

**PENGEMBANGAN SISTEM INFORMASI PEMBAYARAN
BERBASIS WEB PADA USAHA TV KABEL
(Studi Kasus: Linda Cable)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Thariq Al Astagis
NIM: 165150400111040



**PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021**



PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI PEMBAYARAN BERBASIS WEB PADA
USAHA TV KABEL
(STUDI KASUS: LINDA CABLE)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Thariq Al Astagis
NIM: 165150400111040

Skripsi ini telah diuji dan dinyatakan lulus pada
15 Juli 2021

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II



Satrio Agung Wicaksono, S.Kom., M.Kom.
NIP. 198605212012121001

Bondan Sapta Prakoso, S.T., M.Kom.
NIP. 197408072003121002

Mengetahui
Ketua Jurusan Sistem Informasi




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

PERNYATAAN ORISINALITAS

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

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

Jember, 24 Juni 2021



Thariq Al Astagis

NIM: 165150400111040

ABSTRAK

Thariq Al Astagis, Pengembangan Sistem Informasi Pembayaran Berbasis Web pada Usaha TV Kabel (Studi Kasus: Linda Cable)

Pembimbing: Satrio Agung Wicaksono, S.Kom., M.Kom., Bondan Sapta Prakoso, S.T., M.Kom.

Linda Cable merupakan usaha TV kabel yang berada di Kecamatan Panggul, Kabupaten Trenggalek yang memiliki 1400 pelanggan aktif. Setiap bulan Linda Cable membebaskan tagihan layanan yang dibayarkan kepada petugas penagihan. Namun terdapat beberapa permasalahan yang terjadi dalam menjalankan pencatatan pembayaran tagihan terkait dengan pengelolaan pembayaran. Pembayaran yang diterima petugas penagihan harus dicatat ulang oleh administrator dalam laporan bulanan untuk pembukuan. Hal ini juga menyebabkan pengelolaan bukti transaksi menjadi tidak efisien karena terjadi duplikasi arsip yang disimpan. Selain itu proses bisnis dari Linda Cable yang akan mengalami perubahan yang mana pelanggan dapat memilih untuk membayar di kantor. Berdasarkan pemaparan masalah tersebut, penelitian ini bertujuan untuk mengembangkan sistem informasi yang dapat membantu Linda Cable dalam melakukan pencatatan pembayaran, mengelola penagihan, mencetak slip tagihan dan membuat laporan bulanan. Metode pengembangan yang digunakan oleh penelitian ini merupakan metode *waterfall* dan dalam penerapannya, aplikasi yang dikembangkan merupakan aplikasi berbasis web. Pengujian yang dilakukan pada penelitian ini adalah pengujian validasi untuk menguji kesesuaian dari fungsional sistem dengan kebutuhan pengguna dan pengujian penerimaan pengguna terhadap sistem informasi yang dikembangkan.

Kata kunci: pembayaran, tagihan, TV kabel, pengelolaan

ABSTRACT

Thariq Al Astagis, Development of Web-Based Payment Information System for Cable TV Business (Case Study: Linda Cable)

Supervisors: Satrio Agung Wicaksono, S.Kom., M.Kom., Bondan Sapta Prakoso, S.T., M.Kom.

Linda Cable is a cable TV business located in Trenggalek which has 1400 active subscribers. Each month, Linda Cable charges a service bill that their customers must pay to the collection clerk. However, there are several problems that occur while recording the payments. Payments received by the collection clerks must be re-recorded by the administrator in the monthly report. Payment receipt are poorly managed and inefficient because of duplicated records. Linda Cable's payment business processes will be changed by allowing customers to choose between paying the bills themselves to the Linda Cable's office or being collected by collection clerk but with additional charges. This study aims to develop an information system that can help Linda Cable in recording payments, managing bills, generating billing slips and generating monthly reports. Waterfall method is used in this research to develop a web-based app. User acceptance test and validation test was performed due to ensure the developed systems has fulfilled the user requirements.

Keywords: *payment, bill, cable TV, management*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Penelitian Sebelumnya	5
2.2 Profil Linda Cable	5
2.3 Sistem Informasi	6
2.3.1 Sistem Informasi Manajemen	6
2.3.2 Sistem Pemrosesan Transaksi	7
2.4 <i>Systems Development Life Cycle</i>	7
2.5 <i>Business Process Model and Notation</i>	8
2.6 <i>Unified Modelling Language</i>	11
2.6.1 <i>Use Case Diagram</i>	11
2.6.2 <i>Activity Diagram</i>	13
2.6.3 <i>Sequence Diagram</i>	14
2.6.4 <i>Class Diagram</i>	16
2.7 <i>Database</i>	17
2.7.1 <i>Data Model</i>	17
2.7.2 <i>MySQL</i>	18



2.8 Perancangan Antarmuka Pengguna	18
2.9 PHP	18
2.9.1 Model-View-Controller	18
2.9.2 Kerangka Kerja Laravel	19
2.10 Pengujian Sistem	19
2.10.1 Pengujian Validasi	20
2.10.2 Pengujian Penerimaan Pengguna	20
BAB 3 METODOLOGI PENELITIAN	21
3.1 Identifikasi Masalah	21
3.2 Studi Literatur	22
3.3 Analisis Kebutuhan Sistem	22
3.4 Perancangan Sistem	22
3.5 Implementasi Sistem	23
3.6 Pengujian Sistem	23
3.7 Kesimpulan dan Saran	23
BAB 4 REKAYASA KEBUTUHAN	24
4.1 Aturan Penomoran	24
4.2 Proses Bisnis Saat Ini (<i>As-Is</i>)	24
4.2.1 Proses Bisnis Pencatatan Pembayaran Saat Ini (<i>As-Is</i>)	25
4.2.2 Proses Bisnis Pendaftaran Pelanggan (<i>As-Is</i>)	26
4.2.3 Proses Bisnis Pembuatan Laporan Bulanan (<i>As-Is</i>)	27
4.3 Analisis Permasalahan Proses Bisnis Saat Ini (<i>As-Is</i>)	27
4.4 Rancangan Proses Bisnis (<i>To-Be</i>)	29
4.4.1 Rancangan Proses Bisnis Pencatatan Pembayaran (<i>To-Be</i>)	29
4.4.2 Rancangan Proses Bisnis Pendaftaran Pelanggan	31
4.4.3 Rancangan Proses Bisnis Pembuatan Laporan Bulanan	31
4.5 Analisis Hasil Perbaikan Proses Bisnis	32
4.6 Identifikasi Pengguna	35
4.7 Identifikasi Fitur	35
4.8 Identifikasi Persyaratan Fungsional	35
4.9 Identifikasi Persyaratan Non-fungsional	36
4.10 Use Case Diagram	36

4.11 Use Case Scenario	38
4.11.1 Skenario Use Case Melihat Daftar Penagihan	39
4.11.2 Skenario Use Case Menambah Pelanggan ke Daftar Penagihan.....	39
4.11.3 Skenario Use Case Menghapus Pelanggan dari Daftar Penagihan.....	40
4.11.4 Skenario Use Case Melihat Data Tagihan Pelanggan.....	40
4.11.5 Skenario Use Case Mencetak Slip Tagihan	40
4.11.6 Skenario Use Case Mencatat Pembayaran Tagihan	41
4.11.7 Skenario Use Case Mencatat Pembayaran Tagihan di Kantor.....	41
4.11.8 Skenario Use Case Memverifikasi Pembayaran Tagihan	42
4.11.9 Skenario Use Case Membuat Laporan Bulanan.....	43
4.12 Activity Diagram	43
4.12.1 Activity Diagram Melihat Daftar Penagihan.....	43
4.12.2 Activity Diagram Menambah Pelanggan ke Daftar Penagihan.....	44
4.12.3 Activity Diagram Menghapus Pelanggan dari Daftar Penagihan.....	44
4.12.4 Activity Diagram Melihat Data Tagihan Pelanggan.....	45
4.12.5 Activity Diagram Mencetak Slip Tagihan	45
4.12.6 Activity Diagram Mencatat Pembayaran Tagihan.....	45
4.12.7 Activity Diagram Mencatat Pembayaran Tagihan di Kantor.....	46
4.12.8 Activity Diagram Memverifikasi Pembayaran Tagihan.....	46
4.12.9 Activity Diagram Membuat Laporan Bulanan.....	47
BAB 5 PERANCANGAN DAN IMPLEMENTASI	48
5.1 Perancangan Sistem	48
5.1.1 Sequence Diagram	48
5.1.2 Class Diagram	54
5.1.3 Pemetaan Class Diagram ke Relational Data Model.....	56
5.1.4 Perancangan Antarmuka Pengguna.....	57
5.1.5 Perancangan Pengujian Penerimaan Pengguna	60
5.2 Implementasi Sistem	62



5.2.1 Implementasi <i>Database</i>	62
5.2.2 Implementasi Fungsi.....	66
5.2.3 Implementasi Antarmuka Pengguna	72
BAB 6 PENGUJIAN	76
6.1 Pengujian Validasi.....	76
6.1.1 Skenario Uji Fungsi Melihat Daftar Penagihan	76
6.1.2 Skenario Uji Fungsi Menambah Pelanggan ke Daftar Penagihan.....	76
6.1.3 Skenario Uji Fungsi Menghapus Pelanggan dari Daftar Penagihan.....	77
6.1.4 Skenario Uji Fungsi Melihat Data Tagihan Pelanggan	78
6.1.5 Skenario Uji Fungsi Mencetak Slip Tagihan	78
6.1.6 Skenario Uji Fungsi Mencatat Pembayaran Tagihan	79
6.1.7 Skenario Uji Fungsi Memverifikasi Pembayaran Tagihan	80
6.1.8 Skenario Uji Fungsi Membuat Laporan Bulanan.....	81
6.2 Hasil Pengujian Validasi	81
6.3 Pengujian Penerimaan Pengguna	82
6.4 Hasil Pengujian Penerimaan Pengguna	82
BAB 7 PENUTUP	84
7.1 Kesimpulan	84
7.2 Saran	85
DAFTAR REFERENSI	86
LAMPIRAN A RANCANGAN WAWANCARA	89
LAMPIRAN B HASIL WAWANCARA	91
LAMPIRAN C HASIL PENGUJIAN PENERIMAAN PENGGUNA	94
LAMPIRAN D SKENARIO USE CASE	96
LAMPIRAN E ACTIVITY DIAGRAM	100
LAMPIRAN F SEQUENCE DIAGRAM	104

DAFTAR TABEL

Tabel 2.1 Notasi Elemen pada BPMN	9
Tabel 2.2 Tipe Aktivitas pada BPMN	10
Tabel 2.3 Notasi Elemen pada <i>Use Case Diagram</i>	12
Tabel 2.4 Notasi Elemen pada <i>Activity Diagram</i>	13
Tabel 2.5 Notasi Elemen pada <i>Sequence Diagram</i>	15
Tabel 2.6 Notasi Elemen pada Class Diagram	16
Tabel 2.7 Tabel Skenario Uji	20
Tabel 4.1 Identifikasi Proses Bisnis Saat Ini	27
Tabel 4.2 Tabel Analisis Permasalahan	28
Tabel 4.3 Identifikasi Masalah	28
Tabel 4.4 Identifikasi Rancangan Proses Bisnis	32
Tabel 4.5 Analisis Aktivitas Rancangan Proses Bisnis	33
Tabel 4.6 Identifikasi Pengguna	35
Tabel 4.7 Identifikasi Fitur	35
Tabel 4.8 Identifikasi Persyaratan Fungsional	36
Tabel 4.9 Identifikasi Aktor <i>Use Case</i>	37
Tabel 4.10 Identifikasi <i>Use Case</i>	38
Tabel 4.11 Skenario <i>Use Case</i> Melihat Daftar Penagihan	39
Tabel 4.12 Skenario <i>Use Case</i> Menambah Daftar Penagihan	39
Tabel 4.13 Skenario <i>Use Case</i> Menghapus Daftar Penagihan	40
Tabel 4.14 Skenario <i>Use Case</i> Melihat Data Tagihan Pelanggan	40
Tabel 4.15 Skenario <i>Use Case</i> Mencetak Slip Tagihan	40
Tabel 4.16 Skenario <i>Use Case</i> Mencatat Pembayaran Tagihan	41
Tabel 4.17 Skenario <i>Use Case</i> Mencatat Pembayaran Tagihan di Kantor	41
Tabel 4.18 Skenario <i>Use Case</i> Memverifikasi Pembayaran Tagihan	42
Tabel 4.19 Skenario <i>Use Case</i> Membuat Laporan Bulanan	43
Tabel 5.1 Rancangan Pengujian Penerimaan Pengguna oleh Administrator	60
Tabel 5.2 Rancangan Pengujian Penerimaan Pengguna oleh Petugas Penagihan	61
Tabel 5.3 Implementasi Tabel Roles	62
Tabel 5.4 Implementasi Tabel Users	62
Tabel 5.5 Implementasi Tabel Notifications	63

Tabel 5.6 Implementasi Tabel Kecamatan.....	63
Tabel 5.7 Implementasi Tabel Desa	64
Tabel 5.8 Implementasi Tabel Pelanggan	64
Tabel 5.9 Implementasi Tabel Penagihan.....	65
Tabel 5.10 Implementasi Tabel Bulan.....	65
Tabel 5.11 Implementasi Tabel Status Tagihan.....	65
Tabel 5.12 Implementasi Tabel Tagihan	66
Tabel 5.13 Fungsi Melihat Daftar Penagihan.....	67
Tabel 5.14 Fungsi Menambah Pelanggan ke Daftar Penagihan	67
Tabel 5.15 Fungsi Menghapus Pelanggan dari Daftar Penagihan	68
Tabel 5.16 Fungsi Melihat Data Tagihan Pelanggan.....	68
Tabel 5.17 Fungsi Mencetak Slip Tagihan.....	69
Tabel 5.18 Fungsi Mencatat Pembayaran Tagihan.....	70
Tabel 5.19 Fungsi Memverifikasi Pembayaran Tagihan	70
Tabel 5.20 Fungsi Membuat Laporan Bulanan	71
Tabel 6.1 Skenario Uji Fungsi Melihat Daftar Penagihan.....	76
Tabel 6.2 Skenario Uji Fungsi Menambah Pelanggan ke Daftar Penagihan	77
Tabel 6.3 Skenario Uji Fungsi Menghapus Pelanggan dari Daftar Penagihan	77
Tabel 6.4 Skenario Uji Fungsi Melihat Data Tagihan Pelanggan.....	78
Tabel 6.5 Skenario Uji Fungsi Mencetak Slip Tagihan.....	78
Tabel 6.6 Skenario Uji Fungsi Mencatat Pembayaran Tagihan	79
Tabel 6.7 Skenario Uji Fungsi Mencatat Pembayaran Tagihan di Kantor.....	79
Tabel 6.8 Skenario Uji Fungsi Memverifikasi Pembayaran Tagihan	80
Tabel 6.9 Skenario Uji Fungsi Meverifikasi Pembayaran Tagihan Alternatif 1	80
Tabel 6.10 Skenario Uji Fungsi Membuat Laporan Bulanan	81
Tabel 6.11 Hasil Pengujian Validasi.....	81
Tabel 6.12 Hasil Pengujian Penerimaan Pengguna oleh Administrator	82
Tabel 6.13 Hasil Pengujian Penerimaan Pengguna oleh Petugas Penagihan	83



DAFTAR GAMBAR

Gambar 2.1 Siklus Hidup Pengembangan Perangkat Lunak.....	7
Gambar 2.2 Metode <i>Waterfall</i>	7
Gambar 2.3 Contoh Pemodelan Proses Bisnis.....	11
Gambar 2.4 Contoh <i>Use Case Diagram</i>	12
Gambar 2.5 Contoh <i>Activity Diagram</i>	14
Gambar 2.6 Contoh <i>Sequence Diagram</i>	15
Gambar 2.7 Contoh <i>Class Diagram</i>	17
Gambar 2.8 Model-View-Controller.....	19
Gambar 2.9 Cara Kerja Laravel.....	19
Gambar 3.1 Diagram Alir Penelitian.....	21
Gambar 4.1 Model Proses Bisnis Pencatatan Pembayaran Saat Ini (<i>As-Is</i>).....	25
Gambar 4.2 Model Subproses Bisnis Penarikan Tagihan Saat Ini (<i>As-Is</i>).....	25
Gambar 4.3 Model Proses Bisnis Pendaftaran Pelanggan Saat ini (<i>As-Is</i>).....	26
Gambar 4.4 Model Proses Bisnis Pembuatan Laporan Bulanan Saat Ini (<i>As-Is</i>).....	27
Gambar 4.5 Model Rancangan Proses Bisnis Pencatatan Pembayaran oleh Petugas Penagihan (<i>To-Be</i>).....	29
Gambar 4.6 Model Rancangan Subproses Penarikan Tagihan (<i>To-Be</i>).....	29
Gambar 4.7 Model Rancangan Proses Bisnis Pencatatan Pembayaran di Kantor (<i>To-Be</i>).....	30
Gambar 4.8 Model Rancangan Proses Bisnis Pendaftaran Pelanggan (<i>To-Be</i>).....	31
Gambar 4.9 Model Rancangan Proses Bisnis Pembuatan Laporan Bulanan (<i>To-Be</i>).....	32
Gambar 4.10 Diagram <i>Use Case</i> Sistem Informasi Pembayaran.....	37
Gambar 4.11 <i>Activity Diagram</i> Melihat Daftar Penagihan.....	43
Gambar 4.12 <i>Activity Diagram</i> Menambah Pelanggan ke Daftar Penagihan.....	44
Gambar 4.13 <i>Activity Diagram</i> Menghapus Pelanggan Daftar Penagihan.....	44
Gambar 4.14 <i>Activity Diagram</i> Melihat Data Tagihan Pelanggan.....	45
Gambar 4.15 <i>Activity Diagram</i> Mencetak Slip Tagihan.....	45
Gambar 4.16 <i>Activity Diagram</i> Mencatat Pembayaran Tagihan.....	46
Gambar 4.17 <i>Activity Diagram</i> Mencatat Pembayaran Tagihan di Kantor.....	46
Gambar 4.18 <i>Activity Diagram</i> Memverifikasi Pembayaran Tagihan.....	47

Gambar 4.19 <i>Activity Diagram</i> Membuat Laporan Bulanan	47
Gambar 5.1 <i>Sequence Diagram</i> Melihat Daftar Penagihan	48
Gambar 5.2 <i>Sequence Diagram</i> Menambah Pelanggan ke Daftar Penagihan	49
Gambar 5.3 <i>Sequence Diagram</i> Menghapus Pelanggan dari Daftar Penagihan ..	49
Gambar 5.4 <i>Sequence Diagram</i> Melihat Data Tagihan Pelanggan	50
Gambar 5.5 <i>Sequence Diagram</i> Mencetak Slip Tagihan	51
Gambar 5.6 <i>Sequence Diagram</i> Mencatat Pembayaran Tagihan	51
Gambar 5.7 <i>Sequence Diagram</i> Mencatat Pembayaran Tagihan di Kantor	52
Gambar 5.8 <i>Sequence Diagram</i> Memverifikasi Pembayaran Tagihan	53
Gambar 5.9 <i>Sequence Diagram</i> Membuat Laporan Bulanan	53
Gambar 5.10 <i>Class Diagram Controller</i>	54
Gambar 5.11 <i>Class Diagram Model</i>	55
Gambar 5.12 Relasi <i>Controller</i> dan <i>Model</i>	55
Gambar 5.13 <i>Relational Data Model</i>	56
Gambar 5.14 Rancangan Antarmuka Halaman Detail Pelanggan	57
Gambar 5.15 Rancangan Antarmuka Halaman Daftar Petugas Penagihan	58
Gambar 5.16 Rancangan Antarmuka Halaman Daftar Penagihan	58
Gambar 5.17 Rancangan Antarmuka Halaman Daftar Tagihan	59
Gambar 5.18 Rancangan Antarmuka Halaman Laporan Bulanan	60
Gambar 5.19 Antarmuka Halaman Detail Pelanggan	72
Gambar 5.20 Antarmuka Halaman Daftar Petugas Penagihan	73
Gambar 5.21 Antarmuka Halaman Daftar Penagihan	73
Gambar 5.22 Antarmuka Halaman Daftar Tagihan	74
Gambar 5.23 Antarmuka Halaman Laporan Bulanan	75

BAB 1 PENDAHULUAN

1.1 Latar Belakang

TV kabel merupakan jasa televisi berlangganan yang digunakan sebagian masyarakat untuk mengakses televisi nasional maupun internasional melalui media kabel. Pelanggan yang menggunakan jasa TV kabel harus membayar dalam kurun waktu tertentu agar tetap mendapatkan layanan dari penyedia TV kabel. Penyedia layanan TV kabel perlu mengelola pembayaran ini untuk meminimalisasi kerugian yang terjadi. Selain itu, sebagai usaha yang berbasis pada pelanggan, maka pemenuhan kepuasan pelanggan diperlukan untuk menjaga hubungan pelanggan agar tetap menggunakan jasa yang ditawarkan oleh perusahaan (Morauszki, József & Attila, 2015). Oleh karena itu, bukti pembayaran akan memenuhi kepuasan pelanggan melalui jaminan atas pelayanan jika terjadi kesalahan pada catatan kantor, sehingga pelanggan tidak akan merasa rugi.

Linda Cable merupakan usaha TV kabel yang berlokasi di Kecamatan Panggul, Kabupaten Trenggalek dan memiliki sekitar 1400 pelanggan aktif. Linda Cable memiliki lebih dari 30 kanal televisi yang disiarkan kepada pelanggannya. Linda Cable membebaskan pelanggannya tagihan setiap bulan dan proses penagihan dilakukan oleh petugas penagihan yang datang ke rumah-rumah pelanggan. Linda Cable memiliki kebijakan untuk melakukan pemutusan saluran pelanggan ketika pelanggan tidak membayar tagihan 3 bulan berturut-turut. Linda Cable memiliki jenis tagihan untuk masing-masing pelanggan, pelanggan satu televisi, lebih dari satu televisi, pelanggan kurang mampu dan pelanggan usaha lain (seperti hotel).

Berdasarkan hasil wawancara kepada pengelola Linda Cable, pencatatan pembayaran tagihan pelanggan dilakukan oleh petugas penagihan setiap bulan, kemudian petugas penagihan menyerahkan daftar pelanggan yang membayar kepada administrator untuk disalin ke buku pembayaran di kantor. Hal ini menurut pengelola Linda Cable tidak efisien dari segi waktu dan tenaga, dan juga sangat banyak arsip fisik yang disimpan hanya untuk catatan pembayaran. Selain itu manusia juga memiliki kemampuan yang terbatas dan mungkin melakukan kesalahan yaitu salah tindakan atau salah ingatan (Butlewski et al., 2014). Linda Cable juga hendak mengubah model pembayaran tagihannya. Proses pembayaran sebelumnya hanya dipungut petugas penagihan, selanjutnya pelanggan akan diberikan opsi untuk membayar di kantor ataupun meminta jasa penagihan dengan biasa tambahan. Oleh karena itu, Linda Cable perlu mencatat pelanggan yang menginginkan jasa penagihan serta membuat bukti pembayaran untuk pelanggan yang membayar di kantor.

Masalah yang terjadi pada Linda Cable tersebut merupakan masalah pada proses bisnis dan operasional dari pembayaran. Menurut Marakas dan O'Brien (2010), sistem informasi berperan sebagai pendukung dalam menjalankan proses bisnis dan operasional, sehingga sistem informasi dapat menjadi salah satu solusi dari permasalahan Linda Cable. Sistem informasi yang dikembangkan pada penelitian dirancang untuk meningkatkan automasi proses bisnis yang sekarang

sedang berjalan menggunakan aplikasi sistem informasi agar proses bisnis pembayaran menjadi lebih akurat, terpadu, efektif dan efisien. Linda Cable dapat menggunakan sistem informasi ini untuk membantu mencatat transaksi serta membuat bukti transaksi secara otomatis. Sistem informasi pembayaran juga dapat membuat laporan bulanan dengan format laporan yang lebih terstruktur.

Pengembangan sistem informasi pembayaran menerapkan siklus hidup pengembangan perangkat lunak atau *Systems Development Life Cycle (SDLC)*. SDLC merupakan tahapan-tahapan untuk memenuhi kebutuhan bisnis, merancang, membangun serta menyampaikan kepada *user* sebuah sistem (Dennis, 2012). Metode SDLC yang digunakan adalah metode *waterfall* untuk meminimalisasi usaha pembangunan sistem dengan mendefinisikan kebutuhan selengkap mungkin di awal (Pressman, 2010). Implementasi kebutuhan ke dalam sistem perlu diuji sebagai verifikasi antara ekspektasi pengguna dengan fitur dari sistem (Leung & Wong, 1997). Perancangan sistem informasi ini didasari oleh proses bisnis yang berjalan di Linda Cable dan memberikan rancangan proses bisnis yang melibatkan sistem. Pemodelan proses bisnis menggunakan *business process model and notation (BPMN)* yang memiliki standar notasi untuk memodelkan proses bisnis. Selain itu pada tahap perancangan sistem juga menggunakan *unifiend modelling language (UML)* yang merupakan standar pemodelan untuk diagram sistem, baik dari fungsional sistem, alur sistem maupun struktur data sistem. Sistem informasi mengaplikasikan *database* sebagai penyimpanan data pada komputer untuk nantinya diolah menjadi informasi oleh sistem.

Berdasarkan latar belakang permasalahan yang telah dipaparkan, perlu dilakukan penelitian ini sebagai upaya untuk menyelesaikan masalah tersebut melalui pengembangan sistem informasi dengan judul "**Pengembangan Sistem Informasi Pembayaran Berbasis Web pada Usaha TV Kabel (Studi Kasus: Linda Cable)**". Melalui penelitian ini, penulis akan melakukan rangkaian proses di dalam siklus hidup pengembangan sistem informasi yang ditujukan untuk menyelesaikan masalah pada Linda Cable. Rangkaian proses ini akan menghasilkan aplikasi pembayaran yang dapat digunakan oleh Linda Cable sebagai instrumen dalam melaksanakan pencatatan pembayaran.

1.2 Rumusan Masalah

Berdasarkan permasalahan pada latar belakang yang telah dipaparkan sebelumnya, dapat diangkat rumusan masalah sebagai berikut:

1. Bagaimana hasil analisis permasalahan pembayaran dan kebutuhan sistem informasi pembayaran pada Linda Cable?
2. Bagaimana rancangan sistem informasi pembayaran yang dapat memenuhi kebutuhan Linda Cable?
3. Bagaimana implementasi sistem informasi pembayaran yang sesuai dengan rancangan?
4. Bagaimana hasil pengujian dari sistem informasi pembayaran yang telah diimplementasikan?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dipaparkan sebelumnya, maka dapat didefinisikan tujuan dari penelitian ini sebagai berikut:

1. Untuk menganalisis permasalahan pembayaran yang terjadi di Linda Cable dan menganalisis kebutuhan sistem informasi pembayaran Linda Cable.
2. Untuk merancang sistem informasi yang dapat memenuhi kebutuhan Linda Cable.
3. Untuk mengimplementasikan sistem informasi pembayaran yang sesuai dengan rancangan sistem informasi.
4. Untuk melakukan pengujian terhadap sistem informasi yang telah diimplementasikan.

1.4 Manfaat

Penelitian ini diharapkan dapat memiliki manfaat penelitian terhadap beberapa hal berikut ini:

1. Sebagai pengetahuan bagi penulis dan pembaca tentang alur pembayaran tagihan TV kabel serupa yang efektif dan efisien.
2. Sebagai bentuk aplikasi dari pengetahuan penulis untuk mengembangkan sebuah sistem informasi.
3. Sebagai bahan referensi untuk penelitian selanjutnya yang berkaitan dengan peningkatan sistem informasi pembayaran TV kabel serupa dan.

1.5 Batasan Masalah

Beberapa batasan masalah penelitian ini didefinisikan untuk mengarahkan penelitian agar fokus pada masalah dan tujuan penelitian. Batasan masalah didefinisikan sebagai berikut:

1. Objek TV yang dilakukan penelitian yaitu Linda Cable yang berlokasi di Kecamatan Panggul, Kabupaten Trenggalek.
2. Data yang digunakan merupakan data hasil observasi dan wawancara kepada pengelola serta petugas Linda Cable.
3. Menggunakan metode *waterfall* dalam kegiatan *Systems Development Life Cycle* (SDLC).
4. Desain sistem yang dilakukan adalah desain berbasis *object oriented* dan menggunakan *unified modelling language* (UML).
5. Aplikasi dikembangkan berbasis pada web.
6. Pengembangan aplikasi menggunakan PHP versi 7.2 sebagai bahasa pemrograman, MySQL versi 5.7 sebagai DBMS, Laravel versi 7.28 sebagai kerangka kerja *backend* dan Bootstrap versi 4.5 sebagai kerangka kerja *frontend*.

1.6 Sistematika Pembahasan

Untuk memberikan gambaran dan kemudahan bagi pembaca dalam memahami isi, penulis memberikan sistematika pembahasan yang terbagi dalam lima bab. Sistematika tersebut terdiri dari:

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang penelitian, rumusan masalah penelitian, tujuan penelitian, manfaat penelitian, batasan masalah penelitian dan sistematika penulisan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi landasan teori yang digunakan penulis untuk mendukung penyelesaian penelitian ini. Teori berasal dari teori yang telah dibuat sebelumnya dan berhubungan dengan judul penelitian ini.

BAB 3 METODE PENELITIAN

Bab ini berisi tentang metode penelitian yang digunakan dalam penelitian ini serta tahapan-tahapan dalam melakukan penelitian.

BAB 4 REKAYASA KEBUTUHAN

Bab ini berisi tentang penggalian kebutuhan pengguna untuk menjadi persyaratan sistem. Rekayasa kebutuhan berdasarkan wawancara kepada pengguna untuk kemudian dilakukan identifikasi fitur.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi tentang perancangan sistem berdasarkan rekayasa kebutuhan dan implementasi berdasarkan rancangan sistem.

BAB 6 PENGUJIAN

Bab ini berisi tentang pengujian sistem yang melihat kesesuaian implementasi sistem dengan kebutuhan pengguna.

BAB 7 PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan pada bab sebelumnya dan saran untuk penelitian selanjutnya yang berkaitan dengan penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Penelitian Sebelumnya

Penelitian terkait dengan pembayaran tagihan telah dilakukan oleh Sutoyo (2020) dengan judul “Perancangan Sistem Informasi Pembayaran SPP Terpadu Menggunakan Model Prototipe”. Pada penelitian tersebut, diangkat masalah tentang proses pembayaran SPP yang lambat dan tingginya tingkat kesalahan yang dapat terjadi, sulitnya pencarian informasi data siswa, serta pembuatan bukti bayar yang lama ketika dilakukan secara manual. Solusi yang ditawarkan penelitian tersebut untuk menyelesaikan masalah pembuatan sistem informasi pembayaran SPP. Sistem informasi yang ditawarkan memiliki fitur untuk *login* untuk membedakan pengguna, mengelola transaksi pembayaran SPP, bukti transaksi dan pelaporan pembayaran SPP. Penelitian tersebut menyatakan bahwa prosedur manual dapat digantikan dengan interaksi dengan sistem informasi pembayaran SPP sehingga kendala pembayaran dapat teratasi.

Penelitian terkait dengan pengembangan sistem informasi pembayaran dengan metode *waterfall* salah satunya telah dilakukan oleh Surniandari dan Rachmi (2020) dengan judul “Sistem Informasi Pengelolaan Transaksi Pembayaran dengan Metode *Waterfall* Untuk Kasir *Event* Multi UKM”. Masalah yang diangkat pada penelitian tersebut adalah kelemahan pada kasir acara UMKM yang dicatat secara manual, sehingga lebih cenderung terjadi kesalahan. Penelitian ini mengembangkan sistem informasi pengelolaan transaksi pembayaran yang dapat mencatat transaksi dan membuat laporan untuk menyelesaikan masalah yang didefinisikan. Hasil dari penelitian tersebut menyatakan bahwa sistem informasi yang dikembangkan dapat mempermudah proses transaksi, memperbaiki dokumentasi, serta mempermudah pembuatan laporan.

Penelitian terkait pembayaran berbasis web juga dilakukan oleh Ramadhan (2019) pada penelitian berjudul “Analisis Dan Perancangan Sistem Pembayaran Jasa Internet Berbasis Web Dan SMS *Gateway*”. Penelitian tersebut memaparkan masalah terkait pencarian dan penyimpanan data secara fisik sulit untuk dikelola dan rawan rusak ataupun kehilangan. Solusi yang ditawarkan ialah perancangan sistem informasi pembayaran yang berbasis web untuk menyelesaikan masalah yang telah digali. Sistem informasi yang dirancang telah mencapai tahap implementasi dan dapat membantu proses pembayaran dan penagihan.

2.2 Profil Linda Cable

Linda Cable merupakan penyedia jasa TV kabel yang berlokasi di Kecamatan Panggul, Kabupaten Trenggalek. Linda Cable menyediakan akses lebih dari 30 kanal televisi kepada sekitar 1400 pelanggannya. Linda Cable memiliki pelanggan paling banyak di Kecamatan Panggul di antara kompetitornya karena memberikan tawaran kanal televisi yang paling banyak. Linda Cable membebaskan sejumlah nominal setiap bulan kepada pelanggannya. Pelanggan dapat melaporkan keluhan melalui media Whatsapp ke kantor Linda Cable untuk ditangani.

Linda Cable dikelola oleh Heri Prasetyo dan memiliki 1 petugas administrasi, 1 petugas teknisi dan 5 petugas penagihan. Pengelola mengelola kebijakan yang akan diambil oleh Linda Cable. Petugas administrasi bertugas untuk mencatat seluruh pembayaran dari pelanggan ke dalam buku pembukuan. Petugas teknisi melakukan pemeliharaan terhadap alat-alat penyiaran dan melakukan pemasangan serta pemutusan saluran TV kabel jika dibutuhkan. Petugas penagihan memiliki tanggung jawab terhadap 300 pelanggan untuk ditagih selama satu bulan penuh. Setiap 10 hari petugas penagihan menyerahkan data pelanggan yang membayar dan uang pembayaran ke kantor.

2.3 Sistem Informasi

Sistem pada umumnya merupakan rangkaian proses *input*, proses dan *output* yang menghasilkan produk dengan *value* bagi organisasi, sehingga sistem informasi merupakan sistem yang berkaitan dengan informasi. Namun definisi sistem informasi meluas menjadi sistem untuk memanfaatkan informasi dalam sebuah organisasi dilakukan oleh rangkaian dari beberapa hal, yaitu orang, perangkat lunak, perangkat keras, jaringan komunikasi, *data resource* serta melibatkan kebijakan dan prosedur di dalam organisasi (Marakas & O'Brien, 2010). Masing-masing elemen yang dilibatkan di dalam sebuah sistem informasi bersinergi untuk menciptakan lingkungan sistem yang dapat meningkatkan kinerja organisasi.

Sistem informasi dapat mengidentifikasi dan menyelesaikan masalah dalam sebuah organisasi, meningkatkan stabilitas finansial, serta meningkatkan kualitas dari strategi organisasi, sehingga organisasi sangat perlu untuk berinvestasi ke arah sistem informasi (Lipaj & Davidavičienė, 2013). Solusi yang ditawarkan menggunakan sistem informasi harus dirancang berdasarkan kebutuhan pemangku kepentingan, sehingga sistem informasi dapat digunakan untuk membantu menyelesaikan masalah di dalam organisasi.

2.3.1 Sistem Informasi Manajemen

Sistem informasi manajemen merupakan bagian dari sistem pendukung manajemen yang berfungsi untuk menyediakan laporan kepada manajer dari sebuah organisasi (Marakas & O'Brien, 2010). Masalah dalam sebuah organisasi dapat diidentifikasi dari laporan yang dikeluarkan oleh sistem informasi manajemen, sehingga langkah penyelesaian dapat segera diambil.

Data laporan merupakan data dari kinerja organisasi yang sekarang maupun yang sudah lalu dan biasanya laporan disusun dengan format tertentu (Arif, Rahaman & Uddin, 2012). Sistem informasi manajemen dapat digunakan sebagai tolak ukur kinerja dari organisasi, sehingga dapat dilakukan perbandingan dari waktu ke waktu. Sistem informasi menghasilkan keluaran yang berguna bagi manajer untuk mengambil keputusan berdasarkan laporan yang dibuat.

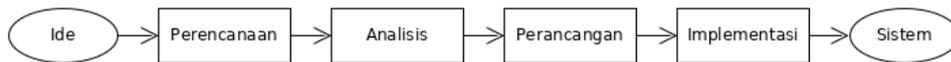
2.3.2 Sistem Pemrosesan Transaksi

Sistem pemrosesan transaksi merupakan sistem yang mendukung operasional dari suatu area sendiri pada organisasi dengan memproses transaksi pada organisasi tersebut (Arif, Rahaman & Uddin, 2012). Sistem pemrosesan transaksi digunakan untuk mencatat transaksi sehari-hari oleh pegawai ataupun manajer operasional yang masing-masing fungsinya dilakukan oleh operator yang berbeda (Amin, Alauddin & Azad, 2012). Informasi yang dihasilkan dari sistem pemrosesan transaksi berupa laporan kinerja yang menjadi masukan manajer untuk mengambil keputusan.

Menurut Marakas dan O'Brien (2010) terdapat dua cara yang dapat diterapkan pada sistem pemrosesan transaksi, *real-time* atau *batch*. Pemrosesan *real-time* dilakukan pada saat data masuk serta *batch* dilakukan ketika data sudah terkumpul pada jumlah tertentu lalu diproses secara bersamaan. Masing-masing cara dapat digunakan berdasarkan situasi dan kondisi di dalam lingkungan organisasi tersebut.

2.4 Systems Development Life Cycle

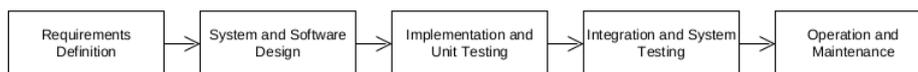
Sebuah sistem dapat disajikan kepada pengguna setelah melalui *systems development life cycle* yang mana melalui beberapa tahapan (Marakas & O'Brien, 2010). Tahapan dalam SDLC pada umumnya adalah: (1) identifikasi masalah, (2) analisis, (3) perancangan, (4) implementasi, dan (5) perawatan. Tahapan paling vital dan mendasar dari keseluruhan siklus ini adalah identifikasi masalah (Ali, 2017). Oleh karena itu analisis menjadi kunci dari sebuah pengembangan sistem informasi karena harus bisa mengidentifikasi kondisi bisnis, kesempatan dan aspek yang dapat ditingkatkan dari sebuah bisnis (Dennis, 2012).



Gambar 2.1 Siklus Hidup Pengembangan Perangkat Lunak

Sumber: Dennis (2012)

Metode *waterfall* merupakan salah satu metode SDLC yang dijalankan secara berurutan yang akan bergerak ke fase setelahnya atas persetujuan pemangku kepentingan (Dennis, 2012). Metode *waterfall* mengharuskan rancangan sistem selesai sebelum melakukan pemrograman, hal ini juga menjadi keuntungan ketika kebutuhan sudah jelas di awal dan tidak perlu melakukan perubahan ketika proses implementasi sedang berjalan.



Gambar 2.2 Metode Waterfall

Sumber: Sommerville (2016)

Menurut Sommerville (2016), metode *waterfall* memiliki tahapan-tahapan untuk menyelesaikan sebuah pengembangan sistem, yaitu:

1. *Requirements analysis and definition*, pada tahapan ini spesifikasi sistem didefinisikan secara lengkap berdasarkan kebutuhan pengguna.
2. *System and software design*, tahapan untuk merancang arsitektur sistem berdasarkan spesifikasi sistem sebelumnya. Rancangan perlu mencakup bagian-bagian utama sistem dan bagaimana hubungan antar bagian yang sudah dibuat.
3. *Implementation and unit testing*, rancangan diimplementasikan ke dalam unit-unit program untuk dirangkai menjadi satu. Setiap unit akan diuji untuk memastikan kesesuaiannya dengan spesifikasi sistem.
4. *Integration dan system testing*, unit program yang telah sesuai akan diintegrasikan pada tahapan ini. Keseluruhan sistem akan diuji berdasarkan kebutuhan pengguna sebelum sistem diberikan kepada pengguna.
5. *Operation and maintenance*, tahapan yang dilakukan selama masa pemakaian sistem oleh pengguna. Tahapan perawatan ini meliputi perbaikan ketika ada *error*, meningkatkan kualitas dan menambahkan fitur sistem.

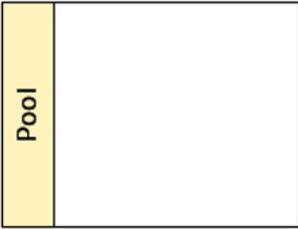
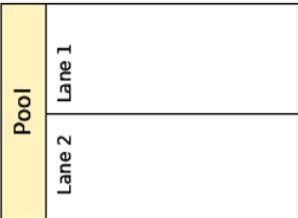
2.5 Business Process Model and Notation

Proses bisnis merupakan serangkaian aktivitas yang dilakukan untuk mencapai sebuah tujuan dan terbentuk atas aktivitas itu sendiri, kejadian pemicu, percabangan, aktor, serta keluaran yang diharapkan (Dumas et al., 2018). Proses bisnis juga perlu dipahami dan dikelola oleh organisasi agar meningkatkan efektivitas dan daya saing organisasi. Proses bisnis dapat dimodelkan untuk menggambarkan alur yang terjadi untuk setiap proses, agar dapat dikelola dengan baik. Pemodelan proses bisnis meningkatkan kecakapan sebuah rancangan melalui penggambaran alur, aktivitas, inisiator dan eksekutor untuk masing-masing aktivitas, urutan aktivitas, serta keluaran yang dihasilkan masing-masing aktivitas (Barjis, 2008). Pemodelan proses bisnis digambarkan dengan notasi-notasi yang sudah menjadi standar, notasi ini disebut dengan *business process model and notation*.

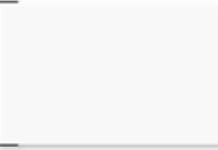
Sebuah standar notasi grafis yang dapat dimanfaatkan oleh analis bisnis untuk memodelkan proses dapat disebut dengan *business process model and notation* (Natschläger, 2011). BPMN dapat diposisikan sebagai pengganti *activity diagram* dalam pengembangan aplikasi sebab menurut Aagesen dan Krogstie (2015), BPMN diadaptasi dari berbagai notasi terutama *activity diagram*.

Elemen proses bisnis dimodelkan menggunakan notasi-notasi BPMN yang dapat digunakan untuk menggambarkan proses bisnis yang menjadi kebutuhan pengguna. Notasi-notasi yang dapat digunakan untuk memodelkan elemen proses bisnis dapat dilihat sebagai berikut:

Tabel 2.1 Notasi Elemen pada BPMN

Elemen	Deskripsi	Notasi
<i>Event</i>	<i>Event</i> menggambarkan bahwa ada sebuah “kejadian” pada sebuah proses atau koreografi.	
<i>Activity</i>	<i>Activity</i> menggambarkan sesuatu yang dilakukan oleh organisasi.	
<i>Gateway</i>	<i>Gateway</i> menggambarkan percabangan atau penyatuan sebuah <i>sequence flow</i> .	
<i>Sequence Flow</i>	<i>Sequence flow</i> digunakan sebagai penunjuk urutan aktivitas dalam sebuah proses.	
<i>Message Flow</i>	<i>Message flow</i> menunjukkan aliran antar <i>message</i> antara dua partisipan atau artinya berbeda <i>pool</i> .	
<i>Association</i>	<i>Association</i> digunakan sebagai penghubung informasi dan artefak yang ada dalam proses.	
<i>Pool</i>	<i>Pool</i> merepresentasikan partisipan dalam kolaborasi proses bisnis. <i>Pool</i> juga menjadi wadah untuk rangkaian aktivitas.	
<i>Lane</i>	<i>Lane</i> menggambarkan sub-partisi dari proses yang mengatur dan mengorganisir aktivitas.	
<i>Data Object</i>	<i>Data Object</i> menggambarkan apa yang menjadi kebutuhan ataupun hasil dari sebuah aktivitas.	

Tabel 2.1 Notasi Elemen pada BPMN (lanjutan)

Elemen	Deskripsi	Notasi
<i>Message</i>	<i>Message</i> merepresentasikan isi dari komunikasi antara dua partisipan.	
<i>Group</i>	<i>Group</i> mengelompokkan serangkaian elemen yang memiliki kategori yang sama.	
<i>Text Annotaion</i>	<i>Text anotation</i> memberikan informasi tambahan terkait proses bisnis.	

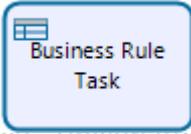
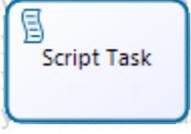
Sumber: Natschläger (2011)

Setiap aktivitas atau *task* di dalam proses bisnis juga dapat dibedakan jenis-jenisnya berdasarkan bagaimana aktivitas tersebut dikerjakan. Berikut jenis-jenis *task* yang dapat digunakan:

Tabel 2.2 Tipe Aktivitas pada BPMN

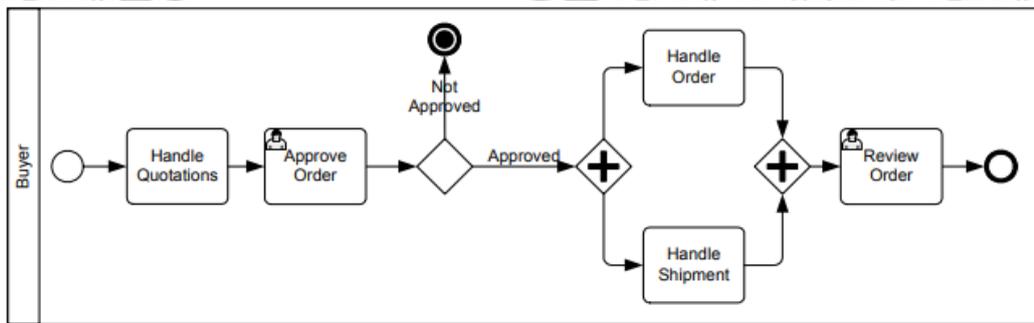
Jenis	Deskripsi	Notasi
<i>Service Task</i>	<i>Task</i> yang memanfaatkan layanan untuk menjalankannya, bisa <i>web service</i> atau aplikasi yang berjalan otomatis.	 Service Task
<i>Send Task</i>	<i>Task</i> yang mengirimkan pesan ke partisipan eksternal. <i>Task</i> dikatakan selesai ketika pesan sudah terkirim.	 Send Task
<i>Receive Task</i>	<i>Task</i> yang dirancang untuk menunggu pesan yang datang dari partisipan eksternal. <i>Task</i> dikatakan selesai ketika pesan sudah diterima.	 Receive Task
<i>User Task</i>	<i>Task</i> yang dilakukan oleh manusia yang dibantu perangkat lunak.	 User Task
<i>Manual Task</i>	<i>Task</i> yang dilakukan oleh manusia tanpa bantuan perangkat lunak ataupun <i>business rule engine</i> .	 Manual Task

Tabel 2.2 Tipe Aktivitas pada BPMN (lanjutan)

Jenis	Deskripsi	Notasi
<i>Business Rule Task</i>	Task yang memberikan masukan ke <i>business rule engine</i> untuk memproses sesuai dengan aturan bisnis yang sudah ditentukan oleh organisasi dan menerima keluarannya.	
<i>Script Task</i>	Task yang dilakukan oleh <i>business process engine</i> sesuai dengan <i>script</i> yang telah didefinisikan sebelumnya.	

Sumber: OMG (2011)

Menggunakan elemen-elemen BPMN, disertakan contoh pemodelan proses bisnis yang dapat dilihat pada gambar berikut ini:



Gambar 2.3 Contoh Pemodelan Proses Bisnis

Sumber: OMG (2011)

2.6 Unified Modelling Language

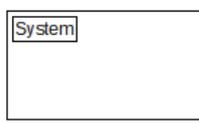
Unified Modelling Language (UML) merupakan sebuah standar yang membantu pengembang perangkat lunak sebagai cetak biru untuk perangkat lunak, layaknya cetak biru pada arsitektur bangunan (Pressman, 2010). UML diatur oleh *Object Management Group* (OMG) yang disesuaikan untuk pengembangan sistem berorientasi objek dan dikembangkan sebagai solusi untuk pemodelan yang memiliki masalah *bottle neck* (Habeeb, 2018). UML menggunakan beberapa diagram untuk memodelkan sebuah sistem, diantaranya: *use case diagram*, *sequence diagram*, dan *class diagram*.

2.6.1 Use Case Diagram

Pengguna dapat mengetahui fungsional dan fitur sistem dari sudut pandang pengguna, serta langkah-langkah untuk menjalankan fitur melalui spesifikasi *use case* (Pressman, 2010). *Use case diagram* memiliki karakteristik yang sederhana dan jelas untuk berkomunikasi dengan pengguna, oleh karena itu *use case diagram* dibuat pada awal perencanaan (Dennis, 2012). Oleh karena itu *use case diagram* merupakan alat bantu bagi perancang sistem untuk menggambarkan fungsional dan skenario sistem dengan sederhana kepada pengguna.

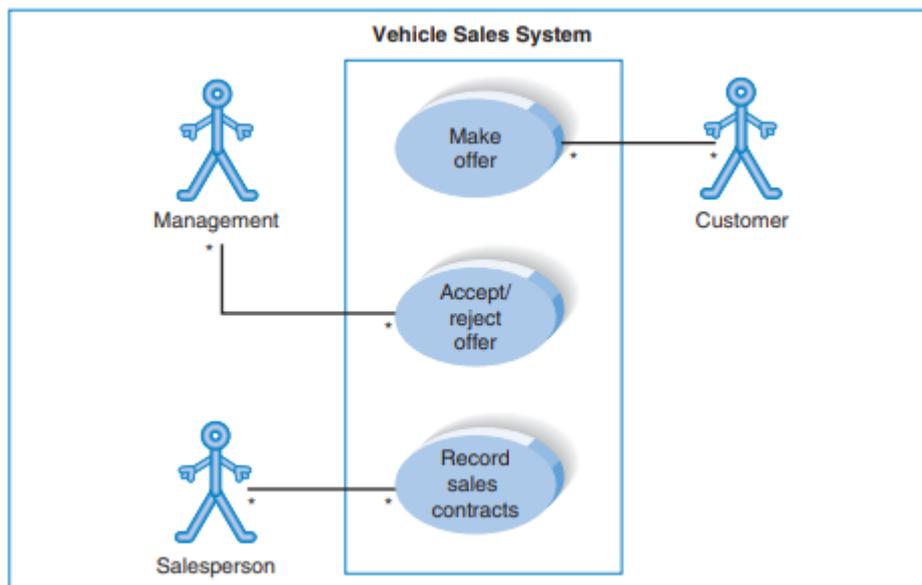
Elemen penyusun *use case diagram* digambarkan menggunakan notasi-notasi sebagai berikut:

Tabel 2.3 Notasi Elemen pada Use Case Diagram

Elemen	Deskripsi	Notasi
<i>Actor</i>	<i>Actor</i> merupakan bagian eksternal dari sistem yang dapat berupa orang atau sistem lainnya. <i>Actor</i> dapat diasosiasikan dengan <i>actor</i> lain dengan asosiasi <i>specialization/superclass</i> .	
<i>Use case</i>	<i>Use case</i> merupakan representasi dari fungsional sistem dan bisa digambarkan <i>extend</i> dan <i>include</i> use case lain.	
<i>System boundary</i>	<i>System boundary</i> merupakan gambaran dari <i>scope</i> dari sistem.	
<i>Association relationship</i>	<i>Association relationship</i> digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> .	

Sumber: Dennis (2012)

Menggunakan elemen-elemen *use case diagram*, disertakan contoh yang dapat dilihat pada gambar berikut ini:



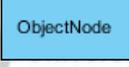
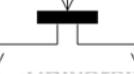
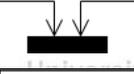
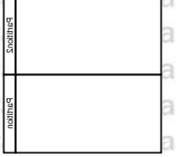
Gambar 2.4 Contoh Use Case Diagram

Sumber: Dennis (2012)

2.6.2 Activity Diagram

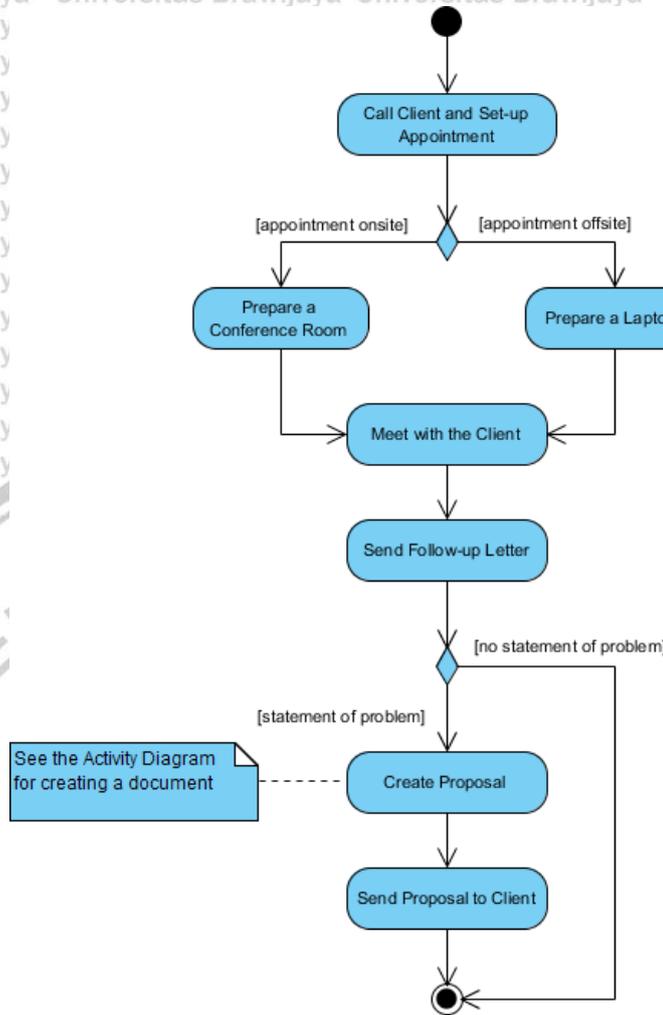
Activity diagram merupakan alur proses bisnis dari sebuah sistem dengan menunjukkan aliran dari aktivitas yang satu ke aktivitas yang lain (Pressman, 2010). Diagram ini bisa menunjukkan aktivitas yang dilakukan oleh aktor dan sistem dan menggambarkan alur untuk *use case*, sehingga diagram ini menjelaskan masing-masing *use case*. Berikut merupakan notasi yang ada pada *activity diagram*:

Tabel 2.4 Notasi Elemen pada Activity Diagram

Elemen	Deskripsi	Notasi
<i>Activity</i>	<i>Activity</i> merupakan notasi yang merepresentasikan serangkaian dari aksi.	
<i>Action</i>	<i>Action</i> merupakan tugas yang harus diselesaikan.	
<i>Control Flow</i>	<i>Control flow</i> merupakan urutan dari pengerjaan tugas.	
<i>Object Flow</i>	<i>Object flow</i> menggambarkan aliran objek dari satu aksi/aktivitas ke aksi/aktivitas yang lain.	
<i>Initial Node</i>	<i>Initial node</i> merupakan awal dari alur aktivitas atau aksi.	
<i>Activity Final Node</i>	<i>Activity final node</i> merupakan akhir dari semua rangkaian aktivitas.	
<i>Object Node</i>	<i>Object node</i> merupakan representasi dari object yang dihubungkan oleh <i>object flow</i> .	
<i>Decision Node</i>	<i>Decision node</i> merupakan kondisi yang menentukan arah alur dari objek atau aksi.	
<i>Merge Node</i>	<i>Merge node</i> merupakan notasi yang menggabungkan alur-alur yang terpisah oleh <i>decision node</i> .	
<i>Fork Node</i>	<i>Fork node</i> memisahkan alur menjadi rangkaian paralel.	
<i>Join Node</i>	<i>Join node</i> menggabungkan alur yang telah dipisahkan oleh <i>fork node</i> .	
<i>Swimlane and Partition</i>	<i>Swimlane and Partition</i> merupakan representasi aktor yang mengerjakan aktivitas atau aksi.	

Sumber: Visual Paradigm (n.d.)

Menggunakan elemen-elemen *activity diagram*, disertakan contoh yang dapat dilihat pada gambar berikut ini:



Gambar 2.5 Contoh Activity Diagram

Sumber: Visual Paradigm (n.d.)

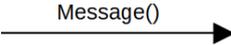
2.6.3 Sequence Diagram

Sequence diagram menunjukkan bagaimana sebuah *task* dapat diselesaikan dengan gambaran komunikasi antar objek secara runtut waktu (Pressman, 2010). *Sequence diagram* memuat pesan yang dikirimkan saat terjadi komunikasi antar objek sehingga bisa dilihat secara runtut perubahan dan perkembangan sistem dan partisipan yang terlibat (Dennis, 2012). *Sequence diagram* bisa dikatakan sebagai gambaran urutan interaksi antar objek dalam sebuah sistem berdasarkan urutan waktu.

Sequence diagram digambarkan dengan notasi yang menggambarkan aktor, kelas yang terkait dan *method* yang digunakan untuk berinteraksi antar kelas. Notasi *sequence diagram* disusun atas beberapa elemen yang digambarkan dalam notasi-notasi berikut:

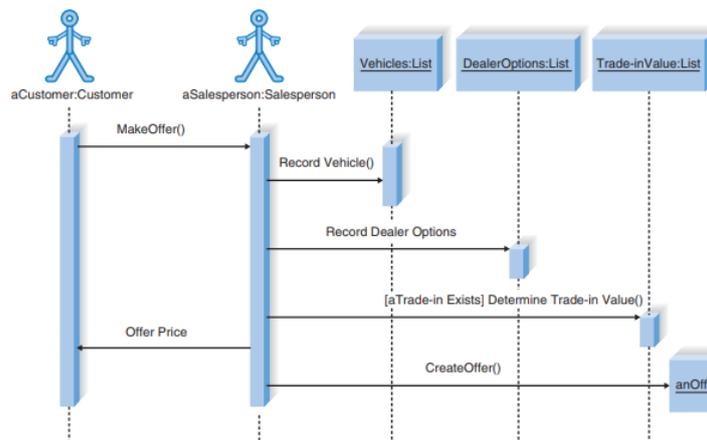


Tabel 2.5 Notasi Elemen pada Sequence Diagram

Elemen	Deskripsi	Notasi
Actor	Actor merupakan bagian eksternal dari sistem yang dapat berupa orang atau sistem lainnya. Actor mengirim atau menerima message saat berinteraksi dengan objek lain.	
Object	Object merupakan bagian yang berinteraksi dengan aktor ataupun objek lain melalui message.	Object : Class
Lifeline	Lifeline merepresentasikan hidup sebuah objek selama masih berinteraksi dalam sebuah sistem.	
Focus of control	Focus of control digunakan ketika sebuah objek yang mengirim atau menerima message dan diletakkan pada lifeline.	
Message	Message mengandung informasi yang dikirimkan dari satu objek ke objek lain.	
Object Destruction	Object Destruction adalah ketika sebuah objek tidak lagi dibutuhkan lagi dalam sistem.	

Sumber: Dennis (2012)

Menggunakan elemen-elemen sequence diagram, disertakan contoh yang dapat dilihat pada gambar berikut ini:



Gambar 2.6 Contoh Sequence Diagram

Sumber: Dennis (2012)

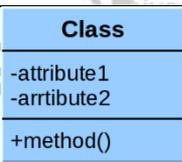
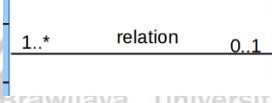
2.6.4 Class Diagram

Class diagram adalah diagram statis yang menggambarkan struktur dari sebuah sistem yang mana masing-masing kelas memiliki atribut dan operasi (Pressman, 2010). Atribut merupakan sesuatu yang melekat pada sebuah kelas dan dapat dimanfaatkan datanya oleh sistem. Operasi pada sebuah kelas menunjukkan apa yang dapat dilakukan oleh sebuah kelas. Penggambaran *class diagram* akan menjadi sangat rumit ketika mencakup keseluruhan sistem yang sebenarnya, oleh karena itu, Dennis (2012) mengatakan dalam hal ini perlu adanya penyederhanaan *class diagram* melalui dua pendekatan:

6. *Views*, yaitu penggambaran sederhana dari keseluruhan model diagram dengan membatasi informasi yang ditampilkan pada diagram
7. Pengelompokan, yaitu mengelompokkan kelas yang serupa ke dalam sebuah *package*. *Package* merupakan kelompok dari notasi yang secara logis memiliki kesamaan.

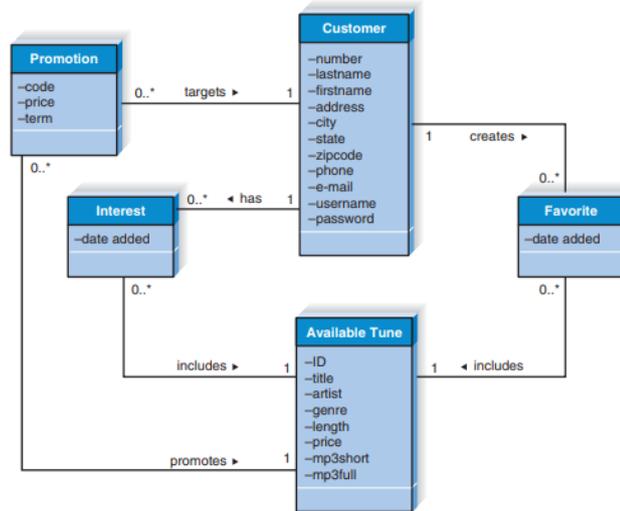
Elemen pada *class diagram* digambarkan menggunakan notasi-notasi sebagai berikut:

Tabel 2.6 Notasi Elemen pada Class Diagram

Elemen	Deskripsi	Notasi
Class	Class merepresentasikan orang, tempat atau apapun yang perlu disimpan informasinya agar dapat dimanfaatkan oleh sistem.	
Attribute	Attribute merupakan properti dari kelas yang mendeskripsikan kondisi dari kelas tersebut.	-attribute
Method	Method merupakan sebuah aksi atau fungsi yang dapat dilakukan oleh sebuah kelas.	+method()
Association	Association merepresentasikan relasi antar kelas yang dilengkapi dengan kata kerja.	

Sumber: Dennis (2012)

Menggunakan elemen-elemen *class diagram*, disertakan contoh yang dapat dilihat pada gambar berikut ini:



Gambar 2.7 Contoh Class Diagram

Sumber : Dennis (2012)

2.7 Database

Database pada sistem merujuk pada sekumpulan data yang dikelola menggunakan *database management system* (DBMS) dan data disimpan untuk jangka waktu yang lama (Garcia-Molina, Ullman & Widom, 2008). *Database* menyimpan data dengan jumlah besar untuk dikelola sistem dan mengubahnya menjadi informasi yang berguna bagi pengguna sistem. Menurut Bourgeois dan Bourgeois (2014), masing-masing data yang disimpan di dalam *database* seharusnya memiliki relasi antar satu data dengan yang lain, sehingga dapat diolah menjadi informasi oleh sistem. Umumnya *database* memiliki bentuk relasional, di mana data disimpan ke dalam tabel-tabel yang berhubungan satu sama lain. Satu tabel dalam *database* relasional mencerminkan satu objek tertentu yang memiliki atribut untuk menyimpan data-data dari objek terkait.

Perancangan *database* merupakan hal yang dibutuhkan sebelum menerapkan *database* untuk dipasangkan dengan sistem yang dibuat. Perancangan *database* membuat gambaran dari struktur sistem dari sisi data, sehingga pada saat melakukan prosesnya, sistem memiliki masukan dan keluaran yang terstruktur. *Database* dapat dirancang dengan memodelkan kelas-kelas dari objek di dunia nyata menggunakan pemodelan data.

2.7.1 Data Model

Model data merupakan model dari elemen data serta hubungan antar elemennya yang membentuk struktur data yang dibutuhkan untuk menjalankan proses bisnis yang ada (Marakas & O'Brien, 2010). Proses pemodelan data menggunakan *conceptual model* untuk menghasilkan *logical design* yang kemudian digunakan untuk membuat *physical design*. *Conceptual model* berupa terjemahan dari kebutuhan pengguna yang digambarkan dalam *entity relationship model*, yang pada penelitian ini berupa *class diagram*.

2.7.2 MySQL

MySQL merupakan DBMS yang bersifat *open source*, *relational* dan dapat berjalan dengan sistem *client/server* serta dijalankan dengan *structured query language* sebagai bahasa untuk mengakses *database* (MySQL, 2020). Menurut Manesh Chand (Chand, 2019) pada artikelnya, MySQL menjadi DBMS gratis dan *open source* yang paling populer di dunia. MySQL sering digunakan sebagai DBMS untuk masa pengembangan dari sistem yang baru dikembangkan karena kapabilitasnya yang cukup tinggi dan tidak berbayar, sehingga pengembangan tidak mengeluarkan biaya berlebih. MySQL juga dapat berjalan pada berbagai *platform*, sehingga penggunaannya jadi lebih fleksibel untuk pengembangan sistem.

2.8 Perancangan Antarmuka Pengguna

Antarmuka pengguna merupakan komponen sistem yang paling dekat dengan pengguna, oleh karena itu perancangannya perlu dibuat semenarik dan semudah mungkin untuk digunakan oleh pengguna (Marakas & O'Brien, 2010). Perancangan antarmuka pengguna didasarkan dari kebutuhan pengguna yang telah dianalisis, dan biasanya akan menerima masukan dari pengguna.

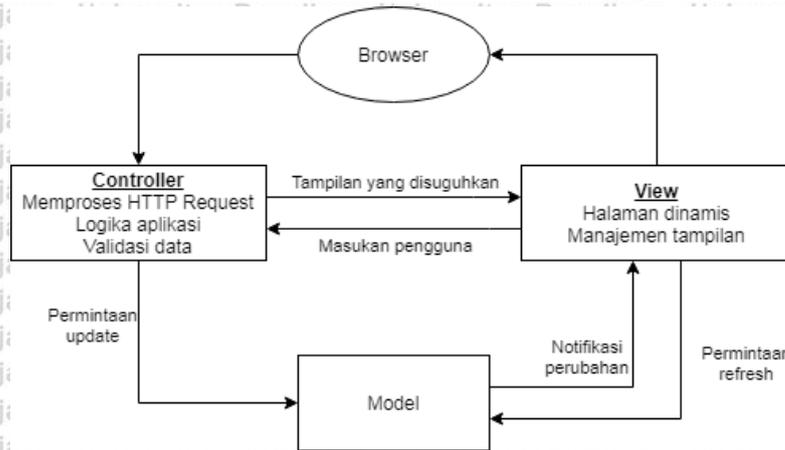
2.9 PHP

PHP merupakan bahasa pemrograman web yang bertujuan untuk mengembangkan halaman web dinamis lebih cepat (PHP Documentation Group, 2012). PHP dapat menjadi bahasa untuk mengembangkan aplikasi berbasis web, sehingga aplikasi memiliki kompatibilitas yang tinggi dan menjadi *cross platform*. Namun pada pengembangannya, PHP dapat memanfaatkan server lokal pada komputer. Hal ini membuat pengembangan menjadi lebih cepat karena tidak melalui jaringan internet.

Seiring waktu berjalan, PHP memiliki beberapa kerangka kerja yang memudahkan pengembang dalam membuat halamannya. Selain itu, menggunakan kerangka kerja PHP membuat aplikasi menjadi lebih cepat dan penggunaan memori yang lebih cepat bergantung pada kerangka kerja yang dipilih (Das & Saikia, 2016). Kerangka kerja PHP telah banyak beredar, seperti CodeIgniter, Symfony, Phalcon dan Laravel.

2.9.1 Model-View-Controller

Model-View-Controller atau disebut MVC menurut Sommerville (2016) merupakan pola pengembangan sistem yang membagi sistem menjadi tiga bagian yang menangani tiga hal berbeda dari sistem. *Model* menangani data, *view* menangani tampilan yang disuguhkan kepada pengguna serta *controller* menangani interaksi sistem dan proses logika sistem.

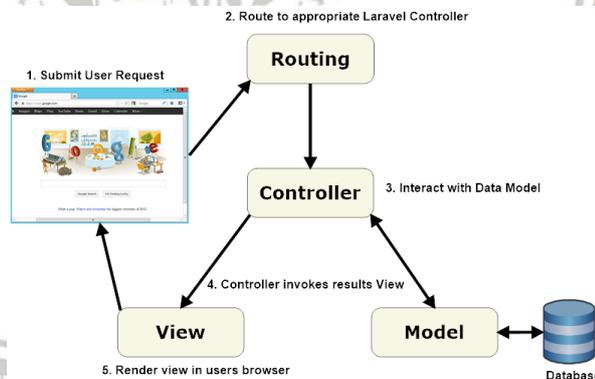


Gambar 2.8 Model-View-Controller

Sumber: Sommerville (2016)

2.9.2 Kerangka Kerja Laravel

Salah satu kerangka kerja PHP yang berfokus pada penulisan *syntax* dan tujuan utamanya untuk memudahkan pengembang saat mengembangkan web (Otwell, 2020). Laravel menjadi salah satu kerangka kerja yang populer karena kemudahan dan kecepatannya dalam mengakses server. Berdasarkan penelitian yang dilakukan oleh Das dan Saikira (2016), Laravel merupakan kerangka yang memiliki kecepatan eksekusi paling cepat namun penggunaan memori yang paling lama dibandingkan Codeigniter dan PHP murni. Penelitian tersebut menyatakan bahwa Laravel akan cocok ketika digunakan untuk proyek berskala besar.



Gambar 2.9 Cara Kerja Laravel

Sumber: Malhotra (2019)

2.10 Pengujian Sistem

Pengujian merupakan tahapan dalam SDLC untuk memastikan sistem bekerja sesuai dengan kebutuhan sistem yang telah dirancang di awal SDLC. Pengujian bertujuan untuk menemukan kesalahan sebanyak mungkin pada sistem yang telah dibuat dengan usaha yang sedikit mungkin (Pressman, 2010). Pengujian merupakan hal yang tak bisa dipisahkan dari pemrograman, pemrograman sistem dikatakan selesai ketika telah melewati serangkaian pengujian (Dennis, 2012).

2.10.1 Pengujian Validasi

Menurut Sommerville (2016) *validation testing* merupakan pengujian melalui rangkaian *test case* untuk memastikan sistem telah bekerja sesuai rancangan. *Test case* disiapkan untuk mencakup pengujian yang menyeluruh terhadap sistem, yaitu semua fungsi memenuhi kebutuhan, perilaku sistem telah sesuai dengan rancangan, konten disajikan dengan akurat, performa yang baik, dokumentasi yang benar serta kebutuhan lain telah terpenuhi (Pressman, 2010).

Tabel 2.7 Tabel Skenario Uji

Kode Fungsional	
Skenario Uji	
Nama Uji	
Kasus Uji	
Prakondisi	
Prosedur Uji	
Hasil yang Diharapkan	
Hasil Pengujian	
Status Pengujian	

Sumber: Diadaptasi dari Ryser dan Glinz (2000)

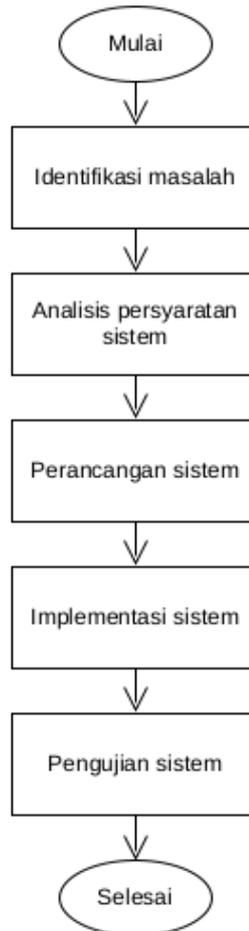
2.10.2 Pengujian Penerimaan Pengguna

User acceptance testing merupakan pengujian yang melibatkan pengguna dari sistem untuk menguji sistem sendiri sebelum digunakan oleh pengguna (Sommerville, 2016). Pengguna dapat memilih *test case* yang akan digunakan untuk diuji ke sistem (Leung & Wong, 1997). Pengujian penerimaan pengguna dilakukan sebelum sistem diterapkan di lingkungan pengguna. UAT menguji kesesuaian sistem yang dikembangkan dengan kebutuhan sistem (Dennis, 2012), dalam hal ini persyaratan sistem dapat digunakan sebagai acuan pengujian.

Menurut Naik dan Tripathy (2018), terdapat kriteria-kriteria untuk mengukur penerimaan pengguna. Kriteria-kriteria tersebut dapat dipilih dan disesuaikan dengan kebutuhan pengguna. Kriteria yang dimaksud antara lain: *functional correctness and completeness, data integrity, data conversion, usability, performance, timeliness, confidentiality and availability, installability and upgradability, scalability, dan documentation.*

BAB 3 METODOLOGI PENELITIAN

Penelitian ini dilakukan menggunakan metode *waterfall* untuk menyelesaikan penelitian terkait pengembangan sistem informasi pembayaran TV kabel. Pendekatan yang digunakan merupakan pendekatan berorientasi objek. Penelitian ini melakukan langkah-langkah yang terus berlanjut hingga kebutuhan pengguna dapat terpenuhi sepenuhnya. Tahapan yang dilakukan di dalam iterasi adalah perancangan, implementasi dan pengujian sistem.



Gambar 3.1 Diagram Alir Penelitian

3.1 Identifikasi Masalah

Tahapan ini merupakan tahapan untuk menggali masalah yang terjadi di Linda Cable melalui wawancara yang dilakukan kepada pemangku kepentingan di Linda Cable. Identifikasi masalah menghasilkan rumusan masalah dan tujuan yang dapat diangkat pada penelitian ini. Manfaat dan batasan penelitian didefinisikan juga pada tahapan ini.

3.2 Studi Literatur

Tahapan ini merupakan tahapan untuk mempelajari teori yang ada untuk mendukung dan memperkuat pelaksanaan penelitian. Teori didapatkan dari karya ilmiah, jurnal, buku dan penelitian sebelumnya yang terkait dengan penelitian ini. Dasar teori yang digunakan adalah pengembangan sistem informasi melalui *systems development life cycle* dengan metode *waterfall* sebagai dasar utama alur penelitian ini. Dasar perancangan dari sistem yang menggunakan diagram-diagram seperti *business process model and notation* sebagai notasi menggambarkan proses bisnis dan *unified modelling language* sebagai diagram yang menggambarkan struktur sistem. Dasar teori selanjutnya yaitu dasar untuk melakukan implementasi dari sistem yaitu dengan PHP sebagai bahasa pemrograman dan *database* sebagai penyimpanan data. Pengujian sistem didasari teori untuk melakukan pengujian validasi, efisiensi dan penerimaan pengguna untuk mengukur performa dari sistem.

3.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahapan untuk menyusun kebutuhan sistem berdasarkan wawancara kepada pemangku kepentingan terkait permasalahan yang telah diidentifikasi. Wawancara dilakukan untuk menggali proses bisnis yang dikerjakan di Linda Cable dengan rancangan wawancara kebutuhan pada LAMPIRAN A poin A.1. Proses bisnis tersebut akan diperbaiki dengan agar menjadi lebih efisien dengan interaksi sistem, sehingga dari proses bisnis perbaikan ini akan menghasilkan kebutuhan sistem.

Pada tahapan ini diagram-diagram dihasilkan menggunakan notasi-notasi yang telah menjadi standar perancangan kebutuhan, di antaranya:

- a. BPMN digambarkan untuk mengidentifikasi alur proses bisnis yang sedang berjalan dari sistem informasi musyawarah sebelumnya untuk disesuaikan dengan kondisi termutakhir.
- b. Notasi diagram UML yang berupa notasi *use case diagram* untuk menggambarkan fungsional sistem dan interaksi aktor kepada fungsional tersebut.

3.4 Perancangan Sistem

Perancangan sistem merupakan tahap pembuatan cetak biru dari sistem sesuai dengan analisis kebutuhan sistem. Perancangan sistem mencakup alur sistem dan struktur data di dalam sistem.

Perancangan dalam menggambarkan diagram-diagram sistem yang memanfaatkan notasi UML sebagai standar, di antaranya:

- a. *Sequence diagram* untuk menggambarkan alur sistem dalam berinteraksi antar objek.
- b. *Class diagram* untuk menggambarkan struktur data untuk diimplementasikan ke *database*.

3.5 Implementasi Sistem

Implementasi sistem merupakan tahapan untuk merealisasikan rancangan ke dalam bentuk sistem yang sebenarnya. Implementasi sistem memanfaatkan *database* dan kode program untuk membangun sistem.

- a. PHP merupakan bahasa pemrograman yang digunakan pada penelitian ini untuk mengimplementasikan aplikasi berbasis web. Versi PHP yang digunakan untuk pengembangan ini adalah PHP 7.2.
- b. Laravel merupakan alat yang digunakan pada penelitian ini sebagai kerangka kerja PHP untuk meningkatkan kualitas dari kode program yang dihasilkan. Laravel yang digunakan untuk penelitian ini adalah Laravel versi 7.28.
- c. *Database* dimanfaatkan sebagai penyimpan data yang akan diolah oleh sistem menggunakan PHP. DBMS yang digunakan untuk pengembangan ini adalah MySQL versi 5.7.

3.6 Pengujian Sistem

Pengujian sistem adalah tahapan melakukan serangkaian skenario uji untuk memastikan sistem telah bekerja sesuai dengan rancangan dan kebutuhan pengguna. Skenario uji dibuat untuk mendapatkan segala kemungkinan yang bisa terjadi pada sistem ketika skenario dijalankan. Pengujian yang dilakukan pada penelitian ini adalah pengujian validasi, pengujian efisiensi dan pengujian penerimaan pengguna.

3.7 Kesimpulan dan Saran

Tahapan ini merupakan tahapan penarikan kesimpulan terhadap penelitian yang telah dilakukan untuk masalah yang sudah diidentifikasi. Selain itu pada tahapan ini saran diberikan untuk pengembangan terhadap sistem lebih lanjut.

BAB 4 REKAYASA KEBUTUHAN

Pembahasan pada bab ini adalah tentang aturan penomoran dan rekayasa kebutuhan sistem. Rekayasa kebutuhan sistem merupakan tahapan yang merupakan tahapan awal dari siklus pengembangan sistem sebagai pendefinisian spesifikasi sistem berdasarkan kebutuhan pengguna (Sommerville, 2016). Analisis kebutuhan sistem digali berdasarkan wawancara yang dilakukan kepada pemangku kepentingan LAMPIRAN B poin B.1. Analisis kebutuhan sistem dilakukan untuk memperoleh kebutuhan organisasi atau pemangku kepentingan dan dianalisis untuk menjadi kebutuhan sistem. Hasil dari penggalan ini adalah penggambaran proses bisnis, identifikasi fitur, kebutuhan fungsional serta diagram *use case*.

4.1 Aturan Penomoran

Aturan penomoran memberikan cara memberikan nomor yang terdiri dari angka serta huruf. Nomor mewakili kode artefak sistem yang bertujuan untuk memudahkan penelusuran artefak sistem (Akachain.io, 2020). Berikut aturan penomoran pada dokumen penelitian ini yang berlaku untuk semua bab:

AA-SIP-XXX

Keterangan:

1. **AA** : Representasi kode jenis artefak
2. **SIP** : Representasi nama program – Sistem Informasi Pembayaran
3. **XXX** : Representasi nomor urut artefak

Kode jenis artefak dapat didefinisikan sebagai berikut:

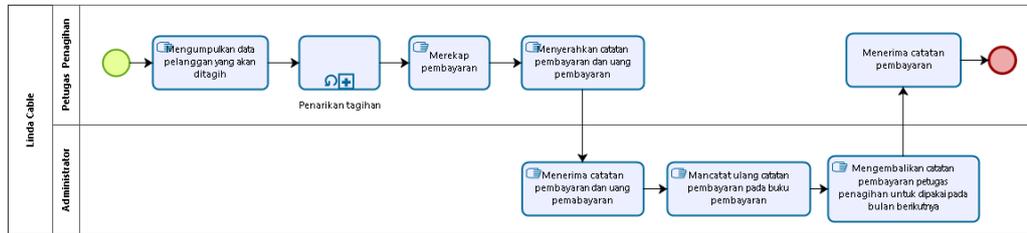
1. **PBA** : Proses bisnis saat ini (*as-is*)
2. **PBT** : Proses bisnis rekomendasi (*to-be*)
3. **IM** : Identifikasi masalah
4. **APBT** : Aktivitas proses bisnis rekomendasi (*to-be*)
5. **IF** : Identifikasi fitur
6. **PF** : Persyaratan fungsional
7. **PNF** : Persyaratan nonfungsional
8. **UC** : *Uce case*
9. **UCS** : *Use case scenario*
10. **TS** : Skenario uji/*test scenario*

4.2 Proses Bisnis Saat Ini (As-Is)

Proses bisnis merupakan serangkaian aktivitas yang menggambarkan alur untuk mencapai tujuan dan proses bisnis dimodelkan menggunakan BPMN sebagai standar notasinya (Dumas et al., 2018; Natschläger, 2011). Berdasarkan hasil wawancara yang dilakukan pada LAMPIRAN B poin B.1, dapat diidentifikasi proses bisnis yang berkaitan dengan kebutuhan pemangku kepentingan.

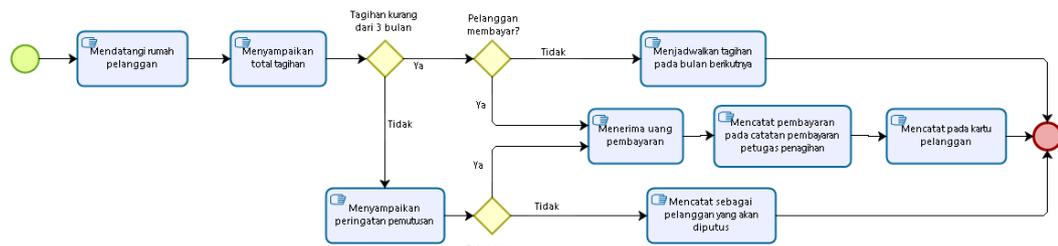
4.2.1 Proses Bisnis Pencatatan Pembayaran Saat Ini (As-Is)

Proses bisnis untuk pencatatan pembayaran dilakukan secara rutin setiap bulan. Proses bisnis pencatatan pembayaran dimodelkan pada Gambar 4.1 dan Gambar 4.2 berikut.



Gambar 4.1 Model Proses Bisnis Pencatatan Pembayaran Saat Ini (As-Is)

Proses bisnis pencatatan pembayaran yang dimodelkan pada Gambar 4.1 memiliki subproses bisnis untuk penarikan tagihan yang digambarkan pada Gambar 4.2 berikut ini:



Gambar 4.2 Model Subproses Bisnis Penarikan Tagihan Saat Ini (As-Is)

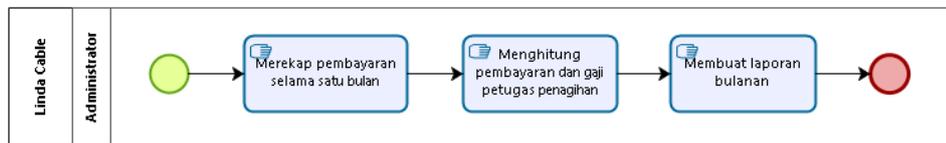
Proses bisnis melibatkan administrator dan petugas penagihan. Proses diinisiasi oleh petugas penagihan. Gambar 4.1 menunjukkan bahwa penarikan tagihan berulang, artinya proses penarikan tagihan yang dimodelkan pada Gambar 4.2 dilakukan berulang hingga penarikan tagihan mencapai targetnya. Berikut penjelasan alur proses bisnis pencatatan pembayaran:

1. Petugas penagihan mengumpulkan daftar data pelanggan yang akan ditagih dari catatan bulan-bulan sebelumnya.
2. Petugas penagihan melakukan penarikan tagihan yang berulang hingga seluruh pelanggan selesai ditagih. Sub proses penarikan tagihan sebagai berikut:
 - a. Petugas penagihan mendatangi rumah pelanggan yang masuk dalam daftar tagihan.

3. Jika pelanggan belum pernah menyambung TV kabel, maka pelanggan akan diberikan formulir data pelanggan dan persetujuan. Kemudian data pelanggan dicatat pada buku daftar.
4. Jika pelanggan pernah menyambung sebelumnya, maka akan diperiksa tagihan yang belum dibayar pada buku daftar pelanggan bagian pembayaran. Jika ada tagihan belum dibayar maka pelanggan harus menyelesaikan dahulu semua tagihan tersebut.
5. Administrator memberikan data pelanggan terbaru kepada petugas penagih.

4.2.3 Proses Bisnis Pembuatan Laporan Bulanan (As-Is)

Proses bisnis ini dijalankan secara setiap bulan setelah proses bisnis pencatatan pembayaran selesai dilakukan. Pemodelan proses bisnis pembuatan laporan bulanan dapat dilihat pada



Powered by
bizagi
Modeler

Gambar 4.4 Model Proses Bisnis Pembuatan Laporan Bulanan Saat Ini (As-Is)

Proses bisnis pembuatan laporan bulanan dilakukan administrator membuat laporan dari pembayaran yang telah ditagihkan selama satu bulan. Berikut merupakan penjelasan alur proses bisnis pembuatan laporan bulanan:

1. Administrator merekap pembayaran selama satu bulan.
2. Administrator menghitung pembayaran dan gaji petugas penagihan.
3. Administrator membuat laporan bulanan berdasarkan perhitungan.

4.3 Analisis Permasalahan Proses Bisnis Saat Ini (As-Is)

Proses bisnis saat ini pada Linda Cable yang dimodelkan pada Gambar 4.1 hingga Gambar 4.4 dapat diidentifikasi pada Tabel 4.1.

Tabel 4.1 Identifikasi Proses Bisnis Saat Ini

Kode Proses Bisnis Saat Ini	Nama Proses Bisnis Saai Ini
PBA-SIP-01	Pencatatan Pembayaran
PBA-SIP-02	Pendaftaran Pelanggan
PBA-SIP-03	Pembuatan Laporan Bulanan

Proses bisnis saat ini yang sudah diidentifikasi terdapat beberapa masalah yang terjadi, sehingga dapat dilakukan analisa terhadap permasalahan, dampak dan solusi yang dapat diberikan untuk menyelesaikan masalah-masalah tersebut dampak dari masalah tersebut dapat diminimalisasi. Hasil analisis permasalahan yang dapat dilihat pada Tabel 4.2 berikut ini:

Tabel 4.2 Tabel Analisis Permasalahan

Masalah	<ol style="list-style-type: none"> 1. Duplikasi pencatatan pembayaran. 2. Data pelanggan terbaru hanya dapat diakses pada buku daftar pelanggan yang disimpan di kantor. 3. Kalkulasi data tagihan dilakukan secara manual. 4. Catatan pembayaran hanya dapat ditelusuri secara manual.
Dampak	<ol style="list-style-type: none"> 1. Proses pencatatan pembayaran menjadi lama dan berpotensi terjadi kesalahan pencatatan. 2. Perlu dilakukan pembaruan secara berkala antara data pelanggan milik petugas penagihan dan dan buku daftar pelanggan di kantor. 3. Proses kalkulasi tagihan lebih lama dan berpotensi untuk terjadi kesalahan kalkulasi. 4. Penelusuran data pembayaran lebih sulit dan lama.
Solusi	<ol style="list-style-type: none"> 1. Membangun sistem informasi yang dapat mencatat pembayaran yang dapat mencatat pembayaran dari petugas penagihan dan administrator. 2. Membangun sistem informasi yang dapat mengelola data pelanggan dengan satu pusat data. 3. Membangun sistem informasi yang dapat menghitung data tagihan secara otomatis. 4. Membangun sistem informasi yang dapat menyimpan dan menelusuri catatan pembayaran.

Analisis permasalahan pada Tabel 4.2 menunjukkan masalah, dampak dan solusi yang mungkin untuk diterapkan. Oleh karena itu disusun identifikasi masalah pada sebagai acuan untuk solusi rancangan sistem informasi yang dibuat

Tabel 4.3 Identifikasi Masalah

Kode Masalah	Deskripsi Masalah	Kode Proses Bisnis Saat Ini
IM-SIP-01	Duplikasi proses pencatatan pembayaran	PBA-SIP-01
IM-SIP-02	Data pelanggan terbaru hanya dapat diakses pada buku daftar pelanggan yang disimpan di kantor.	PBA-SIP-02
IM-SIP-03	Kalkulasi data tagihan dilakukan secara manual.	PBA-SIP-01, PBA-SIP-02, PBA-SIP-03
IM-SIP-04	Catatan pembayaran hanya dapat ditelusuri secara manual.	PBA-SIP-01, PBA-SIP-02

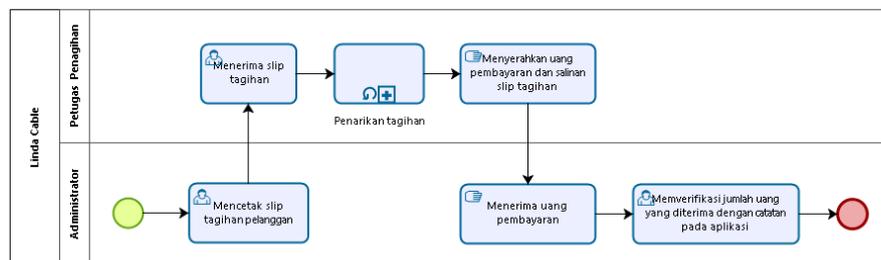
4.4 Rancangan Proses Bisnis (*To-Be*)

Berdasarkan analisis permasalahan pada Tabel 4.2, dapat dimodelkan proses bisnis perbaikan yang dirancang untuk diterapkan. Rancangan proses bisnis dimodelkan dengan interaksi aktor dengan sistem, sehingga aktivitas proses bisnis dapat dimodifikasi baik melalui penambahan aktivitas maupun pengurangan aktivitas.

4.4.1 Rancangan Proses Bisnis Pencatatan Pembayaran (*To-Be*)

Masalah pada proses bisnis pencatatan pembayaran saat ini adalah pengulangan pencatatan dari petugas penagihan kepada administrator, sehingga memakan waktu lebih lama. Berdasarkan wawancara yang dilakukan pada LAMPIRAN B poin B.1, dibutuhkan proses bisnis pembayaran yang dapat dilakukan di kantor Linda Cable. Oleh karena itu terdapat dua proses bisnis yang dimodelkan untuk pencatatan pembayaran di kantor dan pembayaran yang ditagihkan ke rumah pelanggan.

Proses bisnis pencatatan pembayaran oleh petugas penagihan yang dimodelkan pada Gambar 4.5 dan Gambar 4.6 merepresentasikan alur yang harus dikerjakan oleh petugas penagihan secara rutin setiap bulan.



Gambar 4.5 Model Rancangan Proses Bisnis Pencatatan Pembayaran oleh Petugas Penagihan (*To-Be*)

Proses bisnis pencatatan pembayaran oleh petugas penagihan pada Gambar 4.5 memiliki subproses bisnis penarikan tagihan yang dimodelkan pada Gambar 4.6 berikut ini:

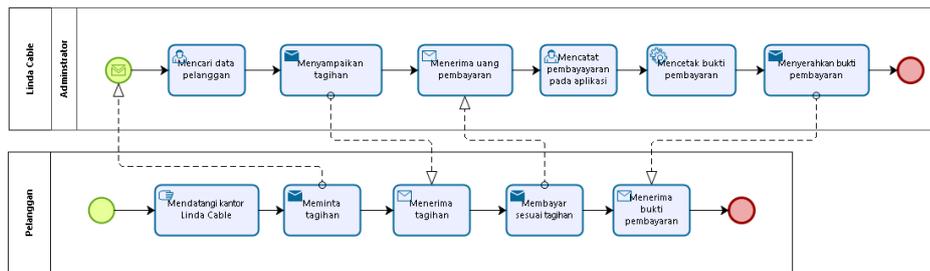


Gambar 4.6 Model Rancangan Subproses Penarikan Tagihan (*To-Be*)

Perbaikan proses bisnis ini memodelkan alur proses bisnis untuk petugas penagihan yang berinteraksi dengan aplikasi pembayaran. Interaksi aktor mengubah aktivitas pencatatan yang sebelumnya dilakukan oleh administrator dan petugas penagihan menjadi pencatatan pembayaran pada aplikasi pembayaran. Kalkulasi tagihan pelanggan dilakukan oleh aplikasi pembayaran sehingga status tagihan akan diperbarui secara otomatis. Berikut merupakan penjelasan alur proses pencatatan pembayaran oleh petugas penagihan:

1. Administrator mencetak slip tagihan pelanggan sesuai dengan petugas penagihan yang bertanggung jawab.
2. Petugas penagihan menerima slip tagihan yang telah dicetak.
3. Petugas penagihan menyampaikan slip tagihan yang berulang hingga semua tagihan di aplikasi telah disampaikan. Sub proses penarikan tagihan sebagai berikut:
 - a. Petugas penagihan mendatangi rumah pelanggan.
 - b. Petugas penagihan menyampaikan tagihan kepada pelanggan
 - c. Jika pelanggan membayar, maka petugas penagihan menerima uang pembayaran, mencatat pada aplikasi pembayaran dan mencatat pada kartu pelanggan. Jika tidak membayar maka proses selesai.
4. Petugas penagihan menyerahkan uang pembayaran kepada administrator dan salinan slip tagihan yang sudah ditandatangani.
5. Administrator menerima uang pembayaran.
6. Administrator memverifikasi jumlah uang yang diterima dan yang tercatat pada aplikasi oleh petugas penagihan.

Proses bisnis selanjutnya yang digambarkan pada Gambar 4.7 merupakan proses bisnis yang dirancang berdasarkan kebutuhan pemangku kepentingan untuk mengadakan proses pembayaran di kantor.



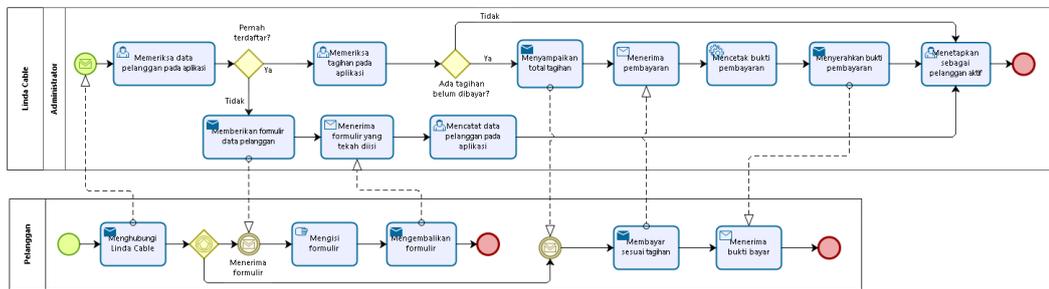
Gambar 4.7 Model Rancangan Proses Bisnis Pencatatan Pembayaran di Kantor (To-Be)

Proses bisnis pencatatan pembayaran di kantor melibatkan interaksi administrator dengan aplikasi pembayaran. Proses bisnis diinisiasi oleh pelanggan yang mendatangi kantor Linda Cable untuk meminta tagihan. Berikut penjelasan alur proses bisnis pencatatan pembayaran di kantor:

1. Pelanggan mendatangi kantor Linda Cable dan meminta tagihan.
2. Administrator mencari data pelanggan pada sistem dan menyerahkan tagihan.
3. Pelanggan membayar sesuai tagihan.
4. Administrator mencatat pada aplikasi dan mencetak surat tagihan.
5. Administrator menandatangani surat tagihan bersama pelanggan.
6. Administrator menyerahkan surat tagihan yang telah ditandatangani sebagai bukti transaksi dan menyimpan lembar arsip.

4.4.2 Rancangan Proses Bisnis Pendaftaran Pelanggan

Proses bisnis pendaftaran pelanggan saat ini terdapat masalah pencatatan data pelanggan yang mengharuskan petugas penagihan mencatat ulang data tagihan pelanggan baru. Proses bisnis pendaftaran pelanggan perbaikan dimodelkan pada Gambar 4.8.



Gambar 4.8 Model Rancangan Proses Bisnis Pendaftaran Pelanggan (To-Be)

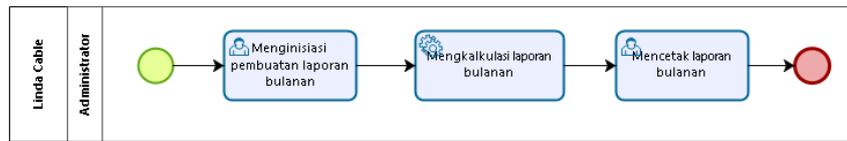
Perbaikan proses bisnis pendaftaran pelanggan yang digambarkan pada Gambar 4.8 diperbaiki dengan membuat interaksi antara administrator dengan sistem, sehingga pemeriksaan pelanggan yang pernah terdaftar dan pemeriksaan tagihan yang belum dibayar lebih cepat menggunakan bantuan sistem. Perbaikan juga pada data pelanggan terbaru tidak perlu disalin oleh petugas penagihan. Berikut penjelasan dari alur proses bisnis pendaftaran pelanggan:

1. Pelanggan menginisiasi dengan mendatangi kantor Linda Cable.
2. Administrator memeriksa data pelanggan pada buku daftar pelanggan
3. Jika pelanggan belum pernah menyambung TV kabel, maka pelanggan akan diberikan formulir data pelanggan dan persetujuan. Kemudian data pelanggan dicatat pada buku daftar.
4. Jika pelanggan pernah menyambung sebelumnya, maka akan diperiksa tagihan yang belum dibayar pada buku daftar pelanggan bagian pembayaran. Jika ada tagihan belum dibayar maka pelanggan harus menyelesaikan dahulu semua tagihan tersebut.
5. Administrator menerapkan status aktif pada pelanggan.

4.4.3 Rancangan Proses Bisnis Pembuatan Laporan Bulanan

Proses bisnis pembuatan laporan bulanan saat ini terdapat masalah terkait kalkulasi data pembayaran yang dilakukan manual, sehingga proses lebih lama dan

berpotensi untuk salah perhitungan. Proses bisnis pembuatan laporan bulanan perbaikan dimodelkan pada Gambar 4.9.



Powered by bizagi Modeler

Gambar 4.9 Model Rancangan Proses Bisnis Pembuatan Laporan Bulanan (To-Be)

Proses bisnis pembuatan laporan bulanan dikerjakan oleh administrator. Administrator menginisiasi proses ini dan selanjutnya kalkulasi dilakukan oleh aplikasi pembayaran. Berikut penjelasan alur proses bisnis pelaporan bulanan:

1. Administrator menginisiasi sistem untuk menghitung laporan bulanan
2. Administrator mencetak laporan bulanan yang dihasilkan oleh sistem.

4.5 Analisis Hasil Perbaikan Proses Bisnis

Proses bisnis rekomendasi yang telah dimodelkan menjadi hasil perbaikan untuk proses bisnis saat ini. Identifikasi pada Tabel 4.4 berikut ini merupakan identifikasi rancangan proses bisnis terhadap proses bisnis saat ini.

Tabel 4.4 Identifikasi Rancangan Proses Bisnis

Kode Proses Bisnis Rekomendasi	Nama Proses Bisnis	Kode Proses Bisnis Saat Ini
PBT-SIP-01	Pencatatan Pembayaran oleh Petugas Penagihan	PBA-SIP-01
PBT-SIP-02	Pencatatan Pembayaran di Kantor	PBA-SIP-01
PBT-SIP-03	Pendaftaran Pelanggan	PBA-SIP-02
PBT-SIP-04	Pembuatan Laporan Bulanan	PBA-SIP-03

Tabel 4.4 tersebut memuat daftar proses bisnis rekomendasi yang mendapatkan perbaikan pada aktivitas di dalamnya. Aktivitas pada rancangan proses bisnis dapat menyelesaikan masalah yang sebelumnya telah diidentifikasi sebelumnya pada Tabel 4.3. Analisis aktivitas pada rancangan proses bisnis terhadap masalah yang diidentifikasi dapat dilihat pada Tabel 4.5.

Tabel 4.5 Analisis Aktivitas Rancangan Proses Bisnis

Kode Aktivitas Rekomendasi	Kode Proses Bisnis Rekomendasi	Lane	Nama Aktivitas Rekomendasi	Kode Masalah yang Diselesaikan	Keterangan
APBT-SIP-01	PBT-SIP-01	Petugas Penagihan	Memeriksa data tagihan pelanggan yang akan ditagih pada aplikasi	IM-SIP-04	Ketika mengakses aplikasi pembayaran, petugas penagihan dapat mengakses tagihan yang sudah disediakan oleh aplikasi pembayaran sehingga proses penelusuran lebih cepat dan mudah.
APBT-SIP-02	PBT-SIP-01	Petugas Penagihan	Mencatat pembayaran pada aplikasi	IM-SIP-01	Petugas penagihan mencatat pembayaran pada aplikasi pembayaran sehingga catatan tersimpan pada aplikasi. Hal ini menghindari duplikasi pencatatan pembayaran.
APBT-SIP-03	PBT-SIP-01	Administrator	Memverifikasi jumlah uang yang diterima dengan catatan pada aplikasi	IM-SIP-01	Administrator melakukan pemeriksaan jumlah yang telah dikalkulasi oleh aplikasi pembayaran, sehingga tidak ada duplikasi proses pencatatan.
APBT-SIP-04	PBT-SIP-02	Administrator	Mencari data pelanggan	IM-SIP-04	Administrator menelusuri catatan pembayaran yang sudah dilakukan untuk mengetahui data tagihan pelanggan terkait, sehingga penelusuran menjadi lebih cepat.
APBT-SIP-05	PBT-SIP-02	Administrator	Mencatat pembayaran pada aplikasi	IM-SIP-01	Pencatatan pembayaran pada aplikasi pembayaran menghindari duplikasi proses pencatatan pembayaran sehingga dapat diakses oleh setiap penggunanya.
APBT-SIP-06	PBT-SIP-03	Administrator	Memeriksa data pelanggan pada aplikasi	IM-SIP-02	Data pelanggan yang diakses pada aplikasi pembayaran merupakan data terbaru, sehingga tidak perlu proses pembaruan berkala untuk mendapatkan data pelanggan terbaru.

Tabel 4.5 Analisis Aktivitas Rancangan Proses Bisnis (lanjutan)

APBT-SIP-07	PBT-SIP-03	Administrator	Mencatat data pelanggan pada aplikasi	IM-SIP-02	Pencatatan data pelanggan pada aplikasi memberikan data pelanggan terbaru kepada semua yang mengakses aplikasi pembayaran.
APBT-SIP-08	PBT-SIP-04	Administrator	Menghitung laporan bulanan	IM-SIP-03	Kalkulasi data tagihan yang dilakukan oleh sistem mempercepat proses pembuatan laporan dan meminimalisasi kesalahan hitung yang dilakukan oleh administrator.



4.6 Identifikasi Pengguna

Identifikasi proses bisnis perbaikan yang telah dilakukan pada Tabel 4.5 terdapat *lane* sebagai aktor yang melakukan aktivitas tersebut, sehingga dapat diidentifikasi pengguna yang berinteraksi dengan sistem.

Tabel 4.6 Identifikasi Pengguna

Pengguna	Deskripsi
Administrator	Pengguna yang memiliki hak terhadap pengelolaan data pelanggan, pembayaran, bukti transaksi dan pelaporan.
Petugas Penagihan	Pengguna yang memiliki hak pengelolaan pembayaran, bukti transaksi dan akses terhadap informasi pelanggan.

4.7 Identifikasi Fitur

Fitur dari sistem informasi yang dikembangkan dapat dilihat aktivitas proses bisnis perbaikan yang melibatkan aplikasi pembayaran pada Tabel 4.5 dan LAMPIRAN B poin B.1. Fitur dari sistem informasi ini merupakan fitur yang dapat menyelesaikan masalah terkait. Hasil identifikasi fitur sistem terdapat pada Tabel 4.7 berikut ini.

Tabel 4.7 Identifikasi Fitur

Kode Fitur	Deskripsi Fitur	Kode Aktivitas Rekomendasi
IF-SIP-01	Sistem dapat mengenali identitas pengguna yang masuk ke dalam sistem dan membedakan akses berdasarkan peran pengguna.	APBT-SIP-03, APBT-SIP-08
IF-SIP-02	Sistem dapat mengelola data pelanggan, di antaranya: menambah, melihat, memperbaiki, menghapus dan mencari data pelanggan. Sistem dapat mengelola daftar penagihan, di antaranya: melihat daftar penagihan, menambah dan menghapus pelanggan dari daftar penagihan.	APBT-SIP-04, APBT-SIP-06, APBT-SIP-07
IF-SIP-03	Sistem dapat mengelola tagihan dan pembayaran, di antaranya: menampilkan tagihan berdasarkan statusnya, memverifikasi tagihan, mencetak slip tagihan dan mencatat pembayaran tagihan pelanggan. Pengguna dapat menggunakan sistem saat mobilitas tinggi.	APBT-SIP-01, APBT-SIP-02, APBT-SIP-03, APBT-SIP-05
IF-SIP-04	Sistem dapat menghitung tagihan dan pembayaran untuk membuat laporan bulanan.	APBT-SIP-08

4.8 Identifikasi Persyaratan Fungsional

Persyaratan fungsional sistem dapat diidentifikasi berdasarkan fitur yang telah diidentifikasi pada Tabel 4.7. Persyaratan fungsional didefinisikan pada Tabel 4.8 berikut ini:

Tabel 4.8 Identifikasi Persyaratan Fungsional

Kode Persyaratan Fungsional	Deskripsi	Kode Fitur
PF-SIP-01	Sistem dapat mengautentikasi pengguna.	IF-SIP-01
PF-SIP-02	Sistem dapat membedakan akses pengguna.	IF-SIP-01
PF-SIP-03	Sistem dapat melakukan pencarian data pelanggan.	IF-SIP-02
PF-SIP-04	Sistem dapat menampilkan data pelanggan.	IF-SIP-02
PF-SIP-05	Sistem dapat menambah data pelanggan.	IF-SIP-02
PF-SIP-06	Sistem dapat memperbaiki data pelanggan.	IF-SIP-02
PF-SIP-07	Sistem dapat menghapus data pelanggan.	IF-SIP-02
PF-SIP-08	Sistem dapat menampilkan daftar penagihan.	IF-SIP-02
PF-SIP-09	Sistem dapat menambahkan data pelanggan ke dalam daftar penagihan.	IF-SIP-02
PF-SIP-10	Sistem dapat menghapus data pelanggan ke dalam daftar penagihan.	IF-SIP-02
PF-SIP-11	Sistem dapat menampilkan data tagihan pelanggan.	IF-SIP-03
PF-SIP-12	Sistem dapat mencetak slip tagihan.	IF-SIP-03
PF-SIP-13	Sistem dapat mencatat pembayaran tagihan.	IF-SIP-03
PF-SIP-14	Sistem dapat memverifikasi pembayaran tagihan.	IF-SIP-03
PF-SIP-15	Sistem dapat menghitung laporan bulanan.	IF-SIP-04

4.9 Identifikasi Persyaratan Non-fungsional

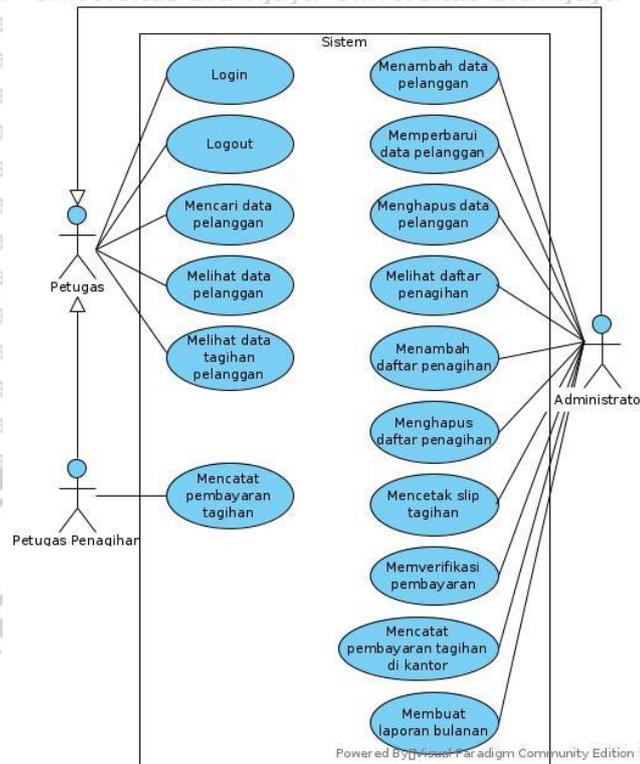
Persyaratan non-fungsional merupakan persyaratan yang diterapkan pada sistem sebagai pemenuhan kebutuhan pengguna melalui perilaku sistem. Persyaratan non-fungsional didefinisikan pada:

Kode Persyaratan Non-fungsional	Deskripsi	Kode Fitur
PNF-SIP-01	Sistem dapat digunakan pada <i>platform desktop</i> dan <i>mobile device</i> .	IF-SIP-03
PNF-SIP-02	Sistem dapat mencatat waktu, penerima dan verifikator pembayaran	IF-SIP-02

4.10 Use Case Diagram

Diagram *use case* merupakan diagram yang dapat menggambarkan fungsional sistem dari sudut pandang pengguna (Pressman, 2010). Pengguna yang telah diidentifikasi pada Tabel 4.6 dapat digambarkan sebagai aktor di dalam diagram *use case* dan persyaratan fungsional yang telah diidentifikasi pada Tabel 4.8 dapat

digambarkan *use case* atau fitur dari sistem. Diagram *use case* dapat dilihat pada Gambar 4.10 berikut:



Gambar 4.10 Diagram Use Case Sistem Informasi Pembayaran

Diagram *use case* memiliki aktor yang berinteraksi dengan sistem dengan *use case* sebagai kegiatannya. Identifikasi aktor yang terdapat pada Gambar 4.10 didefinisikan pada Tabel 4.9 berikut ini.

Tabel 4.9 Identifikasi Aktor Use Case

Nama Aktor	Deskripsi Aktor
Petugas	Petugas merupakan aktor <i>parent</i> yang memiliki hak akses untuk bisa <i>login</i> dan <i>logout</i> , melihat data pelanggan, mencari data pelanggan dan melihat data tagihan pelanggan.
Petugas Penagihan	Petugas penagihan merupakan aktor turunan dari petugas yang memiliki hak untuk mencatat pembayaran tagihan pelanggan.
Administrator	Administrator merupakan aktor turunan dari petugas yang memiliki hak akses milik petugas dengan hak akses khusus administrator, yaitu: menambah, memperbarui, menghapus data pelanggan, menambah pelanggan ke daftar penagihan, menghapus pelanggan dari daftar penagihan, memverifikasi pembayaran tagihan, mencetak slip tagihan, membuat laporan bulanan serta memperbarui status tagihan di kantor.

Diagram *use case* pada Gambar 4.10 menggambarkan apa yang bisa dilakukan oleh sistem dan bagaimana hubungannya dengan aktor atau pengguna. Identifikasi *use case* pada Tabel 4.10 merupakan hubungan antara *use case* yang telah digambarkan dengan persyaratan fungsional yang telah diidentifikasi pada Tabel 4.8.

Tabel 4.10 Identifikasi Use Case

Kode Use Case	Nama Use Case	Kode Persyaratan Fungsional
UC-SIP-01	Login	PF-SIP-01
UC-SIP-02	Logout	PF-SIP-01
UC-SIP-03	Mencari data pelanggan	PF-SIP-03
UC-SIP-04	Melihat data pelanggan	PF-SIP-04
UC-SIP-05	Menambah data pelanggan	PF-SIP-05
UC-SIP-06	Memperbarui data pelanggan	PF-SIP-06
UC-SIP-07	Menghapus data pelanggan	PF-SIP-07
UC-SIP-08	Melihat daftar penagihan	PF-SIP-08
UC-SIP-09	Menambah pelanggan ke daftar penagihan	PF-SIP-09
UC-SIP-10	Menghapus pelanggan dari daftar penagihan	PF-SIP-10
UC-SIP-11	Melihat data tagihan pelanggan	PF-SIP-11
UC-SIP-12	Mencetak slip tagihan	PF-SIP-12
UC-SIP-13	Mencatat pembayaran tagihan	PF-SIP-13
UC-SIP-14	Mencatat pembayaran tagihan di kantor	PF-SIP-13
UC-SIP-15	Memverifikasi pembayaran tagihan	PF-SIP-14
UC-SIP-16	Membuat laporan bulanan	PF-SIP-15

4.11 Use Case Scenario

Skenario use case merupakan rangkaian dari diagram use case yang menggambarkan langkah-langkah melakukan use case (Pressman, 2010). Skenario use case merupakan skenario yang dapat terjadi untuk masing-masing use case yang telah diidentifikasi pada Tabel 4.10 dengan mengimplementasikan persyaratan fungsional. Skenario use case menampilkan secara rinci informasi use case serta menjelaskan alur dari use case tersebut. Alur yang digambarkan merupakan alur utama dan alur alternatif yang mungkin terjadi dengan kondisi-kondisi tertentu. Skenario use case juga menggambarkan kondisi yang harus dipenuhi sebelum alur use case juga kondisi yang didapatkan setelah alur use case dijalankan.

Skenario use case yang menjadi fokus bahasan pada penelitian ini merupakan skenario dari use case yang berkaitan dengan pembayaran dan tagihan pelanggan. Skenario dari use case otentikasi pengguna dan pengelolaan pelanggan akan dilampirkan pada bagian lampiran.

4.11.1 Skenario *Use Case* Melihat Daftar Penagihan

Tabel 4.11 Skenario *Use Case* Melihat Daftar Penagihan

Scenario ID	UCS-SIP-08	Use Case	UC-SIP-08
Description	<i>Use case</i> melihat daftar penagihan merupakan alur untuk membuka daftar penagihan yang perlu dilakukan oleh petugas ke rumah-rumah.		
Actor	Administrator		
Goal	Aktor dapat melihat daftar penagihan.		
Preconditions	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Aktor telah membuka layar utama sistem. - Data pelanggan dan tagihan yang dituju sudah tersimpan pada sistem. 		
Main Flow	<ol style="list-style-type: none"> 1. Aktor membuka menu daftar penagihan 2. Sistem menampilkan daftar petugas penagihan. 3. Aktor memilih petugas penagihan yang dituju. 4. Sistem menampilkan daftar penagihan petugas terkait. 		
Alternative Flow	-		
Postconditions	Daftar pelanggan pada penagihan telah ditampilkan.		

4.11.2 Skenario *Use Case* Menambah Pelanggan ke Daftar Penagihan

Tabel 4.12 Skenario *Use Case* Menambah Daftar Penagihan

Scenario ID	UCS-SIP-09	Use Case	UC-SIP-09
Description	<i>Use case</i> menambah daftar penagihan merupakan <i>use case</i> yang dilakukan aktor untuk menambahkan data pelanggan pada petugas penagihan yang bertanggung jawab atas penagihan ke rumah.		
Actor	Administrator		
Goal	Aktor dapat menambahkan data pelanggan pada daftar penagihan untuk petugas penagihan yang ditunjuk.		
Preconditions	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Data pelanggan dan petugas yang dituju tersimpan pada sistem. 		
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol tambah data pelanggan. 2. Sistem menampilkan data pelanggan yang tidak ada dalam penagihan. 3. Aktor memilih data pelanggan yang ditambah. 4. Aktor menekan tombol tambahkan ke daftar penagihan. 5. Sistem menyimpan data penagihan. 		
Alternative Flow	-		
Postconditions	Daftar penagihan telah bertambah.		

4.11.3 Skenario *Use Case* Menghapus Pelanggan dari Daftar Penagihan

Tabel 4.13 Skenario *Use Case* Menghapus Daftar Penagihan

<i>Scenario ID</i>	UCS-SIP-10	<i>Use Case</i>	UC-SIP-10
<i>Description</i>	<i>Use case</i> menambah daftar penagihan merupakan <i>use case</i> yang dilakukan aktor untuk menghapus data pelanggan dari daftar penagihan petugas penagihan yang bertanggung jawab atas penagihan ke rumah.		
<i>Actor</i>	Administrator		
<i>Goal</i>	Aktor dapat menghapus data pelanggan dari daftar penagihan untuk petugas penagihan yang ditunjuk.		
<i>Preconditions</i>	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Data pelanggan dan petugas yang dituju tersimpan pada sistem. 		
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih data pelanggan yang akan dihapus. 2. Aktor menekan tombol hapus dari daftar penagihan 3. Sistem menyimpan data penagihan. 		
<i>Alternative Flow</i>	-		
<i>Postconditions</i>	Data pelanggan telah dihapus dari daftar penagihan.		

4.11.4 Skenario *Use Case* Melihat Data Tagihan Pelanggan

Tabel 4.14 Skenario *Use Case* Melihat Data Tagihan Pelanggan

<i>Scenario ID</i>	UCS-SIP-11	<i>Use Case</i>	UC-SIP-11
<i>Description</i>	<i>Use case</i> melihat data tagihan merupakan <i>use case</i> yang merupakan melihat data tagihan untuk masing-masing pelanggan.		
<i>Actor</i>	Petugas		
<i>Goal</i>	Aktor dapat melihat data tagihan untuk pelanggan yang dituju.		
<i>Preconditions</i>	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Aktor telah membuka daftar pelanggan. - Data pelanggan dan tagihan yang dituju sudah tersimpan pada sistem. 		
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor membuka data pelanggan yang dituju 2. Sistem menampilkan seluruh tagihan pelanggan. 		
<i>Alternative Flow</i>			
<i>Postconditions</i>	Data tagihan ditampilkan.		

4.11.5 Skenario *Use Case* Mencetak Slip Tagihan

Tabel 4.15 Skenario *Use Case* Mencetak Slip Tagihan

<i>Scenario ID</i>	UCS-SIP-12	<i>Use Case</i>	UC-SIP-12
<i>Description</i>	<i>Use case</i> mencetak slip tagihan untuk diberikan kepada petugas penagihan dan di distribusikan kepada pelanggan ke rumah-rumah.		

Actor	Administrator
Goal	Aktor dapat mencetak slip tagihan yang dipilih.
Preconditions	- Aktor telah masuk ke sistem. - Data pelanggan dan tagihan yang dituju sudah tersimpan pada sistem.
Main Flow	1. Aktor membuka menu tagihan. 2. Sistem menampilkan daftar tagihan. 3. Aktor memilih tagihan yang akan dicetak. 4. Sistem menampilkan konfirmasi cetak tagihan. 5. Aktor menekan tombol konfirmasi. 6. Sistem mencetak slip tagihan yang dipilih.
Alternative Flow	
Postconditions	Slip tagihan berhasil dicetak.

4.11.6 Skenario Use Case Mencatat Pembayaran Tagihan

Tabel 4.16 Skenario Use Case Mencatat Pembayaran Tagihan

Scenario ID	UCS-SIP-13	Use Case	UC-SIP-13
Description	Use case memperbarui status tagihan merupakan use case yang mengubah status tagihan menjadi dibayar.		
Actor	Petugas Penagihan		
Goal	Aktor dapat memperbarui status tagihan menjadi dibayar.		
Preconditions	- Aktor telah masuk ke sistem. - Data pelanggan dan tagihan yang dituju sudah tersimpan pada sistem.		
Main Flow	1. Petugas penagihan memilih tagihan yang dituju. 2. Petugas penagihan menekan tombol bayar. 3. Sistem menyimpan status tagihan. 4. Sistem menampilkan pesan tagihan menunggu verifikasi.		
Alternative Flow	-		
Postconditions	Status tagihan berubah menjadi menunggu verifikasi.		

4.11.7 Skenario Use Case Mencatat Pembayaran Tagihan di Kantor

Tabel 4.17 Skenario Use Case Mencatat Pembayaran Tagihan di Kantor

Scenario ID	UCS-SIP-14	Use Case	UC-SIP-14
Description	Use case memperbarui status tagihan di kantor merupakan use case yang mengubah status tagihan menjadi dibayar dari sisi administrator ketika ada pelanggan yang membayar ke kantor.		
Actor	Administrator		
Goal	Aktor dapat memperbarui status tagihan menjadi dibayar.		

Preconditions	- Aktor telah masuk ke sistem. - Data pelanggan dan tagihan yang dituju sudah tersimpan pada sistem.
Main Flow	1. Aktor membuka data pelanggan yang dituju. 2. Sistem menampilkan daftar pelanggan. 3. Aktor menekan tombol informasi tagihan. 4. Sistem menampilkan seluruh tagihan pelanggan. 5. Aktor memilih tagihan yang dituju. 6. Aktor menekan tombol bayar. 7. Sistem menyimpan status tagihan. 8. Sistem menampilkan pesan status tagihan diperbarui.
Alternative Flow	
Postconditions	Status tagihan berubah menjadi dibayar.

4.11.8 Skenario Use Case Memverifikasi Pembayaran Tagihan

Tabel 4.18 Skenario Use Case Memverifikasi Pembayaran Tagihan

Scenario ID	UCS-SIP-15	Use Case	UC-SIP-15
Description	Use case yang dilakukan oleh administrator ketika mendapatkan pembayaran dari petugas penagihan untuk memverifikasi jumlah uang yang diterima dengan catatan tagihan yang sudah dibayar.		
Actor	Administrator		
Goal	Aktor memverifikasi pembayaran yang tercatat pada sistem.		
Preconditions	- Aktor telah masuk ke sistem. - Aktor telah membuka layar utama sistem. - Pembayaran sudah dicatat pada sistem oleh petugas penagihan.		
Main Flow	1. Aktor membuka menu tagihan. 2. Sistem menampilkan rekap pembayaran yang telah dilakukan. 3. Aktor memeriksa kesesuaian pembayaran dengan uang yang diterima. 4. Aktor menekan tombol verifikasi. 5. Sistem menyimpan data pembayaran.		
Alternative Flow	3a. Pembayaran tidak sesuai dengan uang yang diterima. 3a. 1. Aktor menekan tombol tolak pembayaran. 3a. 2. Sistem mengirim notifikasi kepada petugas penagihan terkait. 3a. 3. Kembali ke flow nomor (2).		
Postconditions	Pembayaran berhasil diverifikasi.		

4.11.9 Skenario Use Case Membuat Laporan Bulanan

Tabel 4.19 Skenario Use Case Membuat Laporan Bulanan

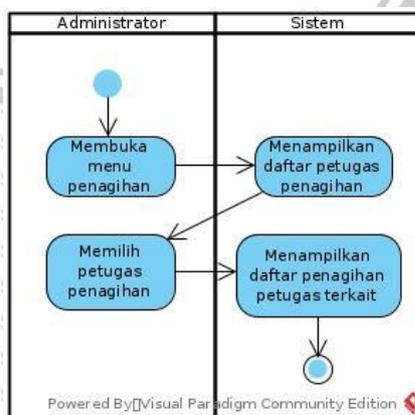
Scenario ID	UCS-SIP-16	Use Case	UC-SIP-16
Use Case	Membuat laporan bulanan (UC-SIP-16)		
Description	Use case yang dilakukan oleh administrator untuk membuat laporan bulanan yang dibantu oleh sistem.		
Actor	Administrator		
Goal	Aktor membuat laporan bulanan.		
Preconditions	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Aktor telah membuka layar utama sistem. 		
Main Flow	<ol style="list-style-type: none"> 1. Aktor membuka menu laporan bulanan. 2. Aktor menekan tombol cetak laporan bulanan. 3. Sistem mencetak laporan bulanan. 		
Alternative Flow	-		
Postconditions	Pembayaran berhasil diverifikasi.		

4.12 Activity Diagram

Activity diagram merupakan salah satu diagram UML yang digunakan untuk menggambarkan alur proses bisnis sebuah sistem dari satu aktivitas ke aktivitas yang lain (Pressman, 2010). Activity diagram memodelkan skenario use case yang sudah di definisikan ke dalam notasi UML.

4.12.1 Activity Diagram Melihat Daftar Penagihan

Skenario melihat daftar penagihan yang telah didefinisikan pada Tabel 4.11 digambarkan ke dalam activity diagram pada Gambar 4.11 berikut dengan action yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.11 Activity Diagram Melihat Daftar Penagihan

4.12.2 Activity Diagram Menambah Pelanggan ke Daftar Penagihan

Skenario menambah pelanggan ke daftar penagihan yang telah didefinisikan pada Tabel 4.12 digambarkan ke dalam *activity diagram* pada Gambar 4.12 berikut dengan *action* yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.12 Activity Diagram Menambah Pelanggan ke Daftar Penagihan

4.12.3 Activity Diagram Menghapus Pelanggan dari Daftar Penagihan

Skenario menghapus pelanggan dari daftar penagihan yang telah didefinisikan pada Tabel 4.13 digambarkan ke dalam *activity diagram* pada Gambar 4.13 berikut dengan *action* yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.13 Activity Diagram Menghapus Pelanggan Daftar Penagihan

4.12.4 Activity Diagram Melihat Data Tagihan Pelanggan

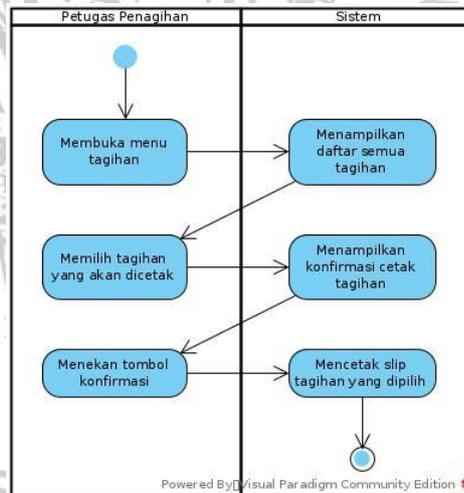
Skenario melihat data tagihan pelanggan yang telah didefinisikan pada Tabel 4.14 digambarkan ke dalam *activity diagram* pada Gambar 4.14 berikut dengan *action* yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.14 Activity Diagram Melihat Data Tagihan Pelanggan

4.12.5 Activity Diagram Mencetak Slip Tagihan

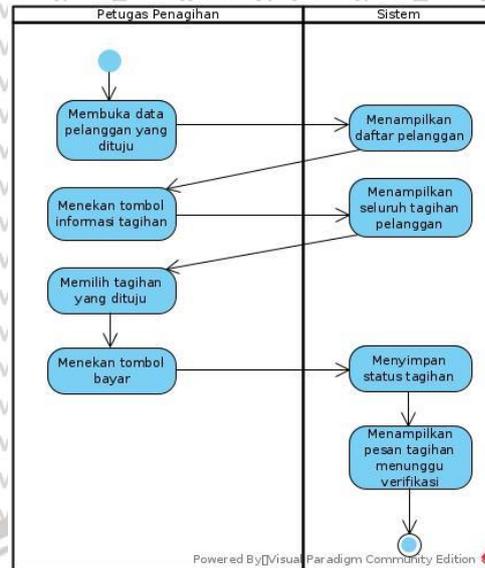
Skenario mencetak slip tagihan yang telah didefinisikan pada Tabel 4.15 digambarkan ke dalam *activity diagram* pada Gambar 4.15 berikut dengan *action* yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.15 Activity Diagram Mencetak Slip Tagihan

4.12.6 Activity Diagram Mencatat Pembayaran Tagihan

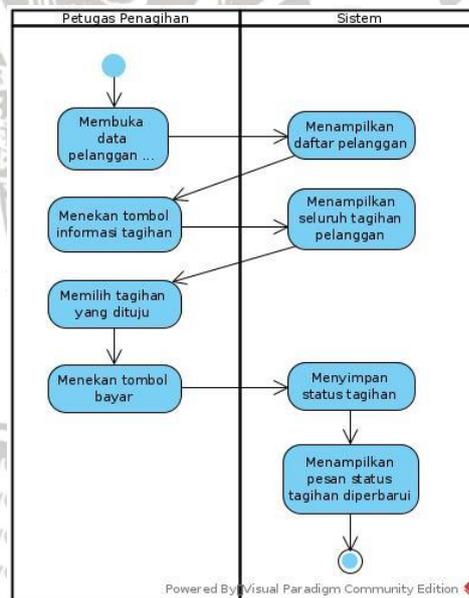
Skenario mencatat pembayaran tagihan yang telah didefinisikan pada Tabel 4.16 digambarkan ke dalam *activity diagram* pada Gambar 4.16 berikut dengan *action* yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.16 Activity Diagram Mencatat Pembayaran Tagihan

4.12.7 Activity Diagram Mencatat Pembayaran Tagihan di Kantor

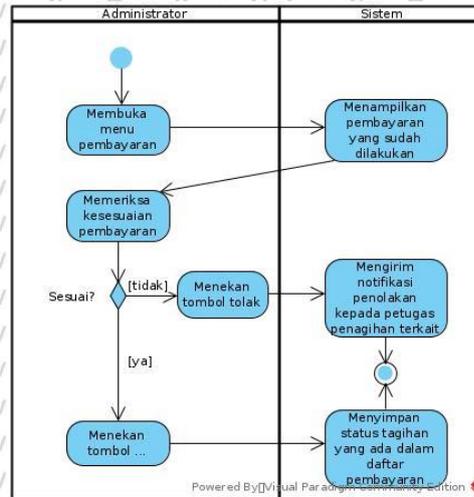
Skenario mencatat pembayaran tagihan di kantor yang telah didefinisikan pada Tabel 4.17 digambarkan ke dalam *activity diagram* pada Gambar 4.17 berikut dengan *action* yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.17 Activity Diagram Mencatat Pembayaran Tagihan di Kantor

4.12.8 Activity Diagram Memverifikasi Pembayaran Tagihan

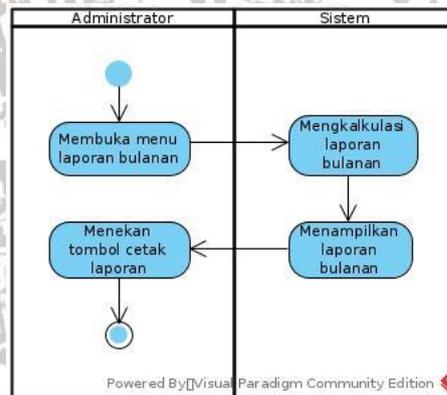
Skenario memverifikasi pembauaran tagihan yang telah didefinisikan pada Tabel 4.18 digambarkan ke dalam *activity diagram* pada Gambar 4.18 berikut dengan *action* yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.18 Activity Diagram Memverifikasi Pembayaran Tagihan

4.12.9 Activity Diagram Membuat Laporan Bulanan

Skenario membuat laporan bulanan yang telah didefinisikan pada Tabel 4.19 digambarkan ke dalam *activity diagram* pada Gambar 4.19 berikut dengan *action* yang menggambarkan interaksi yang diberikan oleh aktor kepada sistem dan reaksi dari sistem.



Gambar 4.19 Activity Diagram Membuat Laporan Bulanan

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas terkait perancangan sistem dan implementasi dari rekayasa kebutuhan. Perancangan sistem memuat artefak sistem yang mendukung kebutuhan sistem. Implementasi sistem memuat artefak sistem yang dapat digunakan untuk menjalankan sistem sesuai dengan rancangan yang dibuat pada perancangan sistem.

5.1 Perancangan Sistem

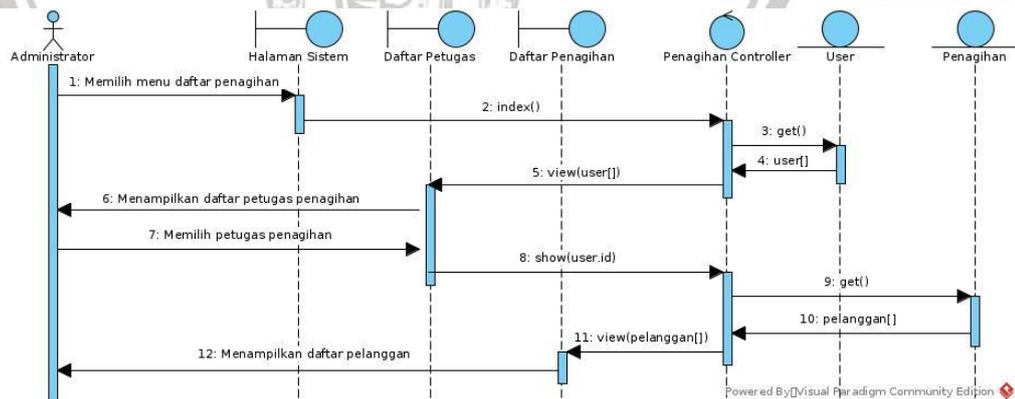
Perancangan sistem merupakan tahapan yang merancang spesifikasi sistem berdasarkan kebutuhan pengguna yang telah dianalisis pada tahapan analisis kebutuhan sistem (Sommerville, 2016). Pada tahapan ini memuat perancangan sistem menggunakan artefak yang mendukung spesifikasi sistem, terdiri dari: *sequence diagram*, *class diagram*, *data model* dan rancangan antarmuka pengguna.

5.1.1 Sequence Diagram

Sequence diagram merupakan artefak rancangan sistem yang memuat komunikasi antar objek pada sistem, sehingga dapat dilihat secara runtuh perubahan dan perkembangan pada sistem (Dennis, 2012). *Sequence diagram* memuat aktor, interaksi aktor dan pesan yang dikirim antar elemen sistem.

5.1.1.1 Sequence Diagram Melihat Daftar Penagihan

Sequence diagram untuk *use case* melihat daftar penagihan menggambarkan interaksi aktor dengan elemen sistem maupun interaksi antar elemen sistem. *Sequence diagram* melihat daftar penagihan dapat dilihat pada Gambar 5.1 berikut ini:

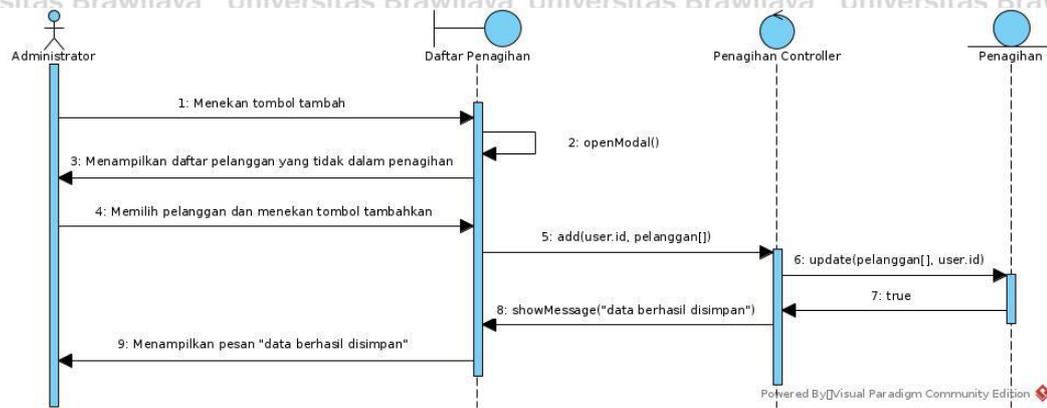


Gambar 5.1 Sequence Diagram Melihat Daftar Penagihan

Diagram *sequence* pada Gambar 5.1 menggambarkan interaksi aktor dengan *boundary* layar utama untuk memilih menu. Sistem akan memanggil operasi pada Penagihan Controller untuk mendapatkan data petugas penagihan. Aktor memilih petugas penagihan dan sistem akan mendapatkan data pelanggan petugas terkait untuk ditampilkan sebagai daftar penagihan.

5.1.1.2 Sequence Diagram Menambah Pelanggan ke Daftar Penagihan

Sequence diagram untuk *use case* menambah pelanggan ke daftar penagihan menggambarkan interaksi aktor dengan entitas sistem maupun interaksi antar elemen sistem. *Sequence diagram* menambah pelanggan ke daftar dapat dilihat pada Gambar 5.2 berikut ini:

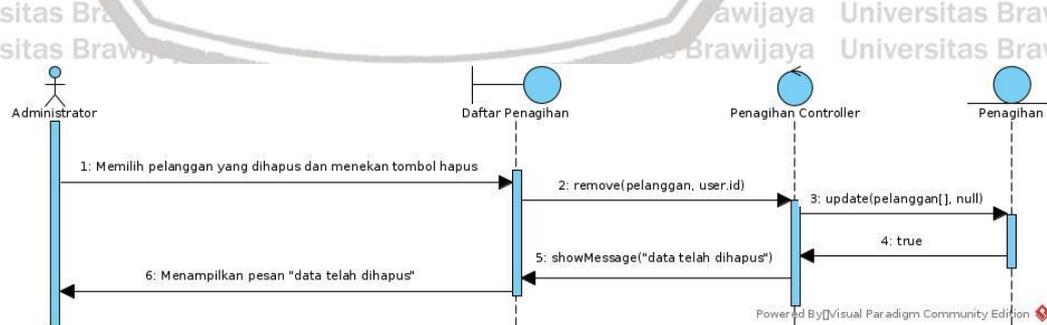


Gambar 5.2 Sequence Diagram Menambah Pelanggan ke Daftar Penagihan

Diagram *sequence* menambah pelanggan ke daftar penagihan pada Gambar 5.2 menggambarkan alur proses dimulai dari interaksi aktor dengan *boundary* sistem untuk mendapatkan data *user* dan pelanggan melalui Penagihan Controller, lalu ditampilkan melalui *boundary* Daftar Petugas dan Daftar Penagihan. Penagihan Controller mengakses *entity* User dan Penagihan untuk mendapatkan data dari *database*. Data yang dikirim dari aktor berupa data pelanggan yang dipilih dan akan ditambah ke daftar penagihan melalui *controller* dan *entity* terlibat.

5.1.1.3 Sequence Diagram Menghapus Pelanggan dari Daftar Penagihan

Sequence diagram untuk *use case* menghapus pelanggan dari daftar penagihan menggambarkan interaksi aktor dengan elemen sistem maupun interaksi antar elemen sistem. *Sequence diagram* menghapus pelanggan dari daftar penagihan dapat dilihat pada Gambar 5.3 berikut ini:



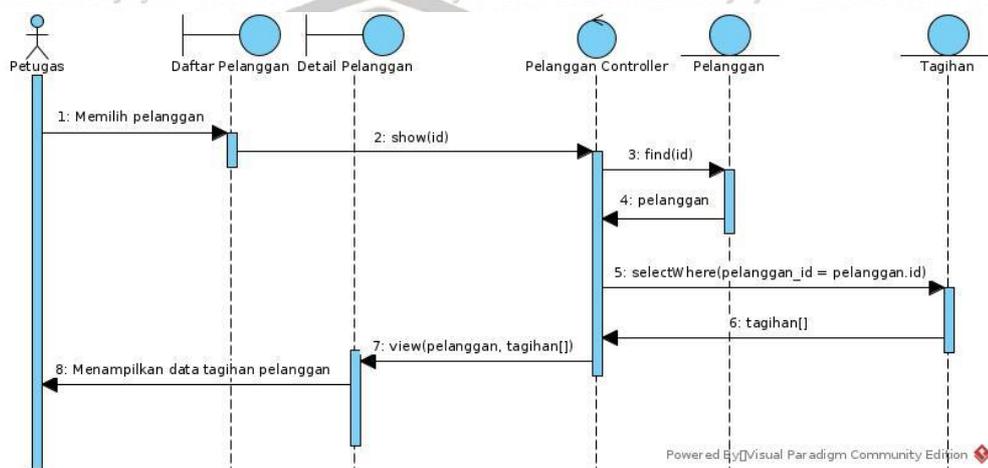
Gambar 5.3 Sequence Diagram Menghapus Pelanggan dari Daftar Penagihan

Diagram *sequence* menghapus pelanggan dari daftar penagihan pada Gambar 5.3 menggambarkan alur proses dimulai dari interaksi aktor dengan *boundary* sistem untuk mendapatkan data *user* dan pelanggan melalui Penagihan Controller,

lalu ditampilkan melalui *boundary* Daftar Petugas dan Daftar Penagihan. Penagihan Controller mengakses *entity* User dan Penagihan untuk mendapatkan data dari *database*. Data yang dikirim dari aktor berupa data pelanggan yang dipilih dan akan dihapus dari daftar penagihan melalui *controller* dan *entity* terlibat.

5.1.1.4 Sequence Diagram Melihat Data Tagihan Pelanggan

Sequence diagram untuk *use case* melihat data tagihan pelanggan menggambarkan interaksi aktor dengan elemen sistem maupun interaksi antar elemen sistem. *Sequence diagram* melihat data tagihan pelanggan dapat dilihat pada Gambar 5.4 berikut ini:

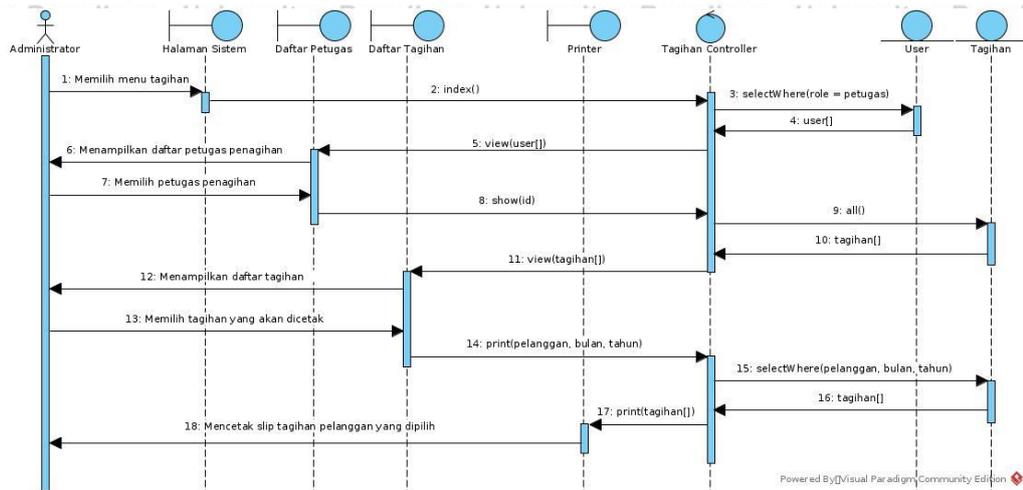


Gambar 5.4 Sequence Diagram Melihat Data Tagihan Pelanggan

Diagram *sequence* melihat data tagihan pelanggan pada Gambar 5.4 menggambarkan alur proses dimulai dari interaksi aktor dengan *boundary* sistem untuk mendapatkan pelanggan dan tagihan melalui Pelanggan Controller dan Tagihan Controller, lalu ditampilkan melalui *boundary* Daftar Pelanggan dan Detail Pelanggan. Tagihan Controller mengakses *entity* Pelanggan dan Tagihan untuk mendapatkan data dari *database*.

5.1.1.5 Sequence Diagram Mencetak Slip Tagihan

Sequence diagram untuk *use case* mencetak slip menggambarkan interaksi aktor dengan elemen sistem maupun interaksi antar elemen sistem. *Sequence diagram* mencetak slip tagihan dapat dilihat pada Gambar 5.5 berikut ini:

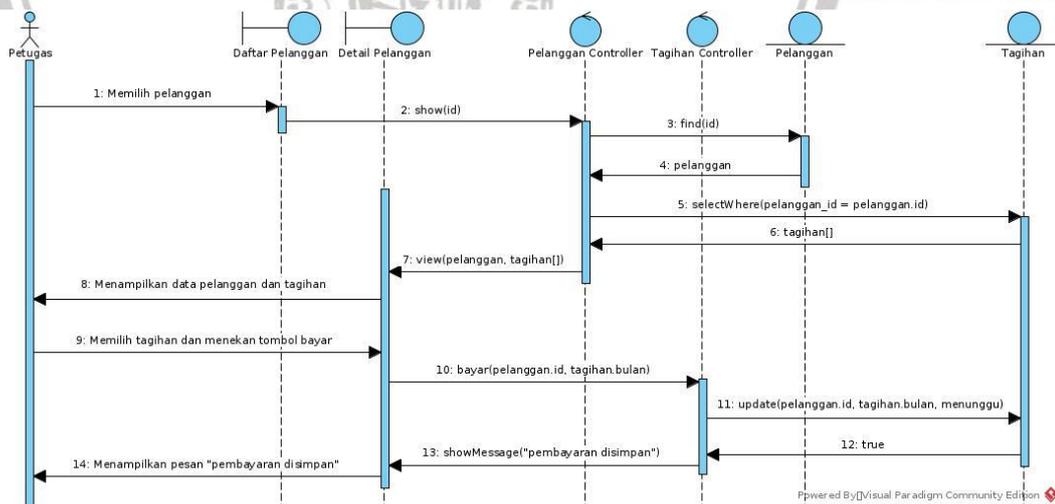


Gambar 5.5 Sequence Diagram Mencetak Slip Tagihan

Diagram *sequence* mencetak slip tagihan pada Gambar 5.5 menggambarkan alur proses dimulai dari interaksi aktor dengan *boundary* sistem untuk mendapatkan data *tagihan* melalui Tagihan Controller, lalu ditampilkan melalui *boundary* Layar Tagihan dan Printer yang berinteraksi dengan mesin pencetak. Tagihan Controller mengakses *entity* Tagihan untuk mendapatkan data dari *database*.

5.1.1.6 Sequence Diagram Mencatat Pembayaran Tagihan

Sequence diagram untuk *use case* mencatat pembayaran tagihan menggambarkan interaksi aktor dengan elemen sistem maupun interaksi antar elemen sistem. *Sequence diagram* mencatat pembayaran tagihan dapat dilihat pada Gambar 5.6 berikut ini:



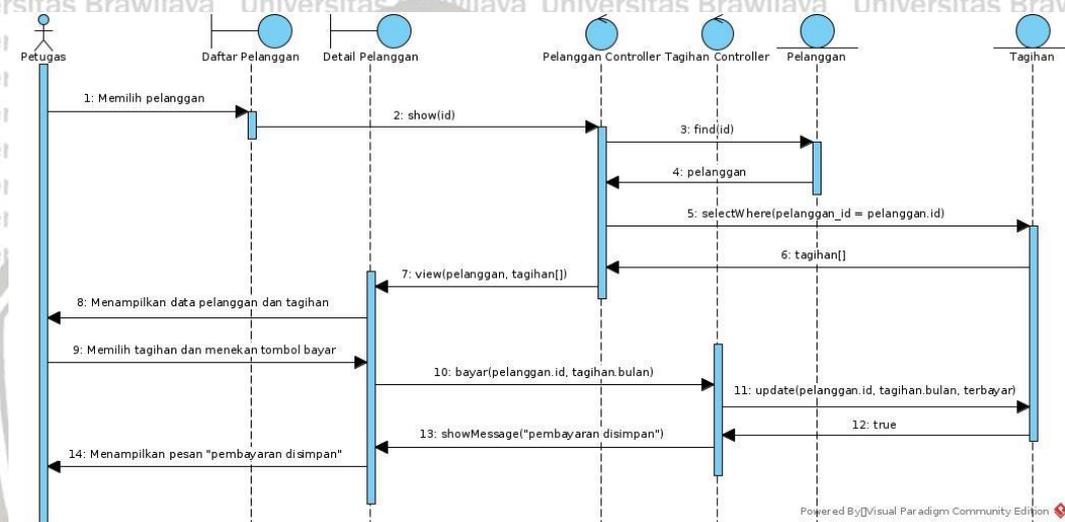
Gambar 5.6 Sequence Diagram Mencatat Pembayaran Tagihan

Diagram *sequence* mencatat pembayaran tagihan pada Gambar 5.6 menggambarkan alur proses dimulai dari interaksi aktor dengan *boundary* sistem untuk mendapatkan data tagihan dan pelanggan melalui Pelanggan Controller dan Tagihan Controller, lalu ditampilkan melalui *boundary* Daftar Pelanggan dan Detail

Pelanggan. Tagihan Controller dan Pelanggan Controller mengakses *entity* Tagihan dan Penagihan untuk mendapatkan data dari *database*. Data yang dikirim dari aktor berupa data tagihan yang dipilih dan akan diperbarui status tagihannya melalui *controller* dan *entity* terlibat. Pada diagram ini, ditekankan pada aturan bisnis yang mengubah status tagihan menjadi “menunggu verifikasi”, sehingga perlu melewati proses verifikasi untuk menjadi pembayaran yang sah.

5.1.1.7 Sequence Diagram Mencatat Pembayaran Tagihan di Kantor

Sequence diagram untuk *use case* mencatat pembayaran tagihan di kantor menggambarkan interaksi aktor dengan elemen sistem maupun interaksi antar elemen sistem. *Sequence diagram* mencatat pembayaran tagihan di kantor dapat dilihat pada Gambar 5.7 berikut ini:

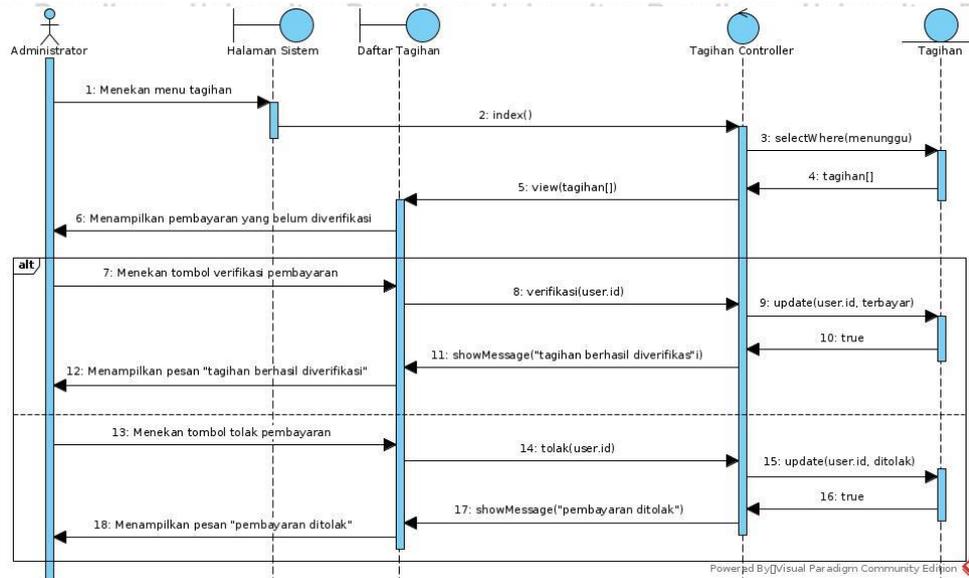


Gambar 5.7 Sequence Diagram Mencatat Pembayaran Tagihan di Kantor

Diagram *sequence* mencatat pembayaran tagihan di kantor pada Gambar 5.7 menggambarkan alur proses dimulai dari interaksi aktor dengan *boundary* sistem untuk mendapatkan data tagihan dan pelanggan melalui Pelanggan Controller dan Tagihan Controller, lalu ditampilkan melalui *boundary* Daftar Pelanggan dan Detail Pelanggan. Tagihan Controller dan Pelanggan Controller mengakses *entity* Tagihan dan Penagihan untuk mendapatkan data dari *database*. Data yang dikirim dari aktor berupa data tagihan yang dipilih dan akan diperbarui status tagihannya melalui *controller* dan *entity* terlibat. Pada diagram ini, ditekankan pada aturan bisnis yang mengubah status tagihan menjadi “diverifikasi”, sehingga tidak melewati proses verifikasi.

5.1.1.8 Sequence Diagram Memverifikasi Pembayaran Tagihan

Sequence diagram untuk *use case* memverifikasi pembayaran tagihan menggambarkan interaksi aktor dengan elemen sistem maupun interaksi antar elemen sistem. *Sequence diagram* memverifikasi pembayaran tagihan dapat dilihat pada Gambar 5.8 berikut ini:

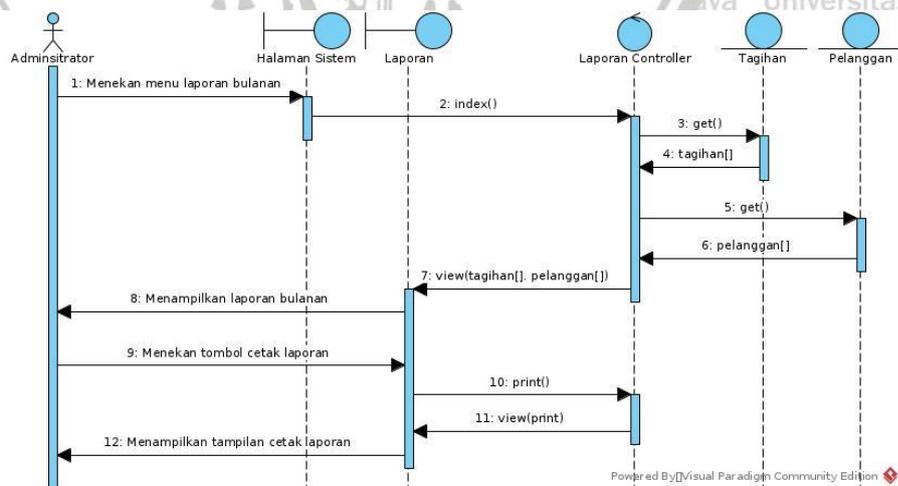


Gambar 5.8 Sequence Diagram Memverifikasi Pembayaran Tagihan

Diagram *sequence* memverifikasi pembayaran tagihan pada Gambar 5.8 menggambarkan alur proses dimulai dari interaksi aktor dengan *boundary* sistem untuk mendapatkan data tagihan melalui Tagihan Controller, lalu ditampilkan melalui *boundary* Pembayaran. Tagihan Controller mengakses *entity* Tagihan untuk mendapatkan data dari *database*. Masukan dari aktor berupa keputusan untuk memverifikasi atau menolak pembayaran sehingga terjadi pembaruan pada status tagihan.

5.1.1.9 Sequence Diagram Membuat Laporan Bulanan

Sequence diagram untuk *use case* membuat laporan bulanan menggambarkan interaksi aktor dengan elemen sistem maupun interaksi antar elemen sistem. *Sequence diagram* membuat laporan bulanan dapat dilihat pada Gambar 5.9 berikut ini:



Gambar 5.9 Sequence Diagram Membuat Laporan Bulanan

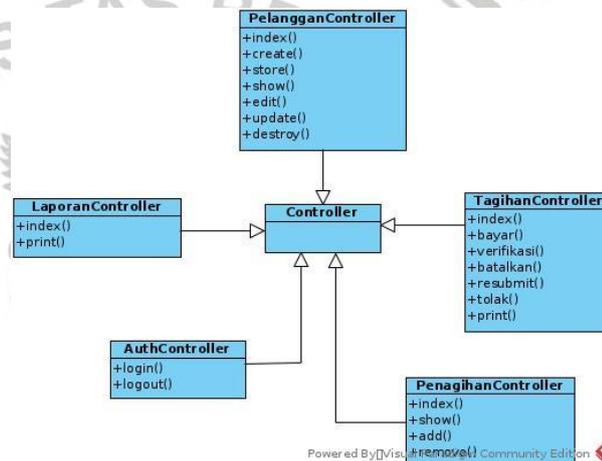
Diagram *sequence* membuat laporan bulanan pada Gambar 5.9 menggambarkan alur proses dimulai dari interaksi aktor dengan *boundary* sistem untuk mendapatkan data tagihan dan pelanggan melalui Laporan Controller, lalu ditampilkan melalui *boundary* Laporan. Laporan Controller mengakses *entity* Tagihan dan User untuk mendapatkan data dari *database*. Laporan bulanan dikalkulasi dari data tagihan dan *user* sebagai petugas penagihan.

5.1.2 Class Diagram

Class diagram merupakan diagram yang menggambarkan struktur dari sebuah sistem menjadi kelas-kelas dan masing-masing kelas memiliki atribut dan operasi (Pressman, 2010). *Class diagram* menggambarkan elemen *controller* dan *entity* sebagai elemen yang memiliki atribut atau operasi di dalamnya.

5.1.2.1 Class Diagram Controller

Diagram kelas untuk bagian *controller* dari sistem informasi pembayaran dapat dilihat pada Gambar 5.10 berikut ini.

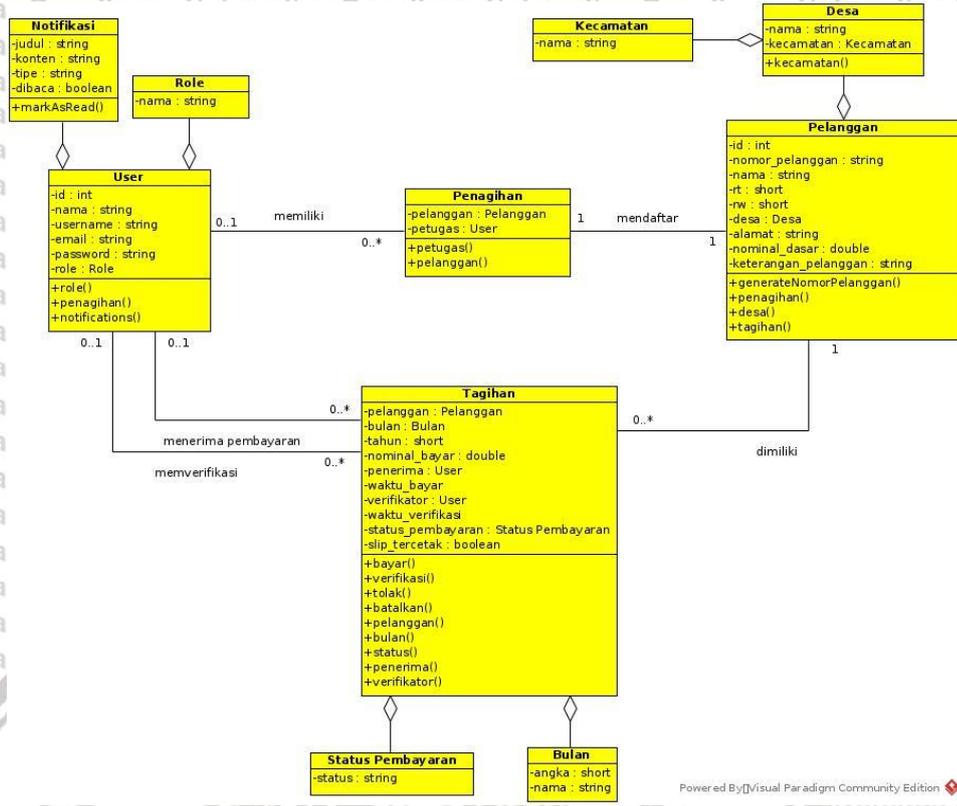


Gambar 5.10 Class Diagram Controller

Semua kelas *controller* merupakan kelas turunan dari *superclass* yang disediakan oleh Laravel yaitu kelas Controller. Kelas yang digunakan pada penelitian ini ada 5 kelas *controller*, yaitu TagihanController, PenagihanController, AuthController, LaporanController dan PelangganController. Masing-masing memiliki fungsi pengelolaan masing masing, TagihanController sebagai pengelola tagihan untuk melakukan pembayaran dan verifikasi, PenagihanController untuk mengelola penagihan, AuthController untuk mengurus segala otentikasi pengguna, LaporanController mengelola laporan bulanan dan PelangganController mengelola pelanggan untuk melakukan CRUD.

5.1.2.2 Class Diagram Model

Diagram kelas untuk bagian *model* dari sistem informasi pembayaran dapat dilihat pada Gambar 5.11 berikut ini.

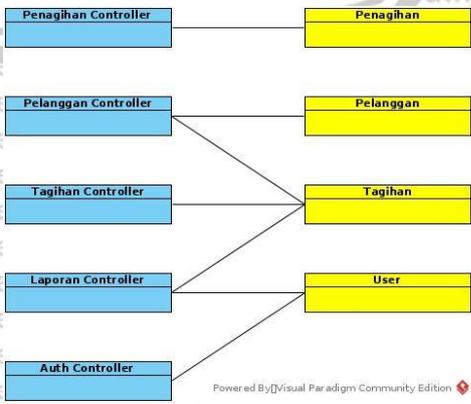


Gambar 5.11 Class Diagram Model

Semua kelas pada Gambar 5.11 merupakan kelas turunan dari *superclass* yang disediakan oleh Laravel yaitu kelas Model, namun pada model tersebut tidak digambarkan generalisasi tersebut untuk keterbacaan gambar. Model ini memiliki 4 kelas utama yaitu User, Penagihan, Pelanggan dan Tagihan. Kelas lain merupakan kelas untuk kebutuhan normalisasi dan digunakan untuk memodelkan relasi saat menggunakan kerangka kerja Laravel menggunakan Eloquent ORM.

5.1.2.3 Relasi Controller dan Model

Kelas-kelas dari bagian *controller* dan *model* memiliki hubungan antar kelas yang digambarkan pada Gambar 5.12 berikut ini.



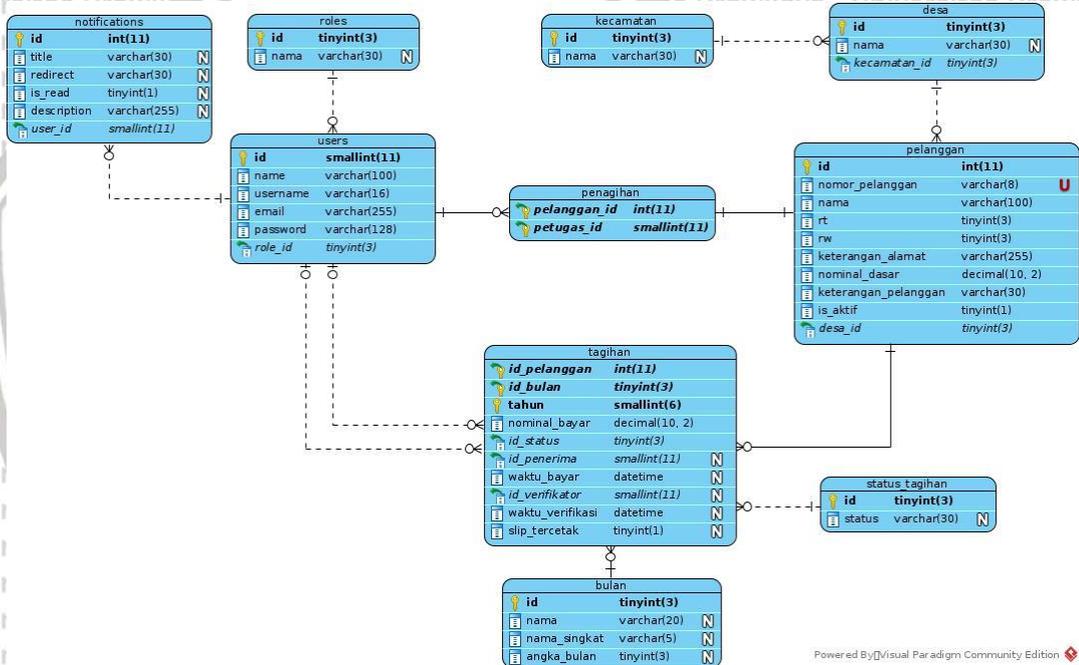
Gambar 5.12 Relasi Controller dan Model

Relasi untuk kelas-kelas *controller* dengan model merupakan penggambaran *model* yang diakses oleh *controller* untuk menjalankan logika sistem. Relasi antar kelas ini melibatkan semua *controller* dan model utama.

5.1.3 Pemetaan Class Diagram ke Relational Data Model

Model data merupakan model dari elemen data serta hubungan antar elemennya yang membentuk struktur data yang dibutuhkan untuk menjalankan proses bisnis yang ada (Marakas & O'Brien, 2010). Model data berikut ini menggambarkan model data untuk diimplementasikan ke dalam sistem menjadi struktur *database*.

Class diagram pada Gambar 5.11 menjadi dasar untuk membuat pemetaan menjadi *relational data model* untuk diimplementasikan sebagai struktur *database* untuk sistem informasi pembayaran TV kabel yang digambarkan pada Gambar 5.13 berikut.



Gambar 5.13 Relational Data Model

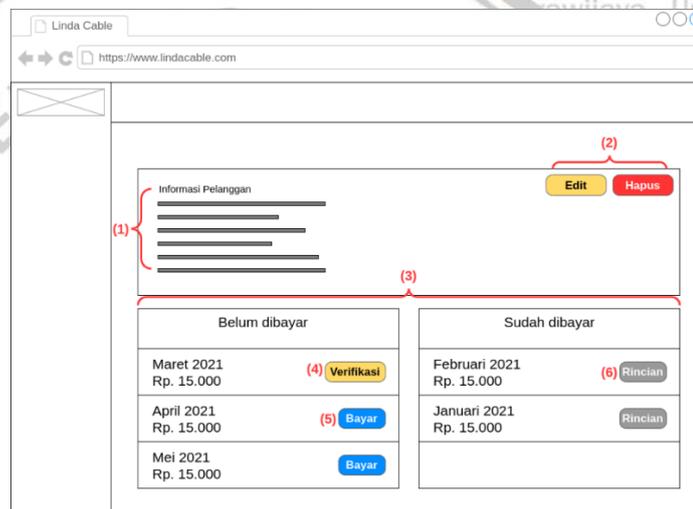
Model data tersebut memuat kelas dan atribut yang telah dimodelkan pada *class diagram* menjadi tabel-tabel untuk diimplementasikan pada *database*, yaitu: User, Penagihan, Tagihan dan Pelanggan. Tipe data untuk masing-masing atribut didefinisikan model ini. *Constraint* untuk *primary key* dan *foreign key* didefinisikan untuk memberikan identitas bagi masing-masing tabel dan mendefinisikan atribut yang menjadi kunci dari relasi masing-masing tabel. Pada model ini juga telah dilakukan normalisasi untuk mengurangi redundansi data yang sifatnya repetitif jika dimasukkan menjadi baris data.

5.1.4 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan acuan untuk mengimplementasi antarmuka pengguna yang merupakan bagian *view*. Antarmuka pengguna merupakan komponen sistem yang paling dekat dengan pengguna, oleh karena itu perancangannya perlu dibuat semenarik dan semudah mungkin untuk digunakan oleh pengguna (Marakas & O'Brien, 2010). Perancangan ini perlu agar saat melakukan implementasi, antarmuka pengguna sesuai untuk menampilkan informasi yang diperlukan untuk menjalankan sistem sesuai dengan proses bisnis.

5.1.4.1 Rancangan Antarmuka Halaman Detail Pelanggan

Halaman detail pelanggan merupakan halaman yang menampilkan data pelanggan secara terperinci beserta dengan tagihan-tagihan untuk pelanggan terkait. Rancangan antarmuka halaman detail pelanggan dapat dilihat pada Gambar 5.14 berikut ini.



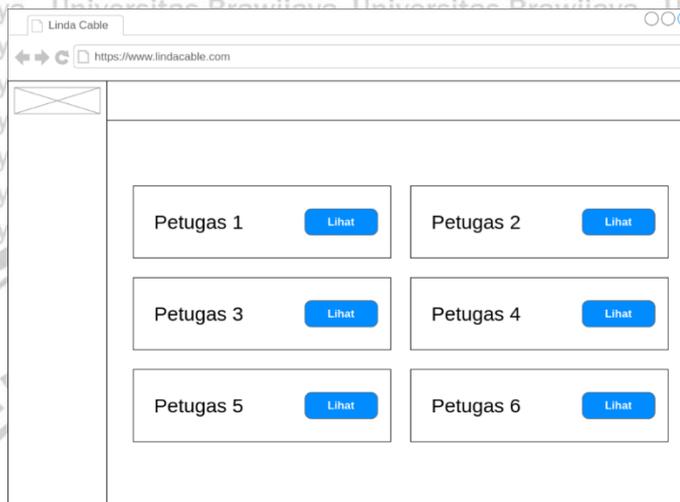
Gambar 5.14 Rancangan Antarmuka Halaman Detail Pelanggan

Rancangan antarmuka halaman detail pelanggan menerapkan beberapa poin pada perancangannya untuk menerapkan fitur-fitur yang dibutuhkan untuk halaman detail pelanggan. Poin (1) merupakan elemen antarmuka pengguna yang menampilkan informasi pelanggan sesuai dengan data yang disimpan pada sistem. Poin (2) menunjukkan elemen untuk melakukan pengelolaan data pelanggan, yaitu memperbarui dan menghapus pelanggan. Elemen ini hanya dapat diakses oleh aktor administrator sebagai pembatasan hak akses antar pengguna. Poin (3) merupakan bagian dari halaman detail pelanggan yang menunjukkan daftar tagihan dari pelanggan terkait, di mana terdapat subbagian dari daftar tagihan untuk membedakan tagihan yang sudah dibayar dan belum dibayar. Pada bagian ini terdapat 3 elemen yang memungkinkan untuk muncul berdasarkan dari kondisi tagihan, yaitu verifikasi (4), bayar (5) dan rincian (6). Poin (4) adalah elemen untuk verifikasi yang hanya dapat diakses oleh aktor administrator. Poin (5) adalah elemen untuk melakukan pencatatan pembayaran dan akan diproses oleh sistem.

Poin (5) adalah elemen untuk melihat rincian pembayaran yang sudah dilakukan dan diverifikasi untuk melihat ataupun mencetak bukti transaksinya.

5.1.4.2 Rancangan Antarmuka Halaman Daftar Petugas Penagihan

Halaman daftar petugas penagihan merupakan halaman untuk menampilkan daftar petugas penagihan yang dapat merujuk pada halaman daftar penagihan petugas terkait. Rancangan antarmuka halaman daftar petugas penagihan dapat dilihat pada Gambar 5.15 berikut ini.

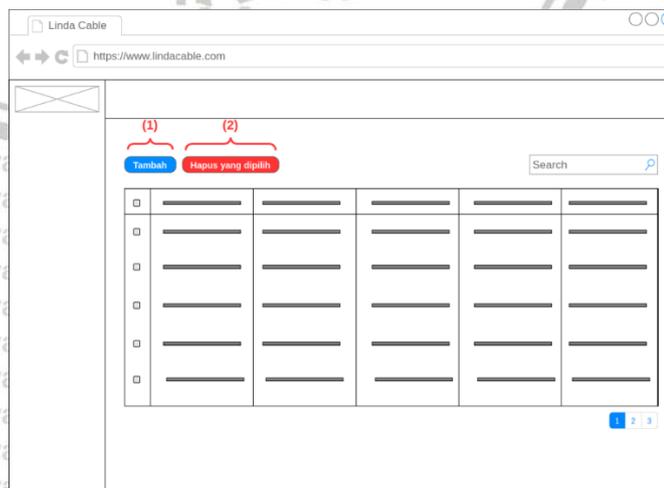


Gambar 5.15 Rancangan Antarmuka Halaman Daftar Petugas Penagihan

Rancangan antarmuka halaman daftar petugas penagihan menampilkan semua petugas penagihan dan dapat merujuk ke daftar penagihan petugas terkait.

5.1.4.3 Rancangan Antarmuka Halaman Daftar Penagihan

Halaman daftar penagihan menampilkan daftar pelanggan yang masuk ke dalam penagihan petugas terkait. Rancangan antarmuka halaman daftar penagihan dapat dilihat pada Gambar 5.16 berikut ini.

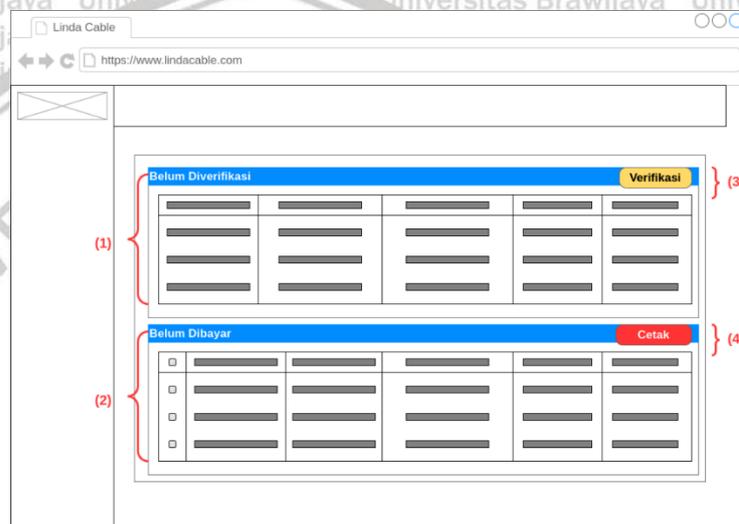


Gambar 5.16 Rancangan Antarmuka Halaman Daftar Penagihan

Rancangan antarmuka halaman daftar penagihan memiliki daftar pelanggan yang masuk ke dalam penagihan petugas terkait. Elemen yang disorot adalah bagian untuk mengakses fitur tambah pelanggan ke daftar penagihan (1) dan hapus pelanggan dari daftar penagihan (2).

5.1.4.4 Rancangan Antarmuka Halaman Daftar Tagihan

Halaman daftar tagihan merupakan halaman yang menampilkan tagihan pelanggan dalam satu halaman yang mencakup tagihan-tagihan yang belum dicetak slip tagihannya dan tagihan-tagihan yang belum diverifikasi. Rancangan antarmuka halaman daftar tagihan akan berfokus pada halaman daftar tagihan untuk aktor administrator yang memiliki semua fitur untuk mengakses menu daftar tagihan. Rancangan antarmuka halaman daftar tagihan dapat dilihat pada Gambar 5.17 berikut ini.

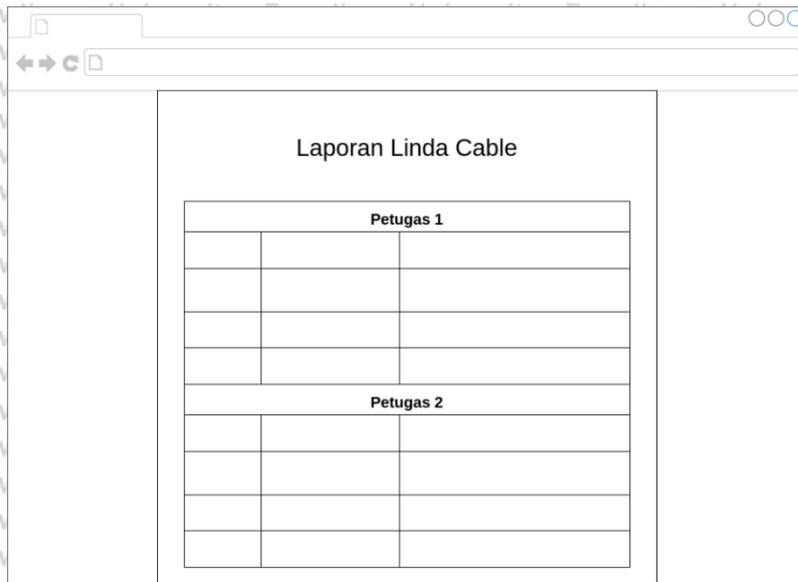


Gambar 5.17 Rancangan Antarmuka Halaman Daftar Tagihan

Rancangan antarmuka halaman daftar tagihan memiliki beberapa elemen yang digunakan untuk menampilkan tagihan-tagihan pelanggan. Elemen yang ditunjuk Poin (1) dan (2) merupakan bagian tabel yang menampilkan tagihan pelanggan untuk yang belum diverifikasi dan belum dibayar. Poin (3) dan (4) merupakan elemen untuk mengakses fitur verifikasi dan cetak tagihan.

5.1.4.5 Rancangan Antarmuka Halaman Laporan Bulanan

Halaman laporan bulanan menampilkan laporan bulanan yang telah dihitung oleh sistem berdasarkan pembayaran tagihan pada bulan tersebut. Rancangan antarmuka halaman laporan bulanan dapat dilihat pada Gambar 5.18 berikut ini.



Gambar 5.18 Rancangan Antarmuka Halaman Laporan Bulanan

Antarmuka halaman laporan bulanan memuat informasi tagihan yang diterima oleh masing-masing petugas dalam bentuk tabel.

5.1.5 Perancangan Pengujian Penerimaan Pengguna

Pengujian ini disebut juga *user acceptance testing* (UAT) yang mana pengujian ini melibatkan pengguna dari sistem untuk menguji sistem sendiri sebelum digunakan (Sommerville, 2016). Berdasarkan wawancara kebutuhan pada LAMPIRAN B poin B.1, tabel pengujian penerimaan pengguna dibuat dengan beberapa kriteria ajuan Naik dan Tripathy (2018), yaitu kriteria *functional correctness and completeness, usability* dan *confidentiality and availability*.

Pengguna dari sistem terdapat dua peran, yaitu administrator dan petugas penagihan. Rancangan pengujian penerimaan pengguna oleh administrator dapat dilihat pada tabel Tabel 5.1 berikut ini.

Tabel 5.1 Rancangan Pengujian Penerimaan Pengguna oleh Administrator

No.	Pertanyaan	Jawaban	
		Ya	Tidak
Functional Correctness and Completeness			
1	Apakah fitur penagihan membantu administrator untuk mengelola penagihan milik petugas penagihan?		
2	Apakah fitur tagihan membantu administrator untuk mengelola pencatatan pembayaran di kantor?		
3	Apakah fitur laporan membantu administrator untuk membuat laporan bulanan?		

Tabel 5.1 Rancangan Pengujian Penerimaan Pengguna oleh Administrator (lanjutan)

<i>Usability</i>		
4	Apakah sistem membantu administrator mengerjakan aktivitas pembayaran dan pengelolaan sehari-hari?	
5	Apakah alur sistem informasi pembayaran sudah jelas saat menjalankan fungsional sistem?	
6	Apakah keseluruhan fitur sistem informasi pembayaran mudah digunakan?	
<i>Confidentiality and Availability</i>		
7	Apakah sistem informasi pembayaran telah memberikan hak akses sesuai dengan peran pengguna?	
8	Apakah sistem informasi pembayaran menjamin keamanan data dari bukan pengguna?	
Total		

Rancangan pengujian penerimaan pengguna selanjutnya oleh petugas penagihan dapat dilihat pada Tabel 5.2 berikut ini.

Tabel 5.2 Rancangan Pengujian Penerimaan Pengguna oleh Petugas Penagihan

No.	Pertanyaan	Jawaban	
		Ya	Tidak
<i>Functional Correctness and Completeness</i>			
1	Apakah fitur tagihan membantu petugas untuk mengelola pencatatan pembayaran?		
<i>Usability</i>			
2	Apakah sistem membantu petugas penagihan mengerjakan aktivitas pembayaran sehari-hari?		
3	Apakah alur sistem informasi pembayaran sudah jelas saat menjalankan fungsional sistem?		
4	Apakah keseluruhan fitur sistem informasi pembayaran mudah digunakan?		
<i>Confidentiality and Availability</i>			
5	Apakah sistem informasi pembayaran telah memberikan hak akses sesuai dengan peran pengguna?		
6	Apakah sistem informasi pembayaran menjamin keamanan data dari bukan pengguna?		
Total			

Rancangan pengujian pengguna diserahkan kepada pengguna ketika implementasi sistem telah dilakukan. Hasil pengujian dapat dianalisis untuk ditarik kesimpulan terkait penelitian ini.

5.2 Implementasi Sistem

Implementasi sistem merupakan tahapan penerapan rancangan sistem yang diimplementasikan ke dalam unit-unit program untuk dirangkai menjadi satu (Sommerville, 2016). Pada tahapan ini dilakukan implementasi sistem dari beberapa bagian, yaitu: *database*, fungsi dan antarmuka pengguna dari sistem. Implementasi sistem pada penelitian ini menggunakan MySQL versi 5.7, PHP versi 7.2 untuk menjalankan Laravel versi 7.28 dan Bootstrap versi 4.5 sebagai kerangka kerja *frontend*.

5.2.1 Implementasi Database

Database merupakan sekumpulan data yang dikelola menggunakan *database management system* (DBMS) dan data disimpan untuk jangka waktu yang lama (Garcia-Molina, Ullman & Widom, 2008). Implementasi *database* dilakukan menggunakan fitur *migration* dari Laravel untuk menerapkan struktur tabel sesuai dengan rancangan pada Gambar 5.13.

5.2.1.1 Implementasi Tabel Roles

Implementasi tabel roles menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.3 berikut ini.

Tabel 5.3 Implementasi Tabel Roles

No.	Kode
1	Schema::create('roles', function (Blueprint \$table) {
2	\$table->tinyIncrements('id');
3	\$table->string('name', 30);
4	});

Tabel roles digunakan untuk menyimpan peran-peran yang mungkin dimiliki oleh petugas yang terlibat pada sistem. Tabel roles menggunakan *id* sebagai *primary key* dengan tipe data *tiny integer* serta fungsi *auto increment*.

5.2.1.2 Implementasi Tabel Users

Implementasi tabel users menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.4 berikut ini.

Tabel 5.4 Implementasi Tabel Users

No.	Kode
1	Schema::create('users', function (Blueprint \$table) {
2	\$table->smallIncrements('id');
3	\$table->string('name', 100);
4	\$table->string('username', 16)->unique();
5	\$table->string('email', 255)->unique();
6	\$table->timestamp('email_verified_at')->nullable();
7	\$table->string('password', 128);
8	\$table->unsignedTinyInteger('role_id');
9	\$table->rememberToken();
10	\$table->timestamps();
11	\$table->foreign('role_id')->references('id')->on('roles');
12	});

Tabel *users* merupakan tabel yang merepresentasikan petugas sebagai pengguna dari sistem dengan menggunakan tabel yang sudah disediakan Laravel. Tabel *users* menggunakan *id* sebagai *primary key* dengan tipe data *small integer* serta fungsi *auto increment*. *Foreign key* pada tabel ini adalah *role_id* yang merujuk pada tabel *roles* untuk menunjukkan peran yang dimiliki oleh pengguna terkait.

5.2.1.3 Implementasi Table Notifications

Implementasi tabel *notifications* menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.5 berikut ini.

Tabel 5.5 Implementasi Tabel Notifications

No.	Kode
1	Schema::create('notifications', function (Blueprint \$table) {
2	\$table->integerIncrements('id');
3	\$table->string('title', 30);
4	\$table->string('redirect', 30);
5	\$table->boolean('is_read');
6	\$table->string('description', 255);
7	\$table->unsignedSmallInteger('user_id');
8	\$table->timestamps();
9	\$table->foreign('user_id')->references('id')->on('users');
10	});

Tabel *notifications* digunakan untuk menyimpan data pemberitahuan untuk petugas yang disimpan pada tabel *users*. Tabel *notifications* menggunakan *id* sebagai *primary key* dengan tipe data *integer* dan fungsi *auto increment*. Tabel *notifications* memiliki *foreign key* berupa *user_id* yang menunjukkan pengguna yang memiliki notifikasi tersebut.

5.2.1.4 Implementasi Tabel Kecamatan

Implementasi tabel kecamatan menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.6 berikut ini.

Tabel 5.6 Implementasi Tabel Kecamatan

No.	Kode
1	Schema::create('kecamatan', function (Blueprint \$table) {
2	\$table->tinyIncrements('id');
3	\$table->string('nama', 30);
4	});

Tabel kecamatan merupakan tabel untuk menyimpan data wilayah kecamatan yang terkait dengan pelanggan. Tabel kecamatan menggunakan *id* sebagai *primary key* dengan tipe data *tiny integer* dan fungsi *auto increment*.

5.2.1.5 Implementasi Tabel Desa

Implementasi tabel *roles* menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.7 berikut ini.

Tabel 5.7 Implementasi Tabel Desa

No.	Kode
1	Schema::create('desa', function (Blueprint \$table) {
2	\$table->tinyIncrements('id');
3	\$table->string('nama', 30);
4	\$table->unsignedTinyInteger('kecamatan_id');
5	\$table->foreign('kecamatan_id')->references('id')->
6	on('kecamatan');

Tabel desa merupakan tabel untuk menyimpan data wilayah desa yang terkait dengan pelanggan. Tabel desa menggunakan id sebagai *primary key* dengan tipe data *tiny integer* dan fungsi *auto increment*. Tabel desa memiliki *foreign key* kecamatan_id sebagai kolom yang memuat id dari kecamatan yang terkait dengan desa tersebut.

5.2.1.6 Implementasi Tabel Pelanggan

Implementasi tabel roles menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.8 berikut ini.

Tabel 5.8 Implementasi Tabel Pelanggan

No.	Kode
1	Schema::create('pelanggan', function (Blueprint \$table) {
2	\$table->integerIncrements('id');
3	\$table->string('nomor_pelanggan', 8)->unique();
4	\$table->string('nama', 100);
5	\$table->unsignedTinyInteger('desa_id');
6	\$table->unsignedTinyInteger('rt');
7	\$table->unsignedTinyInteger('rw');
8	\$table->string('keterangan_alamat', 255);
9	\$table->decimal('nominal_dasar', 10, 2);
10	\$table->string('keterangan_pelanggan', 30);
11	\$table->boolean('is_aktif');
12	\$table->timestamps();
13	\$table->foreign('desa_id')->references('id')->on('desa');
14	});

Tabel pelanggan merupakan tabel yang memuat data pelanggan beserta informasi terkait pelanggan tersebut. Tabel pelanggan menggunakan id sebagai *primary key* dengan tipe data *integer* dan fungsi *auto increment*. Pada tabel ini dimuat kolom nomor_pelanggan yang bersifat unik untuk masing-masing data pelanggan. *Foreign key* pada tabel pelanggan adalah kolom desa_id yang merujuk ke tabel desa untuk menunjukkan desa tempat pemasangan tv kabel milih pelanggan.

5.2.1.7 Implementasi Tabel Penagihan

Implementasi tabel roles menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.9 berikut ini.

Tabel 5.9 Implementasi Tabel Penagihan

No.	Kode
1	Schema::create('penagihan', function (Blueprint \$table) {
2	\$table->unsignedInteger('pelanggan_id');
3	\$table->unsignedSmallInteger('petugas_id');
4	\$table->primary('pelanggan_id');
5	\$table->foreign('pelanggan_id')->references('id')->
6	on('pelanggan');
7	\$table->foreign('petugas_id')->references('id')->on('users');
	});

Tabel penagihan merupakan tabel yang memuat penagihan untuk masing-masing pelanggan ditagihkan kepada petugas atau kantor berdasarkan permintaan pelanggan. Tabel penagihan menggunakan pelanggan_id sebagai *primary key* dengan tipe data *integer* dan juga sebagai *foreign key* yang merujuk kepada tabel *users*. Selain itu *foreign key* juga pada kolom petugas_id yang menunjukkan petugas yang bertanggung jawab atas penagihan pelanggan.

5.2.1.8 Implementasi Tabel Bulan

Implementasi tabel roles menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.10 berikut ini.

Tabel 5.10 Implementasi Tabel Bulan

No.	Kode
1	Schema::create('bulan', function (Blueprint \$table) {
2	\$table->tinyIncrements('id');
3	\$table->string('nama', 20);
4	\$table->string('nama_singkat', 5);
5	\$table->unsignedTinyInteger('angka_bulan');
6	});

Tabel bulan merupakan tabel yang menyimpan data waktu berupa bulan untuk menjadi acuan tagihan. Tabel bulan menggunakan id sebagai *primary key* dengan tipe data *tiny integer* dan fungsi *auto increment*.

5.2.1.9 Implementasi Tabel Status Tagihan

Implementasi tabel roles menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.11 berikut ini.

Tabel 5.11 Implementasi Tabel Status Tagihan

No.	Kode
1	Schema::create('status_tagihan', function (Blueprint \$table) {
2	\$table->tinyIncrements('id');
3	\$table->string('status', 30);
4	});

Tabel status_tagihan merupakan tabel yang menyimpan status dari tagihan. Tabel status_tagihan menggunakan id sebagai *primary key* dengan tipe data *tiny integer* dan fungsi *auto increment*.

5.2.1.10 Implementasi Tabel Tagihan

Implementasi tabel roles menggunakan fitur *migration* dari Laravel dapat dilihat pada Tabel 5.12 berikut ini.

Tabel 5.12 Implementasi Tabel Tagihan

No.	Kode
1	Schema::create('tagihan', function (Blueprint \$table) {
2	\$table->unsignedInteger('pelanggan_id');
3	\$table->unsignedTinyInteger('bulan_id');
4	\$table->unsignedSmallInteger('tahun');
5	\$table->decimal('nominal_bayar', 10, 2);
6	\$table->unsignedTinyInteger('status_id')->default('1');
7	\$table->boolean('slip_tercetak')->default('0');
8	\$table->unsignedSmallInteger('penerima_id')->nullable();
9	\$table->dateTime('waktu_bayar')->nullable();
10	\$table->unsignedSmallInteger('verifikator_id')->nullable();
11	\$table->dateTime('waktu_verifikasi')->nullable();
12	\$table->timestamps();
13	\$table->primary(['pelanggan_id', 'bulan_id', 'tahun']);
14	\$table->foreign('pelanggan_id')->references('id')->on('pelanggan');
15	\$table->foreign('bulan_id')->references('id')->on('bulan');
16	\$table->foreign('status_id')->references('id')->on('status_tagihan');
17	\$table->foreign('penerima_id')->references('id')->on('users');
18	\$table->foreign('verifikator_id')->references('id')->on('users');
19	});

Tabel tagihan merupakan tabel yang menyimpan tagihan untuk pelanggan berdasarkan bulan dan tahun. Tabel tagihan menggunakan *composite primary key* pada struktur tabelnya, terbentuk dari kolom pelanggan_id, bulan_id dan tahun. Kolom pelanggan_id bertipe data *integer* yang menjadi *foreign key* dan merujuk ke tabel pelanggan. Kolom bulan_id bertipe data *tiny integer* yang menjadi *foreign key* dan merujuk ke tabel bulan. Kolom tahun bertipe data *small integer* untuk menyimpan angka tahun yang nilainya ribuan. *Foreign key* lainnya adalah kolom yang merujuk pada tabel *users* yaitu penerima_id untuk mencatat petugas yang menerima pembayaran dan verifikator_id untuk mencatat petugas yang memverifikasi pembayaran.

5.2.2 Implementasi Fungsi

Tahap implementasi fungsi merupakan tahap untuk penerapan logika dalam bentuk kode program untuk menjalankan sistem sesuai alur yang sudah dirancang. Implementasi fungsi menggunakan bahasa pemrograman PHP dan kerangka kerja Laravel dalam proses pengembangannya. Berikut merupakan potongan kode sumber dari fungsi-fungsi utama sistem.

5.2.2.1 Implementasi Fungsi Melihat Daftar Penagihan

Fungsi melihat daftar penagihan bertujuan untuk menampilkan daftar penagihan dari petugas terkait. Terdapat dua fungsi yang dilibatkan untuk alur melihat daftar penagihan dan dapat dilihat pada Tabel 5.13 berikut ini.

Tabel 5.13 Fungsi Melihat Daftar Penagihan

No.	Kode
1	public function index()
2	{
3	Gate::authorize('access-penagihan');
4	\$petugas = Role::find(2)->users;
5	
6	return view('penagihan.index', ['petugas' => \$petugas]);
7	}
8	
9	public function show(Request \$request)
10	{
11	Gate::authorize('access-penagihan');
12	\$pelanggan['kantor'] = Penagihan::where('petugas_id', null)-
13	>get()->pluck('pelanggan');
14	if (\$request->petugas_id == 'kantor') {
15	return view('penagihan.kantor', ['pelanggan' => \$pelanggan]);
16	}
17	\$petugas = User::find(\$request->petugas_id);
18	if (!\$petugas \$petugas->isAdmin()) {
19	abort(404);
20	}
21	\$pelanggan['penagihan'] = Penagihan::where('petugas_id',
22	\$request->petugas_id)->get()->pluck('pelanggan');
23	
24	return view('penagihan.list', ['petugas' => \$petugas, 'pelanggan'
25	=> \$pelanggan]);
26	}

Melihat daftar penagihan melibatkan 2 fungsi untuk menjalankan keseluruhan alurnya, yaitu fungsi index dan fungsi show. Fungsi index bertujuan untuk menampilkan daftar petugas penagihan untuk ditampilkan daftar penagihannya. Fungsi selanjutnya yang dilibatkan adalah fungsi show yang menampilkan daftar pelanggan yang dalam petugas penagihan dipilih. Data pelanggan yang diolah pada fungsi ini dikelompokkan menjadi data pelanggan dalam penagihan dan data pelanggan yang tidak dalam penagihan petugas manapun.

5.2.2.2 Implementasi Fungsi Menambah Pelanggan ke Daftar Penagihan

Fungsi menambah data pelanggan ke daftar penagihan bertujuan untuk menambahkan data pelanggan ke dalam daftar penagihan petugas agar petugas terkait memiliki akses ke data pelanggan tersebut. Fungsi menambah data pelanggan ke daftar penagihan dapat dilihat pada Tabel 5.14 berikut ini.

Tabel 5.14 Fungsi Menambah Pelanggan ke Daftar Penagihan

No.	Kode
1	public function add(Request \$request)
2	{
3	Penagihan::whereIn('pelanggan_id', \$request->tambah)-
4	>update(['petugas_id' => \$request->petugas_id]);
5	\$messages = array();
6	\$messages['added']['type'] = 'primary';
7	\$messages['added']['body'] = 'Data pelanggan telah ditambahkan ke
8	daftar penagihan!';
9	session(['messages' => \$messages]);
10	return true;
11	}

Menambah pelanggan ke daftar penagihan menggunakan fungsi *add* yang melakukan *update* terhadap data pelanggan yang dipilih pada halaman daftar penagihan. Pesan sukses ditampilkan dengan memanfaatkan *session*.

5.2.2.3 Implementasi Fungsi Menghapus Pelanggan dari Daftar Penagihan

Fungsi menghapus pelanggan dari daftar penagihan bertujuan untuk menghapus data pelanggan dari daftar penagihan petugas dan menjadi pelanggan tidak dalam penagihan ke rumah. Fungsi menghapus data pelanggan dari daftar penagihan dapat dilihat pada Tabel 5.15 berikut ini.

Tabel 5.15 Fungsi Menghapus Pelanggan dari Daftar Penagihan

No.	Kode
1	<code>public function remove(Request \$request)</code>
2	<code>{</code>
3	<code> \$penagihan::whereIn('pelanggan_id', \$request->hapus)-</code>
4	<code> >update(['petugas_id' => null]);</code>
5	<code> \$messages = array();</code>
6	<code> \$messages['removed']['type'] = 'danger';</code>
7	<code> \$messages['removed']['body'] = 'Data pelanggan telah dihapus ke</code>
8	<code> daftar penagihan!';</code>
9	<code> session(['messages' => \$messages]);</code>
10	<code> return true;</code>
11	<code>}</code>

Menghapus pelanggan dari daftar penagihan menggunakan fungsi *remove* yang melakukan *update* terhadap data pelanggan yang dipilih pada halaman daftar penagihan menjadi *null* atau tidak dalam penagihan. Pesan sukses ditampilkan dengan memanfaatkan *session*.

5.2.2.4 Implementasi Fungsi Melihat Data Tagihan Pelanggan

Fungsi melihat data tagihan pelanggan bertujuan untuk menampilkan tagihan pelanggan untuk dilakukan aksi pembayaran terhadap tagihan yang dibayar. Fungsi melihat data tagihan pelanggan dapat dilihat pada Tabel 5.16 berikut ini.

Tabel 5.16 Fungsi Melihat Data Tagihan Pelanggan

No.	Kode
1	<code>public function show(\$nomor_pelanggan)</code>
2	<code>{</code>
3	<code> \$pelanggan = Pelanggan::where('nomor_pelanggan',</code>
4	<code> \$nomor_pelanggan)->first();</code>
5	<code> if (Auth::user()->cannot('view', \$pelanggan)) {</code>
6	<code> abort(403);</code>
7	<code> } else if (\$pelanggan == null) {</code>
8	<code> abort(404);</code>
9	<code> }</code>
10	<code> \$tagihan = \$pelanggan->tagihan->groupBy('status.status');</code>
11	<code> return view('pelanggan.details', ['pelanggan' => \$pelanggan,</code>
12	<code> 'tagihan' => \$tagihan]);</code>
13	<code>}</code>

Melihat data tagihan pelanggan melibatkan fungsi *show* pada *controller* pelanggan yang menampilkan data pelanggan beserta tagihan pelanggan tersebut yang dikelompokkan berdasarkan status pembayarannya.

5.2.2.5 Implementasi Fungsi Mencetak Slip Tagihan

Fungsi mencetak slip tagihan bertujuan untuk mencetak slip tagihan yang dipilih oleh pengguna sehingga slip tagihan dapat dicetak melalui *printer* oleh pengguna. Fungsi mencetak slip tagihan dapat dilihat pada Tabel 5.17 berikut ini.

Tabel 5.17 Fungsi Mencetak Slip Tagihan

No.	Kode
1	public function index()
2	{
3	if (Auth::user()->isAdmin()) {
4	\$petugas = Role::find(2)->users;
5	return view('tagihan.index', ['petugas' => \$petugas]);
6	} else {
7	return route()->redirect('tagihan.show', Auth::user()->id);
8	}
9	}
10	
11	public function show(Request \$request)
12	{
13	if (\$request->petugas_id == 'kantor') {
14	\$pelanggan_id = Penagihan::where('petugas_id', null)-
15	>pluck('pelanggan_id');
16	} else {
17	\$petugas_id = Penagihan::where('petugas_id', \$request-
18	>petugas_id)->pluck('pelanggan_id');
19	} else {
20	\$petugas = User::find(\$request->petugas_id);
21	\$tagihan = Tagihan::whereIn('pelanggan_id', \$pelanggan_id)-
22	>get()->groupBy(['status.status']);
23	}
24	return view('tagihan.list', ['tagihan' => \$tagihan, 'petugas' =>
25	\$petugas]);
26	}
27	public function print(Request \$request)
28	{
29	\$json_request = json_decode(\$request->print);
30	\$tagihan = [];
31	foreach (\$json_request as \$print) {
32	\$current = Tagihan::where('pelanggan_id', \$print-
33	>pelanggan_id)->where('bulan_id', \$print->bulan_id)-
34	>where('tahun', \$print->tahun)->first();
35	\$current->markAsPrinted();
36	array_push(\$tagihan, \$current);
37	}
38	}
39	
40	\$pdf = PDF::loadView('tagihan.print', ['tagihan' => \$tagihan]);
41	return \$pdf->stream('tagihan-pdf');
42	}

Mencetak slip tagihan melibatkan 3 fungsi dalam menjalankan keseluruhan alurnya, yaitu fungsi *index*, *show* dan *print*. Fungsi *index* berperan untuk menampilkan daftar petugas yang mungkin untuk memuat tagihan. Fungsi *show* bertujuan untuk menampilkan keseluruhan tagihan yang bisa dicetak. Kemudian fungsi *print* bertujuan untuk melakukan *generate* laporan berdasarkan bulan yang dipilih oleh pengguna. Fungsi *print* memanfaatkan *pdf* untuk membuat dokumen agar dapat dicetak oleh *browser* ke perangkat *printer*.



5.2.2.6 Implementasi Fungsi Mencatat Pembayaran Tagihan

Fungsi mencatat pembayaran tagihan bertujuan untuk memperbarui status tagihan pembayaran sesuai dengan kondisi pembayaran. Fungsi mencatat pembayaran tagihan dapat dilihat pada Tabel 5.18 berikut ini.

Tabel 5.18 Fungsi Mencatat Pembayaran Tagihan

No.	Kode
1	public function bayar(Request \$request)
2	{
3	\$petugas = \$request->user();
4	\$pelanggan = Pelanggan::where('nomor_pelanggan', \$request->nomor_pelanggan)->first();
5	\$tagihan = \$pelanggan->tagihan()->where(['bulan_id', \$request->bulan_id], ['tahun', \$request->tahun])->first();
6	
7	\$tagihan->bayar();
8	
9	if (\$petugas->role->id === 1) {
10	\$tagihan->verifikasi();
11	}
12	
13	\$messages = array();
14	\$messages['paid']['type'] = 'primary';
15	\$messages['paid']['body'] = 'Tagihan telah dibayar';
16	session(['messages' => \$messages]);
17	
18	return redirect()->back();
19	}

Mencatat pembayaran tagihan melibatkan fungsi bayar yang melakukan *update* terhadap tagihan yang dituju. Fungsi bayar mencakup fitur untuk melakukan pembayaran dan pembayaran di kantor melalui seleksi kondisi pada kode program fungsi bayar. Fungsi ini melakukan *update* terhadap status pembayaran menjadi menunggu verifikasi, dan jika pengguna memiliki *role* admin maka status diperbarui kembali menjadi terbayar. Pesan sukses ditampilkan dengan memanfaatkan *session*.

5.2.2.7 Implementasi Fungsi Memverifikasi Pembayaran Tagihan

Fungsi memverifikasi pembayaran tagihan bertujuan untuk mengubah status pembayaran tagihan menjadi terbayar. Fungsi memverifikasi pembayaran tagihan dapat dilihat pada Tabel 5.19 berikut ini.

Tabel 5.19 Fungsi Memverifikasi Pembayaran Tagihan

No.	Kode
1	public function verifikasi(Request \$request)
2	{
3	Gate::authorize('verifikasi-tagihan');
4	\$tagihan = Tagihan::where('penerima_id', \$request->petugas_id)->where('status_id', 2)->get();
5	foreach (\$tagihan as \$tghn) {
6	\$tghn->verifikasi();
7	}
8	
9	\$messages = array();
10	
11	\$messages['verified']['type'] = 'primary';

```

12 $messages['verified']['body'] = 'Tagihan telah diverifikasi';
13 session(['messages' => $messages]);
14
15 return redirect()->back();
16 }
17
18 public function tolak(Request $request)
19 {
20     Gate::authorize('verifikasi-tagihan');
21     $tagihan = Tagihan::where('penerima_id', $request->petugas_id)-
22 >where('status_id', 2)->get();
23     foreach ($tagihan as $tghn) {
24         $tghn->tolak();
25     }
26     Notification::create([
27         'title' => "Pembayaran Ditolak",
28         'type' => "danger",
29         'redirect' => "tagihan",
30         'description' => $request->alasan_penolakan,
31         'user_id' => $request->petugas_id
32     ]);
33     $messages = array();
34     $messages['verified']['type'] = 'danger';
35     $messages['verified']['body'] = 'Tagihan telah ditolak';
36     session(['messages' => $messages]);
37
38     return redirect()->back();
39 }

```

Memverifikasi pembayaran tagihan melibatkan fungsi verifikasi yang mengubah status pembayaran dengan menggunakan model Tagihan. Masing-masing tagihan yang dicatat pada *request* diperbarui menggunakan pengulangan. Kemudian pesan sukses ditampilkan dengan memanfaatkan *session*.

5.2.2.8 Implementasi Fungsi Membuat Laporan Bulanan

Fungsi membuat laporan bulanan bertujuan untuk membuat laporan berdasarkan data pembayaran yang sudah dicatat sebelumnya. Fungsi membuat laporan bulanan dapat dilihat pada Tabel 5.20 berikut ini.

Tabel 5.20 Fungsi Membuat Laporan Bulanan

No.	Kode
1	public function print(Request \$request)
2	{
3	Gate::authorize('access-laporan');
4	\$bulan = Bulan::find(\$request->bulan);
5	\$tahun = \$request->tahun;
6	\$admin_id = Role::find(1)->users->pluck('id');
7	\$tagihan['Kantor'] = Tagihan::whereMonth('waktu_bayar', \$bulan->id)->where('status_id', 4)->whereIn('penerima_id', \$admin_id)->orderBy('bulan_id', 'asc')->get();
8	\$petugas = Role::find(2)->users;
9	foreach (\$petugas as \$ptgs) {
10	\$tagihan[\$ptgs->name] = Tagihan::whereMonth('waktu_bayar', \$bulan->id)->where('status_id', 4)->where('penerima_id', \$ptgs->id)->orderBy('bulan_id', 'asc')->get();
11	}
12	\$pdf = PDF::loadView('laporan.print', ['bulan' => \$bulan, 'tahun' => \$tahun, 'tagihan' => \$tagihan]);
13	return \$pdf->stream('laporan');
14	}

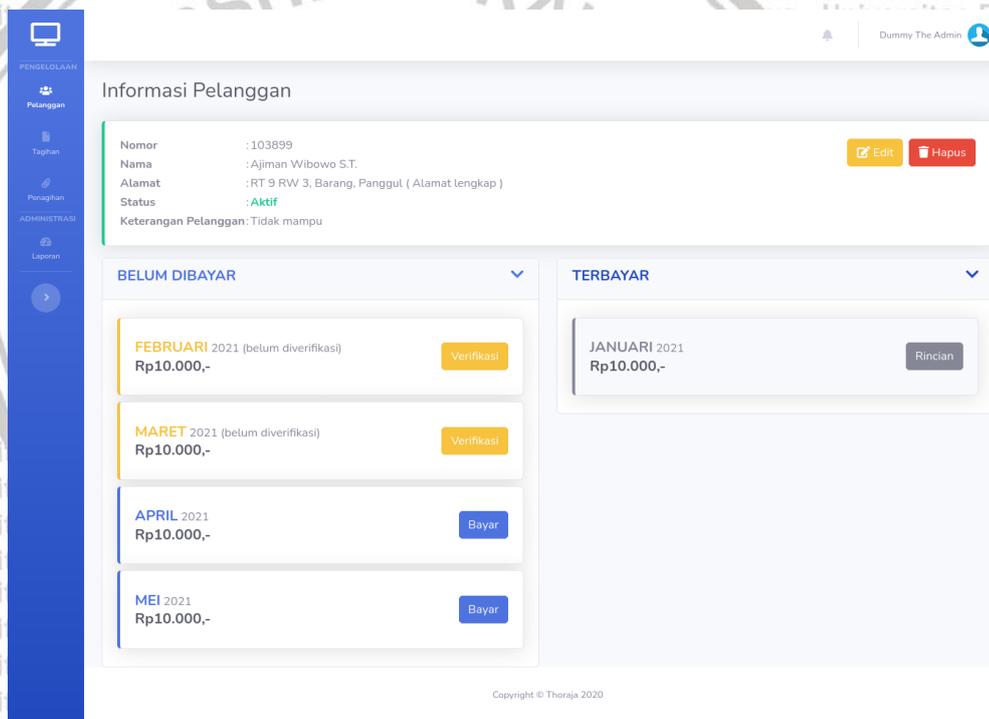
Membuat laporan bulanan melibatkan fungsi print yang mencetak laporan bulanan dalam bentuk dokumen PDF. Fungsi print melakukan seleksi data dari *request* yang dilakukan oleh pengguna yang berupa bulan dan tahun laporan. Tagihan yang diambil ada tagihan yang dibayar pada bulan dan tahun sesuai dengan masukan dari pengguna. PDF dibuat dan diserahkan kepada *browser* untuk dicetak melalui perangkat *printer* yang tersedia.

5.2.3 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna merupakan realisasi antarmuka pengguna dari rancangan antarmuka pada tahapan perancangan sistem. Antarmuka pengguna merupakan bagian sistem yang berinteraksi dengan pengguna untuk menerima masukan dan menampilkan informasi kepada pengguna.

5.2.3.1 Implementasi Antarmuka Halaman Detail Pelanggan

Implementasi antarmuka halaman detail pengguna merupakan realisasi dari rancangan antarmuka yang digambarkan pada Gambar 5.14. Implementasi antarmuka halaman detail pelanggan dapat dilihat pada Gambar 5.19 berikut ini.

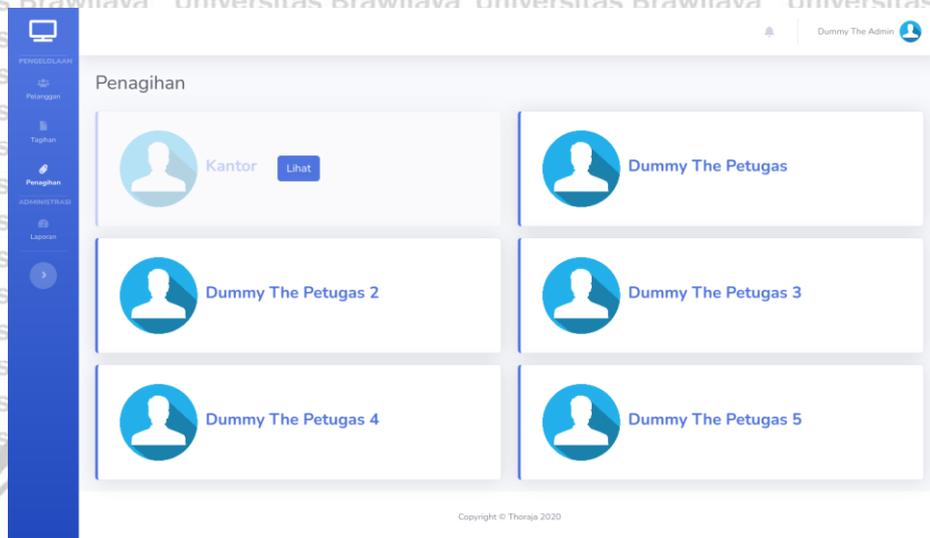


Gambar 5.19 Antarmuka Halaman Detail Pelanggan

Antarmuka detail pelanggan memuat data tagihan untuk pelanggan tersebut dengan pengelompokan berdasarkan status pembayaran. Semua tagihan ditampilkan pada seksi halaman yang terbagi dua untuk tagihan yang belum dibayar dan terbayar. Tagihan yang belum dibayar memiliki aksi yang dapat mengubah status pembayaran berdasarkan pengguna yang melakukan. Aksi-aksi pada halaman detail pelanggan disembunyikan berdasarkan pengguna yang mengakses halaman tersebut, aksi yang dimaksud seperti edit, hapus dan verifikasi dimana hal tersebut merupakan hak akses yang hanya dimiliki admin.

5.2.3.2 Implementasi Antarmuka Daftar Petugas Penagihan

Implementasi antarmuka daftar petugas penagihan merupakan realisasi dari rancangan antarmuka yang digambarkan pada Gambar 5.15. Implementasi antarmuka halaman daftar petugas penagihan dapat dilihat pada Gambar 5.20 berikut ini.

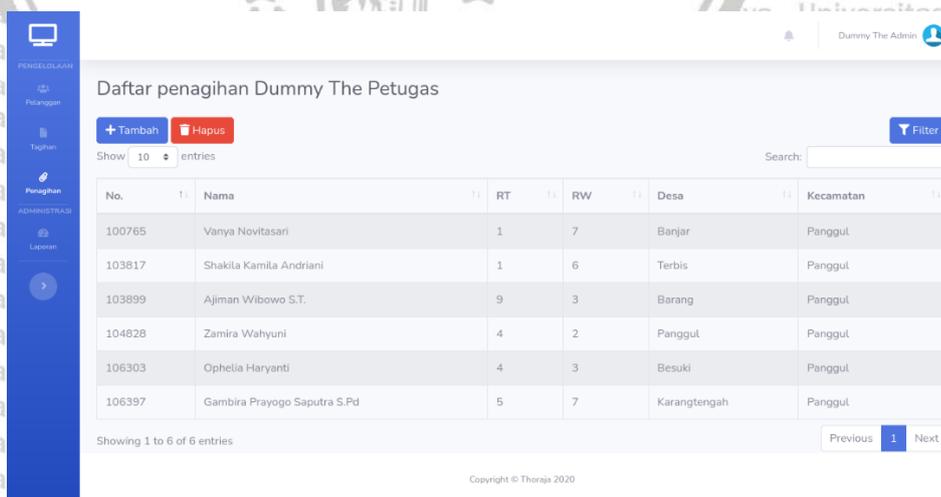


Gambar 5.20 Antarmuka Halaman Daftar Petugas Penagihan

Halaman antarmuka daftar petugas penagihan memuat seluruh pengguna yang memiliki peran sebagai petugas penagihan. Masing-masing elemen yang memuat petugas penagihan mengarah pada halaman penagihan petugas terkait.

5.2.3.3 Implementasi Antarmuka Daftar Penagihan

Implementasi antarmuka halaman daftar penagihan merupakan realisasi dari rancangan antarmuka yang digambarkan pada Gambar 5.16. Implementasi antarmuka halaman detail pelanggan dapat dilihat pada Gambar 5.21 berikut ini.

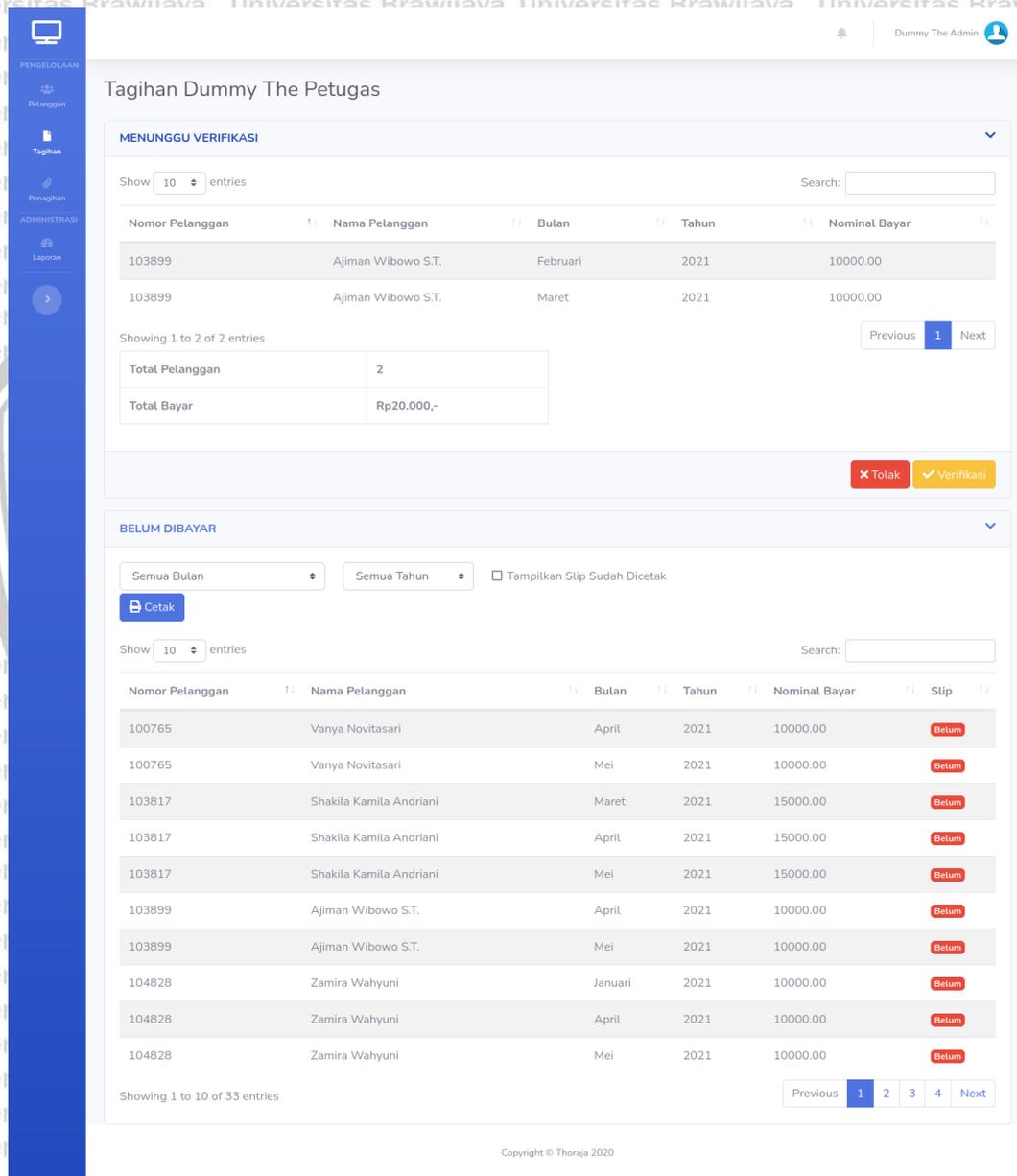


Gambar 5.21 Antarmuka Halaman Daftar Penagihan

Halaman antarmuka daftar penagihan menampilkan data pelanggan yang masuk ke dalam daftar penagihan petugas terkait. Elemen yang ada pada daftar penagihan memberikan akses kepada pengguna terhadap fitur manajemen daftar penagihan.

5.2.3.4 Implementasi Antarmuka Daftar Tagihan

Implementasi antarmuka daftar tagihan merupakan realisasi dari rancangan antarmuka yang digambarkan pada Gambar 5.17. Implementasi antarmuka halaman detail pelanggan dapat dilihat pada Gambar 5.22 berikut ini.



Gambar 5.22 Antarmuka Halaman Daftar Tagihan

Antarmuka halaman daftar tagihan memuat tagihan yang disorot untuk diperhatikan oleh pengguna. Pada bagian atas memuat tagihan yang memiliki status menunggu verifikasi. Penempatan tagihan menunggu verifikasi pada bagian

atas agar pengguna dapat memberikan perhatian pertama kali pada hal tersebut dan memberikan aksi pada tagihan tersebut. Pada bagian selanjutnya terdapat bagian tagihan yang memiliki status tagihan belum dibayar. Bagian ini memuat semua tagihan yang belum dibayar dan perlu dicetak slip tagihannya.

5.2.3.5 Implementasi Antarmuka Halaman Laporan Bulanan

Implementasi antarmuka daftar tagihan merupakan realisasi dari rancangan antarmuka yang digambarkan pada Gambar 5.18. Implementasi antarmuka halaman detail pelanggan dapat dilihat pada Gambar 5.23 berikut ini.

Kantor				
No. Pelanggan	Nama Pelanggan	Bulan	Tahun	Nominal Bayar
114030	Nasim Sihombing	Januari	2021	Rp10.000,-
137353	Wirda Ajeng Agustina	Januari	2021	Rp10.000,-
137523	Humaira Nasyidah	Januari	2021	Rp10.000,-
137957	Ratih Ade Yolanda	Januari	2021	Rp15.000,-
114030	Nasim Sihombing	Februari	2021	Rp10.000,-
137353	Wirda Ajeng Agustina	Februari	2021	Rp10.000,-
137523	Humaira Nasyidah	Februari	2021	Rp10.000,-
137957	Ratih Ade Yolanda	Februari	2021	Rp15.000,-
120561	Hafshah Aryani	Februari	2021	Rp15.000,-
114030	Nasim Sihombing	Maret	2021	Rp10.000,-
137353	Wirda Ajeng Agustina	Maret	2021	Rp10.000,-
120561	Hafshah Aryani	Maret	2021	Rp15.000,-
114030	Nasim Sihombing	April	2021	Rp10.000,-
120561	Hafshah Aryani	April	2021	Rp15.000,-
114030	Nasim Sihombing	Mei	2021	Rp10.000,-
120561	Hafshah Aryani	Mei	2021	Rp15.000,-
Total				Rp190.000,-
Dummy The Petugas				
No. Pelanggan	Nama Pelanggan	Bulan	Tahun	Nominal Bayar

Gambar 5.23 Antarmuka Halaman Laporan Bulanan

Antarmuka halaman laporan bulanan diterapkan dalam bentuk dokumen yang dapat dicetak, dalam penelitian ini berupa PDF. Halaman laporan bulanan memuat informasi laporan dan tabel yang memuat data tagihan yang diterima oleh masing-masing petugas penagihan.

BAB 6 PENGUJIAN

Bab ini membahas terkait pengujian sistem sebagai tahapan terakhir dari penelitian ini, karena pengujian merupakan hal yang tak bisa dipisahkan dari pemrograman, pemrograman sistem dikatakan selesai ketika telah melewati serangkaian pengujian (Dennis, 2012). Pengujian merupakan tahapan dalam SDLC untuk memastikan sistem yang sudah dikembangkan dari tahapan rekayasa kebutuhan, perancangan hingga implementasi, dapat bekerja sesuai fungsionalnya. Pengujian bertujuan untuk menemukan kesalahan sebanyak mungkin pada sistem yang telah dibuat dengan usaha yang sedikit mungkin (Pressman, 2010).

6.1 Pengujian Validasi

Menurut Sommerville (2016) *validation testing* merupakan pengujian melalui rangkaian *test case* untuk memastikan sistem telah bekerja sesuai rancangan. Oleh karena itu, pada pengujian ini didefinisikan skenario uji dengan kasus-kasus tertentu untuk melihat kesesuaian masing-masing fungsional sistem.

6.1.1 Skenario Uji Fungsi Melihat Daftar Penagihan

Skenario uji fungsi melihat daftar penagihan pada Tabel 6.1 berikut dibuat berdasarkan skenario *use case* melihat daftar penagihan pada Tabel 4.11.

Tabel 6.1 Skenario Uji Fungsi Melihat Daftar Penagihan

Kode Skenario Uji	TS-SIP-01	Kode Persyaratan Fungsional	PF-SIP-08
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi melihat daftar penagihan.		
Kasus Uji	Memeriksa keluaran sistem ketika fungsi melihat daftar penagihan diakses.		
Prakondisi	- Penguji telah masuk ke sistem sebagai administrator. - Penguji telah membuka layar utama sistem.		
Prosedur Uji	1. Penguji membuka menu daftar penagihan. 2. Penguji menekan tombol lihat pada salah satu petugas penagihan.		
Hasil yang Diharapkan	Sistem berhasil menampilkan halaman daftar penagihan untuk petugas terkait.		
Hasil Pengujian	Sistem berhasil menampilkan halaman daftar penagihan untuk petugas terkait.		
Status Pengujian	Valid		

6.1.2 Skenario Uji Fungsi Menambah Pelanggan ke Daftar Penagihan

Skenario uji fungsi menambah pelanggan ke daftar penagihan pada Tabel 6.2 berikut ini dibuat berdasarkan skenario *use case* menambah pelanggan ke daftar penagihan pada Tabel 4.12.

Tabel 6.2 Skenario Uji Fungsi Menambah Pelanggan ke Daftar Penagihan

Kode Skenario Uji	TS-SIP-02	Kode Persyaratan Fungsional	PF-SIP-09
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi menambah pelanggan ke daftar penagihan.		
Kasus Uji	Memeriksa hasil penyimpanan data pelanggan pada daftar penagihan setelah dilakukan penambahan.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem sebagai administrator. - Penguji telah membuka menu daftar penagihan. 		
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji menekan tombol tambah. 2. Penguji memilih beberapa data pelanggan dengan mencentang. 3. Penguji menekan tombol simpan. 4. Penguji menekan tombol konfirmasi. 		
Hasil yang Diharapkan	Sistem berhasil menyimpan data pelanggan yang dipilih pada daftar penagihan dan data baru dapat ditampilkan.		
Hasil Pengujian	Sistem berhasil menyimpan data pelanggan yang dipilih pada daftar penagihan dan data baru dapat ditampilkan.		
Status Pengujian	Valid		

6.1.3 Skenario Uji Fungsi Menghapus Pelanggan dari Daftar Penagihan

Skenario uji fungsi menghapus pelanggan ke daftar penagihan pada Tabel 6.3 berikut ini dibuat berdasarkan skenario *use case* pada Tabel 4.13.

Tabel 6.3 Skenario Uji Fungsi Menghapus Pelanggan dari Daftar Penagihan

Kode Skenario Uji	TS-SIP-03	Kode Persyaratan Fungsional	PF-SIP-10
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi menghapus pelanggan dari daftar penagihan		
Kasus Uji	Memeriksa hasil penyimpanan data pelanggan pada daftar penagihan setelah dilakukan penghapusan.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem sebagai administrator. - Penguji telah membuka menu daftar penagihan. 		
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji menekan tombol hapus. 2. Penguji memilih beberapa data pelanggan dengan mencentang. 3. Penguji menekan tombol hapus yang dipilih. 4. Penguji menekan tombol konfirmasi. 		
Hasil yang Diharapkan	Sistem berhasil menghapus data pelanggan yang dipilih pada daftar penagihan.		
Hasil Pengujian	Sistem berhasil menghapus data pelanggan yang dipilih pada daftar penagihan.		
Status Pengujian	Valid		

6.1.4 Skenario Uji Fungsi Melihat Data Tagihan Pelanggan

Skenario uji fungsi melihat data tagihan pelanggan pada Tabel 6.4 berikut ini dibuat berdasarkan skenario *use case* melihat data tagihan pada Tabel 4.14.

Tabel 6.4 Skenario Uji Fungsi Melihat Data Tagihan Pelanggan

Kode Skenario Uji	TS-SIP-04	Kode Persyaratan Fungsional	PF-SIP-11
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi melihat data tagihan pelanggan.		
Kasus Uji	Memeriksa keluaran sistem ketika fungsi melihat data tagihan pelanggan.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem. - Penguji telah membuka daftar pelanggan. 		
Prosedur Uji	1. Penguji memilih salah satu data pelanggan.		
Hasil yang Diharapkan	Sistem berhasil menampilkan halaman data tagihan untuk pelanggan yang dipilih.		
Hasil Pengujian	Sistem berhasil menampilkan halaman data tagihan untuk pelanggan yang dipilih.		
Status Pengujian	Valid		

6.1.5 Skenario Uji Fungsi Mencetak Slip Tagihan

Skenario uji fungsi mencetak slip tagihan pada Tabel 6.5 berikut ini dibuat berdasarkan skenario *use case* mencetak slip tagihan pada Tabel 4.15.

Tabel 6.5 Skenario Uji Fungsi Mencetak Slip Tagihan

Kode Skenario Uji	TS-SIP-05	Kode Persyaratan Fungsional	PF-SIP-12
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi mencetak slip tagihan.		
Kasus Uji	Memeriksa keluaran sistem untuk slip tagihan yang dipilih untuk dicetak.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem sebagai administrator. - Penguji telah membuka layar utama sistem. 		
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji membuka menu tagihan. 2. Penguji menekan tombol cetak pada seksti tagihan belum dibayar. 3. Penguji memilih beberapa tagihan yang akan dicetak. 4. Penguji menekan tombol cetak yang dipilih. 5. Penguji menekan tombol konfirmasi. 		
Hasil yang Diharapkan	Sistem berhasil membuat berkas yang memuat slip tagihan terpilih.		
Hasil Pengujian	Sistem berhasil membuat berkas yang memuat slip tagihan terpilih.		
Status Pengujian	Valid		

6.1.6 Skenario Uji Fungsi Mencatat Pembayaran Tagihan

Skenario uji fungsi mencatat pembayaran tagihan pada Tabel 6.6 berikut ini dibuat berdasarkan skenario *use case* mencatat pembayaran tagihan pada Tabel 5.18.

Tabel 6.6 Skenario Uji Fungsi Mencatat Pembayaran Tagihan

Kode Skenario Uji	TS-SIP-06	Kode Persyaratan Fungsional	PF-SIP-13
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi mencatat pembayaran tagihan.		
Kasus Uji	Memeriksa hasil penyimpanan status pembayaran tagihan pada sistem sebagai petugas penagihan.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem sebagai petugas penagihan. - Penguji telah membuka data tagihan pelanggan. 		
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji membuka menekan tombol bayar pada salah satu tagihan. 2. Penguji menekan tombol konfirmasi. 		
Hasil yang Diharapkan	Sistem berhasil mengubah status tagihan pembayaran menjadi belum diverifikasi.		
Hasil Pengujian	Sistem berhasil mengubah status tagihan pembayaran menjadi belum diverifikasi.		
Status Pengujian	Valid		

Skenario uji fungsi mencatat pembayaran tagihan di kantor pada Tabel 6.7 berikut ini dibuat berdasarkan skenario *use case* mencatat pembayaran tagihan di kantor pada Tabel 4.17.

Tabel 6.7 Skenario Uji Fungsi Mencatat Pembayaran Tagihan di Kantor

Kode Skenario Uji	TS-SIP-07	Kode Persyaratan Fungsional	PF-SIP-13
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi mencatat pembayaran tagihan di kantor.		
Kasus Uji	Memeriksa hasil penyimpanan status pembayaran tagihan pada sistem sebagai petugas administrator.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem sebagai administrator. - Penguji telah membuka data tagihan pelanggan. 		
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji membuka menekan tombol bayar pada salah satu tagihan. 2. Penguji menekan tombol konfirmasi. 		
Hasil yang Diharapkan	Sistem berhasil mengubah status tagihan pembayaran menjadi sudah dibayar.		
Hasil Pengujian	Sistem berhasil mengubah status tagihan pembayaran menjadi sudah dibayar.		
Status Pengujian	Valid		

6.1.7 Skenario Uji Fungsi Memverifikasi Pembayaran Tagihan

Skenario uji fungsi mencatat pembayaran tagihan pada Tabel 6.8 berikut ini dibuat berdasarkan skenario *use case* memverifikasi pembayaran tagihan pada Tabel 5.19 bagian *main flow*.

Tabel 6.8 Skenario Uji Fungsi Memverifikasi Pembayaran Tagihan

Kode Skenario Uji	TS-SIP-08	Kode Persyaratan Fungsional	PF-SIP-14
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi memverifikasi pembayaran tagihan.		
Kasus Uji	Memeriksa perubahan status pembayaran tagihan yang diterima.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem sebagai administrator. - Penguji telah membuka layar utama sistem 		
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji membuka menu tagihan. 2. Penguji menekan tombol verifikasi pada bagian tagihan menunggu verifikasi. 3. Penguji menekan tombol konfirmasi. 		
Hasil yang Diharapkan	Sistem berhasil mengubah status pembayaran menjadi sudah dibayar.		
Hasil Pengujian	Sistem berhasil mengubah status pembayaran menjadi sudah dibayar.		
Status Pengujian	Valid		

Skenario uji fungsi mencatat pembayaran tagihan alternatif 1 pada Tabel 6.9 berikut ini dibuat berdasarkan skenario *use case* memverifikasi pembayaran tagihan pada Tabel 5.19 bagian *alternative flow 3a*.

Tabel 6.9 Skenario Uji Fungsi Memverifikasi Pembayaran Tagihan Alternatif 1

Kode Skenario Uji	TS-SIP-09	Kode Persyaratan Fungsional	PF-SIP-14
Skenario Uji	Skenario 2: <i>Alternative flow 3a</i>		
Nama Uji	Pengujian fungsi memverifikasi pembayaran tagihan alternatif 1.		
Kasus Uji	Memeriksa perubahan status pembayaran tagihan yang ditolak.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem sebagai administrator. - Penguji telah membuka layar utama sistem 		
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji membuka menu tagihan. 2. Penguji menekan tombol tolak pada bagian tagihan menunggu verifikasi. 3. Penguji menekan tombol konfirmasi. 		
Hasil yang Diharapkan	Sistem berhasil mengubah status pembayaran menjadi ditolak.		
Hasil Pengujian	Sistem berhasil mengubah status pembayaran menjadi ditolak.		
Status Pengujian	Valid		

6.1.8 Skenario Uji Fungsi Membuat Laporan Bulanan

Skenario uji fungsi membuat laporan bulanan pada Tabel 6.10 berikut ini dibuat berdasarkan skenario *use case* membuat laporan bulanan pada Tabel 5.20 bagian *main flow*.

Tabel 6.10 Skenario Uji Fungsi Membuat Laporan Bulanan

Kode Skenario Uji	TS-SIP-10	Kode Persyaratan Fungsional	PF-SIP-15
Skenario Uji	Skenario 1: <i>Main flow</i>		
Nama Uji	Pengujian fungsi mencetak laporan bulanan.		
Kasus Uji	Memeriksa keluaran sistem ketika laporan bulanan dicetak.		
Prakondisi	<ul style="list-style-type: none"> - Penguji telah masuk ke sistem sebagai administrator. - Penguji telah membuka layar utama sistem 		
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji membuka menu laporan. 2. Penguji menekan tombol cetak. 		
Hasil yang Diharapkan	Sistem berhasil membuat berkas laporan bulanan.		
Hasil Pengujian	Sistem berhasil membuat berkas laporan bulanan.		
Status Pengujian	Valid		

6.2 Hasil Pengujian Validasi

Skenario uji yang sudah dilakukan menghasilkan pemetaan hasil uji validasi yang dapat dilihat pada Tabel 6.11 berikut ini.

Tabel 6.11 Hasil Pengujian Validasi

Kode Skenario Uji	Kode Skenario <i>Use Case</i>	Kode Fungsional	Valid	
			Ya	Tidak
TS-SIP-01	US-SIP-08	PF-SIP-08	✓	
TS-SIP-02	US-SIP-09	PF-SIP-09	✓	
TS-SIP-03	US-SIP-10	PF-SIP-10	✓	
TS-SIP-04	US-SIP-11	PF-SIP-11	✓	
TS-SIP-05	US-SIP-12	PF-SIP-12	✓	
TS-SIP-06	US-SIP-13	PF-SIP-13	✓	
TS-SIP-07	US-SIP-14	PF-SIP-13	✓	
TS-SIP-08	US-SIP-15	PF-SIP-14	✓	
TS-SIP-09	US-SIP-15	PF-SIP-14	✓	
TS-SIP-10	US-SIP-16	PF-SIP-15	✓	
Total			10	0

Berdasarkan tabel hasil pengujian validasi, dapat dilihat bahwa seluruh skenario uji valid terhadap fungsional yang telah digali pada tahapan rekayasa kebutuhan. Hasil dari pengujian validasi menunjukkan bahwa seluruh persyaratan sistem telah diimplementasikan ke dalam sistem informasi pembayaran dengan benar.

6.3 Pengujian Penerimaan Pengguna

Pengujian penerimaan pengguna dilakukan berdasarkan rancangan pengujian penerimaan pengguna yang sudah dibuat pada tahap perancangan. Rancangan pengujian diberikan kepada pengguna untuk diisi jawaban sesuai dengan pengujian pengguna terhadap sistem yang telah dikembangkan. Jawaban pengujian penerimaan pengguna dicantumkan LAMPIRAN C.

6.4 Hasil Pengujian Penerimaan Pengguna

Pengujian penerimaan pengguna oleh administrator berdasarkan pada rancangan yang sudah dibuat dan jawaban pengguna yang dapat dilihat pada lampiran. Berdasarkan jawaban pengguna dapat dibuat tabel hasil pengujian penerimaan pengguna yang ditunjukkan pada Tabel 6.12 berikut ini.

Tabel 6.12 Hasil Pengujian Penerimaan Pengguna oleh Administrator

No.	Kriteria	Jumlah Pertanyaan	Jawaban "Ya"	Jawaban "Tidak"	Persentase Jawaban "Ya"
1	<i>Functional Correctness and Completeness</i>	3	3	0	100%
2	<i>Usability</i>	3	3	0	100%
3	<i>Confidentiality and Availability</i>	2	2	0	100%

Hasil pengujian pada Tabel 6.12 menunjukkan bahwa kriteria *functional correctness and completeness* menghasilkan 100% jawaban "ya", yang berarti seluruh fungsional sistem lengkap dan sesuai dengan kebutuhan administrator sebagai pengguna. Hasil dari kriteria *usability* menunjukkan 100% jawaban "ya" untuk masing-masing pada kriteria ini yang berarti sistem yang dikembangkan sudah cukup mudah bagi administrator sebagai pengguna. Kriteria *confidentiality and availability* menunjukkan hasil 100% jawaban "ya" yang berarti sistem telah menyediakan data dan menjaga data berdasarkan pembatasan hak akses yang ditentukan.

Pengujian penerimaan pengguna oleh petugas penagihan berdasarkan pada rancangan yang sudah dibuat dan jawaban pengguna yang dapat dilihat pada lampiran. Berdasarkan jawaban pengguna dapat dibuat tabel hasil pengujian penerimaan pengguna yang ditunjukkan pada Tabel 6.13 berikut ini.

Tabel 6.13 Hasil Pengujian Penerimaan Pengguna oleh Petugas Penagihan

No.	Kriteria	Jumlah Pertanyaan	Jawaban "Ya"	Jawaban "Tidak"	Persentase Jawaban "Ya"
1	Functional Correctness and Completeness	1	1	0	100%
2	Usability	3	3	0	100%
3	Confidentiality and Availability	2	2	0	100%

Hasil pengujian pada Tabel 6.13 menunjukkan bahwa kriteria *functional correctness and completeness* menghasilkan 100% jawaban "ya", yang berarti fungsi pembayaran telah sesuai dengan kebutuhan petugas penagihan untuk sebagai pengguna. Hasil dari kriteria *usability* menunjukkan 100% jawaban "ya" untuk masing-masing pada kriteria ini yang berarti sistem yang dikembangkan sudah cukup mudah bagi petugas penagihan sebagai pengguna. Kriteria *confidentiality and availability* menunjukkan hasil 100% jawaban "ya" yang berarti sistem telah menyediakan data dan menjaga data berdasarkan pembatasan hak akses yang ditentukan.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan hal-hal terkait dengan permasalahan yang dirumuskan pada penelitian ini. Kesimpulan dari penelitian ini sebagai berikut:

1. Hasil analisis kebutuhan menunjukkan menghasilkan 4 proses bisnis rekomendasi yang melibatkan sistem informasi pembayaran sebagai penunjang pemangku kepentingan menjalankan aktivitas pembayaran. Kemudian analisis kebutuhan mengidentifikasi 2 peran sebagai pengguna, 4 fitur utama, 15 persyaratan fungsional dan 1 persyaratan non-fungsional. Diagram *use case* diidentifikasi berdasarkan persyaratan fungsional dan menghasilkan masing-masing 16 *use case*, skenario *use case* dan *activity diagram*.
2. Hasil perancangan sistem dilakukan berdasarkan hasil analisis kebutuhan sistem dengan struktur *model-view-controller* (MVC). Alur jalannya sistem digambarkan dengan *sequence diagram* melalui urutan interaksi antara aktor dengan sistem dan interaksi antar komponen di dalam sistem. Elemen-elemen pada *sequence diagram* digambarkan ke dalam kelas-kelas pada *class diagram*. Diidentifikasi terdapat 4 kelas *controller*, 4 kelas utama untuk model dan 6 kelas pendukung untuk model. Kemudian kelas-kelas model dipetakan menjadi 10 tabel di dalam *relational data model* untuk dijadikan dasar implementasi *database*. *View* pada MVC dirancang dalam perancangan antarmuka dan terdapat 5 rancangan antarmuka pengguna untuk fitur utama.
3. Implementasi sistem informasi pembayaran dilakukan menggunakan PHP versi 7.2 sebagai bahasa pemrograman, MySQL versi 5.7 sebagai DBMS, Laravel versi 7.28 sebagai kerangka kerja *backend* dan Bootstrap versi 4.5 sebagai kerangka kerja *frontend*. Implementasi basis data menggunakan fitur *migration* pada Laravel dengan DBMS MySQL untuk 10 tabel yang telah dirancang. Fungsi-fungsi diimplementasikan menggunakan PHP dan Laravel sebagai kerangka kerjanya dan menghasilkan 12 fungsi PHP untuk implementasi fungsi utama sistem. Bootstrap digunakan untuk implementasi antarmuka pengguna dan terdapat 5 halaman untuk fitur utama sistem.
4. Pengujian sistem informasi pembayaran menghasilkan 10 dari 10 skenario uji untuk pengujian validasi bernilai valid dan dapat disimpulkan seluruh fungsi sistem telah sesuai dengan persyaratan sistem. Kemudian pengujian penerimaan pengguna menunjukkan nilai 100% untuk jawaban ya pada masing-masing kriteria penerimaan pengguna yang berarti sistem dapat diterima untuk dilakukan instalasi pada lingkungan pengguna.



7.2 Saran

Saran dari penelitian ini dapat digunakan sebagai acuan dan pertimbangan untuk melakukan penelitian terkait pengembangan lanjut pada sistem informasi pembayaran. Saran dari penelitian ini adalah:

1. Fitur yang dapat dikembangkan lebih lanjut adalah transaksi pembayaran secara digital yang terintegrasi dengan layanan pembayaran digital atau layanan pembayaran bank sebagai fitur yang meningkatkan efisiensi verifikasi pembayaran.
2. Pengembangan aplikasi pembayaran berbasis *mobile* diperlukan untuk kemudahan mobilisasi petugas penagihan dalam melakukan pencatatan pembayaran ke rumah-rumah.
3. Pengembangan lanjut dapat dilakukan pada sistem pengelolaan keuangan yang terintegrasi dengan sistem pencatatan pembayaran sehingga catatan pemasukan dan pengeluaran pada laporan bulan lebih terperinci.



DAFTAR REFERENSI

- Aagesen, G. & Krogstie, J., 2015. BPMN 2.0 for Modeling Business Processes. *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, hal.219–250. https://doi.org/10.1007/978-3-642-45100-3_10.
- Akachain.io, 2020. *Top 5 criteria of a great traceability system that blockchain can upgrade*. [daring] Tersedia pada: <<https://akachainio.medium.com/top-5-criteria-of-a-great-traceability-system-that-blockchain-can-upgrade-17d98954acb5>> [Diakses 3 Des 2020].
- Ali, K., 2017. A Study of Software Development Life Cycle Process Models. *International Journal of Advanced Research in Computer Science*, 8(1), hal.2–7.
- Amin, M., Alauddin, M. & Azad, D., 2012. Business Transaction Processing System. *International Journal of Computer Information Systems*, 4, hal.11–15.
- Arif, M.Z.U., Rahaman, D.M. & Uddin, M., 2012. Role of Management Information Systems in Managerial Decision Making of Organizations in the Global Business World. 2, hal.14–18.
- Barjis, J., 2008. The importance of business process modeling in software systems design. *Science of Computer Programming*, [daring] 71(1), hal.73–87. <https://doi.org/https://doi.org/10.1016/j.scico.2008.01.002>.
- Bourgeois, D. & Bourgeois, D.T., 2014. *Chapter 4: Data and Databases*. Tersedia pada: <<https://bus206.pressbooks.com/chapter/chapter-4-data-and-databases/>> [Diakses 3 Nov 2020].
- Butlewski, M., Jasiulewicz-Kaczmarek, M., Misztal, A. & Sławińska, M., 2014. Design methods of reducing human error in practice. hal.1101–1106. <https://doi.org/10.1201/b17399-154>.
- Chand, M., 2019. *Most Popular Databases In The World*. [daring] Tersedia pada: <<https://www.c-sharpcorner.com/article/what-is-the-most-popular-database-in-the-world/>> [Diakses 7 Nov 2020].
- Das, R.P. & Saikia, D.L.P., 2016. Comparison of Procedural PHP with Codeigniter and Laravel Framework.
- Dennis, A., 2012. *Systems Analysis and Design*. 5th ed. Wiley Publishing.
- Dumas, M., Rosa, M. La, Mendling, J. & Reijers, H.A., 2018. *Fundamentals of Business Process Management*. 2nd ed. Springer Publishing Company, Incorporated.
- Garcia-Molina, H., Ullman, J.D. & Widom, J., 2008. *Database Systems: The Complete Book*. 2 ed. USA: Prentice Hall Press.

Habeeb, A., 2018. *Introduction to Unified Modeling Language*.
<https://doi.org/10.13140/RG.2.2.28706.32963>.

Leung, H.K.N. & Wong, P.W.L., 1997. A study of user acceptance tests. *Software Quality Journal*, [daring] 6(2), hal.137–149.
<https://doi.org/10.1023/A:1018503800709>.

Lipaj, D. & Davidavičienė, V., 2013. Influence of Information Systems on Business Performance. 5. <https://doi.org/10.3846/mla.2013.06>.

Malhotra, M., 2019. *Why Laravel Is The Best PHP Framework In 2019?* [daring] Tersedia pada: <<https://www.valuecoders.com/blog/technology-and-apps/laravel-best-php-framework-2017/>> [Diakses 3 Des 2020].

Marakas, G. & O'Brien, J., 2010. *Introduction to Information Systems*. Fifteenth ed.

Moraszki, K., József, M. & Attila, L., 2015. *Customer Satisfaction or How We Can Keep Satisfied Customer*.

MySQL, 2020. MySQL 8.0 Reference Manual. *MySQL 8.0 Reference Manual*, [daring] 1, hal.1692. Tersedia pada: <<https://dev.mysql.com/doc/relnotes/mysql/8.0/en/>>.

Naik, K. & Tripathy, P., 2018. *Software Testing and Quality Assurance: Theory and Practice*. 2nd ed. Wiley Publishing.

Natschläger, C., 2011. Towards a BPMN 2.0 Ontology BT - Business Process Model and Notation. In: R. Dijkman, J. Hofstetter & J. Koehler, ed. Berlin, Heidelberg: Springer Berlin Heidelberg. hal.1–15.

OMG, 2011. *Business Process Model and Notation (BPMN)*.

Otwell, T., 2020. Laravel Documentation - 6.x. *Driade Laravel-Book*, [daring] (February 7), hal.1–495. Tersedia pada: <<https://github.com/driade/laravel-book/blob/master/laravel-docs-6.x.pdf>>.

PHP Documentation Group, 2012. PHP Manual. [daring] hal.undefined-undefined. Tersedia pada: <www.php.net/manual/> [Diakses 23 Jun 2020].

Pressman, R.S., 2010. *Software Engineering: A Practitioner's Approach*. McGraw-Hill higher education. [daring] McGraw-Hill Education. Tersedia pada: <https://books.google.co.id/books?id=y4k_AQAAIAAJ>.

Ramadhan, I., 2019. Analisis Dan Perancangan Sistem Pembayaran Jasa Internet Berbasis Web Dan SMS Gateway. *Systematics*, 1(1), hal.1. <https://doi.org/10.35706/sys.v1i1.1976>.

Ryser, J. & Glinz, M., 2000. SCENT: A Method Employing Scenarios to Systematically Derive Test Cases for System Test.

Sommerville, I., 2016. *Software engineering (10th edition)*. Pearson Education Limited.

Surniandari, A. & Rachmi, H., 2020. Sistem Informasi Pengelolaan Transaksi Pembayaran dengan Metode Waterfall Untuk Kasir Event Multi UKM. *Jurnal Teknik Komputer*, 6, hal.53–60. <https://doi.org/10.31294/jtk.v6i1.6762>.

Sutoyo, I., 2020. Perancangan Sistem Informasi Pembayaran SPP Terpadu Menggunakan Model Prototipe. *INTI Nusa Mandiri*, [daring] 14(2), hal.145–152. <https://doi.org/10.33480/inti.v14i2.1125>.

Visual Paradigm, 2020. *What is Activity Diagram?* [daring] Tersedia pada: <<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>> [Diakses 3 Des 2020].



LAMPIRAN A RANCANGAN WAWANCARA

A.1 Draf Wawancara Latar Belakang Permasalahan

Lampiran ini berupa rancangan wawancara untuk menggali latar belakang permasalahan yang terjadi pada Linda Cable. Berikut daftar pertanyaan yang diajukan kepada narasumber untuk menggali permasalahan:

Nama Narasumber	:	
Jabatan Narasumber	:	
Tanggal	:	
Tempat	:	

No.	Pertanyaan
1.	Apakah Linda Cable sudah menerapkan sistem informasi untuk mengelola pembayaran? Jawaban:
2.	Bagaimana kendala yang terjadi saat melakukan kegiatan pembayaran atau penagihan saat ini? Jawaban:
3.	Seberapa sering kendala atau masalah pada pertanyaan poin 2 terjadi? Jawaban:
4.	Apakah Linda Cable memiliki rencana kedepan mengenai pengelolaan pembayaran? Jawaban:

A.2 Draf Wawancara Kebutuhan

Lampiran ini berupa rancangan wawancara untuk menggali kebutuhan sistem yang digunakan untuk pengembangan sistem informasi pembayaran pada penelitian ini. Berikut daftar pertanyaan yang diajukan kepada narasumber untuk menggali kebutuhan:

Nama Narasumber	:	
Jabatan Narasumber	:	
Tanggal	:	
Tempat	:	

No.	Pertanyaan
1.	<p>Bagaimana alur proses penagihan yang berjalan di Linda Cable?</p> <p>Jawaban:</p>
2.	<p>Bagaimana alur proses pencatatan pembayaran di Linda Cable?</p> <p>Jawaban:</p>
3.	<p>Bagaimana alur proses pendaftaran pelanggan baru di Linda Cable?</p> <p>Jawaban:</p>
4.	<p>Bagaimana alur proses pembaruan data pelanggan di Linda Cable?</p> <p>Jawaban:</p>
5.	<p>Apa saja yang diperlukan oleh sistem dari sistem informasi pembayaran Linda Cable?</p> <p>Jawaban:</p>

LAMPIRAN B HASIL WAWANCARA

B.1 Hasil Wawancara Latar Belakang Permasalahan

Nama Narasumber	: Heri Prasetyo
Jabatan Narasumber	: Pemilik Linda Cable
Tanggal	: 26-10-2020
Tempat	: Daring

No.	Pertanyaan
1.	<p>Bagaimana alur proses penagihan yang berjalan di Linda Cable?</p> <p>Jawaban:</p> <p>Penagihan dilakukan setiap bulan dibagi atas 3 periode penarikan dari rumah ke rumah oleh petugas penarik tagihan. Jika pelanggan sudah menunggak 3 bulan dan tidak membayar, maka pelanggan akan dicatat untuk diputus. Penagihan hendak diperbarui proses bisnisnya dengan memberikan pilihan kepada pelanggan untuk membayar di kantor atau meminta jasa penagihan ke rumah-rumah.</p>
2.	<p>Bagaimana alur proses pencatatan pembayaran di Linda Cable?</p> <p>Jawaban:</p> <p>Pencatatan dilakukan setiap periode ketika petugas penagihan telah menagihkan ke semua pelanggan yang ada di periode tersebut. Administrator menyalin semua pembayaran yang sudah di rekap oleh petugas penagihan.</p>
3.	<p>Bagaimana alur proses pendaftaran pelanggan baru di Linda Cable?</p> <p>Jawaban:</p> <p>Calon pelanggan perlu menghubungi kantor Linda Cable. Calon pelanggan akan diberikan formulir untuk data pelanggan dan persetujuan kontrak pelayanan. Setelah formulir diisi, data pelanggan dicatat untuk dipasang jaringan TV kabel dan datanya diberikan kepada petugas yang memiliki cakupan wilayah pelanggan. Calon pelanggan yang sudah pernah menjadi pelanggan tidak perlu mengisi formulir, tetapi harus menyelesaikan tunggakannya. Tunggakan yang diselesaikan adalah tunggakannya sebelum pelanggan diputus sebelumnya.</p>
4.	<p>Bagaimana alur proses pembuatan laporan bulanan di Linda Cable?</p> <p>Jawaban:</p> <p>Laporan bulanan dilakukan dengan menghitung jumlah pembayaran bulan ini dan penghitungan gaji petugas penagihan, penghitungan dilakukan oleh administrator.</p>

5. Apa saja yang diperlukan oleh sistem dari sistem informasi pembayaran Linda Cable?

Jawaban:

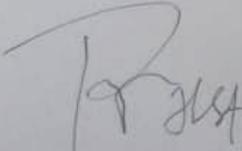
- Login
- Pendaftaran pelanggan
- Perubahan data pelanggan
- Pembayaran tagihan
- Laporan bulanan

Prasetyo
Heri Prasetyo

B.2 Hasil Wawancara Kebutuhan

Nama Narasumber	: Heri Prasetyo
Jabatan Narasumber	: Pemilik Linda Cable
Tanggal	: 21-8-2020
Tempat	: Daring

No.	Pertanyaan
1.	<p>Apakah Linda Cable sudah menerapkan sistem informasi untuk mengelola pembayaran?</p> <p>Jawaban:</p> <p>Belum, semua kegiatan dilakukan secara manual, semua data pembayaran dan laporan diarsipkan dalam bentuk fisik.</p>
2.	<p>Bagaimana kendala yang terjadi saat melakukan kegiatan pembayaran atau penagihan saat ini?</p> <p>Jawaban:</p> <p>Pencatatan pembayaran tagihan pelanggan dilakukan oleh pegawai administrasi kantor pada buku pembukuan. Tidak efisien karena setiap petugas penagihan tagihan menyerahkan daftar nama pelanggan yang membayar pada bulan itu, lalu pegawai administrasi menyalin ulang ke dalam catatan kantor.</p>
3.	<p>Seberapa sering kendala atau masalah pada pertanyaan poin 2 terjadi?</p> <p>Jawaban:</p> <p>Setidaknya setiap bulan akan ada masa dimana arsip menumpuk dan rawan untuk terjadi kesalahan, karena manual menggunakan tenaga orang.</p>
4.	<p>Apakah Linda Cable memiliki rencana kedepan mengenai pengelolaan pembayaran?</p> <p>Jawaban:</p> <p>Proses pembayaran sebelumnya hanya dipungut petugas penagihan, selanjutnya pelanggan akan diberikan opsi untuk membayar di kantor ataupun meminta jasa penagihan dengan biasa tambahan. Jadi perlu pembeda antara pelanggan satu dengan yang lain perlu ditagih oleh petugas siapa dan perlu pengelolaan transaksi yang lebih baik.</p>


Heri Prasetyo

LAMPIRAN C HASIL PENGUJIAN PENERIMAAN PENGGUNA

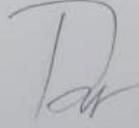
C.1 Jawaban Kuisiner UAT Administrator

KUISIONER PENERIMAAN PENGGUNA (Administrator)

Nama : Deni Rahmawati

Jabatan : Administrator

No.	Pertanyaan	Jawaban	
		Ya	Tidak
<i>Functional Correctness and Completeness</i>			
1	Apakah fitur penagihan membantu administrator untuk mengelola penagihan milik petugas penagihan?	✓	
2	Apakah fitur tagihan membantu administrator untuk mengelola pencatatan pembayaran di kantor?	✓	
3	Apakah fitur laporan membantu administrator untuk membuat laporan bulanan?	✓	
<i>Usability</i>			
4	Apakah sistem membantu administrator mengerjakan aktivitas pembayaran dan pengelolaan sehari-hari?	✓	
5	Apakah alur sistem informasi pembayaran sudah jelas saat menjalankan fungsional sistem?	✓	
6	Apakah keseluruhan fitur sistem informasi pembayaran mudah digunakan?	✓	
<i>Confidentiality and Availability</i>			
7	Apakah sistem informasi pembayaran telah memberikan hak akses sesuai dengan peran pengguna?	✓	
8	Apakah sistem informasi pembayaran menjamin keamanan data dari bukan pengguna?	✓	
Total		8	


Deni Rahmawati

C.2 Jawaban Kuisisioner UAT Petugas Penagihan

KUISISIONER PENERIMAAN PENGGUNA

(Petugas Penagihan)

Nama : Samsul Huda
Jabatan : Petugas Penagihan

No.	Pertanyaan	Jawaban	
		Ya	Tidak
<i>Functional Correctness and Completeness</i>			
1	Apakah fitur tagihan membantu petugas untuk mengelola pencatatan pembayaran?	✓	
<i>Usability</i>			
2	Apakah sistem membantu petugas penagihan mengerjakan aktivitas pembayaran sehari-hari?	✓	
3	Apakah alur sistem informasi pembayaran sudah jelas saat menjalankan fungsional sistem?	✓	
4	Apakah keseluruhan fitur sistem informasi pembayaran mudah digunakan?	✓	
<i>Confidentiality and Availability</i>			
5	Apakah sistem informasi pembayaran telah memberikan hak akses sesuai dengan peran pengguna?	✓	
6	Apakah sistem informasi pembayaran menjamin keamanan data dari bukan pengguna?	✓	
Total		6	


Samsul Huda

LAMPIRAN D SKENARIO USE CASE

D.1 Skenario Use Case Login

Scenario ID	UCS-SIP-01	Use Case ID	UC-SIP-01
Description	Use case login merupakan use case untuk mengotentikasi pengguna dengan cara masuk ke sistem dengan identitas khusus dengan hak akses tertentu.		
Actor	Petugas		
Goal	Aktor dapat masuk ke sistem		
Preconditions	Aktor telah membuka sistem		
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu login. 2. Sistem menampilkan layar login. 3. Aktor memasukkan username dan password. 4. Aktor menekan tombol login. 5. Sistem mengotentikasi masukan aktor. 6. Sistem menampilkan layar utama sistem. 		
Alternative Flow	<p>6a. Otentikasi gagal</p> <ol style="list-style-type: none"> 6a. 1. Sistem menampilkan pesan bahwa otentikasi gagal. 6a. 2. Kembali ke flow nomor (2). 		
Postconditions	Sistem mengotentikasi pengguna.		

D.2 Skenario Use Case Logout

Scenario ID	UCS-SIP-02	Use Case	UC-SIP-02
Description	Use case logout merupakan use case untuk membuat sistem menghapus otentikasi pengguna yang telah diotentikasi.		
Actor	Petugas		
Goal	Aktor dapat keluar dari sistem		
Preconditions	Aktor telah masuk ke dalam sistem		
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol logout 2. Sistem menampilkan konfirmasi 3. Aktor menekan tombol konfirmasi 4. Sistem menampilkan layar login 		
Alternative Flow			
Postconditions	Sistem menghapus otentikasi pengguna.		

D.3 Skenario Use Case Mencari Data Pelanggan

Scenario ID	UCS-SIP-03	Use Case	UC-SIP-03
Description	Use case mencari data pelanggan merupakan use case untuk aktor mencari data pelanggan sesuai dengan nomor pelanggan.		
Actor	Petugas		
Goal	Aktor dapat melihat data pelanggan yang dicari.		
Preconditions	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Aktor telah membuka menu data pelanggan. 		
Main Flow	<ol style="list-style-type: none"> 1. Aktor membuka menu pencarian pelanggan. 2. Sistem menampilkan layar pencarian pelanggan. 3. Aktor memasukkan nomor pelanggan atau identitas pelanggan. 4. Sistem mencari data pelanggan terkait. 5. Sistem menampilkan daftar pelanggan sesuai masukan aktor. 		
Alternative Flow	<p>5a. Pelanggan tidak ditemukan</p> <p>5a. 1. Menampilkan pesan tidak ada data yang cocok.</p> <p>5a. 2. Kembali ke flow nomor (2)</p>		
Postconditions	Daftar pelanggan yang sesuai dengan pencarian ditampilkan.		

D.4 Skenario Use Case Melihat Data Pelanggan

Scenario ID	UCS-SIP-04	Use Case	UC-SIP-04
Use Case	Melihat Data Pelanggan (UC-SIP-02)		
Description	Use case melihat data pelanggan merupakan use case untuk aktor melihat data pelanggan.		
Actor	Petugas		
Goal	Aktor dapat melihat data pelanggan.		
Preconditions	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Aktor telah membuka layar utama 		
Main Flow	<ol style="list-style-type: none"> 1. Aktor membuka menu data pelanggan. 2. Sistem menampilkan daftar pelanggan. 3. Aktor memilih pelanggan. 4. Aktor membuka data pelanggan. 5. Sistem menampilkan data pelanggan. 		
Alternative Flow	-		
Postconditions	Layar data pelanggan telah ditampilkan.		

D.5 Skenario Use Case Menambah Data Pelanggan

Scenario ID	UCS-SIP-05	Use Case	UC-SIP-05
Description	Use case menambah data pelanggan merupakan kegiatan pengelolaan data pelanggan untuk data pelanggan baru yang belum pernah terdaftar.		
Actor	Administrator		
Goal	Aktor dapat menambahkan data pelanggan.		
Preconditions	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Aktor telah membuka menu data pelanggan. 		
Main Flow	<ol style="list-style-type: none"> 1. Aktor membuka menu tambah pelanggan 2. Sistem menampilkan formulir masukan data pelanggan. 3. Aktor memasukkan data pelanggan. 4. Sistem menyimpan data pelanggan. 5. Sistem menampilkan pesan data pelanggan telah disimpan. 		
Alternative Flow	-		
Postconditions	Data pelanggan berhasil disimpan.		

D.6 Skenario Use Case Memperbarui Data Pelanggan

Scenario ID	UCS-SIP-06	Use Case	UC-SIP-06
Description	Use case memperbarui data pelanggan merupakan use case terkait pengelolaan data pelanggan untuk memperbarui data pelanggan yang sudah ada.		
Actor	Administrator		
Goal	Aktor dapat memperbarui data pelanggan.		
Preconditions	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Aktor telah membuka menu data pelanggan. - Data pelanggan yang dituju sudah tersimpan pada sistem 		
Main Flow	<ol style="list-style-type: none"> 1. Aktor membuka data pelanggan yang dituju. 2. Sistem menampilkan data pelanggan. 3. Aktor membuka menu pembaruan data pelanggan. 4. Sistem menampilkan formulir dengan data awal pelanggan terkait. 5. Aktor mengubah data pelanggan 6. Sistem menampilkan pesan konfirmasi pembaruan. 7. Aktor menekan tombol konfirmasi pembaruan. 8. Sistem menyimpan data pelanggan. 9. Sistem menampilkan pesan data pelanggan telah diperbarui. 		
Alternative Flow	7a. Aktor tidak mengkonfirmasi pembaruan.		

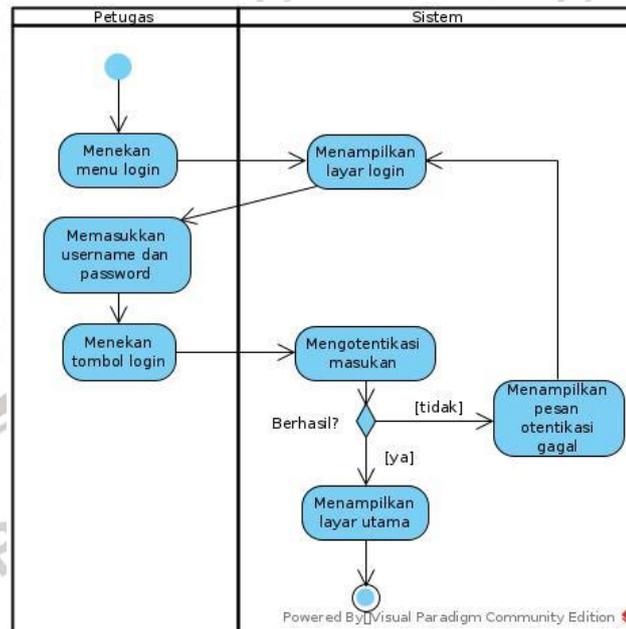
	7a. 1. Aktor menekan tombol batal konfirmasi 7a. 2. Kembali ke <i>flow</i> nomor (4)
Postconditions	Data pelanggan telah diperbarui.

D.7 Skenario *Use Case* Menghapus Data Pelanggan

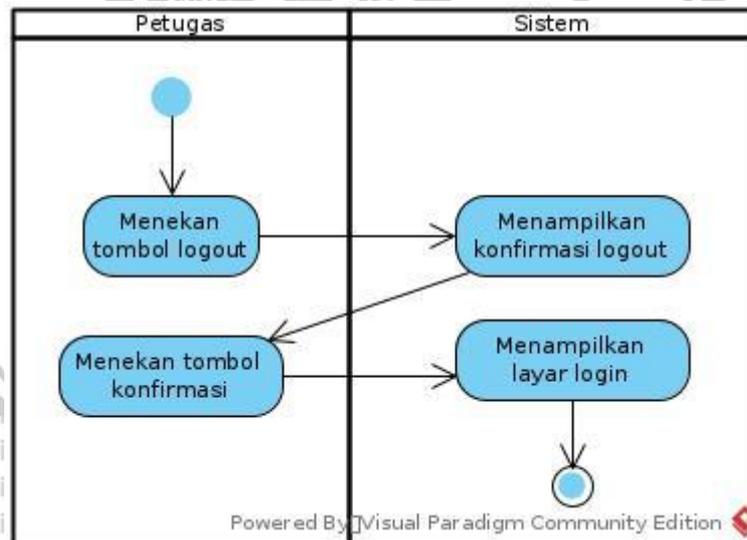
Scenario ID	UCS-SIP-07	Use Case	UC-SIP-07
Description	<i>Use case</i> menghapus data pelanggan merupakan <i>use case</i> terkait pengelolaan data pelanggan menghapus data pelanggan yang sudah tidak diperlukan lagi disimpan pada sistem.		
Actor	Administrator		
Goal	Aktor dapat memperbaiki data pelanggan.		
Preconditions	<ul style="list-style-type: none"> - Aktor telah masuk ke sistem. - Aktor telah membuka menu data pelanggan. - Data pelanggan yang dituju sudah tersimpan pada sistem 		
Main Flow	<ol style="list-style-type: none"> 1. Aktor membuka data pelanggan yang dituju. 2. Sistem menampilkan data pelanggan. 3. Aktor menekan tombol hapus data pelanggan. 4.. Sistem menampilkan pesan konfirmasi penghapusan. 5. Aktor menekan tombol konfirmasi penghapusan. 6. Sistem menyimpan data pelanggan. 7. Sistem menampilkan pesan data pelanggan telah diperbarui. 		
Alternative Flow	5a. Aktor tidak mengkonfirmasi penghapusan. 5a. 1. Aktor menekan tombol batal konfirmasi 5a. 2. Kembali ke <i>flow</i> nomor (2)		
Postconditions	Data pelanggan dihapus dari sistem.		

LAMPIRAN E ACTIVITY DIAGRAM

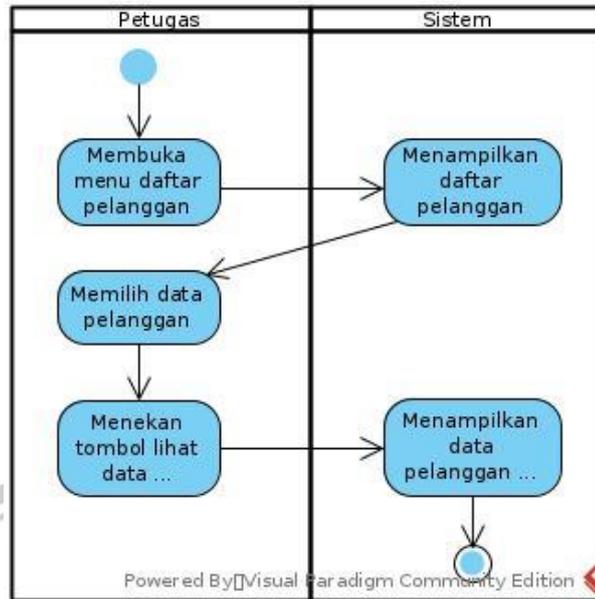
E.1 Activity Diagram Login



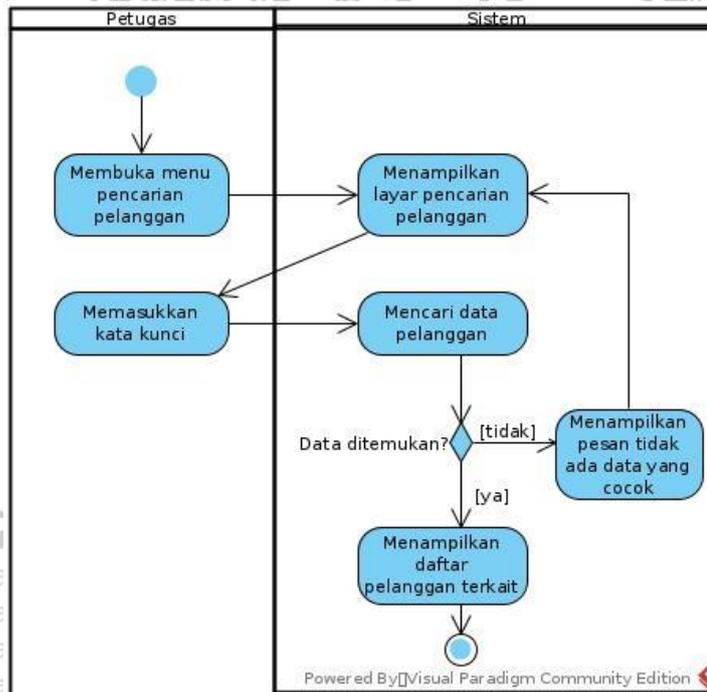
E.2 Activity Diagram Logout



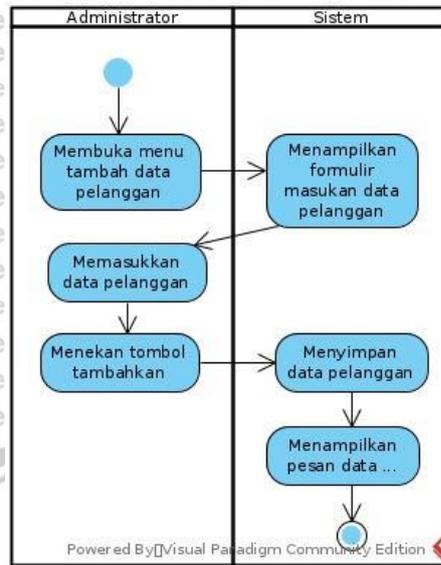
E.3 Activity Diagram Melihat Data Pelanggan



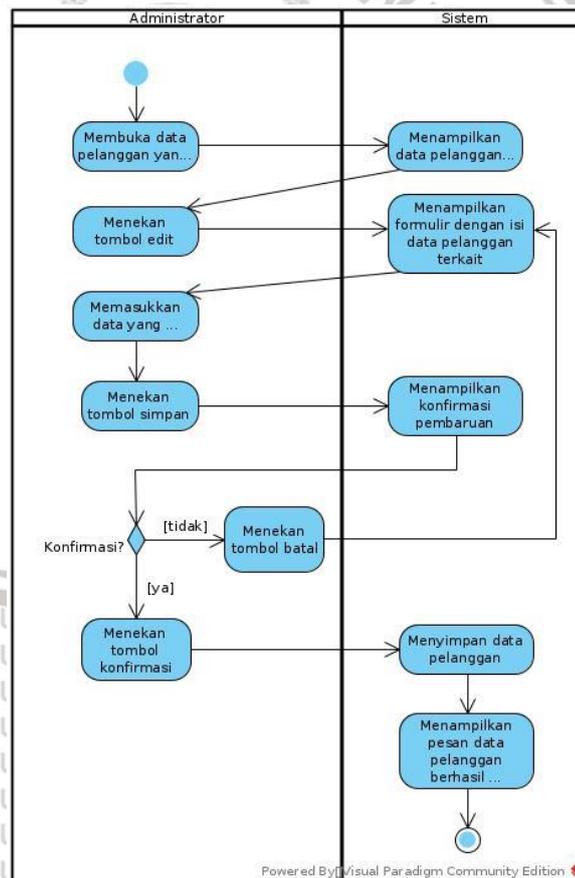
E.4 Activity Diagram Mencari Data Pelanggan



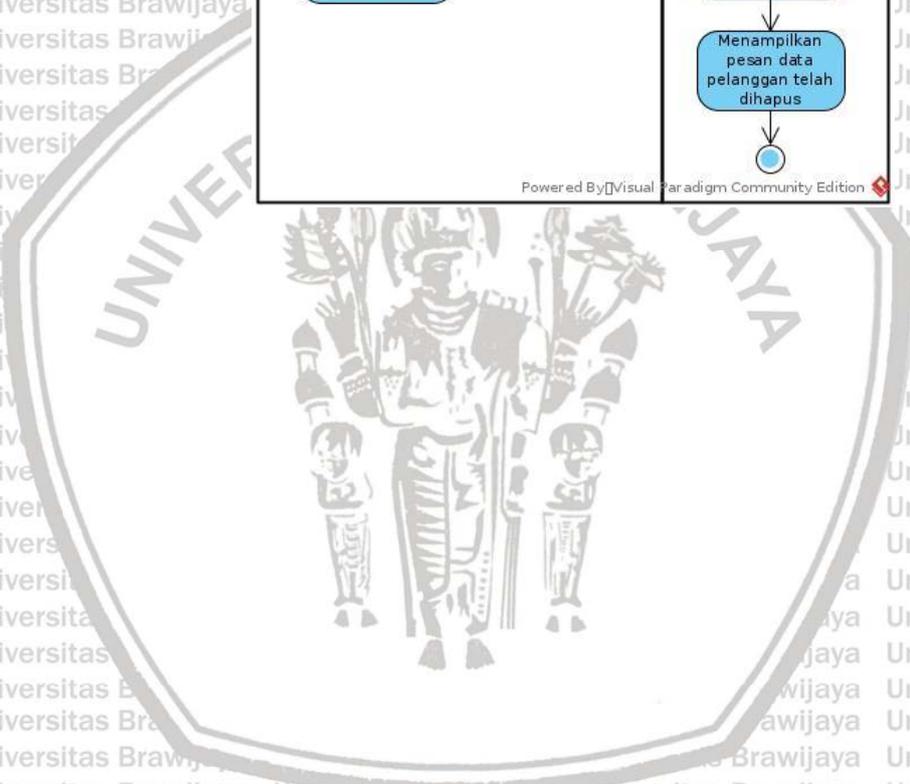
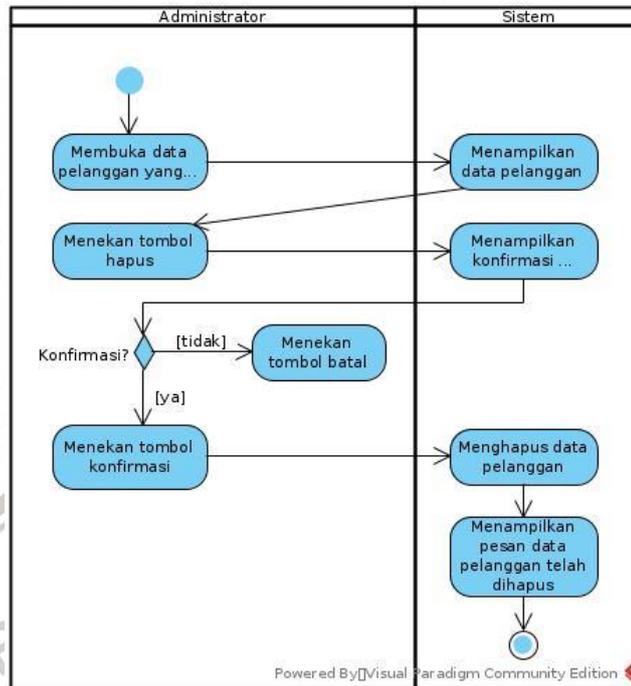
E.5 Activity Diagram Menambah Data Pelanggan



E.6 Activity Diagram Memperbarui Data Pelanggan

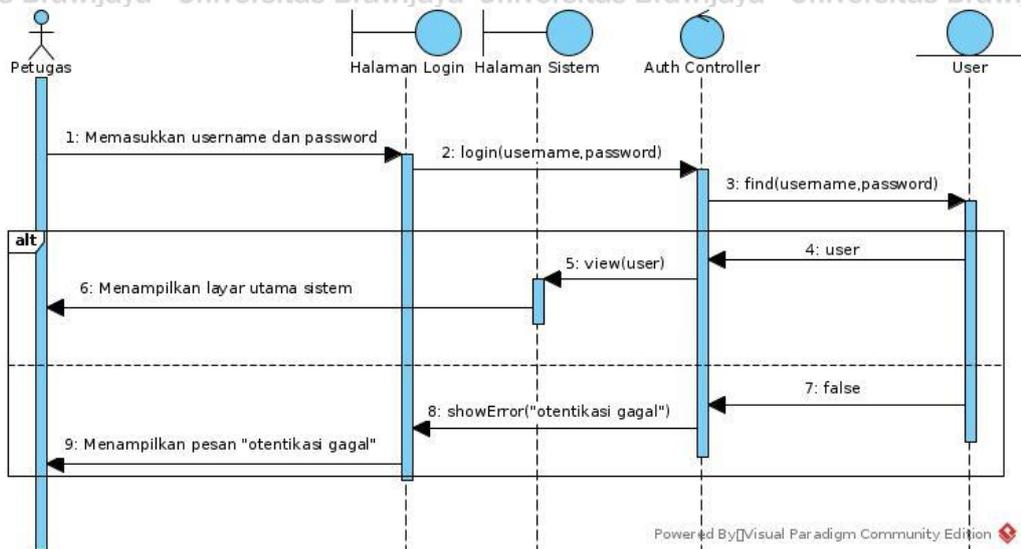


E.7 Activity Diagram Menghapus Data Pelanggan

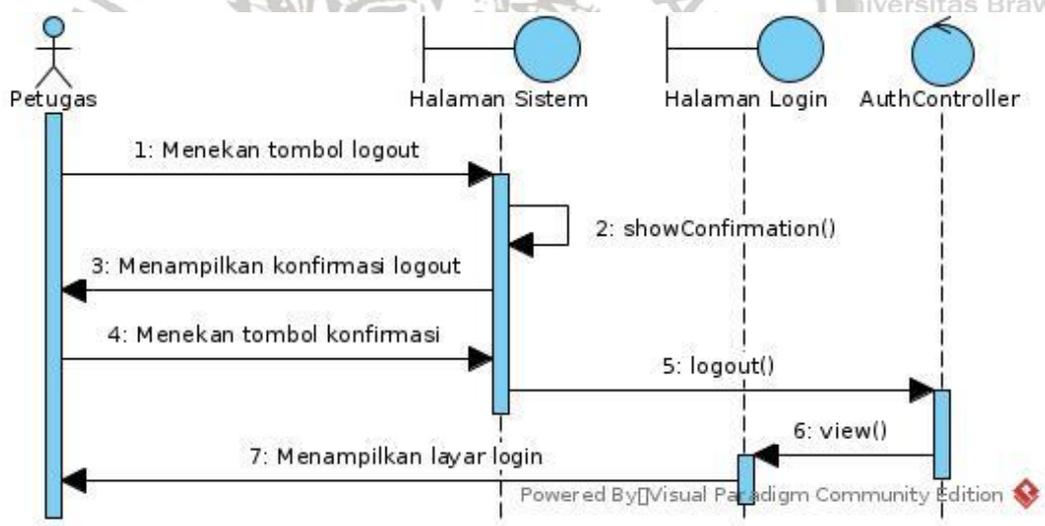


LAMPIRAN F SEQUENCE DIAGRAM

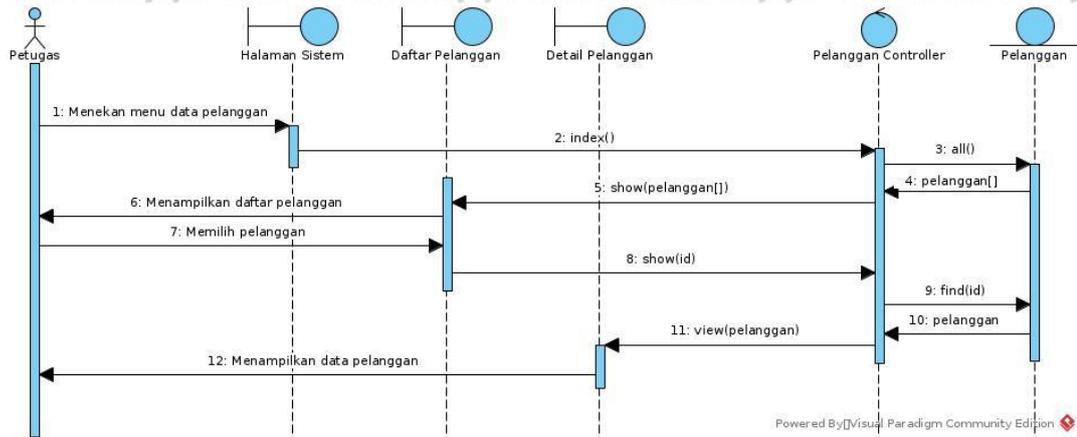
F.1 Sequence Diagram Login



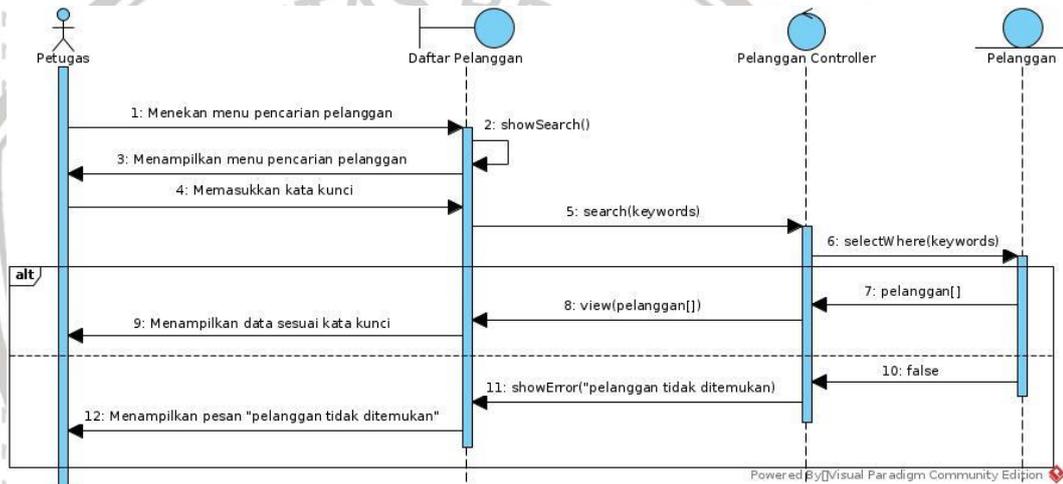
F.2 Sequence Diagram Logout



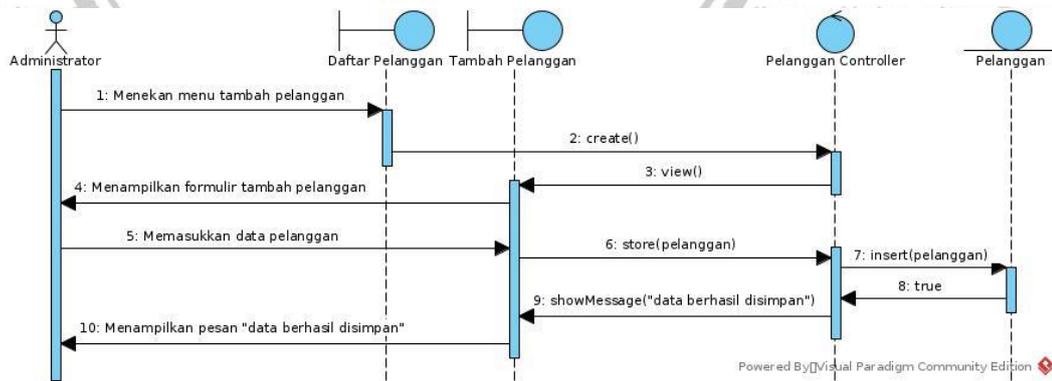
F.3 Sequence Diagram Melihat Data Pelanggan



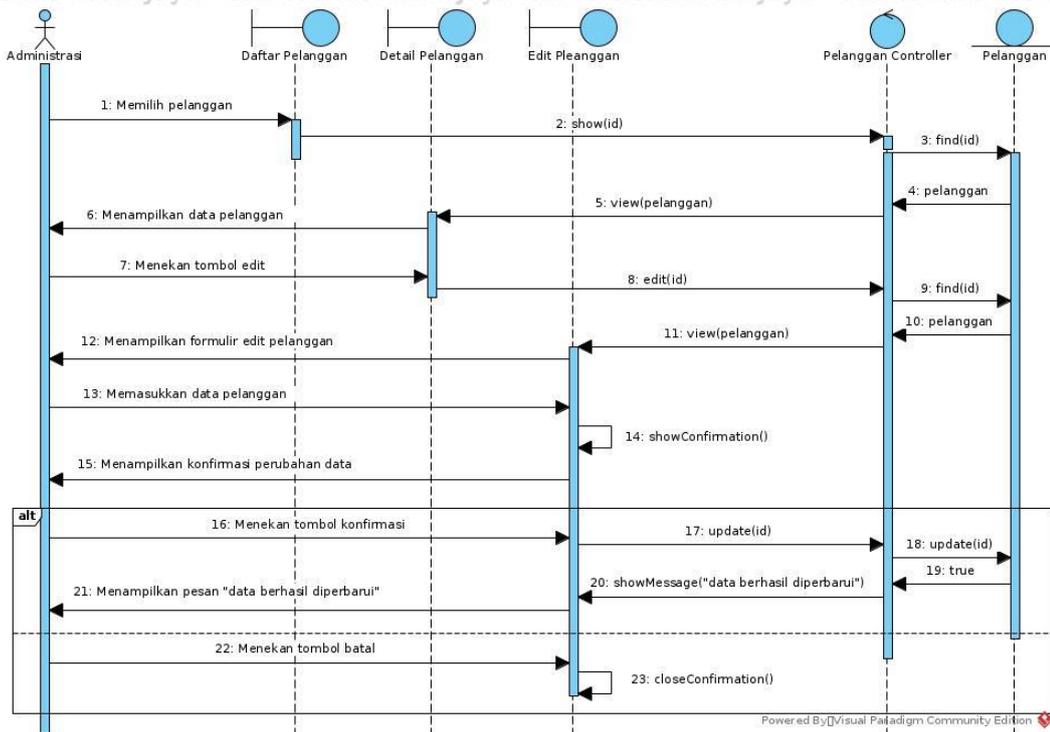
F.4 Sequence Diagram Mencari Data Pelanggan



F.5 Sequence Diagram Menambah Data Pelanggan



F.6 Sequence Diagram Memperbarui Data Pelanggan



F.7 Sequence Diagram Menghapus Data Pelanggan

