biaya pengembangan Sitem Informasi atau aplikasi serupa yang mahal.

Biaya pengembangan aplikasi yang mahal merupakan salah satu penghambat sebuah bisnis dapat berkembang. Analisis biaya yang dikeluarkan untuk mengembangkan Sistem Informasi Pondok Pesantren dengan perencanaan jangka panjang (5 Tahun) lebih dari 1 milyar (Firdausi, 2013). Biaya tersebut merupakan data pada tahun 2013 yang terdiri dari biaya pengembangan Sistem Informasi pada setiap bagian dari Pondok Pesantren seperti Sistem Informasi Manajemen Penerimaan Siswa, Surat-menyurat, Kepegawaian, Keamanan dan Ketertiban, Asrama, dan lain sebagainya.

Perusahaan dengan sumber daya finansial yang terbatas, akan sangat berdampak dalam menjalankan bisnisnya di kemudian hari. Sebab, anggaran dari perusahaan tidak hanya ditujukan untuk pengembangan aplikasi saja. Itupun jika perusahaan tersebut hanya bergerak pada satu lini bisnis saja. Jika suatu perusahaan mempunyai beberapa lini bisnis dan memiliki proses bisnis yang berbeda serta pelaporan yang harus terintegrasi, maka akan menjadi sebuah masalah yang besar terhadap sumber daya finansialnya.

2. Proses bisnis yang dengan cepat berubah.

Dalam menjalankan proses bisnis terkadang tidak sesuai rencana. Ada yang memutuskan untuk merubah proses bisnis tersebut dan juga ada yang bertahan dengan proses bisnis yang ada. Namun, dalam keadaan apapun, rasionalitas seorang pemimpin perusahaan akan mengarahkan perusahaannya

kepada sebuah kesuksesan. Sehingga besar kemungkinan proses bisnis cepat berubah jika hasil tidak sesuai dengan tujuan yang ingin dicapai. Kemampuan beradaptasi merupakan konsep praktis yang dapat menjadi basis bagi suatu perusahaan dalam menghadapi lingkungan yang berubah-ubah. Siasat ini melibatkan tidak hanya cara yang berbeda dari operasi bisnis yang biasa tetapi juga cara yang sangat berbeda dari melaksanakan strategi bisnis. (Pamulu, 2016)

Dalam menjalankan bisnis, Syariahrooms mengalami perubahan proses bisnis dengan sangat cepat. Hal itu disebabkan oleh beberapa faktor, antara lain munculnya kompetitor baru, adanya *staff* baru, penjalinan kerjasama dengan pihak ketiga, dan lain sebagainya. Jika pihak manajemen Syariahrooms tidak merubah proses bisnis yang sedang digunakan, cepat atau lambat mereka tidak akan bisa bertahan untuk bersaing.

3. Banyaknya level akses user dalam satu sistem yang mempunyai hak aksesnya masing-masing.

Dalam penggunaan sistem *informasi* atau aplikasi serupa membutuhkan seorang *administrator* untuk mengelolanya. Namun, saat ini pihak pimpinan atau eksekutif, HRD, *supervisor*, akuntan juga menjadi pengguna yang bisa mengelola sistem *informasi* tersebut. Semua level akses tersebut mempunyai hak aksesnya masing-masing. Sehingga aplikasi juga mempunyai alurnya masing-masing. Pemberian hak akses tersebut guna meminimalisir serta mendeteksi penyimpangan-penyimpangan yang mungkin terjadi (Sutedi, 2018).

Pada awal pengembangan sistem *informasinya*, Syariahrooms hanya mempunyai 2 level akses, yaitu *owner* dan *admin*. Namun berkembangnya perusahaan membuat level akses tersebut bertambah menjadi *owner*, *admin*, *supervisor*, *finance*, *housekeeper*, *mitra*, dan lain sebagainya. Dengan bertambahnya level akses tersebut yang memiliki hak akses yang berbeda-beda, sehingga perlu untuk merubah struktur kode yang ada pada sistem *informasi* tersebut.

Dari penjelasan di atas dapat disimpulkan bahwa *BPM System* sangat penting adanya agar permasalahan-permasalahan tersebut dapat diatasi. Sehingga penulis

mengangkat judul “*Business Process Management System untuk Sistem Informasi Penginapan Pada Syariahrooms Hospitality*”.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, maka dirumuskan masalah adalah bagaimana merancang dan membangun *Business Process Management System* yang *flexible* dan *reusable* untuk Sistem Informasi pada Syariahrooms Hospitality?

### **1.3. Tujuan Penelitian**

Berdasarkan latar belakang yang telah diuraikan, maka tujuan penulisan adalah merancang dan membangun *Business Process Management System* yang *flexible* dan *reusable* untuk Sistem Informasi pada Syariahrooms Hospitality.

### **1.4. Manfaat Penelitian**

Manfaat yang dapat diambil dari penulisan ini antara lain:

1. Mempermudah pengelolaan sistem *informasi* suatu perusahaan dalam perubahan proses bisnis yang berlaku.
2. Meningkatkan *flexibility* dan *reusability* sistem *informasi* dalam suatu perusahaan.

### **1.5. Batasan Masalah**

Supaya pembahasan dalam penelitian ini tidak menyimpang diperlukan batasan-batasan masalah antara lain:

1. Sistem yang dibangun menggunakan Codeigniter Framework.
2. Pemodelan proses menggunakan Bizagi Process Modeller.
3. Data uji pada sistem yang dibangun menggunakan file yang berbentuk *XML Process Definition Language (XPDL)*.
4. Model proses didapat dari wawancara secara langsung dengan CEO Syariahrooms.
5. Fitur yang akan dimasukkan dalam model proses hanya terbatas pada manajemen *user*, manajemen *property*, manajemen *booking* dan manajemen pembayaran.
6. *Web service* yang disediakan sesuai dengan model proses yang dimasukkan.

7. Bisnis model yang dibuat adalah bisnis model yang terstruktur, memiliki 1 *Start Event* dan 1 *End Event*.
8. BPMN yang mempunyai menu hanya bisa mengakomodasi gerbang XOR.

### **1.6. Sistematika Penulisan**

Sistematika penulisan ini terdiri dari lima bab yang terdiri dari pendahuluan, tinjauan pustaka, metode penelitian, pembahasan, dan penutup. Masing-masing bab akan dijelaskan sebagai berikut :

#### **BAB I : PENDAHULUAN**

Pendahuluan berisi latar belakang masalah yang menjadi landasan penelitian, tujuan dan manfaat dari penelitian, kemudian batasan-batasan masalah dalam penelitian, dan sistematika penulisan.

#### **BAB II: TINJAUAN PUSTAKA**

Tinjauan pustaka akan membahas mengenai analisis permasalahan-permasalahan pada penelitian dan penelitian-penelitian terkait *Business Process Management System* untuk Sistem Informasi Penginapan Pada *Syariahrooms Hospitality*.

#### **BAB III : METODE PENELITIAN**

Metode penelitian akan membahas tentang penjelasan rancangan struktur program dan antarmuka dari *Business Process Management System* yang dibuat.

#### **BAB IV : HASIL DAN PEMBAHASAN**

Hasil dan pembahasan menjelaskan tentang implementasi sistem berdasarkan rancangan yang telah dibuat dan pengujian aplikasi terhadap data uji untuk membuktikan bahwa aplikasi sesuai dengan tujuan penelitian yang disebutkan.

#### **BAB V : PENUTUP**

Bab terakhir atau penutup bab akan membahas tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang dapat dilakukan untuk mengembangkan penelitian ini di masa mendatang.

## **BAB II**

### **KAJIAN PUSTAKA**

Pada bab kajian pustaka membahas tentang kajian-kajian dasar penulisan yang dilakukan antara lain *Business Process Management (BPM)*, *Business Process Model Noation (BPMN)*, *Service Oriented Architecture (SOA)*, *Business Process Execution Language (BPEL)*, *Business Process Management System (BPM System)*, *McCall's Software Quality Framework*, *Black Box Testing* serta produk-produk yang terkait dengan *BPM System* yang akan dibangun.

#### **2.1. Business Process Management (BPM)**

BPM merupakan sebuah disiplin manajemen berfokus pada peningkatan kinerja organisasi melalui peningkatan, mengelola dan mengendalikan proses bisnis (Dallas & Wynn, 2014). BPM merupakan salah satu metode pendekatan yang sangat bermanfaat. Dalam beberapa tahun terakhir BPM telah banyak diperhatikan karena potensinya yang secara signifikan meningkatkan produktivitas dan menghemat biaya dalam mengelola proses bisnis. Terdapat 3 level yang perlu diperhatikan dalam BPM (Chotijah, 2019), yaitu:

a. *Enterprise Level*

Tingkatan ini berpusat pada metode perencanaan dan penunjang perusahaan pada umumnya mempersiapkan pondasi atau proses bisnis dalam mengatur agar sesuai dengan strategi bisnis perusahaan.

b. *Business Process Level*

Di tingkat ini, pusatnya ada pada rencana dan pelaksanaan proses bisnis tertentu.

c. *Implementation Level*

Level ini berpusat pada bagian TI (Teknologi Informasi) dan SDM (Sumber Daya Manusia) dalam membuat sumber daya yang dibutuhkan untuk menjalankan proses bisnis organisasi.

BPM harus memperhatikan karakteristik unik masing-masing perusahaan. Skala organisasi adalah salah satu karakteristik terpenting untuk diingat. Penciptaan model kematangan untuk proses bisnis adalah salah satu bidang utama BPM yang mendapat banyak perhatian. Organisasi harus menggunakan tingkat kematangan

untuk mempertimbangkan proses manajemen yang ada dan menemukan area di mana mereka dapat meningkatkan kematangan mereka.

Tahapan sistematis tingkat *maturity* BPM ada 4 (McCormack et al., 2009), yaitu *ad-hoc*, *defined*, *linked*, dan *integrated*. Setiap fase memiliki kumpulan atributnya sendiri, dan atribut pada tahap berikutnya didasarkan pada tahapan sebelumnya untuk menaikkan tingkat kematangan. Definisi berikut diberikan untuk tahap di mana organisasi berorientasi proses bisnis:

- ***Ad-Hoc***: Prosesnya tidak terstruktur dan tidak jelas. Tidak ada langkah-langkah proses, dan peran serta struktur organisasi dibangun di sekitar tugas-tugas tradisional daripada proses horizontal. Level ini berada di level 0-4.
- ***Defined***: Proses-proses fundamental telah dijelaskan, terdaftar, dan terwakili dalam diagram alur. Perubahan pada fase ini sekarang harus melalui saluran terstruktur. Metode ini digabungkan ke dalam pekerjaan dan kerangka kerja organisasi, tetapi tetap fungsional. Perwakilan dari area fungsional (penjualan, manufaktur, dll.) Bertemu secara teratur untuk berkolaborasi satu sama lain, tetapi hanya dalam peran konvensional mereka. Penyesuaian metode harus mengikuti proses terstruktur, pekerjaan dan organisasi, serta struktur proses, dan memiliki hubungan yang sering untuk berkomunikasi satu sama lain. Tingkat kesulitan untuk level ini adalah 4-5.5.
- ***Linked***: Tingkat pencapaian. Manajemen sistem digunakan oleh manajer untuk mencapai tujuan strategis. Kerja sistem dan proses yang ekstensif diletakkan di luar peran konvensional. Level ini antara 5,55 dan 6,5.
- ***Integrated***: Kerja sama dibawa ke tingkat berikutnya oleh organisasi, vendor, dan pemasoknya. Proses mendukung struktur organisasi dan pekerjaan, dan peran konvensional mulai sama atau, dalam beberapa kasus, tunduk pada proses. Fase proses dan sistem manajemen berakar kuat di perusahaan. Level ini berada di antara level 6,5 dan 7.

Konsep utama dari BPM adalah fokus tanpa henti pada proses. Proses adalah aset terpenting dari suatu organisasi dan yang menghasilkan nilai-nilai yang membenarkan keberadaan organisasi. Agar proses dikelola dan ditingkatkan secara efektif, BPM mendukung struktur organisasi yang berorientasi pada proses. Tiga jenis struktur berorientasi proses telah diidentifikasi (Chang, 2016), yaitu *Process*

*organization, Case management organization dan Horizontal process management organization.*

*Process organization* adalah salah satu jenis yang menyelaraskan struktur organisasi di sepanjang garis proses. Setiap unit proses akan berisi berbagai fungsi yang mendukung proses. Keuntungan dari proses organisasi adalah mengoptimalkan kinerja proses. Namun, hal ini dilakukan dengan biaya duplikasi fungsi untuk proses. Ini dapat mengakibatkan pekerjaan fungsional tidak dilakukan secara konsisten dan kenaikan biaya karena duplikasi fungsi.

Dalam *Case management organization*, karyawan masih akan melapor ke kepala fungsional. Selain itu, mereka akan melapor kepada manajer kasus. Manajer kasus memiliki tanggung jawab untuk mengawasi proses ujung ke ujung dari sebuah kasus individual. Kasusnya dapat berupa pasien (struktur ini dipelopori dalam industri perawatan kesehatan untuk memberikan layanan pasien yang lebih baik mulai dari masuk hingga pulang) atau jenis proyek apapun.

Dalam struktur *Horizontal process management organization*, pelaporan garis solid karyawan akan ke kepala proses, sedangkan kepala fungsional akan memiliki hubungan garis putus-putus. Namun, ini tidak perlu bagi organisasi manajemen proses untuk bekerja. Pengaturan dalam organisasi ini mengurangi duplikasi fungsi tetapi masih memungkinkan organisasi untuk mengambil manfaat dari fokus proses. Kelemahannya adalah sulit untuk diimplementasikan. Organisasi yang ingin mengadopsi struktur organisasi ini harus memiliki tingkat keahlian manajemen yang canggih karena organisasi matriks sering menimbulkan kebingungan di antara karyawan tentang peran dan tanggung jawab. Adopsi BPM dan teknik manajemen proses oleh organisasi telah menyebabkan minat yang kuat di dunia teknologi untuk mengembangkan produk dan solusi untuk mendukung BPM.

## **2.2. Business Process Model Notation (BPMN)**

*Business Process Model Notation (BPMN)* adalah deskripsi proses bisnis. Sedangkan model proses bisnis adalah metode memahami, merancang, dan menganalisis proses bisnis (Dewi et al., 2017). BPMN dapat membantu suatu perusahaan dalam memahami proses bisnis dengan baik sehingga resiko pekerjaan yang tidak sesuai akan terminimalisir. Hal tersebut akan meningkatkan kepercayaan

para *stakeholder* pada perusahaan tersebut, karena mereka bukan hanya dapat memahami, juga mereka dapat melakukan evaluasi terkait proses bisnis yang dinilai tidak sesuai. Berikut adalah aturan dalam memodelkan suatu proses bisnis (Dewi et al., 2017):

- Buat model peristiwa yang menjelaskan proses mulai, proses itu sendiri, dan hasil dari alur proses.
- Gateway mewakili keputusan dalam alur percabangan.
- Sebuah proses dapat memiliki sub-proses yang dapat diwakili oleh *Business Process Diagram* (BPD) lain yang ditautkan ke simbol proses melalui hyperlink.
- Tanda "+" menggambarkan bahwa proses tersebut diuraikan; jika tidak, maka proses tersebut disebut sebagai tugas.

Beberapa notasi telah ada sebelum munculnya BPMN pada bulan Maret 2007 yang dikenalkan oleh Object Management Group, Inc. (OMG) antara lain (Vega-Márquez et al., 2019):

- *Flowcharts* pada awal tahun 1900
- *Event-Driven Process Chains* (EPC) pada tahun 1992
- *Unified Modeling Language* (UML) pada tahun 1995
- *Yet Another Workflow Language* (YAWL) pada tahun 2002

Menurut Object Management Group, BPMN telah menjadi standar *de facto* untuk diagram proses bisnis. Ini dimaksudkan untuk digunakan secara langsung oleh para pemangku kepentingan yang merancang, mengelola dan merealisasikan proses bisnis, tetapi pada saat yang sama cukup tepat untuk memungkinkan BPMN digambarkan untuk diterjemahkan ke dalam komponen proses perangkat lunak. BPMN memiliki notasi seperti *flowchart* yang mudah digunakan yang tidak tergantung pada lingkungan implementasi tertentu. Tujuan utama BPMN adalah menyediakan dokumentasi yang mudah dipahami oleh semua klien bisnis, mulai dari penguji bisnis yang menyusun *formulir* awal hingga insinyur khusus yang mempresentasikan kemajuan yang akan melaksanakan *formulir* tersebut, dan akhirnya, individu bisnis yang akan mengontrol dan melacak bentuk-bentuk itu. Sehingga BPMN menjembatani kesenjangan antara desain proses bisnis dan eksekusi proses. Tujuan lain yang tidak kalah penting adalah untuk memastikan

bahwa bahasa XML dirancang untuk pelaksanaan proses bisnis, seperti *Web Service Business Process Execution Language* (WSBPEL) yang dapat divisualisasikan dengan notasi yang berorientasi bisnis.

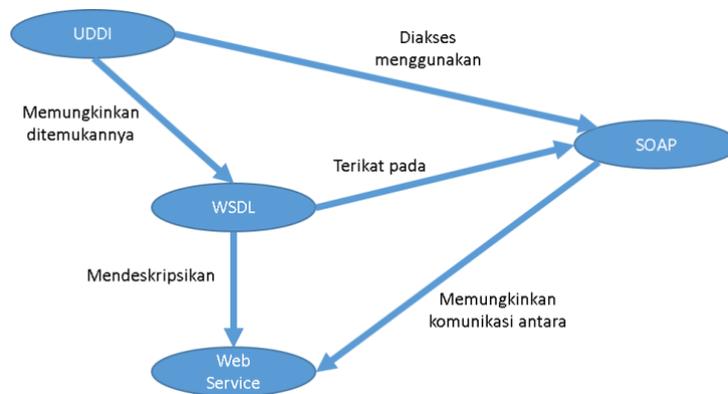
Workflow Management Coalition (WfMC) membuat *format* standar untuk mengubah konsep proses bisnis antara alur kerja yang dimanakan XML Process Definition Language (XPDL). XPDL dibuat untuk dapat berbagi deskripsi sebuah proses dari alur kerja proses bisnis. Format terbaik dalam pertukaran informasi dari BPMN adalah XPDL karena di dalamnya terdapat semua informasi-informasi dari BPMN. Dalam versi yang saat ini digunakan adalah XPDL 2.0 yang salah satunya dapat di ekspor melalui perangkat lunak dari perusahaan *software* Bizagi yang bernama Bizagi Modeler.

### **2.3. Service Oriented Architecture (SOA)**

*Web service* adalah sebuah sistem terdistribusi berelemen yang dapat di-*deploy* dan diakses menggunakan *Hyper Text Transport Protocol* (HTTP) ataupun HTTPS (HTTP Secure). Perkembangan pesat layanan web telah mengakibatkan munculnya ekstensi layanan web yang memperluas fungsi itu sendiri (Erl, 2005). *Web service* memiliki dua teknik populer, yaitu *Simple Object Access Protocol* (SOAP) dan *REpresentational State Transfer* (REST) (Yusrizal et al., 2017). Keduanya digunakan sebagai metode pertukaran pesan dalam komunikasi. Namun dalam penulisan *format* pesan yang digunakan SOAP lebih khusus, yaitu *Extensible Markup Language* (XML). Sedangkan REST tidak ada spesifikasi khusus dalam penulisan *format* pesan.

Sebuah aplikasi *web service* tentunya tidak mungkin untuk memprediksi bagaimana program layanan web akan digunakan secara langsung. Akibatnya, Deskripsi Layanan diperlukan untuk aplikasi ini. Web Service Description Language (WSDL) adalah standar untuk jenis deskripsi ini (WSDL). Penemuan dan Integrasi Deskripsi Universal, mekanisme pencarian aplikasi layanan web biasa, juga disediakan oleh layanan web (UDDI). Di dunia maya, UDDI mirip dengan katalog layanan web, yang memungkinkan untuk menemukan layanan web yang akan digunakan sesuai dengan kebutuhan. Namun sampai saat ini UDDI belum

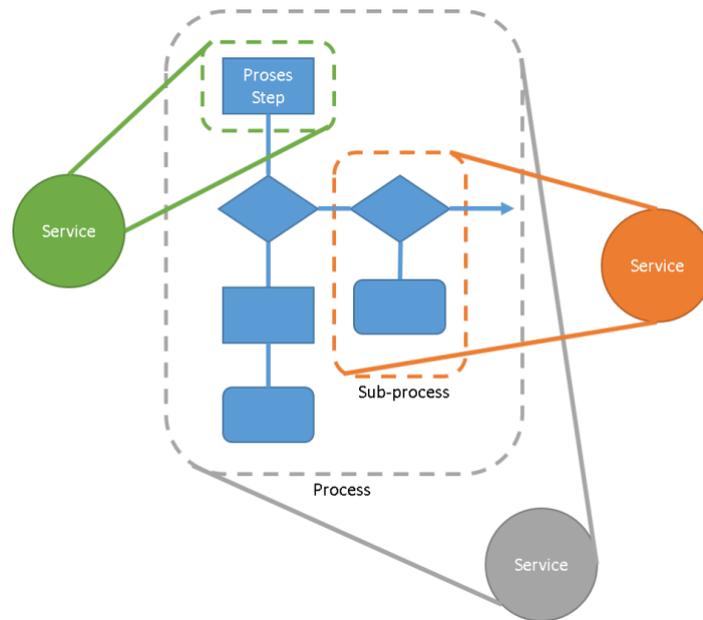
digunakan secara luas. Gambaran dari WSDL dapat dilihat pada gambar 2.1. (Sarwosri & Naja, 2011)



Gambar 2. 1 Keterhubungan Antar Komponen *Web Service*

Saat ini telah banyak perusahaan yang menggunakan aplikasi perangkat lunak guna memenuhi kebutuhan bisnis. Namun tidak semua perusahaan dapat mengembangkan aplikasi itu sendiri, sehingga membutuhkan banyak aplikasi yang berbeda yang dibangun dengan teknologi berbeda serta standar dan oleh *vendor* yang berbeda. Agar perusahaan dapat menghasilkan fungsionalitas bisnis yang sesuai dengan harapan atau bahkan bisa lebih baik, mereka harus membuat aplikasi-aplikasi tersebut agar dapat terintegrasi. Hal itu dibuktikan dengan adanya kolaborasi atau kerjasama antar perusahaan atau *startup* yang semakin lama semakin banyak.

SOA adalah sebuah arsitektur teknologi yang berprinsip pada *service-orientation* (berorientasi layanan). Istilah *service-oriented* telah ada sejak lama dan telah digunakan dalam berbagai konteks untuk memecahkan masalah tertentu dan mencapai tujuan tertentu. Prinsip orientasi-layanan menggunakan strategi yang membagi sebuah layanan kompleks menjadi serangkaian layanan kecil yang sederhana dengan tujuan untuk menyelesaikannya. SOA tidak memiliki standar khusus dalam pembuatannya, melainkan dengan pendekatan modular untuk pembuatan perangkat lunak. (Kapojos et al., 2012)



Gambar 2. 2 Enkapsulasi *business process* dengan service

Dalam konteks SOA, *Service* adalah kumpulan fungsi, prosedur, atau proses yang akan merespons ketika klien memintanya. Seperti yang diilustrasikan pada Gambar 2.2, cakupan layanan tidak terbatas; layanan dapat mencakup seluruh proses atau satu proses kecil saja. Ini dapat diubah sesuai kebutuhan. Karena masalah dapat dipartisi menjadi banyak administrasi, pengaturan untuk masalah ini harus melibatkan kepentingan dari semua administrasi. Untuk ini, administrasi harus menjelaskan sejumlah masalah, menghitung bagaimana administrasi terkait, bagaimana mereka terkait, bagaimana mereka diatur, dan bagaimana pesan antar administrasi digambarkan. (Erl, 2005)

Divisi berbasis layanan ini bukanlah hal baru, karena sudah dimanfaatkan secara luas. Keunikan dari prosedur berorientasi layanan ini, di sisi lain, terkait dengan karakteristiknya (Erl, 2005), yaitu:

- *Loosely coupled*, yaitu setiap layanan mandiri dan berjalan tanpa bantuan layanan lain. Ketergantungan layanan dikurangi seminimal mungkin, hanya membutuhkan mekanisme kontak.
- *Service contract*, yaitu setiap layanan memiliki protokol komunikasi sendiri.
- *Autonomy*, yaitu layanan ini memiliki kendali penuh atas logika algoritma yang dienkapsulasi.

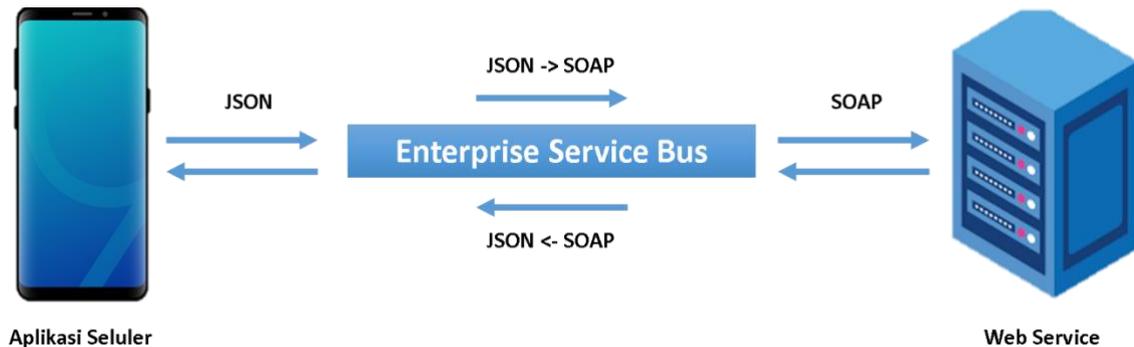
- *Abstraction*, yaitu layanan tidak mendemonstrasikan bagaimana logika algoritmik diterapkan.
- *Reusability*, yaitu logika algoritma dipecah menjadi sejumlah layanan yang dapat digunakan kembali.
- *Statelessness*, yaitu operasi yang dilakukan layanan tidak memiliki status khusus.
- *Discoverability*, yaitu layanan dibuat deskriptif sehingga dapat diidentifikasi dan digunakan oleh mesin pencari.
- *Composability*, yaitu layanan satu dapat dipasangkan dengan layanan lain. Ini memfasilitasi penggunaan kembali dan pembentukan lapisan dengan memungkinkan logika algoritme diekspresikan pada berbagai tingkat perincian.

Klarifikasi di atas hanya menjelaskan bahwa SOA terdiri dari sekumpulan layanan yang tidak akan cukup untuk membentuk arsitektur tersebut. Terdapat empat komponen yang harus dipenuhi, antara lain (Erl, 2005):

- a. *Message*, yaitu informasi yang diperlukan untuk menyelesaikan suatu bagian atau unit kerja, yang diteruskan dari satu operasi ke operasi berikutnya.
- b. *Operation*, yaitu tugas layanan untuk memproses pesan untuk mengirimkan sesuatu. Untuk melengkapi suatu unit kerja, fungsi-fungsi tersebut akan saling berkomunikasi.
- c. *Service*, merupakan kumpulan operasi serupa yang harus diselesaikan agar sekumpulan unit kerja terkait dapat diselesaikan.
- d. *Process*, merupakan aturan bisnis yang menunjukkan operasi mana yang digunakan untuk mencapai tujuan tertentu.

SOA adalah sebuah model untuk mendefinisikan dan mengevaluasi arsitektur teknologi informasi yang memberdayakan aplikasi yang berbeda untuk berbagi informasi dan mengambil bagian dalam proses bisnis. SOA membagi fungsi menjadi beberapa unit (layanan) yang dapat tersebar di seluruh jaringan, terintegrasi, dan digunakan kembali untuk membuat aplikasi bisnis. Layanan ini berkomunikasi satu sama lain dengan berbagi data dengan pelanggan atau mengoordinasikan aktivitas antara dua atau lebih layanan.

*Enterprise Integration* merupakan sebuah konsep dalam rekayasa perusahaan untuk memberikan informasi yang relevan. EI berfokus pada interkoneksi sistem, pertukaran data elektronik dan produk serta lingkungan komputasi terdistribusi. Dengan konsep tersebut dapat memungkinkan komunikasi antara manusia, mesin serta komputer juga memungkinkan kerja sama dan koordinasi mereka yang efisien. Aplikasi yang dirancang untuk dapat menjalankan konsep tersebut dikenal dengan Enterprise Service Bus (ESB).



Gambar 2. 3 Enkapsulasi *business process* dengan service

ESB memungkinkan beragam aplikasi, layanan, dan sistem untuk saling berinteraksi yang sesuai dengan pola integrasi perusahaan atau yang dikenal dengan *Enterprise Integration Pattern* (EIP). ESB dapat digunakan sebagai lapisan integrasi perantara antara dua atau lebih sistem dan layanan yang berbeda (Indrasiri, 2016). Oleh karena itu, seperti yang diilustrasikan pada Gambar 2.3, ESB dapat ditempatkan antara aplikasi seluler berbasis JSON dan layanan web berbasis SOAP. Tanpa menulis kode apa pun, pengguna dapat mengonfigurasi ESB untuk melakukan konversi *format* pesan. Jika kasus penggunaan bisnis perlu diperluas lebih lanjut untuk mencakup layanan lain, pengembang dapat mengintegrasikan layanan itu ke ESB, dan alih-alih memiliki tautan *point-to-point*, ketiga sistem dapat dihubungkan melalui lapisan integrasi ESB yang terpadu.

Namun seiring berubahnya waktu dan perkembangan teknologi yang sangat pesat, muncullah *Application Programming Interface* atau yang biasa kita kenal dengan API. API dinilai lebih baik daripada ESB dari segi integrasi dan biaya (Daly, 2017). API yang identik dengan RESTfull API ini menyediakan integrasi *multi-platform* yang juga bisa diakses kapanpun karena API sudah berada pada *cloud system*. Dengan *cloud system* yang digunakan dalam penerapan API ini, biaya

pengembangan aplikasi dapat ditekan. Berbeda dengan ESB yang biaya *software*nya sudah mahal serta hanya dapat diakses oleh jaringan lokal. Dilansir oleh shadow-soft.com (Johnston, 2017), beberapa *tools* ESB memiliki harga langganan (dalam satu tahun) yang sangat tinggi bagi perusahaan yang memiliki sumber daya finansial yang terbatas, antara lain Red Hat JBoss Fuse mulai harga \$23.760 USD, Mule ESB mulai harga \$80.000 USD, IBM Websphere ESB mulai harga \$630.400 USD, Oracle ESB mulai harga \$73.600 USD, Microsoft BizTalk mulai harga \$173.360 USD. Jika ingin berada pada *cloud system*, perusahaan harus mengeluarkan biaya lagi karena ESB tidak dirancang untuk *cloud*.

WSO2 ESB merupakan salah satu produk yang disediakan oleh WSO2, Inc. WSO2 ESB datang dengan penawaran yang sangat elegan, yaitu *cloud-enabled and 100% free* (WSO2 Inc., 2016). Dengan hal itu, statement dari Daly telah terjawab dengan jelas bahwa saat ini ESB dapat bersaing dengan perkembangan API.

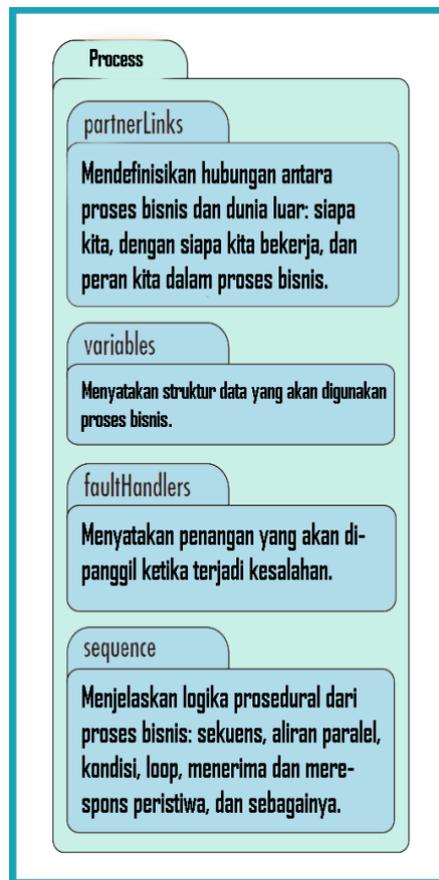
#### **2.4. Business Process Execution Language (BPEL)**

*Business Process Execution Language* (BPEL) adalah wadah atau *container* di mana dapat mendeklarasikan hubungan dengan mitra eksternal, deklarasi untuk data proses, penanganan untuk berbagai tujuan dan, yang paling penting, kegiatan yang akan dieksekusi. *Container* proses memiliki beberapa atribut, yaitu nama (wajib) dan deklarasi (juga wajib) dari *namespace*. (Barreto et al., 2007)

BPEL menawarkan kemungkinan untuk menggabungkan layanan web dan menentukan logika bisnis di antara masing-masing interaksi layanan ini. Juga dikatakan bahwa BPEL mengatur interaksi layanan web. Setiap interaksi layanan dapat dianggap sebagai komunikasi dengan mitra bisnis. Menerapkan proses bisnis memerlukan bahasa standar dan spesifik untuk proses bisnis yang menggunakan bahasa penerima publik serta menunjukkan proses bisnis. BPEL adalah bahasa cepat dan standar dominan. Tujuan utama BPEL adalah untuk membakukan otomatisasi proses layanan web (Juric et al., 2006).

Dalam SOA, layanan adalah komponen kunci arsitektur. Layanan adalah komponen kunci dari proses BPEL. Layanan ini adalah tugas yang disebut layanan perusahaan. Operasi layanan bisnis umumnya merujuk pada kegiatan komersial. Layanan bisnis ini digunakan untuk menjalankan prosedur bisnis, atau BPEL menggunakannya untuk menjalankan aktivitas proses. BPEL, dengan kata lain,

adalah kumpulan layanan bisnis. BPEL berisi aliran proses, sedangkan layanan bisnis menyediakan fungsionalitas (Juric et al., 2006).

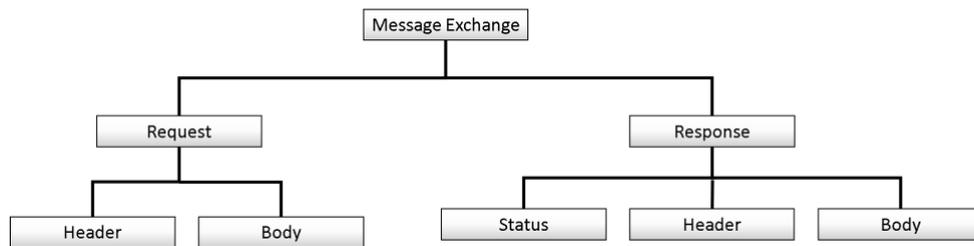


Gambar 2. 4 Struktur BPEL

Berdasarkan gambar 2.4, BPEL terdiri dari empat komponen utama, yaitu (Louridas, 2008) :

- Bagian `<partnerLinks>` mendefinisikan pihak yang berkolaborasi untuk menyelesaikan proses bisnis
- Bagian `<variables>` mendefinisikan struktur data yang akan digunakan oleh proses bisnis ditentukan.
- Bagian `<faultHandlers>`, ketika terjadi kesalahan, metode ini digunakan untuk mengelola aplikasi yang akan membuat eksekusi penanganan itu.
- Bagian `<sequence>` mendefinisikan prosedur metode yang terdiri dari aktivitas BPEL.

BPEL adalah bahasa berbasis XML yang memodelkan proses bisnis sebagai komposisi dari serangkaian layanan web dasar. Mesin BPEL disebut sebagai komponen perangkat lunak khusus yang dapat menjalankan definisi proses BPEL. Setiap proses BPEL dapat diakses sebagai layanan web dari mesin BPEL juga. Spesifikasi BPEL tergantung pada standar W3C, WSDL untuk deskripsi layanan web, skema XML untuk definisi struktur data, dan XPath untuk pengambilan elemen XML.



Gambar 2. 5 URI dalam BPEL

BPEL tidak memiliki pemodelan status eksplisit, tetapi implisit melalui konstruk `<scope>`. Dengan URI diperkenalkan ke BPEL, maka setiap interaksi URI sinkron dan tupel (kumpulan data) permintaan dan respons dikelompokkan ke dalam *Message Exchange* sebagaimana digambarkan pada gambar 2.5. Dengan menggunakan URI dalam BPEL dapat memodelkan *resource* dan URI sehingga dapat terbaca oleh *Web Service* yang berbasis HTTP dengan bantuan *Web Application Description Language (WADL)*. Seperti pada gambar 2.6 berikut:

```

<rbpel:onMessage path="/.item/new" messageExchange="createItem">
  <wadl:method name="POST" />
  <sequence>
    <assign>
      <copy>
        <from>rbpel:generate-
uri("./item/{itemNumber}")</from>
        <to variable="newItemURI" />
      </copy>
    </assign>
    <rbpel:reply messageExchange="createItem">
      <rbpel:status>201</rbpel:status>
      <rbpel:param name="Location"
style="header">$newItemURI</rbpel:param>
    </rbpel:reply>
  </sequence>
</rbpel:onMessage>
  
```

Gambar 2. 6 Contoh Kode BPEL

BPEL dapat diperoleh dengan menggunakan BPMN sebagai komponen utama dalam proses pengeksporannya (Ouyang et al., 2006). BPMN akan dipetakan sehingga *output* akhirnya adalah kode BPEL yang tergenerasi. Pemetaan yang dilakukan berfokus pada perspektif aliran kontrol. Berikut adalah langkah pemetaan BPMN ke BPEL:

1. mendefinisikan sintaks dari subset inti *Business Process Diagrams* (BPD) yang digunakan untuk pemetaan.
2. Mengurai BPD menjadi komponen yang merupakan bagian dari BPD yang memiliki titik masuk (*entry point*) dan titik keluar (*exit point*).
3. Menyusun terjemahan *activity-based* yang terdiri dari *sequence*, *flow*, *switch*, *pick* dan *first preference* jika struktur yang sesuai muncul di BPD. Komponen yang dapat dipetakan dengan tepat ke salah satu dari lima tersebut dianggap sebagai *well-structured*.
4. Menerjemahkan komponen yang *non-well-structured* ke dalam *scope activity* dengan mengeksploitasi konstruksi "*event handler*" di BPEL.
5. Mendefinisikan algoritma untuk menerjemahkan BPD ke BPEL.
6. Kode BPEL telah tergenerasi.

## **2.5. Business Process Management System (BPM System)**

*BPM System* merupakan terobosan dalam penggunaan dan implementasi sistem informasi. Metodologi implementasi sistem tradisional difokuskan pada fungsi dan objek. Proses diturunkan ke alur kerja, yang biasanya tidak mendapat perhatian besar selama implementasi. *BPM System* mematahkan mentalitas desain objek, yang lebih mirip blok bangunan lego dengan kemampuan desain proses yang terbatas. Ini menempatkan proses pada fokus utama dari desain solusi. Ini menyelaraskan solusi TI agar lebih sejalan dengan realitas proses fokus untuk organisasi BPM. Desain proses dapat dilakukan dan simulasi dilakukan di *BPM System*.

*BPM System* menyediakan alat yang kuat dalam siklus desain proses - implement - analisis - perbaiki. Dalam organisasi BPM, *BPM System* adalah alat penting untuk sukses. Dalam organisasi BPM yang telah menggunakan *BPM System* untuk manajemen proses, *BPM System* tidak hanya berfungsi sebagai alat desain proses sistemik, tetapi juga menyediakan alat perbaikan proses dengan data

proses dan *informasi* untuk mempercepat implementasi peningkatan proses. *BPM System* adalah alat teknologi yang digunakan bersama dengan semua alat peningkatan proses lainnya. *BPM System* menyediakan fungsi desain berharga yang akan membantu baik dalam proyek peningkatan proses yang bertahap atau lebih dramatis. Ini juga merupakan dasar teknologi untuk BPM. Setelah *BPM System* diterapkan, *BPM System* menyediakan data proses dan analitik yang membantu dalam pengelolaan proses bisnis sehari-hari. Data proses yang berisi *BPM System* juga berharga selama proyek peningkatan proses. Singkatnya, BPM memberikan dasar dan praktik yang memungkinkan suatu organisasi untuk mencapai keadaan perusahaan proses. Sebagai sintesis dari pendekatan peningkatan proses lainnya, BPM menyatukan pendekatan sebelumnya dalam satu kerangka kerja. Sebagai kerangka kerja, BPM menyediakan *informasi* proses, struktur organisasi, dan budaya perusahaan untuk membantu berbagai alat berhasil dalam perbaikan proses.

*BPM System* adalah kelas baru dalam perangkat lunak yang memungkinkan organisasi merancang solusi teknologi *informasi* proses sentris. Sentris-proses berarti solusi *BPM System* dapat mengintegrasikan orang, sistem, dan data. Organisasi yang memanfaatkan *BPM System* untuk mencapai perubahan proses bisnis yang didukung TI akan mendapatkan kapabilitas berikut (Chang, 2016):

- Keterlibatan bisnis yang lebih dekat dalam merancang solusi proses bisnis yang didukung TI
- Kemampuan dalam mengintegrasikan pengguna dan sistem yang terlibat dalam proses bisnis
- Kemampuan dalam mensimulasikan proses bisnis untuk merancang proses yang paling optimal untuk implementasi
- Kemampuan untuk memantau, mengendalikan, dan meningkatkan proses bisnis secara real time
- Kemampuan untuk melakukan perubahan pada proses bisnis yang ada secara real time tanpa upaya konversi proses yang rumit

Untuk membantu pemilik proses bisnis dan analis bisnis dalam desain proses, *BPM System* menyediakan simulasi proses dan kemampuan pemodelan. Menggunakan perancang proses bisnis *BPM System*, pemilik atau analis proses bisnis dapat merancang proses bisnis awal dan menjalankan desain proses dalam

mode simulasi. Mode simulasi mencakup distribusi probabilitas waktu untuk setiap aktivitas dalam proses simulasi.

*BPM System* berfungsi sebagai sistem pengawasan yang mengawasi proses bisnis setelah solusi proses diimplementasikan. Aspek pengawasan *BPM System* menyediakan kemampuan untuk memantau, mengendalikan, dan meningkatkan proses bisnis. Karena *BPM System* mengawasi semua langkah, baik manual maupun otomatis dalam proses bisnis, *BPM System* dapat memberikan informasi proses yang berharga. Perangkat lunak *BPM System* berfungsi sebagai monitor kinerja untuk proses. Kemampuan pemantauan dapat memungkinkan manajer bisnis untuk diberitahu tentang kejadian yang tidak biasa.

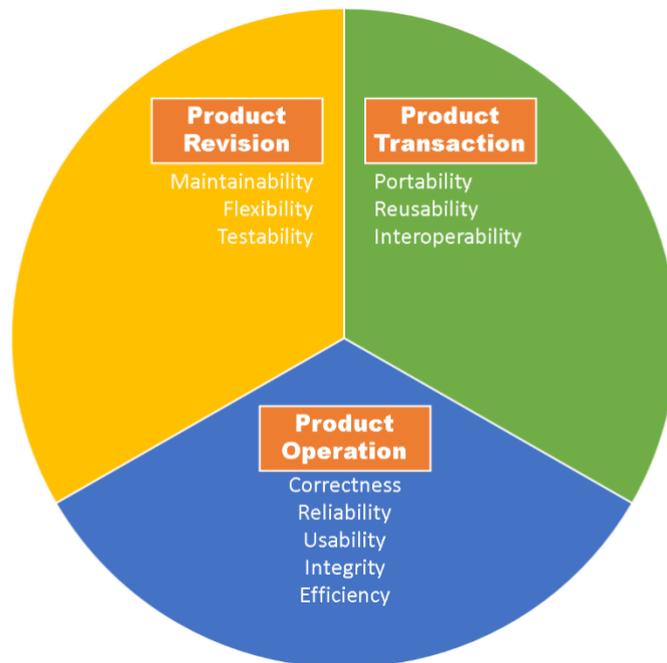
*BPM System* memberi kemampuan kepada organisasi untuk mengimplementasikan peningkatan proses *realtime* tanpa upaya konversi proses yang luas. Proses bisnis asli sudah ada dalam perancang proses bisnis. Ini menghilangkan kebutuhan untuk mengumpulkan informasi proses saat pada saat itu. Ketika kemacetan proses telah ditentukan, pemilik atau analis proses bisnis tersebut dapat memasukkan peningkatan pada proses menggunakan perancang proses bisnis. Sistem ini memungkinkan proses asli dan yang ditingkatkan untuk ada sampai semua pekerjaan dari proses asli selesai.

Teknologi BPM memungkinkan analis bisnis untuk berkolaborasi lebih dekat dengan orang-orang TI dalam melaksanakan proyek. Berbagai alat yang ditawarkan *BPM System* memberikan paradigma baru tentang bagaimana solusi dapat diimplementasikan. Organisasi tidak lagi terikat pada proses bisnis yang tertanam dalam aplikasi bisnis mereka. Dengan pembuatan alur kerja otomatis dan kemampuan portal Web, alur kerja dapat dengan mudah digunakan di berbagai aplikasi, sehingga mengintegrasikan orang ke dalam proses bisnis. Inovasi teknologi ini memungkinkan teknologi untuk memenuhi cita-cita manajemen proses dengan lebih baik. Proses dapat dikelola dalam kerangka proses. Kerangka kerja ini memungkinkan organisasi untuk memantau, mengukur, dan meningkatkan proses mereka.

## **2.6. McCall's Software Quality Framework**

McCall's *Software Quality Framework* merupakan sebuah struktur untuk mengkategorikan kriteria yang mempengaruhi kualitas perangkat lunak. Standar

tersebut dibagi menjadi tiga kategori, seperti yang ditunjukkan pada Gambar 2.7 di bawah ini (Mulyanto, 2016).



Gambar 2. 7 McCall's Software Quality Framework

McCall mengartikan *flexibility* sebagai kemampuan suatu perangkat lunak untuk melakukan modifikasi terhadap perubahan. Sedangkan *reusability* diartikan sebagai kemampuan suatu perangkat lunak untuk dapat menggunakan kembali modul-modulnya untuk aplikasi lainnya.

*Flexibility* merupakan kemudahan dimana sistem atau komponen dapat disesuaikan untuk dijalankan dalam aplikasi atau kondisi yang berbeda dari yang dibuat sebelumnya (Eden & Mens, 2006). *Flexibility* berdampak pada efisiensi operasional dan kemandirian sistem informasi dalam bisnis, tetapi jarang dipertimbangkan secara langsung dalam desain dan implementasi sistem ini.

*Flexibility* dapat dilihat dalam berbagai cara dalam konteks sistem informasi, termasuk pendekatan, produk, strategi, proses, dan sistem. Teknik dan komponen produk berkaitan dengan penggunaan atau pemanfaatan sistem informasi yang tentu saja terkait dengan arsitektur sistem dan pemeliharaan sistem. Kemampuan beradaptasi suatu sistem informasi menunjukkan bahwa sistem informasi yang digunakan berkualitas tinggi. Kemampuan sistem informasi untuk beradaptasi dalam menanggapi kebutuhan pengguna adalah fokus dari fleksibilitas. Pengguna

akan lebih puas ketika menggunakan sistem informasi yang dapat disesuaikan dengan kebutuhan mereka.

*Reusability* dalam sistem informasi kurang lebih sama dengan konsep *Object Oriented Programming* yang mana sebuah fungsi atau modul dapat digunakan kembali pada program lainnya. Sistem yang diperoleh secara eksternal dijelaskan dengan istilah seperti *commercial-off-the-shelf* (COTS) atau perangkat lunak pihak ketiga, *commercially available software* (CAS), dan *non-developmental item* (NDI). Ketika digunakan sebagai bagian dari sistem, mereka dapat disebut sebagai komponen. Sistem itu sendiri dikenal sebagai perangkat lunak berbasis komponen atau sistem sistem (Voas, 1998).

Penggunaan kembali perangkat lunak adalah konsep yang mengacu pada potensi penggunaan kembali komponen perangkat lunak. Karena semakin banyaknya organisasi yang ingin menggunakan perangkat lunak untuk berbagai aplikasi, banyak dari mereka mengadopsi pendekatan yang lebih berorientasi komponen di mana sistem atau aplikasi baru dapat menggunakan dari komponen yang ada. Dalam peran baru ini, perangkat lunak yang diperoleh harus terintegrasi dengan fungsionalitas perangkat lunak lainnya (Gill, 2003).

## **2.7. Black Box Testing**

Pengujian perangkat lunak merupakan sebuah teknik yang digunakan dalam memverifikasi serta memvalidasi kualitas sebuah perangkat lunak. Dalam *Software Development Life Cycle* (SDLC), pengujian perangkat lunak merupakan sebuah tahapan penting untuk membantu *software developer* memperoleh kepercayaan dari *client* bahwa perangkat lunak yang dikembangkan sesuai dengan apa yang telah dirancang. Pada umumnya, teknik pengujian perangkat lunak yang sering digunakan adalah *black box testing* dan *white box testing*.

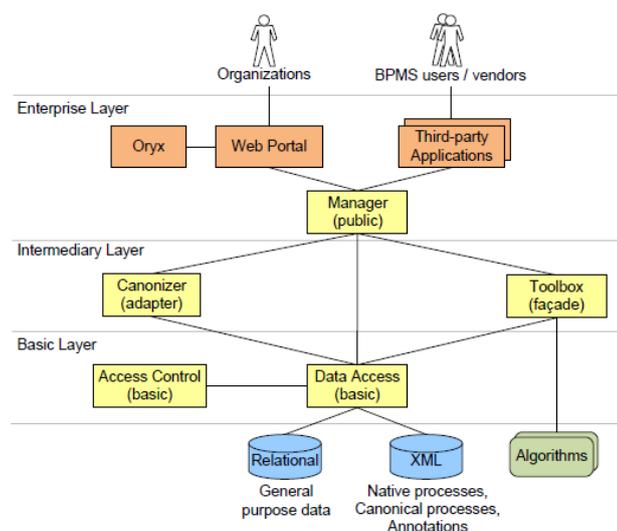
*White box testing* dapat juga disebut dengan pengujian struktural perangkat lunak yang bertujuan untuk menentukan apakah program telah dirancang dengan benar. Pengujian dilakukan dengan memeriksa *source code* dari perangkat lunak. Sedangkan *black box testing* dapat juga disebut dengan pengujian fungsionalitas dari sebuah program yang bertujuan untuk membuktikan apakah program yang dibuat sudah tepat. Hal tersebut mempunyai arti bahwa kesesuaian antara proses *input* dan *output* yang diinginkan. *Black box testing* tidak memerhatikan apa yang

ada di dalam sistem tersebut, namun hanya berfokus pada *output* yang dihasilkan. (Khan, 2011)

## 2.8. *Advanced Process Model Repository (APROMORE)*

*Advanced Process Model Repository* atau APROMORE adalah repositori model proses yang melampaui fitur khas repositori tradisional, seperti impor / ekspor model dan kontrol versi. Salah satu masalah yang sering dihadapi dalam menggunakan model proses adalah jika model proses yang disusun dari sejumlah model yang sangat banyak dan hal itu perlu untuk dikonsultasikan, diubah dan digunakan kembali dalam waktu yang cukup lama. APROMORE menawarkan serangkaian fitur untuk memelihara, menganalisis, dan mengeksploitasi konten model proses.

APROMORE mendukung berbagai bahasa pemodelan proses, termasuk EPC, BPMN, YAWL dan BPEL. APROMORE juga menawarkan *open-platform* dan dapat diperluas untuk membangun fitur-fitur canggih yang secara khusus menangani koleksi model proses besar, seperti pencarian kesamaan, penggabungan proses, dan permintaan model yang efisien. APROMORE merupakan *platform* analisis proses *online open-source* dan dapat dikembangkan dengan kemampuan canggih untuk mengelola dan menganalisis koleksi model proses besar. APROMORE diimplementasikan melalui *three-layered SOA* dan digunakan melalui internet. Implementasi tersebut dapat dilihat pada gambar 2.8 (Fauvet et al., 2010).



Gambar 2. 8 *Three-Layered SOA in APROMORE*

*Enterprise Layer* meng-host *manager*, yaitu layanan perusahaan publik yang memperlihatkan semua fitur repositori melalui operasi *web service* untuk integrasi dengan aplikasi pihak ketiga yang salah satunya adalah *BPM System*. *Intermediary layer* meng-host *Canonizer* yang merupakan adaptor yang bertanggung jawab untuk (membatalkan) model proses kanonisasi ketika mereka diimpor / diekspor ke / dari repositori. *Toolbox* adalah *facade* atas operasi lanjutan yang dapat dilakukan pada koleksi model proses yang disimpan. Akses ke model ini dicapai melalui layanan *data access* di *Basic Layer*, yang merangkum operasi pusat data untuk membaca / menulis data berdasarkan permintaan yang dibuat oleh layanan lain. Akhirnya, layanan *Access Control* mengontrol aspek keamanan seperti otentikasi dan otorisasi pengguna. (Fauvet et al., 2010)

### **2.9. Workflow Analyzer (Woflan)**

Dalam menggunakan model proses juga memungkinkan adanya kesalahan dalam proses *workflow* yang telah dibuat. Teknologi pengelolaan *workflow* menjanjikan solusi fleksibel yang memfasilitasi pembuatan proses bisnis baru dan modifikasi yang sudah ada dengan mudah. Namun, sebagian besar produk aplikasi *workflow* saat ini memungkinkan proses yang salah untuk dimasukkan ke dalamnya. Dengan kata lain, aplikasi tersebut tidak memiliki mekanisme verifikasi yang tepat dalam alat definisi proses dari *workflow* yang dibuat atau dimodifikasi. Dalam hal ini, Woflan merupakan salah satu *tools* terbaik untuk melakukan verifikasi dan koreksi terhadap *workflow* tersebut.

Woflan merupakan *workflow analyzer* yang digunakan sebagai verifikasi pada proses *workflow*. Woflan hanya dapat digunakan pada *workflow* berbasis Petri Net. Menggunakan teknik berbasis Petri-net, Woflan mendiagnosis definisi proses sebelum diproduksi. Dalam proses analisis tersebut, Woflan akan memberikan informasi mengenai *structural properties*. Selanjutnya, Woflan akan memandu pemodel atau pembuat *workflow* tersebut untuk menemukan dan memperbaiki kesalahan yang mungkin terjadi. Dengan adanya proses tersebut, kesalahan-kesalahan seperti *deadlocks* dan *livelocks* BPMN dapat terdeteksi.

### **2.10. Produk Terkait**

Telah banyak *BPM System* di internet yang dapat diunduh, baik itu yang *Open Source* maupun *Commercial*. Terdapat *survey* perbandingan antara Bonita dan

Camunda dalam segi *Modelling, Design, Deployment, Monitoring, Analysis* dan *Execution* (Meidan et al., 2017). Serta Bizagi menjadi pembanding ketiga untuk memperluas obyek perbandingan. Perbandingan tersebut dirangkum dalam tabel 2.1 berikut.

Tabel 2. 1 Perbandingan *BPM System* Bonita, Camunda dan Bizagi

| <i>Category</i>  |   | <i>BPM System</i>  |                                      |   |                               |
|------------------|---|--|--------------------------------------|---|-------------------------------|
|                  |   | <b>Bonita</b>  | <b>Camunda</b>                       | <b>Bizagi</b>   | <b>Proposed</b>               |
| <b>Modeling</b>  | <i>Language</i>                           | <i>BPMN 2.0</i>  | <i>BPMN 2.0</i>                      | <i>BPMN 2.0</i>   | <i>BPMN 2.0</i>               |
|                  | <i>Interoperability and compatibility</i> | <i>I : jBPM XML 3.2, BPMN 2.0, XPDL 1.0<br/>E : BPMN 2.0</i> | <i>I : BPMN 2.0<br/>E : BPMN 2.0</i> | <i>I : Visio, XPDL 2.1, BPMN 2.0<br/>E : XPDL 2.1, Image, Visio, BPMN 2.0</i> | <i>I: XPDL 1.0</i>            |
|                  | <i>Reuse</i>                              | <i>✗</i>   | <i>✗</i>                             | <i>✓</i>  | <i>✓</i>                      |
|                  | <i>View</i>                               | <i>All</i>   | <i>All</i>                           | <i>All</i>  | <i>Flow</i>                   |
|                  | <i>Business Rules</i>                     | <i>Partially</i>   | <i>Partially</i>                     | <i>Independently</i>  | <i>Independently</i>          |
|                  | <i>Modeling PPI</i>                       | <i>✗</i>   | <i>Partially</i>                     | <i>✗</i>  | <i>✗</i>                      |
|                  | <i>Generate Process Documentation</i>     | <i>✗</i>   | <i>✗</i>                             | <i>✓</i>  | <i>✗</i>                      |
| <b>Designing</b> | <i>Language</i>                           | <i>Java</i>  | <i>Java, Javascript</i>              | <i>Java, Javascript</i>   | <i>PHP</i>                    |
|                  | <i>UI</i>                                 | <i>Auto and Manual</i>                                       | <i>Auto and Manual</i>               | <i>Auto and Manual</i>  | <i>✗</i>                      |
|                  | <i>Describing Roles</i>                   | <i>Names/descriptions</i>                                    | <i>Names/descriptions</i>            | <i>Names/description/type</i>   | <i>Names/description/type</i> |
|                  | <i>Import Organizational Structure</i>    | <i>✓</i>   | <i>✓</i>                             | <i>✓</i>  | <i>✗</i>                      |
|                  | <i>Assigning User to a Role</i>           | <i>Manual</i>  | <i>Manual</i>                        | <i>Manual</i>   | <i>Manual</i>                 |

|                               |  |                 |                      |                      |                      |
|-------------------------------|--|-----------------|----------------------|----------------------|----------------------|
|                               | <i>Adding SLA and Linking SLA KPIs</i>             | x               | x                    | ✓                    | ✓                    |
|                               | <i>Exception Handling</i>                          | ✓               | ✓                    | ✓                    | ✓                    |
|                               | <i>Transaction Control</i>                         | x               | ✓                    | ✓                    | ✓                    |
|                               | <i>Translation Into Executable Models</i>          | Auto            | Auto                 | Auto                 | Auto                 |
|                               | <i>Bp Execution Languages</i>                      | x               | x                    | x                    | x                    |
| <b>Deployment</b>             | <i>Distributed Execution</i>                       | x               | x                    | Partially            | x                    |
|                               | <i>Integration Into Other Systems and Services</i> | More than 80    | REST and Web Service | REST and Web Service | REST and Web Service |
| <b>Execution</b>              | <i>Version Management of BP Models</i>             | ✓               | ✓                    | ✓                    | ✓                    |
|                               | <i>Calendar Management</i>                         | Google Calendar | x                    | ✓                    | x                    |
|                               | <i>Informing Users</i>                             | ✓               | ✓                    | ✓                    | ✓                    |
|                               | <i>Document Management</i>                         | ✓               | x                    | ✓                    | x                    |
| <b>Monitoring and Control</b> | <i>Technical Monitoring and Control</i>            | ✓               | x                    | ✓                    | x                    |
|                               | <i>Business monitoring bam</i>                     | Partially       | ✓                    | ✓                    | x                    |
|                               | <i>Change the role or resource</i>                 | ✓               | ✓                    | ✓                    | ✓                    |
|                               | <i>Changing business rules</i>                     | x               | x                    | ✓                    | ✓                    |

|                        |                                      |                  |                  |                                     |                     |
|------------------------|--------------------------------------|------------------|------------------|-------------------------------------|---------------------|
|                        | <i>Optimized execution</i>           | <i>x</i>         | <i>x</i>         | <i>✓</i>                            | <i>x</i>            |
|                        | <i>Ability to deal with failures</i> | <i>Partially</i> | <i>✓</i>         | <i>✓</i>                            | <i>✓</i>            |
|                        | <i>Changing the workload balance</i> | <i>✓</i>         | <i>✓</i>         | <i>✓</i>                            | <i>x</i>            |
|                        | <i>Dashboards and reports</i>        | <i>x</i>         | <i>✓</i>         | <i>✓</i>                            | <i>x</i>            |
|                        | <i>Detail levels</i>                 | <i>x</i>         | <i>Partially</i> | <i>✓</i>                            | <i>✓</i>            |
|                        | <i>Different views of monitoring</i> | <i>x</i>         | <i>Partially</i> | <i>Partially</i>                    | <i>Partially</i>    |
| <b><i>Analysis</i></b> | <i>Process verification</i>          | <i>✓</i>         | <i>✓</i>         | <i>✓</i>                            | <i>x</i>            |
|                        | <i>Process simulation</i>            | <i>✓</i>         | <i>x</i>         | <i>✓</i>                            | <i>x</i>            |
|                        | <i>Analysis historical data</i>      | <i>Partially</i> | <i>Log Files</i> | <i>Operational Data Store (ODS)</i> | <i>Activity Log</i> |
|                        | <i>Suggestions on improvement</i>    | <i>Partially</i> | <i>x</i>         | <i>✓</i>                            | <i>x</i>            |
|                        | <i>Bi and process mining tools</i>   | <i>Partially</i> | <i>Partially</i> | <i>✓</i>                            | <i>x</i>            |

Berdasarkan tabel 2.1, penulis akan membangun sebuah *BPM System* yang *reusable* dan *flexible*. *BPM System* yang akan dibangun menggunakan BPMN untuk model proses bisnisnya yang sebelumnya telah diekspor ke dalam *format XPD*L, sehingga aplikasi yang telah dibuat dengan *BPM System* dapat disesuaikan dengan bisnis proses yang telah ditentukan. PHP digunakan sebagai basis bahasa program dalam sistem ini dimana Bonita, Camunda dan Bizagi menggunakan java dan javascript sebagai basis bahasa program.

*Flexibility* pada sistem ini mempunyai arti bahwa meskipun bisnis proses berubah dalam waktu yang singkat, sistem dapat menyesuaikan dengan perubahan tersebut. Perubahan yang dimaksud dapat berupa penggantian urutan proses, penambahan proses, pengurangan proses, dan lain sebagainya.

Sedangkan *reusability* pada sistem ini mempunyai arti bahwa *method* atau fungsi yang berada dalam aplikasi tersebut dapat digunakan oleh aplikasi lain. Oleh karena itu, sistem ini juga akan menyediakan *web service repository* yang mana pada Bonita, Camunda dan Bizagi tidak menyediakan *repository* tersebut. Mereka hanya menyediakan *repository* untuk aplikasi yang sedang dibangun.

## **BAB III**

### **METODE PENELITIAN**

Pada bab ini menjelaskan terkait metodologi yang akan digunakan sebagai panduan untuk menyelesaikan penulisan tugas akhir ini. Metodologi penulisan ini dimulai dari analisis permasalahan uji coba sistem.

#### **3.1. Survei Literatur**

Saat ini, BPM telah menjadi salah satu pemodelan konsep yang paling populer. Sebuah model proses dideskripsikan sebagai notasi gambar yang menjelaskan bagaimana sebuah prosedur dapat berjalan dengan tugas yang berbeda-beda dengan menggunakan sumber daya serta obyek yang dapat dimanipulasi. Model proses dapat digunakan untuk sebuah pengembangan teknologi informasi dan juga dapat digunakan untuk tujuan bisnis.

APROMORE merupakan sebuah platform analisis proses bisnis menawarkan fitur untuk mengelola dan menganalisis koleksi model proses yang besar seperti impor dan ekspor model proses serta pengelolaan versi aplikasi. Penulis merefensikan fitur tersebut ke dalam sistem yang akan dibangun, yaitu impor model proses dan pengelolaan versi aplikasi.

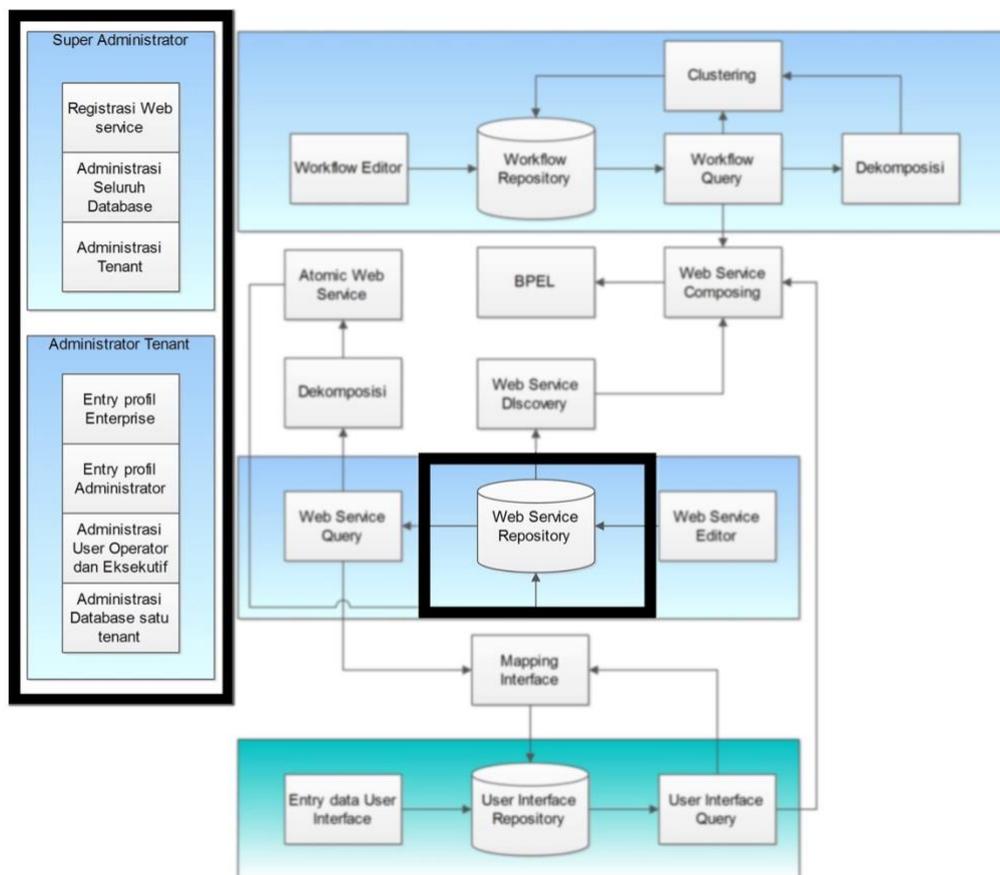
Dalam pembuatan *workflow* proses bisnis juga tidak dapat asal-asalan karena dapat membahayakan bisnis itu sendiri. Woflan hadir sebagai alat untuk menganalisis *workflow* agar kesalahan-kesalahan seperti *deadlocks* dan *livelocks* dapat terdeteksi melalui informasi *structural properties* yang didapatkan. Sehingga *workflow* dapat diperbaiki pada bagian yang bermasalah tersebut. Penulis merefensikan fitur tersebut ke dalam sistem yang akan dibangun, yaitu pemberian informasi *structural properties* pada sistem yang akan menjadi sebuah definisi pada masing-masing proses. Sehingga proses dapat dipetakan dengan *web service* yang telah disediakan.

#### **3.2. Merancang BPM System**

Dalam penulisan ini, sistem yang akan dibangun bersifat *flexible* dan *reusable*. *Flexibility* dan *reusability* termasuk dalam kriteria yang dimaksud oleh *framework* dari McCall. McCall mengartikan *flexibility* sebagai kemampuan suatu sistem dalam beradaptasi terhadap modifikasi atau perubahan. Sedangkan

*reusability* diartikan sebagai kemampuan suatu sistem dalam penggunaan kembali suatu fitur atau fungsi dari atau untuk sistem lain.

Pada proses merancang *BPM System* dilakukan dengan 2 perancangan utama, yaitu Arsitektur Data dan Arsitektur Aplikasi. Arsitektur Data merupakan perancangan *database* sedangkan Arsitektur Aplikasi merupakan perancangan alur kerja serta tampilan aplikasi. Perancangan-perancangan tersebut merujuk pada arsitektur BPMS dengan skala besar. Penulis merujuk pada bagian fitur dari *super administrator* dan *administrator tenant* serta *web service repository* seperti yang ditunjukkan pada gambar 3.1 berikut.



Gambar 3. 1 Arsitektur BPMS skala besar

Perancangan dimulai dari *design database* yang di dalamnya dapat menyimpan data-data yang diperlukan oleh *BPM System* yang akan dibangun meliputi:

- *Data user*
- *Data web service*

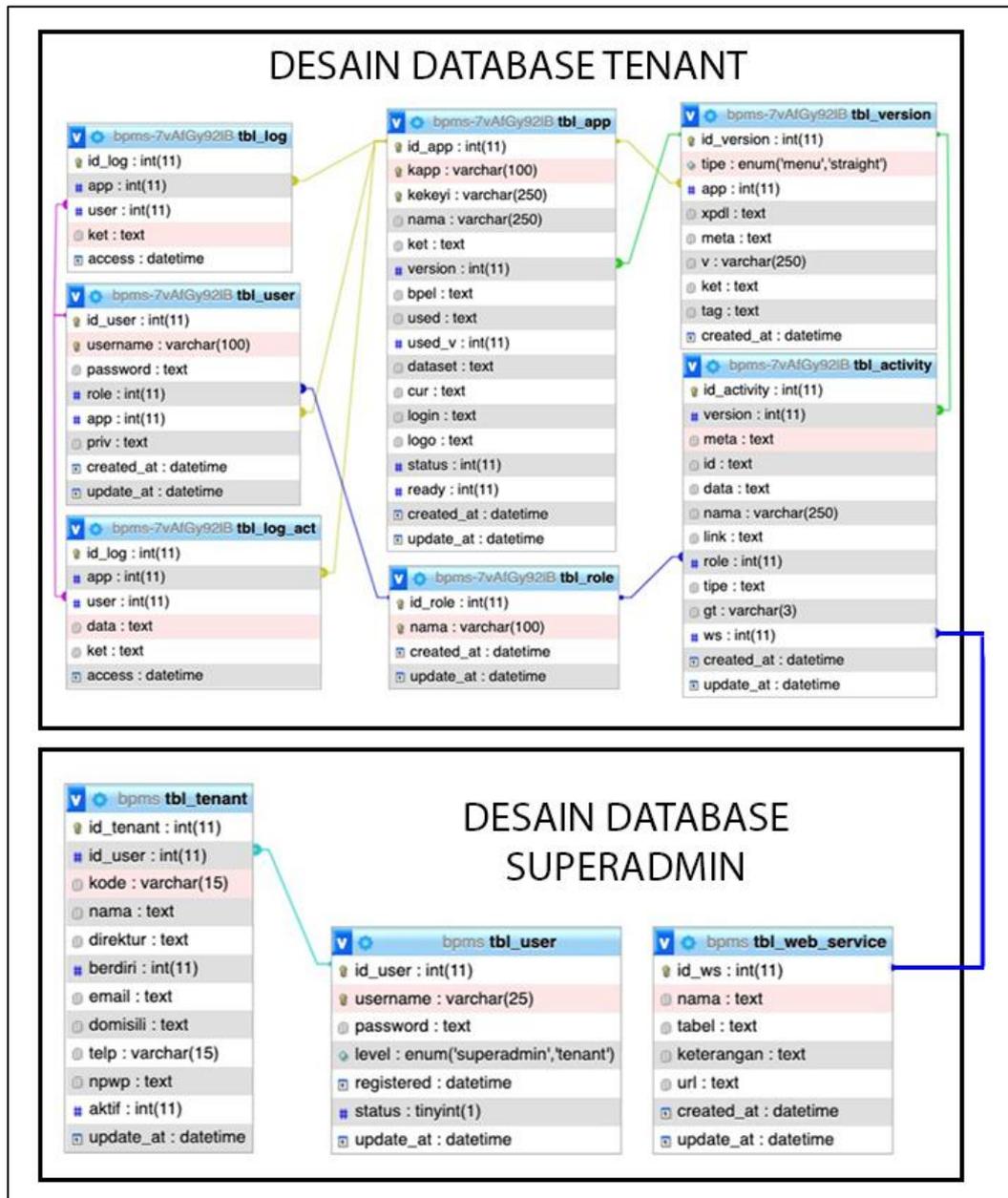
- *Data tenant*
- *Data role user*
- *Data applications*

Perancangan *flowchart* pada setiap fitur yang akan dibangun pada *BPM System*. Penggambaran alur proses aplikasi bertujuan agar mempermudah implementasi, pengujian dan analisis dari *BPM System*. *Flowchart* tersebut meliputi:

- *User management*
- *Web service management*
- *Tenant management*
- *Application management*
- *Role management*

### **3.2.1. Arsitektur Data**

*Database BPM System* yang akan dibangun mempunyai skema ERD seperti tabel 3.2 berikut.



Gambar 3. 2 Entity Relational Diagram BPM System

Pada sistem dengan login sebagai *tenant*, terdapat fitur untuk membuat aplikasi yang mana proses-prosesnya berdasarkan file XPDL yang diunggah. Pada pembuatan aplikasi tersebut memiliki tabel masing-masing. Tabel-tabel tersebut akan ditambahkan secara otomatis sesuai dengan apa yang telah disediakan oleh sistem. Sehingga tabel-tabel tersebut tidak masuk pada desain *database* di atas. Berikut adalah rincian tabel pada *database BPM System*.

- Tabel user (pada *superadmin*)

Tabel ini merupakan tabel untuk menyimpan data login pengguna sistem. Terdapat 2 *level user* yang menggunakan tipe data *enum*, yaitu *superadmin* dan *tenant*. Jika *level user* tersebut adalah *tenant*, maka akan ada detail info dari *tenant* tersebut pada tabel tenant yang dijelaskan pada tabel 3.2. Tabel *user* dapat dilihat pada tabel 3.1.

Tabel 3. 1 Struktur tabel user

| Nama       | Jenis                               | Nilai Default              | Keterangan         |
|------------|-------------------------------------|----------------------------|--------------------|
| id_user    | <i>int(11)</i>                      |                            | <i>Primary Key</i> |
| username   | <i>varchar(25)</i>                  |                            | Not Null           |
| password   | <i>text</i>                         |                            | Not Null           |
| level      | <i>enum('superadmin', 'tenant')</i> |                            | Not Null           |
| regsitered | <i>datetime</i>                     | <i>current_timestamp()</i> | Not Null           |
| status     | <i>tinyint(1)</i>                   | 0                          | Not Null           |
| update_at  | <i>datetime</i>                     | NULL                       |                    |

- Tabel tenant

Tabel ini merupakan tabel untuk menyimpan detail info dari pengguna sistem dengan *level user tenant*. Setiap *tenant* memiliki kode unik masing-masing yang mana kode tersebut digunakan sebagai identitas *database tenant* terkait. Tabel tenant dapat dilihat pada tabel 3.2.

Tabel 3. 2 Struktur tabel tenant

| Nama      | Jenis              | Nilai Default | Keterangan         |
|-----------|--------------------|---------------|--------------------|
| id_tenant | <i>int(11)</i>     |               | <i>Primary Key</i> |
| Id_user   | <i>int(11)</i>     |               | <i>Foreign Key</i> |
| kode      | <i>varchar(15)</i> |               | Not Null           |
| nama      | <i>text</i>        |               | Not Null           |
| direktur  | <i>text</i>        |               | Not Null           |
| berdiri   | <i>int(11)</i>     |               | Not Null           |
| email     | <i>text</i>        |               | Not Null           |
| domisili  | <i>text</i>        |               | Not Null           |
| telp      | <i>int(12)</i>     |               | Not Null           |
| npwp      | <i>text</i>        |               | Not Null           |
| aktif     | <i>int(11)</i>     | 0             | Not Null           |
| update_at | <i>datetime</i>    | NULL          |                    |

- Tabel *web\_service*

Tabel ini merupakan tabel untuk menyimpan data *web service* atau pada penelitian ini disebut sebagai *web service repository*. Data yang disimpan terdapat kolom tabel yang mana berisi daftar tabel yang harus digunakan dalam penggunaan *web service* tersebut. Terdapat juga kolom *url* yang merupakan *endpoint* dari *web service* dalam mengaksesnya. Tabel *web\_service* dapat dilihat pada tabel 3.3.

Tabel 3. 3 Struktur tabel *web\_service*

| <b>Nama</b>       | <b>Jenis</b>    | <b>Nilai Default</b>       | <b>Keterangan</b>  |
|-------------------|-----------------|----------------------------|--------------------|
| <i>id_ws</i>      | <i>int(11)</i>  |                            | <i>Primary Key</i> |
| <i>nama</i>       | <i>text</i>     |                            | Not Null           |
| <i>tabel</i>      | <i>text</i>     | NULL                       |                    |
| <i>keterangan</i> | <i>text</i>     |                            | Not Null           |
| <i>url</i>        | <i>text</i>     |                            | Not Null           |
| <i>created</i>    | <i>datetime</i> | <i>current_timestamp()</i> | Not Null           |
| <i>update_at</i>  | <i>datetime</i> | NULL                       |                    |

- Tabel *app*

Tabel ini merupakan tabel untuk menyimpan data aplikasi yang dibuat oleh *tenant*. Data yang disimpan terdapat kolom *version* yang mana berisi versi aplikasi yang digunakan. Terdapat juga kolom *bpel* yang merupakan alur proses pada aplikasi berdasarkan XPDL yang diunggah. Tabel *app* dapat dilihat pada tabel 3.4.

Tabel 3. 4 Struktur tabel *app*

| <b>Nama</b>           | <b>Jenis</b>        | <b>Nilai Default</b> | <b>Keterangan</b>  |
|-----------------------|---------------------|----------------------|--------------------|
| <b><i>id_app</i></b>  | <i>int(11)</i>      |                      | <i>Primary Key</i> |
| <b><i>kapp</i></b>    | <i>varchar(100)</i> |                      | Not Null           |
| <b><i>kekeyi</i></b>  | <i>varchar(250)</i> |                      | Not Null           |
| <b><i>nama</i></b>    | <i>varchar(250)</i> |                      | Not Null           |
| <b><i>ket</i></b>     | <i>text</i>         |                      | Not Null           |
| <b><i>version</i></b> | <i>int(11)</i>      |                      | <i>Foreign Key</i> |
| <b><i>bpel</i></b>    | <i>text</i>         |                      | Not Null           |
| <b><i>used</i></b>    | <i>text</i>         | NULL                 |                    |
| <b><i>used_v</i></b>  | <i>int(11)</i>      | NULL                 |                    |
| <b><i>dataset</i></b> | <i>text</i>         | NULL                 |                    |
| <b><i>cur</i></b>     | <i>text</i>         | NULL                 |                    |

|                  |          |                     |          |
|------------------|----------|---------------------|----------|
| <b>login</b>     | text     | NULL                |          |
| <b>logo</b>      | text     |                     | Not Null |
| <b>status</b>    | int(11)  | 0                   | Not Null |
| <b>ready</b>     | int(11)  | 0                   | Not Null |
| <b>created</b>   | datetime | current_timestamp() | Not Null |
| <b>update_at</b> | datetime | NULL                |          |

- Tabel version

Tabel ini merupakan tabel untuk menyimpan versi-versi aplikasi yang dibuat oleh *tenant*. Data yang disimpan terdapat kolom *version* yang mana berisi versi aplikasi yang digunakan. Terdapat juga kolom *bpel* yang merupakan alur proses pada aplikasi berdasarkan XPDL yang diunggah. Tabel version dapat dilihat pada tabel 3.5.

Tabel 3. 5 Struktur tabel version

| <b>Nama</b>       | <b>Jenis</b>              | <b>Nilai Default</b> | <b>Keterangan</b> |
|-------------------|---------------------------|----------------------|-------------------|
| <b>id_version</b> | int(11)                   |                      | Primary Key       |
| <b>tipe</b>       | enum('menu', 'stratight') | straight             | Not Null          |
| <b>app</b>        | int(11)                   |                      | Not Null          |
| <b>xpdl</b>       | text                      |                      | Not Null          |
| <b>meta</b>       | text                      |                      | Not Null          |
| <b>v</b>          | varchar(250)              |                      | Not Null          |
| <b>ket</b>        | text                      |                      | Not Null          |
| <b>tag</b>        | text                      |                      | Not Null          |
| <b>created_at</b> | datetime                  | current_timestamp()  | Not Null          |

- Tabel activity

Tabel ini merupakan tabel untuk menyimpan aktivitas-aktivitas atau proses-proses dari aplikasi yang dibuat oleh *tenant* berdasarkan XPDL yang diunggah. Data yang disimpan terdapat kolom *role* yang mana berisi data akses berdasarkan akses pengguna aplikasi. Terdapat juga kolom *ws* yang merupakan *web service* pada aplikasi berdasarkan pemetaan XPDL dengan *web service*. Terdapat juga kolom *version* yang mana proses tersebut terikat pada versi yang sedang digunakan oleh aplikasi. Tabel activity dapat dilihat pada tabel 3.6.

Tabel 3. 6 Struktur tabel activity

| Nama               | Jenis        | Nilai Default       | Keterangan  |
|--------------------|--------------|---------------------|-------------|
| <b>id_activity</b> | int(11)      |                     | Primary Key |
| <b>version</b>     | int(11)      |                     | Foreign Key |
| <b>meta</b>        | text         |                     | Not Null    |
| <b>id</b>          | text         |                     | Not Null    |
| <b>data</b>        | text         | NULL                |             |
| <b>nama</b>        | varchar(250) |                     | Not Null    |
| <b>link</b>        | text         |                     | Not Null    |
| <b>role</b>        | int(11)      |                     | Not Null    |
| <b>tipe</b>        | text         |                     | Not Null    |
| <b>gt</b>          | varchar(3)   |                     | Not Null    |
| <b>ws</b>          | int(11)      |                     | Not Null    |
| <b>created_at</b>  | datetime     | current_timestamp() | Not Null    |
| <b>update_at</b>   | datetime     | NULL                |             |

- Tabel user (pada *tenant*)

Tabel ini merupakan tabel untuk menyimpan data login pengguna aplikasi yang dibuat oleh *tenant*. Data yang disimpan terdapat kolom *role* dan *app* yang mana berisi akses pengguna aplikasi yang terikat pada aplikasi tertentu. Tabel user dapat dilihat pada tabel 3.7.

Tabel 3. 7 Struktur tabel user (pada *tenant*)

| Nama              | Jenis        | Nilai Default       | Keterangan  |
|-------------------|--------------|---------------------|-------------|
| <b>id_user</b>    | int(11)      |                     | Primary Key |
| <b>username</b>   | varchar(100) |                     | Not Null    |
| <b>password</b>   | text         |                     | Not Null    |
| <b>role</b>       | int(11)      |                     | Foreign Key |
| <b>app</b>        | int(11)      |                     | Foreign Key |
| <b>priv</b>       | text         |                     | Not Null    |
| <b>created_at</b> | datetime     | current_timestamp() | Not Null    |
| <b>update_at</b>  | datetime     | NULL                |             |

- Tabel role

Tabel ini merupakan tabel untuk menyimpan data akses pengguna aplikasi yang dibuat oleh *tenant*. Data akses ini merupakan hak akses pengguna pada aplikasi tertentu. Tabel user dapat dilihat pada tabel 3.8.

Tabel 3. 8 Struktur tabel role

| Nama              | Jenis        | Nilai Default       | Keterangan  |
|-------------------|--------------|---------------------|-------------|
| <b>id_role</b>    | int(11)      |                     | Primary Key |
| <b>nama</b>       | varchar(100) |                     | Not Null    |
| <b>created_at</b> | datetime     | current_timestamp() | Not Null    |
| <b>update_at</b>  | datetime     | NULL                |             |

- Tabel log

Tabel ini merupakan tabel untuk menyimpan data riwayat aktivitas pengguna sistem, bukan pengguna aplikasi. Tabel user dapat dilihat pada tabel 3.9.

| Nama          | Jenis    | Nilai Default       | Keterangan  |
|---------------|----------|---------------------|-------------|
| <b>id_log</b> | int(11)  |                     | Primary Key |
| <b>app</b>    | int(11)  |                     | Foreign Key |
| <b>user</b>   | int(11)  |                     | Foreign Key |
| <b>ket</b>    | text     |                     | Not Null    |
| <b>access</b> | datetime | current_timestamp() |             |

- Tabel log\_act

Tabel ini merupakan tabel untuk menyimpan data riwayat aktivitas pengguna aplikasi tertentu. Tabel user dapat dilihat pada tabel 3.10.

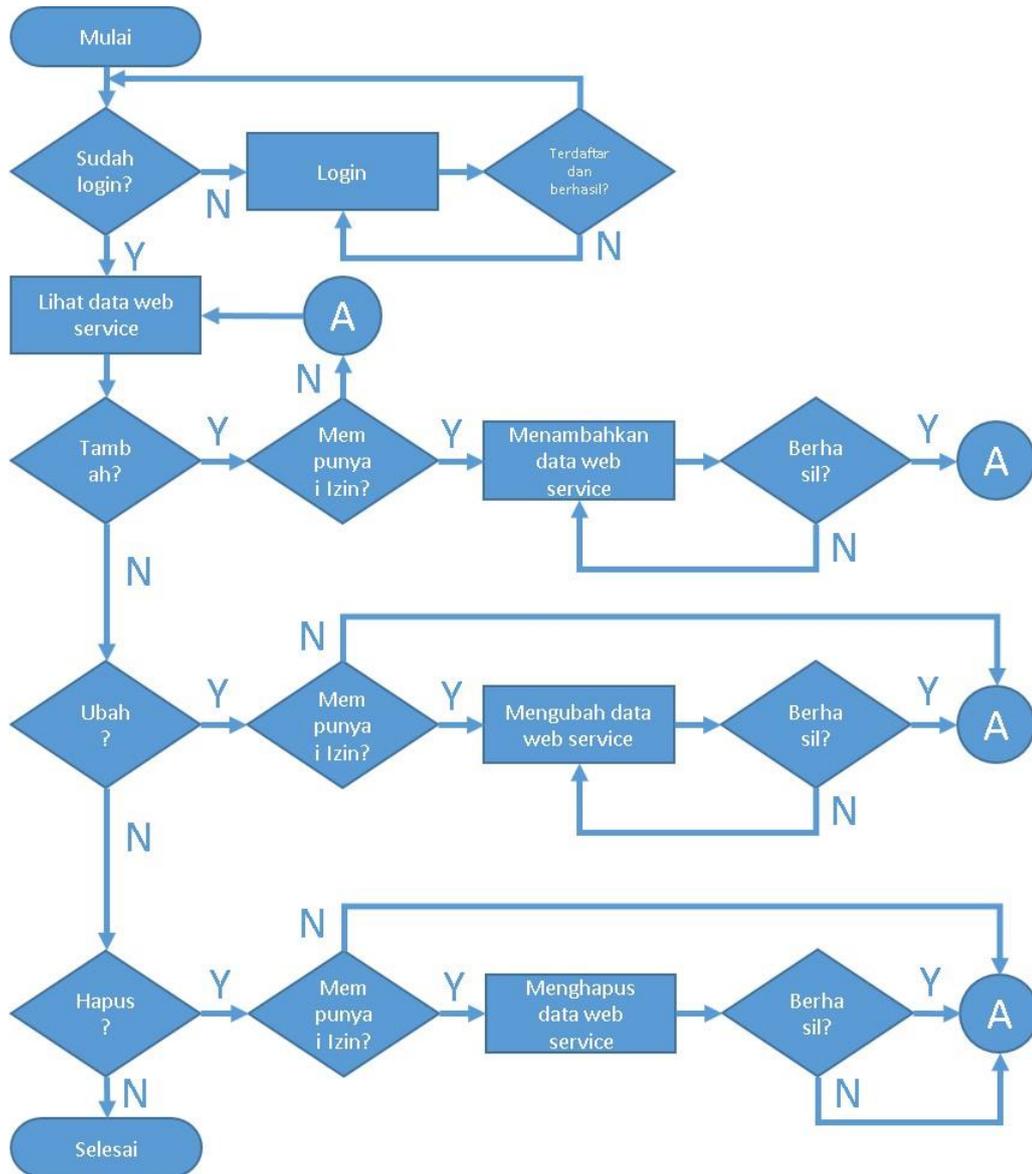
| Nama          | Jenis    | Nilai Default       | Keterangan  |
|---------------|----------|---------------------|-------------|
| <b>id_log</b> | int(11)  |                     | Primary Key |
| <b>app</b>    | int(11)  |                     | Foreign Key |
| <b>user</b>   | int(11)  |                     | Foreign Key |
| <b>data</b>   | text     |                     | Not Null    |
| <b>ket</b>    | text     |                     | Not Null    |
| <b>access</b> | datetime | current_timestamp() |             |

### 3.2.2. Arsitektur Aplikasi

Pada arsitektur aplikasi menjelaskan *workflow* dari sistem yang dibangun, yaitu perancangan *flowchart*. Perancangan *flowchart BPM System* adalah sebagai berikut:

- Manajemen *Web Service*

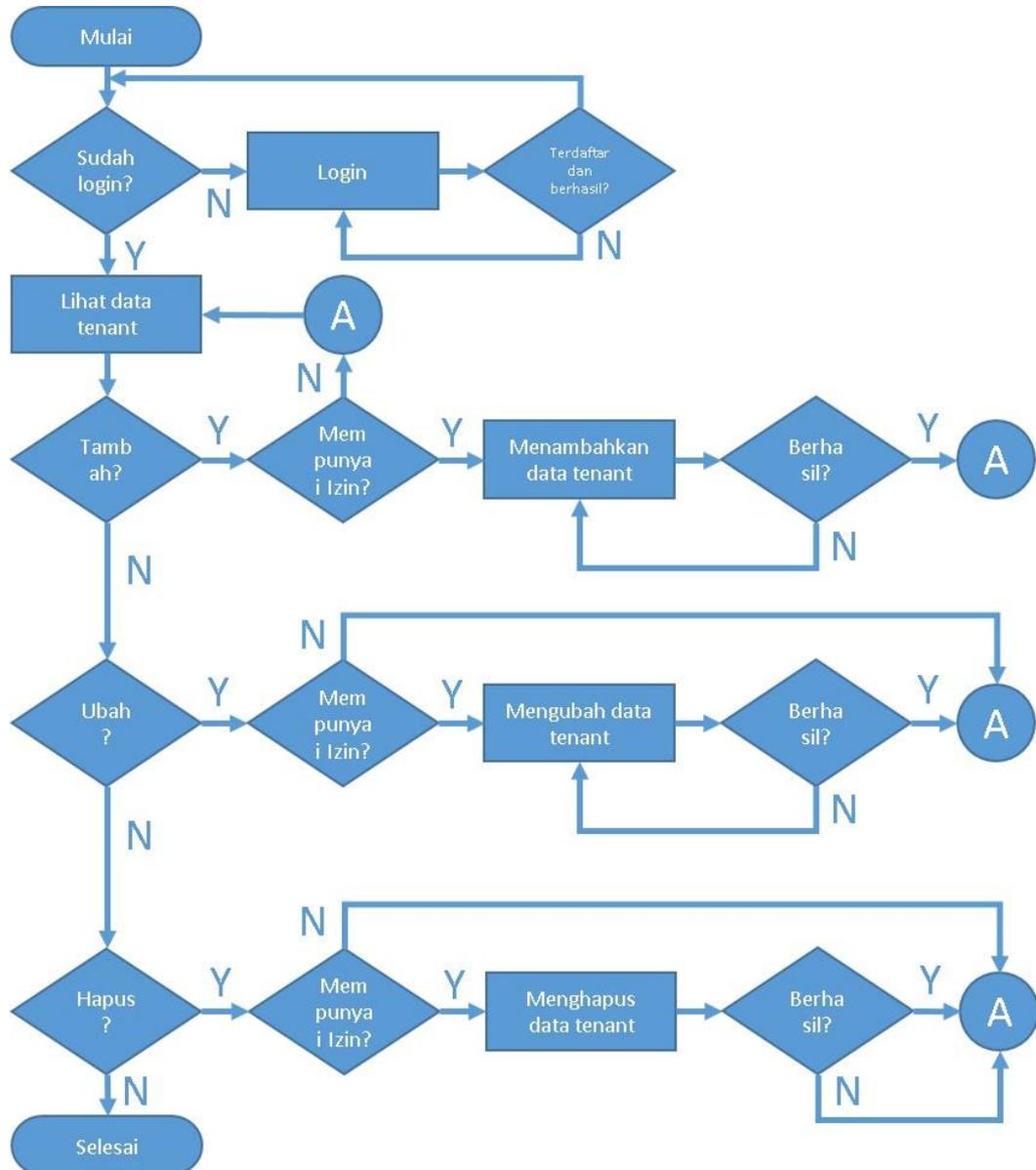
Flowchart manajemen web service dapat dilihat pada gambar 3.3.



Gambar 3.3 Flowchart web service management

- Manajemen Tenant

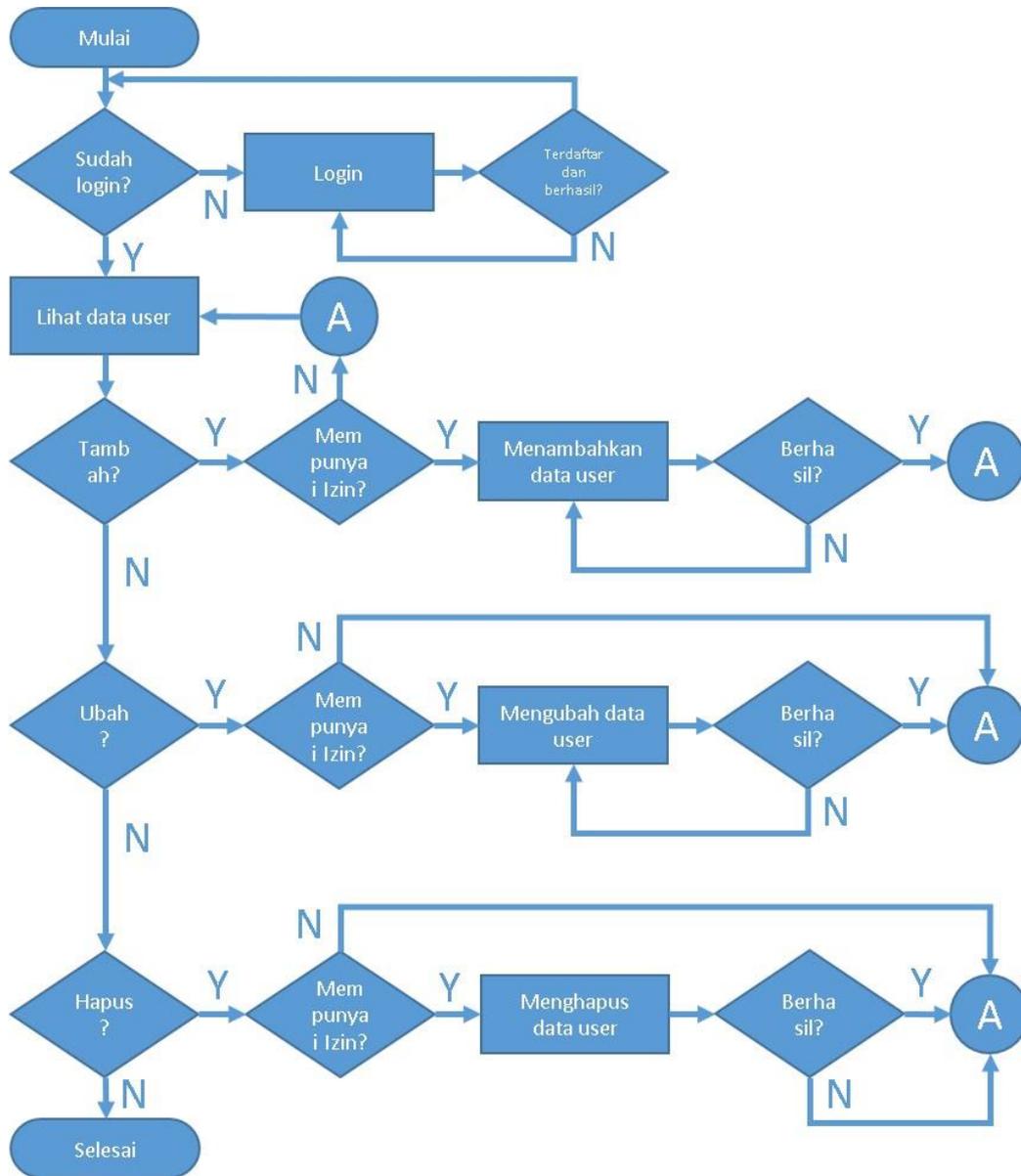
Flowchart manajemen tenant dapat dilihat pada gambar 3.4.



Gambar 3. 4 Flowchart tenant management

- Manajemen User

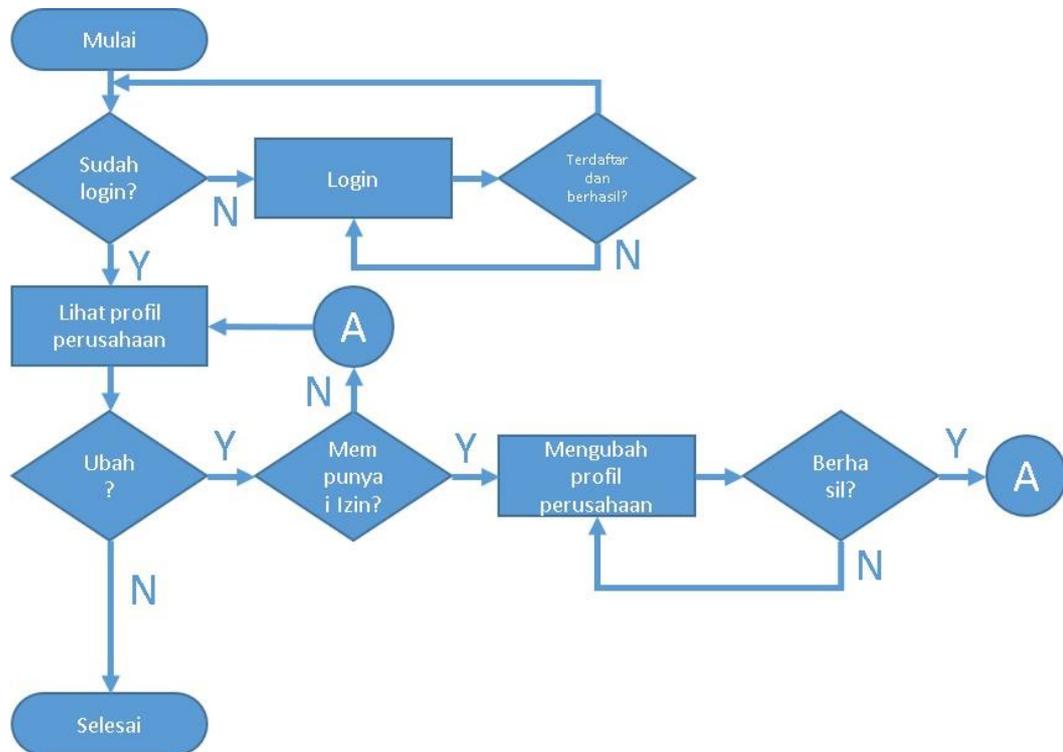
Flowchart manajemen user dapat dilihat pada gambar 3.5.



Gambar 3. 5 Flowchart user management

- *Entry Profil Perusahaan*

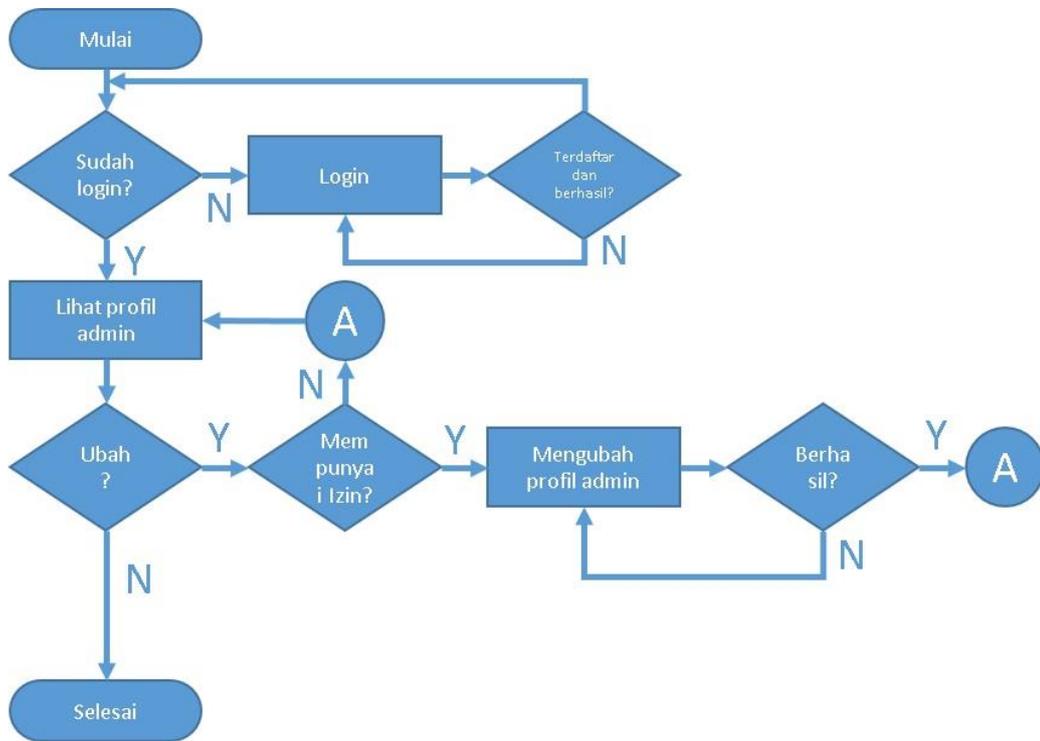
*Flowchart entry profil perusahaan dapat dilihat pada gambar 3.6.*



Gambar 3. 6 *Flowchart entry profil perusahaan*

- *Entry Profil Admin*

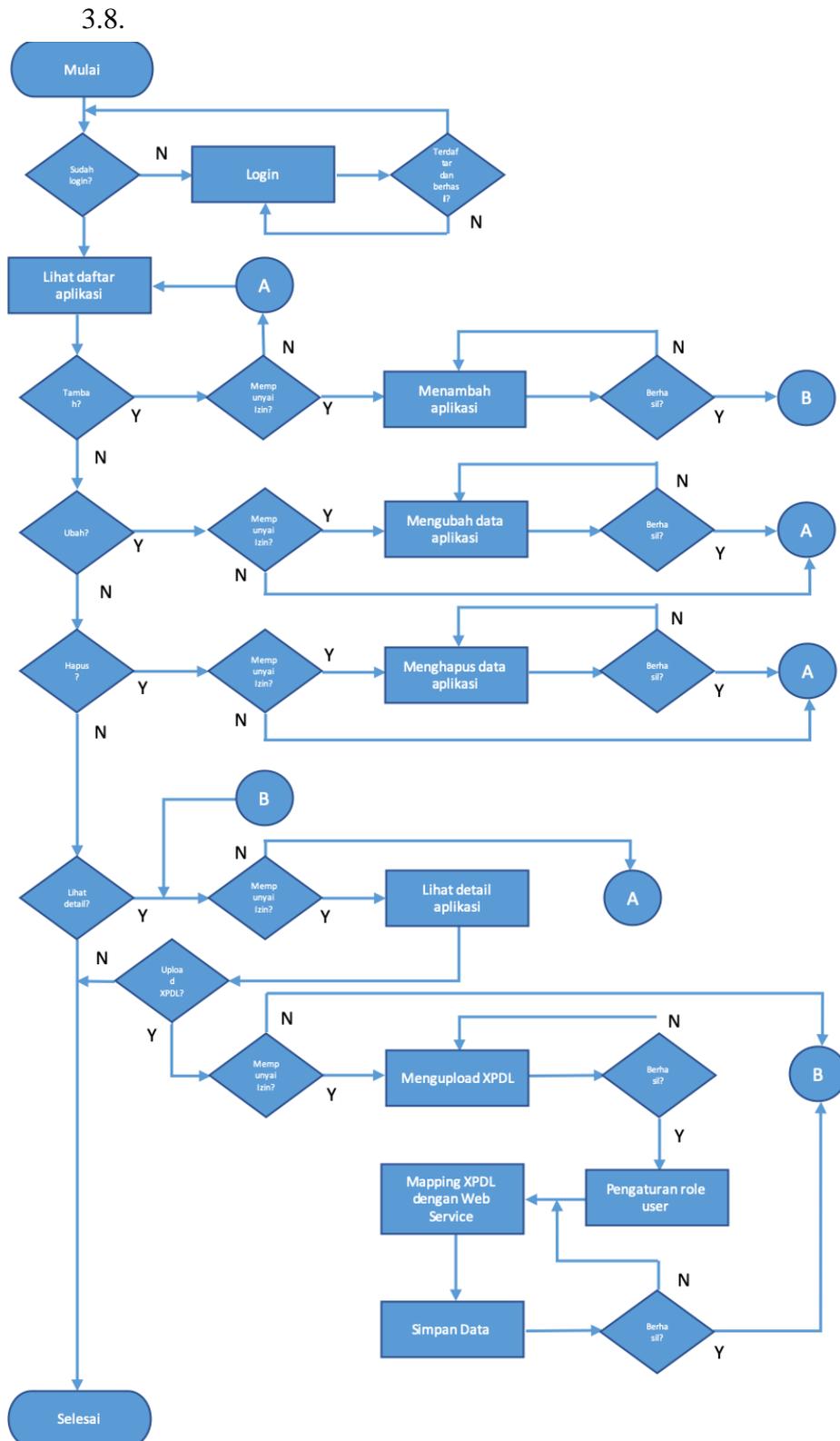
*Flowchart entry profil admin dapat dilihat pada gambar 3.7.*



Gambar 3. 7 *Flowchart entry profil admin*

- *Application management*

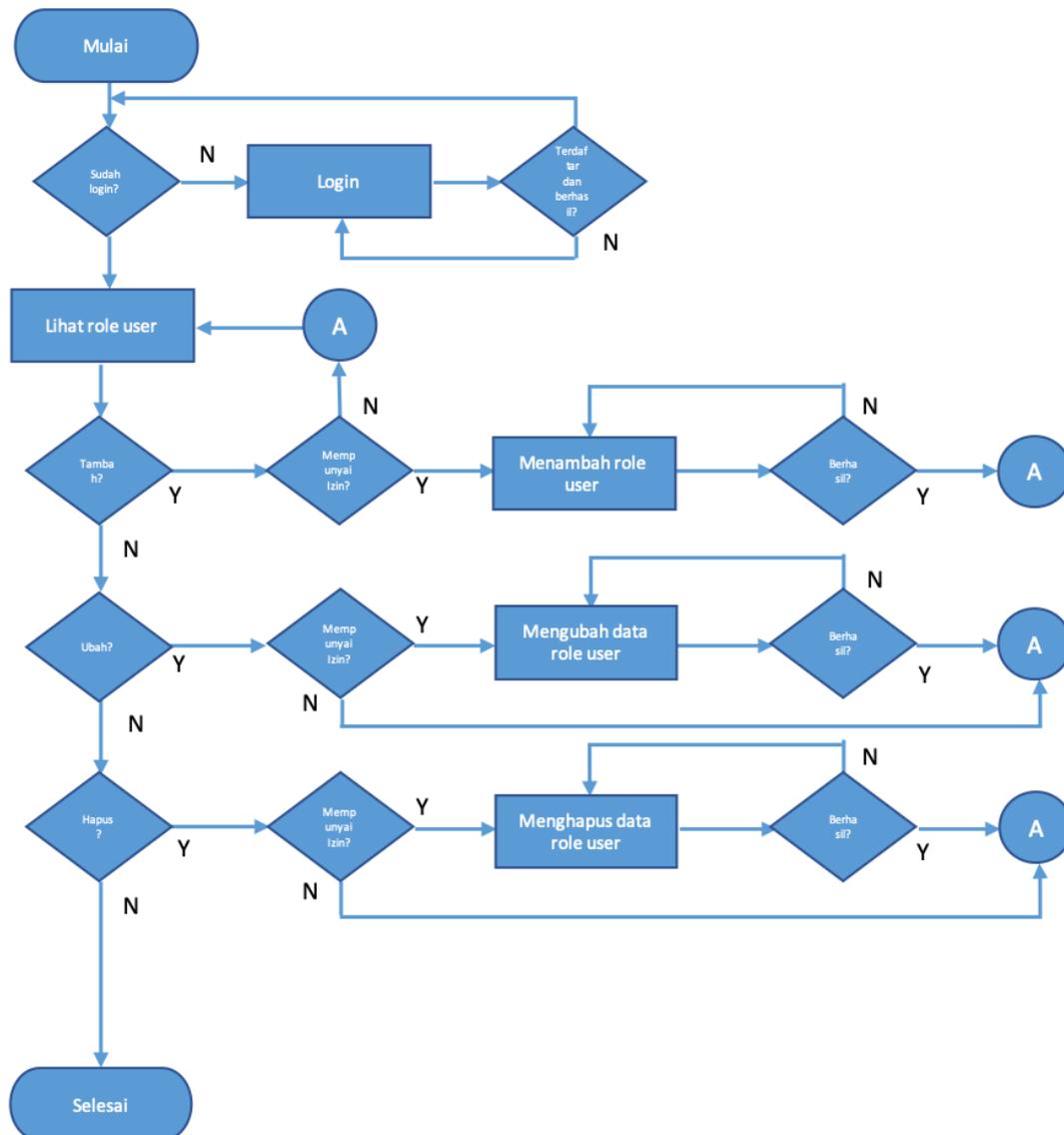
*Flowchart application management dapat dilihat pada gambar*



Gambar 3. 8 *Flowchart application management*

- *Role management*

*Flowchart role management* dapat dilihat pada gambar 3.9.



Gambar 3.9 *Flowchart role management*

### 3.3. Membangun *BPM System*

Dalam membangun sistem, diperlukan peralatan-peralatan yang digunakan untuk mendukung suksesnya pembangunan sistem ini yang meliputi perangkat keras dan perangkat lunak sebagai berikut.

a. Laptop

Fungsi dari laptop adalah untuk menjalankan aplikasi BPMS yang dibangun. Dalam menjalankan aplikasi, penulis menggunakan *web server* dan

*database server* dari aplikasi pihak ketiga, yaitu XAMPP serta Google Chrome sebagai *compiler*.

b. PHP 7.2.28

PHP merupakan bahasa pemrograman untuk *server side* yang nantinya akan menjalankan fungsi-fungsi yang ada pada *server*.

c. Apache 2.4.41

Apache merupakan salah satu aplikasi *web server* yang membantu sebuah aplikasi atau *script server side* untuk dapat dijalankan.

d. MySQL 5.0.12-dev

MySQL merupakan salah satu aplikasi *database server* yang membantu dalam mengelola basis data dari sebuah sistem.

e. XAMPP 3.2.4

XAMPP merupakan salah satu paket aplikasi yang dapat menjalankan *web server* dan *database server*.

f. Google Chrome 81.0.4044.129

Google Chrome merupakan salah satu *web browser* yang dapat membantu untuk mengkompilasi kode-kode menjadi aplikasi yang dapat dibaca oleh pengguna.

g. Sublime Text 3.2.2 Unregistered Version

Sublime Text merupakan salah satu *text editor* yang dapat membantu dalam menyunting *source code* aplikasi.

h. Codeigniter Framework 3.1.11

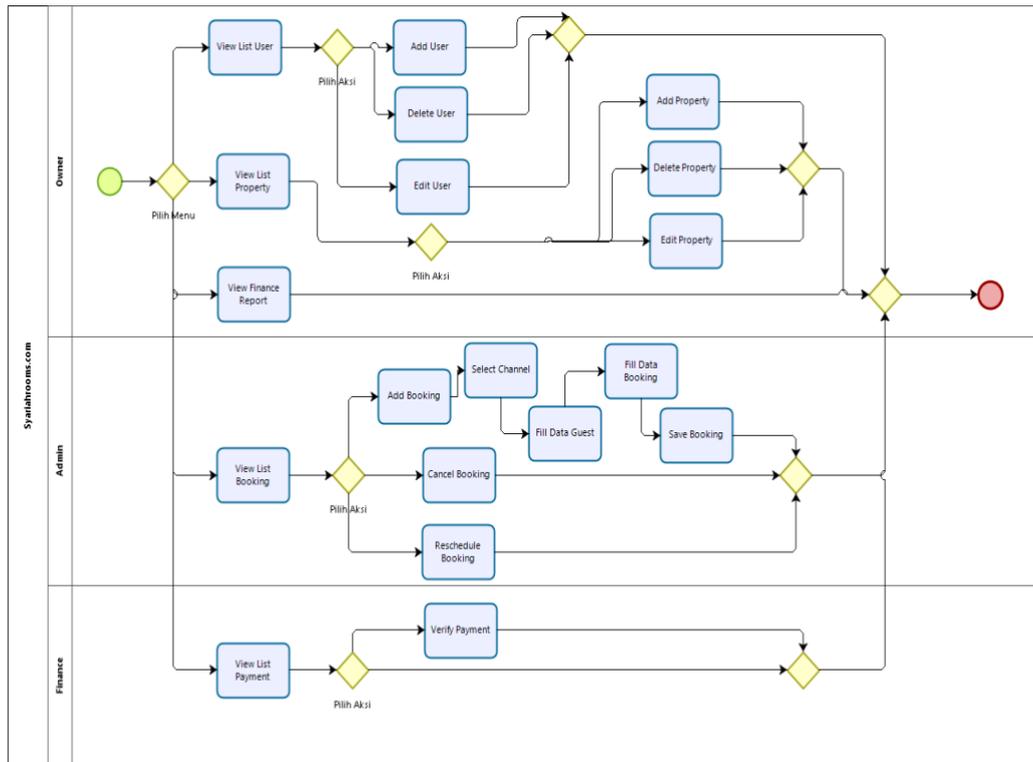
Codeigniter Framework merupakan salah satu *framework* PHP yang menerapkan pola *Model View Controller* (MVC). *Framework* ini dapat membantu dalam penyusunan kode dengan lebih cepat dan lebih mudah.

### 3.4. Uji Coba Sistem

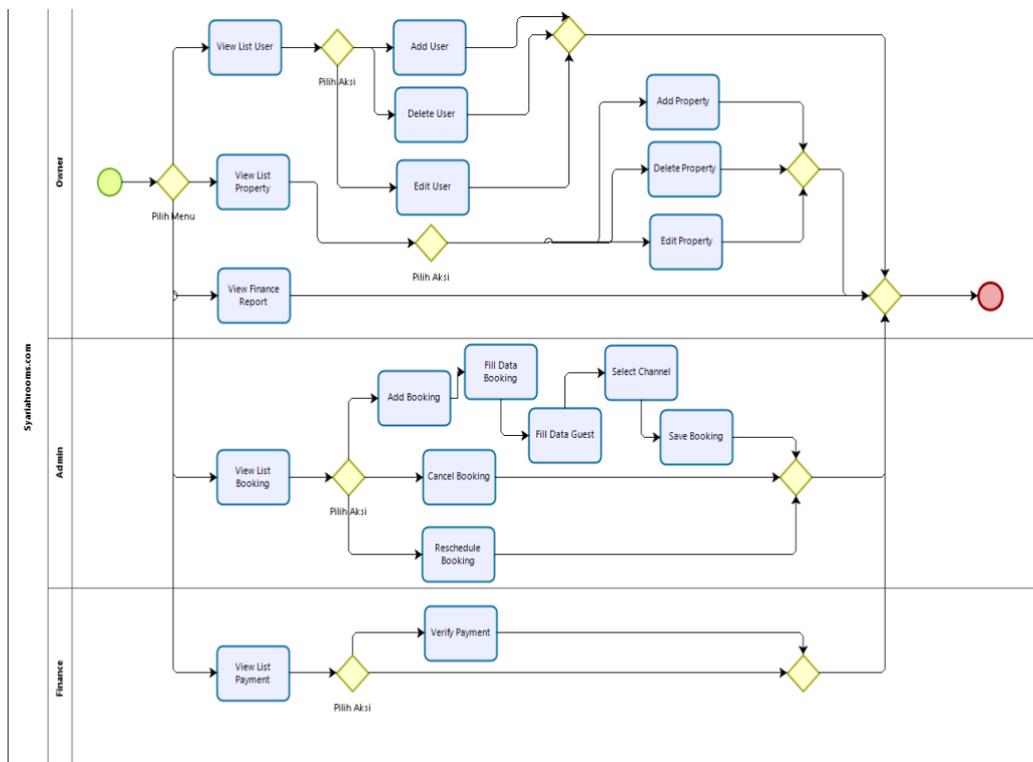
Teknik pengujian yang digunakan pada tahap uji coba ini adalah *black box testing*. Dengan mengetahui fungsi-fungsi yang terdapat pada *BPM System*, pengujian dapat dilakukan sehingga terlihat fungsi-fungsi tersebut dapat berjalan sepenuhnya yang pada waktu yang sama juga mencari kesalahan dari setiap fungsi yang ada.

Lingkup uji coba *BPM System* ini terbatas pada sistem informasi syariahrooms.com yang saat ini berjalan. Dari beberapa fitur yang ada pada sistem informasi tersebut, penulis mengambil 4 fitur yang akan dijadikan sebagai bahan pengujian dari *BPM System*. Fitur-fitur tersebut meliputi *user management*, *property management*, *payment management* dan *booking management*. Penulis akan membuat model proses yang sama dengan proses bisnis yang berlaku saat penulisan ini disusun.

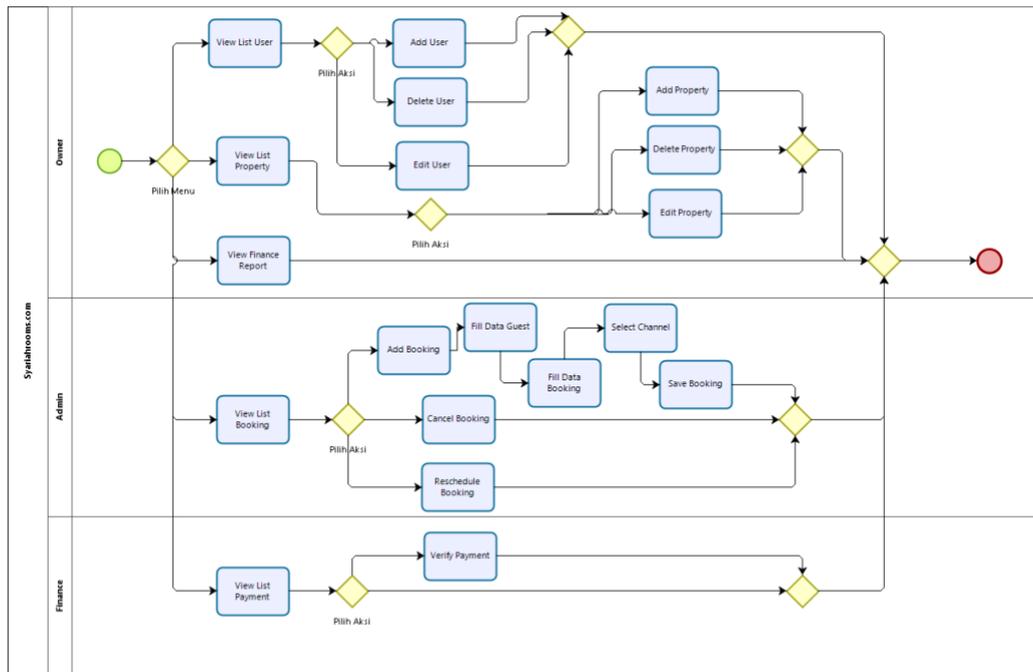
Uji coba ini dilakukan untuk membuktikan bahwa fungsi-fungsi yang ada pada *BPM System* dapat digunakan kembali (*reusable*) dan dapat disesuaikan sesuai dengan kebutuhan (*flexible*). Dalam tahap pengujian perlu ada uji kasus yang mana dalam hal ini menggunakan 4 proses bisnis berbeda. Perbedaan proses bisnis tersebut terletak pada alur proses *booking* yang ditunjukkan pada gambar 3.10, gambar 3.11, gambar 3.12 dan gambar 3.13 berikut.



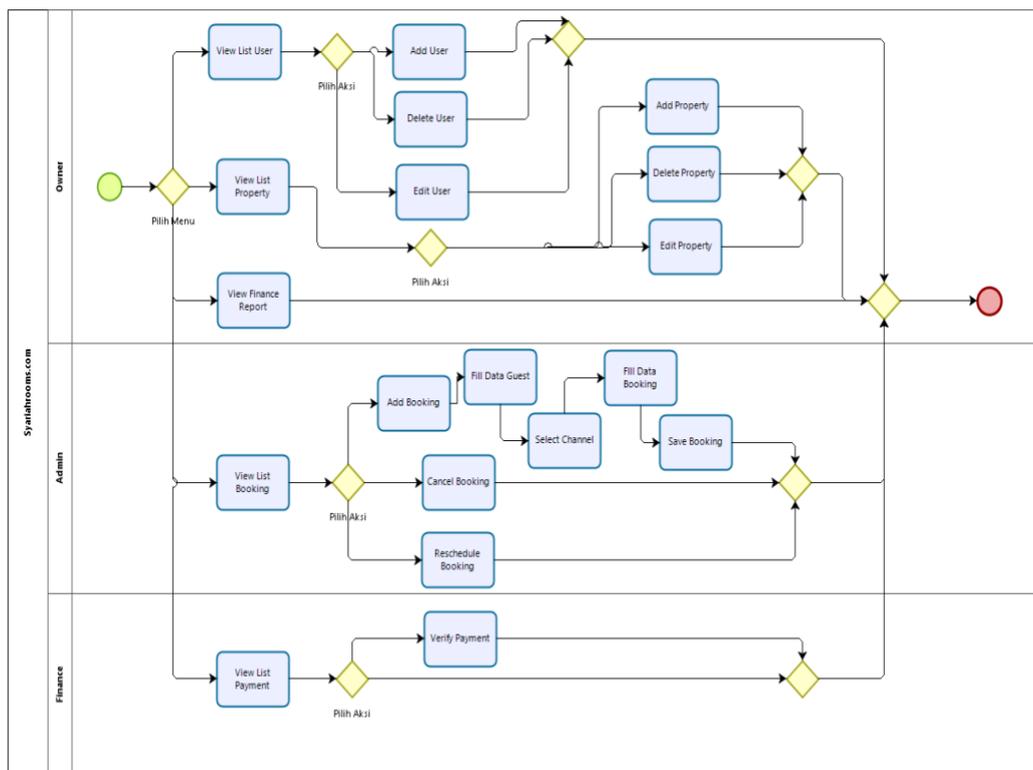
Gambar 3. 10 BPMN Syariahrooms Alur *Booking 1*



Gambar 3. 11 BPMN Syariahrooms Alur *Booking 2*



Gambar 3. 13 BPMN Syariahrooms Alur *Booking 3*



Gambar 3. 12 BPMN Syariahrooms Alur *Booking 4*

## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini akan membahas mengenai pengujian sistem beserta analisa hasil sistem yang telah dibangun. Tujuan dari pengujian tersebut adalah untuk mengetahui bahwa sistem yang dibangun sesuai dengan rancangan yang dibuat serta dapat berjalan dengan baik. Tujuan akhir dari penulisan ini adalah untuk membuktikan bahwa sistem yang dibangun bersifat *flexible* dan *reusable*.

#### 4.1. Langkah Pengujian

##### 4.1.1. Menentukan Modul

Sistem yang dibangun memiliki dua *access level*, yaitu superadmin dan admin tenant. Modul-modul tersebut dijabarkan pada tabel 4.1 berikut.

Tabel 4. 1 Tabel Modul *BPM System*

| No | Modul                     | Level Access                      | Keterangan   |
|----|---------------------------|-----------------------------------|--|
| 1  | <i>Authentication</i>     | Superadmin<br>Dan<br>Admin Tenant | Superadmin dan Admin Tenant <i>login</i> dengan menggunakan data autentikasi yang sudah disiapkan                  |
| 2  | Tambah <i>Web Service</i> | Superadmin                        | Superadmin menambah <i>web service</i> yang nantinya digunakan dalam <i>running application</i> pada <i>tenant</i> |
| 3  | <i>Accept Tenant</i>      | Superadmin                        | Superadmin menerima pendaftaran Admin Tenant   |
| 4  | Pendaftaran <i>Tenant</i> | Admin Tenant                      | Admin Tenant melakukan pendaftaran mandiri pada <i>BPM System</i>  |
| 5  | Tambah <i>Application</i> | Admin Tenant                      | Admin Tenant menambahkan aplikasi yang dapat digunakan oleh <i>tenant</i> itu sendiri                              |
| 6  | Tambah <i>Role User</i>   | Admin Tenant                      | Admin Tenant menambahkan <i>role user</i> yang nantinya dapat digunakan sebagai pemisah hak                        |

|    |   |              |  |
|----|---|--------------|--|
|    |   |              | akses pengguna dalam aplikasi yang dibuat  |
| 7  | Tambah <i>User</i> pada <i>Application</i>    | Admin Tenant | Admin Tenant menambahkan <i>user</i> pada aplikasi yang telah dibuat dengan <i>role user</i> masing-masing                               |
| 8  | <i>Upload</i> XPDL                            | Admin Tenant | Admin Tenant mengunggah file XPDL yang berisi proses bisnis ke dalam aplikasi yang telah dibuat  |
| 9  | <i>Mapping</i> XPDL dengan <i>Web Service</i> | Admin Tenant | Admin Tenant melakukan pemetaan antara proses bisnis yang telah diunggah dengan <i>web service</i> yang telah disediakan oleh Superadmin |
| 10 | <i>Running</i> <i>Application</i>             | Admin Tenant | Admin Tenant menjalankan aplikasi yang telah dipetakan proses bisnis dengan <i>web service</i>   |

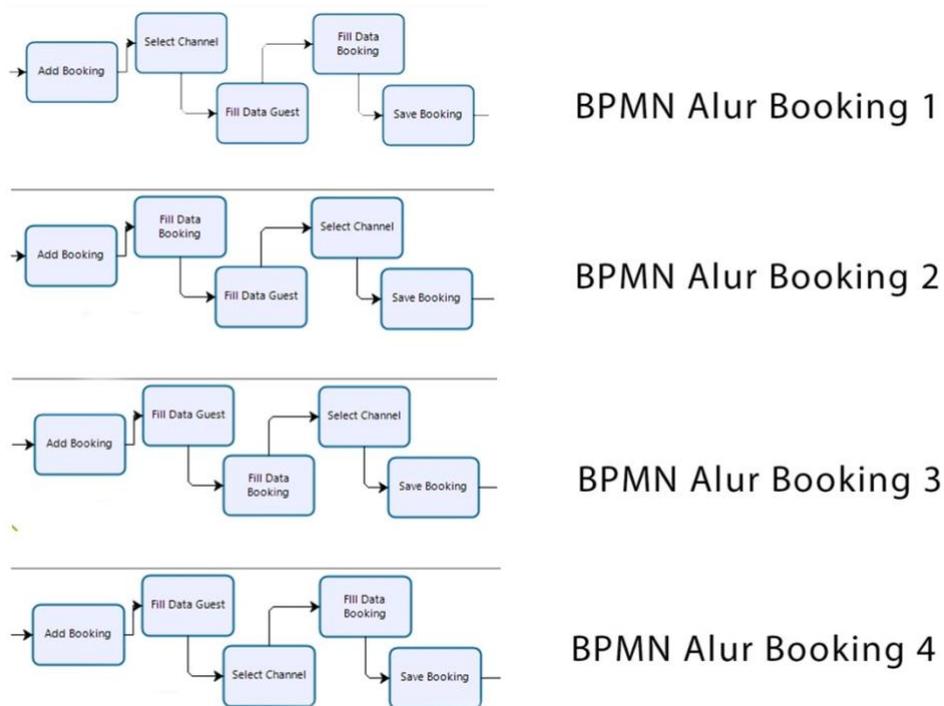
#### 4.1.2. Skenario Pengujian

Skenario pengujian ini dilakukan bertujuan untuk mengetahui *output* yang dihasilkan dengan *input* yang berbeda. Dalam hal ini *black box testing* digunakan untuk menguji skenario tersebut. Skenario yang digunakan dalam penulisan ini antara lain menggunakan data *valid*, menggunakan data *invalid* dan mengisi data sebagian. Skenario tersebut digunakan untuk menguji modul nomor 1-9 sesuai dengan tabel 4.1.

Sedangkan skenario pengujian untuk modul nomor 10 pada tabel 4.1 bertujuan untuk mengetahui bahwa sistem bersifat *flexible* dan *reusable*. Dalam pengujian ini juga menggunakan *black box testing* sebagai metode pengujiannya dan menggunakan uji kasus yang disebutkan pada gambar 3.10, gambar 3.11, gambar 3.12 dan gambar 3.13.

Skenario pengujian untuk membuktikan bahwa sistem bersifat *flexible* yaitu perubahan urutan proses, penambahan proses, pengurangan proses dan perubahan gerbang logika. Namun, skenario yang digunakan adalah perubahan

urutan proses. Berdasarkan uji kasus terdapat urutan proses yang berbeda yaitu pada alur pemesanan. Perbedaan tersebut dapat dilihat pada gambar 4.1 berikut.



Gambar 4. 1 Perbedaan alur pemesanan

Sedangkan skenario pengujian untuk membuktikan bahwa sistem yang dibangun bersifat *reusable* yaitu penggunaan *web service* pada aplikasi atau sistem yang dibuat dapat juga digunakan oleh aplikasi versi baru atau aplikasi lain dalam satu *tenant*. Berdasarkan uji kasus, proses yang digunakan oleh keempat BPMN tersebut tidak ada perbedaan yang signifikan, hanya saja terdapat urutan berbeda yang telah dijelaskan pada gambar 4.1 di atas. Oleh karena itu, skenario pengujian ini bertujuan untuk membuktikan bahwa *web service* yang telah disediakan oleh sistem dapat dijalankan pada keempat BPMN dari uji kasus tersebut.

#### 4.2. Hasil Pengujian

Hasil pengujian dari skenario dengan data *input* yang berbeda, yaitu menggunakan data *valid*, menggunakan data *invalid* dan mengisi data sebagian pada modul 1-9 terdapat pada tabel 4.2 berikut.

Tabel 4. 2 Tabel Hasil Pengujian Modul *BPM System*

| No | Modul                     | Skenario      | Output   | Keterangan   |
|----|---------------------------|---------------|--|--|
| 1  | <b>Authentication</b>     | Data Valid    | Masuk ke dalam Halaman <i>Dashboard</i> pada masing-masing <i>role</i> | Sukses   |
|    |                           | Data Invalid  | Menampilkan <i>alert</i> gagal   | Gagal  |
|    |                           | Data Sebagian | Menampilkan <i>alert warning</i> pada <i>form</i> yang belum diisi     | Warning  |
| 2  | <b>Tambah Web Service</b> | Data Valid    | Diarahkan ke Halaman <i>List Web Service</i>                           | Sukses   |
|    |                           | Data Invalid  | -  | Tidak ada kemungkinan karena <i>field</i> berjenis <i>text</i> semua |
|    |                           | Data Sebagian | Menampilkan <i>alert warning</i> pada <i>form</i> yang belum diisi     | Warning  |
| 3  | <b>Accept Tenant</b>      | Data Valid    | Menampilkan <i>alert</i> sukses  | Sukses   |
|    |                           | Data Invalid  | Menampilkan <i>alert</i> “ID Tenant harus berupa nomor”                | Gagal  |
|    |                           | Data Sebagian | Menampilkan <i>alert</i> “ID Tenant tidak dimasukkan”                  | Gagal  |
| 4  | <b>Pendaftaran Tenant</b> | Data Valid    | Menampilkan <i>alert</i> bahwa pendaftaran                             | Sukses   |

|   |                           |               |   |  |
|---|---------------------------|---------------|---|--|
|   |                           |               | sukses dan masuk ke dalam <i>List Tenant</i> dengan status <i>Need an Acception</i> |  |
|   |                           | Data Invalid  | Menampilkan <i>alert</i> “Password tidak sama”                                      | <i>Warning</i>   |
|   |                           | Data Sebagian | Menampilkan <i>alert warning</i> pada <i>form</i> yang belum diisi                  | <i>Warning</i>   |
| 5 | Tambah <i>Application</i> | Data Valid    | Diarahkan ke Halaman <i>List Application</i>  | Sukses   |
|   |                           | Data Invalid  | Menampilkan <i>alert</i> “Aplikasi gagal dibuat. Logo harus berformat JPEG/JPG/PNG” | Gagal  |
|   |                           | Data Sebagian | Menampilkan <i>alert warning</i> pada <i>form</i> yang belum diisi                  | <i>Warning</i>   |
| 6 | Tambah <i>Role User</i>   | Data Valid    | Diarahkan ke Halaman <i>List Role User</i>  | Sukses   |
|   |                           | Data Invalid  | -   | Tidak ada kemungkinan karena <i>field</i> berjenis <i>text</i> semua |
|   |                           | Data Sebagian | Menampilkan <i>alert warning</i> pada <i>form</i> yang belum diisi                  | <i>Warning</i>   |

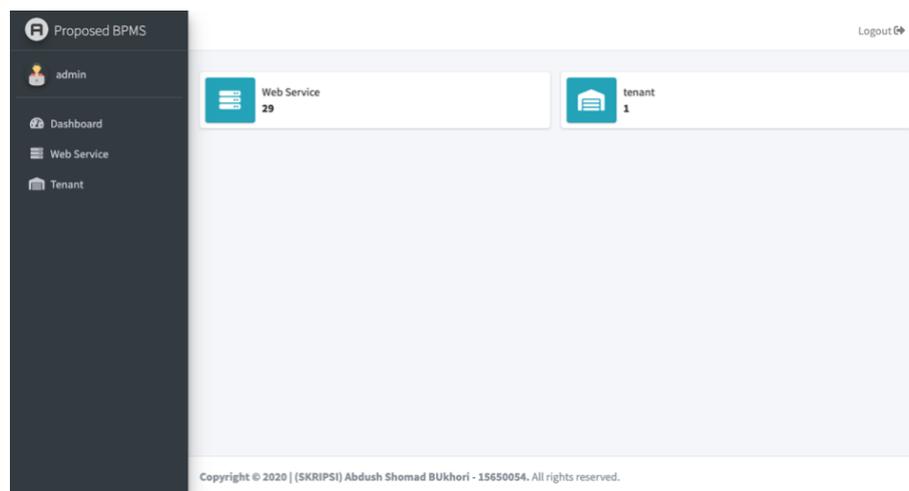
|   |                                 |               |  |  |
|---|---------------------------------|---------------|--|--|
| 7 | Tambah User pada Application    | Data Valid    | Diarahkan ke Halaman Detail Aplikasi dan <i>user</i> yang ditambahkan akan terdaftar di tabel <i>List User</i> pada <i>tab Users</i> | Sukses   |
|   |                                 | Data Invalid  | -  | Tidak ada kemungkinan karena <i>field</i> berjenis <i>text</i> dan <i>dropdown</i> |
|   |                                 | Data Sebagian | Menampilkan <i>alert warning</i> pada <i>form</i> yang belum diisi   | Warning  |
| 8 | Upload XPDL                     | Data Valid    | Diarahkan ke Halaman <i>Setup Role User</i> dari Rekomendasi Sistem  | Sukses   |
|   |                                 | Data Invalid  | Menampilkan <i>alert</i> “File yang diupload harus berekstensi .XPDL”  | Gagal  |
|   |                                 | Data Sebagian | Menampilkan <i>alert warning</i> pada <i>form</i> yang belum diisi   | Warning  |
| 9 | Mapping XPDL dengan Web Service | Data Valid    | Diarahkan ke Halaman Detail Aplikasi dan <i>value</i> dari <i>Current Version</i> akan sama dengan                                   | Sukses   |

|  |  |                     |  |  |
|--|--|---------------------|--|--|
|  |  |                     | apa yang dimasukkan  |  |
|  |  | Data <i>Invalid</i> | -  | Tidak ada kemungkinan karena <i>field</i> berjenis <i>dropdown</i> |
|  |  | Data Sebagian       | Menampilkan <i>alert warning</i> pada <i>form</i> yang belum diisi | <i>Warning</i>   |

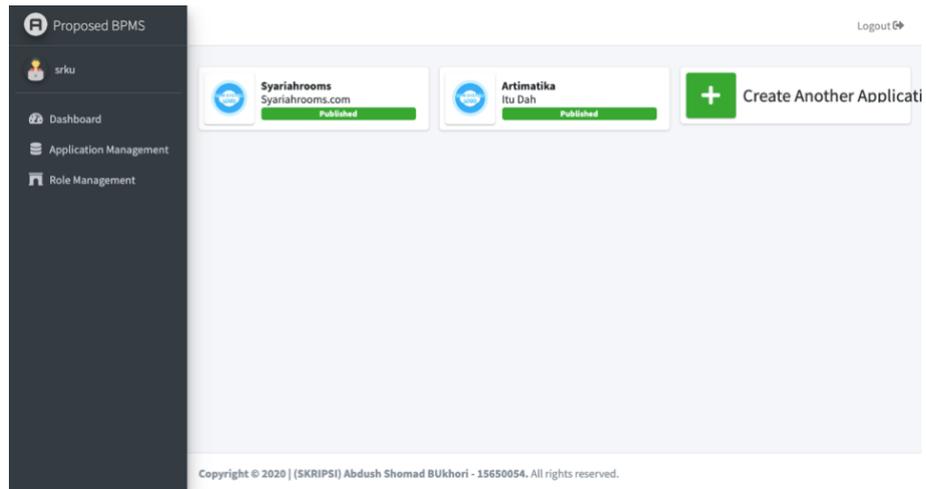
Penjelasan tabel 4.2 dijabarkan sebagai berikut:

- **Modul *Authentication***

Pada modul *authentication*, super admin maupun admin tenant sukses menjalankan dengan data *valid* ditunjukkan dengan masuk ke dalam halaman dashboard seperti gambar 4.2 dan gambar 4.3. Hasil gagal keluar ketika data yg digunakan *invalid* ditunjukkan dengan *alert* seperti gambar 4.4. Sedangkan ketika menggunakan data yang tidak lengkap akan keluar *warning* pada *form* yang ditunjukkan pada gambar 4.5.



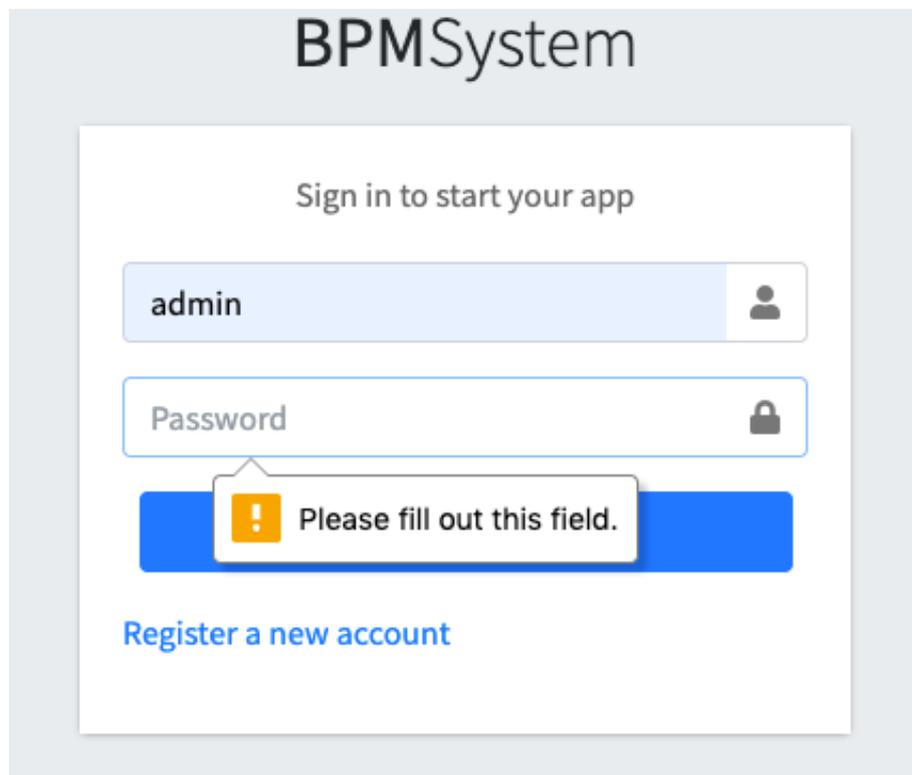
Gambar 4. 2 Halaman *Dashboard* Super Admin



Gambar 4. 5 Halaman *Dashboard* Admin tenant



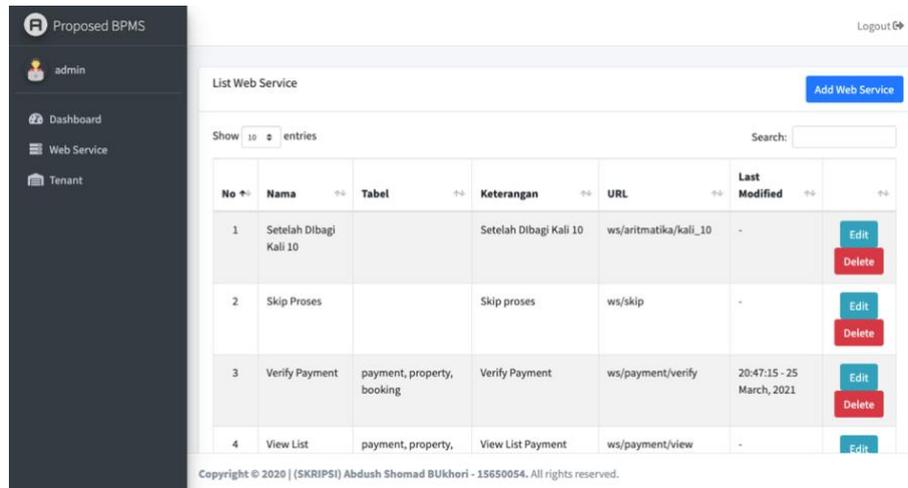
Gambar 4. 4 *Alert* ketika data *invalid*



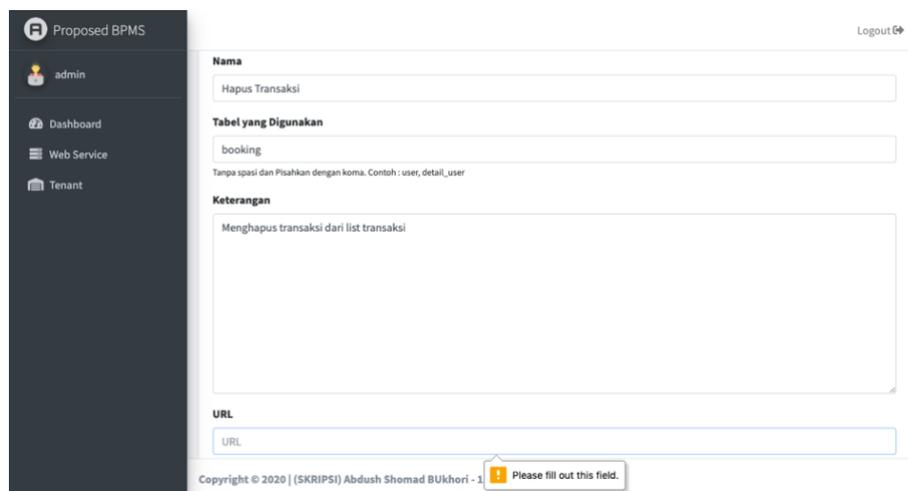
Gambar 4. 3 *Warning* Pada *Form Login*

- **Modul Tambah *Web Service***

Pada modul tambah *web service*, hasil sukses akan mengarahkan ke halaman *list web service* sebagaimana ditunjukkan pada gambar 4.6. Ketika data yang dimasukkan tidak lengkap akan muncul *warning* seperti gambar 4.7.



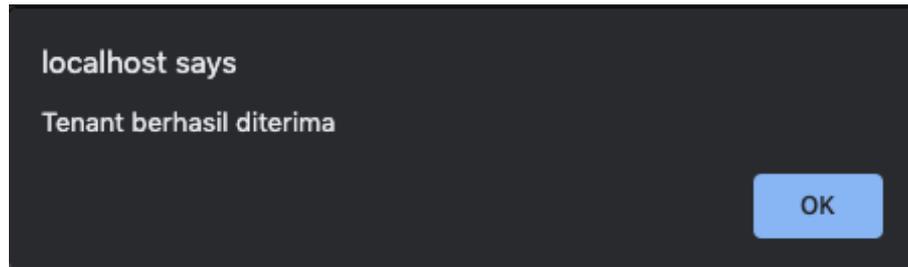
Gambar 4. 6 Halaman *List Web Service*



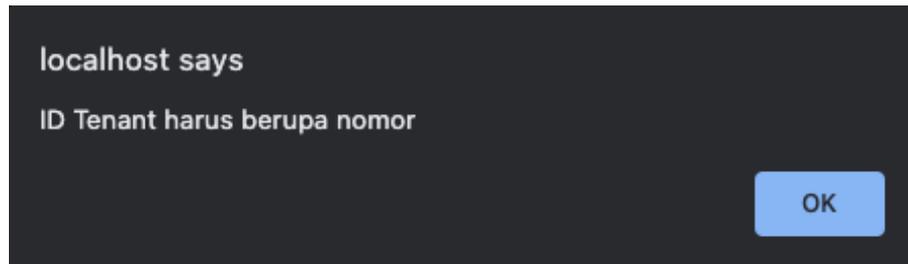
Gambar 4. 7 *Warning Pada Form Tambah Web Service*

- **Modul *Accept Tenant***

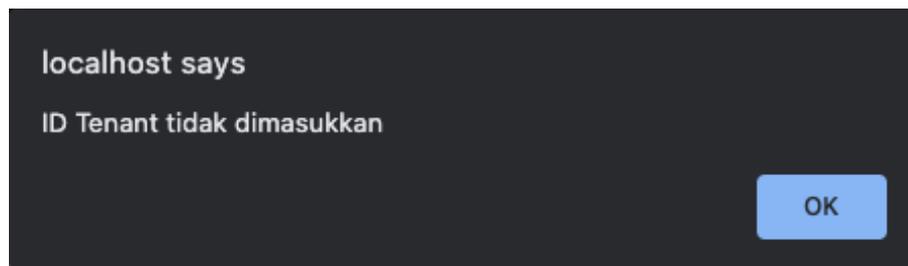
Pada modul *accept tenant*, hasil sukses ditunjukkan dengan *alert* seperti gambar 4.8. Hasil gagal keluar ketika data yg digunakan *invalid* ditunjukkan dengan *alert* seperti gambar 4.9. Sedangkan ketika menggunakan data yang tidak lengkap akan keluar *alert* ditunjukkan pada gambar 4.10.



Gambar 4. 8 *Alert* Ketika Berhasil Menerima Pendaftaran *Tenant*



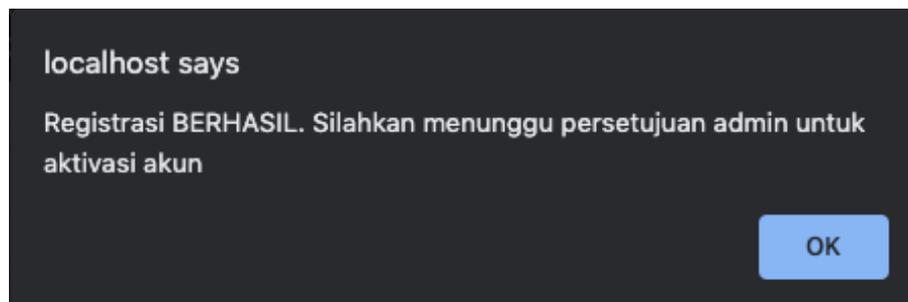
Gambar 4. 9 *Alert* Ketika Gagal Menerima Pendaftaran *Tenant*



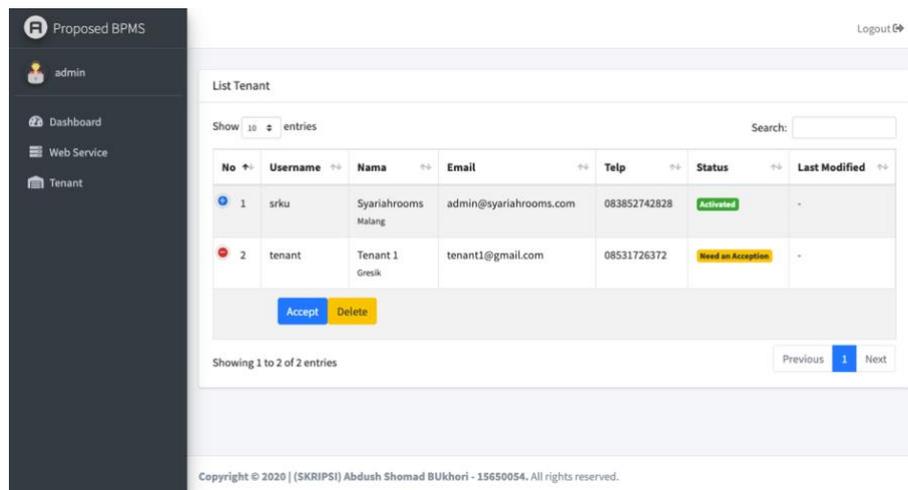
Gambar 4. 10 *Alert* Ketika Data Tidak Lengkap Menerima Pendaftaran *Tenant*

- **Modul Pendaftaran *Tenant***

Pada modul pendaftaran *tenant*, hasil sukses akan memunculkan *alert* bahwa pendaftaran sukses sebagaimana ditunjukkan pada gambar 4.11 dan data tersebut masuk pada *list* tenant dengan status *Need an Acception* sebagaimana ditunjukkan pada gambar 4.12. Ketika data yang dimasukkan *invalid* akan muncul *alert* seperti gambar 4.13. Sedangkan ketika data yang dimasukkan tidak lengkap akan muncul *warning* seperti gambar 4.14.



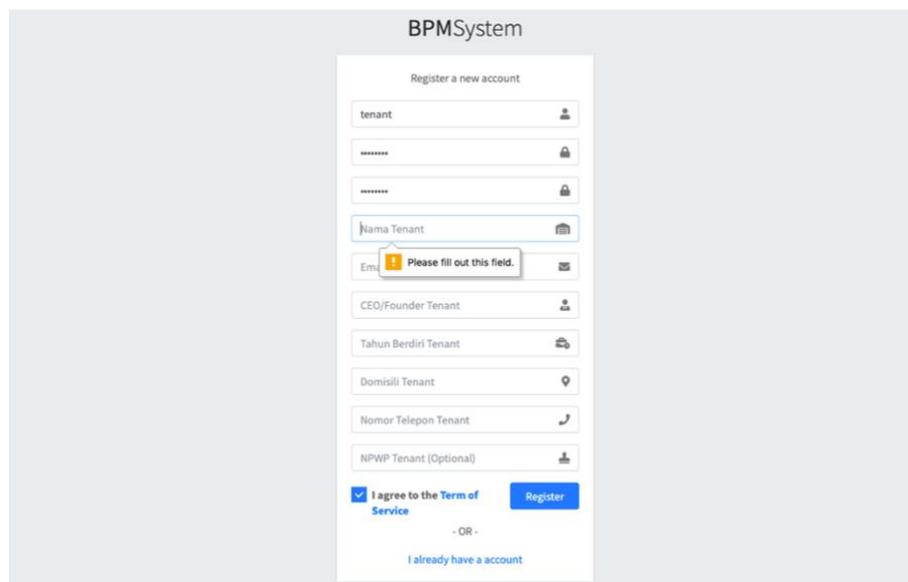
Gambar 4. 11 *Alert* bahwa pendaftaran berhasil



Gambar 4. 12 Halaman *List Tenant*



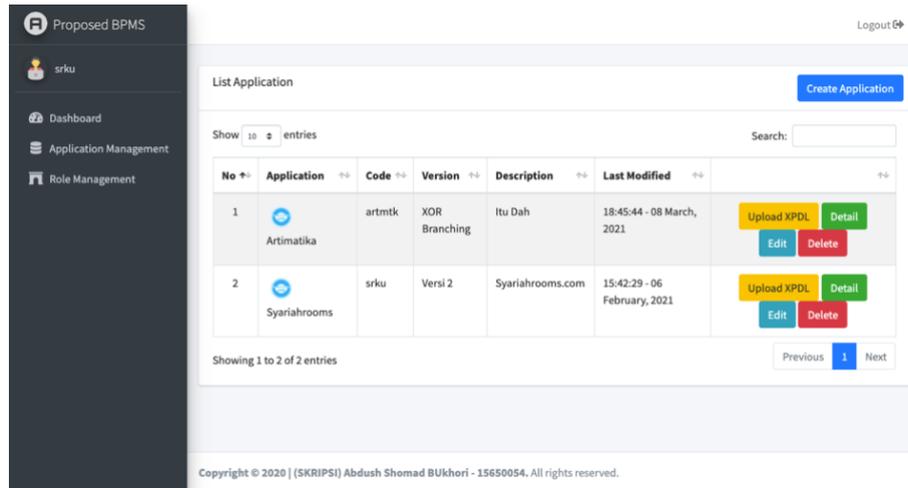
Gambar 4. 14 *Alert* bahwa pendaftaran gagal



Gambar 4. 13 *Warning* Pada *Form* Pendaftaran *Tenant*

- **Modul Tambah *Application***

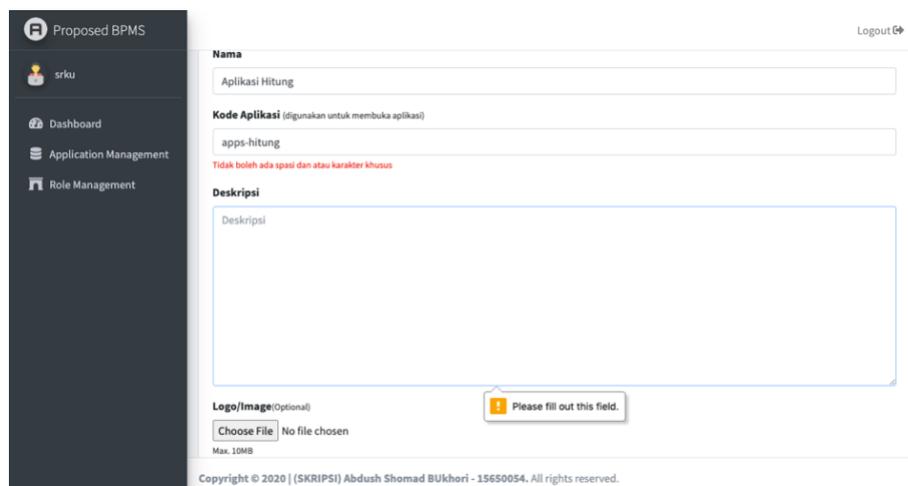
Pada modul tambah *application*, hasil sukses akan muncul pada *list* aplikasi sebagaimana ditunjukkan pada gambar 4.15. Ketika data yang dimasukkan *invalid* akan muncul *alert* seperti gambar 4.16. Sedangkan ketika data yang dimasukkan tidak lengkap akan muncul *warning* seperti gambar 4.17.



Gambar 4. 16 Halaman *List Application*



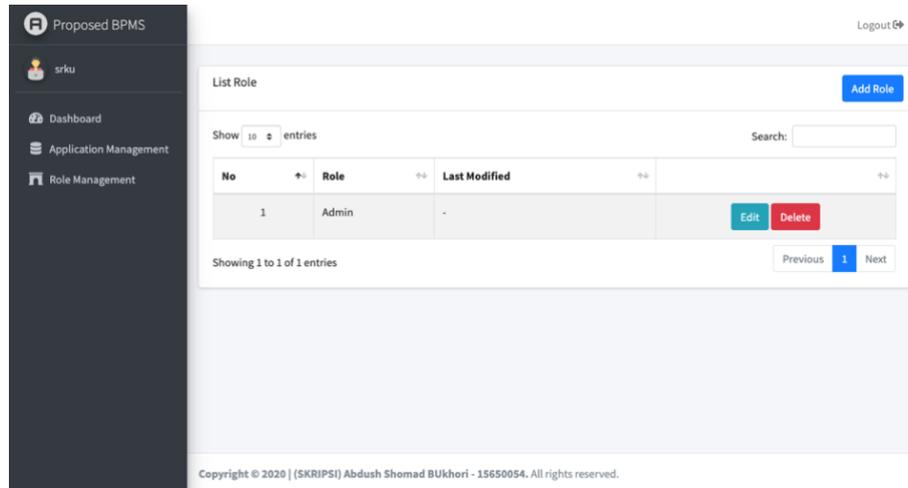
Gambar 4. 17 *Alert* bahwa pendaftaran gagal



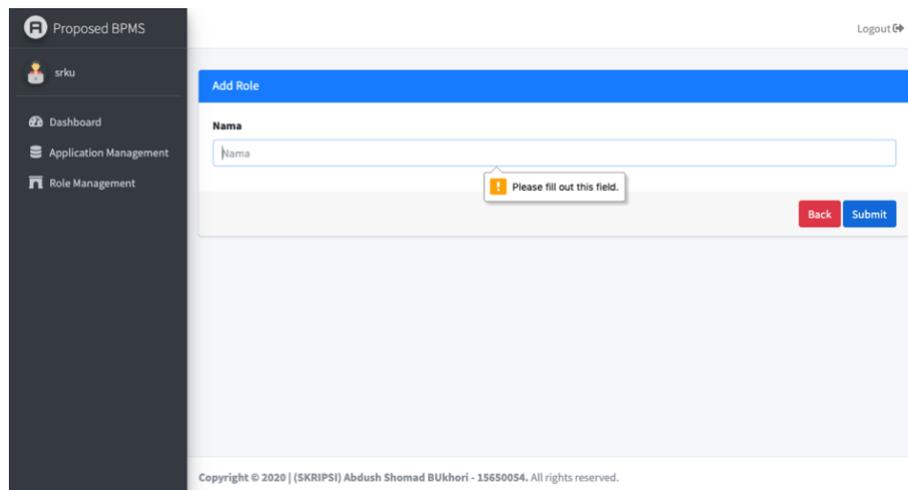
Gambar 4. 15 *Warning* Pada *Form* Tambah Aplikasi

- **Modul Tambah *Role User***

Pada modul tambah *role user*, hasil sukses jika data yang dimasukkan tampil pada *list* role user sebagaimana ditunjukkan pada gambar 4.18. Ketika data yang dimasukkan tidak lengkap akan muncul *warning* seperti gambar 4.19.



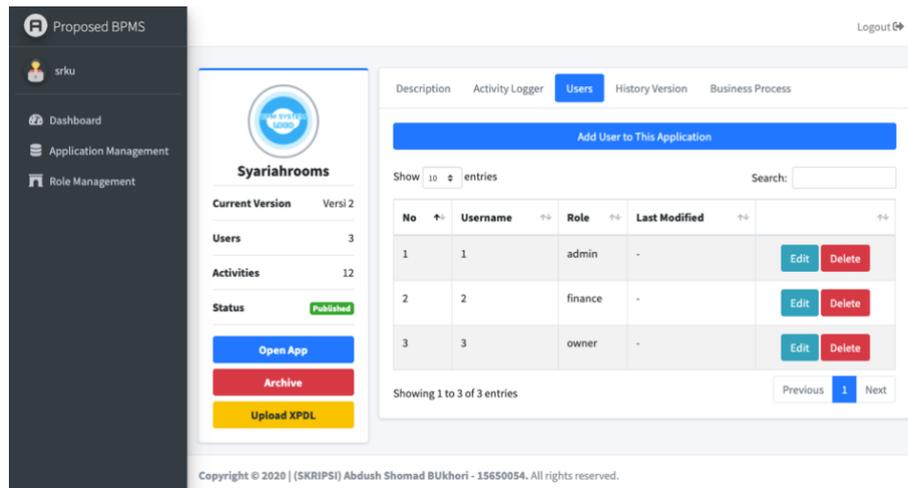
Gambar 4. 18 Halaman *List Role User*



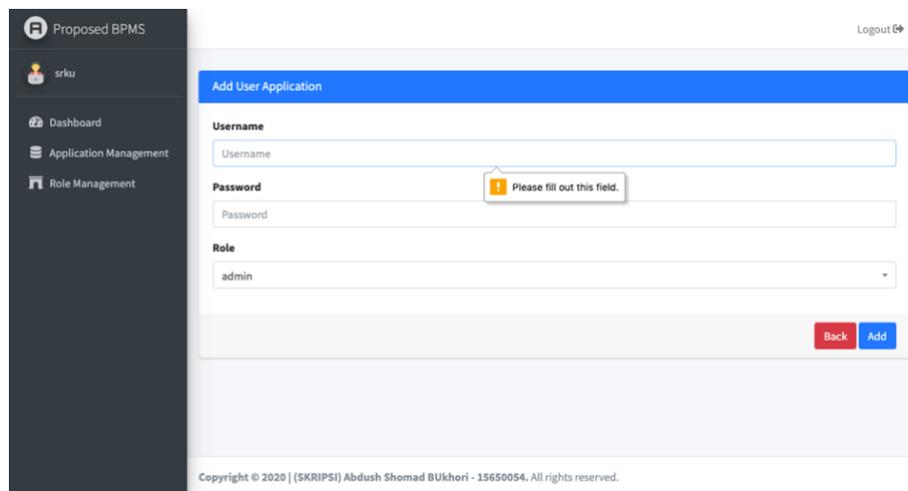
Gambar 4. 19 *Warning* Pada *Form* Tambah *Role User*

- **Modul Tambah *User* pada *Application***

Pada modul tambah *user* pada *application*, hasil sukses jika data yang dimasukkan tampil pada *list* user sebagaimana ditunjukkan pada gambar 4.20. Ketika data yang dimasukkan tidak lengkap akan muncul *warning* seperti gambar 4.21.



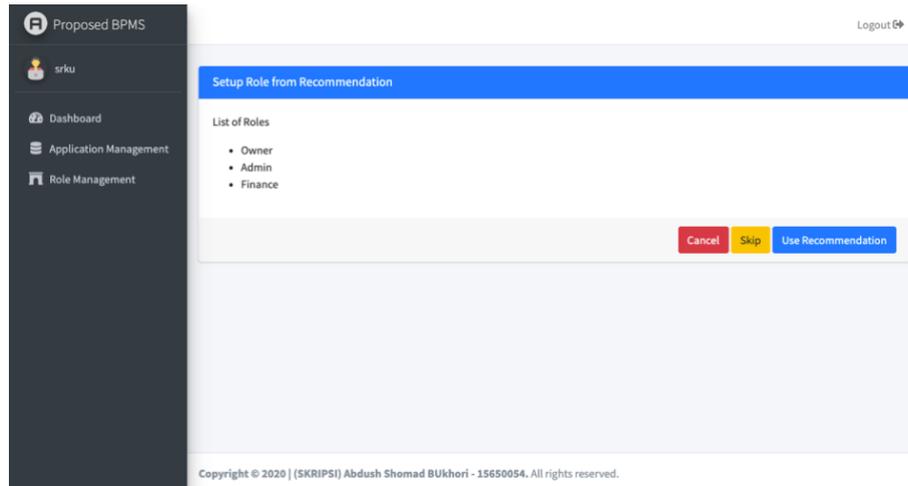
Gambar 4. 21 Halaman *List User* pada Aplikasi



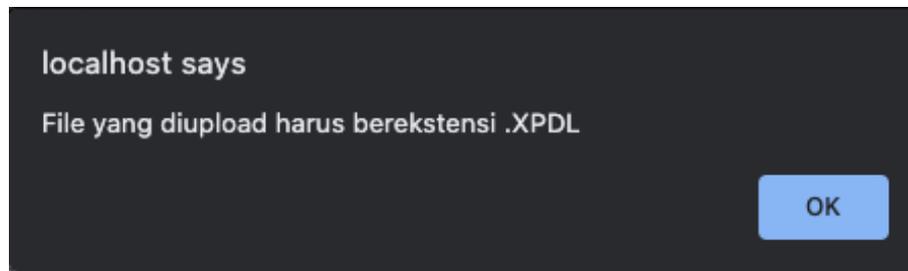
Gambar 4. 20 *Warning* Pada *Form Tambah User* Aplikasi

- **Modul Upload XPDL**

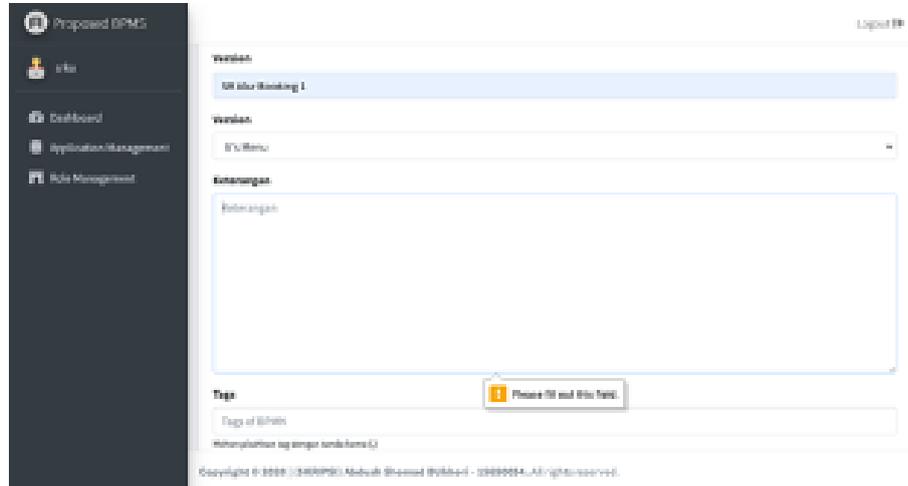
Pada modul *upload* XPDL, hasil sukses jika diarahkan ke halaman setup role dari rekomendasi sistem sebagaimana ditunjukkan pada gambar 4.22. Ketika data yang dimasukkan tidak lengkap akan muncul *alert* seperti gambar 4.23. Sedangkan ketika data yang dimasukkan tidak lengkap akan muncul *warning* seperti gambar 4.24.



Gambar 4. 23 Halaman *Setup Role User* dari Rekomendasi Sistem



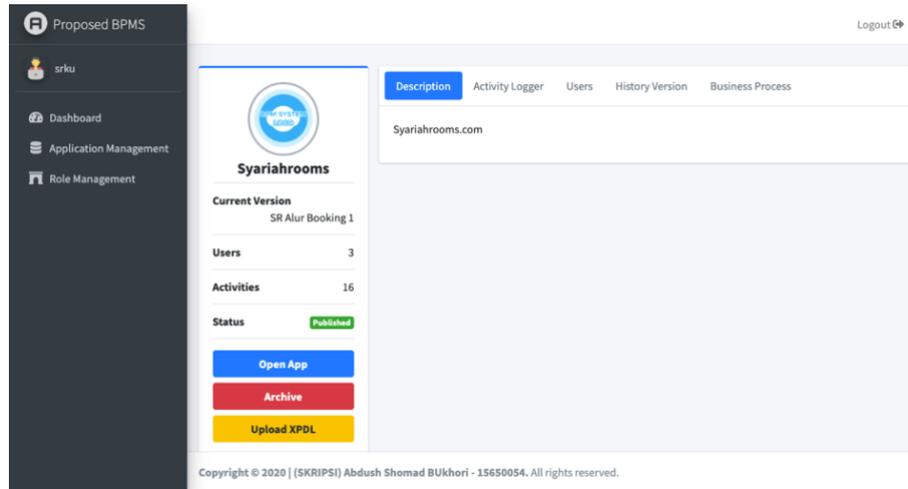
Gambar 4. 24 *Alert* bahwa *upload XPDL* gagal



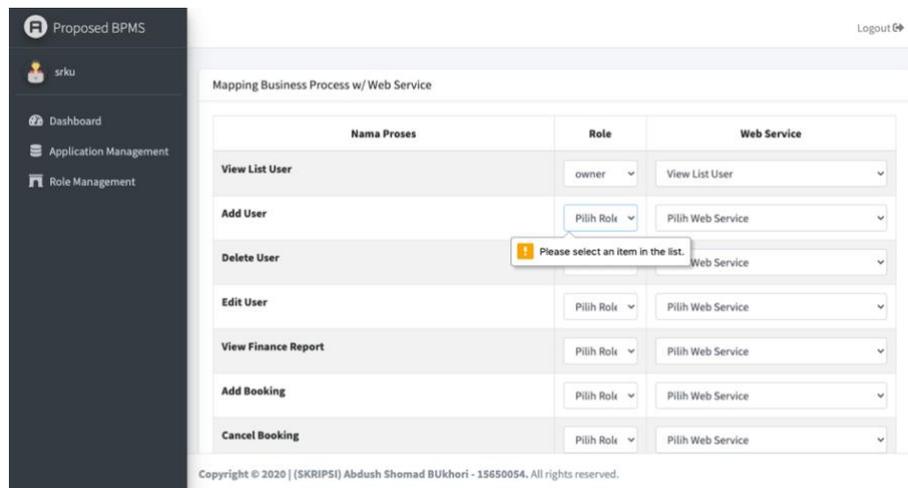
Gambar 4. 22 *Warning* Pada *Form Upload XPDL*

- **Modul *Mapping XPDL* dengan *Web Service***

Pada modul *mapping* XPDL dengan *web service*, hasil sukses jika versi yang sedang digunakan sesuai dengan nama versi yang dimasukkan sebagaimana ditunjukkan pada gambar 4.25. Ketika data yang dimasukkan tidak lengkap akan muncul *warning* seperti gambar 4.26.



Gambar 4. 25 Halaman Detail Aplikasi



Gambar 4. 26 Warning Pada Form Mapping Web Service

Modul terakhir pada tabel 4.1 no 10, yaitu *running application* merupakan modul inti dari *BPM System* yang dibangun. Fungsi tersebut digunakan untuk menjalankan aplikasi berdasarkan proses bisnis yang telah diunggah. Proses bisnis yang digunakan antara lain : gambar 3.10 sebagai BPMN Alur *Booking 1*, gambar 3.11 sebagai BPMN Alur *Booking 2*, gambar 3.12 sebagai BPMN Alur *Booking 3* dan gambar 3.13 sebagai BPMN Alur *Booking 4*. Proses bisnis tersebut digunakan sebagai bahan uji pada *BPM System* untuk membuktikan bahwa sistem yang dibangun bersifat *flexible* dan *reusable*.

*Flexibility* mempunyai arti bahwa sistem dapat menyesuaikan dengan perubahan proses bisnis. Perubahan yang dimaksud dapat berupa penggantian urutan proses, penambahan proses, pengurangan proses, dan lain sebagainya.

Sedangkan *reusability* mempunyai arti bahwa *method* atau fungsi yang berada dalam aplikasi tersebut dapat digunakan oleh aplikasi lain. Hasil dari pengujian keempat BPMN tersebut ditampilkan pada tabel 4.3 berikut.

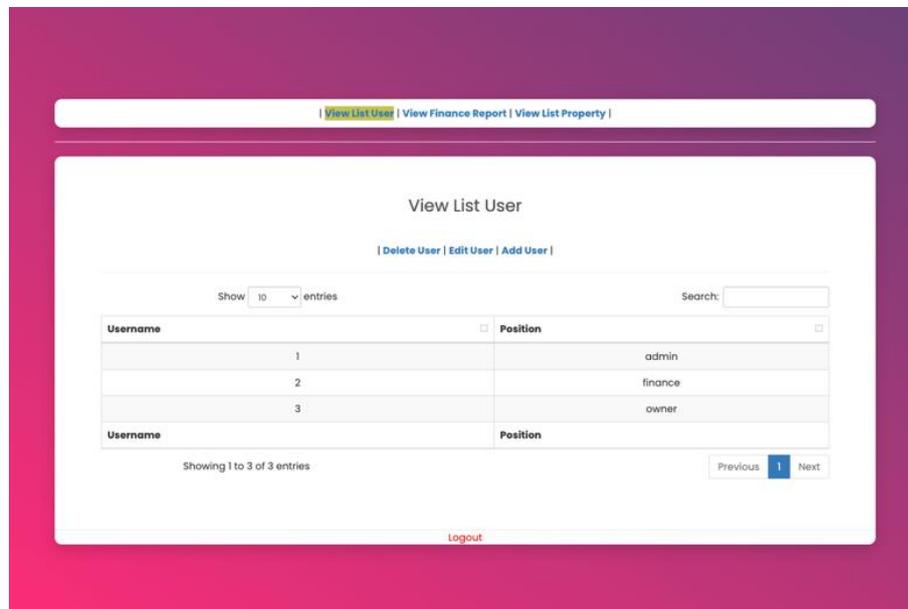
Tabel 4. 3 Tabel Hasil *Running Application*

| Skenario                          | Hasil    |
|-----------------------------------|----------|
| <b>BPMN Alur <i>Booking 1</i></b> | Berhasil |
| <b>BPMN Alur <i>Booking 2</i></b> | Berhasil |
| <b>BPMN Alur <i>Booking 3</i></b> | Berhasil |
| <b>BPMN Alur <i>Booking 4</i></b> | Berhasil |

Berikut adalah penjelasan dari setiap proses pada semua BPMN berdasarkan tabel 4.3 untuk pengujian *reusability* sistem.

- *View List User*

Proses ini adalah proses menampilkan daftar pengguna dari aplikasi. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.27 berikut.

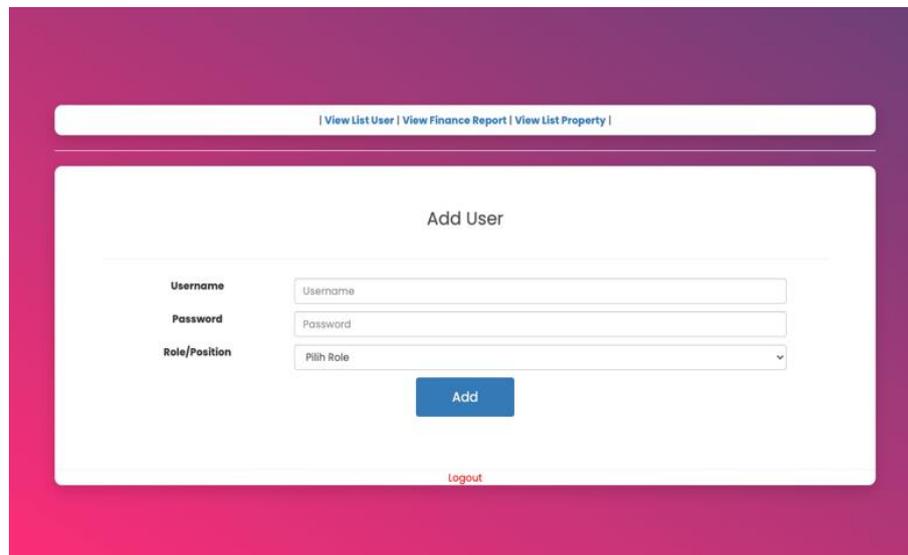


Gambar 4. 27 *View List User*

- *Add User*

Proses ini adalah proses penambahan pengguna dari aplikasi. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar

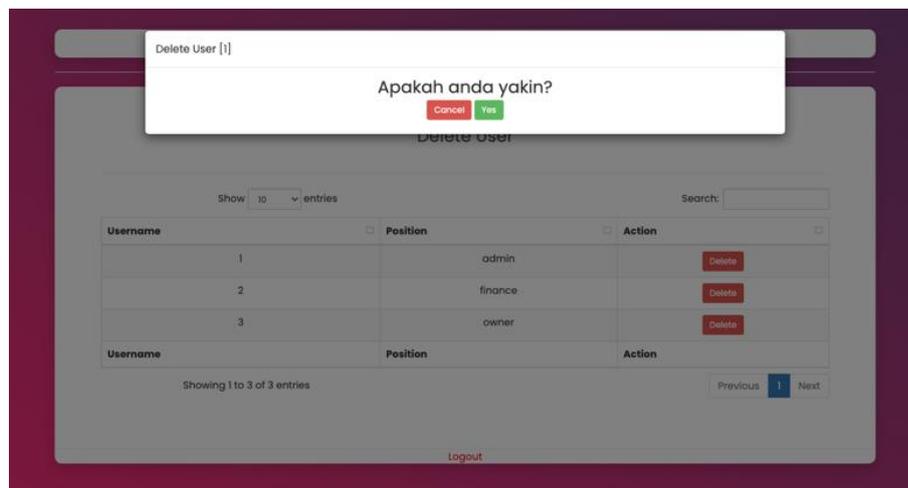
pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.28 berikut.



Gambar 4. 28 *Add User*

- *Delete User*

Proses ini adalah proses untuk menghapus data pengguna dari aplikasi. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.29 berikut.

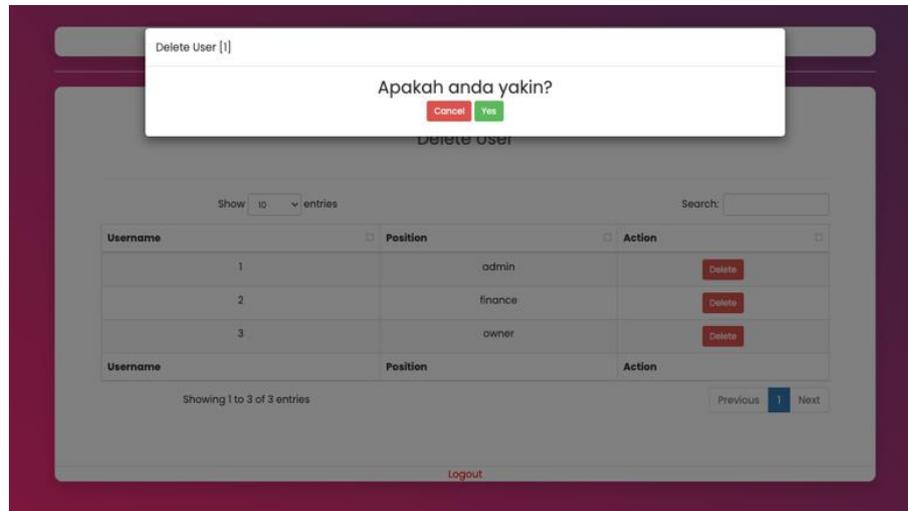


Gambar 4. 29 *Delete User*

- *Edit User*

Proses ini adalah proses untuk merubah data pengguna dari aplikasi. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan

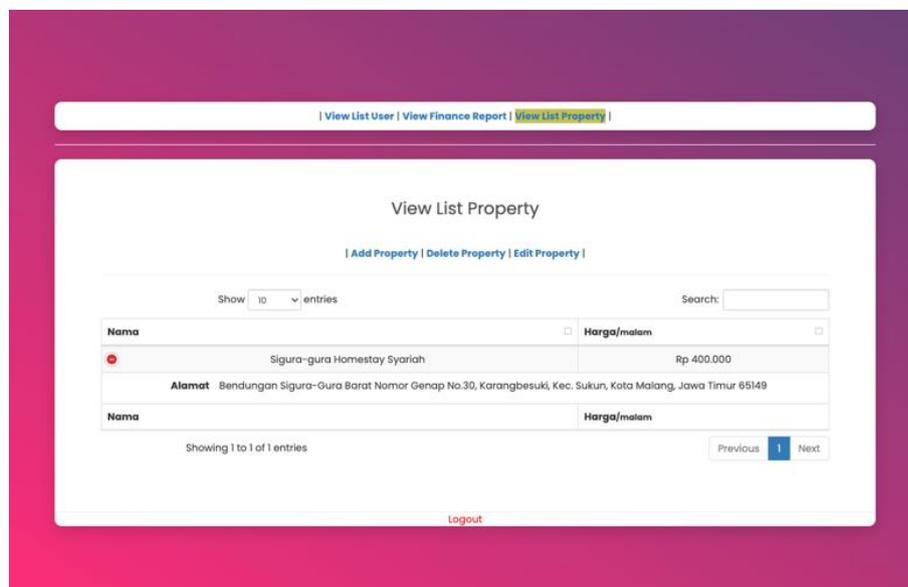
lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.30 berikut.



Gambar 4. 30 *Edit User*

- *View List Property*

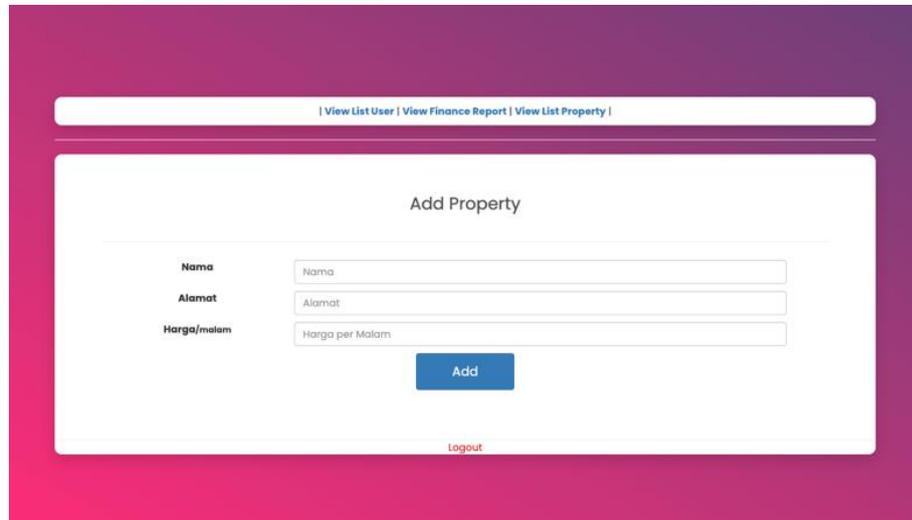
Proses ini adalah proses untuk menampilkan daftar properti. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.31 berikut.



Gambar 4. 31 *View List Property*

- *Add Property*

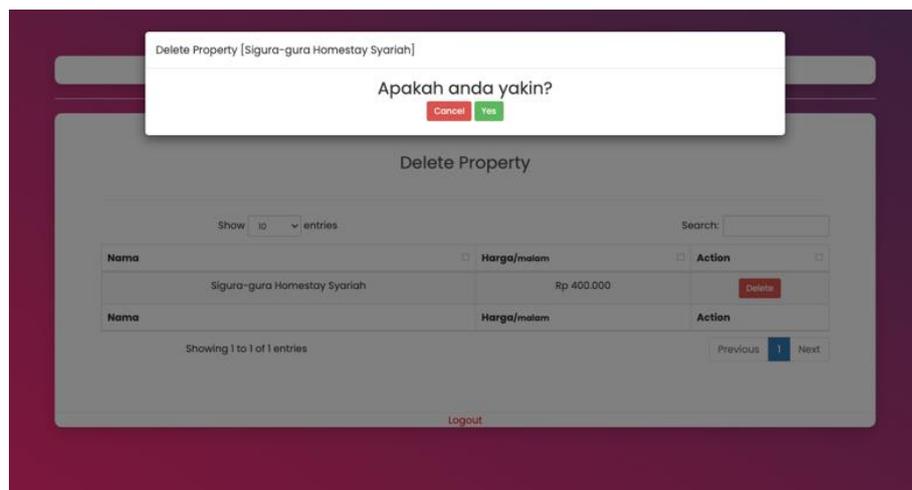
Proses ini adalah proses penambahan data properti. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.32 berikut.



Gambar 4. 32 *Add Property*

- *Delete Property*

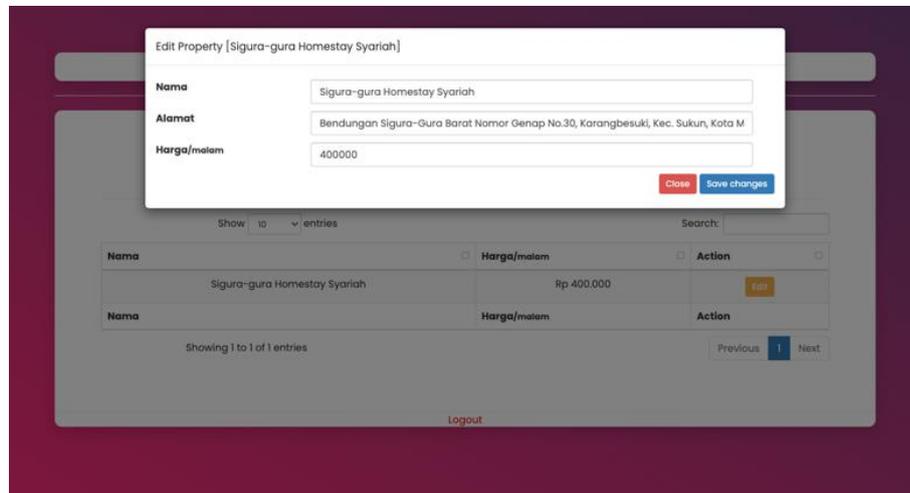
Proses ini adalah proses untuk menghapus data properti. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.33 berikut.



Gambar 4. 33 *Delete Property*

- *Edit Property*

Proses ini adalah proses untuk merubah data properti. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.34 berikut.



Gambar 4. 34 *Edit Property*

- *View Finance Report*

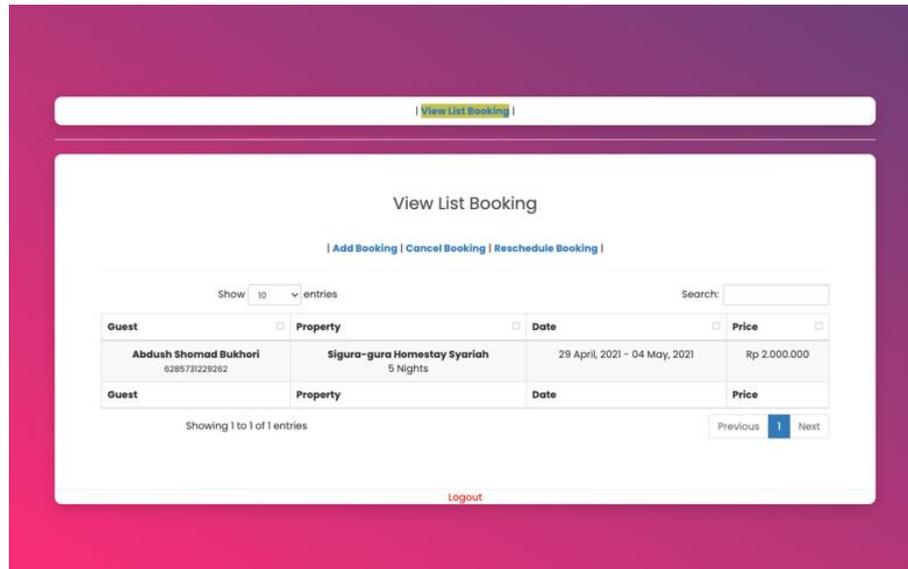
Proses ini adalah proses untuk menampilkan laporan keuangan. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.35 berikut.



Gambar 4. 35 *View Finance Report*

- *View List Booking*

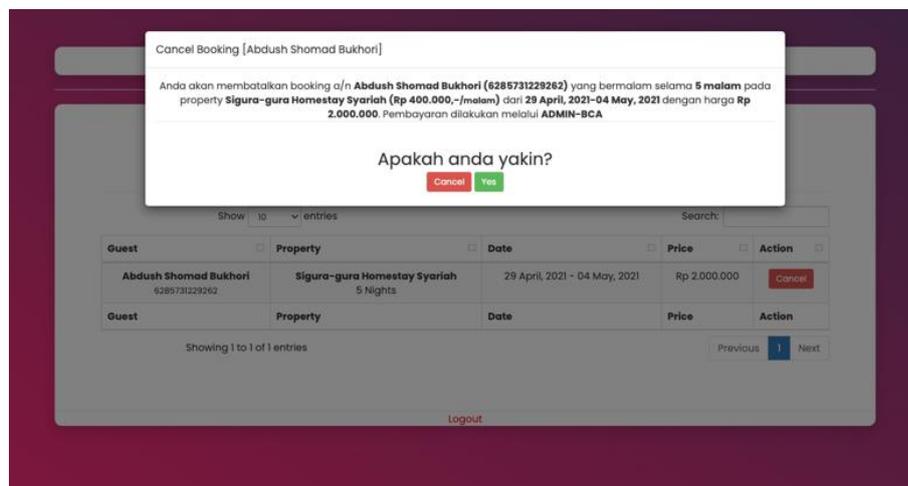
Proses ini adalah proses untuk menampilkan daftar pesanan kamar atau properti. Pada proses ini yang memiliki hak akses adalah admin. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.36 berikut.



Gambar 4. 36 *View List Booking*

- *Cancel Booking*

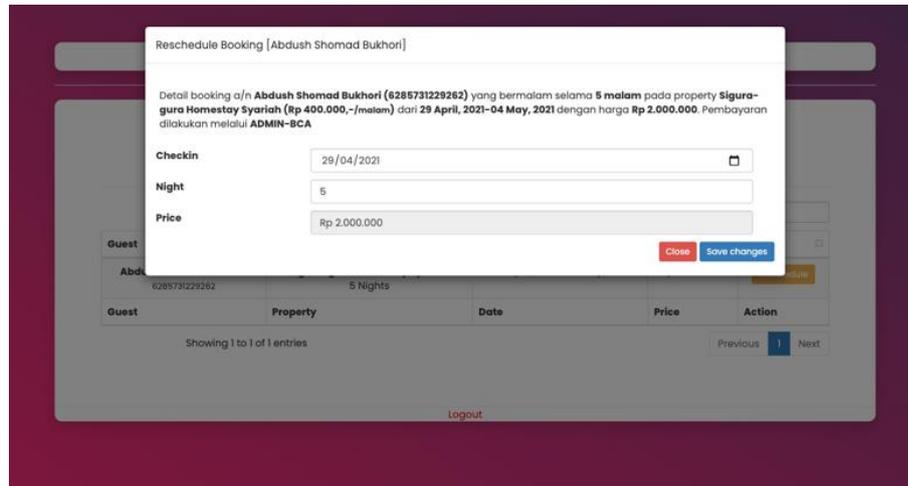
Proses ini adalah proses untuk membatalkan pemesanan kamar atau properti. Pada proses ini yang memiliki hak akses adalah owner. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.37 berikut.



Gambar 4. 37 *Cancel Booking*

- *Reschedule Booking*

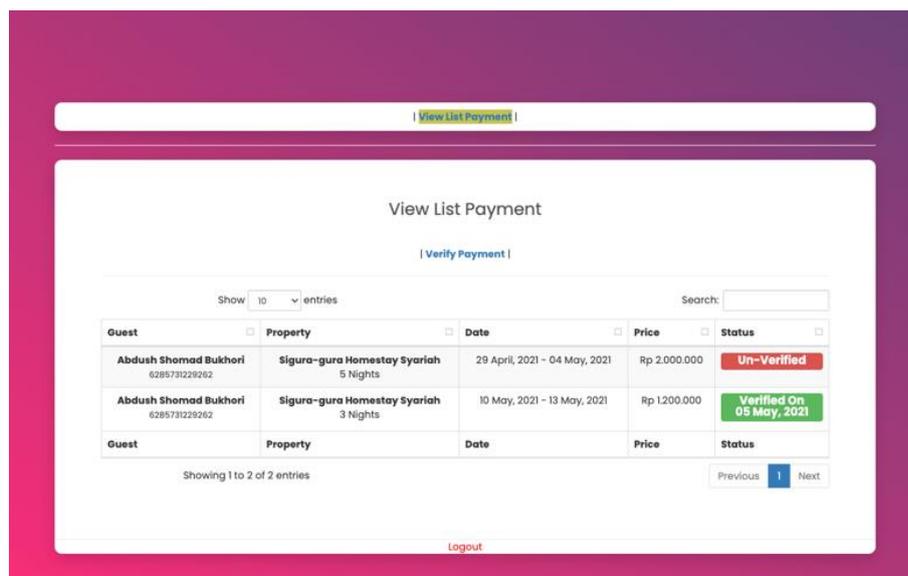
Proses ini adalah proses untuk merubah tanggal checkin pemesanan kamar atau properti. Pada proses ini yang memiliki hak akses adalah admin. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.38 berikut.



Gambar 4. 38 *Reschedule Booking*

- *View List Payment*

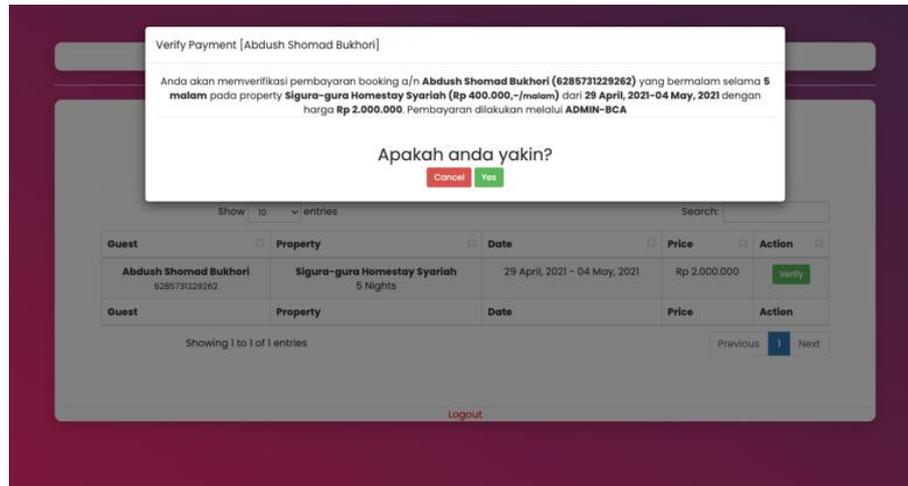
Proses ini adalah proses untuk menampilkan daftar pembayaran dari pemesanan kamar atau properti. Pada proses ini yang memiliki hak akses adalah finance. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.39 berikut.



Gambar 4. 39 *View List Payment*

- Verify Payment

Proses ini adalah proses untuk meverifikasi pembayaran dari pemesanan kamar atau properti. Pada proses ini yang memiliki hak akses adalah finance. Proses ini berjalan lancar pada keempat BPMN. Tampilan dari proses ini sebagaimana gambar 4.40 berikut.



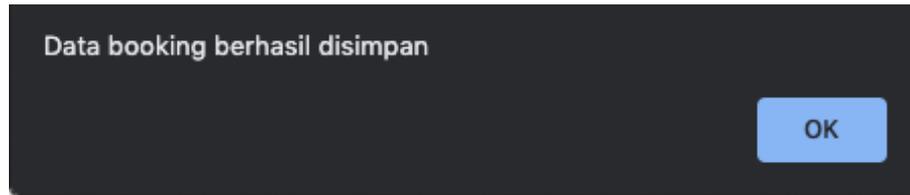
Gambar 4. 40 *Verify Payment*

Berdasarkan penjelasan di atas, sistem yang dibangun dapat menyediakan *web service* yang dapat digunakan pada versi aplikasi lain. Meskipun terdapat perbedaan alur pemesanan seperti yang ditunjukkan pada gambar 4.1, *web service* yang disediakan dapat berjalan dengan lancar. Sehingga dapat disimpulkan bahwa sistem bersifat *reusable*.

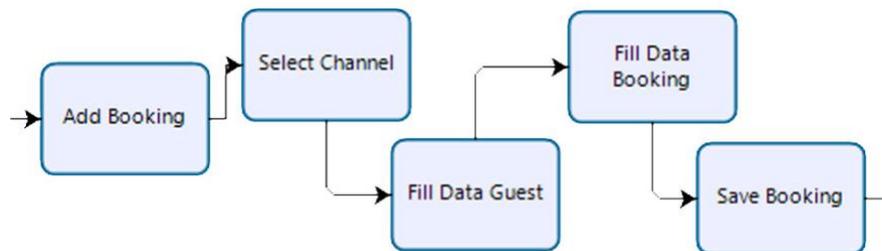
Sedangkan penjelasan dari setiap proses pada semua BPMN berdasarkan tabel 4.3 untuk pengujian *flexibility* sistem adalah sebagai berikut.

- Alur Pemesanan Berdasarkan Gambar 3.10

Alur pemesanan ini dimulai dari menampilkan halaman pengantar untuk menambahkan pemesanan. Diteruskan dengan memilih *payment channel* untuk mengetahui bahwa pemesanan tersebut datang darimana. Kemudian diarahkan ke pengisian data tamu yang dilanjutkan dengan pengisian data pemesanan. Data pemesanan tersebut harus memilih *property* atau penginapan serta tanggal *check in* dan *check out*. Proses akhir adalah menyimpan data pemesanan yang mempunyai tampilan yang sama ditunjukkan dengan gambar 4.41. Alur proses tersebut dapat dilihat pada gambar 4.42 berikut.



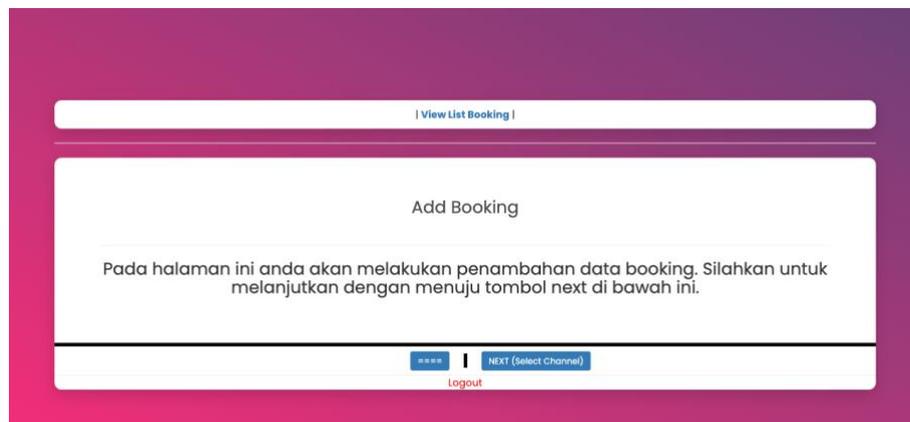
Gambar 4. 41 Alur Proses Pemesanan *Save Booking*



Gambar 4. 42 Alur Proses Pemesanan Berdasarkan BPMN 1

Hasil dari alur proses pemesanan di atas ditampilkan pada gambar 4.43 sebagai *add booking*, gambar 4.44 sebagai *select channel*, gambar 4.45 sebagai *fill data guest*, gambar 4.46 sebagai *fill data booking*.

1) *Add Booking*



Gambar 4. 43 *Add Booking* BPMN 1

2) *Select Channel*

| View List Booking |

Select Channel Booking

Anda telah memilih channel **TRAVELOKA**  
Anda dapat melanjutkan proses dengan mengakses tombol next di bawah

Channel: Traveloka

Save

PREV (Add Booking) | NEXT (Fill Data Guest)

Logout

Gambar 4. 44 *Select Channel BPMN 1*

3) *Fill Data Guest*

| View List Booking |

Fill Data Guest Booking

Anda telah mengisi data tamu a/n **Abdush Shomad Bukhori (6285731229262)**  
Anda dapat melanjutkan proses dengan mengakses tombol next di bawah

Nama: Abdush Shomad Bukhori

Nomor Whatsapp: 6285731229262

Save

PREV (Select Channel) | NEXT (Fill Data Booking)

Logout

Gambar 4. 45 *Fill Data Guest BPMN 1*

#### 4) *Fill Data Booking*

Fill Data Booking

Anda telah mengisi data booking  
**Sigura-gura Homestay Syariah (3 Nights : Rp 1.200.000)**  
10 May, 2021-13 May, 2021  
Anda dapat melanjutkan proses dengan mengklik tombol next di bawah

Property: Sigura-gura Homestay Syariah (Rp 400.000/night)

Checkin: 10/05/2021

Night: 3

Price: Rp 1.200.000

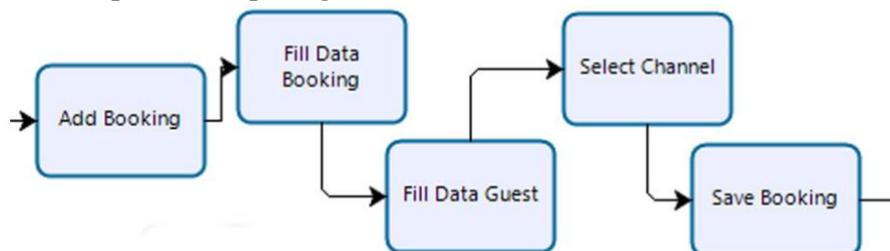
Save

PREV (Fill Data Guest) | NEXT (Save Booking) | Logout

Gambar 4. 46 *Fill Data Booking* BPMN 1

- Alur Pemesanan Berdasarkan Gambar 3.11

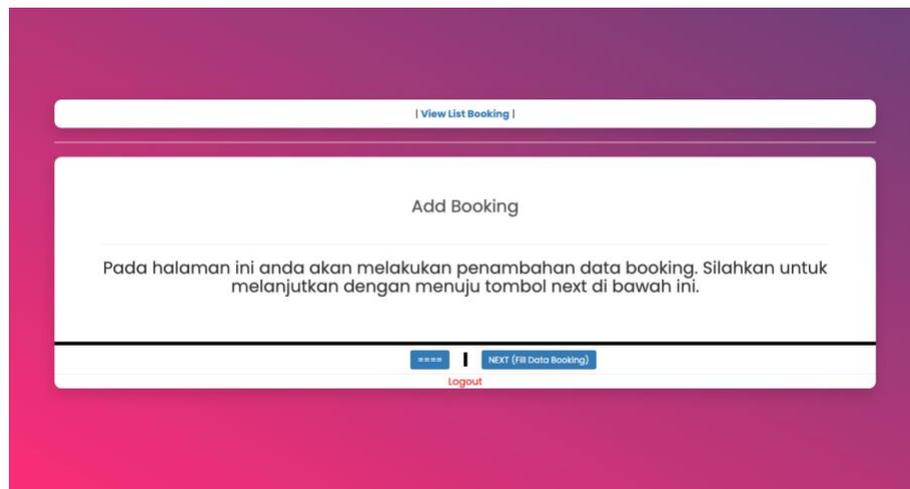
Alur pemesanan ini dimulai dari menampilkan halaman pengantar untuk menambahkan pemesanan. Diteruskan dengan mengisi data pemesanan yang mana harus memilih *property* atau penginapan serta tanggal *check in* dan *check out*. Kemudian diarahkan ke pengisian data tamu yang dilanjutkan dengan memilih *payment channel* untuk mengetahui bahwa pemesanan tersebut datang darimana. Proses akhir adalah menyimpan data pemesanan yang mempunyai tampilan yang sama ditunjukkan dengan gambar 4.41. Alur di atas dapat dilihat pada gambar 4.47 berikut.



Gambar 4. 47 Alur Proses Pemesanan Berdasarkan BPMN 2

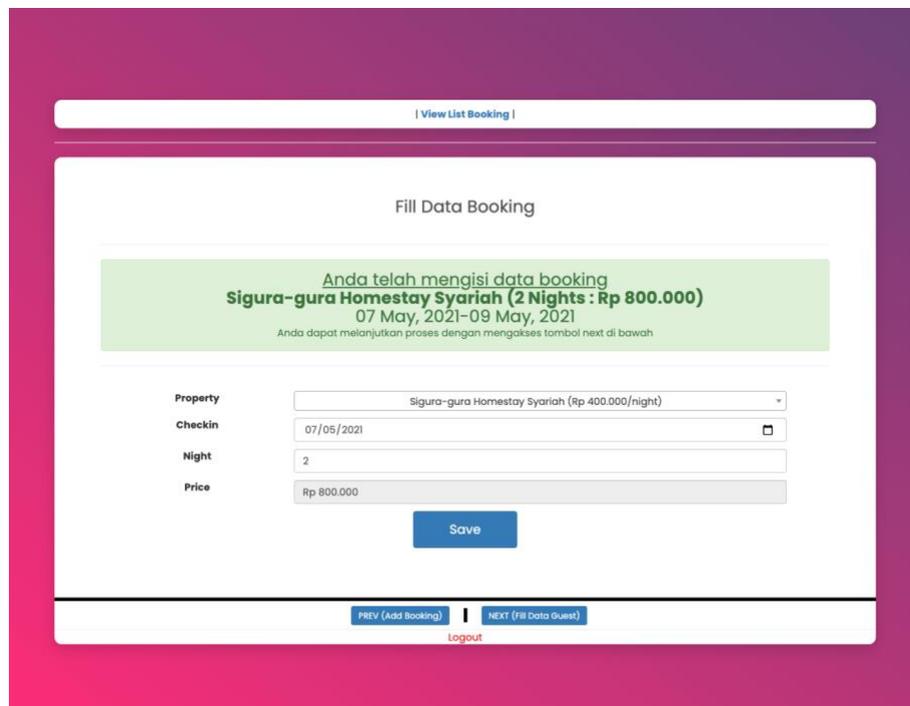
Hasil dari alur proses pemesanan di atas ditampilkan pada gambar 4.48 sebagai *add booking*, gambar 4.49 sebagai *fill data booking*, gambar 4.50 sebagai *fill data guest*, gambar 4.51 sebagai *select channel*.

1) *Add Booking*



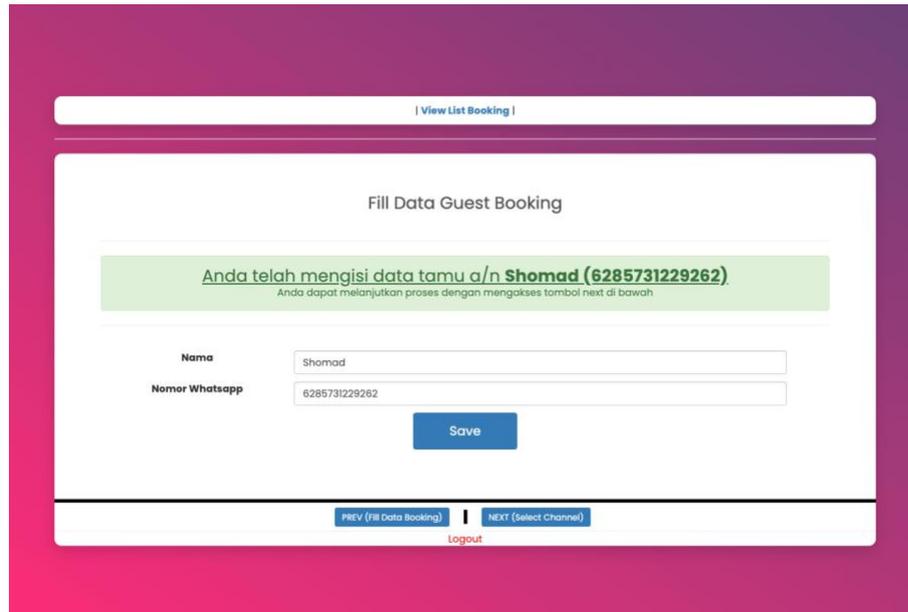
Gambar 4. 48 *Add Booking* BPMN 2

2) *Fill Data Booking*



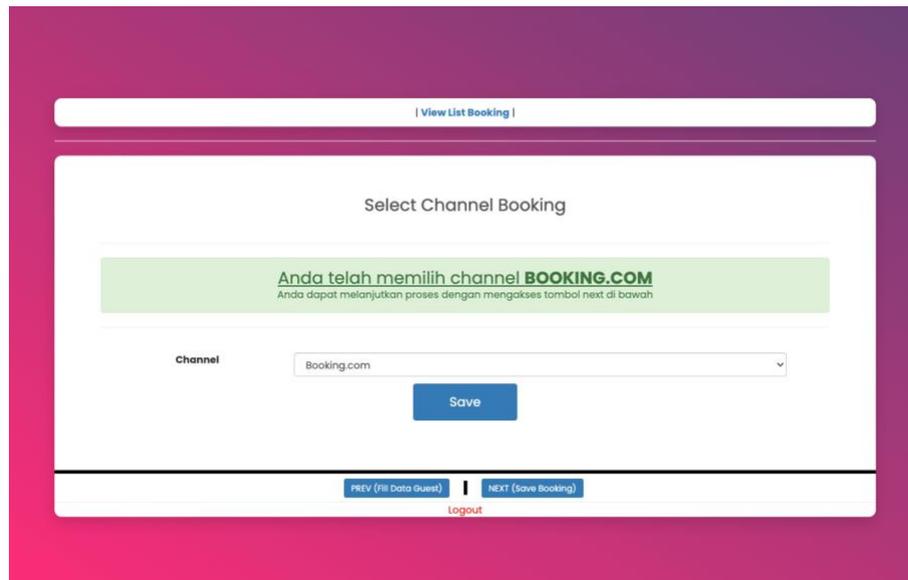
Gambar 4. 49 *Fill Data Booking* BPMN 2

### 3) *Fill Data Guest*



Gambar 4. 50 *Fill Data Guest* BPMN 2

### 4) *Select Channel*

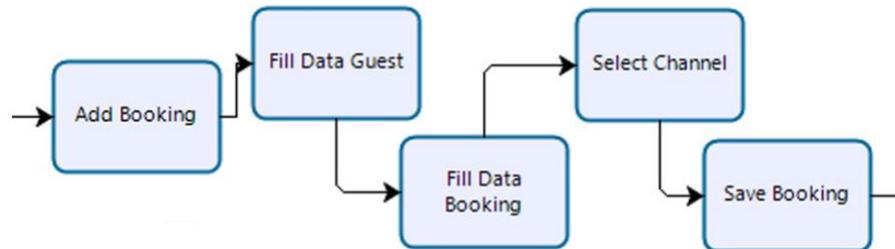


Gambar 4. 51 *Select Channel* BPMN 2

- Alur Pemesanan Berdasarkan Gambar 3.12

Alur pemesanan ini dimulai dari menampilkan halaman pengantar untuk menambahkan pemesanan. Diteruskan dengan pengisian data tamu yang dilanjutkan dengan pengisian data pemesanan. Data pemesanan tersebut harus memilih *property* atau penginapan serta tanggal *check in* dan

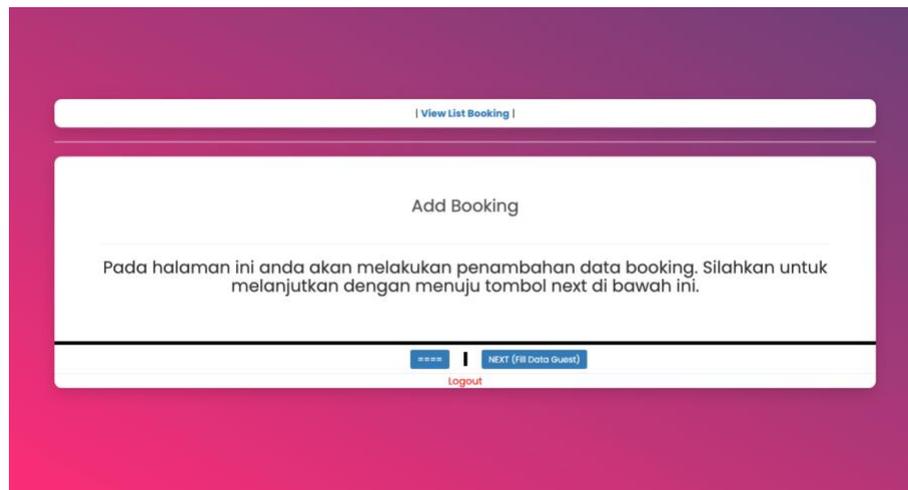
*check out*. Kemudian diarahkan ke pemilihan *payment channel* untuk mengetahui bahwa pemesanan tersebut datang darimana. Proses akhir adalah menyimpan data pemesanan yang mempunyai tampilan yang sama ditunjukkan dengan gambar 4.41. Alur di atas dapat dilihat pada gambar 4.52 berikut.



Gambar 4. 52 Alur Proses Pemesanan Berdasarkan BPMN 3

Hasil dari alur proses pemesanan di atas ditampilkan pada gambar 4.53 sebagai *add booking*, gambar 4.54 sebagai *fill data guest*, gambar 4.55 sebagai *fill data booking*, gambar 4.56 sebagai *select channel*.

1) *Add Booking*



Gambar 4. 53 *Add Booking* BPMN 3

2) *Fill Data Guest*

Fill Data Guest Booking

Anda telah mengisi data tamu a/n **Shomad (coba alur 3) (6285731229262)**  
Anda dapat melanjutkan proses dengan mengakses tombol next di bawah

Nama: Shomad (coba alur 3)  
Nomor Whatsapp: 6285731229262

Save

PREV (Add Booking) | NEXT (Fill Data Booking)  
Logout

Gambar 4. 54 *Fill Data Guest* BPMN 3

3) *Fill Data Booking*

Fill Data Booking

Anda telah mengisi data booking **Sigura-gura Homestay Syariah (4 Nights : Rp 1.600.000)**  
19 May, 2021-23 May, 2021  
Anda dapat melanjutkan proses dengan mengakses tombol next di bawah

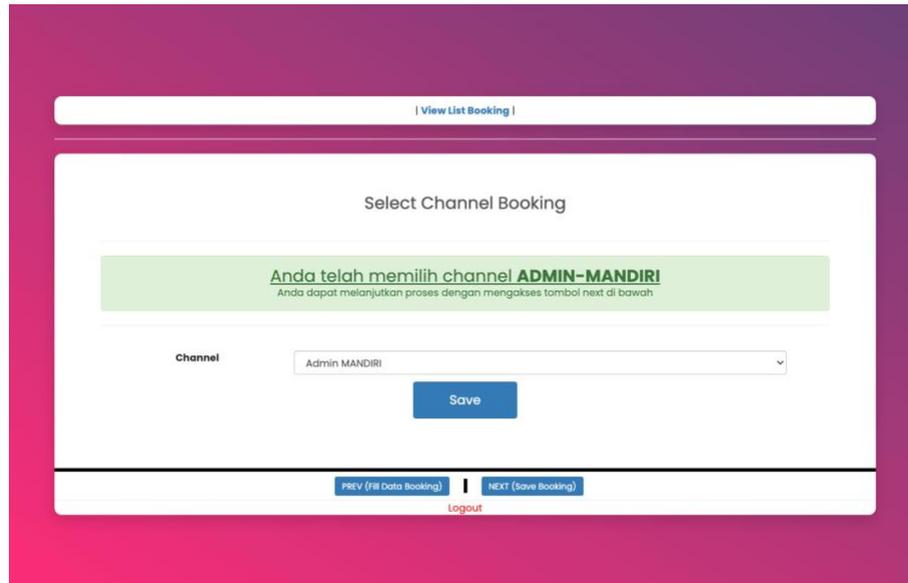
Property: Sigura-gura Homestay Syariah (Rp 400.000/night)  
Checkin: 19/05/2021  
Night: 4  
Price: Rp 1.600.000

Save

PREV (Fill Data Guest) | NEXT (Select Channel)  
Logout

Gambar 4. 55 *Fill Data Booking* BPMN 3

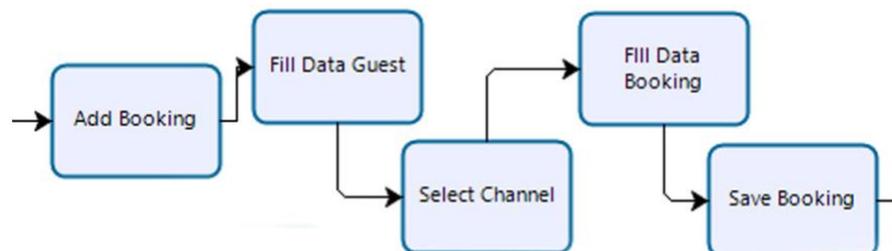
#### 4) *Select Channel*



Gambar 4. 56 *Select Channel* BPMN 3

- Alur Pemesanan Berdasarkan Gambar 3.13

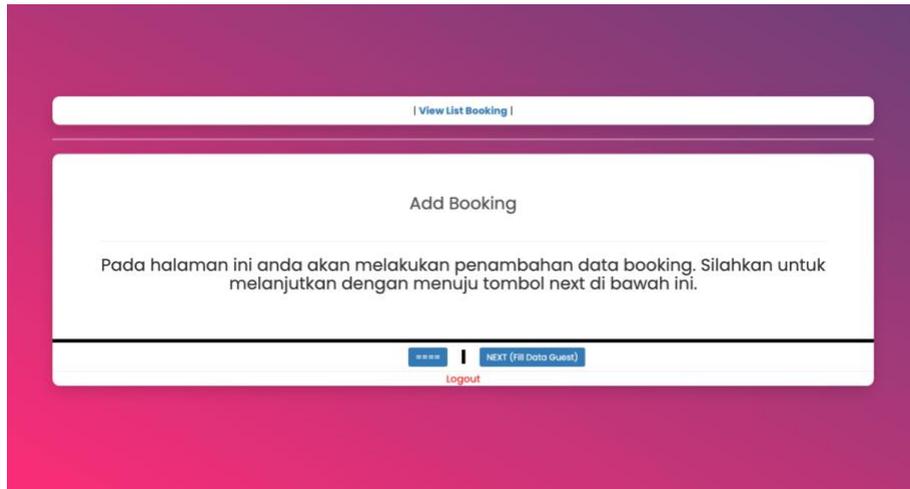
Alur pemesanan ini dimulai dari menampilkan halaman pengantar untuk menambahkan pemesanan. Diteruskan dengan pengisian data tamu yang dilanjutkan dengan pemilihan *payment channel* untuk mengetahui bahwa pemesanan tersebut datang darimana. Kemudian diarahkan ke pengisian data pemesanan. Data pemesanan tersebut harus memilih *property* atau penginapan serta tanggal *check in* dan *check out*. Proses akhir adalah menyimpan data pemesanan yang mempunyai tampilan yang sama ditunjukkan dengan gambar 4.41. Alur di atas dapat dilihat pada gambar 4.57 berikut.



Gambar 4. 57 Alur Proses Pemesanan Berdasarkan BPMN 4

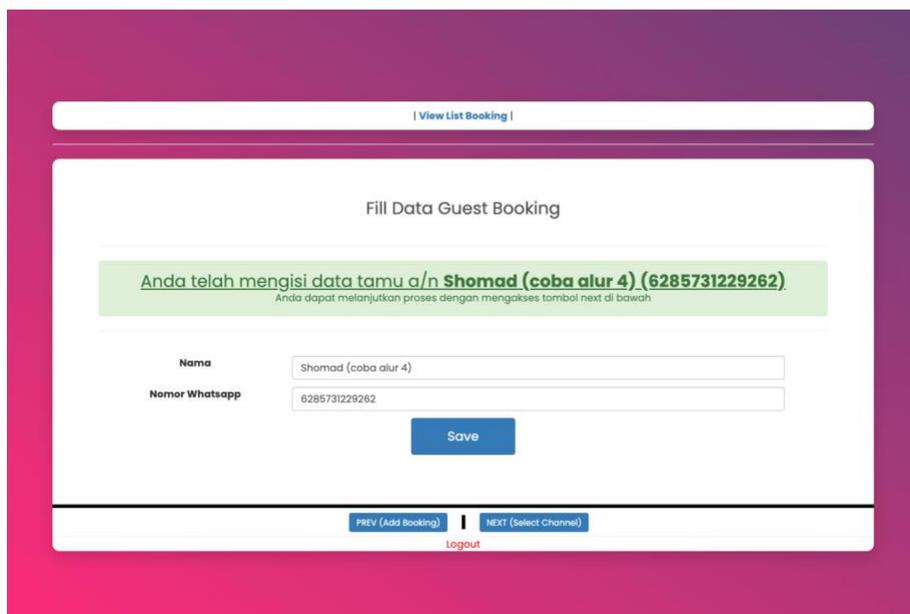
Hasil dari alur proses pemesanan di atas ditampilkan pada gambar 4.58 sebagai *add booking*, gambar 4.59 sebagai *fill data guest*, gambar 4.60 sebagai *select channel*, gambar 4.61 sebagai *fill data booking*.

### 1) Add Booking



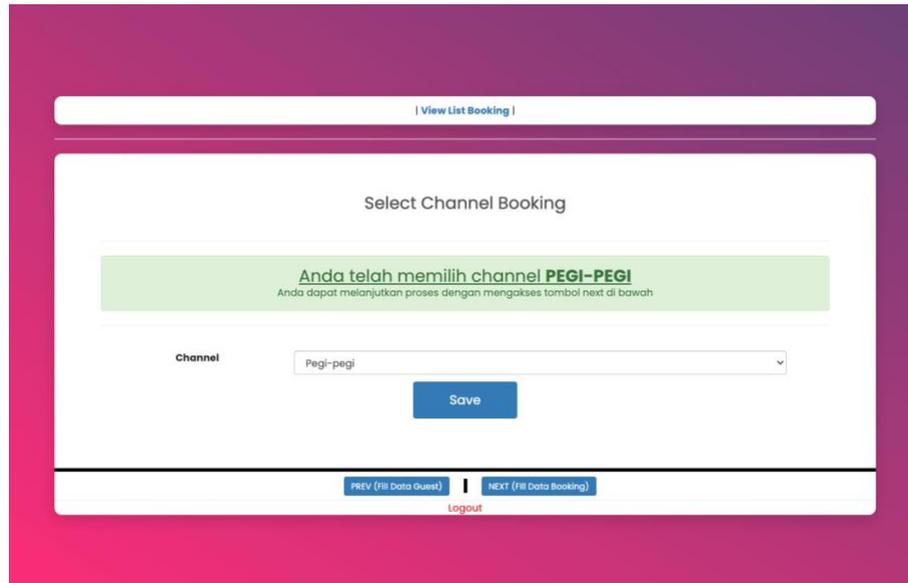
Gambar 4. 58 Add Booking BPMN 4

### 2) Fill Data Guest



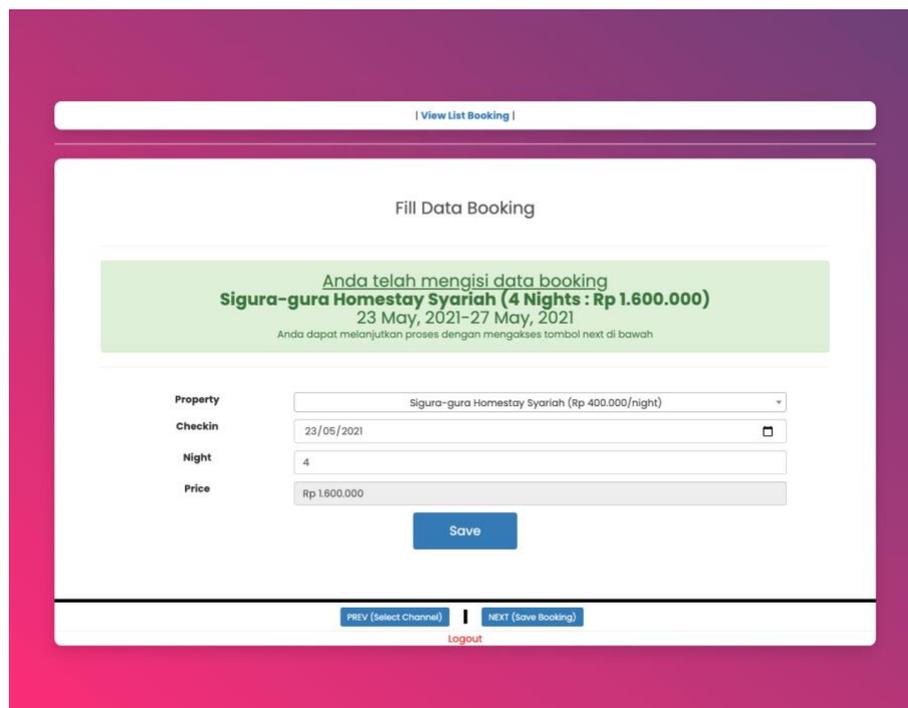
Gambar 4. 59 Fill Data Guest BPMN 4

### 3) *Select Channel*



Gambar 4. 60 *Select Channel* BPMN 4

### 4) *Fill Data Booking*



Gambar 4. 61 *Fill Data Booking* BPMN 4

Berdasarkan penjelasan di atas, sistem yang dibangun dapat mengakomodasi proses bisnis yang berbeda-beda. Proses bisnis tersebut didapatkan dari bahan uji sistem dimana ada empat BPMN yang memiliki perbedaan pada alur pemesanan

properti sebagaimana ditunjukkan pada gambar 4.42, gambar 4.47, gambar 4.52 dan gambar 4.57.

Pada BPMN Alur *Booking* 1, alur pemesanan properti dimulai dengan memilih metode pemesanan, kemudian mengisi data tamu dan data pemesanan serta yang terakhir adalah proses menyimpan pemesanan. Pada BPMN Alur *Booking* 2, alur pemesanan properti dimulai dari mengisi data pemesanan dan data tamu, kemudian memilih metode pemesanan serta yang terakhir adalah proses menyimpan pemesanan. Pada BPMN Alur *Booking* 3, proses pemesanan properti dimulai dari mengisi data tamu dan data pemesanan, kemudian memilih metode pemesanan serta yang terakhir adalah proses menyimpan pemesanan. Sedangkan pada BPMN Alur *Booking* 4, proses pemesanan properti dimulai dari mengisi data tamu, kemudian memilih metode pemesanan dan dilanjutkan mengisi data pemesanan serta yang terakhir adalah proses menyimpan pemesanan. Berdasarkan tabel 4.3, keempat BPMN tersebut dapat dijalankan dengan baik meskipun terdapat perbedaan alur proses bisnis. Sehingga dapat disimpulkan bahwa sistem bersifat *flexible*.

### **4.3. Pembahasan dan Integrasi Islam**

*Business Process Management System (BPM System)* merupakan sistem yang dirancang khusus untuk menjalankan suatu program dari proses bisnis yang dimasukkan. Sistem ini sangat bermanfaat dalam industri bisnis karena memudahkan pelaku bisnis untuk dapat menggunakan sistem sesuai dengan proses bisnis yang diberlakukan. Dengan penggunaan proses bisnis dalam sistem tersebut, pelaku bisnis ataupun para pekerja dapat bekerja dengan disiplin dan jujur. Disamping itu, sistem ini juga dapat menekan biaya pengeluaran untuk pembuatan sistem yang pada umumnya terpisah. Dengan hanya satu sistem yang dapat mengakomodasi beberapa bagian atau divisi dari sebuah perusahaan, semua informasi akan terpusat yang dapat dipantau secara penuh oleh atasan atau direktur perusahaan.

Dalam islam membahas tentang etika dalam berbisnis yang salah satunya adalah kejujuran. Dalam hal ini kejujuran berarti semua tindakan dalam berbisnis dilakukan dengan sikap jujur tanpa adanya penipuan atau yang dapat disebut sebagai bertanggung jawab. Kejujuran merupakan salah satu hal terpenting dalam

kehidupan yang tak luput juga dalam berbisnis, karena dengan kejujuran itulah yang akan membuat pelaku bisnis mendapat kepercayaan dari pelaku bisnis lain.

Allah SWT sangat menganjurkan untuk bersikap jujur dalam surat Al-Anfaal ayat 58:

وَأَمَّا تَخَافَنَّ مِنْ قَوْمٍ خِيَانَةً فَأَنْذِرْ إِلَيْهِمْ عَلَىٰ سَوَاءٍ ۗ إِنَّ اللَّهَ لَا يُحِبُّ الْخَائِبِينَ

Artinya: "Dan jika kamu khawatir akan (terjadinya) pengkhianatan dari suatu golongan, maka kembalikanlah perjanjian itu kepada mereka dengan cara yang jujur. Sesungguhnya Allah tidak menyukai orang-orang yang berkhianat."

Dari Abu Sa'id Al Khudri r.a., Rasulullah s.a.w. bersabda : *Pedagang yang jujur dan dapat dipercaya termasuk golongan para nabi, orang yang benar-benar tulus dan para syuhada* (Tirmidzi, Darimi dan Daraqutni).

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Dari penulisan yang telah dilakukan disimpulkan bahwa sistem bersifat *flexible* dimana dapat mengakomodasi proses bisnis yang berbeda-beda dan *reusable* dimana *web service* yang disediakan oleh sistem dapat digunakan oleh lebih dari satu proses bisnis. Hal ini dibuktikan dengan berjalannya program dengan baik berdasarkan proses bisnis yang dimasukkan.

#### **5.2 Saran**

Dari hasil penulisan ini, maka penulis memberikan saran yang dapat bermanfaat bagi pengembang aplikasi, pembaca dan penulis selanjutnya antara lain:

1. Penulis selanjutnya dapat menambahkan jumlah perusahaan sebagai studi kasus penelitian.
2. Penulis selanjutnya dapat menambahkan tujuan penelitian dimana sistem bersifat *scalable*.

## DAFTAR PUSTAKA

- Barreto, C., Bullard, V., Erl, T., Evdemon, J., Jordan, D., Kand, K., König, D., Moser, S., Stout, R., Ten-Hove, R., Trickovic, I., Rijn, D. van der, & Yiu, A. (2007). Web Services Business Process Execution Language Version 2.0. *OASIS Open 2007, May*, 1–66.
- Chang, J. F. (2016). *Business Process Management Systems*. Auerbach Publications. <https://doi.org/10.1201/9781420031362>
- Chotijah, U. (2019). Assesment Of Business Process Management For MSMEs In East Java. *Jurnal Sistem Informasi*, 15(1), 1–9. <https://doi.org/10.21609/jsi.v15i1.770>
- Dallas, I., & Wynn, M. T. (2014). *Business Process Management in Small Business: A Case Study*. 25–46. [https://doi.org/10.1007/978-3-642-38244-4\\_2](https://doi.org/10.1007/978-3-642-38244-4_2)
- Daly, M. (2017). *5 reasons why ESB-led integration is out and API-led integration is in*. <https://blogs.mulesoft.com/learn-apis/api-led-connectivity/5-reasons-why-esb-led-integration-is-out-and-api-led-integration-is-in/>
- Dewi, L. P., Indahyanti, U., & Hari S., Y. (2017). Pemodelan Proses Bisnis Menggunakan Activity Diagram UML Dan BPMN ( Studi Kasus Frs Online ). *Jurnal Informatika*, 1–9.
- Eden, A. H., & Mens, T. (2006). Measuring software flexibility. *IEE Proceedings: Software*, 153(3), 113–125. <https://doi.org/10.1049/ip-sen:20050045>
- Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall International.
- Fauvet, M. C., La Rosa, M., Sadegh, M., Alshareef, A., Dijkman, R. M., García-Bañuelos, L., Reijers, H. A., Van Der Aalst, W. M. P., Dumas, M., & Mendling, J. (2010). Managing process model collections with AProMoRe. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6470 LNCS(May 2014), 699–701. [https://doi.org/10.1007/978-3-642-17358-5\\_62](https://doi.org/10.1007/978-3-642-17358-5_62)
- Firdausi, A. Z. (2013). *Perencanaan Strategis Teknologi Informasi Pondok Pesantren Al-Ishlah Bondowoso Dengan Metode Tozer* [UIN Malulana Malik Ibrahim Malang]. <http://theses.uin-malang.ac.id/7571/>

- Gill, N. S. (2003). Reusability issues in component-based development. *ACM SIGSOFT Software Engineering Notes*, 28(4), 4–4. <https://doi.org/10.1145/882240.882255>
- Gricke, A., Bayer, F., & Kühn, H. (2013). Einleitung. In F. Bayer & H. Kühn (Eds.), *Prozessmanagement für Experten* (pp. 3–10). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-36995-7>
- Indrasiri, K. (2016). Beginning WSO2 ESB. *Beginning WSO2 ESB*, 1–16. <https://doi.org/10.1007/978-1-4842-2343-7>
- Irawan, A., Risa, M., Muttaqien, M. A., & Shinnay, A. E. (2017). Perancangan Sistem Informasi Penjualan Pakaian Pada Cv Nonninth Inc Berbasis Online. *POSITIF: Jurnal Sistem Dan Teknologi Informasi*, 3(2), 74–82. <https://doi.org/10.31961/positif.v3i2.417>
- Johnston, T. (2017). *Enterprise Service Bus (ESB) Tools: Technical Comparison and Review*. <https://shadow-soft.com/enterprise-service-bus-esb-tools/>
- Juric, M. B., Ma, B., & Sarang, P. (2006). Business Process Execution Language for Web Services second edition. In *Standards proposal by BEA Systems, International Business Machines Corporation, and Microsoft Corporation* (Issue May). <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Business+Process+Execution+Language+for+Web+Services#0>
- Kapojos, F., Wowor, H. F., Rumagit, A. M., & Wowor, A. P. R. (2012). Implementasi Service-Oriented Architecture Dengan Web Service Untuk Aplikasi Informasi Akademik. *Jurnal Teknik Elektro Dan Komputer*, 1(1), 1–5. <https://doi.org/10.35793/jtek.1.1.2012.554>
- Khan, M. E. (2011). Different approaches to white box testing technique for finding errors. *International Journal of Software Engineering and Its Applications*, 5(3), 1–14. <https://doi.org/10.5121/ijsea.2011.2404>
- Louridas, P. (2008). Orchestrating web services with BPEL. *IEEE Software*, 25(2), 85–87. <https://doi.org/10.1109/MS.2008.42>
- Mas'adi, F. (2014). *Hadits Meringankan Beban Orang Lain*. <https://fromlearntoearn.wordpress.com/2014/11/04/hadis-meringankan->

beban-orang-lain-fitri-masadi/

- McCormack, K., Willems, J., van den Bergh, J., Deschoolmeester, D., Willaert, P., Štemberger, M. I., Škrinjar, R., Trkman, P., Ladeira, M. B., de Oliveira, M. P. V., Vuksic, V. B., & Vlahovic, N. (2009). A global investigation of key turning points in business process maturity. *Business Process Management Journal*, *15*(5), 792–815. <https://doi.org/10.1108/14637150910987946>
- Meidan, A., García-García, J. A., Escalona, M. J., & Ramos, I. (2017). A survey on business processes management suites. *Computer Standards and Interfaces*, *51*, 71–86. <https://doi.org/10.1016/j.csi.2016.06.003>
- Mulyanto, A. (2016). Pengujian Sistem Informasi Akademik Menggunakan McCall's Software Quality Framework. *Jurnal Informatika Sunan Kalijaga*, *1*(1), 47–57.
- Ouyang, C., Dumas, M., Ter Hofstede, A. H. M., & Van Der Aalst, W. M. P. (2006). From BPMN process models to BPEL Web services. *Proceedings - ICWS 2006: 2006 IEEE International Conference on Web Services*, 285–292. <https://doi.org/10.1109/ICWS.2006.67>
- Pamulu, M. S. (2016). *Mensiasati Perubahan Lingkungan Bisnis yang Kian Tidak Menentu* (Issue August). ResearchGate. <https://www.researchgate.net/publication/307334553>
- Ramdhani, M. A. (2015). Pemodelan Proses Bisnis Sistem Akademik Menggunakan Pendekatan Business Process Modelling Notation (BPMN) (Studi Kasus Institusi Perguruan Tinggi Xyz). *Jurnal Informasi*, *7*(2), 83–93.
- Sarno, R., Fauzan, A. C., Nurlaili, A. N., Rahmawati, D., Sungkono, K. R., & Effendi, Y. A. (2017). *Manajemen Proses Bisnis*. ITS Tekno Sains.
- Sarwosri, & Naja, F. (2011). Aplikasi Pelayanan Kesehatan Berbasis Service-Oriented Architecture. *MediaNeliti*, *5*(Semantik), 1–8.
- Sutedi, S. (2018). *Rancangan Aplikasi Manajemen User untuk Mendukung Aplikasi Sistem Informasi*. July 2013. <https://doi.org/10.31224/osf.io/c6vdm>
- Vega-Márquez, O. L., Chavarriaga, J., Linares-Vásquez, M., & Sánchez, M. (2019). Requirements Comprehension Using BPMN: An Empirical Study. *Empirical*

*Studies on the Development of Executable Business Processes*, 85–111.  
[https://doi.org/10.1007/978-3-030-17666-2\\_5](https://doi.org/10.1007/978-3-030-17666-2_5)

Voas, J. M. (1998). Challenges of using COTS software in component-based development. *Computer*, 31(6), 44–45.  
<https://doi.org/10.1109/MC.1998.683006>

Yusrizal, Roslidar, R., & Dawood, R. (2017). Rancang Bangun Aplikasi Rekam Medis Untuk Praktik Pribadi Dokter Berbasis Android Dan Memanfaatkan Layanan Web. *Kitektro*, 2(1), 1–8.