

**PENGEMBANGAN SISTEM INFORMASI PERHOTELAN
MENGUNAKAN METODE *RATIONAL UNIFIED PROCESS*
(STUDI PADA EDOTEL PAMEKASAN)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Radea Zulindra Ardisukma

NIM: 155150400111066



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI PERHOTELAN
MENGUNAKAN METODE *RATIONAL UNIFIED PROCESS*
(STUDI PADA EDOTEL PAMEKASAN)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Radea Zulindra Ardisukma
NIM: 155150400111066

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Juli 2019

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II


Yusi Tyroni Mursitoyo, S.Kom., M.AB.
NIP: 198002282006041001


Issa Arwani, S.Kom., M.Sc.
NIP: 198309222012121003

Mengetahui
Ketua Jurusan Sistem Informasi




Herman Tolle, S.T., M.T.
NIP: 197408232000121001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 30 Juni 2019



Radea Zulindra Ardisukma

NIM: 155150400111066

PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga memberikan kemudahan dan kelancaran dalam menyelesaikan skripsi yang berjudul “Pengembangan Sistem Informasi Perhotelan Menggunakan Metode *Rational Unified Process* (Studi pada Edotel Pamekasan)”.

Penulis menyadari bahwa penyusunan dan penulisan skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku Ketua Program Studi Sistem Informasi dan Dosen Pembimbing I atas petunjuk, arahan, dan bimbingan yang diberikan kepada penulis selama proses pengerjaan skripsi ini.
2. Bapak Issa Arwani, S.Kom., M.Sc. selaku Dosen Pembimbing II atas petunjuk, arahan, dan bimbingan yang diberikan kepada penulis selama proses pengerjaan skripsi ini.
3. Bapak Herman Tolle, Dr.Eng., S.T., M.T. selaku Ketua Jurusan Sistem Informasi.
4. Bapak Wayan Firdaus, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer.
5. Ibu Sri Indrawati, S.Pd, M.M. selaku Kepala SMK Negeri 3 Pamekasan yang telah memberikan izin untuk melakukan penelitian di Edotel Pamekasan.
6. Kedua orang tua Bapak Iskandar Zulkarnain dan Ibu Yuli Indrawati serta Prameswari Putri Kinanti selaku adik yang selalu memberikan kasih sayang, doa, dukungan moril dan materil kepada penulis.
7. Regita Cahyani Pramesti, Rizka Amalia Zeny Putri, Agatha Shella Agtika, Fachri Rizky Wibisono, dan Filza Dikaputra pasukan “ena – ena rangers” atas dukungan, doa, kerjasama dalam perkuliahan, dan kehidupan sehari-hari sejak hari pertama kuliah hingga saat ini dan selalu hadir untuk penulis.
8. Badan Pengurus Harian (BPH) Lembaga KBMSI, teman – teman, dan adik – adik Lembaga KBMSI Periode 5 (tahun 2018/2019) terutama untuk keluarga Departemen Penelitian dan Pengembangan Studi (P2S) Periode 5 yang selalu memberikan kesempatan dan semangat untuk terus melangkah, memberikan pengalaman dalam berorganisasi dan bekerjasama di bangku perkuliahan.
9. Seluruh Keluarga Besar Sistem Informasi angkatan 2015 atas kerjasama, doa, semangat yang diberikan.
10. Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu dan berjasa dalam proses penyelesaian skripsi ini.

Akhir kata, atas segala bantuan dan dukungan semua pihak, penulis mengucapkan terima kasih banyak dan dengan kerendahan hati penulis menyadari masih bahwa skripsi ini masih jauh dari kata sempurna, maka dari itu penulis mengharapkan kritik dan saran yang dapat membangun skripsi ini.

Malang, 30 Juni 2019

Penulis

rzulindra@gmail.com



ABSTRAK

Radea Zulindra Ardisukma, Pengembangan Sistem Informasi Perhotelan Menggunakan Metode *Rational Unified Process* (Studi pada Edotel Pamekasan)

Pembimbing: Yusi Tyroni Mursityo, S.Kom., M.AB. dan Issa Arwani, S.Kom., M.Sc.

Edotel Pamekasan merupakan hotel milik SMK Negeri 3 Pamekasan yang beroperasi sebagai hotel sekaligus laboratorium pembelajaran siswa. Lokasinya yang berada di tengah kota dan dekat dengan kantor pemerintah Kabupaten Pamekasan menjadi kelebihan yang dimiliki oleh Edotel Pamekasan. Hal tersebut juga membuat Edotel Pamekasan menjadi hotel yang sering dipilih oleh tamu kedinasan. Namun sayangnya Edotel Pamekasan mulai kalah saing dengan hotel – hotel baru karena kegiatan operasionalnya masih dilakukan secara manual. Kegiatan operasional yang masih dilakukan secara manual menimbulkan berbagai permasalahan, seperti masalah tumpang tindih reservasi kamar di tanggal yang sama dan masalah kehabisan stok barang di gudang. Untuk mengatasi masalah tersebut, diusulkan solusi perbaikan proses bisnis dengan implementasi sistem informasi perhotelan. Pengembangan sistem informasi perhotelan dilakukan dengan menggunakan metode *Rational Unified Process* (RUP). Metode RUP dipilih karena kemampuan iteratifnya yang dapat menanggapi perubahan yang terjadi, ketika pemangku kepentingan tidak dapat menjelaskan kebutuhannya secara rinci. Pertama, pada fase *inception* dilakukan analisis terhadap 5 proses bisnis, yaitu proses bisnis reservasi, registrasi *check-in*, *check-out*, pengelolaan stok gudang, dan pelaporan operasional yang kemudian dibuatkan usulan perbaikannya. Setelah itu, pada fase *elaboration* dengan melalui 2 kali iterasi, didapatkan 7 fitur dengan 32 kebutuhan fungsional dan 1 kebutuhan non-fungsional sebagai dasar perancangan. Lalu, pada fase *construction* dilakukan implementasi kemudian dilanjutkan pengujian. Hasil pengujian *black-box* menunjukkan implementasi telah sesuai dengan perancangannya, sedangkan pengujian *compatibility* menghasilkan daftar rekomendasi peramban, salah satunya Google Chrome. Fase terakhir, yakni fase *transition* dilakukan instalasi sistem lalu dilanjutkan *User Acceptance Testing*. UAT yang dilakukan pada pegawai hotel menghasilkan rata – rata nilai sebesar 93,4% yang artinya pengguna sangat setuju pada solusi perbaikan proses bisnis dengan implementasi sistem informasi perhotelan. Pada akhir penelitian dihasilkan sebuah sistem informasi perhotelan yang dapat melakukan pengelolaan reservasi, registrasi tamu, pencarian kamar, pencatatan barang masuk, pencatatan barang keluar, dan pelaporan operasional.

Kata kunci: Sistem informasi, Hotel, Sistem Informasi Perhotelan, Rational Unified Process (RUP), Iterative, User Acceptance Testing (UAT)

ABSTRACT

Radea Zulindra Ardisukma, *Development of Hospitality Information Systems Using The Rational Unified Process Method (Study on Edotel Pamekasan)*

Supervisors: Yusi Tyroni Mursityo, S.Kom., M.AB. dan Issa Arwani, S.Kom., M.Sc.

Edotel Pamekasan is a hotel owned by Public Vocational High School 3 Pamekasan which is a hotel as well as a student learning laboratory. Located in the downtown and close to the government office of Pamekasan District is its advantage. It also made Edotel Pamekasan as a government employee's main choice. Unfortunately, Edotel Pamekasan began to lose competitiveness with new hotels because of its operation was done manually. Operational activities that still done manually cause various problems, such as overlapping room reservations and the warehouse's running out of stock problems. To solve those problems, business process improvement solutions with the implementation of the hospitality information system were proposed. Hospitality information systems developed with the Rational Unified Process (RUP) method. RUP was chosen because of its iterative ability that can respond to change when stakeholders are unable to explain their requirements clearly. Firstly, in the inception phase, analysis of 5 business processes and its recommendations were made, namely the business process of reservation, check-in registration, check-out, warehouse stock management, and operational reports. After that, in the elaboration phase, through 2 iterations, 7 features were obtained with 32 functional and 1 non-functional requirements for design background. Then, in the construction phase, implementation was done then continued to testing. Black-box testing shows that the implementation is in accordance with the design, while compatibility testing shows a list of recommended browsers, one of which is Google Chrome. In the final phase, in the transition phase, the system was installed then proceed to User Acceptance Testing. UAT that was done to hotel employees produces a 93.4% mean score, which means users strongly agreed on business process improvement solutions with the implementation of the hospitality information system. At the end of the study, produced a hospitality information system that can manage reservations, guest, room search, managing incoming and outgoing goods, and operational reports.

Keywords: Information Systems, Hotel, Hospitality Information Systems, Rational Unified Process (RUP), Iterative, User Acceptance Testing (UAT)

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Edotel Pamekasan.....	13
2.2.1 Visi	13
2.2.2 Misi.....	13
2.3 Sistem Informasi	14
2.4 <i>Rational Unified Process</i>	15
2.4.1 <i>Fase Rational Unified Process</i>	15
2.5 Proses Bisnis.....	17
2.5.1 <i>Business Process Modelling Notation (BPMN)</i>	17
2.6 Analisis Kebutuhan Sistem.....	22
2.7 <i>Unified Modeling Language</i>	22
2.7.1 <i>Use Case Diagram</i>	22
2.7.2 <i>Sequence Diagram</i>	24
2.7.3 <i>Class Diagram</i>	25



2.8 Data Model	26
2.8.1 Conceptual Data Model	26
2.8.2 Physical Data Model.....	26
2.9 Pengujian Sistem.....	27
2.9.1 Black-Box Testing	27
2.9.2 Compatibility Testing	28
2.10 Deployment.....	28
2.11 User Acceptance Testing (UAT).....	28
2.12 Skala Likert.....	30
BAB 3 METODOLOGI PENELITIAN	32
3.1 Studi Literatur	33
3.2 Pengumpulan Data	33
3.3 Fase <i>Inception</i>	33
3.3.1 Pemodelan Proses Bisnis Saat Ini (<i>As-Is</i>).....	34
3.3.2 Evaluasi Proses Bisnis Saat Ini (<i>As-Is</i>)	34
3.3.3 Pemodelan Proses Bisnis Usulan (<i>To-Be</i>).....	34
3.4 Fase <i>Elaboration</i>	34
3.4.1 Analisis Kebutuhan.....	34
3.4.2 Perancangan Sistem.....	35
3.5 Fase <i>Construction</i>	35
3.5.1 Implementasi Sistem.....	35
3.5.2 Pengujian Sistem	36
3.6 Fase <i>Transition</i>	36
3.6.1 <i>Deployment</i>	36
3.6.2 <i>User Acceptance Testing (UAT)</i>	36
3.7 Kesimpulan dan Saran	37
BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS KEBUTUHAN	38
4.1 Pemodelan Proses Bisnis Saat Ini (<i>As-Is</i>)	38
4.1.1 Proses Bisnis Reservasi Saat ini (<i>As-Is</i>).....	38
4.1.2 Proses Bisnis Registrasi <i>Check-In</i> Saat Ini (<i>As-Is</i>).....	43
4.1.3 Proses Bisnis <i>Check-Out</i> Saat Ini (<i>As-Is</i>)	48
4.1.4 Proses Bisnis Pengelolaan Stok Gudang Saat Ini (<i>As-Is</i>).....	50



4.1.5 Proses Bisnis Pelaporan Operasional Saat Ini (<i>As-Is</i>)	52
4.2 Evaluasi Proses Bisnis Saat Ini (<i>As-Is</i>).....	54
4.3 Pemodelan Proses Bisnis Usulan (<i>To-Be</i>).....	57
4.3.1 Proses Bisnis Reservasi Usulan (<i>To-Be</i>)	57
4.3.2 Proses Bisnis Registrasi <i>Check-In</i> Usulan (<i>To-Be</i>)	62
4.3.3 Proses Bisnis <i>Check-Out</i> Usulan (<i>To-Be</i>)	67
4.3.4 Proses Bisnis Pengelolaan Stok Gudang Usulan (<i>To-Be</i>).....	70
4.3.5 Proses Bisnis Pelaporan Operasional Usulan (<i>To-Be</i>)	73
4.4 Analisis Kebutuhan	76
4.4.1 Identifikasi Aktor	76
4.4.2 Identifikasi Fitur	77
4.4.3 Kebutuhan Fungsional Sistem.....	79
4.4.4 Kebutuhan Non-Fungsional Sistem.....	82
4.4.5 <i>Use Case Diagram</i>	83
4.4.6 <i>Use Case Scenario</i>	84
BAB 5 PERANCANGAN SISTEM.....	97
5.1 <i>Sequence Diagram</i>	97
5.2 <i>Class Diagram</i>	103
5.3 <i>Conceptual Data Model (CDM)</i>	105
5.4 <i>Physical Data Model (PDM)</i>	107
5.5 Desain Antarmuka Pengguna.....	108
BAB 6 IMPLEMENTASI SISTEM	115
6.1 Spesifikasi Sistem	115
6.1.1 Spesifikasi Perangkat Keras.....	115
6.1.2 Spesifikasi Perangkat Lunak	115
6.2 Implementasi <i>Class</i>	116
6.3 Implementasi <i>Database</i>	118
6.4 Implementasi Antarmuka Pengguna	122
BAB 7 PENGUJIAN DAN <i>DEPLOYMENT</i>	131
7.1 Pengujian Sistem.....	131
7.1.1 Pengujian Fungsional Sistem.....	131
7.1.2 Pengujian Non-Fungsional Sistem.....	150

7.2 <i>Deployment</i>	153
7.2.1 Spesifikasi Perangkat Keras di Edotel Pamekasan	153
7.2.2 Spesifikasi Perangkat Lunak di Edotel Pamekasan	153
7.3 <i>User Acceptance Testing (UAT)</i>	154
7.3.1 Perancangan <i>User Acceptance Testing (UAT)</i>	154
7.3.2 Hasil <i>User Acceptance Testing (UAT)</i>	158
7.3.3 Analisis Hasil <i>User Acceptance Testing (UAT)</i>	163
BAB 8 PENUTUP	166
8.1 Kesimpulan.....	166
8.2 Saran	167
DAFTAR REFERENSI	168



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	9
Tabel 2.2 Notasi <i>Event</i>	18
Tabel 2.3 Notasi <i>Gateway</i>	19
Tabel 2.4 Notasi <i>Activity</i>	20
Tabel 2.5 Notasi <i>Data</i>	20
Tabel 2.6 Notasi <i>Connection</i>	21
Tabel 2.7 Notasi <i>Swimlanes</i>	21
Tabel 2.8 Notasi <i>Artifacts</i>	22
Tabel 2.9 Notasi <i>Use Case Diagram</i>	23
Tabel 2.10 Dokumen <i>Use Case Description</i>	24
Tabel 2.11 Kategori Persentase	31
Tabel 4.1 Deskripsi <i>Task</i> Proses Bisnis Reservasi Saat Ini (<i>As-Is</i>).....	40
Tabel 4.2 Deskripsi <i>Task</i> Proses Bisnis Registrasi <i>Check-In</i> Saat Ini (<i>As-Is</i>)	45
Tabel 4.3 Deskripsi <i>Task</i> Proses Bisnis <i>Check-Out</i> Saat Ini (<i>As-Is</i>).....	49
Tabel 4.4 Deskripsi <i>Task</i> Proses Bisnis Pengelolaan Stok Gudang Saat Ini (<i>As-Is</i>)	51
Tabel 4.5 Deskripsi <i>Task</i> Proses Bisnis Pelaporan Operasional Saat Ini (<i>As-Is</i>)	53
Tabel 4.6 Analisis Evaluasi Proses Bisnis Reservasi.....	54
Tabel 4.7 Analisis Evaluasi Proses Bisnis Registrasi <i>Check-In</i>	54
Tabel 4.8 Analisis Evaluasi Proses Bisnis <i>Check-Out</i>	55
Tabel 4.9 Analisis Evaluasi Proses Bisnis Pengelolaan Stok Gudang.....	56
Tabel 4.10 Analisis Proses Bisnis Pelaporan Operasional	56
Tabel 4.11 Deskripsi <i>Task</i> Proses Bisnis Reservasi Usulan (<i>To-Be</i>)	59
Tabel 4.12 Analisis Hasil Perbaikan Proses Bisnis Reservasi	62
Tabel 4.13 Deskripsi <i>Task</i> Proses Bisnis Registrasi <i>Check-In</i> Usulan (<i>To-Be</i>)	64
Tabel 4.14 Analisis Hasil Perbaikan Proses Bisnis Registrasi <i>Check-In</i>	67
Tabel 4.15 Deskripsi <i>Task</i> Proses Bisnis <i>Check-Out</i> Usulan (<i>To-Be</i>).....	69
Tabel 4.16 Analisis Hasil Perbaikan Proses Bisnis <i>Check-Out</i>	70
Tabel 4.17 Deskripsi <i>Task</i> Proses Bisnis Pengelolaan Stok Gudang Usulan (<i>To-Be</i>)	72
Tabel 4.18 Analisis Hasil Perbaikan Proses Bisnis Pengelolaan Stok Gudang.....	73
Tabel 4.19 Deskripsi <i>Task</i> Proses Bisnis Pelaporan Operasional Usulan (<i>To-Be</i>)..	75

Tabel 4.20 Analisis Hasil Perbaikan Proses Bisnis Pelaporan Operasional	76
Tabel 4.21 Identifikasi Aktor	76
Tabel 4.22 Identifikasi Fitur.....	77
Tabel 4.23 Kebutuhan Fungsional Sistem	79
Tabel 4.24 Kebutuhan Fungsional yang Dihapus	82
Tabel 4.25 Kebutuhan Non-Fungsional Sistem	82
Tabel 4.26 Hubungan <i>Use Case</i> dengan Fitur	83
Tabel 4.27 <i>Use Case Scenario</i> Mengelola Reservasi	84
Tabel 4.28 <i>Use Case Scenario</i> Mengelola Registrasi Tamu	86
Tabel 4.29 <i>Use Case Scenario</i> Melihat Kamar Tersedia	89
Tabel 4.30 <i>Use Case Scenario</i> Mengelola Barang Masuk	90
Tabel 4.31 <i>Use Case Scenario</i> Mengelola Barang Keluar	92
Tabel 4.32 <i>Use Case Scenario</i> Melihat Laporan	94
Tabel 6.1 Spesifikasi Perangkat Keras	115
Tabel 6.2 Spesifikasi Perangkat Lunak	116
Tabel 6.3 Implementasi Class	116
Tabel 7.1 Rancangan Skenario Mengelola Reservasi.....	131
Tabel 7.2 Rancangan Kasus Uji Menambah Reservasi.....	132
Tabel 7.3 Rancangan Kasus Uji Gagal Menambah Reservasi.....	132
Tabel 7.4 Rancangan Skenario Mengelola Registrasi Tamu	133
Tabel 7.5 Rancangan Kasus Uji Menambah Registrasi <i>Check-In</i>	134
Tabel 7.6 Rancangan Kasus Uji Gagal Menambah Registrasi <i>Check-In</i>	134
Tabel 7.7 Rancangan Kasus Uji Menambah Tagihan Tamu	135
Tabel 7.8 Rancangan Kasus Uji Gagal Menambah Tagihan Tamu	136
Tabel 7.9 Rancangan Skenario Melihat Kamar Tersedia.....	137
Tabel 7.10 Rancangan Kasus Uji Melihat Kamar Tersedia	137
Tabel 7.11 Rancangan Kasus Uji Gagal Melihat Kamar Tersedia	138
Tabel 7.12 Rancangan Skenario Mengelola Barang Masuk.....	139
Tabel 7.13 Rancangan Kasus Uji Menambah Barang Masuk.....	139
Tabel 7.14 Rancangan Kasus Uji Gagal Menambah Barang Masuk.....	140
Tabel 7.15 Rancangan Kasus Uji Gagal Menambah Barang Masuk.....	140
Tabel 7.16 Rancangan Skenario Mengelola Barang Keluar	141

Tabel 7.17 Rancangan Kasus Uji Menambah Barang Keluar	142
Tabel 7.18 Rancangan Kasus Uji Gagal Menambah Barang Keluar	143
Tabel 7.19 Rancangan Skenario Melihat Laporan.....	144
Tabel 7.20 Rancangan Pengujian Melihat Laporan.....	144
Tabel 7.21 Rancangan Pengujian Gagal Melihat Laporan.....	144
Tabel 7.22 Hasil Pengujian Fungsional Sistem	146
Tabel 7.23 Rancangan Peramban yang Digunakan.....	150
Tabel 7.24 Spesifikasi Perangkat Keras di Edotel Pamekasan	153
Tabel 7.25 Spesifikasi Perangkat Lunak di Edotel Pamekasan.....	154
Tabel 7.26 Perancangan <i>User Acceptance Testing</i> pada Pengguna Resepsionis	155
Tabel 7.27 Perancangan <i>User Acceptance Testing</i> pada Pengguna <i>Store Keeper</i>	156
Tabel 7.28 Perancangan <i>User Acceptance Testing</i> pada Pengguna Manajer.....	156
Tabel 7.29 Perancangan <i>Acceptance Test Plan (ATP)</i>	157
Tabel 7.30 Hasil <i>User Acceptance Testing</i> pada Pengguna Resepsionis.....	158
Tabel 7.31 Rekapitulasi Hasil <i>User Acceptance Testing</i> pada Pengguna Resepsionis	160
Tabel 7.32 Hasil <i>User Acceptance Testing</i> pada Pengguna <i>Store Keeper</i>	160
Tabel 7.33 Rekapitulasi Hasil <i>User Acceptance Testing</i> pada Pengguna <i>Store Keeper</i>	161
Tabel 7.34 Hasil <i>User Acceptance Testing</i> pada Pengguna Manajer	162
Tabel 7.35 Rekapitulasi Hasil <i>User Acceptance Testing</i> pada Pengguna Manajer	163
Tabel 7.36 Hasil <i>User Acceptance Testing</i> Pengguna Resepsionis.....	164
Tabel 7.37 Hasil <i>User Acceptance Testing</i> Pengguna <i>Store Keeper</i>	164
Tabel 7.38 Hasil <i>User Acceptance Testing</i> Pengguna Manajer	165
Tabel 7.39 Rata – Rata Hasil <i>User Acceptance Testing</i>	165

DAFTAR GAMBAR

Gambar 2.1 Edotel Pamekasan	13
Gambar 2.2 <i>Information Systems Knowledge</i>	14
Gambar 2.3 <i>Rational Unified Process</i>	16
Gambar 2.4 <i>Use Case Diagram</i>	23
Gambar 2.5 <i>Sequence Diagram</i>	25
Gambar 2.6 <i>Class Diagram</i>	25
Gambar 2.7 <i>Conceptual Data Model</i>	26
Gambar 2.8 <i>Physical Data Model</i>	27
Gambar 3.1 Diagram Alir Penelitian	32
Gambar 4.1 Pemodelan Proses Bisnis Reservasi Saat Ini (<i>As-Is</i>)	39
Gambar 4.2 Pemodelan Proses Bisnis Registrasi <i>Check-In</i> Saat Ini (<i>As-Is</i>)	44
Gambar 4.3 Pemodelan Proses Bisnis <i>Check-Out</i> Saat Ini (<i>As-Is</i>)	48
Gambar 4.4 Pemodelan Proses Bisnis Pengelolaan Stok Gudang Saat ini (<i>As-Is</i>)	50
Gambar 4.5 Pemodelan Proses Bisnis Pelaporan Operasional Saat Ini (<i>As-Is</i>)	52
Gambar 4.6 Pemodelan Proses Bisnis Reservasi Usulan (<i>To-Be</i>)	58
Gambar 4.7 Pemodelan Proses Bisnis Registrasi <i>Check-In</i>	63
Gambar 4.8 Pemodelan Proses Bisnis <i>Check-Out</i>	68
Gambar 4.9 Pemodelan Proses Bisnis Pengelolaan Stok Gudang	71
Gambar 4.10 Pemodelan Proses Bisnis Pelaporan Operasional	74
Gambar 4.11 <i>Use Case Diagram</i>	83
Gambar 5.1 <i>Sequence Diagram</i> Menambah Reservasi	97
Gambar 5.2 <i>Sequence Diagram</i> Menambah Registrasi <i>Check-In</i>	98
Gambar 5.3 <i>Sequence Diagram</i> Menambah Tagihan Tamu	99
Gambar 5.4 <i>Sequence Diagram</i> <i>Check-Out</i>	100
Gambar 5.5 <i>Sequence Diagram</i> Melihat Kamar Tersedia	100
Gambar 5.6 <i>Sequence Diagram</i> Menambah Barang Masuk	101
Gambar 5.7 <i>Sequence Diagram</i> Menambah Barang Keluar	102
Gambar 5.8 <i>Sequence Diagram</i> Melihat Laporan Bulanan Transaksi Tamu	103
Gambar 5.9 <i>Class Diagram</i> <i>Controller</i>	104
Gambar 5.10 <i>Class Diagram</i> <i>Model</i>	105
Gambar 5.11 <i>Conceptual Data Model</i>	106

Gambar 5.12 <i>Physical Data Model</i>	107
Gambar 5.13 <i>Wireframe Login</i>	108
Gambar 5.14 <i>Wireframe Beranda Reservasi</i>	109
Gambar 5.15 <i>Wireframe Tambah Reservasi</i>	109
Gambar 5.16 <i>Wireframe Tambah Registrasi Check-In</i>	110
Gambar 5.17 <i>Wireframe Beranda Guest List</i>	111
Gambar 5.18 <i>Wireframe Detail Tamu</i>	111
Gambar 5.19 <i>Wireframe Cari Kamar</i>	112
Gambar 5.20 <i>Wireframe Beranda Barang Masuk</i>	112
Gambar 5.21 <i>Wireframe Tambah Barang Masuk</i>	113
Gambar 5.22 <i>Wireframe Tambah Beranda Laporan</i>	113
Gambar 6.1 Implementasi Tabel reservasi	118
Gambar 6.2 Implementasi Tabel reservasi_kamar	119
Gambar 6.3 Implementasi Tabel tamu_hotel	119
Gambar 6.4 Implementasi Tabel tagihan_tamu	120
Gambar 6.5 Implementasi Tabel kamar	120
Gambar 6.6 Implementasi Tabel barang_masuk	121
Gambar 6.7 Implementasi Tabel barang_keluar	121
Gambar 6.8 Implementasi Tabel barang.....	122
Gambar 6.9 Implementasi Antarmuka Login	122
Gambar 6.10 Implementasi Antarmuka Beranda Reservasi	123
Gambar 6.11 Implementasi Antarmuka Tambah Reservasi	124
Gambar 6.12 Implementasi Antarmuka Tambah Registrasi <i>Check-In</i>	125
Gambar 6.13 Implementasi Antarmuka Beranda <i>Guest List</i>	126
Gambar 6.14 Implementasi Antarmuka Detail Tamu	127
Gambar 6.15 Implementasi Antarmuka Detail Tamu yang Sudah <i>Check-Out</i>	127
Gambar 6.16 Implementasi Antarmuka Cari Kamar	128
Gambar 6.17 Implementasi Antarmuka Beranda Barang Masuk	128
Gambar 6.18 Implementasi Antarmuka Tambah Barang Masuk	129
Gambar 6.19 Implementasi Antarmuka Beranda Laporan	130
Gambar 7.1 Kategori <i>Compatibility Issues</i>	151
Gambar 7.2 Hasil Pengujian <i>Compatibility</i>	152



DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA PERMASALAHAN	170
LAMPIRAN B PERSETUJUAN KEBUTUHAN PENGGUNA	172
LAMPIRAN C HASIL PENGUJIAN <i>USER ACCEPTANCE TESTING</i> (UAT).....	173



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Semakin canggih dan pesatnya perkembangan teknologi informasi membuat pemanfaatannya menjadi semakin luas di dalam kehidupan manusia dan di dalam berbagai kegiatan bisnis, termasuk kegiatan bisnis perhotelan. Banyak bisnis perhotelan yang telah memanfaatkan teknologi informasi sebagai salah satu cara untuk menghadapi permasalahan globalisasi, persaingan bisnis yang semakin ketat, maraknya kecenderungan pelanggan untuk mudah berpindah, yang diikuti oleh meningkatnya ekspektasi (Melian-Gonzalez dan Bulchand-Gidumal, 2016). Masalah tersebut menyebabkan banyak hotel mulai sadar untuk melakukan investasi pada teknologi informasi, termasuk Edotel Pamekasan. Edotel Pamekasan juga ingin mengatasi permasalahan yang terjadi dan ingin meningkatkan kualitasnya sehingga mampu bersaing dengan hotel – hotel yang lain.

Edotel Pamekasan merupakan hotel milik sekolah kejuruan negeri yang ada di Kabupaten Pamekasan, yaitu SMK Negeri 3 Pamekasan. Edotel Pamekasan merupakan sebuah hotel yang juga digunakan sebagai laboratorium pembelajaran siswa jurusan perhotelan di SMK Negeri 3 Pamekasan. Hal tersebut membuat kegiatan operasional hotel banyak dibantu oleh siswa jurusan perhotelan yang sedang magang. Awal mulanya, hotel yang berdiri sejak tahun 2003 ini hanya memiliki 2 kamar dan diperuntukkan sebagai laboratorium pembelajaran siswa. Namun seiring berjalannya waktu, Edotel Pamekasan mendapatkan banyak bantuan dari Direktorat Pendidikan Menengah dan Kejuruan (DIKMENJUR) dan bantuan dana *sharing* dari pemerintah Kabupaten Pamekasan sehingga Edotel Pamekasan dapat mengembangkan fasilitas hotelnya. Bantuan yang didapatkan, dipergunakan untuk menambah jumlah kamar hotel, hingga saat ini Edotel Pamekasan memiliki total 18 kamar dengan rincian 14 kamar dengan tipe *standart*, 1 kamar dengan tipe *superior*, 2 kamar dengan tipe *deluxe*, dan 1 kamar dengan tipe *executive*. Sekarang, hotel yang pada awal berdirinya hanya digunakan sebagai laboratorium pembelajaran siswa, kini beroperasi sebagai hotel sekaligus laboratorium pembelajaran siswa.

Hal yang menarik dari Edotel Pamekasan adalah letak atau lokasinya yang termasuk di tengah kota dan tidak jauh dari kantor pemerintah Kabupaten Pamekasan yang menjadi kelebihan yang dimiliki oleh Edotel Pamekasan. Hal tersebut juga membuat Edotel Pamekasan menjadi hotel yang direkomendasikan dan sering dipilih oleh tamu kedinasan hingga pejabat pemerintahan. Namun sayangnya Edotel Pamekasan mulai kalah saing dengan hotel – hotel yang baru bermunculan, hal tersebut dikarenakan seluruh kegiatan operasionalnya masih dilakukan secara manual. Kegiatan operasional yang masih dilakukan secara manual menimbulkan berbagai macam permasalahan.

Berdasarkan hasil observasi dan wawancara dengan *general manager* Edotel Pamekasan pada 30 Desember 2018, Ibu Yuli memaparkan lebih lanjut

permasalahan yang dihadapi. Permasalahan pertama, yaitu terjadinya tumpang tindih pesanan kamar antara reservasi dengan reservasi lainnya di tanggal yang sama ataupun dengan tamu yang melakukan *check-in* langsung. Dari permasalahan tersebut, timbul dampak yang merugikan pihak hotel. Akibat terjadinya tumpang tindih pesanan kamar, pelanggan merasa kecewa karena tidak mendapatkan kamar yang dipesan yang menyebabkan hilangnya kepercayaan pelanggan terhadap Edotel Pamekasan. Kerugian finansial juga dialami Edotel Pamekasan ketika harus memberikan kompensasi kelalaian pihak hotel berupa penggantian kamar ke kamar baru yang memiliki *room type* satu tingkat lebih tinggi dari kamar yang dipesan. Kedua, terjadinya masalah kehabisan stok barang di gudang yang disebabkan karena tidak ada proses bisnis pengelolaan maupun pencatatan inventaris beserta stok barang di gudang. Untuk melakukan kegiatan *re-stock* barang, pegawai langsung mengambil barangnya di gudang. Namun jika ternyata barang yang dibutuhkan habis, maka pegawai melapor ke bagian *accounting* yang kemudian pegawai *accounting* akan memesan barang berdasarkan laporan yang masuk. Ketika barang habis, untuk melakukan *re-stock* perlu menunggu pesanan barang datang dari Surabaya yang dapat memakan waktu sehari – hari. Masalah ketiga, kegiatan operasional yang masih dilakukan secara manual dapat menyebabkan hilang atau terselipnya tagihan tamu hotel (*bill*) maupun nota – nota pembelian barang yang dibutuhkan untuk membuat laporan operasional. Potensi kerugian yang dapat terjadi adalah hilangnya *bill* maupun nota pembelian yang dapat mengakibatkan kecacatan pada laporan keuangan. Selain itu, tulisan yang ada pada nota dapat hilang seiring berjalannya waktu sehingga tidak lagi bisa dibaca.

Untuk mengatasi permasalahan dan dampak yang ada, diusulkan sebuah solusi perbaikan proses bisnis dengan menggunakan implementasi sistem informasi perhotelan yang mampu melakukan pengelolaan kegiatan operasional reservasi, registrasi *check-in*, *check-out*, pengelolaan inventaris dan stok barang di gudang, serta otomatisasi laporan bulanan transaksi tamu dan laporan bulanan penggunaan barang. Solusi ini dipilih karena sistem informasi memungkinkan untuk mempercepat pekerjaan, meningkatkan pengetahuan dalam memahami pekerjaan, meningkatkan efektivitas, mempermudah pekerjaan, dan dapat melakukan kontrol pekerjaan, sehingga akan sulit jika pengerjaannya masih dilakukan secara manual (Regar, Areros dan Asaloei, 2018). Pengembangan sistem informasi perhotelan dilakukan dengan melakukan penerapan *Software Development Life Cycle* metode *Rational Unified Process*. Metode *Rational Unified Process* yang disingkat RUP ialah sebuah proses pemodelan perangkat lunak yang menyajikan pengembangan perangkat lunak ke dalam 4 fase, yakni fase pertama adalah fase *inception*, fase kedua merupakan fase *elaboration*, fase ketiga ialah fase *construction*, dan fase terakhir sekaligus fase keempat adalah fase *transition*. *Inception* memiliki tugas menetapkan proses bisnis, *elaboration* mendefinisikan kebutuhan dan perancangan, *construction* melakukan implementasi lalu mengujinya, dan *transition* mengenalkan sistem pada lingkungannya (Sommerville, 2011). RUP dipilih karena beberapa keunggulannya, yaitu pengembangannya dapat dilakukan secara iteratif sehingga pengembangan

dapat dimulai dari prioritas kebutuhan, serta memiliki pengelolaan kebutuhan yang secara jelas mendokumentasikan kebutuhan dan melacak perubahan yang terjadi ke dalam dokumen, untuk keperluan analisis dampak perubahan sistem (Sommerville, 2011). RUP dipilih karena pemangku kepentingan tidak dapat menjelaskan kebutuhan dan keinginannya secara rinci, sehingga diperlukan iterasi untuk dapat menanggapi perubahan kebutuhan yang terjadi. Selain itu, arsitektur sistem yang menjadi fokus RUP dapat membantu memastikan sistem yang dikembangkan mempunyai kualitas yang baik dan memenuhi ekspektasi pemangku kepentingan.

Berdasarkan latar belakang permasalahan yang terjadi dan dampaknya, peneliti ingin melakukan penelitian dengan tujuan menyelesaikan masalah serta membantu Edotel Pamekasan mengembangkan bisnisnya dengan judul "Pengembangan Sistem Informasi Perhotelan Menggunakan Metode Rational Unified Process (Studi pada Edotel Pamekasan)". Sistem informasi yang dihasilkan penelitian ini, harapannya bisa digunakan oleh Edotel Pamekasan sebagai pengganti kegiatan operasional yang masih dilakukan secara manual.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka disusun rumusan masalah sebagai berikut:

1. Bagaimana hasil pemodelan proses bisnis saat ini, hasil evaluasi proses bisnis saat ini, dan hasil pemodelan proses bisnis usulan dalam fase *inception* ketika mengembangkan sistem informasi perhotelan di Edotel Pamekasan?
2. Bagaimana hasil analisis kebutuhan dan hasil perancangan sistem dalam fase *elaboration* ketika mengembangkan sistem informasi perhotelan di Edotel Pamekasan?
3. Bagaimana hasil implementasi sistem dan analisis hasil pengujian sistem dalam fase *construction* ketika mengembangkan sistem informasi perhotelan di Edotel Pamekasan?
4. Bagaimana hasil *deployment* dan analisis hasil *User Acceptance Testing* (UAT) dalam fase *transition* ketika mengembangkan sistem informasi perhotelan di Edotel Pamekasan?

1.3 Tujuan

Tujuan dari penelitian adalah sebagai berikut:

1. Mendapatkan hasil pemodelan proses bisnis saat ini, hasil evaluasi proses bisnis saat ini, dan hasil pemodelan proses bisnis usulan dalam fase *inception* ketika mengembangkan sistem informasi perhotelan di Edotel Pamekasan.
2. Mendapatkan hasil analisis kebutuhan dan hasil perancangan sistem dalam fase *elaboration* ketika mengembangkan sistem informasi perhotelan di Edotel Pamekasan.

3. Mendapatkan hasil implementasi sistem dan analisis hasil pengujian sistem dalam fase *construction* ketika mengembangkan sistem informasi perhotelan di Edotel Pamekasan.
4. Mendapatkan hasil *deployment* dan analisis hasil *User Acceptance Testing* (UAT) dalam fase *transition* ketika mengembangkan sistem informasi perhotelan di Edotel Pamekasan.

1.4 Manfaat

Adapun manfaat yang dapat diberikan setelah penelitian ini dilakukan adalah sebagai berikut:

1. Bagi pegawai Edotel Pamekasan, hasil penelitian ini dapat digunakan untuk mengatasi permasalahan yang ada pada kegiatan operasional di Edotel Pamekasan serta dapat meningkatkan efisiensi pelaksanaan kegiatan operasional hotel.
2. Bagi manajer Edotel Pamekasan, hasil penelitian ini dapat membantu manajer dalam melakukan pengawasan kegiatan operasional di Edotel Pamekasan.

1.5 Batasan Masalah

Berdasarkan latar belakang, rumusan masalah serta tujuan penelitian yang telah dijabarkan di atas, dapat disimpulkan bahwa batasan masalah pada penelitian ini adalah sebagai berikut:

1. Studi kasus dari penelitian ini adalah hotel Edotel Pamekasan.
2. Lingkup kerja yang diteliti adalah manajemen operasional hotel berupa kegiatan reservasi, registrasi *check-in*, *check-out*, pengelolaan inventaris dan stok barang di gudang, dan pelaporan bulanan transaksi tamu hotel serta penggunaan barang di edoTEL Pamekasan.
3. Solusi pengembangan sistem informasi perhotelan dikembangkan dengan menerapkan bahasa pemrograman PHP dan *database* MySQL.
4. Menggunakan metode *Rational Unified Process* (RUP) yang menerapkan fase *inception*, fase *elaboration*, fase *construction*, serta fase *transition*.
5. Tahap *deployment* pada fase *transition* dilakukan hanya sebatas melakukan instalasi sistem untuk menguji penerimaan pengguna.

1.6 Sistematika Pembahasan

Sistematika pembahasan bertujuan untuk memberikan uraian dari penulisan skripsi secara garis besar, yaitu:

BAB 1

PENDAHULUAN

Pendahuluan berisi bahasan mengenai latar belakang penelitian, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika pembahasan.

BAB 2**LANDASAN KEPUSTAKAAN**

Landasan kepastakaan memuat kajian pustaka yang berisi penelitian sejenis yang digunakan sebagai acuan dalam melakukan penelitian ini dan juga dasar teori yang digunakan dalam melakukan pengembangan sistem informasi perhotelan Edotel Pamekasan.

BAB 3**METODOLOGI PENELITIAN**

Metodologi penelitian membahas metode serta tahapan – tahapan yang dilakukan mulai dari studi kepastakaan hingga penarikan kesimpulan dan saran penelitian.

BAB 4**PEMODELAN PROSES BISNIS DAN ANALISIS KEBUTUHAN**

Pemodelan proses bisnis dan analisis kebutuhan membahas mengenai kondisi organisasi saat ini beserta permasalahannya yang dimodelkan ke dalam pemodelan proses bisnis saat ini (*as-is*), rekomendasi perbaikan yang dimodelkan ke dalam pemodelan proses bisnis usulan (*to-be*), dan analisis kebutuhan berupa identifikasi aktor, hingga pembuatan *use case diagram* dan *use case scenario*.

BAB 5**PERANCANGAN SISTEM**

Perancangan sistem berisi uraian mengenai perancangan sistem yang terdiri dari *sequence diagram*, *class diagram*, *data model*, serta rancangan desain antarmuka pengguna.

BAB 6**IMPLEMENTASI SISTEM**

Implementasi sistem berisi uraian mengenai spesifikasi lingkungan dimana sistem dibangun dan proses implementasi sesuai perancangan yang telah dilakukan.

BAB 7**PENGUJIAN DAN DEPLOYMENT**

Pengujian dan *deployment* membahas proses pengujian sistem yang telah diimplementasikan, *deployment* instalasi sistem pada perangkat computer yang ada di Edotel Pamekasan, dan pengujian penerimaan pengguna.

BAB 8**PENUTUP**

Penutup membahas kesimpulan yang didapat dari hasil penelitian yang sudah dilakukan beserta saran yang ditujukan kepada penelitian selanjutnya yang memiliki keterkaitan dengan penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada penelitian ini, kajian pustaka yang digunakan sebagai panduan dan sumber referensi didapatkan dari jurnal maupun penelitian – penelitian terdahulu yang telah dilakukan. Penelitian pertama adalah penelitian yang dilakukan oleh Walerych dan Wojciech Zabierowski pada tahun 2012. Penelitian ini mendeskripsikan aplikasi pendukung manajemen hotel. Disebutkan bahwa aplikasi yang dibuat, mampu membantu kegiatan operasional hotel yang mana dapat membantu secara signifikan pengelolaan hotel dan bisnis sejenis. Penelitian ini tidak menyebutkan metode yang digunakan dalam pengembangan perangkat lunaknya, namun dalam pengembangan sistemnya digunakan platform Java untuk membuat aplikasi *web*. Penelitian ini kemudian membahas keuntungan yang didapatkan jika hotel memiliki dan menggunakan sistem, contohnya sistem dapat mengontrol daftar kamar yang dapat disewakan dan menghitung tagihan tamu permalamnya secara otomatis. Kemudian penelitian ini membahas fitur – fitur yang dimiliki oleh sistem, contohnya fitur pengelolaan tambah atau hapus kamar. Hasil dari penelitian ini adalah penjelasan bahwa penggunaan sistem dalam pengelolaan hotel memberikan banyak keuntungan, yaitu sebuah sistem dapat menghemat pengeluaran manajemen, pelatihan staff, dan keseluruhan pelayanan. Lalu, pengoprasiaannya mudah dan jelas, sehingga dapat menghemat waktu. Sistem juga dapat mengontrol kamar dan menghitung tagihan tamu permalamnya secara otomatis. Jika menggunakan sistem berbasis web, maka tidak perlu *install software*, implementasi cepat, dan hanya perlu internet dengan *browser* ketika ingin menggunakan sistem. Oleh karena itu, perancangan sistem yang tepat dapat mempermudah manajemen hotel bahkan motel, *guest house* dan sejenisnya. Penelitian yang dilakukan (Walerych dan Zabierowski, 2012) akan digunakan sebagai referensi yang menguatkan solusi perbaikan proses bisnis dengan melakukan pengembangan sistem informasi untuk mengatasi masalah operasional hotel dan referensi mengembangkan sistem informasi berbasis *website*.

Bersumber dari jurnal internasional, penelitian kedua merupakan penelitian yang dilakukan pada tahun 2011 oleh Irena Petrijevcin Vuksanovic dan Bojan Sudarevic. Penelitian yang dilakukan membahas mengenai perbandingan dari proses pengembangan 2 buah versi perangkat lunak dari aplikasi yang sama. Aplikasi pertama menggunakan *pure PHP* tanpa *framework*, sedangkan aplikasi kedua menggunakan *framework* CodeIgniter dalam pengembangannya. Penelitian dilakukan dimulai dari melakukan perancangan aplikasi yang ingin dibandingkan, yaitu *Aleph Integrated Library System*. Kemudian, setelah mendefinisikan fitur dan teknologi yang akan digunakan, 2 versi aplikasi dikembangkan secara bergantian. Versi pertama dikembangkan terlebih dahulu tanpa menggunakan *framework*, kemudian dilanjutkan dengan mengembangkan versi kedua dengan menggunakan *framework* CodeIgniter. Setelah 2 versi aplikasi selesai dikembangkan, dilakukan perbandingan terhadap 2 versi *Aleph Integrated Library System*. Hasil dari

penelitian ini adalah pengembangan aplikasi versi 1 yang menggunakan PHP murni memakan waktu 19 jam *coding* dengan waktu pengerjaan selama 3 hari, sedangkan pengembangan aplikasi versi 2 dengan menggunakan *framework* CodeIgniter hanya menghabiskan waktu 11 jam dalam waktu 1 hari. Penggunaan *framework* juga mengurangi waktu yang dihabiskan dalam pemeliharaan kode program serta pengembangan lanjutan. Pengembangan menggunakan *framework* pada *small application* disarankan oleh penelitian ini karena menguntungkan terhadap kesiapan pengembang dalam kecepatan implementasi pembaruan pengembangan sistem yang akan datang yang sulit diprediksi dan bisa datang sewaktu – waktu permintaan pembaruannya. Penelitian yang dilakukan (Vuksanovic dan Sudarevic, 2011) akan digunakan sebagai referensi yang menguatkan penggunaan *framework* pada pengembangan sistem informasi perhotelan.

Lan Jin dan Xiujuan Liang merupakan nama peneliti yang penelitiannya yang digunakan sebagai kajian pustaka. Penelitian yang dilakukan pada tahun 2016 membahas mengenai menggabungkan dua metode, yaitu *Rational Unified Process* (RUP) dan *Unified Modeling Language* (UML) sebagai anjuran langkah pemodelan pengembangan sistem perangkat lunak skala kecil dan sedang, dan menerapkannya dalam analisis, desain, dan implementasi *instant messaging system*. Pengembangan perangkat lunak *instant messaging system* menggunakan metode *Rational Unified Process* (RUP) dan pemodelan *Unified Modeling Language* (UML). Hasil dan kesimpulan dari penelitian ini adalah RUP menyediakan templat, dan pedoman, dan kriteria yang dibutuhkan dalam banyak aspek metodologi pengembangan. Lalu dengan menggunakan UML dapat menghasilkan *model* sistem yang bervariasi dan menjadikannya sebuah persiapan yang baik saat akan melakukan *coding*. Dengan menggabungkan RUP dan UML, risiko produk hasil pengembangan dapat dikurangi, memberikan standarisasi manajemen beserta proses pengembangannya, dan meningkatkan kualitas, efisiensi, serta perawatan pengembangan perangkat lunak. Penelitian yang dilakukan (Jin dan Liang, 2016) akan digunakan sebagai referensi pengembangan sistem informasi yang menggunakan kombinasi metode RUP dan pemodelan UML.

Selanjutnya penelitian keempat yang adalah penelitian yang dilakukan oleh Andhika Akbar Saputra, Yusi Tyrone Mursityo, dan Nanang Yudi Setiawan pada tahun 2019. Penelitian membahas pengembangan sistem informasi untuk mengoptimalkan proses bisnis dan mengatasi permasalahan yang ada pada proses operasional CV. Dwi Artha Indah Samarinda, contohnya seperti waktu proses pengecekan ketersediaan peralatan sering melebihi batas waktu prosesur. Metode penelitian pengembangan sistem informasi ini dilakukan dengan menggunakan metode *Rational Unified Process* (RUP). Tahapan pengembangannya dimulai dari analisis proses bisnis, analisis kebutuhan, perancangan, pengujian *black-box*, pengujian kompatibilitas, hingga tahap terakhir, yaitu *User Acceptance Testing* (UAT). Hasil penelitian ini adalah metode RUP dapat dengan baik melakukan identifikasi masalah, memodelkannya ke dalam proses bisnis, dan merancang rekomendasi proses bisnis. Analisis kebutuhan yang memiliki perubahan dapat diatasi oleh kemampuan RUP yang *iterative*. Kegiatan

perancangan, implementasi, dan pengujian yang dilakukan dengan menggunakan RUP juga berhasil dilakukan dengan baik yang dibuktikan oleh penerimaan 3 aktor yang sangat setuju dengan adanya sistem reservasi. Penelitian yang dilakukan (Saputra, Mursityo dan Setiawan, 2019) akan digunakan sebagai referensi melakukan pengembangan sistem informasi menggunakan metode RUP terutama pada tahapan penelitiannya.

Penelitian yang dilakukan oleh Hendra Agusvianto menjadi penelitian kelima yang digunakan sebagai kajian pustaka. Penelitian yang dilakukan pada tahun 2017 membahas pengembangan aplikasi sistem *inventory* gudang berbasis *web* yang dapat digunakan untuk melakukan pengelolaan inventaris produk yang ada di gudang meliputi kegiatan mencatat, mengolah, dan melaporkan data persediaan barang yang ada di gudang. Pengembangan aplikasi ini dilakukan dengan tujuan untuk memudahkan kantor pusat dalam mendapatkan laporan dari gudang secara akurat, dan tepat sasaran, efisien. Penelitian ini menggunakan metode terstruktur karena pengembangan sistemnya dilakukan dengan alur membuat *flowchart*, *context diagram*, dan *data flow diagram* (DFD) pada tahap analisis dan desain sistemnya. Penelitian ini menghasilkan sebuah aplikasi sistem inventori pada gudang memiliki fungsi untuk dapat melakukan pencatatan keluar masuk barang membantu staf PT. Alaisys dalam mengefisienkan waktu kegiatan mencatat dan menyusun data barang. Lalu juga dapat memberikan kantor pusat laporan – laporan dari gudang yang akurat, tepat sasaran, dan efisien. Penelitian yang dilakukan (Agusvianto, 2017) akan digunakan sebagai referensi sistem informasi perhotelan bagian pengelolaan inventori karena tidak adanya proses bisnis pengelolaan inventori di Edotel Pamekasan.

Tabel 2.1 Kajian Pustaka

No.	Nama Penelitian	Tujuan Penelitian	Metode Penelitian	Hasil dan Kesimpulan
1	<p>Nama Peneliti: Mieszko Walerych dan Wojciech Zabierowski</p> <p>Judul Jurnal: <i>Application Supporting Hotel Management as an Example of Web Technologies Usage</i></p> <p>Nama Jurnal: <i>Proceedings of International Conference on Modern Problem of Radio Engineering, Telecommunications and Computer Science</i></p> <p>Tahun Terbit: 2012</p>	<p>Mendeskripsikan aplikasi pendukung manajemen hotel, yang mana dapat membantu secara signifikan pengelolaan hotel dan bisnis sejenis.</p>	<p>Tidak dijelaskan metode yang digunakan dalam pengembangan perangkat lunaknya, namun dalam pengembangan sistemnya digunakan platform Java untuk membuat aplikasi <i>web</i>. Kemudian membahas keuntungan sistemnya, seperti dapat mengontrol kamar dan menghitung tagihan tamu permalamnya. Lalu dilanjutkan menjelaskan fitur – fitur yang dimiliki oleh sistem.</p>	<p>Penggunaan sistem dalam pengelolaan hotel memberikan banyak keuntungan, yaitu sebuah sistem dapat menghemat pengeluaran manajemen, pelatihan staff, dan keseluruhan pelayanan. Lalu, operasi mudah dan jelas, menghemat waktu. Dapat mengontrol kamar dan menghitung tagihan tamu permalamnya. Tidak perlu <i>install software</i>, implementasi cepat, dan hanya perlu internet dengan <i>browser</i> ketika ingin menggunakan sistem. Oleh karena itu, perancangan sistem yang tepat dapat mempermudah manajemen hotel bahkan motel, <i>guest house</i> dan sejenisnya.</p>

Tabel 2.1 Kajian Pustaka (lanjutan)

No.	Nama Penelitian	Tujuan Penelitian	Metode Penelitian	Hasil dan Kesimpulan
2	<p>Nama Peneliti: Irena Petrijevcenin Vuksanovic dan Bojan Sudarevic</p> <p>Judul Jurnal: <i>Use of Web Application Frameworks in the Development of Small Applications</i></p> <p>Nama Jurnal: <i>MIPRO, 2011 Proceedings of the 34th International Convention</i></p> <p>Tahun Terbit: 2011</p>	<p>Membandingkan proses pengembangan perangkat lunak dari aplikasi yang sama. Aplikasi pertama menggunakan <i>pure PHP</i> tanpa <i>framework</i>, sedangkan aplikasi kedua menggunakan <i>framework</i> CodeIgniter dalam pengembangannya.</p>	<p>Melakukan perancangan aplikasi yang ingin dibandingkan, yaitu <i>Aleph Integrated Library System</i>. Setelah mendefinisikan fitur dan teknologi yang akan digunakan, 2 versi aplikasi dikembangkan secara bergantian. Kemudian dilakukan perbandingan terhadap aplikasi 2 versi <i>Aleph Integrated Library System</i>. Aplikasi versi pertama dibuat menggunakan <i>PHP murni</i> tanpa <i>framework</i>, lalu aplikasi versi kedua dibuat menggunakan <i>framework</i> CodeIgniter.</p>	<p>Pengembangan aplikasi versi 1 yang menggunakan <i>PHP murni</i> memakan waktu 19 jam coding selama 3 hari, sedangkan pengembangan aplikasi versi 2 dengan menggunakan CodeIgniter menghabiskan waktu 11 jam hanya dalam waktu 1 hari. Penggunaan <i>framework</i> juga mengurangi waktu yang dihabiskan dalam pemeliharaan kode program dan pengembangannya lain waktu. Pengembangan menggunakan <i>framework</i> pada <i>small application</i> disarankan karena menguntungkan kesiapan kecepatan implementasi pada pengembangan yang akan datang yang sulit diprediksi permintaan pembaruannya.</p>
3	<p>Nama Peneliti: Lan Jin dan Xiujuan Liang</p>	<p>Menggabungkan dua metode, yaitu <i>Rational Unified Process (RUP)</i> dan <i>Unified Modeling Language (UML)</i> sebagai anjuran</p>	<p>Mengembangkan perangkat lunak menggunakan metode <i>Rational Unified Process (RUP)</i></p>	<p>RUP menyediakan templat, dan pedoman, dan kriteria yang dibutuhkan dalam banyak aspek metodologi pengembangan. Lalu</p>

Tabel 2.1 Kajian Pustaka (lanjutan)

No.	Nama Penelitian	Tujuan Penelitian	Metode Penelitian	Hasil dan Kesimpulan
	<p>Judul Jurnal: <i>Modeling of Instant Messaging System Based on RUP and UML</i></p> <p>Nama Jurnal: <i>2016 International Conference on Computational Intelligence and Applications</i></p> <p>Tahun Terbit: 2016</p>	<p>langkah pemodelan pengembangan sistem perangkat lunak skala kecil dan sedang, dan menerapkannya dalam analisis, desain, dan implementasi <i>instant messaging system</i>.</p>	<p>dan pemodelan <i>Unified Modeling Language</i> (UML).</p>	<p>menggunakan UML dapat menghasilkan model sistem yang bervariasi dan membuatnya menjadi persiapan yang baik saat akan melakukan <i>coding</i>. Menggabungkan RUP dan UML dapat mengurangi risiko hasil pengembangan, memberikan standarisasi manajemen beserta proses pengembangannya, dan meningkatkan efisiensi, kualitas, serta perawatan perangkat lunak yang dikembangkan.</p>
4	<p>Nama Peneliti: Andhika Akbar Saputra, Yusi Tyroni Mursityo, dan Nanang Yudi Setiawan</p> <p>Judul Jurnal: Pengembangan Sistem Informasi Reservasi Pada CV. Dwi Artha Indah Samarinda Menggunakan</p>	<p>Mengembangkan sistem informasi untuk mengoptimalkan proses bisnis dan permasalahan yang ada pada kegiatan operasional CV. Dwi Artha Indah Samarinda seperti waktu proses pengecekan ketersediaan peralatan sering melebihi batas waktu prosesur.</p>	<p>Pengembangan sistem informasi dilakukan dengan menerapkan metode pengembangan <i>Rational Unified Process</i> (RUP). Tahap pengembangan dimulai dari analisis proses bisnis, analisis kebutuhan, perancangan, pengujian <i>black-box</i>, pengujian kompatibilitas, lalu</p>	<p>Metode RUP dapat dengan baik meakukan identifikasi masalah, memodelkannya ke dalam proses bisnis, dan merancang rekomendasi proses bisnis. Analisis kebutuhan yang memiliki perubahan dapat diatasi oleh kemampuan RUP yang <i>iterative</i>. Perancangan, implementasi, dan pengujian yang dilakukan dengan RUP juga berhasil dilakukan</p>

Tabel 2.1 Kajian Pustaka (lanjutan)

No.	Nama Penelitian	Tujuan Penelitian	Metode Penelitian	Hasil dan Kesimpulan
	<p>Metode <i>Rational Unified Process</i></p> <p>Nama Jurnal: Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer</p> <p>Tahun Terbit: 2019</p>		<p>tahap terakhir <i>User Acceptance Testing (UAT)</i>.</p>	<p>dengan baik yang dibuktikan dari penerimaan 3 aktor terhadap sistem reservasi mendapatkan skor sangat setuju.</p>
5	<p>Nama Peneliti: Hendra Agusvianto</p> <p>Judul Jurnal: Sistem Informasi Inventori Gudang Untuk Mengontrol Persediaan Barang Pada Gudang Studi Kasus: PT. Alaisys Sidoarjo</p> <p>Nama Jurnal: <i>Journal Information Engineering and Educational Technology</i></p> <p>Tahun Terbit: 2017</p>	<p>Mengembangkan aplikasi sistem <i>inventory</i> gudang berbasis <i>web</i> yang dapat digunakan untuk melakukan pengelolaan inventaris produk yang ada pada gudang meliputi kegiatan mencatat, mengolah, dan melaporkan data persediaan gudang. Pengembangan aplikasi dilakukan untuk memudahkan kantor pusat mengetahui laporan dari gudang secara akurat, tepat sasaran, dan efisien.</p>	<p>Menggunakan metode terstruktur karena pengembangan dilakukan dengan alur membuat <i>flowchart</i>, <i>context diagram</i>, dan <i>data flow diagram (DFD)</i> pada tahap analisis dan desain sistem.</p>	<p>Aplikasi sistem inventori pada gudang yang memiliki fungsi untuk dapat mengerjakan pencatatan keluar dan masuknya barang berhasil membantu staf PT. Alaisys dalam mengefisiensi waktu mencatat dan menyusun data barang. Aplikasi juga dapat memberikan laporan dari gudang ke kantor pusat secara akurat, tepat sasaran, dan efisien.</p>

2.2 Edotel Pamekasan

Pada tahun ajaran 2003/2004, Program Keahlian Akomodasi Perhotelan SMK Negeri 3 Pamekasan membuat sebuah laboratorium berupa mini hotel dengan nama “Mini Hotel Srikandi” yang hanya memiliki 2 buah kamar. Kemudian pada awal tahun 2004, SMK Negeri 3 Pamekasan melakukan penambahan kamar hotel setelah mendapat bantuan dari Direktorat Pendidikan Menengah dan Kejuruan (DIKMENJUR) dan dana *sharing* Pemerintah Kabupaten Pamekasan.

Seiring dengan bertambahnya jumlah kamar, hotel yang pada awalnya hanya dipergunakan untuk laboratorium siswa kini dibuka untuk umum, dan berganti nama menjadi “Shofa Hotel”. Lalu, pada tanggal 16 hingga 19 Desember 2005, menurut penjelasan Ibu Yuli Indrawati selaku manajer hotel, beliau menuturkan bahwa dalam *Workshop* dan Peresmian Hotel *Training* Se-Indonesia, diluncurkan sebuah nama untuk seluruh Hotel Training SMK Se-Indonesia dengan nama Edotel yang diikuti oleh nama kabupaten/kota SMK tersebut berada. Sehingga “Shofa Hotel” berganti nama menjadi Edotel Pamekasan. Seiring dengan berjalannya waktu, kini Edotel Pamekasan telah memiliki total 19 kamar dengan tipe kamar berbeda mulai dari *standart* hingga *executive*. Edotel Pamekasan ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Edotel Pamekasan

2.2.1 Visi

Mewujudkan Edotel Pamekasan sebagai *education* hotel yang professional

2.2.2 Misi

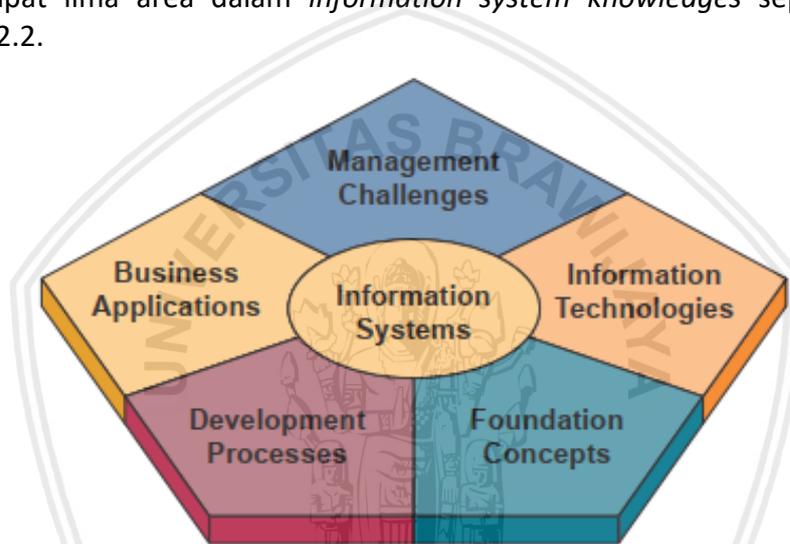
1. Memberikan layanan prima sesuai standar industry
2. Meningkatkan citra positif hotel sebagai sarana akomodasi

- Menjadikan Edotel Pamekasan sebagai pusat pelatihan kerja bidang perhotelan

2.3 Sistem Informasi

Sistem informasi merupakan gabungan dari manusia, perangkat lunak, perangkat keras, sumber data, jaringan komunikasi, dan prosedur serta kebijakan yang menerima, menyimpan, mengolah, dan menyebarkan informasi pada sebuah organisasi. Beragam instruksi maupun prosedur pemrosesan informasi yakni *software*, perangkat fisik yakni *hardware*, data tersimpan yakni *data resources*, kanal komunikasi yakni *networks* merupakan tumpuan dari penggunaan sistem informasi modern (O'Brien dan Marakas, 2010).

Terdapat lima area dalam *information system knowledges* seperti dalam Gambar 2.2.



Gambar 2.2 Information Systems Knowledge

Sumber: O'Brien dan Marakas (2010)

Menurut O'Brien dan Marakas (2010), area pertama dalam *information system knowledge* adalah *Foundation Concepts*, yaitu dasar dari konsep mengenai perilaku, teknis, bisnis, hingga manajerial mengenai peran dan komponen dalam sistem informasi. Setelah itu *Information Technologies*, yaitu membahas konsep pokok, pengembangan, serta masalah manajemen dalam teknologi informasi seperti perangkat lunak, perangkat keras, manajemen data, jaringan, dan berbagai teknologi *Internet-based*. Aplikasi Bisnis atau *Business Applications* memiliki bahasan utama kegiatan operasional, manajemen, dan keuntungan kompetitif bisnis sebagai kegunaan sistem informasi yang sangat penting. *Development Processes* membahas bagaimana *business professional* dan *information specialist* melakukan perancangan, melakukan pengembangan, serta melakukan penerapan sistem informasi dalam mendapatkan peluang bisnis. Area terakhir adalah tantangan manajemen atau *Management Challenges*, yaitu tantangan efektivitas dan etika mengelola teknologi informasi kepada pengguna akhir, *enterprise*, dan bisnis dengan skala global.

2.4 Rational Unified Process

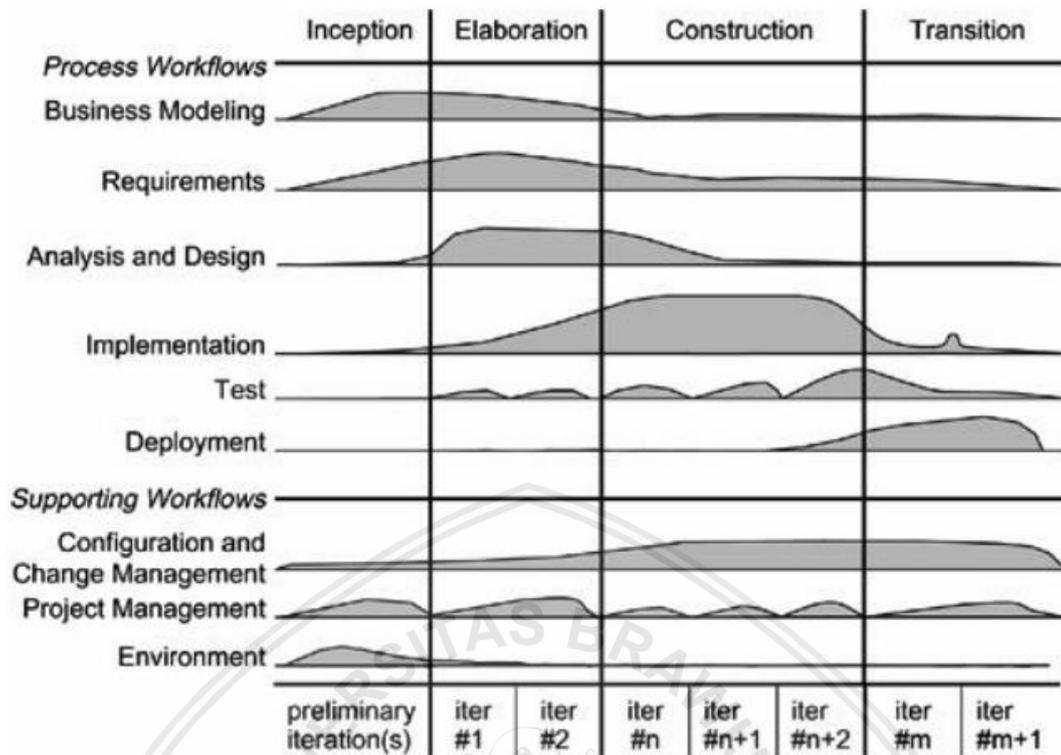
Rational Unified Process (RUP) merupakan salah satu pendekatan *software development life cycle* yang cocok digunakan dengan *Unified Modeling Language*. Tujuan utama dari RUP adalah membuat sebuah produk perangkat lunak yang dengan kualitas yang tinggi sehingga dapat menjawab kebutuhan pengguna akhir sesuai dengan jadwal dan sesuai dengan anggaran. Pendekatannya adalah salah satu metode praktik pengembangan perangkat lunak terbaik saat ini karena fleksibel di berbagai proyek dan organisasi. Terdapat tujuh karakteristik RUP menurut Booch, Rumbaugh, dan Jacobson (1998).

Pertama, RUP merupakan proses *iterative*, yang meningkatkan pemahaman masalah melalui perbaikan yang dilakukan berturut – turut dan mengembangkan solusi efektif yang berlanjut di dalam beberapa siklus. Kedua, aktivitas RUP menekankan pada pembuatan dan pemeliharaan model seperti UML daripada dokumen tertulis, dengan tujuan meminimalisir pengeluaran tambahan seperti perawatan dokumen dan memaksimalkan konten informasi yang relevan. Ketiga, pengembangan RUP adalah *architecture-centric* yang mana berfokus pada perancangan perangkat lunak.

Keempat, aktivitas pengembangan RUP bersifat *use case – driven* yang menekankan pada pemahaman menyeluruh bagaimana sistem digunakan. Kelima, RUP mendukung teknik *object – oriented* karena model yang digunakan menggunakan UML dan menggunakan konsep objek, kelas, dan relasinya. Keenam, RUP adalah proses *configurable* yang mana RUP fleksibel terhadap kebutuhan proyek baik untuk tim kecil pengembangan perangkat lunak maupun organisasi besar. Terakhir, RUP mendorong *quality control* dan manajemen risiko yang objektif dan berkelanjutan (Booch, Rumbaugh dan Jacobson, 1998).

2.4.1 Fase Rational Unified Process

Metode RUP yang memiliki kepanjangan *Rational Unified Process* memiliki 4 fase, yakni fase *inception*, fase *elaboration*, fase *construction*, dan terakhir fase *transition*. *Inception* dan *elaboration* merupakan aktivitas perancangan dalam SDLC, sedangkan *construction* dan *transition* merupakan tahap produksi (Booch, Rumbaugh dan Jacobson, 1998). Fokus pengerjaan pada setiap fasenya ditunjukkan dalam Gambar 2.3.



Gambar 2.3 Rational Unified Process

Sumber: Booch, Rumbaugh, dan Jacobson (1998)

2.4.1.1 Fase Inception

Dalam fase *inception*, dilakukan pembuatan *business case* untuk sistem dan batas *scope*. *Business case* terdiri dari kriteria kesuksesan, analisis risiko, estimasi kebutuhan sumber daya, dan perencanaan penjadwalan. Pada akhir fase, tujuan *project life cycle* diperiksa dan memutuskan apakah akan melanjutkan pengembangan dengan skala penuh (Booch, Rumbaugh dan Jacobson, 1998).

2.4.1.2 Fase Elaboration

Fase *elaboration* merupakan sebuah fase yang dilakukan setelah fase *inception* dengan tujuan untuk menganalisa domain masalah, membuat dasar arsitektur, mengembangkan *project plan*, dan mengeliminasi elemen risiko tertinggi dalam proyek. Arsitektur harus dibuat dengan pemahaman menyeluruh terhadap sistem, yang artinya kebutuhan sistem harus dijelaskan. Untuk memverifikasi arsitektur yang dibuat, sistem diimplementasi untuk mendemonstrasikan arsitektur dan mengeksekusi *use cases*. Fase *elaboration* berakhir dengan pengembang melakukan pengecekan kembali secara detail arah dan cakupan sistem, bagaimana arsitekturnya dibangun, dan menentukan apakah akan lanjut ke dalam fase *construction* (Booch, Rumbaugh dan Jacobson, 1998).

2.4.1.3 Fase Construction

Dalam fase *construction*, dikembangkan sebuah produk yang nantinya siap digunakan oleh pengguna secara berulang (*iterative*) dan bertahap (*incremental*).

Fase ini menjelaskan sisa kebutuhan, kriteria penerimaan, melakukan penyempurnaan desain, menyelesaikan implementasi, dan menguji perangkat lunak. Di akhir fase *construction* dilakukan pemeriksaan pada perangkat lunak yang dibangun, *server*, dan pengguna yang bertujuan untuk mengetahui kesiapannya dalam mengoperasikan perangkat lunak (Booch, Rumbaugh dan Jacobson, 1998).

2.4.1.4 Fase *Transition*

Fase terakhir, yaitu fase *transition* dilakukan *deployment* atau memperkenalkan perangkat lunak kepada pengguna. Biasanya setelah sistem diberikan ke pengguna, muncul isu atau masalah yang membutuhkan tambahan pengembangan lanjut guna menyesuaikan sistem, membenahi masalah yang belum atau tidak terdeteksi, dan atau menyelesaikan beberapa fitur yang sempat tertunda. Fase *transition* biasanya dimulai dengan melakukan perilis sistem ke dalam beta *version*, yang mana kemudian diganti oleh sistem yang sudah selesai dan siap untuk di produksi. Akhir fase *transition*, diteliti apakah tujuan pengembangan proyek telah terpenuhi dan tentukan apakah sudah saatnya memulai pengembangan yang lain. Kemudian pelajaran yang didapat dari pengembangan sistem informasi dirangkum agar dapat melakukan improvisasi pada proses pengembangan yang akan dilakukan nantinya (Booch, Rumbaugh dan Jacobson, 1998).

2.5 Proses Bisnis

Menurut Monk dan Wagner (2013) Sekumpulan kegiatan yang membutuhkan minimal satu atau masukan yang lebih dan dapat menciptakan keluaran, seperti misalnya pelaporan atau prakiraan, sehingga memberikan *value* kepada pelanggan merupakan definisi dari proses bisnis. Proses bisnis dapat digambarkan menggunakan notasi grafik dengan melakukan pemodelan proses bisnis. Pemodelan proses bisnis ialah suatu aktivitas atau tahapan atau kegiatan pemodelan yang dilakukan dengan tujuan membantu meningkatkan pemahaman proses yang sedang berlangsung di dalam sebuah organisasi maupun di dalam sebuah perusahaan (Saputra dan Christianti, 2013). Terdapat banyak model yang dapat digunakan untuk menggambarkan proses bisnis, *Business Process Modelling Notation* (BPMN) ialah salah satunya.

2.5.1 *Business Process Modelling Notation* (BPMN)

Business Process Modelling Notation (BPMN) yang dikembangkan oleh Object Management Group merupakan aktivitas pemodelan dengan menggunakan notasi grafis yang dilakukan untuk mempermudah pengguna dalam memahami proses yang terjadi dan mengetahui aktor yang terlibat dalam alur bisnis. Tujuan utamanya adalah menyajikan notasi yang dapat dengan mudah dimengerti oleh seluruh *business users*, mulai dari seorang bisnis analis yang bertanggung jawab membuat konsep proses bisnis, hingga pengembang yang bertanggung jawab mengimplementasi teknologi yang menjalankan proses bisnis, dan juga pada *business people* yang bertanggung jawab mengawasi dan mengelola proses –

prosesnya (Weske, 2007). BPMN mempunyai lima kategori elemen dasar yang digunakan untuk membuat diagram (Object Management Group, 2011). Lima elemen tersebut adalah:

1. *Flow Objects*

Flow objects ialah elemen yang fundamental yang digunakan untuk mendefinisikan perilaku sebuah proses bisnis. Terdiri dari *events*, *activities*, dan *gateways* (Object Management Group, 2011).

2. *Data*

Data direpresentasikan dengan empat tipe, yaitu *data objects*, *data inputs*, *data outputs*, dan *data stores* (Object Management Group, 2011).

3. *Connecting Objects*

Merupakan penghubung *flow objects* dengan satu sama lainnya atau dengan informasi lain. Terdapat empat macam, yaitu *sequence flows*, *message flows*, *associations*, dan *data associations* (Object Management Group, 2011).

4. *Swimlanes*

Terdapat dua cara mengelompokkan model elemen utama melalui “Swimlanes”, yaitu *Pools* dan *Lanes*.

5. *Artifacts*

Artifacts biasanya digunakan untuk memberikan informasi tambahan mengenai proses seperti *group*, dan *text annotation*.

2.5.1.1 Flow Objects

Flow object terdiri dari *event*, *gateway*, dan *activity*.

1. *Event* merupakan sesuatu kejadian yang memberikan dampak pada proses bisnis. Sebuah *event* dapat berasal dari internal atau eksternal proses. Terdapat tiga tipe *event*, antara lain *start event*, *intermediate event*, dan *end event*. Semua proses selalu diawali dengan sebuah *start event* sebagai awal dari proses bisnis. Penjelasan tipe – tipe *activity* ditunjukkan pada Tabel 2.2.

Tabel 2.2 Notasi Event

No.	Tipe	Deskripsi	Notasi
1	<i>Start</i>	Menunjukkan dimulainya suatu proses	
2	<i>Intermediate</i>	Biasanya terjadi antara <i>Start Event</i> dengan <i>End Event</i> . Alur proses akan terpengaruh jika menggunakan tipe <i>intermediate</i> , namun tipe	

Tabel 2.2 Notasi Event (lanjutan)

No.	Tipe	Deskripsi	Notasi
		ini tidak akan memulai atau secara langsung memberhentikan proses	
3	<i>End</i>	Menunjukkan akhir dari suatu proses	

Sumber: Object Management Group (OMG) (2011)

2. *Gateway* memiliki tanggung jawab dalam melakukan kontrol alur sebuah proses bisnis. Penjelasan tipe – tipe *activity* terdapat pada Tabel 2.3.

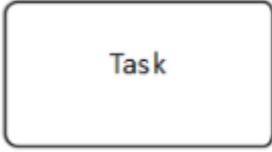
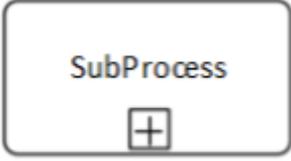
Tabel 2.3 Notasi Gateway

No.	Tipe	Deskripsi	Notasi
1	<i>Exclusive</i>	Sebagai <i>divergence</i> : digunakan untuk membuat jalur alternatif dalam sebuah proses, tapi hanya satu yang dipilih Sebagai <i>convergence</i> : digunakan untuk menggabungkan jalur alternative	
2	<i>Parallel</i>	Mendeskripsikan proses yang berjalan bersama	
3	<i>Inclusive</i>	Mendeskripsikan sebuah proses yang dipecah ke dalam beberapa jalur	
4	<i>Complex</i>	Mendeskripsikan alur kompleks dalam sebuah proses bisnis	

Sumber: Object Management Group (OMG) (2011)

3. *Activity* merupakan sebuah tugas yang dilakukan dalam sebuah proses bisnis. *Activity* ditujukan dengan kotak dengan sudut bulat berisi nama yang menunjukkan aktivitas yang dilakukan. Memiliki dua macam tipe *activity*, yakni *task* dan *sub process*. Deskripsi tipe – tipe *activity* ditunjukkan pada Tabel 2.4.

Tabel 2.4 Notasi Activity

No.	Type	Deskripsi	Notasi
1	<i>Task</i>	Merupakan aktivitas yang terjadi pada alur proses	
2	<i>Sub Process</i>	Merupakan sebuah aktivitas yang beragam yang dimasukkan ke dalam proses. Aktivitas yang beragam tersebut dapat dimodelkan lebih detail	

Sumber: Object Management Group (OMG) (2011)

2.5.1.2 Data

Data direpresentasikan ke dalam empat elemen, yaitu *data objects*, *data inputs*, *data outputs*, dan *data stores*. Penjelasan tipe – tipe *data* ditunjukkan pada Tabel 2.5.

Tabel 2.5 Notasi Data

No.	Type	Deskripsi	Notasi
1	<i>Data Object</i>	Mempresentasikan informasi yang digunakan dalam sebuah proses seperti surat, dokumen keperluan bisnis, surat, email dan lain – lain	
2	<i>Data Input</i>	<i>Data Input</i> merupakan sebuah deklarasi data tipe tertentu akan digunakan sebagai masukan	
3	<i>Data Output</i>	<i>Data Output</i> merupakan sebuah deklarasi data tipe tertentu akan digunakan sebagai keluaran	
4	<i>Data Store</i>	Menyediakan mekanisme aktivitas mendapatkan atau memperbarui informasi yang disimpan	

Sumber: Object Management Group (OMG) (2011)

2.5.1.3 Connecting Objects

Merupakan elemen yang digunakan untuk menjembatani *flow objects*. *Connecting Objects* terdiri dari *sequence flow*, *association*, dan *message flow*. Tipe – tipe *connections* dijelaskan pada Tabel 2.6.

Tabel 2.6 Notasi Connection

No.	Tipe	Deskripsi	Notasi
1	<i>Sequence Flow</i>	Mengunjukkan urutan kegiatan sebuah proses	
2	<i>Association</i>	Mengunjukkan relasi dalam sebuah proses antara data, teks, artefak yang lain, dengan <i>flow object</i>	
3	<i>Message Flow</i>	Mengunjukkan alur pesan antara dua partisipan yang mampu menerima dan mengirim pesan	

Sumber: Object Management Group (OMG) (2011)

2.5.1.4 Swimlanes

Swimlanes ialah wadah grafis yang membagi sebuah set aksi dengan aksi yang lain. *Swimlanes* memiliki 2 tipe, yaitu *pool* dan *lane*. Penjelasan tipe – tipe *swimlanes* ditunjukkan pada Tabel 2.7.

Tabel 2.7 Notasi Swimlanes

No.	Tipe	Deskripsi	Notasi
1	<i>Pool</i>	Melambangkan wadah yang berisi satu proses dan memiliki <i>sequence flow</i> yang dapat menghubungkan aktivitas	
2	<i>Lane</i>	Memiliki fungsi untuk mempresentasikan tanggung jawab kegiatan dalam sebuah proses	

Sumber: Object Management Group (OMG) (2011)

2.5.1.5 Artifacts

Artifacts mempresentasikan objek yang ada diluar proses. *Artifact* mempresentasikan data maupun catatan yang menjelaskan sebuah proses atau yang dapat digunakan untuk mengelola tugas ataupun proses. Penjelasan tipe – tipe *artifacts* ditunjukkan pada Tabel 2.8.

Tabel 2.8 Notasi Artifacts

No.	Type	Deskripsi	Notasi
1	<i>Annotation</i>	Menunjukkan tambahan informasi kepada pembaca <i>diagram</i> BPMN	
2	<i>Group</i>	Secara informal dimungkinkan untuk dapat digunakan dalam mengelompokkan elemen	

Sumber: Object Management Group (OMG) (2011)

2.6 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahap pengembangan sebuah sistem yang dilakukan dengan mendefinisikan kebutuhan yang dibutuhkan pengguna sistem. Kebutuhan suatu sistem sendiri adalah penjelasan dari apa yang seharusnya sistem dapat kerjakan, layanan apa yang dimiliki, dan kendala dalam pengoperasiannya. Kebutuhan dibuat berdasarkan keinginan pengguna akan sistem. Kebutuhan sering diklasifikasi menjadi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional sendiri berisi pernyataan layanan yang harus disediakan sistem, bagaimana reaksi sistem terhadap masukan, dan bagaimana tingkah laku sistem dalam situasi tertentu. Kemudian kebutuhan non-fungsional berisi batasan layanan yang dimiliki sistem, contohnya batas waktu (Sommerville, 2011).

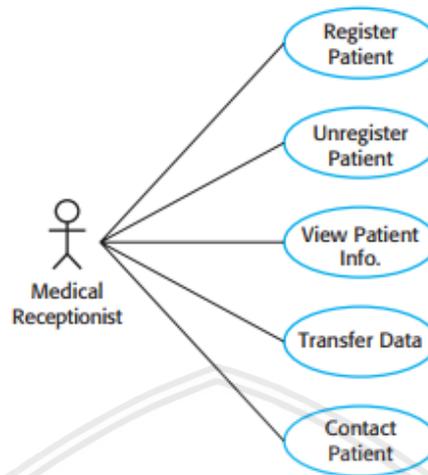
2.7 Unified Modeling Language

Menurut Pressman (2010), *Unified Modeling Language* (UML) adalah sebuah bahasa standar yang memiliki kegunaan untuk dapat melakukan pemodelan perangkat lunak. Kegunaan UML adalah untuk memberikan gambaran, untuk membangun, dan untuk menyajikan dokumentasi artefak – artefak yang dibutuhkan pada kegiatan pengembangan sistem. UML digunakan dengan maksud untuk menggambar pemodelan yang dapat memudahkan pemahaman ketika melakukan aktivitas atau kegiatan pembangunan suatu sistem. Di dalam UML, ada banyak macam pemodelan diagram yang dapat digunakan oleh pengembang, seperti *use case diagram*, *sequence diagram*, serta *class diagram*.

2.7.1 Use Case Diagram

Use case adalah gambaran mengenai interaksi *end user* dan sistem dalam keadaan tertentu (Pressman, 2010). *Use case* juga merupakan teknik penggalian kebutuhan yang fundamental dalam pemodelan UML. Secara sederhana, *use case* mengidentifikasi aktor yang terlibat dan nama interaksi dalam sistemnya. *Use cases* kemudian didokumentasikan dalam *use case diagram* (Sommerville, 2011). Contoh *use case diagram* milik Sommerville (2011) serta notasi yang umum

digunakan dalam *use case diagram* ditunjukkan dalam Gambar 2.4 dan pada Tabel 2.9.



Gambar 2.4 Use Case Diagram

Sumber: Sommerville (2011)

Tabel 2.9 Notasi Use Case Diagram

No.	Tipe	Deskripsi	Notasi
1	Aktor	Inisialisasi penggambaran sebuah peran yang dilakukan oleh manusia, proses, dan lain – lain yang berinteraksi dengan sistem, subsistem, atau <i>class</i> .	
2	<i>Use Case</i>	Satuan dari fungsi yang telah digolongkan dan berisi pesan yang dimiliki oleh satu atau lebih aktor	
3	<i>Association</i>	Merupakan sebuah garis lurus yang menandakan jalur hubungan aktor dengan <i>use case</i>	
4	<i>Extend</i>	Tambahan perilaku aktor pada dasar <i>use case</i>	«extend»
5	<i>Include</i>	Tambahan perilaku <i>use case</i> yang masih masuk bagian dasar <i>use case</i>	«include»
6	<i>Use Case Generalization</i>	Relasi antara sebuah <i>general use case</i> terhadap <i>use case</i> yang lebih spesifik yang mewarisinya	

Sumber: Rumbaugh, Jacobson dan Booch (2005)

Pada dasarnya, *use case* tidak hanya sebuah bulatan yang memiliki deskripsi singkat. *use case* memiliki *use case description* yang memiliki tujuan untuk menceritakan *use case* secara menyeluruh. Tidak jarang ditemukan *use case description* direferensikan sebagai *use case properties* berisikan detail dari *use case*. Kunci dari *use case description* adalah *flow of events*, *preconditions*, dan *postconditions* (Bittner dan Spence, 2002). Format *use case description* ditunjukkan pada Tabel 2.10.

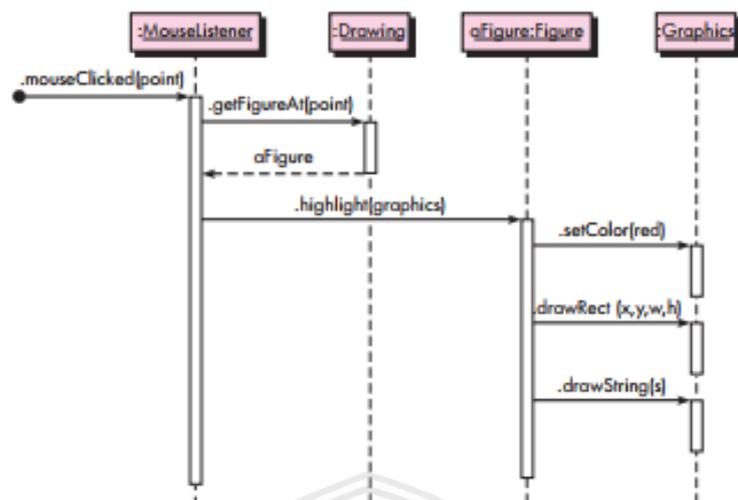
Tabel 2.10 Dokumen Use Case Description

Use Case Code	Berisi kode yang digunakan dalam <i>use case</i>
Use Case Name	Berisi nama <i>use case</i>
Actor	Berisi nama pengguna atau aktor yang memiliki interaksi dengan sistem
Description	Berisi penjelasan dari tujuan dan peran <i>use case</i>
Pre-conditions	Menjabarkan kondisi – kondisi yang harus terpenuhi sebelum menjalankan <i>use case</i>
Main Flow	Biasa dikenal sebagai <i>Basic Flow</i> atau <i>Basic Path</i> , merupakan alur normal <i>use case</i>
Alternative Flow	Merupakan alur opsional yang berjalan saat <i>use case</i> melenceng dari alur normal
Sub Flow	Merupakan alur yang digunakan untuk menyederhanakan dan membagi alur normal yang terlalu mendetail
Post-conditions	Berisi deksripsi kondisi yang harus terjadi saat <i>use case</i> selesai

Sumber: Bittner dan Spence (2002)

2.7.2 Sequence Diagram

Sequence diagram memiliki pengertian, yaitu sebuah kegiatan pemodelan komunikasi yang dinamis antar objek – objek dalam menjalankan perintah. *Sequence diagram* menggambarkan urutan sementara ketika objek satu sama lain saling berkirim pesan dalam mengerjakan sebuah perintah. *Diagram* ini biasanya digunakan oleh pengembang dengan tujuan memberikan petunjuk kejadian interaksi di dalam kasus (Pressman, 2010). Contoh *Sequence diagram* digambarkan dalam Gambar 2.5.

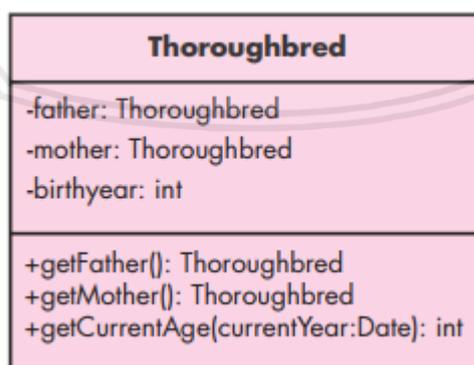


Gambar 2.5 Sequence Diagram

Sumber: Pressman (2010)

2.7.3 Class Diagram

UML menyediakan *class diagram* untuk memodelkan kelas – kelas, termasuk atribut, operasi, serta relasi dan asosiasi terhadap kelas yang lain. *Class diagram* merupakan sebuah diagram yang memiliki kegunaan untuk memberikan gambaran dari sudut pandang statis maupun struktural dalam sebuah sistem. Elemen pokok yang menunjukkan ciri – ciri *class diagram* adalah *boxes* atau kotak – kotak, sebagai ikon untuk merepresentasikan kelas – kelas dan antarmuka. Setiap kotak dibagi secara horizontal. Bagian teratas berisi nama kelasnya. Di tengah terdapat atribut dari kelasnya. Dan bagian ketiga dari *class diagram* berisi operasi atau perilaku kelas tersebut (Pressman, 2010). Contoh *Class diagram* yang dikemukakan oleh Pressman ditunjukkan dalam Gambar 2.6.



Gambar 2.6 Class Diagram

Sumber: Pressman (2010)

2.8 Data Model

Pemodelan data bertanggung jawab membuat *model* dari informasi milik organisasi yang menjangkau tingkatan abstraksi yang berbeda, mulai dari konseptual hingga fisik. *Model* konseptual merupakan model yang independen dan dapat didiskusikan dengan orang bisnis awam hingga ahli domain, sebagai konsep dasar yang dapat dibahas, diganti, dan disepakati. Sedangkan *model* fisik sudah menerapkan data yang spesifik dan pemilihan teknologinya pada *model*. *Model* fisik hanya memperbolehkan pengembang berdiskusi dan menyetujui penggunaan teknologinya sebagai tahap persiapan implementasi seperti mendiskusikan *database management system* (DBMS) (Sparx Systems, 2017).

2.8.1 Conceptual Data Model

Conceptual Data Model (CDM) bisa juga disebut domain *model* merupakan bentuk *data model* yang paling abstrak. CDM yang sangat berguna untuk membahas ide – ide kepada pemangku kepentingan karena kesederhanaannya. Oleh karenanya, informasi khusus seperti tipe data, indeks, dan key dihilangkan dari *Conceptual Data Model* (Sparx Systems, 2017). Dalam Gambar 2.7 ditunjukkan contoh *Conceptual Data Model*.

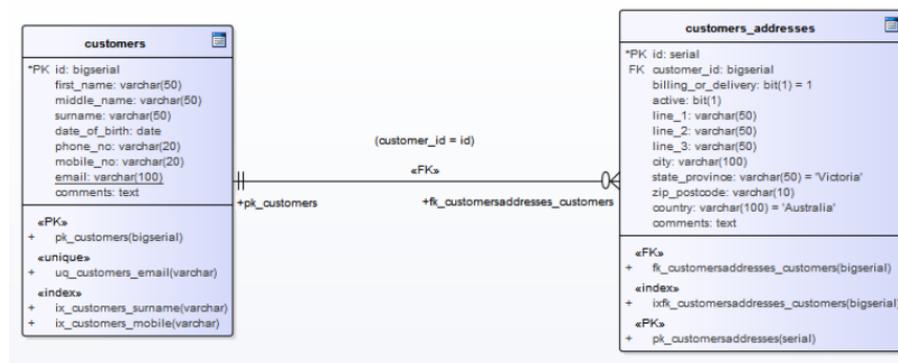


Gambar 2.7 *Conceptual Data Model*

Sumber: Sparx Systems (2017)

2.8.2 Physical Data Model

Physical Data Model (PDM) merepresentasikan visual struktur data menyerupai yang diimplementasikan pada skema *database* relasional. Sebagai tambahan tampilan abstraksi struktur basis data, keuntungan penting yang dari mendefinisikan *Physical Data Model* adalah bisa secara otomatis mendapatkan skema *database* dari model yang dibuat. Hal tersebut mungkin terjadi karena meta-data yang digunakan pada *Physical Data Model* mendekati skema *database*, meliputi *table*, *column*, *primary key*, serta *foreign key*. Dalam Gambar 2.8 ditunjukkan contoh *Physical Data Model*.



Gambar 2.8 Physical Data Model

Sumber: Sparx Systems (2017)

2.9 Pengujian Sistem

Pengertian atau definisi dari pengujian sistem ialah kegiatan yang dapat melibatkan pengujian dan *debugging* perangkat lunak, pengujian performa *website*, atau menguji *hardware* baru. Hal terpenting dalam melakukan pengujian adalah *review* tampilan *prototype*, hasilnya, dan *output* lainnya. *Prototype* seharusnya diulas oleh *end users* sistem untuk mencari kemungkinan kesalahan (O'Brien dan Marakas, 2010). Sedangkan menurut Sommerville (2011), pengujian merupakan sebuah proses yang ditujukan untuk menunjukkan sistem dapat melakukan hal yang seharusnya dan untuk dapat mengetahui serta membenahi cacat program sebelum program dirilis. Terdapat 2 tujuan dari proses pengujian, antara lain:

1. Sebagai pembuktian pada pengguna dan juga pada pengembang bahwasanya *software* sudah sesuai kebutuhan. Untuk perangkat lunak *custom*, untuk menguji kebutuhan minimal harus ada 1 tes yang dilakukan.
2. Untuk mendapatkan penemuan situasi dimana karakter perangkat lunak melakukan kesalahan, atau tidak memenuhi spesifikasi (Sommerville, 2011).

2.9.1 Black-Box Testing

Pengertian atau definisi dari *Black-box testing*, yang memiliki nama lain *behavioral testing* ialah pengujian yang memiliki menguji kebutuhan fungsional perangkat lunak sebagai fokusnya. Di dalam aktivitas melakukan *black-box testing*, penguji hanya tau *input* yang bisa diberikan dan harapan *output*-nya (Agarwal, Tayal dan Gulpa, 2010). *Black-box testing* dilakukan dengan tujuan menemukan kesalahan pada fungsi yang hilang atau salah, *user interface*, struktur data, dan perilaku sistem (Pressman, 2010).

2.9.1.1 Validation Testing

Validation testing merupakan sebuah pengujian yang mengharapkan sistem dapat beroperasi dengan baik pada kasus uji sesuai dengan ekspektasi. Target validasi adalah untuk meyakinkan bahwa perangkat lunak sesuai dengan ekspektasi pengguna. Validasi menjadi penting karena analisis kebutuhan tidak selalu dapat menggambarkan keinginan maupun kebutuhan pengguna dengan baik. Oleh karenanya, tujuan utama validasi adalah untuk membuktikan perangkat lunak yang dikembangkan telah sesuai dengan tujuan (Sommerville, 2011).

2.9.2 Compatibility Testing

Compatibility testing dilakukan pada aplikasi yang dikembangkan dengan cara menguji aplikasi pada peramban yang berbeda. Karena penggunaan peramban yang berbeda dapat memberikan pengaruh yang signifikan pada operasi *WebApp*. Contohnya, terkadang perbedaan peramban dapat menampilkan tampilan yang berbeda, terkecualikan dari tingkat standarisasi HTML yang ada menurut *WebApp*. Seperti misalnya mungkin tidak tersedianya *plug-in*. Dari beberapa kasus yang ada, masalah yang kecil dari kompatibilitas tidak memberikan dampak yang signifikan, namun tidak terlepas dari kemungkinan memunculkan masalah yang serius pada yang lain. Contoh lainnya adalah perbedaan peramban dapat menyebabkan terjadinya perubahan dramatis pada *layout* halaman, *font stye* yang tidak sesuai, atau formulir menjadi berantakan. Oleh karena itu, *compatibility testing* dilakukan untuk mendeteksi masalah lalu mengeliminasinya sebelum aplikasi dirilis (Pressman, 2010).

2.10 Deployment

Deployment adalah proses membuat sistem bisa digunakan oleh pengguna, mentransfer data dari sistem yang ada, dan menyetor dengan sistem lain yang ada di tempat itu (Sommerville, 2011). *Deployment* membahas proses konfigurasi sistem yang akan dikenalkan ke lingkungannya. Dalam RUP, *Deployment* merupakan salah satu dari 9 proses alur kerja. Secara lebih rinci, alur kerja *deployment* berisikan informasi bagaimana perangkat lunak dikemas, dikirim, diinstal, dan dijalankan pada target lingkungannya (Booch, Rumbaugh dan Jacobson, 1998).

2.11 User Acceptance Testing (UAT)

Pengertian atau definisi dari *User Acceptance Testing* adalah aktivitas menguji sistem yang dikerjakan bersama pengguna dengan tujuan mendapatkan kepastian sistem yang dibuat berhasil memenuhi kriteria tingkat penerimaan pengguna. UAT biasanya dijalankan kelompok pengguna yang berbeda, seperti *quality assurance engineer*, asosiasi bisnis, dan *customer support engineer* (Naik dan Tripathy, 2008). Secara umum UAT dikerjakan sebelum sistem dirilis dan dilakukan ketika sudah setelah melakukan pengujian sistem. Target responden pengujian ini adalah pengguna atau pemangku kepentingan yang terlibat dalam pengembangan

sistem. UAT memiliki kriteria penerimaan yang digunakan untuk membantu menilai penerimaan sistem.

Kriteria penerimaan terdiri dari beberapa aspek atribut. Atribut – atribut tersebut yang menentukan ada atau tidaknya kualitas dalam suatu sistem. Kriteria penerimaan memberikan ide yang luas mengenai kebutuhan dan ekspektasi pengguna, sehingga membuatnya menjadi terlalu banyak dan umum. Oleh karenanya diperbolehkan untuk memilih kriteria dan atribut yang cocok dengan situasi yang dibutuhkan (Naik dan Tripathy, 2008). Berikut kriteria penerimaan yang ada pada *User Acceptance Testing* menurut Naik dan Tripathy.

- *Functional Correctness and Completeness*, memiliki fokus untuk menjawab pertanyaan ‘apakah sistem yang dibuat benar – benar dapat melakukan apa yang pengguna inginkan?’.
- *Accuracy*, berfokus menjawab pertanyaan ‘apakah sistem memberikan hasil yang benar?’, dalam hal ini perhitungan yang dilakukan sistem dapat mendekati nilai yang diharapkan.
- *Data Integrity*, mengacu kepada performa yang dimiliki oleh sistem ketika melakukan menyediakan atau menyimpan data.
- *Data Conversion*, membahas kemampuan sistem dalam melakukan konversi data dan memastikan tidak ada masalah karenanya.
- *Backup and Recovery*, berfokus menjawab pertanyaan ‘bagaimana kemampuan sistem dalam proses *backup* dan pemulihan data?’.
- *Competitive Edge*, memiliki fokus menjawab kelebihan apa yang dimiliki oleh sistem yang dikembangkan dibandingkan dengan sistem yang lain.
- *Usability*, memiliki fokus pada menjawab pertanyaan ‘bagaimana kemudahan pengguna dalam mengoperasikan sistem, dan seberapa mudah sistem dipelajari?’.
- *Performance*, memiliki fokus mengukur kinerja sistem.
- *Start-Up Time*, memiliki fokus menjawab pertanyaan ‘berapa lama waktu yang dibutuhkan oleh sistem untuk mulai beroperasi?’.
- *Stress*, memiliki fokus pada kemampuan sistem yang harus mampu mengatasi beban proses yang sangat sulit dan tinggi.
- *Reliability and Availability*, memiliki fokus pada kemungkinan sebuah sistem dijalankan tanpa mengalami kegagalan pada waktu dan tempat yang telah ditentukan.
- *Maintainability and Serviceability*, memiliki fokus pada kemampuan yang dimiliki sistem untuk melakukan perbaikan dan mengalami evolusi.
- *Robustness*, mendefinisikan kemampuan yang dimiliki sistem untuk dapat memulihkan diri ketika terjadi kesalahan, kemampuan sistem untuk dapat

tetap beroperasi dalam kondisi yang terburuk, dan kemampuan sistem untuk dapat tetap andal dalam jangka waktu lama.

- *Timeliness*, merupakan kriteria yang memiliki fokus pada waktu penyelesaian sesuai perjanjian dengan pemangku kepentingan.
- *Confidentiality and Availability*, memiliki fokus pada persyaratan bahwasanya data harus dilindungi dari pembukaan data tanpa adanya tanggung jawab dari pengguna yang berwenang dan data harus bisa diamankan dari serangan (DoS).
- *Compatibility and Interoperability*, memiliki fokus pada kemampuan sistem untuk kesamaan pengoprasian walaupun dalam platform dan konfigurasi jaringan yang berbeda serta di berbagai aplikasi yang berbeda.
- *Compliance*, memiliki fokus pada kepatuhan sistem terhadap standar teknis yang relevan.
- *Installability and Upgradability*, untuk memastikan instalasi sistem dilakukan dengan benar dan dapat diperbarui ketika diterapkan di lingkungan pengguna.
- *Scalability*, adalah secara efektif sistem mampu menunjukkan kinerjanya.
- *Documentation*, memastikan mutu *user guide* harus tinggi.

2.12 Skala Likert

Definisi atau pengertian dari *Skala likert* adalah sebuah jenis pengukuran skala yang kegunaannya adalah sebagai alat ukur pendapat, sikap, dan persepsi dari satu orang maupun dari sekelompok orang mengenai suatu kejadian. Kejadian atau fenomena yang ditentukan oleh peneliti disebut sebagai variabel penelitian. Dalam *skala likert*, variabel dideskripsikan menjadi indikator variabel. Setelah itu indikator dijadikan titik tolak dalam menyusun item – item instrumen seperti pernyataan ataupun pertanyaan. Jawaban yang didapatkan dari tiap instrumen dapat berupa kata – kata yang diberi skor antara lain: (a) sangat setuju diberikan skor 5, (b) setuju diberikan skor 4, (c) netral diberikan skor 3, (d) tidak setuju diberikan skor 2, (e) sangat tidak setuju diberikan skor 1 (Sugiyono, 2012).

Untuk mendapatkan indeks persentase, langkah pertama yang perlu dilakukan adalah mencari total nilai. Untuk mendapatkan Total Nilai, dilakukan dengan cara menjumlahkan masing – masing skor jawaban *instrument* yang sebelumnya telah dikalikan dengan variabel '*n*' yang merupakan jumlah responden. Didapatkan rumus total nilai adalah:

$$\text{Total Nilai} = (5 \times n) + (4 \times n) + (3 \times n) + (2 \times n) + (1 \times n) \quad (2.1)$$

Langkah kedua adalah mencari Nilai Y. Nilai Y didapatkan dari mengalikan nilai '*N₁*' yang merupakan skor atau nilai paling tinggi dari *skala likert* yang digunakan dengan nilai '*n*', kemudian mengalikannya juga dengan nilai '*U*' yang merupakan jumlah kasus uji. Sehingga didapatkan rumus nilai Y adalah:

$$\text{Nilai Y} = N_1 \times n \times U \quad (2.2)$$



Setelah berhasil mendapat Total Nilai beserta Nilai Y, dicari Indeks Persentasenya. Menghitung nilai indeks persentase dapat dilakukan dengan cara total nilai dibagi dengan nilai Y, setelah itu dikali dengan 100%. Sehingga didapatkan rumus indeks persentase adalah:

$$\text{Index \%} = \left(\frac{\text{Total Nilai}}{\text{Nilai Y}} \right) \times 100 \quad (2.3)$$

Indeks persentase yang didapat kemudian dikategorikan sesuai kategori persentase. Kategori persentase diterangkan pada Tabel 2.11.

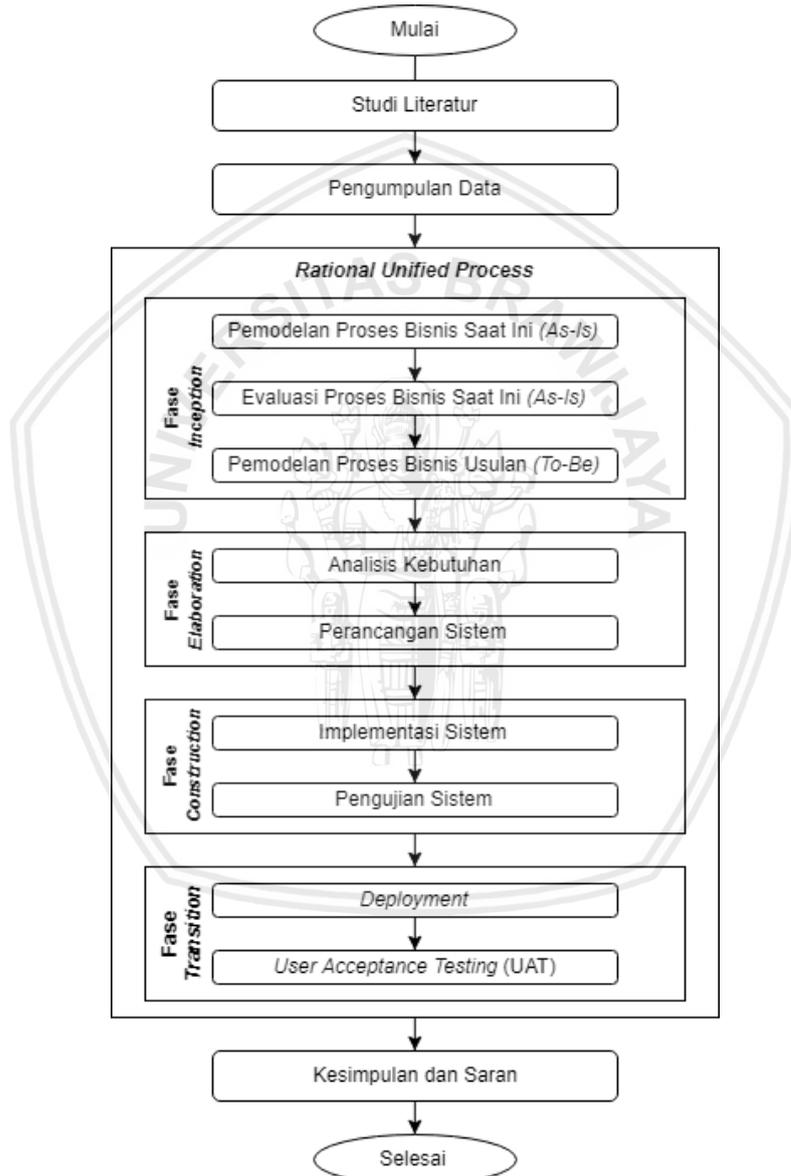
Tabel 2.11 Kategori Persentase

No.	Range (dalam Persen)	Kategori
1	80 s/d 100	Sangat Setuju
2	60 s/d 79,9	Setuju
3	40 s/d 59,9	Netral
4	20 s/d 39,9	Tidak Setuju
5	0 s/d 19,9	Sangat Tidak Setuju



BAB 3 METODOLOGI PENELITIAN

Penelitian yang dilakukan adalah penelitian yang bersifat implementatif pengembangan. Implementasinya dilakukan dengan menggunakan pendekatan berorientasi objek, penelitian ini menghasilkan sistem informasi yang dapat dimanfaatkan oleh hotel Edotel Pamekasan sebagai sarana untuk meningkatkan kegiatan proses bisnisnya. Berikut adalah langkah – langkah yang dilakukan oleh peneliti dalam melakukan penelitian ini, disajikan dalam Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian



3.1 Studi Literatur

Studi literatur pada penelitian ini merupakan sebuah tahap pembelajaran dengan mencari dan mengumpulkan referensi sebagai landasan penelitian maupun sebagai acuan teori yang dapat mendukung penelitian ini sehingga didapatkan kajian pustaka dan dasar teori yang akan membantu proses pengembangan sistem informasi perhotelan pada Edotel Pamekasan. Tujuan melakukan studi literatur adalah untuk mendapatkan kajian pustaka yang berisi bahasan penelitian terdahulu serta kaitannya dengan penelitian yang akan dilakukan dan bertujuan untuk mendapatkan informasi yang diperlukan untuk memperkuat dasar teori penelitian. Literatur yang akan digunakan adalah literatur yang berkaitan dengan *Rational Unified Process*, *Business Process*, *Unified Modelling Language*, *User Acceptance Testing*, sistem informasi perhotelan, sistem informasi reservasi, serta sistem informasi inventori yang dapat diperoleh dari pencarian di internet, jurnal, artikel, buku, *e-book*, dan dari beberapa sumber penelitian yang telah dilakukan sebelumnya.

3.2 Pengumpulan Data

Pada penelitian ini, pengumpulan data akan dilakukan dengan memakai teknik wawancara beserta Teknik observasi langsung terhadap objek penelitian. Wawancara dilakukan secara langsung kepada *general manager* Edotel Pamekasan untuk mendapatkan latar belakang permasalahan dan menggali kebutuhan. Sedangkan observasi dilakukan dengan mengamati aktivitas proses bisnis yang terjadi di Edotel Pamekasan untuk mendapatkan informasi lebih lanjut dan untuk menambah pengetahuan peneliti mengenai objek penelitian yang tidak didapatkan dari hasil wawancara. Pengumpulan data memiliki tujuan untuk mendapatkan latar belakang permasalahan, melihat dan menentukan urgensi dilakukannya penelitian, serta menggali sebanyak – banyaknya informasi yang diperlukan oleh penelitian.

3.3 Fase Inception

Fase pertama dari metode *Rational Unified Process* ialah Fase *Inception* yang dilakukan dengan tujuan untuk mendapatkan gambaran mengenai cakupan atau jangkauan penelitian dan mendapatkan proses bisnis pada objek penelitian berdasarkan informasi yang sudah didapatkan saat melakukan tahapan pengumpulan data. Terdapat 3 tahap yang akan dikerjakan pada fase *inception*, yaitu melakukan pemodelan proses bisnis saat ini yang sedang berjalan, kemudian dilanjutkan melakukan evaluasi proses bisnis yang saat ini sedang berjalan, dan terakhir tahap melakukan pemodelan solusi evaluasi berupa pemodelan proses bisnis usulan. Hasil yang telah didapatkan dari fase *inception* adalah pemodelan proses bisnis saat ini (*as-is*), evaluasi proses bisnis saat ini (*as-is*), dan pemodelan proses bisnis usulan (*to-be*).

3.3.1 Pemodelan Proses Bisnis Saat Ini (*As-Is*)

Pada tahap pemodelan proses bisnis saat ini (*as-is*), pemodelannya dibuat berdasarkan hasil yang diperoleh dari kegiatan wawancara dan observasi secara langsung kepada objek penelitian. Proses bisnis yang akan dimodelkan adalah reservasi, registrasi *check-in*, *check-out*, pengelolaan stok gudang, dan pelaporan operasional. Proses yang telah diidentifikasi akan dimodelkan ke dalam *Business Process Modeling Notation* (BPMN) agar lebih mudah dibaca dan dipahami. Hasil yang didapatkan dari tahap pemodelan proses bisnis saat ini (*as-is*) adalah pemodelan proses bisnis saat ini (*as-is*) di Edotel Pamekasan.

3.3.2 Evaluasi Proses Bisnis Saat Ini (*As-Is*)

Tahap selanjutnya adalah evaluasi proses bisnis saat ini (*as-is*) yang berisi evaluasi mengenai proses bisnis yang sedang berjalan. Tahap ini akan mendefinisikan kendala atau permasalahan yang terjadi pada proses bisnis yang sedang berjalan, kemudian mendefinisikan dampak dari berbagai masalah yang ada. Lalu setelah mendefinisikan permasalahan dan dampaknya, dilakukan analisis yang menghasilkan solusi yang digunakan untuk mengatasi permasalahan yang ada dan dampaknya. Hasil dari tahap ini adalah evaluasi proses bisnis saat ini (*as-is*) yang membahas kendala atau permasalahan yang terjadi, dampak dari permasalahan, dan solusi yang dapat dilakukan.

3.3.3 Pemodelan Proses Bisnis Usulan (*To-Be*)

Lalu tahap terakhir dalam fase *inception* adalah pemodelan proses bisnis usulan (*to-be*). Tahap ini akan memodelkan usulan proses bisnis yang dibuat berdasarkan solusi pada evaluasi proses bisnis saat ini (*as-is*) yang telah dilakukan sebelumnya. Usulan proses bisnis dimodelkan ke dalam *Business Process Modeling Notation* (BPMN) agar lebih mudah dibaca dan dipahami. Kemudian dilakukan analisis hasil perbaikan proses bisnis dengan cara membandingkan selisih waktu pengerjaan proses bisnis yang sedang berjalan dengan waktu pengerjaan usulan proses bisnis, yang menghasilkan persentase efisiensi waktu. Hasil dari tahap ini adalah pemodelan proses bisnis usulan (*to-be*) di Edotel Pamekasan.

3.4 Fase *Elaboration*

Fase berikutnya dalam metode *Rational Unified Process* adalah fase *elaboration*. Tujuan dari fase *elaboration* adalah untuk membuat kerangka sistem yang akan dikembangkan berdasarkan proses bisnis yang telah dimodelkan pada fase *inception*. Fase *elaboration* akan melakukan 2 tahap pengerjaan, yaitu tahap menganalisis hal – hal yang dibutuhkan sistem kemudian membuat rancangannya. Hasil dari fase *elaboration* adalah analisis kebutuhan dan perancangan sistem.

3.4.1 Analisis Kebutuhan

Tahap analisis kebutuhan adalah tahap pertama dalam fase *elaboration* yang bertujuan untuk menjelaskan dan menggambarkan kebutuhan sistem. Langkah – langkah yang akan dikerjakan pada tahap analisis kebutuhan adalah identifikasi

aktor yang terlibat dalam sistem, kemudian melakukan identifikasi fitur yang dibutuhkan, lalu melakukan analisis kebutuhan fungsional dan non-fungsional sistem berdasarkan identifikasi fitur yang telah dilakukan. Setelah selesai melakukan identifikasi kebutuhan, dilanjutkan dengan menggambar *use case diagram* lalu dihubungkan dengan kebutuhan fitur. Terakhir, mendeskripsikan perilaku dari setiap *use case* yang ada, yang dalam penelitian ini disebut sebagai membuat *use case scenario*. Hasil dari tahap ini adalah analisis kebutuhan sebagai dasar perancangan sistem.

3.4.2 Perancangan Sistem

Setelah melakukan analisis kebutuhan, fase *elaboration* dilanjutkan ke tahap perancangan sistem. Tahap ini memiliki tujuan untuk membuat kerangka yang dapat dimanfaatkan sebagai dasar implementasi sistem dengan cara memodelkan analisis kebutuhan ke dalam diagram – diagram serta merancang antarmuka pengguna. Tahap perancangan sistem dimulai dari memodelkan *sequence diagram* berdasarkan *use case scenario*, setelah itu melakukan pemodelan *class diagram*, lalu merancang *data model* dengan membuat abstrak basis data menggunakan *conceptual data model* lalu disempurnakan oleh *physical data model*. Akhir dari tahap perancangan sistem adalah membuat rancangan desain antarmuka pengguna atau *wireframe*. Hasil yang didapatkan dari tahapan ini adalah perancangan sistem yang dapat digunakan sebagai dasar implementasi sistem.

3.5 Fase Construction

Fase ketiga yang dilakukan dalam metode RUP adalah fase *construction*. Fase *construction* memiliki tujuan untuk merealisasikan perancangan sistem yang telah dibuat pada fase *elaboration* agar siap digunakan. Dalam fase *construction* terdapat 2 tahapan yang harus dilakukan, yakni melakukan implementasi kemudian menguji sistemnya. Hasil dari fase *construction* ialah implementasi sistem serta pengujian sistem.

3.5.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap pertama yang dilakukan dalam fase *construction*. Hal yang dilakukan pada tahap implementasi sistem adalah mengimplementasikan rancangan yang telah dikerjakan ke dalam baris kode program. Pada penelitian ini, implementasi sistem akan dilakukan menggunakan *framework* untuk memudahkan dan mempercepat pengerjaan. Tahap implementasi sistem dimulai dengan mengidentifikasi spesifikasi perangkat yang akan digunakan untuk mengimplementasikan sistem dimulai dari mengidentifikasi spesifikasi perangkat lunak hingga mengidentifikasi spesifikasi perangkat keras. Setelah mendapatkan identifikasi spesifikasi perangkat, dilanjutkan dengan melakukan implementasi *class* yang dibuat berdasarkan perancangan *class diagram*, lalu implementasi *database* berdasarkan *data model* yang telah dibuat, dan terakhir mengimplementasi antarmuka pengguna berdasarkan rancangan

antarmuka pengguna. Hasil dari tahap ini adalah implementasi sistem berupa sistem informasi perhotelan Edotel Pamekasan.

3.5.2 Pengujian Sistem

Sistem informasi perhotelan di Edotel Pamekasan merupakan sebuah sistem yang belum teruji. Oleh karenanya, akan dilakukan tahap pengujian sistem untuk menguji sistem yang telah dibangun. Tahapan pengujian sistem bertujuan untuk memastikan bahwa implementasi sistem telah sesuai dengan kebutuhan dan perancangan yang telah dilakukan. Pengujian akan dilakukan dengan menggunakan 2 metode, yaitu metode *black-box validation testing* yang digunakan untuk menguji kesesuaian kebutuhan fungsional dengan implementasinya dan metode *compatibility testing* yang digunakan untuk menguji kesesuaian kebutuhan non-fungsional dengan implementasinya. Hasil dari tahap ini adalah analisis hasil pengujian sistem.

3.6 Fase Transition

Fase keempat sekaligus fase terakhir dari metode *Rational Unified Process* adalah fase *transition*. Fase *transition* dilakukan dengan tujuan mengenalkan sistem yang telah dibangun kepada penggunanya yang dalam penelitian ini merupakan pegawai Edotel Pamekasan. Terdapat 2 tahapan yang akan dilakukan dalam fase *transition*, yaitu memasang sistem ke dalam lingkungan penggunanya kemudian melakukan pengujian penerimaan pengguna terhadap sistem. Hasil dari fase ini adalah *deployment* dan *User Acceptance Testing (UAT)*.

3.6.1 Deployment

Pada tahap *deployment*, sistem yang telah melewati tahap pengujian akan dikenalkan kepada lingkungan penggunanya. Pengenalan sistem dimulai dari melakukan identifikasi spesifikasi perangkat yang ada di Edotel Pamekasan mulai dari perangkat lunak hingga perangkat keras, setelah itu dilakukan instalasi sistem pada perangkat yang ada di Edotel Pamekasan. Hasil dari tahap ini adalah *deployment* sistem ke dalam lingkungannya.

3.6.2 User Acceptance Testing (UAT)

Tahap akhir dalam fase *transition* adalah melakukan *User Acceptance Testing*. Tujuan dari *User Acceptance Testing* adalah untuk dapat melihat sejauh mana penerimaan pengguna terhadap sistem informasi perhotelan di Edotel Pamekasan. *User Acceptance Testing* akan ditujukan kepada pengguna yang terlibat dalam pengembangan sistem dengan setiap pengguna menjalankan *test plan* yang berisi perintah yang harus dikerjakan. Setelah menjalankan *test plan*, pengguna akan mengisi dokumen kuesioner yang telah dirancang oleh peneliti. Hasil dari tahapan ini adalah analisis hasil penerimaan pengguna terhadap sistem.

3.7 Kesimpulan dan Saran

Langkah terakhir pada penelitian ini adalah menarik kesimpulan dan saran. Penarikan kesimpulan dibuat berdasarkan hasil penelitian yang telah dilakukan dan bertujuan untuk menjawab rumusan masalah yang telah didefinisikan pada pendahuluan. Selain itu terdapat juga penulisan saran yang bertujuan untuk memberi rekomendasi penelitian lebih lanjut mengenai pengembangan sistem informasi perhotelan Edotel Pamekasan.



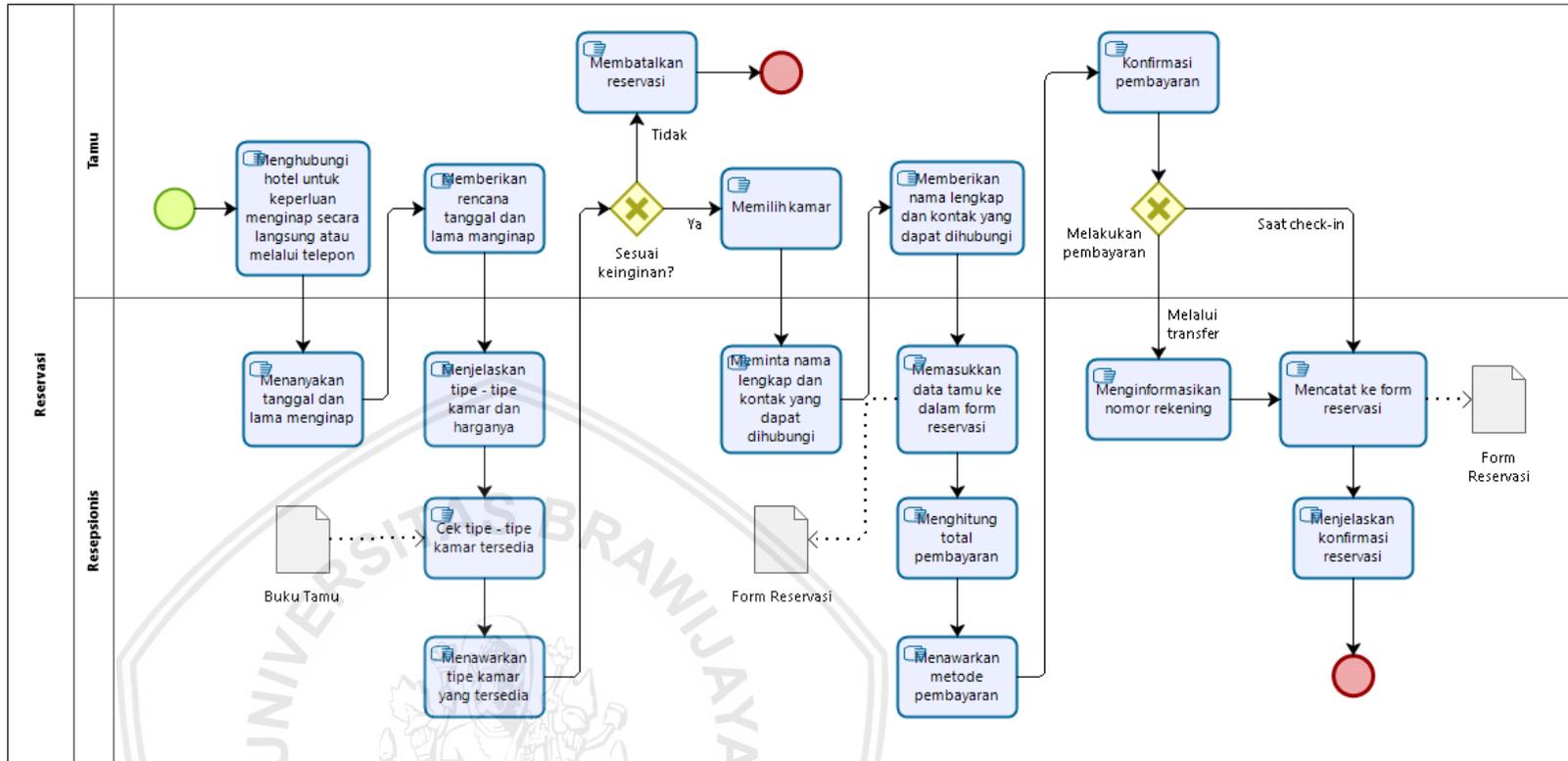
BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS KEBUTUHAN

4.1 Pemodelan Proses Bisnis Saat Ini (*As-Is*)

Pemodelan proses bisnis merupakan metode pemodelan yang memiliki tujuan untuk menggambarkan proses bisnis ke dalam notasi grafik untuk mempermudah pengguna dalam memahaminya. Pemodelan proses bisnis merupakan tahap awal dari fase *inception* di dalam RUP. Fase ini akan memodelkan proses bisnis yang sedang berjalan (*as-is*), lalu melakukan evaluasi yang kemudian digunakan sebagai dasar membuat proses bisnis usulan (*to-be*). Penelitian ini akan memodelkan proses bisnis milik Edotel Pamekasan yang berfokus pada proses reservasi, registrasi *check-in*, *check-out*, pengelolaan stok barang, dan pelaporan operasional.

4.1.1 Proses Bisnis Reservasi Saat ini (*As-Is*)

Proses bisnis utama yang dijalankan oleh Edotel Pamekasan adalah proses bisnis reservasi. Proses bisnis diawali tamu hotel menghubungi resepsionis untuk keperluan menginap secara langsung maupun dengan menggunakan telepon. Yang kemudian ditanggapi oleh resepsionis dengan menanyakan tanggal menginap yang diinginkan tamu. Setelah tamu memberikan rencana tanggal menginapnya, resepsionis menawarkan tipe – tipe kamar. Lalu, resepsionis mengecek kamar yang tersedia di tanggal yang diinginkan tamu. Kemudian resepsionis menawarkan kamar yang tersedia. Jika tidak cocok maka tamu dapat membatalkan reservasinya. Namun, jika cocok maka tamu dapat memilih kamar yang ingin dipesan. Kemudian resepsionis akan meminta data diri tamu lalu mengisinya ke dalam formulir reservasi. Setelahnya resepsionis akan menghitung total pembayaran secara manual, lalu menawarkan metode pembayaran transfer dan langsung. Setelah itu tamu memilih metode yang diinginkan, dan kemudian resepsionis mencatatnya di formulir reservasi. Terakhir resepsionis akan menginformasikan bahwa akan melakukan konfirmasi reservasi h-2 kedatangan tamu, dan proses bisnis selesai. Proses bisnis reservasi saat ini (*as-is*) dimodelkan dalam gambar 4.1 dan dijelaskan pada tabel 4.1.



Gambar 4.1 Pemodelan Proses Bisnis Reservasi Saat Ini (As-Is)

Tabel 4.1 Deskripsi *Task* Proses Bisnis Reservasi Saat Ini (*As-Is*)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	Tamu	Menghubungi hotel untuk keperluan menginap secara langsung atau melalui telepon	Tamu menghubungi pihak hotel secara langsung atau melalui telepon untuk keperluan menginap		10s	<i>Manual Task</i>
2	Resepsionis	Menanyakan tanggal dan lama menginap	Resepsionis menanyakan tanggal <i>check-in</i> dan tanggal <i>check-out</i> atau lama menginap kepada tamu		10s	<i>Manual Task</i>
3	Tamu	Memberikan rencana tanggal dan lama menginap	Tamu memberikan rencana tanggal <i>check-in</i> dan tanggal <i>check-out</i> atau lama menginapnya kepada resepsionis		30s	<i>Manual Task</i>
4	Resepsionis	Menjelaskan tipe - tipe kamar dan harganya	Resepsionis menjelaskan tipe – tipe kamar dan harganya kepada tamu		3m	<i>Manual Task</i>
5	Resepsionis	Cek tipe - tipe kamar tersedia	Resepsionis melakukan pengecekan tipe – tipe kamar yang tersedia dengan memeriksa satu persatu tanggal reservasi dengan nomor kamarnya, dan tanggal tamu yang sedang <i>check-in</i> dengan nomor kamarnya	Buku Tamu	3m	<i>Manual Task</i>
6	Resepsionis	Menawarkan tipe kamar yang tersedia	Resepsionis menawarkan tipe kamar yang tersedia kepada tamu setelah melakukan pengecekan tipe kamar		1m	<i>Manual Task</i>

Tabel 4.1 Deskripsi Task Proses Bisnis Reservasi Saat Ini (As-Is) (lanjutan)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
7	Tamu	Membatalkan reservasi	Tamu dapat membatalkan reservasi jika tipe kamar yang tersedia dirasa tidak ada yang cocok		10s	Manual Task
8	Tamu	Memilih kamar	Tamu memilih kamar jika tipe kamar yang tersedia dirasa cocok		10s	Manual Task
9	Resepsionis	Meminta nama lengkap dan kontak yang dapat dihubungi	Resepsionis meminta nama lengkap dan kontak nomor telepon yang dapat dihubungi kepada tamu		10s	Manual Task
10	Tamu	Memberikan nama lengkap dan kontak yang dapat dihubungi	Tamu memberi nama lengkap dan kontak nomor telepon yang dapat dihubungi kepada resepsionis		10s	Manual Task
11	Resepsionis	Memasukkan data tamu ke dalam formulir reservasi	Resepsionis memasukkan data tamu seperti nama, nomor telepon, tanggal serta lama menginap, dan pilihan kamar ke dalam formulir reservasi	Formulir Reservasi	2m	Manual Task
12	Resepsionis	Menghitung total pembayaran	Resepsionis menghitung total pembayaran yang harus dibayarkan oleh tamu		2m	Manual Task
13	Resepsionis	Menawarkan metode pembayaran	Resepsionis menawarkan metode pembayaran transfer atau pembayaran sesaat sebelum <i>check-in</i> kepada tamu		10s	Manual Task
14	Tamu	Konfirmasi pembayaran	Tamu memilih pembayaran yang diinginkan		10s	Manual Task
15	Resepsionis	Menginformasikan nomor rekening	Resepsionis menginformasikan nomor rekening kepada tamu		10s	Manual Task

Tabel 4.1 Deskripsi Task Proses Bisnis Reservasi Saat Ini (As-Is) (lanjutan)

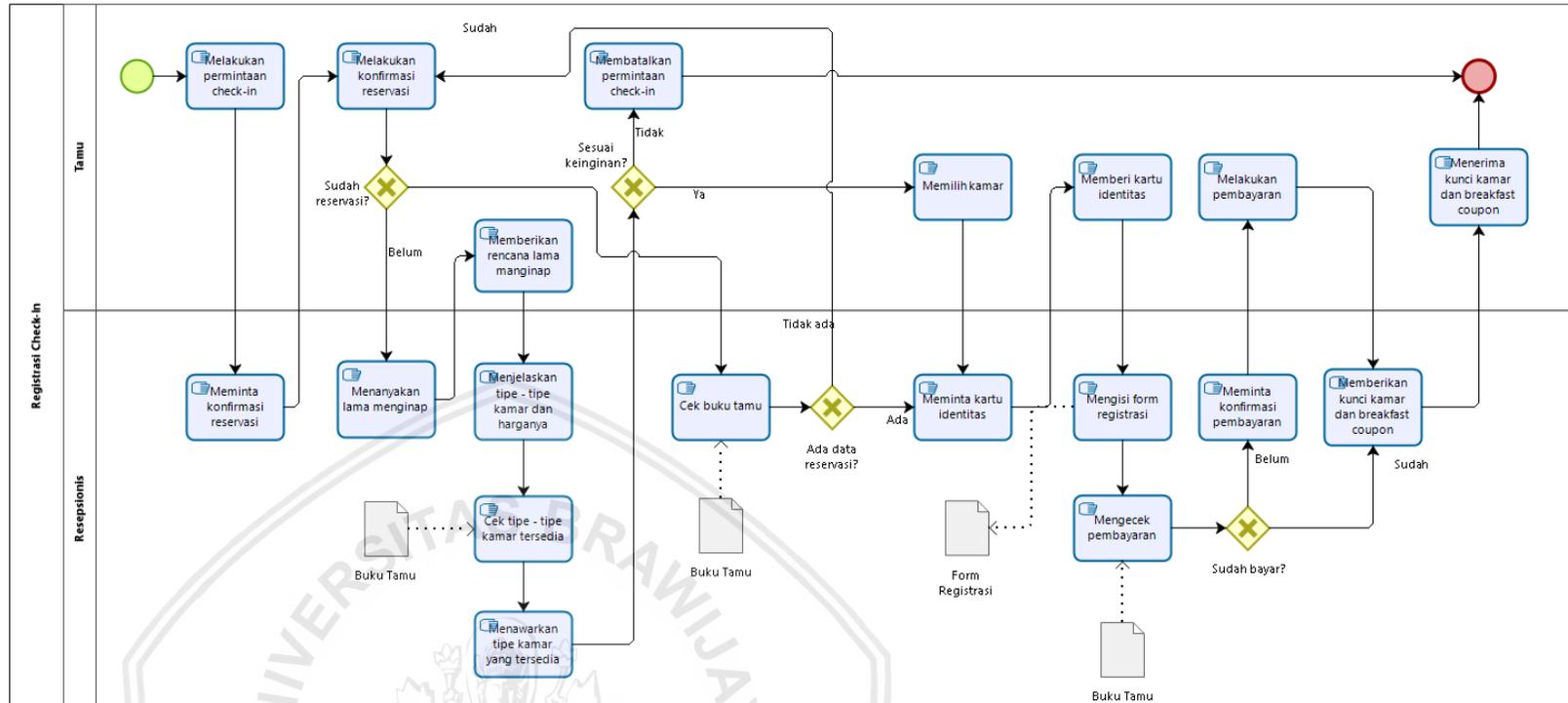
No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
16	Resepsionis	Mencatat ke formulir reservasi	Resepsionis mencatat metode pembayaran yang diinginkan tamu	Formulir Reservasi	10s	<i>Manual Task</i>
17	Resepsionis	Menjelaskan konfirmasi reservasi	Resepsionis menjelaskan perihal konfirmasi reservasi yang akan dilakukan pada h-2 kedatangan tamu		10s	<i>Manual Task</i>



4.1.2 Proses Bisnis Registrasi *Check-In* Saat Ini (*As-Is*)

Proses bisnis kedua adalah proses registrasi *check-in*. Proses bisnis dimulai dari tamu melakukan permintaan *check-in*. Kemudian resepsionis akan meminta konfirmasi reservasi dengan menanyakan apakah tamu telah melakukan reservasi atau belum. Lalu, tamu melakukan konfirmasi reservasi. Jika sudah reservasi, maka resepsionis akan mengecek formulir reservasi. Jika tidak ada data reservasinya, maka tamu melakukan konfirmasi reservasi ulang. Jika ada, maka resepsionis akan meminta kartu id tamu untuk ditahan sementara di hotel selama masa menginap. Kemudian resepsionis lanjut mengisi formulir registrasi. Jika tamu belum bayar, maka tamu diminta untuk melakukan pembayaran terlebih dahulu. Jika sudah bayar, tamu diberi kunci kamar dan kupon sarapan. Jika belum reservasi, resepsionis akan menginformasikan kamar – kamar yang ada. Lalu, resepsionis akan mengecek kamar yang tersedia dan kemudian menawarkan kamar yang tersedia ke tamu. Kemudian tamu memilih kamar. Lalu menjalankan proses bisnis meminta kartu id tamu hingga tamu menerima kunci kamar dan kupon sarapannya. Setelah tamu menerima kunci kamar dan kupon sarapannya, maka proses bisnis selesai. Proses bisnis ini dimodelkan dalam gambar 4.2 dan dijelaskan pada tabel 4.2.





Gambar 4.2 Pemodelan Proses Bisnis Registrasi *Check-In* Saat Ini (*As-Is*)

Tabel 4.2 Deskripsi *Task* Proses Bisnis Registrasi *Check-In* Saat Ini (*As-Is*)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	Tamu	Melakukan permintaan <i>check-in</i>	Tamu melakukan permintaan <i>check-in</i> kepada resepsionis		10s	Manual Task
2	Resepsionis	Meminta konfirmasi reservasi	Resepsionis meminta konfirmasi apakah sudah melakukan reservasi atau tidak kepada tamu		10s	Manual Task
3	Tamu	Melakukan konfirmasi reservasi	Tamu melakukan konfirmasi sudah melakukan reservasi atau tidak melakukan reservasi sebelumnya		10s	Manual Task
4	Resepsionis	Menanyakan lama menginap	Resepsionis menanyakan berapa malam tamu akan menginap jika tamu tidak melakukan reservasi		10s	Manual Task
5	Tamu	Memberikan rencana lama menginap	Tamu memberitahukan berapa lama ia akan menginap		30s	Manual Task
6	Resepsionis	Menjelaskan tipe - tipe kamar dan harganya	Resepsionis menjelaskan tipe – tipe kamar yang dimiliki hotel		3m	Manual Task
7	Resepsionis	Cek tipe - tipe kamar tersedia	Resepsionis melakukan pengecekan tipe – tipe kamar yang tersedia dengan memeriksa satu persatu tanggal reservasi dengan nomor kamarnya, dan tanggal tamu yang sedang <i>check-in</i> dengan nomor kamarnya	Buku Tamu	3m	Manual Task
8	Resepsionis	Menawarkan tipe kamar yang tersedia	Resepsionis menawarkan tipe kamar yang tersedia kepada tamu setelah melakukan pengecekan tipe kamar		1m	Manual Task

Tabel 4.2 Deskripsi *Task* Proses Bisnis Registrasi *Check-In* Saat Ini (*As-Is*) (lanjutan)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
9	Tamu	Membatalkan permintaan <i>check-in</i>	Tamu dapat membatalkan permintaan <i>check-innya</i> jika tipe kamar yang tersedia dirasa tidak ada yang cocok		10s	Manual Task
10	Tamu	Memilih kamar	Tamu memilih kamar jika tipe kamar yang tersedia dirasa cocok		10s	Manual Task
11	Resepsionis	Cek buku tamu	Resepsionis melakukan pengecekan buku tamu untuk mencari data reservasi tamu	Buku Tamu	3m	Manual Task
12	Resepsionis	Meminta kartu identitas	Resepsionis meminta kartu identitas tamu untuk ditahan sementara sebagai jaminan		10s	Manual Task
13	Tamu	Memberi kartu identitas	Tamu memberikan kartu identitasnya		10s	Manual Task
14	Resepsionis	Mengisi formulir registrasi	Resepsionis memasukkan data tamu seperti nomor identitas, nama, nomor telepon, tanggal serta lama menginap, dan pilihan kamar ke dalam formulir registrasi	Formulir Registrasi	2m	Manual Task
15	Resepsionis	Mengecek pembayaran	Resepsionis memeriksa catatan reservasi apakah tamu sudah melakukan pembayaran	Buku Tamu	10s	Manual Task
16	Resepsionis	Meminta konfirmasi pembayaran	Resepsionis meminta tamu melakukan pembayaran kamar		10s	Manual Task
17	Tamu	Melakukan pembayaran	Tamu melakukan pembayaran kamar		10s	Manual Task

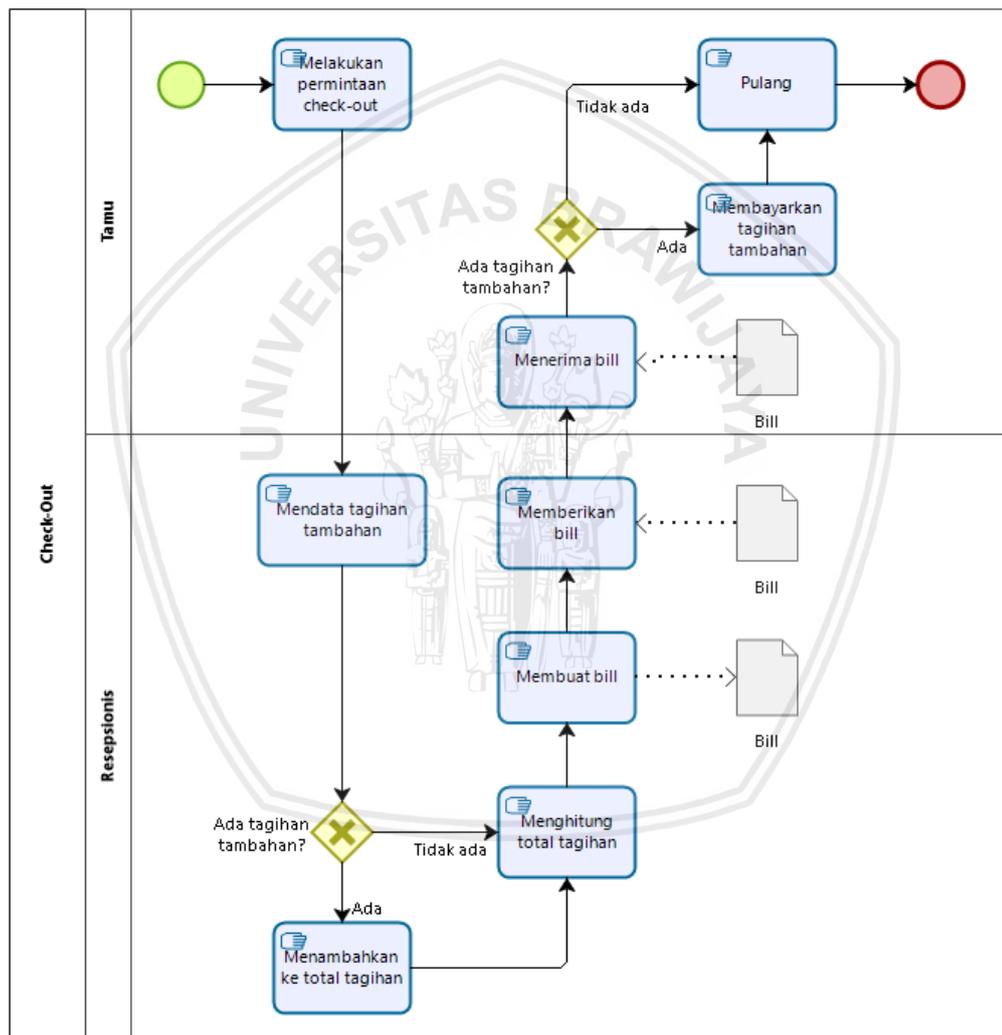
Tabel 4.2 Deskripsi *Task* Proses Bisnis Registrasi *Check-In* Saat Ini (*As-Is*) (lanjutan)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
18	Resepsionis	Memberikan kunci kamar dan <i>breakfast coupon</i>	Resepsionis memberikan kunci kamar dan <i>breakfast coupon</i> kepada tamu		10s	<i>Manual Task</i>
19	Tamu	Menerima kunci kamar dan <i>breakfast coupon</i>	Tamu menerima kunci kamar dan <i>breakfast couponnya</i>		10s	<i>Manual Task</i>



4.1.3 Proses Bisnis *Check-Out* Saat Ini (*As-Is*)

Proses bisnis ketiga adalah proses *check-out* saat ini. Proses bisnis diawali dengan tamu melakukan permintaan *check-out*. Kemudian resepsionis akan mendaftarkan kemungkinan tagihan tambahan tamu. Jika ada tagihan tambahan, resepsionis menambahkan ke total tagihan. Jika tidak ada, resepsionis langsung menghitung total tagihan. Lalu resepsionis membuat *bill* kemudian memberikan *bill*-nya ke tamu. Setelah menerima *bill*, tamu membayar tagihan tambahannya jika ada. Jika tidak ada, tamu bisa langsung pulang dan proses bisnis selesai. Proses bisnis ini dimodelkan dalam gambar 4.3 dan dijelaskan pada tabel 4.3.



Powered by
bizagi
Modeler

Gambar 4.3 Pemodelan Proses Bisnis *Check-Out* Saat Ini (*As-Is*)

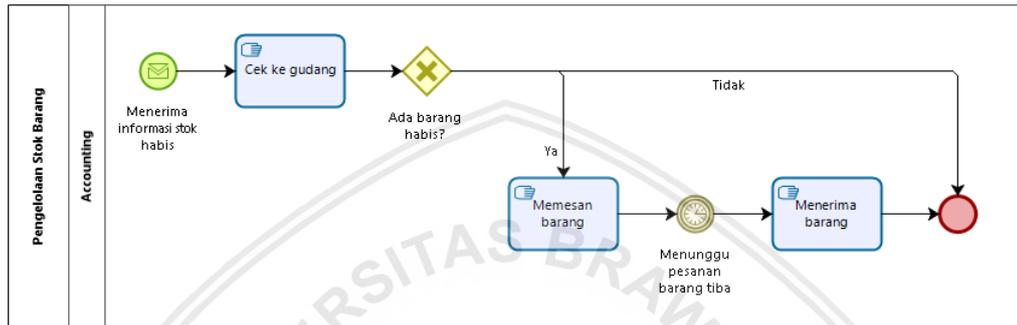


Tabel 4.3 Deskripsi Task Proses Bisnis Check-Out Saat Ini (As-Is)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	Tamu	Melakukan permintaan <i>check-out</i>	Tamu mendatangi resepsionis untuk keperluan <i>check-out</i>		10s	Manual Task
2	Resepsionis	Mendata tagihan tambahan	Resepsionis mendata apakah tamu memiliki tagihan tambahan		3m	Manual Task
3	Resepsionis	Menambahkan ke total tagihan	Resepsionis menambahkan data tagihan ke total tagihan jika ada tagihan tambahan yang dimiliki tamu		2m	Manual Task
4	Resepsionis	Menghitung total tagihan	Resepsionis menghitung total tagihan yang harus dibayarkan tamu		2m	Manual Task
5	Resepsionis	Membuat <i>bill</i>	Resepsionis membuat <i>bill</i>	<i>Bill</i>	2m	Manual Task
6	Resepsionis	Memberikan <i>bill</i>	Resepsionis memberikan <i>bill</i> kepada tamu	<i>Bill</i>	10s	Manual Task
7	Tamu	Menerima <i>bill</i>	Tamu menerima <i>bill</i> yang diserahkan resepsionis	<i>Bill</i>	10s	Manual Task
8	Tamu	Membayarkan tagihan tambahan	Tamu membayarkan sisa tagihan tambahannya jika ada		10s	Manual Task
9	Tamu	Pulang	Tamu pulang		10s	Manual Task

4.1.4 Proses Bisnis Pengelolaan Stok Gudang Saat Ini (As-Is)

Proses bisnis keempat adalah proses pengelolaan stok gudang saat ini. Proses dimulai saat *start message* aktor menerima informasi ada stok barang yang habis, kemudian *accounting* akan melakukan cek langsung ke gudang. Jika ternyata stok barang masih ada, maka proses bisnis selesai. Jika stok habis, maka *accounting* akan memesan barang, lalu menunggu barang datang. Setelah barang diterima, maka proses bisnis selesai. Proses bisnis ini dimodelkan dalam gambar 4.4 dan dijelaskan pada tabel 4.4.



Gambar 4.4 Pemodelan Proses Bisnis Pengelolaan Stok Gudang Saat ini (As-Is)

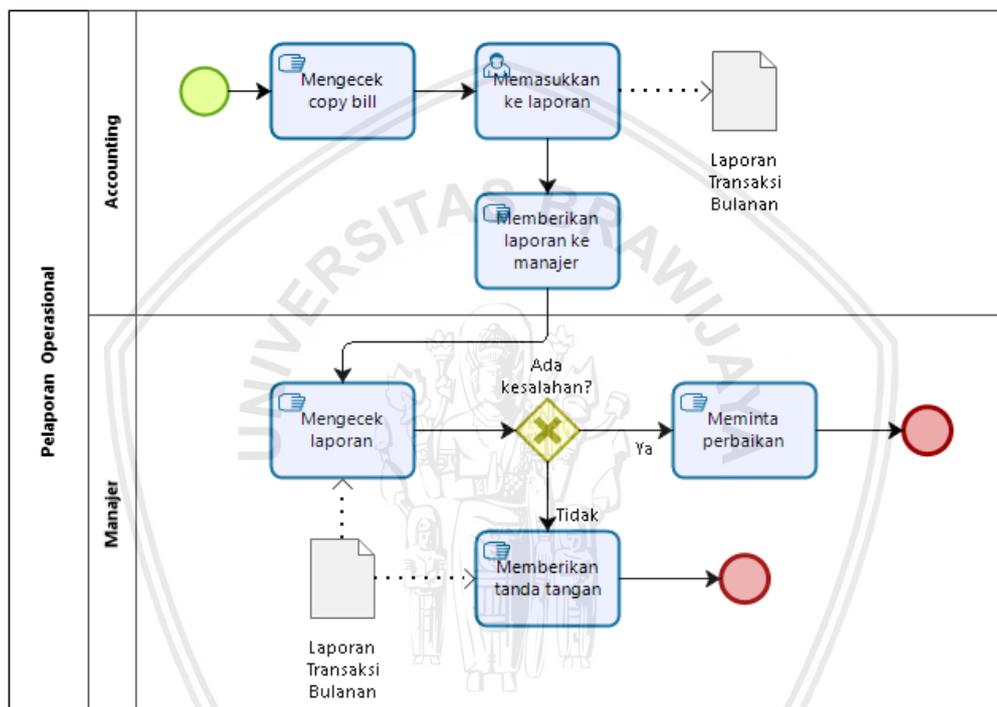
Tabel 4.4 Deskripsi *Task* Proses Bisnis Pengelolaan Stok Gudang Saat Ini (*As-Is*)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	<i>Accounting</i>	Cek ke gudang	<i>Accounting</i> melakukan pengecekan barang yang diinformasikan habis secara langsung ke gudang		5m	<i>Manual Task</i>
2	<i>Accounting</i>	Memesan barang	<i>Accounting</i> memesan barang yang habis		5m	<i>Manual Task</i>
3	<i>Accounting</i>	Menerima barang	<i>Accounting</i> menerima barang yang dipesan		10s	<i>Manual Task</i>



4.1.5 Proses Bisnis Pelaporan Operasional Saat Ini (As-Is)

Proses bisnis kelima adalah proses pelaporan operasional saat ini. Proses dimulai dengan *accounting* melakukan pengecekan terhadap copy bill transaksi tamu. Kemudian memasukkannya ke laporan. Setelah selesai membuat laporannya, *accounting* menyerahkan laporan ke manajer. Kemudian, manajer akan mengecek laporannya. Jika terdapat kesalahan, manajer akan meminta perbaikan laporan. Jika tidak, manajer akan memberikan tanda tangan lalu proses bisnis selesai. Proses bisnis ini dimodelkan dalam gambar 4.5 dan dijelaskan pada tabel 4.5.



Powered by
bizagi
Modeler

Gambar 4.5 Pemodelan Proses Bisnis Pelaporan Operasional Saat Ini (As-Is)



Tabel 4.5 Deskripsi *Task* Proses Bisnis Pelaporan Operasional Saat Ini (*As-Is*)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	Accounting	Mengecek <i>copy bill</i>	Accounting mengecek <i>copy bill</i> transaksi tamu hotel		2h	Manual Task
2	Accounting	Memasukkan ke laporan	Accounting memasukkan <i>copy bill</i> yang sudah di cek ke dalam laporan	Laporan Bulanan Transaksi	5h	User Task
3	Accounting	Memberikan laporan ke manajer	Accounting memberikan laporan yang sudah dikerjakan		10s	Manual Task
4	Manajer	Mengecek laporan	Manajer mengecek laporan	Laporan Bulanan Transaksi	1h	Manual Task
5	Manajer	Meminta perbaikan	Manajer meminta perbaikan jika terdapat kesalahan pada laporan		10s	Manual Task
6	Manajer	Memberikan tanda tangan	Manajer memberikan tanda tangan tanda laporan telah disetujui	Laporan Bulanan Transaksi	10s	Manual Task

4.2 Evaluasi Proses Bisnis Saat Ini (*As-Is*)

Evaluasi proses bisnis saat ini (*as-is*) memiliki tujuan untuk menganalisis permasalahan atau kendala yang ada pada proses bisnis sekarang. Kemudian melakukan analisis dampak yang terjadi dari permasalahan atau kendala yang ada. Lalu menawarkan solusi untuk mengatasi kendala atau permasalahan, dan dampaknya.

Tabel 4.6 Analisis Evaluasi Proses Bisnis Reservasi

Kendala/ Permasalahan	<ol style="list-style-type: none"> 1. Kegiatan reservasi yang dilakukan masih secara konvensional 2. Pencatatan buku tamu tidak dilakukan dengan benar
Dampak	<ol style="list-style-type: none"> 1. Proses reservasi memakan waktu yang lama karena proses mencari kamar tersedia 2. Dapat terjadi tumpang tindih pesanan kamar karena pencatatan yang tidak rapi
Solusi	<ol style="list-style-type: none"> 1. Menyediakan sistem informasi yang memfasilitasi proses reservasi kamar hotel 2. Menyediakan sistem yang mampu menunjukkan kamar – kamar yang tersedia di tanggal tertentu, sehingga kamar yang sudah dipesan tidak akan ditawarkan ke tamu

Tabel 4.6 mendeskripsikan evaluasi proses bisnis reservasi mulai dari kendala atau permasalahan yang terjadi, seperti kegiatan reservasi yang dilakukan masih secara konvensional. Dampak dari kendala atau permasalahan, seperti proses reservasi memakan waktu yang lama karena proses mencari kamar tersedia. Lalu solusinya, yaitu menyediakan sistem informasi yang memfasilitasi proses reservasi kamar hotel.

Tabel 4.7 Analisis Evaluasi Proses Bisnis Registrasi *Check-In*

Kendala/ Permasalahan	<ol style="list-style-type: none"> 1. Kegiatan registrasi yang dilakukan masih secara konvensional 2. Pencatatan buku tamu tidak dilakukan dengan rapi 3. Resepsionis harus menulis ulang formulir registrasi ketika tamu akan melakukan <i>check-in</i> walau sudah melakukan reservasi
Dampak	<ol style="list-style-type: none"> 1. Mencari kamar tersedia memakan waktu dan tidak efisien

Tabel 4.7 Analisis Evaluasi Proses Bisnis Registrasi *Check-In* (lanjutan)

	<ol style="list-style-type: none"> 2. Dapat terjadi tumpang tindih pesanan kamar karena pencatatan yang tidak rapi 3. Kegiatan menulis ulang formulir registrasi merupakan proses yang tidak efisien
Solusi	<ol style="list-style-type: none"> 1. Menyediakan sistem informasi yang memfasilitasi proses registrasi 2. Menyediakan sistem yang mampu menunjukkan kamar – kamar yang tersedia di tanggal tertentu, sehingga kamar yang sudah dipesan tidak akan ditawarkan ke tamu 3. Menyediakan sistem yang mampu memindahkan formulir reservasi ke dalam formulir registrasi sehingga tidak perlu menulis ulang isi yang ada di formulir reservasi

Tabel 4.7 mendeskripsikan evaluasi proses bisnis registrasi *check-in* mulai dari kendala atau permasalahan yang terjadi, seperti resepsionis harus menulis ulang formulir registrasi ketika tamu akan melakukan *check-in* walau sudah melakukan reservasi. Dampak dari kendala atau permasalahan, adalah menulis ulang formulir registrasi merupakan proses yang tidak efisien. Lalu solusinya, yaitu menyediakan sistem yang mampu memindahkan formulir reservasi ke dalam formulir registrasi sehingga tidak perlu menulis ulang isi yang ada di formulir reservasi.

Tabel 4.8 Analisis Evaluasi Proses Bisnis *Check-Out*

Kendala/ Permasalahan	<ol style="list-style-type: none"> 1. <i>Bill</i> yang ditulis dapat terselip oleh pegawai 2. Total tagihan dihitung secara manual
Dampak	<ol style="list-style-type: none"> 1. Kehilangan <i>record</i> transaksi tamu hotel 2. Terjadi salah hitung
Solusi	<ol style="list-style-type: none"> 1. Menyediakan sistem informasi yang dapat mencatat transaksi tamu hotel 2. Menyediakan sistem yang mampu menghitung total tagihan tamu hotel sekaligus mencetak <i>bill</i>

Tabel 4.8 mendeskripsikan evaluasi proses bisnis *check-out* mulai dari kendala atau permasalahan yang terjadi, seperti total tagihan dihitung secara manual. Dampak dari kendala atau permasalahan, adalah terjadi salah hitung. Lalu solusinya, yaitu menyediakan sistem yang mampu menghitung total tagihan tamu hotel sekaligus mencetak *bill*.

Tabel 4.9 Analisis Evaluasi Proses Bisnis Pengelolaan Stok Gudang

Kendala/ Permasalahan	<ol style="list-style-type: none"> 1. Tidak ada pencatatan barang dan stoknya di gudang 2. Ketika stok barang habis baru memesan barang
Dampak	<ol style="list-style-type: none"> 1. Jumlah stok barang tidak diketahui 2. Tidak ada catatan stok barang 3. Harus menunggu pesanan sampai, sehingga terpaksa mengurangi fasilitas dengan mengganti dengan yang ada
Solusi	<ol style="list-style-type: none"> 1. Menyediakan sistem informasi yang dapat mencatat barang masuk dan barang keluar dari gudang sehingga stok barang dapat dipantau

Tabel 4.9 mendeskripsikan evaluasi proses bisnis pengelolaan stok gudang mulai dari kendala atau permasalahan yang terjadi, seperti tidak ada pencatatan barang dan stoknya di gudang. Dampak dari kendala atau permasalahan, adalah jumlah stok barang tidak diketahui. Lalu solusinya, yaitu menyediakan sistem informasi yang dapat mencatat barang masuk dan barang keluar dari gudang sehingga stok barang dapat dipantau.

Tabel 4.10 Analisis Proses Bisnis Pelaporan Operasional

Kendala/ Permasalahan	<ol style="list-style-type: none"> 1. Pengecekan <i>bill</i> dan pembuatan laporan memakan waktu 1 hari kerja dan bisa lebih 2. Tidak ada pelaporan penggunaan barang
Dampak	<ol style="list-style-type: none"> 1. Permintaan laporan tidak bisa diminta sewaktu – waktu oleh manajer 2. Tidak ada catatan stok barang
Solusi	<ol style="list-style-type: none"> 1. Menyediakan sistem informasi yang dapat menampilkan laporan bulanan transaksi secara langsung sehingga dapat diminta oleh manajer sewaktu – waktu 2. Menyediakan sistem informasi yang dapat melakukan pelaporan penggunaan barang

Tabel 4.10 mendeskripsikan evaluasi proses bisnis pelaporan operasional mulai dari kendala atau permasalahan yang terjadi, seperti pengecekan *bill* dan pembuatan laporan memakan waktu 1 hari kerja dan bisa lebih. Dampak dari kendala atau permasalahan, adalah permintaan laporan tidak bisa diminta sewaktu – waktu oleh manajer. Lalu solusinya, yaitu menyediakan sistem

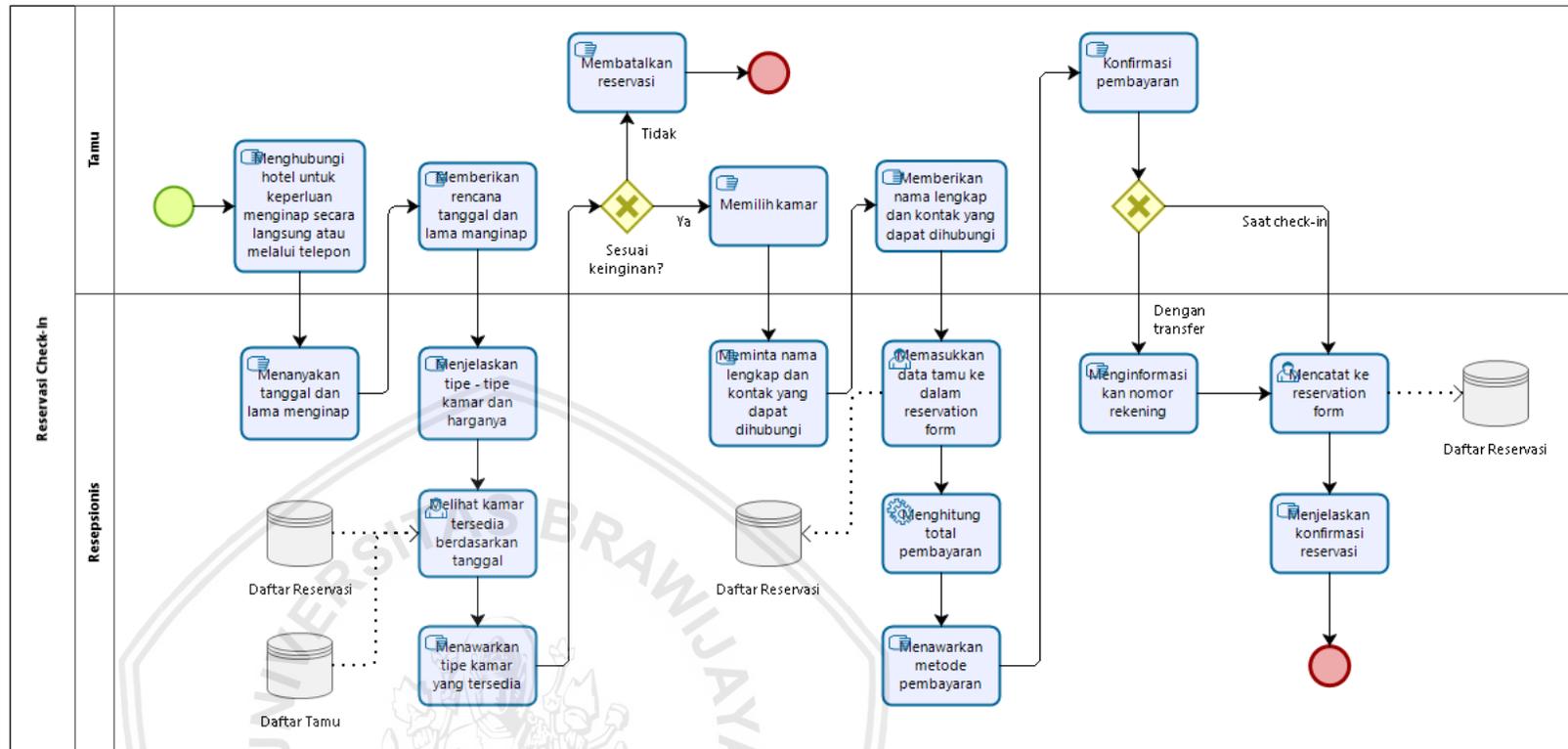
informasi yang dapat menampilkan laporan bulanan transaksi secara langsung sehingga dapat diminta oleh manajer sewaktu – waktu.

4.3 Pemodelan Proses Bisnis Usulan (*To-Be*)

Pemodelan proses bisnis usulan (*to-be*) merupakan tahapan dalam fase *inception* yang melakukan pemodelan pada usulan proses bisnis berdasarkan solusi pada evaluasi proses bisnis saat ini (*as-is*) yang diajukan kepada Edotel Pamekasan. Tahap ini diharapkan dapat menyelesaikan permasalahan yang terjadi dan dampaknya. Fokus pemodelan proses bisnis usulan (*to-be*) terletak pada proses bisnis reservasi, registrasi *check-in*, *check-out*, stok gudang, dan pelaporan operasional hotel.

4.3.1 Proses Bisnis Reservasi Usulan (*To-Be*)

Perbaikan proses bisnis utama yang dijalankan oleh Edotel Pamekasan adalah proses bisnis reservasi. Proses bisnis usulan ini diawali dengan tamu hotel menghubungi resepsionis untuk keperluan menginap secara langsung maupun dengan menggunakan telepon. Yang kemudian ditanggapi oleh resepsionis dengan menanyakan tanggal menginap yang diinginkan tamu. Setelah tamu memberikan rencana tanggal menginapnya, resepsionis menawarkan tipe – tipe kamar. Lalu, resepsionis melakukan input terhadap tanggal menginap yang diinginkan tamu, kemudian sistem akan menampilkan kamar yang tersedia pada tanggal yang diinginkan. Kemudian, resepsionis menawarkan kamar yang tersedia. Jika tidak cocok maka tamu dapat membatalkan reservasinya. Namun, jika cocok maka tamu dapat memilih kamar yang ingin dipesan. Kemudian, resepsionis akan meminta data diri tamu lalu mengisinya ke dalam formulir reservasi yang ada dalam sistem. Setelahnya sistem secara otomatis akan menampilkan jumlah yang harus dibayarkan, lalu menawarkan metode pembayaran transfer dan langsung. Setelah itu tamu memilih metode yang diinginkan, dan kemudian resepsionis mencatatnya di formulir reservasi. Terakhir resepsionis akan menginformasikan bahwa akan melakukan konfirmasi reservasi h-2 kedatangan tamu, dan proses bisnis selesai. Proses bisnis reservasi usulan (*to-be*) dimodelkan dalam gambar 4.6 dan dideskripsikan pada tabel 4.11.



Gambar 4.6 Pemodelan Proses Bisnis Reservasi Usulan (To-Be)

Tabel 4.11 Deskripsi Task Proses Bisnis Reservasi Usulan (To-Be)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	Tamu	Menghubungi hotel untuk keperluan menginap secara langsung atau melalui telepon	tamu menghubungi pihak hotel secara langsung atau melalui telepon untuk keperluan menginap		10s	Manual Task
2	Resepsionis	Menanyakan tanggal dan lama menginap	Resepsionis menanyakan tanggal <i>check-in</i> dan tanggal <i>check-out</i> atau lama menginap kepada tamu		10s	Manual Task
3	Tamu	Memberikan rencana tanggal dan lama manginap	Tamu memberikan rencana tanggal <i>check-in</i> dan tanggal <i>check-out</i> atau lama menginapnya kepada resepsionis		30s	Manual Task
4	Resepsionis	Menjelaskan tipe - tipe kamar dan harganya	Resepsionis menjelaskan tipe – tipe kamar dan harganya kepada tamu		3m	Manual Task
5	Resepsionis	Melihat kamar tersedia berdasarkan tanggal	Resepsionis melakukan <i>input</i> tanggal yang diinginkan tamu kemudian sistem secara otomatis akan menampilkan daftar kamar yang tersedia	Daftar Reservasi, Daftar Tamu	30s	User Task
6	Resepsionis	Menawarkan tipe kamar yang tersedia	Resepsionis menawarkan tipe kamar yang tersedia kepada tamu berdasarkan daftar kamar yang tersedia		1m	Manual Task

Tabel 4.11 Deskripsi Task Proses Bisnis Reservasi Usulan (To-Be) (lanjutan)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
7	Tamu	Membatalkan reservasi	Tamu dapat membatalkan reservasi jika tipe kamar yang tersedia dirasa tidak ada yang cocok		10s	Manual Task
8	Tamu	Memilih kamar	Tamu memilih kamar jika tipe kamar yang tersedia dirasa cocok		10s	Manual Task
9	Resepsionis	Meminta nama lengkap dan kontak yang dapat dihubungi	Resepsionis meminta nama lengkap dan kontak nomor telepon yang dapat dihubungi kepada tamu		10s	Manual Task
10	Tamu	Memberikan nama lengkap dan kontak yang dapat dihubungi	Tamu memberi nama lengkap dan kontak nomor telepon yang dapat dihubungi kepada resepsionis		10s	Manual Task
11	Resepsionis	Memasukkan data tamu ke dalam formulir reservasi	Resepsionis menggunakan sistem untuk memasukkan data tamu seperti nama, nomor telepon, tanggal serta lama menginap, dan pilihan kamar ke dalam formulir reservasi	Daftar Reservasi	1m	User Task
12	Resepsionis	Menghitung total pembayaran	Setelah memilih kamar, sistem secara otomatis akan menghitung total pembayaran yang harus dibayarkan		5s	Service Task
13	Resepsionis	Menawarkan metode pembayaran	Resepsionis menawarkan metode pembayaran transfer atau pembayaran sesaat sebelum <i>check-in</i> kepada tamu		10s	Manual Task

Tabel 4.11 Deskripsi *Task* Proses Bisnis Reservasi Usulan (*To-Be*) (lanjutan)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
14	Tamu	Konfirmasi pembayaran	Tamu memilih pembayaran yang diinginkan		10s	<i>Manual Task</i>
15	Resepsionis	Menginformasikan nomor rekening	Resepsionis menginformasikan nomor rekening kepada tamu		10s	<i>Manual Task</i>
16	Resepsionis	Mencatat ke formulir reservasi	Resepsionis mencatat metode pembayaran yang diinginkan tamu menggunakan sistem	Daftar Reservasi	10s	<i>User Task</i>
17	Resepsionis	Menjelaskan konfirmasi reservasi	Resepsionis menjelaskan perihal konfirmasi reservasi yang akan dilakukan pada h-2 kedatangan tamu		10s	<i>Manual Task</i>



Berdasarkan dampak dari permasalahan yang ada pada proses bisnis reservasi, yaitu proses pencarian kamar yang memakan waktu, dibuatlah sebuah sistem informasi yang mengganti proses pencarian kamar dan beberapa proses lainnya yang masih dilakukan secara manual (*manual task*) menjadi proses yang dilakukan dengan menggunakan sistem informasi (*user task*). Dengan perubahan ini, proses bisnis resevasi mengalami peningkatan efisiensi waktu sebesar 43,13%. Detail waktu proses bisnis saat ini (*as-is*) dan proses bisnis usulan (*to-be*) ditunjukkan pada tabel 4.12.

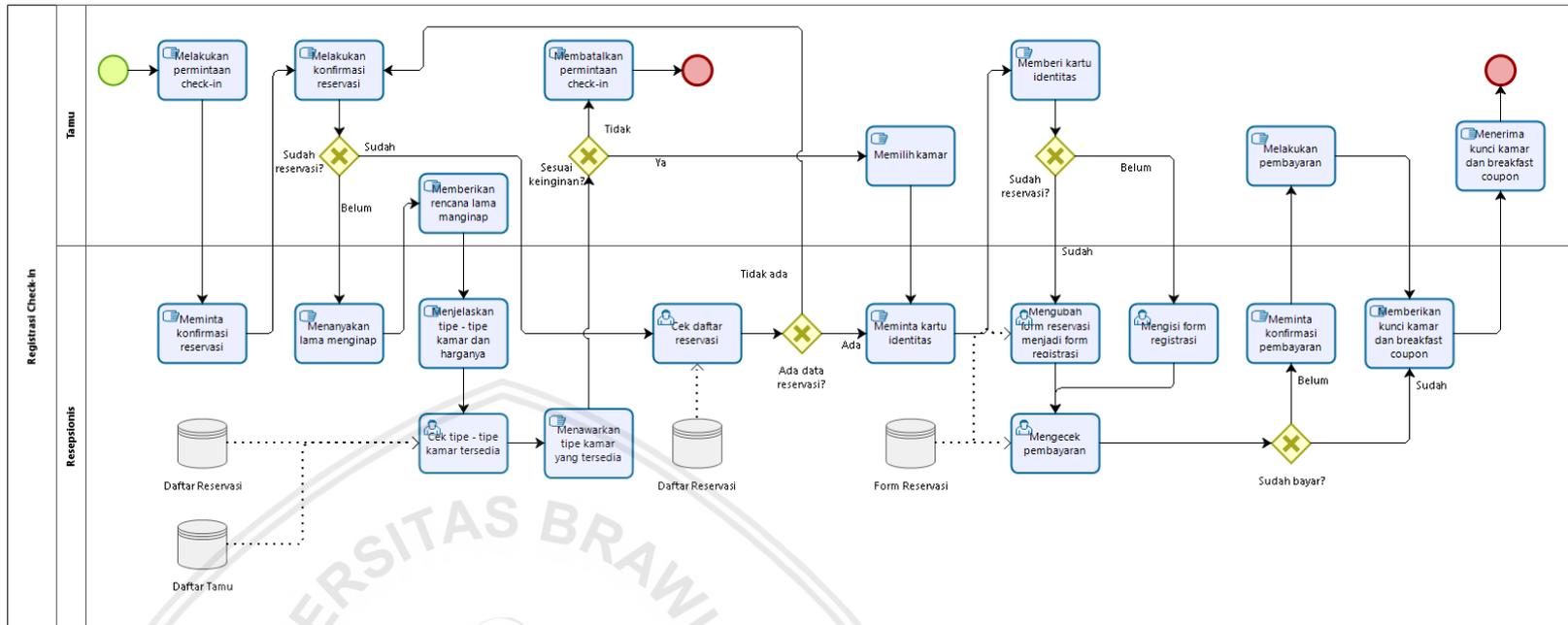
Tabel 4.12 Analisis Hasil Perbaikan Proses Bisnis Reservasi

No.	Proses Bisnis	Waktu <i>As-Is</i>	Waktu <i>To-Be</i>	Efisiensi Waktu
1	Reservasi	13 menit 20 detik	7 menit 35 detik	43,13%

Tabel 4.12 menjelaskan total waktu dari proses bisnis saat ini (*as-is*), proses bisnis usulan (*to-be*), dan persentase efisiensi waktu dari proses bisnis reservasi. Total waktu pengerjaan proses bisnis saat ini (*as-is*) adalah 13 menit 20 detik, sedangkan total waktu pengerjaan proses bisnis usulan (*to-be*) adalah 7 menit 35 detik. Sehingga, proses bisnis reservasi mendapat peningkatan efisiensi waktu sebesar 43,13%

4.3.2 Proses Bisnis Registrasi *Check-In* Usulan (*To-Be*)

Proses bisnis usulan kedua adalah proses registrasi *check-in*. Proses bisnis dimulai dari tamu melakukan permintaan *check-in*. Kemudian, resepsionis akan meminta konfirmasi reservasi dengan menanyakan apakah tamu telah melakukan reservasi atau belum. Lalu, tamu melakukan konfirmasi reservasi. Jika sudah reservasi, maka resepsionis akan mengecek data reservasi untuk mencari apakah ada nama tamunya. Jika tidak ada data reservasinya, maka tamu dapat melakukan *walk-in guest check-in* atau *check-in* langsung. Jika sudah reservasi, maka resepsionis akan meminta kartu id tamu untuk ditahan sementara di hotel selama masa menginap. Kemudian resepsionis melengkapi formulir registrasinya. Jika tamu belum bayar, maka tamu diminta untuk melakukan pembayaran terlebih dahulu. Jika sudah bayar, tamu diberi kunci kamar dan kupon sarapan. Jika belum reservasi, maka tamu akan melakukan *walk-in guest check-in*. Resepsionis akan menanyakan lama menginap. Kemudian, setelah tamu menginformasikan lama menginapnya, resepsionis akan menginformasikan kamar – kamar yang ada. Lalu, resepsionis akan mengecek kamar yang tersedia menggunakan sistem dan kemudian menawarkan kamar yang tersedia ke tamu. Kemudian tamu memilih kamar. Lalu menjalankan proses bisnis meminta kartu id tamu sama seperti ketika tamu sudah melakukan reservasi hingga tamu menerima kunci kamar dan kupon sarapannya. Setelah tamu menerima kunci kamar dan kupon sarapannya, maka proses bisnis selesai. Proses bisnis ini dimodelkan dalam gambar 4.7 dan dideskripsikan pada tabel 4.13.



Powered by
bizagi
Modeler

Gambar 4.7 Pemodelan Proses Bisnis Registrasi *Check-In*
Usulan (*To-Be*)

Tabel 4.13 Deskripsi Task Proses Bisnis Registrasi Check-In Usulan (To-Be)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	Tamu	Melakukan permintaan <i>check-in</i>	Tamu melakukan permintaan <i>check-in</i> kepada resepsionis		10s	Manual Task
2	Resepsionis	Meminta konfirmasi reservasi	Resepsionis meminta konfirmasi apakah sudah melakukan reservasi atau tidak kepada tamu		10s	Manual Task
3	Tamu	Melakukan konfirmasi reservasi	Tamu melakukan konfirmasi sudah melakukan reservasi atau tidak melakukan reservasi sebelumnya		10s	Manual Task
4	Resepsionis	Menanyakan lama menginap	Resepsionis menanyakan berapa malam tamu akan menginap jika tamu tidak melakukan reservasi		10s	Manual Task
5	Tamu	Memberikan rencana lama menginap	Tamu memberitahukan berapa lama ia akan menginap		30s	Manual Task
6	Resepsionis	Menjelaskan tipe - tipe kamar dan harganya	Resepsionis menjelaskan tipe – tipe kamar yang dimiliki hotel		3m	Manual Task
7	Resepsionis	Cek tipe - tipe kamar tersedia	Resepsionis melakukan pengecekan tipe – tipe kamar yang tersedia dengan melakukan <i>input</i> tanggal yang diinginkan tamu kemudian sistem secara otomatis akan menampilkan daftar kamar yang tersedia	Daftar Reservasi, Daftar Tamu	30s	User Task
8	Resepsionis	Menawarkan tipe kamar yang tersedia	Resepsionis menawarkan tipe kamar yang tersedia kepada tamu setelah melakukan pengecekan tipe kamar		1m	Manual Task

Tabel 4.13 Deskripsi *Task* Proses Bisnis Registrasi *Check-In* Usulan (*To-Be*) (lanjutan)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
9	Tamu	Membatalkan permintaan <i>check-in</i>	Tamu dapat membatalkan permintaan <i>check-in</i> nya jika tipe kamar yang tersedia dirasa tidak ada yang cocok		10s	Manual Task
10	Tamu	Memilih kamar	Tamu memilih kamar jika tipe kamar yang tersedia dirasa cocok		10s	Manual Task
11	Resepsionis	Cek daftar reservasi	Resepsionis melakukan pengecekan daftar reservasi untuk mencari data reservasi tamu menggunakan sistem	Daftar Reservasi	10s	User Task
12	Resepsionis	Meminta kartu identitas	Resepsionis meminta kartu identitas tamu untuk ditahan sementara sebagai jaminan		10s	Manual Task
13	Tamu	Memberi kartu identitas	Tamu memberikan kartu identitasnya		10s	Manual Task
14	Resepsionis	Mengisi formulir registrasi	Resepsionis memasukkan data tamu seperti nomor identitas, nama, nomor telepon, tanggal serta lama menginap, dan pilihan kamar ke dalam formulir registrasi menggunakan sistem	Formulir Registrasi	1m	User Task
15	Resepsionis	Mengubah formulir reservasi menjadi formulir registrasi	Resepsionis mengubah reservasi menjadi formulir registrasi dengan melengkapi isian formulir menggunakan sistem		30s	User Task
16	Resepsionis	Mengecek pembayaran	Resepsionis memeriksa catatan reservasi apakah tamu sudah melakukan pembayaran	Formulir Reservasi	10s	User Task

Tabel 4.13 Deskripsi *Task* Proses Bisnis Registrasi *Check-In* Usulan (*To-Be*) (lanjutan)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
17	Resepsionis	Meminta konfirmasi pembayaran	Resepsionis meminta tamu melakukan pembayaran kamar		10s	Manual Task
18	Tamu	Melakukan pembayaran	Tamu melakukan pembayaran kamar		10s	Manual Task
19	Resepsionis	Memberikan kunci kamar dan <i>breakfast coupon</i>	Resepsionis memberikan kunci kamar dan <i>breakfast coupon</i> kepada tamu		10s	Manual Task
20	Tamu	Menerima kunci kamar dan <i>breakfast coupon</i>	Tamu menerima kunci kamar dan <i>breakfast couponnya</i>		10s	Manual Task



Berdasarkan dampak dari permasalahan yang ada pada proses bisnis registrasi *check-in*, yaitu proses menulis ulang formulir registrasi, dibuatlah sebuah sistem informasi yang mengganti proses penulisan ulang formulir registrasi, dan beberapa proses lainnya yang masih dilakukan secara manual (*manual task*) menjadi proses yang dilakukan dengan menggunakan sistem informasi (*user task*). Dengan perubahan ini, proses bisnis registrasi *check-in* mengalami peningkatan efisiensi waktu sebesar 39,36%. Detail waktu proses bisnis saat ini (*as-is*) dan proses bisnis usulan (*to-be*) ditunjukkan pada tabel 4.14.

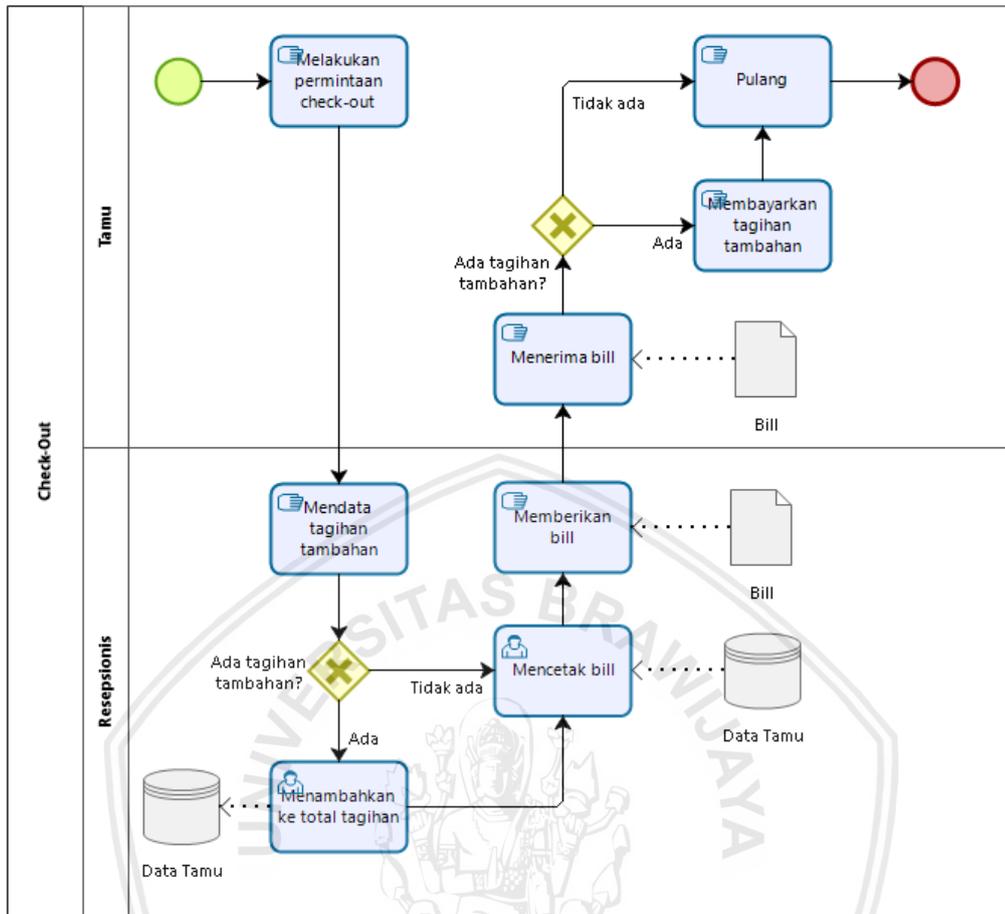
Tabel 4.14 Analisis Hasil Perbaikan Proses Bisnis Registrasi *Check-In*

No.	Proses Bisnis	Waktu <i>As-Is</i>	Waktu <i>To-Be</i>	Efisiensi Waktu
1	Registrasi <i>Check-In</i>	15 menit 40 detik	9 menit 30 detik	39,36%

Tabel 4.14 menjelaskan total waktu dari proses bisnis saat ini (*as-is*), proses bisnis usulan (*to-be*), dan persentase efisiensi waktu dari proses bisnis registrasi *check-in*. Total waktu pengerjaan proses bisnis saat ini (*as-is*) adalah 15 menit 40 detik, sedangkan total waktu pengerjaan proses bisnis usulan (*to-be*) adalah 9 menit 30 detik. Sehingga, proses bisnis registrasi *check-in* mendapat peningkatan efisiensi waktu sebesar 39,36%.

4.3.3 Proses Bisnis *Check-Out* Usulan (*To-Be*)

Proses bisnis ketiga adalah proses *check-out* usulan. Proses bisnis diawali dengan tamu melakukan permintaan *check-out*. Kemudian resepsionis akan mendata kemungkinan tagihan tambahan tamu. Jika ada tagihan tambahan, resepsionis menambahkan ke daftar tagihan di sistem. Jika tidak ada, resepsionis dapat langsung mencetak *bill*. Lalu resepsionis membuat *bill* kemudian memberikan *bill*nya ke tamu. Setelah menerima *bill*, tamu membayar tagihan tambahannya jika ada. Jika tidak ada, tamu bisa langsung pulang dan proses bisnis selesai. Proses bisnis ini dimodelkan dalam gambar 4.8 dan dijelaskan pada tabel 4.15.



Powered by **bizagi** Modeler

Gambar 4.8 Pemodelan Proses Bisnis *Check-Out* Usulan (*To-Be*)

Tabel 4.15 Deskripsi *Task* Proses Bisnis *Check-Out* Usulan (*To-Be*)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	Tamu	Melakukan permintaan <i>check-out</i>	Tamu mendatangi resepsionis untuk keperluan <i>check-out</i>		10s	<i>Manual Task</i>
2	Resepsionis	Mendata tagihan tambahan	Resepsionis mendata apakah tamu memiliki tagihan tambahan		3m	<i>Manual Task</i>
3	Resepsionis	Menambahkan ke total tagihan	Resepsionis menambahkan data tagihan ke total tagihan jika ada tagihan tambahan yang dimiliki tamu	Data Tamu	1m	<i>User Task</i>
4	Resepsionis	Mencetak <i>bill</i>	Resepsionis mencetak <i>bill</i> menggunakan sistem	Data Tamu	30s	<i>User Task</i>
5	Resepsionis	Memberikan <i>bill</i>	Resepsionis memberikan <i>bill</i> kepada tamu	<i>Bill</i>	10s	<i>Manual Task</i>
6	Tamu	Menerima <i>bill</i>	Tamu menerima <i>bill</i> yang diserahkan resepsionis	<i>Bill</i>	10s	<i>Manual Task</i>
7	Tamu	Membayarkan tagihan tambahan	Tamu membayarkan sisa tagihan tambahannya jika ada		10s	<i>Manual Task</i>
8	Tamu	Pulang	Tamu pulang		10s	<i>Manual Task</i>

Berdasarkan dampak dari permasalahan yang ada pada proses bisnis *check-out*, yaitu kehilangan *record* transaksi tamu hotel, dibuatlah sebuah sistem informasi yang mengganti proses penyimpanan transaksi tamu hotel dan beberapa proses lainnya yang masih dilakukan secara manual (*manual task*) menjadi proses yang dilakukan dengan menggunakan sistem informasi (*user task*). Dengan perubahan ini, proses bisnis *check-out* mengalami peningkatan efisiensi waktu sebesar 44,83%. Detail waktu proses bisnis saat ini (*as-is*) dan proses bisnis usulan (*to-be*) ditunjukkan pada tabel 4.16.

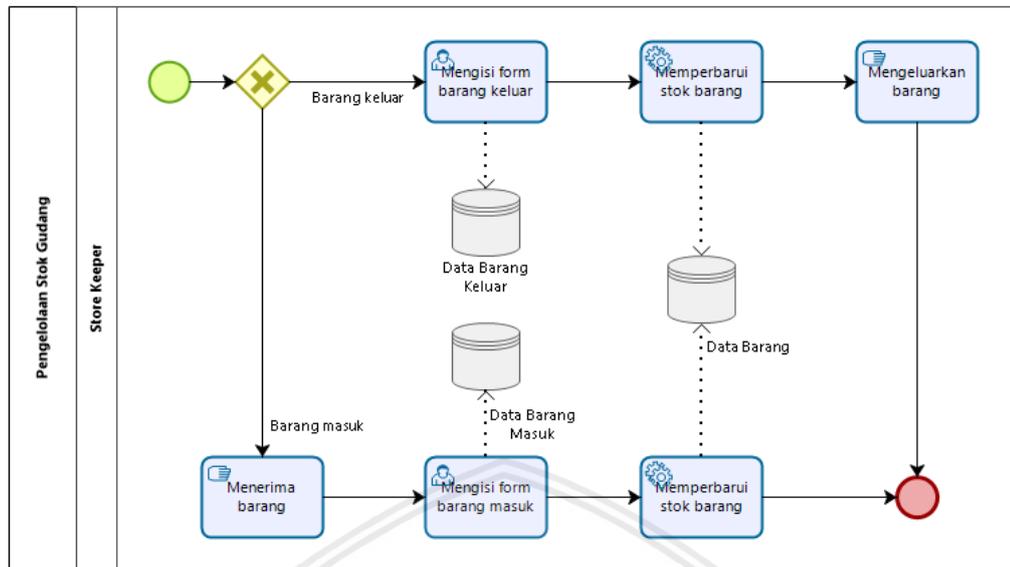
Tabel 4.16 Analisis Hasil Perbaikan Proses Bisnis *Check-Out*

No.	Proses Bisnis	Waktu <i>As-Is</i>	Waktu <i>To-Be</i>	Efisiensi Waktu
1	<i>Check-Out</i>	9 menit 40 detik	5 menit 20 detik	44,83%

Tabel 4.16 menjelaskan total waktu dari proses bisnis saat ini (*as-is*), proses bisnis usulan (*to-be*), dan persentase efisiensi waktu dari proses bisnis *check-out*. Total waktu pengerjaan proses bisnis saat ini (*as-is*) adalah 9 menit 40 detik, sedangkan total waktu pengerjaan proses bisnis usulan (*to-be*) adalah 5 menit 20 detik. Sehingga, proses bisnis *check-out* mendapat peningkatan efisiensi waktu sebesar 44,83%.

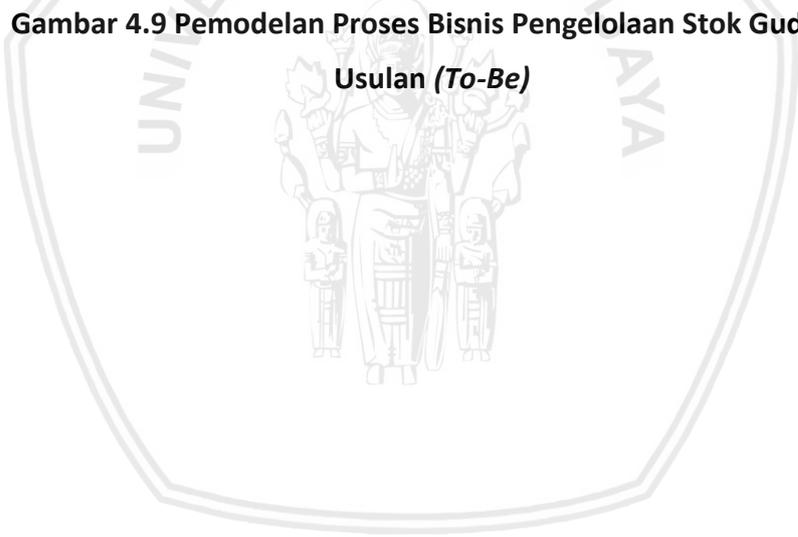
4.3.4 Proses Bisnis Pengelolaan Stok Gudang Usulan (*To-Be*)

Proses bisnis keempat adalah proses pengelolaan stok gudang usulan. Proses dimulai saat ada barang masuk atau keluar. Jika ada barang masuk, setelah *store keeper* menerima barang, maka *store keeper* mengisi formulir barang masuk. Kemudian sistem secara otomatis akan memperbarui stok. Lalu proses selesai. Jika ada permintaan barang keluar, maka *store keeper* mengisi formulir barang keluar. Setelah itu sistem akan secara otomatis memperbarui stok. Setelah itu barang dikeluarkan dan proses selesai. Proses bisnis ini dimodelkan dalam gambar 4.9 dan dijelaskan pada tabel 4.17.



Powered by
bizagi
Modeler

Gambar 4.9 Pemodelan Proses Bisnis Pengelolaan Stok Gudang Usulan (To-Be)



Tabel 4.17 Deskripsi *Task* Proses Bisnis Pengelolaan Stok Gudang Usulan (*To-Be*)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	<i>Store Keeper</i>	Menerima barang	<i>Store Keeper</i> menerima barang masuk		1m	<i>Manual Task</i>
2	<i>Store Keeper</i>	Mengisi formulir barang masuk	<i>Store Keeper</i> mengisi formulir data barang masuk mulai dari nama barang, jumlah barang, hingga foto nota	Data Barang Masuk	1m	<i>User Task</i>
3	<i>Store Keeper</i>	Memperbarui stok barang	Sistem secara otomatis akan menambah stok barang	Data Barang	5s	<i>Service Task</i>
4	<i>Store Keeper</i>	Mengisi formulir barang keluar	<i>Store Keeper</i> mengisi formulir data barang keluar mulai dari nama barang, jumlah barang, hingga keperluan barang keluar	Data Barang Keluar	1m	<i>User Task</i>
5	<i>Store Keeper</i>	Memperbarui stok barang	Sistem secara otomatis akan mengurangi stok barang	Data Barang	5s	<i>Service Task</i>
6	<i>Store Keeper</i>	Mengeluarkan barang	<i>Store Keeper</i> mengeluarkan barang dari gudang		1m	<i>Manual Task</i>

Berdasarkan dampak dari permasalahan yang ada pada proses bisnis pengelolaan stok gudang, yaitu tidak ada catatan stok barang, dibuatlah sebuah sistem informasi yang mengganti pengelolaan stok barang dan beberapa proses lainnya yang masih dilakukan secara manual (*manual task*) menjadi proses yang dilakukan dengan menggunakan sistem informasi (*user task*). Dengan perubahan ini, proses bisnis pengelolaan stok gudang mengalami peningkatan efisiensi waktu sebesar 59,02%. Detail waktu proses bisnis saat ini (*as-is*) dan proses bisnis usulan (*to-be*) ditunjukkan pada tabel 4.18.

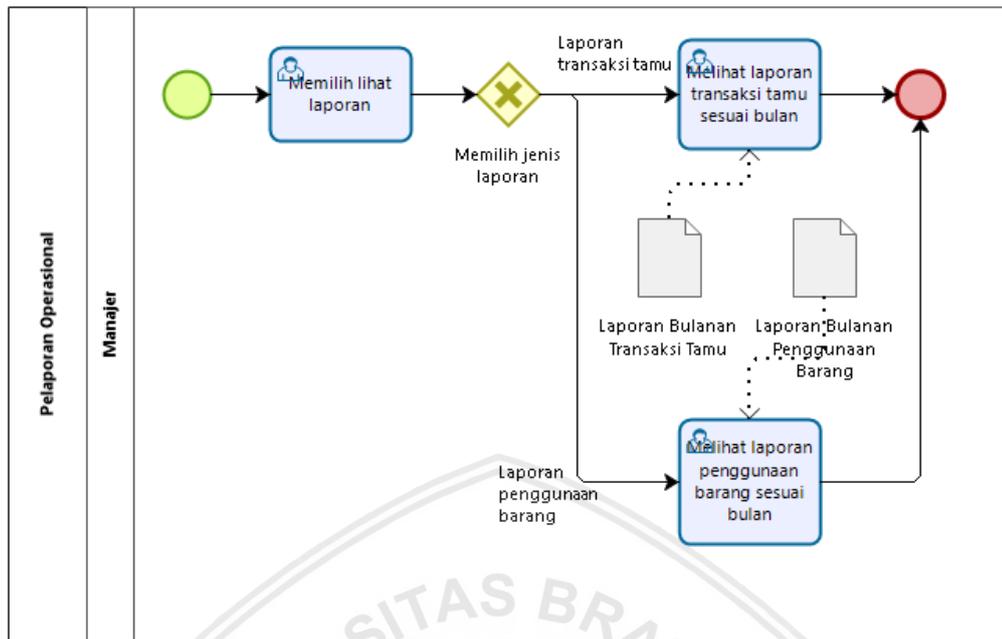
Tabel 4.18 Analisis Hasil Perbaikan Proses Bisnis Pengelolaan Stok Gudang

No.	Proses Bisnis	Waktu <i>As-Is</i>	Waktu <i>To-Be</i>	Efisiensi Waktu
1	Pengelolaan Stok Gudang	10 menit 10 detik	4 menit 10 detik	59,02%

Tabel 4.18 menjelaskan total waktu dari proses bisnis saat ini (*as-is*), proses bisnis usulan (*to-be*), dan persentase efisiensi waktu dari proses bisnis pengelolaan stok gudang. Total waktu pengerjaan proses bisnis saat ini (*as-is*) adalah 10 menit 10 detik, sedangkan total waktu pengerjaan proses bisnis usulan (*to-be*) adalah 4 menit 10 detik. Sehingga, proses bisnis pengelolaan stok gudang mendapat peningkatan efisiensi waktu sebesar 59,02%.

4.3.5 Proses Bisnis Pelaporan Operasional Usulan (*To-Be*)

Proses bisnis kelima adalah proses pelaporan operasional usulan. Proses dimulai dengan manajer memilih menu lihat laporan pada sistem. Kemudian manajer memilih jenis laporan yang ingin dilihat. Manajer dapat melihat laporan bulanan transaksi dan laporan bulanan penggunaan barang langsung melalui sistem. setelah melihat laporannya, proses bisnis selesai. Proses bisnis ini dimodelkan dalam gambar 4.10 dan dijelaskan pada tabel 4.19.



Powered by
bizagi
Modeler

Gambar 4.10 Pemodelan Proses Bisnis Pelaporan Operasional Usulan (To-Be)



Tabel 4.19 Deskripsi *Task* Proses Bisnis Pelaporan Operasional Usulan (*To-Be*)

No.	Aktor	Task	Deskripsi	Dokumen	Waktu	Tipe
1	Manajer	Memilih lihat laporan	Manajer memilih menu lihat laporan bulanan yang disediakan sistem		30s	User Task
2	Manajer	Melihat laporan transaksi tamu sesuai bulan	Manajer memilih melihat laporan bulanan transaksi tamu	Laporan Bulanan Transaksi Tamu	5m	User Task
3	Manajer	Melihat laporan penggunaan barang sesuai bulan	Manajer memilih melihat laporan bulanan penggunaan barang	Laporan Bulanan Penggunaan Barang	5m	User Task



Berdasarkan dampak dari permasalahan yang ada pada proses bisnis pelaporan operasional, yaitu manajer tidak dapat meminta laporan sewaktu – waktu, dibuatlah sebuah sistem informasi yang mengganti pelaporan operasional dan beberapa proses lainnya yang masih dilakukan secara manual (*manual task*) menjadi proses yang dilakukan dengan menggunakan sistem informasi (*user task*). Dengan perubahan ini, proses bisnis pelaporan operasional mengalami peningkatan efisiensi waktu sebesar 97,81%. Detail waktu proses bisnis saat ini (*as-is*) dan proses bisnis usulan (*to-be*) ditunjukkan pada tabel 4.20.

Tabel 4.20 Analisis Hasil Perbaikan Proses Bisnis Pelaporan Operasional

No.	Proses Bisnis	Waktu <i>As-Is</i>	Waktu <i>To-Be</i>	Efisiensi Waktu
1	Pelaporan Operasional	8 jam 30 detik	10 menit 30 detik	97,81%

Tabel 4.20 menjelaskan total waktu dari proses bisnis saat ini (*as-is*), proses bisnis usulan (*to-be*), dan persentase efisiensi waktu dari proses bisnis pelaporan operasional. Total waktu pengerjaan proses bisnis saat ini (*as-is*) adalah 8 jam 30 detik, sedangkan total waktu pengerjaan proses bisnis usulan (*to-be*) adalah 10 menit 30 detik. Sehingga, proses bisnis pengelolaan stok gudang mendapat peningkatan efisiensi waktu sebesar 97,81%.

4.4 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan yang pertama kali dilakukan dari fase *elaboration*. Pada analisis kebutuhan dilakukan identifikasi aktor, identifikasi fitur, kebutuhan fungsional, kebutuhan non-fungsional, *use case diagram*, dan *use case scenario*. Pada tahap ini terjadi iterasi yang dilakukan untuk menyamakan persepsi kebutuhan pengguna dengan pengembang. Penggalan kebutuhan dilakukan berdasarkan hasil wawancara, observasi, evaluasi proses bisnis saat ini (*as-is*), dan proses bisnis usulan (*to-be*).

4.4.1 Identifikasi Aktor

Identifikasi aktor adalah kegiatan mendeskripsikan peran yang dilakukan pengguna ketika melakukan interaksi dengan sistem. Identifikasi aktor dibuat berdasarkan analisis proses bisnis saat ini (*as-is*) dan analisis proses bisnis usulan (*to-be*) yang telah dilakukan. Identifikasi aktor pada sistem informasi perhotelan Edotel Pamekasan dideskripsikan pada tabel 4.21.

Tabel 4.21 Identifikasi Aktor

No.	Kode Aktor	Nama Aktor	Deskripsi Aktor
1	AKSI-01	Resepsionis	Resepsionis merupakan orang yang memiliki tanggung jawab untuk berinteraksi dengan tamu

Tabel 4.21 Identifikasi Aktor (lanjutan)

No.	Kode Aktor	Nama Aktor	Deskripsi Aktor
			untuk keperluan reservasi, registrasi <i>check-in</i> , maupun <i>check-out</i>
2	AKSI-02	<i>Accounting</i>	<i>Accounting</i> pada Edotel Pamekasan bertanggung jawab mengelola keuangan merangkap pembelian barang. Aktor ini tidak dilibatkan dalam pengembangan sistem saat ini
3	AKSI-03	<i>Store Keeper</i>	<i>Store Keeper</i> adalah aktor usulan yang bertanggung jawab menggantikan <i>accounting</i> dalam mengelola stok barang di gudang
4	AKSI 04	Manajer	Manajer merupakan orang yang bertanggung jawab memantau seluruh kegiatan operasional hotel
5	AKSI 05	Pengguna	Pengguna merupakan aktor yang belum melewati autentikasi sistem

Tabel 4.21 mendeskripsikan aktor dan interaksinya dengan sistem informasi. Terdapat 5 aktor yang diidentifikasi, namun hanya terdapat 4 aktor yang akan berinteraksi dengan sistem informasi. Pertama adalah aktor resepsionis dengan kode AK-01, bertanggung jawab terhadap kegiatan reservasi, *check-in*, hingga *check-out* tamu hotel. Kedua adalah aktor *store keeper* dengan kode AK-03, bertanggung jawab mengelola stok barang di gudang. Ketiga adalah aktor manajer dengan kode AK-04, bertanggung jawab terhadap seluruh kegiatan di operasional di sistem. Aktor terakhir adalah aktor pengguna dengan kode AK-05, merupakan generalisasi dari ketiga aktor sebelum masuk ke dalam sistem.

4.4.2 Identifikasi Fitur

Identifikasi fitur yang dilakukan dalam penelitian ini merupakan tahap menentukan fitur yang dibutuhkan berdasarkan solusi pada evaluasi proses bisnis saat ini (*as-is*). Hasil dari identifikasi fitur digunakan sebagai dasar kebutuhan fungsional dan non-fungsional sistem informasi yang akan dikembangkan. Identifikasi fitur sistem informasi perhotelan Edotel Pamekasan dijelaskan pada tabel 4.22.

Tabel 4.22 Identifikasi Fitur

No.	Kode Fitur	Solusi pada Evaluasi Proses Bisnis <i>As-Is</i>	Deskripsi
1	FTSI-01	A. Menyediakan sistem informasi yang memfasilitasi proses reservasi kamar hotel	Sistem dapat mencatat, menyimpan, dan mengelola reservasi kamar tamu hotel

Tabel 4.22 Identifikasi Fitur (lanjutan)

No.	Kode Fitur	Solusi pada Evaluasi Proses Bisnis As-Is	Deskripsi
2	FTSI-02	<p>A. Menyediakan sistem informasi yang memfasilitasi proses registrasi</p> <p>B. Menyediakan sistem yang mampu memindahkan formulir reservasi ke dalam formulir registrasi sehingga tidak perlu menulis ulang isi yang ada di formulir reservasi</p> <p>C. Menyediakan sistem informasi yang dapat mencatat transaksi tamu hotel</p> <p>D. Menyediakan sistem yang mampu menghitung total tagihan tamu hotel sekaligus mencetak <i>bill</i></p>	<p>Sistem dapat mencatat, menyimpan, dan mengelola transaksi tamu mulai dari <i>check-in</i> hingga <i>check-out</i> kamar tamu hotel, termasuk mengelola tagihan tamu dan mencetak <i>bill</i></p>
3	FTSI-03	<p>A. Menyediakan sistem yang dapat menunjukkan kamar – kamar yang tersedia di tanggal tertentu, sehingga kamar yang sudah dipesan tidak akan ditawarkan ke tamu</p>	<p>Sistem dapat menampilkan daftar kamar yang dapat dipesan berdasarkan masukan tanggal</p>
4	FTSI-04	<p>A. Menyediakan sistem informasi yang dapat mencatat barang masuk dan barang keluar dari gudang sehingga stok barang dapat dipantau</p>	<p>Sistem dapat mencatat, menyimpan, dan mengelola data barang masuk gudang</p>
5	FTSI-05	<p>A. Menyediakan sistem informasi yang dapat mencatat barang masuk dan barang keluar dari gudang sehingga stok barang dapat dipantau</p>	<p>Sistem dapat mencatat, menyimpan, dan mengelola data barang keluar gudang</p>
6	FTSI-06	-	<p>Sistem dapat mengetahui pengguna dan membatasi akses pengguna sesuai tanggung jawab pengguna</p>
7	FTSI-07	<p>A. Menyediakan sistem informasi yang dapat menampilkan laporan bulanan transaksi secara</p>	<p>Sistem dapat menampilkan laporan bulanan transaksi tamu hotel, dan penggunaan barang</p>



Tabel 4.22 Identifikasi Fitur (lanjutan)

No.	Kode Fitur	Solusi pada Evaluasi Proses Bisnis As-Is	Deskripsi
		<p>langsung sehingga dapat diminta oleh manajer sewaktu – waktu</p> <p>B. Menyediakan sistem informasi yang dapat melakukan pelaporan penggunaan barang</p>	

Tabel 4.22 mendeskripsikan fitur yang harus dimiliki sistem, kaitannya dengan evaluasi proses bisnis, dan nomor kodenya. Terdapat 7 fitur yang dikembangkan. Fitur pertama dengan kode FTSI-01 merupakan fitur sistem dapat mengelola reservasi tamu hotel. Fitur kedua dengan kode FTSI-02 merupakan fitur sistem dapat mengelola transaksi tamu mulai dari *check-in* hingga *check-out* tamu hotel, dan sistem juga dapat mengelola tagihan tamu. Fitur ketiga dengan kode FTSI-03 merupakan fitur sistem dapat menampilkan daftar kamar yang dapat dipesan berdasarkan masukan tanggal. Fitur keempat dengan kode FTSI-04 merupakan fitur sistem dapat mengelola data barang masuk gudang. Fitur kelima dengan kode FTSI-05 merupakan fitur sistem dapat mengelola data barang keluar gudang. Fitur keenam dengan kode FTSI-06 merupakan fitur sistem dapat mengetahui pengguna dan membatasi akses pengguna sesuai tanggung jawab pengguna. Fitur ketujuh dengan kode FTSI-07 merupakan fitur sistem dapat menampilkan laporan bulanan transaksi tamu hotel, dan penggunaan barang.

4.4.3 Kebutuhan Fungsional Sistem

Kebutuhan fungsional sistem adalah gambaran terkait apa yang seharusnya dilakukan oleh sistem. Kebutuhan fungsional secara spesifik mendeskripsikan fungsi sistem, *input*, dan *output*-nya (Sommerville, 2011). Kebutuhan fungsional pada penelitian ini dibuat berdasarkan identifikasi fitur. Kebutuhan fungsional akan dideskripsikan pada tabel 4.23.

Tabel 4.23 Kebutuhan Fungsional Sistem

No.	Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
1	FTSI-01	FGSSI-01	Sistem dapat menampilkan data reservasi
2		FGSSI-02	Sistem dapat menampilkan formulir reservasi
3		FGSSI-03	Sistem dapat menyimpan data reservasi

Tabel 4.23 Kebutuhan Fungsional Sistem (lanjutan)

No.	Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
4		FGSSI-05	Sistem dapat mengganti status belum konfirmasi reservasi jadi sudah konfirmasi kedatangan
5		FGSSI-06	Sistem dapat mengganti status belum konfirmasi reservasi jadi batal reservasi
6		FGSSI-32	Sistem dapat mengganti status sudah konfirmasi menjadi sudah <i>check-in</i> (ditambahkan pada iterasi ke-1)
7	FTSI-02	FGSSI-07	Sistem dapat menampilkan data tamu yang sudah <i>check-in</i>
8		FGSSI-08	Sistem dapat menampilkan formulir data registrasi <i>check-in</i>
9		FGSSI-09	Sistem dapat menyimpan data registrasi <i>check-in</i>
10		FGSSI-10	Sistem dapat menampilkan data detail tamu
11		FGSSI-12	Sistem dapat menyimpan tagihan tamu
12		FGSSI-33	Sistem dapat menghapus tagihan tamu (ditambahkan pada iterasi ke-1 mengganti FGSSI-13)
13		FGSSI-14	Sistem dapat mengganti status tamu menjadi <i>check-out</i>
14		FGSSI-15	Sistem dapat mencetak <i>bill</i>
15	FTSI-03	FGSSI-16	Sistem dapat menampilkan data kamar yang tersedia untuk disewa tamu hotel berdasarkan masukan tanggal <i>check-in</i> dan tanggal <i>check-out</i>
16	FTSI-04	FGSSI-17	Sistem dapat menampilkan data barang masuk
17		FGSSI-18	Sistem dapat menampilkan formulir barang masuk
18		FGSSI-19	Sistem dapat menyimpan data barang masuk
19		FGSSI-20	Sistem dapat mengubah data barang masuk
20		FGSSI-34	Sistem dapat menghapus data barang masuk (ditambahkan pada iterasi ke-2)

Tabel 4.23 Kebutuhan Fungsional Sistem (lanjutan)

No.	Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
21		FGSSI-21	Sistem dapat menampilkan nota barang masuk
22	FTSI-05	FGSSI-22	Sistem dapat menampilkan data barang keluar
23		FGSSI-23	Sistem dapat menampilkan formulir barang keluar
24		FGSSI-24	Sistem dapat menyimpan data barang keluar
25		FGSSI-25	Sistem dapat mengubah data barang keluar
26		FGSSI-35	Sistem dapat menghapus data barang keluar (ditambahkan pada iterasi ke-2)
27	FTSI-06	FGSSI-26	Sistem dapat menampilkan halaman <i>login</i>
28		FGSSI-27	Sistem dapat mengautentikasi pengguna
29	FTSI-07	FGSSI-28	Sistem dapat menampilkan laporan bulanan transaksi tamu hotel
30		FGSSI-29	Sistem dapat menampilkan laporan bulanan penggunaan barang
31		FGSSI-30	Sistem dapat mencetak laporan bulanan transaksi tamu hotel
32		FGSSI-31	Sistem dapat mencetak laporan bulanan penggunaan barang

Tabel 4.23 menunjukkan bahwa terdapat 31 kebutuhan fungsional pada iterasi ke-0. Kemudian, pada iterasi ke-1 dalam fase *elaboration* terjadi perubahan kebutuhan fungsional dengan menambah 2 kebutuhan fungsional, dan menghapus 3 kebutuhan fungsional. Lalu, pada iterasi ke-2 dalam fase *elaboration* terjadi penambahan kebutuhan fungsional sejumlah 2 kebutuhan fungsional, yaitu fungsional sistem dapat menghapus catatan data barang masuk, dan barang keluar. Iterasi ke-2 dilakukan saat peneliti merasa ada fungsional yang kurang ketika dalam fase *construction*.

Iterasi, perubahan, dan penghapusan kebutuhan dilakukan karena terdapat kebutuhan yang dirasa kurang perlu untuk dikembangkan sekarang dan kebutuhan yang perlu diganti. Kebutuhan fungsional yang dihapus ditunjukkan pada tabel 4.24.

Tabel 4.24 Kebutuhan Fungsional yang Dihapus

No	Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
1	FTSI-01	FGSSI-04	Sistem dapat mengubah data reservasi (dihapus pada iterasi ke-1)
2	FTSI-02	FGSSI-11	Sistem dapat mengubah data detail tamu (dihapus pada iterasi ke-1)
3		FGSSI-13	Sistem dapat mengubah tagihan tamu (dihapus pada iterasi ke-1)

Tabel 4.24 menunjukkan kebutuhan fungsional yang dihapus. Terdapat 3 kebutuhan fungsional yang dihapus, yakni kebutuhan mengubah data reservasi, kebutuhan mengubah data detail tamu, dan kebutuhan mengubah tagihan tamu. Kebutuhan fungsional dihapus karena dirasa tidak dibutuhkan oleh pengguna sistem. Sehingga total akhir kebutuhan fungsional berjumlah 32 kebutuhan.

4.4.4 Kebutuhan Non-Fungsional Sistem

Kebutuhan non-fungsional sistem adalah kebutuhan yang tidak berkaitan secara langsung dengan layanan sistem kepada pengguna, contohnya seperti waktu respon sistem. Atau, dapat didefinisikan sebagai batasan implementasi sistem (Sommerville, 2011). Kebutuhan non-fungsional sistem dideskripsikan pada tabel 4.25.

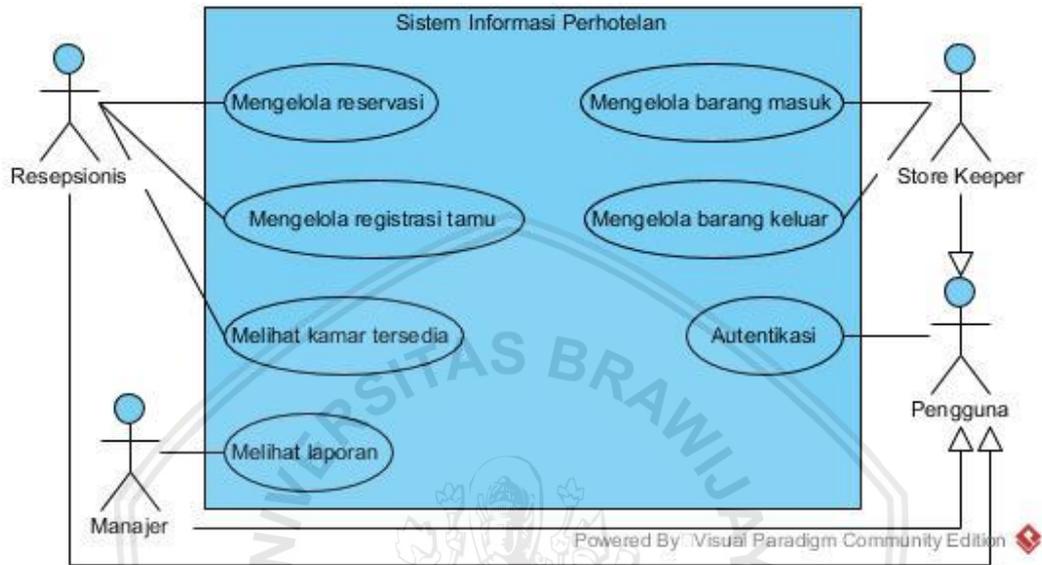
Tabel 4.25 Kebutuhan Non-Fungsional Sistem

No.	Kode Fitur	Kode Kebutuhan Non-Fungsional	Deskripsi
1	FTSI-01	NFGSSI-01	Sistem dapat dijalankan di berbagai peramban (<i>browser</i>) tanpa kendala besar (<i>Compatibility</i>)
	FTSI-02		
	FTSI-03		
	FTSI-04		
	FTSI-05		
	FTSI-06		
	FTSI-07		

Tabel 4.25 menunjukkan kebutuhan non-fungsional sistem. Terdapat satu buah kebutuhan non-fungsional, yaitu kebutuhan *compatibility*, dimana sistem dapat dijalankan diberbagai *browser* seperti Google Chrome atau Mozilla Firefox tanpa ada masalah besar.

4.4.5 Use Case Diagram

Use case merupakan teknik penggalan kebutuhan yang fundamental dalam pemodelan UML. Secara sederhana, *use case* mengidentifikasi aktor yang terlibat dan nama interaksi dalam sistemnya. *Use cases* kemudian didokumentasikan dalam *use case diagram* (Sommerville, 2011). *Use case diagram* sistem informasi perhotelan ditunjukkan dalam gambar 4.11.



Gambar 4.11 Use Case Diagram

Use case diagram yang telah dibuat kemudian dihubungkan dengan fitur solusi yang ditawarkan. Mengubungkan *use case* dengan fitur memiliki tujuan untuk menunjukkan dan mempertegas pemodelan *use case* telah dibuat sesuai analisis kebutuhan. Hubungan *use case* dengan fitur ditunjukkan pada tabel 4.26.

Tabel 4.26 Hubungan Use Case dengan Fitur

No.	Kode Use Case	Use Case	Nomor Fitur
1	UCSI-01	Mengelola reservasi	FTSI-01
2	UCSI-02	Mengelola registrasi tamu	FTSI-02
3	UCSI-03	Melihat kamar tersedia	FTSI-03
4	UCSI-04	Mengelola barang masuk	FTSI-04
5	UCSI-05	Mengelola barang keluar	FTSI-05
6	UCSI-06	Melihat laporan	FTSI-07
7	UCSI-07	Autentikasi	FTSI-06

Tabel 4.26 memetakan hubungan antara *use case* dengan fitur sistem informasi. Pertama, *use case* mengelola reservasi dengan kode UCSI-01 berkaitan dengan fitur FTSI-01, yaitu fitur sistem dapat mengelola reservasi tamu hotel.

Kedua, *use case* mengelola registrasi tamu dengan kode UCSI-02 berkaitan dengan fitur FTSI-02, yakni fitur sistem dapat mengelola transaksi tamu mulai dari *check-in* hingga *check-out* tamu hotel, dan sistem juga dapat mengelola tagihan tamu. Ketiga, *use case* melihat kamar tersedia dengan kode UCSI-03 berkaitan dengan fitur FTSI-03, yaitu fitur sistem dapat menampilkan daftar kamar yang dapat dipesan berdasarkan masukan tanggal. Keempat, *use case* mengelola barang masuk dengan kode UCSI-04 berkaitan dengan fitur FTSI-04 yang merupakan fitur sistem dapat mengelola data barang masuk gudang. Kelima, *use case* mengelola barang keluar dengan kode UCSI-05 berkaitan dengan fitur FTSI-05 yang merupakan fitur sistem dapat mengelola data barang keluar gudang. Keenam, *use case* melihat laporan dengan kode UCSI-06 berkaitan dengan fitur FTSI-07 yang merupakan fitur sistem dapat menampilkan laporan bulanan transaksi tamu hotel dan penggunaan barang. Terakhir, *use case* autentikasi dengan kode UCSI-07 berkaitan dengan fitur FTSI-06, yakni fitur sistem dapat mengetahui pengguna dan membatasi akses pengguna sesuai tanggung jawab pengguna.

4.4.6 Use Case Scenario

Use case description yang pada penelitian ini disebut sebagai *use case scenario* merupakan penjelasan mendetail mengenai bagaimana aktor dan perilakunya dalam sebuah *use case* (Bittner dan Spence, 2002). Dalam *use case scenario* terdapat nama *use case*, nama aktor, deskripsi tujuan *use case*, kondisi prasyarat yang harus terpenuhi sebelum menjalankan *use case*, alur utama *use case* dalam keadaan normal, alur alternatif ketika alur tidak berjalan sesuai alur utama, dan *sub flow* yang merupakan percabangan dari alur utama.

Tabel 4.27 Use Case Scenario Mengelola Reservasi

Use Case Code	UCSI-01
Use Case Name	Mengelola reservasi
Actor	Resepsionis
Description	<i>Use case</i> mengelola reservasi kamar hotel menjelaskan bagaimana aktor resepsionis menggunakan sistem untuk menambah data reservasi, melihat data reservasi, melakukan konfirmasi kedatangan tamu, melakukan pembatalan reservasi, dan melakukan <i>check-in</i> setelah konfirmasi reservasi
Pre-conditions	Aktor berada pada halaman reservasi
Main Flow	<ol style="list-style-type: none"> Aktor memilih tambah reservasi {Sistem menampilkan formulir reservasi} Aktor mengisi data formulir reservasi yang terdiri dari nama, nomor telpon yang dapat dihubungi, catatan tambahan (jika ada), tanggal <i>check-in</i>, tanggal <i>check-out</i>, dan pilihan nomor kamar

Tabel 4.27 Use Case Scenario Mengelola Reservasi (lanjutan)

	<p>3. Aktor memilih simpan reservasi {Sistem menyimpan data reservasi tamu} {Sistem menyimpan data reservasi kamar} {Sistem menampilkan kembali halaman reservasi}</p> <p>4. Aktor menjalankan <i>sub flow</i> Mengganti Status Belum Konfirmasi Kedatangan Menjadi Sudah Konfirmasi Kedatangan</p> <p>5. Aktor menjalankan <i>sub flow</i> Mengganti Status Belum Konfirmasi Kedatangan Menjadi Batal Reservasi</p> <p>6. Aktor menjalankan <i>sub flow</i> Mengganti Status Sudah Konfirmasi Kedatangan Menjadi <i>Check-In</i></p> <p>7. <i>Use case</i> selesai</p>
Alternative Flow	<p>A1. Formulir isian kosong</p> <p>Jika formulir isian dengan <i>mandatory</i> tanda * dibiarkan kosong, maka saat {Sistem menyimpan reservasi}, sistem akan menampilkan pesan error dan memberi perintah untuk wajib mengisi form, kemudian sistem akan kembali ke {Sistem menampilkan formulir reservasi}</p>
Sub Flow	<p>S1. Mengganti Status Belum Konfirmasi Kedatangan Menjadi Sudah Konfirmasi Kedatangan</p> <ol style="list-style-type: none"> 1. Aktor memilih data reservasi dengan status belum konfirmasi yang ingin diganti statusnya 2. Aktor memilih konfirmasi kedatangan {Sistem menampilkan alert untuk meyakinkan aktor} 3. Aktor memilih ya {Sistem menyimpan perubahan status} {Sistem menampilkan kembali halaman reservasi} 4. <i>Use case</i> selesai <p>S2. Mengganti Status Belum Konfirmasi Kedatangan Menjadi Batal Reservasi</p> <ol style="list-style-type: none"> 1. Aktor memilih data reservasi dengan status belum konfirmasi yang ingin diganti statusnya 2. Aktor memilih batal reservasi {Sistem menampilkan alert untuk meyakinkan aktor}



Tabel 4.27 *Use Case Scenario* Mengelola Reservasi (lanjutan)

	<p>3. Aktor memilih ya {Sistem menyimpan perubahan status} {Sistem menampilkan kembali halaman reservasi}</p> <p>4. <i>Use case</i> selesai</p> <p>S3. Mengganti Status Sudah Konfirmasi Kedatangan Menjadi <i>Check-In</i></p> <p>1. Aktor memilih salah satu data reservasi dengan status sudah konfirmasi yang ingin diganti statusnya</p> <p>2. Aktor memilih <i>check-in</i> tamu hotel {Sistem menampilkan halaman formulir registrasi <i>check-in</i>}</p> <p>3. Aktor melengkapi formulir registrasi <i>check-in</i></p> <p>4. Aktor memilih simpan registrasi <i>check-in</i> {Sistem menyimpan registrasi <i>check-in</i>} {Sistem menyimpan perubahan status} {Sistem menampilkan halaman <i>guest list</i>}</p> <p>5. <i>Use case</i> selesai</p>
Post-conditions	Aktor berhasil membuat reservasi kamar

Tabel 4.27 menunjukkan *use case scenario* mengelola reservasi. *Use case scenario* mengelola reservasi menjelaskan tujuan aktor dan alur kegiatan yang dilakukan oleh aktor resepsionis saat menggunakan sistem. *Use case scenario* mengelola reservasi memiliki *main flow* menambah reservasi dengan *sub flow* mengganti status reservasi menjadi sudah *check-in*, sudah konfirmasi, dan batal reservasi. *Use case scenario* mengelola reservasi memiliki *alternate flow*, yaitu formulir isian kosong yang akan menampilkan pesan *error* saat terdapat formulir isian yang belum diisi.

Tabel 4.28 *Use Case Scenario* Mengelola Registrasi Tamu

Use Case Code	UCSI-02
Use Case Name	Mengelola registrasi tamu
Actor	Resepsionis
Description	<i>Use case</i> mengelola registrasi tamu menjelaskan bagaimana aktor resepsionis menggunakan sistem untuk menambah data registrasi <i>check-in</i> , melihat data tamu yang sudah <i>check-in</i> , melihat data detail tamu, menambah tagihan tamu,

Tabel 4.28 Use Case Scenario Mengelola Registrasi Tamu (lanjutan)

	menghapus tagihan tamu, mengganti status tamu menjadi <i>check-out</i> , dan mencetak <i>bill</i>
Pre-conditions	Aktor berada pada halaman reservasi
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih tambah registrasi <i>check-in</i> {Sistem menampilkan formulir registrasi} 2. Aktor mengisi data formulir registrasi terdiri dari nomor identitas, nama, alamat, nomor telpon yang dapat dihubungi, catatan tambahan (jika ada), tanggal <i>check-in</i>, tanggal <i>check-out</i>, dan memilih kamar 3. Aktor memilih simpan registrasi <i>check-in</i> {Sistem menyimpan registrasi <i>check-in</i>} {Sistem menampilkan halaman <i>guest list</i>} 4. Aktor menjalankan <i>sub flow</i> Menampilkan Data Detail Tamu 5. Aktor menjalankan <i>sub flow</i> Menambah Tagihan Tamu 6. Aktor menjalankan <i>sub flow</i> Menghapus Tagihan Tamu 7. Aktor menjalankan <i>sub flow</i> Mengganti Status Tamu Menjadi Check-Out 8. Aktor menjalankan <i>sub flow</i> Mencetak <i>Bill</i> 9. <i>Use case</i> selesai
Alternative Flow	<p>A1. Formulir isian kosong</p> <p>Jika formulir isian dengan <i>mandatory</i> tanda * dibiarkan kosong, maka saat {Sistem menyimpan registrasi <i>check-in</i>}, sistem akan menampilkan pesan <i>error</i> dan memberi perintah untuk wajib mengisi form, kemudian sistem akan kembali ke {Sistem menampilkan formulir registrasi}</p> <p>A2. Formulir isian kosong pada <i>sub flow</i> Menambah Tagihan Tamu</p> <p>Jika formulir isian dibiarkan kosong, maka saat {Sistem menyimpan tagihan}, sistem akan menampilkan pesan <i>error</i> dan memberi perintah bahwa formulir harus sepenuhnya diisi, kemudian sistem akan kembali ke {Sistem menampilkan halaman detail tamu}</p>
Sub Flow	<p>S1. Menampilkan Data Detail Tamu</p> <ol style="list-style-type: none"> 1. Aktor memilih menu <i>guest list</i>

Tabel 4.28 *Use Case Scenario* Mengelola Registrasi Tamu (lanjutan)

	<p>{Sistem menampilkan halaman <i>guest list</i> / daftar tamu}</p> <ol style="list-style-type: none"> 2. Aktor melihat halaman <i>guest list</i> 3. Aktor memilih lihat detail tamu. <p>{Sistem menampilkan halaman detail tamu}</p> <ol style="list-style-type: none"> 4. Aktor melihat data detail tamu 5. <i>Use case</i> selesai <p>S2. Menambah Tagihan Tamu</p> <ol style="list-style-type: none"> 1. Aktor menjalankan <i>sub flow</i> Menampilkan Data Detail Tamu 2. Aktor mengisi data formulir tagihan yang terdiri dari nomor kamar, nama tagihan, besaran tagihan, dan jenis tagihan 3. Aktor memilih simpan tagihan <p>{Sistem menyimpan tambahan tagihan}</p> <p>{Sistem menampilkan detail tamu}</p> <ol style="list-style-type: none"> 4. <i>Use case</i> selesai <p>S3. Menghapus Tagihan Tamu</p> <ol style="list-style-type: none"> 1. Aktor menjalankan <i>sub flow</i> Menampilkan Data Detail Tamu 2. Aktor memilih tagihan yang ingin dihapus 3. Aktor memilih hapus <p>{Sistem menampilkan <i>alert</i> untuk meyakinkan aktor}</p> <ol style="list-style-type: none"> 4. Aktor memilih ya <p>{Sistem menghapus tagihan yang dipilih}</p> <p>{Sistem menampilkan detail tamu}</p> <ol style="list-style-type: none"> 5. <i>Use case</i> selesai <p>S4. Mengganti Status Tamu Menjadi <i>Check-Out</i></p> <ol style="list-style-type: none"> 1. Aktor menjalankan <i>sub flow</i> Menampilkan Data Detail Tamu 2. Aktor memilih <i>check-out</i> <p>{Sistem menampilkan <i>alert</i> untuk meyakinkan aktor}</p>
--	---

Tabel 4.28 *Use Case Scenario* Mengelola Registrasi Tamu (lanjutan)

	<ol style="list-style-type: none"> 3. Aktor memilih ya {Sistem menyimpan perubahan status tamu} {Sistem menampilkan detail tamu} 4. <i>Use case</i> selesai <p>S5. Mencetak <i>Bill</i></p> <ol style="list-style-type: none"> 1. Aktor menjalankan <i>sub flow</i> Menampilkan Data Detail Tamu 2. Aktor memilih menu cetak <i>bill</i> {Sistem menampilkan <i>print preview</i> nota tamu} 3. Aktor mencetak <i>bill</i> 4. <i>Use case</i> selesai
Post-conditions	Aktor berhasil membuat registrasi <i>check-in</i> tamu

Tabel 4.28 menunjukkan *use case scenario* mengelola registrasi tamu. *Use case scenario* mengelola registrasi tamu menjelaskan tujuan aktor dan alur kegiatan yang dilakukan oleh aktor resepsionis saat menggunakan sistem. *Use case scenario* mengelola registrasi tamu memiliki *main flow* menambah mengelola registrasi *check-in* dengan *sub flow* mengelola tagihan, melakukan *check-out*, dan cetak *bill*. *Use case scenario* mengelola registrasi tamu memiliki *alternate flow*, yaitu formulir isian kosong yang akan menampilkan pesan *error* saat terdapat formulir isian yang belum diisi.

Tabel 4.29 *Use Case Scenario* Melihat Kamar Tersedia

Use Case Code	UCSI-03
Use Case Name	Melihat kamar tersedia
Actor	Resepsionis
Description	<i>Use case</i> melihat kamar tersedia menjelaskan bagaimana aktor resepsionis menggunakan sistem untuk melihat daftar kamar yang dapat dipesan oleh tamu
Pre-conditions	Aktor berada pada halaman cari kamar
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengisi formulir tanggal <i>check-in</i> dan tanggal <i>check-out</i> 2. Aktor memilih lihat kamar {Sistem mengolah masukan tanggal <i>check-in</i> dan tanggal <i>check-out</i>} {Sistem menampilkan daftar kamar yang dapat dipesan pada tanggal yang telah diisi}

Tabel 4.29 *Use Case Scenario* Melihat Kamar Tersedia (lanjutan)

	3. Aktor melihat data kamar tersedia 4. Use case selesai
Alternative Flow	A1. Formulir isian kosong Jika formulir isian dengan <i>mandatory</i> tanda * dibiarkan kosong, maka saat {Sistem mengolah masukan tanggal <i>check-in</i> dan tanggal <i>check-out</i>} , sistem akan menampilkan pesan error dan memberi perintah untuk wajib mengisi form, kemudian sistem akan kembali ke {Sistem menampilkan halaman cari kamar}
Sub Flow	-
Post-conditions	Aktor berhasil melihat kamar tersedia

Tabel 4.29 menunjukkan *use case scenario* melihat kamar tersedia. *Use case scenario* melihat kamar tersedia menjelaskan tujuan aktor dan alur kegiatan yang dilakukan oleh aktor resepsionis saat menggunakan sistem. *Use case scenario* melihat kamar tersedia memiliki *main flow* melihat kamar yang tersedia. *Use case scenario* melihat kamar tersedia memiliki *alternate flow*, yaitu formulir isian kosong yang akan menampilkan pesan *error* saat terdapat formulir isian yang belum diisi.

Tabel 4.30 *Use Case Scenario* Mengelola Barang Masuk

Use Case Code	UCSI-04
Use Case Name	Mengelola barang masuk
Actor	<i>Store Keeper</i>
Description	<i>Use case</i> mengelola barang masuk menjelaskan bagaimana aktor <i>store keeper</i> menggunakan sistem untuk menambah barang masuk, melihat data barang masuk, mengubah data barang masuk, dan menampilkan nota pembelian barang masuk
Pre-conditions	Aktor berada pada halaman barang masuk
Main Flow	1. Aktor memilih tambah data barang masuk {Sistem menampilkan formulir data barang masuk} 2. Aktor mengisi data formulir barang masuk yang terdiri dari nama barang, tanggal barang masuk, jumlah barang, dan foto nota 3. Aktor memilih simpan data barang masuk {Sistem menyimpan data barang masuk} {Sistem menampilkan halaman barang masuk}

Tabel 4.30 Use Case Scenario Mengelola Barang Masuk (lanjutan)

	<p>4. Aktor menjalankan <i>sub flow</i> Mengubah Data Barang Masuk</p> <p>5. Aktor menjalankan <i>sub flow</i> Menghapus Data Barang Masuk</p> <p>6. Aktor menjalankan <i>sub flow</i> Menampilkan Nota Barang Masuk</p> <p>7. Use case selesai</p>
Alternative Flow	<p>A1. Formulir isian kosong</p> <p>Jika formulir isian dengan <i>mandatory</i> tanda * dibiarkan kosong, maka saat {Sistem menyimpan data barang masuk}, sistem akan menampilkan pesan <i>error</i> dan memberi perintah untuk wajib mengisi form, kemudian sistem akan kembali ke {Sistem menampilkan formulir data barang masuk}</p> <p>A2. Formulir isian nama barang tidak sesuai</p> <p>Jika formulir isian nama barang diisi diluar daftar nama barang yang disediakan, maka saat {Sistem menyimpan data barang masuk}, sistem akan menampilkan pesan <i>error</i> dan memberi perintah untuk mengisi nama barang sesuai dengan nama barang yang ada pada daftar, kemudian sistem akan kembali ke {Sistem menampilkan formulir data barang masuk}</p>
Sub Flow	<p>S1. Mengubah Data Barang Masuk</p> <ol style="list-style-type: none"> 1. Aktor memilih data barang masuk yang ingin diubah 2. Aktor memilih edit <p>{Sistem menampilkan formulir data barang masuk yang telah terisi data barang}</p> <ol style="list-style-type: none"> 3. Aktor mengubah data barang masuk yang ingin diubah 4. Aktor memilih simpan perubahan data barang masuk <p>{Sistem menyimpan perubahan data barang masuk}</p> <p>{Sistem menampilkan kembali halaman barang masuk}</p> <ol style="list-style-type: none"> 5. Use case selesai <p>S2. Menghapus Data Barang Masuk</p> <ol style="list-style-type: none"> 1. Aktor memilih data barang masuk yang ingin dihapus 2. Aktor memilih hapus



Tabel 4.30 Use Case Scenario Mengelola Barang Masuk (lanjutan)

	<p>{Sistem menampilkan <i>alert</i> untuk meyakinkan aktor}</p> <p>3. Aktor memilih ya</p> <p>{Sistem menghapus data barang masuk yang dipilih}</p> <p>{Sistem menampilkan kembali halaman barang masuk}</p> <p>4. <i>Use case</i> selesai</p> <p>S3. Menampilkan Nota Barang Masuk</p> <p>1. Aktor memilih nota dari salah satu data barang masuk</p> <p>{Sistem menampilkan nota barang masuk}</p> <p>2. Aktor melihat nota</p> <p>3. <i>Use case</i> selesai</p>
Post-conditions	Aktor berhasil membuat data barang masuk

Tabel 4.30 menunjukkan *use case scenario* mengelola barang masuk. *Use case scenario* mengelola barang masuk menjelaskan tujuan aktor dan alur kegiatan yang dilakukan oleh aktor *store keeper* saat menggunakan sistem. *Use case scenario* mengelola barang masuk memiliki *main flow* menambah data barang masuk dengan *sub flow* mengubah dan menghapus data barang masuk. *Use case scenario* mengelola barang masuk memiliki *alternate flow*, yaitu formulir isian kosong yang akan menampilkan pesan *error* saat terdapat formulir isian yang belum diisi dan formulir isian nama barang tidak sesuai yang akan menampilkan pesan *error* saat nama barang masuk tidak sesuai dengan daftar nama barang yang ada.

Tabel 4.31 Use Case Scenario Mengelola Barang Keluar

Use Case Code	UCSI-05
Use Case Name	Mengelola barang keluar
Actor	<i>Store Keeper</i>
Description	<i>Use case</i> mengelola barang keluar menjelaskan bagaimana aktor <i>store keeper</i> menggunakan sistem untuk menambah barang keluar, melihat data barang keluar, dan mengubah data barang keluar
Pre-conditions	Aktor berada pada halaman barang keluar
Main Flow	1. Aktor memilih tambah data barang keluar {Sistem menampilkan formulir data barang keluar}



Tabel 4.31 Use Case Scenario Mengelola Barang Keluar (lanjutan)

	<p>2. Aktor mengisi data formulir barang keluar yang terdiri dari nama barang, tanggal barang keluar, jumlah barang, nama pegawai, dan tujuan pengambilan barang</p> <p>3. Aktor memilih simpan data barang keluar</p> <p>{Sistem menyimpan data barang keluar}</p> <p>{Sistem menampilkan halaman barang keluar}</p> <p>4. Aktor menjalankan <i>sub flow</i> Mengubah Data Barang Keluar</p> <p>5. Aktor menjalankan <i>sub flow</i> Menghapus Data Barang Keluar</p> <p>6. Use case selesai</p>
Alternative Flow	<p>A1. Formulir isian kosong</p> <p>Jika formulir isian dengan <i>mandatory</i> tanda * dibiarkan kosong, maka saat {Sistem menyimpan data barang keluar}, sistem akan menampilkan pesan error dan memberi perintah untuk wajib mengisi form, kemudian sistem akan kembali ke {Sistem menampilkan formulir data barang masuk}</p> <p>A2. Formulir isian nama barang tidak sesuai</p> <p>Jika formulir isian nama barang diisi diluar daftar nama barang yang disediakan, maka saat {Sistem menyimpan data barang masuk}, sistem akan menampilkan pesan error dan memberi perintah untuk mengisi nama barang sesuai dengan nama barang yang ada pada daftar, kemudian sistem akan kembali ke {Sistem menampilkan formulir data barang masuk}</p> <p>A3. Formulir isian jumlah barang keluar tidak sesuai</p> <p>Jika formulir isian jumlah diisi lebih besar daripada jumlah stok barang, maka saat {Sistem menyimpan data barang keluar}, sistem akan menampilkan pesan error dan memberi perintah untuk mengisi formulir jumlah dengan bilangan yang lebih kecil atau sama dengan jumlah barang, kemudian sistem akan kembali ke {Sistem menampilkan formulir data barang masuk}</p>
Sub Flow	<p>S1. Mengubah Data Barang Keluar</p> <ol style="list-style-type: none"> 1. Aktor memilih data barang keluar yang ingin diubah 2. Aktor memilih edit

Tabel 4.31 Use Case Scenario Mengelola Barang Keluar (lanjutan)

	<p>{Sistem menampilkan formulir data barang keluar yang telah terisi data barang}</p> <ol style="list-style-type: none"> Aktor mengubah data barang keluar yang ingin diubah Aktor memilih simpan perubahan data barang keluar <p>{Sistem menyimpan perubahan data barang keluar}</p> <p>{Sistem menampilkan kembali halaman barang keluar}</p> <ol style="list-style-type: none"> Use case selesai <p>S2. Menghapus Data Barang Keluar</p> <ol style="list-style-type: none"> Aktor memilih data barang keluar yang ingin dihapus Aktor memilih hapus <p>{Sistem menampilkan alert untuk meyakinkan aktor}</p> <ol style="list-style-type: none"> Aktor memilih ya <p>{Sistem menghapus data barang keluar yang dipilih}</p> <p>{Sistem menampilkan kembali halaman barang keluar}</p> <ol style="list-style-type: none"> Use case selesai
Post-conditions	Aktor berhasil membuat data barang keluar

Tabel 4.31 menunjukkan *use case scenario* mengelola barang keluar. *Use case scenario* mengelola barang keluar menjelaskan tujuan aktor dan alur kegiatan yang dilakukan oleh aktor *store keeper* saat menggunakan sistem. *Use case scenario* mengelola barang keluar memiliki *main flow* menambah data barang keluar dengan *sub flow* mengubah dan menghapus catatan barang keluar. *Use case scenario* mengelola barang keluar memiliki *alternate flow*, yaitu formulir isian kosong yang akan menampilkan pesan *error* saat terdapat formulir isian yang belum diisi, formulir isian nama barang tidak sesuai yang akan menampilkan pesan *error* saat nama barang masuk tidak sesuai dengan daftar nama barang yang ada, dan formulir isian jumlah barang tidak sesuai yang akan menampilkan pesan *error* saat jumlah barang keluar melebihi jumlah barang yang ada.

Tabel 4.32 Use Case Scenario Melihat Laporan

Use Case Code	UCSI-06
Use Case Name	Melihat laporan
Actor	Manajer

Tabel 4.32 Use Case Scenario Melihat Laporan (lanjutan)

Description	Use case melihat laporan menjelaskan bagaimana aktor manajer menggunakan sistem untuk melihat laporan bulanan transaksi tamu hotel, mencetak laporan bulanan transaksi tamu hotel, melihat laporan bulanan penggunaan barang, dan mencetak laporan bulanan penggunaan barang
Pre-conditions	Aktor berada pada halaman laporan
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengisi formulir bulan dan tahun 2. Aktor memilih lihat laporan bulanan transaksi tamu hotel {Sistem mengolah data laporan} {Sistem menampilkan laporan bulanan transaksi tamu hotel} 3. Aktor melihat laporan bulanan transaksi tamu 4. Aktor menjalankan <i>sub flow</i> Mencetak Laporan Bulanan Transaksi Tamu Hotel 5. Aktor menjalankan <i>sub flow</i> Melihat Laporan Bulanan Penggunaan Barang 6. Aktor menjalankan <i>sub flow</i> Mencetak Laporan Bulanan Penggunaan Barang 7. Use case selesai
Alternative Flow	<p>A1, Formulir isian kosong</p> <p>Jika formulir isian dengan <i>mandatory</i> tanda * dibiarkan kosong, maka saat {Sistem mengolah data laporan}, sistem akan menampilkan pesan <i>error</i> dan memberi perintah untuk wajib mengisi form, kemudian sistem akan kembali ke halaman laporan</p>
Sub Flow	<p>S1. Mencetak Laporan Bulanan Transaksi Tamu Hotel</p> <ol style="list-style-type: none"> 1. Aktor memilih cetak laporan {Sistem menampilkan print preview laporan} 2. Aktor mencetak laporan {Sistem mencetak laporan} 3. Use case selesai <p>S2. Melihat Laporan Bulanan Penggunaan Barang</p> <ol style="list-style-type: none"> 1. Aktor mengisi formulir bulan dan tahun 2. Aktor memilih lihat laporan bulanan penggunaan barang {Sistem mengolah data laporan}

Tabel 4.32 Use Case Scenario Melihat Laporan (lanjutan)

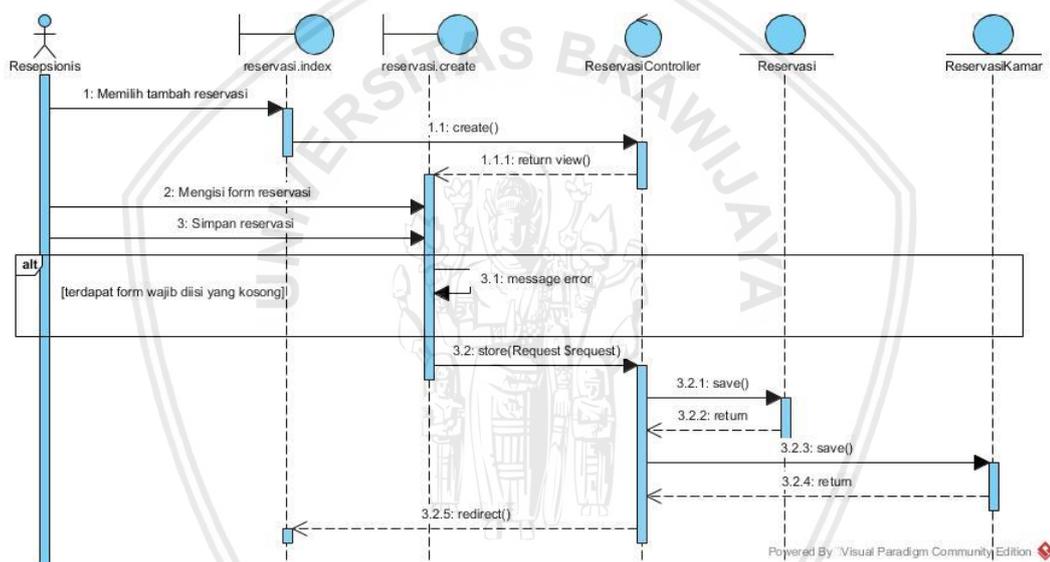
	<p>{Sistem menampilkan laporan bulanan penggunaan barang}</p> <ol style="list-style-type: none"> 3. Aktor melihat laporan bulanan penggunaan barang 4. <i>Use case</i> selesai <p>S3. Mencetak Laporan Bulanan Penggunaan Barang</p> <ol style="list-style-type: none"> 1. Aktor memilih cetak laporan <p>{Sistem menampilkan <i>print preview</i> laporan}</p> <ol style="list-style-type: none"> 2. Aktor cetak laporan <p>{Sistem mencetak laporan}</p> <ol style="list-style-type: none"> 3. <i>Use case</i> selesai
Post-conditions	Aktor berhasil melihat laporan bulanan transaksi tamu hotel

Tabel 4.32 menunjukkan *use case scenario* melihat laporan. *Use case scenario* melihat laporan menjelaskan tujuan aktor dan alur kegiatan yang dilakukan oleh aktor manajer saat menggunakan sistem. *Use case scenario* melihat laporan memiliki *main flow* melihat laporan bulanan transaksi tamu dengan *sub flow* mencetak laporan bulanan transaksi tamu, melihat laporan bulanan penggunaan barang, dan mencetak laporan bulanan penggunaan barang. *Use case scenario* melihat laporan memiliki *alternate flow*, yaitu formulir isian kosong yang akan menampilkan pesan *error* saat terdapat formulir isian yang belum diisi.

BAB 5 PERANCANGAN SISTEM

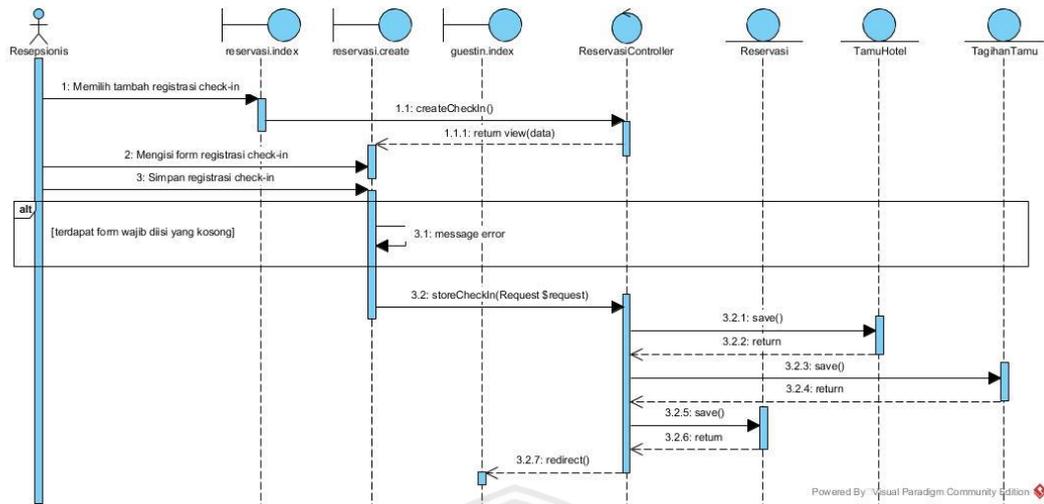
5.1 Sequence Diagram

Sequence diagram merupakan sebuah diagram *Unified Modeling Language* yang menunjukkan bagaimana *event* melakukan transformasi dari objek ke objek (Pressman, 2010). *Sequence diagram* dalam UML biasanya sering digunakan untuk memodelkan interaksi antara aktor, objek dalam sistem, dan interaksi antar mereka (Sommerville, 2011). Pada tahap ini, *sequence diagram* dibuat berdasarkan *main flow* dan beberapa *sub flow* pada *use case scenario* yang telah dibuat sebelumnya. Pemilihan pemodelan *sequence diagram* yang dibuat berdasarkan kemampuan utama atau *core* sistem. Total *sequence diagram* yang dimodelkan berjumlah 8 *diagram*.



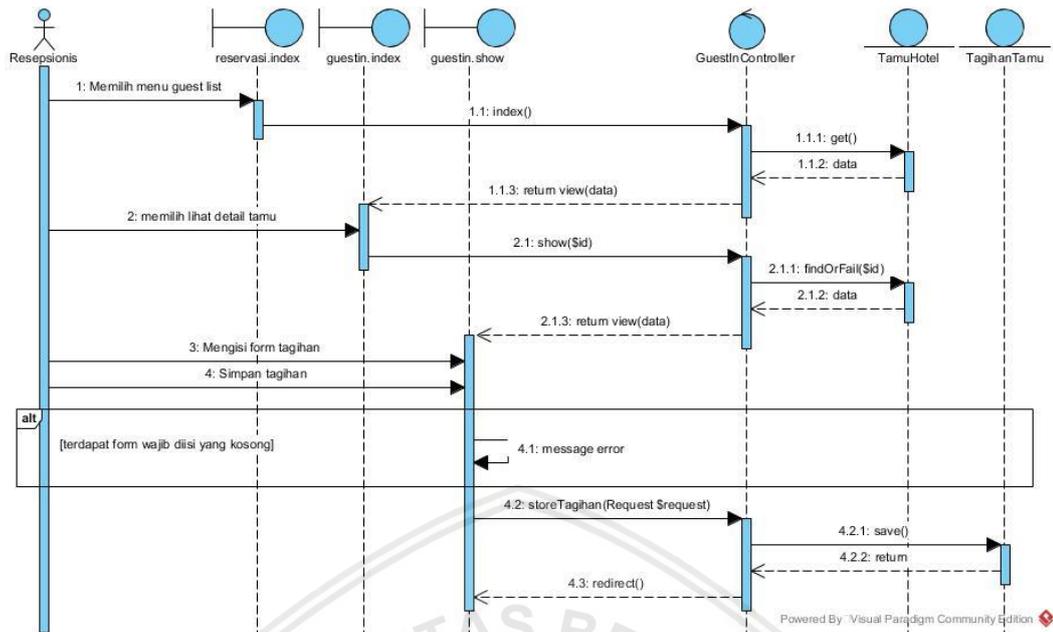
Gambar 5.1 Sequence Diagram Menambah Reservasi

Gambar 5.1 menggambarkan alur melakukan reservasi. Aktivitas dimulai saat aktor resepsionis memilih tambah reservasi pada halaman *index* reservasi. Setelah menekan tombol tambah, sistem akan memanggil *method* *create()* yang ada pada *controller* *ReservasiController*. *Method* *create()* kemudian menampilkan halaman *create* reservasi yang didalamnya *include* formulir reservasi. Lalu aktor resepsionis mengisi formulir reservasi. Setelah selesai, aktor resepsionis menyimpan formulir pada halaman *create* reservasi. Setelah menyimpan reservasi, sistem akan mengecek apakah ada formulir kosong. Jika ada maka sistem memanggil pesan *error*, jika tidak maka sistem memanggil *method* *store(Request \$request)* pada *controller* *ReservasiController*. *Method* *store(Request \$request)* melakukan penyimpanan data ke dalam *model* *Reservasi* dan *ReservasiKamar* menggunakan *method* *save()*. Alur berakhir dengan sistem menampilkan kembali halaman *index* reservasi.



Gambar 5.2 Sequence Diagram Menambah Registrasi Check-In

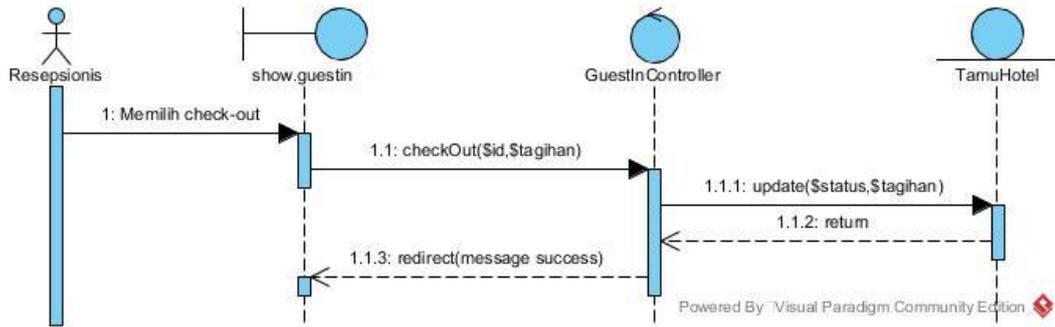
Gambar 5.2 menggambarkan alur menambah registrasi *check-in*. Aktivitas dimulai saat aktor resepsionis memilih tambah pada halaman *index* reservasi. Setelah menekan tombol *check-in*, sistem akan memanggil *method* *createCheckIn()* yang ada pada *controller* *ReservasiController*. *Method* *createCheckIn()* kemudian menampilkan halaman *create* reservasi yang didalamnya *include* formulir registrasi. Lalu aktor resepsionis mengisi formulir registrasi *check-in*. Setelah selesai, aktor resepsionis menyimpan formulir pada halaman *create* reservasi. Setelah menyimpan reservasi, sistem akan mengecek apakah ada formulir kosong. Jika ada maka sistem memanggil pesan *error*, jika tidak maka sistem memanggil *method* *storeCheckIn(Request \$request)* pada *controller* *ReservasiController*. *Method* *storeCheckIn(Request \$request)* melakukan penyimpanan data ke dalam *model* *Tamuhotel*, *TagihanTamuh*, dan *Reservasi* menggunakan *method* *save()*. Alur berakhir dengan sistem menampilkan halaman *index* daftar tamu.



Gambar 5.3 Sequence Diagram Menambah Tagihan Tamu

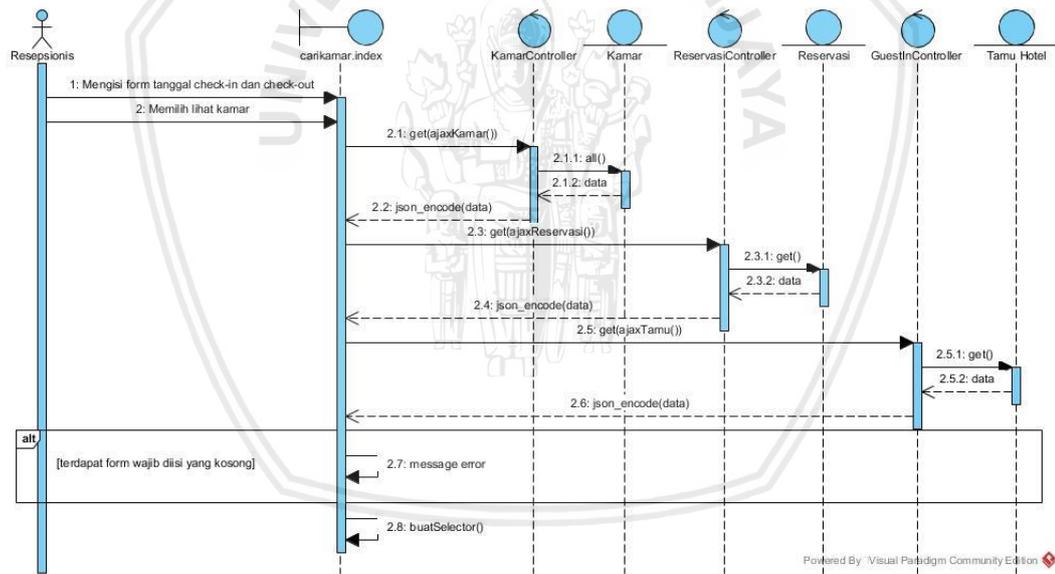
Gambar 5.3 menggambarkan alur menambah tagihan tamu. Aktivitas dimulai saat aktor resepsionis memilih menu *guest list* atau daftar tamu pada halaman *sidebar*. Setelah menekan tombol *guest list* di halaman manapun, contohnya di halaman *index* reservasi, sistem akan memanggil *method* `index()` yang ada pada *controller* `GuestInController`. *Method* `index()` melakukan pengambilan data pada *model* `TamuHotel` dengan menggunakan *method* `get()` yang kemudian menampilkan halaman *index* daftar tamu. Setelah itu aktor resepsionis memilih lihat detail tamu pada halaman *index* daftar tamu. Setelah menekan tombol *view*, sistem akan memanggil *method* `show($id)` yang ada pada *controller* `GuestInController`. *Method* `Show($id)` melakukan pengambilan data pada *model* `TamuHotel` dengan menggunakan *method* `findOrFail($id)` yang kemudian akan menampilkan halaman detail tamu *include* formulir tagihan. Lalu aktor resepsionis mengisi formulir tagihan tamu. Setelah selesai, aktor resepsionis menyimpan formulir pada halaman detail tamu. Setelah menyimpan tagihan, sistem akan mengecek apakah ada formulir kosong. Jika ada maka sistem memanggil pesan *error*, jika tidak maka sistem memanggil *method* `storeTagihan(Request $request)` pada *controller* `GuestInController`. *Method* `storeTagihan (Request $request)` melakukan penyimpanan data ke dalam *model* `TagihanTamu` menggunakan *method* `save()`. Alur berakhir dengan sistem menampilkan halaman detail tamu.





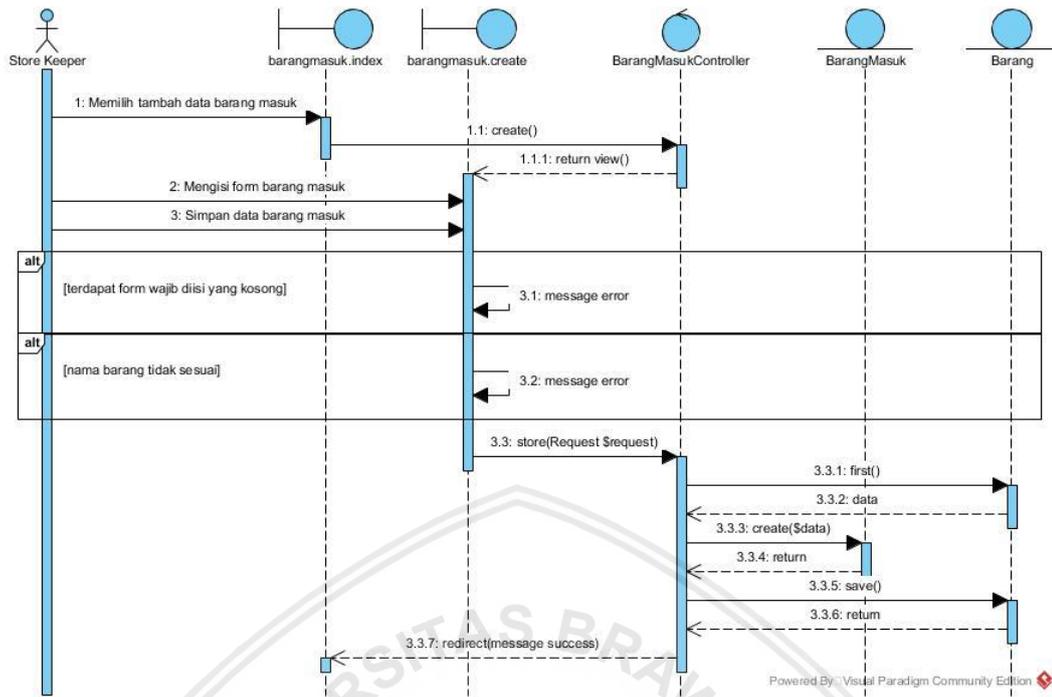
Gambar 5.4 Sequence Diagram Check-Out

Gambar 5.4 menggambarkan alur *check-out*. Aktivitas dimulai saat aktor resepsionis memilih *check-out* pada halaman *show* detail tamu. Setelah menekan tombol *check-out*, sistem akan memanggil *method* *checkOut(\$id,\$tagihan)* yang ada pada *controller* *GuestInController*. *Method* *checkOut(\$id,\$tagihan)* melakukan penyimpanan data ke dalam *model* *TamuHotel* menggunakan *method* *update(\$status,\$tagihan)*. Alur berakhir dengan sistem menampilkan halaman *show* detail tamu dengan pesan berhasil.



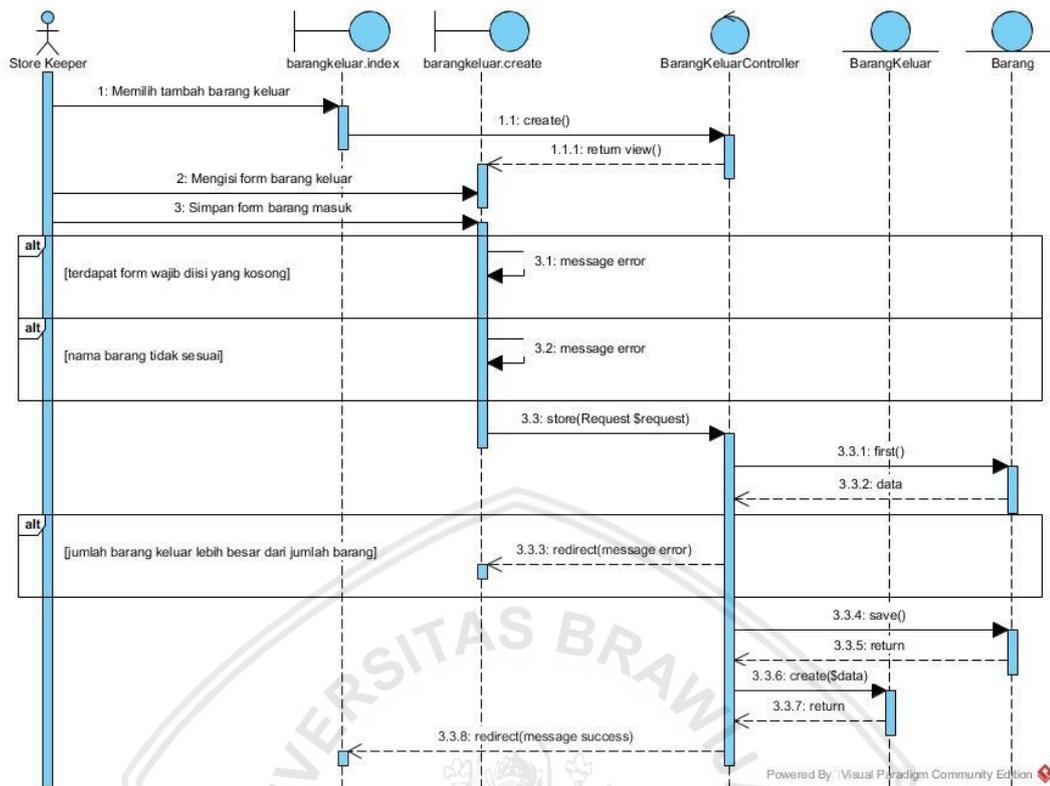
Gambar 5.5 Sequence Diagram Melihat Kamar Tersedia

Gambar 5.5 menggambarkan alur melihat kamar tersedia. Aktivitas dimulai saat aktor resepsionis mengisi formulir tanggal *check-in* dan *check-out* kemudian memilih lihat kamar pada halaman *index* cari kamar. Setelah menekan tombol lihat, sistem akan mengecek apakah ada formulir kosong. Jika ada maka sistem memanggil pesan *error*, jika tidak maka sistem memanggil *method* *buatSelector()* yang ada pada halaman *index* cari kamar. *Method* *buatSelector()* melakukan pengambilan dan pengolahan data pada API kamar, reservasi, dan registrasi *check-in*. Alur berakhir dengan sistem menampilkan halaman *index* cari kamar dengan daftar kamar tersedia.



Gambar 5.6 Sequence Diagram Menambah Barang Masuk

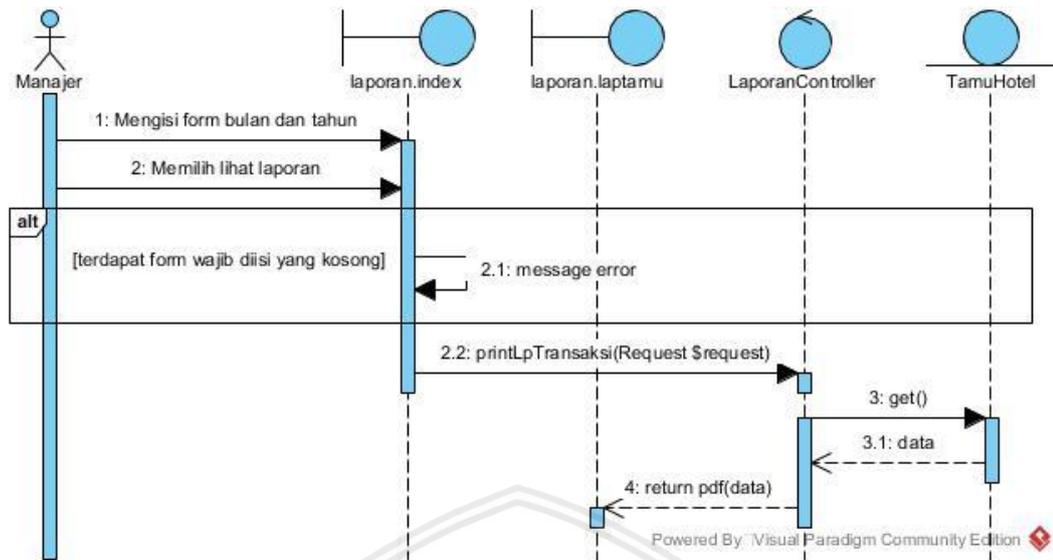
Gambar 5.6 menggambarkan alur menambah barang masuk. Aktivitas dimulai saat aktor *store keeper* memilih tambah barang masuk pada halaman *index* barang masuk. Setelah menekan tombol tambah, sistem akan memanggil *method create()* yang ada pada *controller* *BarangMasukController*. *Method create()* kemudian akan menampilkan halaman *create* barang masuk. Lalu aktor *store keeper* mengisi formulir barang masuk. Setelah selesai, aktor *store keeper* menyimpan formulir pada halaman *create* barang masuk. Setelah menekan tombol tambah, sistem akan mengecek apakah ada formulir kosong atau nama barang tidak sesuai. Jika ada maka sistem memanggil pesan *error*, jika tidak maka sistem memanggil *method store(Request \$request)* pada *controller* *BarangMasukController*. *Method store(Request \$request)* melakukan pengambilan data pada *model* *Barang* dengan menggunakan *method first()*. Kemudian data disimpan ke dalam *model* *BarangMasuk* menggunakan *method create(\$data)* dan ke dalam *model* *Barang* menggunakan *method save()*. Alur berakhir dengan sistem menampilkan halaman *index* barang masuk dengan pesan berhasil.



Gambar 5.7 Sequence Diagram Menambah Barang Keluar

Gambar 5.7 menggambarkan alur menambah barang keluar. Aktivitas dimulai saat aktor *store keeper* memilih tambah barang keluar pada halaman *index* barang keluar. Setelah menekan tombol tambah, sistem akan memanggil *method create()* yang ada pada *controller* *BarangKeluarController*. *Method create()* kemudian akan menampilkan halaman *create* barang keluar. Lalu aktor *store keeper* mengisi formulir barang keluar. Setelah selesai, aktor *store keeper* menyimpan formulir pada halaman *create* barang keluar. Setelah menekan tombol tambah, sistem akan mengecek apakah ada formulir kosong atau nama barang tidak sesuai. Jika ada maka sistem memanggil pesan *error*, jika tidak maka sistem memanggil *method store(Request \$request)* pada *controller* *BarangMasukController*. *Method store(Request \$request)* melakukan pengambilan data pada *model* *Barang* dengan menggunakan *method first()*. Lalu sistem akan mengecek apakah formulir jumlah barang keluar lebih besar dari jumlah barang. Jika ada maka sistem memanggil pesan *error*, jika tidak maka data disimpan ke dalam *model* *Barang* menggunakan *method save()* dan ke dalam *model* *BarangKeluar* menggunakan *method create(\$data)*. Alur berakhir dengan sistem menampilkan halaman *index* barang masuk dengan pesan berhasil.



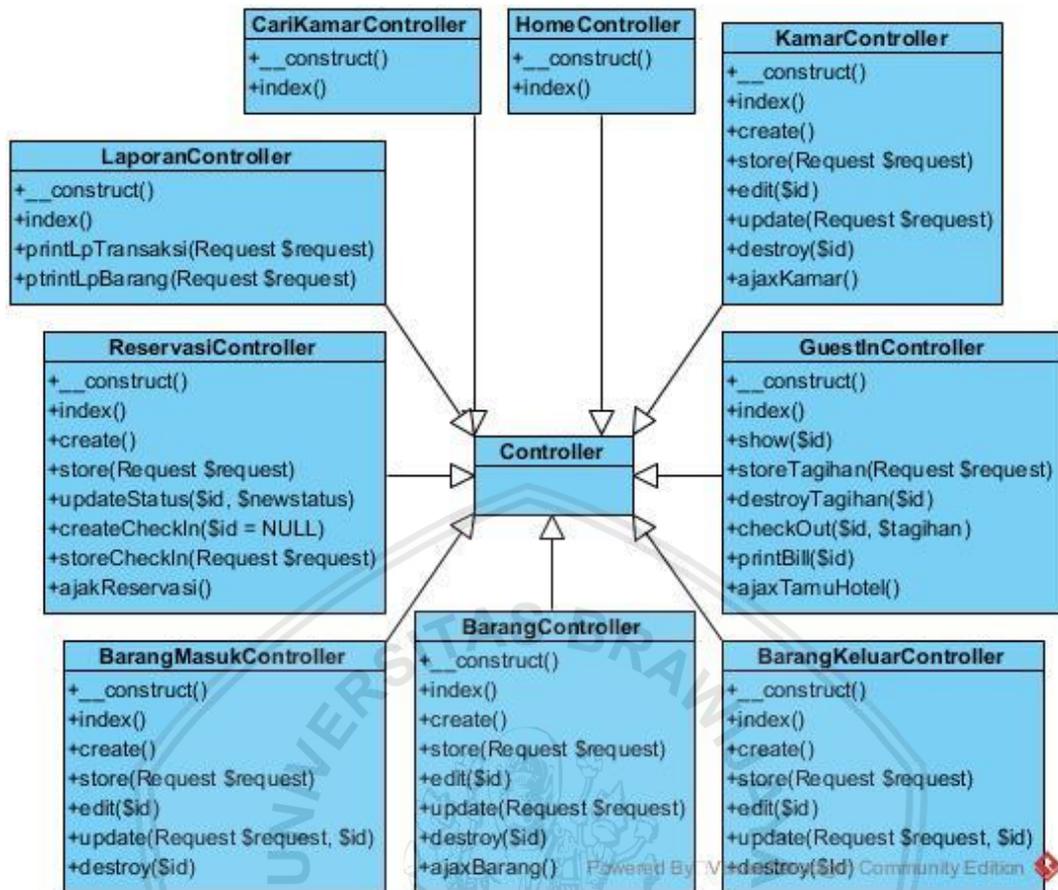


Gambar 5.8 Sequence Diagram Melihat Laporan Bulanan Transaksi Tamu

Gambar 5.8 menggambarkan alur melihat laporan bulanan transaksi tamu. Aktivitas dimulai saat aktor manajer mengisi formulir bulan dan tahun kemudian memilih lihat laporan pada halaman *index* laporan. Setelah menekan tombol lihat, sistem akan mengecek apakah ada formulir kosong. Jika ada maka sistem memanggil pesan *error*, jika tidak maka sistem memanggil *method* *printLpTransaksi*(Request \$request) yang ada pada *controller* LaporanController. *Method* *printLpTransaksi* Request \$request) melakukan pengambilan data pada *model* TamuHotel dengan menggunakan *method* *get*(). Alur berakhir dengan sistem menampilkan halaman PDF laporan bulanan transaksi tamu.

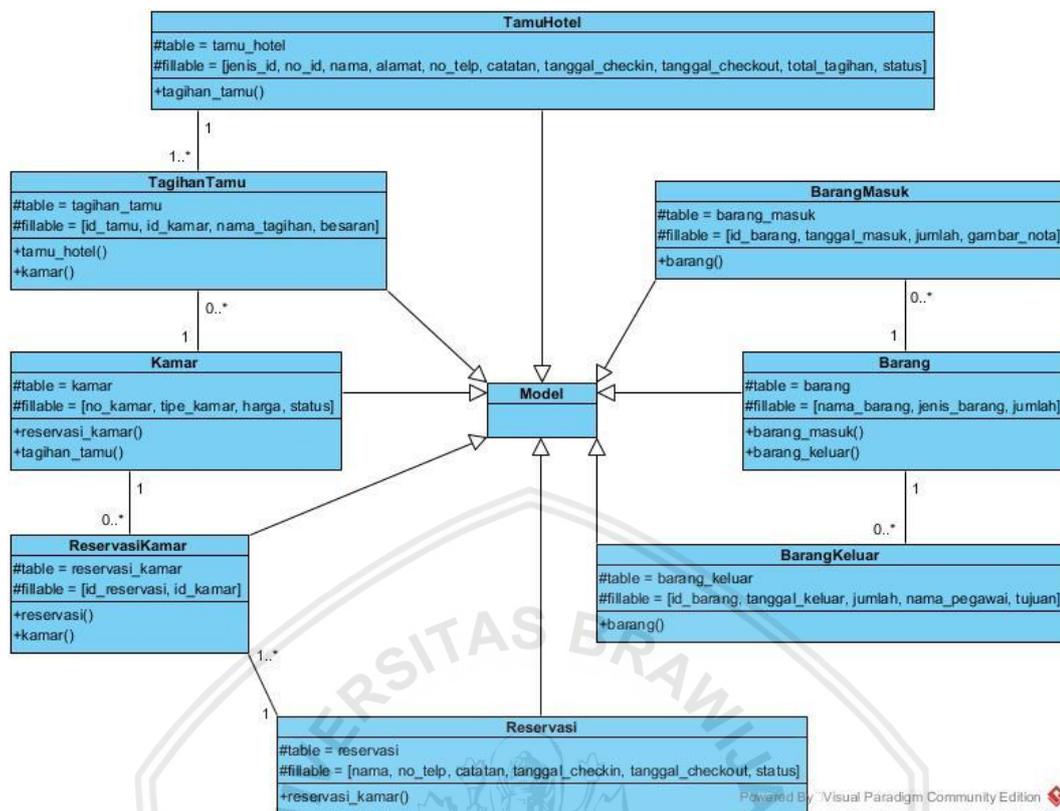
5.2 Class Diagram

Class diagram dibuat sesuai dengan *sequence diagram*. *Class diagram* dibuat menjadi 2 diagram, yaitu *controller* yang dibuat berdasarkan *control classes* pada *sequence diagram* dan *model* yang dibuat berdasarkan *entity classes* pada *sequence diagram*. *Class diagram controller* ditunjukkan oleh Gambar 5.7 sedangkan *class diagram model* ditunjukkan oleh Gambar 5.8.



Gambar 5.9 Class Diagram Controller

Gambar 5.9 merupakan *class diagram controller* sistem informasi perhotelan Edotel Pamekasan. Terdapat 9 *controller* yang merupakan turunan dari *class Controller* yang disediakan oleh *framework* Laravel. *Class* *ReservasiController* berfokus pada pengelolaan reservasi kamar hingga melakukan *check-in*. *Class* *GuestInController* berfokus pada pengelolaan tamu hingga *check-out*. *Class* *CariKamarController* berfokus pada pengelolaan pencarian kamar. *Class* *BarangMasukController* berfokus pada pengelolaan barang masuk. *Class* *BarangKeluarController* berfokus pada pengelolaan barang keluar. *Class* *KamarController* berfokus pada pengelolaan kamar. *Class* *BarangController* berfokus pada pengelolaan barang. *Class* *LaporanController* berfokus pada pengelolaan melihat laporan. Lalu, terakhir *HomeController* berfokus pada pengelolaan *dashboard*.

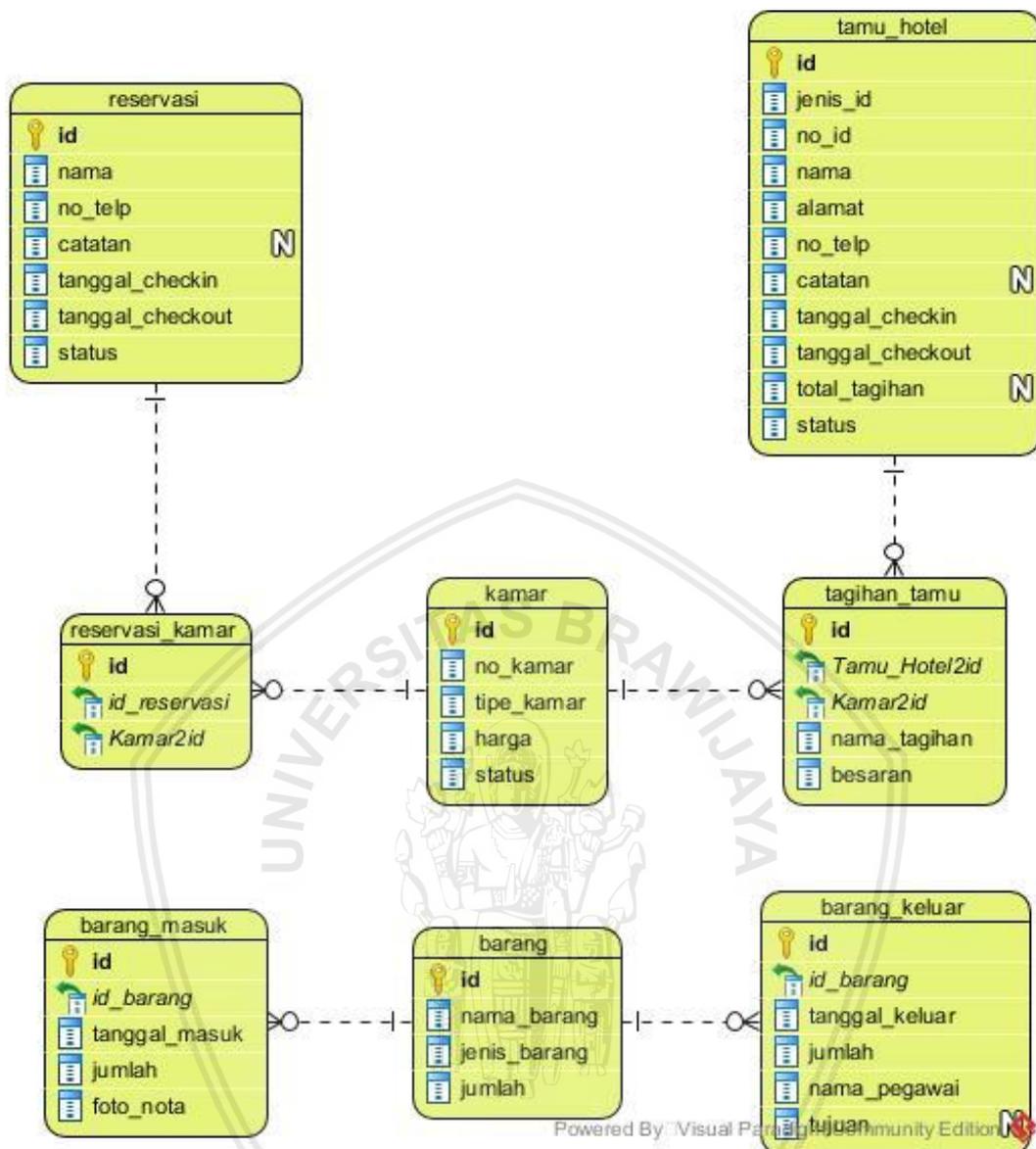


Gambar 5.10 Class Diagram Model

Gambar 5.10 merupakan *class diagram model* sistem informasi perhotelan Edotel Pamekasan. Terdapat 8 *model* yang merupakan turunan dari *class Model* merupakan *class bawaan* yang disediakan oleh *framework Laravel*. Pengembangannya dilakukan dengan menggunakan *Eloquent ORM (Object Relation Mapping)* yang dimiliki oleh *Laravel*. Hal ini dilakukan dengan tujuan memudahkan kegiatan pengembangan sistem. Untuk dapat menggunakan fitur ini, maka diharuskan membuat 2 variabel wajib, yaitu *table* dengan *fillable*. Variabel bernama *table* memiliki kegunaan untuk inisialisasi nama tabel, sedangkan variabel *fillable* memiliki kegunaan untuk inisialisasi nama kolom dalam tabelnya. Lalu untuk dapat menghubungkan relasi antar model, model harus membuat *method* dengan nama model yang dituju.

5.3 Conceptual Data Model (CDM)

Conceptual data model yang juga dikenal sebagai *conceptual schema* merupakan pemodelan informasi sebagai dasar pembuatan *database* yang didapatkan dari analisis kebutuhan yang telah dilakukan. Merupakan bentuk paling sederhana dari perancangan *data model*. *Conceptual data model* ditunjukkan pada Gambar 5.11.



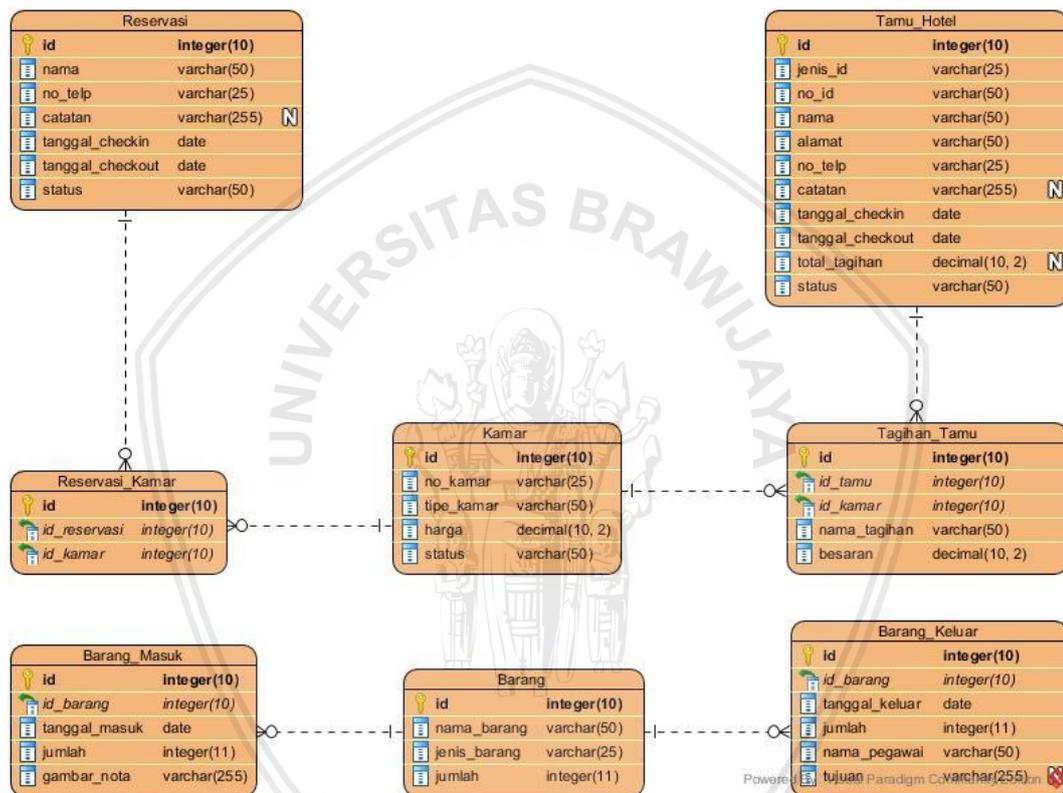
Gambar 5.11 Conceptual Data Model

Gambar 5.11 menunjukkan *entity relationship diagram* dalam bentuk *conceptual*. Ditunjukkan bahwa terdapat 8 tabel yang akan digunakan untuk menyimpan data hasil pemrosesan sistem informasi perhotelan. Tabel – tabel tersebut adalah tabel reservasi, reservasi_kamar, tamu_hotel, tagihan_tamu, kamar, barang, barang_masuk, dan barang_keluar. Tabel reservasi bertujuan untuk menyimpan data reservasi tamu mulai dari identitas hingga tanggal *check-in* dan *check-out*. Tabel reservasi kamar berfungsi untuk menyimpan data kamar – kamar yang dipesan oleh tamu. Lalu tabel tamu_hotel digunakan untuk menyimpan data tamu hotel, mirip seperti reservasi yang menyimpan identitas hingga tanggal *check-in* dan *check-out*. Kemudian tabel tagihan_tamu yang tidak hanya digunakan untuk menyimpan data kamar – kamar yang dipesan oleh tamu, namun juga digunakan untuk menyimpan tagihan – tagihan tambahan tamu. Tabel kamar bertujuan mengelola daftar kamar. Tabel barang_masuk dan barang_keluar

digunakan untuk melakukan pencatatan masing – masing barang masuk dan barang keluar. Terakhir tabel barang yang bertujuan untuk mengelola daftar barang.

5.4 Physical Data Model (PDM)

Physical data model ialah tahap lanjutan perancangan *data model* dari *conceptual data model*. *Physical data model* digunakan sebagai representasi *blueprint* dari struktur basis data. *Physical data model* memberikan gambaran mengenai bagaimana data tersusun dan saling berelasi dalam DBMS (*DataBase Management System*). *Physical data model* ditunjukkan oleh Gambar 5.12.



Gambar 5.12 Physical Data Model

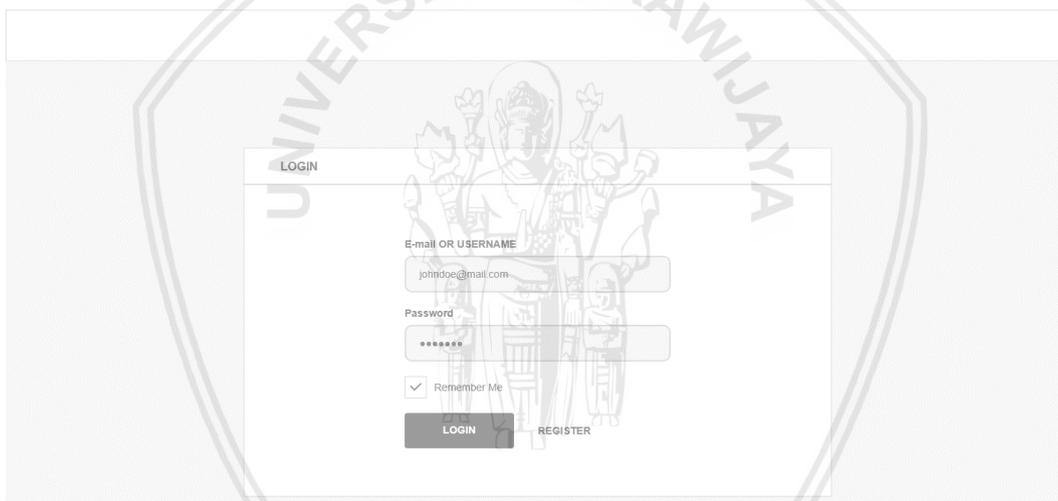
Gambar 5.12 menunjukkan *entity relationship diagram* dalam bentuk *physical*. *Physical data model* dimodelkan untuk menyempurnakan perancangan model data yang sebelumnya sudah dimodelkan dengan menggunakan *conceptual data model*. Perbedaan *physical data model* dengan *conceptual data model* adalah keberadaan perancangan tipe data pada tiap kolomnya sebagai *blueprint* implementasi sistem. Tabel reservasi menggunakan tipe data integer untuk id, varchar untuk data diri tamu, dan date untuk tanggal *check-in* serta *check-out*. Pada tabel reservasi_kamar, hanya digunakan tipe data integer untuk id. Pada tabel tamu_hotel, digunakan tipe data integer untuk id, varchar untuk data diri tamu yang menginap, date untuk tanggal *check-in* dan *check-out*, juga decimal untuk menyimpan data tagihan. Pada tabel tagihan_tamu, digunakan tipe data integer untuk id, varchar dan decimal untuk pengolahan nama tagihan dan

besarnya. Lalu tabel kamar berisi integer untuk id, informasi kamar menggunakan varchar, dan harga menggunakan decimal. Lalu tabel barang yang menggunakan tipe data integer dan varchar untuk kolom – kolomnya. Tabel barang_masuk dan barang_keluar sama – sama menggunakan integer untuk id dan jumlah, date untuk tanggal barang masuk dan barang keluar, lalu varchar untuk informasi barang masuk dan barang keluar.

5.5 Desain Antarmuka Pengguna

Perancangan desain antarmuka pengguna atau yang biasa dikenal sebagai *wireframe* merupakan rancangan tampilan atau visual yang berupa sketsa sistem informasi perhotelan. Sketsa yang ditampilkan adalah sketsa antarmuka login, beranda reservasi, tambah reservasi, tambah registrasi *check-in*, beranda *guest list*, detail tamu, cari kamar, beranda barang masuk, dan tambah barang masuk. Sketsa yang ditampilkan merupakan sketsa yang peneliti rasa penting untuk ditunjukkan dan dapat mewakili sketsa sejenis.

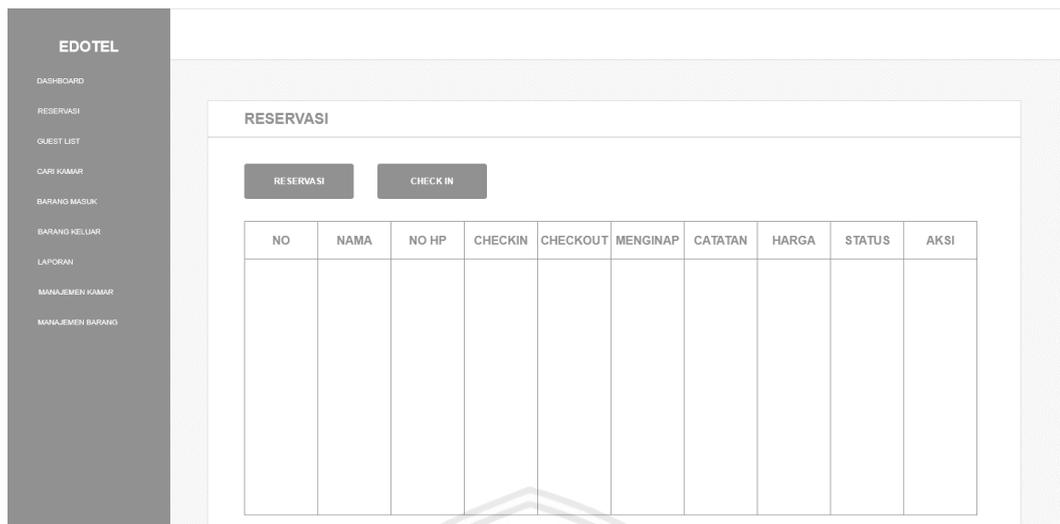
1. Login



Gambar 5.13 Wireframe Login

Gambar 5.13 merupakan rancangan halaman *login*. Halaman *login* digunakan oleh pengguna untuk masuk ke dalam sistem agar bisa mengakses fitur – fitur sistem. Pada rancangan halaman login terdapat formulir – formulir yang dapat diisi seperti formulir *username* yang dapat digunakan untuk memasukkan *username* pengguna, lalu formulir *password* yang dapat digunakan untuk memasukkan *password* pengguna. Juga terdapat tombol login yang dapat digunakan untuk masuk ke dalam sistem.

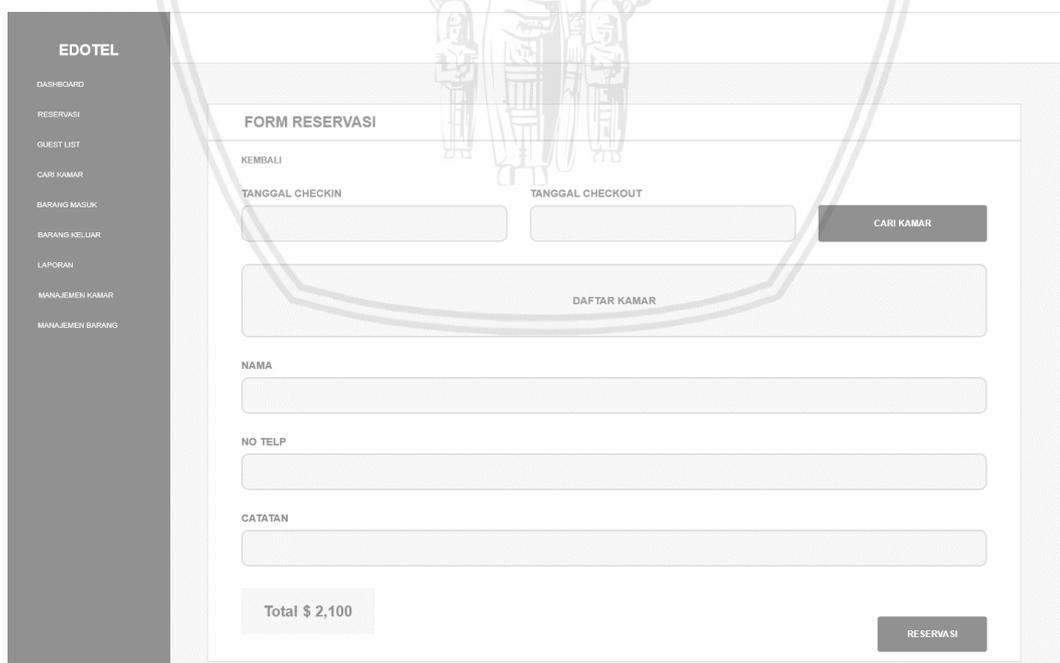
2. Beranda reservasi



Gambar 5.14 Wireframe Beranda Reservasi

Gambar 5.14 merupakan rancangan halaman beranda reservasi. Halaman beranda reservasi digunakan oleh pengguna untuk mengelola kegiatan reservasi. Pada rancangan halaman beranda reservasi, terdapat tombol reservasi yang digunakan untuk melakukan tambah reservasi, tombol *check-in* untuk melakukan *check-in*, dan tabel yang berisi informasi mengenai reservasi tamu. Pada kolom aksi akan ada tombol yang dapat digunakan untuk mengganti status reservasi tamu.

3. Tambah reservasi



Gambar 5.15 Wireframe Tambah Reservasi

Gambar 5.15 merupakan rancangan halaman tambah reservasi. Halaman tambah reservasi digunakan oleh pengguna untuk menambah reservasi tamu. Pada halaman tambah reservasi terdapat formulir – formulir reservasi, yaitu pilih



tanggal *check-in* serta *check-out*, pilihan kamar, formulir identitas tamu yang terdiri dari nama dan nomor telepon, dan juga total pembayaran tamu. Setelah mengisi formulir, pada pojok kanan bawah terdapat tombol reservasi yang digunakan untuk melakukan reservasi.

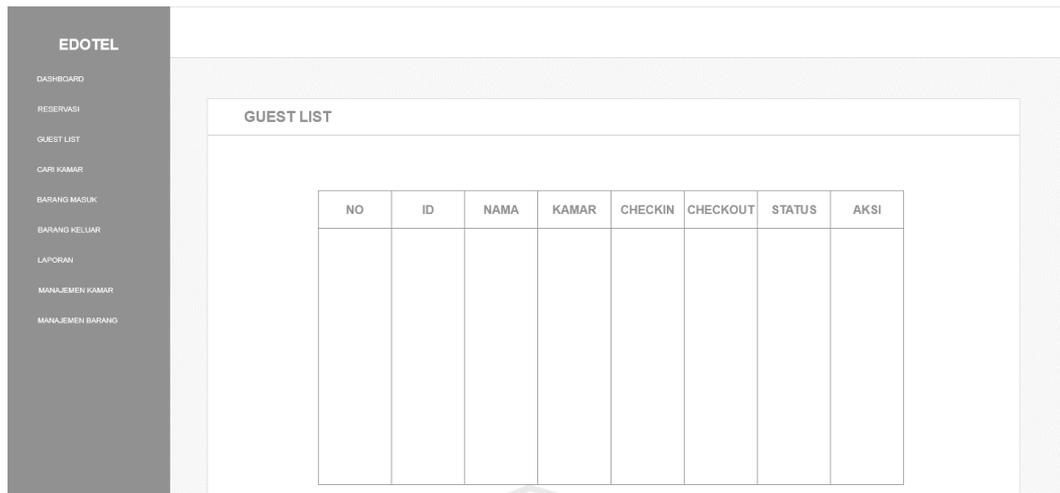
4. Tambah registrasi *check-in*

The wireframe shows a web interface for adding a check-in registration. On the left is a sidebar menu for 'EDOTEL' with items: DASHBOARD, RESERVASI, GUEST LIST, CARI KAMAR, BARANG MASUK, BARANG KELUAR, LAPORAN, MANAJEMEN KAMAR, and MANAJEMEN BARANG. The main content area is titled 'FORM REGISTRASI' and includes a 'KEMBALI' link. It features two date input fields for 'TANGGAL CHECKIN' and 'TANGGAL CHECKOUT', followed by a 'CARI KAMAR' button. Below these is a 'DAFTAR KAMAR' button. The form then has several text input fields for 'JENIS ID', 'NO ID', 'NAMA', 'ALAMAT', and 'NO TELP'. A 'CATATAN' field is also present. At the bottom left, it shows 'Total \$ 2,100' and a 'CHECK IN' button at the bottom right.

Gambar 5.16 Wireframe Tambah Registrasi *Check-In*

Gambar 5.16 merupakan rancangan halaman tambah registrasi *check-In*. Halaman tambah registrasi *check-In* digunakan oleh pengguna untuk menambah *check-in* tamu. Penggunaan halaman tambah registrasi *check-In* mirip dengan halaman tambah reservasi. Yang menjadi pembeda adalah terdapat formulir tambahan pada rancangan halaman tambah registrasi *check-In*, yaitu formulir jenis identitas, nomor identitas, dan alamat. Setelah mengisi formulir, pada pojok kanan bawah terdapat tombol *check-In* yang digunakan untuk melakukan *check-In*.

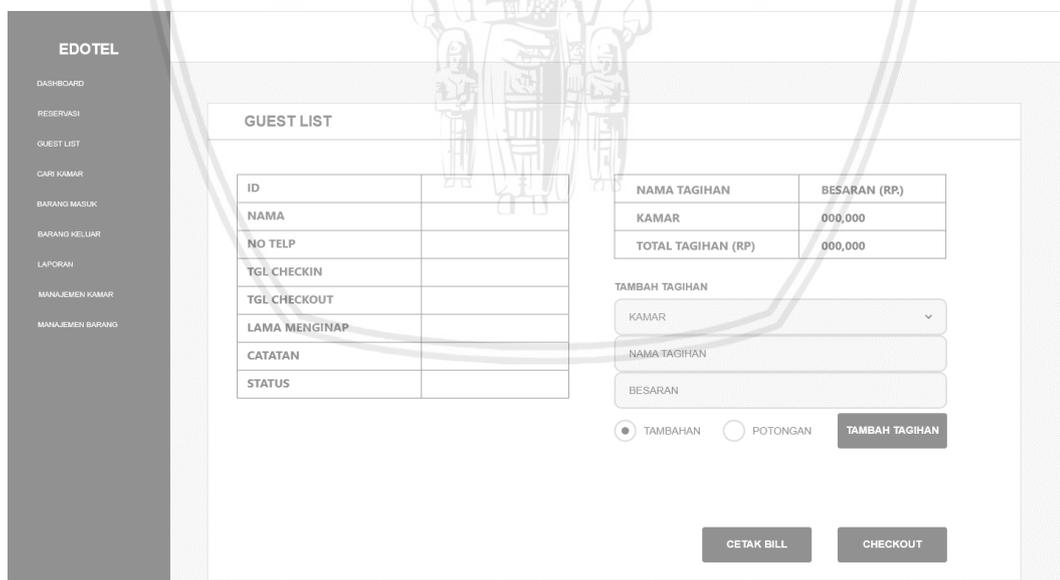
5. Beranda *guest list*



Gambar 5.17 Wireframe Beranda Guest List

Gambar 5.17 merupakan rancangan halaman beranda *guest list*. Halaman beranda *guest list* digunakan oleh pengguna untuk mengelola daftar tamu yang menginap di hotel. Pada halaman beranda *guest list* terdapat tabel yang berisi informasi mengenai tamu yang sedang menginap maupun tamu yang pernah menginap. Tabel informasi tamu terdiri dari nomor identitas, nama tamu, kamar yang digunakan, tanggal menginap, statusnya, dan aksi. Kolom aksi memiliki tombol *view* yang digunakan untuk melihat lebih detail informasi tamu.

6. Detail tamu



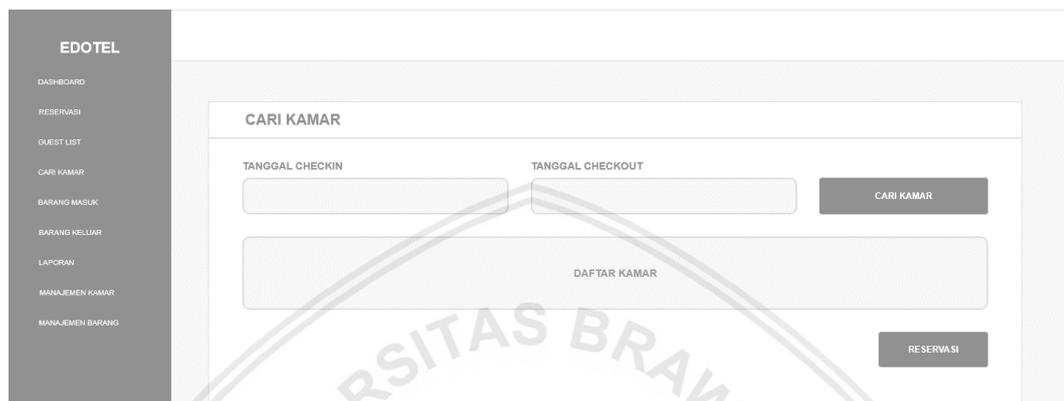
Gambar 5.18 Wireframe Detail Tamu

Gambar 5.18 merupakan rancangan halaman detail tamu. Halaman detail tamu digunakan oleh pengguna untuk mengelola tagihan tamu yang menginap di hotel, melakukan *check-out*, dan mencetak *bill*. Halaman detail tamu merupakan halaman yang bisa diakses setelah menekan tombol *view* pada halaman beranda *guest list*. Pada halaman detail tamu terdapat informasi lengkap mengenai tamu yang ditampilkan dalam bentuk tabel. Informasi data diri tamu terdiri dari nomor



identitas, nama, no telp, tanggal menginap, berapa malam tamu menginap, catatan tambahan, dan statusnya. Selain itu, halaman detail tamu juga menyajikan pengelolaan tagihan tamu yang berisi tabel daftar tagihan tamu, total tagihan, dan formulir untuk melakukan penambahan tagihan. Kemudian untuk melakukan *check-out* atau cetak *bill*, terdapat tombol *check-out* dan cetak *bill* di pojok kanan bawah.

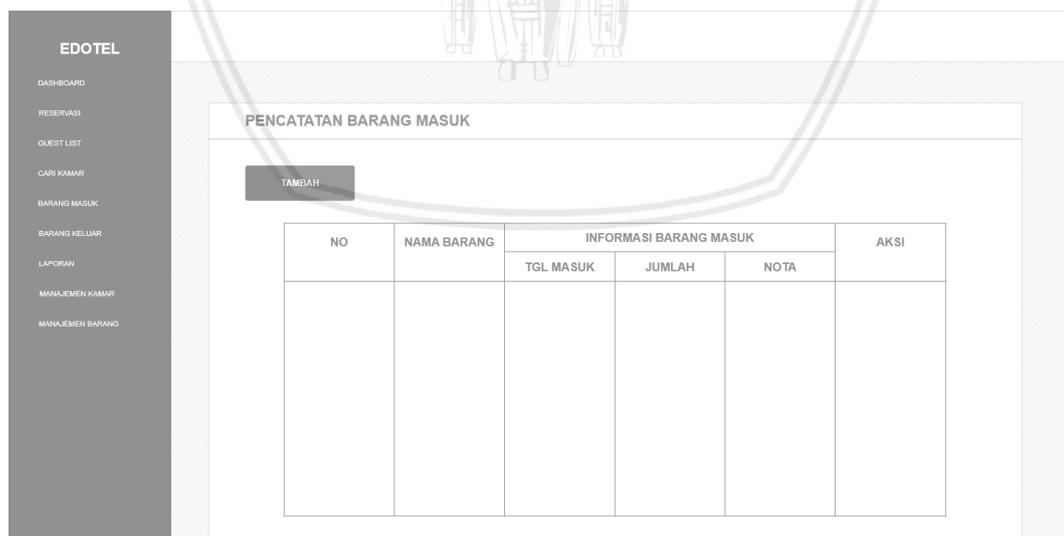
7. Cari kamar



Gambar 5.19 Wireframe Cari Kamar

Gambar 5.19 merupakan rancangan halaman cari kamar. Halaman cari kamar digunakan oleh pengguna untuk mencari kamar yang tersedia. Pada halaman cari kamar terdapat 2 formulir tanggal *check-in* dan tanggal *check-out*. Kedua formulir tersebut wajib diisi sebelum menekan tombol cari kamar. Ketika menekan tombol cari kamar, sistem akan menampilkan daftar kamar beserta statusnya.

8. Beranda barang masuk



Gambar 5.20 Wireframe Beranda Barang Masuk

Gambar 5.20 merupakan rancangan halaman beranda barang masuk. Halaman beranda barang masuk digunakan oleh pengguna untuk mengelola pencatatan barang masuk. Pada halaman beranda barang masuk terdapat tombol tambah



yang dapat digunakan untuk menambah data barang masuk. Ketika menekan tombol tambah, sistem akan memanggil halaman tambah barang masuk. Selain tombol tambah, halaman beranda barang masuk juga menyediakan informasi barang masuk yang ditulis ke dalam tabel. Tabel barang masuk terdiri dari nama barang, tanggal barang masuk, jumlah barang masuk, gambar nota, dan aksi. Untuk melakukan pengelolaan memperbarui dan menghapus data, dapat menggunakan tombol edit dan hapus yang berada pada kolom aksi.

9. Tambah barang masuk

Gambar 5.21 Wireframe Tambah Barang Masuk

Gambar 5.21 merupakan rancangan halaman tambah barang masuk. Halaman tambah barang masuk digunakan oleh pengguna untuk menambah pencatatan barang masuk. Terdapat formulir – formulir yang digunakan untuk mengisi data, antara lain formulir nama barang, tanggal barang masuk, jumlahnya, dan unggah gambar nota. Setelah mengisi formulir yang ada, disediakan tombol tambah pada pojok kanan bawah yang dapat digunakan untuk menyimpan informasi catatan barang masuk.

10. Beranda Laporan

Gambar 5.22 Wireframe Tambah Beranda Laporan

Gambar 5.22 merupakan rancangan halaman beranda laporan. Halaman beranda laporan digunakan oleh pengguna untuk melihat dan mencetak laporan

bulanan transaksi tamu maupun penggunaan barang. Pada halaman beranda laporan terdapat formulir bulan dan tahun yang diisi pilihan bulan dan tahunnya. Kemudian untuk menampilkan laporannya, pengguna dapat menekan tombol lihat laporan.



BAB 6 IMPLEMENTASI SISTEM

6.1 Spesifikasi Sistem

Spesifikasi sistem yang dilakukan dalam tahap implementasi sistem yang merupakan awal fase *construction* pada metode RUP adalah kegiatan mendeskripsikan rincian perangkat yang digunakan untuk melakukan pengembangan sistem informasi. Spesifikasi sistem terbagi menjadi 2 pendeskripsian perangkat, yakni spesifikasi perangkat keras (*hardware*) yang membahas detail perangkat keras yang digunakan dan spesifikasi perangkat lunak (*software*) yang membahas detail perangkat lunak yang digunakan.

6.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras menjelaskan komponen perangkat keras yang digunakan untuk melakukan pengembangan sistem informasi perhotelan. Komponen perangkat keras yang dideskripsikan adalah *model pc/laptop*, *processor*, *graphic card*, dan *random access memory (RAM)*. Spesifikasi perangkat keras ditunjukkan pada Tabel 6.1.

Tabel 6.1 Spesifikasi Perangkat Keras

No.	Komponen	Spesifikasi
1	<i>Model PC / Laptop</i>	HP Pavilion 14 Notebook PC K8U52PA#AR6
2	<i>Processor</i>	Intel® Core™ i7-5500U CPU @ 2.40GHz (4 CPUs), ~2.4GHz
3	<i>Graphic Card</i>	Nvidia GeForce 840M
4	<i>Random Access Memory (RAM)</i>	8192MB RAM / 8GB RAM

Tabel 6.1 menunjukkan spesifikasi perangkat keras yang digunakan untuk melakukan pengembangan sistem informasi perhotelan. Terdiri dari *model laptop* yang menggunakan HP Pavilion 14 Notebook PC K8U52PA#AR6, dengan *processor* Intel® Core™ i7-5500U CPU @ 2.40GHz (4 CPUs), ~2.4GHz. Perangkat keras yang digunakan juga didukung oleh *graphic card* Nvidia GeForce 840M, dan memiliki *random access memory (RAM)* sebesar 8GB.

6.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak menjelaskan komponen perangkat lunak yang digunakan dalam membangun sistem informasi perhotelan. Komponen perangkat lunak yang dideskripsikan adalah *operating system (OS)*, *code editor*, *server*, *framework*, *UI template*, *programming language*, *database management system (DBMS)*, *version control*, dan *web browser*. Spesifikasi perangkat lunak ditunjukkan pada Tabel 6.2.

Tabel 6.2 Spesifikasi Perangkat Lunak

No.	Komponen	Spesifikasi
1	<i>Operating System (OS)</i>	Windows 10 Home Single Language 64-bit (10.0, Build 17763)
2	<i>Code Editor</i>	Visual Studio Code v1.35.0
3	<i>Server</i>	XAMPP v3.2.3 Apache localhost
4	<i>Framework</i>	Laravel Framework 5.7.28
5	<i>UI Template</i>	SB Admin 2
6	<i>Programming Language</i>	PHP v7.3.4, HTML, dan Javascript
7	<i>Database Management System (DBMS)</i>	MySQL
8	<i>Version Control</i>	Git v2.21.0
9	<i>Web Browser</i>	Google Chrome v74.0.3729.169

Tabel 6.2 menunjukkan spesifikasi perangkat lunak yang digunakan dalam mengembangkan sistem informasi perhotelan. Terdiri dari *operating system (OS)* Windows 10 Home Single Language 64-bit (10.0, Build 17763), menggunakan *code editor* Visual Studio Code v1.35.0, menggunakan *server* XAMPP v3.2.3 Apache localhost, dengan *framework* Laravel Framework 5.7.28, lalu *UI template* SB Admin 2, *programming language* PHP v7.3.4, HTML, dan Javascript. Serta *database management system (DBMS)* MySQL, *version control* Git v2.21.0, dan *web browser* Google Chrome v74.0.3729.169.

6.2 Implementasi Class

Implementasi *class* merupakan tahapan pada fase *construction*. Tahap ini mengimplementasi *class – class* yang telah dirancang pada pemodelan *class diagram* dalam fase *elaboration*. Terdapat 2 pemodelan yang diimplementasikan, yaitu *class controller* dan *class model* dengan rincian masing – masing 9 *class controller* dan 8 *class model*. Implementasi class ditunjukkan pada Tabel 6.3.

Tabel 6.3 Implementasi Class

No.	Nama Package	Nama Class	Nama File
1	<i>Controllers</i>	ReservasiController	ReservasiController.php
2	<i>Controllers</i>	GuestInController	GuestInController.php
3	<i>Controllers</i>	CariKamarController	CariKamarController.php
4	<i>Controllers</i>	BarangMasukController	BarangMasukController.php

Tabel 6.3 Implementasi *Class* (lanjutan)

No.	Nama <i>Package</i>	Nama <i>Class</i>	Nama <i>File</i>
5	<i>Controllers</i>	BarangKeluarController	BarangKeluarController.php
6	<i>Controllers</i>	LaporanController	LaporanController.php
7	<i>Controllers</i>	KamarController	KamarController.php
8	<i>Controllers</i>	BarangController	BarangController.php
9	<i>Controllers</i>	HomeController	HomeController.php
10	<i>App</i>	Reservasi	Reservasi.php
11	<i>App</i>	ReservasiKamar	ReservasiKamar.php
12	<i>App</i>	TamuHotel	TamuHotel.php
13	<i>App</i>	TagihanTamud	TagihanTamud.php
14	<i>App</i>	Kamar	Kamar.php
15	<i>App</i>	BarangMasuk	BarangMasuk.php
16	<i>App</i>	BarangKeluar	BarangKeluar.php
17	<i>App</i>	Barang	Barang.php

Tabel 6.3 membahas implementasi *class* yang telah dilakukan dalam tahap implementasi sistem. *Class* diimplementasi sesuai dengan perancangan *class diagram* yang telah dilakukan sebelumnya. *Controller class* pertama adalah *ReservasiController* yang digunakan untuk mengolah data kegiatan reservasi seperti tambah reservasi sebelum ditampilkan kepada pengguna. Kemudian *GuestInController* merupakan *class* kedua yang diimplementasikan yang digunakan untuk mengolah data tamu seperti tambah tagihan yang akan ditampilkan kepada pengguna. Setelah itu, *class CariKamarController* diimplementasi yang digunakan untuk mengatur data yang dikirimkan pada halaman cari kamar. Keempat, yaitu implementasi *class BarangMasukController* yang berisi perintah pengolahan data barang masuk. Selanjutnya, *BarangKeluarController* merupakan *class* yang mirip dengan *class BarangMasukController* yang mana *class* ini juga memiliki fokus pada pengelolaan data barang, yaitu barang keluar seperti tambah, perbarui, dan hapus barang keluar. *Class* kelima, yaitu *LaporanController* merupakan *class* yang mengolah data laporan yang ingin ditampilkan kepada pengguna. Kemudian, *KamarController* dan *BarangController* yang masing – masing memiliki fungsi untuk mengelola penambahan, pembaruan, dan penghapusan daftar kamar dan daftar barang. *Class* terakhir dalam *controller class* yang diimplementasi adalah *HomeController*. *Class HomeController* memiliki fungsi mengolah data yang ingin ditampilkan pada *dashboard*.

Selanjutnya terdapat implementasi *model class* sejumlah 8 *class*. *Model class* pertama adalah *model class* Reservasi yang memiliki fungsi untuk mengelola penyimpanan, pembaruan dan penghapusan data reservasi tamu. *Class* berikutnya adalah ReservasiKamar yang digunakan sebagai pengelolaan simpan, *update*, dan hapus data kamar yang dipeservasi tamu. Selanjutnya implementasi *model class* TamuHotel dan TagihanTamu yang masing – masing digunakan untuk mengelola penyimpanan, pembaruan dan penghapusan data tamu hotel dan data tagihan tamu hotel termasuk kamar dan tagihan lainnya. Kelima, *model class* Kamar diimplementasi untuk pengelolaan simpan, *update*, dan hapus data kamar yang dimiliki hotel. Kemudian implementasi *class* BarangMasuk dan BarangKeluar dengan tujuan untuk mengelola mengelola penyimpanan, pembaruan dan penghapusan data masing – masing barang masuk dan barang keluar. Terakhir, yaitu implementasi *model class* Barang yang berfungsi mengelola penyimpanan, pembaruan dan penghapusan daftar barang yang dimiliki hotel.

6.3 Implementasi Database

Implementasi *database* ialah tahapan yang dilakukan dalam fase *construction*. Implementasinya dilakukan dengan memakai MySQL sebagai DBMS dan memakai phpMyAdmin sebagai aplikasi pengolahnya. Terdapat 8 tabel yang diimplementasi sesuai dengan perancangan *conceptual data model (CDM)* dan *physical data model (PDM)* yang telah dilakukan pada fase *elaboration*. Implementasi 8 tabel ditunjukkan dalam gambar berikut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	nama	varchar(50)	utf8mb4_unicode_ci		No	None		
3	no_telp	varchar(25)	utf8mb4_unicode_ci		No	None		
4	catatan	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
5	tanggal_checkin	date			No	None		
6	tanggal_checkout	date			No	None		
7	status	varchar(50)	utf8mb4_unicode_ci		No	None		

Gambar 6.1 Implementasi Tabel reservasi

Gambar 6.1 menunjukkan hasil implementasi tabel reservasi. Tabel ini digunakan untuk menyimpan data reservasi tamu hotel. Memiliki *primary key* dengan nama kolom id yang bersifat *auto increment* dan tipe datanya adalah *integer*. Kemudian kolom nama, no_telp, catatan dengan tipe data varchar yang digunakan untuk menyimpan data diri tamu. Selanjutnya kolom tanggal_checkin dan tanggal_checkout dengan tipe data *date* mempunyai fungsi untuk mencatat tanggal reservasi yang diinginkan tamu. Terakhir kolom status yang digunakan untuk mengetahui status reservasi tamu.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	id_reservasi	int(10)		UNSIGNED	No	None		
3	id_kamar	int(10)		UNSIGNED	No	None		

Gambar 6.2 Implementasi Tabel reservasi_kamar

Gambar 6.2 menunjukkan hasil implementasi tabel reservasi_kamar. Tabel ini digunakan untuk menyimpan data kamar – kamar yang dipesan oleh tamu hotel. Memiliki *primary key* dengan nama kolom id dengan tipe data *integer* dan memiliki sifat *auto increment*. Tabel reservasi_kamar juga memiliki *foreign key* dengan nama kolom id_reservasi yang memiliki tipe data *integer* dan juga memiliki relasi dengan tabel reservasi yang berguna untuk mencatat identitas reservasi tamu, dan memiliki *foreign key* dengan nama kolom id_kamar dan tipe datanya *integer* yang memiliki relasi dengan tabel kamar yang digunakan untuk mencatat data kamar yang dipesan tamu.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	jenis_id	varchar(25)	utf8mb4_unicode_ci		No	None		
3	no_id	varchar(50)	utf8mb4_unicode_ci		No	None		
4	nama	varchar(50)	utf8mb4_unicode_ci		No	None		
5	alamat	varchar(50)	utf8mb4_unicode_ci		No	None		
6	no_telp	varchar(25)	utf8mb4_unicode_ci		No	None		
7	catatan	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
8	tanggal_checkin	date			No	None		
9	tanggal_checkout	date			No	None		
10	total_tagihan	decimal(10,2)			Yes	NULL		
11	status	varchar(50)	utf8mb4_unicode_ci		No	None		

Gambar 6.3 Implementasi Tabel tamu_hotel

Gambar 6.3 menunjukkan hasil implementasi tabel tamu_hotel. Tabel ini digunakan untuk menyimpan data transaksi tamu hotel. Memiliki *primary key* dengan nama kolom id yang bersifat *auto increment* dan memiliki tipe data *integer*. Kemudian terdapat kolom jenis_id, no_id, nama, alamat, no_telp, dan catatan yang memiliki tipe data *varchar* digunakan untuk menyimpan informasi tamu lengkap mengenai yang menginap di hotel. Selanjutnya kolom tanggal_checkin dan tanggal_checkout dengan tipe data *date* digunakan untuk mencatat tanggal tamu menginap dan digunakan untuk menghitung berapa malam tamu menginap. Kolom kesepuluh, yaitu kolom total_tagihan dengan tipe data *decimal* memiliki fungsi untuk mencatat total tagihan tamu yang digunakan untuk kebutuhan laporan bulanan transaksi tamu. Terakhir implementasi kolom status dengan tipe data *varchar* yang digunakan untuk mengetahui status tamu apakah sudah *check-out* atau masih berada di hotel.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	id_tamu	int(10)		UNSIGNED	No	None		
3	id_kamar	int(10)		UNSIGNED	No	None		
4	nama_tagihan	varchar(50)	utf8mb4_unicode_ci		No	None		
5	besaran	decimal(10,2)			No	None		

Gambar 6.4 Implementasi Tabel tagihan_tamu

Gambar 6.4 menunjukkan hasil implementasi tabel tagihan_tamu. Tabel ini digunakan untuk menyimpan data rincian tagihan transaksi tamu hotel. Memiliki *primary key* dengan nama kolom id dengan tipe data *integer* yang bersifat *auto increment*. Tabel tagihan *foreign key* dengan nama kolom id_tamu dan menggunakan tipe data *integer* memiliki relasi dengan tabel tamu_hotel yang berfungsi untuk mendapatkan informasi pemilik tagihan, dan memiliki *foreign key* dengan nama kolom id_kamar dan tipe datanya *integer* yang memiliki relasi dengan tabel kamar yang digunakan untuk mengetahui tagihan berada di kamar yang mana. Tabel ini juga memiliki kolom nama_tagihan dan besaran dengan masing – masing tipe datanya adalah *varchar* dan *decimal*. Kedua kolom tersebut digunakan untuk menjelaskan tagihan beserta harganya.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	no_kamar	varchar(25)	utf8mb4_unicode_ci		No	None		
3	tipe_kamar	varchar(50)	utf8mb4_unicode_ci		No	None		
4	harga	decimal(10,2)			No	None		
5	status	varchar(50)	utf8mb4_unicode_ci		No	None		

Gambar 6.5 Implementasi Tabel kamar

Gambar 6.5 menunjukkan hasil implementasi tabel kamar. Tabel ini digunakan untuk menyimpan data kamar hotel. Memiliki *primary key* dengan nama kolom id yang memiliki tipe data *integer* dan bersifat *auto increment*, dan juga memiliki *unique key* dengan nama kolom no_kamar dan tipe datanya *varchar*. *Unique key* digunakan agar tidak terjadi duplikasi nomor kamar. Lalu terdapat kolom tipe_kamar, harga, dan status kamar dengan tipe data *varchar* dan *decimal* yang digunakan untuk mendeskripsikan tipe, harga, dan status hotel.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 🔑	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	id_barang 🔑	int(10)		UNSIGNED	No	None		
3	tanggal_masuk	date			No	None		
4	jumlah	int(11)			No	None		
5	gambar_nota	varchar(255)	utf8mb4_unicode_ci		No	None		

Gambar 6.6 Implementasi Tabel barang_masuk

Gambar 6.6 menunjukkan hasil implementasi tabel `barang_masuk`. Tabel ini digunakan untuk menyimpan data pencatatan barang masuk. Memiliki *primary key* dengan nama kolom `id` dengan tipe data *integer* yang memiliki sifat *auto increment*, dan memiliki *foreign key* dengan nama kolom `id_barang` yang juga menggunakan tipe data *integer* yang memiliki relasi dengan tabel `barang` dengan tujuan menghubungkan nama barang dengan pencatatan barang masuknya. Juga terdapat kolom yang menyimpan informasi data barang masuk seperti `tanggal_masuk` barang dengan tipe data *date*, kemudian kolom `jumlah` dengan tipe data *integer*, dan yang terakhir adalah kolom `gambar_nota` dengan tipe data *varchar* yang digunakan untuk menyimpan lokasi penyimpanan gambar.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 🔑	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	id_barang 🔑	int(10)		UNSIGNED	No	None		
3	tanggal_keluar	date			No	None		
4	jumlah	int(11)			No	None		
5	nama_pegawai	varchar(50)	utf8mb4_unicode_ci		No	None		
6	tujuan	varchar(255)	utf8mb4_unicode_ci		No	None		

Gambar 6.7 Implementasi Tabel barang_keluar

Gambar 6.7 menunjukkan hasil implementasi tabel `barang_keluar`. Tabel ini digunakan untuk menyimpan data pencatatan barang keluar. Memiliki *primary key* dengan nama kolom `id`, tipe data *integer*, dan bersifat *auto increment*. Tabel ini memiliki *foreign key* dengan nama kolom `id_barang` yang juga menggunakan tipe data *integer* yang memiliki relasi dengan tabel `barang` dengan tujuan menghubungkan nama barang dengan pencatatan barang keluarannya. Juga terdapat kolom yang menyimpan informasi data barang keluar seperti `tanggal_keluar` barang dengan tipe data *date*, kemudian kolom `jumlah` dengan tipe data *integer*, dan yang terakhir adalah kolom `nama_pegawai` dan kolom `tujuan` dengan tipe data *varchar* yang digunakan untuk menyimpan informasi siapa yang mengeluarkan barang dari gudang dan apa tujuannya.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	nama_barang	varchar(50)	utf8mb4_unicode_ci		No	None		
3	jenis_barang	varchar(25)	utf8mb4_unicode_ci		No	None		
4	jumlah	int(11)			No	None		

Gambar 6.8 Implementasi Tabel barang

Gambar 6.8 menunjukkan hasil implementasi tabel barang. Tabel ini digunakan untuk menyimpan data barang di gudang hotel. Memiliki *primary key* dengan nama kolom id yang menggunakan tipe data *integer* dan bersifat *auto increment*. Tabel barang memiliki *unique key* dengan nama kolom nama_barang dan tipe datanya adalah *varchar*. *Unique key* dibutuhkan untuk menghindari duplikasi nama barang di gudang. Kemudian terdapat kolom jenis barang dan jumlah dengan tipe data *varchar* dan *integer* yang digunakan untuk mencatat tipe barang dan jumlah stoknya.

6.4 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna merupakan salah satu tahapan implementasi sistem yang ada dalam fase *construction*. Implementasi antarmuka pengguna dilakukan berdasarkan *wireframe* perancangan antarmuka pengguna yang telah dibuat pada fase *elaboration*. Antarmuka pengguna yang ditampilkan adalah antarmuka login, beranda reservasi, tambah reservasi, tambah registrasi *check-in*, beranda *guest list*, detail tamu, cari kamar, beranda barang masuk, tambah barang masuk, dan beranda laporan. Antarmuka pengguna yang ditampilkan merupakan antarmuka yang termasuk dalam kebutuhan utama sistem dan antarmuka yang dapat mewakili antarmuka sejenis. Berikut adalah implementasi antarmuka pengguna:

1. Login

Gambar 6.9 Implementasi Antarmuka Login

Gambar 6.9 merupakan hasil implementasi antarmuka login. Halaman *login* digunakan oleh pengguna untuk masuk ke dalam sistem agar bisa mengakses fitur

– fitur sistem. Untuk masuk ke dalam sistem, pengguna dapat memasukkan *username* dan *password* yang dimiliki, kemudian menekan tombol *login* untuk masuk ke dalam *dashboard*.

2. Beranda reservasi

The screenshot displays the reservation management interface for EDOTEL SMKN 3 PAMEKASAN. The interface includes a sidebar menu with options like Dashboard, Reservasi, Guest List, Cari Kamar, Manajemen Kamar, Manajemen Barang, Barang Masuk, Barang Keluar, and Laporan. The main content area is titled 'Reservasi' and contains four sections:

- Tamu Sudah Konfirmasi Reservasi**: Shows 2 confirmed reservations.

#	Nama	Nomor Telepon	Nomor Kamar	Tanggal Check-In	Tanggal Check-Out	Lama Menginap	Catatan	Harga (Rp.)	Status	Aksi
1	Radea Zulindra Ardisukma	081231458001	105	2019-06-28	2019-06-29	1 Hari		160,000	Sudah Konfirmasi	Check-In
2	Putri	573521	104	2019-06-29	2019-06-30	1 Hari		160,000	Sudah Konfirmasi	Check-In
- Tamu Belum Konfirmasi Reservasi**: Shows 1 unconfirmed reservation.

#	Nama	Nomor Telepon	Nomor Kamar	Tanggal Check-In	Tanggal Check-Out	Lama Menginap	Catatan	Harga (Rp.)	Status	Aksi
1	Putri	573521	104	2019-06-29	2019-06-30	1 Hari		160,000	Belum Konfirmasi	Konfirmasi, Batal
- Tamu Sudah Check-In**: Shows 3 checked-in guests.

#	Nama	Nomor Telepon	Nomor Kamar	Tanggal Check-In	Tanggal Check-Out	Lama Menginap	Catatan	Harga (Rp.)	Status	Aksi
1	Zulindra	081231458001	104	2019-05-22	2019-05-24	2 Hari		320,000	Sudah Check-In	
2	A	6	201	2019-05-23	2019-05-24	1 Hari		160,000	Sudah Check-In	
3	Radea	081231457997	100	2019-05-30	2019-06-02	3 Hari	Tambah Extra Bed	1,125,000	Sudah Check-In	
- Tamu Batal Reservasi**: Shows 1 canceled reservation.

#	Nama	Nomor Telepon	Nomor Kamar	Tanggal Check-In	Tanggal Check-Out	Lama Menginap	Catatan	Harga (Rp.)	Status	Aksi
1	Rizky	5453413142	104	2019-06-21	2019-06-22	1 Hari		160,000	Batal Reservasi	

Gambar 6.10 Implementasi Antarmuka Beranda Reservasi

Gambar 6.10 merupakan hasil implementasi antarmuka beranda reservasi. Halaman beranda reservasi digunakan oleh pengguna resepsionis untuk mengelola kegiatan reservasi. Pada antarmuka antarmuka beranda reservasi

terdapat tabel mengenai informasi reservasi tamu yang terdiri dari nama tamu yang melakukan reservasi, nomor teleponnya, kamar yang dipesan, tanggal beserta lama menginap, harga, status, dan aksi. Selain itu, pengguna resepsionis dapat menambah reservasi dengan cara menekan tombol reservasi untuk menambah reservasi yang kemudian akan diarahkan oleh sistem untuk masuk ke dalam antarmuka formulir tambah reservasi. Lalu jika ingin menambah data *check-in* langsung, pengguna resepsionis dapat menekan tombol *walk in guest check-in* yang kemudian akan diarahkan oleh sistem untuk masuk ke dalam antarmuka formulir registrasi *check-in*. Selain itu, pengguna resepsionis juga dapat mengganti status reservasi tamu seperti mengganti status belum konfirmasi menjadi sudah konfirmasi dengan menekan tombol konfirmasi berwarna hijau, atau mengganti status belum konfirmasi menjadi batal reservasi dengan menekan tombol batal berwarna merah. Pengguna resepsionis juga dapat mengganti status reservasi yang sudah konfirmasi menjadi sudah *check-in* dengan menekan tombol *check-in* berwarna biru yang kemudian akan diarahkan oleh sistem untuk melengkapi formulir registrasi *check-in* tamu.

3. Tambah reservasi

The screenshot displays the 'Form Reservasi' interface for EDOTEL SMKN 3 PAMEKASAN. The interface includes a sidebar menu with options like Dashboard, Reservasi, Guest List, Cari Kamar, Manajemen Kamar, Manajemen Barang, Barang Masuk, Barang Keluar, and Laporan. The main form area contains the following elements:

- Form Fields:**
 - Tanggal Check-In: 06/29/2019
 - Tanggal Check-Out: 06/30/2019
 - Button: **Pilih Kamar**
 - Nama:
 - Nomor Telepon:
 - Catatan:
 - Total: Rp. 320,000
 - Button: **Reservasi**
- Room Selection Grid:**

100 Executive Room	101 Deluxe Room	102 Deluxe Room	103 Superior Room	104 Standart Room	105 Standart Room	106 Standart Room	107 Standart Room	108 Standart Room	109 Standart Room	110 Standart Room	201 Standart Room
202 Standart Room	203 Standart Room	204 Standart Room	205 Standart Room	206 Standart Room							

Gambar 6.11 Implementasi Antarmuka Tambah Reservasi

Gambar 6.11 merupakan hasil implementasi antarmuka tambah reservasi. Halaman tambah reservasi digunakan oleh pengguna resepsionis untuk melakukan tambah reservasi tamu. Ketika ingin melakukan reservasi, pengguna resepsionis dapat mengisi formulir tanggal *check-in* kemudian mengisi formulir tanggal *check-out*, setelah itu tekan tombol pilih kamar maka sistem akan menampilkan daftar kamar. Kamar dengan warna hijau menandakan bahwa

kamar sedang kosong dan dapat dipesan, kamar berwarna merah menunjukkan kamar sedang tidak tersedia atau sudah dipesan orang lain, dan kamar berwarna biru merupakan kamar yang dipilih tamu. Setelah memilih kamar, pengguna resepsionis wajib mengisi formulir identitas nama dan nomor telepon, sedangkan formulir catatan merupakan formulir yang kondisional. Setelah selesai mengisi formulir, pengguna resepsionis dapat menekan tombol reservasi untuk menambah data reservasi.

4. Tambah registrasi *check-in*

The screenshot shows the 'Form Registrasi' interface for EDOTEL PAMEKASAN. The form is titled 'Form Registrasi' and includes a 'Kembali' link. It features two date pickers for 'Tanggal Check-In' (06/14/2019) and 'Tanggal Check-Out' (06/16/2019). A room selection button shows '104 Standart Room'. The form includes fields for 'Jenis ID' (with a hint 'cth: KTP/SIM'), 'Nomor ID', 'Nama' (filled with 'Raeda'), 'Alamat', 'Nomor Telepon' (filled with '081234567'), and 'Catatan'. A 'Total' of 'Rp. 320.000' is displayed at the bottom left, and a green 'Check-in' button is at the bottom right. A large watermark of Universitas Brawijaya is overlaid on the form.

Gambar 6.12 Implementasi Antarmuka Tambah Registrasi *Check-In*

Gambar 6.12 merupakan hasil implementasi antarmuka tambah registrasi *check-in*. Halaman tambah registrasi *check-in* digunakan oleh pengguna resepsionis untuk menambah *check-in* tamu. Cara mengisi formulir registrasi *check-in* sama dengan cara mengisi form reservasi. Yang membedakan hanya pada tambahan formulir jenis kartu identitas dan nomor identitas yang nantinya digunakan sebagai jaminan tamu saat menginap.

5. Beranda *guest list*

EDOTEL PAMEKASAN

Logout

Dashboard

Reservasi

Guest List

Cari Kamar

Manajemen Kamar

Manajemen Barang

Barang Masuk

Barang Keluar

Laporan

Guest List

Tamu Sedang Check-In

Show 10 entries Search:

#	ID	Nama	Nomor Kamar	Tanggal Check-In	Tanggal Check-Out	Status	Aksi
1	KTP - 35223132313123	Ardi Sukma	100	2019-06-14	2019-06-15	Check-In	View

Showing 1 to 1 of 1 entries Previous 1 Next

Tamu Sudah Check-Out

Show 10 entries Search:

#	ID	Nama	Nomor Kamar	Tanggal Check-In	Tanggal Check-Out	Status	Aksi
1	KTP - 3521906970001	Ardi	105	2019-05-22	2019-05-23	Check-Out	View
2	KTM - 155150400111066	Zulindra	104	2019-05-22	2019-05-24	Check-Out	View
3	KTP - 12345678	A	201	2019-05-23	2019-05-24	Check-Out	View
4	KTP - 54738290	Radea	100	2019-05-30	2019-06-02	Check-Out	View

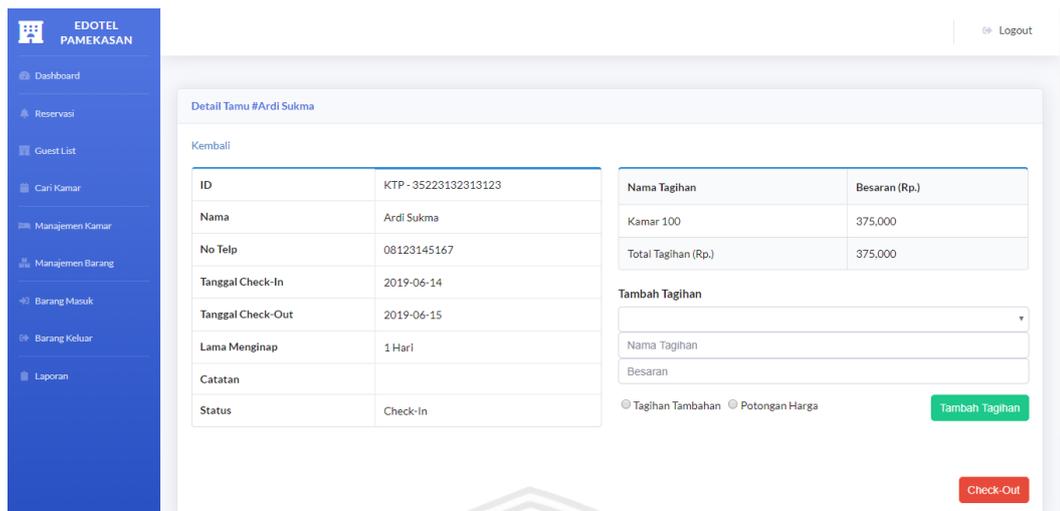
Showing 1 to 4 of 4 entries Previous 1 Next

Copyright © EdoTEL Pamekasan 2019

Gambar 6.13 Implementasi Antarmuka Beranda *Guest List*

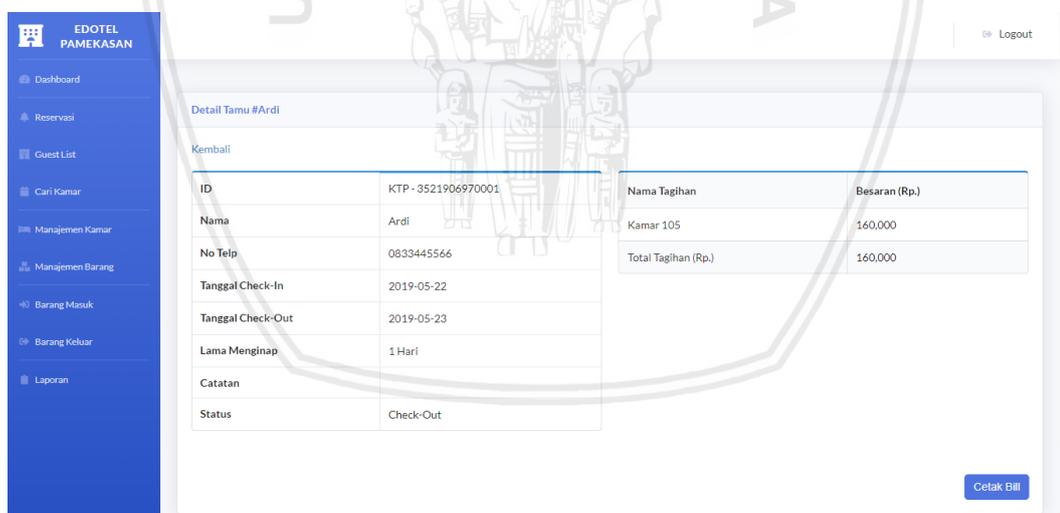
Gambar 6.13 merupakan hasil implementasi antarmuka beranda *guest list*. Halaman beranda *guest list* digunakan oleh pengguna resepsionis untuk mengelola daftar tamu yang menginap di hotel. Pada halaman beranda *guest list* terdapat tabel yang berisi informasi mengenai tamu yang sedang menginap di hotel dan tamu yang pernah menginap. Tabel berisi informasi identitas tamu, nama tamu, nomor kamarnya, tanggal menginap, status, dan aksi. Untuk melihat informasi yang lebih detail mengenai tamu, pengguna resepsionis dapat menekan tombol *view* yang ada di kolom aksi.

6. Detail tamu



Gambar 6.14 Implementasi Antarmuka Detail Tamu

Gambar 6.14 merupakan hasil implementasi antarmuka detail tamu. Halaman detail tamu digunakan oleh pengguna resepsionis untuk mendapatkan informasi lebih detail mengenai tamu yang sedang menginap. Halaman detail tamu juga digunakan untuk mengelola tagihan tamu yang menginap di hotel, mulai dari menambah tagihan, memberikan diskon, beserta menghapus tagihan. Proses *check-out* tamu juga dilakukan pada halaman detail tamu hanya dengan menekan 1 tombol *check-out*.

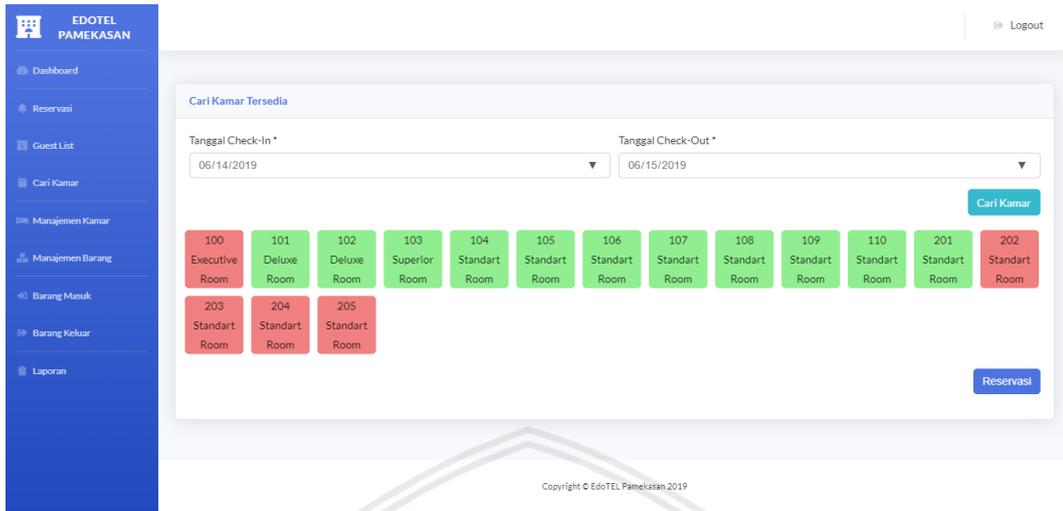


Gambar 6.15 Implementasi Antarmuka Detail Tamu yang Sudah *Check-Out*

Gambar 6.15 merupakan hasil implementasi antarmuka detail tamu yang sudah *check-out*. Halaman detail tamu yang sudah *check-out* digunakan oleh pengguna resepsionis untuk mendapatkan informasi mendetail mengenai tamu hotel dan juga untuk mencetak *bill*. Halaman detail tamu yang sudah *check-out* muncul ketika status tamu sudah melakukan *check-out*. Pada kondisi ini pengguna resepsionis sudah tidak dapat melakukan kegiatan kelola tagihan tamu, sehingga satu – satunya hal yang dapat dilakukan adalah mencetak *bill*. Dan untuk dapat melakukan cetak *bill*, tamu harus melakukan *check-out* terlebih dahulu.



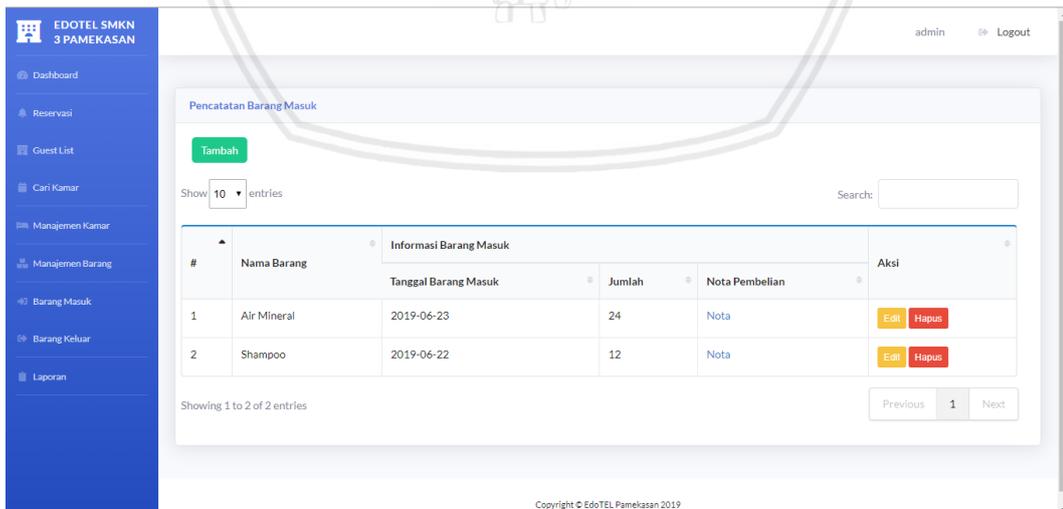
7. Cari kamar



Gambar 6.16 Implementasi Antarmuka Cari Kamar

Gambar 6.16 merupakan hasil implementasi antarmuka cari kamar. Halaman cari kamar digunakan oleh pengguna resepsionis untuk mengetahui status – status kamar. Kamar dengan warna merah menunjukkan kamar tersebut tidak dapat dipesan, bisa jadi karena sudah dipesan orang lain, atau karena kamar sedang dalam perawatan. Sedangkan kamar berwarna hijau menandakan bahwa kamar tersebut dapat dipesan oleh tamu. Untuk menampilkan daftar kamar beserta statusnya, pengguna resepsionis perlu mengisi formulir tanggal *check-in* kemudian dilanjutkan dengan mengisi formulir tanggal *check-out*, kemudian menekan tombol cari kamar. Setelah itu, tunggu sistem menampilkan daftar kamar beserta status kamarnya.

8. Beranda barang masuk

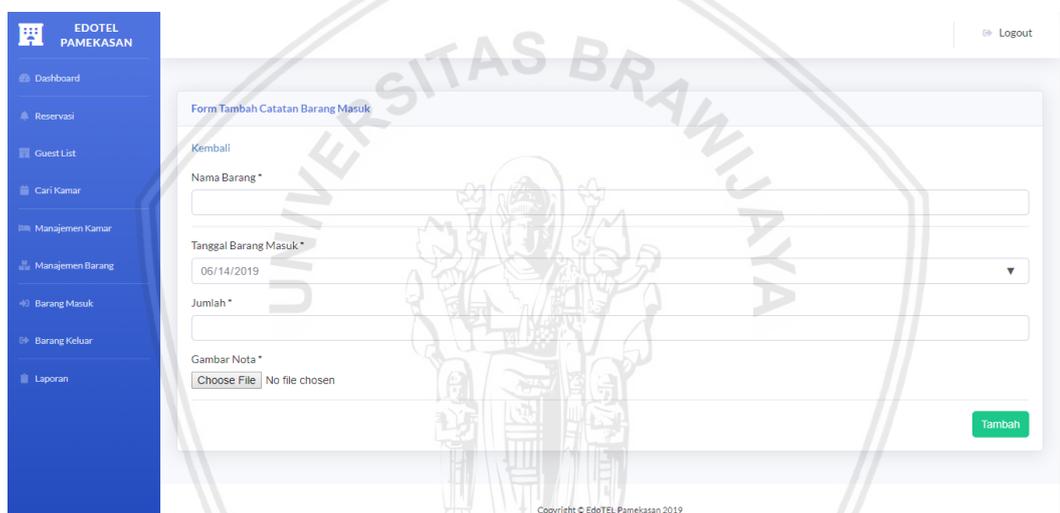


Gambar 6.17 Implementasi Antarmuka Beranda Barang Masuk

Gambar 6.17 merupakan hasil implementasi antarmuka beranda barang masuk. Halaman beranda barang masuk digunakan oleh pengguna *store keeper*

untuk mengelola pencatatan barang masuk. Pada halaman beranda barang masuk terdapat tabel yang berisi informasi pencatatan barang masuk yang sudah dilakukan. Informasi tersebut terdiri dari nama barang masuk, tanggal masuknya, jumlah, foto nota, dan aksi. Selain itu, pada halaman beranda barang masuk pengguna dapat melakukan tambah catatan barang masuk dengan cara menekan tombol tambah yang kemudian akan diarahkan oleh sistem menuju halaman tambah barang masuk. Untuk melakukan kegiatan perbaikan catatan barang masuk, pengguna dapat menekan tombol edit yang berada di kolom aksi. Dan untuk melakukan kegiatan menghapus catatan barang masuk, pengguna dapat menekan tombol hapus yang berada di kolom aksi. Halaman beranda barang masuk memiliki tampilan dan perintah yang sejenis dengan halaman beranda barang keluar, yang membedakannya hanyalah data – data yang dimiliki oleh barang masuk dengan barang keluar.

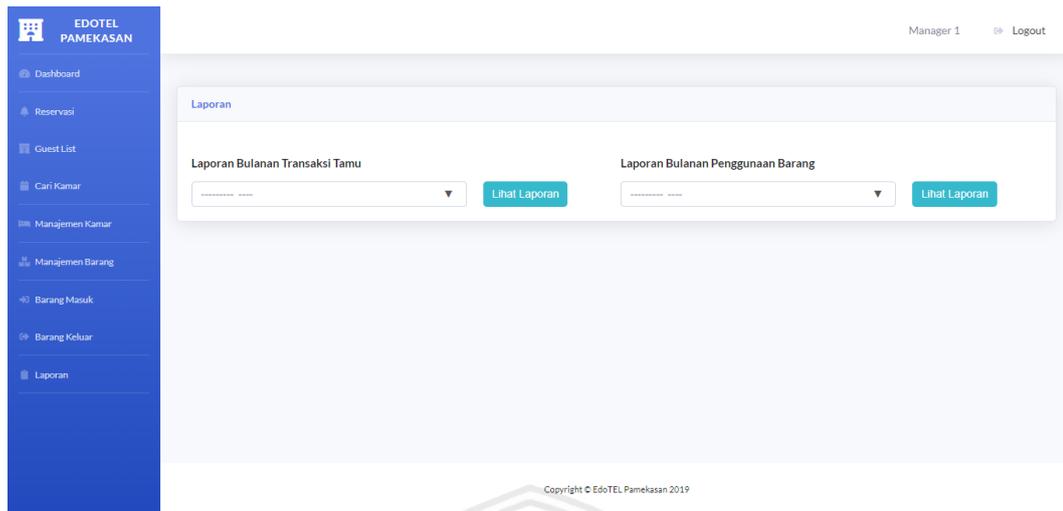
9. Tambah barang masuk



Gambar 6.18 Implementasi Antarmuka Tambah Barang Masuk

Gambar 6.18 merupakan hasil implementasi antarmuka tambah barang masuk. Halaman tambah barang masuk digunakan oleh pengguna untuk menambah catatan barang masuk. Untuk dapat menambah barang masuk, pengguna dapat mengisi formulir nama barang, tanggal barang masuk, jumlah barang yang akan dimasukkan ke gudang, dan bukti foto nota pembeliannya. Setelah mengisi formulir, pengguna kemudian menekan tombol tambah untuk menyimpan tambahan catatan barang masuk. Halaman tambah barang masuk memiliki tampilan dan perintah yang sejenis dengan halaman tambah barang keluar, yang membedakannya hanyalah data – data yang harus diisi pada formulir.

10. Beranda Laporan



Gambar 6.19 Implementasi Antarmuka Beranda Laporan

Gambar 6.19 merupakan hasil implementasi antarmuka beranda laporan. Halaman beranda laporan digunakan oleh pengguna untuk melihat dan mencetak laporan. Terdapat 2 jenis laporan bulanan pada halaman beranda laporan. Laporan pertama merupakan laporan bulanan transaksi tamu, sedangkan laporan lainnya adalah laporan bulanan penggunaan barang. Untuk dapat melihat laporan bulanan, pengguna manajer harus mengisi formulir yang berisi masukan bulan dan tahun laporan. Setelah memilih bulan dan tahun, pengguna manajer menekan tombol lihat laporan. Sistem akan secara otomatis membuat laporan bulanan berdasarkan bulan dan tahun yang diinginkan pengguna. Cara melihat laporan berlaku untuk kedua jenis laporan.

BAB 7 PENGUJIAN DAN DEPLOYMENT

7.1 Pengujian Sistem

Pengujian sistem merupakan tahap terakhir dalam fase *construction*. Pengujian dilakukan berdasarkan analisis kebutuhan dan perancangan yang telah dilakukan. Kemudian, memverifikasinya terhadap implementasi sistem. Pengujian sistem dilakukan menggunakan 2 jenis pengujian. Pengujian kebutuhan fungsional menggunakan metode pengujian *black-box validation testing*. Sedangkan pengujian kebutuhan non-fungsional menggunakan metode pengujian *compatibility testing*.

7.1.1 Pengujian Fungsional Sistem

Kebutuhan fungsional diuji dengan menerapkan metode *black-box validation testing*, yang dilakukan untuk menguji dan menjamin kesesuaian implementasi sistem dengan analisis kebutuhan dan perancangan. Pengujian dilakukan terhadap beberapa fungsi pada sistem, yaitu mengelola reservasi, mengelola registrasi tamu, melihat kamar tersedia, mengelola barang masuk, mengelola barang keluar, dan melihat laporan dengan menggunakan skenario uji. Skenario uji yang digunakan dalam pengujian ini dibuat berdasarkan alur *use case scenario* yang telah dilakukan pada analisis kebutuhan.

7.1.1.1 Perancangan Pengujian Mengelola Reservasi

Perancangan pengujian mengelola reservasi menjelaskan rancangan pengujian terhadap fungsi yang terlibat dalam kegiatan mengelola reservasi. Perancangan pengujian terdiri dari rancangan skenario dan rancangan kasus uji. Rancangan skenario menjelaskan alur awal dan alur alternatif yang dibuat berdasarkan alur *use case scenario* mengelola reservasi, khususnya menambah data reservasi menggunakan sistem. Rancangan kasus uji menjelaskan tujuan pengujian, prosedur uji, dan hasil yang diharapkan yang dibuat berdasarkan rancangan skenario.

Tabel 7.1 Rancangan Skenario Mengelola Reservasi

No.	Kode Skenario	Alur Awal	Alur Alternatif
1	SKVT-01	<i>Main Flow</i>	-
2	SKVT-02	<i>Main Flow</i>	A1. Formulir isian kosong

Tabel 7.1 menjelaskan skenario mengelola reservasi. Terdapat 2 skenario dengan kode SKVT-01 dan SKVT-02. Skenario SKVT-01 merupakan skenario berhasil menambah reservasi. Sedangkan skenario SKVT-02 merupakan skenario gagal menambah reservasi karena ada formulir yang kosong atau tidak diisi.

Tabel 7.2 Rancangan Kasus Uji Menambah Reservasi

Kode Pengujian	PVT-01
Kode Skenario	SKVT-01
Kode Kebutuhan Fungsional	FGSSI-03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data reservasi, menampilkan formulir reservasi, dan menyimpan data reservasi dengan berhasil menambah reservasi
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah reservasi 2. Penguji mengisi data formulir reservasi yang terdiri dari nama, nomor telpon yang dapat dihubungi, catatan tambahan (jika ada), tanggal <i>check-in</i>, tanggal <i>check-out</i>, dan pilihan nomor kamar 3. Penguji memilih simpan reservasi
Hasil yang Diharapkan	Sistem berhasil menampilkan formulir reservasi, menyimpan reservasi, dan menampilkan data reservasi yang telah disimpan di dalam sistem

Tabel 7.2 menjelaskan kasus uji menambah reservasi dengan kode PVT-01. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data reservasi, menampilkan formulir reservasi, dan menyimpan data reservasi dengan berhasil menambah reservasi. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan formulir reservasi, menyimpan reservasi, dan menampilkan data reservasi yang telah disimpan di dalam sistem.

Tabel 7.3 Rancangan Kasus Uji Gagal Menambah Reservasi

Kode Skenario	SKVT-02
Kode Kebutuhan Fungsional	FGSSI-03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data reservasi, menampilkan formulir reservasi, namun tidak dapat menyimpan data reservasi dan sistem menampilkan pesan <i>error</i> formulir wajib diisi karena terdapat formulir isian dengan <i>mandatory</i> tanda * yang tidak terisi
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah reservasi

Tabel 7.3 Rancangan Kasus Uji Gagal Menambah Reservasi (lanjutan)

	<p>2. Penguji mengisi data formulir reservasi yang terdiri dari nama, nomor telpon yang dapat dihubungi, catatan tambahan (jika ada), tanggal <i>check-in</i>, tanggal <i>check-out</i>, dan pilihan nomor kamar namun mengosongkan salah satu isian formulir</p> <p>3. Penguji memilih simpan reservasi</p>
Hasil yang Diharapkan	Sistem berhasil menampilkan pesan <i>error</i> untuk melengkapi formulir yang kosong

Tabel 7.3 menjelaskan kasus uji gagal menambah reservasi dengan kode PVT-02. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data reservasi, menampilkan formulir reservasi, namun tidak dapat menyimpan data reservasi dan sistem menampilkan pesan *error* formulir wajib diisi karena terdapat formulir isian dengan *mandatory* tanda * yang tidak terisi. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan pesan *error* untuk melengkapi formulir yang kosong.

7.1.1.2 Perancangan Pengujian Mengelola Registrasi Tamu

Perancangan pengujian mengelola registrasi tamu menjelaskan rancangan pengujian terhadap fungsi yang terlibat dalam kegiatan mengelola registrasi tamu. Perancangan pengujian terdiri dari rancangan skenario dan rancangan kasus uji. Rancangan skenario menjelaskan alur awal dan alur alternatif yang dibuat berdasarkan alur *use case scenario* mengelola registrasi tamu, khususnya menambah data registrasi *check-in* dan menambah tagihan tamu menggunakan sistem. Rancangan kasus uji menjelaskan tujuan pengujian, prosedur uji, dan hasil yang diharapkan yang dibuat berdasarkan rancangan skenario.

Tabel 7.4 Rancangan Skenario Mengelola Registrasi Tamu

No.	Kode Skenario	Alur Awal	Alur Alternatif
1	SKVT-03	<i>Main Flow</i>	-
2	SKVT-04	<i>Main Flow</i>	A1. Formulir isian kosong
3	SKVT-05	<i>Main Flow</i>	S2. Menambah Tagihan Tamu
4	SKVT-06	<i>Main Flow</i>	A2. Formulir isian kosong pada <i>sub flow</i> Menambah Tagihan Tamu

Tabel 7.4 menjelaskan skenario mengelola registrasi tamu. Terdapat 4 skenario dengan kode SKVT-03, SKVT-04, SKVT-05, dan SKVT-06. Skenario SKVT-03 merupakan skenario berhasil menambah registrasi *check-in*. Skenario SKVT-04 merupakan skenario gagal menambah registrasi *check-in* karena ada formulir yang kosong atau tidak diisi. Skenario SKVT-05 merupakan skenario berhasil menambah

tagihan tamu. Skenario SKVT-06 merupakan skenario gagal menambah tagihan tamu karena ada formulir yang kosong atau tidak diisi.

Tabel 7.5 Rancangan Kasus Uji Menambah Registrasi *Check-In*

Kode Pengujian	PVT-03
Kode Skenario	SKVT-03
Kode Kebutuhan Fungsional	FGSSI-09
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data tamu yang sudah <i>check-in</i> , menampilkan formulir data registrasi <i>check-in</i> , dan menyimpan data registrasi <i>check-in</i> dengan berhasil menambah registrasi <i>check-in</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah registrasi <i>check-in</i> 2. Penguji mengisi data formulir registrasi terdiri dari jenis identitas, nomor identitas, nama, alamat, nomor telpon yang dapat dihubungi, catatan tambahan (jika ada), tanggal <i>check-in</i>, tanggal <i>check-out</i>, dan memilih kamar 3. Penguji memilih simpan registrasi <i>check-in</i>
Hasil yang Diharapkan	Sistem berhasil menampilkan formulir registrasi <i>check-in</i> , menyimpan registrasi <i>check-in</i> , dan menampilkan data tamu yang sudah <i>check-in</i> yang telah disimpan di dalam sistem

Tabel 7.5 menjelaskan kasus uji menambah registrasi *check-in* dengan kode PVT-03. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data tamu yang sudah *check-in*, menampilkan formulir data registrasi *check-in*, dan menyimpan data registrasi *check-in* dengan berhasil menambah registrasi *check-in*. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan formulir registrasi *check-in*, menyimpan registrasi *check-in*, dan menampilkan data tamu yang sudah *check-in* yang telah disimpan di dalam sistem.

Tabel 7.6 Rancangan Kasus Uji Gagal Menambah Registrasi *Check-In*

Kode Pengujian	PVT-04
Kode Skenario	SKVT-04

Tabel 7.6 Rancangan Kasus Uji Gagal Menambah Registrasi *Check-In* (lanjutan)

Kode Kebutuhan Fungsional	FGSSI-09
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data tamu yang sudah <i>check-in</i> , menampilkan formulir data registrasi <i>check-in</i> , namun tidak dapat menyimpan data registrasi <i>check-in</i> sehingga sistem menampilkan pesan <i>error</i> formulir wajib diisi karena terdapat formulir isian dengan <i>mandatory</i> tanda * yang tidak terisi
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah registrasi <i>check-in</i> 2. Penguji mengisi data formulir registrasi terdiri dari jenis identitas, nomor identitas, nama, alamat, nomor telpon yang dapat dihubungi, catatan tambahan (jika ada), tanggal <i>check-in</i>, tanggal <i>check-out</i>, dan memilih kamar namun mengosongkan salah satu isian formulir 3. Penguji memilih simpan registrasi <i>check-in</i>
Hasil yang Diharapkan	Sistem berhasil menampilkan pesan <i>error</i> untuk melengkapi formulir yang kosong

Tabel 7.6 menjelaskan kasus uji gagal menambah registrasi *check-in* dengan kode PVT-04. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data tamu yang sudah *check-in*, menampilkan formulir data registrasi *check-in*, namun tidak dapat menyimpan data registrasi *check-in* sehingga sistem menampilkan pesan *error* formulir wajib diisi karena terdapat formulir isian dengan *mandatory* tanda * yang tidak terisi. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan pesan *error* untuk melengkapi formulir yang kosong.

Tabel 7.7 Rancangan Kasus Uji Menambah Tagihan Tamu

Kode Pengujian	PVT-05
Kode Skenario	SKVT-05
Kode Kebutuhan Fungsional	FGSSI-12
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data detail tamu, dan menyimpan tagihan tamu
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih menu <i>guest list</i> 2. Penguji memilih lihat detail tamu 3. Penguji melihat data detail tamu

Tabel 7.7 Rancangan Kasus Uji Menambah Tagihan Tamu (lanjutan)

	<ol style="list-style-type: none"> 4. Penguji mengisi data formulir tagihan yang terdiri dari nomor kamar, nama tagihan, besaran tagihan, dan jenis tagihan 5. Penguji memilih simpan tagihan
Hasil yang Diharapkan	Sistem berhasil menampilkan data detail tamu, dan menyimpan tagihan tamu

Tabel 7.7 menjelaskan kasus uji menambah tagihan tamu dengan kode PVT-05. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data detail tamu, dan menyimpan tagihan tamu. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan data detail tamu, dan menyimpan tagihan tamu.

Tabel 7.8 Rancangan Kasus Uji Gagal Menambah Tagihan Tamu

Kode Pengujian	PVT-06
Kode Skenario	SKVT-06
Kode Kebutuhan Fungsional	FGSSI-12
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data detail tamu, menambah tagihan tamu, namun tidak dapat menyimpan tagihan tamu sehingga sistem menampilkan pesan error formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih menu <i>guest list</i> 2. Penguji memilih lihat detail tamu 3. Penguji melihat data detail tamu 4. Penguji mengisi data formulir tagihan yang terdiri dari nomor kamar, nama tagihan, besaran tagihan, dan jenis tagihan namun mengosongkan salah satu isian formulir 5. Penguji memilih simpan tagihan
Hasil yang Diharapkan	Sistem berhasil menampilkan pesan <i>error</i> untuk melengkapi formulir yang kosong

Tabel 7.8 menjelaskan kasus uji gagal menambah tagihan tamu dengan kode PVT-06. Tujuan pengujian adalah Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data detail tamu, menambah tagihan tamu, namun tidak dapat menyimpan tagihan tamu sehingga sistem menampilkan pesan error

formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan pesan *error* untuk melengkapi formulir yang kosong.

7.1.1.3 Perancangan Pengujian Melihat Kamar Tersedia

Perancangan pengujian melihat kamar tersedia menjelaskan rancangan pengujian terhadap fungsi yang terlibat dalam kegiatan melihat kamar tersedia. Perancangan pengujian terdiri dari rancangan skenario dan rancangan kasus uji. Rancangan skenario menjelaskan alur awal dan alur alternatif yang dibuat berdasarkan alur *use case scenario* melihat kamar tersedia menggunakan sistem. Rancangan kasus uji menjelaskan tujuan pengujian, prosedur uji, dan hasil yang diharapkan yang dibuat berdasarkan rancangan skenario.

Tabel 7.9 Rancangan Skenario Melihat Kamar Tersedia

No.	Kode Skenario	Alur Awal	Alur Alternatif
1	SKVT-07	Main Flow	-
2	SKVT-08	Main Flow	A1. Formulir isian kosong

Tabel 7.9 menjelaskan skenario melihat kamar tersedia. Terdapat 2 skenario dengan kode SKVT-07 dan SKVT-08. Skenario SKVT-07 merupakan skenario berhasil melihat kamar tersedia. Skenario SKVT-08 merupakan skenario gagal melihat kamar tersedia karena ada formulir yang kosong atau tidak diisi.

Tabel 7.10 Rancangan Kasus Uji Melihat Kamar Tersedia

Kode Pengujian	PVT-07
Kode Skenario	SKVT-07
Kode Kebutuhan Fungsional	FGSSI-16
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data kamar yang tersedia untuk disewa tamu hotel berdasarkan masukan tanggal <i>check-in</i> dan <i>tanggal check-out</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi formulir <i>tanggal check-in</i> dan <i>tanggal check-out</i> 2. Penguji memilih lihat kamar 3. Penguji melihat data kamar tersedia
Hasil yang Diharapkan	Sistem berhasil menampilkan data kamar yang tersedia untuk disewa tamu

Tabel 7.10 menjelaskan kasus uji melihat kamar tersedia dengan kode PVT-07. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data kamar yang tersedia untuk disewa tamu hotel berdasarkan masukan tanggal *check-in* dan *tanggal check-out*. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan data kamar yang tersedia untuk disewa tamu.

Tabel 7.11 Rancangan Kasus Uji Gagal Melihat Kamar Tersedia

Kode Pengujian	PVT-08
Kode Skenario	SKVT-08
Kode Kebutuhan Fungsional	FGSSI-16
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji tidak dapat menampilkan data kamar yang tersedia untuk disewa tamu hotel berdasarkan masukan tanggal <i>check-in</i> dan tanggal <i>check-out</i> dan sistem menampilkan pesan <i>error</i> formulir wajib diisi karena terdapat formulir isian dengan <i>mandatory</i> tanda * yang tidak terisi
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi formulir tanggal <i>check-in</i> dan tanggal <i>check-out</i> namun mengosongkan salah satu isian formulir 2. Penguji memilih lihat kamar 3. Penguji melihat data kamar tersedia
Hasil yang Diharapkan	Sistem berhasil menampilkan pesan <i>error</i> untuk melengkapi formulir yang kosong

Tabel 7.11 menjelaskan kasus uji gagal melihat kamar tersedia dengan kode PVT-08. Tujuan pengujian adalah untuk memastikan bahwa penguji tidak dapat menampilkan data kamar yang tersedia untuk disewa tamu hotel berdasarkan masukan tanggal *check-in* dan tanggal *check-out* dan sistem menampilkan pesan *error* formulir wajib diisi karena terdapat formulir isian dengan *mandatory* tanda * yang tidak terisi. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan pesan *error* untuk melengkapi formulir yang kosong.

7.1.1.4 Perancangan Pengujian Mengelola Barang Masuk

Perancangan pengujian mengelola barang masuk menjelaskan rancangan pengujian terhadap fungsi yang terlibat dalam kegiatan mengelola barang masuk. Perancangan pengujian terdiri dari rancangan skenario dan rancangan kasus uji. Rancangan skenario menjelaskan alur awal dan alur alternatif yang dibuat berdasarkan alur *use case scenario* mengelola barang masuk, khususnya menambah data barang masuk menggunakan sistem. Rancangan kasus uji

menjelaskan tujuan pengujian, prosedur uji, dan hasil yang diharapkan yang dibuat berdasarkan rancangan skenario.

Tabel 7.12 Rancangan Skenario Mengelola Barang Masuk

No.	Kode Skenario	Alur Awal	Alur Alternatif
1	SKVT-09	Main Flow	-
2	SKVT-10	Main Flow	A1. Formulir isian kosong
3	SKVT-11	Main Flow	A2. Formulir isian nama barang tidak sesuai

Tabel 7.12 menjelaskan skenario mengelola barang masuk. Terdapat 3 skenario dengan kode SKVT-09, SKVT-10, dan SKVT-11. Skenario SKVT-09 merupakan skenario berhasil menambah barang masuk. Skenario SKVT-10 merupakan skenario gagal menambah barang masuk karena ada formulir yang kosong atau tidak diisi. Skenario SKVT-11 merupakan skenario gagal menambah barang masuk karena isian formulir nama barang yang tidak sesuai.

Tabel 7.13 Rancangan Kasus Uji Menambah Barang Masuk

Kode Pengujian	PVT-09
Kode Skenario	SKVT-09
Kode Kebutuhan Fungsional	FGSSI-19
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, dan menyimpan data barang masuk dengan berhasil menambah barang masuk
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah data barang masuk 2. Penguji mengisi data formulir barang masuk yang terdiri dari nama barang, tanggal barang masuk, jumlah barang, dan foto nota 3. Penguji memilih simpan data barang masuk
Hasil yang Diharapkan	Sistem berhasil menampilkan formulir barang masuk, menyimpan data barang masuk, dan menampilkan data barang masuk yang telah disimpan di dalam sistem

Tabel 7.13 menjelaskan kasus uji menambah barang masuk dengan kode PVT-09. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, dan menyimpan data barang masuk dengan berhasil melakukan penambahan barang masuk. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan

formulir barang masuk, menyimpan data barang masuk, dan menampilkan data barang masuk yang telah disimpan di dalam sistem.

Tabel 7.14 Rancangan Kasus Uji Gagal Menambah Barang Masuk

Kode Pengujian	PVT-10
Kode Skenario	SKVT-10
Kode Kebutuhan Fungsional	FGSSI-19
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, namun tidak dapat menyimpan data barang masuk sehingga sistem menampilkan pesan error formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah data barang masuk 2. Penguji mengisi data formulir barang masuk yang terdiri dari nama barang, tanggal barang masuk, jumlah barang, dan foto nota namun mengosongkan salah satu isian formulir 3. Penguji memilih simpan data barang masuk
Hasil yang Diharapkan	Sistem berhasil menampilkan pesan <i>error</i> untuk melengkapi formulir yang kosong

Tabel 7.14 menjelaskan kasus uji gagal menambah barang masuk dengan kode PVT-10. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, namun tidak dapat menyimpan data barang masuk sehingga sistem menampilkan pesan error formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan pesan *error* untuk melengkapi formulir yang kosong.

Tabel 7.15 Rancangan Kasus Uji Gagal Menambah Barang Masuk

Kode Pengujian	PVT-11
Kode Skenario	SKVT-11

Tabel 7.15 Rancangan Kasus Uji Gagal Menambah Barang Masuk (lanjutan)

Kode Kebutuhan Fungsional	FGSSI-19
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, namun tidak dapat menyimpan data barang masuk sehingga sistem menampilkan pesan error nama barang salah karena formulir nama barang diisi tidak sesuai dengan nama barang yang ada pada daftar nama barang
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah data barang masuk 2. Penguji mengisi data formulir barang masuk yang terdiri dari nama barang dengan nama barang diluar daftar nama barang yang ada, tanggal barang masuk, jumlah barang, dan foto nota 3. Penguji memilih simpan data barang masuk
Hasil yang Diharapkan	Sistem berhasil menampilkan pesan <i>error</i> untuk mengganti nama barang sesuai dengan nama barang yang ada pada daftar nama barang

Tabel 7.15 menjelaskan kasus uji gagal menambah barang masuk dengan kode PVT-11. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, namun tidak dapat menyimpan data barang masuk sehingga sistem menampilkan pesan error nama barang salah karena formulir nama barang diisi tidak sesuai dengan nama barang yang ada pada daftar nama barang. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan pesan *error* untuk mengganti nama barang sesuai dengan nama barang yang ada pada daftar nama barang.

7.1.1.5 Perancangan Pengujian Mengelola Barang Keluar

Perancangan pengujian mengelola barang keluar menjelaskan rancangan pengujian terhadap fungsi yang terlibat dalam kegiatan mengelola barang keluar. Perancangan pengujian terdiri dari rancangan skenario dan rancangan kasus uji. Rancangan skenario menjelaskan alur awal dan alur alternatif yang dibuat berdasarkan alur *use case scenario* mengelola barang keluar, khususnya menambah data barang keluar menggunakan sistem. Rancangan kasus uji menjelaskan tujuan pengujian, prosedur uji, dan hasil yang diharapkan yang dibuat berdasarkan rancangan skenario.

Tabel 7.16 Rancangan Skenario Mengelola Barang Keluar

No.	Kode Skenario	Alur Awal	Alur Alternatif
1	SKVT-12	<i>Main Flow</i>	-

Tabel 7.16 Rancangan Skenario Mengelola Barang Keluar (lanjutan)

No.	Kode Skenario	Alur Awal	Alur Alternatif
2	SKVT-13	Main Flow	A3. Formulir isian jumlah barang keluar tidak sesuai

Tabel 7.16 menjelaskan skenario mengelola barang keluar. Terdapat 2 skenario dengan kode SKVT-12 dan SKVT-13. Skenario SKVT-12 merupakan skenario berhasil menambah barang keluar. Skenario SKVT-13 merupakan skenario gagal menambah barang keluar karena isian formulir jumlah barang keluar yang tidak sesuai.

Tabel 7.17 Rancangan Kasus Uji Menambah Barang Keluar

Kode Pengujian	PVT-12
Kode Skenario	SKVT-12
Kode Kebutuhan Fungsional	FGSSI-24
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang keluar, menampilkan formulir barang keluar, dan menyimpan data barang keluar dengan berhasil menambah barang keluar
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah data barang keluar 2. Penguji mengisi data formulir barang keluar yang terdiri dari nama barang, tanggal barang keluar, jumlah barang, nama pegawai, dan tujuan pengambilan barang 3. Penguji memilih simpan data barang keluar
Hasil yang Diharapkan	Sistem berhasil menampilkan formulir barang keluar, menyimpan data barang keluar, dan menampilkan data barang keluar yang telah disimpan di dalam sistem

Tabel 7.17 menjelaskan kasus uji menambah barang keluar dengan kode PVT-12. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data barang keluar, menampilkan formulir barang keluar, dan menyimpan data barang keluar dengan berhasil menambah barang keluar. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan formulir barang keluar, menyimpan data barang keluar, dan menampilkan data barang keluar yang telah disimpan di dalam sistem.

Tabel 7.18 Rancangan Kasus Uji Gagal Menambah Barang Keluar

Kode Pengujian	PVT-13
Kode Skenario	SKVT-13
Kode Kebutuhan Fungsional	FGSSI-24
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang keluar, menampilkan formulir barang keluar, namun tidak dapat menyimpan data barang keluar sehingga sistem menampilkan pesan <i>error</i> jumlah barang keluar salah karena formulir jumlah diisi melebihi jumlah barang yang ada
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji memilih tambah data barang keluar 2. Penguji mengisi data formulir barang keluar yang terdiri dari nama barang, tanggal barang keluar, jumlah barang melebihi jumlah barang yang ada, nama pegawai, dan tujuan pengambilan barang 3. Penguji memilih simpan data barang keluar
Hasil yang Diharapkan	Sistem berhasil menampilkan pesan <i>error</i> untuk mengganti jumlah barang keluar tidak lebih dari atau sama dengan jumlah barang yang ada

Tabel 7.18 menjelaskan kasus uji gagal menambah barang keluar dengan kode PVT-13. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan data barang keluar, menampilkan formulir barang keluar, namun tidak dapat menyimpan data barang keluar sehingga sistem menampilkan pesan *error* jumlah barang keluar salah karena formulir jumlah diisi melebihi jumlah barang yang ada. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan pesan *error* untuk mengganti jumlah barang keluar tidak lebih dari atau sama dengan jumlah barang yang ada.

7.1.1.6 Perancangan Pengujian Melihat Laporan

Perancangan pengujian melihat laporan menjelaskan rancangan pengujian terhadap fungsi yang terlibat dalam kegiatan melihat laporan. Perancangan pengujian terdiri dari rancangan skenario dan rancangan kasus uji. Rancangan skenario menjelaskan alur awal dan alur alternatif yang dibuat berdasarkan alur *use case scenario* melihat laporan, khususnya melihat laporan bulanan transaksi tamu menggunakan sistem. Rancangan kasus uji menjelaskan tujuan pengujian,

prosedur uji, dan hasil yang diharapkan yang dibuat berdasarkan rancangan skenario.

Tabel 7.19 Rancangan Skenario Melihat Laporan

No.	Kode Skenario	Alur Awal	Alur Alternatif
1	SKVT-14	Main Flow	-
2	SKVT-15	Main Flow	A1. Formulir isian kosong

Tabel 7.19 menjelaskan skenario melihat laporan. Terdapat 2 skenario dengan kode SKVT-14 dan SKVT-15. Skenario SKVT-14 merupakan skenario berhasil melihat laporan. Skenario SKVT-15 merupakan skenario gagal melihat laporan karena ada formulir yang kosong atau tidak diisi.

Tabel 7.20 Rancangan Pengujian Melihat Laporan

Kode Pengujian	PVT-14
Kode Skenario	SKVT-14
Kode Kebutuhan Fungsional	FGSSI-28
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan laporan bulanan transaksi tamu hotel
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi formulir bulan dan tahun 2. Penguji memilih lihat laporan bulanan transaksi tamu hotel 3. Penguji melihat laporan bulanan transaksi tamu
Hasil yang Diharapkan	Sistem berhasil menampilkan laporan bulanan transaksi tamu hotel

Tabel 7.20 menjelaskan kasus uji melihat laporan dengan kode PVT-14. Tujuan pengujian adalah untuk memastikan bahwa penguji dapat menampilkan laporan bulanan transaksi tamu hotel. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan laporan bulanan transaksi tamu hotel.

Tabel 7.21 Rancangan Pengujian Gagal Melihat Laporan

Kode Pengujian	PVT-15
Kode Skenario	SKVT-15
Kode Kebutuhan Fungsional	FGSSI-28

Tabel 7.21 Rancangan Pengujian Gagal Melihat Laporan (lanjutan)

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa penguji tidak dapat menampilkan laporan bulanan transaksi tamu hotel dan sistem menampilkan pesan error formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi formulir bulan dan tahun namun mengosongkan salah satu field form 2. Penguji memilih lihat laporan bulanan transaksi tamu hotel 3. Penguji melihat laporan bulanan transaksi tamu
Hasil yang Diharapkan	Sistem berhasil menampilkan pesan <i>error</i> untuk melengkapi formulir yang kosong

Tabel 7.21 menjelaskan kasus uji gagal melihat laporan dengan kode PVT-15. Tujuan pengujian adalah untuk memastikan bahwa penguji tidak dapat menampilkan laporan bulanan transaksi tamu hotel dan sistem menampilkan pesan error formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi. Hasil yang diharapkan dari pengujian adalah sistem berhasil menampilkan pesan *error* untuk melengkapi formulir yang kosong.

7.1.1.7 Hasil Pengujian Fungsional Sistem

Hasil pengujian fungsional sistem menjelaskan hasil pengujian dari perancangan yang telah dilakukan sebelumnya. Hasil pengujian membahas 15 kasus uji yang telah dirancang sebelumnya yang berasal dari 6 *use case scenario* yang berbeda, yaitu mengelola reservasi, mengelola registrasi tamu, melihat kamar tersedia, mengelola barang masuk, mengelola barang keluar, dan melihat laporan. Hasil pengujian berisi hasil rancangan pengujian yang terdiri dari tujuan pengujian yang telah di definisikan, hasil pengujian, dan status.

Tabel 7.22 Hasil Pengujian Fungsional Sistem

No.	Kode Pengujian	Tujuan Pengujian	Hasil Pengujian	Status
1	PVT-01	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data reservasi, menampilkan formulir reservasi, dan menyimpan data reservasi dengan berhasil menambah reservasi	Sistem berhasil menampilkan formulir reservasi, menyimpan reservasi, dan menampilkan data reservasi yang telah disimpan di dalam sistem	Valid
2	PVT-02	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data reservasi, menampilkan formulir reservasi, namun tidak dapat menyimpan data reservasi dan sistem menampilkan pesan <i>error</i> formulir wajib diisi karena terdapat formulir isian dengan <i>mandatory</i> tanda * yang tidak terisi	Sistem berhasil menampilkan pesan <i>error fill out this field</i> dan belum memilih kamar	Valid
3	PVT-03	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data tamu yang sudah <i>check-in</i> , menampilkan formulir data registrasi <i>check-in</i> , dan menyimpan data registrasi <i>check-in</i> dengan berhasil menambah registrasi <i>check-in</i>	Sistem berhasil menampilkan formulir registrasi <i>check-in</i> , menyimpan registrasi <i>check-in</i> , dan menampilkan data tamu yang sudah <i>check-in</i> yang telah disimpan di dalam sistem	Valid
4	PVT-04	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data tamu yang sudah <i>check-in</i> , menampilkan formulir data registrasi <i>check-in</i> , namun tidak dapat menyimpan data registrasi <i>check-in</i> sehingga sistem menampilkan pesan <i>error</i> formulir wajib diisi karena terdapat formulir isian dengan <i>mandatory</i> tanda * yang tidak terisi	Sistem berhasil menampilkan pesan <i>error fill out this field</i> dan belum memilih kamar	Valid

Tabel 7.22 Hasil Pengujian Fungsional Sistem (lanjutan)

No.	Kode Pengujian	Tujuan Pengujian	Hasil Pengujian	Status
5	PVT-05	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data detail tamu, dan menyimpan tagihan tamu	Sistem berhasil menampilkan data detail tamu, dan menyimpan tagihan tamu	Valid
6	PVT-06	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data detail tamu, menambah tagihan tamu, namun tidak dapat menyimpan tagihan tamu sehingga sistem menampilkan pesan error formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi	Sistem berhasil menampilkan pesan <i>error select an item in the list, fill out this field, select one of this action</i>	Valid
7	PVT-07	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data kamar yang tersedia untuk disewa tamu hotel berdasarkan masukan tanggal <i>check-in</i> dan <i>tanggal check-out</i>	Sistem berhasil menampilkan daftar kamar bisa disewa tamu dan kamar yang tidak bisa disewa tamu	Valid
8	PVT-08	Pengujian dilakukan untuk memastikan bahwa penguji tidak dapat menampilkan data kamar yang tersedia untuk disewa tamu hotel berdasarkan masukan tanggal <i>check-in</i> dan tanggal <i>check-out</i> dan sistem menampilkan pesan <i>error</i> formulir wajib diisi karena terdapat formulir isian dengan <i>mandatory</i> tanda * yang tidak terisi	Sistem berhasil menampilkan pesan <i>error fill out this field</i>	Valid
9	PVT-09	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, dan menyimpan data barang masuk dengan berhasil menambah barang masuk	Sistem berhasil menampilkan formulir barang masuk, menyimpan data barang masuk, dan menampilkan data barang	Valid

Tabel 7.22 Hasil Pengujian Fungsional Sistem (lanjutan)

No.	Kode Pengujian	Tujuan Pengujian	Hasil Pengujian	Status
			masuk yang telah disimpan di dalam sistem	
10	PVT-10	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, namun tidak dapat menyimpan data barang masuk sehingga sistem menampilkan pesan error formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi	Sistem berhasil menampilkan pesan <i>error fill out this field</i>	Valid
11	PVT-11	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang masuk, menampilkan formulir barang masuk, namun tidak dapat menyimpan data barang masuk sehingga sistem menampilkan pesan error nama barang salah karena formulir nama barang diisi tidak sesuai dengan nama barang yang ada pada daftar nama barang	Sistem berhasil menampilkan pesan <i>error select valid value</i>	Valid
12	PVT-12	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang keluar, menampilkan formulir barang keluar, dan menyimpan data barang keluar dengan berhasil menambah barang keluar	Sistem berhasil menampilkan formulir barang keluar, menyimpan data barang keluar, dan menampilkan data barang keluar yang telah disimpan di dalam sistem	Valid
13	PVT-13	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan data barang keluar, menampilkan formulir barang keluar, namun tidak dapat menyimpan data barang keluar	Sistem berhasil menampilkan pesan <i>error value must be less or equal</i>	Valid

Tabel 7.22 Hasil Pengujian Fungsional Sistem (lanjutan)

No.	Kode Pengujian	Tujuan Pengujian	Hasil Pengujian	Status
		sehingga sistem menampilkan pesan <i>error</i> jumlah barang keluar salah karena formulir jumlah diisi melebihi jumlah barang yang ada		
14	PVT-14	Pengujian dilakukan untuk memastikan bahwa penguji dapat menampilkan laporan bulanan transaksi tamu hotel	Sistem berhasil menampilkan laporan bulanan transaksi tamu hotel dalam bentuk PDF	Valid
15	PVT-15	Pengujian dilakukan untuk memastikan bahwa penguji tidak dapat menampilkan laporan bulanan transaksi tamu hotel dan sistem menampilkan pesan error formulir wajib diisi karena terdapat formulir isian dengan mandatory tanda * yang tidak terisi	Sistem berhasil menampilkan pesan <i>error fill out this field</i>	Valid



Tabel 7.22 menjelaskan hasil pengujian dari beberapa fungsi yang ada. Pengujian validasi yang dilakukan pada 15 kasus uji menghasilkan status valid. Status valid didapatkan ketika hasil pengujian sesuai dengan hasil yang diharapkan, yang artinya kinerja sistem sesuai dengan spesifikasi sehingga memenuhi harapan pengguna (Pressman, 2010).

7.1.1.8 Analisis Hasil Pengujian Fungsional Sistem

Analisis hasil pengujian dilakukan untuk mencari persentase keberhasilan pengujian validasi. Persentase keberhasilan pengujian validasi didapatkan dari membagi jumlah kasus uji dengan status valid dengan total kasus uji yang dilakukan, kemudian mengalikannya dengan 100. Hasil perhitungan persentase pengujian validasi adalah:

$$\text{Persentase} = (15/15) \times 100 = 100\% \quad (7.1)$$

Dari 15 kasus uji dengan 6 *use case scenario* berbeda antara lain mengelola reservasi, mengelola registrasi tamu, melihat kamar tersedia, mengelola barang masuk, mengelola barang keluar, dan melihat laporan menunjukkan hasil pengujian status valid dengan persentase 100%. Dapat diartikan bahwa implementasi sistem dapat diterima karena telah sesuai dengan analisis kebutuhan dan perancangannya.

7.1.2 Pengujian Non-Fungsional Sistem

Kebutuhan non-fungsional diuji dengan menggunakan metode *compatibility testing*. *Compatibility testing* dilakukan untuk menguji persyaratan kebutuhan non-fungsional dengan kode NFGSSI-01. Pada *compatibility testing*, sistem akan diukur tingkat kesesuaiannya dengan berbagai peramban (*browser*) yang ada. Pengujian dilakukan menggunakan bantuan aplikasi SortSite 5 Trial.

7.1.2.1 Perancangan Pengujian Non-Fungsional Sistem

Perancangan pengujian non-fungsional menjelaskan rancangan pengujian kompatibilitas. Mulai dari rancangan peramban yang digunakan yang menjelaskan nama peramban yang akan diuji beserta versinya. Juga menjelaskan kategori masalah pada aplikasi SortSite 5 Trial. Daftar peramban yang digunakan untuk menguji sistem ditunjukkan pada Tabel 7.23.

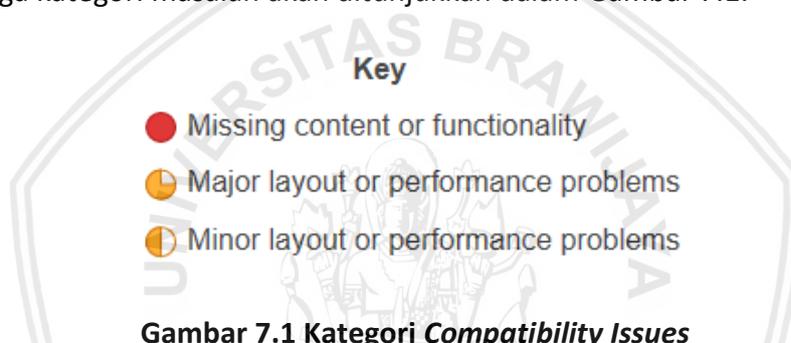
Tabel 7.23 Rancangan Peramban yang Digunakan

No.	Nama Peramban	Versi
1	Internet Explorer	Versi 11
2	Microsoft Edge	Versi 18
3	Mozilla Firefox	Versi 66
4	Safari	Versi 12
5	Opera	Versi 60

Tabel 7.23 Rancangan Peramban yang Digunakan (lanjutan)

No.	Nama Peramban	Versi
6	Google Chrome	Versi 74
7	iOS	Versi ≤ 10, 11, 12
8	Android	Versi ≤ 3, 4

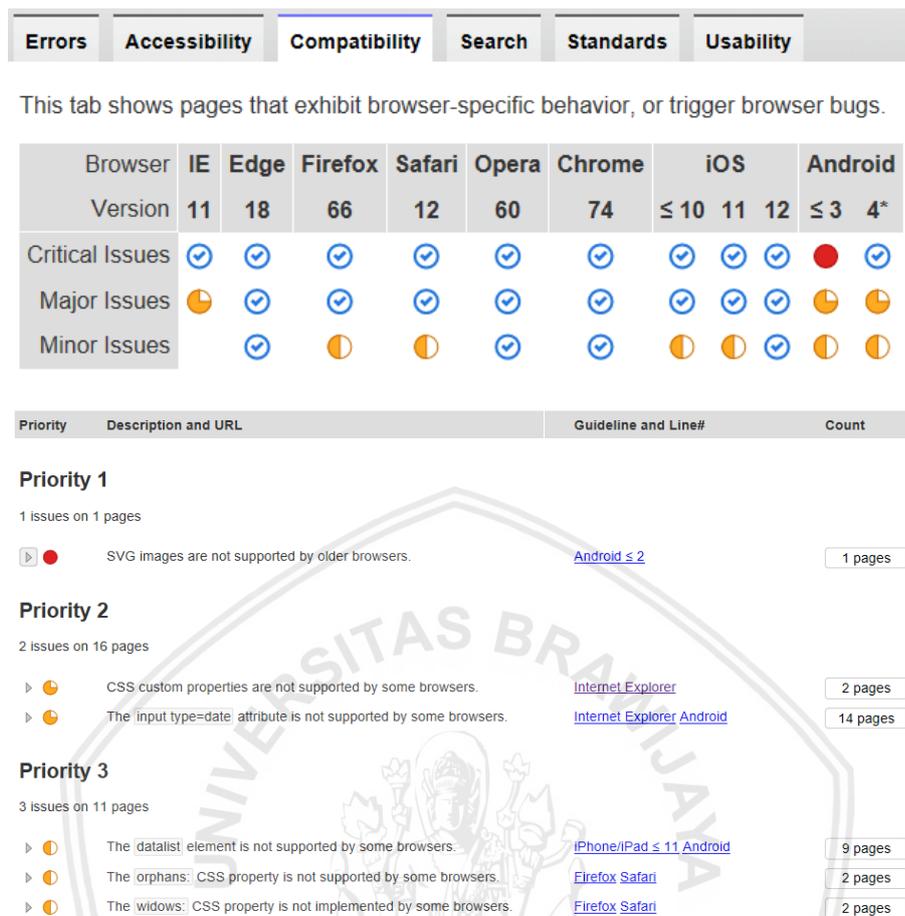
Tabel 7.23 menjelaskan jenis peramban dan versinya yang digunakan sebagai kasus uji. Peramban yang digunakan adalah Internet Explorer versi 11, Microsoft Edge versi 18, Mozilla Firefox versi 66, Safari versi 12, Opera versi 12, Google Chrome versi 74, iOS versi ≤ 10, 11, dan 12, serta Android versi ≤ 3, dan 4. Jenis dan versi peramban yang diuji ditentukan secara langsung oleh aplikasi SortSite 5 Trial. Pada pengujian kompatibilitas, aplikasi SortSite 5 Trial mengidentifikasi kategori masalah menjadi 3 kategori, yaitu *critical issues*, *major issues*, dan *minor issues*. Tiga kategori masalah akan ditunjukkan dalam Gambar 7.1.



Dalam Gambar 7.1 tanda lingkaran penuh berwarna merah merupakan *critical issues* yang menandakan terdapat masalah yang dapat menyebabkan hilangnya konten atau fungsionalitas. Lalu tanda tiga perempat lingkaran berwarna jingga merupakan *major issues* yang menandakan terdapat masalah major pada tampilan atau performa sistem. Kemudian tanda setengah lingkaran berwarna jingga merupakan *minor issues* yang menandakan terdapat masalah minor pada tampilan atau performa sistem. Terakhir tanda centang dalam lingkaran berwarna biru yang menunjukkan bahwa tidak ada masalah kompatibilitas.

7.1.2.2 Hasil Pengujian Non-Fungsional Sistem

Hasil pengujian menunjukkan hasil pengujian dari perancangan pengujian yang telah dilakukan sebelumnya. Hasil pengujian menunjukkan hasil uji kompatibilitas dari 8 peramban dan 11 versinya. Hasil pengujian menunjukkan ada atau tidaknya masalah kompatibilitas pada peramban sesuai kategori *compatibility issues*.



Gambar 7.2 Hasil Pengujian *Compatibility*

Gambar 7.2 menunjukkan hasil pengujian kebutuhan non-fungsional dengan menggunakan aplikasi SortSite 5 Trial. Pengujian dilakukan dengan memasukkan *url website* ke dalam *address bar* yang telah disediakan oleh aplikasi SortSite 5 Trial. Kemudian, tekan tombol “Go” untuk menampilkan halaman *website*. Lalu, tekan *dropdown* di sebelah tombol “Check” pada *toolbar*, setelah itu pilih “Check Entire Site”. Tunggu aplikasi SortSite 5 Trial melakukan pengujian pada *website*, setelah itu, pilih “Compatibility” pada *summary* pengujian. Sistem akan menampilkan hasil pengujian *compatibility*.

7.1.2.3 Analisis Hasil Pengujian Non-Fungsional Sistem

Analisis hasil pengujian menjelaskan secara rinci hasil pengujian kompatibilitas. Berdasarkan hasil pengujian, ditunjukkan bahwa terdapat *critical issues* pada peramban Android versi ≤ 3, lebih tepatnya pada peramban Android versi ≤ 2. Menurut aplikasi SortSite 5 Trial, *critical issues* terjadi disebabkan oleh peramban Android versi ≤ 2 tidak mendukung penggunaan gambar dengan format SVG. Hasil pengujian juga menunjukkan bahwa terdapat 2 *major issues* pada peramban Internet Explorer dan Android. *Major issues* pertama disebabkan oleh peramban Internet Explorer yang tidak mendukung *CSS property custom*. Kedua, peramban Internet Explorer dan Android tidak mendukung *input* dengan tipe *date*. Sedangkan pada *minor issues* terdapat 3 masalah berdasarkan hasil pengujian,

yaitu elemen *datalist* yang tidak didukung oleh peramban iOS versi ≤ 11 dan Android, serta CSS *orphans* dan *widows* yang merupakan bawaan template SB Admin 2 tidak didukung maupun tidak dapat diimplementasi oleh peramban Mozilla Firefox dan Safari. Terakhir, pengujian juga menunjukkan bahwa tidak ada masalah kompatibilitas atau *compatibility issues* pada peramban Microsoft Edge, Opera, Google Chrome, dan iOS versi 12. Dari hasil pengujian kompatibilitas yang telah dilakukan, dapat disarankan untuk penggunaan sistem menggunakan peramban Microsoft Edge minimal v11, Opera minimal v60, atau Google Chrome minimal v74.

7.2 Deployment

Deployment merupakan tahap pertama yang dilakukan dalam fase *transition*. Pada tahap *deployment* inilah sistem yang telah dibangun dan yang telah diuji diperkenalkan kepada lingkungan penggunaannya. Pengenalan sistem yang dilakukan adalah dengan melakukan instalasi pada perangkat yang ada di Edotel Pamekasan.

7.2.1 Spesifikasi Perangkat Keras di Edotel Pamekasan

Spesifikasi perangkat keras di Edotel Pamekasan menjelaskan mengenai komponen perangkat keras yang dimiliki Edotel Pamekasan. Komponen perangkat keras yang dideskripsikan adalah *model pc/laptop*, *processor*, *graphic card*, dan *random access memory (RAM)*. Spesifikasi perangkat keras ditunjukkan pada Tabel 7.24.

Tabel 7.24 Spesifikasi Perangkat Keras di Edotel Pamekasan

No.	Komponen	Spesifikasi
1	<i>Model PC / Laptop</i>	ASUS X453S
2	<i>Processor</i>	Intel® Celeron® CPU N3040 @1.60GHz (2 CPUs), ~1.6GHz
3	<i>Graphic Card</i>	Intel® HD Graphics
4	<i>Random Access Memory (RAM)</i>	2048MB RAM / 2GB RAM

Tabel 7.24 menunjukkan spesifikasi perangkat keras yang dimiliki Edotel Pamekasan. Terdiri dari *model laptop* ASUS X453S, dengan *processor* Intel® Celeron® CPU N3040 @1.60GHz (2 CPUs), ~1.6GHz. Perangkat keras didukung oleh *graphic card* Intel® HD Graphics, dan memiliki *random access memory (RAM)* sebesar 2GB.

7.2.2 Spesifikasi Perangkat Lunak di Edotel Pamekasan

Spesifikasi perangkat lunak di Edotel Pamekasan menjelaskan komponen perangkat lunak yang dimiliki Edotel Pamekasan. Komponen perangkat lunak yang dideskripsikan adalah *operating system (OS)*, *server*, *database management*

system (DBMS), dan web browser. Spesifikasi perangkat lunak ditunjukkan pada Tabel 7.25.

Tabel 7.25 Spesifikasi Perangkat Lunak di Edotel Pamekasan

No.	Komponen	Spesifikasi
1	<i>Operating System (OS)</i>	Windows 10 Pro 64-bit
2	<i>Server</i>	XAMPP v3.2.3 Apache localhost
3	<i>Database Management System (DBMS)</i>	MySQL
4	<i>Web Browser</i>	Microsoft Edge v44.17763.1.0

Tabel 7.25 menunjukkan spesifikasi perangkat lunak yang dimiliki Edotel Pamekasan. Terdiri dari *operating system (OS)* Windows 10 Pro 64-bit, menggunakan *server* XAMPP v3.2.3 Apache localhost, menggunakan *database management system (DBMS)* MySQL, dan *web browser* Microsoft Edge v44.17763.1.0.

7.3 User Acceptance Testing (UAT)

User Acceptance Testing merupakan tahap akhir fase *transition* sekaligus fase akhir pengembangan RUP. *User Acceptance Testing* dilakukan untuk mengetahui penerimaan pengguna terhadap sistem yang telah dibangun. Pengujian dilakukan dengan cara membuat rancangan kriteria pengujian, kemudian membuat *acceptance test plan*, lalu melakukan penilaian dari rancangan pernyataan yang telah dibuat.

7.3.1 Perancangan User Acceptance Testing (UAT)

Dalam melakukan kegiatan perancangan *User Acceptance Testing* atau UAT dimulai dengan melakukan pendefinisian kriteria penerimaan pengguna yang kemudian dijadikan dasar membuat pernyataan untuk dinilai oleh pengguna menggunakan *skala likert*. Pelaksanaan *User Acceptance Testing* dibagi berdasarkan pengguna utama yang akan berinteraksi dengan sistem. Terdapat 3 pengguna utama yang akan berinteraksi dengan sistem, yakni pengguna resepsionis, pengguna *store keeper*, dan pengguna manajer. Kriteria dan pernyataan yang akan dinilai oleh setiap aktor atau pengguna masing – masing resepsionis, *store keeper*, dan manajer ditunjukkan pada Tabel 7.26, Tabel 7.27, dan Tabel 7.28.

Tabel 7.26 Perancangan *User Acceptance Testing* pada Pengguna Resepsionis

No.	Kriteria	Pernyataan
1	<i>Functional Correctness and Completeness</i>	Sistem mampu membantu saya mengelola seluruh kegiatan reservasi kamar tamu hotel, termasuk konfirmasi reservasi dan pembatalan reservasi
		Sistem mampu membantu saya mengelola seluruh kegiatan registrasi tamu mulai dari <i>check-in</i> hingga <i>check-out</i> , termasuk mengelola tagihan tamu dan mencetak <i>bill</i>
		Sistem mampu membantu saya melihat kamar tersedia yang dapat dipesan berdasarkan masukan tanggal
2	<i>Accuracy</i>	Perhitungan total biaya reservasi menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya
		Perhitungan total biaya registrasi tamu menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya
3	<i>Usability</i>	Sistem ini mudah untuk saya gunakan
		Saya merasa mudah mempelajari sistem ini
		Sistem ini membantu saya melakukan pekerjaan saya
		Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem
		Menu, perintah, dan tampilan yang ada sudah jelas bagi saya
		Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas

Tabel 7.26 mendeskripsikan perancangan *User Acceptance Testing* pada pengguna resepsionis. Terdapat 3 kriteria yang didefinisikan antara lain *functional correctness and completeness*, *accuracy*, dan *usability*. Kriteria dipilih karena dinilai dapat menguji analisis kebutuhan dengan implementasi yang telah dilakukan sebelumnya. Dari 3 kriteria yang didefinisikan, menghasilkan 11 pernyataan yang harus dinilai oleh pengguna resepsionis.

Tabel 7.27 Perancangan *User Acceptance Testing* pada Pengguna *Store Keeper*

No.	Kriteria	Pernyataan
1	<i>Functional Correctness and Completeness</i>	Sistem mampu membantu saya mengelola seluruh kegiatan pencatatan barang masuk
		Sistem mampu membantu saya mengelola seluruh kegiatan pencatatan barang keluar
2	<i>Accuracy</i>	Perhitungan jumlah barang menggunakan sistem telah memenuhi harapan saya dalam perhitungan jumlah barang sebenarnya
3	<i>Usability</i>	Sistem ini mudah untuk saya gunakan
		Saya merasa mudah mempelajari sistem ini
		Sistem ini membantu saya melakukan pekerjaan saya
		Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem
		Menu, perintah, dan tampilan yang ada sudah jelas bagi saya
		Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas

Tabel 7.27 mendeskripsikan perancangan *User Acceptance Testing* pada pengguna *store keeper*. Terdapat 3 kriteria yang didefinisikan antara lain *functional correctness and completeness*, *accuracy*, dan *usability*. Kriteria dipilih karena dinilai dapat menguji analisis kebutuhan dengan implementasi yang telah dilakukan sebelumnya. Dari 3 kriteria yang didefinisikan, menghasilkan 9 pernyataan yang harus dinilai oleh pengguna *store keeper*.

Tabel 7.28 Perancangan *User Acceptance Testing* pada Pengguna Manajer

No.	Kriteria	Pernyataan
1	<i>Functional Correctness and Completeness</i>	Sistem mampu membantu saya melihat laporan bulanan transaksi tamu hotel, termasuk mencetaknya
		Sistem mampu membantu saya melihat laporan bulanan penggunaan barang, termasuk mencetaknya
2	<i>Accuracy</i>	Perhitungan pada laporan bulanan transaksi tamu hotel menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya
		Perhitungan pada laporan bulanan penggunaan barang menggunakan sistem telah memenuhi

Tabel 7.28 Perancangan *User Acceptance Testing* pada Pengguna Manajer (lanjutan)

No.	Kriteria	Pernyataan
		harapan saya dalam perhitungan total biaya sebenarnya
3	<i>Usability</i>	Sistem ini mudah untuk saya gunakan
		Saya merasa mudah mempelajari sistem ini
		Sistem ini membantu saya melakukan pekerjaan saya
		Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem
		Menu, perintah, dan tampilan yang ada sudah jelas bagi saya
		Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas

Tabel 7.28 mendeskripsikan perancangan *User Acceptance Testing* pada pengguna manajer. Terdapat 3 kriteria yang didefinisikan antara lain *functional correctness and completeness, accuracy, dan usability*. Kriteria dipilih karena dinilai dapat menguji analisis kebutuhan dengan implementasi yang telah dilakukan sebelumnya. Dari 3 kriteria yang didefinisikan, menghasilkan 10 pernyataan yang harus dinilai oleh pengguna *store* manajer.

Setelah mendeskripsikan kriteria dan pernyataannya, langkah berikutnya adalah membuat *acceptance test plan*. *Acceptance test plan* berisi tugas – tugas yang harus dilakukan oleh masing masing pengguna. Perancangan ATP ditunjukkan pada Tabel 7.29.

Tabel 7.29 Perancangan *Acceptance Test Plan* (ATP)

No.	Task
Resepsionis	
1	Membuat data reservasi baru
2	Membuat data registrasi tamu baru
3	Mencari kamar tersedia berdasarkan masukan tanggal
Store Keeper	
1	Membuat data pencatatan barang masuk baru
2	Membuat data pencatatan barang keluar baru
Manajer	
1	Melihat laporan bulanan transaksi tamu hotel berdasarkan masukan bulan dan tahun

Tabel 7.29 Perancangan *Acceptance Test Plan* (ATP) (lanjutan)

2	Melihat laporan bulanan penggunaan barang berdasarkan masukan bulan dan tahun
---	---

Tabel 7.29 menunjukkan tugas – tugas yang harus dilakukan pengguna yang dibuat ke dalam ATP. Terdapat 3 tugas yang harus dijalankan resepsionis, 2 tugas untuk *store keeper*, lalu 2 tugas untuk manajer. Tugas dirancang untuk mempermudah pengguna dalam memberikan penilaian penerimaan pengguna karena secara langsung berinteraksi dengan sistem.

7.3.2 Hasil *User Acceptance Testing* (UAT)

Hasil *User Acceptance Testing* merupakan hasil penilaian penerimaan pengguna terhadap sistem. Terdapat total 4 responden, dengan resepsionis sebanyak 2 orang responden, *store keeper* sebanyak 1 orang responden, dan manajer sebanyak 1 orang responden. Hasil *User Acceptance Testing* ditunjukkan pada tabel 7.30, 7.32, dan 7.34.

Tabel 7.30 Hasil *User Acceptance Testing* pada Pengguna Resepsionis

No.	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Sistem mampu membantu saya mengelola seluruh kegiatan reservasi kamar tamu hotel, termasuk konfirmasi reservasi dan pembatalan reservasi	0	0	0	0	2
2	Sistem mampu membantu saya mengelola seluruh kegiatan registrasi tamu mulai dari <i>check-in</i> hingga <i>check-out</i> , termasuk mengelola tagihan tamu dan mencetak <i>bill</i>	0	0	0	0	2
3	Sistem mampu membantu saya melihat kamar tersedia yang dapat dipesan berdasarkan masukan tanggal	0	0	0	0	2
4	Perhitungan total biaya reservasi menggunakan sistem telah memenuhi	0	0	0	1	1

Tabel 7.30 Hasil *User Acceptance Testing* pada Pengguna Resepsionis (lanjutan)

No.	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	harapan saya dalam perhitungan total biaya sebenarnya					
5	Perhitungan total biaya registrasi tamu menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya	0	0	0	1	1
6	Sistem ini mudah untuk saya gunakan	0	0	0	2	0
7	Saya merasa mudah mempelajari sistem ini	0	0	0	1	1
8	Sistem ini membantu saya melakukan pekerjaan saya	0	0	0	1	1
9	Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem	0	0	0	1	1
10	Menu, perintah, dan tampilan yang ada sudah jelas bagi saya	0	0	0	2	0
11	Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas	0	0	0	1	1

Tabel 7.30 menunjukkan formulir penilaian *User Acceptance Testing* pada pengguna resepsionis. Terdapat 11 poin pernyataan yang harus diisi responden. Jumlah responden pada pengguna resepsionis adalah sebanyak 2 orang. Hasil rekapitulasi pengujian ditunjukkan pada tabel 7.31.

Tabel 7.31 Rekapitulasi Hasil *User Acceptance Testing* pada Pengguna Resepsionis

No.	Penilaian	Jumlah
1	Sangat Tidak Setuju	0
2	Tidak Setuju	0
3	Netral	0
4	Setuju	10
5	Sangat Setuju	12

Tabel 7.31 berisi rekapitulasi hasil pengujian dengan 11 pernyataan. Dari 11 pernyataan yang diajukan kepada 2 responden, pilihan “Sangat Setuju” merupakan pilihan yang paling banyak dipilih dengan jumlah 12 kali, lalu “Setuju” dengan jumlah 10 kali dipilih, dan responden tidak memilih “Netral”, “Tidak Setuju”, maupun “Sangat Tidak Setuju” sama sekali.

Tabel 7.32 Hasil *User Acceptance Testing* pada Pengguna *Store Keeper*

No.	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Sistem mampu membantu saya mengelola seluruh kegiatan pencatatan barang masuk	0	0	0	0	1
2	Sistem mampu membantu saya mengelola seluruh kegiatan pencatatan barang keluar	0	0	0	0	1
3	Perhitungan jumlah barang menggunakan sistem telah memenuhi harapan saya dalam perhitungan jumlah barang sebenarnya	0	0	0	0	1
4	Sistem ini mudah untuk saya gunakan	0	0	0	1	0
5	Saya merasa mudah mempelajari sistem ini	0	0	0	1	0

Tabel 7.32 Hasil *User Acceptance Testing* pada Pengguna *Store Keeper* (lanjutan)

No.	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
6	Sistem ini membantu saya melakukan pekerjaan saya	0	0	0	0	1
7	Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem	0	0	0	0	1
8	Menu, perintah, dan tampilan yang ada sudah jelas bagi saya	0	0	0	1	0
9	Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas	0	0	0	0	1

Tabel 7.32 menunjukkan formulir penilaian *User Acceptance Testing* pada pengguna *store keeper*. Terdapat 9 poin pernyataan yang harus diisi responden. Jumlah responden pada pengguna *store keeper* adalah sebanyak 1 orang. Hasil rekapitulasi pengujian ditunjukkan pada tabel 7.33.

Tabel 7.33 Rekapitulasi Hasil *User Acceptance Testing* pada Pengguna *Store Keeper*

No.	Penilaian	Jumlah
1	Sangat Tidak Setuju	0
2	Tidak Setuju	0
3	Netral	0
4	Setuju	3
5	Sangat Setuju	6

Tabel 7.33 berisi rekapitulasi hasil pengujian dengan 9 pernyataan. Dari 9 pernyataan yang diajukan kepada 1 responden, pilihan “Sangat Setuju” merupakan pilihan yang paling banyak dipilih dengan jumlah 6 kali, lalu “Setuju” dengan jumlah 3 kali dipilih, dan responden tidak memilih “Netral”, “Tidak Setuju”, maupun “Sangat Tidak Setuju” sama sekali.

Tabel 7.34 Hasil *User Acceptance Testing* pada Pengguna Manajer

No.	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Sistem mampu membantu saya melihat laporan bulanan transaksi tamu hotel, termasuk mencetaknya	0	0	0	0	1
2	Sistem mampu membantu saya melihat laporan bulanan penggunaan barang, termasuk mencetaknya	0	0	0	0	1
3	Perhitungan pada laporan bulanan transaksi tamu hotel menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya	0	0	0	0	1
4	Perhitungan pada laporan bulanan penggunaan barang menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya	0	0	0	0	1
5	Sistem ini mudah untuk saya gunakan	0	0	0	0	1
6	Saya merasa mudah mempelajari sistem ini	0	0	0	0	1
7	Sistem ini membantu saya melakukan pekerjaan saya	0	0	0	0	1
8	Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem	0	0	0	1	0

Tabel 7.34 Hasil *User Acceptance Testing* pada Pengguna Manajer (lanjutan)

No.	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
9	Menu, perintah, dan tampilan yang ada sudah jelas bagi saya	0	0	0	0	1
10	Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas	0	0	0	1	0

Tabel 7.34 menunjukkan formulir penilaian *User Acceptance Testing* pada pengguna manajer. Terdapat 10 poin pernyataan yang harus diisi responden. Jumlah responden pada pengguna manajer adalah sebanyak 1 orang. Hasil rekapitulasi pengujian ditunjukkan pada tabel 7.35.

Tabel 7.35 Rekapitulasi Hasil *User Acceptance Testing* pada Pengguna Manajer

No.	Penilaian	Jumlah
1	Sangat Tidak Setuju	0
2	Tidak Setuju	0
3	Netral	0
4	Setuju	2
5	Sangat Setuju	8

Tabel 7.35 berisi rekapitulasi hasil pengujian dengan 10 pernyataan. Dari 10 pernyataan yang diajukan kepada 1 responden, pilihan “Sangat Setuju” merupakan pilihan yang paling banyak dipilih dengan jumlah 8 kali, lalu “Setuju” dengan jumlah 2 kali dipilih, dan responden tidak memilih “Netral”, “Tidak Setuju”, maupun “Sangat Tidak Setuju” sama sekali.

7.3.3 Analisis Hasil *User Acceptance Testing* (UAT)

Analisis hasil UAT dilakukan dengan menghitung hasil pengujian yang telah dilakukan sehingga menghasilkan indeks persentase. Nilai indeks persentase digunakan untuk mengkategorikan penerimaan pengguna terhadap sistem. Hasil perhitungan Indeks Persentase pengguna resepsionis adalah:

1. Pengguna Resepsionis

Jawaban Sangat Setuju (A) = 60

Jawaban Setuju (B) = 40

Jawaban Netral (C) = 0

Jawaban Tidak Setuju (D) = 0

Jawaban Sangat Tidak Setuju (E) = 0

Tabel 7.36 Hasil User Acceptance Testing Pengguna Resepsionis

No.	Nama Pengguna	Total Nilai	Nilai Y	Indeks Persentase
1	Resepsionis	100	110	90,90%

Tabel 7.36 menunjukkan pengguna resepsionis memiliki indeks persentase sebesar 90,90%. Indeks persentase didapatkan berdasarkan perhitungan total nilai sebesar 100, dibagi nilai Y dengan nilai sebesar 110, kemudian dikalikan dengan 100%. Indeks persentase tersebut termasuk ke dalam kategori "Sangat Setuju" dalam *skala likert*. Selanjutnya perhitungan indeks persentase pengguna *store keeper*, dengan hasil perhitungannya:

2. Pengguna *Store Keeper*

Jawaban Sangat Setuju (A) = 30

Jawaban Setuju (B) = 12

Jawaban Netral (C) = 0

Jawaban Tidak Setuju (D) = 0

Jawaban Sangat Tidak Setuju (E) = 0

Tabel 7.37 Hasil User Acceptance Testing Pengguna Store Keeper

No.	Nama Pengguna	Total Nilai	Nilai Y	Indeks Persentase
1	<i>Store Keeper</i>	42	45	93,33%

Tabel 7.37 menunjukkan pengguna *store keeper* memiliki indeks persentase sebesar 93,33%. Indeks persentase didapatkan berdasarkan perhitungan total nilai sebesar 42, dibagi nilai Y dengan nilai sebesar 45, kemudian dikalikan dengan 100%. Indeks persentase tersebut termasuk ke dalam kategori "Sangat Setuju" dalam *skala likert*. Selanjutnya perhitungan indeks persentase pengguna manajer, dengan hasil perhitungannya:

3. Pengguna Manajer

Jawaban Sangat Setuju (A) = 40

Jawaban Setuju (B) = 8

Jawaban Netral (C) = 0

Jawaban Tidak Setuju (D) = 0

Jawaban Sangat Tidak Setuju (E) = 0

Tabel 7.38 Hasil User Acceptance Testing Pengguna Manajer

No.	Nama Pengguna	Total Nilai	Nilai Y	Indeks Persentase
1	Manajer	48	50	96%

Tabel 7.38 menunjukkan pengguna manajer memiliki indeks persentase sebesar 96%. Indeks persentase didapatkan berdasarkan perhitungan total nilai sebesar 48, dibagi nilai Y dengan nilai sebesar 50, kemudian dikalikan dengan 100%. Indeks persentase tersebut termasuk ke dalam kategori “Sangat Setuju” dalam *skala likert*. Sehingga berdasarkan 3 pengguna yang diuji, rata – rata nilai penerimaan pengguna adalah:

Tabel 7.39 Rata – Rata Hasil User Acceptance Testing

No.	Nama Pengguna	Total Nilai	Nilai Y	Indeks Persentase
1	Resepsionis	110	110	90,90%
2	Store Keeper	42	45	93,33%
3	Manajer	48	50	96%
Rata – rata				93,41%

Tabel 7.39 menunjukkan rata – rata nilai indeks persentase penerimaan pengguna sebesar 93,41%. Nilai indeks persentase didapatkan dari menjumlahkan total nilai indeks persentase 3 pengguna, kemudian dibagi 3. Berdasarkan *skala likert*, rata – rata nilai indeks persentase penerimaan pengguna masuk ke dalam kategori “Sangat Setuju” karena memiliki persentase diatas 80%.

BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan penelitian yang telah selesai dilakukan dengan menggunakan metode *Rational Unified Process*, terdapat kesimpulan dalam pengembangan sistem informasi perhotelan Edotel Pamekasan sebagai berikut:

1. Pada fase pertama yaitu fase *inception* dilakukan pemodelan proses bisnis saat ini (*as-is*), evaluasi proses bisnis saat ini (*as-is*), dan pemodelan proses bisnis usulan (*to-be*). Berdasarkan hasil pemodelan proses bisnis saat ini (*as-is*), didapatkan model 5 permasalahan yang terjadi di Edotel Pamekasan yang kemudian dijelaskan pada evaluasi proses bisnis saat ini (*as-is*), masalah tersebut antara lain kegiatan reservasi dan registrasi yang dilakukan masih secara konvensional, pencatatan buku yang tamu tidak dilakukan dengan benar, dan resepsionis harus menulis ulang formulir registrasi ketika tamu akan melakukan *check-in* walau sudah melakukan reservasi, lalu *bill* dan tagihan yang masih dihitung secara manual dapat terselip atau hilang, kemudian tidak ada pencatatan barang dan stoknya di gudang sehingga terjadi kehabisan stok barang, dan juga pengecekan *bill* dan pembuatan laporan memakan waktu 1 hari kerja atau bisa lebih dan tidak memiliki pelaporan penggunaan barang. Dari permasalahan tersebut, dihasilkan usulan perbaikan proses bisnis dengan menerapkan sistem informasi ke dalam proses bisnisnya. Perbaikan proses bisnis dengan menambahkan peran sistem informasi ke dalam proses bisnis menghasilkan pemodelan proses bisnis usulan (*to-be*).
2. Kemudian pada fase *elaboration* dilakukan analisis kebutuhan dan perancangan sistem. Analisis kebutuhan yang dilakukan berdasarkan evaluasi proses bisnis saat ini (*as-is*) dan proses bisnis usulan (*to-be*). Melalui 2 kali iterasi, analisis kebutuhan menghasilkan 5 identifikasi aktor, 7 identifikasi fitur, 32 kebutuhan fungsional, 1 kebutuhan non-fungsional, juga 7 *use case*, dan pemodelan *use case scenario*. Selanjutnya perancangan sistem menghasilkan *sequence diagram* yang dimodelkan berdasarkan pada *use case scenario*, model *class diagram* dengan rincian 9 *class diagram controller* dan 8 *class diagram model*, *conceptual data model* yang menghasilkan inialisasi 8 tabel, kemudian *physical data model* menambahkan tipe data dan relasinya pada tiap tabel sehingga menghasilkan *blueprint* perancangan *database*, dan terakhir rancangan desain antarmuka pengguna menghasilkan *wireframe*.
3. Fase *construction* melakukan tahapan implementasi sistem dan pengujian sistem. Hasil Implementasi sistem adalah pendeskripsian spesifikasi sistem yang digunakan peneliti dalam mengembangkan sistem, kemudian menghasilkan implementasi *class*, implementasi *database*, dan implementasi antarmuka pengguna berdasarkan perancangan sistem yang telah dilakukan pada fase *elaboration*. Kemudian, dilakukan pengujian sistem sebanyak 2 kali. Pengujian pertama adalah pengujian fungsional sistem menggunakan *black-box validation testing*. Pengujian yang dilakukan dengan menjalankan 15 kasus

uji menghasilkan persentase status valid sebesar 100%. Hasil tersebut menunjukkan bahwa implementasi fungsional telah sesuai dengan analisis kebutuhan dan perancangannya. Lalu pengujian kedua adalah pengujian non-fungsional sistem menggunakan *compatibility testing*. Pengujian yang dilakukan dengan menggunakan aplikasi SortSite Trial 5 menunjukkan bahwa sistem dapat berjalan dengan baik tanpa ada kendala apapun jika menggunakan peramban yang direkomendasikan, seperti Microsoft Edge minimal v11, Opera minimal v60, atau Google Chrome minimal v74.

4. Terakhir dalam metode RUP, yaitu fase *transition* dilakukan *deployment* dan *User Acceptance Testing*. *Deployment* yang dilakukan menghasilkan deskripsi perangkat keras serta perangkat lunak yang dipergunakan untuk melakukan instalasi sistem di Edotel Pamekasan. Kemudian dilakukan pengujian penerimaan sistem menggunakan *User Acceptance Testing* (UAT). UAT dilakukan kepada 3 jenis pengguna, yakni 2 orang responden pengguna resepsionis, 1 orang responden pengguna *store keeper*, dan 1 orang responden pengguna manajer. Hasil pengujian menunjukkan tingkat penerimaan pengguna resepsionis sebesar 90,90%, pengguna *store keeper* sebesar 93,33%, manajer sebesar 96%, dan menghasilkan rata – rata penerimaan pengguna sebesar 93,41%. Berdasarkan *skala likert*, hasil pengujian tersebut masuk ke dalam kategori “Sangat Setuju” karena memiliki persentase diatas 80%. Sehingga dapat disimpulkan bahwa sistem dapat diterima dengan baik oleh pengguna.

8.2 Saran

Berdasarkan penelitian pengembangan sistem informasi perhotelan Edotel Pamekasan yang telah dilakukan, beberapa saran yang dapat diberikan oleh peneliti untuk penelitian selanjutnya ialah:

1. Melakukan evaluasi dan perbaikan tampilan sistem informasi perhotelan Edotel Pamekasan.
2. *Deployment* pada penelitian ini hanya melakukan instalasi sistem ke perangkat yang ada di Edotel Pamekasan, saran untuk penelitian selanjutnya adalah melakukan *deployment* sistem informasi perhotelan sampai pada fase sistem benar – benar digunakan untuk kegiatan operasionalnya.

DAFTAR REFERENSI

- Agarwal, B.B., Tayal, S.P. dan Gulpa, M., 2010. *Software Engineering & Testing*. Sadbury, Massachusetts: Jones And Bartlett Publishers.
- Agusvianto, H., 2017. Sistem Informasi Inventori Gudang Untuk Mengontrol Persediaan Barang Pada Gudang Studi Kasus : PT.Alaisys Sidoarjo. *Journal Information Engineering and Educational Technology*, 01, hal.40–46.
- Bittner, K. dan Spence, I., 2002. *Use Case Modeling*. Addison Wesley.
- Booch, G., Rumbaugh, J. dan Jacobson, I., 1998. *The Unified Modeling Language User Guide*. Massachusetts: Addison Wesley.
- Jin, L. dan Liang, X., 2016. Modeling of Instant Messaging System Based on RUP and UML. *Proceedings - 2016 International Conference on Computational Intelligence and Applications, ICCIA 2016*, hal.61–66.
- Melian-Gonzalez, S. dan Bulchand-Gidumal, J., 2016. A Model That Connects Information Technology and Hotel Performance. *Tourism Management*, 53, hal.30–37.
- Monk, E.F. dan Wagner, B.J., 2013. *Concepts in Enterprise Resource Planning*. 4th Editio ed. Boston: Course Technology, Cengage Learning.
- Naik, K. dan Tripathy, P., 2008. *Software Testing and Quality Assurance Theory And Practice*. New Jersey: A John Wiley & Sons, Inc.
- O'Brien, J.A. dan Marakas, G.M., 2010. *Management Information Systems*. 10th ed. New York: McGraw-Hill/Irwin.
- Object Management Group, 2011. *Business Process Model and Notation (BPMN) Version 2.0*. Object Management Group, Inc.
- Pressman, R.S., 2010. *Software Engineering : A Practitioner's Approach*. New York: Mc-Graw Hill.
- Regar, T.A., Areros, W.A. dan Asaloei, S.I., 2018. Perepsi Karyawan Terhadap Penggunaan Teknologi Informasi Pada Hotel Sintesa Peninsula Manado. *Jurnal Administrasi Bisnis*, 6(2), hal.17–26.
- Rumbaugh, J., Jacobson, I. dan Booch, G., 2005. *The Unified Modeling Language Reference Manual Second Edition*. Addison-Wesley.
- Saputra, A.A., Mursityo, Y.T. dan Setiawan, N.Y., 2019. Pengembangan Sistem Informasi Reservasi Pada CV. Dwi Artha Indah Samarinda Menggunakan Metode Rational Unified Process. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(3), hal.2379–2387.
- Saputra, F.Y.E. dan Christianti, M., 2013. Pemodelan Proses Bisnis Menggunakan Idef0 dengan Studi Kasus PT. Bank Central Asia Tbk Subang. *Jurnal Teknologi Informasi-Aiti*, 10.

- Sommerville, I., 2011. *Software Engineering*. 9th Editio ed. New York: Addison-Wesley.
- Sparx Systems, 2017. *Database Models*. Enterprise Architect.
- Sugiyono, 2012. *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- Vuksanovic, I.P. dan Sudarevic, B., 2011. Use of Web Application Frameworks in the Development of Small Applications. *MIPRO, 2011 Proceedings of the 34th International Convention, Opatija, Croatia*, (January).
- Walerych, M. dan Zabierowski, W., 2012. Application Supporting Hotel Management as an Example of Web Technologies Usage. *Proceedings of International Conference on Modern Problem of Radio Engineering, Telecommunications and Computer Science*, hal.327–328.
- Weske, M., 2007. *Business Process Management*. Berlin: Springer.



LAMPIRAN A WAWANCARA PERMASALAHAN

HASIL WAWANCARA

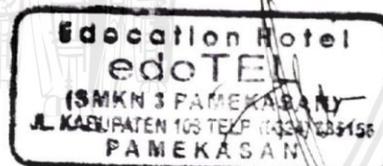
Narasumber : Yuli Indrawati
 Jabatan : *General Manager* Edotel Pamekasan
 Tempat : Edotel Pamekasan. Jl. Kabupaten No. 103, Pamekasan
 Hari, Tanggal : Minggu, 30 Desember 2018

- P: Pewawancara N: Narasumber
- P: Bagaimana gambaran umum mengenai Edotel Pamekasan?
- N: Berawal dari 2 ruang praktik / laboratorium pembelajaran siswa Program Keahlian Akomodasi Perhotelan di SMK Negeri 3 Pamekasan. Hingga pada tahun 2004, SMK Negeri 3 Pamekasan mendapat bantuan dari pemerintah berupa pembangunan 8 kamar hotel melalui program hotel training. Dengan total 10 kamar pada saat itu, hotel milik SMK Negeri 3 Pamekasan berganti menjadi Edotel Pamekasan yang dibuka untuk umum sekaligus tempat pelatihan siswa praktik menangani dan melayani tamu. Kemudian pada tahun 2006 dan 2007, Edotel Pamekasan kembali mendapat bantuan hotel training sehingga saat ini total kamar sejumlah 18 kamar.
- P: Apa saja proses bisnis yang berjalan pada Edotel Pamekasan?
- N: Edotel Pamekasan memiliki beberapa proses bisnis utama, yaitu pemesanan atau reservasi kamar, registrasi *check-in*, dan *check-out*. Lalu didukung proses bisnis administrasi kegiatan operasional mencakup keuangan, pengarsipan nota – nota.
- P: Apakah proses bisnis utama sudah berjalan optimal atau masih terdapat kendala?
- N: Proses bisnis utama masih belum berjalan secara optimal. Seluruh proses bisnisnya, tidak hanya proses bisnis utama masih dilakukan secara manual, belum terkomputerisasi sama sekali. Untuk proses bisnis utama, terdapat masalah pada reservasi dan *check-in* kamar, yaitu terjadi kasus tumpang tindih pesanan reservasi kamar dengan reservasi yang lain, ataupun dengan tamu yang *check-in* secara langsung. Masalah tersebut berdampak pada kekecewaan pelanggan. Mengatasi hal tersebut biasanya hotel memberi kompensasi berupa kamar dengan *room type* yang satu tingkat diatas *room type* yang dipesan, tanpa tambahan biaya. Hal tersebut membuat hotel merugi karena harus menomboki tambahan pembayaran.
- P: Untuk proses bisnis pendukungnya apakah sudah optimal?
- N: Untuk pembuatan laporan keuangan tidak ada masalah, kecuali tidak bisa meminta laporan sewaktu – waktu diperlukan karena bagian *accounting* harus merekap nota – nota salinan *bill* dan pembelian barang terlebih dahulu. Masalah baru terjadi ketika ada nota terselip yang menyebabkan masalah di penulisan laporan, ataupun nota hilang sehingga harus buat baru yang menyebabkan makin lama proses pembuatan laporannya.

- P: Mengenai pembelian barang, bagaimana proses bisnis pengelolaan barang yang terjadi disana?
- N: Tidak ada kegiatan pengelolaan ataupun pemantauan stok barang. Yang ada hanya proses pembelian barang yang dilakukan ketika bagian *accounting* mendapatkan laporan barang habis, kemudian di cek di gudang, jika benar habis maka melakukan pemesanan barang. Untuk pemesanan barang dilakukan pada *supplier* di Surabaya yang pengirimannya bisa memakan waktu 1 s/d 3 hari. Hal tersebut menjadi masalah ketika ketika barang yang habis merupakan barang yang diperlukan, sehingga terkadang harus diganti dengan barang seadanya
- P: Jika peneliti mengusulkan untuk membangun sistem informasi untuk mengatasi masalah yang terjadi, apa saja yang diharapkan?
- N: Harapannya adalah ada sistem informasi yang dapat membantu hotel meningkatkan kualitas pelayanan dengan memudahkan karyawan dalam menangani reservasi, *check-in*, maupun pembuatan *bill* saat *check-out*, mampu meminimalisir kesalahan penanganan tamu yang sebelumnya masih melakukan pengecekan kamar secara manual. Memudahkan pengecekan persediaan barang (*guest supplies*) sehingga tidak perlu menunggu kehabisan stok terlebih dahulu, dan dapat melihat laporan kapanpun saat dibutuhkan.

Pamekasan, 15 Mei 2019

Mengetahui



Yuli Indrawati

LAMPIRAN B PERSETUJUAN KEBUTUHAN PENGGUNA

PERSETUJUAN KEBUTUHAN

Nama : YULI INDRAMATI
Tanggal : 15 MEI 2019
Tempat : EDOTEL PAMEKASAN

Dengan ini menyatakan bahwa pemodelan proses bisnis yang sedang berjalan saat ini di Edotel Pamekasan telah sesuai dengan keadaan yang sebenarnya. Serta menyetujui perbaikan proses bisnis usulan, dan analisis kebutuhan yang telah didiskusikan sebelumnya.

Pamekasan, 15 Mei 2019



LAMPIRAN C HASIL PENGUJIAN USER ACCEPTANCE TESTING (UAT)

USER ACCEPTANCE TESTING

Sistem Informasi Perhotelan edoTEL Pamekasan

Nama : Fitriana Novita
 Jabatan : Resepsionis
 Tanggal Pengujian : 22 Juni 2019

No	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Sistem mampu membantu saya mengelola seluruh kegiatan reservasi kamar tamu hotel, termasuk konfirmasi reservasi dan pembatalan reservasi					✓
2	Sistem mampu membantu saya mengelola seluruh kegiatan registrasi tamu mulai dari <i>check-in</i> hingga <i>check-out</i> , termasuk mengelola tagihan tamu dan mencetak <i>bill</i>					✓
3	Sistem mampu membantu saya melihat kamar tersedia yang dapat dipesan berdasarkan masukan tanggal					✓
4	Perhitungan total biaya reservasi menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya				✓	
5	Perhitungan total biaya registrasi tamu menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya				✓	



No	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
6	Sistem ini mudah untuk saya gunakan				✓	
7	Saya merasa mudah mempelajari sistem ini					✓
8	Sistem ini membantu saya melakukan pekerjaan saya				✓	
9	Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem				✓	
10	Menu, perintah, dan tampilan yang ada sudah jelas bagi saya				✓	
11	Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas					✓

Pamekasan, 22 Juni 2019



 Fitriana Novita



USER ACCEPTANCE TESTING

Sistem Informasi Perhotelan edoTEL Pamekasan

Nama : Wasih Arrohman

Jabatan : Resepsionis

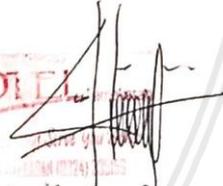
Tanggal Pengujian : 22 Juni 2019

No	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Sistem mampu membantu saya mengelola seluruh kegiatan reservasi kamar tamu hotel, termasuk konfirmasi reservasi dan pembatalan reservasi					✓
2	Sistem mampu membantu saya mengelola seluruh kegiatan registrasi tamu mulai dari <i>check-in</i> hingga <i>check-out</i> , termasuk mengelola tagihan tamu dan mencetak <i>bill</i>					✓
3	Sistem mampu membantu saya melihat kamar tersedia yang dapat dipesan berdasarkan masukan tanggal					✓
4	Perhitungan total biaya reservasi menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya					✓
5	Perhitungan total biaya registrasi tamu menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya					✓



No	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
6	Sistem ini mudah untuk saya gunakan				✓	
7	Saya merasa mudah mempelajari sistem ini				✓	
8	Sistem ini membantu saya melakukan pekerjaan saya					✓
9	Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem					✓
10	Menu, perintah, dan tampilan yang ada sudah jelas bagi saya				✓	
11	Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas					✓

Pamekasan, 22 Juni 2019


Wasih Arrohmah



USER ACCEPTANCE TESTING

Sistem Informasi Perhotelan edoTEL Pamekasan

Nama : *Emelda Martha*
 Jabatan : *Store Keeper*
 Tanggal Pengujian : *22 Juni 2019*

No	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Sistem mampu membantu saya mengelola seluruh kegiatan pencatatan barang masuk					✓
2	Sistem mampu membantu saya mengelola seluruh kegiatan pencatatan barang keluar					✓
3	Perhitungan jumlah barang menggunakan sistem telah memenuhi harapan saya dalam perhitungan jumlah barang sebenarnya					✓
4	Sistem ini mudah untuk saya gunakan				✓	
5	Saya merasa mudah mempelajari sistem ini				✓	
6	Sistem ini membantu saya melakukan pekerjaan saya					✓
7	Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem					✓



No	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
8	Menu, perintah, dan tampilan yang ada sudah jelas bagi saya				✓	
9	Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas					✓

Pamekasan, 22 Juni 2019



Emelda Martha

Emelda Martha



USER ACCEPTANCE TESTING

Sistem Informasi Perhotelan edoTEL Pamekasan

Nama : YULI INDRAWATI, S.Pd,MM

Jabatan : Manajer

Tanggal Pengujian : 22 JUNI 2019

No	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Sistem mampu membantu saya melihat laporan bulanan transaksi tamu hotel, termasuk mencetaknya					✓
2	Sistem mampu membantu saya melihat laporan bulanan penggunaan barang, termasuk mencetaknya					✓
3	Perhitungan pada laporan bulanan transaksi tamu hotel menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya					✓
4	Perhitungan pada laporan bulanan penggunaan barang menggunakan sistem telah memenuhi harapan saya dalam perhitungan total biaya sebenarnya					✓
5	Sistem ini mudah untuk saya gunakan					✓
6	Saya merasa mudah mempelajari sistem ini					✓
7	Sistem ini membantu saya melakukan pekerjaan saya					✓



No	Pernyataan	Penilaian				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
8	Produktivitas, kepuasan pelanggan, kemampuan, dan kualitas kerja saya meningkat karena menggunakan sistem				✓	
9	Menu, perintah, dan tampilan yang ada sudah jelas bagi saya					✓
10	Penanganan <i>error</i> yang dimiliki oleh sistem meningkatkan keandalan dan produktivitas				✓	

Pamekasan, 22 JUNI 2019





RADEA ZULINDRA ARDISUKMA

ABOUT

I am an all rounder person because i'd love to learn something new where i can enrich my knowledge and skills for continuous improvement..

PROFESSIONAL SKILLS

System design and analysis
Website development
Project management
Database design
Graphic design

PERSONAL SKILLS

Willing to learn
Team player
Professional
Fast learner
Organized

CONTACT

Phone: 081231458001
Email: rzulindra@gmail.com
LinkedIn: Radea Zulindra
Address: Perumahan Graha Kencana
Blok Z No. 41, Pamekasan, Jawa Timur

EXPERIENCE

SOFTWARE DEVELOPER

Bejana Investidata Globalindo | Dec 2018 - Feb 2019

- Improved an banking apps using ionic and codeigniter

HEAD OF STUDY RESEARCH AND DEVELOPMENT DEPARTMENT

Eksekutif Mahasiswa Sistem Informasi | Mar 2018 - Feb 2019

- Managed staff in my department
- Designed and executed department's work program

DATABASE PRACTICUM ASSISTANT

Faculty of Computer Science, Brawijaya University | Aug 2017 - Dec 2018

- Taught basic and administration database

EDUCATION

FACULTY OF COMPUTER SCIENCE, MAJOR IN INFORMATION SYSTEM

Brawijaya University | 2015 - 2019

SENIOR HIGH SCHOOL 1 PAMEKASAN

Senior High School | 2012 - 2015

JUNIOR HIGH SCHOOL 1 PAMEKASAN

Junior High School | 2009 - 2012

PROJECT

HOSPITALITY INFORMATION SYSTEM

2019

TIBOX

2017

