

**PENGEMBANGAN SISTEM INFORMASI PENDATAAN RAK
DAN PERANGKAT SERVER BERBASIS WEB (STUDI KASUS
PADA PT. INDOSAT OOREDOO SURABAYA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Dinda Ayu Rudyana Putri

NIM: 145150401111055



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

PENGESAHAN

PENGESAHAN

Pengembangan Sistem Informasi Pendataan Rak dan Perangkat *Server* Berbasis Web (Studi Kasus Pada PT. Indosat Ooredoo Surabaya)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
Dinda Ayu Rudyana Putri
NIM: 145150401111055

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Januari 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ismiarta Aknuranda, S.T., M.Sc, Ph.D
NIP: 201006 740719 1 001

Nanang Yudi Setiawan S.T., M.Kom
NIP: 19760619 200604 1 001

Mengetahui
Ketua Jurusan Sistem Informasi



Dr. Eng. Herman Tolle, S.T., M.T.
NIP: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

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 pustaka.

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, 3 Januari 2019



Dinda Ayu Rudyana Putri

NIM: 145150401111055

PRAKATA

Segala puji bagi Allah SWT atas limpahan rahmat yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem Informasi Pendataan Rak dan Perangkat *Server* Berbasis Web (Studi Kasus Pada PT. Indosat Ooredoo Surabaya)”.

Dari aspek akademis, penelitian ini mengajarkan banyak ilmu yang belum saya dapatkan di kelas perkuliahan. Selain itu, penelitian ini juga mengajarkan bagaimana penulis dapat berkomunikasi dengan banyak orang, menyamakan persepsi yang saya miliki dengan pemangku kepentingan dan dosen pembimbing. Proses penelitian yang dilakukan juga mengajarkan untuk menjadi pribadi yang lebih sabar, menurunkan ego, dan senantiasa berbagi hal-hal yang bermanfaat yang berkaitan dengan kebutuhan pembelajaran. Penelitian ini tidak dapat diselesaikan tanpa dukungan dan bantuan dari orang-orang terdekat, sehingga saya ingin menyampaikan rasa terima kasih kepada:

1. Ibu saya, Ibu Nunik Prihartini yang selalu memberikan dukungan doa, moril, materiil, mengajarkan ketegasan, kedisiplinan dalam menyelesaikan tugas, dan senantiasa memberikan contoh yang baik untuk menjadi pribadi yang sabar dan pekerja keras. Ayah saya, Bapak Rudy Poerwanto yang sangat saya yakini senantiasa memberikan dukungan doa dan senantiasa menjadi pribadi yang bersahabat dengan saya. Serta adik-adik saya Dimas Bintang Pangestu dan Divana Ayu Rudiana Putri yang saya sayangi.
2. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D, selaku pembimbing satu yang telah membimbing yang sabar, solutif, dan memberikan contoh sebagai pribadi yang profesional dalam memberikan bimbingan kepada mahasiswa.
3. Bapak Nanang Yudi Setiawan, S.Kom, M.Kom, selaku pembimbing dua yang mengajarkan ketegasan dan kedisiplinan dalam menyelesaikan kewajiban. Memberikan contoh sebagai pribadi yang sabar, solutif, dan profesional dalam memberikan bimbingan kepada mahasiswa.
4. Bapak dan Ibu Dosen Fakultas Ilmu Komputer, Universitas Brawijaya yang telah memberikan ilmu kepada penulis selama proses perkuliahan hingga penulis mampu menyelesaikan skripsi.
5. Bapak Rudy dan Bapak Rizal yang telah memberikan izin penulis untuk melakukan penelitian pada PT. Indosat Ooredoo Surabaya.
6. Pendamping tercinta yang telah menemani saya selama ini dalam perkuliahan dan menemani saya dalam keadaan susah maupun senang Rizki Indra Fanani.
7. Sahabat-sahabat terkasih saya yang selalu menemani Siti Saudah, Hilwa Aminatus, Ikrima Nuha, Anita P.S, Irfan Asnawi.
8. Teman-teman Jurusan Sistem Informasi, Fakultas Ilmu Komputer, yang tidak dapat saya sebutkan namanya satu per satu, yang telah memberikan saya



dukungan moral, menjadi teman dan sahabat yang baik dan menyenangkan selama masa perkuliahan.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada proses penelitian ini. Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan, diskusi mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya.

Malang, 3 Januari 2019

Penulis

Pdinda903@gmail.com



ABSTRAK

Dinda Ayu Rudyana Putri, Pengembangan Sistem Informasi Pendataan Rak dan Perangkat Server Berbasis Web (Studi Kasus pada PT. Indosat Ooredoo Surabaya)

Pembimbing: Ismiarta Aknuranda, S.T, M.Sc, Ph.D. dan Nanang Yudi Setiawan S.Kom., M.Kom

PT. Indosat Ooredoo merupakan salah satu perusahaan penyedia layanan telekomunikasi di Indonesia. Semakin berkembangnya teknologi, Indosat juga mengembangkan layanan jaringannya hingga kini telah mencapai 4Gplus. Dengan demikian, proses telekomunikasi secara cepat bisa dilakukan. Untuk mendukung kelancaran pelayanan yang maksimal, tentu diperlukan pengelolaan ruang server jaringan yang lebih baik. Dapat dikatakan server merupakan bagian yang sangat vital dalam aktivitas jaringan telekomunikasi. Pada PT. Indosat Ooredoo Surabaya terdapat masalah pada Divisi Teknik yaitu belum ada sumber daya yang mampu mengelola ruangan server dengan baik. Kegiatan pendataan pada setiap rak dan perangkat server masih menggunakan teknik manual dengan menggunakan Microsoft Word dan menyalin data tersebut kedalam aplikasi penggambaran 3D dengan menggunakan bantuan aplikasi SketchUp.

Berdasarkan uraian permasalahan diatas, maka diperlukan perbaikan pada proses pendataan informasi isi dari ruangan server yang telah dilakukan oleh petugas server dengan mempertimbangkan perbaikan menggunakan sistem berbasis komputer yaitu membangun sebuah sistem informasi pendataan rak dan perangkat server. Untuk mengembangkan sistem informasi permasalahan diatas maka peneliti menggunakan salah satu metode yaitu *Waterfall Model*. *Waterfall Model* konsisten dengan fase yang dimilikinya dan dokumentasi dihasilkan pada setiap fase. Masalah utamanya adalah partisi yang tidak fleksibel dari proyek menjadi tahap-tahap yang berbeda. Oleh karena itu, untuk mencegah kesalahan dalam menentukan persyaratan kebutuhan pengguna maka peneliti menggunakan pendekatan *Object-oriented Analysis and Design (OOAD)*. Setelah analisis persyaratan dan perancangan dilakukan selanjutnya melakukan pengujian pada implementasi sistem informasi pendataan rak dan perangkat server dengan menggunakan pengujian fungsional dan nonfungsional. Pada pengujian fungsional menggunakan uji validasi dengan hasil 100% valid dan pengujian nonfungsional menggunakan uji validasi kompatibilitas peramban dengan menggunakan software SortSite. Dari hasil pengujian perangkat lunak tersebut dapat disimpulkan bahwa petugas server setuju dengan sistem.

Kata kunci: pengembangan, sistem informasi, waterfall model, pendataan rak dan perangkat server

ABSTRACT

Dinda Ayu Rudyana Putri, Pengembangan Sistem Informasi Pendataan Rak dan Perangkat Server Berbasis Web (Studi Kasus pada PT. Indosat Ooredoo Surabaya)

Pembimbing: Ismiarta Aknuranda, S.T, M.Sc, Ph.D. dan Nanang Yudi Setiawan S.Kom., M.Kom

PT. Indosat Ooredoo is one of the telecommunications service providers in Indonesia. As technology develops, Indosat also develops its network services to reach 4Gplus. Thus, telecommunication processes can be done quickly. To support the smooth running of services, it is certainly necessary to better manage network server space. It can be said that the server is a very vital part of telecommunication network activity. At PT. Surabaya Indosat Ooredoo has a problem in the Engineering Division, namely there are no resources capable of managing the server room properly. Data collection activities on each rack and server device still use manual techniques by using Microsoft Word and copying the data into 3D drawing applications using the help of the SkechUp application.

Based on the description of the above problems, it is necessary to improve the data collection process on the contents of the server room that has been carried out by server officers by considering improvements using a computer-based system that is building an information system for rack and server devices. To develop the problem information system above, the researcher used one of the methods, namely the Waterfall Model. The Waterfall Model is consistent with the phase it has and documentation is generated at each phase. The main problem is that inflexible partitions from the project are different stages. Therefore, to prevent errors in determining the requirements of user requirements, the researcher uses the Object-oriented Analysis and Design (OOAD) approach. After the requirements analysis and design are carried out, the next step is to test the implementation of information systems on rack and server data collection using functional and nonfunctional tests. On functional testing using a validation test with 100% valid results and nonfunctional testing using the browser compatibility validation test using SortSite software. From the results of the software testing it can be concluded that the server clerk agrees with the system.

Keywords: *development, information system, waterfall model, rack and server device data collection*

DAFTAR ISI

PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	4
1.4 Manfaat.....	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	5
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Landasan Kepustakaan	6
2.2 Profil PT Indosat Ooredoo Surabaya.....	7
Logo Perusahaan	7
Visi dan Misi	8
Struktur Organisasi.....	8
2.3 Sketchup	8
2.3.1 Fungsi-fungsi tools skethcup.....	9
2.4 Proses Bisnis.....	9
2.5 Business Proses Modeling Notation (BPMN).....	10
2.5.1 Flow Object	10
2.5.2 Data	11
2.5.3 Connections <i>Objects</i>	11
2.5.4 Swimlanes	12
2.5.5 Artifacts	12
2.6 Pemodelan <i>Use Case</i>	13

2.6.1 Identifikasi Tipe Pemangku Kepentingan.....	14
2.6.2 Analisis Masalah	15
2.6.3 Identifikasi Kebutuhan Pemangku Kepentingan.....	15
2.6.4 Identifikasi Fitur	16
2.6.5 Identifikasi Persyaratan Fungsional dan Nonfungsional.....	16
2.6.6 Pemodelan <i>Use Case</i>	17
2.7 <i>Software Development Life Cycle</i> (SLDC)	18
2.7.1 <i>Waterfall Model</i>	18
2.8 Unified Model Language (UML)	19
2.8.1 <i>Use Case Diagram</i>	19
2.8.2 <i>Activity Diagram</i>	21
2.8.3 <i>Sequence Diagram</i>	23
2.8.4 <i>Class Diagram</i>	26
2.9 <i>Physical Data Model</i>	28
2.10 <i>Model View Controller</i> (MVC)	28
2.11 Framework.....	29
2.11.1 <i>Hypertext Markup Language</i> (HTML)	29
2.11.2 <i>Casanding Style Sheet</i> (CSS)	30
2.11.3 <i>Hypertext Processor</i> (PHP)	30
2.12 <i>Codeigniter</i> (CI)	31
2.13 Pengujian Perangkat Lunak	31
2.13.1 Pengujian Validasi	32
2.13.2 Pengujian Kompatibilitas	32
BAB 3 METODOLOGI PENELITIAN	33
3.1 Studi Literatur	33
3.2 Analisis Persyaratan dan Pemodelan Proses Bisnis.....	33
3.3 Perancangan	34
3.4 Implementasi	34
3.5 Pengujian	35
3.6 Kesimpulan dan saran.....	35
BAB 4 ANALISIS PERSYARATAN DAN PEMODELAN PROSES BISNIS	36
4.1 Pemodelan Proses Bisnis	36



4.1.1	Proses Bisnis <i>As-is</i>	36
4.1.2	Analisis Masalah Pendataan Rak <i>Server</i> dan Perangkat <i>Server</i> ..	40
4.1.3	Proses Bisnis <i>To-be</i>	41
4.2	Analisis Persyaratan	46
4.2.1	Identifikasi Tipe Pemangku Kepentingan	46
4.2.2	Identifikasi Kebutuhan Pemangku Kepentingan	47
4.2.3	Identifikasi Pengguna	48
4.2.4	Identifikasi Fitur	49
4.2.5	Persyaratan Fungsional	50
4.2.6	Persyaratan Nonfungsional	52
4.3	Pemodelan Use case	52
4.3.1	Use case Diagram	53
4.3.2	Deskripsi Aktor	54
4.3.3	Use Case Specification	55
BAB 5	PERANCANGAN	66
5.1	Perancangan Arsitektur	66
5.2	Pemodelan Interaksi Objek	66
5.2.1	Sequence Diagram Login	66
5.2.2	Sequence Diagram Mengelola Rak	67
5.2.3	Sequence Diagram Mengelola Perangkat	69
5.2.4	Sequence Diagram Mengelola Lantai	70
5.2.5	Sequence Diagram Mengelola User	71
5.2.6	Sequence Diagram Memantau Rak dan Perangkat <i>Server</i>	72
5.3	Pemodelan Class	74
5.4	Perancangan Basis Data	75
5.4.1	Tabel Indosat Perangkat	76
5.4.2	Tabel Indosat Rak	76
5.4.3	Tabel Indosat Lantai	77
5.4.4	Tabel Indosat Users	77
5.5	Perancangan Algoritme	77
5.5.1	Menambah Rak <i>Server</i>	77
5.5.2	Menambah Perangkat	78

5.5.3 Mengelola User	79
5.6 Perancangan Antarmuka	79
5.6.1 Antarmuka Login	79
5.6.2 Antarmuka Mengelola Rak.....	80
5.6.3 Antarmuka Mengelola Perangkat	81
5.6.4 Antarmuka Mengelola Lantai.....	82
5.6.5 Antarmuka Mengelola User	83
BAB 6 IMPLEMENTASI	84
6.1 Lingkungan Implentasi	84
6.1.1 Lingkungan Perangkat Keras	84
6.1.2 Lingkungan Perangkat Lunak	84
6.2 Implementasi <i>Class</i>	85
6.3 Implementasi Program	85
6.4 Implementasi Antarmuka	87
6.4.1 Antamuka Login	88
6.4.2 Antarmuka Mengelola Rak.....	88
6.4.3 Antarmuka Mengelola Perangkat	89
6.4.4 Antarmuka Mengelola Lantai.....	91
6.4.5 Antarmuka Mengelola User	91
6.4.6 Antarmuka Memantau Rak dan Perangkat.....	92
BAB 7 PENGUJIAN	94
7.1 Pengujian Fungsional	94
7.2 Pengujian Nonfungsional	96
BAB 8 PENUTUP	97
8.1 Kesimpulan.....	97
8.2 Saran	98
DAFTAR PUSTAKA.....	99
LAMPIRAN A HASIL WAWANCARA.....	101
LAMPIRAN B HASIL PENGUJIAN	105

DAFTAR TABEL

Tabel 2.1 Notasi <i>Event</i>	10
Tabel 2.2 Notasi Data	11
Tabel 2.3 Notasi <i>Connections Objects</i>	12
Tabel 2.4 Notasi Swimlanes	12
Tabel 2.5 Simbol-Simbol <i>Artifacts</i> BPMN.....	13
Tabel 2.6 Format Dokumentasi Spesifikasi <i>Use Case</i>	14
Tabel 2.7 Tabel Analisis	15
Tabel 2.8 Contoh Tabel Fitur	16
Tabel 2.9 Format Dokumentasi Spesifikasi <i>Use Case</i>	17
Tabel 2.10 Notasi pada <i>Use Case Diagram</i>	20
Tabel 2.11 Notasi pada Activity Diagram	21
Tabel 2.12 Notasi pada <i>Sequence Diagram</i>	24
Tabel 2.13 Notasi pada <i>Class Diagram</i>	27
Tabel 4.1 Analisis Masalah	40
Tabel 4.2 Perubahan Aktivitas Proses Bisnis.....	45
Tabel 4.3 Tabel Tipe Pemangku Kepentingan	47
Tabel 4.4 Analisis Kebutuhan Nonfungsional	52
Tabel 4.5 Tabel Hubungan Use Case dengan Pengguna	53
Tabel 4.6 Hubungan Use Case dengan Fitur	54
Tabel 4.7 Identifikasi Aktor	54
Tabel 4.8 Spesifikasi <i>Use Case Login</i>	55
Tabel 4.9 Spesifikasi <i>Use Case Mengelola Rak</i>	56
Tabel 4.10 Spesifikasi <i>Use Case Mengelola Perangkat</i>	58
Tabel 4.11 Spesifikasi <i>Use Case Mengelola Lantai</i>	60
Tabel 4.12 Spesifikasi <i>Use Case Mengelola User</i>	62
Tabel 4.13 Spesifikasi <i>Use Case Memantau Rak dan Perangkat</i>	64
Tabel 5.1 Tabel Indosat Perangkat	76
Tabel 5.2 Tabel Indosat Rak	76
Tabel 5.3 Tabel Indosat Lantai	77
Tabel 5.4 Tabel Indosat <i>Users</i>	77
Tabel 5.5 Tabel Pseudocode Menambah Rak <i>Server</i>	78



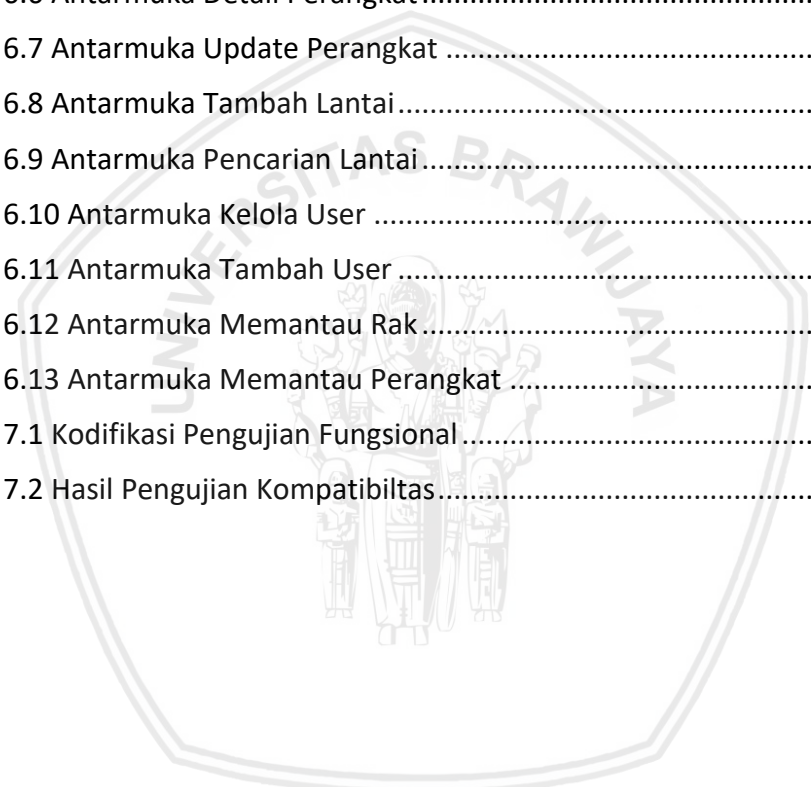
Tabel 5.6 Menampilkan Informasi Perangkat	78
Tabel 6.1 Implementasi <i>Class</i> pada folder kode program	85
Tabel 6.2 Kode Program Tambah Rak	85
Tabel 6.3 Kode Program Semua Perangkat	86
Tabel 6.4 Kode Program User	87
Tabel 7.1 Rencana Pengujian Validasi Fungsi <i>Login</i>	94
Tabel 7.2 Rencana Pengujian Validasi Fungsi Menambah Rak	95
Tabel 7.3 Validation Testing Mengelola Perangkat	95



DAFTAR GAMBAR

Gambar 2.1 Logo Perusahaan Indosat Ooredoo.....	7
Gambar 2.2 Struktur Organisasi PT. Indosat Ooredoo	8
Gambar 2.3 <i>Waterfall Model</i>	18
Gambar 2.4 Contoh <i>Use Case Diagram</i>	20
Gambar 2.5 Contoh <i>Activity Diagram</i>	21
Gambar 2.6 Contoh <i>Sequence Diagram</i>	24
Gambar 3.1 Metodologi Penelitian.....	33
Gambar 4.1 Proses Bisnis <i>As-is</i> Pendataan Rak dan Perangkat <i>Server</i>	39
Gambar 4.2 Proses Bisnis <i>To-be</i> Pendataan Rak dan Perangkat <i>Server</i>	44
Gambar 4.3 Identifikasi Kebutuhan Pemangku Kepentingan.....	47
Gambar 4.4 Identifikasi Fitur	49
Gambar 4.5 Persyaratan Fungsional	50
Gambar 4.6 Persyaratan Nonfungsional.....	52
Gambar 4.7 Use case Diagram PT Indosat Ooredoo Surabaya.....	53
Gambar 4.8 Deskripsi Aktor	54
Gambar 5.1 Desain Arsitektur Sistem Informasi Pendataan Rak dan Perangkat .	66
Gambar 5.2 <i>Sequence Diagram Login</i>	67
Gambar 5.3 <i>Sequence Diagram</i> Mengelola Rak <i>Server</i>	68
Gambar 5.4 <i>Sequence Diagram</i> Mengelola Perangkat	70
Gambar 5.5 <i>Sequence Diagram</i> Mengelola Lantai.....	71
Gambar 5.6 <i>Sequence Diagram</i> Mengelola User	72
Gambar 5.7 <i>Sequence Diagram</i> Memantau Rak dan Perangkat	73
Gambar 5.8 Diagram Kelas Perancangan <i>Controller</i>	74
Gambar 5.9 <i>Class Diagram</i> untuk Domain <i>Model</i>	75
Gambar 5.10 Diagram Kelas Relasi <i>Controller</i> dengan <i>Model</i>	75
Gambar 5.11 <i>Physical Data Model</i> Sistem Informasi Rak dan Perangkat <i>Server</i> ..	76
Gambar 5.12 Antarmuka <i>Login</i>	80
Gambar 5.13 Antarmuka Daftar Rak.....	80
Gambar 5.14 Antarmuka Tambah Rak.....	81
Gambar 5.15 Antarmuka Daftar Perangkat	81
Gambar 5.16 Antarmuka Tambah Perangkat	82

Gambar 5.17 Antarmuka Cari Lantai.....	82
Gambar 5.18 Antarmuka Tambah Lantai.....	83
Gambar 5.19 Antarmuka Tambah User	83
Gambar 6.1 Antarmuka Login	88
Gambar 6.2 Antarmuka Daftar Rak Petugas <i>Server</i>	88
Gambar 6.3 Antarmuka Tambah Rak.....	89
Gambar 6.4 Antarmuka Daftar Perangkat Petugas <i>Server</i>	89
Gambar 6.5 Antarmuka Tambah Perangkat	89
Gambar 6.6 Antarmuka Detail Perangkat.....	90
Gambar 6.7 Antarmuka Update Perangkat	90
Gambar 6.8 Antarmuka Tambah Lantai.....	91
Gambar 6.9 Antarmuka Pencarian Lantai.....	91
Gambar 6.10 Antarmuka Kelola User	91
Gambar 6.11 Antarmuka Tambah User	92
Gambar 6.12 Antarmuka Memantau Rak.....	92
Gambar 6.13 Antarmuka Memantau Perangkat	93
Gambar 7.1 Kodifikasi Pengujian Fungsional.....	94
Gambar 7.2 Hasil Pengujian Kompatibilitas.....	96



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	101
A.1 Hasil Wawancara dengan Petugas <i>Server</i>	101
A.2 Hasil Wawancara dengan Kepala Teknik Backboud.....	103
LAMPIRAN B HASIL PENGUJIAN	105
B.1 Hasil Kuisiner Pengujian Validasi	105



BAB 1 PENDAHULUAN

1.1 Latar belakang

Arus perkembangan teknologi yang semakin maju menyebabkan kebutuhan informasi menjadi semakin penting. Dengan adanya perkembangan teknologi, menyebabkan penyebaran informasi dan data sangatlah cepat dan mudah. Banyak perusahaan berskala besar maupun kecil menggunakan teknologi internet untuk mendukung kegiatan dalam perusahaannya. Teknologi internet memiliki keuntungan dalam penyampaian informasi yang lebih cepat, dapat diakses oleh siapa saja, kapan saja dan dimana saja. Internet dapat diartikan sebagai suatu jaringan komputer dengan skala luas yang menghubungkan antara pemakai komputer, dari satu komputer dengan komputer lainnya (Kusumawati, 2013). Banyak cara dalam memanfaatkan teknologi internet, situs web merupakan salah satu teknologi internet yang terdiri dari kumpulan halaman-halaman web yang berfungsi sebagai media penyampaian informasi, komunikasi, hiburan, atau transaksi (Kusumawati, 2013). Situs web dapat dimanfaatkan untuk banyak hal seperti pembuatan profil perusahaan, situs jual beli, sistem *e-complaint*, sistem informasi, dan lain sebagainya.

Sistem informasi adalah salah satu teknologi informasi yang menyediakan informasi untuk membantu pengambilan keputusan manajemen dan memberikan informasi yang layak (Rizki, et al., 2014). Sehingga dapat disimpulkan bahwa sistem informasi adalah sebuah sistem yang dapat mengelola data baik yang sederhana maupun yang kompleks untuk dijadikan sebuah informasi yang dapat dimanfaatkan sebagai pengambilan keputusan. Dengan besarnya fungsi sistem informasi hampir seluruh perusahaan yang berskala besar maupun kecil menggunakan sistem informasi agar kinerja yang dilakukan jauh lebih efektif dan efisien. Banyak sekali macam dari sistem informasi seperti sistem informasi akademik, sistem informasi transaksi penjualan, sistem informasi persediaan barang, sistem informasi kepegawaian, sistem informasi antrian, dan lain-lain.

PT. Indosat Ooredoo merupakan salah satu perusahaan penyedia layanan telekomunikasi di Indonesia. Semakin berkembangnya teknologi, Indosat juga mengembangkan layanan jaringannya hingga kini telah mencapai 4Gplus. Dengan demikian, proses telekomunikasi secara cepat bisa dilakukan. Untuk mendukung kelancaran pelayanan yang maksimal, tentu diperlukan pengelolaan ruang *server* jaringan yang lebih baik. *Server* adalah seperangkat sistem komputer yang menyediakan layanan untuk program atau *device* tertentu yang disebut *client* yang dalam sebuah jaringan komputer. Bisa dikatakan *server* merupakan bagian yang sangat vital dalam aktivitas jaringan telekomunikasi.

Berdasarkan hasil wawancara dengan bapak Rizal dari Divisi Teknik, terdapat beberapa masalah yang mengganggu jalannya proses bisnis pada PT. Indosat Ooredoo Surabaya. Salah satunya, pada Divisi Teknik belum adanya sumber daya yang mampu mengelola ruangan *server* dengan baik. Pihak dari Indosat belum ada upaya untuk menanggulangi masalah yang terjadi. Selama ini Divisi Teknik hanya

memiliki petugas *server* yang mempunyai tugas yaitu memeriksa setiap ruangan *server* dan mengontrol isi tiap rak dari ruangan *server*. Namun, kegiatan pendataan nama, isi, kuota yang ada pada setiap rak *server* tersebut masih menggunakan teknik manual dengan mencatat data rak dan perangkat menggunakan word dan menyalin ke aplikasi penggambaran 3D dengan menggunakan bantuan aplikasi SkechUp untuk memudahkan pencarian posisi dan letak *server*. Hal ini tentu sangat tidak praktis dalam hal pencarian, karena gambar tersebut tidak mendukung pencarian nama perangkat secara spesifik dan harus mencari satu persatu dari banyak ruang yang ada. Petugas *server* itu sendiri juga jarang mengupdate keluar masuknya isi *server* pada setiap ruangan.

Berdasarkan uraian permasalahan diatas, maka diperlukan perbaikan pada proses pendataan informasi isi dari ruangan *server* yang telah dilakukan oleh petugas *server* dengan mempertimbangkan perbaikan menggunakan sistem berbasis komputer yaitu membangun sebuah sistem informasi pendataan rak dan perangkat *server*. Sistem informasi tersebut merupakan aplikasi berbasis *website*, yang dapat membantu dalam proses pendataan sekaligus pemantauan kegiatan masuknya *server* dan perangkat baru maupun *server* dan perangkat keluar pada PT. Indosat Ooredoo Surabaya secara elektronik yang memiliki keunggulan dapat melakukan penyimpanan informasi rak *server*, menyimpan informasi perangkat *server*, mengupdate data informasi isi rak *server*, menambahkan isi rak *server*, menghapus isi rak *server*, melihat lantai mana saja yang sudah terisi, dan mencari perangkat dari rak *server*.

Untuk mengembangkan sistem informasi permasalahan diatas maka peneliti menggunakan salah satu metode yaitu *Waterfall Model*. *Waterfall Model* merupakan model pertama dan merupakan siklus hidup perangkat lunak. *Waterfall Model* mempunyai 5 fase untuk mencerminkan kegiatan pengembangan sistem informasi yaitu *Requirement, System and Software Design, Implementation and Unit Testing, Intregation and System Testing* serta *Operation and Maintanance* (Sommerville, 2011). Untuk menghasilkan sistem informasi yang layak dengan menggunakan metode *Waterfall Model* harus melewati 5 fase tersebut dengan diawali dengan fase menganalisis kebutuhan pengguna. Setelah menganalisis kebutuhan pengguna fase berikutnya perancangan sistem yang dilakukan sesuai berdasarkan analisis kebutuhan pengguna dan kemudian menerapkan pada fase implementasi melalui kode program yang sesuai dengan hasil perancangan. Setelah sistem informasi dibuat maka membutuhkan tahap pengujian. Hal ini dilakukan untuk mengukur kelayakan perangkat lunak yang dibuat. Ketika perangkat lunak dinyatakan layak dan dapat digunakna oleh pengguna maka masuk pada fase terakhir yaitu pemeliharaan. Sebenarnya fase ini tidak harus dilakukan karena mempunyai siklus hidup yang lama. *Waterfall Model* konsisten dengan fase yang dimilikinya dan dokumentasi dihasilkan pada setiap fase. Masalah utamanya adalah partisi yang tidak fleksibel dari proyek menjadi tahap-tahap yang berbeda. Komitmen harus dibuat terlebih dahulu pada tahap awal dalam proses, yang membuatnya sulit untuk menanggapi perubahan persyaratan pelanggan. Pada prinsipnya, *Waterall Model* seharusnya hanya digunakan ketika persyaratan kebutuhannya dipahami dengan baik dan tidak

mungkin berubah (Sommerville, 2011). Oleh karena itu, untuk mencegah terjadinya kesalahan dalam menentukan persyaratan kebutuhan pengguna maka peneliti menggunakan pendekatan *Object-oriented Analysis and Design* (OOAD). Masalah yang dihadapi PT. Indosat Ooredoo Surabaya yang telah dijabarkan diatas memiliki persyaratan atau keperluan yang nantinya harus dipenuhi oleh sebuah sistem informasi. Peneliti menerapkan metode OOAD untuk menganalisis permasalahan yang terjadi dan mengobservasi masalah tersebut agar berkesinambungan antara masalah dengan kebutuhan yang diperlukan.

Krol dan Reich (1999) menjelaskan bahwa OOAD merupakan metode yang menjadi bagian dari aktivitas pengembangan sistem berorientasi objek yang memiliki keunggulan dalam menjamin konsistensi antara rancangan dengan komponen sistem. Menurut Aggarwal (2002), aktivitas yang dapat dilakukan untuk mencapai tujuan OOA adalah dengan memodelkan fungsi sistem dan mengidentifikasi objek-objek sistem, memodelkan objek-objek dan mengidentifikasi hubungan antar objek, serta memodelkan perilaku objek. Untuk memfasilitasi kebutuhan komunikasi mengenai sistem melalui model visual dan memungkinkan persyaratan sistem dapat mudah dipahami oleh pemangku kepentingan, maka Unified Modeling Language (UML) dapat digunakan sebagai bahasa pemodelan (Aggarwal, 2002). Aggarwal (2002) menjelaskan bahwa UML merupakan hasil unifikasi dari beberapa pendekatan pengembangan sistem berorientasi objek yang menawarkan standar pemodelan objek untuk menspesifikasi, memvisualisasi, merancang, dan mendokumentasikan artefak sistem. Aggarwal (2002) menambahkan bahwa UML menjamin efektivitas komunikasi antara pengguna dengan analis, sehingga dapat memudahkan analis mendokumentasikan persyaratan pengguna.

Berdasarkan penjelasan tersebut, penelitian ini dilakukan untuk mengembangkan sistem informasi pendataan rak dan perangkat *server* dengan menerapkan serangkaian tahap analisis, perancangan, dan implementasi berorientasi objek. Selanjutnya, hasil pengembangan sistem informasi pendataan rak dan perangkat *server* menunjukkan bahwa hasil sistem informasi yang dikembangkan sesuai dengan persyaratan yang telah diidentifikasi dibuat dengan untuk mendukung PENGEMBANGAN SISTEM INFORMASI PENDATAAN RAK DAN PERANGKAT *SERVER* BERBASIS WEB PADA PT. INDOSAT OOREDOO SURABAYA.

1.2 Rumusan masalah

Berdasarkan latar belakang diatas, maka diperoleh rumusan masalah penelitian ini. Penentuan rumusan masalah bertujuan agar mempermudah dan membatasi ruang lingkup dari penelitian sehingga penelitian akan tetap dalam jalur dan tidak melebar. Rumusan masalah sebagai berikut:

1. Bagaimana hasil analisis persyaratan dan pemodelan proses bisnis sistem informasi pendataan rak dan perangkat *server* pada PT. Indosat Ooredoo Surabaya?
2. Bagaimana hasil perancangan sistem informasi pendataan rak dan perangkat *server* pada PT. Indosat Ooredoo Surabaya?

3. Bagaimana hasil implementasi sistem informasi pendataan rak dan perangkat *server* pada PT. Indosat Ooredoo Surabaya?
4. Bagaimana hasil pengujian *fungsiional dan nonfungsiional* sistem informasi pendataan rak dan perangkat *server* pada PT. Indosat Ooredoo Surabaya?

1.3 Tujuan

Berdasarkan uraian rumusan masalah, dapat diketahui tujuan dilaksanakan penelitian ini sebagai berikut:

1. Menganalisis sistem informasi kebutuhan data pada *server* masuk dan *server* keluar pada PT. Indosat Ooredoo.
2. Merancang sistem informasi data rak *server* masuk dan *server* keluar pada PT. Indosat Ooredoo.
3. Mengimplentaskani sistem informasi data rak *server* masuk dan *server* keluar pada PT. Indosat Ooredoo.
4. Menguji dengan metode *fungsiional dan nonfungsiional* terhadap sistem informasi data rak *server* masuk dan *server* keluar pada PT. Indosat Ooredoo.

1.4 Manfaat

Manfaat dari pengembangan sistem informasi pendataan rak dan perangkat *server* adalah penelitian pengembangan sistem informasi ini, penulis berharap dapat membantu mempermudah kegiatan pendataan rak *server* masuk dan *server* keluar, membantu pemetaan rak *server* dengan baik, lebih mudah dalam pencarian nama dan isi pada rak *server* pada PT. Indosat Ooredoo Surabaya. Dengan demikian, pembuatan sistem informasi berbasis web ini dapat mengurangi *cost* yang dikeluarkan oleh PT. Indosat Jakarta.

1.5 Batasan masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Penelitian hanya pada kegiatan pendataan satu ruangan *server* pada PT. Indosat Ooredoo.
2. Penelitian ini menggunakan model pengembangan *Software Development Life Cycle (SDLC)* bernama *Waterfall Model* yang difokuskan pada pembuatan sistem dengan tahapan perancangan, implementasi dan pengujian dan pendekatan *Object-oriented Analysis and Design (OOAD)* untuk menganalisis dan melakukan perancangan.
3. Sistem yang dibangun berbasis web dengan menggunakan program *PHP, HTML, CSS* dengan menggunakan framework *Codeigniter* dan menggunakan database *MySql*.

1.6 Sistematika pembahasan

Gambaran secara garis besar pembahasan dari keseluruhan untuk setiap bab adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab pendahuluan membahas mengenai latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan berisi uraian dan pembahasan tentang kajian pustaka yang berkaitan dengan Sistem Informasi Pendataan Rak dan Perangkat Server dan dasar teori yang mendukung tema penelitian.

BAB 3 METODOLOGI PENELITIAN

Bab metodologi penelitian menguraikan tentang metodologi yang akan digunakan dalam analisis perancangan dan implementasi.

BAB 4 ANALISIS PERSYARATAN DAN PEMODELAN PROSES BISNIS

Bab analisis pesyaratan dan pemodelan proses bisnis dan ini menguraikan tentang tentang proses bisnis yang saat ini sedang berjalan (*as-is*) dan proses bisnis yang akan datang (*to-be*), menganalisis masalah yang lebih mendetail, mengidentifikasi persyaratan/kebutuhan perangkat lunak secara fungsional dan non-fungsional dan pemodelan use case, spesifikasi *use case*.

BAB 5 PERANCANGAN

Bab perancangan ini menguraikan bagaimana melakukan perancangan berdasarkan analisis kebutuhan pengguna yang telah ditentukan.

BAB 6 IMPLEMENTASI

Bab implementasi ini membahas tentang bagaimana cara implementasi sistem berdasarkan perancangan yang telah dibuat sebelumnya.

BAB 7 PENGUJIAN

Bab pengujian ini membahas tentang hasil pengujian dari sistem informasi yang telah dibangun dengan tujuan mendapatkan hasil yang diinginkan sesuai dengan kebutuhan.

BAB 8 PENUTUP

Bab penutup ini membahas tentang pengambilan kesimpulan dari bab yang telah dibahas dan yang telah dikerjakan beserta saran dari sistem informasi pendataan rak dan perangkat *server*.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Landasan Kepustakaan

Penelitian ini mengacu pada penelitian sebelumnya yang menggunakan metode *Waterfall Model* dalam pengembangan sistem informasi dan menggunakan pendekatan *Object-oriented Analysis and Design* (OOAD) pada proses analisis perancangan untuk mempermudah dan memahami terhadap perangkat lunak yang akan dibangun. Hasil dari perancangan tersebut akan dibuat acuan untuk implementasi sistem informasi pendataan rak dan perangkat *server* yang akan dikembangkan ke dalam penelitian ini.

Penelitian pertama merujuk pada I Gde Yuda Pratama dengan judul “Pengembangan Sistem Informasi Manajemen Perpustakaan Pada Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya”. Menurut Yuda, ada beberapa masalah yang dihadapi antara lain sistem informasi layanan perpustakaan masih bersifat *offline*, sehingga petugas mengalami kesulitan dalam mengelola peminjaman dan pengembalian buku. Pada penelitian, ini Yuda menggunakan metode *Waterfall Model* yang sesuai untuk mengembangkan sistem informasi manajemen perpustakaan dan menggunakan permodelan *Unified Modeling Language* (UML) untuk mempermudah perancangan terhadap sistem informasi yang akan dibangun. Aktivitas peneliti menggunakan *Waterfall Model* menghasilkan analisis kebutuhan kemudian melakukan perancangan sistem selanjutnya melakukan implementasi. Setelah mengimplementasi sistem informasi langkah selanjutnya melakukan pengujian. Penelitian ini menggunakan pengujian *black-box testing* berupa uji validasi untuk mengetahui sistem telah sesuai dengan kebutuhan pengguna. Selain melakukan pengujian tersebut, peneliti menggunakan pengujian berupa UAT (*User Acceptance Testing*) dan pengujian kapatibilitas (Pratama, 2018).

Penelitian kedua merujuk pada Fathir Izzudin Qitshi dengan judul “Pengembangan Sistem Informasi Kesehatan Primer untuk Mendukung Penerapan Model Organisasi dan Manajemen Kesehatan Primer ANDAL pada Poliklinik Pabrik Gula Kebon Agung”. Menurut Fathir, masalah yang dihadapi yaitu proses memahami persyaratan dan ekspektasi pengguna dan institusi kesehatan yang menyebabkan sistem informasi kesehatan tidak berjalan dengan linear. Penelitian ini menggunakan pengujian *black-box testing*, *Equivalen Partionin*, *Boundary Value Analysis*. Selain melakukan pengujian tersebut, peneliti menggunakan pengujian kapatibilitas (Fathir, 2018).

Perbandingan antara tiga *platform* model pengembangan perangkat lunak yang dilakukan oleh Adel Alshamrani dan Abdullah Bahattab dengan judul “A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model” *Waterfall Model* memiliki beberapa keuntungan yang diantaranya (Alshamrani & Bahattab, 2015) (1) mudah dipahami dan mudah melakukan implementasi; (2) diketahui dan banyak digunakan; (3) desain sistem jelas sebelum dibuat perangkat lunak; (4) penyusunan yang sistematis sehingga

mudah untuk melakukan implementasikan; (5) meminimalisir *overhead* perencanaan; (6) dan fase diproses dan diselesaikan satu per satu. Menurut (Alshamrani & Bahattab, 2015) *Waterfall Model* baik digunakan ketika kebutuhan sudah sangat diketahui, jelas, dan tetap. Pada penelitian yang dilakukan peneliti telah didapatkan bahwa kebutuhan dari pengguna sudah sangat jelas sehingga penulis akan menggunakan model *Waterfall* untuk pengembangan sistem informasi.

Keterkaitan penelitian diatas dengan peneliti yaitu melakukan penelitian pengembangan sistem informasi pendataan rak dan perangkat *server* dengan menggunakan metode *Waterfall Model* dan perancangan menggunakan *Unified Model Language* (UML) selain itu penelitian ini memiliki keterkaitan dengan pengujian sistem yang dilakukan oleh peneliti yaitu dengan menggunakan metode *fungsiional dan nonfungsiional*.

2.2 Profil PT Indosat Ooredoo Surabaya

Indosat didirikan pada tahun 1964 sebagai perusahaan penanaman modal asing pertama di Indonesia yang menyediakan layanan telekomunikasi internasional melalui satelit internasional. Sampai pada tahun 1980 Indosat berkembang menjadi perusahaan telekomunikasi internasional pertama yang dibeli dan dimiliki 100% oleh Pemerintah Indonesia. Dan pada tahun 1994 berhasil terdaftar di Bursa Efek Indonesia dan New York Stock Exchange dengan pemegang saham 65% untuk pemerintah Indonesia dan 35% untuk publik.

Melanjutkan keberhasilan, Indosat mengambil alih saham mayoritas Satelindo, operator selular dan SLI di Indonesia, dan mendirikan PT Indosat Multimedia Mobile (IM3) sebagai pelopor jaringan GPRS dan layanan multimedia di Indonesia. Selain itu Indosat juga bergabung dengan ketiga anak perusahaan yaitu, Satelindo, IM3, dan Bimagraha untuk membentuk operator selular di Indonesia.

Setahun kemudian, Indosat melakukan transformasi untuk menjadi perusahaan yang lebih fokus dan efisien dengan restrukturisasi organisasi, meodernisasi dan ekspsi jaringan selular serta inisiatif untuk mencapai keunggulan operasional. Perubahan terjadi pada tahun 2012, saat Indosat mencapai 58,5 Juta pelanggan yang didukung oleh peningkatan jaringan serta inovasi produk. Hingga pada tahun 2015 Indosat resmi berubah nama menjadi Indosat Ooredoo.

Logo Perusahaan



Gambar 2.1 Logo Perusahaan Indosat Ooredoo

Sumber: Indosat Ooredoo

Visi dan Misi

Visi

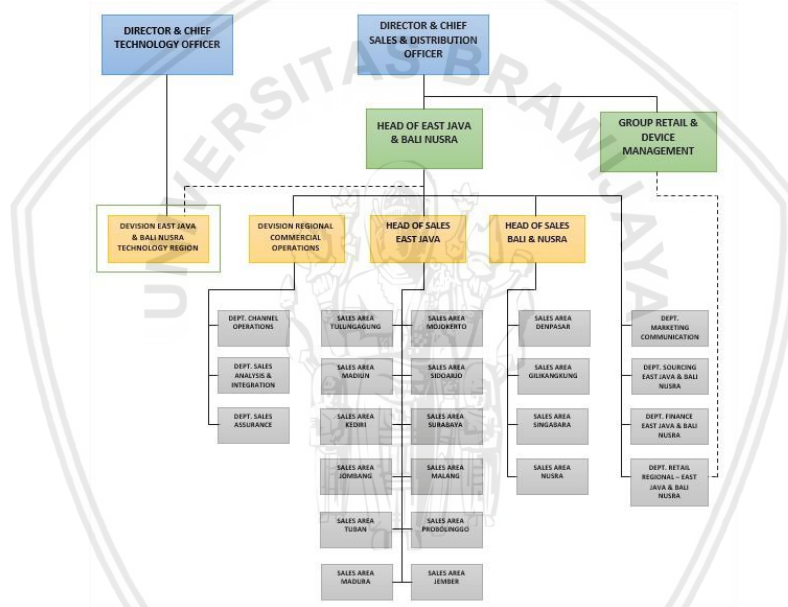
Menjadi perusahaan telekomunikasi digital terdepan di Indonesia.

Misi

1. Layanan produk yang membebaskan.
2. Jaringan data yang unggul.
3. Memperlakukan pelanggan sebagai sahabat.
4. Transformasi digital.

Struktur Organisasi

Berikut ini adalah sebuah struktur organisasi pada PT Indosat Ooredoo Surabaya.



Gambar 2.2 Struktur Organisasi PT. Indosat Ooredoo

Sumber: PT. Indosat Ooredoo

2.3 Sketchup

Skech Up adalah sebuah *software* yang berfungsi untuk melakukan desain dan model dalam bentuk 3D. Model yang dibuat bisa disimpan dalam harddisk lokal, maupun diupload di 3D warehouse yang bertautan dengan akun google masing-masing. Pada awalnya, Skech Up hanya digunakan untuk pembuatan desain bagi para arsitek. Namun pada perkembangannya, aplikasi ini mulai merambah ke hal lain. Seperti melakukan pencatatan secara nyata suatu kondisi ruangan.

2.3.1 Fungsi-fungsi tools skethcup

Untuk mengenal tools-nya bisa dimulai dari dalam software tersebut dengan cara pilih Windows > Instructor maka akan tampil sebuah jendela kecil yang didalamnya menerangkan fungsi tool tersebut.

1. Pull Up Down Menu: menu standar di setiap program, digunakan untuk menyimpan, membuka atau meng edit objek gambar, menambah tool pada toolbar.
2. Toolbar Standard: pada toolbar standard disini terdapat icon-icon perintah untuk menggambar, mengukur, menseleksi, memindahkan, menyisipkan, mewarnai, modifikasi.
3. Large Tool Set: Merupakan pengembangan dan penambahan dari Toolbar standard.
4. Measuremens: merupakan panel untuk mengetahui ukuran atau dimensi gambar/object.

2.4 Proses Bisnis

Definisi dari proses bisnis adalah sekumpulan aktivitas yang membutuhkan sebuah inputan dan dapat menghasilkan output nilai bagi pelanggan (Monk & Wagner, 2013). Monk & Wagner juga menjelaskan bahwa pelanggan untuk proses bisnis dapat dibagi menjadi 2 yaitu pelanggan eksternal dan pelanggan internal. Pengertian dari pelanggan eksternal adalah individu atau kelompok yang membeli produk sedangkan pelanggan internal adalah individu atau kelompok yang berada dalam berbagai macam departemen perusahaan.

Sebagai sebuah proses bisnis terdiri dari serangkaian kegiatan yang dilakukan dalam koordinasi di lingkungan bisnis dan teknis. Sangkaian kegiatan ini bersama-sama mewujudkan strategis bisnis baru. Suatu proses bisnis biasanya diberlakukan dalam suatu organisasi, tapi dapat juga saling berinteraksi dengan proses bisnis yang dilakukan oleh organisasi lain. Adanya manfaat proses bisnis yaitu mampu memudahkan pemahaman alur dari proses secara terintegrasi, sedangkan tujuan dari pemodelan proses bisnis adalah mampu mendefinisikan langkah yang perlu diambil untuk mencapai suatu tujuan bisnis (Weske, 2007).

Tujuan utama dari BPMN adalah untuk menyediakan suatu notasi yang mudah dipahami oleh semua pengguna bisnis, mulai dari bisnis analis yang membuat draft awal dari proses, para pengembang teknis yang bertanggung jawab untuk menerapkan teknologi yang akan melakukan proses-proses tersebut, hingga kepada orang-orang bisnis yang akan mengelola dan memantau proses mereka. Dengan demikian, BPMN menciptakan jembatan standar antara desain proses bisnis dan proses implementasi.

Spesifikasi ini merupakan penggabungan dari praktek terbaik dalam komunitas pemodelan bisnis untuk mendefinisikan notasi dan semantik Kolaborasi diagram, diagram Proses, dan diagram Koreografi. Maksud dari BPMN adalah untuk membakukan model bisnis proses dan notasi dalam menghadapi banyak notasi pemodelan dan sudut pandang yang berbeda. Dalam

melakukannya, BPMN akan menyediakan cara sederhana mengkomunikasikan informasi proses bisnis pengguna, proses pelaksana, pelanggan, dan pemasok.

Penggunaan Business Process diatur oleh pemerintah melalui peraturan Kementerian Pendayagunaan Aparatur Negara Dan Reformasi Birokrasi Nomor 12 Tahun 2011, yang isinya, Business Process Modeling Notation (BPMN) digunakan untuk memberikan acuan bagi kementerian atau lembaga pemerintah daerah dalam menyusun Standard Operating Procedures (SOP).

2.5 Business Proses Modeling Notation (BPMN)

Object Management Group (OMG) mendefinisikan BPMN adalah notasi sederhana yang digunakan untuk memberikan solusi terhadap proses bisnis sehingga menjadi proses pertukaran informasi antar pengguna bisnis dengan baik. Tujuan dari BPMN adalah memberikan notasi yang dimengerti oleh pengguna bisnis, dimulai dari analisis bisnis untuk mempersiapkan konsep bisnis hingga pengguna yang memiliki kepentingan untuk mengelola dan mengawasi proses bisnis berjalan.

BPMN dirancang untuk mempermudah mekanisme dalam pemodelan proses bisnis secara sederhana dan mudah dipahami, namun di saat yang bersamaan, BPMN juga mampu menangani visualisasi kompleksitas proses bisnis. Oleh karena itu, BPMN menyediakan sekumpulan notasi yang dibagi menjadi 5 kategori di antaranya adalah *flow objects*, *data*, *connecting objects*, *swimlanes*, *artifacts*.



2.5.1 Flow Object


Flow Object menjelaskan tentang perilaku yang dimiliki proses bisnis. *Flow Object* dibagi menjadi 3 yaitu *event*, *activity*, dan *gateway*. Berikut penjelasannya:

1. *Event*

Event merupakan notasi yang memberikan alur dari aktivitas. Suatu *event* dapat berasal dari internal dan eksternal suatu proses. *Event* dibagi menjadi tiga yaitu *start event*, *intermediate event*, dan *end event*. Setiap proses selalu memiliki sebuah *start event* untuk menunjukkan awal dari proses bisnis. Pada Tabel 2.1 merupakan penjelasan dari simbol *event*.

Tabel 2.1 Notasi Event

Notasi	Nama	Deskripsi
	<i>Start Event</i>	Notasi yang yang merepresentasikan sebuah proses dimulai.
	<i>Intermediate Event</i>	Notasi ini muncul di antara <i>start event</i> dan <i>end</i> , serta akan memengaruhi alur proses bisnis. Notasi ini tidak dapat digunakan untuk memulai dan mengakhiri sebuah proses secara langsung. Notasi ini memiliki beberapa macam varian untuk






Notasi	Nama	Deskripsi
		merepresentasikan event yang berbeda-beda. Setiap varian diindikasikan dengan penggunaan simbol-simbol yang berbeda di tengah notasi ini.
	<i>End</i>	Notasi yang merepresentasikan sebuah proses telah selesai.

Sumber : Object Management Group (2011)

2.5.2 Data

Data digunakan untuk menggambarkan bagaimana dokumen, data, dan objek lain digunakan dan diperbarui pada sebuah alur proses (IBM Knowledge Center, 2014). Kategori ini mempresentasikan 4 notasi yang terdiri dari *data objects*, *data inputs*, *data outputs*, *data stores*. Pada Tabel 2.2 menjelaskan tentang notasi data.

Tabel 2.2 Notasi Data



Notasi	Nama	Deskripsi
	<i>Data Object</i>	Notasi yang digunakan untuk memberikan informasi tambahan mengenai data atau dokumen atau objek lain yang berkaitan dengan aktivitas atau proses tertentu.
	<i>Data Object (Collection)</i>	Memvisualisasikan <i>data object</i> yang berjumlah lebih dari 1.
	<i>Data Input</i>	Notasi yang digunakan untuk memvisualisasikan data atau dokumen atau objek lain yang menjadi <i>input</i> sebuah atau serangkaian aktivitas.
	<i>Data Output</i>	Notasi yang digunakan untuk memvisualisasikan data atau dokumen atau objek lain yang menjadi <i>output</i> sebuah atau serangkaian aktivitas.
	<i>Data Store</i>	Notasi yang digunakan untuk memberikan informasi mengenai sebuah lokasi penyimpanan dan pembaruan data.

2.5.3 Connections Objects

Connections Objects merupakan notasi yang menghubungkan notasi *flow objects* dengan notasi BPMN lainnya. Pada Tabel 2.4 merupakan penjelasan dari simbol *connections*.



Tabel 2.3 Notasi Connections Objects



Notasi	Nama	Deskripsi
	<i>Pool</i>	Notasi yang digunakan untuk merepresentasikan partisipan sebuah proses bisnis. <i>Pool</i> dapat memiliki detail di dalamnya, sehingga seluruh aktivitas yang dimiliki oleh sebuah partisipan dapat dilihat pada model proses bisnis. Namun <i>pool</i> juga dapat bersifat <i>black-box</i> .
	<i>Lane</i>	<i>Lane</i> merupakan sub-partisi dari sebuah <i>pool</i> , notasi ini merepresentasikan unit-unit yang dimiliki oleh sebuah partisipan proses bisnis.

Sumber : Object Management Group (2011)

2.5.4 Swimlanes

Swimlanes merupakan jenis notasi yang digunakan untuk mengelompokkan aktivitas berdasarkan unit bisnis yang menjalankan aktivitas-aktivitas tertentu. *Swimlanes* pada umumnya digunakan untuk memberikan informasi yang jelas untuk aktivitas *business-to-business* (Object Management Group, 2011). Pada Tabel 2.4 menunjukkan notasi swimlanes





Tabel 2.4 Notasi Swimlanes

Notasi	Nama	Deskripsi
	<i>Pool</i>	Notasi yang digunakan untuk merepresentasikan partisipan sebuah proses bisnis. <i>Pool</i> dapat memiliki detail di dalamnya, sehingga seluruh aktivitas yang dimiliki oleh sebuah partisipan dapat dilihat pada model proses bisnis. Namun <i>pool</i> juga dapat bersifat <i>black-box</i> .
	<i>Lane</i>	<i>Lane</i> merupakan sub-partisi dari sebuah <i>pool</i> , notasi ini merepresentasikan unit-

2.5.5 Artifacts

Artifacts mempresentasikan sebuah objek diluar sebuah proses. *Artifact* dapat mempresentasikan data atau catatan yang menjelaskan sebuah proses atau dapat digunakan untuk mengelola tugas atau proses. Tabel 2.5 merupakan penjelasan simbol *artifacts*.

Tabel 2.5 Simbol-Simbol *Artifacts* BPMN

No	Nama	Gambar	Deskripsi
1.	<i>Data Object</i>		Mempresentasikan informasi yang mengalir pada sebuah proses seperti dokumen bisnis, surat, <i>email</i> dan lain-lain.
2.	<i>Data Store</i>		Tempat dimana proses dapat membaca atau menulis data
3.	<i>Annotation</i>		Menunjukkan informasi tambahan kepada pembaca sebuah diagram BPMN.
4.	<i>Group</i>		Memungkinkan untuk mengelompokkan elemen secara informal

Sumber : (Object Management Group, 2011)

2.6 Pemodelan *Use Case*

Model *use case* digunakan untuk mempresentasikan sistem dalam hal bagaimana penggunaan sistem tersebut (Bittner & Spence, 2002). Bittner dan Spence menambahkan, dalam menentukan *use case*, analis menggunakan persyaratan perangkat lunak sebagai panduan dan seringkali *use case* berisi persyaratan fungsional dari sebuah sistem. Model *use case* berisi dua komponen utama yaitu aktor dan *use Case*.

Aktor adalah representasi individu (manusia) atau objek lain yang berinteraksi dengan sistem. Aktor merupakan objek diluar sistem. Dalam model *use case*, aktor memiliki nama dan deskripsi singkat, selain itu, aktor dihubungkan dengan *use case* sesuai dengan interaksi aktor dengan *use case* tertentu.

Use case adalah representasi sebuah nilai yang diberikan oleh sistem kepada aktor. *Use case* tidak sama dengan fitur atau fungsi dari sebuah sistem. *Use case* memiliki nama dan penjelasan singkat. *Use case* juga memiliki spesifikasi yang lebih detail, yang berisi alur ketika aktor dan sistem berinteraksi.

Model *use case* tidak menyediakan deskripsi yang lengkap mengenai persyaratan sebuah sistem, oleh karena itu, model *use case* harus didukung dengan pemodelan-pemodelan persyaratan dan dokumen-dokumen persyaratan lainnya. Setiap *use case* di dalam model *use case* memiliki spesifikasi *use case* yang berisi penjelasan bagaimana aktor dan sistem berkolaborasi untuk memenuhi tujuan yang direpresentasikan oleh *use case* tersebut.

Tabel 2.6 Format Dokumentasi Spesifikasi *Use Case*

Brief Description	Berisi penjelasan singkat sebuah <i>use case</i> .
Actor	Berisi aktor yang berinteraksi dengan <i>use case</i> .
Pre-condition	Berisi satu atau lebih kondisi atau keadaan yang harus dipenuhi oleh sistem sebelum <i>use case</i> dapat dijalankan.
Post-condition	Berisi satu atau lebih kondisi atau keadaan yang dipenuhi oleh sistem ketika <i>use case</i> berakhir.
Extension	Berisi informasi penggunaan <i>use case</i> lain yang bersifat opsional, yang disertakan pada sebuah <i>use case</i> .
Basic Flow	Berisi alur normal yang dilalui ketika <i>use case</i> berjalan.
Alternative Flow	Berisi alur opsional yang akan dilalui ketika <i>use case</i> tidak berjalan sesuai dengan alur normal.
Subflow	Berisi beberapa kelompok alur yang merupakan penyederhanaan alur <i>use case</i> yang terlalu detail.

2.6.1 Identifikasi Tipe Pemangku Kepentingan

Pemangku kepentingan adalah seseorang yang secara materi terkena dampak dari hasil sebuah sistem atau proyek yang memproduksi sebuah sistem (Bittner & Spence, 2002). Identifikasi tipe pemangku kepentingan merupakan aktivitas analisis persyaratan yang digunakan untuk mengetahui siapa saja pemangku kepentingan dari sebuah sistem atau proyek, kemudian pemangku kepentingan tersebut akan dikelompokkan berdasarkan karakteristik dan hubungan pemangku kepentingan dengan sistem atau proyek yang sedang dikerjakan. Bittner dan Spence menjelaskan bahwa terdapat beberapa kategori pemangku kepentingan, yang terdiri dari:

1. Pengguna (*Users*), merupakan pemangku kepentingan yang berinteraksi secara langsung dengan sistem. Pengguna merupakan individu-individu yang akan memerankan peran sebagai aktor yang ada pada model *use case*.
2. Sponsor (*Sponsors*), merupakan pemangku kepentingan yang menanam investasi kedalam pengembangan sistem atau proyek. Pemangku kepentingan ini pada umumnya adalah pengguna tidak langsung dari sistem,
3. Pengembang (*Developers*), merupakan pemangku kepentingan yang terlibat dalam kegiatan produksi dan mendukung pengembangan sistem atau sebuah proyek.
4. Pihak yang berwenang (*Authorities*), merupakan pemangku kepentingan yang memiliki keahlian khusus dalam memberikan dukungan dalam pengembangan sistem dalam bidang regulasi internal atau eksternal sebuah perusahaan, teknologi, aturan perundangan, dan standar organisasi.

5. Customers (*Pelanggan*), merupakan pemangku kepentingan yang akan membeli sistem yang telah selesai dikembangkan.

Namun Bittner dan Spence menekankan bahwa kategori atau tipe pemangku kepentingan tidak terbatas pada kategori-kategori yang sudah dijelaskan sebelumnya. Tipe pemangku kepentingan bisa saja dikembangkan bergantung pada jenis organisasi atau perusahaan, macam-macam calon pengguna sistem, dan sebagainya.

2.6.2 Analisis Masalah

Analisis masalah adalah sebuah proses untuk memahami masalah yang aktual dan kebutuhan pengguna, serta proses untuk mengusulkan solusi untuk memenuhi kebutuhan pengguna (Leffingwell & Widrig, 2003). Leffingwell dan Widrig menjelaskan tujuan analisis masalah adalah untuk mendapatkan pemahaman yang lebih baik mengenai masalah yang akan diselesaikan sebelum proses pengembangan solusi (sistem) dimulai.

Cara terbaik untuk mendokumentasikan analisis masalah adalah dengan membangun pernyataan masalah (*problem statement*) (Bittner & Spence, 2002). Pernyataan masalah diperoleh dengan melakukan wawancara dengan beberapa pemangku kepentingan, kemudian pernyataan masalah dapat didokumentasikan dalam tabulasi sesuai contoh pada Tabel 2.7

Tabel 2.7 Tabel Analisis

<i>The problem of</i>	Berisi deskripsi masalah.
<i>Affects</i>	Berisi pemangku kepentingan yang terpengaruh oleh masalah.
<i>The impact of which is</i>	Berisi deskripsi dampak masalah terhadap pemangku kepentingan dan kegiatan bisnis.
<i>A successful solution would</i>	Berisi beberapa solusi dan manfaat utama yang ditawarkan untuk menyelesaikan masalah.

2.6.3 Identifikasi Kebutuhan Pemangku Kepentingan

Kebutuhan pemangku kepentingan dan pengguna atau secara singkat disebut kebutuhan pengguna adalah sesuatu yang dapat mewakili masalah bisnis, personal, atau operasional yang harus ditekankan dengan tujuan untuk membenarkan alasan untuk mempertimbangkan, membeli, atau menggunakan sistem baru (Leffingwell & Widrig, 2003). Memperoleh informasi mengenai kebutuhan pengguna dapat membantu dalam memahami bagaimana dan sejauh mana beberapa aspek yang berbeda dari sebuah masalah memengaruhi jenis pemangku kepentingan yang berbeda (Bittner & Spence, 2002).

Bittner dan Spence menambahkan bahwa identifikasi kebutuhan pengguna dapat menyediakan pemahaman yang lebih dalam mengenai pernyataan-pernyataan masalah yang disampaikan oleh pemangku kepentingan. Leffingwell

dan Widrig menjelaskan, identifikasi kebutuhan pengguna, identifikasi fitur, dan persyaratan perangkat lunak dapat dilakukan dengan beberapa teknik yang terdiri dari:

1. Wawancara
2. *Requirement workshop*
3. *Brainstorming and Idea Reduction*
4. *Storyboarding*

2.6.4 Identifikasi Fitur

Fitur merupakan layanan yang disediakan oleh sistem untuk memenuhi satu atau lebih kebutuhan pemangku kepentingan (Leffingwell & Widrig, 2003). Leffingwell dan Widrig menambahkan, fitur merupakan cara berguna dan mudah untuk mendeskripsikan fungsional sistem tanpa harus terhambat dengan penjelasan fungsional sistem yang terlalu detail. Fitur juga menyediakan ringkasan manfaat yang ditawarkan oleh produk (sistem) yang akan dikembangkan (Bittner & Spence, 2002).

Tabel 2.8 Contoh Tabel Fitur

Kode Fitur	Deskripsi
Berisi kode identitas fitur	Berisi penjelasan tentang yang dapat dilakukan oleh sistem

Sumber: Diadaptasi dari Bittner & Spence (2002)

2.6.5 Identifikasi Persyaratan Fungsional dan Nonfungsional

Bittner dan Spence menyebutkan, persyaratan fungsional dan persyaratan non-fungsional merupakan kategori yang umum digunakan untuk mengelompokkan persyaratan perangkat lunak. Persyaratan fungsional dan nonfungsional dapat diekspresikan dalam banyak cara, namun cara yang paling umum digunakan adalah dengan mendeskripsikan persyaratan-persyaratan tersebut dalam kalimat deklaratif.

Persyaratan fungsional adalah layanan yang sebaiknya dapat disediakan oleh sistem, persyaratan fungsional juga merupakan pernyataan yang menyatakan bagaimana sistem bereaksi terhadap *input* tertentu dan bagaimana sistem berjalan ketika berada pada situasi tertentu (Sommerville, 2011). Deskripsi dari persyaratan fungsional sebaiknya berisi penjelasan mengenai apa tindakan yang dapat dilakukan oleh sebuah sistem. Selain itu didalamnya terdapat informasi perilaku sistem ketika menerima *input* dan *output*.

Persyaratan nonfungsional merupakan aspek kualitas atau batasan-batasan yang harus dipenuhi oleh sistem (Bittner & Spence, 2002). Leffingwell dan Widrig menjelaskan bahwa persyaratan non-fungsional muncul karena sangat penting

bagi analisis untuk menentukan aspek-aspek sistem yang berhubungan dengan *usability, reliability, performance, dan supportability* dari sebuah sistem.

2.6.6 Pemodelan Use Case

Model *use case* digunakan untuk mempresentasikan sistem dalam hal bagaimana penggunaan sistem tersebut (Bittner & Spence, 2002). Bittner dan Spence menambahkan, dalam menentukan *use case*, analisis menggunakan persyaratan perangkat lunak sebagai panduan dan seringkali *use case* berisi persyaratan fungsional dari sebuah sistem. Model *use case* berisi dua komponen utama yaitu aktor dan *use Case*.

Aktor adalah representasi individu (manusia) atau objek lain yang berinteraksi dengan sistem. Aktor merupakan objek diluar sistem. Dalam model *use case*, aktor memiliki nama dan deskripsi singkat, selain itu, aktor dihubungkan dengan *use case* sesuai dengan interaksi aktor dengan *use case* tertentu.

Use case adalah representasi sebuah nilai yang diberikan oleh sistem kepada aktor. *Use case* tidak sama dengan fitur atau fungsi dari sebuah sistem. *Use case* memiliki nama dan penjelasan singkat. *Use case* juga memiliki spesifikasi yang lebih detail, yang berisi alur ketika aktor dan sistem berinteraksi.

Model *use case* tidak menyediakan deskripsi yang lengkap mengenai persyaratan sebuah sistem, oleh karena itu, model *use case* harus didukung dengan pemodelan-pemodelan persyaratan dan dokumen-dokumen persyaratan lainnya. Setiap *use case* di dalam model *use case* memiliki spesifikasi *use case* yang berisi penjelasan bagaimana aktor dan sistem berkolaborasi untuk memenuhi tujuan yang direpresentasikan oleh *use case* tersebut. Format dokumentasi spesifikasi *use case* terdapat pada Tabel 2.9.

Tabel 2.9 Format Dokumentasi Spesifikasi Use Case

Brief Description	Berisi penjelasan singkat sebuah <i>use case</i> .
Actor	Berisi aktor yang berinteraksi dengan <i>use case</i> .
Pre-condition	Berisi satu atau lebih kondisi atau keadaan yang harus dipenuhi oleh sistem sebelum <i>use case</i> dapat dijalankan.
Post-condition	Berisi satu atau lebih kondisi atau keadaan yang dipenuhi oleh sistem ketika <i>use case</i> berakhir.
Extension	Berisi informasi penggunaan <i>use case</i> lain yang bersifat opsional, yang disertakan pada sebuah <i>use case</i> .
Basic Flow	Berisi alur normal yang dilalui ketika <i>use case</i> berjalan.
Alternative Flow	Berisi alur opsional yang akan dilalui ketika <i>use case</i> tidak berjalan sesuai dengan alur normal.
Subflow	Berisi beberapa kelompok alur yang merupakan penyederhanaan alur <i>use case</i> yang terlalu detail.

Sumber: Diadaptasi dari Bittner & Spence (2002)

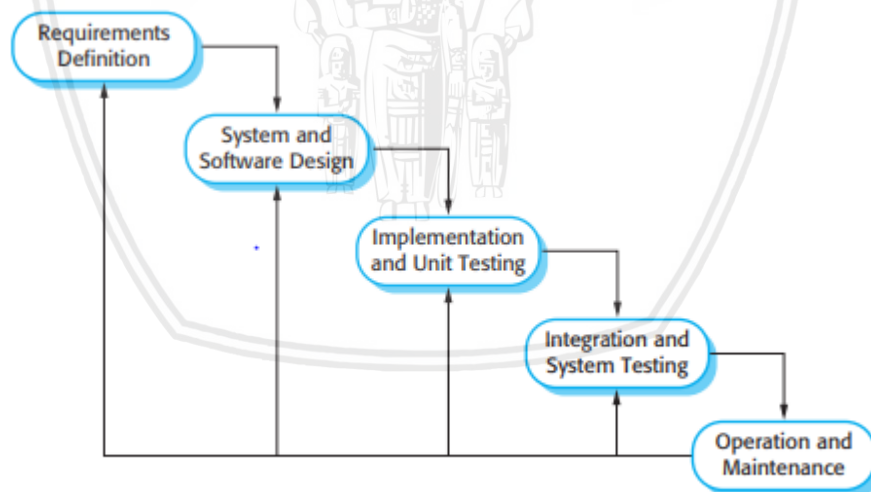
2.7 Software Development Life Cycle (SLDC)

System Development Life Cycle (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik). Sepertinya halnya metamorfosis pada kupu-kupu, untuk menjadi kupu-kupu indah maka dibutuhkan beberapa tahap untuk dilalui, sama halnya dengan membuat perangkat lunak, memiliki daur tahap yang dilalui agar menghasilkan perangkat lunak yang berkualitas (Rosa A. S dan M. Shalahuddin, 2016).

Siklus pada SDLC harus dijalankan secara berurutan dengan tetap mengkaji ulang pada masing-masing tahapan yang telah diselesaikan. Hal ini dilakukan untuk memastikan bahwa tahapan telah sesuai dengan yang diharapkan.

2.7.1 Waterfall Model

Model waterfall mendefinisikan beberapa fase berurutan yang harus diselesaikan satu persatu dan hanya berpindah ke tahapan selanjutnya apabila tahapan sebelumnya sudah diselesaikan.



Gambar 2.3 Waterfall Model

Berikut ini merupakan fase-fase dari model pengembangan *waterfall*:

1. Requirements analysis and definition
Layanan sistem, kendala, dan Tujuannya ditetapkan melalui konsultasi dengan pengguna sistem. Mereka kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem

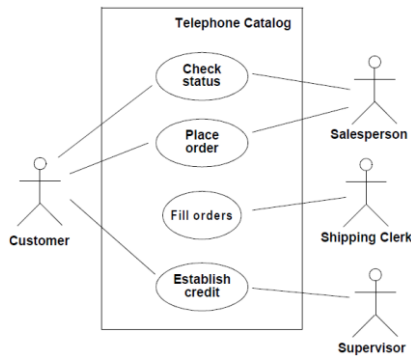
2. **System and software design**
Proses perancangan sistem mengalokasikan persyaratan ke sistem perangkat keras atau perangkat lunak dengan membangun sistem secara keseluruhan Arsitektur. Desain perangkat lunak melibatkan identifikasi dan penggambaran fundamental Abstraksi sistem perangkat lunak dan hubungan mereka.
3. **Implementation and unit testing**
Selama tahap ini, disain software direalisasikan sebagai seperangkat program atau unit program. Pengujian unit melibatkan verifikasi itu setiap unit memenuhi spesifikasi.
4. **Integration and system testing**
Unit atau program perorangan terintegrasi dan diuji sebagai sistem yang lengkap untuk memastikan bahwa perangkat lunak persyaratan telah dipenuhi Setelah pengujian, sistem perangkat lunak dikirim ke pelanggan.
5. **Operation and maintenance**
Biasanya (meski belum tentu), ini adalah fase siklus hidup terpanjang. Sistem dipasang dan dimasukkan ke dalam penggunaan praktis. Pemeliharaan melibatkan koreksi kesalahan yang tidak ditemukan sebelumnya tahapan siklus hidup, peningkatan pelaksanaan unit sistem dan meningkatkan layanan sistem sebagai persyaratan baru ditemukan.

2.8 Unified Model Language (UML)

Unified Modelling Language (UML) merupakan Bahasa yang digunakan untuk menulis rancangan perangkat lunak. Selain itu, *Unified Modelling Language* (UML) ini dapat memvisualisasikan, menspesifikasikan, membangun dan mendokumentasikan kerangka perangkat lunak. Tidak hanya itu, arsitek perangkat lunak juga membuat diagram *Unified Modelling Language* (UML) untuk membantu para pengembang perangkat lunak dalam membangun perangkat lunak. *Unified Modelling Language* (UML) jika dipahami akan memberikan beberapa manfaat seperti lebih mudah memahami sistem, menspesifikasikan sistem dan menjelaskan desain atau sistem.

2.8.1 Use Case Diagram

Use case diagram menampilkan hubungan di antara aktor dengan *use case* (Object Management Group, 2005). Hubungan yang tergambar di dalam *use case diagram* adalah hubungan asosiasi antara aktor dengan satu atau lebih *use case*, hubungan generalisasi antar aktor, dan hubungan generalisasi, *extends*, dan *includes* antar *use case*. Sekumpulan *use case* secara opsional dapat diletakkan di dalam sebuah notasi berbentuk persegi yang merepresentasikan *boundary* sebuah sistem. Contoh *use case diagram* diperlihatkan dalam Gambar 2.4.



Gambar 2.4 Contoh Use Case Diagram

Sumber: Object Management Group (2005)

Use case diagram memiliki beberapa komponen notasi, di antaranya notasi aktor, notasi use case, notasi hubungan use case, dan notasi hubungan aktor. Notasi-notasi use case diagram dijelaskan pada Tabel 2.10.

Tabel 2.10 Notasi pada Use Case Diagram

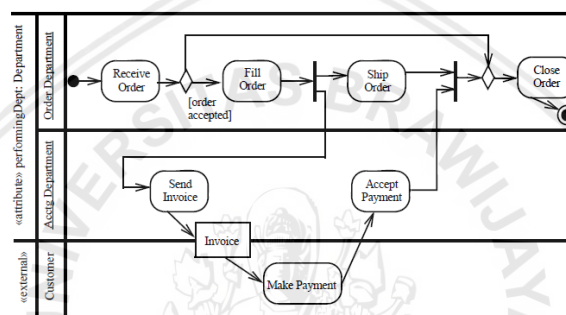
Notasi	Nama	Deskripsi
	Aktor	Aktor merupakan peran yang diperankan oleh pengguna atau sistem lain yang berinteraksi dengan sistem. Satu aktor bisa saja berinteraksi dengan satu atau lebih use case.
	Use case	Use case adalah (pernyataan) dari sekumpulan aksi yang dilakukan oleh sistem yang akan memberikan nilai untuk satu atau lebih aktor atau pemangku kepentingan sebuah sistem.
	Asosiasi	Notasi hubungan antara aktor dengan use case yang menggambarkan aktor berinteraksi dengan sistem dengan melakukan use case tertentu.
	Generalisasi	Notasi hubungan antar aktor yang menghubungkan aktor yang lebih spesifik dengan aktor yang umum.
	Extends	Notasi hubungan antar use case yang menunjukkan bahwa perilaku sebuah use case dapat diperluas (extended) oleh use case yang lain.
	Include	Notasi hubungan antar use case yang menunjukkan bahwa sebuah perilaku dari

Tabel 2.10 Notasi pada Use Case Diagram

Notasi	Nama	Deskripsi
		sebuah <i>use case</i> merupakan bagian dari <i>use case</i> lain.

2.8.2 Activity Diagram

Activity diagram digunakan untuk memodelkan aspek dinamis dari sebuah sistem (Booch, et al., 1998). *Activity diagram* pada umumnya digunakan untuk memodelkan urutan langkah dalam proses komputasi. *Activity diagram* menekankan pemodelan alur antar aktivitas, namun *activity diagram* juga dapat digunakan untuk memodelkan alur perubahan *state* dari sebuah objek. Contoh *activity diagram* terdapat dalam Gambar 2.5.



Gambar 2.5 Contoh Activity Diagram

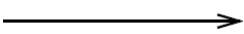


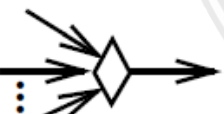
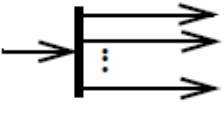
Sumber: Object Management Group (2005)

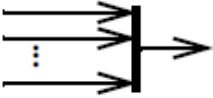


Activity diagram memiliki beberapa notasi, beberapa notasi yang digunakan dalam pemodelan di penelitian ini terdiri dari notasi *initial node*, notasi *actions*, notasi *activity edges*, notasi *decision node* atau *merge node*, notasi *fork node* atau *join node*, notasi *final node (activity final)*, dan notasi *activity partition*. Penjelasan setiap notasi pada activity diagram terdapat pada Tabel 2.11.

Tabel 2.11 Notasi pada Activity Diagram

Notasi	Nama	Deskripsi
	<i>Initial Node</i>	Notasi kontrol yang menandakan titik awal untuk mengeksekusi sebuah aktivitas yang tergambar di dalam <i>activity diagram</i> . Notasi <i>initial node</i> tidak boleh memiliki <i>incoming edges</i> .
	<i>Actions</i>	Notasi yang merepresentasikan tindakan yang dilakukan oleh sebuah unit di dalam sistem atau unit bisnis.



Notasi	Nama	Deskripsi
	<i>Activity Edges</i>	Notasi yang menghubungkan serangkaian notasi di dalam <i>activity diagram</i> .
	<i>Activity Edge Connector</i>	Notasi yang digunakan untuk menghubungkan <i>activity edges</i> . Setiap konektor memiliki pasangan dan ditandai dengan nama yang sama. Sebuah konektor dalam satu pasang hanya boleh memiliki 1 buah <i>incoming edge</i> dan pasangan harus memiliki 1 buah <i>outcoming edge</i> . Satu pasang konektor diletakkan di dalam 1 <i>activity diagram</i> yang sama.
	<i>Decision Node</i>	Notasi yang digunakan untuk menangani alur kondisional. Notasi ini memiliki 1 <i>incoming edges</i> dan memiliki 2 atau lebih <i>outgoing edges</i> . Masing-masing <i>outgoing edges</i> harus memiliki nilai boolean yang digambarkan melalui <i>guards</i> . Notasi ini tidak merepresentasikan proses sinkronisasi, sehingga alur akan berjalan melalui salah satu <i>outgoing edges</i> sesuai dengan kondisi yang dipenuhi.
	<i>Merge Node</i>	Notasi yang digunakan untuk menerima beberapa alur alternatif. Notasi ini tidak merepresentasikan proses sinkronisasi, melainkan menerima salah satu dari beberapa alur alternatif. Notasi ini memiliki 2 atau lebih <i>incoming edges</i> dan memiliki 1 <i>outgoing edges</i> .
	<i>Fork Node</i>	Notasi yang merepresentasikan pembagian sebuah alur menjadi beberapa alur yang berjalan secara <i>concurrent</i> . <i>Fork nodes</i> memiliki 1 <i>incoming edges</i> dan 2 atau lebih <i>outgoing edges</i> . <i>Outgoing edges</i> bisa saja memiliki <i>guards</i> untuk menandakan kondisi tertentu ketika

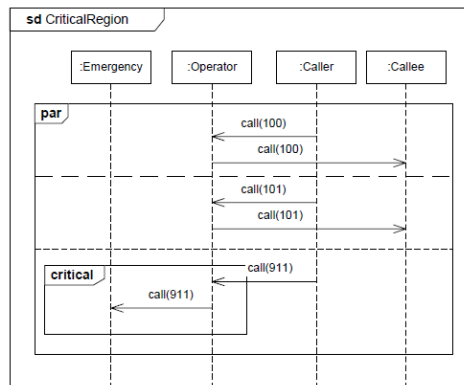
Notasi	Nama	Deskripsi
		alur melewati <i>edges</i> tersebut, hal ini disebut sebagai <i>conditional fork</i> . Jika <i>conditional fork</i> berlaku, maka alur yang tidak memenuhi kondisi tidak dijalankan.
	<i>Join Node</i>	Notasi yang merepresentasikan proses sinkronisasi beberapa flow. Notasi ini memiliki 2 atau lebih <i>incoming edges</i> dan 1 <i>outgoing edges</i> .
	<i>Final Node (Activity Final)</i>	Notasi yang merepresentasikan titik berakhirnya alur aktivitas. <i>Final node</i> tidak boleh memiliki <i>outgoing edges</i> .
	<i>Activity Partition</i>	Notasi yang digunakan untuk mengelompokkan <i>actions</i> berdasarkan unit sistem, unit bisnis, atau entitas diluar sistem yang menjalankan <i>actions</i> tersebut. Penamaan notasi disesuaikan dengan nama unit sistem, nama unit bisnis, atau nama entitas yang melakukan <i>actions</i> tertentu.

Sumber: Diadaptasi dari Object Management Group (2005)

2.8.3 Sequence Diagram

Sequence diagram merupakan varian dari *interaction diagram* yang memvisualisasikan interaksi, yang terdiri dari sekumpulan objek dan hubungan di antaranya, termasuk pesan yang mungkin dikirimkan antar objek (Booch, et al., 1998). *Sequence diagram* menekankan visualisasi pertukaran pesan antar objek didalam sistem dalam urutan waktu. Contoh *sequence diagram* terdapat dalam Gambar 2.6.

Sequence diagram berisi notasi-notasi yang merepresentasikan *instances* dari sebuah kelas, antarmuka, komponen, dan *nodes*, beserta pertukaran pesan di antara *instances* tersebut. Beberapa notasi *sequence diagram* yang digunakan pada penelitian ini dijelaskan pada Tabel 2.12.

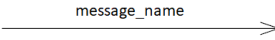
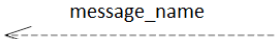
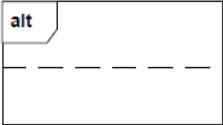


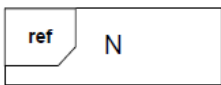
Gambar 2.6 Contoh Sequence Diagram

Sumber: Object Management Group (2005)

Tabel 2.12 Notasi pada Sequence Diagram

Notasi	Nama	Deskripsi
	Frame	Notasi yang berisi sekumpulan interaksi. Didalam notasi ini terdapat serangkaian <i>instances</i> yang saling bertukar pesan sebagai visualisasi aspek dinamis bagian dari sistem. Nama interaksi dituliskan di dalam bangun segi lima di pojok kiri atas notasi <i>frame</i> .
	Lifeline	Notasi ini merupakan representasi sebuah <i>instances</i> yang terlibat dalam sebuah interaksi pertukaran pesan (<i>message</i>).
	Message (call)	Notasi yang merepresentasikan komunikasi yang bersifat <i>synchronous</i> di antara <i>lifeline</i> . Notasi <i>call</i> secara khusus merepresentasikan pemanggilan operasi atau metode. Jika notasi ini digunakan untuk memodelkan pemanggilan operasi dengan argumen, maka argumen tersebut harus disertakan dalam penamaan notasi.

Notasi	Nama	Deskripsi
	<p><i>Message (asynchronous)</i></p>	<p>Notasi yang merepresentasikan komunikasi yang bersifat <i>asynchronous</i> di antara <i>lifeline</i>.</p>
	<p><i>Message (reply)</i></p>	<p>Notasi yang merepresentasikan pesan balasan dari penggunaan notasi <i>message call</i> atau <i>asynchronous</i>.</p>
	<p><i>Combined Fragment (Fragment)</i></p>	<p>Notasi yang digunakan untuk mengelompokkan interaksi kedalam fragmen. Setiap fragmen dapat memiliki beberapa <i>operand</i> yang masing-masing ditempatkan pada jalur yang dipisahkan oleh garis putus-putus di dalam fragmen. Masing-masing <i>operand</i> dapat memiliki sebuah <i>guard</i>, yang menandakan sebuah kondisi yang dipenuhi ketika <i>operand</i> tersebut dijalankan. Notasi ini memiliki beberapa varian yang ditandai dengan penamaan <i>interaction operator</i> di dalam bangun segi lima di pojok kiri atas notasi ini.</p> <p>Varian <i>alternative</i> (alt) merupakan varian yang merepresentasikan interaksi kondisional. Pada varian ini setidaknya 1 <i>operand</i> akan dijalankan.</p> <p>Varian <i>option</i> (opt) merupakan varian yang merepresentasikan interaksi kondisional. Varian ini hanya boleh memiliki 1 <i>operand</i>, yang bisa saja tidak dijalankan apabila kondisi pada <i>guard</i> tidak terpenuhi.</p> <p>Varian <i>loop</i> merupakan varian yang merepresentasikan interaksi yang dilakukan berulang-ulang selama kondisi <i>operand</i> terpenuhi. Varian ini hanya boleh memiliki 1 <i>operand</i>.</p>

Notasi	Nama	Deskripsi
	<i>Interaction Use (Fragment)</i>	Notasi yang digunakan untuk merujuk sebuah interaksi. Pemodelan notasi ini menggunakan notasi <i>combined fragment</i> dengan nama <i>interaction operator</i> ref. Nama interaksi yang dirujuk diletakkan di tengah notasi.

Sumber: Diadaptasi dari Object Management Group (2005)

2.8.4 Class Diagram

Class diagram menunjukkan sekumpulan kelas (*class*), *interfaces*, kolaborasi (*collaboration*) dan hubungan (*relationship*) di antaranya (Booch, et al., 1998). *Class diagram* umumnya berisi beberapa elemen yang terdiri dari kelas, *interfaces*, kolaborasi, *relationship (dependency, generalization, dan association)*.

Class diagram memiliki beberapa tingkat detail yang berbeda-beda, Scott dalam bukunya menjelaskan bahwa terdapat beberapa tingkat detail yang digunakan dalam beberapa tahap pengembangan perangkat lunak di antaranya pada tahap persyaratan, analisis dan perancangan. *Class diagram* digunakan untuk memodelkan aspek statis dari sebuah sistem (Booch, et al., 1998). Beberapa cara pemodelan aspek statis yang dapat dimodelkan ke dalam *class diagram* di antaranya yaitu:

1. *Modeling the Vocabulary of a System*

Pemodelan *class diagram* pada umumnya digunakan untuk memodelkan abstraksi dari sebuah sistem yang didapatkan dari masalah yang ingin diselesaikan dengan sistem yang akan dikembangkan. Masing-masing abstraksi merupakan bagian dari *vocabulary* dari sebuah sistem. Setiap abstraksi dimodelkan ke dalam kelas-kelas beserta atribut dan kemampuan yang dapat dilakukan oleh kelas-kelas tersebut.

2. *Modeling Simple Collaboration*

Kolaborasi dalam hal pemodelan *class diagram* merepresentasikan hubungan antar kelas, *intrefaces*, dan objek lain yang bekerja bersama-sama untuk menunjukkan perilaku dari sebuah sistem.

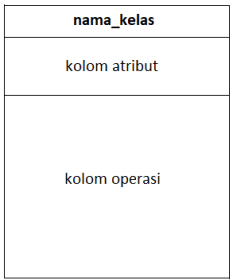




3. *Modeling a Logical Database Schema*

Class diagram dapat digunakan untuk memodelkan skema basis data sebagai cetak biru konsep rancangan basis data. Skema yang dimodelkan merepresentasikan objek-objek yang akan menyimpan data yang akan dibutuhkan oleh sistem.

Notasi kelas memiliki struktur yang terdiri dari 3 kolom di antaranya yaitu kolom nama kelas, kolom atribut, dan kolom operasi. Masing-masing atribut dan

operasi yang dimiliki oleh kelas dapat ditentukan visibilitasnya dan digambarkan dengan notasi *visibility kind* yang diletakkan sebelum nama atribut atau nama operasi. Penjelasan notasi *class diagram* yang digunakan pada penelitian ini terdapat pada Tabel 2.13.

Tabel 2.13 Notasi pada Class Diagram

Notasi	Nama	Deskripsi
	<i>Class</i>	<p>Notasi yang merepresentasikan sebuah objek beserta sekumpulan atribut dan operasi yang dimiliki. Struktur notasi ini memiliki 3 kolom yang terdiri dari kolom nama kelas, kolom atribut, dan kolom operasi.</p> <p>Masing-masing atribut dan operasi dapat ditentukan visibility-nya dengan</p>
	<i>Association</i>	Notasi yang merepresentasikan relasi sebuah objek dengan objek yang lain.
	<i>Generalization</i>	Notasi yang merepresentasikan relasi sebuah objek yang spesifik (<i>subclass</i>) dengan objek yang lebih umum (<i>superclass</i>). <i>Subclass</i> mewarisi atribut dan operasi yang dimiliki oleh <i>superclass</i> .
	<i>Aggregation</i>	Notasi yang merepresentasikan relasi sebuah objek yang lebih luas (<i>whole</i>) dengan objek yang menjadi bagian dari objek tersebut (<i>parts</i>). Objek <i>whole</i> diletakkan pada sisi ujung notasi yang memiliki bentuk belah ketupak, sedangkan objek <i>parts</i> diletakkan pada ujung yang lain.
	<i>Composition</i>	Notasi ini merupakan bentuk lain dari notasi <i>aggregation</i> , namun dalam notasi ini sebuah objek <i>parts</i> merupakan bagian dari hanya 1 objek yang lebih besar.
+	<i>Public (Visibility Kind)</i>	Notasi yang digunakan untuk menjelaskan aksesibilitas atribut atau operasi yang dimiliki objek. <i>Public</i> memungkinkan atribut atau operasi dapat diakses secara langsung oleh objek lain yang memiliki relasi dengan objek yang memilikinya.



Notasi	Nama	Deskripsi
#	<i>Protected</i> (Visibility Kind)	<i>Protected</i> memungkinkan atribut atau operasi dapat diakses secara langsung oleh <i>instance</i> dari objek <i>subclass</i> .
-	<i>Private</i> (Visibility Kind)	<i>Private</i> memungkinkan atribut atau operasi hanya dapat diakses oleh objek yang memilikinya.

Sumber: Diadaptasi dari Object Management Group (2005)

2.9 Physical Data Model

Physical Data Model (PDM) memiliki konsep yang sama dengan pemodelan kelas, namun tujuan PDM adalah untuk memodelkan skema internal basis data, menggambarkan tabel beserta kolom-kolomnya, dan hubungan antar tabel (Ambysoft Inc., 2013).

2.10 Model View Controller (MVC)

Model-View-Controller (MVC) adalah pola arsitektur yang memisahkan aplikasi dalam tiga komponen utama Logis: Model, View dan *Controller*. Masing-masing komponen ini dibangun untuk menangani aspek-aspek tertentu pembangunan aplikasi. MVC adalah salah satu kerangka pembangunan web standar industri paling sering digunakan untuk menciptakan proyek yang terukur an besar dan extensible. Berikut ini merupakan komponen dari MVC:

Model

Komponen Model yang sesuai dengan semua data yang terkait dengan penggunaan logika dalam berkerja. Ini dapat mewakili baik data yang ditransfer antara *View* dan *Controller* komponen atau logika bisnis lain data yang terkait. Sebagai contoh, sebuah objek pelanggan akan mengambil informasi pelanggan dari *database*, memanipulasi itu dan memperbarui data kembali ke *database* atau menggunakannya untuk membuat data.

View

Komponen View digunakan untuk semua UI (*User Interface*) pada logika aplikasi. Misalnya, tampilan pelanggan akan mencakup semua komponen UI seperti *kotak teks*, *dropdown*, dll yang digunakan pengguna untuk berinteraksi.

Controller

Controller bertindak sebagai antar muka antara Model dan View komponen proses semua logika bisnis dan permintaan masuk, memanipulasi data menggunakan komponen *Model* dan berinteraksi dengan *View* untuk membuat hasil akhir. Sebagai contoh, *controller* pelanggan akan menangani semua interaksi

dan masukan dari View pelanggan dan update *database* menggunakan Model pelanggan. *Controller* sama akan digunakan untuk melihat data pelanggan.

2.11 Framework

Framework dapat diartikan sebagai koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi untuk tanpa harus membuat semua kodenya dari awal. Saat ini ada banyak framework PHP, diantaranya: Zend, Cake PHP, Trax, Symfony, Codeigniter dan sebagainya. Tentu saja, setiap framework memiliki kelebihan dan kekurangannya masing-masing. Keuntungan yang dapat diperoleh dari penggunaan framemork adalah (Octafian, 2015) :

1. Waktu pembuatan aplikasi situs web jauh lebih singkat.
2. Kode aplikasi situs web menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok, detailnya adalah kode dari framework.
3. Situs web menjadi lebih mudah diperbaiki, karena tidak perlu fokus ke semua komponen kode situs web, terutama kode sistem framework.
4. Tidak perlu lagi membuat kode penunjang aplikasi situs web seperti koneksi database, validasi form, GUI, dan keamanan.
5. Pikiran pengembang menjadi lebih terfokus ke kode alur permasalahan situs web, apa yang ditampilkan dan layanan apa saja yang diberikan dari aplikasi situs web tersebut.
6. Jika dikerjakan team work, maka akan lebih terarah karena sistem framework, mengharuskan adanya keteraturan peletakan kode. Seperti bagian pengambilan database terpisah dengan bagian pengaturan tampilan untuk penunjang.

2.11.1 Hypertext Markup Language (HTML)

HTML (Hypertext Markup Language) adalah serangkaian kode program yang merupakan dasar dari representasi visual sebuah halaman Web. Didalamnya berisi kumpulan informasi yang disimpan dalam tag-tag tertentu, dimana tag-tag tersebut digunakan untuk melakukan format terhadap informasi yang dimaksud.

Elemen *HTML* mempunyai tiga hal penting yaitu: tag pembuka, isi, dan tag penutup. Sebagai contoh: Elemen html mempunyai tag pembuka "<html>" dan tag penutup "</html>" dan yang berada diantaranya merupakan isi atau konten dari elemen html tersebut. Untuk menuliskan Elemen HTML bisa menggunakan huruf besar maupun huruf kecil. Contoh: <HTML>, <HtMl>, <HTml>, <html>, semuanya adalah sama.

Secara garis besar terdapat 4 jenis elemen dari HTML yaitu:

1. *Structural*

Adalah kode program yang menentukan level atau tingkatan dari sebuah tulisan. Contoh: `<h1>Mozilla</h1>` akan memerintahkan browser untuk menampilkan "Mozilla" sebagai tulisan tebal besar yang menunjukkan sebagai Heading 1.

2. *Presentational*

Adalah kode yang menentukan tampilan dari sebuah tulisan, tidak peduli dengan level dari tulisan tersebut. Contoh: `Cetak Tebal` maka pada browser akan menampilkan "Cetak Tebal". Namun kode-kode presentational saat ini sudah mulai digantikan dengan penggunaan CSS (Cascading Style Sheets) dan tidak direkomendasikan lagi untuk mengatur tampilan tulisan.

3. *HyperText*

Kode program HTML yang menunjukkan hubungan (link) ke bagian lain dari dokumen tersebut atau link ke dokumen lain. Contoh:

`dress` maka pada browser akan menampilkan JURNAL TEKNIK INFORMATIKA, JULI 2015 "dress" sebagai sebuah hyperlink yang menuju url `http://berrybenka.com/`

4. *Elemen*

Widget yang membuat objek-objek lain seperti tombol `<button>`, list ``, dan garis horizontal `<hr>`, Konsep hypertext pada HTML memungkinkan kita untuk membuat link pada suatu kelompok kata atau frase untuk menuju ke bagian manapun dalam World Wide Web (WWW).

2.11.2 *Casanding Style Sheet (CSS)*

Menurut Arief (2011) PHP adalah bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side-scripting* maka sintak dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya akan dikirimkan ke browser dengan format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak terlihat oleh user sehingga halaman web lebih terjamin. PHP dirancang untuk membuat halaman web dinamis, yaitu halaman web yang dapat membentuk sutau tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data di halaman web.

2.11.3 *Hypertext Processor (PHP)*

PHP adalah sebuah bahasa pemrograman yang berjalan dalam sebuah web *server* (*server side*). PHP diciptakan oleh programmer unix dan Perl yang bernama Rasmus Lerdoft pada bulan Agustus-September 1994. Pada awalnya, Rasmus mencoba menciptakan sebuah script dalam wesite pribadinya dengan tujuan untuk memonitor siapa saja yang pernah mengunjungi website-nya. Sistem kerja PHP diawali dengan permintaan yang berasal dari halaman website oleh browser.

Berdasarkan URL atau alamat website dalam jaringan internet, browser akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*. Selanjutnya *webserver* akan mencarikan berkas yang diminta dan menampilkan isi dari halaman website tersebut di browser. Browser yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya. Lalu bagaimana apabila yang dipanggil oleh user adalah halaman yang mengandung script *PHP*? Pada prinsipnya sama dengan memanggil kode *HTML*, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa tipe file yang diminta user. Jika tipe file yang diminta adalah *PHP*, maka akan memeriksa isi script dari halaman *PHP* tersebut. Apabila dalam file tersebut tidak mengandung script *PHP*, permintaan user akan langsung ditampilkan ke browser, namun jika dalam file tersebut mengandung *script PHP*, maka proses akan dilanjutkan ke modul *PHP* sebagai mesin yang menerjemahkan script-script *PHP* dan mengolah script tersebut, sehingga dapat dikonversikan ke kode-kode *HTML* lalu ditampilkan ke browser user.

2.12 Codeigniter (CI)

Codeigniter merupakan sebuah framework yang bersifat open source untuk membangun aplikasi web yang dinamis. Selain itu, penggunaan Codeigniter juga akan menghasilkan suatu struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya. Struktur aplikasi yang rapi tentu sangat diperlukan dari sebuah aplikasi. Misalnya, jika terjadi suatu error dalam aplikasi, dengan code yang rapi kita dapat dengan mudah menemukan kesalahan tersebut. Bukan hanya itu, bayangkan suatu saat aplikasi yang kita bangun membutuhkan fungsi-fungsi lain yang sangat penting, tentu akan diperlukan pengembangan lebih lanjut. Kemudian, hal itu juga sangat memerlukan struktur coding yang rapi. Dengan menggunakan Codeigniter, untuk mewujudkan struktur kode yang rapi sangat-sangat mungkin terjadi.

2.13 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan dengan maksud menunjukkan bahwa sebuah perangkat lunak yang telah dikembangkan dapat berjalan sesuai dengan tujuan dan untuk mengetahui kekurangan (*defects*) sebelum perangkat lunak digunakan oleh pengguna (Sommerville, 2011). Sommerville menambahkan, pengujian perangkat lunak memiliki 2 tujuan, yaitu:

1. Untuk mendemostrasikan kepada pengembang dan pelanggan bahwa perangkat lunak telah memenuhi persyaratan.
2. Untuk mengetahui apabila perangkat lunak tidak berjalan sesuai dengan yang diharapkan atau tidak memenuhi spesifikasi.

2.13.1 Pengujian Validasi

Pengujian validasi merupakan pengujian yang fokus pada tindakan yang dan hasil (*output*) dari sistem yang terlihat oleh pengguna (Pressman, 2010). Pengujian validasi perangkat lunak dilakukan dengan serangkaian kegiatan pengujian untuk mendemonstrasikan perangkat lunak telah memenuhi persyaratan. Setelah pengujian dilakukan, Pressman menjelaskan bahwa akan didapatkan dua hasil, yaitu:

1. Karakteristik fungsi atau performa sistem telah memenuhi spesifikasi dan diterima oleh pelanggan atau pemangku kepentingan.
2. Deviasi dari spesifikasi perangkat lunak akan terlihat dan kekurangan perangkat lunak akan dicatat.

Deviasi atau kesalahan yang telah ditemukan akan sangat jarang dapat diperbaiki karena penyesuaian dengan jadwal perilisan perangkat lunak. Dalam banyak kesempatan, sangat penting untuk melakukan negosiasi dengan pemangku kepentingan untuk dapat menemukan cara untuk menyelesaikan masalah tersebut.

Pengujian validasi perangkat lunak dapat dilakukan dengan menggunakan *use case* sebagai panduan pengujian. *Use case* menyediakan skenario yang memungkinkan untuk mendeteksi deviasi pada perangkat lunak. Selain itu, pengujian *black-box* juga dapat digunakan untuk melakukan pengujian (Pressman, 2010).

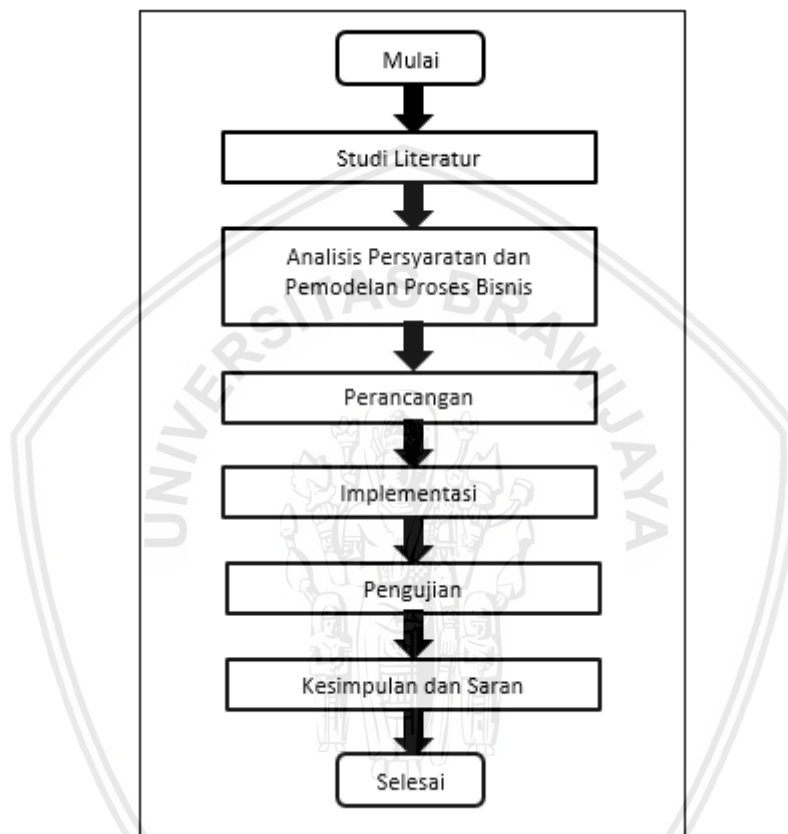
2.13.2 Pengujian Kompatibilitas

Pengujian kompatibilitas merupakan satu dari beberapa pengujian yang dilakukan terhadap antarmuka pengguna pada aplikasi berbasis web untuk mengetahui apakah aplikasi memiliki masalah ketika berjalan pada komputer, perangkat tampilan, sistem operasi, aplikasi peramban, atau kecepatan jaringan yang berbeda (Pressman, 2010). Pengujian kompatibilitas pada aplikasi berbasis web dilakukan dengan mendefinisikan sekumpulan konfigurasi komputasi pada perangkat klien dan menguji aplikasi pada konfigurasi tersebut.

Konfigurasi dapat meliputi penggunaan sistem operasi yang berbeda, perangkat tampilan yang berbeda, aplikasi peramban yang berbeda, atau kecepatan jaringan yang berbeda. Pressman menambahkan, pengujian validasi dilakukan dengan maksud untuk mengetahui kesalahan atau masalah pada eksekusi aplikasi pada beberapa konfigurasi yang berbeda.

BAB 3 METODOLOGI PENELITIAN

Proses tahap penyusunan penelitian mempunyai tujuan untuk membantu kegiatan penelitian yang dilaksanakan dapat dilakukan dengan optimal sehingga menjadi teratur, terencana dan sistematis. Untuk itu, kegiatan penelitian dilaksanakan dalam melalui beberapa proses tahapan. Secara garis besar, kegiatan penelitian dilaksanakan dengan tahapan seperti gambar berikut



Gambar 3.1 Metodologi Penelitian

3.1 Studi Literatur

Studi literatur merupakan langkah awal untuk menjelaskan tinjauan teori yang akan dilaksanakan sebagai referensi dalam pembangunan sistem informasi data rak server pada PT. Indosat Ooredoo Surabaya. Refrensi ini didapatkan dari berbagai sumber seperti buku, jurnal ilmiah, e-book dan informasi dari intrenet.

3.2 Analisis Persyaratan dan Pemodelan Proses Bisnis

Pada tahap analisis masalah yang terdapat pada PT. Indosat Ooredoo Surabaya khususnya pada Divisi Teknik. Pada tahap ini peneliti melakukan wawancara dan obsevasi kepada pihak yang terkait yaitu Divisi Teknik Core and Backbound untuk mendapatkan informasi mengenai Sistem Informasi Pendataan Rak dan Perangkat Server dan kebutuhan pemangku kepentingan dari beberapa

pemangku kepentingan PT. Indosat Ooredoo Surabaya. Kemudian pada tahap selanjutnya adalah melakukan pengumpulan dokumen yang dibutuhkan dan berkaitan dengan penelitian. Beberapa dokumen yang dikumpulkan antara lain informasi rak dan perangkat *server* dalam bentuk MS. Word dan gambar berupa Sketchup.

Kegiatan selanjutnya adalah menganalisis dan memodelkan proses bisnis pada PT. Indosat Ooredoo Surabaya saat ini beserta proses bisnis usulan. Kegiatan berikutnya adalah mengidentifikasi kebutuhan pemangku kepentingan untuk mendapatkan informasi mengenai pemangku kepentingan pada PT. Indosat Ooredoo Surabaya. Kegiatan ini dilakukan untuk mengetahui kebutuhan apa saja bagi pengguna, situasi yang terjadi saat ini untuk kebutuhan pengguna dan solusi yang akan digunakan untuk memenuhi kebutuhan pengguna.

Pada kegiatan selanjutnya adalah mengidentifikasi pengguna dilakukan untuk mengetahui siapa saja calon pengguna sistem. Analisis persyaratan juga dilakukan untuk mengetahui fitur sistem informasi yang akan dikembangkan sebagai bentuk solusi yang akan disediakan oleh sistem bagi pengguna atau pemangku kepentingan untuk memenuhi kebutuhannya.

Kemudian, melakukan pemodelan *use case* untuk mengetahui tujuan aktor dalam menggunakan sistem informasi yang akan dikembangkan. *Use case* juga menunjukkan bagaimana alur penggunaan sistem yang dilakukan oleh aktor. Selanjutnya, alur penggunaan sistem pada spesifikasi *use case* digunakan sebagai panduan pemodelan aktivitas ke dalam diagram aktivitas.

3.3 Perancangan

Perancangan sistem informasi dilakukan untuk memodelkan sistem ke dalam notasi-notasi pada *Unified Modeling Language* (UML) berdasarkan hasil tahap analisis persyaratan. Perancangan sistem berisi hasil pemodelan interaksi objek ke dalam *sequence diagram*, pemodelan objek ke dalam *class diagram*, perancangan basis data yang didokumentasikan ke dalam PDM, perancangan algoritme ke dalam *pseudocode*, dan perancangan antarmuka pengguna.

3.4 Implementasi

Tahap implementasi merupakan tahapan pembuatan Sistem Informasi Pendataan Rak dan Perangkat *Server* dengan menggunakan perancangan sistem. Sistem Informasi Pendataan Rak dan Perangkat menggunakan bahasa pemrograman PHP dengan kerangka kerja CodeIgniter. Hasil tahapan pengembangan Sistem Informasi Pendataan Rak dan Perangkat *Server* berbasis web telah diidentifikasi pada tahapan analisis kebutuhan.

Pendokumentasian juga dilakukan pada tahapan implementasi. Pendokumentasian ini berisi informasi perangkat keras dan perangkat lunak yang digunakan untuk pengembangan sistem informasi. Selain itu, isi dokumentasi berisi dokumentasi algoritma dan dalam kode program dan antarmuka pengguna.

3.5 Pengujian

Tahap pengujian merupakan hal yang sangat penting untuk pembuatan sebuah sistem informasi. Pengujian akan menentukan sistem yang berjalan dengan baik atau buruk pada sebuah sistem. Hal tersebut menjadi suatu pembuktian dan pemenuhan *requirement* yang telah didefinisikan sebelumnya. Dalam penelitian ini, peneliti menggunakan pengujian fungsional dan nonfungsional.

Pengujian fungsional dalam penelitian ini menggunakan *validation testing* yang mengedepankan terhadap fungsionalitas suatu sistem dimana peneliti menyiapkan suatu paket masukan valid atau tidak valid.

Pengujian nonfungsional dilakukan dengan menggunakan aplikasi SortSite untuk menguji kompatibilitas sistem informasi terhadap beberapa browser yang dipilih. Jika hasil pengujian memiliki masalah dalam hal perbaikan maka dibutuhkan kegiatan perbaikan pada implementasi.

3.6 Kesimpulan dan saran

Kesimpulan yang dapat diambil dari penelitian ini adalah jika semua tahapan sudah diselesaikan semua oleh peneliti. Kemudian saran dapat digunakan apabila terdapat kesalahan pada perangkat lunak serta saran tersebut dapat menjadikan perangkat lunak tersebut menjadi lebih baik dan berkembang nantinya.

BAB 4 ANALISIS PERSYARATAN DAN PEMODELAN PROSES BISNIS

4.1 Pemodelan Proses Bisnis

Pada sub bab ini dilakukan pemodelan proses bisnis untuk memodelkan serangkaian aktivitas yang dilakukan oleh fungsi-fungsi pada PT Indosat Ooredoo Surabaya untuk memenuhi pengguna bisnis. Pengambilan informasi mengenai proses bisnis dilakukan dengan cara wawancara dengan kepala teknik backbound PT Indosat Ooredoo Surabaya mengenai alur proses bisnis PT Indosat Ooredoo Surabaya dan dilakukan observasi pada lokasi studi kasus penelitian. Dalam penelitian ini proses bisnis yang akan dilakukan perbaikan adalah proses bisnis pendataan rak dan perangkat *server*.

4.1.1 Proses Bisnis *As-is*

Dalam penelitian ini proses bisnis *as-is* dapat diidentifikasi berdasarkan keadaan proses bisnis yang berjalan pada lokasi studi kasus sebelum solusi yang diusulkan dalam penelitian ini diberikan. Analisis dan pemodelan terhadap proses bisnis *as-is* dilakukan agar peneliti dapat mengidentifikasi kemungkinan perbaikan atau bahkan peningkatan terhadap proses bisnis melalui solusi yang ditawarkan.

4.1.1.1 Proses Bisnis Pendataan Rak *Server* dan Pemasangan Perangkat *Server*

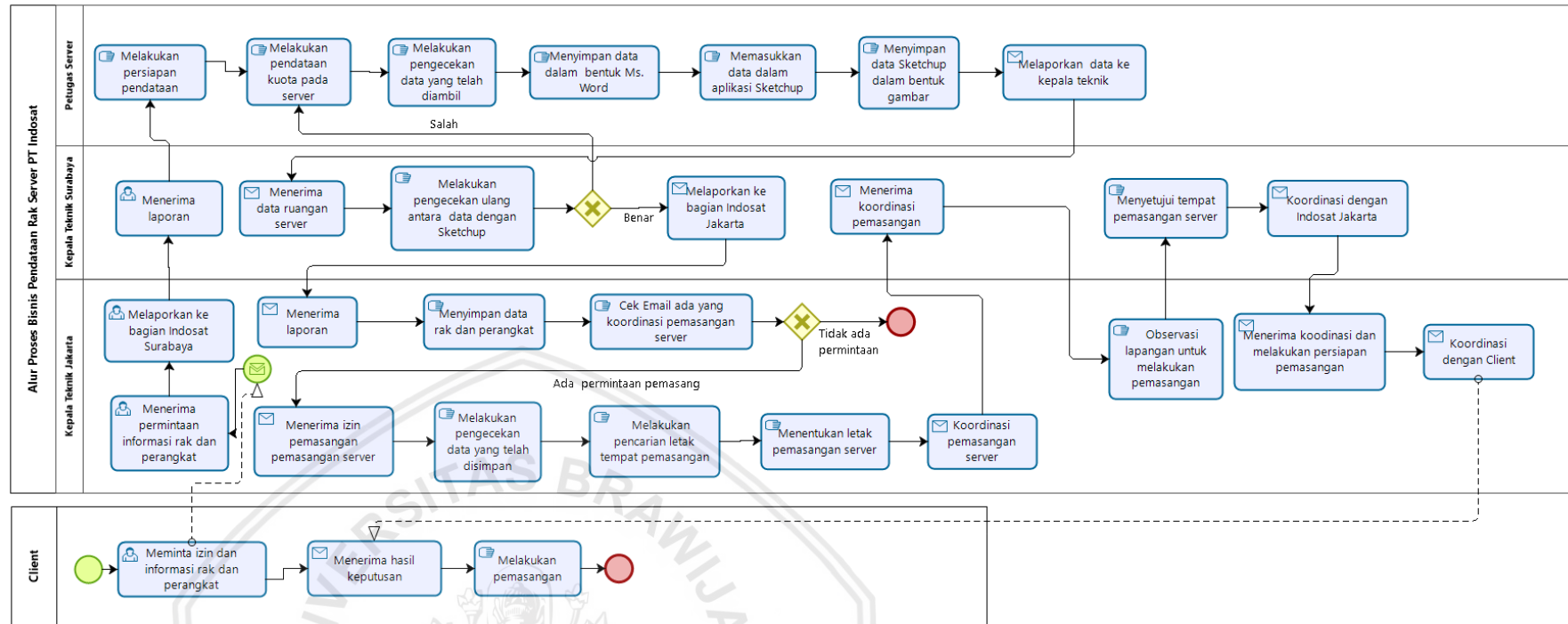
Berdasarkan hasil wawancara yang dilakukan oleh peneliti hasil yang didapatkan adalah aktivitas untuk pendataan rak *server* dan perangkat *server* dimulai dari awal hingga akhir yang akan dipasang pada PT Indosat Ooredoo Surabaya. Proses *as-is* pendataan rak dan perangkat *server* secara alurnya dapat dijelaskan sebagai berikut ini:

1. *Client* meminta izin dan informasi rak dan perangkat server kepada Kepala Teknik Jakarta.
2. Kepala Teknik Jakarta menerima permintaan informasi rak dan perangkat server dan melaporkan ke bagian Kepala Teknik Surabaya.
3. Kepala Teknik Surabaya menerima laporan.
4. Petugas server melakukan persiapan pendataan rak dan perangkat server.
5. Petugas *server* melakukan pendataan nama, kuota dan isi pada rak *server*.
6. Petugas *server* melakukan pengecekan terhadap data yang telah diambil pada proses sebelumnya.
7. Petugas *server* menyimpan data tersebut dalam Microsot Word.
8. Petugas sever memasukkan data dalam aplikasi Sketchup.
9. Petugas *server* menyimpan data rak dan perangkat *server* dalam bentuk gambar pada aplikasi Sketchup.

10. Petugas *server* melaporkan data rak dan perangkat *server* kepada Kepala Teknik.
11. Kepala Teknik menerima laporan dari petugas *server*.
12. Kepala Teknik melakukan pengecekan ulang antara data pada Microsot Word dengan data yang ada pada Sketchup.
13. Jika data tersebut benar, maka akan dilaporkan kepada Kepala Teknik Jakarta dan apabila data itu salah maka melakukan pengulangan dimulai dari proses bisnis melakukan pendataan kuota pada *server*.
14. Kepala Teknik Jakarta menerima laporan dari Kepala Teknik Surabaya.
15. Kepala Teknik Jakarta menyimpan data rak dan perangkat *server* yang telah dilaporkan oleh Kepala Teknik Surabaaya.
16. Kepala Teknik Jakarta melakukan cek email untuk memastikan ada yang melakukan pemasangan *server* kepada PT. Indosat Ooredoo Surabaya. Jika tidak terjadi pemasangan maka proses bisnis tersebut terhenti.
17. Jika ada pemasangan maka Kepala Teknik Jakarta menerima izin pemasangan *server* terhadap *Client*.
18. Kepala Teknik Jakarta melakukan pengecekan data rak dan perangkat *server* yang telah disimpan.
19. Kepala Teknik Jakarta melakukan pencarian tempat pemasangan yang cocok dan masih tersedia pada rak *server*.
20. Kepala Teknik Jakarta menentukan tempat yang cocok untuk pemasangan rak dan perangkat *server*.
21. Kepala Teknik Surabaya melakukan koordinasi pemasangan rak dan perangkat *server* dengan Kepala Teknik Surabaya.
22. Kepala Teknik Surabaya menerima koordinasi pemasangan rak dan perangkat *server*.
23. Kepala Teknik Jakarta melakukan observasi lapangan untuk melakukan pemasangan rak dan perangkat *server*.
24. Kepala Teknik Surabaya menyetujui tempat pemasangan rak dan perangkat *server*.
25. Kepala Teknik Surabaya melakukan koordinasi dengan Kepala Teknik Jakarta untuk melakukan pemasangan rak dan perangkat *server*.
26. Kepala Teknik Jakarta menerima koordinasi dan melakukan persiapan untuk pemasangan rak dan perangkat *server*.
27. Kepala Teknik Jakarta koordinasi dengan *Client* untuk melakukan pemasangan rak dan perangkat *server*.
28. *Client* menerima hasil keputusan dari Kepala Teknik Jakarta.
29. *Client* melakukan pemasangan rak dan perangkat *server*.

30. Proses bisnis selesai.





Gambar 4.1 Proses Bisnis As-is Pendataan Rak dan Pemasangan Perangkat Server

4.1.2 Analisis Masalah Pendataan Rak Server dan Perangkat Server

Analisis masalah digunakan untuk memahami suatu masalah yang terjadi dan harus diselesaikan oleh pemangku kepentingan. Analisis masalah dalam penelitian ini dilakukan dengan menggunakan hasil wawancara dengan petugas *server* dan kepala teknik bakbound PT. Indosat Ooredoo Surabaya. Analisis masalah yang didapatkan dari hasil wawancara petugas *server* dan kepala teknik backbound PT Indosat Ooredoo Surabaya terdapat pada tabel 4.1.

Tabel 4.1 Analisis Masalah

Masalah	<ul style="list-style-type: none"> - Pendataan rak dan perangkat <i>server</i> yang dilakukan secara manual dengan menggunakan Microsoft Word. - Pendataan rak dan perangkat <i>server</i> yang dilakukan secara manual dengan menggunakan aplikasi Sketchup. - Pendataan rak dan perangkat <i>server</i> ketika dimasukkan pada Microsoft Word tidak terjadi pembaharuan sehingga informasi menjadi tidak <i>realtime</i>. - Pendataan rak dan perangkat <i>server</i> ketika dimasukkan pada aplikasi Sketchup tidak terjadi pembaharuan sehingga informasi menjadi tidak <i>realtime</i>. - Pengecekan rak dan perangkat server masih dilakukan secara manual dengan melihat informasi rak dan perangkat server melalui aplikasi Sketchup. - Pencarian rak dan perangkat server masih dilakukan secara manual dengan melihat informasi rak dan perangkat server melalui aplikasi Sketchup.
Mempengaruhi	<ul style="list-style-type: none"> - Pendataan rak dan perangkat server yang tidak <i>realtime</i>. - Memperbaharui informasi rak dan perangkat server membutuhkan waktu lebih lama karena masih secara manual. - Pencarian tata letak rak dan perangkat server membutuhkan waktu yang lebih lama karena datanya masih tidak <i>uptodate</i>.

Berdampak pada	<ul style="list-style-type: none"> - Informasi pendataan rak dan perangkat <i>server</i> yang tidak sesuai dengan kenyataan. - Menyulitkan mencari tempat untuk melakukan pemasangan rak dan perangkat <i>server</i>.
Solusi sukses	<ul style="list-style-type: none"> - Sistem yang mampu menyediakan layanan pendataan rak dan perangkat <i>server</i>. - Sistem yang mampu menyediakan layanan perubahan informasi rak dan perangkat <i>server</i> apabila rak dan perangkat <i>server</i> sudah tidak terpakai lagi.

4.1.3 Proses Bisnis *To-be*

Proses bisnis *to-be* yang dimodelkan dalam penelitian ini merupakan proses bisnis usulan yang ditawarkan dalam penelitian ini kepada pemangku kepentingan perusahaan. Proses bisnis *to-be* diidentifikasi berdasarkan hasil analisis terhadap proses bisnis *as-is*. Dalam penelitian ini, analisis proses bisnis *to-be* dilakukan dengan melakukan penambahan dan penghapusan dari beberapa aktivitas. Selanjutnya, proses bisnis *to-be* dimodelkan untuk menggambarkan perubahan atau tambahan yang diusulkan melalui penelitian ini.

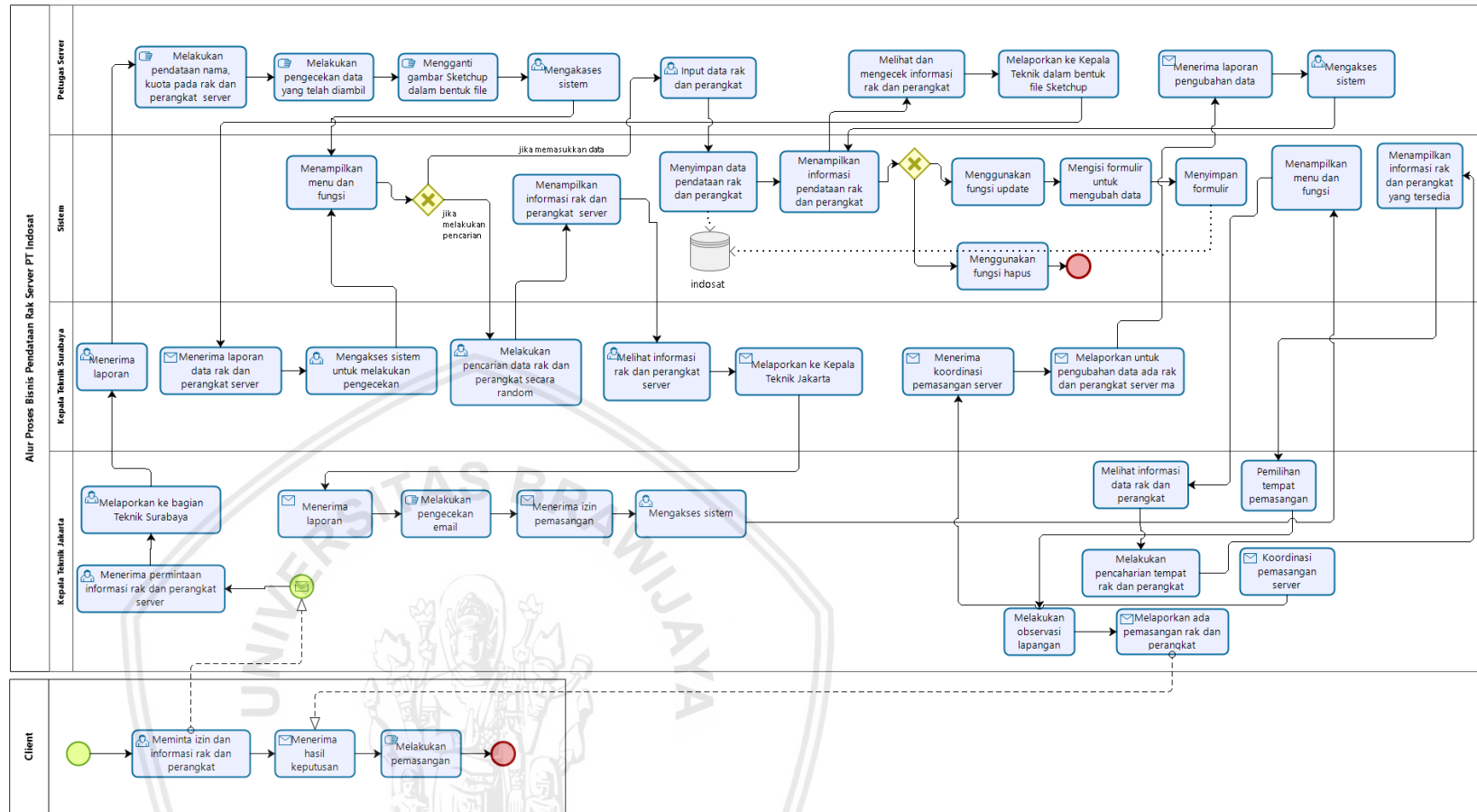
4.1.3.1 Proses Bisnis Pendataan Rak *Server* dan Pemasangan Perangkat *Server*

Berdasarkan hasil analisa permasalahan yang sudah dijelaskan sebelumnya, maka proses bisnis untuk menangani pelanggan akan diubah. Proses bisnis yang baru ditunjukkan pada Gambar 4.2. Proses *to-be* pendataan rak dan perangkat *server* secara alurnya dapat dijelaskan sebagai berikut ini:

1. *Client* meminta izin dan informasi rak dan perangkat server kepada Kepala Teknik Jakarta.
2. Kepala Teknik Jakarta menerima permintaan informasi rak dan perangkat server dan melaporkan ke bagian Kepala Teknik Surabaya.
3. Kepala Teknik Surabaya menerima laporan.
4. Petugas server melakukan persiapan pendataan rak dan perangkat server.
5. Petugas *server* melakukan pendataan nama, kuota dan isi pada rak *server*.
6. Petugas *server* melakukan pengecekan terhadap data yang telah diambil pada proses sebelumnya.
7. Petugas *server* menyimpan data tersebut dalam Microsot Word.
8. Petugas *server* menyimpan data dalam bentuk Sketchup.
9. Petugas *server* mengganti data dari Sketchup menjadi file noted agar bisa diakses oleh sistem.

10. Petugas *server* mengakses sistem informasi pendataan rak dan perangkat *server*.
11. Sistem menampilkan menu dan fungsi sistem informasi pendataan rak dan perangkat *server*.
12. Petugas *server* mengisi data rak dan perangkat *server*.
13. Sistem menyimpan data pendataan rak dan perangkat *server* pada database.
14. Sistem menampilkan informasi pendataan rak dan perangkat *server*.
15. Petugas *server* melakukan pengecekan dengan melihat informasi pada sistem informasi pendataan rak dan perangkat *server*.
16. Petugas *server* melaporkan data rak dan perangkat *server* kepada Kepala Teknik.
17. Kepala Teknik menerima laporan data rak dan perangkat *server*.
18. Kepala Teknik mengakses sistem informasi pendataan rak dan perangkat *server* untuk melakukan pengecekan terhadap sistem telah sesuai dengan data yang diisi.
19. Sistem menampilkan menu dan fungsi sistem informasi pendataan rak dan perangkat *server*.
20. Kepala Teknik melakukan pencarian data rak dan perangkat *server* secara random.
21. Sistem menampilkan informasi rak dan perangkat *server* yang dicari oleh Kepala Teknik.
22. Kepala Teknik melihat informasi rak dan perangkat *server* yang dicari tadi.
23. Kepala Teknik melaporkan informasi rak dan perangkat kepada Kepala Teknik Jakarta.
24. Kepala Teknik Jakarta menerima laporan dari Kepala Teknik Surabaya.
25. *Client* meminta untuk melakukan pemasangan rak dan perangkat *server* kepada Kepala Teknik Jakarta.
26. Kepala Teknik Jakarta mengakses sistem informasi pendataan rak dan perangkat *server*.
27. Sistem menampilkan menu dan fungsi sistem informasi pendataan rak dan perangkat *server*.
28. Kepala Teknik Jakarta melihat informasi rak dan perangkat *server* yang ada pada sistem informasi pendataan rak dan perangkat *server*.
29. Kepala Teknik Jakarta melakukan pencarian tempat rak dan perangkat *server* yang tersedia pada sistem informasi pendataan rak dan perangkat *server* untuk melakukan pemasangan.

30. Sistem menampilkan informasi rak dan perangkat server yang masih tersedia tempat untuk dilakukan pemasangan rak dan perangkat yang baru.
31. Kepala Teknik Jakarta melakukan pemilihan tempat untuk melakukan pemasangan rak dan perangkat *server*.
32. Kepala Teknik Jakarta mengirimkan pesan kepada Kepala Teknik Surabaya untuk melakukan koordinasi pemasangan.
33. Kepala Teknik Surabaya menerima koordinasi pemasangan rak dan perangkat *server*.
34. Kepala Teknik Surabaya melaporkan untuk perubahan data pada sistem karena ada pemasangan rak dan perangkat server yang baru.
35. Petugas *server* menerima laporan untuk melakukan perubahan data pada sistem.
36. Petugas *server* mengakses sistem informasi pendataan rak dan perangkat server untuk melakukan pengubah data pada sistem.
37. Sistem menampilkan informasi pendataan rak dan perangkat server.
38. Jika petugas *server* melakukan perubahan informasi pada rak dan perangkat server maka menggunakan fungsi update dan jika rak dan perangkat server tidak lagi terpakai maka menggunakan fungsi hapus.
39. Sistem menampilkan formulir untuk melakukan perubahan informasi rak dan perangkat *server*.
40. Sistem menyimpan data rak dan perangkat server yang baru.
41. Kepala Teknik Jakarta melakukan observasi lapangan untuk melihat ruangan *server* yang akan dibuat tempat pemasangan.
42. Kepala Teknik Jakarta melaporkan kepada *Client* ada tempat pemasangan rak dan perangkat *server*.
43. *Client* menerima hasil keputusan dari Kepala Teknik Jakarta.
44. *Client* melakukan pemasangan rak dan perangkat *server*.
45. Selesai.



Gambar 4.2 Proses Bisnis *To-be* Pendataan Rak dan Pemasangan Perangkat Server

Tabel 4.2 Perubahan Aktivitas Proses Bisnis

Unit bisnis	Proses bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
Petugas <i>server</i>	Pendataan rak dan perangkat <i>server</i> secara manual menggunakan Microsoft Word.	-	Dieliminasi
	Pendataan rak dan perangkat <i>server</i> secara manual menggunakan aplikasi Sketchup.	Pendataan rak dan perangkat <i>server</i> menggunakan sistem berupa masukan melalui formulir.	Diubah
	Pendataan rak dan perangkat <i>server</i> ketika dimasukkan pada Microsoft Word tidak terjadi pembaharuan sehingga informasi menjadi tidak <i>realtime</i> .	Pendataan rak dan perangkat <i>server</i> menggunakan sistem dapat melakukan perubahan dengan fungsi <i>update</i> dan hapus.	Diubah
	Pendataan rak dan perangkat <i>server</i> ketika dimasukkan pada aplikasi Sketchup tidak terjadi pembaharuan sehingga informasi menjadi tidak <i>realtime</i> .	Pendataan rak dan perangkat <i>server</i> menggunakan sistem tetap menggunakan Sketchup tetapi gambarnya diubah menjadi file.	Diubah
Kepala Teknik (Indosat Surabaya)	Melakukan pengecekan kuota pada rak dan perangkat <i>server</i> pada Sketchup.	Melakukan pengecekan kuota pada rak dan perangkat <i>server</i> pada sistem.	Diubah
	Pencarian tempat pemasangan rak dan	Pencarian tempat pemasangan rak	Diubah

	perangkat server masih menggunakan teknik manual yaitu mencari melalui aplikasi Sketchup.	dan perangkat server menggunakan fungsi cari yang ada pada sistem.	
Kepala Teknik (Indosat Jakarta)	Melakukan pengecekan kuota pada rak dan perangkat server pada Sketchup.	Melakukan pengecekan kuota pada rak dan perangkat server pada sistem.	Diubah
	Pencarian tempat pemasangan rak dan perangkat server masih menggunakan teknik manual yaitu mencari melalui aplikasi Sketchup.	Pencarian tempat pemasangan rak dan perangkat server menggunakan fungsi cari yang ada pada sistem.	Diubah

Pada tabel 4.2 menjelaskan mengenai perubahan aktivitas proses bisnis pendataan rak server dan perangkat server. Pada proses bisnis tersebut mengalami perubahan ada yang diganti dan ada yang ditambahkan. Proses bisnis yang mengalami perubahan diperoleh dengan menganalisis dari hasil masalah yang ada pada proses bisnis *as-is* pendataan rak dan perangkat server. Analisis masalah pada proses bisnis pendataan rak dan perangkat server diperoleh dari wawancara dengan Petugas Server dan Kepala Teknik Backbound. Setelah menganalisis proses bisnis *as-is* menghasilkan proses bisnis usulan yaitu proses bisnis *to-be*. Kemudian hasil proses bisnis *to-be* dilaporkan kepada perusahaan untuk melakukan koordinasi proses bisnis tersebut telah sesuai dengan kebutuhan yang diharapkan.

4.2 Analisis Persyaratan

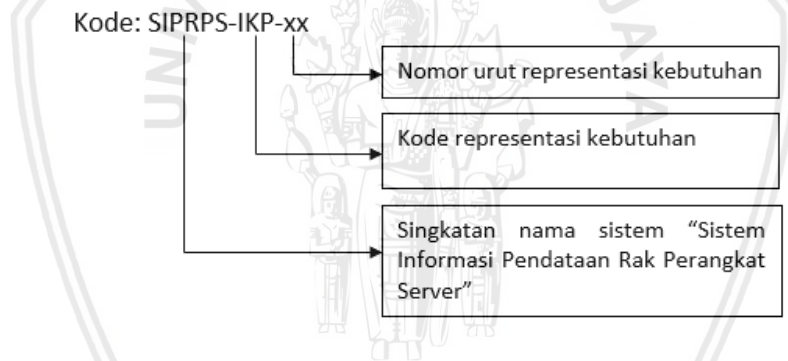
4.2.1 Identifikasi Tipe Pemangku Kepentingan

Pada bagian sub bab ini menjelaskan mengenai identifikasi tipe-tipe pemangku kepentingan yang dilakukan untuk mengetahui dan mengelompokkan pemangku kepentingan yang berkaitan dengan sistem informasi yang dibangun dan mewakili tiap tipe pemangku kepentingan. Penjelasan mengenai tipe pemangku kepentingan dapat dilihat pada Tabel 4.3

Tabel 4.3 Tabel Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku kepentingan
Pengguna	Individu yang berperan langsung dalam menggunakan sistem. Pengguna memiliki peran secara langsung yang di definisikan sebagai aktor dalam <i>usecase</i>	Petugas <i>server</i> , kepala teknik bakbound
Pengembang	Individu yang berperan dalam mengembangkan sistem informasi pendataan rak <i>server</i> dan perangkat <i>server</i>	Peneliti
Pihak yang berwenang	Organisasi atau individu yang berperan memberikan dukungan terhadap pengembangan sistem informasi	PT Indosat Surabaya

4.2.2 Identifikasi Kebutuhan Pemangku Kepentingan



Gambar 4.3 Identifikasi Kebutuhan Pemangku Kepentingan

Tabel 4.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan
IKPK-SIPRPS-01	Sistem harus mampu menyediakan kemudahan layanan dalam hal pendataan rak <i>server</i> dan perangkat <i>server</i> .	Petugas <i>server</i> .	Petugas <i>server</i> mendata rak <i>server</i> dan perangkat <i>server</i> secara manual dengan menggunakan <i>Microsoft Word</i> .	Sistem informasi yang mampu menyediakan layanan berupa form pendataan rak <i>server</i> dan perangkat <i>server</i> .

IKPK-SIPRPS-02	Sistem harus mampu menyediakan kemudahan layanan berupa memperbarui data rak <i>server</i> dan perangkat <i>server</i> .	Petugas <i>server</i> .	Tidak ada.	Sistem informasi yang mampu menyediakan layanan berupa <i>update</i> .
IKPK-SIPRPS-03	Sistem harus mampu menyediakan kemudahan layanan berupa pencarian.	Petugas <i>server</i> .	Tidak ada.	Sistem informasi yang mampu menyediakan layanan berupa pencairan rak <i>server</i> dan perangkat <i>server</i> .
IKPK-SIPRPS-04	Sistem dapat digunakan untuk menampilkan informasi pendataan rak <i>server</i> dan perangkat <i>server</i> .	Petugas <i>server</i> , Kepala Teknik Backbound.	Tidak ada.	Sistem informasi yang mampu menyediakan layanan berupa informasi rak <i>server</i> dan perangkat <i>server</i> .

4.2.3 Identifikasi Pengguna

Identifikasi pengguna adalah tahap yang perlu dilakukan untuk mengidentifikasi individu yang berhubungan dengan sistem. Pengguna sistem merupakan bagian dari kategori pengguna dalam tipe pemangku kepentingan. Pada Tabel 4.4 menjelaskan mengenai identifikasi pengguna dari sistem.

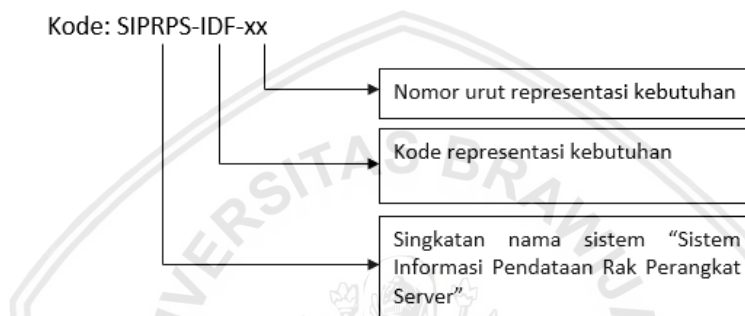
Tabel 4.4 Identifikasi Pengguna

Tipe Pemangku Kepentingan	Tipe Pengguna	Deskripsi
	Petugas <i>Server</i>	Petugas <i>server</i> yang mempunyai wewenang dalam melakukan pendataan rak <i>server</i> dan perangkat



PT. Indosat Ooredoo Surabaya		server yang kemudian dilaporkan kepada kepala teknik backbound.
	Kepala Teknik Backbound.	Kepala teknik backbound selaku pengawas ruangan server PT Indosat Ooredoo yang memiliki jabatan tinggi dalam divisi teknik bertugas mengawasi dan bertanggung jawab terhadap pendataan rak server dan perangkat server.

4.2.4 Identifikasi Fitur



Gambar 4.4 Identifikasi Fitur

Pada sub bab ini menjelaskan mengenai fitur yang terdapat pada sistem yang memberikan beberapa solusi yang ditawarkan untuk memenuhi kebutuhan pengguna, sehingga dapat menyelesaikan masalah dari pemangku kepentingan. Pada Tabel 4.5 menjelaskan mengenai beberapa fitur yang akan dikembangkan oleh sistem.

Tabel 4.5 Identifikasi Fitur

Kode Fitur	Deskripsi
SIPRPS-IDF-01	Sistem mampu mengenali identitas pengguna dan membatasi hak akses layanan terhadap pengguna yang tidak berwenang di dalam sistem.
SIPRPS-IDF-02	Sistem mampu digunakan untuk mengelola pendataan rak server berupa tampilan menambah rak, menghapus rak, menampilkan informasi rak dan mencari nama rak yang dibutuhkan.
SIPRPS-IDF-03	Sistem mampu digunakan untuk mengelola pendataan perangkat server berupa tampilan menambah perangkat, menghapus perangkat, menampilkan informasi perangkat, mencari nama perangkat yang dicari dan memperbarui perangkat.

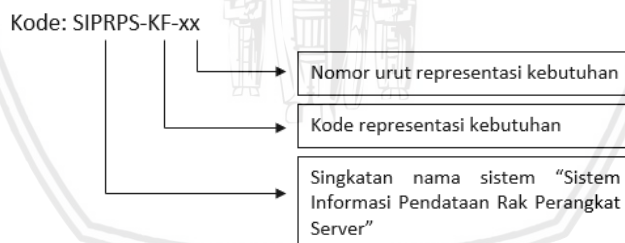
SIPRPS-IDF-04	Sistem mampu digunakan untuk mengelola pendataan lantai berupa tampilan informasi isi lantai ruangan <i>server</i> , menambahkan nama lantai ruangan <i>server</i> dan menampilkan pencarian berdasarkan lantai ruangan <i>server</i> .
SIPRPS-IDF-05	Sistem mampu digunakan untuk mengelola petugas yang terdiri dari menambah petugas dan menghapus petugas yang sudah tidak berkepentingan.

Pada Tabel 4.6 masing-masing fitur telah dikelompokkan berdasarkan keterkaikan terhadap kebutuhan pengguna yang perlu dipenuhi. Pengelompokan ini menggambarkan hubungan relasi antara fitur dengan kebutuhan pengguna. Pada Tabel 4.6 menjelaskan mengenai hubungan relasi fitur dengan kebutuhan pengguna.

Tabel 4.6 Relasi Fitur dengan Kebutuhan Pengguna

Fitur	Kebutuhan Pengguna
SIPRPS-IDF-01	IKPK-SIPRPS-01
SIPRPS-IDF-02	IKPK-SIPRPS-01, IKPK-SIPRPS-02, IKPK-SIPRPS-03
SIPRPS-IDF-03	IKPK-SIPRPS-01, IKPK-SIPRPS-02, IKPK-SIPRPS-03
SIPRPS-IDF-04	IKPK-SIPRPS-01
SIPRPS-IDF-05	IKPK-SIPRPS-02

4.2.5 Persyaratan Fungsional



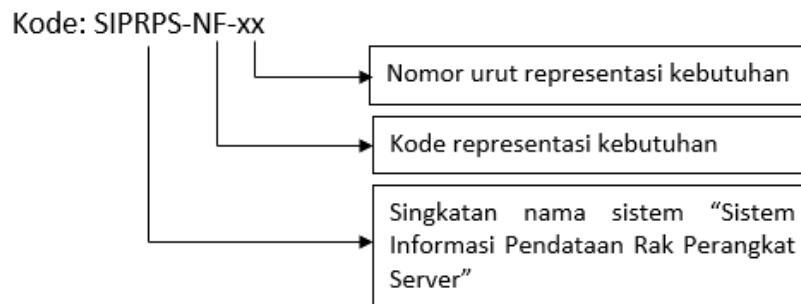
Gambar 4.5 Persyaratan Fungsional

Pada sub bab ini akan menjelaskan tentang persyaratan fungsional sistem informasi pendataan rak *server* dan perangkat *server* pada PT Indosat Ooredoo Surabaya. Persyaratan fungsional adalah proses atau layanan yang harus dipenuhi oleh sistem. Dasar dari persyaratan fungsional adalah identifikasi fitur pada sub bab sebelumnya. Pada Tabel 4.3 menunjukkan hubungan antara persyaratan fungsional dengan fitur pada sistem informasi yang akan dikembangkan. Persyaratan fungsional memiliki kode yang sesuai dengan Gambar 4.5.

Tabel 4.3 Kebutuhan Fungsional

Kode Fitur	Kode Persyaratan	Deskripsi
SIPRPS-IDF-01	SIPRPS-KF-01	Sistem mampu menampilkan antarmuka <i>login</i> .
	SIPRPS-KF-02	Sistem mampu mengidentifikasi pengguna berdasarkan data yang tersimpan.
SIPRPS-IDF-02	SIPRPS-KF-03	Sistem mampu menampilkan antarmuka daftar rak.
	SIPRPS-KF-04	Sistem mampu menampilkan formulir pendataan rak.
	SIPRPS-KF-05	Sistem mampu menambah data rak.
	SIPRPS-KF-06	Sistem mampu menghapus data rak.
	SIPRPS-KF-07	Sistem mampu menampilkan informasi data rak yang dicari.
SIPRPS-IDF-03	SIPRPS-KF-08	Sistem mampu menampilkan formulir pendataan perangkat.
	SIPRPS-KF-09	Sistem mampu menambah data perangkat.
	SIPRPS-KF-10	Sistem mampu menghapus data perangkat.
	SIPRPS-KF-11	Sistem mampu memperbarui informasi perangkat.
	SIPRPS-KF-12	Sistem mampu mengubah informasi perangkat.
	SIPRPS-KF-13	Sistem mampu menampilkan informasi data perangkat yang dicari.
SIPRPS-IDF-04	SIPRPS-KF-14	Sistem mampu menampilkan formulir pendataan lantai.
	SIPRPS-KF-15	Sistem mampu menambah data lantai.
SIPRPS-IDF-05	SIPRPS-KF-16	Sistem mampu menambah <i>user</i> yang diinginkan oleh admin.
	SIPRPS-KF-17	Sistem mampu menghapus <i>user</i> yang tidak diperlukan.
	SIPRPS-KF-18	Sistem mampu memperbarui data <i>user</i> .
	SIPRPS-KF-19	Sistem mampu mencari nama pengguna yang diinginkan.

4.2.6 Persyaratan Nonfungsional



Gambar 4.6 Persyaratan Nonfungsional

Kebutuhan nonfungsional akan menjelaskan mengenai batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu, kemudahan penggunaan sistem, keamanan sistem, dan lain sebagainya. Pada Tabel 4.4 akan dijelaskan mengenai kebutuhan nonfungsional yang terdapat pada sistem informasi berbasis web pada PT Indosat Ooredoo Surabaya yaitu sebagai berikut.

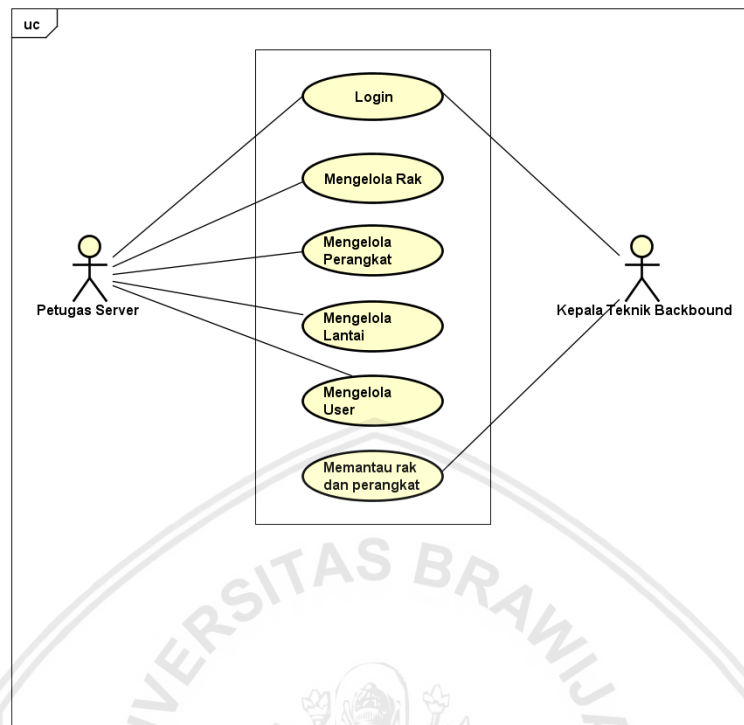
Tabel 4.4 Analisis Kebutuhan Nonfungsional

No.	Kode	Parameter	Deskripsi
1.	SIPRPS-NF-01	<i>Compatibility</i>	Perangkat lunak dapat diakses melalui minimal 3 <i>browser</i> yang berbeda

4.3 Pemodelan Use case

Use case adalah sebuah gambaran hubungan antara aktor dengan sistemnya. Identifikasi aktor pada *use case* dilakukan untuk mengelompokkan pengguna ke dalam beberapa peran. Tujuan dari identifikasi *use case* adalah untuk menentukan aktor dalam menggunakan sistem. Aktor menggunakan sistem mengacu pada fitur yang telah dibuat oleh pengguna. Pada Gambar 4.8 menunjukkan *use case diagram*.

4.3.1 Use case Diagram



Gambar 4.7 Use case Diagram PT Indosat Ooredoo Surabaya

Pada Gambar 4.7 menunjukkan *use case diagram* sistem informasi PT Indosat Ooredoo Surabaya, pengguna dari perangkat lunak ini terdiri dari 2 aktor yaitu petugas server, kepala teknik backbound. Petugas server merupakan pengguna utama pada perangkat lunak ini sehingga petugas server dapat menggunakan semua fungsi yang ada pada sistem informasi diantaranya login, kelola rak, kelola perangkat, kelola lantai, kelola user serta memantau rak dan perangkat. Sedangkan untuk bagian kepala teknik backbound dapat menggunakan fungsi login serta memantau rak dan perangkat server.

Use case dihubungkan dengan pemangku kepentingan untuk membantu memahami aktivitas yang dilakukan oleh pengguna. Pada Tabel 4.5 Menunjukkan hubungan antara *use case* dengan pemangku kepentingan.

Tabel 4.5 Tabel Hubungan Use Case dengan Pengguna

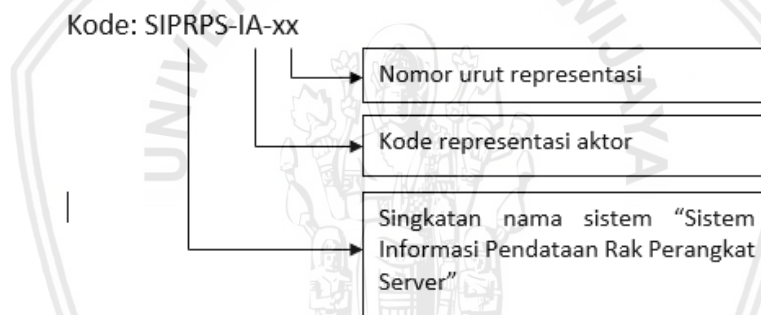
Use Case	Pengguna
<i>Login</i>	Petugas Server, Kepala Teknik Backbound
Mengelola rak	Petugas Server
Mengelola perangkat	Petugas Server
Mengelola lantai	Petugas Sever
Mengelola user	Petugas Server
Memantau rak dan perangkat	Petugas Server, Kepala Teknik Backbound

Use case yang telah teridentifikasi dihubungkan dengan fitur yang telah teridentifikasi. Hal ini dilakukan untuk mempertegas hasil pemodelan use case sama dengan persyaratan. Pada Tabel 4.7 menjelaskan hubungan use case dengan fitur.

Tabel 4.6 Hubungan Use Case dengan Fitur

Use Case	Fitur
<i>Login</i>	SIPRPS-IDF-01
Mengelola rak	SIPRPS-IDF-02, SIPRPS-IDF-05
Mengelola perangkat	SIPRPS-IDF-03, SIPRPS-IDF-05
Mengelola lantai	SIPRPS-IDF-04
Mengelola user	SIPRPS-IDF-06
Memantau rak dan perangkat	SIPRPS-IDF-07

4.3.2 Deskripsi Aktor



Gambar 4.8 Deskripsi Aktor

Pada sub bab ini akan menjelaskan tentang siapa saja yang akan menggunakan sistem pada sistem informasi pada PT Indosat Ooredoo Surabaya. Pada Tabel 4.7 menjelaskan tentang identifikasi aktor yang ada pada sistem informasi PT Indosat Ooredoo Surabaya.

Tabel 4.7 Identifikasi Aktor

Kode aktor	Aktor	Deskripsi
SIPRPS-IA-01	Petugas <i>Server</i>	Petugas <i>server</i> yang mengatur dan mendata seluruh kegiatan yang berhubungan dengan <i>server</i> secara langsung. Oleh sebab itu kegunaan utama sistem ini ditunjukan untuk lebih memudahkan petugas <i>server</i> .

SIPRPS-IA-02	Kepala Teknik Backbound	Kepala Teknik Backbound bertugas untuk melakukan pengecekan dan pemantauan pada sistem informasi web.
--------------	-------------------------	---

4.3.3 Use Case Spesification

Kebutuhan fungsional dari perangkat lunak ini telah ditentukan pada sub bab sebelumnya, agar lebih mempermudah dalam mendefinisikan apa yang akan dilakukan pada kebutuhan fungsional diperlukan penjelasan yang lebih rinci pada tiap kebutuhan fungsional yang telah didefinisikan menggunakan spesifikasi *use case*. Berikut ini adalah penjelasan mengenai spesifikasi *use case* pada saat aktor menjalankan *use case*, kondisi saat *use case* sebelum dan sesudah dijalankan, dan tahap-tahap ketika *use case* dilakukan.

4.3.3.1 Spesifikasi Use Case Login

Penjelasan mengenai spesifikasi *use case login* menggambarkan alur pengguna masuk ke dalam sistem dan melakukan login. Pada Tabel 4.8 menjelaskan alur *login* aktor terhadap sistem.

Tabel 4.8 Spesifikasi Use Case Login

Login (SIPRPS-IDF-01)	
Brief Description	Fungsi spesifikasi <i>use case</i> ini berguna untuk keamanan identitas pengguna yang akan masuk ke dalam sistem.
Actor	Petugas <i>server</i> , kepala teknik backbound
Pre-condition	<ul style="list-style-type: none"> - Komputer yang digunakan dapat mengakses sistem dan terhubung dengan internet. - Sitem dapat terhubung dengan <i>server</i>. - Sistem menyimpan data pengguna.
Post-condition	<ul style="list-style-type: none"> - Identitas aktor dapat dikenali oleh sistem. - Status <i>Login</i> berhasil - Sistem menampilkan informasi hak akses pengguna.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika petugas <i>server</i> memilih masuk ke sistem. <p>{mengisi data}</p> <ol style="list-style-type: none"> 2. Petugas <i>server</i> mengisi <i>username</i> dan <i>password</i> dengan benar. 3. Petugas <i>server</i> mengirimkan <i>username</i> dan <i>password</i>.

	<p>4. Sistem mengecek data <i>username</i> dan <i>password</i>. {cek isi}</p> <p>5. Sistem mengidentifikasi <i>username</i> dan <i>password</i>. {mengidentifikasi email dan password}</p> <p>6. Sistem menyimpan informasi identitas pengguna.</p> <p>7. Status <i>login</i> berhasil.</p> <p>8. <i>Use case</i> selesai</p>
Alternative Flow	<p>A1. Mengatasi email dan password kosong</p> <p>Pada {cek isi} ketika memasukkan email dan <i>password</i> kosong maka sistem menampilkan pesan silahkan isi kolom ini, kemudian kembali pada {mengisi data}.</p> <p>A2. Mengatasi email dan password salah</p> <p>Pada {mengidentifikasi email dan password} ketika email dan password tidak dapat ditemukan maka sistem menampilkan pesan email dan password tidak sesuai.</p>

4.3.3.2 Spesifikasi *Use Case* Mengelola Rak

Penjelasan mengenai spesifikasi *use case* mengelola rak menggambarkan alur pengguna melakukan pendataan rak. Pengguna bisa menambahkan rak, menghapus rak dan memperbarui rak dengan mengisi data pada formulir pendataan yang telah disediakan oleh sistem. Pada Tabel 4.9 menjelaskan alur kegiatan aktor mengelola rak sistem.

Tabel 4.9 Spesifikasi *Use Case* Mengelola Rak

Mengelola Rak (SIPRPS-IDF-02)	
Brief Description	Fungsi spesifikasi use case ini berguna untuk mengelola pendataan rak server sehingga menjadi sebuah informasi rak server yang mempermudah petugas server dalam menjalankan tugas
Actor	Petugas server
Pre-condition	<ul style="list-style-type: none"> - Petugas server berhasil masuk kedalam sistem - Perangkat yang mengakses sistem dapat terhubung dengan internet - Server terhubung dengan sistem
Post-Condition	<ul style="list-style-type: none"> - Petugas server berhasil menyimpan data rak - Petugas server dapat melihat informasi rak

<p>Basic Flow</p>	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika petugas <i>server</i> memilih untuk melakukan tambah rak 2. Sistem menampilkan formulir data tambah rak <i>server</i> 3. Petugas <i>server</i> mengisi formulir data tambah rak <i>server</i> <p>{menyimpan data rak server}</p> <ol style="list-style-type: none"> 4. Petugas <i>server</i> memilih menyimpan data rak <i>server</i> 5. Sistem menyimpan data rak <i>server</i> 6. Menampilkan pesan berhasil disimpan 7. Use case selesai
<p>Alternative Flow</p>	<p>A1. Menghapus rak sever</p> <ol style="list-style-type: none"> 1. Petugas <i>server</i> menggunakan menu daftar rak <p>{penghapusan data rak server}</p> <ol style="list-style-type: none"> 2. Sistem menampilkan halaman informasi daftar rak yang telah tercantum <p>{menampilkan informasi rak server}</p> <ol style="list-style-type: none"> 3. Petugas <i>server</i> memilih menghapus rak <i>server</i> 4. Sistem menampilkan pesan bahwa rak telah terhapus dan menampilkan informasi data rak yang masih tersedia 5. Use case selesai <p>A2. Melihat informasi rak</p> <p>Pada saat {menampilkan informasi rak server} di <i>alternative flow</i>, jika Petugas <i>Server</i> memilih melihat informasi rak <i>server</i>, maka lakukan <i>subflow</i> Melihat Informasi Rak</p> <p>A3. Melakukan pencarian rak</p> <p>Pada saat {menampilkan informasi rak server} di <i>alternative flow</i>, jika Petugas <i>server</i> memilih mencari informasi rak <i>server</i>, maka lakukan <i>subflow</i> Melakukan Pencarian Rak</p>
<p>Sub-flow</p>	<p>S1. Melihat informasi rak</p> <ol style="list-style-type: none"> 1. Petugas <i>server</i> memilih lihat untuk melihat informasi rak <i>server</i> <p>{memuat informasi rak server}</p>



	<p>2. Sistem menampilkan informasi mengenai rak <i>server</i> tersebut</p> <p>{menampilkan informasi rak <i>server</i>}</p> <p>3. <i>Use case</i> selesai</p> <p>S2. Melakukan pencarian rak</p> <p>1. Petugas <i>server</i> menggunakan fungsi cari untuk mencari rak yang diperlukan</p> <p>2. Petugas <i>server</i> memasukkan nama rak yang akan dicari</p> <p>{memuat data rak}</p> <p>3. Sistem memuat informasi rak yang dibutuhkan</p> <p>{menampilkan informasi rak}</p> <p>4. Sistem menampilkan informasi rak yang dibutuhkan</p> <p>5. <i>Use case</i> selesai</p>
--	--

4.3.3.3 Spesifikasi *Use Case* Mengelola Perangkat

Penjelasan mengenai spesifikasi *use case* mengelola perangkat menggambarkan alur pengguna melakukan pendataan perangkat. Pengguna bisa menambahkan perangkat, menghapus perangkat dan memperbaiki perangkat dengan mengisi data pada formulir pendataan yang telah disediakan oleh sistem. Pada Tabel 4.10 menjelaskan alur kegiatan aktor mengelola perangkat sistem.

Tabel 4.10 Spesifikasi *Use Case* Mengelola Perangkat

Mengelola Perangkat (SIPRPS-IDF-03)	
Brief Description	Fungsi pada spesifikasi <i>use case</i> kelola perangkat ini berguna untuk mengelola data perangkat pada suatu rak <i>server</i> agar mempermudah dalam melakukan pencarian dan pemasangan perangkat
Actor	Petugas <i>server</i>
Pre-condition	<ul style="list-style-type: none"> - Petugas <i>server</i> berhasil masuk kedalam sistem - Perangkat yang mengakses sistem dapat terhubung dengan internet - <i>Server</i> terhubung dengan sistem
Post-condition	<ul style="list-style-type: none"> - Petugas <i>server</i> berhasil menyimpan data perangkat - Petugas <i>server</i> dapat melihat informasi perangkat
Basic Flow	{use case dimulai}

	<ol style="list-style-type: none"> 1. Use case dimulai ketika petugas <i>server</i> memilih menu daftar perangkat 2. Sistem menampilkan halaman informasi daftar perangkat 3. Petugas <i>server</i> memilih menambah daftar perangkat <i>server</i> 4. Sistem menampilkan formulir data tambah perangkat <i>server</i> 5. Petugas <i>server</i> mengisi formulir data tambah perangkat <i>server</i> <p style="text-align: center;">{menyimpan data perangkat server}</p> <ol style="list-style-type: none"> 6. Petugas <i>server</i> memilih menyimpan data perangkat <i>server</i> 7. Sistem menyimpan data perangkat <i>server</i> 8. Menampilkan pesan berhasil disimpan 9. Use case selesai
<p>Alternatif Flow</p>	<p>A1. Menghapus perangkat sever</p> <ol style="list-style-type: none"> 1. Petugas <i>server</i> menggunakan menu daftar perangkat 2. Sistem menampilkan halaman informasi daftar perangkat yang telah tercantum <p style="text-align: center;">{penghapusan data perangkat server}</p> <ol style="list-style-type: none"> 3. Petugas <i>server</i> memilih menghapus rak perangkat <i>server</i> <p style="text-align: center;">{menampilkan informasi perangkat server}</p> <ol style="list-style-type: none"> 4. Sistem menampilkan pesan perangkat telah dihapus dan menampilkan informasi data perangkat yang masih tersedia 5. Use case selesai <p>A2. Melihat informasi perangkat server</p> <p>Pada saat {menampilkan informasi perangkat server} di <i>alternative flow</i>, jika Petugas <i>server</i> memilih melihat informasi rak <i>server</i>, maka lakukan <i>subflow</i> Melihat Informasi Rak</p> <p>A3. Mengubah data perangkat server</p> <p>Pada {menampilkan rekap maintenance} di <i>alternative flow</i>, jika Petugas <i>server</i> memilih mengubah data</p>

	perangkat <i>server</i> , maka lakukan <i>subflow</i> Mengubah Data Perangkat Server
Sub-flow	<p>S1. Melihat informasi perangkat server</p> <ol style="list-style-type: none"> 1. Petugas <i>server</i> memilih fungsi lihat untuk melihat informasi perangkat <i>server</i> {menampilkan informasi perangkat server} 2. Sistem menampilkan halaman detail perangkat yang berisi informasi detail perangkat <i>server</i> beserta gambar Sketchup 3. Use case selesai <p>S2. Mengubah data perangkat server {memuat data perangkat yang akan diubah}</p> <ol style="list-style-type: none"> 1. Petugas <i>server</i> menggunakan fungsi <i>update</i> 2. Sistem menampilkan halaman formulir <i>update</i> perangkat yang berisi form perangkat yang akan di perbarui 3. Petugas <i>server</i> mengisi formulir <i>update</i> perangkat <i>server</i> 4. Petugas <i>server</i> memilih menyimpan data perangkat <i>server</i> {mengubah data perangkat server} 4. Sistem menyimpan data perangkat yang telah diubah 5. Sistem menampilkan pesan data perangkat telah diperbarui 6. Use case selesai

4.3.3.4 Spesifikasi Use Case Mengelola Lantai

Penjelasan mengenai spesifikasi *use case* mengelola lantai menggambarkan alur pengguna melakukan penambahan lantai. Pengguna bisa menambahkan lantai yang diperlukan pada formulir yang telah disediakan. Pada Tabel 4.11 menjelaskan alur kegiatan aktor menambah lantai.

Tabel 4.11 Spesifikasi Use Case Mengelola Lantai

Mengelola Lantai (SIPRPS-IDF-04)	
Brief Description	Fungsi pada use case specification ini berguna untuk mengelola data lantai pada suatu gedung <i>server</i> agar mempermudah dalam melakukan pencarian dan pemasangan perangkat



Actor	Petugas server
Pre-condition	<ul style="list-style-type: none"> - Petugas server berhasil masuk kedalam sistem - Perangkat yang mengakses sistem dapat terhubung dengan internet - Server terhubung dengan sistem
	<ul style="list-style-type: none"> - Petugas server berhasil menyimpan data perangkat - Petugas server dapat melihat informasi perangkat
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika petugas server menggunakan fungsi tambah lantai 2. Sistem menampilkan formulir tambah lantai 3. Petugas server mengisi formulir tambah lantai untuk ruangan server yang masih baru 4. Petugas server memilih menyimpan data tambah lantai <p>{menyimpan data tambah lantai}</p> <ol style="list-style-type: none"> 5. Sistem menyimpan data tambah lantai 6. Menampilkan pesan berhasil disimpan 7. Use case selesai
Alternative Flow	<p>A1. Pencarian Lantai</p> <ol style="list-style-type: none"> 1. Petugas server menggunakan fungsi cari 2. Sistem menampilkan halaman informasi rak dan perangkat server pada semua lantai 3. Petugas server memilih lantai tertentu yang dicari 4. Sistem menampilkan informasi rak dan perangkat server pada lantai yang dicari 5. Use case selesai <p>A2. Memilih lantai yang dicari</p> <ol style="list-style-type: none"> 1. Petugas Server menggunakan fungsi memilih lantai 2. Sistem menampilkan informasi lantai yang telah tersedia <p>{memuat data informasi rak dan perangkat}</p> <ol style="list-style-type: none"> 3. Petugas Server memilih salah satu lantai yang akan dicari



	<p>4. Sistem menampilkan informasi rak dan perangkat yang tersedia pada lantai yang dicari</p> <p>5. <i>Use case</i> selesai</p>
--	--

4.3.3.5 Spesifikasi *Use Case* Mengelola User

Penjelasan mengenai spesifikasi *use case* mengelola user menggambarkan alur pengguna melakukan penambahan pengguna yang akan dibutuhkan. Admin bisa menambahkan pengguna lain dengan cara mengisi yang formulir yang telah disediakan. Pada Tabel 4.12 menjelaskan alur kegiatan aktor menambah pengguna.

Tabel 4.12 Spesifikasi *Use Case* Mengelola User

Mengelola User (SIPRPS-IDF-05)	
Brief Description	Fungsi pada use case ini berguna untuk mengelola user
Actor	Petugas server
Pre-condition	<ul style="list-style-type: none"> - Petugas server berhasil masuk kedalam sistem - Perangkat yang mengakses sistem dapat terhubung dengan internet - Server terhubung dengan sistem
Post-condition	<ul style="list-style-type: none"> - Petugas server berhasil menyimpan data perangkat - Petugas server dapat melihat informasi perangkat
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika petugas server menggunakan fitur kelola user 2. Sistem menampilkan halaman identitas user yang sudah terdaftar 3. Petugas server menggunakan fungsi tambah user 4. Sistem menampilkan formulir untuk pengisian identitas pengguna yang akan ditambahkan 5. Petugas server mengisi form identitas pengguna <p>{menyimpan data identitas pengguna}</p> <ol style="list-style-type: none"> 6. Sistem menyimpan formulir identitas pengguna 7. Menampilkan pesan berhasil disimpan 8. Use case selesai
Alternative Flow	A1. Menghapus user



	<ol style="list-style-type: none"> 1. Petugas <i>server</i> menggunakan fungsi kelola <i>user</i> 2. Sistem menampilkan halaman identitas user yang sudah terdaftar {penghapusan data identitas pengguna} 3. Petugas <i>server</i> menggunakan memilih hapus {menampilkan informasi identitas pengguna} 4. Sistem menampilkan informasi pengguna yang masih terdaftar 5. <i>Use case</i> selesai <p>A2. Mengubah pengguna</p> <ol style="list-style-type: none"> 1. Petugas <i>server</i> memilih fungsi kelola <i>user</i> {memuat data yang akan diperbarui} 2. Sistem menampilkan halaman identitas <i>user</i> 3. Petugas <i>server</i> menggunakan fungsi <i>update</i> 4. Sistem menampilkan formulir identitas <i>user</i> {mengubah data identitas pengguna} 5. Petugas <i>server</i> mengisi formulir identitas pengguna 6. Petugas <i>server</i> memilih menyimpan formulir identitas pengguna 7. Sistem menyimpan form identitas dalam database dan menampilkan halaman kelola user 8. Sistem menampilkan pesan data identitas pengguna telah diperbarui 9. <i>Use case</i> selesai
--	---

4.3.3.6 Spesifikasi *Use Case* Memantau Rak dan Perangkat

Penjelasan mengenai spesifikasi *use case* memantau rak dan perangkat menggambarkan alur kepala teknik backbound memantau informasi dari rak dan perangkat didalam sistem. Kepala teknik backbound bisa melihat semua informasi mengenai rak dan perangkat dan melakukan pencarian rak dan perangkat yang akan dibutuhkan. Pada Tabel 4.13 menjelaskan alur kegiatan kepala teknik backbound memantau rak dan perangkat.

Tabel 4.13 Spesifikasi *Use Case* Memantau Rak dan Perangkat

Memantau Rak dan Perangkat (SIPRPS-IDF-06)	
Brief Description	Fungsi pada use case ini berguna untuk melihat informasi mengenai rak dan perangkat pada suatu ruangan <i>server</i>
Actor	Kepala Teknik Backbound
Pre-condition	<ul style="list-style-type: none"> - Kepala Teknik Backbound berhasil masuk kedalam sistem - Perangkat yang mengakses sistem dapat terhubung dengan internet <p><i>Server</i> terhubung dengan sistem</p>
Post-condition	<ul style="list-style-type: none"> - Kepala Teknik Backbound berhasil memantau rak dan perangkat <i>server</i>
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Kepala teknik backbound menggunakan fungsi daftar rak 2. Sistem menampilkan halaman informasi mengenai rak yang tersedia 3. Kepala teknik backbound menggunakan fungsi lihat {memuat data perangkat server} 4. Sistem menampilkan halaman informasi perangkat <i>server</i> 5. Kepala teknik backbound menggunakan fungsi lihat {memuat data detail perangkat} 6. Sistem menampilkan halaman informasi detail perangkat beserta gambar Sketchup 7. Kepala teknik backbound melihat informasi detail perangkat 8. <i>Use case</i> selesai
Alternatif Flow	<p>A1. Pencarian Rak</p> <ol style="list-style-type: none"> 1. Kepala teknik backbound menggunakan fungsi cari rak 2. Kepala teknik backbound memasukkan nama rak {memuat data rak} 3. Sistem menampilkan informasi rak yang dibutuhkan 4. <i>Use case</i> selesai

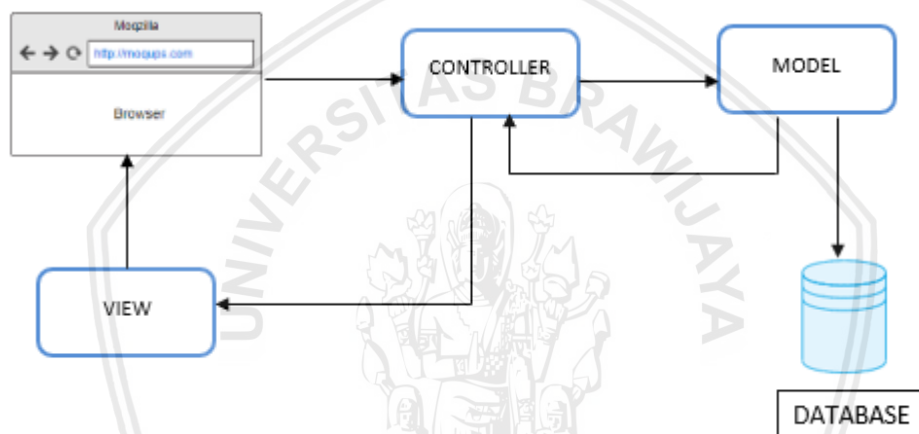
	<p>A2. Pencarian Perangkat</p> <ol style="list-style-type: none"> 1. Kepala teknik backbound menggunakan fungsi cari perangkat 2. Kepala teknik backbound memasukkan nama perangkat yang dibutuhkan {memuat data perangkat} 3. Sistem menampilkan informasi perangkat yang dibutuhkan 4. Use case selesai
--	---



BAB 5 PERANCANGAN

5.1 Perancangan Arsitektur

Perancangan arsitektur sistem informasi pendataan rak dan perangkat *server* dibangun dengan implementasi menggunakan framework *CodeIgniter* sebagai dasar pengembangan yang mengambil konsep pola *MVC* sebagai arsitektur dasar dari perangkat lunak. Alur dari arsitektur sistem informasi pendataan rak dan perangkat dimulai dari pengguna mengakses *web browser* menuju *Controller*, *Controller* memanggil *Model*, *Model* mengambil data yang telah disimpan pada database kemudian mengembalikan data menuju *Controller*. *Controller* mengirimkan data menuju *View*. *View* menampilkan halaman yang diminta oleh pengguna.



Gambar 5.1 Desain Arsitektur Sistem Informasi Pendataan Rak dan Perangkat

5.2 Pemodelan Interaksi Objek

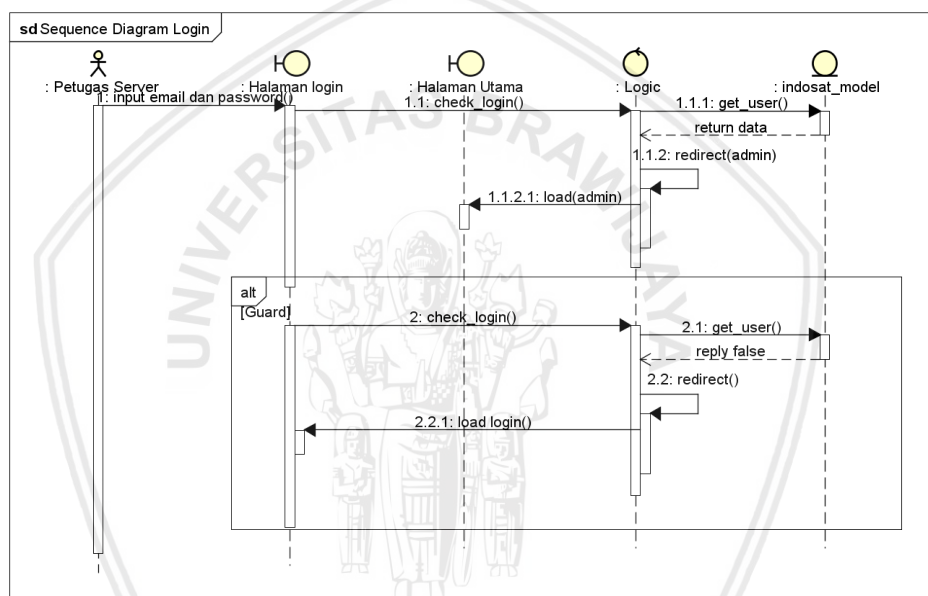
Perancangan interaksi dan kelas ini merupakan kumpulan diagram yang sudah mendekati dengan bahasa pemrograman agar memudahkan implementasi. Pada sistem informasi ini terbagi menjadi 2 jenis pembuatan diagram yaitu (1) Membuat *sequence diagram* untuk mengetahui alur dan interaksi antar *class* dan dilengkapi dengan kejelasan mengenai *Model*, *View*, maupun *Controller* dari setiap *Use Case* pada *Use Case Diagram*; (2) Membuat *class diagram* untuk mengetahui relasi antar *class* yang terdiri dari *class* pada *controller* dan *class* pada *model* yang saling berinteraksi satu sama lain. Dengan adanya dua jenis diagram tersebut akan menjelaskan interaksi yang dilakukan antar *class* dan fungsi yang ada pada setiap *class*nya.

5.2.1 Sequence Diagram Login

Sequence diagram pada merupakan gambaran dari interaksi antar objek ketika aktor petugas *server* menggunakan sistem informasi pendataan rak dan perangkat *server*. Objek yang terlibat pada interaksi untuk login adalah petugas

server sebagai aktor, halaman login, halaman utama, sebagai objek *boundary*, *control* sebagai objek *control* dan *entity* sebagai objek model.

Pengguna berinteraksi dengan sistem dimulai ketika aktor petugas server ingin masuk ke dalam sistem. Petugas server harus mengisi data email dan password pada halaman login. Selanjutnya halaman login yang sebagai *boundary* memanggil fungsi *check_login* pada halaman utama dan meneruskan ke *controller*. Kemudian *controller* meminta fungsi *get_user* dari database melalui *model*. Model akan mengembalikan data ke *controller* ketika data yang dimasukkan benar. Jika data yang dimasukkan salah maka *controller* akan mengembalikan tampilan berupa halaman login lagi. *Sequence diagram* login dapat dilihat pada Gambar 5.2



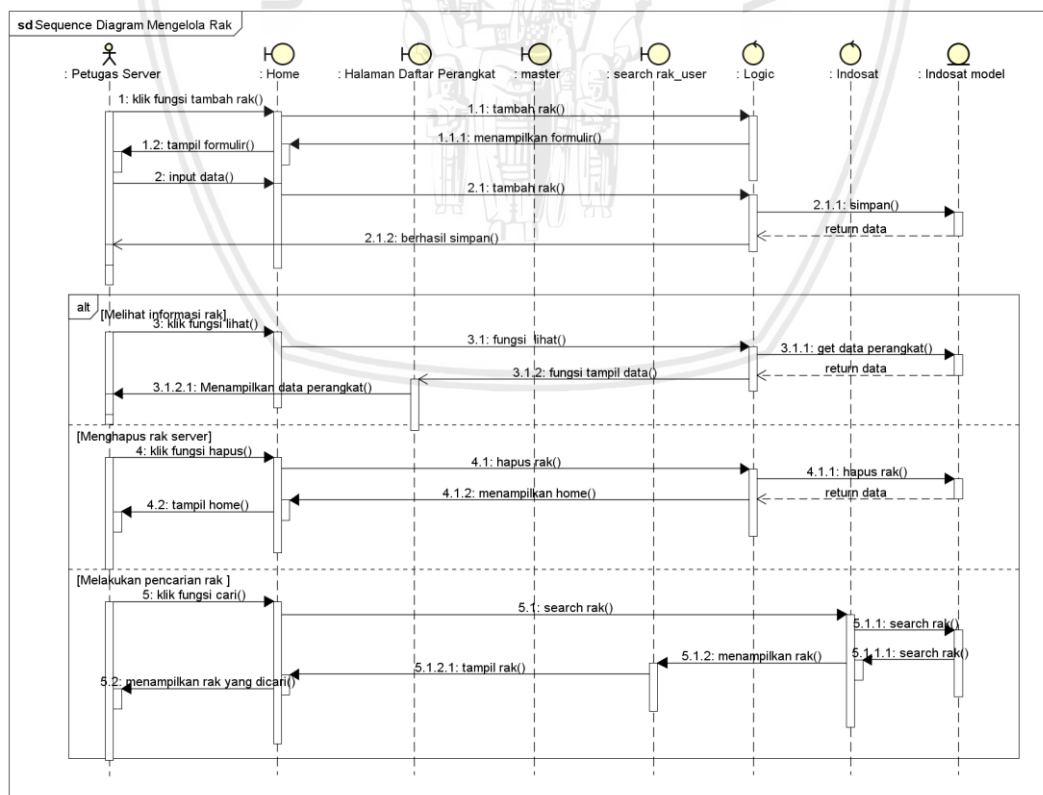
Gambar 5.2 Sequence Diagram Login

5.2.2 Sequence Diagram Mengelola Rak

Sequence diagram pada merupakan gambaran dari interaksi antar objek ketika aktor petugas server menggunakan sistem informasi pendataan rak dan perangkat server. Objek yang terlibat pada interaksi untuk mengelola rak adalah petugas server sebagai aktor, home, halaman daftar perangkat, master, search rak_user sebagai objek *boundary*, *logic*, indosat sebagai objek *control* dan indosat model sebagai objek model.

Pengguna berinteraksi dengan sistem dimulai ketika aktor petugas server telah masuk kedalam sistem. Selanjutnya sistem menampilkan halaman utama daftar rak yang terdapat 4 fungsi utama yaitu menambah rak, menghapus rak, mencari rak dan melihat rak. Jika petugas server ingin melakukan tambah rak maka petugas server harus memilih fungsi tambah rak pada halaman daftar rak melalui *boundary*. Kemudian *boundary* memanggil fungsi tambah rak pada *controller* *logic*. Selanjutnya *controller* mengembalikan view berupa menampilkan formulir

tambah rak dan petugas *server* harus mengisi data formulir tersebut. Data formulir tersebut dikirimkan ke *controller logic* dan meneruskan sampai ke model *indosat_model* dan data tersebut disimpan pada database. Selanjutnya model *indosat_model* akan mengembalikan data melalui *controller logic* dan diteruskan hingga *boundary* berupa tampilan pesan berhasil ditambahkan. Jika petugas *server* memilih melihat rak *server* maka petugas *server* memilih lihat pada *boundary home*. Selanjutnya *boundary home* memanggil fungsi lihat pada *controller logic*. Selanjutnya *controller* meminta data pada model dengan meminta fungsi *get_data_perangkat* pada model *indosat_model*. Model *indosat_model* mengembalikan data tersebut melali *controller logic* dan meneruskan hingga *boundary* dengan menampilkan data perangkat. Jika petugas *server* memilih menghapus rak maka petugas *server* memilih hapus pada *boundary home*. Kemudian memanggil fungsi hapus rak pada *controller logic* dan meneruskan dengan meminta pada model *indosat_model*. Model *indosat_model* mengembalikan data melalui *controller* dan meneruskan hingga *boundary* berupa tampilan halaman utama daftar rak yang terbaru. Jika petugas *server* memilih mencari rak maka petugas *server* memilih mencari rak pada *boundary home*. Selanjutnya *boundary home* memanggil fungsi *search_rak* pada *controller indosat* dan meneruskan meminta fungsi *search rak* pada model *indosat_model*. Model *indosat_model* mengembalikan data ke *controller* dan meneruskan hingga *boundary* dengan menampilkan informasi rak yang dicari. *Sequence diagram* mengelola rak dapat dilihat pada Gambar 5.3

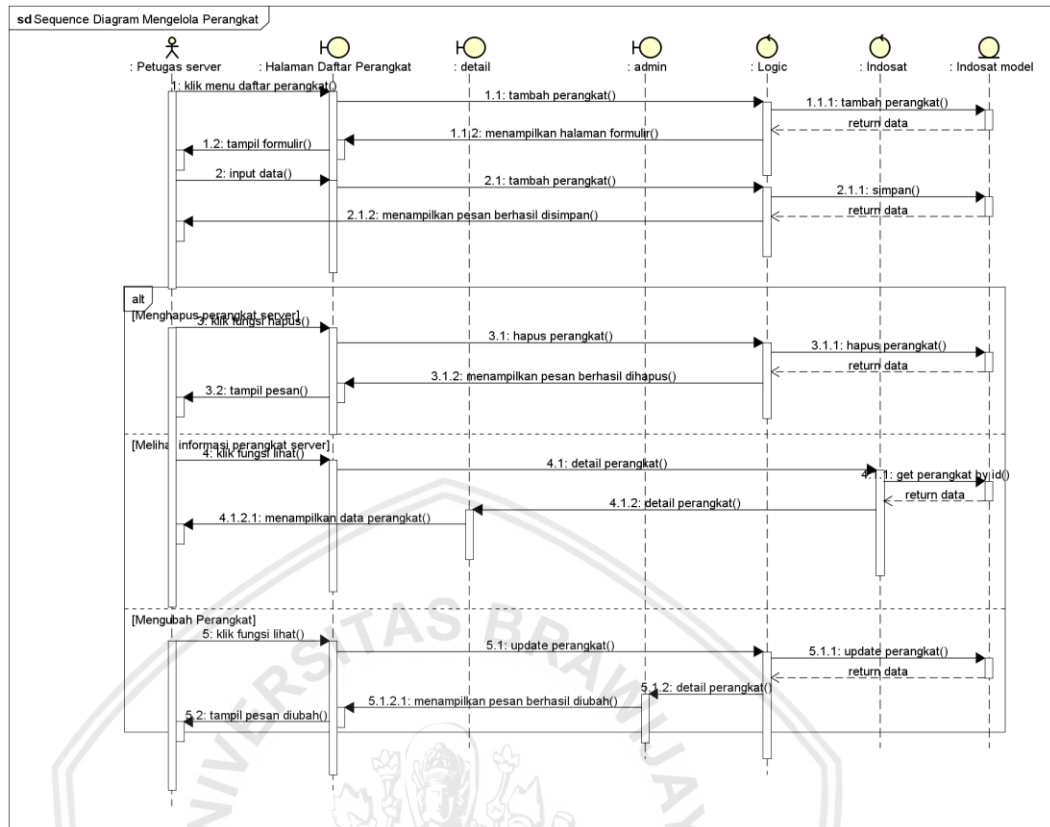


Gambar 5.3 Sequence Diagram Mengelola Rak Server

5.2.3 Sequence Diagram Mengelola Perangkat

Sequence diagram pada merupakan gambaran dari interaksi antar objek ketika aktor petugas *server* menggunakan sistem informasi pendataan rak dan perangkat *server*. Objek yang terlibat pada interaksi untuk mengelola perangkat adalah petugas *server* sebagai aktor, halaman daftar perangkat, detail, admin sebagai objek *boundary*, logic, indosat sebagai objek *control* dan indosat model sebagai objek model.

Pengguna berinteraksi dengan sistem dimulai ketika aktor petugas *server* telah masuk kedalam sistem. Selanjutnya sistem menampilkan halaman utama daftar rak yang terdapat 4 fungsi utama yaitu menambah perangkat, menghapus rak, mencari perangkat dan melihat perangkat. Kemudian *boundary* memanggil fungsi tambah perangkat pada *controller logic*. Selanjutnya *controller* mengembalikan view berupa menampilkan formulir tambah perangkat dan petugas *server* harus mengisi data formulir tersebut. Data formulir tersebut dikirimkan kembali ke *controller logic* dan meneruskan sampai ke *model indosat_model* dan data tersebut disimpan pada database. Selanjutnya *model indosat_model* akan mengembalikan data melalui *controller logic* dan diteruskan hingga *boundary* berupa tampilan pesan berhasil ditambahkan. Jika petugas *server* memilih melihat perangkat *server* maka petugas *server* memilih lihat pada *boundary* daftar perangkat. Selanjutnya *boundary* daftar perangkat memanggil fungsi detail perangkat pada *controller indosat*. Selanjutnya *model indosat_model* mengembalikan data tersebut melalui *controller indosat* dan meneruskan hingga *boundary* dengan menampilkan informasi detail perangkat. Jika petugas *server* memilih menghapus perangkat maka petugas *server* memilih hapus pada *boundary* daftar perangkat. Kemudian memanggil fungsi hapus perangkat pada *controller logic* dan meneruskan dengan meminta pada *model indosat_model*. *Model indosat_model* mengembalikan data melalui *controller* dan meneruskan hingga *boundary* berupa tampilan halaman daftar perangkat yang terbaru. Jika petugas *server* memilih mengubah data perangkat *server* maka petugas memilih lihat pada *boundary* halaman daftar perangkat kemudian memanggil fungsi update perangkat pada *controller logic*. Selanjutnya *controller logic* meminta fungsi update perangkat pada *model indosat_model*. *Model indosat_model* mengembalikan data pada *controller* dan *controller* meneruskan data dengan menampilkan pesan berhasil diubah pada pengguna. *Sequence diagram* mengelola perangkat dapat dilihat pada Gambar 5.4



Gambar 5.4 Sequence Diagram Mengelola Perangkat

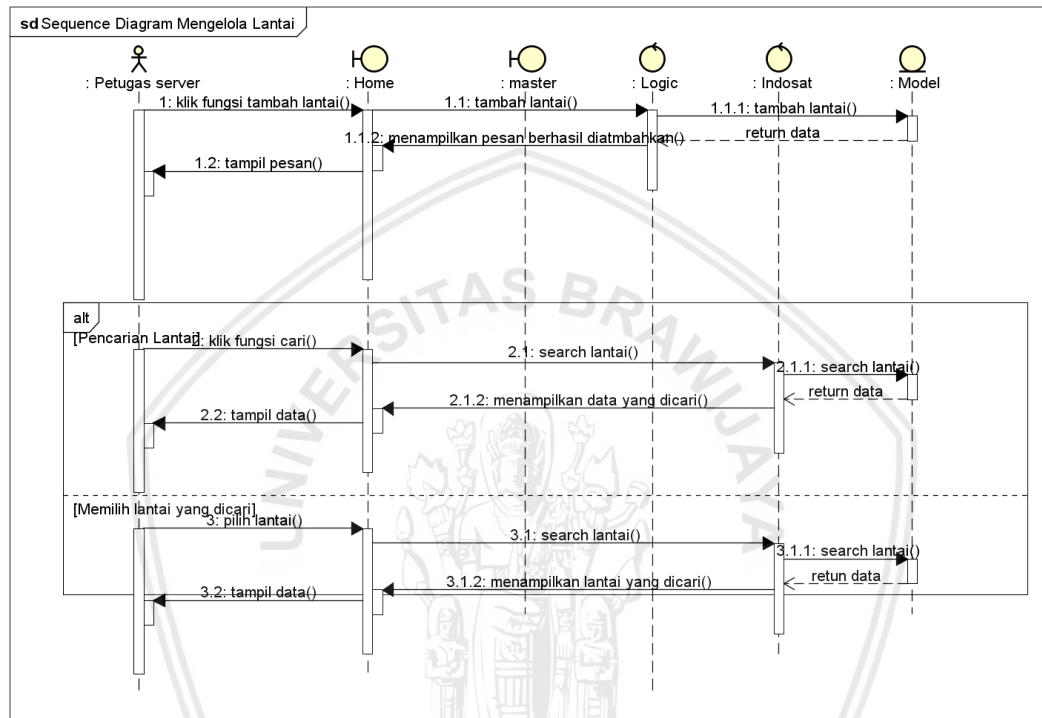
5.2.4 Sequence Diagram Mengelola Lantai

Sequence diagram pada merupakan gambaran dari interaksi antar objek ketika aktor petugas server menggunakan sistem informasi pendataan rak dan perangkat server. Objek yang terlibat pada interaksi untuk mengelola lantai adalah petugas server sebagai aktor, home, master sebagai objek boundary, logic sebagai objek control dan indosat model sebagai objek model.

Pengguna berinteraksi dengan sistem dimulai ketika aktor petugas server telah masuk kedalam sistem. Selanjutnya sistem menampilkan halaman mengelola lantai yang terdapat 3 fungsi utama yaitu menambah lantai gedung, mencari lantai gedung, mencari lantai gedung dan memilih lantai gedung. Jika petugas server ingin melakukan menambah lantai maka petugas server harus memilih fungsi tambah rak pada halaman daftar rak melalui boundary. Selanjutnya boundary home memanggil fungsi tambah lantai pada controller logic. Controller logic meminta fungsi tambah lantai ke model indosat model. Kemudian model indosat model mengembalikan data ke controller logic dan meneruskan hingga boundary home dengan menampilkan pesan berhasil ditambahkan. Jika petugas server akan melakukan pencarian lantai gedung maka petugas server harus memilih fungsi cari pada boundary home. Kemudian boundary home memanggil fungsi search lantai pada controller indosat. Kemudian controller indosat meminta fungsi search lantai pada model indosat model. Model indosat model selanjutnya mengembalikan data pada controller logic dan meneruskan ke boundary home



dengan menampilkan informasi lantai yang dicari. Jika petugas *server* akan melakukan pencarian lantai gedung maka petugas *server* harus memilih fungsi cari pada boundary home. Kemudian boundary home memanggil fungsi search lantai pada controller indosat. Kemudian controller indosat meminta fungsi search lantai pada model indosat model. Model indosat model selanjutnya mengembalikan data pada controller logic dan meneruskan ke boundary home dengan menampilkan informasi lantai yang dicari. *Sequence diagram* mengelola lantai dapat dilihat pada Gambar 5.5



Gambar 5.5 *Sequence Diagram* Mengelola Lantai

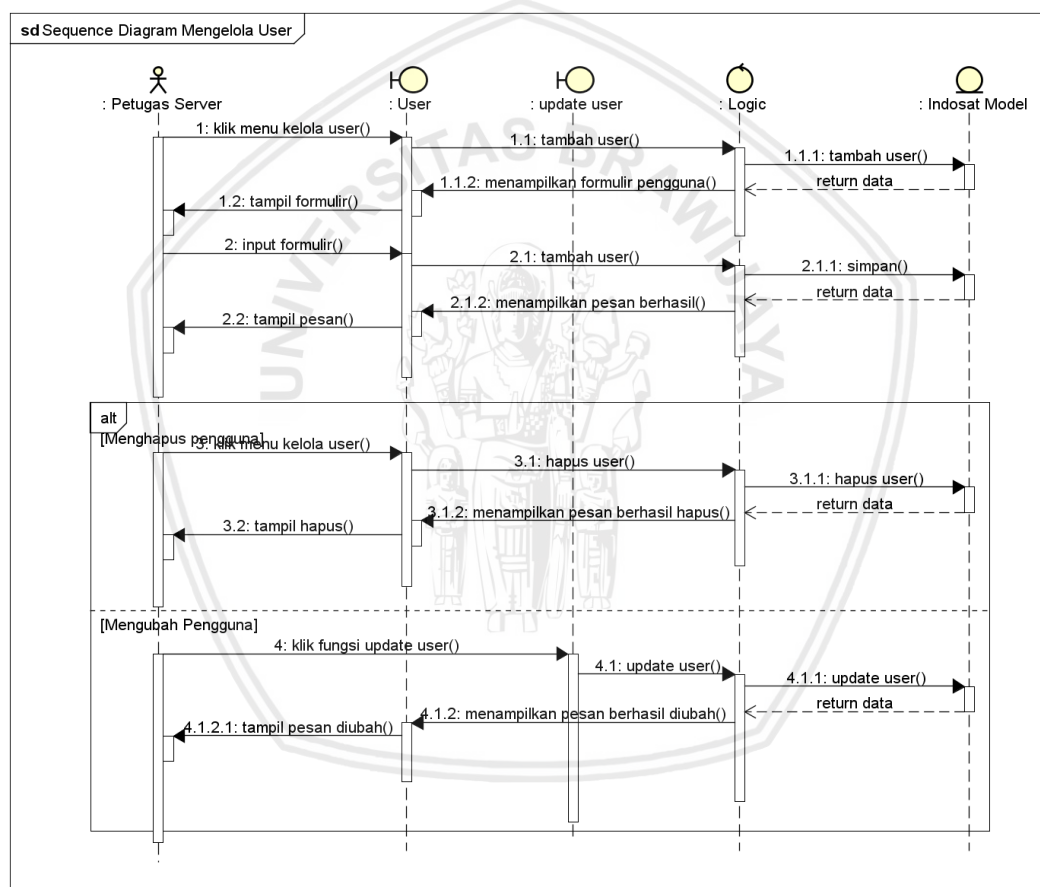
5.2.5 Sequence Diagram Mengelola User

Sequence diagram pada merupakan gambaran dari interaksi antar objek ketika aktor petugas *server* menggunakan sistem informasi pendataan rak dan perangkat *server*. Objek yang terlibat pada interaksi untuk mengelola rak adalah petugas *server* sebagai aktor, user, update user sebagai objek *boundary*, logic sebagai objek *control* dan indosat *model* sebagai objek *model*.

Pengguna berinteraksi dengan sistem dimulai ketika aktor petugas *server* telah masuk kedalam sistem. Selanjutnya sistem menampilkan halaman utama daftar rak yang terdapat 2 fungsi utama yaitu menambah dan menghapus user. Jika petugas *server* ingin melakukan tambah user maka harus memilih menu kelola user. Kemudian boundary user memanggil fungsi tambah user pada controller logi. Selanjutnya controller logic meminta fungsi tambah user dan mengembalikan data ke controller logic dan meneruskan hingga boundary user dengan menampilkan formulir pengguna. Petugas *server* mengisi formulir tersebut dan mengirimkan ke controller logic dan menyimpan data tersebut pada model indosat model. Setelah



disimpan pada indosat model maka model akan mengembalikan data melalui controller dan meneruskan hingga boundary dengan menampilkan pesan berhasil disimpan. Jika petugas *server* memilih menghapus pengguna maka melakukan klik menu kelola user kemudian memanggil fungsi hapus user pada controller logic. Selanjutnya controller logic meminta fungsi hapus user pada model indosat model. Model indosat model mengembalikan data ke controller logic dan meneruskan hingga boundary home dengan menampilkan pesan berhasil dihapus. Jika petugas *server* memilih mengubah data pengguna maka memilih fungsi update pada boundary update user. Kemudian memanggil fungsi update user pada controller logic. Selanjutnya controller logic meminta fungsi update user pada model indosat model. Model indosat model mengembalikan data ke controller dan meneruskan ke boundary dengan menampilkan pesan berhasil diubah

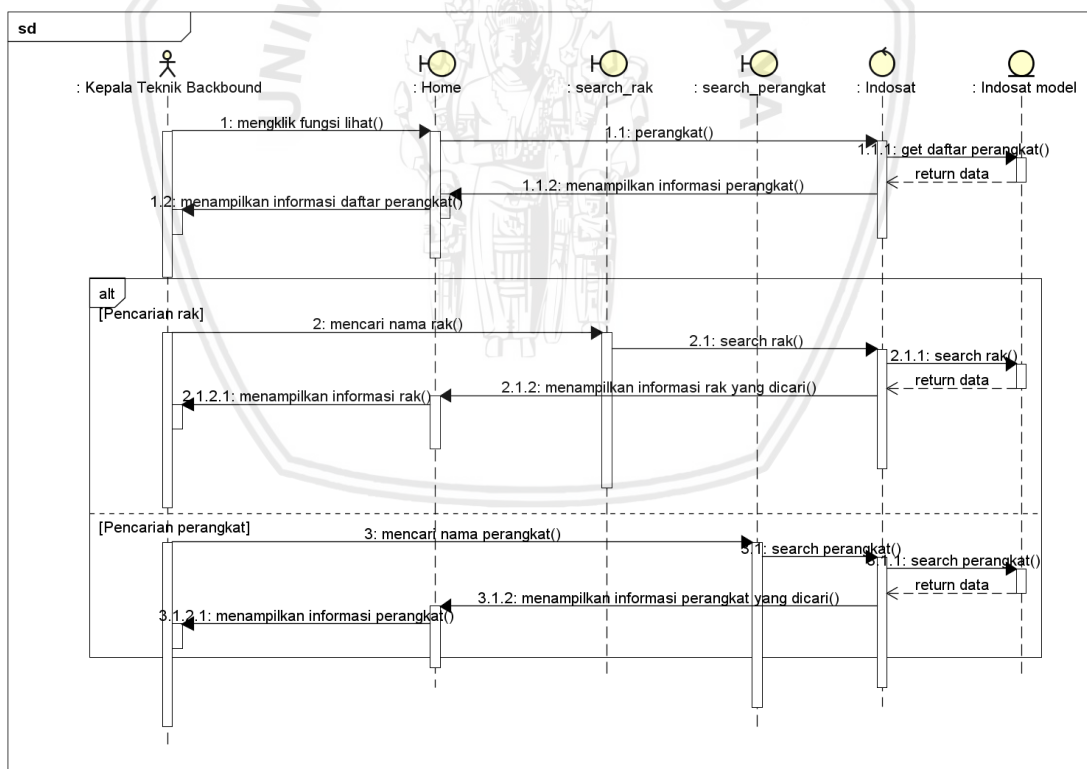


Gambar 5.6 Sequence Diagram Mengelola User

5.2.6 Sequence Diagram Memantau Rak dan Perangkat Server

Sequence diagram pada merupakan gambaran dari interaksi antar objek ketika aktor petugas server menggunakan sistem informasi pendataan rak dan perangkat server. Objek yang terlibat pada interaksi untuk memantau rak dan perangkat server adalah kepala teknik backbound sebagai aktor, search rak, serach perangkat sebagai objek *boundary*, indosat sebagai objek *control* dan indosat model sebagai objek *model*.

Pengguna berinteraksi dengan sistem dimulai ketika aktor kepala teknik bakbound telah masuk kedalam sistem. Selanjutnya sistem menampilkan halaman utama daftar rak yang terdapat 3 fungsi utama yaitu melihat informasi rak dan perangkat *server*, melakukan pencarian berdasarkan nama rak dan nama perangkat. Jika kepala teknik memilih melihat rak maka memilih lihat pada boundary halaman utama. Kemudian boundary memanggil fungsi perangkat pada controller indosat. Selanjutnya controller indosat meminta fungsi get daftar perangkat pada model indosat model. Model indosat model mengembalikan data ke controller dan meneruskan hingga boundary dengan menampilkan informasi perangkat. Jika kepala teknik memilih mencari rak maka mengisi nama rak yang akan dicari pada boundary search rak. Kemudian boundary search rak memanggil fungsi search rak pada model indosat model. Model indosat model mengembalikan data ke controller dan meneruskan hingga boundary home dengan menampilkan informasi rak yang dicari. Jika kepala teknik memilih mencari perangkat maka mengisi nama perangkat yang akan dicari pada boundary search perangkat. Kemudian boundary search perangkat memanggil fungsi search perangkat pada model indosat model. Model indosat model mengembalikan data ke controller dan meneruskan hingga boundary home dengan menampilkan informasi perangkat yang dicari.



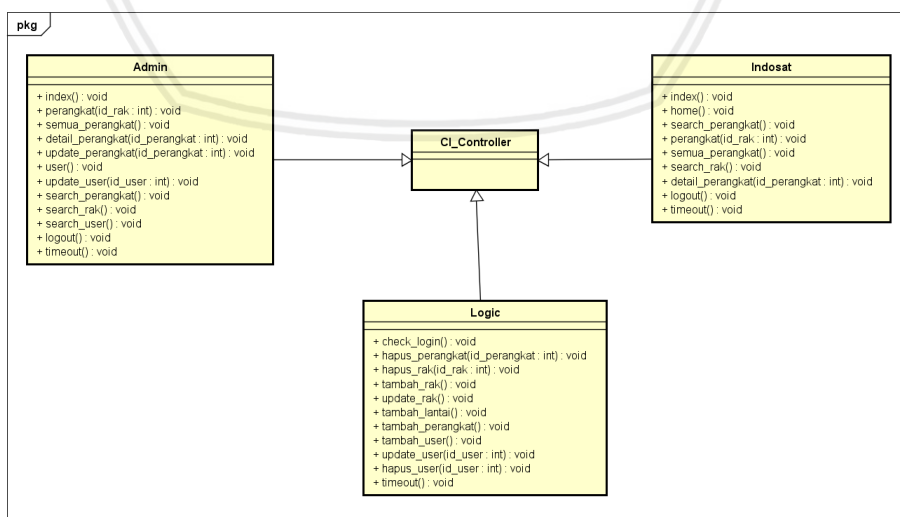
Gambar 5.7 Sequence Diagram Memantau Rak dan Perangkat

5.3 Pemodelan Class

Pada sub bab sebelumnya telah dijelaskan mengenai hubungan antara perilaku terhadap *use case* dengan menggunakan *sequence diagram*. Pada tahap selanjutnya adalah pembuatan *pemodelan class*. Diagram kelas merupakan hasil analisis yang telah dibuat. *Pemodelan class* menggambarkan 3 *controller* dan 1 *model*. Pada Gambar 5.7 menjelaskan diagram kelas *controller* pada sistem. Pada kelas diagram kelas *controller* menyediakan fungsi logika yang mempresentasikan kebutuhan sistem untuk memenuhi kebutuhan pengguna. Pada kelas Admin *controller* merepresentasikan fungsi logika untuk sisi Admin. Pada kelas Indosat *controller* yang mempresentasikan fungsi logika untuk sisi klien dan kelas Logic *controller* mempresentasikan fungsi logika pada sistem seperti menyimpan data, menghapus data.

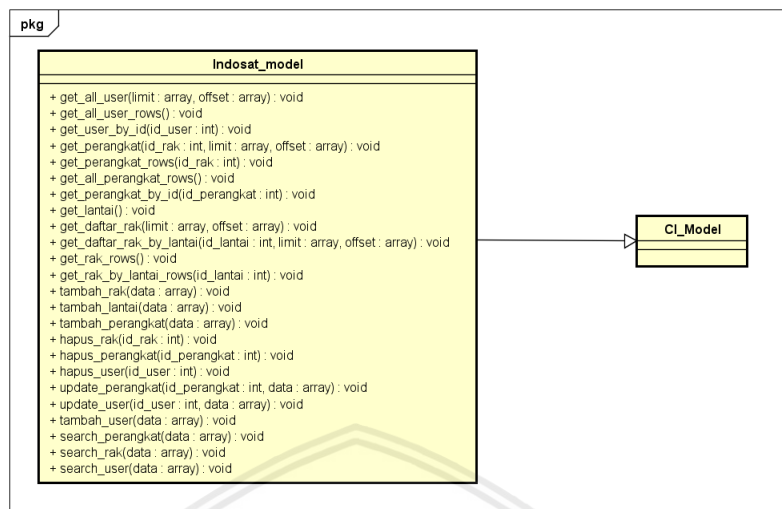
Pada diagram kelas *model* menyediakan fungsi untuk mengakses data pada database. Pada Gambar 5.8 menjelaskan fungsi apa saja yang terdapat pada diagram kelas *model*. Fungsi untuk *login* dapat diakses dengan menggunakan fungsi *get_user*. Fungsi untuk mengelola user dapat diakses dengan menggunakan fungsi *get_all_user*, *get_all_user_rows*, *get_user_by_id*, *update_user*, *tambah user*, *searh_user*. Fungsi untuk mengelola perangkat dapat diakses dengan menggunakan fungsi *tambah perangkat*, *get_perangkat*, *get_semua_perangkat*, *get_perangkat_rows*, *get_all_perangkat*, *get_perangkat_by_id*, *hapus perangkat*, *update_perangkat*, *search_perangkat*. Fungsi untuk mengelola lantai dapat diakses dengan menggunakan fungsi *tambah lantai*, *get_lantai*. Fungsi untuk mengelola rak dapat diakses dengan menggunakan fungsi *tambah rak*, *get_rak*, *get_daftar_rak*, *get_daftar_rak_by_lantai*, *get_rak_rows*, *get_rak_by_lantai_rows*, *hapus rak*, *search_rak*.

Selanjutnya, pada Gambar 5.9 menjelaskan hubungan antara kelas yang ada pada *Controller* dengan kelas *Model*.

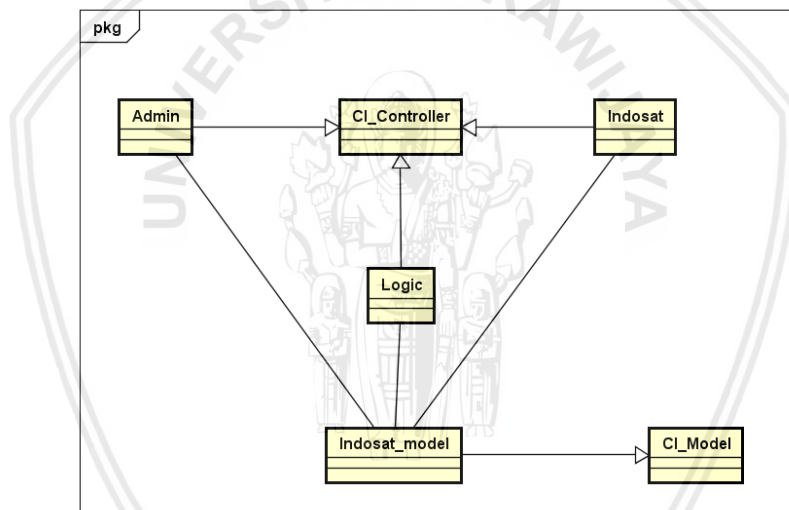


Gambar 5.8 Diagram Kelas Perancangan *Controller*





Gambar 5.9 Class Diagram untuk Domain Model

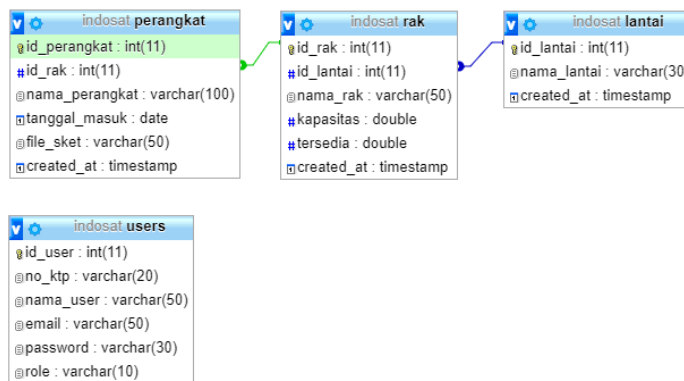


Gambar 5.10 Diagram Kelas Relasi Controller dengan Model

5.4 Perancangan Basis Data

Basis data merupakan kumpulan informasi yang disimpan pada komputer dan dapat diperiksa menggunakan suatu program tertentu untuk memperoleh informasi dari basis data. Basis data pada penelitian ini menggunakan MySQL. Perancangan basis data dibuat untuk menentukan proses isi dan pengaturan data yang dibutuhkan sistem. Pada Gambar 5.7 menunjukkan perancangan basis data.





Gambar 5.11 Physical Data Model Sistem Informasi Rak dan Perangkat Server

5.4.1 Tabel Indosat Perangkat

Tabel indosat perangkat digunakan untuk menyimpan data dari perangkat. Pada Tabel 5.1 akan menjelaskan mengenai struktur tabel Indosat perangkat.

Tabel 5.1 Tabel Indosat Perangkat

No	Nama	Tipe	Panjang	Keterangan
1	id_perangkat	Int	11	id_perangkat sebagai <i>primary key</i>
2	id_rak	Int	11	id rak dari perangkat
3	nama_perangkat	varchar	100	berisi nama perangkat
4	tanggal_masuk	Date		berisi tanggal ketika rak ditambahkan
5	file_sket	varchar	50	berisi <i>inputan file skethcup</i>
6	created_at	timestamp		menunjukkan tanggal, bulan dan tahun saat ini

5.4.2 Tabel Indosat Rak

Tabel indosat rak digunakan untuk menyimpan data dari rak. Pada Tabel 5.2 akan menjelaskan mengenai struktur tabel Indosat rak.

Tabel 5.2 Tabel Indosat Rak

No	Nama	Tipe	Panjang	Keterangan
1	id_rak	Int	11	id_rak sebagai <i>primary key</i>
2	id_lantai	Int	11	id lantai dari rak
3	nama_rak	varchar	50	berisi nama rak
4	kapasitas	double		berisi kapasitas dari rak
5	tersedia	double		berisi sisa kapasitas dari rak



6	created_at	timestamp		menunjukkan tanggal, bulan dan tahun saat ini
---	------------	-----------	--	---

5.4.3 Tabel Indosat Lantai

Tabel indosat lantai digunakan untuk menyimpan data dari lantai. Pada Tabel 5.3 akan menjelaskan mengenai struktur tabel Indosat lantai.

Tabel 5.3 Tabel Indosat Lantai

No	Nama	Tipe	Panjang	Keterangan
1	id_lantai	Int	11	id_lantai sebagai <i>primary key</i>
2	nama_lantai	varchar	30	berisi nama lantai
3	created_at	timestamp		menunjukkan tanggal, bulan dan tahun saat ini

5.4.4 Tabel Indosat Users

Tabel indosat *users* digunakan untuk menyimpan data dari pengguna sistem informasi rak dan perangkat *server*. Pada Tabel 5.4 akan menjelaskan mengenai struktur tabel Indosat perangkat.

Tabel 5.4 Tabel Indosat Users

No	Nama	Tipe	Panjang	Keterangan
1	id_user	Int	11	id_user sebagai <i>primary key</i>
2	no_ktp	Varchar	20	No KTP dari pengguna
3	nama_user	Varchar	50	berisi nama pengguna
4	email	Varchar	50	berisi email pengguna
5	password	Varchar	30	berisi password pengguna
6	role	Varchar	10	menunjukkan peran pengguna

5.5 Perancangan Algoritme

Perancangan algoritme merupakan tahapan yang dilakukan untuk menyusun dan menentukan operasi logika sebagai konsep pembangunan fungsi-fungsi dari sistem informasi yang akan dikembangkan. Hasil perancangan algoritme dalam penelitian ini di dokumentasi ke dalam beberapa *pseudocode*. Algoritme yang telah didokumentasikan akan menjadi panduan untuk pembangunan sistem informasi dalam tahap implementasi.

5.5.1 Menambah Rak Server

Pada sub bab ini berisi rancangan algoritme mengenai fungsi yang disediakan oleh sistem untuk pengguna menggunakan sistem untuk mengelola rak. Pada Tabel 5.5 Merupakan pseudocode rancangan algoritme tambah rak.

Tabel 5.5 Tabel Pseudocode Menambah Rak Server

No	<i>Pseudocode</i>
1.	START
2.	Menggunakan session waktu
3.	Mengirim data pada database
4.	Data yang akan dikirimkan pada database adalah id_lantai
5.	Data yang akan dikirimkan pada database adalah nama_rak
6.	Data yang akan dikirimkan pada database adalah kapasitas
7.	Jika model menyimpan fungsi tambah_rak
8.	Akan muncul pesan bahwa rak berhasil ditambahkan
9.	Kembali ke halaman utama
10.	jika tidak disimpan maka muncul pesan terjadi kesalahan
11.	Penutup

5.5.2 Menambah Perangkat

Pada sub bab ini berisi rancangan algoritme mengenai fungsi yang disediakan oleh sistem untuk pengguna menggunakan sistem untuk mengelola perangkat. Pada Tabel 5.6 Merupakan pseudocode rancangan algoritme menampilkan informasi perangkat.

Tabel 5.6 Menampilkan Informasi Perangkat

No	<i>Pseudocode</i>
1.	START
2.	Menggunakan session waktu
3.	Membuat fuction semua perangkat
4.	Menggunakan session waktu
5.	Memanggil controller semua perangkat pada controller Admin
6.	Memanggil fungsi get_all_perangkat_rows model indosat_model
7.	Jika ingin menampilkan informasi perangkat maka controller memanggil model dengan fungsi get_semua_perangkat
8.	Jika ingin menampilkan informasi rak maka controller memanggil model dengan fungsi get_semua_rak
9.	Mengembalikan data
10.	Menampilkan judul Daftar Perangkat

11.	Menampilkan halaman Daftar Perangkat
12.	Penutup
13.	Mengembalikan tampilan pengguna
14.	Penutup

5.5.3 Mengelola User

Pada sub bab ini berisi rancangan algoritme mengenai fungsi yang disediakan oleh sistem untuk pengguna menggunakan sistem untuk mengelola perangkat. Pada Tabel 5.7 Merupakan pseudocode rancangan algoritme menampilkan informasi pengguna.

No	Pseudocode
1.	START
2.	Menggunakan session waktu
3.	Membuat fuction semua perangkat
4.	Menggunakan session waktu
5.	Memanggil controller semua perangkat pada controller Admin
6.	Memanggil fungsi get_all_users pada model indosat_model
7.	Jika ingin menampilkan informasi pengguna maka controller memanggil model dengan fungsi get_all_users
8.	Mengembalikan data
9.	Menampilkan judul Daftar User
10.	Menampilkan halaman Kelola User
11.	Penutup
12.	Mengembalikan tampilan pengguna
13.	Penutup

5.6 Perancangan Antarmuka

5.6.1 Antarmuka Login

Halaman *Login* merupakan halaman pertama yang akan dilihat oleh pengguna sistem karena untuk menggunakan fungsi lain selain *Login* harus melakukan tahap keamanan yaitu *Login* yang dapat dilakukan pada halaman *Login*. Pada halaman ini terdapat 3 fungsi utama yaitu (1) memasukkan username yang berfungsi untuk masukan identitas username yang telah terdaftar; (2) memasukkan password yang berguna untuk kata sandi sebagai keamanan; dan (3) tombol button *Login* yang berfungsi untuk mengecek username dan password. Antarmuka dari halaman *Login* dapat dilihat pada Gambar 5.12.

LOGIN

The login page features two input fields: 'Email' with an envelope icon and 'Password' with a lock icon. Below these fields is an orange 'LOGIN' button with a right-pointing arrow icon.

Gambar 5.12 Antarmuka Login

5.6.2 Antarmuka Mengelola Rak

Halaman daftar rak merupakan halaman kedua yang akan dilihat oleh petugas *server* setelah *login*. Pada halaman ini terdapat 7 fungsi utama yaitu (1) Aktor pada fungsi ini memberikan informasi mengenai data diri Aktor yang masuk; (2) Daftar Rak salah satu fungsi untuk memberikan informasi isi dari rak pada *server*; (3) Daftar Perangkat salah satu fungsi untuk memberi informasi (4) Kelola User pada fungsi ini berguna untuk menambahkan *user* yang ingin masuk pada web ini; (5) *Logout* pada fungsi ini berguna untuk keluar dari web ini; (6) Pilih lantai pada fungsi ini berguna untuk memudahkan Aktor mencari informasi *server* per lantai; (7) Cari rak pada fungsi ini berguna untuk memudahkan Aktor untuk mencari letak rak yang ingin diketahui dengan memasukkan inputan berupa nama rak. Antarmuka dari halaman utama dapat dilihat pada Gambar 5.13.

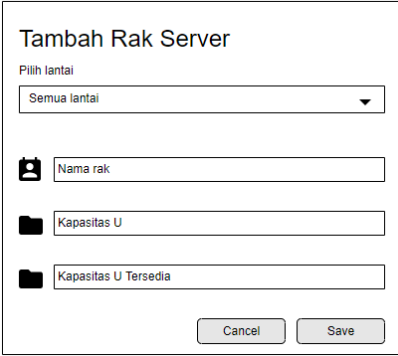
The 'DAFTAR RAK' page includes a navigation menu with 'Admin', 'Daftar Rak', 'Daftar Perangkat', 'Kelola User', and 'Logout'. Below the menu, there is a 'pilih lantai' dropdown menu, a 'Cari' button, and a 'Cari rak' button. A 'Lantai +' button is also present. The main content is a table with the following data:

Lantai	Nama rak	Kapasitas	Tersedia	Aksi
Lantai 1	20XYZ-INDOSAT-OORED00	40 U	7 U	Lihat Hapus

Gambar 5.13 Antarmuka Daftar Rak

Halaman tambah rak merupakan halaman sistem untuk menambahkan rak *server* yang baru dengan menginputkan informasi pada form yang tersedia. Pada halaman ini mempunyai 6 fungsi utama yaitu (1) Aktor melakukan pemilihan lantai; (2) Aktor mengisi nama rak pada form yang tersedia; (3) Aktor mengisi kapasitas U rak kosong yang tersedia; (4) Aktor mengisi kapasitas U yang tersedia pada rak; (5) Aktor menggunakan fungsi save untuk menyimpan informasi form

yang telah diisi (6) Aktor dapat membatalkan penambahan rak. Antarmuka halaman tambah rak dapat dilihat pada gambar 5.14.



Tambah Rak Server

Pilih lantai
Semua lantai

Nama rak

Kapasitas U

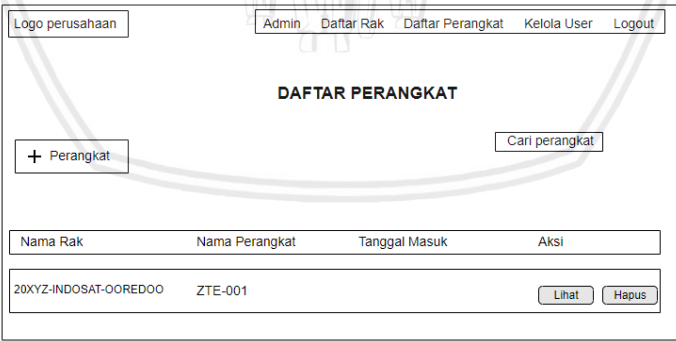
Kapasitas U Tersedia

Cancel Save

Gambar 5.14 Antarmuka Tambah Rak

5.6.3 Antarmuka Mengelola Perangkat

Halaman daftar perangkat merupakan halaman yang akan dilihat oleh petugas *server*. Pada halaman ini terdapat 7 fungsi utama yaitu (1) Aktor pada fungsi ini memberikan informasi mengenai data diri Aktor yang masuk; (2) Daftar Rak salah satu fungsi untuk memberikan informasi isi dari rak pada *server*; (3) Daftar Perangkat salah satu fungsi untuk memberi informasi (4) Kelola User pada fungsi ini berguna untuk menambahkan *user* yang ingin masuk pada web ini; (5) *Logout* pada fungsi ini berguna untuk keluar dari web ini; (6) Pilih lantai pada fungsi ini berguna untuk memudahkan Aktor mencari informasi *server* per lantai; (7) Cari rak pada fungsi ini berguna untuk memudahkan Aktor untuk mencari letak rak yang ingin diketahui dengan memasukkan inputan berupa nama rak. Antarmuka dari halaman utaman dapat dilihat pada Gambar 5.15.



Logo perusahaan Admin Daftar Rak Daftar Perangkat Kelola User Logout

DAFTAR PERANGKAT

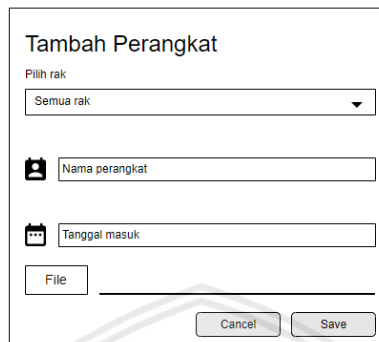
+ Perangkat Cari perangkat

Nama Rak	Nama Perangkat	Tanggal Masuk	Aksi
20XYZ-INDOSAT-OORED00	ZTE-001		Lihat Hapus

Gambar 5.15 Antarmuka Daftar Perangkat

Halaman tambah rak merupakan halaman sistem untuk menambahkan perangkat *server* yang baru dengan menginputkan informasi pada form yang tersedia. Pada halaman ini mempunyai 6 fungsi utama yaitu (1) Aktor melakukan pemilihan rak; (2) Aktor mengisi nama perangkat pada form yang tersedia; (3) Aktor mengisi tanggal masuk perangkat; (4) Aktor memilih file yang akan

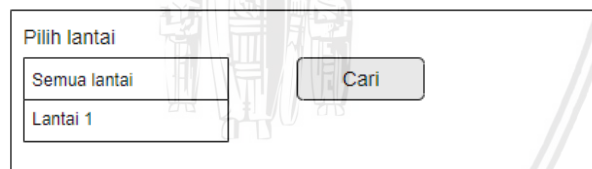
diinputkan dalam form. File ini embeded dari Sketchup; (5) Aktor menggunakan fungsi save untuk menyimpan informasi form yang telah diisi (6) Aktor dapat membatalkan penambahan rak. Antarmuka halaman tambah rak dapat dilihat pada gambar 5.16.



Gambar 5.16 Antarmuka Tambah Perangkat

5.6.4 Antarmuka Mengelola Lantai

Halaman cari lantai merupakan halaman sebuah droplist untuk memilih lantai yang telah diinputkan sebelumnya. Pada halaman ini mempunyai 1 fungsi utama yaitu melakukan pencarian. Pencarian ini berguna untuk memudahkan Aktor dalam melakukan pencarian lokasi lantai. Setiap lokasi lantai terdapat informasi rak dan perangkat yang tersedia. Antarmuka halaman cari lantai dapat dilihat pada Gambar 5.17.



Gambar 5.17 Antarmuka Cari Lantai

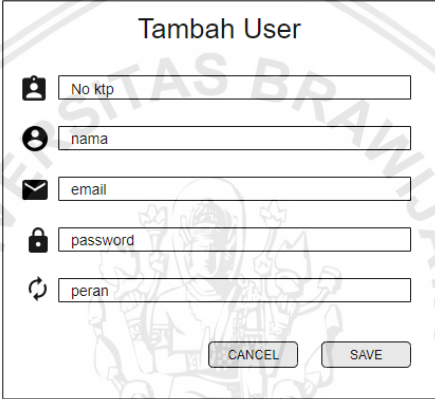
Halaman tambah lantai merupakan halaman sebuah inputan untuk menambahkan lantai ruangan *server*. Pada halaman ini mempunyai 2 fungsi utama yaitu (1) Aktor dapat menginputkan lantai baru kemudian menekan tombol save; (2) Aktor dapat membatalkan inputan lantai baru pada web. Antarmuka halaman tambah lantai dapat dilihat pada Gambar 5.18.



Gambar 5.18 Antarmuka Tambah Lantai

5.6.5 Antarmuka Mengelola User

Halaman tambah user merupakan halaman sistem untuk menambahkan perangkat *server* yang baru dengan menginputkan informasi pada form yang tersedia. Pada halaman ini mempunyai 6 fungsi utama yaitu (1) Aktor melakukan pemilihan rak; (2) Aktor mengisi nama perangkat pada form yang tersedia; (3) Aktor mengisi tanggal masuk perangkat; (4) Aktor memilih file yang akan diinputkan dalam form. File ini embeded dari Sketchup; (5) Aktor menggunakan fungsi save untuk menyimpan informasi form yang telah diisi (6) Aktor dapat membatalkan penambahan rak. Antarmuka halaman tambah rak dapat dilihat pada gambar 5.19



Tambah User

No ktp

nama

email

password

peran

CANCEL SAVE

Gambar 5.19 Antarmuka Tambah User

BAB 6 IMPLEMENTASI

6.1 Lingkungan Implentasi

Pada bab lingkungan implementasi menjelaskan tentang sistem informasi pendataan rak *server* berbasis web pada PT Indosat Ooredoo Surabaya yang digunakan pada lingkungan yang seperti apa, di mulai dari perangkat lunak dan perangkat keras yang digunakan oleh sistem.

6.1.1 Lingkungan Perangkat Keras

Pada bab lingkungan perangkat keras yang akan digunakan dalam sistem informasi pendataan rak sever pada PT Indosat Ooredoo Surabaya dapat dijelaskan sebagai berikut:

No	Nama Perangkat Keras
1.	<i>Processor</i> Intel Inside Core i5
2.	<i>RAM</i> 4 GB
3.	<i>Hardisk</i> 600 GB
4.	<i>Monitor</i> 14''
5.	<i>Display</i> AMD Radeoon R5

6.1.2 Lingkungan Perangkat Lunak

Pada lingkungan perangkat lunak yang akan digunakan dalam sistem informasi pendataan rak sever pada PT Indosat Ooredoo Surabaya dapat dijelaskan sebagai berikut:

No	Nama Perangkat Lunak	Deskripsi
1.	Sistem operasi Windows 10 Pro 64 bit	Sistem operasi yang dikembangkan oleh perusahaan Microsoft
2.	Sublime	Perangkat lunak yang digunakan untuk menulis, meng-compile, dan mencari kesalahan kode program
3.	XAMPP v3.2.2	Perangkat lunak yang berguna untuk pengembangan situs web dengan bahasa pemrograman PHP dan juga sebagai <i>server</i> untuk local dalam pembuatan <i>database</i> dengan MySql.
4.	Google Chrome	Sebuah <i>web browser open source</i> yang dikembangkan oleh google untuk mengaplikasikan sistem informasi berbasis web.

6.2 Implementasi Class

Implementasi sistem informasi pendataan rak *server* pada PT Indosat Ooredoo Surabaya menggunakan *framework CodeIgniter* dengan arsitektur MVC. Untuk memudahkan implementasi sistem ini akan dijabarkan masing-masing dalam class *controller* dan *model* yang terdapat dalam folder “indosatooredoo” (nama folder). Implementasi *Class* dibuat berdasarkan jumlah *class* yang telah dibuat pada sub bab perancangan interaksi dan kelas yang terdiri dari *sequence diagram* dan *class diagram*. *Class* yang telah dirancang pada proses perancangan direalisasikan pada sebuah file dengan ekstensi .php. pada Tabel 6.1 merupakan hubungan antara class yang dirancang dengan file yang dibuat.

Tabel 6.1 Implementasi Class pada folder kode program

No.	Package	Nama Class	Lokasi file	Nama file
1.	<i>Controller</i>	Logic	/application/ controllers/	Logic.php
2.	<i>Controller</i>	Aktor	/application/ controllers/	Aktor.php
3.	<i>Controller</i>	Indosat	/application/ controllers/	Indosat.php
4.	<i>Model</i>	Aktor	/application/ controllers/	Indosat_model.php

6.3 Implementasi Program

Pada sub bab ini disertakan kode program yang sudah diimplementasikan pada sistem informasi pendataan rak dan perangkat *server* yang menggunakan *framework CodeIgniter*.

Berikut ini disertakan kode program tambah rak dari *controller* Logic dengan fungsi *tambah_rak()*, dapat dilihat pada Tabel 6.2

Tabel 6.2 Kode Program Tambah Rak

1	<code>public function tambah_rak()</code>
2	<code>{</code>
3	<code> \$this->timeout();</code>
4	<code> \$data = array(</code>
5	<code> 'id_lantai' => \$_POST['id_lantai'],</code>
6	<code> 'nama_rak' => \$_POST['nama_rak'],</code>
7	<code> 'kapasitas' => \$_POST['kapasitas']</code>

8);
9	if (\$this->indosat_model->tambah_rak(\$data)) {
10	\$this->session->set_flashdata('message', 'Rak Berhasil Ditambahkan.');
11	redirect('admin');
12	}else{
13	\$this->session->set_flashdata('message', 'Terjadi Kesalahan.');
14	redirect('admin');
15	}
16	}

Berikut ini disertakan kode program semua perangkat dari *controller* Indosat dengan fungsi *semua_perangkat()*, dapat dilihat pada Tabel 6.3

Tabel 6.3 Kode Program Semua Perangkat

1	public function semua_perangkat()
2	{
3	\$this->timeout();
4	\$this->load->library('pagination');
5	\$config['base_url'] = base_url() . 'admin/semua_perangkat/';
6	\$config['total_rows'] = \$this->indosat_model-> >get_all_perangkat_rows();
7	\$this->pagination->initialize(\$config);
8);
9	\$perangkat = \$this->indosat_model-> >get_semua_perangkat(15, \$this->uri->segment(4));
10	\$rak = \$this->indosat_model-> >get_daftar_rak();
11	\$data = array(
12	'title' => 'Daftar Perangkat',
13	'content' => \$this->load-> >view('user/perangkat', array('perangkat' => \$perangkat, 'rak' => \$rak), true)

);
14	\$this->load->view('user/master', \$data);
15	}

Berikut ini disertakan kode program *users* dari *controller* Admin dengan fungsi *users()*, dapat dilihat pada Tabel 6.4

Tabel 6.4 Kode Program User

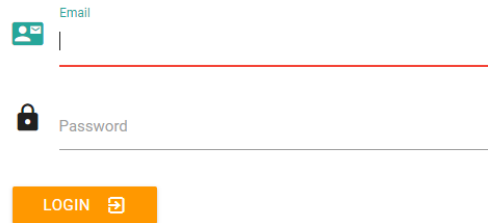
1	public function users()
2	{
3	\$this->timeout();
4	\$this->load->library('pagination');
5	\$config['base_url'] = base_url().'admin/users/';
6	\$config['total_rows'] = \$this->indosat_model->get_all_perangkat_rows();
7	\$this->pagination->initialize(\$config);
8);
9	\$users = \$this->indosat_model->get_all_users(15, \$this->uri->segment(3));
10	\$rak = \$this->indosat_model->get_daftar_rak();
11	\$data = array('title' => 'Daftar User', 'content' => \$this->load->view('users', array('users' => \$users), true));
14	this->load->view('master', \$data);
15	}

6.4 Implementasi Antarmuka

Antarmuka pengguna merupakan bentuk tampilan yang berhubungan langsung dengan sistem. Antarmuka pengguna berfungsi untuk menghubungkan antara pengguna dengan sistem, sehingga komputer dapat digunakan. Berikut ini merupakan implementasi antarmuka pengguna pada sistem informasi pendataan rak dan perangkat *server* PT Indosat Ooredoo Surabaya.

6.4.1 Antarmuka Login

LOGIN



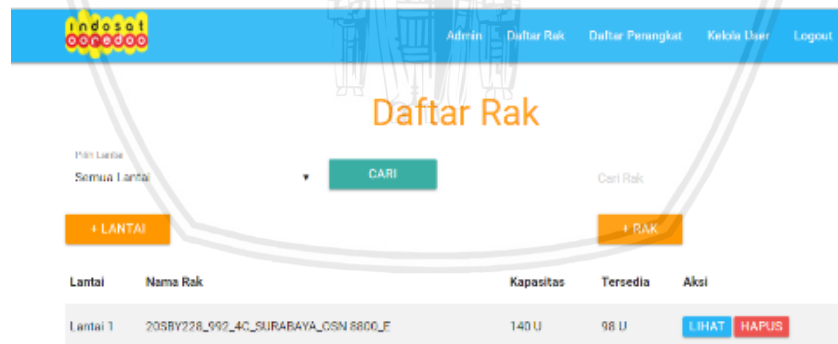
The screenshot shows a login form with two input fields: 'Email' and 'Password'. Below the fields is an orange 'LOGIN' button with a right-pointing arrow icon.

Gambar 6.1 Antarmuka Login

Pada Gambar 6.2 menampilkan implementasi antarmuka halaman login, terdapat 3 fungsi utama pada halaman login yaitu (1) memasukkan username, (2) memasukkan password, (3) tombol login.

6.4.2 Antarmuka Mengelola Rak

Pada Gambar 6.2 menampilkan implementasi antarmuka halaman daftar rak, terdapat 4 fungsi utama pada halaman daftar rak yaitu (1) tambah rak, (2) lihat rak, (3) hapus rak, (4) cari rak.



The screenshot shows the 'Daftar Rak' interface. At the top, there is a blue navigation bar with the 'indosat ooredoo' logo and menu items: 'Admin', 'Daftar Rak', 'Daftar Perangkat', 'Kelola User', and 'Logout'. Below the navigation bar, the title 'Daftar Rak' is displayed in orange. There are two orange buttons: '+ LANTAI' and '+ RAK'. A search section includes a dropdown menu for 'Pilih Lantai' (set to 'Semua Lantai'), a 'CARI' button, and a 'Cari Rak' input field. Below this is a table with the following data:

Lantai	Nama Rak	Kapasitas	Tersedia	Aksi
Lantai 1	20SBY228_092_4C_SURABAYA_OSN 8800_F	140 U	98 U	LIHAT HAPUS

Gambar 6.2 Antarmuka Daftar Rak Petugas Server

Pada Gambar 6.3 menampilkan implementasi antarmuka formulir tambah rak. Di dalam formulir tambah rak menampilkan pilihan untuk memilih lantai, nama rak *server* dan kapasitas dari rak *server*. Selain itu terdapat fungsi untuk menyimpan dan membatalkan.

Gambar 6.3 Antarmuka Tambah Rak

6.4.3 Antarmuka Mengelola Perangkat

Pada Gambar 6.2 menampilkan implementasi antarmuka halaman daftar rak, terdapat 4 fungsi utama pada halaman daftar rak yaitu (1) tambah perangkat, (2) lihat perangkat, (3) hapus perangkat, (4) cari perangkat.

Nama Rak	Nama Perangkat	Tanggal Masuk	Ukuran	Aksi
20SBY228_992_4C_SURABAYA_OSN 8800_E	siemens	10 October, 2018	42 U	LIHAT HAPUS

Gambar 6.4 Antarmuka Daftar Perangkat Petugas Server

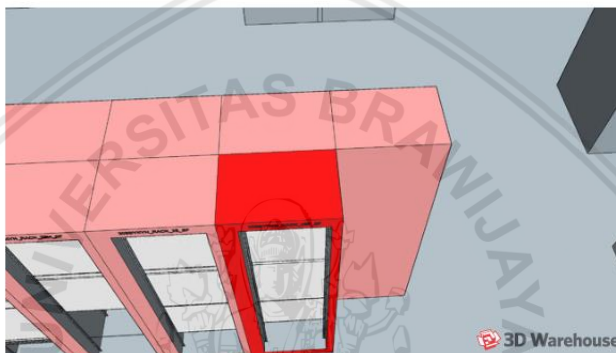
Pada Gambar 6.5 menampilkan implementasi antarmuka tambah perangkat menampilkan. Di dalam formulir tambah perangkat menampilkan pilihan untuk mengisi nama rak *server*, nama perangkat *server*, tanggal masuk, kapasitas dari perangkat *server* dan file *Sketchup*. Selain itu terdapat fungsi untuk menyimpan dan membatalkan.

Gambar 6.5 Antarmuka Tambah Perangkat

Pada Gambar 6.6 menampilkan implementasi antarmuka detail perangkat. Dalam detail perangkat menampilkan informasi berupa informasi perangkat. Selain itu terdapat fungsi *update* yang berguna untuk mengubah informasi perangkat *server* tersebut.

Detail Perangkat

Nama Perangkat: siemens
 Tanggal Masuk: 10 October, 2018 UPDATE
 Nama Rak:
 20SBY228_992_4C_SURABAYA_OSN
 8800_E
 Lantai: Lantai 1



Gambar 6.6 Antarmuka Detail Perangkat

Pada Gambar 6.7 menampilkan formulir dari *update* perangkat. Formulir *update* perangkat berguna untuk mengubah informasi perangkat yang berisi nama perangkat, tanggal masuk, mengisi file skethup. Selanjutnya pengguna bisa memilih fungsi simpan atau batalkan.

Update Perangkat

Nama Perangkat
 siemens

Tanggal Masuk
 01 November, 2018

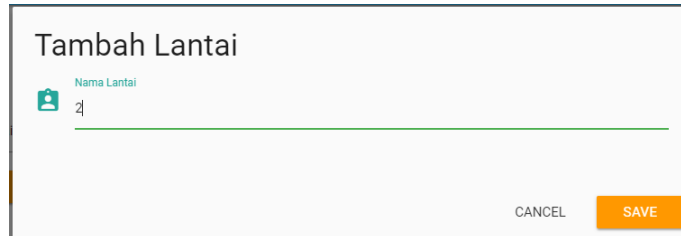
FILE

CANCEL X
SAVE

Gambar 6.7 Antarmuka Update Perangkat

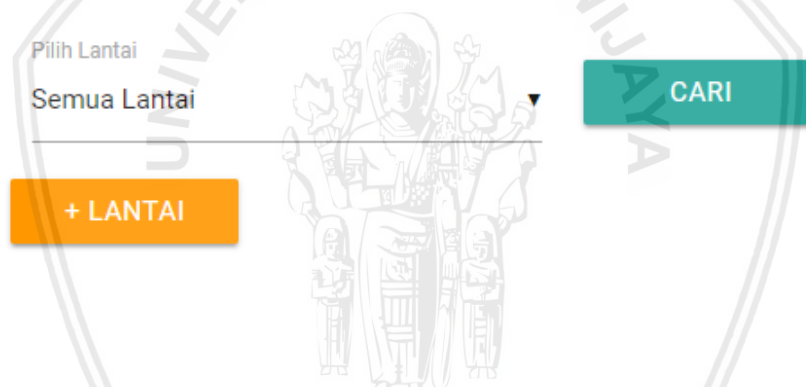
6.4.4 Antarmuka Mengelola Lantai

Pada Gambar 6.8 menampilkan implementasi antarmuka tambah lantai. Pengguna bisa mengisi nama lantai gedung, selanjutnya pengguna dapat memilih menyimpan atau membatalkan.



Gambar 6.8 Antarmuka Tambah Lantai

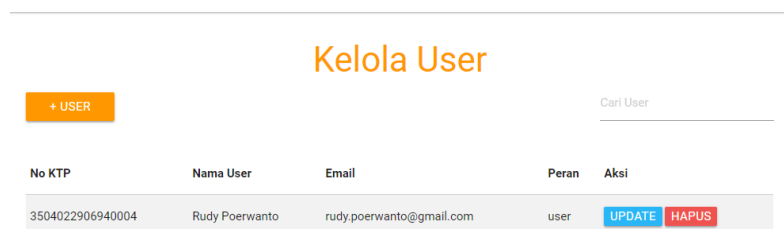
Pada Gambar 6.9 menampilkan implementasi antarmuka cari lantai. Pengguna bisa memilih lantai yang akan dituju dengan antarmuka ini untuk mencari informasi isi lantai gedung.



Gambar 6.9 Antarmuka Pencarian Lantai

6.4.5 Antarmuka Mengelola User

Pada Gambar 6.10 menampilkan implementasi antarmuka halaman kelola user, terdapat 4 fungsi utama pada halaman daftar rak yaitu (1) tambah user, (2) lihat user, (3) hapus user, (4) cari user.



No KTP	Nama User	Email	Peran	Aksi
3504022906940004	Rudy Poerwanto	rudy.poerwanto@gmail.com	user	UPDATE HAPUS

Gambar 6.10 Antarmuka Kelola User

Pada Gambar 6.11 menampilkan implementasi antarmuka formulir tambah user. Di dalam formulir tambah user menampilkan mengisi no KTP, nama pengguna, email pengguna dan password pengguna. Selain itu terdapat fungsi untuk menyimpan dan membatalkan.

Gambar 6.11 Antarmuka Tambah User

6.4.6 Antarmuka Memantau Rak dan Perangkat

Pada Gambar 6.10 menampilkan implementasi antarmuka daftar rak, terdapat 2 fungsi utama pada halaman daftar rak yaitu (1) lihat informasi rak, (2) cari rak. Fungsi lihat menampilkan informasi rak server dan fungsi cari menampilkan informasi nama rak sesuai dengan pencarian nama rak yang dibutuhkan.

Lantai	Nama Rak	Kapasitas	Tersedia	Aksi
Lantai 1	20SBY228_992_4C_SURABAYA_OSN 8800_E	140 U	98 U	LIHAT
Lantai 1	20SBYKYN_RACK_4F_SF_(DMDW HUAWEI)	140 U	68 U	LIHAT

Gambar 6.12 Antarmuka Memantau Rak

Pada Gambar 6.10 menampilkan implementasi antarmuka daftar perangkat, terdapat 2 fungsi utama pada halaman daftar rak yaitu (1) lihat informasi perangkat, (2) cari perangkat. Fungsi lihat menampilkan informasi rak server dan fungsi cari menampilkan informasi nama rak sesuai dengan pencarian nama perangkat yang dibutuhkan.

Indosat Rudy Poerwanto Daftar Rak Daftar Perangkat Logout

Daftar Perangkat

Cari Perangkat

Nama Rak	Nama Perangkat	Tanggal Masuk	Ukuran	Aksi
20SBY228_992_4C_SURABAYA_OSN 8800_E	siemens	10 October, 2018	42 U	LIHAT
20SBYKYN_RACK_4F_SF_(DMDW HUAWEI)	samsung	20 June, 2018	72 U	LIHAT

Gambar 6.13 Antarmuka Memantau Perangkat

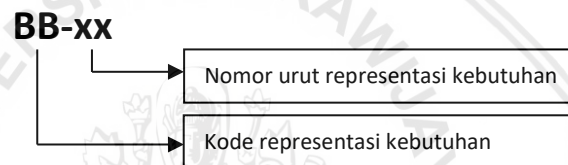


BAB 7 PENGUJIAN

Pada bab ini akan menjelaskan mengenai tahap pengujian yang dilakukan pada sistem informasi pendataan rak dan perangkat *server* PT. Indosat Ooredoo Surabaya. Pengujian dilakukan dengan beberapa cara yaitu pengujian fungsional berupa uji validasi serta menggunakan pengujian nonfungsional berupa uji kompatibilitas terhadap aplikasi peramban.

7.1 Pengujian Fungsional

Pengujian fungsional dilakukan untuk mengetahui bahwa sistem sudah berjalan sesuai dengan ekspektasi pengguna. Pada sub bab ini berisi hasil pengujian fungsional terhadap beberapa fungsi sistem informasi yang digunakan oleh pengguna untuk login, mengelola rak *server*, mengelola perangkat *server*, mengelola lantai, mengelola user serta memantau rak dan perangkat *server*.



Gambar 7.1 Kodifikasi Pengujian Fungsional

Pengujian fungsional dilakukan berdasarkan skenario uji yang diperoleh dari beberapa alur *use case* mengelola rak, mengelola perangkat, mengelola lantai, mengelola user serta memantau rak dan perangkat *server*. Selanjutnya, kasus uji tersebut diidentifikasi berdasarkan skenario uji yang telah ditentukan dan pengujian fungsional dilakukan. Pada Tabel 7.1 menunjukkan gambar kodifikasi pengujian fungsional.

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode BB-01. Kasus uji dengan kode BB-01 dilakukan untuk memastikan bahwa sistem dapat digunakan untuk melakukan login ke dalam sistem. Pengujian dilakukan terhadap penguji yang teridentifikasi sebagai petugas *server*. Pengujian dilakukan pada petugas *server*.

Tabel 7.1 Rencana Pengujian Validasi Fungsi Login

Kode Pengujian	BB-01
Nomor Skenario	Skenario 1
Kode Persyaratan	SIPRPS-KF-01
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan login.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas <i>server</i> .

	<ol style="list-style-type: none"> 2. Penguji memasukkan email: admin@gmail.com dan password: "admin". 3. Penguji menggunakan fungsi login.
Hasil yang diharapkan	Sistem menampilkan halaman home/daftar informasi rak
Status Uji	Valid

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode BB-02. Kasus uji dengan kode BB-02 dilakukan untuk memastikan bahwa sistem dapat digunakan untuk melakukan menambah rak ke dalam sistem. Pengujian dilakukan terhadap penguji yang teridentifikasi sebagai petugas *server*. Pengujian dilakukan pada petugas *server*.

Tabel 7.2 Rencana Pengujian Validasi Fungsi Menambah Rak

Kode Pengujian	BB-05
Nomor Skenario	Skenario 5
Kode Persyaratan	SIPRPS-KF-05
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk menambah rak.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas <i>server</i>. 2. Penguji menggunakan fungsi tambah rak. 3. Penguji mengisi formulir tambah rak. 4. Penguji menyimpan data rak.
Hasil yang diharapkan	Sistem menampilkan informasi rak yang baru
Status Uji	Valid

Pada tabel dibawah ini akan ditunjukkan rencana pengujian validasi dengan kode BB-03. Kasus uji dengan kode BB-03 dilakukan untuk memastikan bahwa sistem dapat digunakan untuk melakukan menambah rak ke dalam sistem. Pengujian dilakukan terhadap penguji yang teridentifikasi sebagai petugas *server*. Pengujian dilakukan pada petugas *serve r*.

Tabel 7.3 Validation Testing Mengelola Perangkat

Kode Pengujian	BB-09
Nomor Skenario	Skenario 9
Kode Persyaratan	SIPRPS-KF-09
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk menambah perangkat.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas <i>server</i>.

	2. Penguji menggunakan fungsi tambah perangkat 3. Penguji mengisi formulir tambah perangkat 4. Penguji menyimpan data perangkat
Hasil yang diharapkan	Sistem menampilkan informasi perangkat yang baru
Status Uji	Valid

$$\text{Hasil} = \frac{\text{Uji Kasus}}{\text{Validitas}} \times 100\%$$

Pada pengujian validasi dengan kode pengujian BB-01, BB-02, BB-03 merupakan beberapa pengujian yang ditampilkan pada pengujian validasi ini. Hasil sebenarnya pengujian validasi memiliki jumlah kasus uji sebanyak 18, dan jumlah fungsi yang sesuai dengan harapan adalah 18. Jadi semua hasil fungsional yang ada sesuai dengan yang diharapkan. Maka disimpulkan bahwa semua kebutuhan fungsional sistem telah dipenuhi.

7.2 Pengujian Nonfungsional

Pada sub bagian ini dijelaskan tentang pengujian nonfungsional berupa uji kompatibilitas peramban untuk mengetahui semua kelayakan sistem pada aplikasi peramban. Pengujian dilakukan dengan menggunakan aplikasi SortSite. Pengujian ini dilakukan dengan menggunakan apache dari perangkat lunak *xampp* sebagai *web server*. Pengujian kompatibilitas diakses pada laman pengguna. Hasil pengujian dapat dilihat pada Gambar 7.2 menunjukkan pada gambar ● menunjukkan bahwa kehilangan konten atau fungsi, pada gambar ◐ menunjukkan masalah tata letak atau kinerja ringan, dan pada gambar ◑ menunjukkan masalah tata letak atau kinerja ringan. Hasil pengujian kompatibilitas yaitu terdapat 2 *critical issues* pada browser *firefox* versi 63 dan *chrome* versi 70. *Critical issues* terjadi pada halaman login. Pada *browser chrome* dan *firefox* menampilkan beberapa peringatan keamanan untuk bidang kata sandi yang tidak menggunakan HTTPS, ini disebabkan karena menggunakan *localhost* dalam menjalankan sistem. Dan terdapat masalah *major issues* apabila situs diakses menggunakan aplikasi peramban internet explorer versi 11 atau sebelumnya. Ini dikarenakan peramban internet explorer terkait tampilan yang tidak mendukung bootstrap.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS			Android	
Version	11	17	63	12	55	70	≤ 10	11	12	≤ 3	4*
Critical Issues	◑	◑	●	◑	◑	●	◑	◑	◑	◑	◑
Major Issues	◐	◑		◐	◑		◑	◑	◑	◑	◑
Minor Issues		◑			◑		◑	◑	◑	◑	◑

Gambar 7.2 Hasil Pengujian Kompatibilitas

BAB 8 PENUTUP

8.1 Kesimpulan

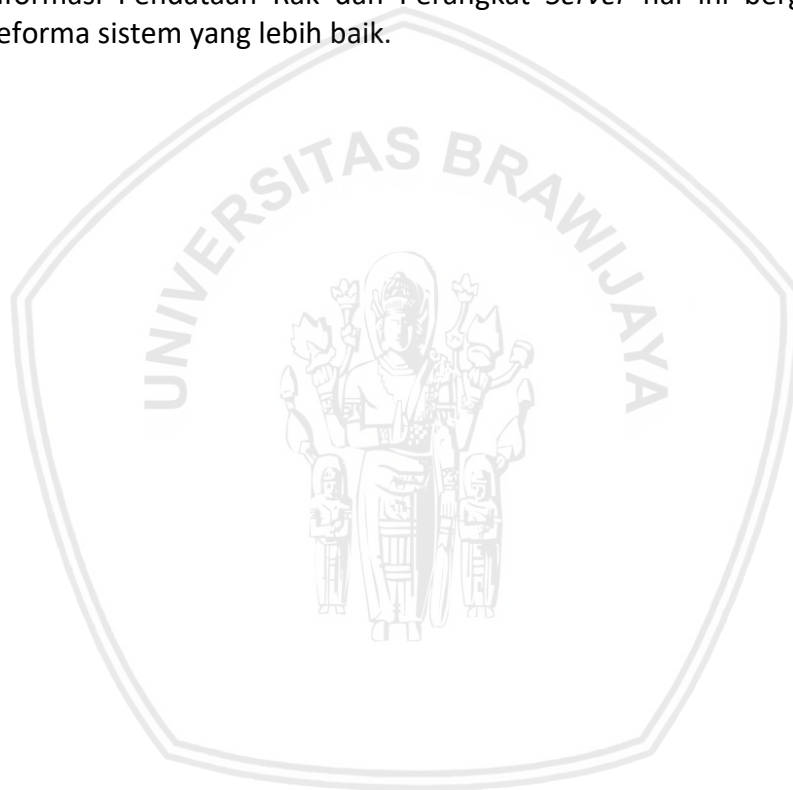
Berdasarkan penelitian yang sudah dilakukan, dapat diambil kesimpulan dari Sistem Informasi Pendataan Rak dan Perangkat *Server* PT. Indosat Ooredoo Surabaya dapat dikembangkan dengan melakukan beberapa kegiatan pengembangan dengan hasil sebagai berikut:

1. Hasil analisis persyaratan dan pemodelan proses bisnis sistem informasi data rak *server* pada PT. Indosat Ooredoo Surabaya merupakan dokumen yang mencakup identifikasi pemangku kepentingan, masalah yang dihadapi oleh pemangku kepentingan, dan deskripsi kebutuhan pengguna. Kemudian, kegiatan analisis terhadap kebutuhan pengguna dilakukan sehingga persyaratan sistem dapat teridentifikasi. Persyaratan sistem memuat informasi mengenai 5 fitur, 18 persyaratan fungsional, dan 1 persyaratan nonfungsional Sistem Informasi Pendataan Rak dan Perangkat *Server* yang mempresentasikan kemampuan sistem secara umum untuk mengelola pendataan rak, mengelola pendataan perangkat, mengelola rantai, mengelola user serta memantau rak dan perangkat *server*. Hasil persyaratan juga memuat mengenai tujuan penggunaan sistem dan urutan aktivitas yang dilakukan oleh pengguna ketika pengguna menggunakan sistem. Informasi tersebut didokumentasikan melalui 6 use case.
2. Perancangan sistem informasi data rak *server* pada PT. Indosat Ooredoo Surabaya berdasarkan hasil analisis persyaratan adalah *model* interaksi objek yang digambarkan dalam *sequence diagram*, *model* objek yang dijelaskan pada kelas diagram, PDM berisi tabel-tabel sebagai penyimpanan media data sistem informasi, beberapa rancangan antarmuka, beberapa rancangan algoritme.
3. Sistem Informasi Pendataan Rak dan Perangkat *Server* pada PT. Indosat Ooredoo Surabaya merupakan sistem informasi berbasis web berdasarkan perancangan sistem yang telah dilakukan. Sistem Informasi Pendataan Rak dan Perangkat *Server* memberikan fitur login, mengelola rak, mengelola perangkat, mengelola rantai dan mengelola pengguna.
4. Hasil pengujian *fungsional* yang berkaitan dengan login, mengelola rak, mengelola perangkat, mengelola rantai, mengelola user dinyatakan valid. Hasil pengujian kompatibilitas menunjukkan adanya masalah pada konten yang hilang pada aplikasi peramban Firefox dan Chrome. Selanjutnya terdapat masalah major terhadap *layout* laman situs apabila diakses melalui peramban Internet Explore dan Safari.

8.2 Saran

Saran yang dapat diberikan untuk bahan pertimbangan perbaikan untuk melakukan pengembangan lebih lanjut pada Sistem Informasi Pendataan Rak dan Perangkat *Server* PT. Indosat Ooredoo Surabaya adalah:

1. Perlu adanya evaluasi pada antarmuka Sistem Informasi Pendataan Rak dan Perangkat *Server* untuk mengetahui perlu dilakukan perbaikan atau pengembangan pada antarmuka pada pengembangan lanjut.
2. Perlu dilakukan penerapan Sistem Informasi Pendataan Rak dan Perangkat *Server* pada lingkungan perusahaan PT. Indosat Ooredoo Surabaya kemudian mengadakan kegiatan evaluasi penerapan Sistem Informasi Pendataan Rak dan Perangkat *Server* hal ini berguna untuk performa sistem yang lebih baik.



DAFTAR PUSTAKA

- Alshamrani, A. & Bahattab, A., 2015. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science*, XII(1), pp. 106-111.
- Ambyssoft Inc., 2013. *Physical Data Model (PDM)s: An Agile Introduction*. [Online] Available at: <http://agilemodeling.com/artifacts/physicalDataModel.htm> [Diakses 17 Desember 2018].
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. Boston: Addison Wesley.
- Booch, G., Rumbaugh, J. & Jacobson, I., 1998. *The Unified Modeling Language User Guide*. Boston: Addison Wesley.
- Fuadi, M.J. 2018. Pengembangan Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(12), pp. 6713-6721
- Kusumawati, T., 2013. PEMBUATAN MEDIA PROMOSI BERBASIS WEBSITE PADA GRAHA PRIMA RESTAURANT PACITAN. II(1), pp. 7-12.
- Krol, P. & Kruchten, P., 2003. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Boston: Addison Wesley.
- Leffingwell, D. & Widrig, D., 2003. *Managing Software Requirements: A Use Case Approach*. 2nd penyunt. Boston: Addison Wesley.
- Monk, E. & Wagner, B., 2013. *Concepts in Enterprise Resource Planning*. 4th penyunt. Boston: Course Technology.
- Object Management Group, 2005. *OMG Unified Modeling Language Specification*. [Online] Available at: <http://www.omg.org/spec/UML/2.0/> [Diakses 31 Oktober 2018].
- Object Management Group, 2005. *Unified Modeling Language: Superstructure*. [Online] Available at: <http://www.omg.org/spec/UML/2.0/Superstructure/PDF/> [Diakses 31 Oktober 2018].
- Object Management Group, 2011. *Business Process Model and Notation (BPMN)*. [Online] Available at: <http://www.omg.org/spec/BPMN/2.0/PDF> [Diakses 20 November 2018].
- Pratama Yuda, I.G. 2018. Pengembangan Sistem Informasi Management Perpustakaan Pada Ruang Baca Ilmu Komputer Universitas Brawijaya. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(12), pp. 6669-6678.
- Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach*. 7th

- Rizki, I., Priadi, R. A. S. & Yuniati, Y., 2014. Pembuatan Sistem Informasi Penjualan Berbasis Web (Studi Kasus pada Toko Ali Computer). *Jurnal Rekayasa dan Teknologi Elektro*, VIII(1), pp. 37-44.
- Rumbaugh, J., Jacobson, I. & Booch, G., 2004. *The Unified Modeling Language Reference Manual*. 2nd penyunt. Boston: Addison Wesley.
- S, Rosa, A dan Shalahuddin, M. 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung : Informatika.
- Scott, K., 2004. *Fast Track UML 2.0*. New York: Springer-Verlag .
- Sommerville, I., 2011. *Software Engineering*. 9th penyunt. Boston: Addison-Wesley.
- Winter, A. et al., 2011. *Health Information Systems: Architectures and Strategies*. 2nd penyunt. London: Springer.



LAMPIRAN A HASIL WAWANCARA

A.1 Hasil Wawancara dengan Petugas Server

LAMPIRAN A HASIL WAWANCARA

A.1 Hasil Wawancara dengan Petugas Server

Berikut ini merupakan hasil wawancara yang tidak terstruktur untuk mengetahui alur proses pendataan rak dan perangkat server PT. Indosat Ooredoo Surabaya. Wawancara dilaksanakan pada:

Tanggal : 8 Januari 2018
Waktu : 09.00 – 10.00 WIB
Tempat : PT Indosat Ooredoo Surabaya
Narasumber : Rizal Pratama
Jabatan : Petugas Server

Pertanyaan: Bagaimana alur proses pendataan rak dan perangkat server?

Jawaban: proses pendataan alur rak dan perangkat dilakukan secara manual. pertama petugas server mendata ruangan pada setiap lantai. Kemudian data tersebut disimpan pada MS. Word. Selanjutnya data tersebut dimasukkan pada aplikasi SetchUp. Setelah itu, data tersebut dilaporkan kepada Kepala Teknik Backbound agar beliau bisa melakukan pengecekan. Selanjutnya data tersebut dirasa sudah benar akan dikirimkan ke Kepala Teknik Backbound Indosat yang di Jakarta.

Pertanyaan: Apakah proses pendataan rak dan perangkat server masih menggunakan teknik manual?

Jawaban: ya, masih menggunakan MS. Word dan SketchUp.

Pertanyaan: Apakah hal tersebut tidak memberatkan untuk petugas server dan perusahaan?

Jawaban: ya, memberat. Karena data tersebut kadang tidak valid dengan kondisi saat ini. Pendataan hanya dilakukan bila ada server yang masuk dan setelah itu tidak pernah mendapat perhatian lagi. Jadi kalo mau melakukan pemasangan ya melakukan pengecekan lagi.

Pertanyaan: Bagaimana bila ada pemasangan rak atau perangkat server pada perusahaan PT. Indosat Ooredoo Surabaya?

Jawaban: Klien harus menghubungi pihak Indosat Jakarta untuk melakukan pemasangan perangkat. Setelah itu, Kepala Teknik Backbound Jakarta menghubungi Kepala Teknik Backbound Surabaya untuk melakukan pengecekan ada tidaknya tempat untuk pemasangan. Selanjutnya Kepala Teknik Backbound Surabaya mengarahkan Petugas Server untuk melakukan pengecekan. Setelah dirasa ada tempat baru menghubungi pihak Indosat Jakarta kembali dan melakukan pemasangan.

Jawaban: ya yang bisa mengelola pendataan rak dan perangkat mbak. Terus ada gambarnya SkethUp dimasukkan biar lebih mempermudah lagi kalo melakukan pencarian.

Pertanyaan: Selain melakukan pendataan rak dan perangkat server, adakah kebutuhan lain yang perlu ditambahkan dalam sistem tersebut?

Jawaban: ada, penambahan informasi lantai ya mbak sama bisa update data. Jadi nanti kalo ada pemasangan lagi bisa diupdate menjadi data yang terbaru

Surabaya, 8 Januari 2018



Rudy Poerwanto



A.2 Hasil Wawancara dengan Kepala Teknik Backboud

A.2 Hasil Wawancara dengan Kepala Teknik Backboud

Berikut ini merupakan hasil wawancara yang tidak terstruktur untuk mengetahui alur proses pendataan rak dan perangkat server PT. Indosat Ooredoo Surabaya. Wawancara dilaksanakan pada:

Tanggal : 8 Januari 2018
 Waktu : 13.00 – 14.00 WIB
 Tempat : PT Indosat Ooredoo Surabaya
 Narasumber : Rudy Poerwanto
 Jabatan : Kepala eknik Bakboud

Pertanyaan: Apakah tugas bapak dalam divisi Teknik ini?

Jawaban: Tugas saya hanya melakukan pemantauan dan bertanggung jawab atas pemasangan rak dan perangkat pada PT. Indosat Ooredoo Subaya.

Pertanyaan: Bagaimana cara Anda melakukan pemantauan rak dan perangkat server?

Jawaban: Selama ini saya hanya melihat data pada SketchUp.

Pertanyaan: Apakah data dalam SketchUp tersebut sesuai dengan kondisi saat ini?

Jawaban: tidak, data yang sudah tercatat hanya data rak dan perangkat server yang sudah terpasang sebelumnya dan tidak mengalami perubahan. Jadi kalo mau pasang rak dan perangkat server ya harus dicek satu-satu terlebih dahulu.

Pertanyaan: Apakah hal itu mempersulit proses pendataan rak dan perangkat server?

Jawaban: ya, hal tersebut menyulitkan saya dan petugas server.

Pertanyaan: Apakah selama ini sudah ada upaya untuk menanggulangi hal tersebut?

Jawaban: belum ada, karna SDM kami yang belum dapat mumpuni untuk membuat sebuah aplikasi yang mempermudah untuk melakukan pendataan rak dan perangkat server.

Pertanyaan: Apakah bapak setuju jika saya melakukan penelitian dengan mengambil topik masalah tersebut?

Jawaban: ya, saya sangat setuju sekali mbak. Karena hal ini dapat mempercepat dan mempermudah untuk kinerja dari petugas server dan saya juga.

Pertanyaan: Apakah bapak tidak merasa keberatan jika nanti saya mengambil data pada salah satu ruangan server?

Jawaban: tidak masalah, selama data tersebut tidak disalah gunakan.

Pertanyaan: Website seperti apakah yang bapak harapkan untuk mempermudah pendataan rak dan perangkat server?

Jawaban: ya yang bisa mengelola pendataan rak dan perangkat mbak. Terus ada gambarnya SkethUp dimasukkan biar lebih mempermudah lagi kalo melakukan pencarian.

Pertanyaan: Selain melakukan pendataan rak dan perangkat server, adakah kebutuhan lain yang perlu ditambahkan dalam sistem tersebut?

Jawaban: ada, penambahan informasi lantai ya mbak sama bisa update data. Jadi nanti kalo ada pemasangan lagi bisa diupdate menjadi data yang terbaru

Surabaya, 8 Jnuari 2018



Rudy Poerwanto



LAMPIRAN B HASIL PENGUJIAN

B.1 Hasil Kuisioner Pengujian Validasi

LAMPIRAN B HASIL PENGUJIAN

B.1 Hasil Kuisioner Pengujian Validasi

Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

- Penguji adalah Petugas Server.
- Penguji menggunakan fungsi login dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
- Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-01
Nomor Skenario	Skenario 1
Kode Persyaratan	SIPRPS-KF-02
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan login.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Penguji memasukkan email: admin@gmail.com dan password: "admin". 3. Penguji menggunakan fungsi login.
Hasil diharapkan	yang Sistem menampilkan halaman home/daftar informasi rak
Status Uji	

Surabaya, 10 Oktober 2018

Rizal Pratama
Rizal Pratama

Pengguna : Kepala Teknik Backbound
 Nama : Rudy Poerwanto
 Umur : 46 tahun
 Divisi : Teknik

Petunjuk Pengisian

- Penguji adalah Kepala Teknik Backbound.
- Penguji menggunakan fungsi login dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
- Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-02
Nomor Skenario	Skenario 2
Kode Persyaratan	SIPRPS-KF-02
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan login.
Prosedur Uji	1. Penguji teridentifikasi sebagai kepala teknik backbound. 2. Penguji memasukkan email: rudy.poorwanto@gmail.com dan password: "admin". 3. Penguji menggunakan fungsi login.
Hasil diharapkan	yang Sistem menampilkan halaman home/daftar informasi rak
Status Uji	

Surabaya, 10 Oktober 2018

Rudy Poerwanto
RUDY POERWANTO

Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

- Penguji adalah Petugas Server.
- Penguji menggunakan fungsi lihat dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
- Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-03
Nomor Skenario	Skenario 3
Kode Persyaratan	SIPRPS-KF-03
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk menampilkan daftar rak yang ada.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Sistem menampilkan halaman daftar rak 3. Penguji menggunakan fungsi lihat pada halaman daftar rak.
Hasil diharapkan	yang Sistem menampilkan halaman informasi perangkat yang terhubung dengan rak.
Status Uji	

Surabaya, 10 Oktober 2018

Rizal Pratama
Rizal Pratama

Pengguna : Kepala Teknik Backbound
 Nama : Rudy Poerwanto
 Umur : 46 tahun
 Divisi : Teknik

Petunjuk Pengisian

- Penguji adalah Kepala Teknik Backbound.
- Penguji menggunakan fungsi lihat dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
- Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-04
Nomor Skenario	Skenario 4
Kode Persyaratan	SIPRPS-KF-03
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk menampilkan daftar rak yang ada.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Sistem menampilkan halaman daftar rak 3. Penguji menggunakan fungsi lihat pada halaman daftar rak.
Hasil diharapkan	yang Sistem menampilkan halaman informasi perangkat yang terhubung dengan rak.
Status Uji	

Surabaya, 10 Oktober 2018

Rudy Poerwanto
RUDY POERWANTO

Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi cari rak dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-07
Nomor Skenario	Skenario 7
Kode Persyaratan	SIPRPS-KF-07
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pencarian informasi rak.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi cari rak pada halaman daftar rak.
Hasil yang diharapkan	Sistem menampilkan informasi rak yang dicari oleh petugas server
Status Uji	

Surabaya, 10 Oktober 2018


 Rizal Pratama

111

Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi cari rak dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-07
Nomor Skenario	Skenario 7
Kode Persyaratan	SIPRPS-KF-07
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pencarian informasi rak.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi cari rak pada halaman daftar rak
Hasil yang diharapkan	Sistem menampilkan informasi rak yang dicari oleh petugas server
Status Uji	

Surabaya, 10 Oktober 2018


 Rizal Pratama

111

Pengguna : Kepala Teknik Backbound
 Nama : Rudy Poerwanto
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Kepala Teknik Backbound.
2. Penguji menggunakan fungsi cari rak dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-08
Nomor Skenario	Skenario 8
Kode Persyaratan	SIPRPS-KF-07
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pencarian informasi rak.
Prosedur Uji	1. Penguji teridentifikasi sebagai kepala teknik backbound. 2. Penguji menggunakan fungsi cari rak pada halaman daftar rak
Hasil yang diharapkan	Sistem menampilkan informasi rak yang dicari oleh petugas server
Status Uji	

Surabaya, 10 Oktober 2018


 Rudy Poerwanto

112

Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi tambah perangkat dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-09
Nomor Skenario	Skenario 9
Kode Persyaratan	SIPRPS-KF-09
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan menambah perangkat.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi tambah perangkat pada halaman daftar perangkat. 3. Penguji mengisi formulir tambah perangkat. 4. Penguji menyimpan formulir tambah perangkat.
Hasil yang diharapkan	Sistem menampilkan informasi perangkat yang baru ditambahkan.
Status Uji	

Surabaya, 10 Oktober 2018


 Rizal Pratama

113



Pengguna : Petugas Server
Nama : Rizal Pratama
Umur : 37 tahun
Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi hapus perangkat dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-10
Nomor Skenario	Skenario 10
Kode Persyaratan	SIPRPS-KF-10
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan menghapus informasi perangkat.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi hapus perangkat pada halaman daftar perangkat. 3. Sistem menghapus informasi perangkat.
Hasil yang diharapkan	Sistem menampilkan informasi perangkat yang masih tersedia.
Status Uji	

Surabaya, 10 Oktober 2018

Rizal Pratama
Rizal Pratama

Pengguna : Petugas Server
Nama : Rizal Pratama
Umur : 37 tahun
Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi update perangkat dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-11
Nomor Skenario	Skenario 11
Kode Persyaratan	SIPRPS-KF-11
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan mengubah informasi perangkat.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi lihat perangkat pada halaman daftar perangkat. 3. Penguji menggunakan fungsi update perangkat. 4. Sistem menampilkan formulir update perangkat. 5. Penguji menyimpan update perangkat.
Hasil yang diharapkan	Sistem menampilkan formulir perangkat dan dapat menyimpan data perangkat terbaru.
Status Uji	

Surabaya, 10 Oktober 2018

Rizal Pratama
Rizal Pratama

Pengguna : Petugas Server
Nama : Rizal Pratama
Umur : 37 tahun
Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi cari rak dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-12
Nomor Skenario	Skenario 12
Kode Persyaratan	SIPRPS-KF-12
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pencarian informasi perangkat.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi cari rak pada halaman daftar rak.
Hasil yang diharapkan	Sistem menampilkan informasi rak yang dicari oleh petugas server
Status Uji	

Surabaya, 10 Oktober 2018

Rizal Pratama
Rizal Pratama

Pengguna : Kepala Teknik Backbound
Nama : Rudy Poerwanto
Umur : 46 tahun
Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Kepala Teknik Backbound.
2. Penguji menggunakan fungsi cari rak dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-13
Nomor Skenario	Skenario 13
Kode Persyaratan	SIPRPS-KF-12
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pencarian informasi rak.
Prosedur Uji	1. Penguji teridentifikasi sebagai kepala teknik backbound. 2. Penguji menggunakan fungsi cari rak pada halaman daftar rak.
Hasil yang diharapkan	Sistem menampilkan informasi rak yang dicari oleh kepala teknik backbound
Status Uji	

Surabaya, 10 Oktober 2018

Rizal Pratama
Rizal Pratama



Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi tambah lantai dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-14
Nomor Skenario	Skenario 14
Kode Persyaratan	SIPRPS-KF-14
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan menambahkan lantai.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi tambah lantai pada halaman daftar rak. 3. Penguji mengisi formulir tambah lantai. 4. Penguji menyimpan formulir tambah lantai.
Hasil yang diharapkan	Sistem menampilkan formulir tambah lantai.
Status Uji	

Surabaya, 10 Oktober 2018


 Rizal Pratama

118

Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi tambah pengguna dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-15
Nomor Skenario	Skenario 15
Kode Persyaratan	SIPRPS-KF-15
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan menambahkan pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi tambah pengguna pada halaman kelola user. 3. Sistem menampilkan formulir tambah pengguna. 4. Penguji mengisi formulir tambah pengguna. 5. Penguji menyimpan formulir tambah pengguna.
Hasil yang diharapkan	Sistem menampilkan formulir tambah pengguna dan menampilkan informasi pengguna yang telah ditambahkan.
Status Uji	

Surabaya, 10 Oktober 2018


 Rizal Pratama

119

Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi hapus user dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-16
Nomor Skenario	Skenario 16
Kode Persyaratan	SIPRPS-KF-16
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan penghapusan pengguna yang tidak diperlukan lagi.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi hapus pengguna pada halaman kelola user. 3. Sistem menampilkan informasi pengguna yang masih tersedia.
Hasil yang diharapkan	Sistem menampilkan informasi pengguna yang masih tersedia.
Status Uji	

Surabaya, 10 Oktober 2018


 Rizal Pratama

120

Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi update user dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-17
Nomor Skenario	Skenario 17
Kode Persyaratan	SIPRPS-KF-17
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk mengubah informasi pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi update pada halaman kelola user. 3. Sistem menampilkan formulir update pengguna. 4. Penguji mengisi formulir update pengguna. 5. Penguji menekan tombol update.
Hasil yang diharapkan	Sistem menampilkan formulir update pengguna dan menampilkan informasi terbaru pengguna.
Status Uji	

Surabaya, 10 Oktober 2018


 Rizal Pratama

121




Pengguna : Petugas Server
 Nama : Rizal Pratama
 Umur : 37 tahun
 Divisi : Teknik

Petunjuk Pengisian

1. Penguji adalah Petugas Server.
2. Penguji menggunakan fungsi cari pengguna dengan prosedur uji dan nilai masukan (input) sesuai dengan rencana pengujian pada tabel dibawah ini.
3. Untuk jawaban hasil pengujian diisi dengan "Valid" apabila keluaran sesuai dengan harapan dan jawaban diisi dengan "Tidak Valid" apabila keluaran tidak sesuai dengan harapan jawaban.

Kode Pengujian	BB-18
Nomor Skenario	Skenario 18
Kode Persyaratan	SIPRPS-KF-18
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat digunakan untuk melakukan pencarian pengguna yang dibutuhkan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas server. 2. Penguji menggunakan fungsi cari pengguna pada halaman kelola user. 3. Sistem menampilkan informasi pengguna yang dicari.
Hasil yang diharapkan	Sistem menampilkan informasi pengguna yang dibutuhkan.
Status Uji	

Surabaya, 10 Oktober 2018


 Rizal Pratama

122

