

**SISTEM PENGELOLAAN DATA PELANGGAN RT/RW NET
BERBASIS *WEB* MENGGUNAKAN METODE
*MODEL VIEW CONTROLLER (MVC)***

SKRIPSI

Diajukan sebagai salah satu syarat untuk kelulusan

Jenjang Strata Satu (S1)

Pada Program Studi Teknik Informatika

Disusun Oleh:

Bayu Ramadhan

361701013



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI**

BANDUNG

2022

LEMBAR PERSETUJUAN

**SISTEM PENGELOLAAN DATA PELANGGAN RT/RW NET
BERBASIS *WEB* MENGGUNAKAN METODE
*MODEL VIEW CONTROLLER (MVC)***

***WEB-BASED RT/RW NET CUSTOMER DATA MANAGEMENT SYSTEM
USING METHOD
MODEL VIEW CONTROLLER (MVC)***

Oleh:

**Bayu Ramadhan
361701013**

Skripsi ini telah di terima dan disahkan
Untuk memenuhi persyaratan mencapai gelar

SARJANA TEKNIK INFORMATIKA

Pada

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI

Bandung, 31 Januari 2022

Disahkan Oleh

**Ketua Program Studi Teknik
Informatika,**

Dosen Pembimbing,

Chalifa Chazar, S.T., M.T.

NIDN: 0421098704

Dr. Chairuddin, Ir., M.M., M. T

NIDN: 0426076901

LEMBAR PERSETUJUAN REVISI TUGAS AKHIR

SISTEM PENGELOLAAN DATA PELANGGAN RT/RW NET BERBASIS *WEB* MENGGUNAKAN METODE *MODEL VIEW CONTROLLER (MVC)*

WEB-BASED RT/RW NET CUSTOMER DATA MANAGEMENT SYSTEM USING METHOD MODEL VIEW CONTROLLER (MVC)

Telah melakukan sidang Skripsi pada hari Rabu, 02 Februari 2022 dan telah melakukan revisi sesuai dengan masukan pada saat sidang Tugas Akhir

Menyetujui,

No	Nama	Keterangan	Tanda Tangan
1	Dr. Chairuddin, Ir., M.M., M. T .,	Pembimbing	
2	Moch. Ali Ramdhani, S.T., M.Kom.	Penguji 1	
3	Chalifa Chazar, S.T., M.T.	Penguji 2	

Bandung, 07 April 2022

Mengetahui,
Ketua program studi

Chalifa Chazar, S.T., M.T.
NIDN. 0421098704

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah Asli dan belum pernah diajukan untuk mendapatkan gelar Akademik, baik di Sekolah Tinggi Manajemen & Komputer Indonesia Mandiri maupun Perguruan Tinggi lainnya.
2. Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
3. Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, 31 Januari 2022
Yang Membuat Pernyataan

Bayu Ramadhan

361701013

SISTEM PENGELOLAAN DATA PELANGGAN RT/RW NET BERBASIS *WEB* MENGGUNAKAN METODE *MODEL VIEW CONTROLLER (MVC)*

ABSTRAK

RT/RW Net adalah konsep di mana beberapa komputer atau gadget dalam suatu perumahan atau blok dapat terhubung ke jaringan internet sehingga dapat saling terhubung dan dapat berbagi data serta informasi. Konsep lain dari RT/RW Net adalah memberdayakan pemakaian internet di mana fasilitas internet tersedia selama 24 jam sehari selama sebulan di mana biaya yang akan dikeluarkan akan murah karena semua biaya pembangunan infrastruktur, operasional dan biaya ISP (*Internet Service Provider*) akan ditanggung bersama. Konsep RT-RW-Net bisa dibilang sama dengan konsep warnet (warung internet), pemilik warnet akan berlangganan internet atau *bandwidth* dari penyedia internet / ISP (*Internet Service Provider*) seperti Telkom, Indosat atau XL, lalu dijual atau disewakan kembali ke pelanggan yang datang menyewa komputer untuk bermain internet baik untuk membuka *Email*, *Chating*, *Browsing*, bermain *Game* dll. RT/RW Net termasuk sangat murah jika dibandingkan dengan warnet karena penyedia layanan RT/RW Net hanya menjual atau menyewakan layanan internetnya saja, tidak beserta dengan perangkatnya seperti konsep warnet. Pada saat ini para usahawan RT/RW Net mulai menggunakan teknologi *website* untuk membantu mengelola data pelanggan, mengunggah produk atau paket layanan internet yang akan tawarkan dan mengatur transaksi pembayaran pelanggan supaya memudahkan pelanggannya untuk melakukan pembayaran tagihan layanan RT/RW Net. Penulis bertujuan untuk membuat aplikasi berbasis *web* yang dapat digunakan oleh para pelaku bisnis RT/RW Net untuk membantu mengelola data pelanggan, melakukan *posting* iklan untuk produk atau paket yang akan dijual dan melakukan transaksi pembayaran dengan pelanggan melalui *payment gateway*.

Kata Kunci: Sistem pengelolaan data pelanggan, RT-RW Net, MVC, *Payment Gateway*.

WEB-BASED RT/RW NET CUSTOMER DATA MANAGEMENT SYSTEM USING METHOD MODEL VIEW CONTROLLER (MVC)

ABSTRACT

RT/RW Net is a concept where several computers or gadgets in a housing or block can be connected to the internet network so that they can be connected to each other and can share data and information. Another concept of RT/RW Net is to empower internet usage where internet facilities are available 24 hours a day for a month where the costs to be incurred will be cheap because all infrastructure development costs, operations and ISP (Internet Service Provider) costs will be shared. The concept of RT-RW-Net is practically the same as the concept of an internet cafe (internet cafe), the cafe owner will subscribe to the internet or bandwidth from an internet provider / ISP (Internet Service Provider) such as Telkom, Indosat or XL, then sell or rent it back to customers who come. rent a computer to play the internet either to open email, chat, browse, play games etc. RT/RW Net is considered very cheap when compared to internet cafes because RT/RW Net service providers only sell or rent internet services, not including the equipment such as the concept of an internet cafe. At this time RT/RW Net entrepreneurs are starting to use website technology to help manage customer data, upload products or internet service packages that will be offered and manage customer payment transactions to make it easier for customers to make RT/RW Net service bill payments. The author aims to create a web-based application that can be used by RT/RW Net business people to help manage customer data, post advertisements for products or packages to be sold and make payment transactions with customers through payment gateways.

Keywords: *Customer data management system, RT-RW Net, MVC, Payment Gateway.*

UCAPAN TERIMAKASIH

Dengan mengucapkan syukur Alhamdulillah, penelitian ini dapat diselesaikan untuk memenuhi syarat tugas akhir. Tugas akhir Skripsi ini dapat tersusun berkat bantuan, bimbingan dan saran-saran serta masukan dari berbagai pihak sampai selesainya laporan ini. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT dengan segala rahmat dan karunia-Nya yang memberikan kekuatan dan kelancaran bagi peneliti dalam menyelesaikan skripsi ini.
2. Kedua Orang Tua yang sangat saya cintai yang telah memberikan doa, bantuan, dukungan, kasih sayang, pengorbanan dan semangat kepada Penulis dalam menyelesaikan tugas akhir skripsi.
3. Bapak Dr. Chairuddin, Ir., M.M., M. T ., selaku ketua Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri sekaligus juga sebagai Dosen Pembimbing. Terimakasih telah meluangkan waktu, memberikan bimbingan, arahan, dan saran kepada penulis dalam menyelesaikan skripsi ini.
4. Ibu Chalifa Chazar, S,T., M.M., M.T., selaku Ketua Program Jurusan Teknik Informatika.
5. Seluruh dosen dan staf Sekolah Tinggi Indonesia Mandiri Bandung yang telah memberikan ilmu dan fasilitas pada penulis hingga dapat menyelesaikan skripsi ini.
6. Seluruh teman - teman penulis yang sama - sama berjuang dalam menyusun skripsi, memberikan semangat dan saling membantu satu sama lain.

7. Semua pihak yang tidak dapat disebutkan satu persatu, terima kasih atas segalanya.

Penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam penyelesaian laporan tugas akhir ini. Semoga laporan tugas akhir ini dapat bermanfaat dan berguna bagi semua pihak.

KATA PENGANTAR

Bismillahirrahmaanirrahim, Puji dan Syukur penulis panjatkan kepada Allah SWT karena atas taufiq dan hidayah-Nya, penulis dapat menyelesaikan skripsi yang berjudul “**SISTEM PENGELOLAAN DATA PELANGGAN RT/RW NET BERBASIS WEB MENGGUNAKAN METODE *MODEL VIEW CONTROLLER (MVC)***”. Skripsi ini disusun sebagai syarat untuk menyelesaikan jenjang pendidikan Strata Satu (S1) di Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri. Penulis menyadari dalam penulisan Laporan Skripsi ini, baik bentuk maupun isinya masih terdapat banyak kekurangan karena keterbatasan pengetahuan dan kemampuan serta pengalaman yang penulis miliki. Untuk itu dengan kerendahan hati penulis mengharapkan kritik dan saran yang membangun untuk kemajuan penulis dikemudian hari. Penulis berharap Skripsi ini bisa menjadi salah satu sumber yang dapat bermanfaat.

Bandung, 07 April 2022
Penulis

Bayu Ramadhan
361701013

DAFTAR ISI

LEMBAR PERSETUJUAN.....	I
LEMBAR PERSETUJUAN REVISI TUGAS AKHIR.....	II
SURAT PERNYATAAN.....	III
ABSTRAK.....	IV
ABSTRACT.....	V
UCAPAN TERIMAKASIH.....	VI
KATA PENGANTAR.....	VIII
DAFTAR ISI.....	IX
DAFTAR TABEL.....	XI
DAFTAR GAMBAR.....	XII
BAB I.....	1
PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 IDENTIFIKASI MASALAH.....	3
1.3 TUJUAN PENELITIAN.....	4
1.4 BATASAN MASALAH.....	4
1.5 METODE PENELITIAN.....	6
1.6 SISTEMATIKA PENULISAN.....	10
BAB II.....	12
LANDASAN TEORI.....	12
2.1 RT/RW-NET.....	12
2.2 GHADHENET.....	12
2.3 KONSEP DASAR SISTEM.....	13
2.4 KONSEP DASAR DATA.....	14

2.5	<i>WEBSITE</i>	15
2.6	LARAVEL	16
2.7	MYSQL	16
2.8	BOOTSTRAP	17
2.9	<i>PAYMENT GATEWAY</i>	18
2.10	MIDTRANS	19
2.11	<i>UNIFIED MODELING LANGUAGE (UML)</i>	19
2.12	BASIS DATA (<i>DATA BASE</i>)	30
2.13	<i>ENTITY RELATIONAL DIAGRAM (ERD)</i>	32
2.14	METODE <i>AGILE</i>	33
2.15	PENGUJIAN WEB	36
BAB III		38
PEMBAHASAN		38
3.1	PENGUMPULAN DATA	39
3.2	REKAYASA PERANGKAT LUNAK (<i>AGILE</i>)	40
BAB IV		79
IMPLEMENTASI DAN UJI COBA		79
4.1	PENGEMBANGAN (<i>DEVELOPMENT</i>)	79
4.2	PENGUJIAN (<i>TESTING</i>)	90
4.3	PENYEBARAN (<i>DEPLOYMENT</i>)	94
4.4	ULASAN (<i>REVIEW</i>)	95
BAB V		96
PENUTUP		96
5.1.	KESIMPULAN	96
5.2.	SARAN	96
DAFTAR PUSTAKA		98
LAMPIRAN		102

DAFTAR TABEL

Tabel 2. 1 Simbol <i>Use Case</i> Diagram (Rosa dan Shalahuddin, 2016:155).....	20
Tabel 2. 2 Simbol Activity Diagram (Rosa dan Shalahuddin, 2016:162).....	24
Tabel 2. 3 Simbol <i>Class</i> Diagram (Rosa dan Shalahuddin, 2016:146).....	26
Tabel 2. 4 Simbol Sequence Diagram (Rosa dan Shalahuddin, 2018:165).....	29
Tabel 2. 5 Simbol Entity Relationship Diagram (Rosa dan Shalahuddin)	33
Tabel 3. 1 Fitur dan Halaman <i>Back-End</i> Sistem.....	41
Tabel 3. 2 <i>Use Case Scenario Management</i> Paket	46
Tabel 3. 3 <i>Use Case Scenario Management Request Order</i>	47
Tabel 3. 4 <i>Use Case Scenario Management Customer</i>	48
Tabel 3. 5 <i>Use Case Scenario Management Transactions</i>	49
Tabel 3. 6 <i>Use Case Scenario</i> Pilih Paket	50
Tabel 3. 7 <i>Use Case Scenario</i> Cek Tagihan	50
Tabel 3. 8 <i>Use Case Scenario</i> Lihat Tentang Kami	51
Tabel 4. 1 Tabel <i>Order</i>	80
Tabel 4. 2 Tabel <i>Product</i>	81
Tabel 4. 3 Tabel <i>Transactions</i>	81
Tabel 4. 4 Hasil Pengujian <i>Back-End (Admin)</i>	90
Tabel 4. 8 Hasil Pengujian <i>Front-End</i>	93

DAFTAR GAMBAR

Gambar 2. 1 Pemrosesan Data (Fitri dan Nia, 2018).....	14
Gambar 2. 2 <i>Use Case Diagram</i> (Mubarak, 2019)	23
Gambar 2. 3 <i>Activity Diagram</i> (Mubarak, 2019)	25
Gambar 2. 4 <i>Class Diagram</i> (Mubarak, 2019).....	28
Gambar 2. 5 Contoh <i>Sequence Diagram</i> (Tri Mardiani, G. 2017).....	30
Gambar 2. 6 Metode <i>Agile</i> (Haryana, 2019).....	35
Gambar 3. 1 Desain Penelitian	38
Gambar 3. 2 <i>Use Case Diagram Admin</i>	44
Gambar 3. 3 <i>Use Case Diagram Customer</i>	45
Gambar 3. 4 <i>Activity Diagram Login (Admin)</i>	52
Gambar 3. 5 <i>Activity Diagram Management Paket (Admin)</i>	53
Gambar 3. 6 <i>Activity Diagram Management Request Order (Admin)</i>	54
Gambar 3. 7 <i>Activity Diagram Management Customer (Admin)</i>	55
Gambar 3. 8 <i>Activity Diagram Management Transaksi (Admin)</i>	56
Gambar 3. 9 <i>Activity Diagram Pendaftaran (Customer)</i>	57
Gambar 3. 10 <i>Activity Diagram Cek Tagihan (Customer)</i>	58
Gambar 3. 11 <i>Activity Diagram Lihat Tentang Kami</i>	59
Gambar 3. 12 <i>Class Diagram Sistem</i>	59
Gambar 3. 13 <i>Sequence Diagram Detail Order (Admin)</i>	59
Gambar 3. 14 <i>Sequence Diagram Login (Admin)</i>	60
Gambar 3. 15 <i>Sequence Diagram Tambah Paket (Admin)</i>	60
Gambar 3. 16 <i>Sequence Diagram Ubah Paket (Admin)</i>	61
Gambar 3. 17 <i>Sequence Diagram Hapus Paket (Admin)</i>	61
Gambar 3. 18 <i>Sequence Diagram Detail Order (Admin)</i>	62
Gambar 3. 19 <i>Sequence Diagram Approve Order (Admin)</i>	62
Gambar 3. 20 <i>Sequence Diagram Delete Order (Admin)</i>	63
Gambar 3. 21 <i>Sequence Diagram Detail Customer (Admin)</i>	63
Gambar 3. 22 <i>Sequence Diagram Edit Data Customer (Admin)</i>	64

Gambar 3. 23 <i>Sequence</i> Diagram Hapus <i>Customer</i> (<i>Admin</i>).....	64
Gambar 3. 24 <i>Sequence</i> Diagram Ubah Tanggal Pembayaran <i>Customer</i> (<i>Admin</i>).....	65
Gambar 3. 25 <i>Sequence</i> Diagram Ubah Status Pembayaran <i>Customer</i> (<i>Admin</i>).....	65
Gambar 3. 26 <i>Sequence</i> Diagram <i>Visit URL Payment</i> (<i>Admin</i>)	66
Gambar 3. 27 <i>Sequence</i> Diagram Buat <i>URL Payment</i> (<i>Admin</i>).....	66
Gambar 3. 28 <i>Sequence</i> Diagram Hapus <i>URL Payment</i> (<i>Admin</i>)	67
Gambar 3. 29 <i>Sequence</i> Diagram Pendaftaran Pelanggan Baru (<i>Customer</i>).....	68
Gambar 3. 30 <i>Sequence</i> Diagram Cek Tagihan (<i>Customer</i>).....	69
Gambar 3. 31 <i>Sequence</i> Diagram Tentang Kami (<i>Customer</i>)	69
Gambar 3. 32 <i>Sequence</i> Diagram Detail <i>Order</i> (<i>Admin</i>).....	69
Gambar 3. 33 Desain Antarmuka Awal (<i>Landing Page</i>).....	70
Gambar 3. 34 Desain Antarmuka Halaman Harga	71
Gambar 3. 35 Desain Antarmuka Halaman Pendaftaran Pelanggan Baru	71
Gambar 3. 36 Desain Antarmuka Halaman Pendaftaran Berhasil.....	72
Gambar 3. 37 Desain Antarmuka Halaman Cek Tagihan	72
Gambar 3. 38 Desain Antarmuka Halaman Histori Pembayaran	73
Gambar 3. 39 Desain Antarmuka Halaman Tentang Kami	73
Gambar 3. 40 Desain Antarmuka Halaman <i>Login</i> (<i>Admin</i>).....	74
Gambar 3. 41 Desain Antarmuka Halaman <i>Dashboard</i> (<i>Admin</i>).....	74
Gambar 3. 42 Desain Antarmuka Halaman <i>Management Paket</i> (<i>Admin</i>)	75
Gambar 3. 43 Desain Antarmuka Halaman <i>Request Order</i> (<i>Admin</i>)	75
Gambar 3. 44 Desain Antarmuka Halaman <i>Management Customer</i> (<i>Admin</i>)	76
Gambar 3. 45 Desain Antarmuka Halaman <i>Management Transaksi</i> (<i>Admin</i>)	76
Gambar 4. 1 <i>Entity Relationship</i> Diagram Sistem.....	80
Gambar 4. 2 <i>Physical Table</i> Sistem.....	82
Gambar 4. 3 Antarmuka Awal (<i>Landing Page</i>).....	82
Gambar 4. 4 Antarmuka Halaman Harga	83
Gambar 4. 5 Antarmuka Halaman Pendaftaran Pelanggan Baru.....	84
Gambar 4. 6 Antarmuka Halaman Pendaftaran Berhasil.....	85
Gambar 4. 7 Antarmuka Halaman Cek Tagihan.....	85
Gambar 4. 8 Antarmuka Halaman Histori Pembayaran	86

Gambar 4. 9 Antarmuka Halaman Tentang Kami	86
Gambar 4. 10 Antarmuka Halaman <i>Login (Admin)</i>	87
Gambar 4. 11 Antarmuka Halaman <i>Dashboard (Admin)</i>	87
Gambar 4. 12 Antarmuka Halaman <i>Management Paket (Admin)</i>	88
Gambar 4. 13 Antarmuka Halaman <i>Request Order (Admin)</i>	88
Gambar 4. 14 Antarmuka Halaman <i>Management Customer (Admin)</i>	89
Gambar 4. 15 Antarmuka Halaman <i>Management Transaksi (Admin)</i>	89

BAB I

PENDAHULUAN

1.1 Latar Belakang

Informasi dan teknologi merupakan dua hal yang tidak bisa untuk dipisahkan, hal ini pula yang dibutuhkan oleh para pelaku usaha beserta dengan konsumennya. Dengan adanya informasi yang tersampaikan, pelaku usaha akan memiliki acuan untuk menentukan langkah dan keputusan, dengan informasi konsumen akan mempunyai suatu kepastian pemahaman, serta kemantapan untuk memberikan suatu *trust* (Wicaksono, B. D., & Anggraeni, S. (2021)). Hal ini berlaku juga pada pelaku bisnis layanan internet, di mana informasi akan sebuah produk internet harus tersampaikan kepada konsumen dengan mudah, cepat, dan efisien, sehingga konsumen akan merasa nyaman sehingga secara tidak langsung *building trust* akan terbentuk di hati konsumen. Untuk itu penggunaan sebuah teknologi sangat dibutuhkan untuk memenuhi kebutuhan yang mudah, cepat dan efisien. Bisnis layanan internet dalam perkembangannya, memerlukan layanan purna jual yang optimal serta mudah untuk diakses oleh konsumen. Pada hal ini tentu akan menimbulkan kepuasan dan kenyamanan tersendiri bagi konsumen dalam memiliki properti tersebut. Di samping dapat memberikan kepuasan pada konsumen, tentu akan memberikan keuntungan bagi pengembang (*developer*) untuk pelayanan jasa tersebut. Kecepatan dan kemudahan konsumen dalam mendapatkan jasa atau pelayanan yang diperlukan, tentu akan meningkatkan kepuasan konsumen serta memudahkan konsumen dalam mengelola dan menggunakan produk properti, di samping akan memberi keuntungan tersendiri kepada pengembang (*developer*).

Harga layanan internet semakin mahal apalagi di saat pandemi ini bagaimana mencari sumber daya yang murah dan baik untuk menghadapi era teknologi informasi yang pesat. Oleh karena itu kebutuhan akan internet dan solusi untuk mendapatkan konektivitas internet yang murah sangat diinginkan oleh masyarakat.

Saat ini konsumen cukup kesulitan dalam melihat perkembangan dan transparansi akan unit yang dibeli. Konsumen tidak dapat melihat riwayat pembayaran yang telah dilakukan, serta *progres* atau perkembangan sejauh mana pembangunan unit dibeli itu dibangun, di samping pembayaran yang kurang fleksibel serta informasi kapan pembayaran atau disebut pengingat (*reminder*) tentang pembayaran, harus dilakukan yang belum tersampaikan dengan tepat dan mudah, dan layanan konsumen yang kurang memadai serta belum adanya fasilitas layanan purna jual (aplikasi) untuk konsumen yang mudah untuk diakses atau digunakan. Kemudian dari sisi internal (pengembang) akan memudahkan dalam penginputan data, penagihan, dan pembuatan laporan. Untuk mewujudkan kecepatan, ketepatan, serta kemudahan bagi konsumen, diperlukan suatu alat atau aplikasi sebagai media yang dapat di kapan dan di mana saja (Al-gawwam dan Benaissa, 2018).

Untuk memudahkan dalam pembayaran tagihan, serta mudah dalam melakukan penagihan, penggunaan sistem tagihan *online* tentu sangat memudahkan serta dapat menghemat waktu serta memiliki keakuratan data. Dengan menggunakan pembayaran *online*, setiap terjadi pembayaran maka data akan otomatis berubah dan dapat dilakukan pembayaran di mana saja dengan mudah. Dibutuhkan *reminder* untuk pembayaran angsuran yang akan jatuh tempo, maka

diperlukan sistem yang menangani *reminder* yang akan jatuh tempo yang berjalan otomatis, di mana menginformasikan jumlah tagihan yang harus dibayar (Kurniawati dan Kusumawardhani, 2017). Dari permasalahan di atas, maka sangat diperlukan sebuah aplikasi yang dapat memberikan jasa pelayanan untuk menunjang proses bisnis secara *online*. Oleh karena itu, Perancangan *Website* Sistem Pengelolaan Data Pelanggan ini akan sangat diperlukan untuk mengoptimalkan pelayanan purna jual layanan internet kepada konsumen.

Dalam penelitian ini peneliti melakukan studi kasus di salah satu perusahaan RT/RW Net yang bernama GhadheNet. GhadheNet merupakan perusahaan RT/RW Net yang berlokasi di KP. Warung lobak No. 5. RT 03 RW 01 Desa Gendasari Kec. Katapang. Untuk area cakupan jaringan internet GhadheNet ini mencapai dua Kecamatan yaitu Kecamatan Katapang dan Soreang dengan semua desa tercakup jaringan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka identifikasi masalahnya adalah sebagai berikut:

1. Sulitnya untuk mengatur data pelanggan yang akan berlangganan layanan konektivitas internet.
2. Sulitnya untuk mengatur transaksi pembayaran tagihan pelanggan.
3. Bagaimana caranya agar pelanggan atau calon pelanggan RT/RW Net mendapatkan kemudahan untuk mendaftar layanan, mengecek tagihan, melihat status pembayaran dan histori pembayaran, dan juga untuk pelaku

bisnis RT/RW Net ini mendapatkan kemudahan untuk mengatur data pelanggan sehingga aktivitas bisnis antara pelanggan dan pelaku bisnis dapat berjalan secara efektif dan fleksibel

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Merancang dan mengembangkan sistem informasi berbasis *website* untuk membantu pelanggan layanan internet untuk melakukan pendaftaran layanan internet dan melakukan transaksi pembayaran layanan internet.
2. Menyediakan wadah untuk para pelaku bisnis layanan internet untuk mengatur data pelanggan serta membantu transaksi pembayaran tagihan internet melalui *web*.

1.4 Batasan Masalah

Di dalam melakukan suatu penelitian diperlukan adanya batasan suatu masalah agar penelitian tersebut lebih terarah dan memudahkan dalam pembahasan sehingga tujuan penelitian akan tercapai. Beberapa batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini membahas mengenai perancangan dan pengembangan sistem pengelolaan data pelanggan berbasis *website* untuk penyedia layanan RT/RW Net, dengan maksud untuk membantu dan mempermudah para usahawan dalam mengatur data pelanggan dan transaksi pembayaran.
2. Metode pengembangan perangkat lunak yang digunakan adalah metode *agile*.

3. Perancangan model sistem ini menggunakan diagram UML yaitu *use case* diagram, *use case scenario*, *activity* diagram, *sequence* diagram dan *class* diagram.
4. Struktur relasi basis data menggunakan pemodelan ERD
5. Bahasa pemrograman yang digunakan adalah PHP dan menggunakan *framework* Laravel, kemudian untuk sistem manajemen basis data menggunakan MySQL.
6. Halaman depan atau bagian *front-end* dari sistem pengelola data pelanggan dapat diakses oleh seluruh pengguna tanpa harus melakukan *login* terlebih dahulu, di mana dapat melihat informasi terkait profil usaha dan paket yang ditawarkan.
7. Untuk halaman belakang atau *back-end* hanya dapat diakses jika pengguna telah melakukan *login*.
8. Menu fitur yang terdapat di *back-end* yaitu: *Dashboard*, *Management Paket*, *Request Order*, *Management Customer*, *Management Transaction*.
9. Sistem dikembangkan dengan beberapa tingkatan hak akses pengguna, yaitu Admin dan Pelanggan.
 - a. Admin merupakan hak akses pengguna utama untuk *admin* sistem, *admin* dapat mengakses berbagai fitur yang ada, diantaranya : *Dashboard*, *Management Paket*, *Request Order*, *Management Customer* dan *Management Transaction*.

- b. Pelanggan merupakan hak akses untuk pelanggan atau tamu yang mengakses halaman depan web, fitur yang dapat diakses untuk pelanggan yaitu: Beranda, Harga, Cek Tagihan dan Tentang Kami.
10. Untuk sistem pembayaran, menggunakan *payment gateway* pihak ketiga yaitu Midtrans dengan membuat link pembayaran secara manual di dalam *website* Midtrans.

1.5 Metode Penelitian

1.5.1 Metode Pengumpulan Data

Adapun metode pengumpulan data yang dilakukan untuk mendapatkan data-data dan referensi yang dibutuhkan dalam menyusun penelitian ini, meliputi:

1. Studi Kepustakaan

Dilakukan dengan cara membaca dan mempelajari buku-buku, referensi yang mendukung dengan topik dengan menghimpun informasi yang relevan pada topik atau masalah yang menjadi objek penelitian. Informasi tersebut dapat diperoleh dari buku-buku, karya ilmiah, tesis, disertasi, ensiklopedia, internet, dan sumber-sumber lain yang akan dibahas dalam penyusunan skripsi ini.

2. Wawancara

Melakukan wawancara kepada pihak terkait untuk mendapatkan data yang diperlukan dalam penelitian.

3. Studi Literatur

Mencari informasi dari berbagai sumber seperti internet, buku, jurnal atau artikel yang membahas tentang pembuatan atau perancangan sistem pengelolaan data dan bagaimana pengujiannya.

1.5.2 Metode Pengembangan Sistem

Dalam penelitian ini peneliti menggunakan metode penelitian kuantitatif. Penelitian kuantitatif digunakan dalam meneliti status kelompok manusia, suatu 6 kondisi, suatu sistem pemikiran atau kelas peristiwa pada waktu tertentu. Sehingga melalui metode ini akan diperoleh data dan informasi tentang gambaran suatu fenomena, fakta, sifat serta hubungan fenomena tertentu secara komprehensif dan integral (Silalahi, 2015: 5). Dengan demikian pengulangan dalam penelitian kuantitatif dilakukan dalam rangka mendapatkan konsistensi atau reliabilitas data penelitian yang ada.

Kerangka berpikir dalam penelitian ini menggunakan metode pengembangan *agile*. Metode *agile* merupakan salah satu metodologi yang dapat digunakan untuk pengembangan perangkat lunak, *agile* memiliki arti bersifat cepat, ringkas, bebas, bergerak dan waspada. Kata dari arti *agile* tersebut yang menggambarkan *agile* berbeda dari konsep model-model proses pengembangan sistem yang sudah ada. Pengertian dari *agile software development* adalah metode pengembangan sistem berjangka waktu pendek yang memerlukan adaptasi cepat terhadap perubahan dalam bentuk apapun selama pengembangan sistem, seperti perubahan kebutuhan perangkat lunak, perubahan anggota tim, perubahan teknologi yang digunakan dan lain sebagainya (Nurzaman, 2020).

Dalam proses metode *agile* antar anggota tim akan selalu saling berinteraksi untuk memenuhi kebutuhan pengguna, sehingga perubahan dapat cepat ditanggapi oleh tim pengembang (Stellman & Greene, 2014) dalam (K. Anwar et al., 2020). Kelebihan menggunakan metode *Agile Software Development* dalam pengembangan perangkat lunak (Raharjana, 2017) dalam (K. Anwar et al., 2020), yaitu:

1. Pengguna dapat melakukan ulasan lebih awal terhadap pengembangan perangkat lunak yang dibuat.
2. Jika terjadi kegagalan nilai kerugian yang ditimbulkan tidak terlalu besar, baik secara material maupun non-material.
3. Kepuasan pelanggan terhadap perangkat lunak lebih tinggi.
4. Mengurangi tingkat risiko kegagalan dari segi non-teknis pada saat implementasi perangkat lunak.

Berikut tahapan-tahapan metode pengembangan perangkat lunak *Agile* :

1. Analisis Kebutuhan (*Requirements*)

Tahap analisis kebutuhan (*Requirements*) merupakan tahap di mana dilakukan pemahaman mengenai jalannya proses bisnis yang ada, seperti bagaimana sistem manual yang dijalankan, prosedur dan aturan yang berlaku.

2. Perancangan (*Design*)

Tahap perancangan (*design*) merupakan tahap yang berhubungan dengan prosedur pembuatan perangkat lunak secara sistematis sebelum ke tahap pengembangan atau penulisan kode.

3. Pengembangan (Development)

Tahap pengembangan (*development*) merupakan serangkaian kegiatan pengembangan perangkat lunak dengan menuliskan kode program, sehingga menjadi perangkat lunak yang sesuai dengan perancangan sistem yang dibuat.

4. Pengujian (Testing)

Tahap pengujian (*testing*) adalah tahap proses menjalankan seluruh fungsi dari bagian perangkat lunak yang telah dikembangkan, dengan tujuan untuk menghindari dan menemukan kesalahan yang mungkin terjadi, sehingga perangkat lunak terjamin kualitasnya.

5. Menyebarkan (Deployment)

Tahap menyebarkan (*deployment*) merupakan tahap membagikan dan menyebarkan perangkat lunak yang telah dikembangkan dan diuji oleh pengembang. Penyebaran dapat dilakukan dengan berbagai macam cara, tergantung dari jenis perangkat lunaknya.

6. Ulasan (Review)

Tahap terakhir yaitu tahap ulasan (*review*) adalah tahap di mana produk perangkat lunak diperiksa secara hati-hati, yang biasanya dilakukan oleh personel proyek, manajer dan pengguna yang selanjutnya akan dilakukan perbaikan secara berkala.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan tujuan dan kegunaan penelitian, yang kemudian diikuti dengan pembatasan masalah, asumsi, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang telah pernah dilakukan sebelumnya termasuk sintesisnya. Terdapat landasan teori yang mendasari pembangunan *website* seperti pengertian *web*, sistem pengelolaan data, sistem informasi, *framework*, *payment gateway*, dan *tools – tools* yang akan digunakan.

BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM

Bab ini membahas tentang analisis kebutuhan dalam membangun *website*, analisis sistem yang akan berjalan sesuai dengan metode pembangunan *website* yang digunakan. Terdapat pula perancangan antarmuka untuk web yang akan dibangun sesuai dengan hasil analisis yang telah dibuat. Model dalam

perancangan yang akan digunakan adalah pemodelan terstruktur. Bab ini juga membahas hasil implementasi dari hasil analisis dan perancangan yang telah dibuat disertai juga dengan hasil pengujian sehingga diketahui apakah sistem yang dibangun sudah memenuhi syarat sebagai *website* yang mudah digunakan.

BAB IV IMPLEMENTASI DAN UJI COBA

Bab ini berisi tentang penjelasan dalam pengoperasian web secara bertahap dan juga evaluasi dari implementasi web ini, termasuk kelebihan dan kekurangannya menggunakan metode kuantitatif.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran yang diperoleh dari hasil penelitian skripsi yang telah dibuat. Kesimpulan didapatkan dari penelitian yang telah dilakukan sampai pembuatan laporan serta saran agar web yang penulis buat menjadi lebih baik lagi untuk pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 RT/RW-Net

RT/RW-Net atau Jaringan WLAN RT/RW adalah suatu konsep di mana beberapa komputer dalam suatu perumahan atau blok saling berhubungan. Dengan adanya jaringan WLAN RT/RW, warga bisa mengakses internet dari mana saja selama masih dalam cakupan jaringan. RT/RW memberdayakan pemakaian internet. Karena semua biaya mulai dari pembangunan infrastruktur, operasional dan langganan ditanggung bersama. Membangun jaringan WLAN RT/RW menggunakan *router mikrotik* sebagai manajemen pengguna, memudahkan pengguna untuk mendapatkan akses internet dengan sistem perhitungan biaya dari *mikrotik router operating system*, pengguna menyesuaikan dengan kebutuhan yang berisi *username* dan *password* untuk proses *login* ke jaringan *wireless* (Mery Andriana, 2019).

2.2 GhadheNet

Dalam penelitian ini peneliti melakukan studi kasus di salah satu perusahaan RT/RW Net yang bernama GhadheNet. GhadheNet merupakan perusahaan RT/RW Net yang berlokasi di KP. Warung lobak No. 5. RT 03 RW 01 Desa Gendasari Kec. Katapang. Untuk area cakupan jaringan internet GhadheNet ini mencapai dua Kecamatan yaitu Kecamatan Katapang dan Soreang dengan semua desa tercakup jaringan.

2.3 Konsep Dasar Sistem

2.3.1 Definisi Sistem

Menurut Sutabri (2016) Sistem Informasi Manajemen : ‘Sistem adalah terdiri atas objek-objek atau unsur-unsur yang berkaitan atau berhubungan satu sama lain sedemikian rupa sehingga unsur-unsur tersebut merupakan suatu kesatuan pemrosesan atau pengolahan yang tertentu’. Berdasarkan pendapat para pakar di atas maka dapat disimpulkan sistem adalah kumpulan elemen-elemen yang saling bekerjasama dan berinteraksi untuk memproses masukan kemudian saling berhubungan untuk mencapai tujuan tertentu.

2.3.2 Karakteristik Sistem

Menurut Sutabri (2016) dalam buku Sistem Informasi Manajemen : ‘Model umum sebuah sistem adalah *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran’.

Adapun karakteristik yang dimaksud:

1. Komponen sistem (*Components*).
2. Batasan Sistem (*Boundary*).
3. Lingkungan Luar Sistem (*Environment*).
4. Penghubung Sistem (*Interface*).
5. Masukan Sistem (*Input*).
6. Keluaran Sistem (*Output*).
7. Pengolah Sistem (*Proses*).

8. Sasaran Sistem (*Objective*).

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik.

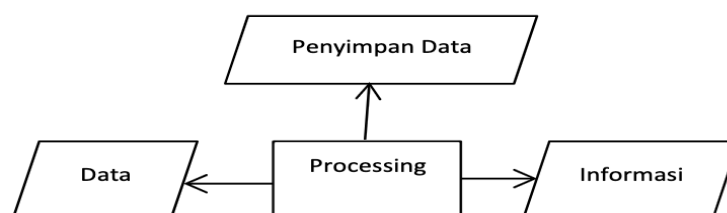
2.4 Konsep Dasar Data

2.3.1 Definisi Data

Data menurut Longkutoy dalam buku Pengenalan Komputer yang tertera di dalam buku Sutabri (2016), yang berjudul Sistem Informasi Manajemen : “yaitu suatu istilah majemuk yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan, simbol-simbol, gambar-gambar, angka-angka, huruf atau simbol yang menunjukkan suatu ide, objek, kondisi, atau situasi

2.3.2 Pengolahan Data

Menurut Sutabri (2016) dalam buku Sistem Informasi Manajemen: ‘Data merupakan bahan mentah untuk diolah, yang hasilnya kemudian menjadi informasi. Dengan kata lain, data yang diperoleh harus diukur dan dinilai baik buruknya, berguna atau tidak dalam hubungannya dengan tujuan yang akan dicapai’.



Gambar 2. 1 Pemrosesan Data (Fitri dan Nia, 2018)

2.5 Website

Pengertian *website* merupakan kumpulan dari halaman yang berisi informasi baik berupa teks, gambar, animasi, suara dan video yang disajikan dalam bentuk data digital melalui jalur koneksi internet, sehingga informasi dapat diakses dan dilihat oleh seluruh orang di berbagai belahan dunia (Abdulloh, 2018). Karakteristik utama *website* merupakan halaman-halaman saling terhubung satu sama lain yang memiliki domain sebagai alamat (url) atau www (*World Wide Web*) yang disimpan pada sebuah hosting, sehingga *website* dapat diakses menggunakan jaringan internet dengan berbagai browser seperti chrome, mozilla firefox, internet explorer (IE), dan lainnya. Suatu website dapat dibangun tanpa dengan menggunakan koneksi internet melalui localhost, dengan menggunakan *software* diantaranya database (MySQL, Oracle), web server apache, PHP editor (macromedia, notepad++), dan browser (chrome, mozilla firefox) (Elgamar, 2020).

Website terdiri dari beberapa jenis (Abdulloh, 2018), diantaranya:

1. *Website* statis yaitu *website* yang isinya tidak diperbarui secara berkala, sehingga isinya selalu tetap dari waktu ke waktu, biasanya digunakan untuk menampilkan profil perusahaan atau organisasi.
2. *Website* dinamis, merupakan kebalikan dari *website* statis yang isinya diperbarui secara berkala, biasanya jenis *website* digunakan untuk menunjang aktivitas bisnis yang berkaitan dengan internet oleh perusahaan atau perorangan, seperti *website* berita atau blog.
3. *Website* interaktif, adalah bagian dari *website* dinamis, hanya saja pembaruan informasi cenderung lebih banyak dilakukan oleh pengguna daripada pengelola

website itu sendiri, contohnya *website* media sosial seperti Facebook dan Twitter atau *website marketplace* seperti Bukalapak dan Tokopedia atau yang lainnya.

2.6 Laravel

Menurut situs resminya (laravel.com), Laravel merupakan sebuah framework aplikasi web dengan sintaks yang ekspresif dan elegan. Kerangka kerja web yang menyediakan struktur dan titik awal untuk membuat aplikasi, memungkinkan kita untuk fokus menciptakan sesuatu yang luar biasa sementara detailnya telah di *handle* oleh Laravel.

Laravel berusaha untuk memberikan pengalaman pengembang yang luar biasa, sambil menyediakan fitur-fitur canggih seperti injeksi ketergantungan menyeluruh, lapisan abstraksi basis data ekspresif, antrian dan pekerjaan terjadwal, pengujian unit dan integrasi, dan banyak lagi.

2.7 MySQL

MYSQL adalah sebuah *database* yang digunakan untuk menyimpan berbagai informasi di dalam *website* dengan cara membaginya berdasarkan kategori tertentu. Informasi – informasi yang disimpan tersebut saling terkait antara satu dengan yang lainnya. MYSQL memiliki konsep RDBMS atau *Relational Database Management System*. RDBMS dapat memungkinkan seorang *admin* menyimpan banyak informasi ke dalam tabel-tabel yang dibuat, di mana tabel-tabel tersebut saling terkait antara satu dengan yang lain (Aristania & Wardati, 2017).

2.8 Bootstrap

Bootstrap dikembangkan oleh Mark Otto dan Jacob Thornton dari Twitter. *Framework* ini diluncurkan sebagai produk *open source* pada Agustus 2011 di GitHub. Bootstrap memiliki fitur-fitur komponen *interface* yang bagus seperti *Typography, Forms, Buttons, Tabela, Navigations, Dropdowns, Alert, Modals, Tabs, Accordion, Carousel*, dan lain sebagainya. Dengan menggunakan bootstrap, anda dapat membuat *layout* situs yang *responsive* dengan mudah. Salah satu kelebihan yang dimiliki bootstrap adalah *framework* ini berisi kumpulan *tool* yang gratis untuk membuat layout web yang fleksibel dan responsif. *Framework* ini juga memiliki komponen *interface* bagus lainnya. Berikut ini kelebihan-kelebihan bootstrap lainnya:

1. **Menghemat waktu** - pengguna dapat menghemat waktu dan tenaga dengan menggunakan berbagai desain *template* dan kelas yang sudah ada dalam bootstrap,
2. **Fitur yang responsive** – dengan bootstrap, pengguna dapat dengan mudah membuat desain yang responsif. Dengan fitur-fitur ini, halaman web akan tampil responsif pada perangkat yang berbeda tanpa perlu adanya perubahan kode *markup*,
3. **Mudah digunakan** - Bootstrap sangat mudah digunakan. Siapa pun dengan pengetahuan dasar HTML dan CSS dapat menggunakan bootstrap,
4. **Semua browser populer** - bootstrap dapat digunakan pada semua *browser* modern seperti Mozilla Firefox, Google Chrome, Safari, Internet Explorer, dan Opera,

5. **Gratis** – bootstrap merupakan *framework open source* yang dapat digunakan secara gratis (N. Nurajizah, 2016).

2.9 Payment Gateway

Payment Gateway adalah layanan aplikasi *e-commerce* penyedia yang menyediakan alat untuk memproses pembayaran antara pelanggan, pedagang, dan bank melalui *World Wide Web* (WWW). *Payment Gateway* membantu mengamankan pembelian dan pelanggan informasi pembayaran dalam suatu transaksi. *Payment Gateway* melindungi informasi pembayaran dengan mengenkripsi informasi sensitif, seperti detail kartu kredit/debit, untuk memastikan bahwa informasi diteruskan dengan aman antara pelanggan dan, pemroses pembayaran. Selain mengenkripsi pembayaran informasi, *payment gateway* juga membantu dalam otorisasi pembayaran dan melindungi dari penipuan keuangan. banyak online pedagang menggunakan *gateway* pembayaran untuk keamanan, keandalannya dan otorisasi pembayaran segera (Kyaw Zay Oo, 2019).

2.10 Midtrans

Sistem pembayaran otomatis menggunakan *Payment Gateway* yang disediakan oleh Midtrans. Dengan melakukan Intergrasi sistem *Payment Gateway*, diharapkan customer bisa mendapatkan pengalaman bertransaksi yang mudah dan cepat.

Pada awalnya Midtrans merupakan sarana sistem pembayaran *online* yang populer di Indonesia maupun internasional yaitu Veritrans dengan jumlah pasar *e-commerce* yang sedikit mereka dapat merajai *Payment Gateway*, namun seiring perkembangan *e-commerce* saat ini kebutuhan yang dibutuhkan juga berbeda maka dari itu mereka mengembangkan Veritrans menjadi Midtrans yang mampu mengontrol aktivitas pembayaran *online* dengan meminimalisir terjadinya risiko saat melakukan transaksi secara *online* dengan keamanan data yang baik Midtrans mampu meningkatkan jumlah pelanggan dan menganalisis data penting yang ingin dimiliki. (Puspitasari dan Maulina, 2019).

2.11 Unified Modeling Language (UML)

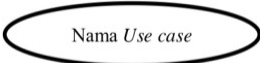
Bahasa yang dapat digunakan untuk pemodelan perangkat lunak adalah UML (*Unified Modeling Language*), yaitu berguna untuk memvisualisasi, menspesifikasikan, membangun, serta mendokumentasikan dari sebuah sistem pengembangan berbasis OO (*Object Oriented*) melalui grafik atau gambar (Mubarak, 2019).

1. Use Case Diagram

Use Case Diagram merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* diagram terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, sistem lain, maupun yang berinteraksi dengan sistem (Kurniawan dan T. Bayu. 2020).



Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel 2. 1 Simbol *Use Case Diagram* (Rosa dan Shalahuddin, 2016:155)

Simbol	Keterangan
<p data-bbox="496 1066 624 1093"><i>Use Case</i></p> 	<p data-bbox="986 1066 1442 1536">Fungsionalitas yang disediakan <i>system</i> sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i>.</p>




Tabel 2. 2 Simbol *Use Case* Diagram (Rosa dan Shalahuddin, 2016:155)

(Lanjutan)

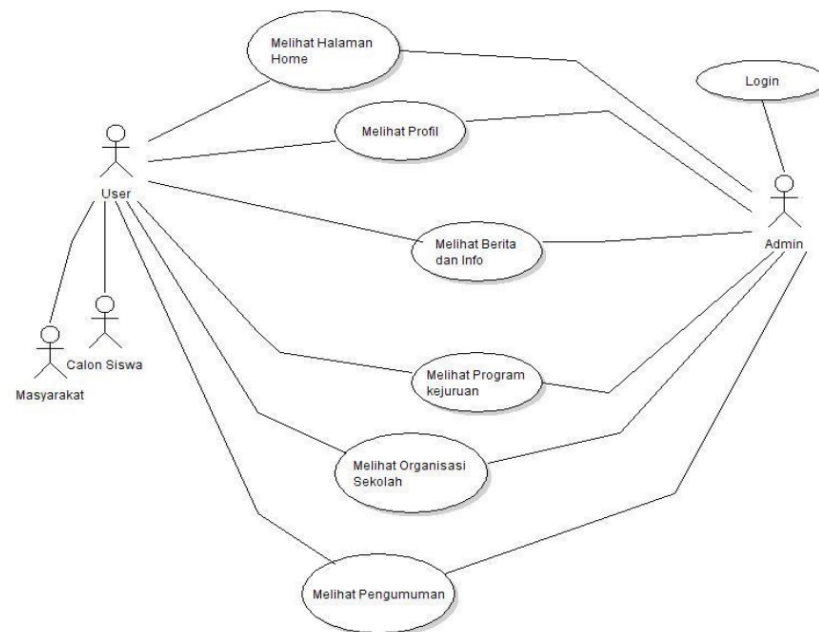
Simbol	Keterangan
<p data-bbox="491 533 671 566"><i>Aktor / Actor</i></p> 	<p data-bbox="986 533 1437 1227">Orang, proses, atau <i>system</i> lain yang berinteraksi dengan <i>system</i> informasi yang akan dibuat di luar <i>system</i> informasi yang akan dibuat itu sendiri jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p data-bbox="491 1267 783 1301"><i>Asosiasi / Association</i></p> 	<p data-bbox="986 1267 1449 1547">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

Tabel 2. 3 Simbol *Use Case* Diagram (Rosa dan Shalahuddin, 2016:155)

(Lanjutan)

Simbol	Keterangan
<p data-bbox="480 528 970 571">Ekstensi / <i>Extend</i></p> 	<p data-bbox="970 528 1445 1030">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. Mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.</p>
<p data-bbox="480 1126 970 1169">Generalisasi / <i>Generalization</i></p> 	<p data-bbox="970 1126 1445 1456">Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p data-bbox="480 1496 970 1538">Menggunakan / <i>include</i></p> 	<p data-bbox="970 1496 1445 1904">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>

Contoh *Use Case* diagram:





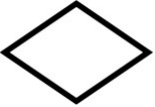


Gambar 2. 2 Use Case Diagram (Mubarak, 2019)

2. *Activity* Diagram

Activity diagram merupakan model yang menggambarkan alur dari sistem yang sedang dibuat, bagaimana serangkaian aktivitas dalam sistem berjalan dari awal hingga akhir (Azwanti, 2017).

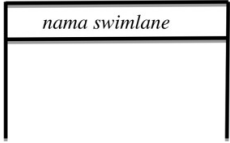
Simbol-simbol yang digunakan dalam *Activity* Diagram yaitu:

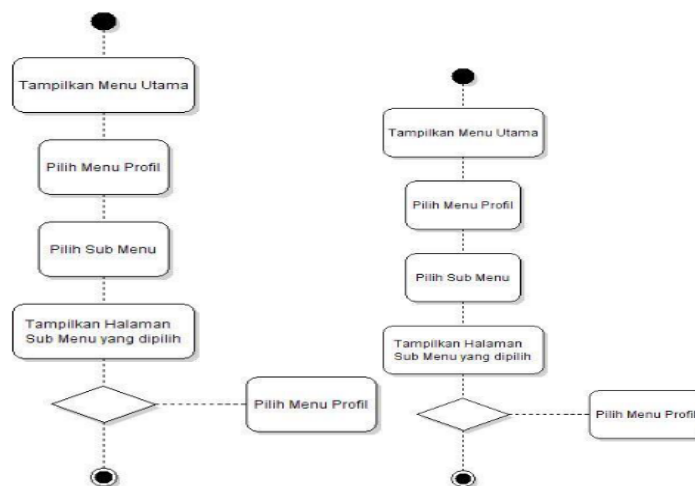
Tabel 2. 4 Simbol Activity Diagram (Rosa dan Shalahuddin, 2016:162)

Simbol	Keterangan
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan <i>system</i> , aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan <i>system</i> , sebuah diagram aktivitas memiliki sebuah status akhir.

Tabel 2. 5 Simbol Activity Diagram (Rosa dan Shalahuddin, 2016:162)

(Lanjutan)

Simbol	Keterangan
<p data-bbox="327 573 454 607">Swimlane</p> 	<p data-bbox="662 573 1348 680">Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

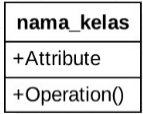


Contoh *Activity Diagram*:**Gambar 2. 3** *Activity Diagram* (Mubarak, 2019)

3. Class Diagram

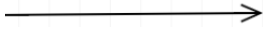

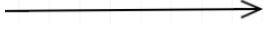

Class diagram berfungsi untuk menggambarkan struktur kelas dari sistem, seperti hubungan antara suatu kelas dengan kelas lain dan penjelasan detail dari tiap-tiap kelas dari suatu sistem yang terdiri nama, atribut dan operasi yang terjadi pada kelas (Hendini, 2016; Mubarak, 2019).

Simbol-simbol yang digunakan dalam *Class* Diagram yaitu:

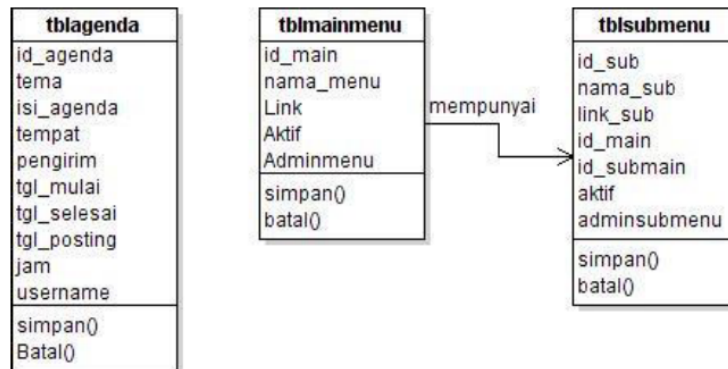
Tabel 2. 6 Simbol *Class* Diagram (Rosa dan Shalahuddin, 2016:146)

Simbol	Keterangan
<p data-bbox="331 931 411 965">Kelas</p> 	<p data-bbox="802 931 1150 965">Kelas pada struktur sistem</p>
<p data-bbox="341 1335 632 1368">Antarmuka / <i>interface</i></p> 	<p data-bbox="802 1335 1437 1447">Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek</p>
<p data-bbox="336 1727 620 1760">Asosiasi / <i>association</i></p> 	<p data-bbox="802 1727 1437 1839">Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>

Tabel 2. 7 Simbol *Class Diagram* (Rosa dan Shalahuddin, 2016:146) (Lanjutan)

Simbol	Keterangan
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Contoh *Class Diagram*:



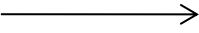
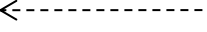


Gambar 2. 4 Class Diagram (Mubarak, 2019)

4. *Sequence Diagram*

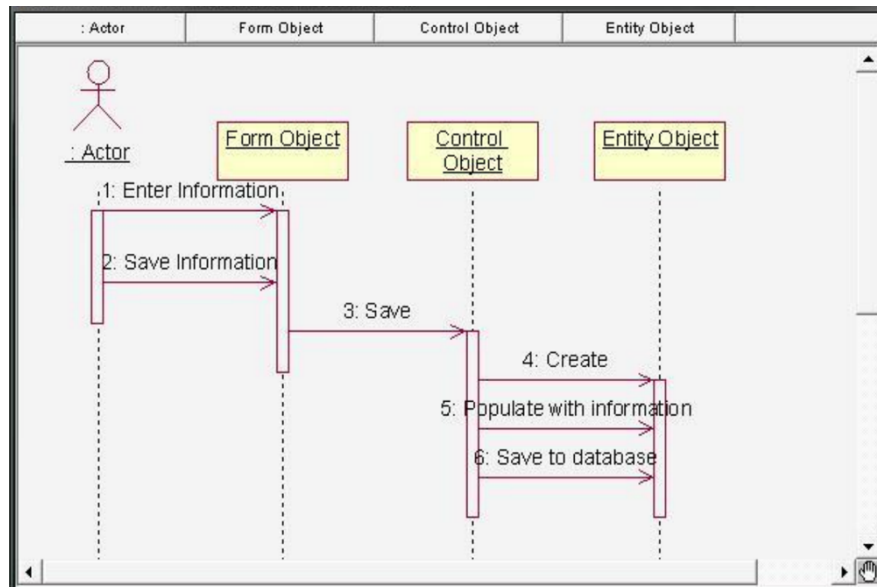
Diagram Sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel 2. 8 Simbol Sequence Diagram (Rosa dan Shalahuddin, 2018:165)

Judul	Keterangan
<< <i>Entity Class</i> >>	<i>Entity class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
<< <i>Boundary Class</i> >>	<i>Boundary class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
<< <i>Control Class</i> >>	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
<i>Message</i> 	<i>Message, symbol</i> mengirim pesan antar <i>class</i> .
<i>Recursive</i> 	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
<i>Activation</i> 	<i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
<i>Lifeline</i> 	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

Contoh *Sequence Diagram*:



Gambar 2. 5 Contoh *Sequence Diagram* (Tri Mardiani, G. 2017)

2.12 Basis Data (*Data Base*)

Secara garis besar, kita bisa mengkategorisasi sistem teknologi informasi sebagai *Online transactional processing (OLTP) system* dan *Online Analytical Processing (OLAP) system*. *OLTP system* dikategorisasikan dari banyaknya jumlah transaksi pendek online (*Read, Insert, Update, Delete*). Di sekitar tahun 2008, terjadi luapan basis data baru yang terciptakan, dan tidak satupun dari sistem tersebut yang mengikuti implementasi relasi tradisional. Basis data yang baru ini, disebut juga sebagai *NoSQL databases*, yang didesain untuk menyimpan dan memproses pertumbuhan secara eksponensial dari kuantitas data ("*Big data*"). Basisdata *NoSQL* ini melakukan oportunitas dan tantangan sistem *OLTP* dan *OLAP*.

Basis data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Sebagai satu kesatuan istilah, Basis Data (Database) sendiri dapat didefinisikan, himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

Berikut merupakan nama beberapa aplikasi database yang dikategorikan menurut komersil dan nonkomersil:

1. Aplikasi database komersil seperti: DB2, Microsoft SQL Server, Oracle, Sybase dan Teradata.
2. Aplikasi database nonkomersil seperti: MySQL, Firebird, PostgreSQL.

Ada juga database gratis yang biasa digunakan developer ketika sedang mengembangkan suatu aplikasi. Beberapa contohnya antara lain : PHPMyAdmin, MySQL Workbench, DBVisualizer, SQL Power Architect, Toad for MySQL, HeidiSQL, Squirrel SQL, pgAdmin, phpPgadmin.

2.13 *Entity Relational Diagram (ERD)*

ERD merupakan sebuah diagram atau gambar yang menunjukkan bagaimana informasi akan di simpan, dibuat dan digunakan di dalam sebuah sistem informasi (Taufik & Ermawati, 2017).

1. *Entity*

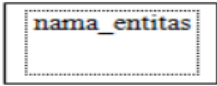
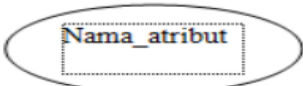
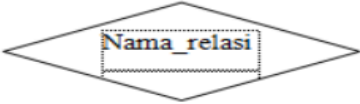

Menurut Indrajani, 2011 dalam (Taufik & Ermawati, 2017) “*Entity* yaitu kumpulan objek-objek dengan sifat (*property*) yang sama, yang diidentifikasi oleh *interface* mempunyai eksistensi yang independen”. Sebuah entitas dapat kita beri nama dengan kata benda atau dapat kita kelompokkan menjadi

- a. Nama Orang
- b. Nama Benda
- c. Nama Lokasi
- d. Nama Kejadian (terdapat unsur waktu didalamnya)

2. *Relationship*

Relationship hubungan antara sebuah *entity* dengan *entity* lainnya yang memiliki keterhubungan kedua tipe entitas tersebut (Taufik & Ermawati, 2017).

Tabel 2. 9 Simbol Entity Relationship Diagram (Rosa, A.S., dan M. Shalahuddin)

No.	Simbol	Keterangan
1.	<p>Entitas / <i>entity</i></p> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data.
2.	<p>Atribut</p> 	Field atau kolom data yang butuh disimpan dalam suatu entitas.
3.	<p>Relasi</p> 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
4.	<p>Asosiasi / <i>association</i></p> 	Garis penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakai.

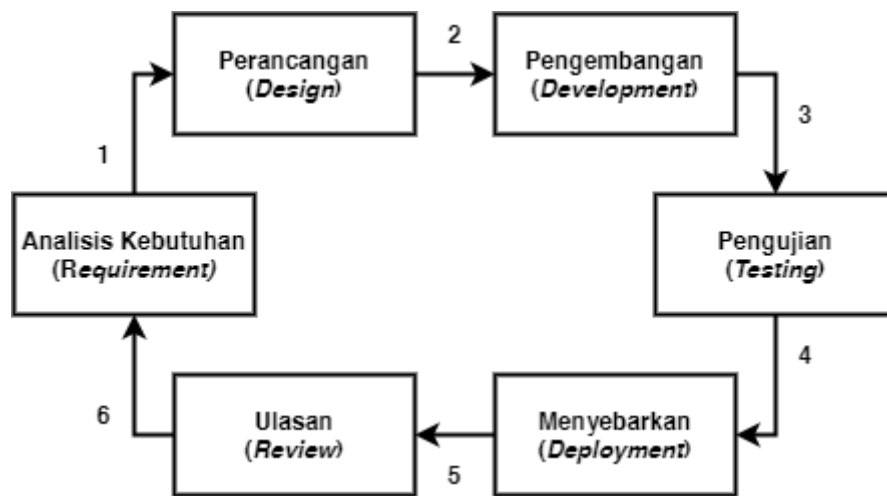
2.14 Metode *Agile*

Metode *agile* merupakan salah satu metodologi yang dapat digunakan untuk pengembangan perangkat lunak, *agile* memiliki arti bersifat cepat, ringkas, bebas, bergerak dan waspada. Kata dari arti *agile* tersebut yang menggambarkan *agile* berbeda dari konsep model-model proses pengembangan sistem yang sudah ada. Pengertian dari *agile software development* adalah metode pengembangan sistem berjangka waktu pendek yang memerlukan adaptasi cepat terhadap perubahan dalam bentuk apapun selama pengembangan sistem, seperti perubahan kebutuhan perangkat lunak, perubahan anggota tim, perubahan teknologi yang digunakan dan lain sebagainya (Nurzaman, 2020).

Dalam proses metode *agile* antar anggota tim akan selalu saling berinteraksi untuk memenuhi kebutuhan pengguna, sehingga perubahan dapat cepat ditanggapi oleh tim pengembang (Stellman & Greene, 2014) dalam (K. Anwar et al., 2020). Kelebihan menggunakan metode *Agile Software Development* dalam pengembangan perangkat lunak (Raharjana, 2017) dalam (K. Anwar et al., 2020), yaitu:

1. Pengguna dapat melakukan ulasan lebih awal terhadap pengembangan perangkat lunak yang dibuat.
2. Jika terjadi kegagalan nilai kerugian yang ditimbulkan tidak terlalu besar, baik secara material maupun non-material.
3. Kepuasan pelanggan terhadap perangkat lunak lebih tinggi.
4. Mengurangi tingkat risiko kegagalan dari segi non-teknis pada saat implementasi perangkat lunak.

Berikut merupakan gambar model dari metode *agile* dan beserta penjelasannya (Haryana, 2019), yaitu:



Gambar 2. 6 Metode *Agile* (Haryana, 2019)

1. Analisis Kebutuhan (*Requirements*)

Tahap analisis kebutuhan (*Requirements*) merupakan tahap dimana dilakukan pemahaman mengenai jalannya proses bisnis yang ada, seperti bagaimana sistem manual yang dijalankan, prosedur dan aturan yang berlaku.

2. Perancangan (*Design*)

Tahap perancangan (*design*) merupakan tahap yang berhubungan dengan prosedur pembuatan perangkat lunak secara sistematis sebelum ke tahap pengembangan atau penulisan kode.

3. Pengembangan (*Development*)

Tahap pengembangan (*development*) merupakan serangkaian kegiatan pengembangan perangkat lunak dengan menuliskan kode program, sehingga menjadi perangkat lunak yang sesuai dengan perancangan sistem yang dibuat.

4. Pengujian (*Testing*)

Tahap pengujian (*testing*) adalah tahap proses menjalankan seluruh fungsi dari bagian perangkat lunak yang telah dikembangkan, dengan tujuan untuk menghindari dan menemukan kesalahan yang mungkin terjadi, sehingga perangkat lunak terjamin kualitasnya.

5. Menyebarkan (*Deployment*)

Tahap menyebarkan (*deployment*) merupakan tahap membagikan dan menyebarkan perangkat lunak yang telah dikembangkan dan diuji oleh pengembang. Penyebaran dapat dilakukan dengan berbagai macam cara, tergantung dari jenis perangkat lunaknya.

6. Ulasan (*Review*)

Tahap terakhir yaitu tahap ulasan (*review*) adalah tahap di mana produk perangkat lunak diperiksa secara hati-hati, yang biasanya dilakukan oleh personel proyek, manajer dan pengguna, yang selanjutnya akan dilakukan perbaikan secara berkala.

2.15 Pengujian Web

Pengujian merupakan suatu kegiatan menguji atau mengevaluasi kebenaran yang diinginkan melalui satu set aktivitas yang direncanakan secara sistematis. Aktivitas pengujian terdiri dari satu set atau sekumpulan langkah di mana dapat menempatkan desain kasus uji yang spesifik (Cholifah et al., 2018). Pengujian pada sistem bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang akan dilakukan pengujian, oleh karena itu pengujian sistem merupakan

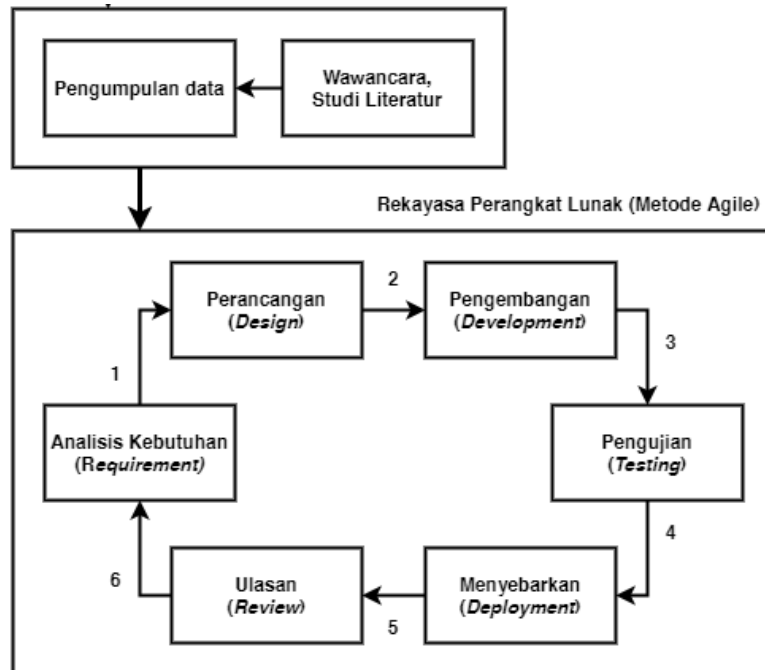
hal yang sangat penting (Salamah & Khasanah, 2017). Pentingnya pengujian pada perangkat lunak khususnya *website* tidak bisa diremehkan, karena fakta menunjukkan bahwa pengujian perangkat lunak merupakan salah satu bagian terpenting dari keseluruhan proses pembuatan perangkat lunak yang memiliki kualitas tinggi. Bahkan dalam proses pembuatan perangkat lunak, 40% waktunya digunakan untuk melakukan pengujian, di mana ini bertujuan untuk benar-benar memastikan kualitas perangkat lunak yang dihasilkan (Myers et al., 2012) dalam (Pamungkas & Rochimah, 2019).

Black box merupakan salah satu metode yang dapat digunakan untuk pengujian perangkat lunak. Pengujian dengan metode *Black box* menguji dari segi spesifikasi fungsional apakah fungsi masukan dan keluaran dari perangkat lunak sudah sesuai dengan spesifikasi yang dibutuhkan, tanpa menguji desain dan kode programnya (Cholifah et al., 2018). Saat melakukan pengujian dengan *black box*, perangkat lunak yang diuji akan dijalankan, lalu kemudian akan diperiksa apakah kebutuhan pengguna yang didefinisikan di awal pembuatan telah terpenuhi, tanpa harus mengetahui *listing* programnya. (Salamah & Khasanah, 2017).

BAB III

PEMBAHASAN

Sebelum melangkah lebih jauh pada proses penelitian, dibuat terlebih dahulu desain penelitian sebagai tahapan-tahapan yang akan dilakukan untuk mempermudah peneliti. Berikut merupakan gambar desain penelitian yang telah dibuat:



Gambar 3. 1 Desain Penelitian

3.1 Pengumpulan Data

Teknik pengumpulan data merupakan proses mencari informasi atau data yang dibutuhkan untuk penelitian, tahap pengumpulan data dilakukan sebelum proses pengembangan sistem, peneliti mengumpulkan data primer melalui wawancara dan data sekunder melalui studi literatur dari internet, buku-buku, jurnal dan artikel yang sesuai dengan penelitian.

3.1.1 Wawancara

Dalam proses pengumpulan data dilakukan wawancara dengan pihak terkait untuk mendapatkan data atau informasi mengenai kebutuhan sistem dan bagaimana alur proses sistem informasi berjalan, sehingga sistem informasi yang dikembangkan nantinya akan tercapai sesuai dengan keinginan.

3.1.2 Studi Literatur

Pada tahap pengumpulan studi literatur peneliti mencari informasi dari berbagai sumber seperti internet, buku, jurnal atau artikel yang membahas tentang pembuatan atau perancangan sistem informasi dan bagaimana pengujian sistem, serta bagaimana proses alur sistem informasi dan pengelolaan data di masyarakat biasanya dilakukan.

3.2 Rekayasa Perangkat Lunak (*Agile*)

Perancangan dan pengembangan sistem akan dibahas detail dari dari setiap langkah-langkah yang terdapat pada metode pengembangan sistem *agile*, mulai dari langkah analisis kebutuhan (*requirement*), perancangan (*design*), pengembangan (*development*), pengujian (*testing*), penyebaran (*deployment*) dan ulasan (*review*).

3.2.1 Analisa Kebutuhan (*Requirements*)

Pada tahap ini peneliti melakukan analisis kebutuhan untuk menentukan apa saja yang diperlukan untuk membangun sistem pengelolaan data pelanggan RT/RW Net, seperti menentukan jumlah tingkatan hak akses pengguna, fitur atau halaman apa saja yang diperlukan, serta gambaran umum dan perancangan prosedur sistem yang diusulkan.

3.2.2.4 Tingkatan Hak Akses Pengguna

Pada sistem pengelolaan data pelanggan RT/RW Net ini terdapat dua tingkatan hak akses pengguna, dan setiap tingkatan hak akses pengguna masing-masing memiliki hak akses fitur yang berbeda-beda, yaitu:

1. *Admin*, merupakan hak akses utama untuk *admin*.
2. *Customer*, merupakan hak akses pengguna untuk pelanggan.

3.2.2.5 Fitur dan Halaman Sistem

1. *Back-End* Sistem

Berikut merupakan tabel yang berisi daftar fitur-fitur dari sistem pengelolaan data pelanggan RT/RW Net, beserta hak akses setiap pengguna terhadap fitur. Tanda (x) berarti fitur dapat diakses oleh pengguna tersebut.

No	Fitur dan Halaman	Hak Akses Pengguna	
		Admin	Customer
1.	Home	x	x
2.	Harga	x	x
3.	Cek Tagihan	x	x
4.	<i>Histori Pembayaran*</i>	x	x
5.	Tentang Kami*	x	x
6.	<i>Login</i>	x	x
7.	<i>Dashboard</i>	x	
8.	Manajemen Paket	x	
9.	Tambah Paket	x	
10.	Manajemen <i>Request Order</i>	x	
11.	Manajemen <i>Customer</i>	x	
12.	Manajemen <i>Transaksi</i>	x	

Tabel 3. 1 Fitur dan Halaman *Back-End* Sistem

Keterangan: Tanda (*) merupakan fitur yang ditambahkan saat pengembangan sistem pada proses ulasan (*review*)

2. *Front-End* Sistem

Front-End atau halaman utama dari sistem pengelolaan data RT/RW Net dapat diakses oleh semua pengguna, termasuk yang belum melakukan *login* ke dalam sistem. Adapun menu atau halaman yang terdapat pada halaman utama yaitu:

- a. Halaman *home*, merupakan halaman utama atau awal dari sistem.
- b. Halaman harga, berfungsi untuk pendaftaran dan pemilihan paket langganan internet untuk konsumen.
- c. Halaman cek tagihan, berfungsi untuk melakukan pengecekan tagihan layanan internet dan juga untuk melakukan pembayaran tagihan layanan internet.
- d. Halaman histori pembayaran, berfungsi untuk pelanggan melihat riwayat pembayaran layanan internet.*
- e. Halaman tentang kami, berfungsi untuk menunjukkan informasi alamat, kontak, sosial media dan foto-foto pelaku bisnis RT/RW Net.*
- f. Halaman *login*, berfungsi untuk melakukan *login* sebagai *admin* ke halaman *dashboard*.

Keterangan: Tanda (*) merupakan fitur yang ditambahkan saat pengembangan sistem pada proses ulasan (*review*).

3.2.2.6 Gambaran Umum Sistem yang Diusulkan

Sistem yang diusulkan adalah suatu langkah untuk membantu dan mengefektifkan aktivitas pengelolaan data dan transaksi antara pelaku bisnis dan pelanggan agar dapat dilakukan secara efisien dan juga pengelolaan data pelanggan dan transaksi seperti pendaftaran paket, cek tagihan dan transaksi pembayaran antara pelaku bisnis dan pelanggan. Sistem yang diusulkan adalah sistem berbasis web yang dapat dijalankan secara *online* melalui jaringan internet. Sebagai media komunikasi dan informasi, sistem ini dapat digunakan untuk membantu aktifitas-aktifitas seperti yang telah disebutkan di atas.

3.2.2 Perancangan (*Design*)

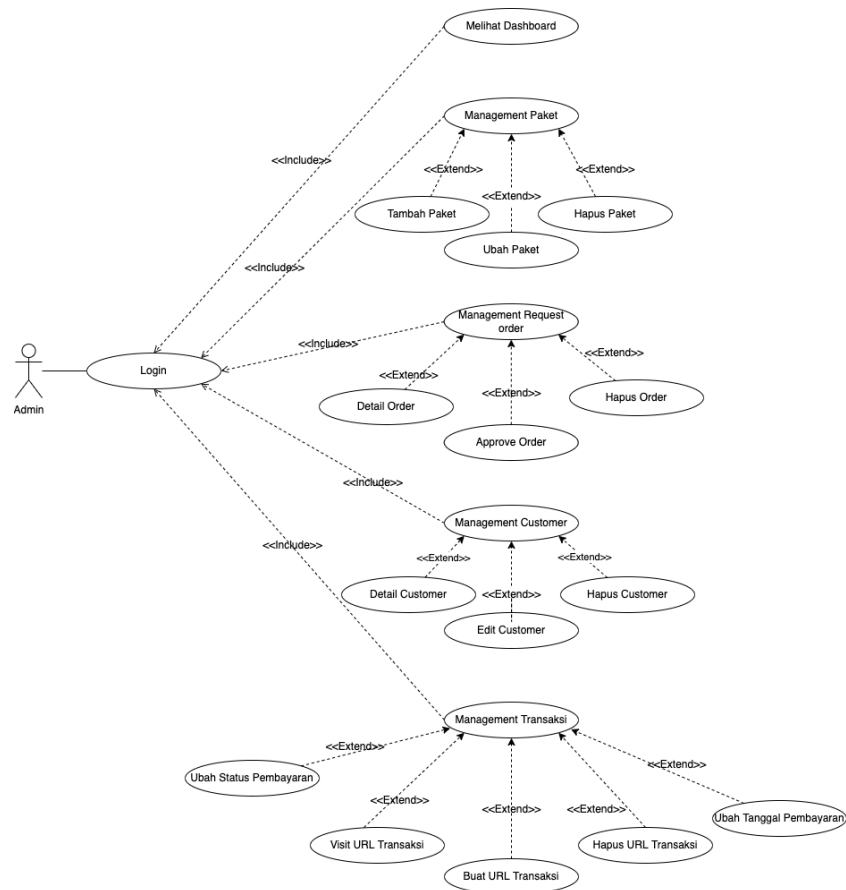
Pada tahap perancangan akan dibahas mengenai perancangan sistem yang telah dibuat, seperti UML (*Unified Modeling Language*) yang meliputi *use case diagram*, *use case scenario*, *activity diagram* dan *class diagram*. Kemudian untuk perancangan basis data menggunakan ERD (*Entity Relationship Diagram*) serta desain dari tampilan antarmuka pengguna sistem.

3.2.2.1 UML (Unified Modeling Language)

1. Use Case Diagram

a. Use Case Diagram Admin

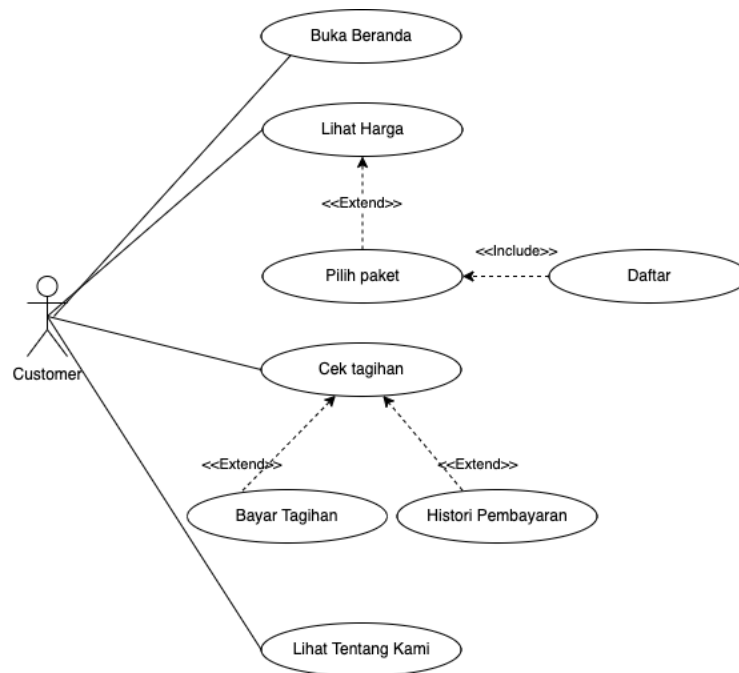
Fungsi *Admin* di sini adalah *user* yang mengelola seluruh data pelanggan mulai dari melihat informasi di *dashboard*, mengelola paket, mengelola *request order*, mengelola *customer* hingga mengelola transaksi pembayaran *customer*.



Gambar 3. 2 Use Case Diagram Admin

b. *Use Case Diagram Customer*

Customer adalah *user* tamu atau pelanggan yang mengunjungi website.



Gambar 3.3 *Use Case Diagram Customer*

2. Use Case Scenario

a. Kelola Paket

Tabel 3. 2 Use Case Scenario Management Paket

Identifikasi	
<i>Use Case</i>	<i>Management Paket</i>
Aktor	<i>Admin</i>
Deskripsi	Merupakan fitur dimana <i>admin</i> dapat menambahkan, mengubah dan menghapus paket layanan internet yang ingin di tampilkan di menu halaman harga.
Kondisi Awal	<ol style="list-style-type: none"> 1. Admin melakukan proses <i>login</i> 2. <i>admin</i> meng-klik menu <i>management</i> paket.
Aktivitas Aktor	Admin melakukan manipulasi data (Menambah, Mengubah, atau menghapus paket)
Aktivitas Sistem	<ol style="list-style-type: none"> 1. Menampilkan tabel informasi paket yang tersedia, 2. Menampilkan form penambahan atau perubahan paket 3. Dan menyimpan hasil <i>request</i> dari admin ke dalam database
Kondisi Akhir	Tampil tabel informasi paket yang telah di <i>update</i>

b. *Management Request Order***Tabel 3. 3** *Use Case Scenario Management Request Order*

Identifikasi	
<i>Use Case</i>	<i>Management Request Order</i>
Aktor	<i>Admin</i>
Deskripsi	Merupakan fitur dimana <i>admin</i> dapat melihat data <i>order</i> dari pelanggan baru yang masuk, dan juga <i>admin</i> dapat melihat detail data pelanggan tersebut, menyetujui <i>order</i> dan menghapus <i>order</i>
Kondisi Awal	<ol style="list-style-type: none"> 1. Admin melakukan proses login 2. <i>Admin</i> meng-klik menu <i>management request order</i>.
Aktivitas Aktor	<ol style="list-style-type: none"> 1. Admin melihat tabel data pelanggan baru yang masuk 2. Admin Melihat detail order baru yang masuk 3. Admin melakukan approve terhadap order baru yang untuk membuat nomor id pelanggan 4. Admin menghapus/menolak <i>order</i> baru
Aktivitas Sistem	<ol style="list-style-type: none"> 1. Menampilkan tabel informasi <i>order</i> baru 2. Men-<i>generate</i> nomor id pelanggan
Kondisi Akhir	Tampil tabel informasi <i>request order</i> yang telah di- <i>update</i>

c. *Management Customer***Tabel 3. 4** *Use Case Scenario Management Customer*

Identifikasi	
<i>Use Case</i>	<i>Kelola Customer</i>
Aktor	<i>Admin</i>
Deskripsi	Merupakan fitur dimana <i>admin</i> dapat melihat data pelanggan yang aktif/berlangganan, dan juga <i>admin</i> dapat melihat detail data pelanggan tersebut dan menghapus data pelanggan jika ada pelanggan yang sudah tidak berlangganan.
Kondisi Awal	<ol style="list-style-type: none"> 1. Admin melakukan proses <i>login</i> 2. <i>Admin</i> meng-klik menu <i>Management Customer</i>.
Aktivitas Aktor	<ol style="list-style-type: none"> 1. <i>Admin</i> melihat tabel data pelanggan yang aktif 2. Admin melihat detail data pelanggan 3. Admin melakukan manipulasi data pelanggan (<i>Update/Delete</i>)
Aktivitas Sistem	<ol style="list-style-type: none"> 1. Menampilkan tabel data informasi pelanggan aktif 2. Memproses data pelanggan sesuai <i>request admin</i>
Kondisi Akhir	Tampil tabel informasi pelanggan aktif yang telah di- <i>update</i>

d. *Management Transactions***Tabel 3. 5** *Use Case Scenario Management Transactions*

Identifikasi	
<i>Use Case</i>	Kelola Transaksi
Aktor	<i>Admin</i>
Deskripsi	Merupakan fitur di mana <i>admin</i> dapat menambahkan link/URL pembayaran melalui <i>payment gateway</i> Midtrans dan memperbarui status pembayaran dan tanggal pembayaran customer
Kondisi Awal	<ol style="list-style-type: none"> 1. Admin melakukan proses login 2. Admin meng-klik menu <i>management</i> transaksi.
Aktivitas Aktor	<ol style="list-style-type: none"> 1. Admin melihat tabel data pelanggan yang aktif beserta jumlah tagihan pelanggan 2. Admin meng-inputkan link/URL pembayaran yang dibuat melalui web Midtrans 3. Admin melihat hasil link pembayaran yang telah di-inputkan 4. Admin menghapus link/URL 5. Admin mengubah status pembayaran <i>customer</i>. 6. Admin mengubah tanggal pembayaran <i>customer</i>
Aktivitas Sistem	<ol style="list-style-type: none"> 1. Menampilkan tabel data informasi pelanggan aktif 2. memproses data pelanggan sesuai <i>request admin</i>
Kondisi Akhir	Tampil tabel informasi pelanggan yang telah di- <i>update</i>

e. Pilih Paket

Tabel 3. 6 *Use Case Scenario* Pilih Paket

Identifikasi	
<i>Use Case</i>	Pilih Paket
Aktor	<i>Customer</i>
Deskripsi	Merupakan fitur di mana <i>customer</i> dapat memilih paket layanan yang diinginkan dan melakukan pendaftaran pelanggan baru.
Kondisi Awal	1. <i>Customer</i> meng-klik menu harga
Aktivitas Aktor	1. <i>Customer</i> memilih paket yang diinginkan 2. <i>Customer</i> melakukan pengisian form pendaftaran pelanggan baru yang telah disediakan
Aktivitas Sistem	1. Menampilkan informasi paket layanan yang tersedia 2. Menyimpan data pelanggan
Kondisi Akhir	Tampil halaman pendaftaran Berhasil

f. Cek Tagihan

Tabel 3. 7 *Use Case Scenario* Cek Tagihan

Identifikasi	
<i>Use Case</i>	Cek Tagihan
Aktor	<i>Customer</i>
Deskripsi	Merupakan fitur di mana <i>customer</i> dapat mengecek tagihan dan melakukan pembayaran tagihan
Kondisi Awal	1. <i>Customer</i> meng-klik menu cek tagihan

Tabel 3. 8 *Use Case Scenario* Cek Tagihan. (Lanjutan)

Identifikasi	
<i>Use Case</i>	Cek Tagihan
Aktor	<i>Customer</i>
Aktivitas Aktor	<ol style="list-style-type: none"> 1. <i>Customer</i> menginputkan nomor id pelanggan atau nomor hp 2. <i>Customer</i> memvalidasi data tagihan sesuai harga paket yang dipilih 3. <i>Customer</i> melakukan pembayaran melalui <i>payment gateway</i> Midtrans 4. <i>Customer</i> melihat histori pembayaran
Aktivitas Sistem	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> inputan nomor id pelanggan dan nomor hp 2. Cek <i>request</i> data pelanggan dan URL pembayaran dari <i>database</i> 3. Men-<i>direct</i> ke halaman pembayaran Midtrans 4. Menampilkan histori pembayaran
Kondisi Akhir	Tampil halaman pembayaran.

g. Lihat Tentang Kami

Tabel 3. 9 *Use Case Scenario* Lihat Tentang Kami

Identifikasi	
<i>Use Case</i>	Lihat Tentang Kami
Aktor	<i>Customer</i>
Deskripsi	Merupakan fitur di mana <i>customer</i> dapat melihat informasi mengenai sejarah, <i>gallery</i> , dan informasi lainnya yang berkaitan dengan si pelaku bisnis.
Kondisi Awal	<ol style="list-style-type: none"> 1. <i>Customer</i> meng-klik menu Tentang Kami

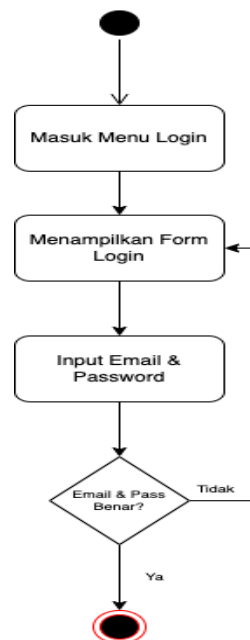
Tabel 3. 10 Use Case Scenario Lihat Tentang Kami. (Lanjutan)

Identifikasi	
<i>Use Case</i>	Lihat Tentang Kami
Aktor	<i>Customer</i>
Aktivitas Aktor	1. <i>Customer</i> melihat halaman informasi pelaku bisnis
Aktivitas Sistem	1. Menampilkan halaman Tentang Kami
Kondisi Akhir	Tampil halaman Tentang Kami

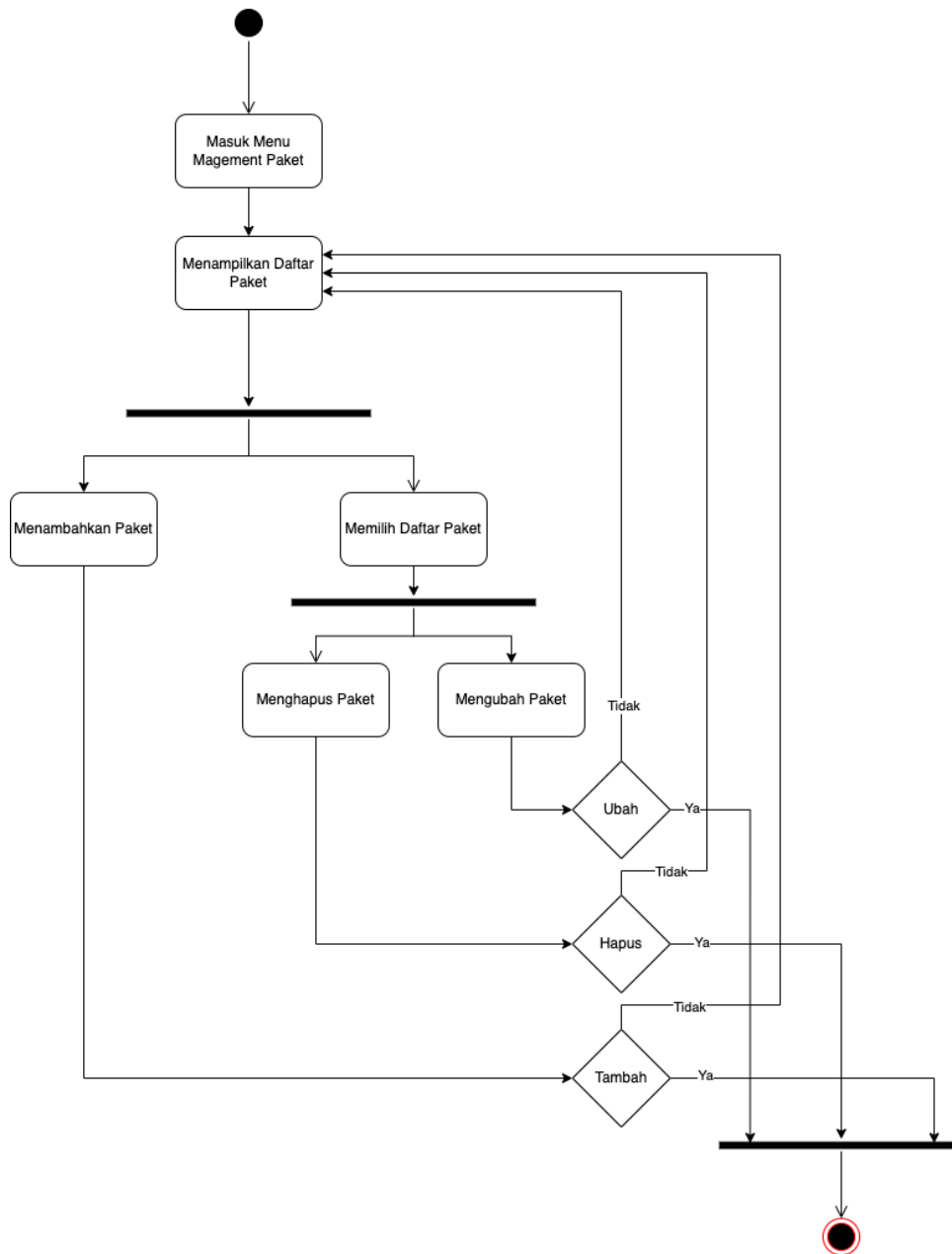
1. Activity Diagram

a. Activity Diagram Admin

1) Activity Diagram Login (Admin)

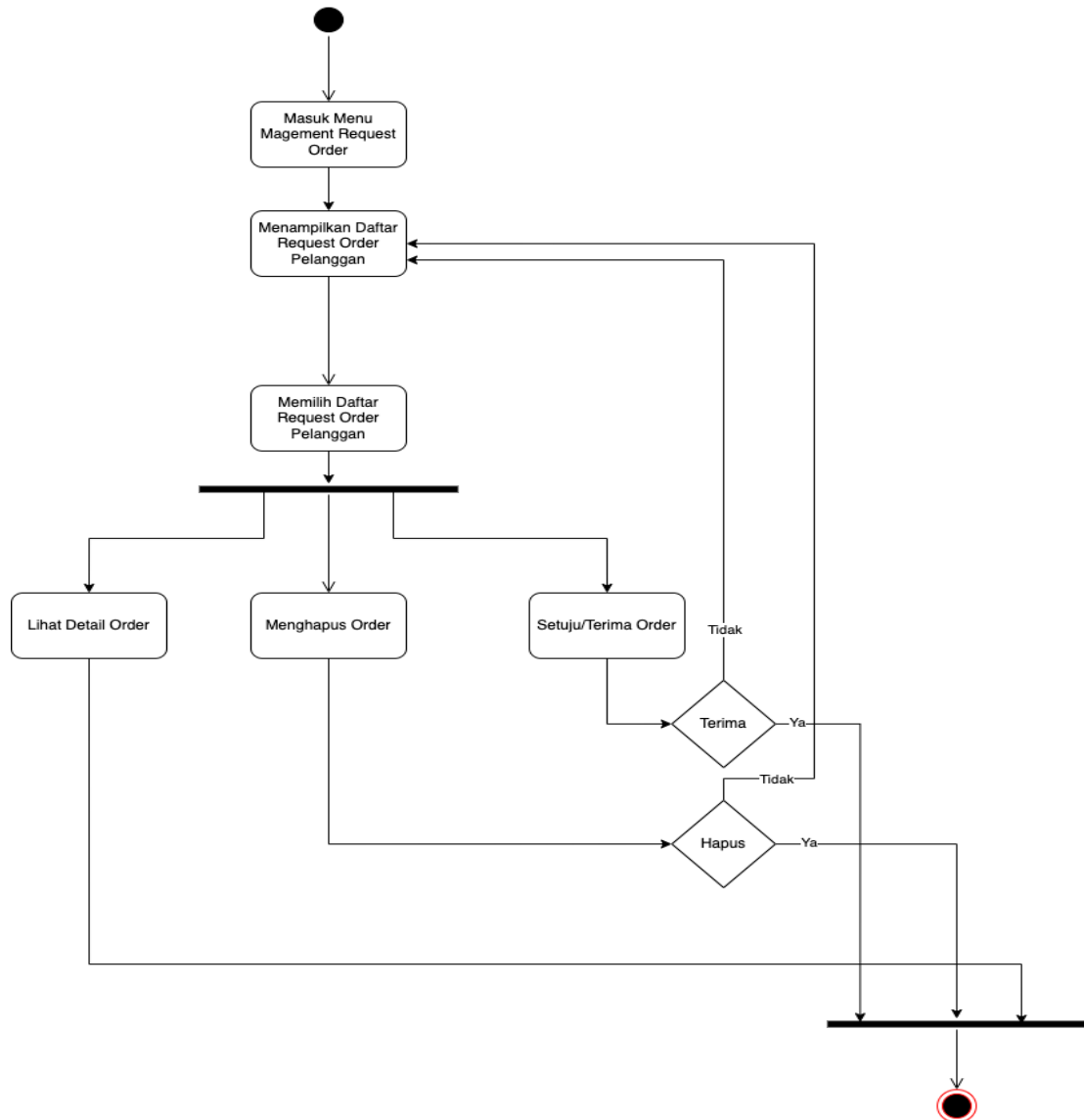


Gambar 3. 4 Activity Diagram Login (Admin)

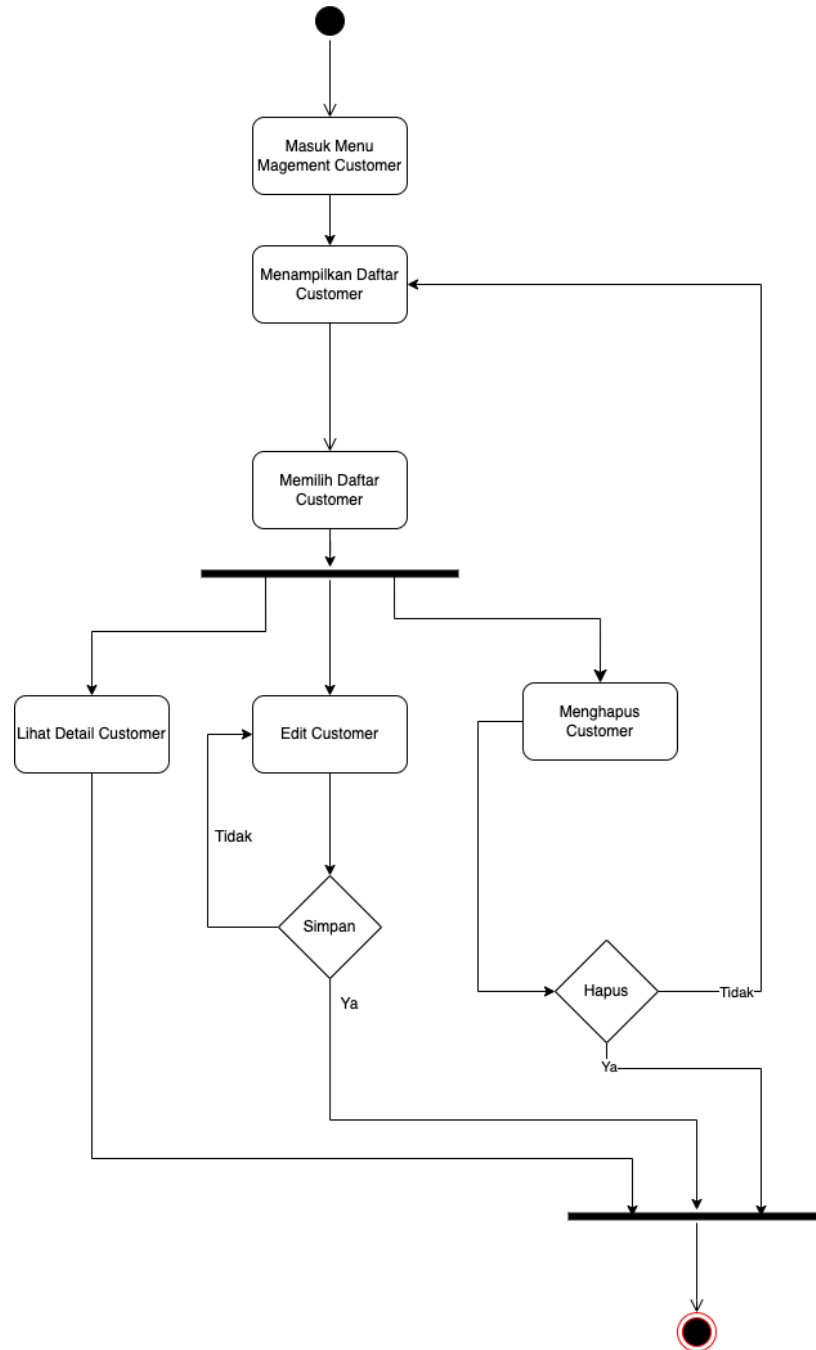
2) *Activity Diagram Management Paket (Admin)*

Gambar 3.5 *Activity Diagram Management Paket (Admin)*

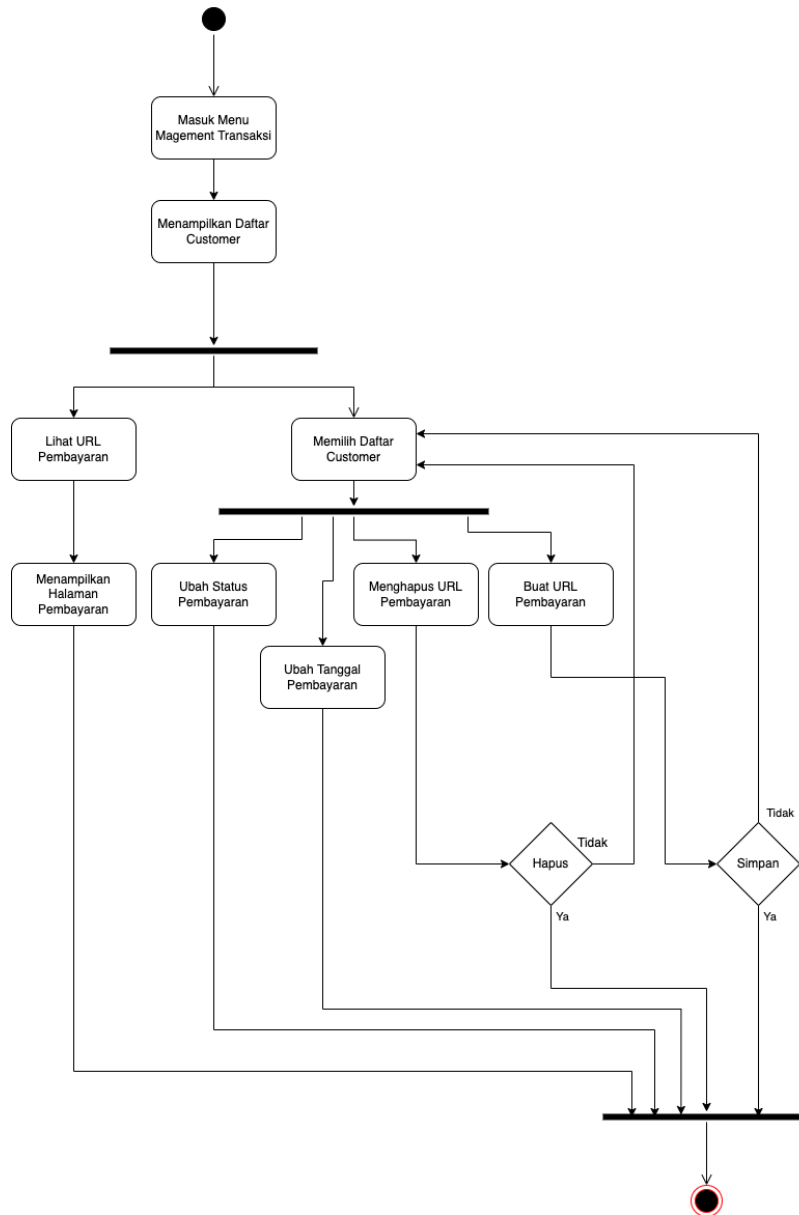
3) *Activity Diagram Management Request Order (Admin)*



Gambar 3. 6 *Activity Diagram Management Request Order (Admin)*

4) *Activity Diagram Management Customer (Admin)***Gambar 3. 7** *Activity Diagram Management Customer (Admin)*

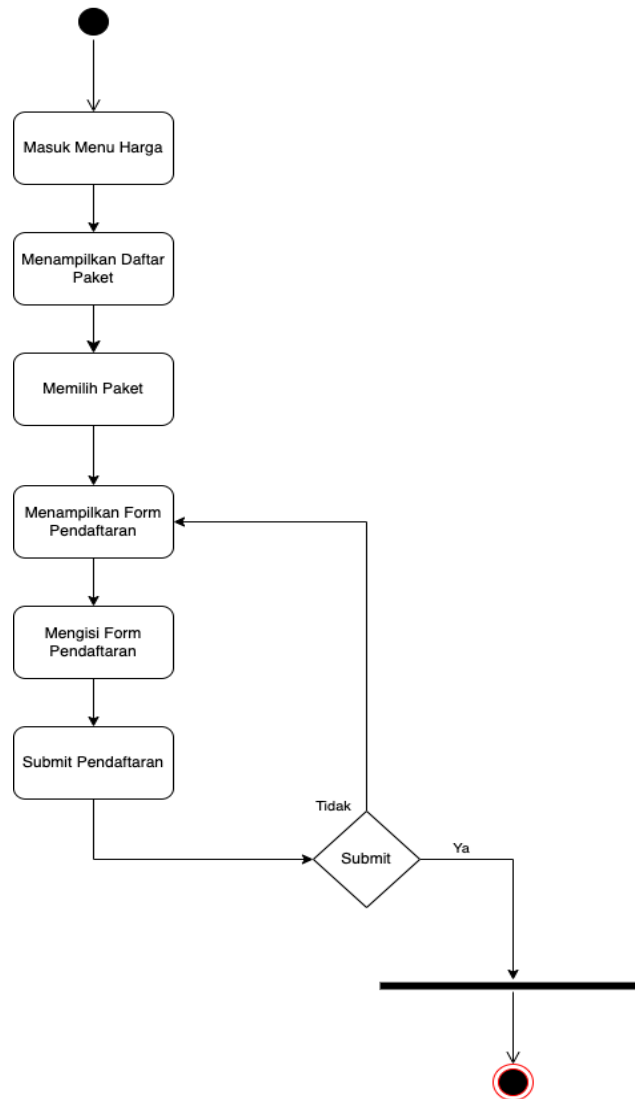
5) Activity Diagram Management Transaksi (Admin)



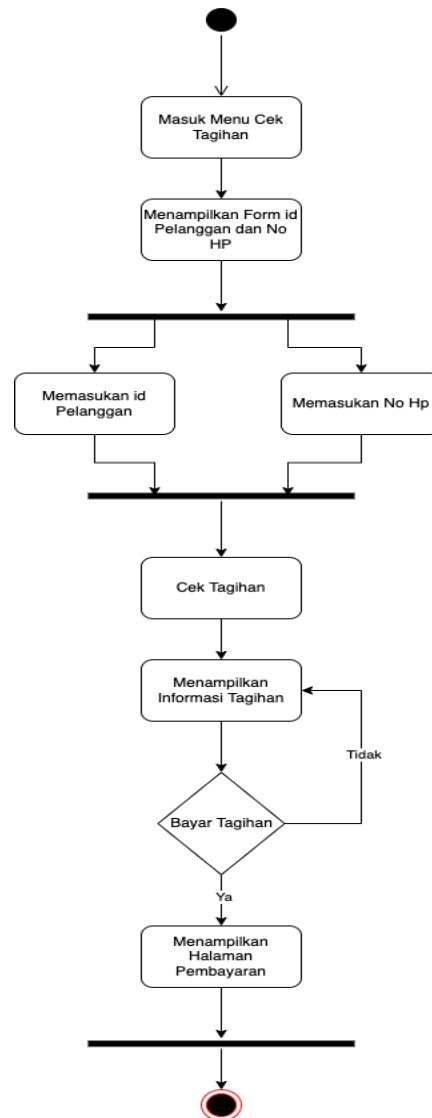
Gambar 3. 8 Activity Diagram Management Transaksi (Admin)

b. *Activity Diagram Customer*

1) *Activity Diagram Pendaftaran Pelanggan Baru (Customer)*



Gambar 3. 9 *Activity Diagram Pendaftaran (Customer)*

2) *Activity Diagram Cek Tagihan (Customer)*

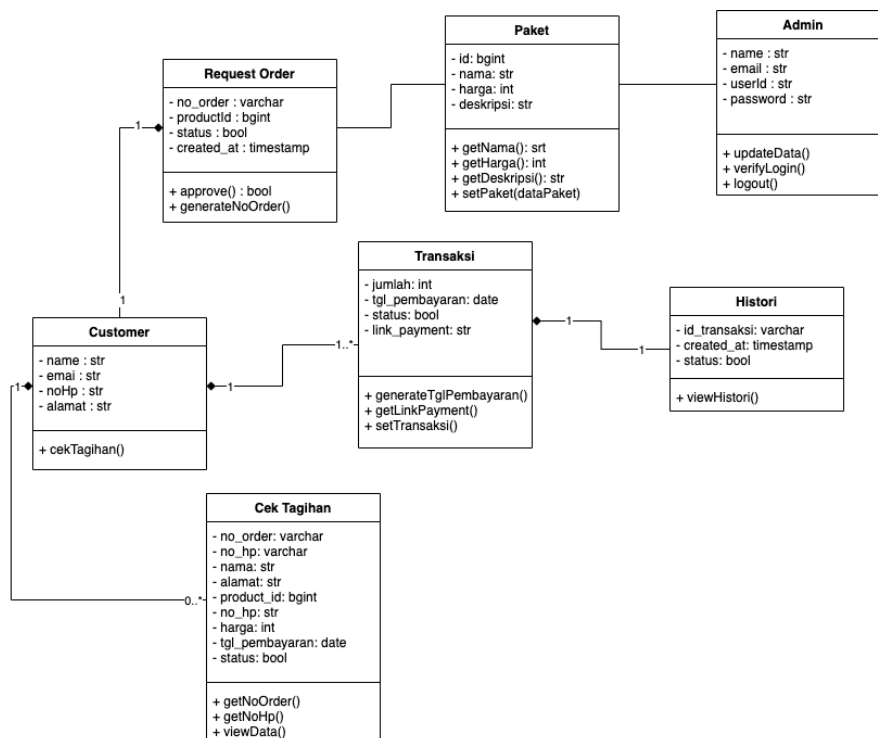
Gambar 3. 10 *Activity Diagram Cek Tagihan (Customer)*

3) Activity Diagram Lihat Tentang Kami (Customer)



Gambar 3. 11 Activity Diagram Lihat Tentang Kami

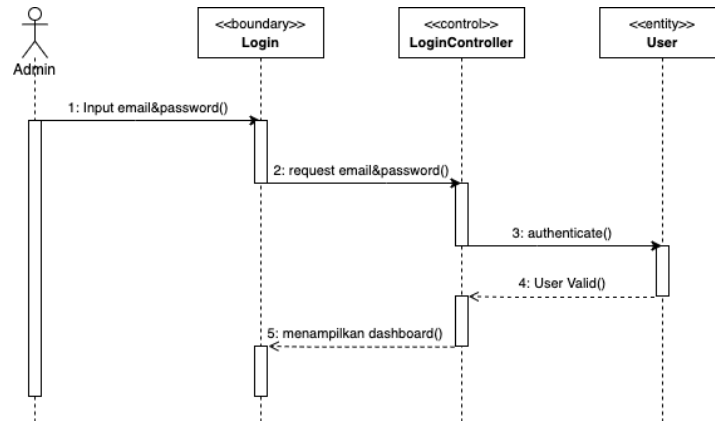
2. Class Diagram Sistem



Gambar 3. 12 Class Diagram Sistem

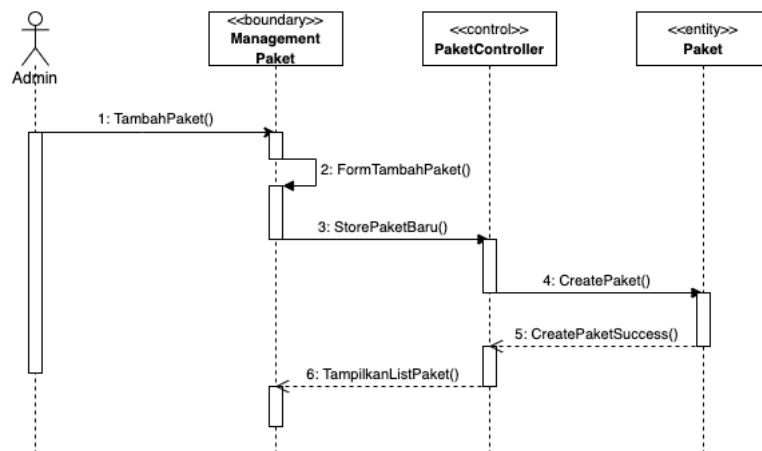
3. Sequence Diagram

1) Sequence Diagram Login

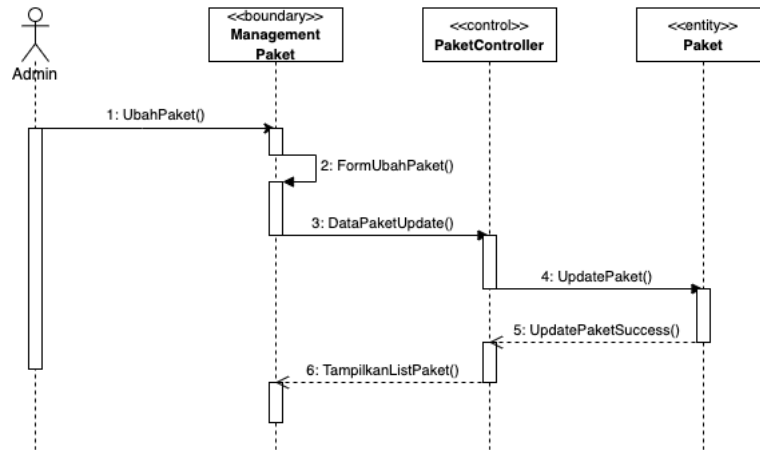
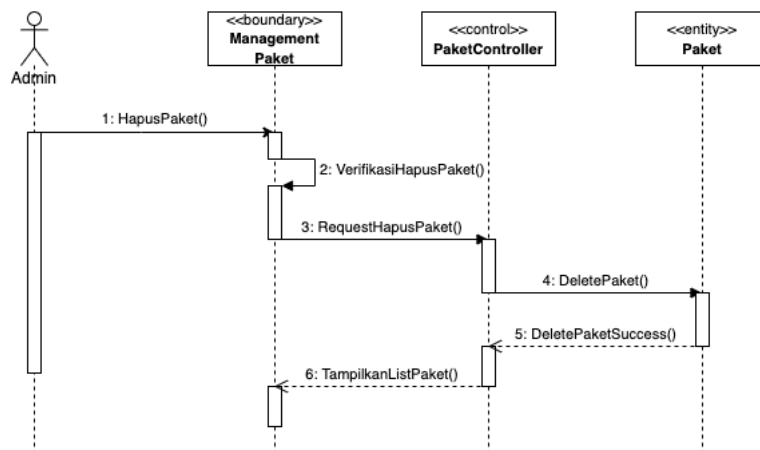


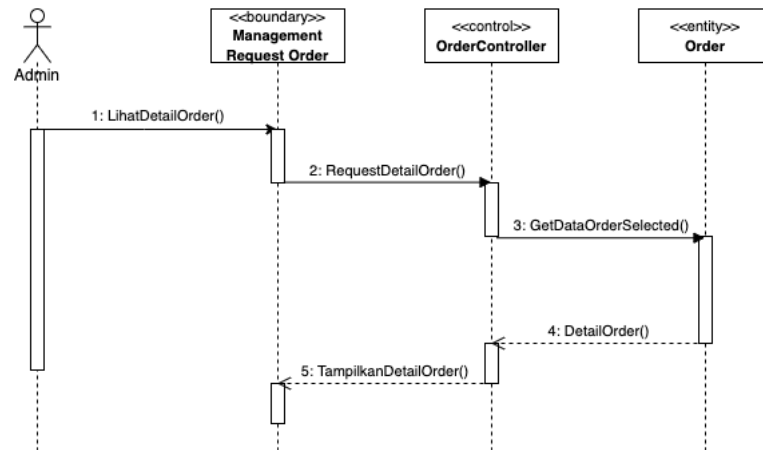
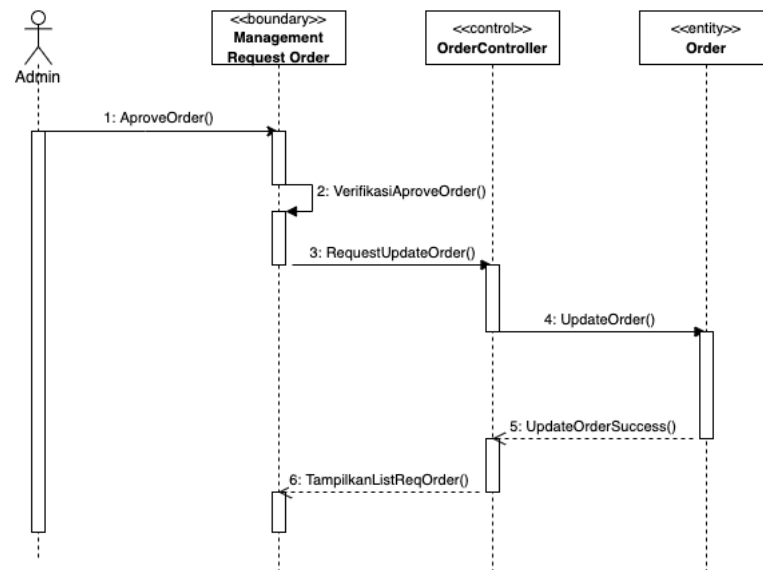
Gambar 3. 14 Sequence Diagram Login (Admin)

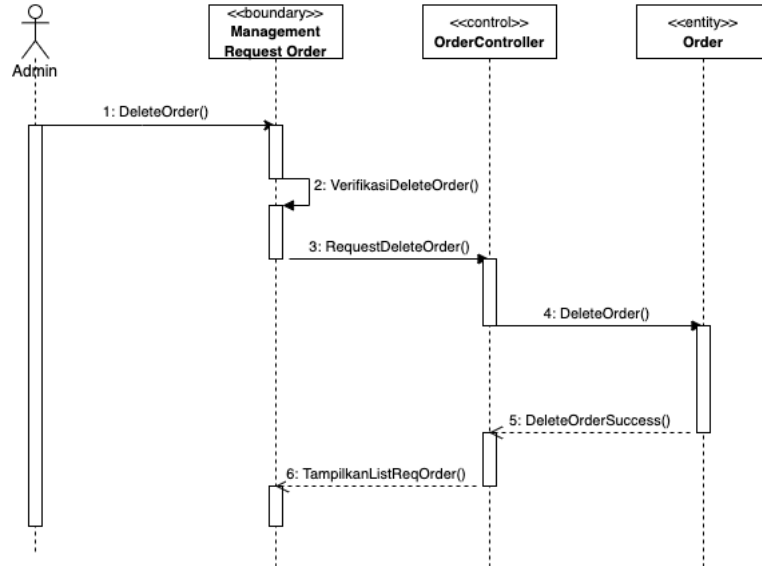
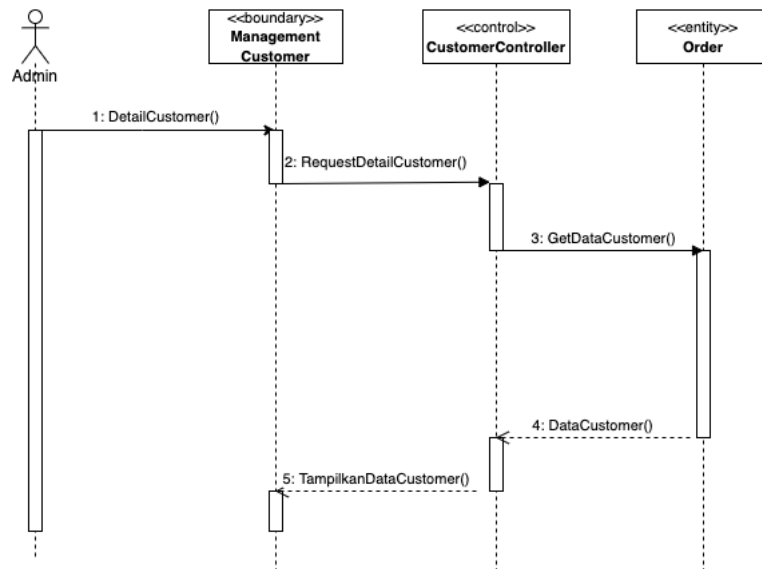
2) Sequence Diagram Tambah Paket

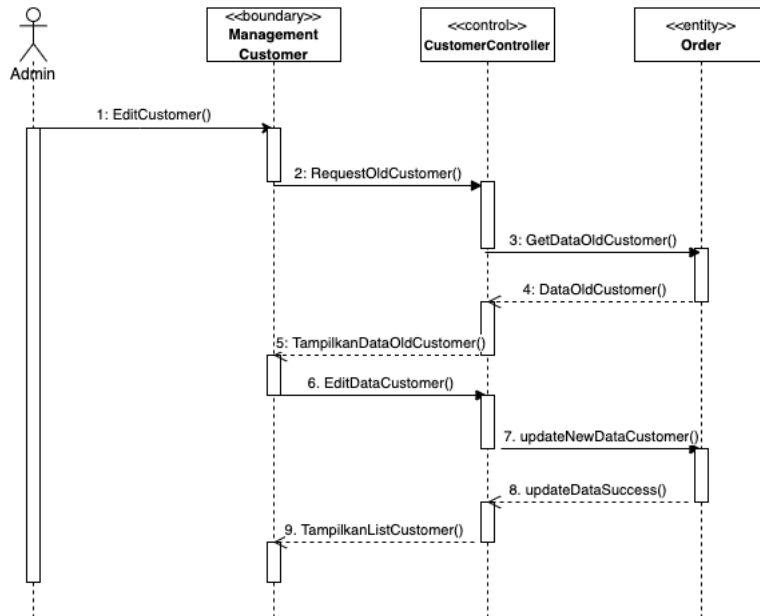
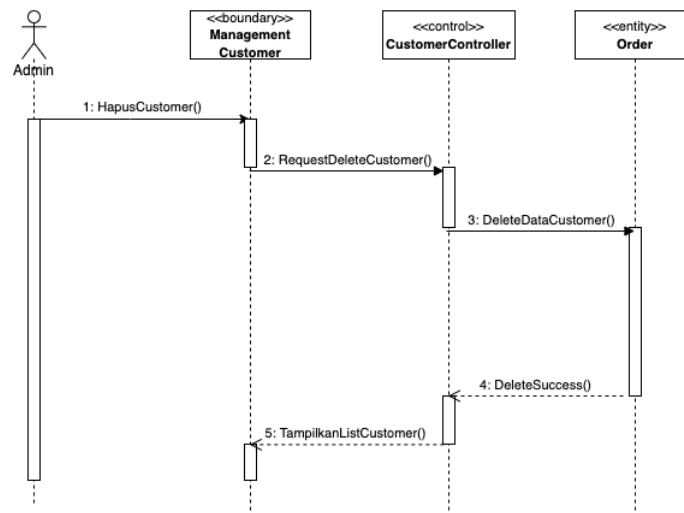


Gambar 3. 15 Sequence Diagram Tambah Paket (Admin)

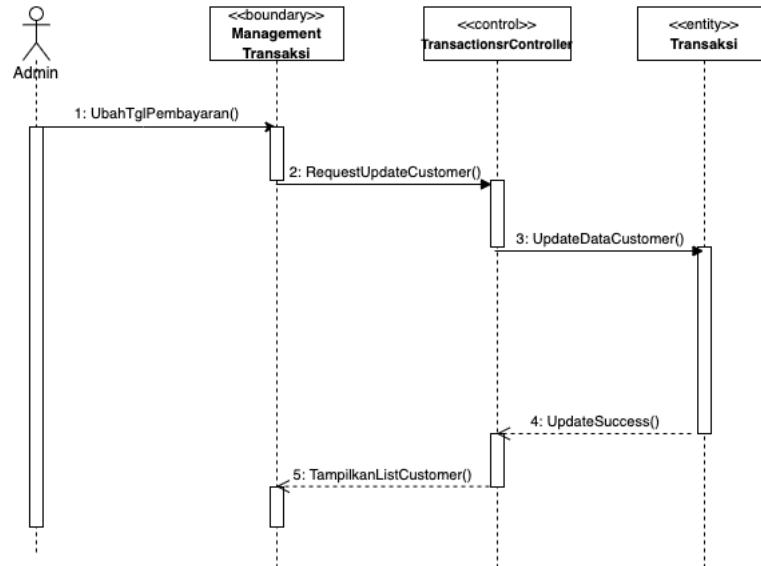
3) *Sequence* Diagram Ubah PaketGambar 3. 16 *Sequence* Diagram Ubah Paket (*Admin*)4) *Sequence* Diagram Hapus PaketGambar 3. 17 *Sequence* Diagram Hapus Paket (*Admin*)

5) *Sequence Diagram Detail Order*Gambar 3. 18 *Sequence Diagram Detail Order (Admin)*6) *Sequence Diagram Approve Order*Gambar 3. 19 *Sequence Diagram Approve Order (Admin)*

7) *Sequence Diagram Delete Order*Gambar 3. 20 *Sequence Diagram Delete Order (Admin)*8) *Sequence Diagram Detail Customer*Gambar 3. 21 *Sequence Diagram Detail Customer (Admin)*

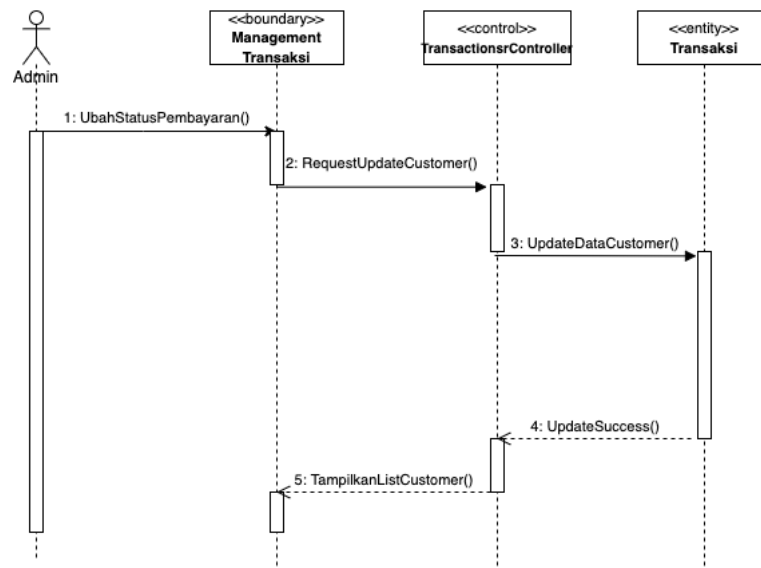
9) Sequence Diagram *Edit Customer*Gambar 3. 22 Sequence Diagram *Edit Data Customer (Admin)*10) Sequence Diagram *Hapus Customer*Gambar 3. 23 Sequence Diagram *Hapus Customer (Admin)*

11) Sequence Diagram Ubah Tanggal Pembayaran Customer

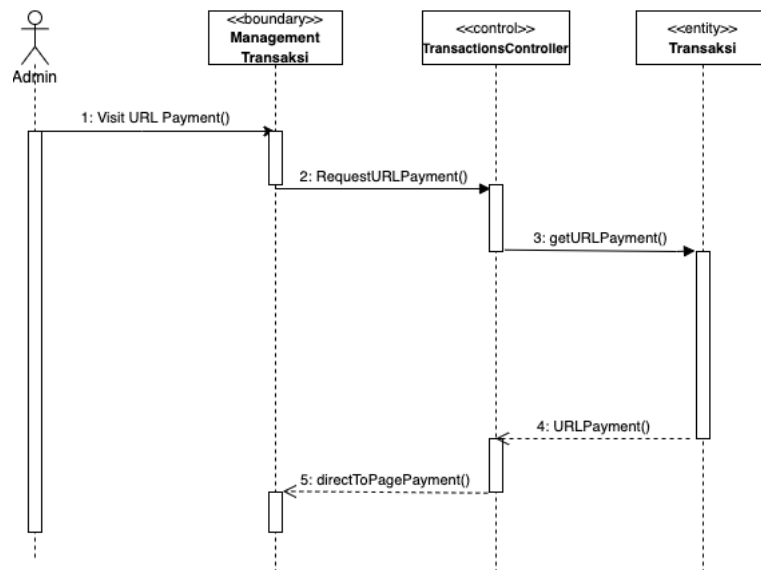
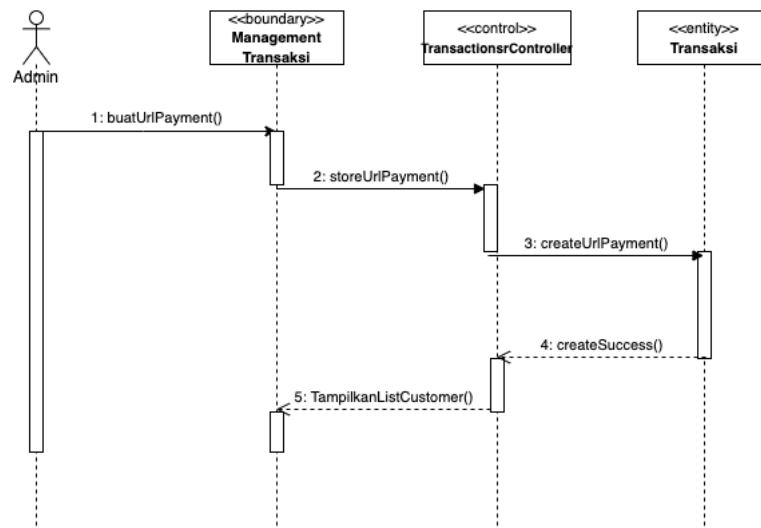


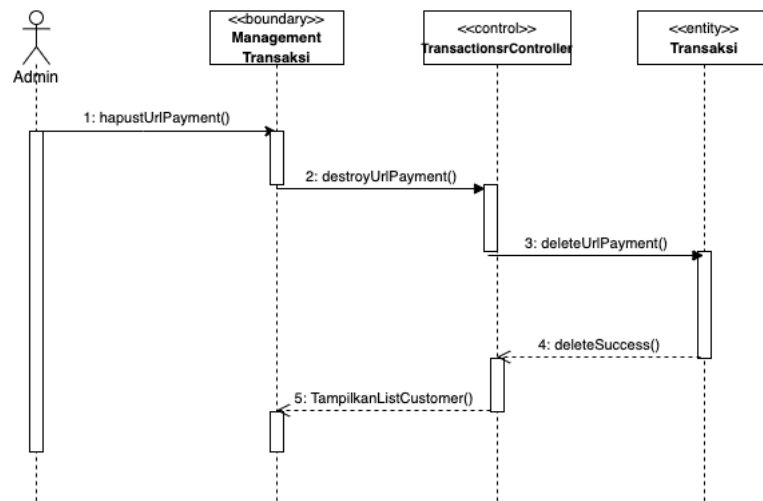
Gambar 3. 24 Sequence Diagram Ubah Tanggal Pembayaran Customer (Admin)

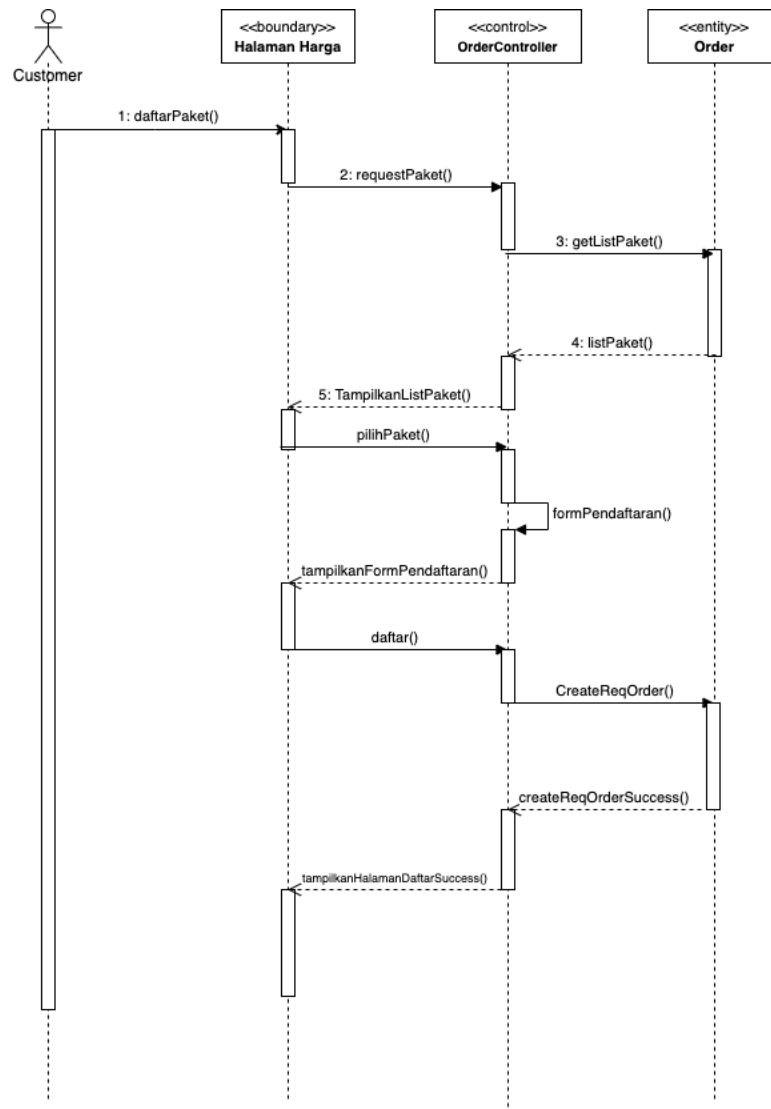
12) Sequence Diagram Ubah Status Pembayaran Customer



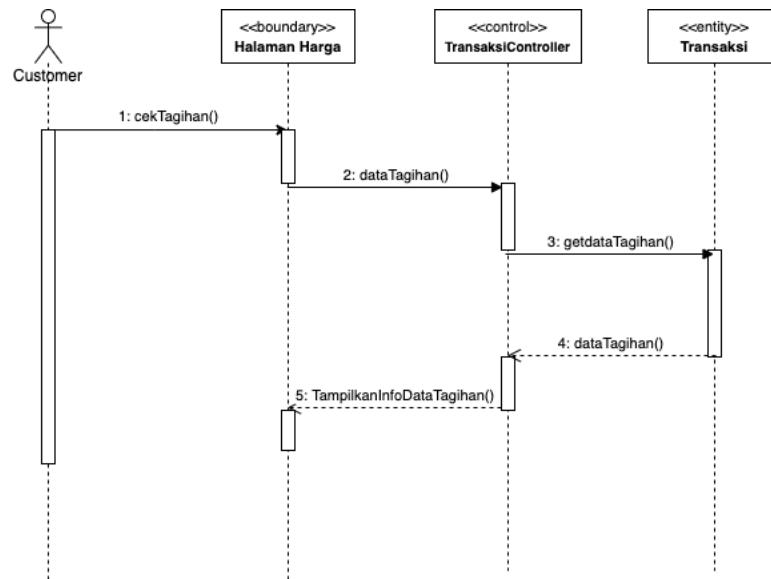
Gambar 3. 25 Sequence Diagram Ubah Status Pembayaran Customer (Admin)

13) Sequence Diagram *Visit URL Payment*Gambar 3. 26 Sequence Diagram *Visit URL Payment (Admin)*14) Sequence Diagram *Buat URL Payment*Gambar 3. 27 Sequence Diagram *Buat URL Payment (Admin)*

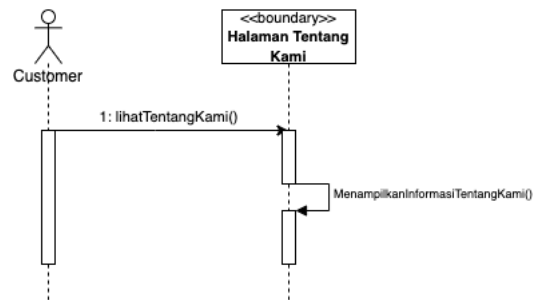
15) *Sequence Diagram Hapus URL Payment***Gambar 3. 28** *Sequence Diagram Hapus URL Payment (Admin)*

16) *Sequence* Diagram Pendaftaran Pelanggan Baru

Gambar 3. 29 *Sequence* Diagram Pendaftaran Pelanggan Baru (*Customer*)

17) *Sequence Diagram Cek Tagihan*

Gambar 3. 30 *Sequence Diagram Cek Tagihan (Customer)*

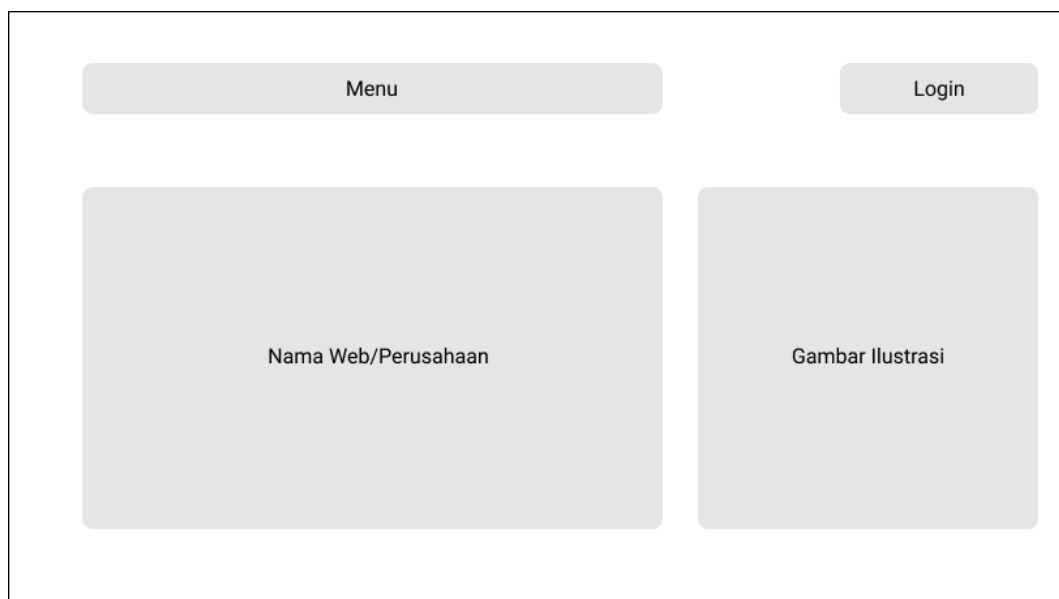
18) *Sequence Diagram Tentang Kami*

Gambar 3. 31 *Sequence Diagram Tentang Kami (Customer)*

3.2.2.2 Desain Tampilan Antarmuka (*User Interface*)

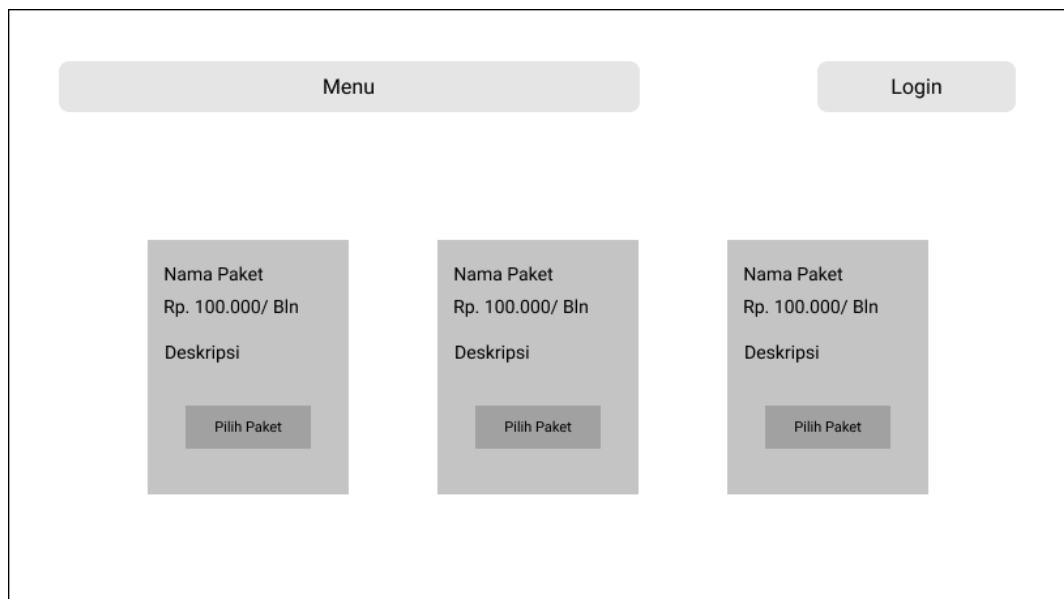
Setelah melakukan analisis diagram pada tahap kali ini akan dilakukan analisis desain *interface* yang akan dibangun. Antarmuka yang digunakan dalam *website* ini adalah *front end (customer)* dan *back-end dashboard management (admin)* sesuai dengan analisis kebutuhan sistem.

1. Desain Antarmuka Awal (*Landing Page*)



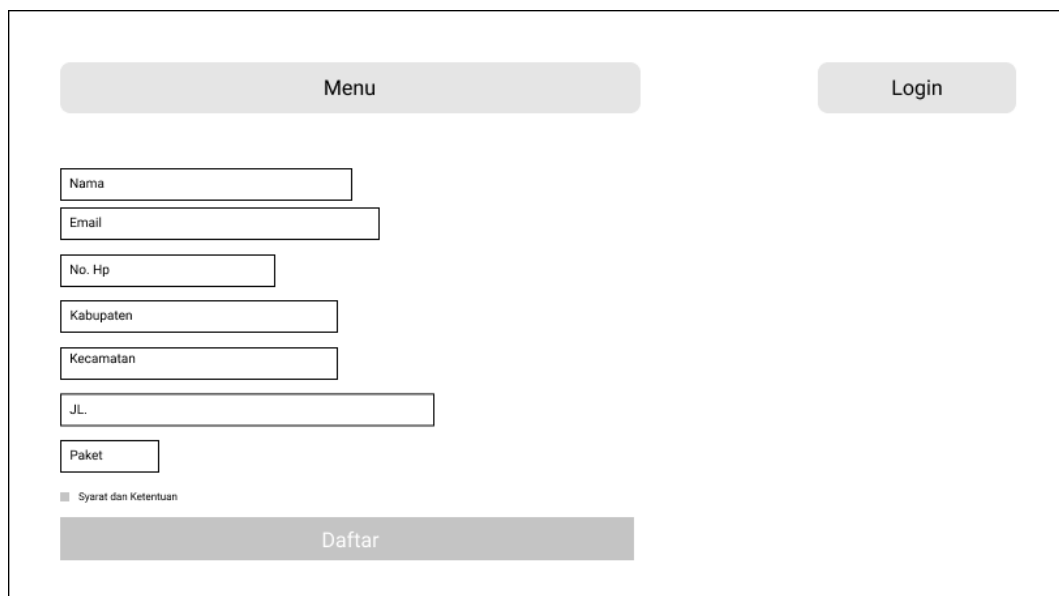
Gambar 3. 33 Desain Antarmuka Awal (*Landing Page*)

2. Desain Antarmuka Halaman Harga



Gambar 3. 34 Desain Antarmuka Halaman Harga

3. Desain Antarmuka Halaman Pendaftaran Pelanggan Baru



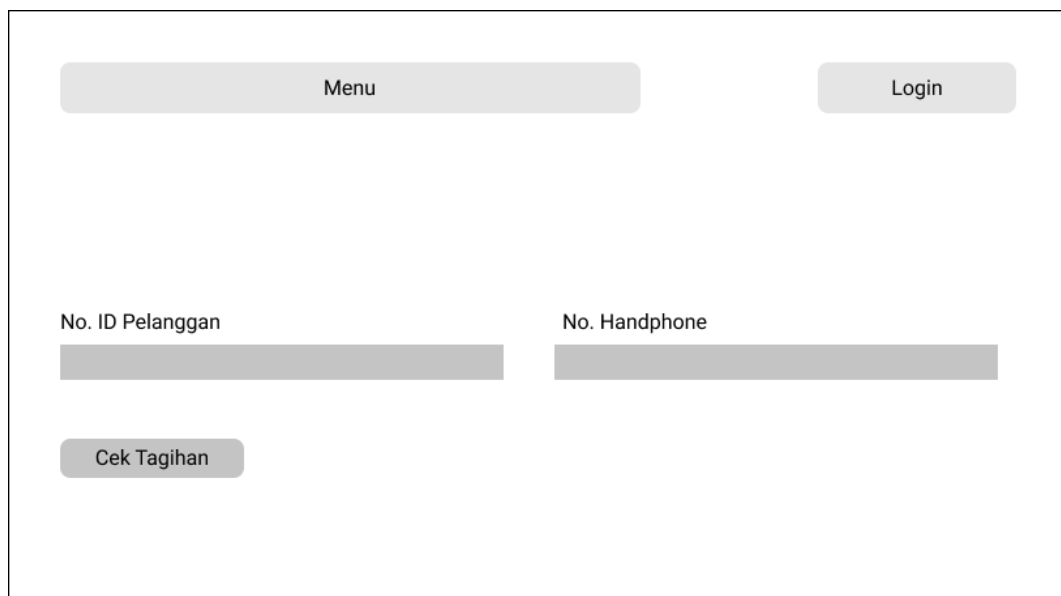
Gambar 3. 35 Desain Antarmuka Halaman Pendaftaran Pelanggan Baru

4. Desain Antarmuka Halaman Pendaftaran Berhasil



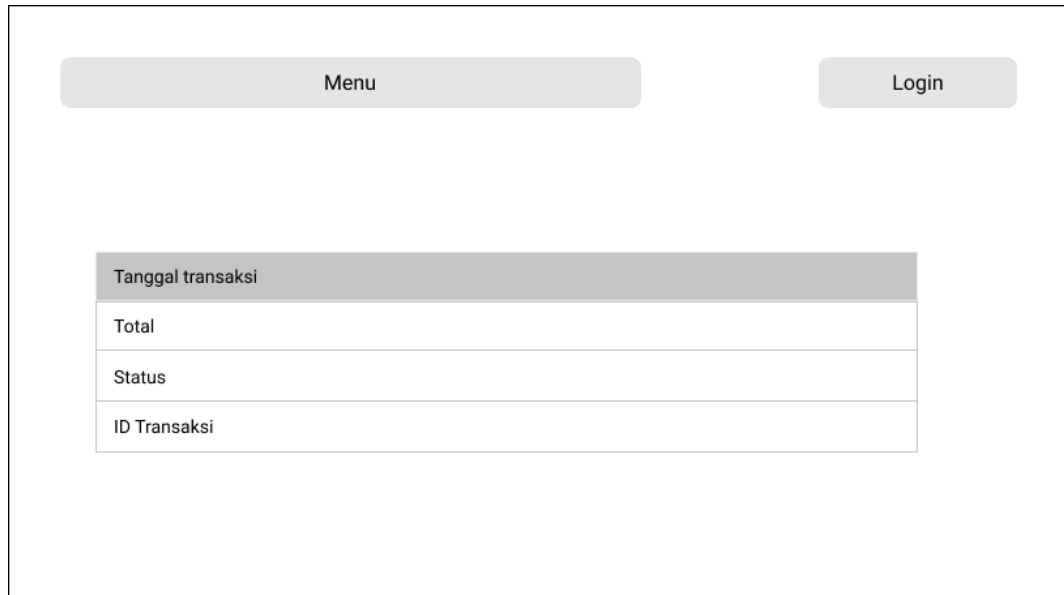
Gambar 3. 36 Desain Antarmuka Halaman Pendaftaran Berhasil

5. Desain Antarmuka Halaman Cek Tagihan



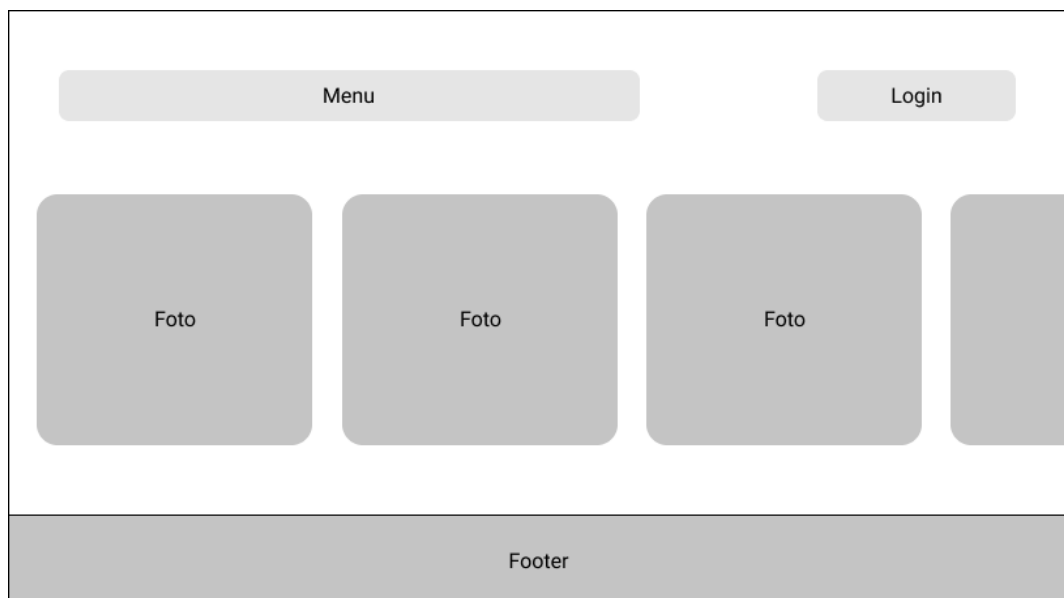
Gambar 3. 37 Desain Antarmuka Halaman Cek Tagihan

6. Desain Antarmuka Halaman Histori Pembayaran



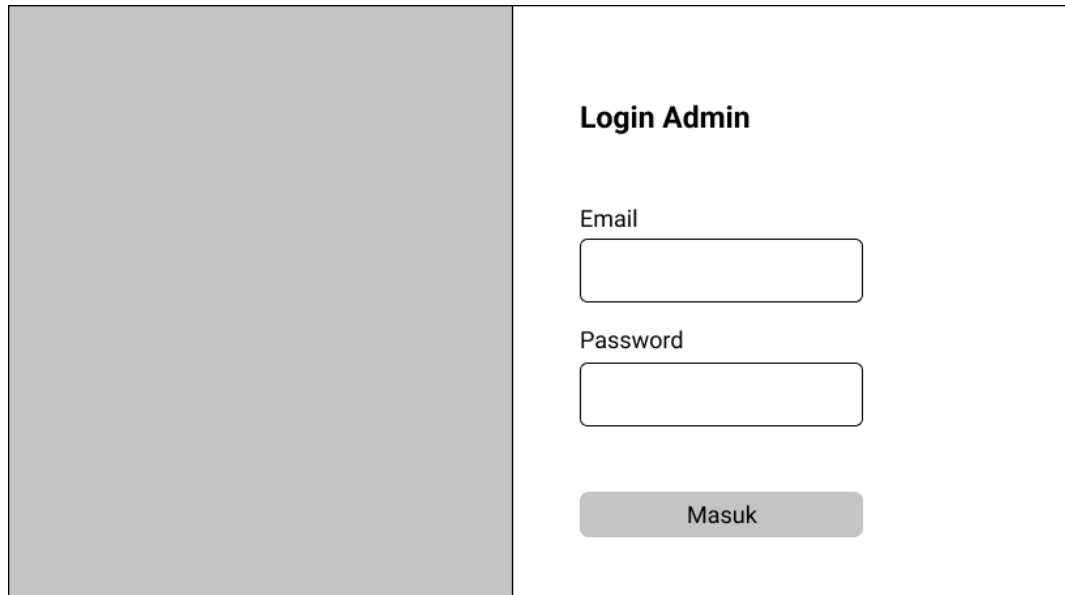
Gambar 3. 38 Desain Antarmuka Halaman Histori Pembayaran

7. Desain Antarmuka Halaman Tentang Kami



Gambar 3. 39 Desain Antarmuka Halaman Tentang Kami

8. Desain Antarmuka Halaman *Login (Admin)*

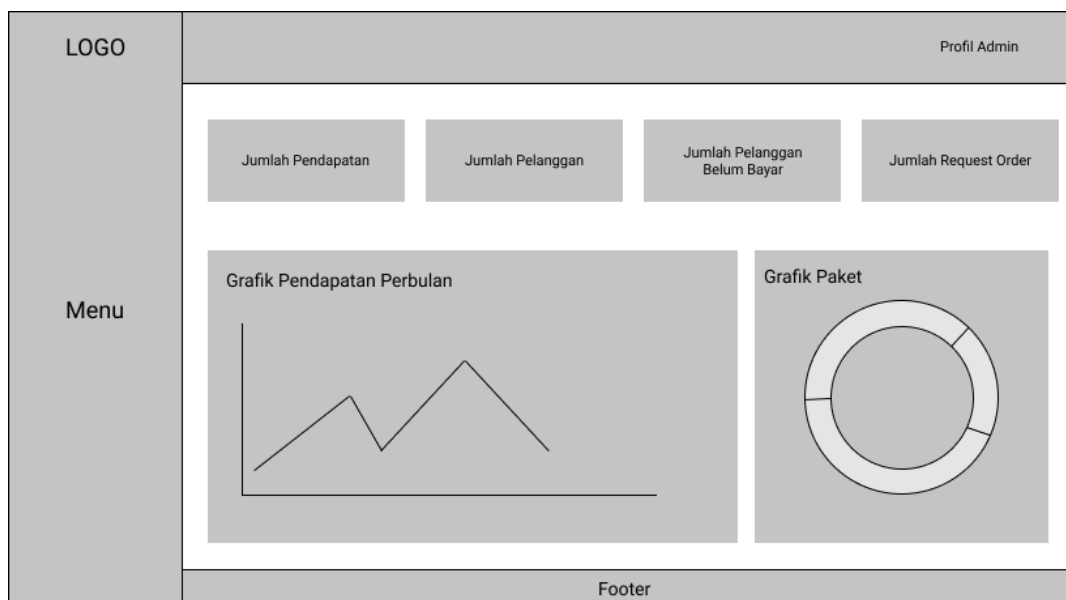


The image shows a login form for an administrator. It is divided into two main sections. The left section is a solid gray vertical bar. The right section is white and contains the following elements:

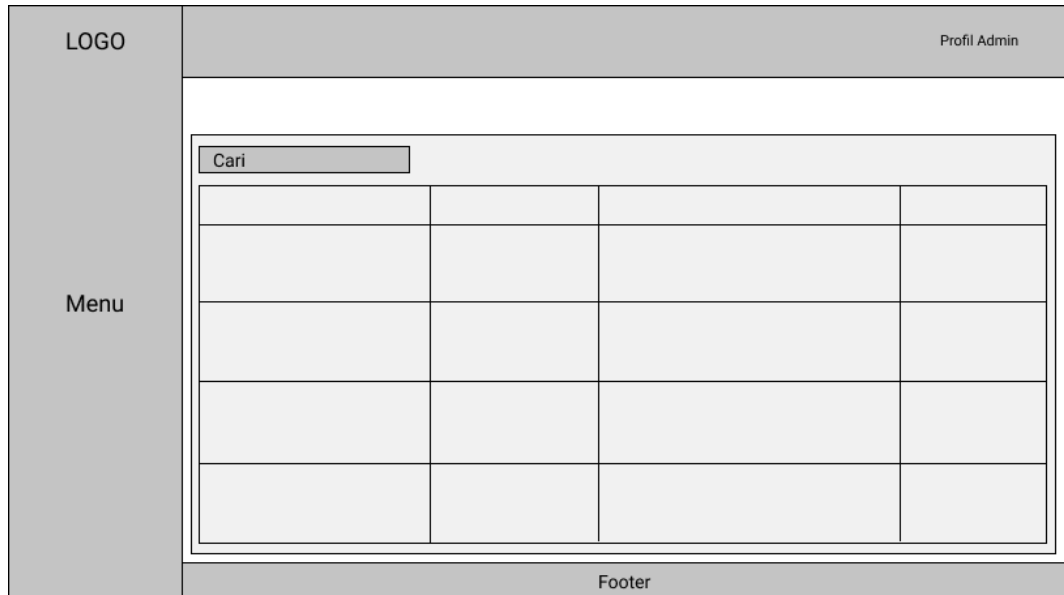
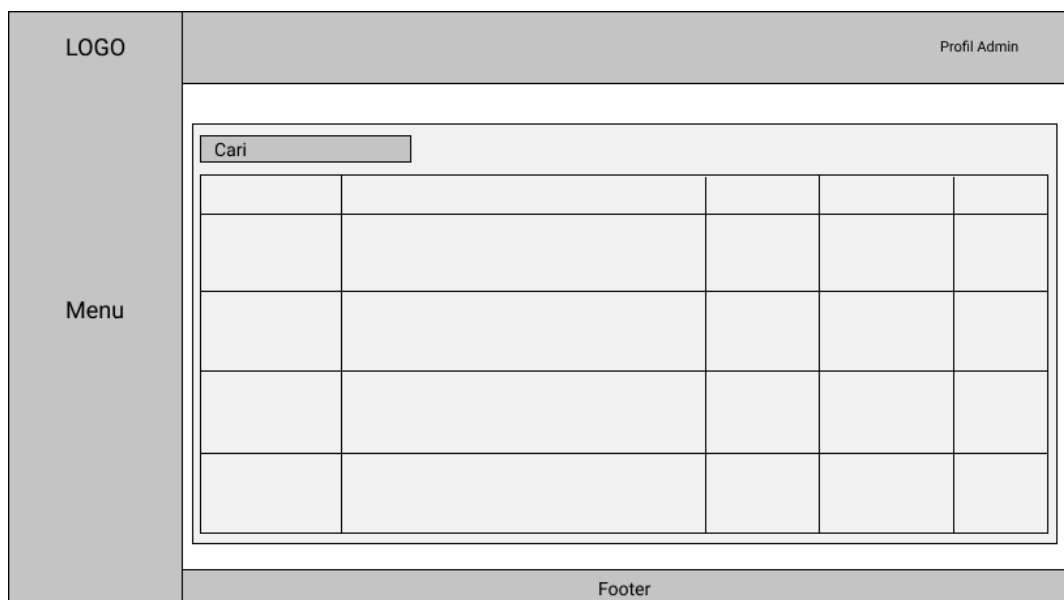
- Login Admin**: A bold heading at the top of the form.
- Email**: A text label above a rectangular input field.
- Password**: A text label above a second rectangular input field.
- Masuk**: A gray button with rounded corners located below the password field.

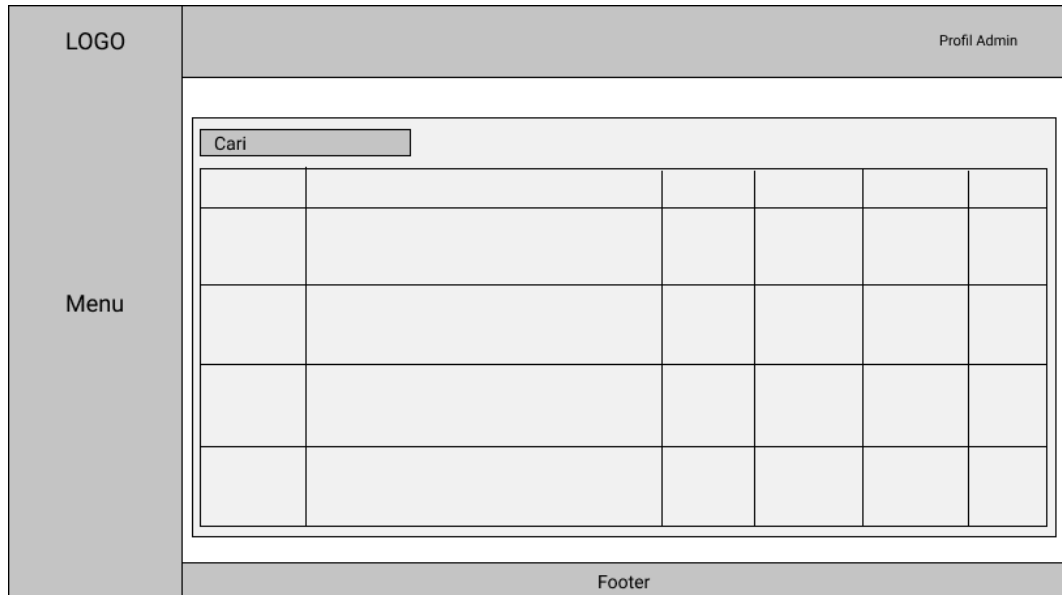
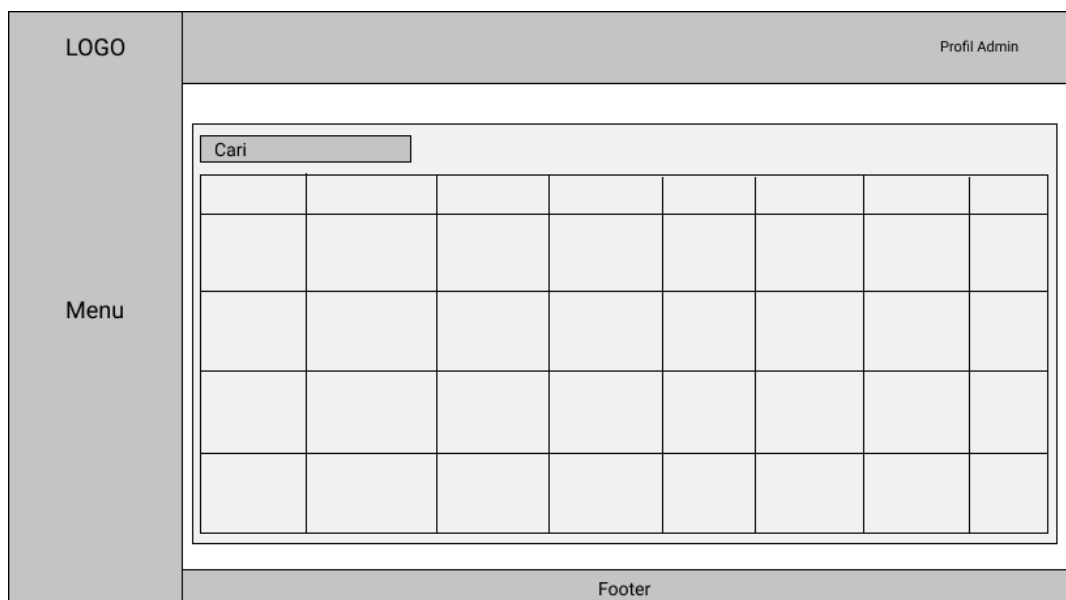
Gambar 3. 40 Desain Antarmuka Halaman *Login (Admin)*

9. Desain Antarmuka Halaman *Dashboard (Admin)*



Gambar 3. 41 Desain Antarmuka Halaman *Dashboard (Admin)*

10. Desain Antarmuka Halaman *Management Paket (Admin)***Gambar 3. 42** Desain Antarmuka Halaman *Management Paket (Admin)*11. Desain Antarmuka Halaman *Request Order (Admin)***Gambar 3. 43** Desain Antarmuka Halaman *Request Order (Admin)*

12. Desain Antarmuka Halaman *Management Customer (Admin)***Gambar 3. 44** Desain Antarmuka Halaman *Management Customer (Admin)*13. Desain Antarmuka Halaman *Management Transaksi (Admin)***Gambar 3. 45** Desain Antarmuka Halaman *Management Transaksi (Admin)*

3.2.3 Pengembangan (*Development*)

Setelah rancangan dibuat, akan dilakukan tahap pengembangan dengan *coding* atau penulisan kode program untuk merealisasikan rancangan yang telah dibuat menjadi sebuah sistem pengelolaan data pelanggan yang dapat digunakan.

3.2.4 Pengujian (*Testing*)

Tahap pengujian dilakukan setelah proses pengembangan sistem selesai dikerjakan, pengujian berfokus pada fungsional dari sistem, untuk memastikan fitur-fitur pada sistem pengelolaan data pelanggan berjalan sesuai yang diinginkan, sehingga *error* dapat diminimalisir.

3.2.5 Menyebarkan (*Deployment*)

Setelah sistem Berhasil diuji selanjutnya sistem akan disebarkan (*deployment*) ke sebuah server atau *hosting*. Kemudian dibuat juga nama domain dari sistem yang telah disebarkan, sehingga sistem pengelolaan data pelanggan bisa dibuka dan diakses secara *online* oleh semua orang melalui domain yang telah dibuat. Di bawah ini merupakan definisi *Server*, *Hosting* dan *Domain*.

1. *Server* adalah sebuah sistem komputer yang menyediakan layanan tertentu melalui satu jaringan komputer.
2. *Hosting* merupakan suatu tempat yang digunakan untuk menyimpan data yang ada pada sebuah website dan sebuah *hosting* disimpan di dalam sebuah *server*.
3. *Domain* merupakan nama yang telah ditentukan dan diatur oleh sebuah badan pengelola nama domain. Domain dapat berguna sebagai alamat dari suatu website.

3.2.6 Ulasan (*Review*)

Pada tahap ini dilakukan pembahasan mengenai sistem yang sudah berjalan, seperti apakah sistem sudah sesuai dengan yang dibutuhkan, fitur apa saja yang perlu dikembangkan atau fitur apa yang perlu ditambahkan pada sistem pengelolaan data RT/RW Net ini.

BAB IV

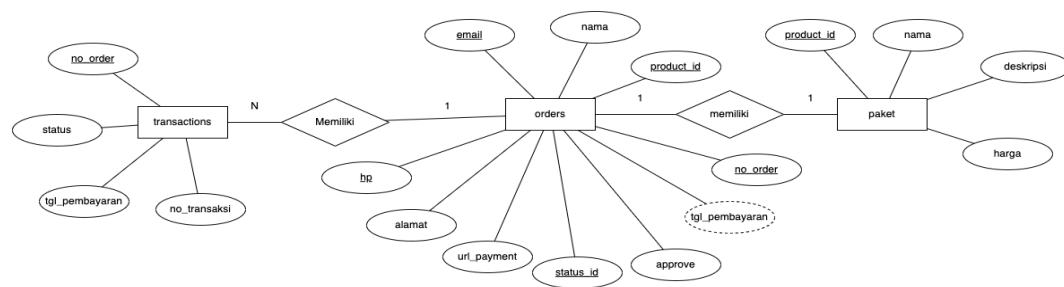
IMPLEMENTASI DAN UJI COBA

4.1 Pengembangan (*Development*)

Setelah proses perancangan selesai, langkah selanjutnya adalah tahap pengembangan yaitu merupakan tahap merealisasikan rancangan yang dibuat dengan menuliskan kode program agar dapat menjadi sistem informasi yang dapat digunakan sesuai dengan harapan. Pengembangan sistem menggunakan bahasa pemrograman PHP dan menggunakan *framework* laravel 8 dengan konsep MVC (*model, view, controller*) dan juga menggunakan bahasa pemrograman javascript untuk membuat tampilan sistem yang interaktif. Pengkodean sistem dilakukan di lokal server dengan menggunakan bantuan *software* XAMPP sebagai server lokal dan *software* visual studio code sebagai teks editor untuk menuliskan kode program. Untuk DBMS (*Database Management System*) yang digunakan adalah MySQL.

4.1.1 Entity Relationship Diagram (ERD)

Entity relationship diagram merupakan suatu model yang menjelaskan hubungan antar data dalam database berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *Entity relationship diagram* memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. ERD Sistem Pengelolaan Data Pelanggan RT/RW Net dapat dilihat pada gambar 4.1



Gambar 4. 1 Entity Relationship Diagram Sistem

Terdapat tiga tabel yang ada pada web sistem pengelolaan data pelanggan RT/RW Net diantaranya tabel *orders*, *products* dan *transactions*. Pada Tabel 3.2 menunjukkan penjelasan dari isi tabel *orders*

Tabel 4. 1 Tabel Order

Nama	Tipe Data	Ukuran	Key
id	bigint	20	<i>Primary-Key</i>
Product_id	bigint	20	<i>Foreign-Key</i>
nama	varchar	255	
hp	varchar	255	<i>Foreign-Key</i>
alamat	varchar	255	
approve	tinyint	1	
No_order	varchar	255	<i>Foreign-Key</i>
Tgl_pembayaran	date	-	
url_payment	varchar	255	

Tabel *Products* berisikan daftar paket yang tersedia atau dibuat oleh *admin*

Tabel 4. 2 Tabel *Product*

Nama	Tipe Data	Ukuran	Key
id	bigint	20	<i>Primary-Key</i>
nama	varchar	255	
harga	int	255	
deskripsi	varchar	255	

Tabel *Transactions* berisikan data dan informasi jumlah tagihan dan histori pembayaran pelanggan.

Tabel 4. 3 Tabel *Transactions*

Nama	Tipe Data	Ukuran	Key
id	bigint	20	<i>Primary-Key</i>
No_order_id	bigint	20	<i>Foreign-Key</i>
Tgl_pembayaran	date	-	
status	varchar	255	
no_transaksi	varchar	255	

4.1.2 *Physical Table*

Physical Table merupakan skema tabel fisik basis data yang terdapat pada DBMS (*Data Base Management System*). Berikut merupakan *Physical Table* sistem pengelolaan data pelanggan RT/RW Net.

db_rw_net orders	db_rw_net transactions	db_rw_net products
id : bigint(20) unsigned	id : bigint(20) unsigned	id : bigint(20) unsigned
product_id : bigint(20) unsigned	no_order_id : bigint(20) unsigned	nama : varchar(255)
nama : varchar(255)	tanggal_pembayaran : date	harga : int(11)
email : varchar(255)	total : int(11)	deskripsi : varchar(1000)
hp : varchar(255)	status : varchar(255)	created_at : timestamp
alamat : varchar(255)	id_transaksi : varchar(255)	updated_at : timestamp
created_at : timestamp	link_pembayaran : varchar(255)	warna : varchar(10)
updated_at : timestamp	created_at : timestamp	total : int(11)
its_true : tinyint(1)	updated_at : timestamp	
no_order : varchar(255)		
jatuh_tempo : date		
status : tinyint(1)		
url_payment : varchar(255)		

Gambar 4. 2 *Physical Table System*

4.1.3 Hasil Tampilan Antarmuka (*User Interface*)

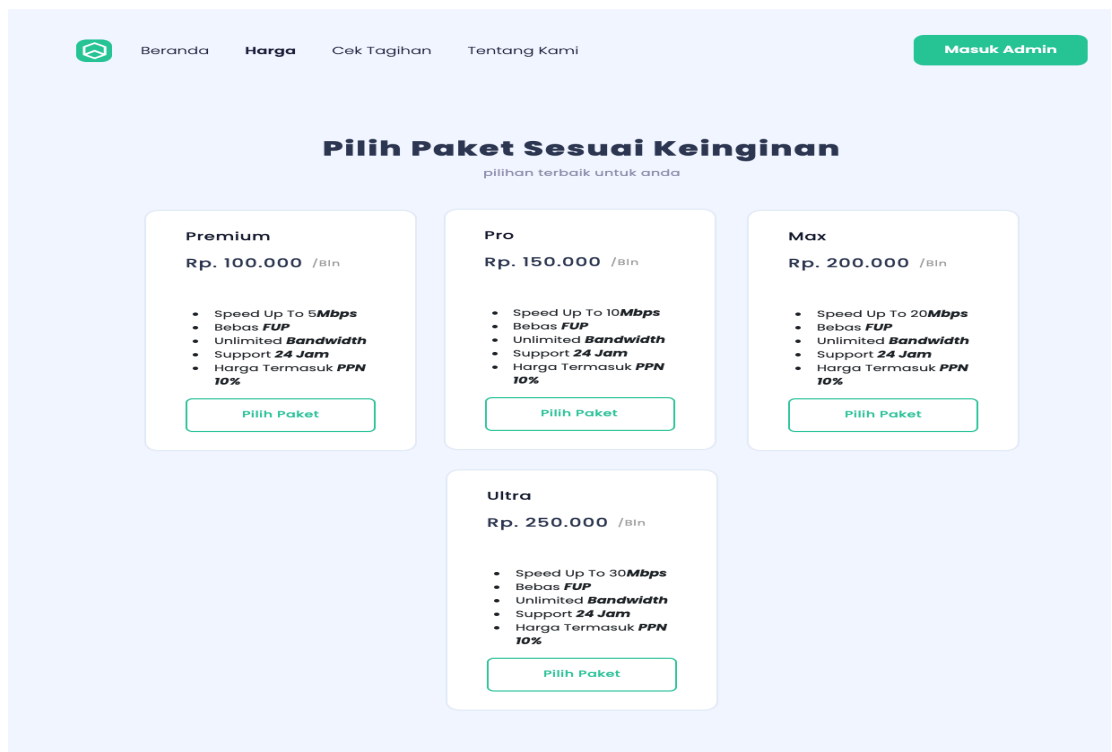
User Interface ditampilkan dalam bentuk antarmuka yang telah dirancang sedemikian rupa agar mudah digunakan. Antarmuka yang digunakan dalam *website* ini adalah *front end customer* dan *back-end dashboard management admin* sesuai dengan analisis kebutuhan sistem.

1. Antarmuka Awal (*Landing Page*)



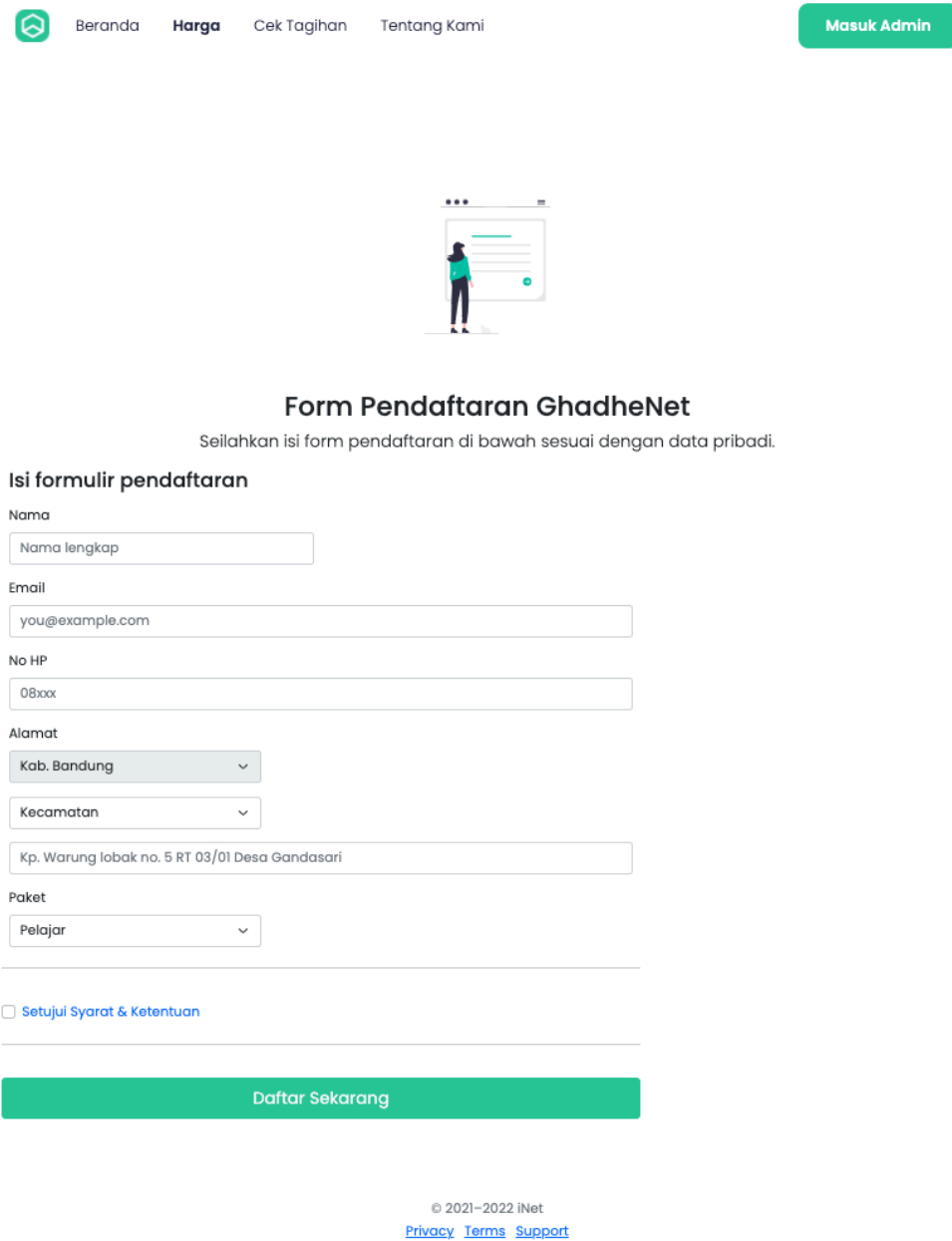
Gambar 4. 3 Antarmuka Awal (*Landing Page*)

2. Antarmuka Halaman Harga




Gambar 4. 4 Antarmuka Halaman Harga

3. Antarmuka Halaman Pendaftaran Pelanggan Baru



The screenshot shows a web application interface for a new customer registration. At the top, there is a navigation bar with a logo on the left and links for 'Beranda', 'Harga', 'Cek Tagihan', and 'Tentang Kami'. A green button labeled 'Masuk Admin' is positioned on the right. Below the navigation bar is a central illustration of a person standing next to a large screen displaying a form. The main heading is 'Form Pendaftaran GhadheNet', followed by the instruction 'Seilahkan isi form pendaftaran di bawah sesuai dengan data pribadi.' The form itself is titled 'Isi formulir pendaftaran' and contains several input fields: 'Nama' (with a placeholder 'Nama lengkap'), 'Email' (with a placeholder 'you@example.com'), 'No HP' (with a placeholder '08xxx'), 'Alamat' (with three sub-fields: a dropdown for 'Kab. Bandung', another dropdown for 'Kecamatan', and a text field for 'Kp. Warung lobak no. 5 RT 03/01 Desa Gandasari'), and 'Paket' (with a dropdown for 'Pelajar'). Below the form is a checkbox labeled 'Setujui Syarat & Ketentuan'. At the bottom of the form is a large green button labeled 'Daftar Sekarang'. The footer contains the copyright notice '© 2021-2022 iNet' and links for 'Privacy', 'Terms', and 'Support'.

Beranda **Harga** Cek Tagihan Tentang Kami Masuk Admin



Form Pendaftaran GhadheNet

Seilahkan isi form pendaftaran di bawah sesuai dengan data pribadi.

Isi formulir pendaftaran

Nama

Email

No HP

Alamat

Paket

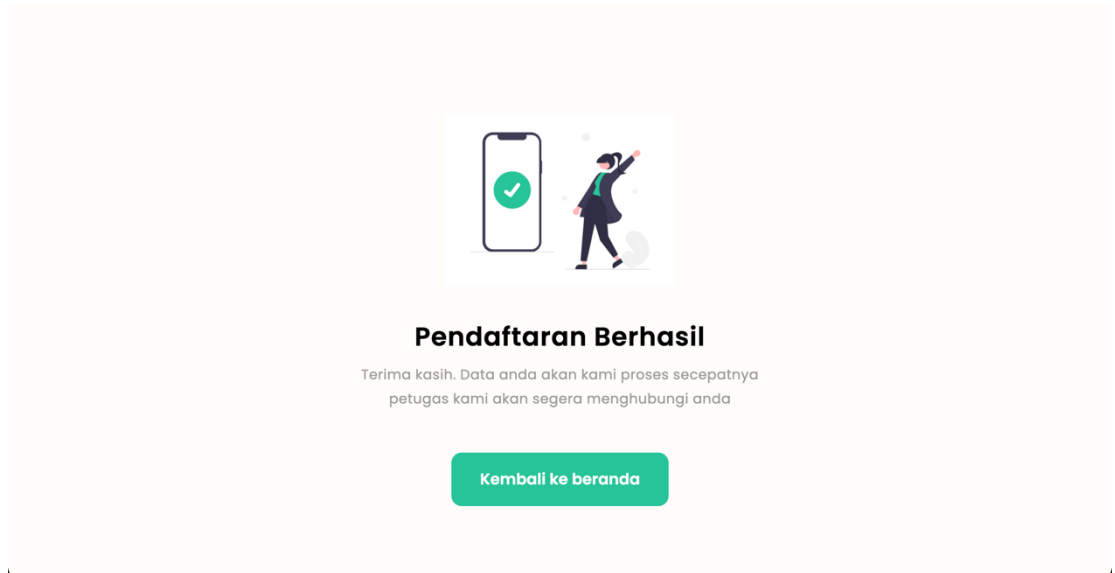
[Setujui Syarat & Ketentuan](#)

Daftar Sekarang

© 2021-2022 iNet
[Privacy](#) [Terms](#) [Support](#)

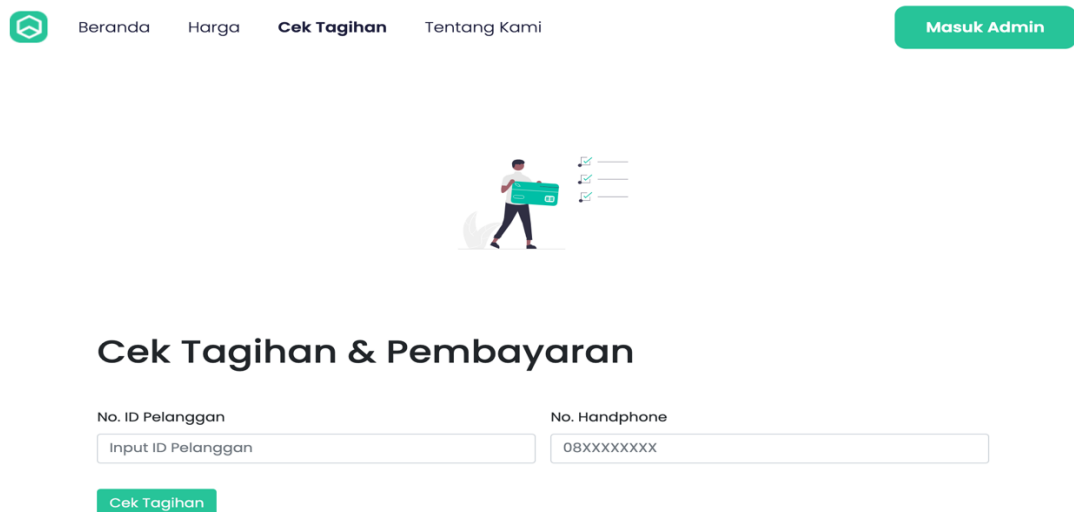
Gambar 4.5 Antarmuka Halaman Pendaftaran Pelanggan Baru

4. Antarmuka Halaman Pendaftaran Berhasil



Gambar 4. 6 Antarmuka Halaman Pendaftaran Berhasil

5. Antarmuka Halaman Cek Tagihan



Gambar 4. 7 Antarmuka Halaman Cek Tagihan

6. Antarmuka Halaman Histori Pembayaran

Beranda Harga Cek Tagihan Tentang Kami [Masuk Admin](#)

Histori Pembayaran

2022-01-10
Total: Rp. 200.000
Status: lunas
ID Transaksi: 46b4f859db94815b4076316969390eb6

2022-01-16
Total: Rp. 100.000
Status: lunas
ID Transaksi: 0db64eb8ef49a19fc2cbcbb76ec3c46




Gambar 4. 8 Antarmuka Halaman Histori Pembayaran

7. Antarmuka Halaman Tentang Kami

Beranda Harga Cek Tagihan Tentang Kami [Masuk Admin](#)

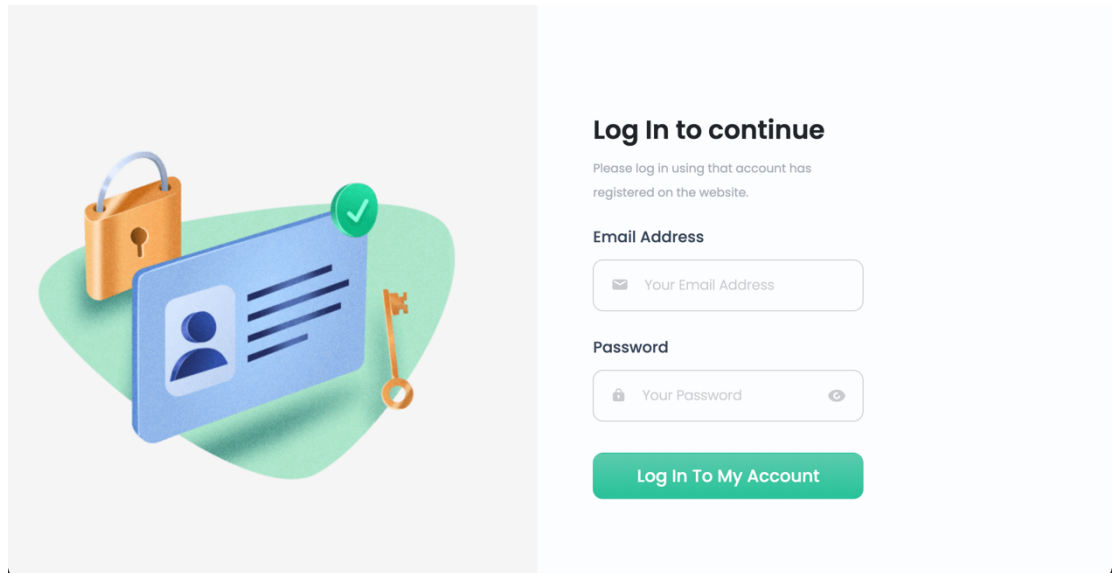
Datang untuk masyarakat

Berawal dari semakin mahalnya harga paket internet dan kurangnya ketersediaan jaringan internet di desa, kami datang untuk menghubungkan seluruh masyarakat desa



Gambar 4. 9 Antarmuka Halaman Tentang Kami

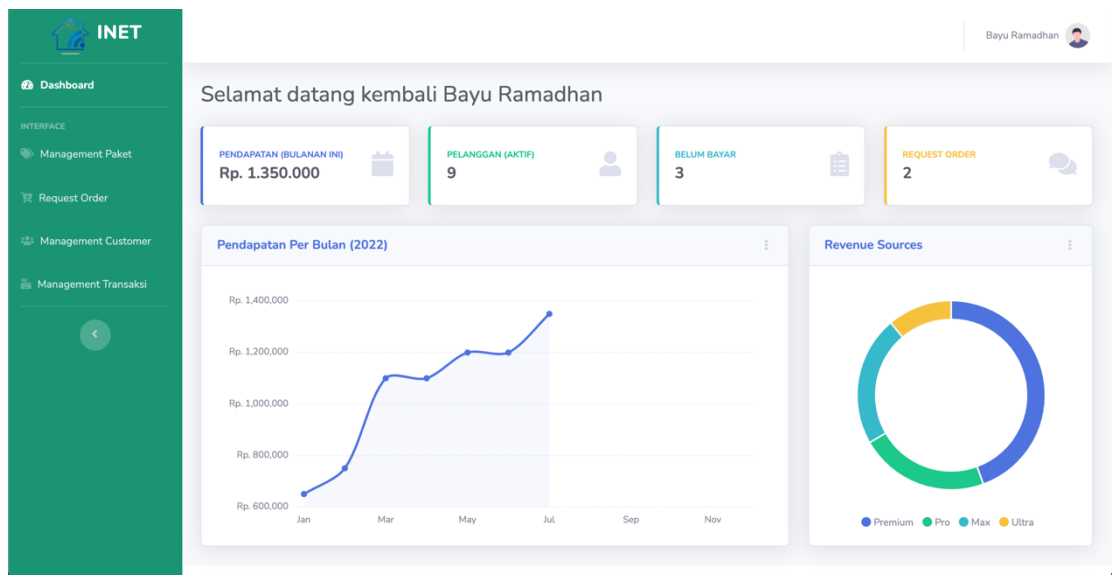
8. Antarmuka Halaman *Login (Admin)*



The screenshot shows an admin login page. On the left, there is a graphic illustration featuring a blue padlock, a blue document with a green checkmark, and a golden key. On the right, the text reads "Log In to continue" followed by "Please log in using that account has registered on the website." Below this, there are two input fields: "Email Address" with a placeholder "Your Email Address" and "Password" with a placeholder "Your Password" and an eye icon. A green button labeled "Log In To My Account" is positioned at the bottom.

Gambar 4. 10 Antarmuka Halaman *Login (Admin)*

9. Antarmuka Halaman *Dashboard (Admin)*



Gambar 4. 11 Antarmuka Halaman *Dashboard (Admin)*

10. Antarmuka Halaman *Management Paket (Admin)*

Daftar Paket Internet

Tambah Paket

Data Daftar Paket

Cari..

Nama	Harga	Deskripsi	Aksi
Pelajar	Rp. 165.000	<ul style="list-style-type: none"> Speed Up To 5Mbps Bebas FUP Unlimited Bandwidth Support 24 Jam Harga Termasuk PPN 10% 	<input checked="" type="checkbox"/> <input type="checkbox"/>
Ekonomis	Rp. 220.000	<ul style="list-style-type: none"> Speed Up To 7Mbps Bebas FUP Unlimited Bandwidth Support 24 Jam Harga Termasuk PPN 10% 	<input checked="" type="checkbox"/> <input type="checkbox"/>
Premium	Rp. 250.000	<ul style="list-style-type: none"> Speed Up To 7Mbps Bebas FUP Unlimited Bandwidth Support 24 Jam 	<input checked="" type="checkbox"/> <input type="checkbox"/>

Gambar 4. 12 Antarmuka Halaman *Management Paket (Admin)*

11. Antarmuka Halaman *Request Order (Admin)*

Daftar Order

Data Order Pelanggan

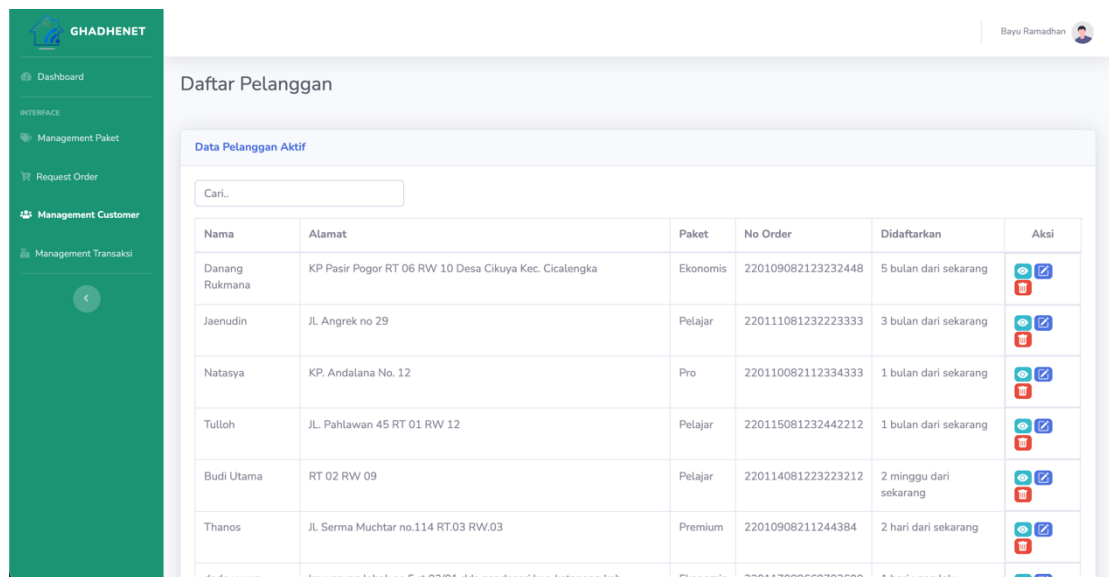
Cari..

Nama	Alamat	Paket	Didaftarkan	Aksi
Ayu Kartika	Gg. Idris no 23, RT 02 RW 09	Pro	3 hari yang lalu	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Nanang	Kec. Katapang Warung Lobak Desa Bungursari	Pro	1 hari yang lalu	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Copyright © Your Website 2021

Gambar 4. 13 Antarmuka Halaman *Request Order (Admin)*

12. Antarmuka Halaman *Management Customer (Admin)*

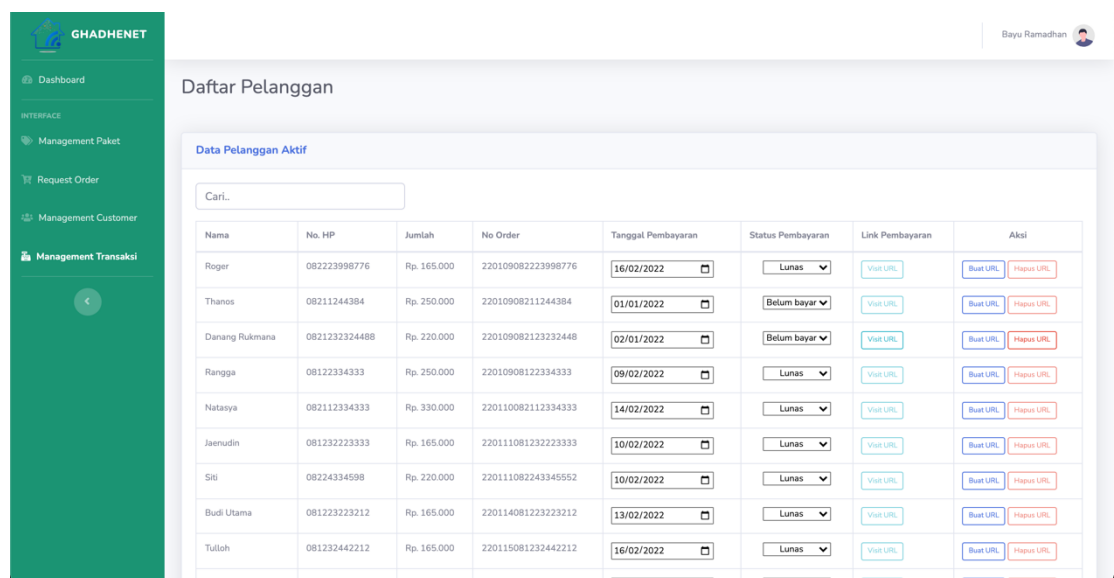


The screenshot shows the 'Daftar Pelanggan' (Customer List) page in the GHADHENET system. The page features a sidebar with navigation options: Dashboard, Management Paket, Request Order, Management Customer (selected), and Management Transaksi. The main content area displays a table of active customers with columns for Nama, Alamat, Paket, No Order, Didaftarkan, and Aksi. A search bar is located above the table.

Nama	Alamat	Paket	No Order	Didaftarkan	Aksi
Danang Rukmana	KP Pasir Pogor RT 06 RW 10 Desa Cikuya Kec. Cicalengka	Ekonomis	220109082123232448	5 bulan dari sekarang	[Icons]
Jaenudin	JL. Angrek no 29	Pelajar	220111081232223333	3 bulan dari sekarang	[Icons]
Natasya	KP. Andalana No. 12	Pro	220110082112334333	1 bulan dari sekarang	[Icons]
Tulloh	JL. Pahtawan 45 RT 01 RW 12	Pelajar	220115081232442212	1 bulan dari sekarang	[Icons]
Budi Utama	RT 02 RW 09	Pelajar	220114081223223212	2 minggu dari sekarang	[Icons]
Thanos	JL. Serma Muchtar no.114 RT.03 RW.03	Premium	22010908211244384	2 hari dari sekarang	[Icons]
derle usuan	kn warunan lohak no 5 rt 03/01 drc nandacari kec. katanann kah	Ekonomis	220117080660702600	1 hari vann lalu	[Icons]

Gambar 4. 14 Antarmuka Halaman *Management Customer (Admin)*

13. Antarmuka Halaman *Management Transaksi (Admin)*



The screenshot shows the 'Daftar Pelanggan' (Customer List) page in the GHADHENET system, specifically the 'Management Transaksi' view. The page features a sidebar with navigation options: Dashboard, Management Paket, Request Order, Management Customer, and Management Transaksi (selected). The main content area displays a table of transactions with columns for Nama, No. HP, Jumlah, No Order, Tanggal Pembayaran, Status Pembayaran, Link Pembayaran, and Aksi. A search bar is located above the table.

Nama	No. HP	Jumlah	No Order	Tanggal Pembayaran	Status Pembayaran	Link Pembayaran	Aksi
Roger	082223998776	Rp. 165.000	220109082223998776	16/02/2022	Lunas	Visit URL	Buat URL Hapus URL
Thanos	08211244384	Rp. 250.000	22010908211244384	01/01/2022	Belum bayar	Visit URL	Buat URL Hapus URL
Danang Rukmana	0821232324488	Rp. 220.000	220109082123232448	02/01/2022	Belum bayar	Visit URL	Buat URL Hapus URL
Rangga	08122334333	Rp. 250.000	22010908122334333	09/02/2022	Lunas	Visit URL	Buat URL Hapus URL
Natasya	082112334333	Rp. 330.000	220110082112334333	14/02/2022	Lunas	Visit URL	Buat URL Hapus URL
Jaenudin	081232223333	Rp. 165.000	220111081232223333	10/02/2022	Lunas	Visit URL	Buat URL Hapus URL
Siti	08224334598	Rp. 220.000	220111082243345552	10/02/2022	Lunas	Visit URL	Buat URL Hapus URL
Budi Utama	081223223212	Rp. 165.000	220114081223223212	13/02/2022	Lunas	Visit URL	Buat URL Hapus URL
Tulloh	081232442212	Rp. 165.000	220115081232442212	16/02/2022	Lunas	Visit URL	Buat URL Hapus URL

Gambar 4. 15 Antarmuka Halaman *Management Transaksi (Admin)*

4.2 Pengujian (*Testing*)

Setelah sistem selesai dikembangkan kemudian akan dilakukan pengujian sistem menggunakan metode *blackbox testing* untuk memastikan bahwa sistem berjalan dengan baik sebelum disebar ke server. Pengujian dibagi menjadi dua bagian yaitu pengujian *back-end* dan *front-end*. Pada pengujian *back-end* akan dilakukan pengujian pada setiap akses penggunanya, yaitu admin dan *customer*.

4.2.1 Pengujian Back-End (Admin)

Berikut merupakan hasil pengujian *back-end* pada hak akses pengguna *admin*.

Tabel 4. 4 Hasil Pengujian *Back-End (Admin)*

No	Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Memilih menu <i>dashboard</i>	Menekan menu <i>dashboard</i> pada <i>sidebar</i>	Sistem menampilkan informasi yang terdapat pada halaman <i>dashboard</i>	Sesuai Harapan	Valid
2.	Memilih menu <i>management</i> paket	Menekan menu <i>management</i> paket pada <i>sidebar</i>	Sistem menampilkan informasi yang terdapat pada halaman <i>management</i> paket	Sesuai Harapan	Valid
3.	Menambahkan paket	Menekan tombol tambah paket	Sistem menampilkan halaman buat paket baru	Sesuai Harapan	Valid
4.	Mengedit paket	Menekan tombol edit paket	Sistem menampilkan halaman edit paket sesuai paket yang dipilih	Sesuai Harapan	Valid

Tabel 4. 5 Hasil Pengujian *Back-End (Admin)* (Lanjutan)

No	Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
5.	Menghapus paket	Menekan tombol hapus paket	Sistem menampilkan <i>window prompt</i> untuk konfirmasi penghapusan dan paket terhapus jika menekan tombol setuju	Sesuai Harapan	Valid
6.	Memilih menu <i>request order</i>	Menekan menu <i>request order</i> pada sidebar	Sistem menampilkan informasi yang terdapat pada halaman <i>request order</i>	Sesuai Harapan	Valid
7.	Melihat <i>detail order</i>	Menekan tombol <i>detail order</i>	Sistem menampilkan informasi terkait order pelanggan yang dipilih	Sesuai Harapan	Valid
8.	Melakukan <i>approval order</i>	Menekan tombol <i>approve order</i>	Sistem menampilkan <i>window prompt</i> untuk konfirmasi persetujuan dan order akan diteruskan menjadi <i>customer</i> jika menekan tombol <i>approve</i>	Sesuai Harapan	Valid
9.	Menghapus order	Menekan tombol hapus order	Sistem menampilkan <i>window prompt</i> untuk konfirmasi penghapusan dan order terhapus jika menekan tombol setuju	Sesuai Harapan	Valid

Tabel 4. 6 Hasil Pengujian *Back-End (Admin)* (Lanjutan)

No	Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
10.	Memilih menu <i>management customer</i>	Menekan menu <i>management customer</i> pada <i>sidebar</i>	Sistem menampilkan informasi yang terdapat pada halaman <i>management customer</i>	Sesuai Harapan	Valid
11.	Melihat detail <i>customer</i>	Menekan tombol detail <i>customer</i>	Sistem menampilkan informasi terkait pelanggan yang dipilih	Sesuai Harapan	Valid
12.	Mengubah data <i>customer</i>	Menekan tombol ubah data <i>customer</i>	Sistem menampilkan halaman ubah data <i>customer</i> sesuai <i>customer</i> yang dipilih	Sesuai Harapan	Valid
13.	Menghapus <i>customer</i>	Menekan tombol hapus <i>customer</i>	Sistem menampilkan <i>window prompt</i> untuk konfirmasi penghapusan dan <i>customer</i> terhapus jika menekan tombol setuju	Sesuai Harapan	Valid
14.	Memilih menu <i>management transaksi</i>	Menekan menu <i>management transaksi</i> pada <i>sidebar</i>	Sistem menampilkan informasi yang terdapat pada halaman <i>management transaksi</i>	Sesuai Harapan	Valid
15.	Mengubah status pembayaran <i>customer</i>	Menekan tombol pilihan belum bayar/lunas	Sistem melakukan perubahan terhadap status pembayaran <i>customer</i>	Sesuai Harapan	Valid

Tabel 4. 7 Hasil Pengujian *Back-End (Admin)* (Lanjutan)

No	Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
16.	Melakukan pengecekan URL Pembayaran	Menekan tombol <i>visit URL</i>	Sistem mengalihkan ke halaman pembayaran	Sesuai Harapan	Valid
17.	Menambahkan URL pembayaran	Menekan tombol buat URL pembayaran	Sistem menampilkan halaman masukan URL	Sesuai Harapan	Valid
18.	Menghapus URL pembayaran	Menekan tombol hapus URL pembayaran	Sistem menampilkan <i>alert</i> konfirmasi penghapusan URL	Sesuai Harapan	Valid

4.2.2 Pengujian *Front-End*

Berikut merupakan hasil pengujian *front-end* atau halaman depan sistem.

Tabel 4. 8 Hasil Pengujian *Front-End*

No	Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Membuka sistem	Membuka halaman utama web	Sistem terbuka dan menampilkan halaman <i>home</i>	Sesuai Harapan	Valid
2.	Memilih menu harga	Menekan menu harga pada <i>navbar</i>	Sistem mengalihkan ke halaman daftar harga paket	Sesuai Harapan	Valid
3.	Memilih paket	Menekan tombol pilih paket	Sistem mengalihkan ke halaman <i>form</i> pendaftaran sesuai paket yang dipilih	Sesuai Harapan	Valid
4.	Memilih menu cek tagihan	Menekan menu cek tagihan pada <i>navbar</i>	Sistem mengalihkan ke halaman cek tagihan	Sesuai Harapan	Valid

Tabel 4. 9 Hasil Pengujian *Front-End* (Lanjutan)

No	Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
5.	Memilih menu tentang kami	Menekan menu tentang kami pada <i>navbar</i>	Sistem mengalihkan ke halaman tentang kami	Sesuai Harapan	Valid

4.3 Penyebaran (*Deployment*)

Tahap penyebaran merupakan tahap menyebarkan (*deployment*) sistem yang telah dikembangkan dan diuji ke sebuah server (*hosting*) dan kemudian ditentukan juga nama domain dari sistem, sehingga sistem pengelolaan data pelanggan RT/RW-Net dapat diakses oleh semua orang melalui nama domain yang telah ditentukan yaitu *ghadhe.my.id*. Peneliti menggunakan layanan jasa penyedia *hosting* dan domain dari rumah web rumahweb.com dengan spesifikasi server sebagai berikut:

- *Unlimited storage*
- *Unlimited traffic*
- *Unlimited email*
- *Unlimited domains*
- *Free weebly sitebuilder*
- *Free SSL*
- *Medium cloud resource*
- *Turbo booster*
- *Security 360*

4.4 Ulasan (*Review*)

Tahap terakhir merupakan tahap memberikan ulasan (*review*) terhadap sistem yang telah dibuat. Pada tahap ini peneliti atau pengembang sistem berdiskusi dengan pihak terkait untuk membahas sistem yang telah berjalan, apakah sistem sudah memenuhi kebutuhan yang telah ditentukan di awal, atau perlu adanya penambahan fitur lain. Dari hasil diskusi yang dilakukan menghasilkan rumusan adanya penambahan fitur pada sistem, diantaranya fitur halaman tentang kami, histori pembayaran dan penambahan fitur status pembayaran. Sehingga pada akhirnya sistem pengelolaan data pelanggan RT/RW Net ini dapat menjadi lebih baik.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai sistem pengelolaan data pelanggan RT/RW Net berbasis *website* dengan menggunakan pendekatan metode *agile*, maka dapat disimpulkan sebagai berikut:

1. Dengan menerapkan sistem pengelolaan data pelanggan pada usaha RT/RW Net dapat mempermudah kegiatan pengelolaan data pelanggan layanan konektivitas internet.
2. Mempermudah proses transaksi antara pelaku usaha dan pelanggan, karena dapat dilakukan di mana saja dan kapan saja secara *online*.
3. Fitur Ghadenet seperti Cek Tagihan, Pendaftaran, *Management* Paket, *Management Order*, *Management Customer* dan *Management* Transaksi yang terdapat pada sistem pengelolaan data pelanggan RT/RW Net dapat menunjang aktivitas bisnis RT/RW Net secara efektif dan fleksibel.

5.2. Saran

Setelah dilakukan penelitian mengenai sistem pengelolaan data pelanggan RT/RW Net berbasis *website*, peneliti bermaksud memberikan saran-saran berdasarkan kesimpulan dan hasil penelitian yang mungkin akan berguna untuk penelitian selanjutnya. Peneliti juga tidak menutup saran-saran dari pengguna sistem, karena pengembangan sistem menggunakan metode *agile* bertujuan untuk

menyesuaikan sistem dengan kebutuhan pengguna. Adapun saran-saran dari peneliti untuk saat ini adalah sebagai berikut:

1. Menambahkan filter berdasarkan tanggal pembayaran untuk mengatur tampilan data dapat mempermudah *admin* dalam melakukan manipulasi data
2. Menambahkan fitur *reminder via* Whatsapp atau Telegram untuk mengingatkan pelanggan mengenai tanggal pembayaran.
3. Menambahkan fitur *report* untuk *customer* bila mana ada kendala dalam menggunakan layanan internet
4. Mengintegrasikan *payment gateway* melalui API agar *monitoring* pembayaran dapat dilakukan secara langsung di dalam web.
5. Menyesuaikan desain antarmuka agar menjadi lebih baik lagi.

DAFTAR PUSTAKA

- A.S, Rosa dan Shalahuddin, M. *Rekayasa Perangkat Lunak Terstruktur dan Beroientasi Objek*. Bandung: Imformatika, 2016.
- Abdulloh, R. (2018). *7 in 1 Pemrograman Web untuk Pemula*. PT Elex Media Komputindo.
- Al-gawwam, S., & Benaissa, M. (2018). *Robust eye blink detection based on eye landmarks and Savitzky–Golay filtering*. *Information*, 9(4), 93.
- Andriana, M., Fatoni, F., & Ibadi, T. (2019, January). *Desain Jaringan WLAN RT/RW Dengan Router Mikrotik Pada Komplek Srimas Plaju*. In Bina Darma Conference on Computer Science (BDCCS) (Vol. 1, No. 1, pp. 50-57).
- Anwar, K., Kurniawan, L. D., Rahman, M. I., & Ani, N. (2020). *Aplikasi Marketplace Penyewaan Lapangan Olahraga Dari Berbagai Cabang Dengan Metode Agile Development*. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 9(2), 264–274.
- Aristania, N., & Wardati, I. U. (2017). *Pembuatan Sistem Informasi Pengelolaan Persediaan Barang Dan Data Proyek Gypsum Pada Utomo Gypsum*. *Journal Speed – Sentra Penelitian Engineering Dan Edukasi*, 9(1), 62–68.
- Ayu, F., & Permatasari, N. (2018). *Perancangan Sistem Informasi Pengolahan Data PKL (Praktek Kerja Lapangan) Di Divisi Humas Pada PT Pegadaian*. *Jurnal Intra Tech*, 2(2), 12-26.

- Azwanti, N. (2017). Sistem Informasi Penjualan Tas Berbasis Web Dengan Pemodelan UML. *KLIK - KUMPULAN JURNAL ILMU KOMPUTER*, 4(1), 14. <https://doi.org/10.20527/klik.v4i1.80>
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 3(2), 206-210.
- Elgamar. (2020). *Buku Ajar Konsep Dasar Pemrograman Website Dengan Php* (N. Pangesti (ed.)). CV. Multimedia Edukasi.
- Haryana, K. S. (2019). *Penerapan Agile Development Methods Dengan Framework Scrum Pada Perancangan Perangkat Lunak Kehadiran Rapat Umum Berbasis Qr-Code. Jurnal Computech & Bisnis*, 13(2), 70–79.
- Hendini, A. (2016). Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak). *Jurnal Khatulistiwa Informatika*, IV(DESEMBER), 107–116.
- Kurniawan, T. B. (2020). *Perancangan sistem aplikasi pemesanan makanan dan minuman pada cafetaria no coffe di tanjung balai karimun menggunakan bahasa pemograman php dan mysql. Jurnal tikar*, 1(2), 192-206.
- Kurniawati, I. D., & Kusumawardhani, A. (2017). *Implementasi Algoritma Canny dalam Pengenalan Wajah menggunakan Antarmuka GUI MATLAB. Institution of Engineering and Technology*, 1-3.

Laravel The PHP Framework for Web Artisans (2022), *About Laravel*. Diakses pada 22 Januari 2022, dari <https://laravel.com>.

Mubarak, A. (2019). *Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek*. *Jurnal Informatika Dan Komputer*, 02(1), 19–25.

N. Nurajizah, “Penerapan Rapid Application Development Pada Sistem Informasi Perpustakaan Sekolah Gema Nuran I Bekasi,” *Jurnal Sistem Informasi STMIK Antar Bangsa*, vol. V, no. 1, pp. 1-6, 2016

Nurzaman, F. (2020). *Pengembangan sistem otomatisasi tagihan menggunakan metode agile s oftware I development*. *Ikra-ith informatika: Jurnal Komputer dan Informatika*, 4(1), 46-57.

Oo, K. Z. (2019). *Design and implementation of electronic payment gateway for secure online payment system*. *Int. J. Trend Sci. Res. Dev*, 3, 1329-1334.

Pamungkas, D. W. L., & Rochimah, S. (2019). *Pengujian Aplikasi Web-Tinjauan Pustaka Sistematis*. *Jurnal IPTEK*, 23(1), 17–24.

Puspitasari, T. M. M., & Maulina, D. (2019). *Implementasi Payment Gateway Menggunakan Midtrans Pada Marketplace Travnesia. Com. Mobile and Forensics*, 1(1), 22-29.

Raharjana, I. K. (2017). *Pengembangan Sistem Informasi Menggunakan Metodologi Agile*. Deepublish.

- Salamah, U., & Khasanah, F. (2017). *Pengujian Sistem Informasi Penjualan Undangan Pernikahan Online Berbasis Web Menggunakan Black Box Testing*. *Information Management for Educators and Professionals*, 2(1), 35–46.
- Silalahi, U., & Atif, N. F. (2015). Metode penelitian sosial kuantitatif.
- Stellman, A., & Greene, J. (2014). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban* (1st ed.). O'Reilly Media.
- Supriyatna, A. (2015). *Perpustakaan Dengan Menggunakan Pieces Framework*. *Pilar Nusa Mandiri*, XI(1), 43–52.
- Sutabri Tata, 2016, *Sistem Informasi Manajemen*, Andi Offset, Yogyakarta
- Taufik, A., & Ermawati. (2017). *Perancangan Sistem Informasi Pemesanan Pentas Seni Berbasis Web Pada Sanggar Seni Getar Pakuan Bogor*. *IJSE - Indonesian Journal on Software Engineering*, 3(2), 1–7.
<http://ejournal.bsi.ac.id/ejurnal/index.php/ijse/article/view/2812/1836>
- Tri Mardiani, G. (2017). Pertemuan 11-Class Diagram & Sequence Diagram.
- Wicaksono, B. D., & Anggraeni, S. (2021). *Perancangan Website Sistem Informasi Transaksi Tagihan Layanan Purna Jual Properti Pada Pollux Properti Indonesia*. *TMJ (Technomedia Journal)* Vol, 5, 132.

LAMPIRAN

Lampiran Source Code

1. Main Code

main.blade.php

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="csrf-token" content="{{ csrf_token() }}" />
    <!-- Bootstrap CSS -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJB
      yhZMI3AhiU" crossorigin="anonymous">
    <link rel="stylesheet" href="css/style.css">
    <link href="css/form-validation.css" rel="stylesheet">
    <style>
      .wrapper {
        margin: 0px !important;
        padding: 0px !important;
        /* width: 100vw; */
        max-height: 100vw;
        max-width: 100vw;
        border: 1px solid #fff;
        display: flex;
        overflow: auto;
```

```
        overflow-y: hidden;
    }
    .wrapper::-webkit-scrollbar {
        width: 0px;
    }
    .wrapper .item {
        min-width: 110px;
        width: 110px;
        height: 110px;
        line-height: 110px;
        text-align: center;
    }
    .acent {
        border-color: #D4BCA9;
        color: #D4BCA9;
    }
    .acent:hover {
        background-color: #D4BCA9;
        color: white;
    }
    .card {
        box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
        transition: transform .4s;
        width: 350px;
        height: 370px;
        border-radius: 20px;
    }
    .card:hover {
        box-shadow: 0 16px 32px 0 rgba(0,0,0,0.2);
        transform: scale(1.01)
    }
}
```

```

</style>
<title>iNet | {{ $title }}</title>
</head>
<body>
  @include('layouts.partials.nav')
  <div class="container p-2">
    @yield('container')
  </div>
  <!-- Optional JavaScript; choose one of the two! -->
  <!-- Option 1: Bootstrap Bundle with Popper -->
  <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.
    min.js" integrity="sha384-
    /bQdsTh/da6pkI1MST/rWKFNjaCP5gBSY4sEBT38Q/9RBh9AH40zEOg7
    Hlq2THRZ" crossorigin="anonymous"></script>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
    icons@1.5.0/font/bootstrap-icons.css">
  <!-- Option 2: Separate Popper and Bootstrap JS -->
  <!--
  <script
    src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.m
    in.js" integrity="sha384-
    W8fXfP3gkOKtndU4JGtKdVXbO53Wy8SZCQHczT5FMiiqmQfUpWbYd
    Til/SxwZgAN" crossorigin="anonymous"></script>
  <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.min.js"
    integrity="sha384-
    skAcpIdS7UcVUC05LJ9Dxay8AXcDYfBJqt1CJ85S/CFujBsIzCIv+19liuY
    LaMQ/" crossorigin="anonymous"></script>
  -->
</body>

```

```
</html>
```

2. *Dashboard Code*

Index.blade.php

```
@extends('dashboard.layouts.main')
@section('container')
<!-- Begin Page Content -->
<div class="container-fluid">
  <!-- Page Heading -->
  <div class="d-sm-flex align-items-center justify-content-between mb-4">
    <h1 class="h3 mb-0 text-gray-800">Selamat datang kembali {{ auth()-
  >user()->name }}</h1>
    {{-- <a href="#" class="d-none d-sm-inline-block btn btn-sm btn-primary
  shadow-sm"><i
      class="fas fa-download fa-sm text-white-50"></i> Generate Report</a> -
  -}}
  </div>
  <!-- Content Row -->
  <div class="row">
    <!-- Earnings (Monthly) Card Example -->
    <div class="col-xl-3 col-md-6 mb-4">
      <div class="card border-left-primary shadow h-100 py-2">
        <div class="card-body">
          <div class="row no-gutters align-items-center">
            <div class="col mr-2">
              <div class="text-xs font-weight-bold text-primary text-
  uppercase mb-1 fs-6">
                Pendapatan (Bulanan Ini)</div>
              <div class="h5 mb-0 font-weight-bold text-gray-800">
                @if (count($order))
                @currency($jumlah->jumlah_harga)
```



```

        @else
        <p>Rp. 0</p>
        @endif
    </div>
</div>
<div class="col-auto">
    <i class="fas fa-calendar fa-2x text-gray-300"></i>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Earnings (Monthly) Card Example -->
<div class="col-xl-3 col-md-6 mb-4">
    <div class="card border-left-success shadow h-100 py-2">
        <div class="card-body">
            <div class="row no-gutters align-items-center">
                <div class="col mr-2">
                    <div class="text-xs font-weight-bold text-success text-uppercase
mb-1">
                        Pelanggan (Aktif)</div>
                    <div class="h5 mb-0 font-weight-bold text-gray-800">{{
$customerCount }}</div>
                </div>
                <div class="col-auto">
                    <i class="fas fa-user fa-2x text-gray-300"></i>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>

```

```

<!-- Earnings (Monthly) Card Example -->
<div class="col-xl-3 col-md-6 mb-4">
  <div class="card border-left-info shadow h-100 py-2">
    <div class="card-body">
      <div class="row no-gutters align-items-center">
        <div class="col mr-2">
          <div class="text-xs font-weight-bold text-info text-uppercase
mb-1">Belum Bayar
        </div>
        <div class="row no-gutters align-items-center">
          <div class="col-auto">
            <div class="h5 mb-0 mr-3 font-weight-bold text-gray-
800">{{ count($status) }}</div>
          </div>
          <div class="col">
            {{-- <div class="progress progress-sm mr-2">
              <div class="progress-bar bg-info" role="progressbar"
                style="width: %" aria-valuenow="50" aria-
valuemin="0"
                aria-valuemax="100"></div>
            </div> --}}
          </div>
        </div>
      </div>
      <div class="col-auto">
        <i class="fas fa-clipboard-list fa-2x text-gray-300"></i>
      </div>
    </div>
  </div>
</div>
</div>

```

```

<!-- Pending Requests Card Example -->
<div class="col-xl-3 col-md-6 mb-4">
  <div class="card border-left-warning shadow h-100 py-2">
    <div class="card-body">
      <div class="row no-gutters align-items-center">
        <div class="col mr-2">
          <div class="text-xs font-weight-bold text-warning text-
uppercase mb-1">
            Request Order</div>
          <div class="h5 mb-0 font-weight-bold text-gray-800">{{
$reqOrder }}</div>
        </div>
        <div class="col-auto">
          <i class="fas fa-comments fa-2x text-gray-300"></i>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
<!-- Content Row -->
<div class="row">
  <!-- Area Chart -->
  <div class="col-xl-8 col-lg-7">
    <div class="card shadow mb-4">
      <!-- Card Header - Dropdown -->
      <div class="card-header py-3 d-flex flex-row align-items-center justify-
content-between">
        <h6 class="m-0 font-weight-bold text-primary">Pendapatan Per
Bulan ({{ date('Y') }})</h6>
        <div class="dropdown no-arrow">

```

```

        <a class="dropdown-toggle" href="#" role="button"
id="dropdownMenuLink" data-toggle="dropdown"
        aria-haspopup="true" aria-expanded="false">
        <i class="fas fa-ellipsis-v fa-sm fa-fw text-gray-400"></i>
    </a>
    <div class="dropdown-menu dropdown-menu-right shadow
animated--fade-in"
        aria-labelledby="dropdownMenuLink">
        <div class="dropdown-header">Dropdown Header:</div>
        <a class="dropdown-item" href="#">Action</a>
        <a class="dropdown-item" href="#">Another action</a>
        <div class="dropdown-divider"></div>
        <a class="dropdown-item" href="#">Something else here</a>
    </div>
</div>
</div>
<!-- Card Body -->
<div class="card-body">
    <div class="chart-area">
        <canvas id="myAreaChart"></canvas>
    </div>
</div>
</div>
</div>
<!-- Pie Chart -->
<div class="col-xl-4 col-lg-5">
    <div class="card shadow mb-4">
        <!-- Card Header - Dropdown -->
        <div class="card-header py-3 d-flex flex-row align-items-center justify-
content-between">

```

```

        <h6 class="m-0 font-weight-bold text-primary">Revenue
Sources</h6>
        <div class="dropdown no-arrow">
            <a class="dropdown-toggle" href="#" role="button"
id="dropdownMenuLink" data-toggle="dropdown"
            aria-haspopup="true" aria-expanded="false">
                <i class="fas fa-ellipsis-v fa-sm fa-fw text-gray-400"></i>
            </a>
            <div class="dropdown-menu dropdown-menu-right shadow
animated--fade-in"
                aria-labelledby="dropdownMenuLink">
                <div class="dropdown-header">Dropdown Header:</div>
                <a class="dropdown-item" href="#">Action</a>
                <a class="dropdown-item" href="#">Another action</a>
                <div class="dropdown-divider"></div>
                <a class="dropdown-item" href="#">Something else here</a>
            </div>
        </div>
    </div>
    <!-- Card Body -->
    <div class="card-body">
        <div class="chart-pie pt-4 pb-2">
            <canvas id="myPieChart"></canvas>
        </div>
        <div class="mt-4 text-center small">
            @foreach ($paket as $item)
            <span class="mr-2">
                <i class="fas fa-circle" style="color: {{ $item->warna }}"></i>
                {{ $item->nama }}
            </span>
            @endforeach
    </div>

```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<!-- /.container-fluid -->
<script>
    let data = [
        '{{ $jan }}',
        '{{ $feb }}',
        '{{ $mar }}',
        '{{ $apr }}',
        '{{ $mei }}',
        '{{ $jun }}',
        '{{ $jul }}',
        '{{ $aug }}',
        '{{ $sept }}',
        '{{ $oct }}',
        '{{ $nov }}',
        '{{ $dec }}',
    ];
    data = data.filter(val => val != "0");
    let jumlahP = <?= json_encode($jumlahP); ?>;
    let warna_p = <?= json_encode($warna_p); ?>;
    let label_p = <?= json_encode($label_p); ?>;
    // jumlahP = jumlahP.filter(val => val != "0");
    let uniq_warna_p = [...new Set(warna_p)];
    let uniq_label_p = [...new Set(label_p)];
    console.log(data);
    console.log(uniq_label_p);

```

```

        console.log(uniq_warna_p);
        console.log(jumlahP);
    </script>
    @endsection

```

3. *Route Code:*

web.php

```

<?php
use App\Models\Order;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BillController;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers>LoginController;
use App\Http\Controllers\OrderController;
use App\Http\Controllers\ProductController;
use App\Http\Controllers\RegisterController;
use App\Http\Controllers\DashboardOrderController;
use App\Http\Controllers\DashboardProductController;
use App\Http\Controllers\DashboardCustomerController;
use App\Http\Controllers\DashboardTransaksiController;
use App\Models\Product;
use App\Models\Transactions;
use SebastianBergmann\CodeCoverage\Report\Html\Dashboard;
Route::get('/', function () {
    return view('home', [
        'title' => 'Beranda'
    ]);
});
Route::get('/products', [
    ProductController::class, 'index'

```

```

]);
Route::get('/products/{product}', [
    ProductController::class, 'show'
]);
Route::post('/order', [
    OrderController::class, 'store'
]);
Route::get('/success', function () {
    return view('success', [
        'title' => 'Success'
    ]);
});
Route::get('/tentang', function () {
    return view('tentang', [
        'title' => 'Tentang Kami'
    ]);
});
Route::get('/bill', [BillController::class, 'index']);
Route::get('/histori', function () {
    $histori = Transactions::where('no_order_id', $_GET['id'])->get();
    return view('histori', [
        'title' => 'Histori Pembayaran',
        'histori' => $histori
    ]);
});
Route::get('/laravolt', function () {
    return view('laravolt', []);
});
Route::get('/login', [LoginController::class, 'index'])->name('login')-
    >middleware('guest');
Route::post('/login', [LoginController::class, 'authenticate']);

```



```

Route::post('/logout', [LoginController::class, 'logout'])->middleware('auth');
Route::get('/dev.register-admin', [RegisterController::class, 'index']);
Route::post('/dev.register-admin', [RegisterController::class, 'store']);
Route::get('/dashboard', [DashboardController::class, 'index'])-
    >middleware('auth');
Route::resource('/dashboard/products', DashboardProductController::class);
Route::resource('/dashboard/orders', DashboardOrderController::class);
Route::resource('/dashboard/customers', DashboardCustomerController::class);
Route::get('/dashboard/customers/edit-customer/{id}',
    [DashboardCustomerController::class, 'editCustomer'])-
    >name('editCustomer');
Route::post('/dashboard/update-customer/{id}',
    [DashboardCustomerController::class, 'updateCustomer'])-
    >name('updateCustomer');
Route::resource('/dashboard/transaksi', DashboardTransaksiController::class);
Route::post('/update-tgl', [DashboardTransaksiController::class, 'updateTgl'])-
    >name('updateTgl');
Route::post('/delete-url', [DashboardTransaksiController::class, 'deleteUrl'])-
    >name('deleteUrl');
Route::post('/cek-tagihan', [Controller::class, 'cekTagihan'])->name('cekTagihan');
Route::post('/buat-url', [DashboardTransaksiController::class, 'buat_url'])-
    >name('buat_url');
Route::post('/status-submit', [DashboardTransaksiController::class,
    'statusSubmit'])->name('statusSubmit');
Route::post('/ubahTgl', [DashboardTransaksiController::class, 'ubahTgl'])-
    >name('ubahTgl');
Route::get('/migrate', function() {
    $output = [];
    \Artisan::call('migrate', $output);
    dd($output);
});

```