

**RANCANG BANGUN APLIKASI *PET GROOMING* BERBASIS
WEBSITE MENGGUNAKAN *FRAMEWORK CODEIGNITER***

LAPORAN PROYEK II

Ditujukan sebagai salah satu syarat
Untuk memperoleh nilai pada kegiatan Proyek II
Program Studi DIV Teknik Informatika

Oleh

Al Novianti Ramadhani Sulaksono 1.21.4.032

Aryka Anisa Pertiwi 1.21.4.035



**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA
UNIVERSITAS LOGISTIK BISNIS INTERNASIONAL
BANDUNG**

2023

LEMBAR PERSETUJUAN

Rancang Bangun Aplikasi *Pet Grooming* Berbasis *Website* Menggunakan *Framework Codeigniter*

LAPORAN PROYEK II

Program Studi DIV Teknik Informatika

Oleh

Al Novianti Ramadhani Sulaksono (1214032)

Aryka Anisa Pertiwi (1204035)

Telah disetujui

Oleh

Pembimbing

Cahyo Prianto S.Pd.,M.T.,CDSP.,SFPC

NIK. 117.84.222

SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME

Yang bertandatangan di bawah ini:

Nama : Al Novianti Ramadhani Sulaksono

NPM : 1214032

Program Studi : D-IV TEKNIK INFORMATIKA

Judul : Rancang Bangun Aplikasi *Pet Grooming* Berbasis *Website*

Menggunakan *Framework Codeigniter*

Menyatakan bahwa :

1. Proyek Pemrograman Aplikasi (PROYEK II) saya ini adalah hasil dan belum pernah diajukan untuk memenuhi kelulusan matakuliah Proyek II pada Program Studi D-IV Teknik Informatika baik di Universitas Logistik dan Bisnis Internasional maupun di Perguruan Tinggi lainnya.
2. Proyek Pemrograman Aplikasi (PROYEK II) ini adalah murni hasil gagasan, rumusan, dan penelitian saya sendiri tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam Proyek Pemrograman Aplikasi (PROYEK II) ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan-penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku diperguruan tinggi lain.

Bandung,

Yang Membuat Pernyataan,

Al Novianti Ramadhani Sulaksono

NPM 1.21.4.032

SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME

Yang bertandatangan di bawah ini:

Nama : Aryka Anisa Pertiwi

NPM : 1214035

Program Studi : D-IV TEKNIK INFORMATIKA

Judul : Rancang Bangun Aplikasi *Pet Grooming* Berbasis *website*

Menggunakan *Framework Codeigniter*

Menyatakan bahwa :

1. Proyek Pemrograman Aplikasi (PROYEK II) saya ini adalah hasil dan belum pernah diajukan untuk memenuhi kelulusan matakuliah Proyek II pada Program Studi D-IV Teknik Informatika baik di Universitas Logistik dan Bisnis Internasional maupun di Perguruan Tinggi lainnya.
2. Proyek Pemrograman Aplikasi (PROYEK II) ini adalah murni hasil gagasan, rumusan, dan penelitian saya sendiri tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam Proyek Pemrograman Aplikasi (PROYEK II) ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan-penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku diperguruan tinggi lain.

Bandung,

Yang Membuat Pernyataan,

Aryka Anisa Pertiwi

NPM 1.21.4.035

ABSTRAK

Website dan teknologi *online* lainnya membuat akses informasi menjadi lebih mudah karena pengguna dapat mengaksesnya kapan saja dan di mana saja, bahkan saat sedang melakukan kegiatan lain. Ini sangat membantu bagi mereka yang sibuk dan tidak memiliki banyak waktu untuk mengunjungi *petshop* untuk perawatan hewan peliharaannya. Selain itu, *website* dan media sosial juga memberikan kemudahan bagi para pemilik *petshop* untuk menawarkan berbagai jenis pelayanan *pet grooming* untuk hewan peliharaan. Hal ini membuka peluang usaha yang menjanjikan bagi para pemilik *petshop* tetapi dalam proses merawat hewan, pemilik hewan peliharaan harus memperhatikan bahwa hewan tersebut mendapat perawatan yang cukup. Namun, kadang-kadang, aktivitas pekerjaan yang padat membuat pemilik hewan peliharaan tidak memiliki cukup waktu untuk mengurus hewan peliharaannya dengan baik, sehingga hewan tersebut bisa sakit akibat tidak dirawat dengan benar. Penelitian ini menggunakan metode waterfall dan menggunakan pemodelan Unified Modeling Language (UML) dalam merancang aplikasi yang dibuat menggunakan bahasa pemrograman PHP dan basis data MySQL. Tujuan dari aplikasi ini adalah untuk memudahkan *customer* dalam mengatur pemesanan layanan hewan serta memberikan informasi yang memadai kepada *customer*. Selain itu, aplikasi ini juga dikelola oleh admin untuk mengelola data pemesanan dari *customer* yang diterima melalui *website*.

Kata Kunci: *Grooming* , *website*, hewan, waterfall, *petshop*.

ABSTRACT

Websites and other online technologies make access to information easier because users can access it anytime and anywhere, even while doing other activities. This is very helpful for those who are busy and don't have much time to visit the pet shop for pet care. In addition, websites and social media also make it easy for pet shop owners to offer various types of pet grooming services for pets. This opens up promising business opportunities for pet shop owners but in the process of caring for animals, pet owners must pay attention that the animal is receiving adequate care. However, sometimes, busy work activities mean that pet owners do not have enough time to take good care of their pets, so that the animals can get sick as a result of not being cared for properly. This research uses the waterfall method and uses the Unified Modeling Language (UML) modeling in designing applications that are made using the PHP programming language and MySQL database. The purpose of this application is to make it easier for customers to arrange animal service orders and provide adequate information to customers. In addition, this application is also managed by the admin to manage order data from customers received through the website.

Keywords: *Grooming, website, animals, waterfall, pet shop.*

KATA PENGANTAR

Segala puji dan syukur kami panjatkan kehadirat Allah SWT karena atas berkat dan rahmatnya kami diberikan kesehatan jasmani dan rohani sehingga penulis dapat menyelesaikan Laporan Proyek II yang berjudul “Rancang Bangun Aplikasi *Pet Grooming* Berbasis *Website* Menggunakan *Framework Codeigniter*”.

Sebelumnya kami juga menyadari banyak kekurangan dalam penulisan laporan ini. Maka dari itu kami mengharapkan kritik dan saran yang baik dari para pembaca demi kesempurnaan laporan ini. Untuk itu kami mengucapkan terima kasih kepada:

1. DR. Ir. Agus Purnomo, M.T. selaku Direktur Universitas Logistik Dan Bisnis Internasional.
2. Kedua orang tua dan keluarga kami yang telah mendorong dan memberi semangat kepada kami .
3. Yang terhormat Roni Andarsyah, ST., M.KOM.,SFPC selaku Ketua Program Studi D-IV Teknik Informatika
4. Nisa Hanum H S.Kom.,M.T selaku Koordinator Proyek II.
5. Cahyo Prianto,.S.Pd.,M.T.,CDSP,SFPC selaku pembimbing, yang telah memberikan bimbingan,dorongan, dan motivasi selama penyusunan hingga selesainya proyek II ini.
6. Rd. Nuraini Siti Fathonah, S.S., M.Hum.,SFPC selaku dosen wali kelas D4 Teknik Informatika 2B.

Kami berharap semoga apa yang telah penulis kembangkan pada Laporan Proyek II ini dapat bermanfaat bagi pihak – pihak yang berkepentingan.

Bandung, 15 Januari 2023

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME	ii
SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	ix
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Tujuan	2
1.4 Lingkup Dokumentasi	2
BAB II LANDASAN TEORI	4
2.1 Kajian Teori	4
2.1.1 Hypertext Preprocessor (PHP).....	4
2.1.2 Hypertext Markup Language (HTML)	4
2.1.3 Cascading Style Sheet (CSS).....	4
2.1.4 MySQL	4
2.1.5 Basis Data	5
2.1.6 Flowmap	5
2.1.7 Website	5
2.1.8 Codeigniter.....	5
2.1.9 Grooming	6
2.1.10 Metode Waterfall.....	6
2.2 Aplikasi yang dipakai.....	7
2.2.1 Visual Studio Code.....	7
2.2.2 XAMPP	7
2.2.3 PhpMyAdmin.....	8
BAB III ANALISIS DAN PERANCANGAN	9
3.1 Analisis Sistem.....	9

3.1.1 Analisis Sistem Berjalan (<i>Current System</i>).....	9
3.1.2 Analisis Sistem Yang Akan Dibangun	10
3.1.3 Analisis Dokumen Yang Digunakan	14
3.1.5 Analisis Kebutuhan Fungsional.....	15
3.1.6 Analisis Kebutuhan Non Fungsional	15
3.2 Perancangan	17
3.2.1 Perancangan Sistem.....	17
3.2.3 Perancangan Struktur Menu	32
3.2.4 Perancangan UI	33
BAB IV IMPLEMENTASI DAN PENGUJIAN	55
4.1 Implementasi	55
4.1.1 Halaman Registrasi	55
4.1.2 Halaman <i>Login</i>	57
4.1.3 Halaman Home.....	59
4.1.4 Halaman <i>Grooming</i>	62
4.1.5 Halaman Order	65
4.1.6 Halaman Buat Pesanan	66
4.1.7 Halaman Profil	68
4.1.8 Halaman Dashboard Admin	70
4.1.9 Halaman Pesanan Admin	72
4.1.10 Halaman Paket <i>Grooming</i> Admin	74
4.1.11 Halaman Tambah Paket <i>Grooming</i> Admin.....	75
4.1.12 Halaman Ubah Paket <i>Grooming</i> Admin	77
4.1.13 Halaman Harga <i>Grooming</i> Admin.....	78
4.1.14 Halaman Tambah Harga <i>Grooming</i> Admin.....	79
4.1.15 Halaman Ubah Harga <i>Grooming</i> Admin	80
4.1.16 Halaman Hewan Peliharaan Admin.....	82
4.1.17 Halaman Tambah Hewan Peliharaan Admin.....	83
4.1.18 Halaman Ubah Hewan Peliharaan Admin	84
4.1.19 Halaman Biaya Antar Jemput Admin	86
4.1.20 Halaman Tambah Biaya Antar Jemput Admin	87
4.1.21 Halaman Ubah Biaya Antar Jemput Admin	88
4.1.22 Halaman <i>User</i> Admin.....	89
4.1.23 Halaman Informasi Akun	91
4.2 Pengujian dan Hasil Pengujian.....	92
4.2.1 Pengujian dan Hasil Pengujian Black Box	92

4.2.2 Pengujian dan Hasil Pengujian Selenium IDE.....	102
BAB V KESIMPULAN DAN SARAN	117
5.1 Kesimpulan	117
5.2 Saran	117
DAFTAR PUSTAKA	118
LAMPIRAN.....	120
GLOSARIUM.....	126

DAFTAR GAMBAR

Gambar 3.1 Flowmap pet <i>grooming</i>	10
Gambar 3.2 Flowmap prosedur register	11
Gambar 3.3 Flowmap prosedur <i>login</i>	12
Gambar 3.4 Flowmap pet <i>grooming</i>	13
Gambar 3.5 Flowmap kelola <i>treatment</i>	14
Gambar 3.6 Metode waterfall	18
Gambar 3.7 Use Case Diagram	18
Gambar 3.8 Sequence diagram register	20
Gambar 3.9 Sequence diagram <i>login</i>	21
Gambar 3.10 Sequence diagram melakukan pemesanan <i>treatment</i>	22
Gambar 3.11 Sequence diagram mengelola pemesanan <i>treatment</i>	22
Gambar 3.12 Sequence diagram logout	23
Gambar 3.13 Activity diagram register.....	24
Gambar 3.14 Activity diagram <i>login</i>	24
Gambar 3.15 Activity diagram melakukan pemesanan <i>treatment</i>	25
Gambar 3.16 Activity diagram kelola <i>treatment</i>	26
Gambar 3. 17 Activity diagram logout	27
Gambar 3.18 Class diagram pelanggan.....	27
Gambar 3.19 Class diagram admin	28
Gambar 3.20 Component diagram	28
Gambar 3.21 Deployment diagram	29
Gambar 3.22 CDM (Conceptual Data model)	30
Gambar 3.23 PDM (Physical Data Model)	31
Gambar 3.24 Perancangan database.....	32
Gambar 3.25 Perancangan struktur mmenu	33
Gambar 3.26 Perancangan UI halaman registrasi	34
Gambar 3.27 Perancangan UI halaman <i>login</i>	35
Gambar 3.28 Perancangan UI halaman home.....	36
Gambar 3.29 Perancangan UI halaman <i>grooming</i>	37
Gambar 3.30 Perancangan UI halaman <i>order</i>	38
Gambar 3.31 Perancangan UI halaman buat pesanan	39

Gambar 3.32 Perancangan UI halaman profil.....	39
Gambar 3.33 Perancangan UI halaman dashboard admin	40
Gambar 3.34 Perancangan UI halaman pesanan admin.....	41
Gambar 3.35 Perancangan UI halaman paket <i>grooming</i> admin	42
Gambar 3.36 Perancangan UI halaman tambah paket <i>grooming</i> admin.....	43
Gambar 3.37 Perancangan UI halaman ubah paket <i>grooming</i> admin.....	43
Gambar 3.38 Perancangan UI halaman harga <i>grooming</i> admin	44
Gambar 3.39 Perancangan UI halaman tambah harga <i>grooming</i> admin	45
Gambar 3.40 Perancangan UI halaman ubah harga <i>grooming</i> admin	46
Gambar 3.41 Perancangan UI halaman hewan peliharaan admin.....	47
Gambar 3.42 Perancangan UI halaman tambah hewan peliharaan admin.....	48
Gambar 3. 43 Perancangan UI halaman ubah hewan peliharaan admin.....	49
Gambar 3.44 Perancangan UI halaman biaya antar jemput	50
Gambar 3.45 Perancangan UI halaman tambah biaya antar jemput admin	51
Gambar 3.46 Perancangan UI halaman ubah biaya antar jemput admin	52
Gambar 3.47 Perancangan UI halaman <i>user</i> admin.....	53
Gambar 3.48 Perancangan UI halaman informasi akun <i>user</i>	53
Gambar 3.49 Perancangan UI halaman informasi akun admin.....	54
Gambar 4.1 Code program registrasi	55
Gambar 4.2 Antarmuka halaman regsiter	56
Gambar 4.3 Code program <i>login</i>	57
Gambar 4.4 Antarmuka halaman <i>login</i>	58
Gambar 4.5 Code program home	59
Gambar 4.6 Antarmuka halaman home 1	60
Gambar 4.7 Antarmuka halaman home 2	60
Gambar 4.8 Antarmuka halaman home 3	61
Gambar 4.9 Code program <i>grooming</i> 1	62
Gambar 4.10 Code program <i>grooming</i> 2	63
Gambar 4.11 Antarmuka halaman <i>grooming</i> 1.....	64
Gambar 4.12 Antarmuka halaman <i>grooming</i> 2.....	64
Gambar 4.13 Code program <i>order</i>	65
Gambar 4.14 Antarmuka halaman <i>order</i>	66

Gambar 4.15 Code program buat pesanan	66
Gambar 4.16 Antarmuka halaman buat pesanan.....	67
Gambar 4.17 Code program profil	68
Gambar 4.18 Antarmuka halaman profil	69
Gambar 4.19 Code program dashboard admin	70
Gambar 4.20 Antarmuka halaman dashboard admin.....	71
Gambar 4.21 Code program pesanan admin	72
Gambar 4.22 Antarmuka halaman pesanan admin	73
Gambar 4.23 Code program paket <i>grooming</i> admin.....	74
Gambar 4.24 Antarmuka halaman paket <i>grooming</i> admin	75
Gambar 4.25 Code program tambah paket <i>grooming</i> admin.....	75
Gambar 4.26 Antarmuka halaman tambah paket <i>grooming</i> admin	76
Gambar 4.27 Code program ubah paket <i>grooming</i> admin.....	77
Gambar 4.28 Antarmuka halaman ubah paket <i>grooming</i> admin	77
Gambar 4.29 Code program harga <i>grooming</i> admin	78
Gambar 4.30 Antarmuka halaman harga <i>grooming</i> admin.....	79
Gambar 4.31 Code program tambah harga <i>grooming</i> admin.....	79
Gambar 4. 32 Antarmuka halaman tambah harga <i>grooming</i> admin	80
Gambar 4.33 Code program ubah harga <i>grooming</i> admin.....	80
Gambar 4.34 Antarmuka halaman ubah harga <i>grooming</i> admin	81
Gambar 4.35 Code program hewan peliharaan admin.....	82
Gambar 4.36 Antarmuka halaman hewan peliharaan admin	83
Gambar 4.37 Code program tambah hewan peliharaan admin	83
Gambar 4.38 Antarmuka halaman tambah hewan peliharaan admin	84
Gambar 4.39 Code program ubah hewan peliharaan admin	84
Gambar 4.40 Code program ubah hewan peliharaan admin	85
Gambar 4.41 Code program biaya antar jemput admin	86
Gambar 4.42 Antarmuka halaman biaya antar jemput admin.....	87
Gambar 4.43 Code program tambah biaya antar jemput admin	87
Gambar 4.44 Antarmuka halaman tambah biaya antar jemput admin.....	88
Gambar 4.45 Code program ubah biaya antar jemput admin	88
Gambar 4.46 Antarmuka halaman ubah biaya antar jemput admin.....	89

Gambar 4.47 Code program <i>user</i> admin	89
Gambar 4.48 Antarmuka halaman <i>user</i> admin	90
Gambar 4.49 Code program informasi akun profil	91
Gambar 4.50 Code program informasi akun logout	91
Gambar 4.51 Antarmuka halaman informasi akun <i>user</i>	92
Gambar 4.52 Antarmuka halaman informasi akun admin	92
Gambar 4.53 Selenium IDE	103
Gambar 4.54 Cara kerja selenium IDE	103
Gambar 4.55 Cara kerja selenium IDE	103
Gambar 4.56 Prsoses testing selenium IDE	104
Gambar 4.57 Record testing	104
Gambar 4.58 Run testing	104
Gambar 4.59 Pengujian dan hasil pengujian menu register	105
Gambar 4.60 Pengujian dan hasil pengujian menu <i>login</i>	106
Gambar 4.61 Pengujian dan hasil pengujian menu home	107
Gambar 4.62 Pengujian dan hasil pengujian menu <i>grooming</i>	107
Gambar 4.63 Pengujian dan hasil pengujian menu <i>order</i>	108
Gambar 4.64 Pengujian dan hasil pengujian menu profil	109
Gambar 4.65 Pengujian dan hasil pengujian menu dashboard admin	109
Gambar 4.66 Pengujian dan hasil pengujian menu pesanan admin	110
Gambar 4.67 Pengujian dan hasil pengujian menu paket <i>grooming</i> admin	111
Gambar 4.68 Pengujian dan hasil pengujian menu harga <i>grooming</i> admin	112
Gambar 4.69 Pengujian dan hasil pengujian menu hewan peliharaan admin	113
Gambar 4.70 Pengujian dan hasil pengujian menu biaya antar jemput admin	114
Gambar 4.71 Pengujian dan hasil pengujian menu <i>user</i> admin	115
Gambar 4.72 Pengujian dan hasil pengujian menu profile admin	115

DAFTAR TABEL

Tabel 3.1 Dokumen pemesanan <i>treatment</i>	14
Tabel 3. 2 Dokumen pembuatan laporan	15
Tabel 3. 3 Kebutuhan perangkat keras 1	16
Tabel 3. 4 Kebutuhan perangkat keras 2.....	16
Tabel 3.5 Kebutuhan perangkat lunak <i>server 1</i>	17
Tabel 3. 6 Kebutuhan perangkat lunak <i>server 2</i>	17
Tabel 3.7 Deskripsi use case	19
Tabel 3.8 Definisi use case.....	19
Tabel 4.1 Pengujian <i>user</i>	93
Tabel 4.2 Hasil pengujian <i>user</i>	93
Tabel 4. 3 Pengujian admin.....	95
Tabel 4.4 Hasil pengujian admin	97

DAFTAR LAMPIRAN

Lampiran 1 Menu Register.....	120
Lampiran 2 Menu Login	120
Lampiran 3 Menu Home	121
Lampiran 4 Menu Grooming	121
Lampiran 5 Menu Order.....	121
Lampiran 6 Menu Profil.....	122
Lampiran 7 Menu Dashboard Admin	122
Lampiran 8 Menu Pesanan Admin.....	122
Lampiran 9 Menu Paket Grooming Admin	123
Lampiran 10 Menu Harga Grooming Admin.....	124
Lampiran 11 Menu Hewan Peliharaan Admin.....	124
Lampiran 12 Menu Biaya Antar Jemput Admin.....	125
Lampiran 13 Menu User Admin	125
Lampiran 14 Menu Profile Admin.....	125

BAB I

PENDAHULUAN

1.1. Latar Belakang

Petshop sering diartikan sebagai toko barang-barang kebutuhan untuk hewan peliharaan. *Petshop* biasanya menyediakan layanan atau jasa perawatan yang biasa disebut dengan *Pet Grooming* (salon hewan). Layanan *pet grooming* ini seperti mandi, gunting kuku, terapi jamur dan lain-lain. Di era digital saat ini semua dapat dilakukan dengan mudah. Dengan segala kesibukan pemilik hewan peliharaan yang mereka harus bekerja serta tidak ada waktu untuk merawat hewan peliharaan mereka. Masalah lain yang ada seperti pelanggan kesulitan untuk mengetahui informasi layanan yang tersedia serta biaya layanan grooming. Pemesanan layanan grooming tersebut juga masih dilakukan secara offline atau via whatsapp dan melalui telepon. Hal itu tidak efektif karena pelanggan harus menunggu konfirmasi dari pihak *pet grooming* apakah layanan *pet grooming* tersebut tersedia. Permasalahan utama yaitu pemesanan layanan grooming pada Frontera *Petshop* masih menggunakan sistem konvensional, yang mana para pelanggan harus datang ke *petshop* dengan membawa hewan peliharaan mereka.

Melihat permasalahan yang ada maka Frontera *Petshop* perlu membuat inovasi baru melalui bidang teknologi informasi, yaitu dengan membuat *website* pemesanan layanan *pet grooming* ini. Dan menambahkan fasilitas antar jemput hewan peliharaan yang akan di *grooming*. Yang bertujuan agar pelanggan tidak perlu datang ke *petshop* karena hewan peliharaan akan langsung dijemput dan diantarkan kembali ke rumah setelah selesai perawatan.

Secara tidak langsung aplikasi *pet grooming* juga sangat menguntungkan bagi pemilik *petshop*, karena pemilik *petshop* bisa memberikan layanan secara mudah dan bisa sebagai media promosi untuk layanan *pet grooming*. Jadi para calon konsumen bisa merawat hewan peliharaan jika konsumen tidak sempat merawat sendiri hewan peliharaanya tersebut. Pemilik *petshop* pun memiliki kesulitan dalam mencatat semua pesanan *pet grooming*. Ada berbagai kendala seperti catatan pesanan yang hilang atau adanya kesalahan pencatatan, sehingga dengan aplikasi *pet grooming* berbasis *website* ini pencatatan dapat dilakukan dengan terkomputerisasi yang ketepatan, kecepatan dan

keakuratannya lebih unggul. Dibanding dengan pencatatan pesanan layanan *pet grooming* secara konvensional atau manual. Mengingat persaingan bisnis yang semakin banyak, dan tentunya sebuah teknologi informasi mampu menjawab dan menjembatani kebutuhan pokok sumber input informasi untuk diolah menjadi bahan pertimbangan, sehingga diharapkan sebuah hasil output yang maksimal.

Dengan adanya aplikasi *pet grooming* berbasis *website* ini diharapkan Frontera *Petshop* dapat menambah daya tarik pelanggan lebih banyak dengan jangkauan yang lebih luas lagi. Serta dapat memberikan kemudahan bagi pihak pemilik *petshop* untuk mencatat pesanan *pet grooming* yang masuk.

1.2. Identifikasi Masalah

Berdasarkan latar belakang diatas dapat dirumuskan permasalahan yaitu :

1. Bagaimana membuat atau merancang aplikasi *pet grooming* berbasis *website* yang berfungsi untuk mempromosikan serta memberikan informasi tentang *grooming* yang ditawarkan.
2. Bagaimana memberikan kemudahan bagi konsumen untuk melakukan pemesanan *pet grooming* secara online.
3. Bagaimana memberikan kemudahan pencatatan data pesanan *pet grooming* bagi pihak *petshop*.

1.3. Tujuan

Tujuan dari penulisan laporan ini adalah sebagai berikut :

1. Merancang atau membuat aplikasi *pet grooming* berbasis *website* yang berfungsi untuk mempromosikan serta memberikan informasi tentang *grooming* yang ditawarkan.
2. Memberikan kemudahan bagi konsumen untuk melakukan pemesanan *grooming* hewan melalui *pet grooming* secara online.
3. Memberikan kemudahan bagi pihak *petshop* dalam melakukan pencatatan data pesanan *pet grooming* konsumen.

1.4 Lingkup Dokumentasi

Dalam hal ini, kami akan membatasi masalah sebagai lingkup dokumentasi untuk menghindari pembahasan yang menyimpang dari tujuannya. Adapun lingkup dokumentasi nya adalah :

1. Tidak ada menu transaksi.
2. Nomor antrean pelanggan diberikan via whatsapp.
3. Menggunakan *framework* Codeigniter 3 dan bahasa pemrograman PHP.

BAB II

LANDASAN TEORI

2.1 Kajian Teori

2.1.1 Hypertext Preprocessor (PHP)

Hypertext Preprocessor (PHP) merupakan salah satu bahasa pemrograman yang berjalan dalam sebuah web *server* dan berfungsi sebagai pengolah data pada sebuah *server*. Data yang dikirim oleh *user client* akan diolah dan disimpan pada database web *server* dan dapat ditampilkan kembali apabila diakses. Untuk menjalankan code-code program PHP, file harus di *upload* kedalam *server*. *Upload* adalah proses mentransfer data atau file dari komputer *client* ke dalam web *server*. Untuk membuat *website* yang dinamis dan mudah di *update* setiap saat dari browser, dibutuhkan sebuah program yang mampu mengolah data dari komputer *client* atau dari komputer *server* itu sendiri sehingga mudah dan nyaman disajikan di browser. Salah satu program yang dapat dijalankan di *server* dan cukup andal adalah PHP. [1]

2.1.2 Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) adalah bahasa pemrograman yang digunakan untuk menampilkan sebuah *website*. HTML termasuk dalam bahasa pemrograman gratis, artinya tidak dimiliki oleh siapapun, pengembangannya dilakukan oleh banyak orang di banyak negara dan bisa dikatakan sebagai sebuah bahasa yang dikembangkan bersama-sama secara global. [2]

2.1.3 Cascading Style Sheet (CSS)

Cascading Style Sheets (CSS). CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / *markup language*. Jika kita berbicara dalam konteks web, bisa di artikan secara bebas sebagai : CSS merupakan bahasa yang digunakan untuk mengatur tampilan / desain suatu halaman HTML. [3]

2.1.4 MySQL

MySQL adalah program database *server* yang mampu menerima dan mengirimkan data dengan sangat cepat, multi *user* serta menggunakan perintah standar *Structured Query Language* (SQL) dan baik digunakan sebagai *client* maupun *server*. [4]

2.1.5 Basis Data

Basis data terdistribusi (*distributed database*) adalah suatu basis data yang berada di bawah kendali sistem manajemen basis data (DBMS) terpusat dengan peranti penyimpanan (*storage devices*) yang terpisah-pisah satu dari yang lainnya. Tempat penyimpanan ini dapat berada di satu lokasi yang secara fisik berdekatan (misal: dalam satu bangunan) atau terpisah oleh jarak yang jauh dan terhubung melalui jaringan internet. [5]

2.1.6 Flowmap

Flowmap atau juga dapat disebut *block chart* atau *Flow Of Document (FOD)* / bagan alir merupakan penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Flowmap efektif digunakan untuk menggambarkan proses maupun prosedur dalam sebuah organisasi. Flowmap digunakan untuk menggambarkan urutan prosedur/proses kerja dalam pembuatan sistem. Ada beberapa hal yang harus diperhatikan oleh seorang analis dan programmer akan membuat flowmap, diantaranya beberapa petunjuk yang harus diperhatikan secara detail, seperti :

- a. Flowmap digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
- b. Durasi waktu aktivitas dimulai dan berakhir harus ditentukan secara jelas.
- c. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja yang jelas. [6]

2.1.7 Website

Website merupakan penyedia informasi yang lengkap dan telah menjadi salah satu aspek penting dalam berbagai bidang terutama pendidikan karena dapat diakses dengan mudah dan tanpa batasan. Hal inilah yang membuat *website* sebagai wadah informasi untuk para mahasiswa, dosen maupun masyarakat luas. Pelayanan *website* tentunya didukung dengan banyak fitur sebagai informasi seperti data mahasiswa, kegiatan akademik, pembayaran, nilai, bahkan layanan akademik kemahasiswaan. [7]

2.1.8 Codeigniter

Codeigniter adalah sebuah *framework* bahasa pemrograman php. Codeigniter bisa dikatakan *framework* PHP paling populer di Indonesia karena kemudahan yang ditawarkan dalam penggunaannya. Codeigniter menawarkan kemudahan dalam proses pengembangan *website* dan aplikasi berbasis web. Dengan menggunakan codeigniter

proses pengembangan *website* menjadi lebih cepat dan terstandar. Codeigniter juga menyediakan library dan helper yang berguna untuk mempermudah proses development. [8]

2.1.9 Grooming

Grooming merupakan salah satu tingkah laku pada hewan untuk merawat dirinya dari ektoparasit yang melekat pada rambut di permukaan tubuhnya. *Grooming* yang dilakukan hewan pada dirinya sendiri disebut *autogrooming*, dan *grooming* yang dilakukan secara perpasangan disebut *alogrooming*. [9]

2.1.10 Metode Waterfall

Waterfall merupakan salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan yaitu setiap fase dalam waterfall harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya [10]. Metode waterfall terbagi atas lima tahap, yaitu:

a. Requirement

Analisis ini tahapan awal yang dilakukan di mana tahapan ini dilakukan untuk mengembangkan program yang akan dibuat. Dalam tahapan ini penulis mencari data pada Frontera Petshop, di mana pengumpulan data secara non fisik.

b. Design

Dalam tahapan ini penulis melakukan perencanaan perancangan pembuatan program sebelum memulai pengkodean pada program. Dalam pendesainan sebelum membuat program ini penulis menggunakan Unified Modelling Language (UML) yang terdiri dari use case diagram, sequence diagram, activity diagram, class diagram, deployment diagram, component diagram, dan desain database serta pendesainan halaman-halaman pada program.

c. Implementation

Dilakukan untuk pengujian terhadap program yang telah dibuat yang bertujuan untuk mengetahui kinerja pada program tersebut. Penulis melakukan pengujian program menggunakan teknik pengujian black box. Teknik pengujian ini berfokus pada fungsionalitas program.

d. Integration & Testing

Dalam tahapan ini program diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin kebutuhan sistem yang telah terpenuhi oleh program.

e. Operation & Maintenance

Pada tahap terakhir kita melakukan tahap *maintenance* atau pemeliharaan dan perbaikan terhadap program yang sudah dibuat.

2.2 Aplikasi yang dipakai

2.2.1 Visual Studio Code

Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java, dst). Banyak sekali fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya Intellisense, Git Integration, Debugging, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VS Code dengan teks editor-teks editor yang lain.

Teks editor VS Code juga bersifat *open source*, yang mana code sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Code sumber dari VS Code ini pun dapat dilihat di link Github. Hal ini juga yang membuat VS Code menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan VS Code ke depannya. [10]

2.2.2 XAMPP

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP berperan sebagai *server* web pada komputer lokal. XAMPP juga dapat disebut sebuah Cpanel *server* virtual, yang dapat membantu melakukan preview sehingga dapat dimodifikasi *website* tanpa harus online atau terakses dengan internet. XAMPP merupakan pengembangan dari lamp (linux, apache, MySQL, PHP, dan PERL). XAMPP adalah proyek non profit yang dikembangkan oleh apache friend yang didirikan oleh kai oswald seidler dan ayvogelgesang pada tahun 2002 proyek ini bertujuan mempromosikan penggunaan apache web *server* Paket amp (apache, mysql, php), salah satunya adalah XAMPP yang sudah terintegrasi amp di dalamnya dan

menghemat *resource* computer dari pada menginstal amp satu persatu. Untuk mempermudah proses instalasi ketiga produk tersebut secara instant dapat menggunakan XAMPP dalam suatu proses install. [11]

2.2.3 PhpMyAdmin

Pada perkembangannya, MYSQL disebut juga SQL yang merupakan singkatan dari Structured Query Language. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. SQL pertama kali didefinisikan *oleh American National Standards Institute (ANSI)* pada tahun 1986. MYSQL adalah sebuah sistem manajemen database yang setbersifat *open source*. MYSQL merupakan sistem manajemen database yang bersifat relational. Artinya, data yang dikelola dalam database yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. MYSQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar. [12]

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis adalah suatu kegiatan berpikir untuk menguraikan suatu hal menjadi bagian-bagian tertentu sehingga bisa diketahui ciri atau tanda pada setiap bagian, hubungan antar bagian satu sama lain, dan juga fungsi dari masing-masing bagian. Sedangkan pengertian dari sistem adalah suatu kesatuan, baik objek nyata atau abstrak yang terdiri dari berbagai komponen yang saling berkaitan, saling tergantung, saling mendukung, dan secara keseluruhan bersatu dalam satu kesatuan untuk mencapai tujuan tertentu secara efektif dan efisien.

Analisis sistem adalah penjabaran dari suatu sistem informasi yang utuh ke dalam berbagai macam bagian komponennya dengan maksud agar kita dapat mengidentifikasi atau mengevaluasi berbagai macam masalah maupun hambatan yang akan timbul pada sistem sehingga nantinya dapat dilakukan penanggulangan, perbaikan atau juga pengembangan.

Pada bagian ini, dibahas tentang analisis prosedur yang digambarkan dalam bentuk flowmap, pengkodean dan analisis *system* non fungsional yang meliputi perangkat keras dan perangkat lunak yang digunakan, serta analisis *user* yang terlibat dalam aplikasi tersebut. Tahapan ini sangat penting dalam membantu melanjutkan tahapan yang selanjutnya yaitu tahapan perancangan.

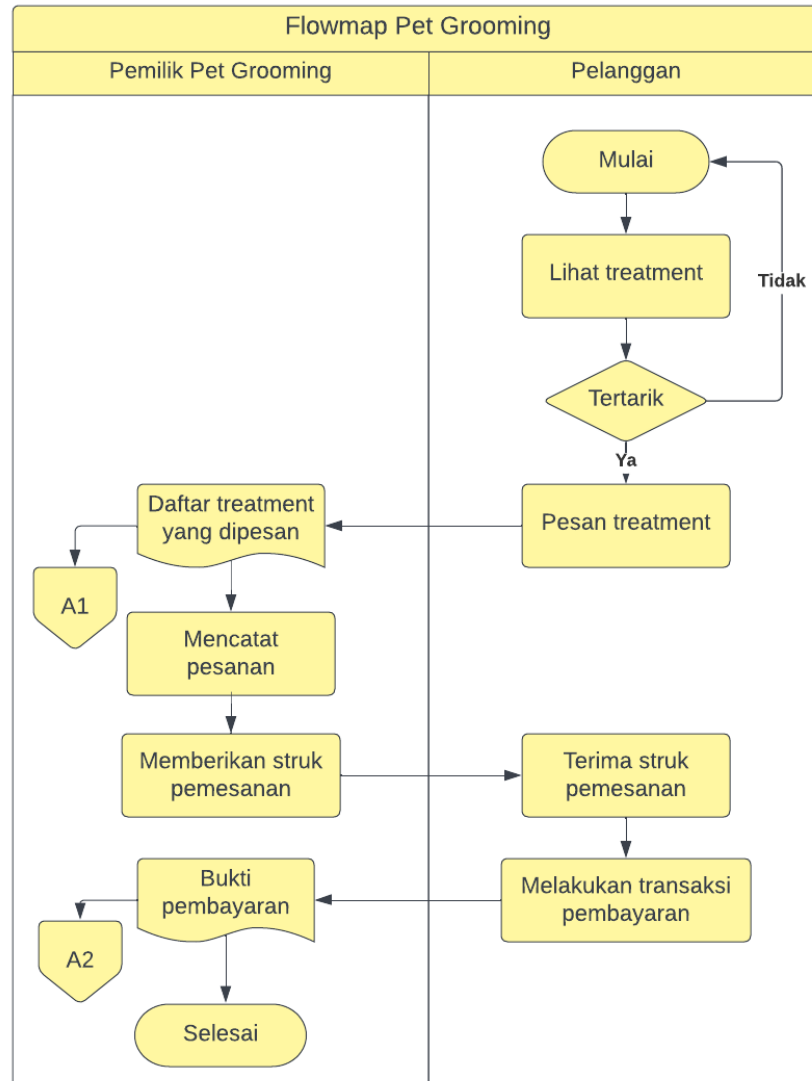
3.1.1 Analisis Sistem Berjalan (*Current System*)

Sebelum melakukan perancangan sistem, diperlukan analisis terhadap sistem yang sedang berjalan. Tujuan dari analisis dan evaluasi sistem adalah untuk menganalisis sistem pengelolaan data serta pengembangan sistem melalui perbaikan sehingga sistem informasi pemesanan dan transaksi pada aplikasi *pet grooming* ini dapat menghasilkan informasi yang akurat dan relevan.

3.1.1.1 Analisis Sistem Yang Berjalan Pada *Pet Grooming*

Pelanggan datang ke *petshop* lalu melihat daftar *treatment*, apabila tidak tertarik maka pelanggan akan keluar *petshop*. Pelanggan yang tertarik akan melakukan pemesanan *treatment*. *Treatment* yang dipesan akan masuk ke data daftar *treatment* yang dipesan. Pemilik akan mencatat pesanan, lalu memberikan struk pemesanan.

Pelanggan menerima struk pemesanan dan melakukan transaksi pembayaran. Pemilik *petshop* akan memberi bukti pembayaran. Adapun flowmap prosedur pada *pet grooming*, adalah sebagai berikut :



Gambar 3.1 Flowmap pet grooming

Keterangan :

A1 : Arsip daftar *treatment* yang dipesan pelanggan

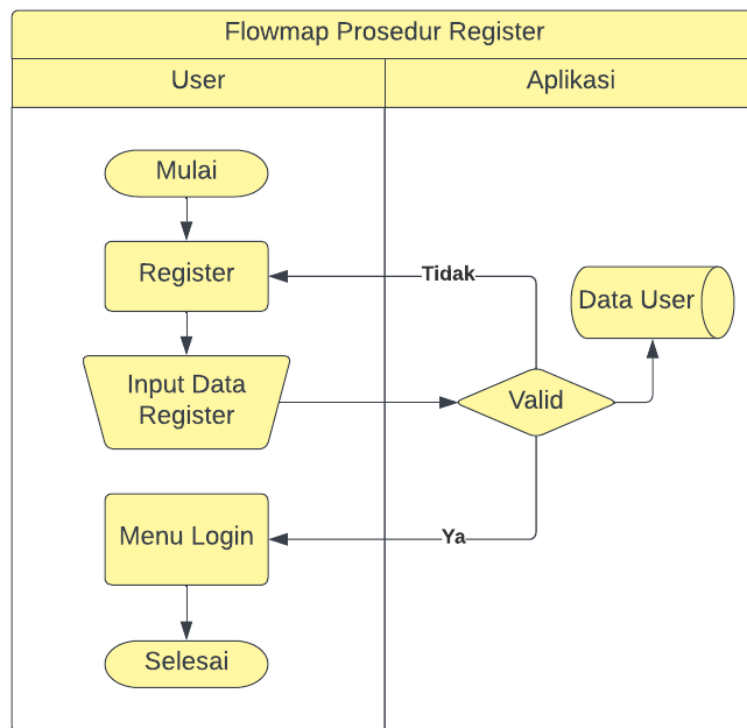
A2 : Arsip bukti pembayaran

3.1.2 Analisis Sistem Yang Akan Dibangun

Analisis kebutuhan yang dimaksud disini berupa analisis *flowmap* mengenai sistem yang akan dibangun pada aplikasi *pet grooming* berbasis *website*.

3.1.2.1 Analisis Sistem Yang Akan Dibangun Pada Prosedur Register

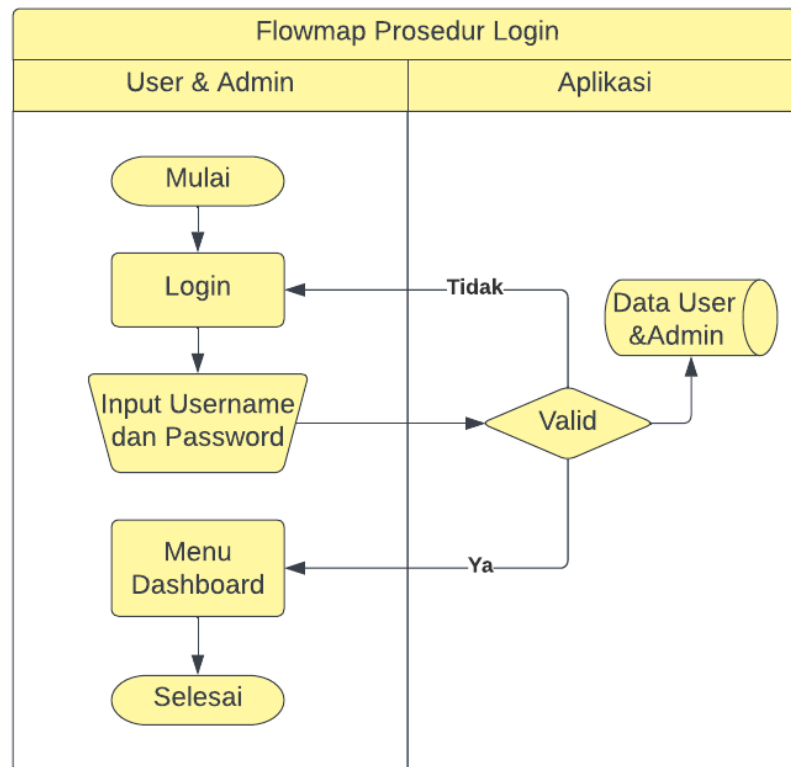
Pada prosedur register melibatkan 1 entitas (aktor) yaitu pelanggan, dimana memiliki hak akses register terhadap sistem atau aplikasi yang dibuat. Untuk menu register pelanggan harus memasukkan nama dan no. telepon, alamat, email dan password, apabila setelah itu akan di arahkan pada menu *login*. Masukkan *username* (email) dan password yang telah di buat tadi, jika valid maka akan masuk ke menu dashboard pelanggan.



Gambar 3.2 Flowmap prosedur register

3.1.2.2 Analisis Sistem Yang Akan Dibangun Pada Prosedur Login

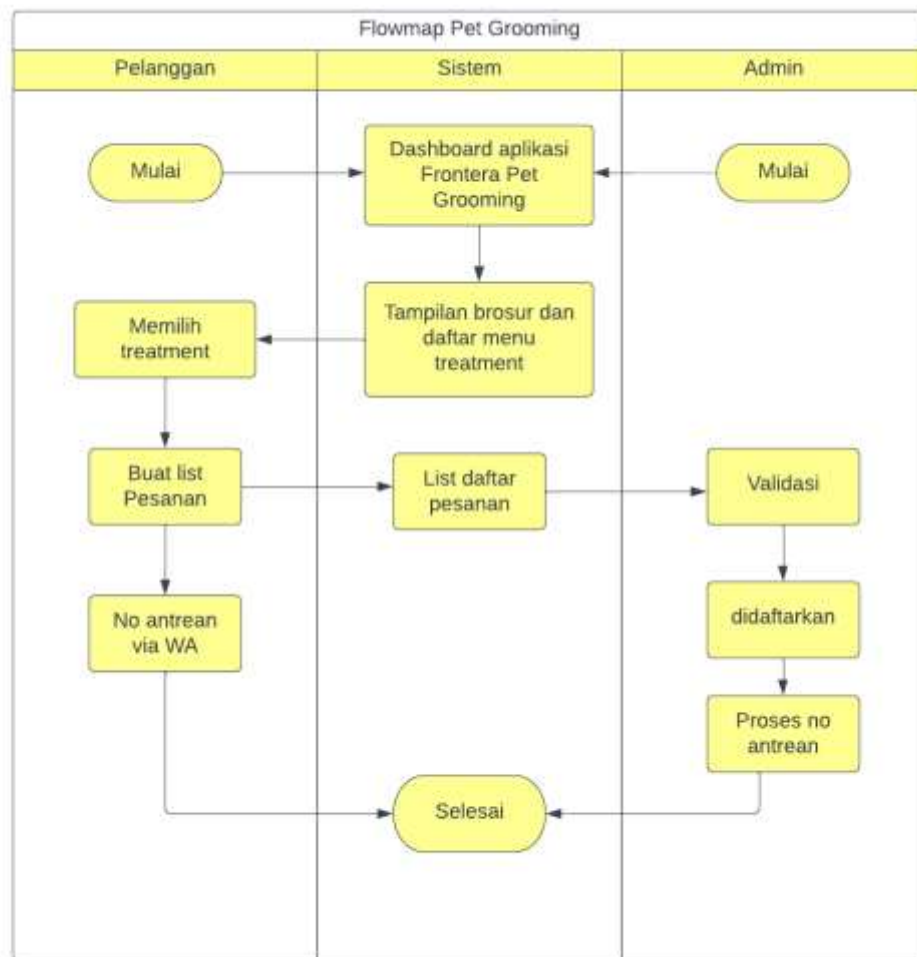
Pada prosedur *login* melibatkan 2 entitas (aktor) yaitu admin dan *user* atau pelanggan, dimana memiliki hak akses *login* terhadap sistem atau aplikasi yang dibuat. Untuk menu *login* admin harus memasukkan username dan password, apabila username dan password valid maka admin atau *user* akan masuk ke menu dashboard. Jika admin atau *user* salah menginputkan username dan password maka akan kembali ke menu *login*. Adapun flowmap prosedur *login* adalah sebagai berikut :



Gambar 3.3 Flowmap prosedur login

3.1.2.3 Analisis Sistem Yang Akan Dibangun Pada Prosedur Pemesanan *Pet Grooming*

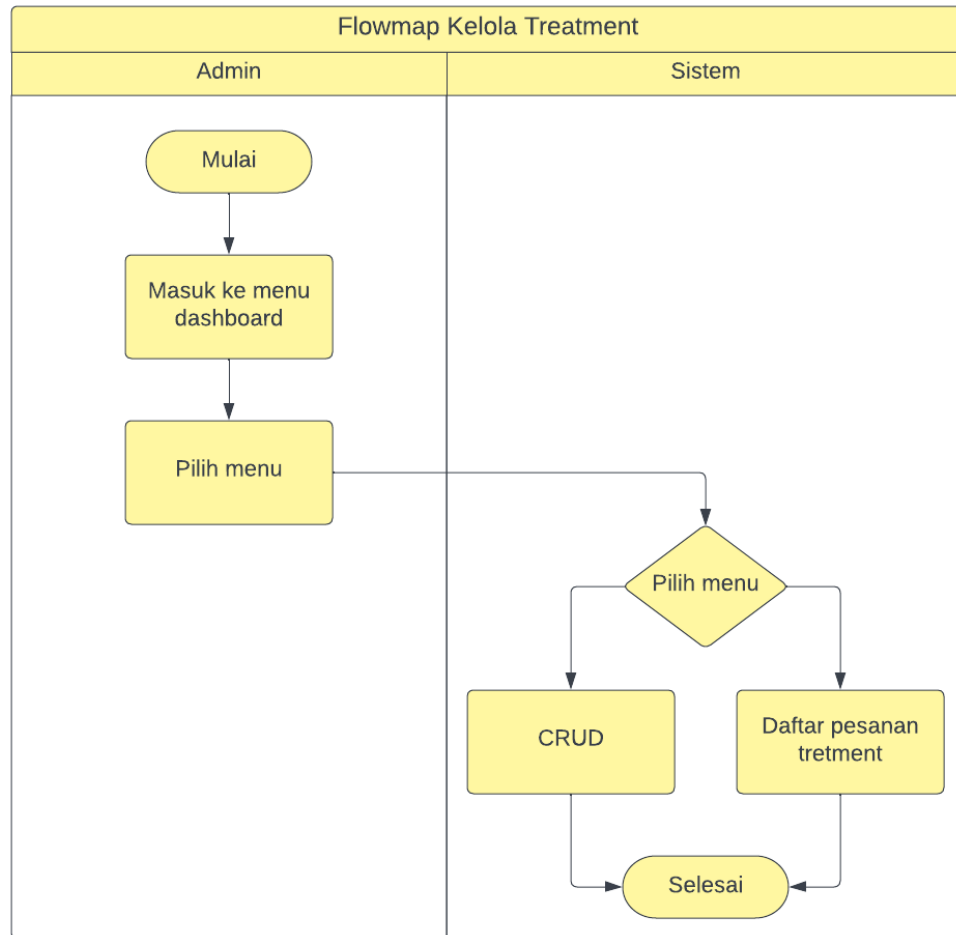
Proses dimulai dari pelanggan yang melakukan proses registrasi dan/atau *login*, begitu juga admin yang melakukan *login* lalu masuk ke halaman dashboard aplikasi *pet grooming*. Lalu pelanggan dapat memilih lalu memesan menu *treatment* yang ada pada aplikasi *pet grooming*, setelah itu admin akan melakukan validasi pada pesanan yang dipesan oleh pelanggan, apabila valid maka admin akan memberikan no antrean pesanan kepada pelanggan melalui WA. Selanjutnya untuk proses pembayaran dilakukan secara manual atau *cash* saat datang langsung ke tempat *pet grooming* atau melalui petugas yang melakukan antar jemput hewan peliharaan ke rumah pelanggan. Adapun flowmap prosedur pada *pet grooming*, adalah sebagai berikut:



Gambar 3.4 Flowmap pet grooming

3.1.2.4 Analisis Sistem Yang Akan Dibangun Pada Admin Prosedur Kelola Pemesanan *Treatment*

Pada prosedur kelola *treatment* melibatkan 1 entitas (aktor) yaitu admin, dimana memiliki hak akses kelola *treatment* terhadap sistem atau aplikasi yang dibuat. Admin masuk ke menu dashboard dengan cara *login* terlebih dahulu. Kemudian admin akan memilih menu ingin melakukan CRUD *treatment* atau ke menu daftar pesanan *treatment*. Adapun flowmap prosedur pada kelola *treatment*, adalah sebagai berikut:



Gambar 3.5 Flowmap kelola treatment

3.1.3 Analisis Dokumen Yang Digunakan

Dari hasil analisis yang dilakukan, dokumen yang digunakan oleh *pet grooming* diantaranya adalah dokumen penjualan *treatment* dan pembuatan laporan. Untuk mempermudah pendokumentasian suatu *treatment* maka dibutuhkan suatu sistem yang dapat mempermudah dalam proses pencarian datanya. Adapun dokumen yang dimaksud adalah sebagai berikut :

Dokumen pemesanan *treatment*

Tabel 3.1 Dokumen pemesanan treatment

Dibuat oleh	Pemilik <i>pet grooming</i>
Dibuat untuk	Pelanggan
Isi	Berupa form <i>treatment</i> yang dipesan
Frekuensi	Dibuat sesuai <i>treatment</i> yang dipesan
Tujuan	Menyediakan <i>treatment</i> untuk pelanggan

Dokumen pembuatan laporan

Tabel 3. 2 Dokumen pembuatan laporan

Dibuat oleh	Pemilik <i>pet grooming</i>
Dibuat untuk	Pemilik <i>pet grooming</i> sebagai bukti pemesanan <i>treatment</i>
Isi	Laporan rekap data <i>treatment</i> yang dipesan oleh pelanggan
Frekuensi	Dibuat sesuai rekap data <i>treatment</i> yang dibeli pelanggan
Tujuan	Menyediakan laporan rekap data <i>treatment</i> yang dipesan pelanggan

3.1.4 Analisis User

Aplikasi yang akan dibuat ini digunakan dalam lingkungan *petshop* khususnya untuk mengelola data pesanan *grooming* hewan. Antara lain melibatkan Admin dan *User* (pelanggan).

3.1.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan suatu kebutuhan yang berhubungan dengan kebutuhan sistem yang akan dibuat. Dimana menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem, adapun kebutuhan fungsional sebagai berikut:

A. Kebutuhan fungsi fungsional pelanggan

- 1) Pelanggan dapat melakukan register dan *login* aplikasi
- 2) Pelanggan dapat melihat menu *treatment*
- 3) Pelanggan dapat mengisi form pemesanan *pet grooming*
- 4) Pelanggan akan mendapatkan no antrean melalui WA

B. Kebutuhan fungsi fungsional admin

- 1) Admin dapat melakukan *login* aplikasi
- 2) Admin dapat memvalidasi data pesanan pelanggan
- 3) Admin dapat memberikan no antrean kepada pelanggan melalui WA
- 4) Admin dapat mengedit, hapus dan menambahkan pada menu *treatment*

3.1.6 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional adalah spesifikasi kebutuhan *system* seperti perangkat keras (*hardware*), perangkat lunak (*software*). Analisis ini dilakukan bertujuan untuk mengetahui spesifikasi kebutuhan *system* spesifikasi

kebutuhan yang melibatkan analisis perangkat keras (*hardware*) dan perangkat lunak (*software*).

A. Kebutuhan Perangkat Keras

Tabel 3. 3 Kebutuhan perangkat keras 1

No.	Nama Perangkat	Spesifikasi	Keterangan
1.	Hardisk	465.76 GB (500,105,249,280 bytes)	Media untuk menyimpan data aplikasi website yang dibuat.
2.	Memory	4GB	Memory sistem yang digunakan.
3.	Processor	Intel® Celeron ® 2957U @1.40GHz (2 CPUs), ~1.4GHz	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer.
4.	Infrastruktur Jaringan	Internet / jaringan lokal	Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses.
5.	Sistem Operasi	Windows 10 pro 64-bit (10.0, Build 10240)	Mengatur fungsi program <i>software</i> agar terhubung dengan perangkat keras tersebut.
6.	Database	MySQL	Sebagai tempat untuk menyimpan data.

Tabel 3. 4 Kebutuhan perangkat keras 2

No.	Nama Perangkat	Spesifikasi	Keterangan
1.	Hardisk	1TB	Media untuk menyimpan data aplikasi website yang dibuat.
2.	Memory	8GB	Memory sistem yang digunakan.
3.	Processor	Intel® Core™ i3-4170 CPU @3.70GHz (4 CPUs), ~3.7GHz	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer.

4.	Infrastruktur Jaringan	Internet / jaringan lokal	Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses.
5.	Sistem Operasi	Windows 10 pro 64-bit (10.0, Build 19041)	Mengatur fungsi program <i>software</i> agar terhubung dengan perangkat keras tersebut.
6.	Database	MySQL	Sebagai tempat untuk menyimpan data.

B. Kebutuhan Perangkat Lunak Server

Tabel 3.5 Kebutuhan perangkat lunak server 1

No.	Tools / Software	Fungsi
1.	<i>Windows 10</i>	Sistem Operasi
2.	<i>Xampp 3.3.0</i>	Server Basis Data
3.	<i>Canva</i>	Desain Antarmuka
4.	<i>PHP, HTML, CSS, JavaScript</i>	Bahasa Pemograman Yang digunakan
5.	<i>PDF</i>	Document
6.	<i>Google Chrome</i>	Browser

Tabel 3.6 Kebutuhan perangkat lunak server 2

No.	Tools / Software	Fungsi
1.	<i>Windows 10</i>	Sistem Operasi
2.	<i>Xampp 3.3.0</i>	Server Basis Data
3.	<i>Canva</i>	Desain Antarmuka
4.	<i>PHP, HTML, CSS, JavaScript</i>	Bahasa Pemograman Yang digunakan
5.	<i>PDF</i>	Document
6.	<i>Google Chrome</i>	Browser

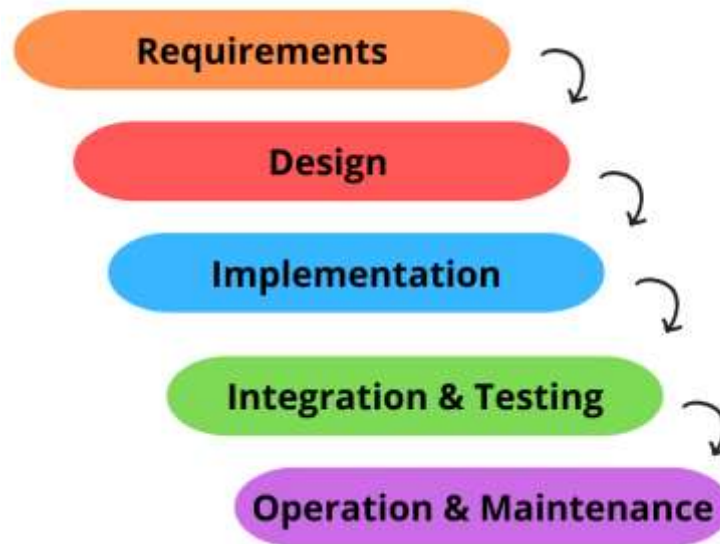
3.2 Perancangan

3.2.1 Perancangan Sistem

Perancangan sistem adalah tahapan lanjutan setelah tahap analisis sistem dalam alur pengembangan serta mendefinisikan setiap kebutuhan fungsional. Perancangan sistem dikerjakan setelah melakukan tahap analisis, karena setelah dilakukan kegiatan

analisis sistem seorang analis sistem sudah memiliki gambaran apa yang harus dikerjakan.

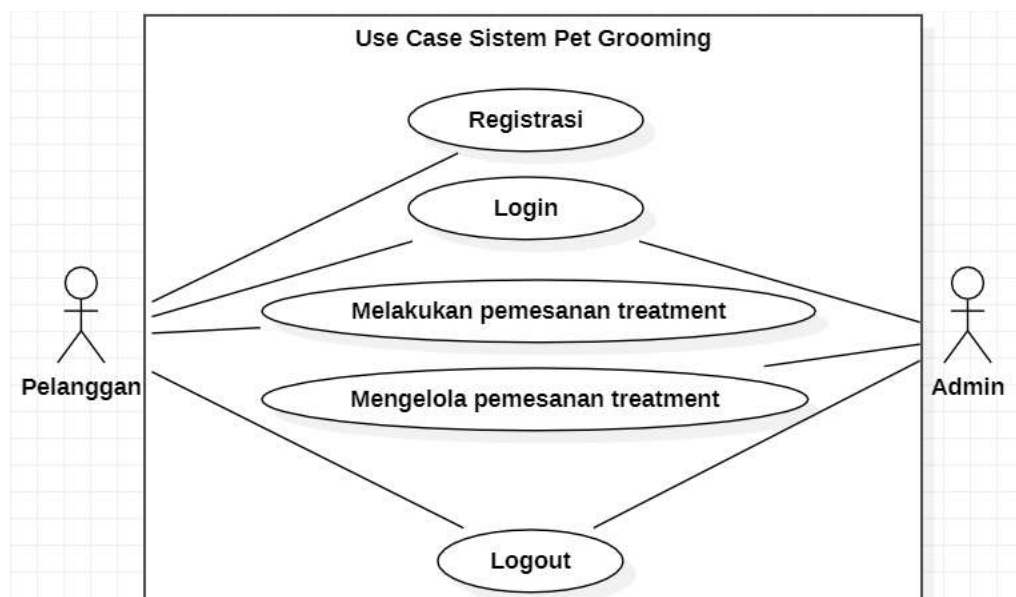
Metode yang digunakan dalam pembangunan Aplikasi *Pet Grooming* Berbasis *Website* ini adalah Metode Waterfall. Tahapan dari metode waterfall adalah sebagai berikut :



Gambar 3.6 Metode waterfall

3.2.1.1 Use Case Diagram

Use case diagram adalah diagram yang menunjukkan suatu kelompok use case dan aktor serta kaitan atau hubungannya



Gambar 3.7 Use Case Diagram

A. Definisi Aktor

Pada bagian ini akan dijelaskan aktor-aktor yang terlibat dalam Aplikasi *Pet Grooming*.

Tabel 3.7 Deskripsi use case

No	Aktor	Deskripsi
1.	Admin	<ul style="list-style-type: none"> • <i>Login</i> • Mengelola pemesanan <i>treatment</i> • <i>Logout</i>
2.	Pelanggan	<ul style="list-style-type: none"> • Register • <i>Login</i> • Melakukan pemesanan <i>treatment</i> • <i>Logout</i>

B. Definisi Use Case

Tabel 3.8 Definisi use case

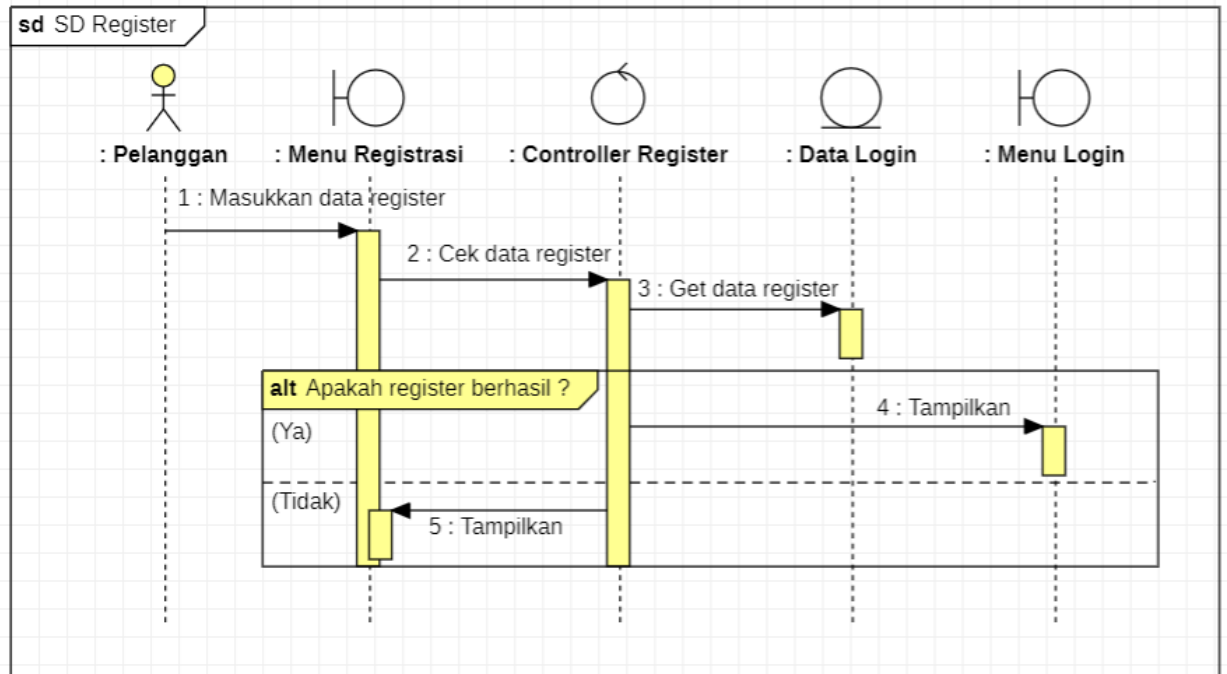
No.	Use Case	Definisi
1.	<i>Login</i>	Merupakan aktivitas <i>login</i> yang dilakukan oleh admin dan <i>user</i> . Sebelum admin dan <i>user</i> masuk ke aplikasi ia harus melakukan <i>login</i> terlebih dahulu.
2.	Register	Merupakan aktivitas pendaftaran untuk menambahkan data <i>login</i> untuk dapat masuk kedalam aplikasi.
3.	Mengelola pemesanan <i>treatment</i>	Merupakan aktivitas pengelolaan <i>treatment</i> yang dapat dilakukan oleh admin dapat menambahkan dan menghapus <i>treatment</i> .
4.	<i>Logout</i>	Merupakan aktivitas yang melibatkan admin untuk keluar dari akun pada <i>website</i> .
5.	Melakukan pemesanan <i>treatment</i>	Merupakan aktivitas untuk melakukan pemesanan <i>treatment</i> yang telah ditawarkan yang dilakukan oleh pelanggan.

3.2.1.2 Sequence Diagram

Sequence diagram di sini adalah untuk menggambarkan kolaborasi dinamis antara sejumlah objek, yang termasuk ke dalam sistem yang akan di bangun.

A. Sequence Diagram Register

Berikut ini merupakan Sequence Diagram register pada pelanggan.



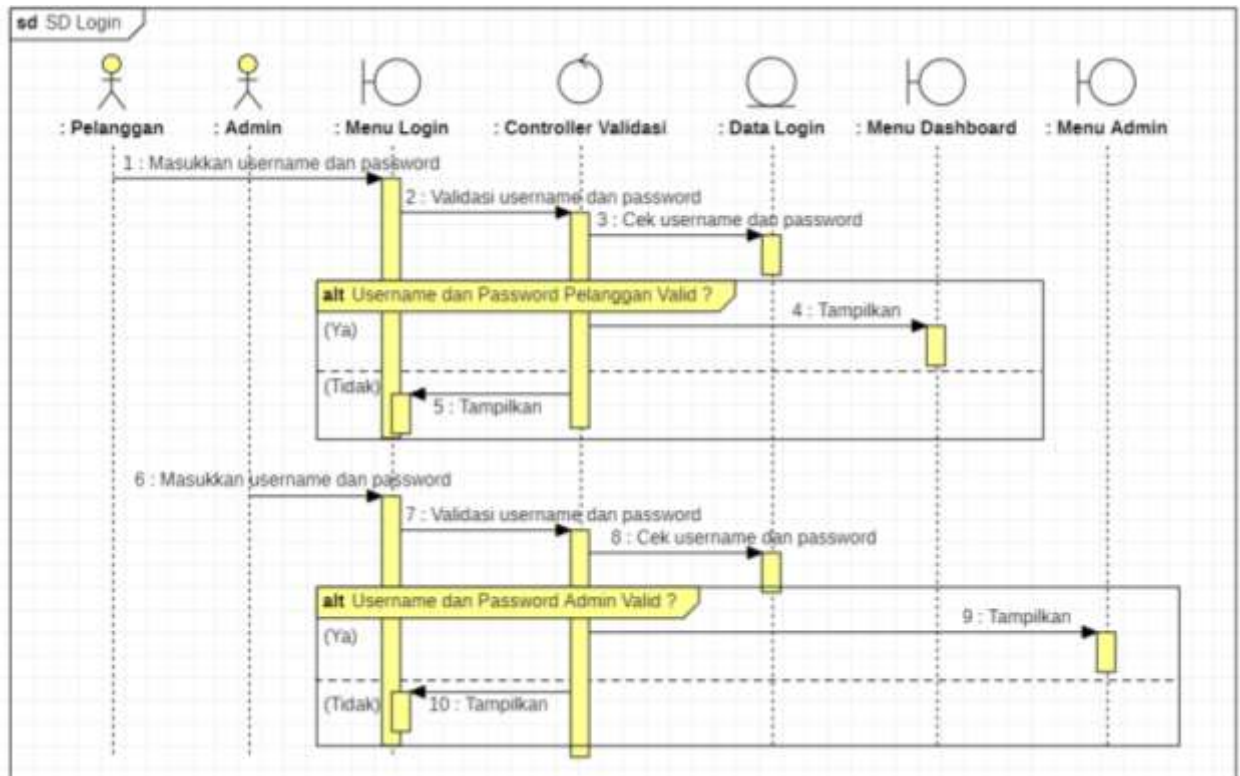
Gambar 3.8 Sequence diagram register

Keterangan :

1. Pelanggan masuk ke menu registrasi.
2. Controller register mengambil data register dari data pelanggan.
3. Jika register berhasil maka akan menampilkan halaman *login*.
4. Jika register tidak berhasil maka akan tetap berada di halaman registrasi.

B. Sequence Diagram Login

Berikut ini merupakan Sequence Diagram *login* pada admin dan pelanggan.



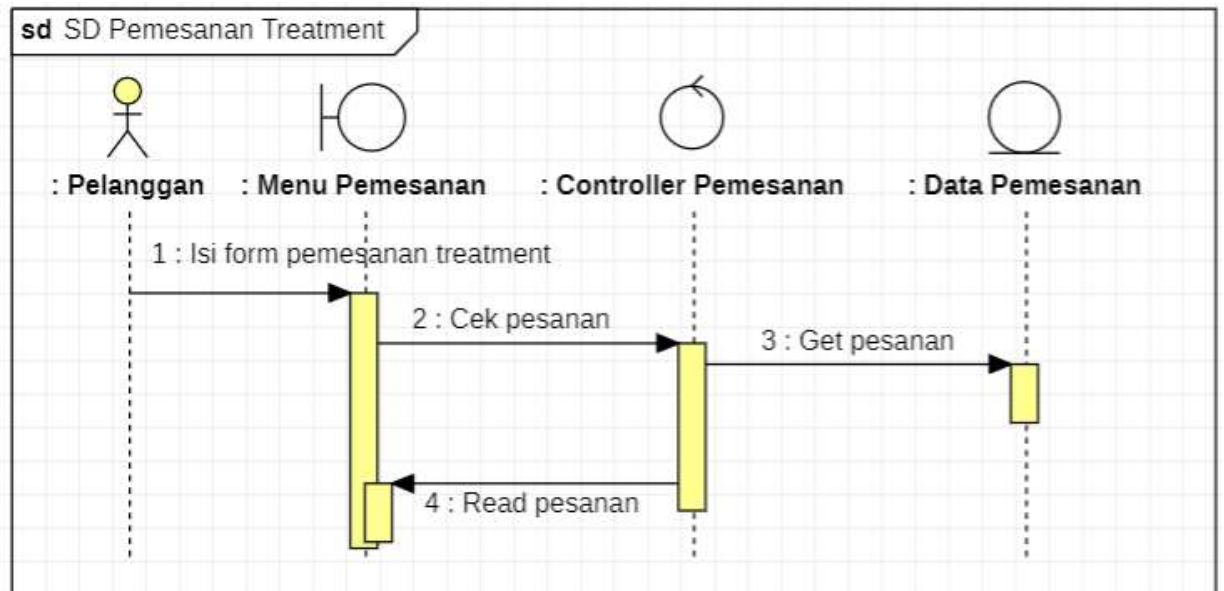
Gambar 3.9 Sequence diagram login

Keterangan :

1. Pelanggan dan admin masuk ke menu *login*.
2. Memvalidasi email dan password.
3. Controller *login* mengambil data email dan password dari data pelanggan dan admin.
4. Jika valid maka akan menampilkan halaman dashboard untuk pelanggan dan menampilkan halaman admin untuk admin.
5. Jika tidak valid maka akan tetap berada di halaman *login*.

C. Sequence Diagram Melakukan Pemesanan *Treatment*

Berikut ini merupakan Sequence Diagram pemesanan *treatment* pada pelanggan.



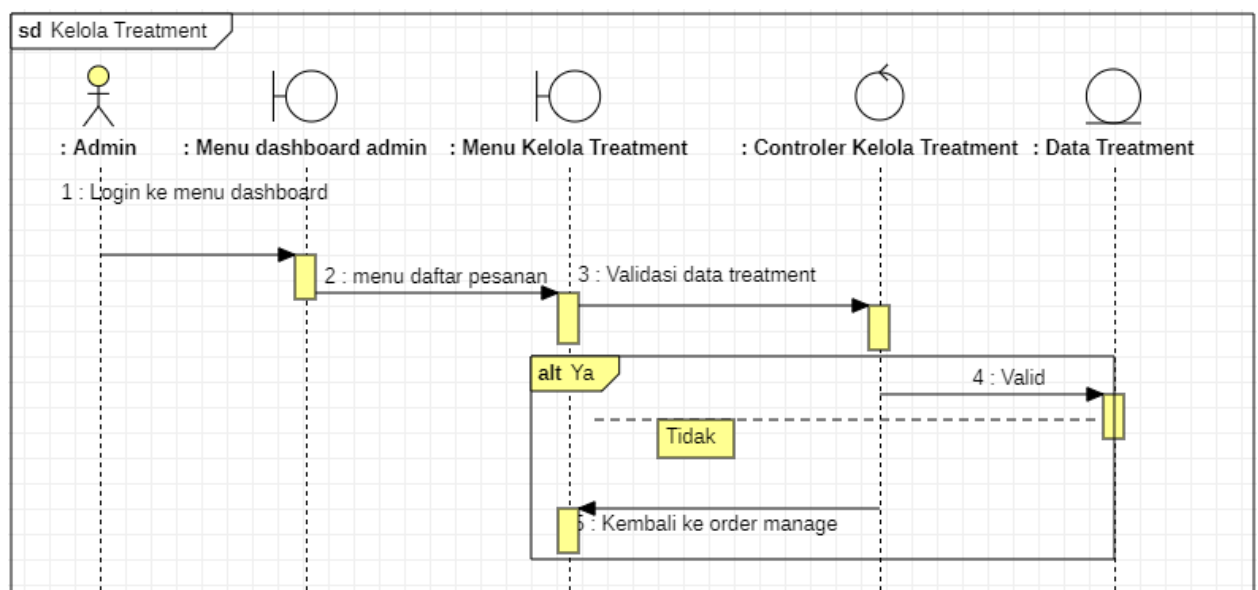
Gambar 3.10 Sequence diagram melakukan pemesanan treatment

Keterangan :

1. Pelanggan mengisi form pemesanan *treatment* pada menu pemesanan.
2. Controller mengecek pesanan dan mengambil data pesanan untuk data *treatment*.
3. Setelah data pesanan sudah masuk akan kembali ke menu pemesanan.

D. Sequence Diagram Mengelola Pemesanan *Treatment*

Berikut ini merupakan Sequence Diagram kelola *treatment* pada admin.



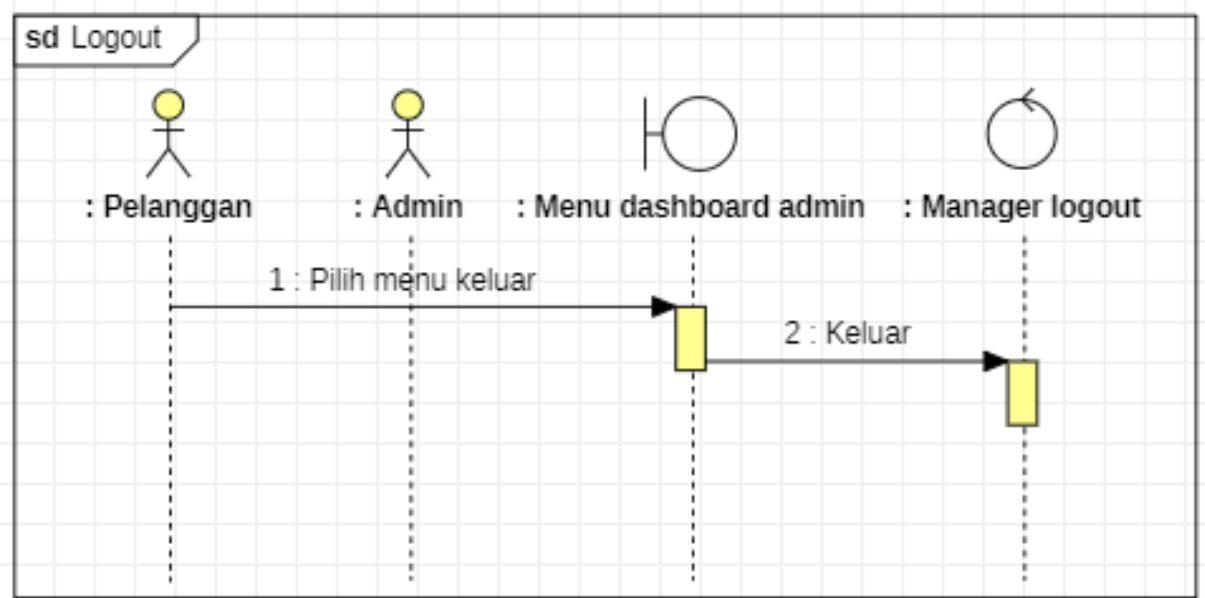
Gambar 3.11 Sequence diagram mengelola pemesanan treatment

Keterangan :

1. Admin *login* ke menu dashboard.
2. Admin memilih untuk melihat daftar pesanan *treatment* atau ingin CRUD *treatment*.
3. Controller memvalidasi data *treatment*.
4. Admin akan kembali ke menu kelola *treatment*.

E. Sequence Diagram Logout

Berikut ini merupakan Sequence Diagram *logout* pada pelanggan dan admin.



Gambar 3.12 Sequence diagram logout

Keterangan :

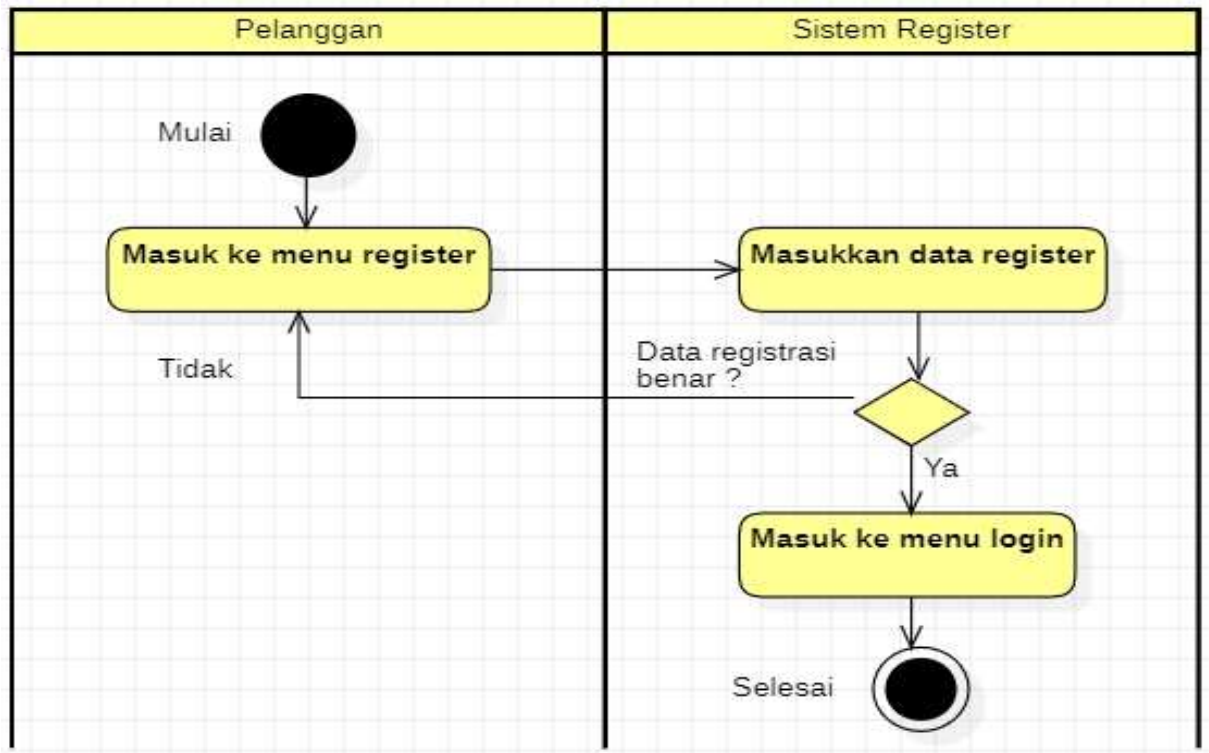
1. Pelanggan atau admin pilih menu keluar.
2. Manager *logout* akan menerima perintah keluar.

3.2.1.3 Activity Diagram

Activity diagram disini berfungsi untuk menggambarkan rangkaian aliran aktivitas yang ada di dalam sistem yang akan dibangun .

A. Activity Diagram Register

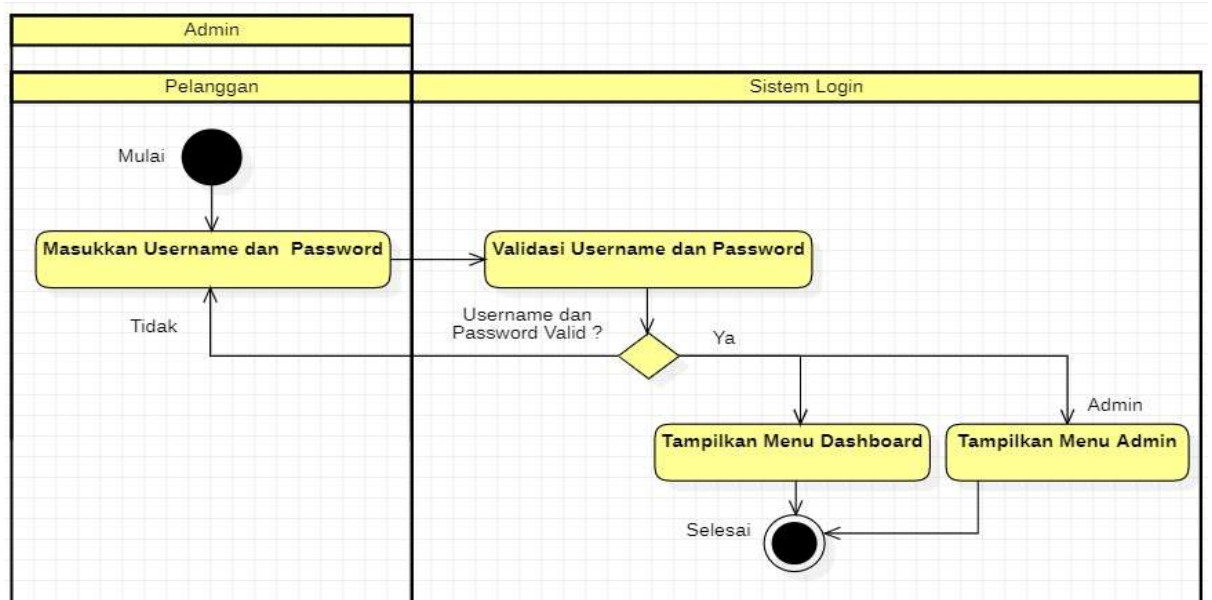
Berikut ini merupakan tampilan Activity Diagram Registrasi pada pelanggan.



Gambar 3.13 Activity diagram register

B. Activity Diagram Login

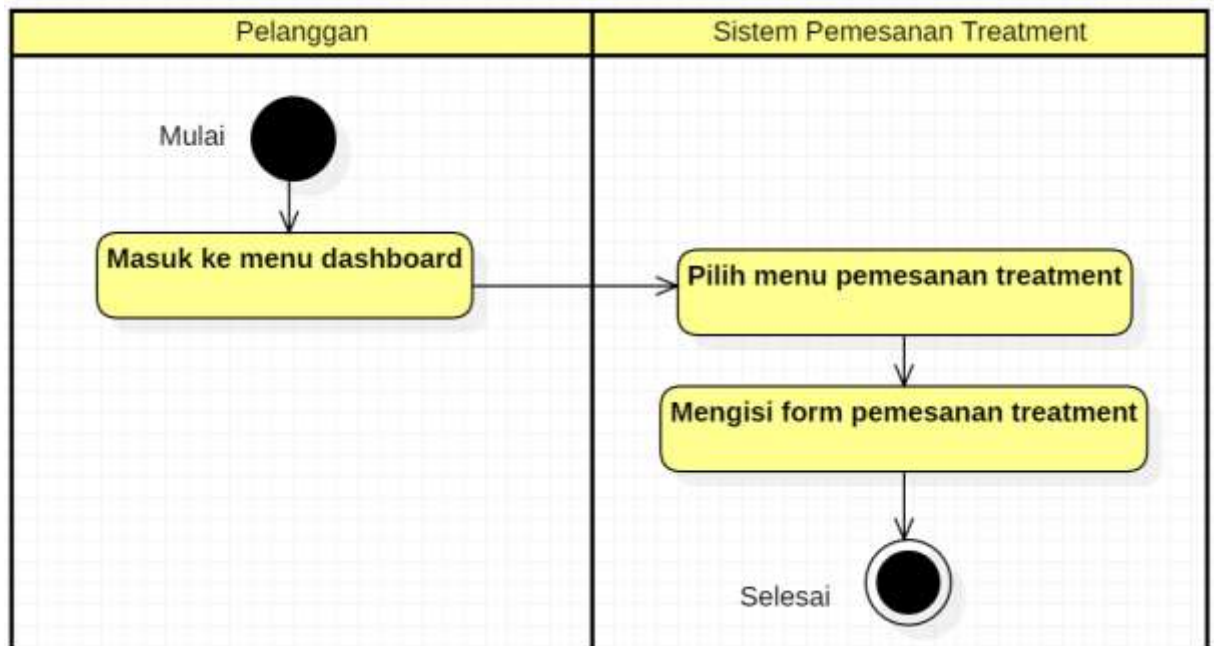
Berikut ini merupakan tampilan Activity Diagram *Login* pada admin dan pelanggan.



Gambar 3.14 Activity diagram login

C. Activity Diagram Melakukan Pemesanan *Treatment*

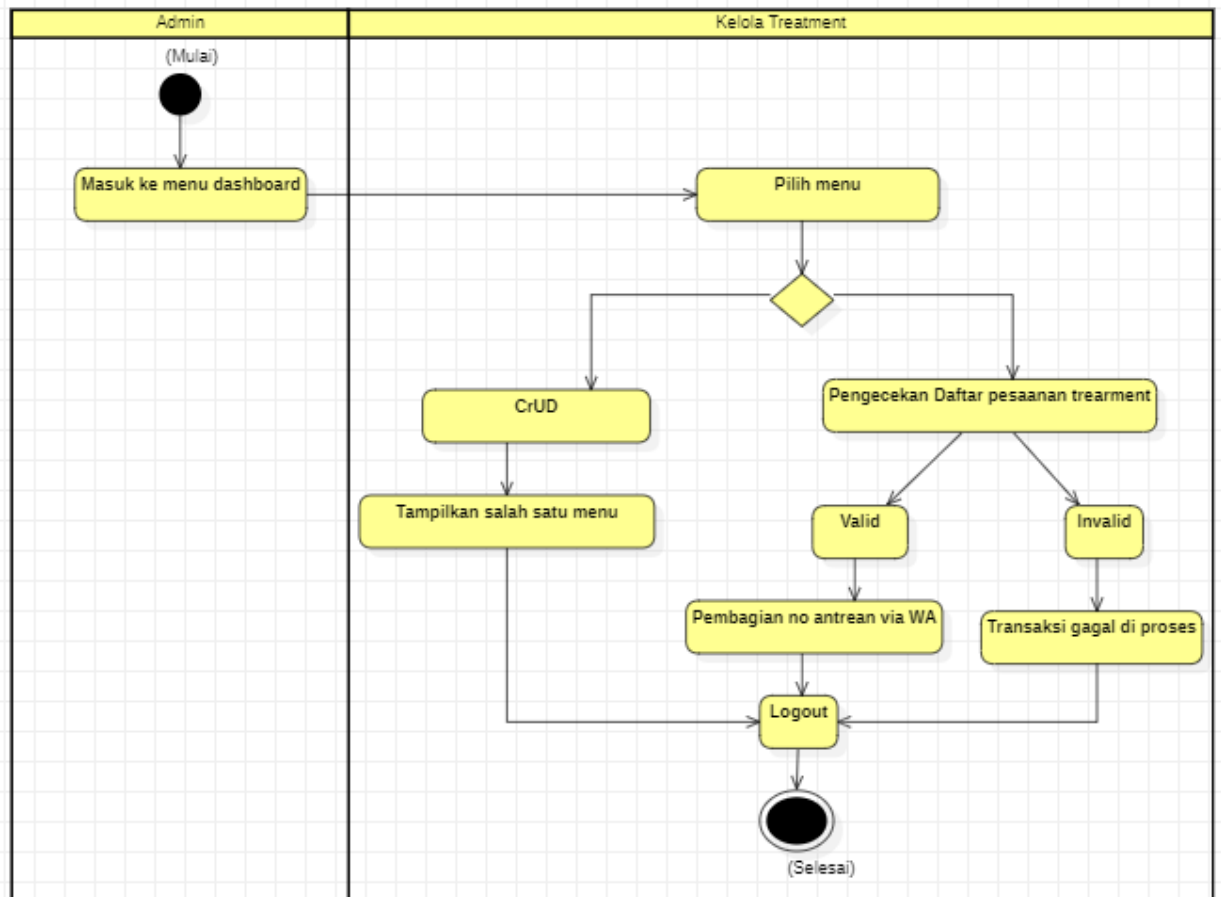
Berikut ini merupakan tampilan Activity Diagram Melakukan *Treatment* pada pelanggan.



Gambar 3.15 Activity diagram melakukan pemesanan treatment

D. Activity Diagram Kelola *Treatment*

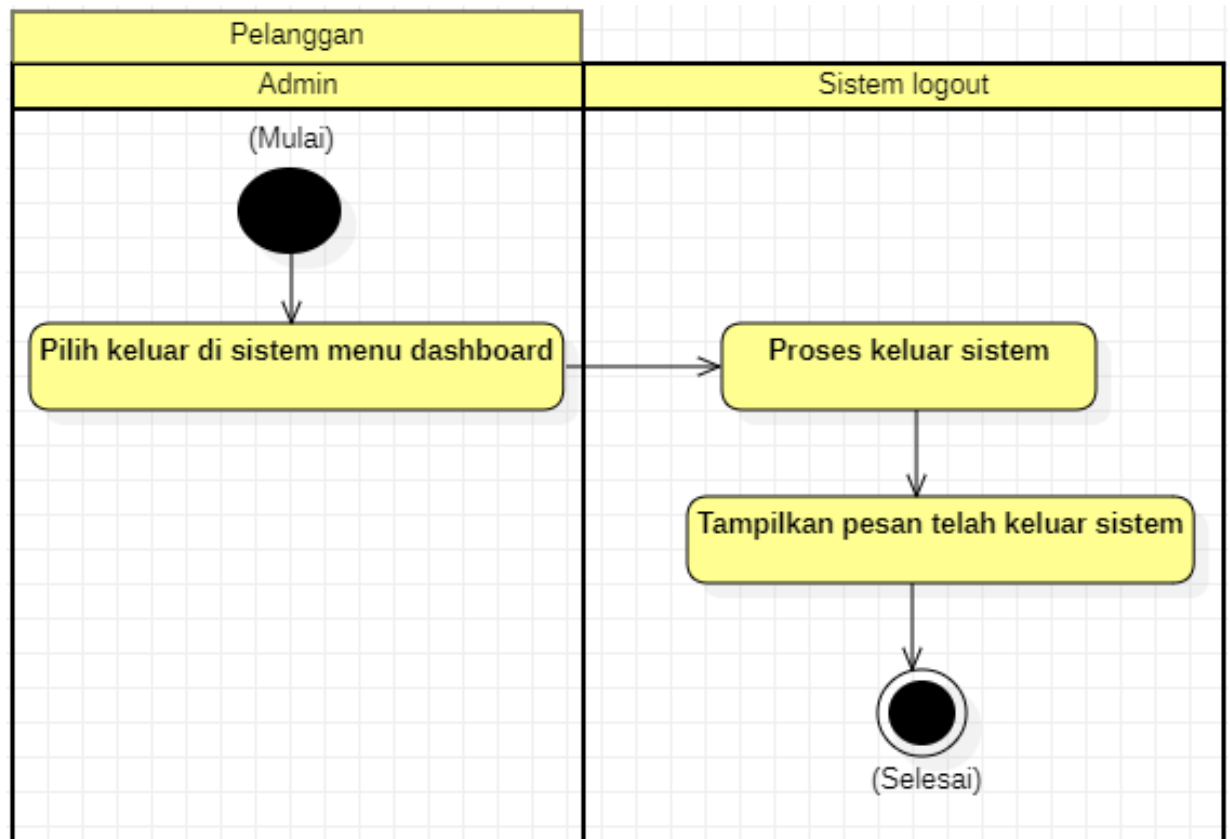
Berikut ini merupakan tampilan Activity Diagram Kelola *Treatment* pada admin.



Gambar 3.16 Activity diagram kelola treatment

E. Activity Diagram *Logout*

Berikut ini merupakan tampilan Activity Diagram *Logout* pada admin dan pelanggan.

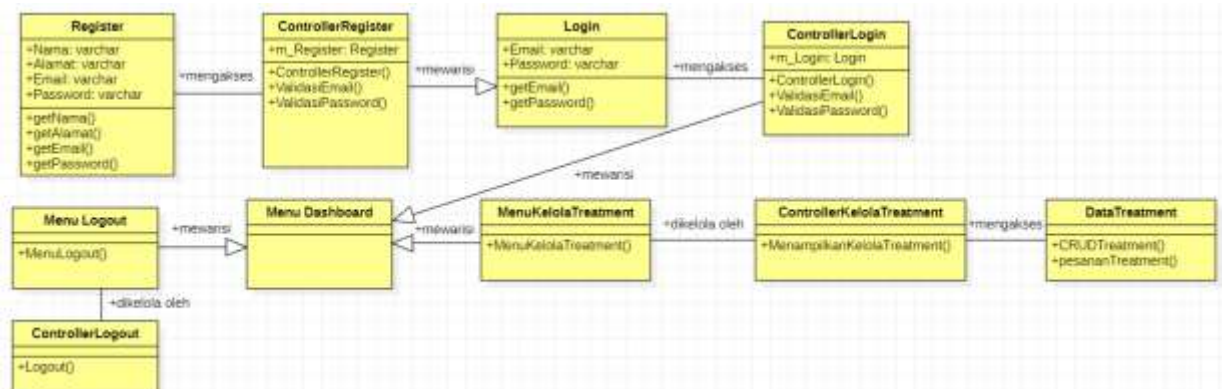


Gambar 3.17 Activity diagram logout

3.2.1.4 Class Diagram

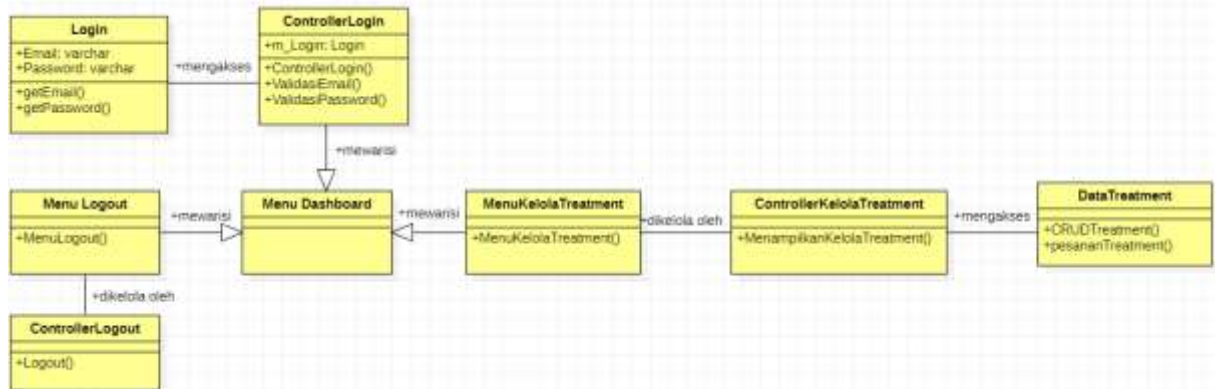
Class Diagram adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan hubungannya secara logika. Class diagram menggambarkan struktur statis dari sebuah sistem, karena itu class diagram merupakan tulang punggung atau kekuatan dasar dari setiap metode berorientasi objek termasuk UML.

A. Class Diagram Pelanggan



Gambar 3.18 Class diagram pelanggan

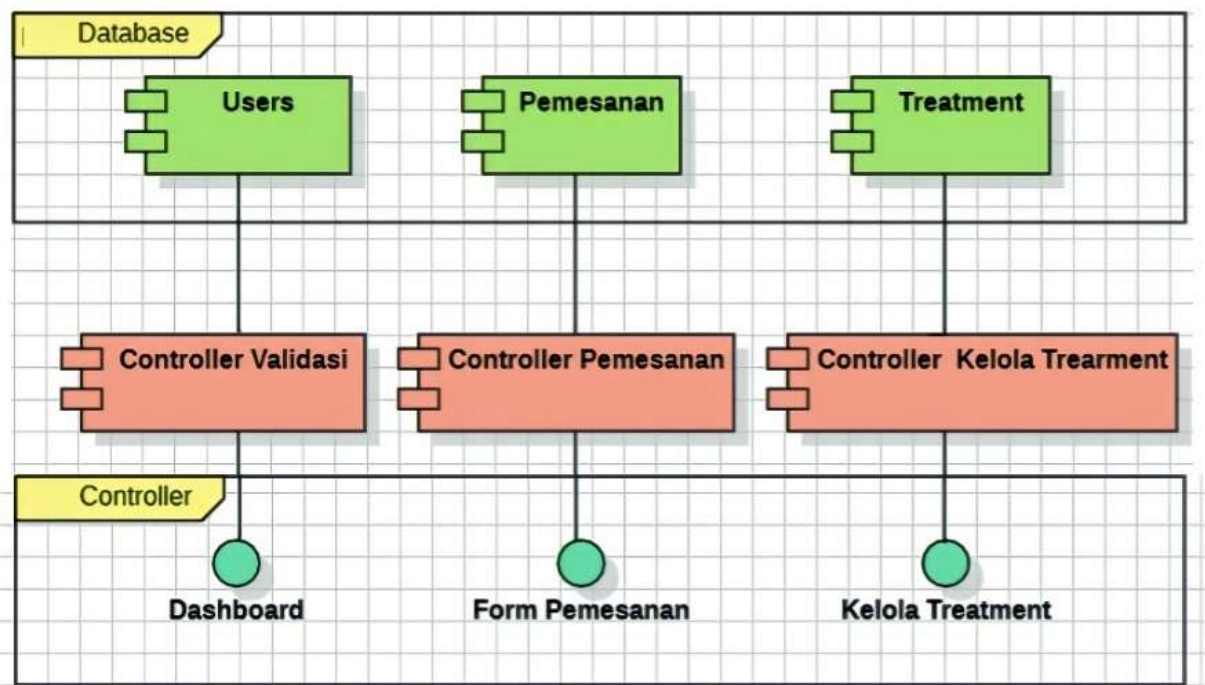
B. Class Diagram Admin



Gambar 3.19 Class diagram admin

3.2.1.5 Component Diagram

Diagram komponen menjelaskan struktur fisik code Anda dan memetakan tampilan logis kelas proyek Anda ke code aktual yang mengimplementasikan logika tersebut.

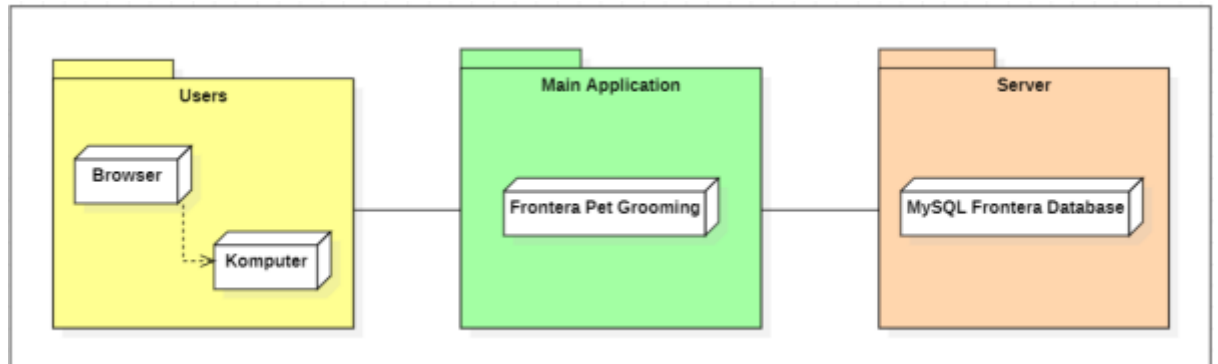


Gambar 3.20 Component diagram

3.2.1.6 Deployment Diagram

Deployment Diagram merupakan komponen dalam pengekskusion aplikasi, dengan menggambarkan detail komponen yang di deploy dalam infrastruktur aplikasi. Dimana package merupakan sebuah bungkusn dari satu atau lebih node. Sebuah node biasanya mengacu pada perangkat keras dan perangkat lunak yang tidak dibuat sendiri,

seperti browser. Kemudian hubungan antar node merupakan *depencey* atau ketergantungan.

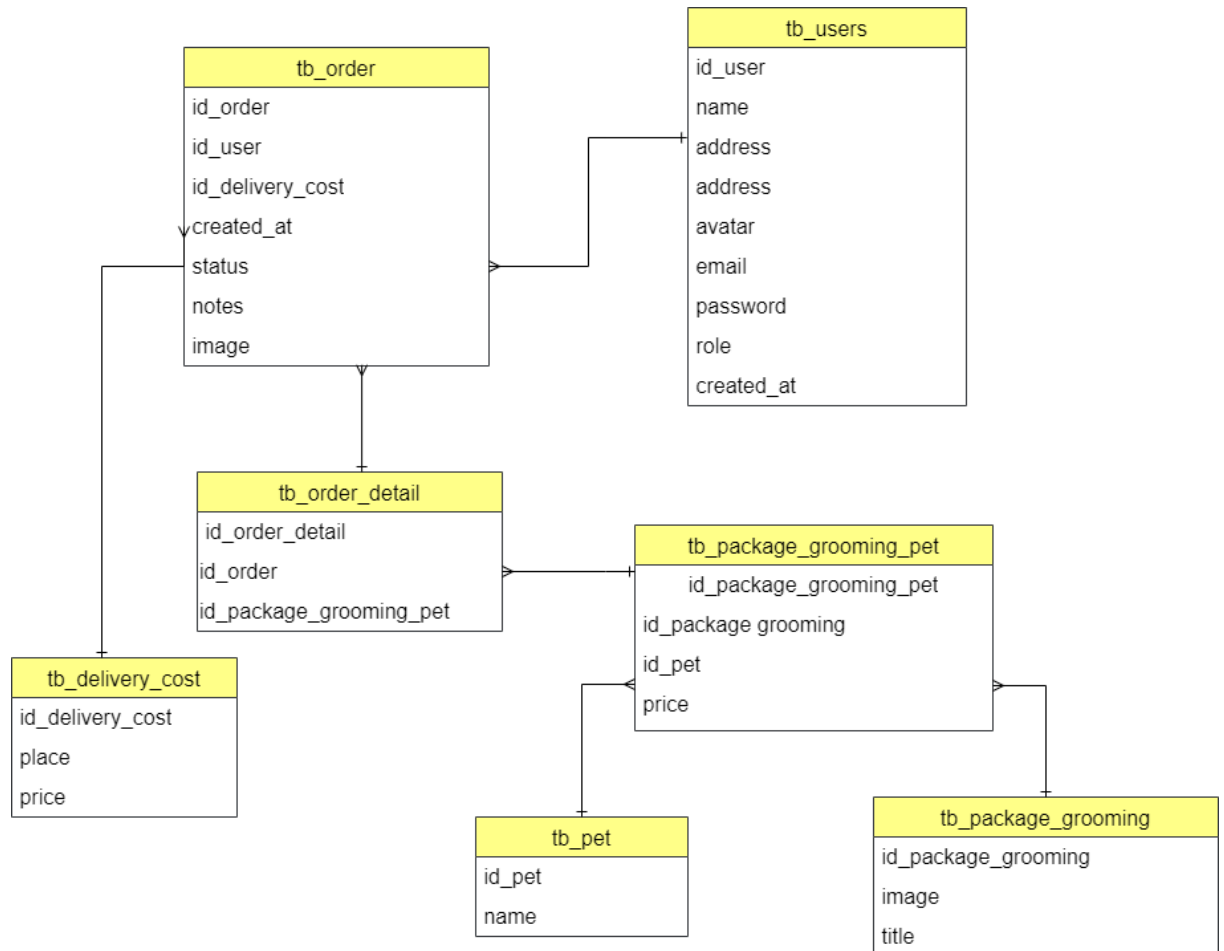


Gambar 3.21 Deployment diagram

3.2.1.7 Model Data

A. CDM (Conceptual Data Model)

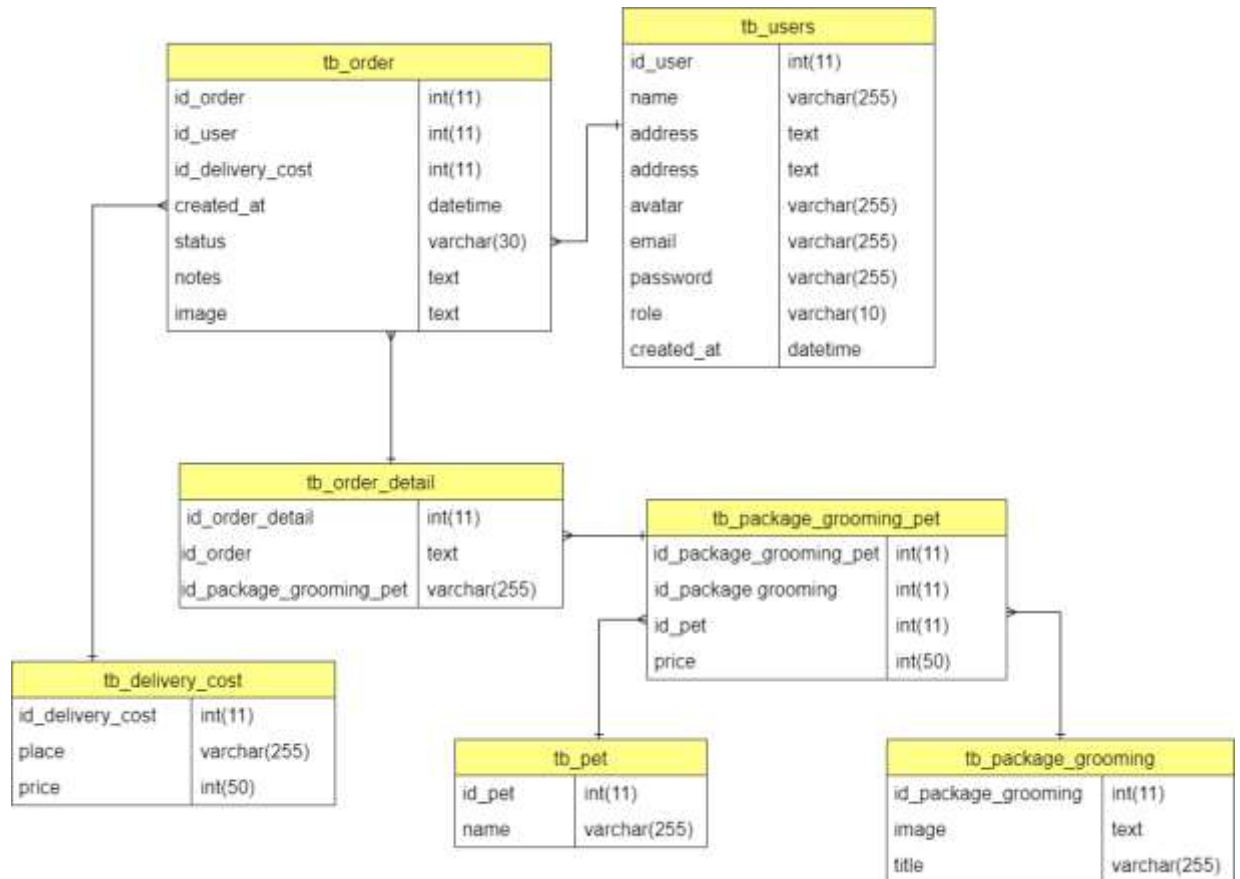
Conceptual Data Model atau dapat disebut CDM. CDM adalah memodelkan struktur logis dari keseluruhan aplikasi data, tidak tergantung pada pertimbangan model dari struktur data atau tergantung pada *software*. CDM yang benar dapat dikonversikan ke PDM.



Gambar 3.22 CDM (Conceptual Data model)

B. PDM (Physical Data Model)

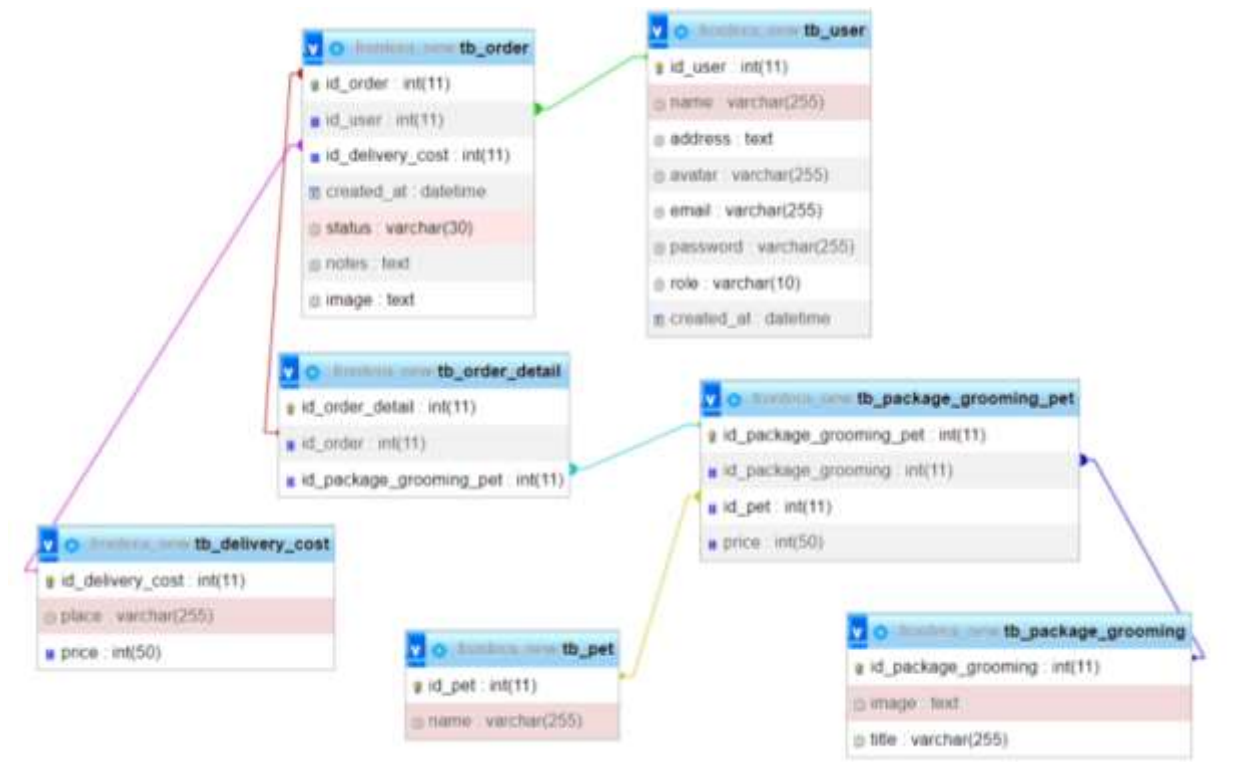
Physical Data Model atau dapat disebut PDM. PDM adalah bentuk fisik dari database yang dibuat dengan pertimbangan DBMS yang akan dipakai. PDM dapat dihasilkan dari CDM yang benar.



Gambar 3.23 PDM (Physical Data Model)

3.2.1.8 Perancangan Database

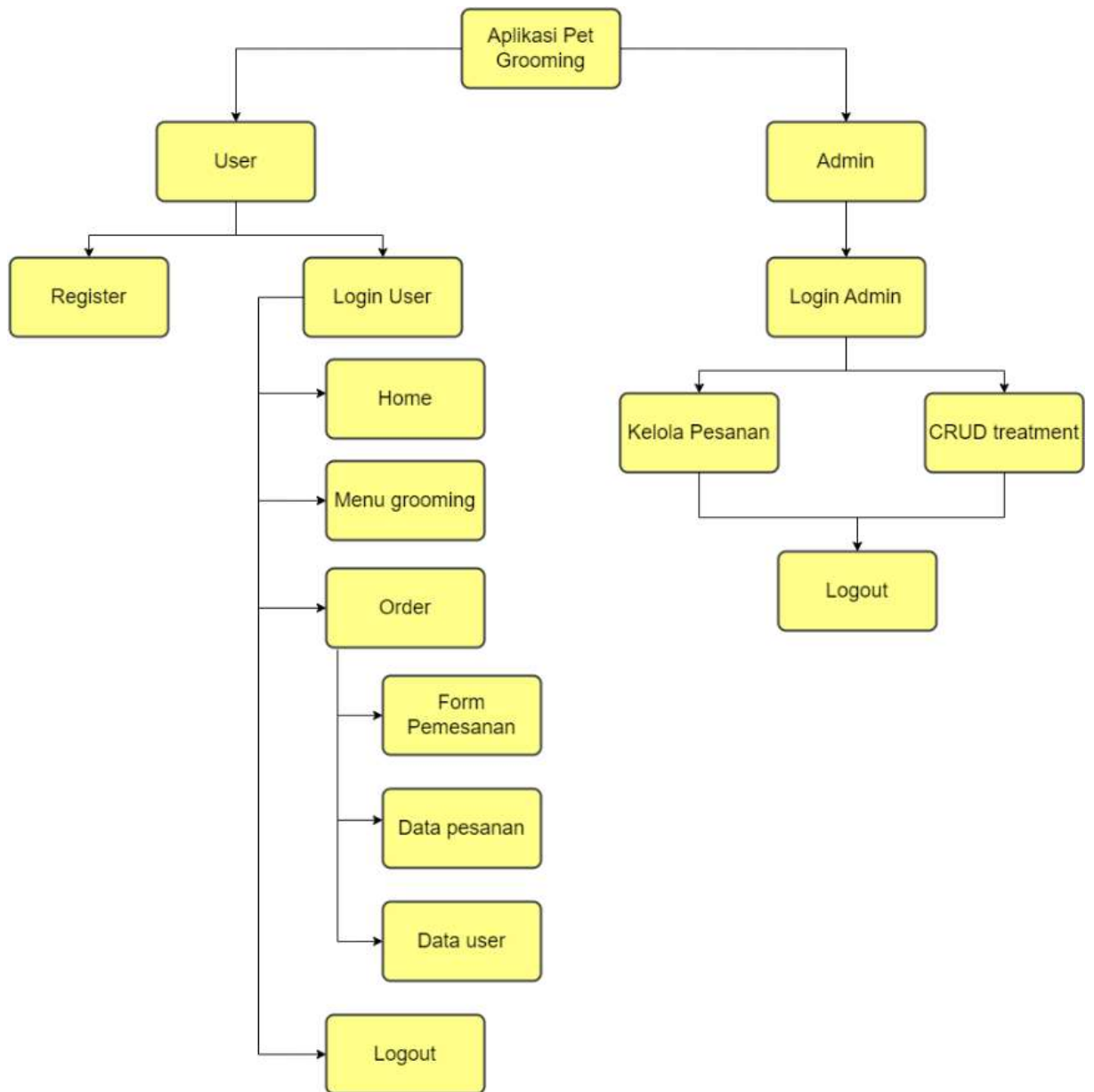
Perancangan database merupakan proses untuk menentukan dan pengaturan data yang dibutuhkan untuk menyimpan data data aplikasi, agar terciptanya pemrosesan data yang lebih efisien. Struktur tabel meliputi nama tabel, tipe data, nama atribut dan relasi antar table dengan primary dan foreign key.



Gambar 3.24 Perancangan database

3.2.3 Perancangan Struktur Menu

Struktur menu adalah bentuk pemrograman umum yang memudahkan pengguna untuk menjalankan program komputer. Sehingga *system* administrator saat menjalankan program tidak mengalami kesulitan dalam memilih menu yang diinginkan. Saat merancang perangkat lunak, menu dibuat seperti yang ditunjukkan pada gambar berikut:

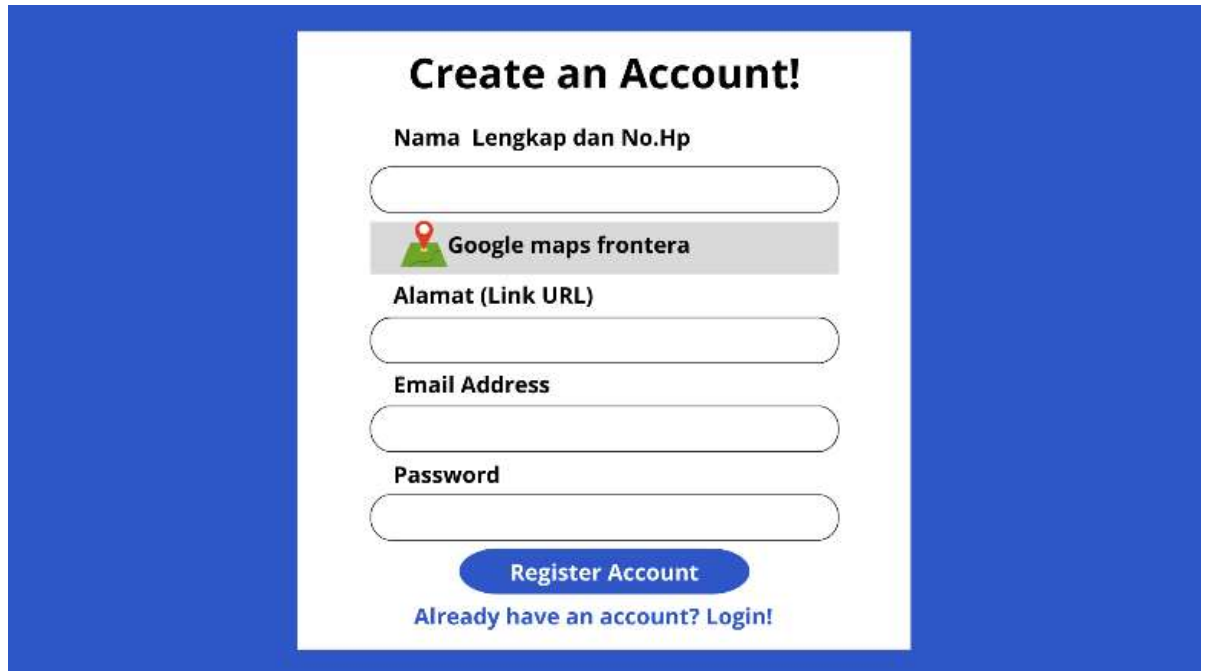


Gambar 3.25 Perancangan struktur mmenu

3.2.4 Perancangan UI

Interface atau antarmuka merupakan suatu media yang digunakan untuk komunikasi antara manusia (*user*) dengan komputer, oleh sebab itu aplikasi ini di rancang agar *user* dapat dengan mudah menggunakannya atau mengoperasikannya.

3.2.4.1 Perancangan UI Halaman Register



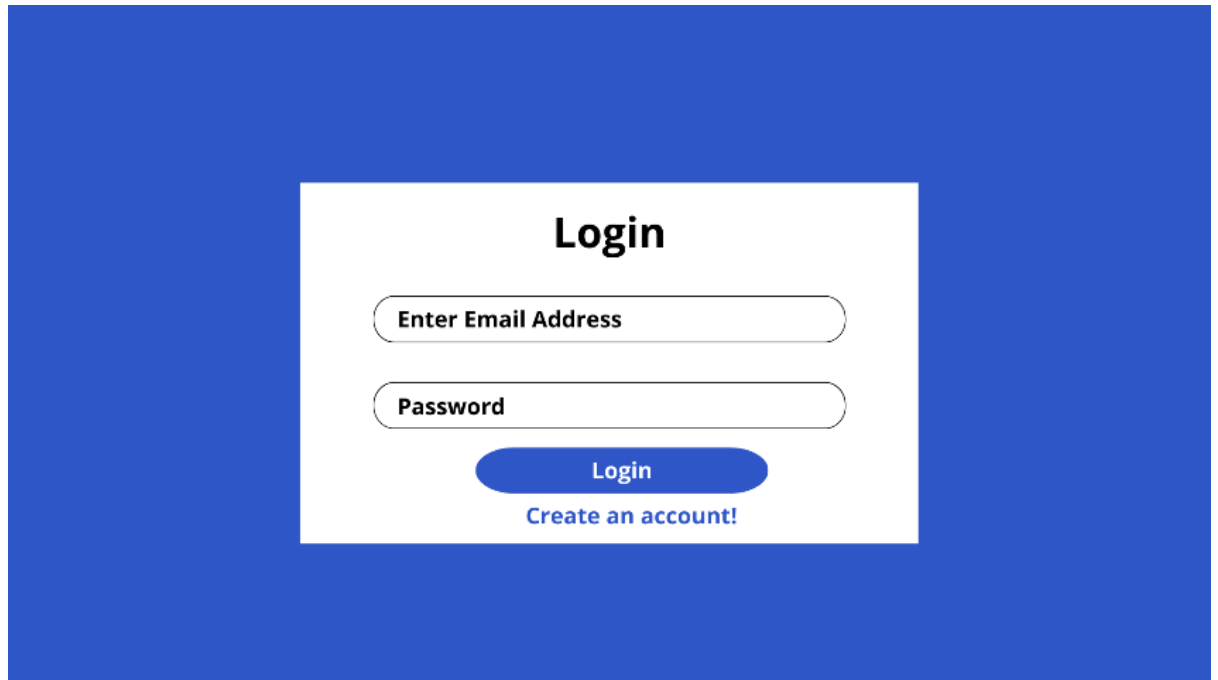
The image shows a registration form titled "Create an Account!" centered on a blue background. The form is white and contains the following elements:

- Title:** "Create an Account!" in bold black text.
- Form Fields:**
 - Nama Lengkap dan No.Hp:** A single-line text input field.
 - Google maps frontera:** A button with a red location pin icon and the text "Google maps frontera".
 - Alamat (Link URL):** A single-line text input field.
 - Email Address:** A single-line text input field.
 - Password:** A single-line text input field.
- Buttons:**
 - Register Account:** A blue button with white text.
 - Already have an account? Login!:** A blue link text.

Gambar 3.26 Perancangan UI halaman registrasi

Gambar diatas merupakan Perancangan UI dari halaman register. Pada halaman register terdapat form yang harus diisi yaitu nama lengkap dan no. hp, alamat, email, dan password. Terdapat *button register account* yang berfungsi untuk menyimpan data regsiter.

3.2.4.2 Perancangan UI Halaman *Login*

The image shows a login form centered on a solid blue background. The form is a white rectangle with a black border. At the top of the form, the word "Login" is written in a bold, black, sans-serif font. Below this, there are two input fields. The first field is a rounded rectangle with a light gray border and the placeholder text "Enter Email Address" in a small, gray, sans-serif font. The second field is similar, with the placeholder text "Password" in the same font. Below the password field is a blue, rounded rectangular button with the word "Login" in white, sans-serif font. At the bottom of the form, the text "Create an account!" is written in a small, blue, sans-serif font.

Gambar 3.27 Perancangan UI halaman login

Gambar diatas merupakan Perancangan UI dari halaman *login*. Halaman *login* berisi form yang harus di isi yaitu email dan password yang sudah didaftarkan pada saat register. Terdapat *button login* yang berfungsi untuk mengarahkan *user* ke halaman home.

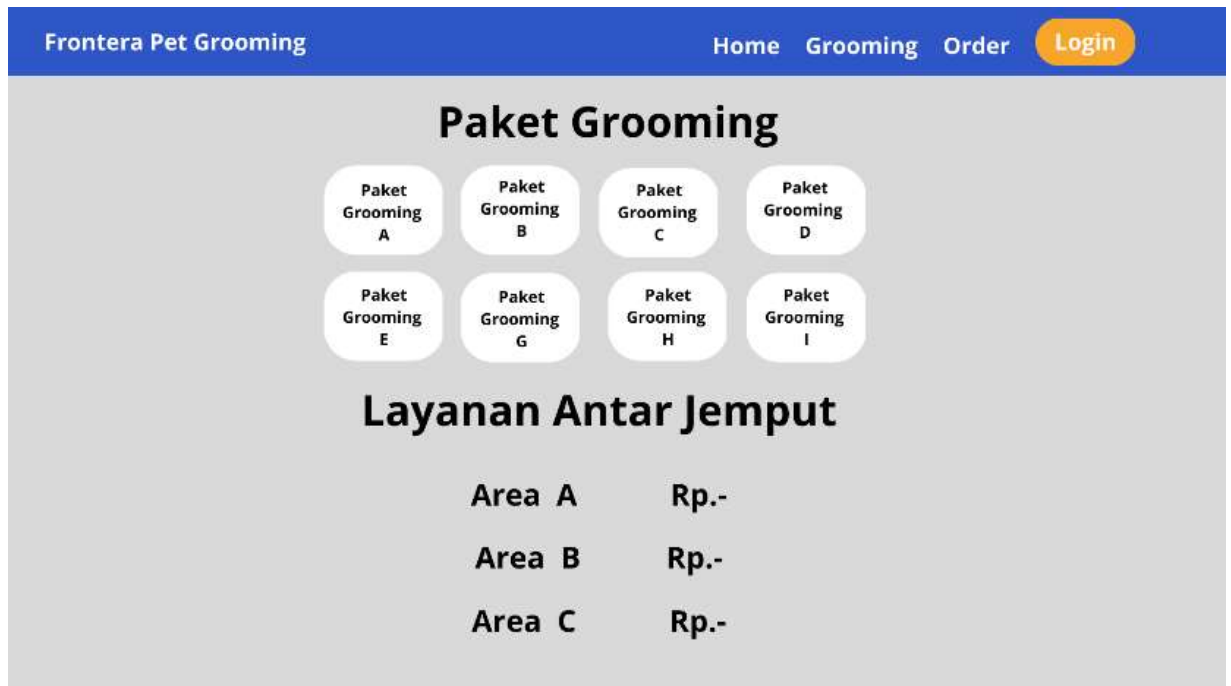
3.2.4.2 Perancangan UI Halaman Home



Gambar 3.28 Perancangan UI halaman home

Gambar diatas merupakan Perancangan UI dari halaman home. Pada navbar terdapat menu home, *grooming*, dan *order* serta *button login*. Pada halaman home ini berisi informasi mengenai *Frontera Pet Grooming*.

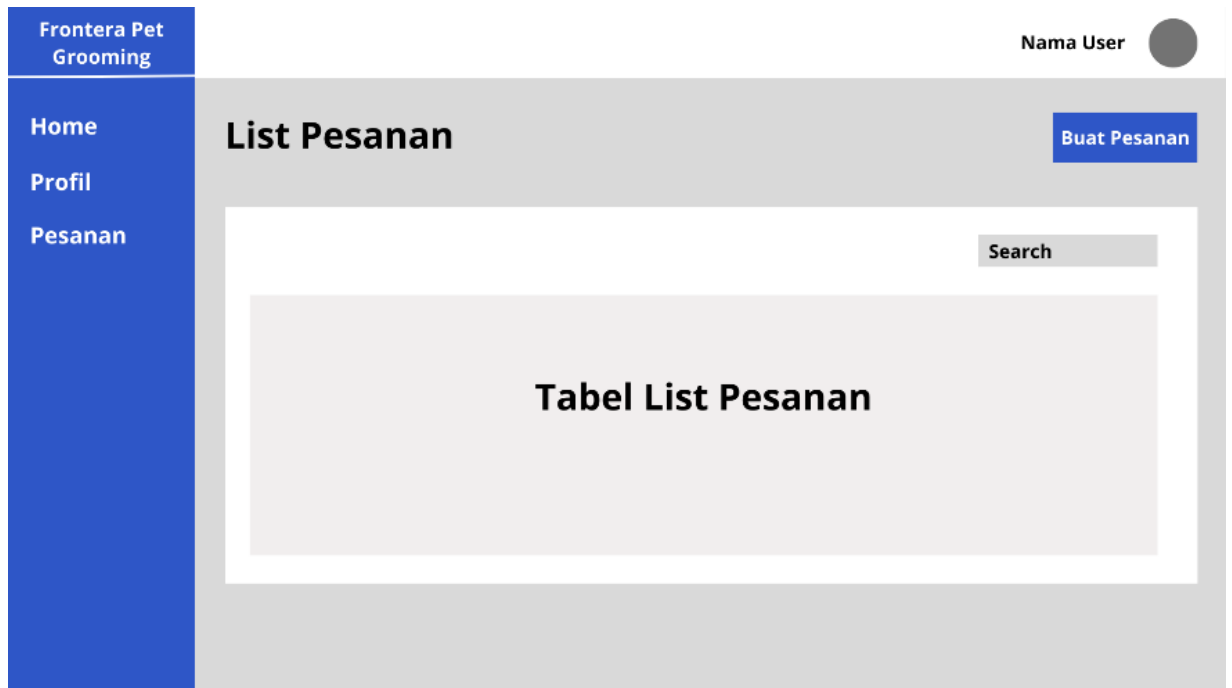
3.2.4.3 Perancangan UI Halaman *Grooming*



Gambar 3.29 Perancangan UI halaman grooming

Gambar diatas merupakan Perancangan UI dari halaman *grooming*. Pada navbar terdapat menu *home*, *grooming*, dan *order* serta *button login*. Halaman *grooming* berisi tentang katalog layanan *grooming* yang ditawarkan serta harganya dan berisi harga dari layanan antar jemput.

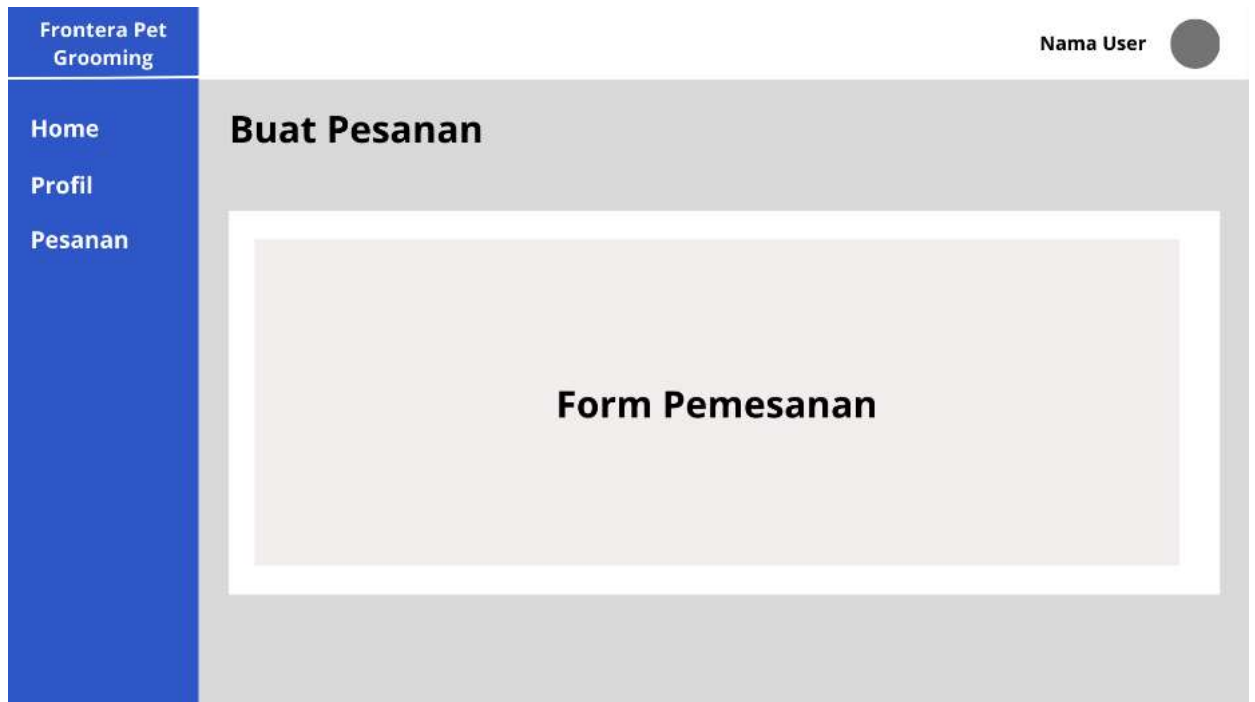
3.2.4.4 Perancangan UI Halaman Order



Gambar 3.30 Perancangan UI halaman order

Gambar diatas merupakan Perancangan UI dari halaman *order*. Pada sidebar terdapat menu home, profil dan pesanan. Terdapat *button* buat pesanan yang berfungsi untuk masuk ke halaman form buat pesanan. Terdapat menu *search* serta tabel list pesanan.

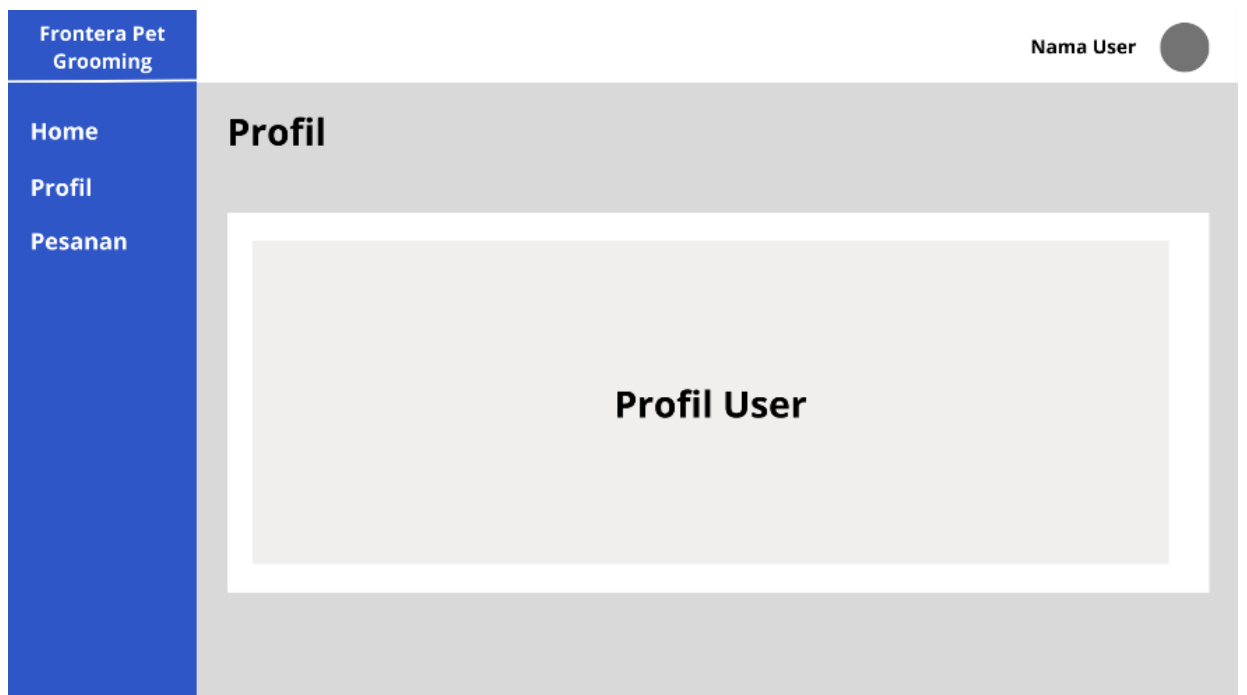
3.2.4.5 Perancangan UI Halaman Buat Pesanan



Gambar 3.31 Perancangan UI halaman buat pesanan

Gambar diatas merupakan Perancangan UI dari halaman buat pesanan. Pada sidebar terdapat menu home, profil dan pesanan. Terdapat form pemesanan.

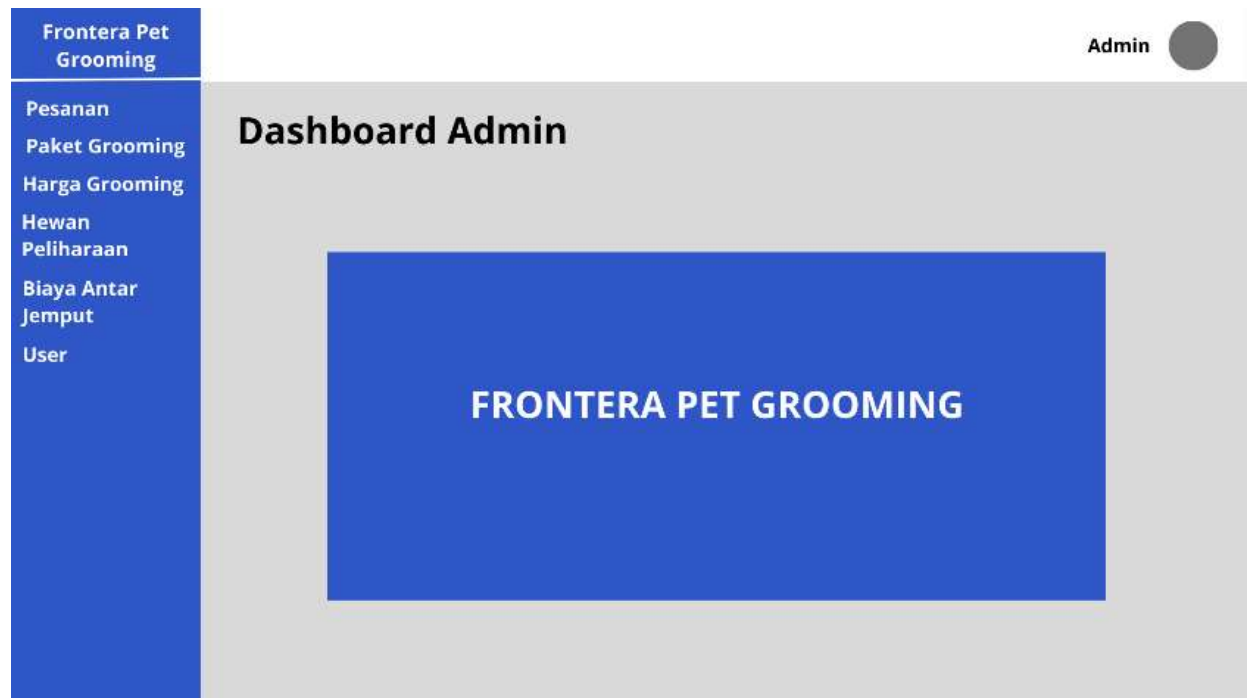
3.2.4.6 Perancangan UI Halaman Profil



Gambar 3.32 Perancangan UI halaman profil

Gambar diatas merupakan Perancangan UI dari halaman profil. Pada sidebar terdapat menu home, profil dan pesanan. Terdapat tabel profil *user*.

3.2.4.7 Perancangan UI Halaman Dashboard Admin



Gambar 3.33 Perancangan UI halaman dashboard admin

Gambar diatas merupakan Perancangan UI dari halaman dashboard admin. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Serta menampilkan foto Frontera Pet *Grooming*.

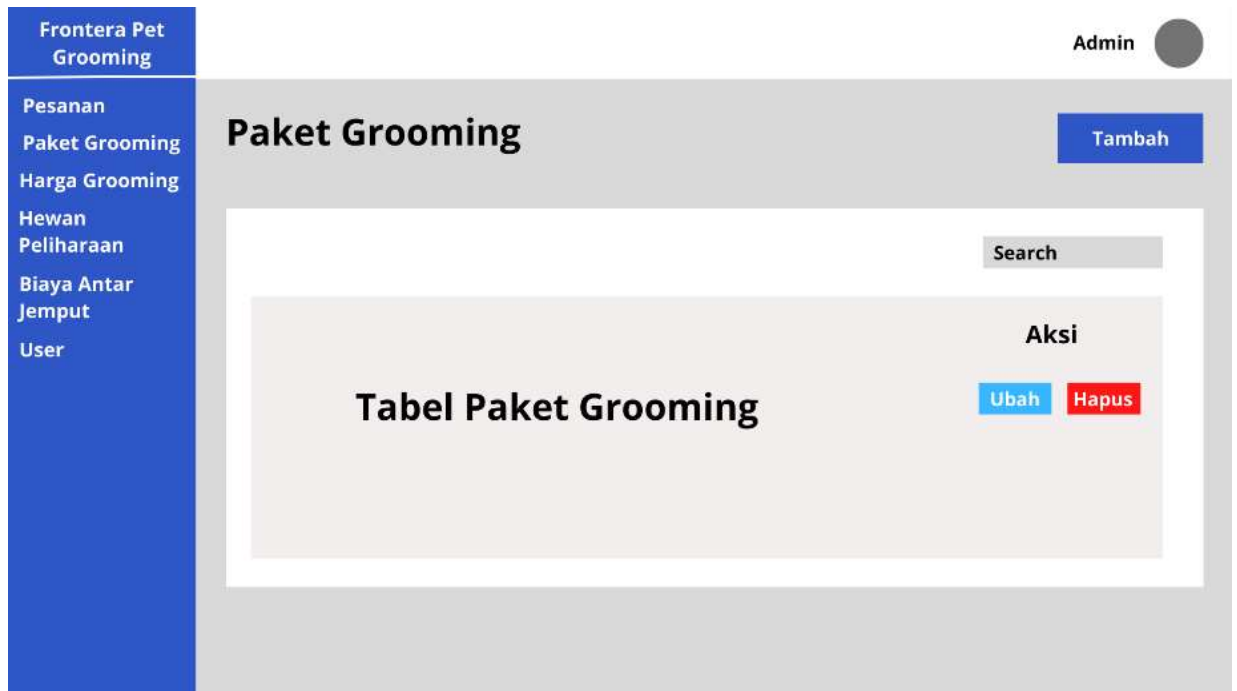
3.2.4.8 Perancangan UI Halaman Pesanan Admin



Gambar 3.34 Perancangan UI halaman pesanan admin

Gambar diatas merupakan Perancangan UI dari halaman pesanan admin. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat *button* buat pesanan, menu *search*, dan tabel list pesanan.

3.2.4.9 Perancangan UI Halaman Paket *Grooming* Admin



Gambar 3.35 Perancangan UI halaman paket grooming admin

Gambar diatas merupakan Perancangan UI dari halaman paket *grooming*. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat *button* tambah, menu *search*, dan pada tabel paket *grooming* yang didalamnya terdapat *button* aksi ubah dan hapus.

3.2.4.10 Perancangan UI Halaman Tambah Paket *Grooming* Admin



Gambar 3.36 Perancangan UI halaman tambah paket grooming admin

Gambar diatas merupakan Perancangan UI dari tambah paket *grooming*. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat form tambah paket *grooming*.

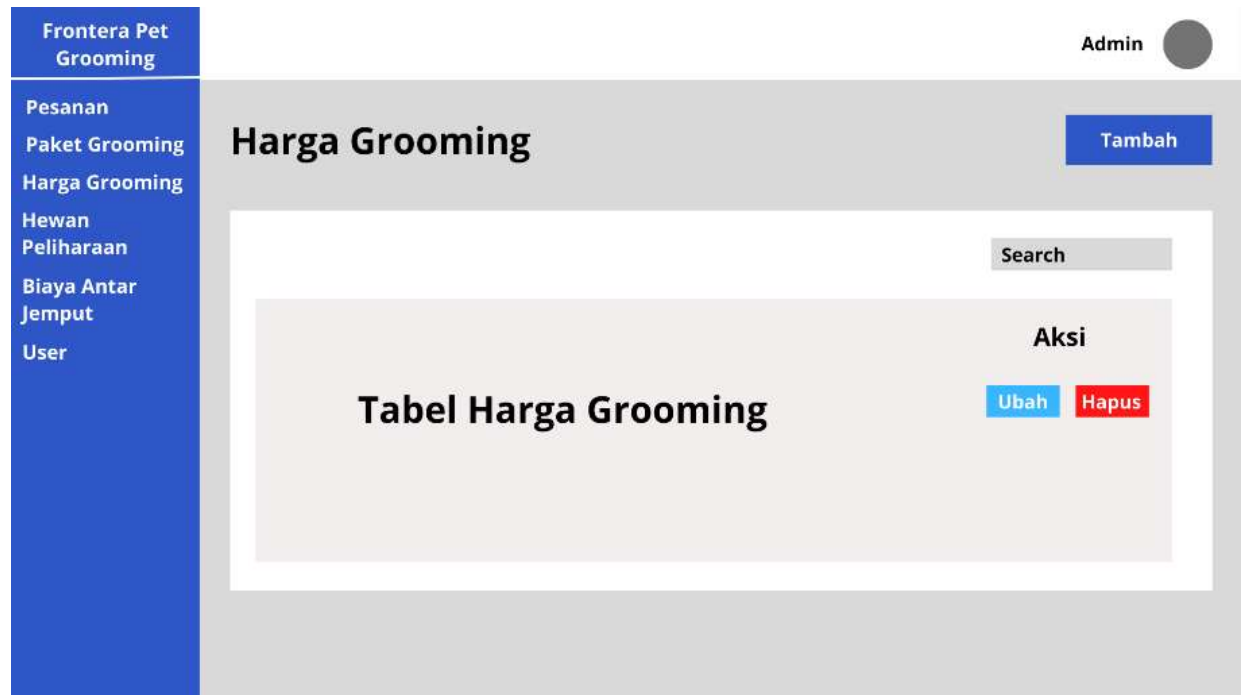
3.2.4.11 Perancangan UI Halaman Ubah Paket *Grooming* Admin



Gambar 3.37 Perancangan UI halaman ubah paket grooming admin

Gambar diatas merupakan Perancangan UI dari ubah paket *grooming*. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat form ubah paket *grooming*.

3.2.4.12 Perancangan UI Halaman Harga *Grooming* Admin



Gambar 3.38 Perancangan UI halaman harga grooming admin

Gambar diatas merupakan Perancangan UI dari halaman harga *grooming*. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat *button* tambah, menu *search*, dan pada tabel harga *grooming* yang didalamnya terdapat *button* aksi ubah dan hapus.

3.2.4.13 Perancangan UI Halaman Tambah Harga *Grooming* Admin



Gambar 3.39 Perancangan UI halaman tambah harga grooming admin

Gambar diatas merupakan Perancangan UI dari tambah harga *grooming*. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat form tambah harga *grooming*.

3.2.4.14 Perancangan UI Halaman Ubah Harga *Grooming* Admin



Gambar 3.40 Perancangan UI halaman ubah harga grooming admin

Gambar diatas merupakan Perancangan UI dari ubah harga *grooming*. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat form ubah harga *grooming*.

3.2.4.15 Perancangan UI Halaman Hewan Peliharaan Admin



Gambar 3.41 Perancangan UI halaman hewan peliharaan admin

Gambar diatas merupakan Perancangan UI dari halaman hewan peliharaan. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat *button* tambah, menu *search*, dan pada hewan peliharaan yang didalamnya terdapat *button* aksi ubah dan hapus.

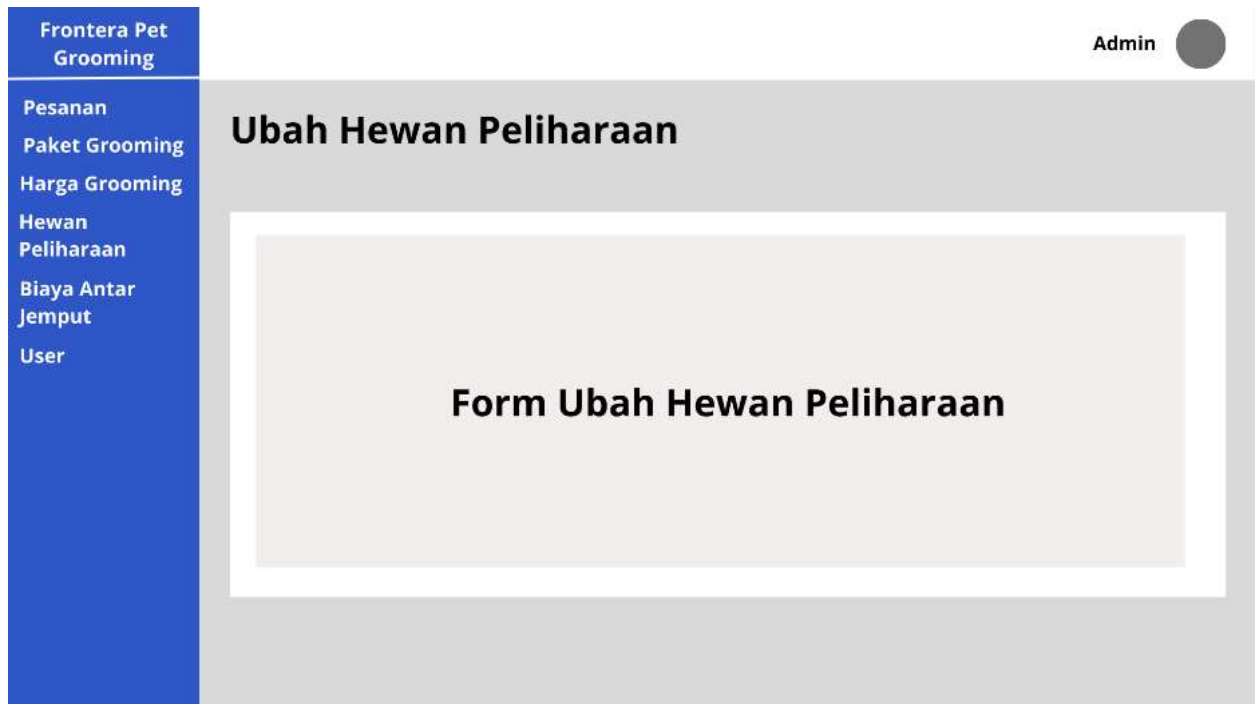
3.2.4.16 Perancangan UI Halaman Tambah Hewan Peliharaan Admin



Gambar 3.42 Perancangan UI halaman tambah hewan peliharaan admin

Gambar diatas merupakan Perancangan UI dari tambah hewan peliharaan. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat form tambah hewan peliharaan.

3.2.4.17 Perancangan UI Halaman Ubah Hewan Peliharaan Admin



Gambar 3. 43 Perancangan UI halaman ubah hewan peliharaan admin

Gambar diatas merupakan Perancangan UI dari ubah hewan peliharaan. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat form ubah hewan peliharaan.

3.2.4.18 Perancangan UI Halaman Biaya Antar Jemput Admin



Gambar 3.44 Perancangan UI halaman biaya antar jemput

Gambar diatas merupakan Perancangan UI dari biaya antar jemput. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat *button* tambah, menu *search*, dan pada tabel biaya antar jemput yang didalamnya terdapat *button* aksi ubah dan hapus.

3.2.4.19 Perancangan UI Halaman Tambah Biaya Antar Jemput Admin



Gambar 3.45 Perancangan UI halaman tambah biaya antar jemput admin

Gambar diatas merupakan Perancangan UI dari tambah layanan antar jemput. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat form tambah biaya antar jemput.

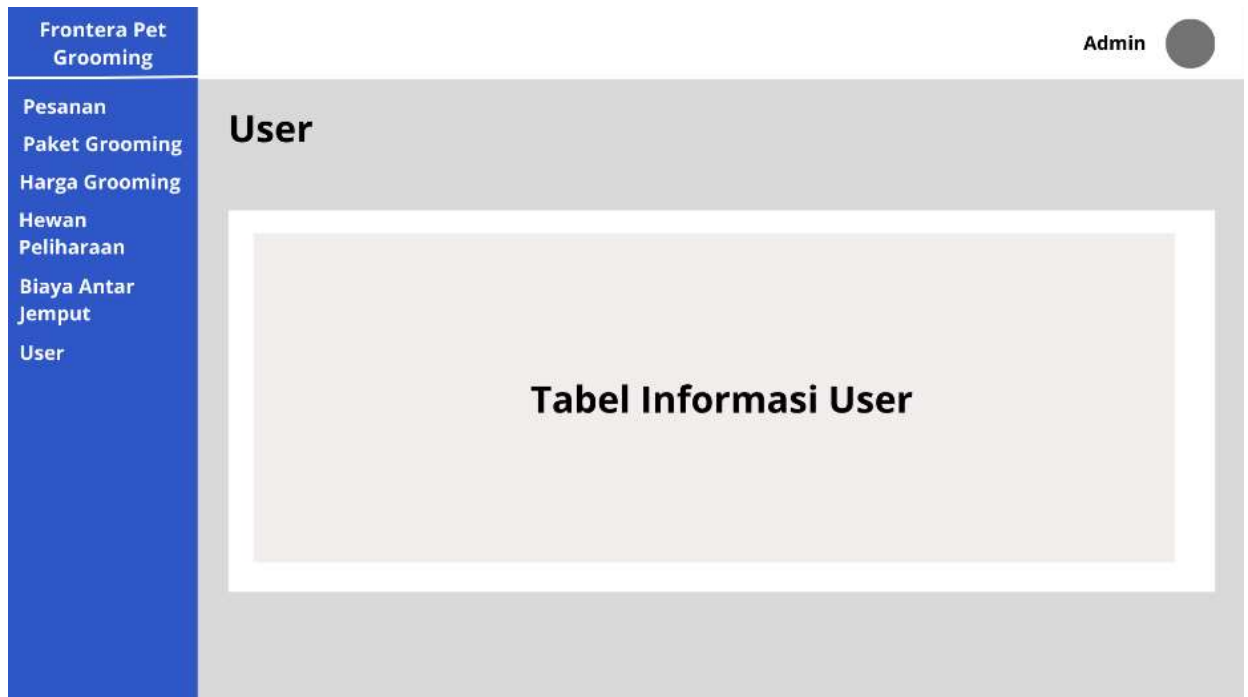
3.2.4.20 Perancangan UI Halaman Ubah Biaya Antar Jemput Admin



Gambar 3.46 Perancangan UI halaman ubah biaya antar jemput admin

Gambar diatas merupakan Perancangan UI dari ubah layanan antar jemput. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat form ubah biaya antar jemput.

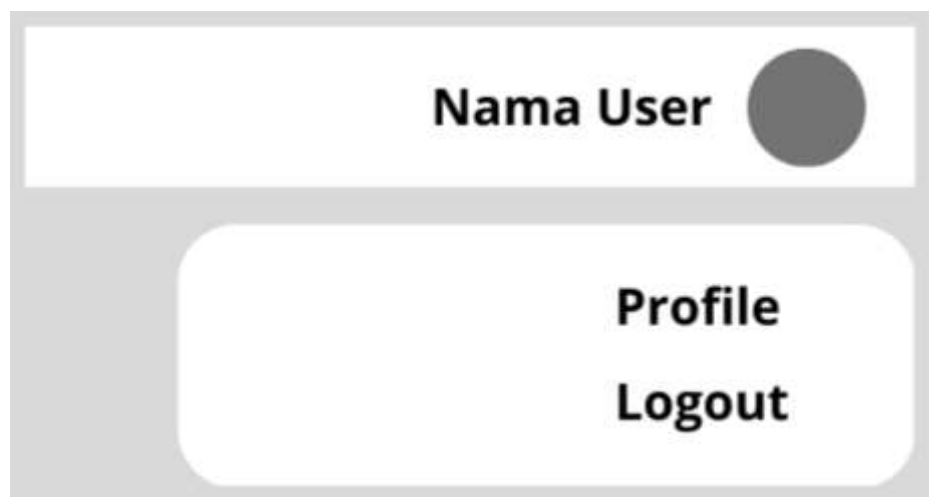
3.2.4.21 Perancangan UI Halaman *User Admin*



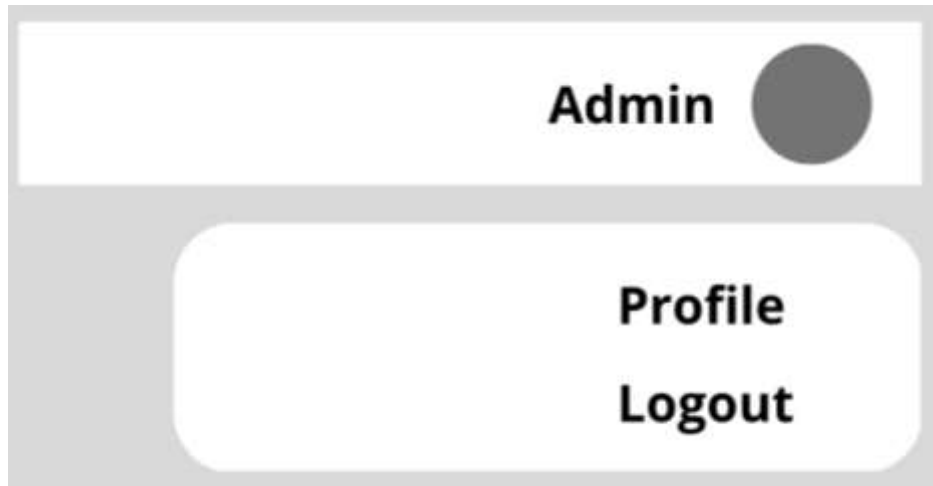
Gambar 3.47 Perancangan UI halaman user admin

Gambar diatas merupakan Perancangan UI dari *user*. Pada sidebar terdapat menu pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan menu *user*. Terdapat tabel informasi *user*.

3.2.4.22 Perancangan UI Halaman Informasi Akun



Gambar 3.48 Perancangan UI halaman informasi akun user



Gambar 3.49 Perancangan UI halaman informasi akun admin

Gambar diatas merupakan Perancangan UI dari informasi akun dari *user* dan admin. Terdapat menu profile dan logout pada *user* dan admin.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Setelah melakukan penganalisisan sistem dan pendesainan terperinci, maka selanjutnya menuju tahap implementasi. Implementasi merupakan tahap meletakkan sistem sehingga siap untuk di gunakan. Tujuan dari implementasi adalah untuk mengonfirmasi modul-modul perancangan sehingga *user* dapat memberikan masukan kepada developer.

4.1 Implementasi

Pada bagian implementasi aplikasi, dipaparkan code-code pada setiap proses berjalannya sistem yang telah dibangun. Pada *framework* Codeigniter 3 fungsi fungsi yang biasa untuk misalnya memasukkan data ke database, memindahkan *user* dari *login* ke home, dan lainnya. Semua fungsi itu di Codeigniter 3 menggunakan controllers yang berfungsi untuk mengontrol web sama seperti fungsi fungsi pada php.

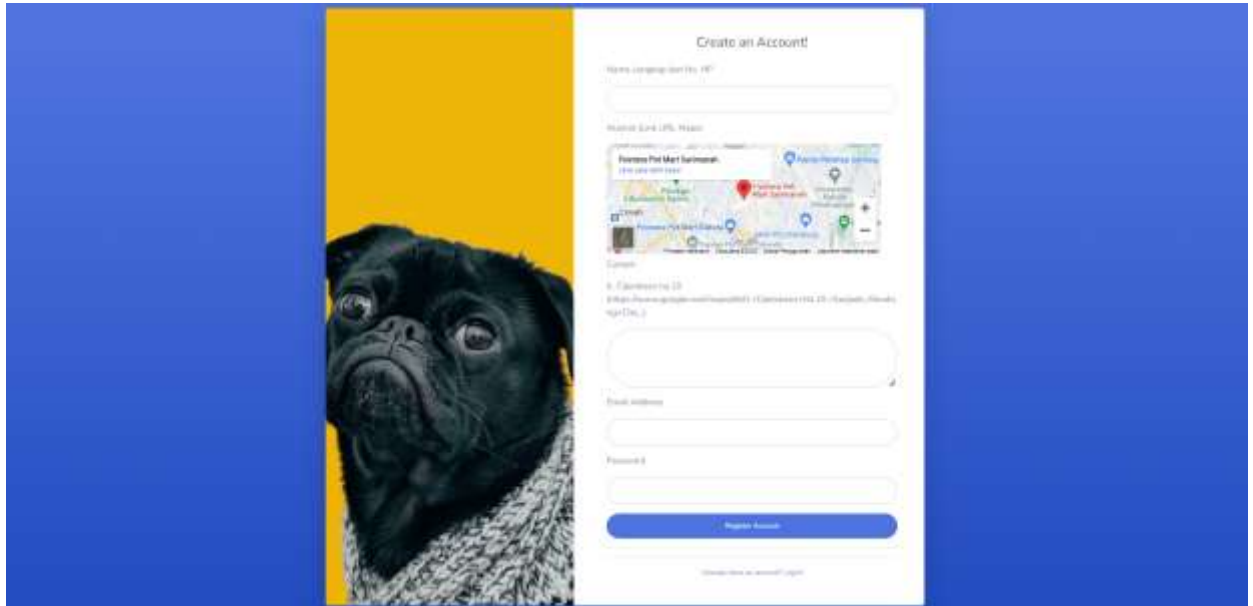
4.1.1 Halaman Registrasi

```
application > controllers > Register.php > Register > save
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3
4  class Register extends CI_Controller
5  {
6      public function index()
7      {
8          if ($this->session->userdata('logged_in') == TRUE) {
9              redirect(base_url('profile'));
10             }
11             $this->load->view('register_view');
12         }
13         public function save()
14         {
15             // Check Email
16             $check_email = $this->db->where('email', $this->input->post('email'))->get('tb_user')->row();
17             if ($check_email != '') {
18                 $this->session->set_flashdata('message', '<div class="alert alert-danger">Email sudah terdaftar !</div>');
19                 redirect('register');
20             }
21
22             // Insert Database
23             $data['name'] = $this->input->post('name');
24             $data['address'] = $this->input->post('address');
25             $data['email'] = $this->input->post('email');
26             $data['password'] = $this->bcrypt->hash_password($this->input->post('password'));
27             $data['role'] = 'user';
28             $this->db->insert('tb_user', $data);
29
30             $this->session->set_flashdata('message', '<div class="alert alert-success">Selamat anda berhasil terdaftar, silahkan login !</div>');
31             redirect('login');
32         }
33     }
```

Gambar 4.1 Code program registrasi

Pada code registrasi terdapat function save, disini dilakukan pengecekan email yang sudah terdaftar tidak dapat melakukan registrasi kembali. Pada insert database, *user* akan menginputkan name, address, email, password yang akan masuk ke tabel

database `tb_user`. Apabila registrasi berhasil maka akan muncul alert “Selamat anda berhasil terdaftar, silahkan *login*!” dan akan dibawa ke halaman *login*. Redirect berfungsi untuk membawa ke halaman yang dituju.



Gambar 4.2 Antarmuka halaman register

Gambar diatas merupakan implementasi dari halaman register. Dan hanya *user* yang dapat melakukan registrasi. Jika *user* belum memiliki akun, *user* harus mendaftarkan diri dibagian registrasi. Pada halaman registrasi *user* harus menginputkan nama lengkap dan nomor hp, alamat, email, dan password. Apabila sudah membuat akun baru *user* akan diarahkan ke halaman *login*.

4.1.2 Halaman *Login*

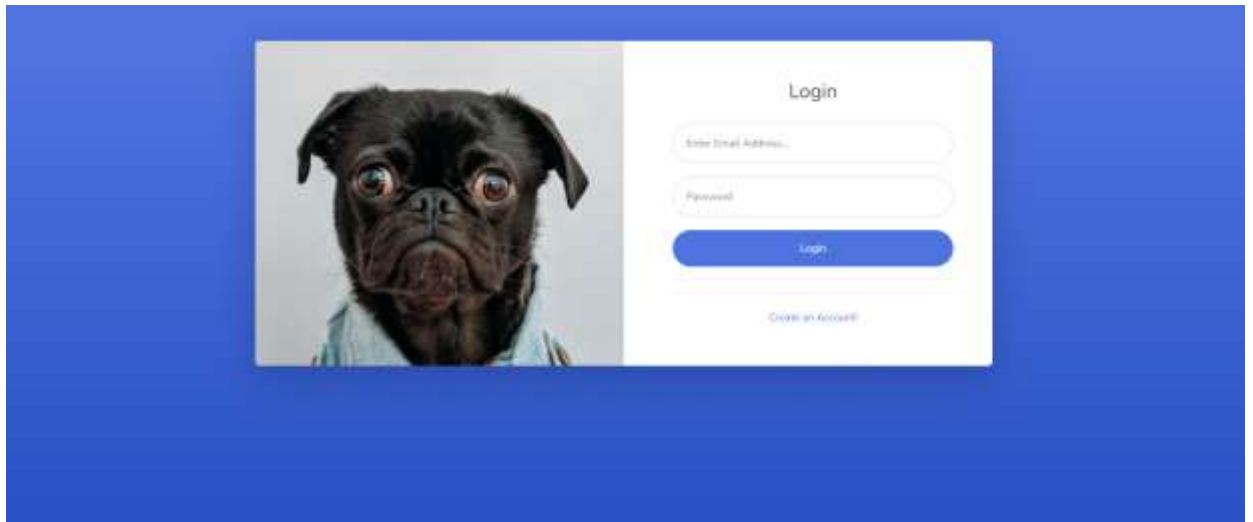
```

application > controllers > Login.php > Login > process
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3
4  class Login extends CI_Controller
5  {
6      public function index()
7      {
8          if ($this->session->userdata('logged_in') == TRUE) {
9              redirect(base_url('profile'));
10         }
11         $this->load->view('login_view');
12     }
13     public function process()
14     {
15         $email = $this->input->post('email');
16         $password = $this->input->post('password');
17
18         // Check User By Email
19         $user = $this->db->where('email', $email)->get('tb_user')->row();
20         if ($user == '') {
21             $this->session->set_flashdata('message', '<div class="alert alert-danger">Pengguna tidak ditemukan !</div>');
22             redirect('login');
23         }
24
25         // Invalid Password
26         if ($this->bcrypt->check_password($password, $user->password) == false) {
27             $this->session->set_flashdata('message', '<div class="alert alert-danger">Password salah !</div>');
28             redirect('login');
29         }
30
31         // Set Session
32         $sess_data['logged_in'] = TRUE;
33         $sess_data['id_user'] = $user->id_user;
34         $sess_data['role'] = $user->role;
35         $this->session->set_userdata($sess_data);
36
37         if ($user->role == 'admin') {
38             redirect('dashboard');
39         } else {
40             redirect('home');
41         }
42     }

```

Gambar 4.3 Code program login

Pada kodingan diatas dapat dilihat bagian diatas merupakan cara membuat sebuah class dan function. Pada bagian `$this->load->view('login_view');` dengan code tersebut maka halaman *login* sudah dapat diakses dan akan menampilkan sesuai dengan tampilan yang ada pada halaman login.php. Proses selanjutnya yaitu inputkan atau masukkan email dan password. Apabila email yang di inputkan tidak terdaftar pada database maka akan muncul pemberitahuan “Pengguna tidak ditemukan”. Selanjutnya adalah check password yang di inputkan apabila salah akan muncul "Password Salah.



Gambar 4.4 Antarmuka halaman login

Gambar diatas merupakan implementasi dari halaman *login*. Halaman *login* ini dapat diakses oleh *user* dan admin. Untuk *user* sebelum mereka melakukan pemesanan *grooming* mereka harus *login* terlebih dahulu untuk dapat masuk ke halaman *order*. Sedangkan admin harus *login* dulu untuk dapat masuk ke halaman dashboard admin. Di halaman *login* ini *user* dan admin harus menginputkan email dan password. Apabila *user* belum memiliki akun, *user* harus melakukan registrasi dahulu.

4.1.3 Halaman Home

```
application > controllers > Home.php > ...
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Home extends CI_Controller {
5      public function index()
6      {
7          $data['main_view'] = 'home_view';
8          $this->load->view('template_home_view', $data);
9      }
10     public function grooming()
11     {
12         $data['main_view'] = 'grooming_view';
13         $this->load->view('template_home_view', $data);
14     }
15     public function order()
16     {
17         redirect('order');
18     }
19 }
20
```

Gambar 4.5 Code program home

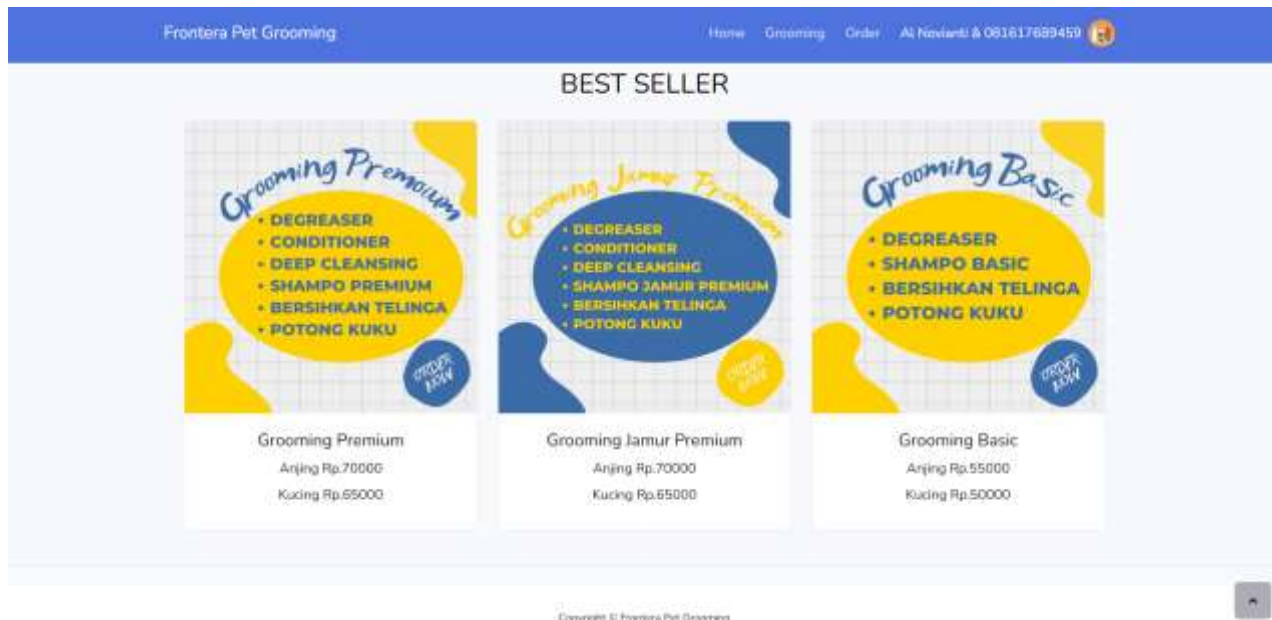
Pada code halaman home akan berisikan function dari *grooming* dan *order*. Ketika *user* sudah *login* maka otomatis akan dibawa ke halaman home.



Gambar 4.6 Antarmuka halaman home 1



Gambar 4.7 Antarmuka halaman home 2



Gambar 4.8 Antarmuka halaman home 3

Gambar diatas merupakan implementasi dari halaman home. Halaman home ini adalah halaman promosi serta menjelaskan informasi tentang *grooming*. Halaman home ini dapat diakses oleh *user* sebelum *user login*, jadi *user* dapat melihat terlebih dahulu informasi mengenai Frontera *Pet Grooming*.

4.1.4 Halaman *Grooming*

pplication > views > grooming_view.php > script

```

1  <section style="margin-top: 7em;">
2      <div class="container">
3          <div class="row text-center mb-3">
4              <div class="col">
5                  <h2>Paket Grooming</h2>
6              </div>
7          </div>
8          <div class="row">
9              <?php
10                 $package_grooming = $this->db->get('tb_package_grooming')->result();
11                 foreach ($package_grooming as $key) { ?>
12                     <div class="col-md-3 mb-3">
13                         <div class="card" onclick="order(<?=$key->id_package_grooming ?>)">
14                             title ?>">
15                             <div class="card-body">
16                                 <h5 class="card-title text-center"><?=$key->title ?></h5>
17                                 <?php
18                                     $pets = $this->db
19                                     ->where('id_package_grooming', $key->id_package_grooming)
20                                     ->join('tb_pet', 'tb_pet.id_pet = tb_package_grooming_pet.id_pet')
21                                     ->get('tb_package_grooming_pet')->result();
22                                 foreach ($pets as $pet) { ?>
23                                     <h6 class="card-title text-center"><?=$pet->name . ' Rp.' . $pet->price ?></h6>
24                                 <?php }
25                                 ?>
26                             </div>
27                         </div>
28                     </div>
29                 <?php }
30                 ?>
31             </div>
32         </div>
33     </section>
34     <hr class="my-4">

```

Gambar 4.9 Code program grooming 1

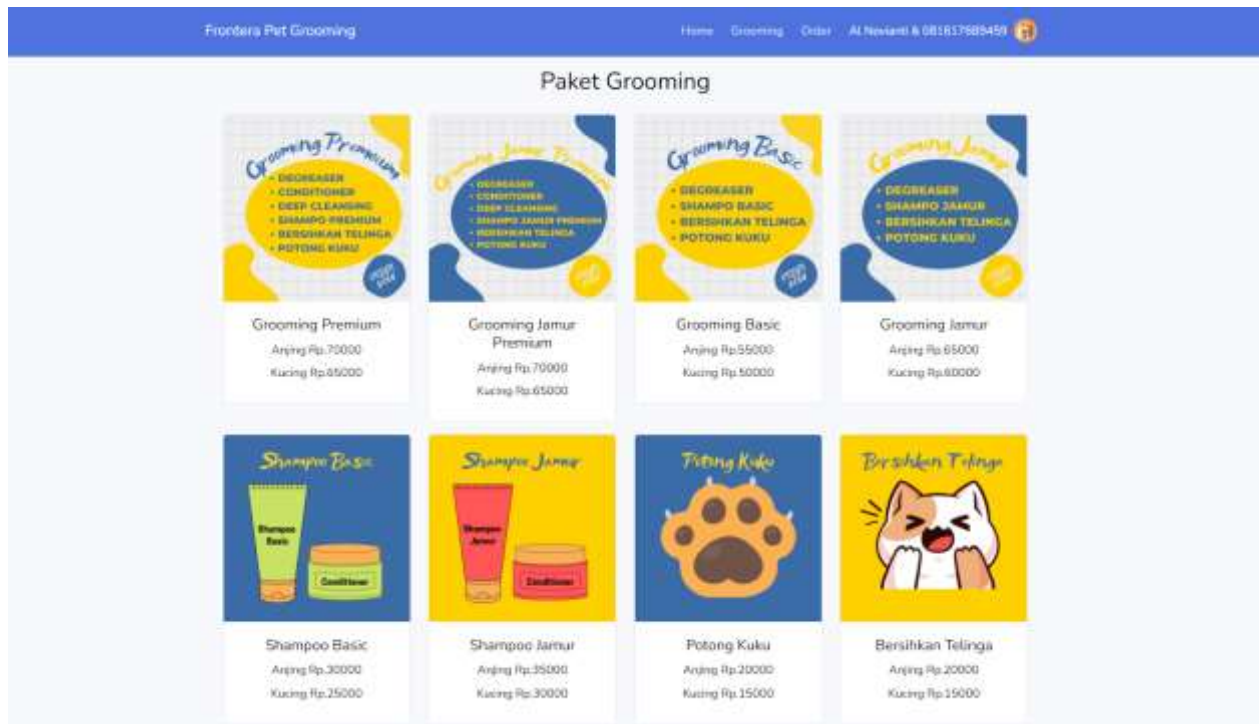
```

36 <div class="container">
37   <div class="row text-center mb-3">
38     <div class="col">
39       <h2>Kami Melayani Antar-Jemput</h2>
40       
41     </div>
42   </div>
43   <div class="row justify-content-center fs-5">
44     <div class="col-md-8">
45       <p></p>
46       <p class="text-center">Area Sarijadi | Rp.5.000</p>
47       <p class="text-center">Area Setia Budi | Rp.10.000</p>
48       <p class="text-center">Area Pasteur | Rp.10.000</p>
49     </div>
50   </div>
51 </div>
52
53 <script>
54   var base_url = "<?= base_url() ?>";
55   function order(id_package_grooming){
56     window.location.href= base_url+'order/add/'+id_package_grooming
57   }
58 </script>

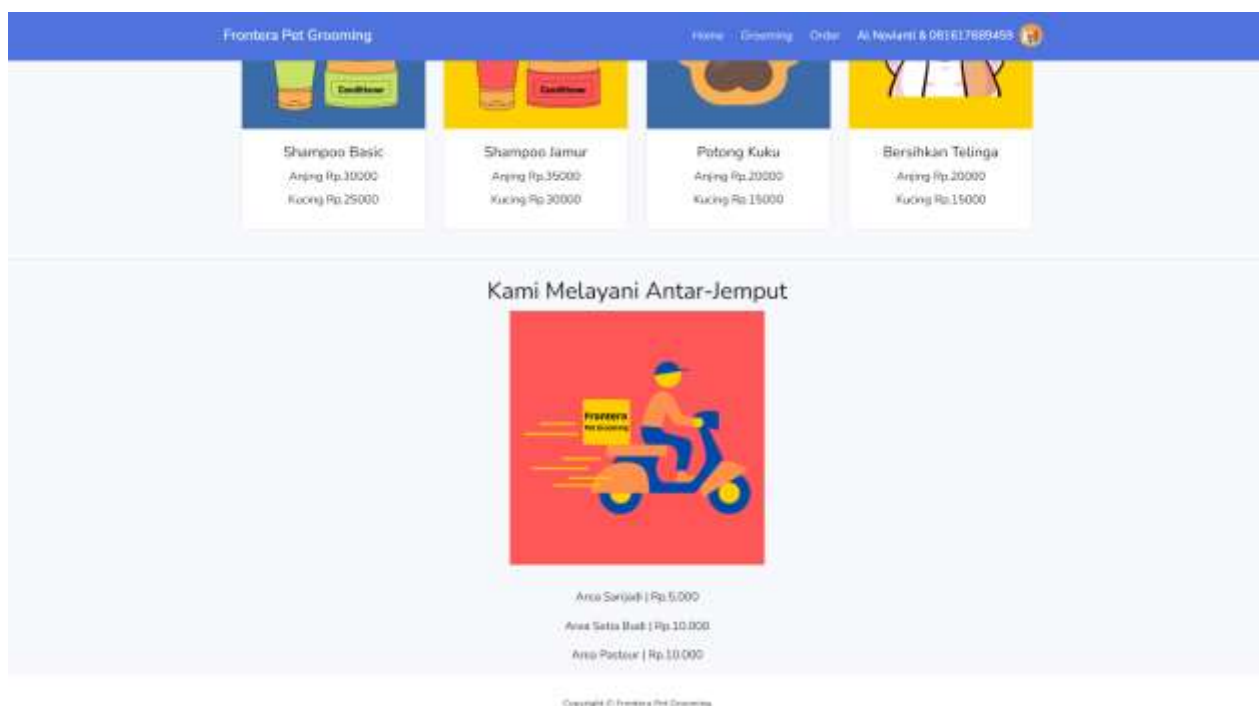
```

Gambar 4.10 Code program grooming 2

Pada code halaman *grooming*, disini berada pada bagian views terdapat card body terhubung dengan database dari pets. Sehingga card body akan menampilkan sesuai dengan yang sudah ada dalam database.

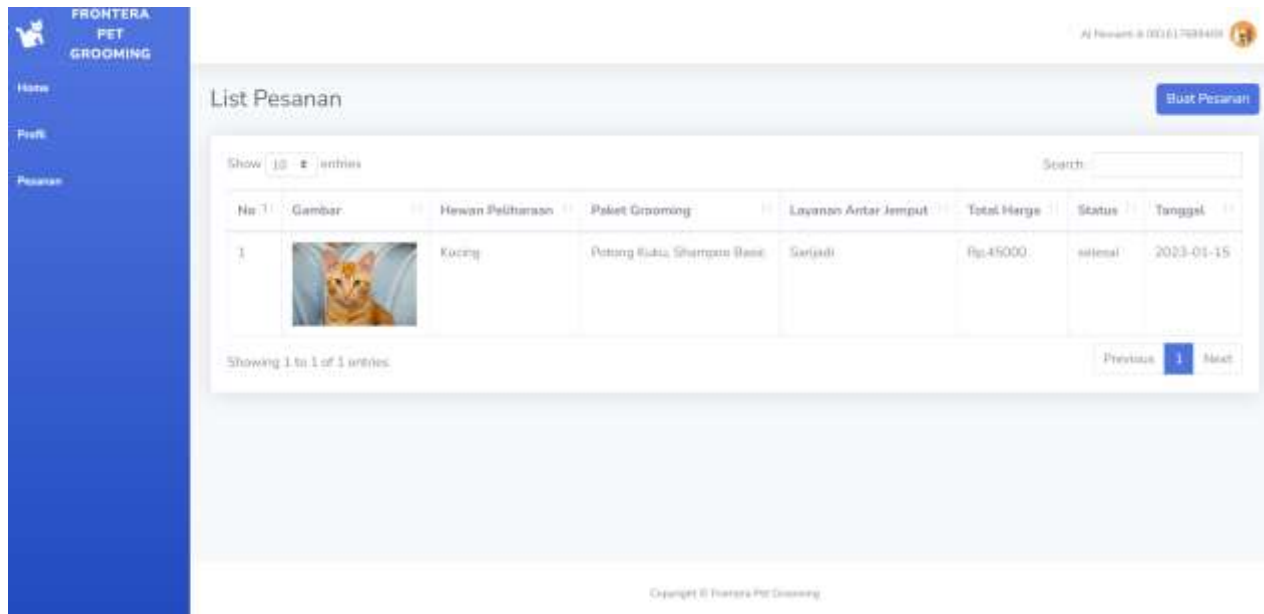


Gambar 4.11 Antarmuka halaman grooming 1



Gambar 4.12 Antarmuka halaman grooming 2

Gambar diatas merupakan implementasi dari halaman *grooming*. Halaman *grooming* berisi tentang katalog layanan *grooming* yang ditawarkan serta harganya dan berisi harga dari layanan antar jemput. Halaman ini dapat diakses oleh *user*



Gambar 4.14 Antarmuka halaman order

Gambar diatas merupakan implementasi dari halaman *order*. Untuk dapat masuk ke halaman *order user* harus memiliki akun terlebih dahulu. Pada sidebar terdapat menu untuk ke home, profil, pesanan. Dihalaman ini terdapat list pesanan *grooming* yang sudah kita pesan.

4.1.6 Halaman Buat Pesanan

```

13 public function index()
14 {
15
16     $order = $this->db
17     ->select('tb_order.*,tb_user.name as customer_name,SUM(tb_package_grooming_pet.price) as package_grooming_pet_price, tb_delivery_cost.id_delivery_cost , tb_delivery
18     if ($this->session->role == 'user') {
19         $order = $this->db
20         ->where('tb_order.id_user', $this->session->id_user);
21     }
22
23     $order = $this->db
24     ->join('tb_order_detail', 'tb_order_detail.id_order = tb_order.id_order')
25     ->join('tb_package_grooming_pet', 'tb_package_grooming_pet.id_package_grooming_pet = tb_order_detail.id_package_grooming_pet')
26     ->join('tb_package_grooming', 'tb_package_grooming.id_package_grooming = tb_package_grooming_pet.id_package_grooming')
27     ->join('tb_pet', 'tb_pet.id_pet = tb_package_grooming_pet.id_pet')
28     ->join('tb_delivery_cost', 'tb_delivery_cost.id_delivery_cost = tb_order.id_delivery_cost', 'left')
29     ->join('tb_user', 'tb_user.id_user = tb_order.id_user')
30     ->group_by('tb_order_detail.id_order')
31     ->get('tb_order')
32     ->result();
33     // print_r($order);
34     $data['order'] = $order;
35     $data['main_view'] = 'order/index_view';
36     $this->load->view('template_view', $data);
37 }

```

Gambar 4.15 Code program buat pesanan

Pada code buat pesanan diatas, terdapat pada bagian controller. Pada bagian *order* langsung terhubung kedalam database. Apabila *user* melakukan *order* maka *order* akan langsung masuk ke dalam tabel database.

The screenshot shows a web application interface for 'FRONTESA PET GROOMING'. On the left is a blue sidebar with navigation links: 'Home', 'Pet', and 'Pesanan'. The main content area is titled 'Buat Pesanan' (Create Order). It contains several form fields:

- 'Paket Grooming' with a '+ Tambahkan Paket' button.
- 'Pilih Paket Grooming' dropdown menu.
- 'Hewan Peliharaan' section with a 'Pilih Hewan Peliharaan' dropdown.
- 'Layanan Antar Jemput' section with a 'Tolak' button.
- 'Total' section with a text input field.
- 'Foto Hewan Peliharaan' section with a 'Choose File' button and 'No file chosen' text.
- 'Catatan (Optional)' text area.

 At the bottom of the form are two buttons: 'Kembali' (Back) in orange and 'Simpan' (Save) in blue.

Gambar 4.16 Antarmuka halaman buat pesanan

Gambar diatas merupakan implementasi dari halaman buat pesanan. Disini *user* harus menginputkan paket *grooming* yang dipilih, jenis hewan peliharaan, menggunakan layanan antar jemput atau tidak, foto hewan peliharaan yang akan di *grooming*, dan catatan untuk pihak *grooming*. *User* juga dapat menambah kolom paket *grooming* dan hewan peliharaan dengan cara mengklik *button* tambah pesanan maka otomatis form akan bertambah. Apabila data yang diinputkan dirasa sudah benar maka *user* tambah mengklik *button* simpan. Maka otomatis akan diarahkan ke halaman list pesanan.

4.1.7 Halaman Profil

```

application > controllers > Profile.php > _
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3
4  class Profile extends CI_Controller
5  {
6      public function __construct()
7      {
8          parent::__construct();
9          if ($this->session->userdata('logged_in') != TRUE) {
10             redirect('login');
11         }
12     }
13     public function index()
14     {
15         $data['user'] = $this->db->where('id_user', $this->session->id_user)->get('tb_user')->row();
16         $data['main_view'] = 'profile_view';
17         $this->load->view('template_view', $data);
18     }
19     public function save()
20     {
21         // Upload File
22         $destination_path = 'uploads/avatar/';
23         $config['upload_path'] = './' . $destination_path;
24         $config['allowed_types'] = '*';
25         $this->load->library('upload', $config);
26         if ($this->upload->do_upload('avatar')) {
27             $fileData = $this->upload->data();
28             $update['avatar'] = $destination_path . $fileData['file_name'];
29         }
30         $update['name'] = $this->input->post('name');
31         $update['address'] = $this->input->post('address');
32         $this->db->where('id_user', $this->session->id_user)->update('tb_user', $update);
33
34         $this->session->set_flashdata('message', '<div class="alert alert-success">Profil berhasil diperbarui !</div>');
35         redirect('profile');
36     }
37 }
38

```

Gambar 4.17 Code program profil

Pada code program profil, yang terdapat pada bagian controller. Disini berisi beberapa function yaitu function construct yang berfungsi untuk memastikan bahwa *user* sudah melakukan *login*. Function index yang berfungsi untuk menampilkan data

dari *user*. Function save yang berfungsi untuk menyimpan perubahan yang sudah dilakukan oleh *user*.

Gambar 4.18 Antarmuka halaman profil

Gambar diatas merupakan implementasi dari halaman profil. Pada halaman profil ini berfungsi untuk *user* dapat mengedit profil akunya seperti avatar, nama lengkap dan nomor hp, dan alamat. Setelah dirasa profilnya sudah benar dapat mengklik *button* simpan dan data berhasil diperbaharui akan muncul alert profil berhasil diperbarui.

4.1.8 Halaman Dashboard Admin

```

application > controllers > Dashboard.php > ...
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Dashboard extends CI_Controller {
5
6      public function __construct(){
7          parent::__construct();
8          if ($this->session->userdata('logged_in') != TRUE) {
9              redirect('login');
10         }
11     }
12     public function index()
13     {
14         $data['main_view'] = 'dashboard_view';
15         $this->load->view('template_view', $data);
16     }
17 }
18

```

Gambar 4.19 Code program dashboard admin

Pada code dashboard admin, yang terdapat pada bagian controller. Disini berisi 2 function yaitu function construct dan function index. Function construct yang

berfungsi untuk memastikan bahwa *user* sudah melakukan *login*. Function index yang berfungsi untuk menampilkan *template_view*.



Gambar 4.20 Antarmuka halaman dashboard admin

Gambar diatas merupakan implementasi dari halaman dashboard admin. Ketika admin ingin mengakses halaman dashboard admin harus melakukan *login* terlebih dahulu. Pada sidebar terdapat menu untuk ke halaman pesanan, paket *grooming*, harga *grooming*, hewan peliharaan, biaya antar jemput, dan *user*.

4.1.9 Halaman Pesanan Admin

```

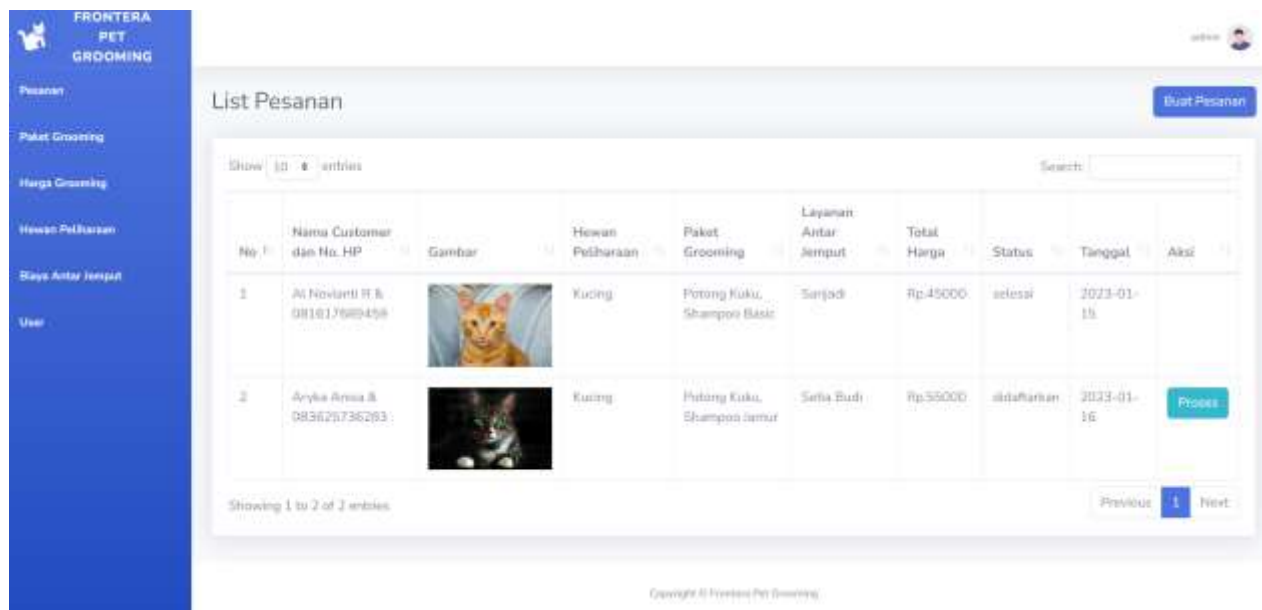
application > views > order > index_view.php > div.d-sm-flex.align-items-center.justify-content-between.mb-4
1  <div class="d-sm-flex align-items-center justify-content-between mb-4">
2      <h1 class="h3 mb-0 text-gray-800">List Pesanan </h1>
3      <a href="{<?>= base_url('order/add') ?>" class="ml-auto btn btn-primary">Buat Pesanan</a>
4  </div>
5
6  <?php echo $this->session->flashdata('message'); ?>
7
8  <div class="card shadow">
9      <div class="card-body">
10         <div class="table-responsive">
11             <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
12                 <thead>
13                     <tr>
14
15                         <?php
16                         if ($this->session->role == 'admin') { ?>
17                             <th>No</th>
18                             <th>Nama Customer dan No. HP</th>
19                             <th>Gambar</th>
20                             <th>Hewan Peliharaan</th>
21                             <th>Paket Grooming</th>
22                             <th>Layanan Antar Jemput</th>
23                             <th>Total Harga</th>
24                             <th>Status</th>
25                             <th>Tanggal</th>
26                             <th>Aksi</th>
27                         <?php } else { ?>
28                             <th>No</th>
29                             <th>Gambar</th>
30                             <th>Hewan Peliharaan</th>
31                             <th>Paket Grooming</th>
32                             <th>Layanan Antar Jemput</th>
33                             <th>Total Harga</th>
34                             <th>Status</th>
35                             <th>Tanggal</th>
36                         <?php }
37                         ?>
38

```

Gambar 4.21 Code program pesanan admin

Pada code program pesanan admin, yang terdapat pada views. Disini terdapat `if ($this->session->role == 'admin')`, yang artinya apabila yang mengakses ini adalah admin akan menampilkan tabel no, nama customer dan no.hp, gambar, hewan

peliharaan, paket *grooming*, layanan antar jemput, total harga, status, tanggal, dan aksi (disini admin dapat mengatur status menjadi proses dan selesai).



Gambar 4.22 Antarmuka halaman pesanan admin

Gambar diatas merupakan implementasi dari halaman pesanan admin. Di halaman ini memuat informasi dari pesanan *grooming* apa saja yang di pesan oleh *user*. Di halaman ini admin dapat mengetahui informasi nama dan no.hp customer, foto hewan yang akan *grooming*, jenis hewan peliharaan, paket *grooming* yang dipesan, menggunakan layanan antar jemput atau tidak, total harga, status pesanan *grooming* (didaftarkan, proses, dan selesai), tanggal, dan aksi (ketika admin mengklik *button* proses maka status otomatis akan berubah menjadi proses).

4.1.10 Halaman Paket *Grooming* Admin

```

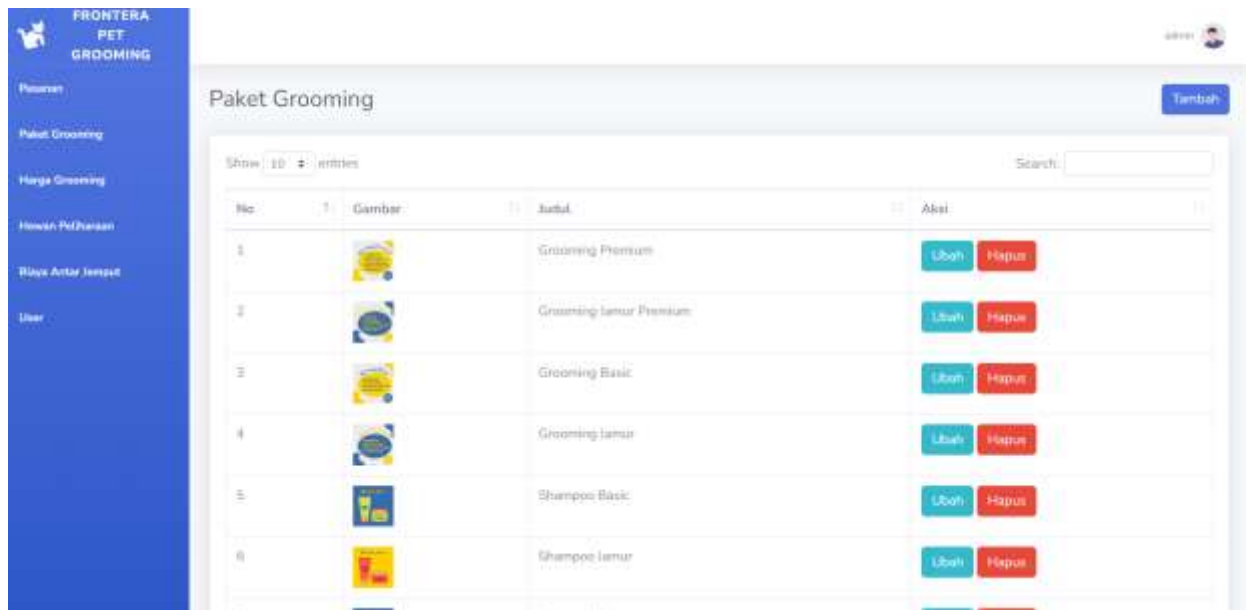
application > views > package_grooming > index_view.php > div.card.shadow
1 <div class="d-sm-flex align-items-center justify-content-between mb-4">
2   <h1 class="h3 mb-0 text-gray-800">Paket Grooming</h1>
3   <a href="<?>= base_url('package_grooming/add') ?>" class="ml-auto btn btn-primary">Tambah</a>
4 </div>
5
6 <?php echo $this->session->flashdata('message'); ?>
7
8 <div class="card shadow">
9   <div class="card-body">
10    <div class="table-responsive">
11      <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
12        <thead>
13          <tr>
14            <th>No</th>
15            <th>Gambar</th>
16            <th>Judul</th>
17            <th>Aksi</th>
18          </tr>
19        </thead>
20        <tbody>
21          <?php
22            $no = 1;
23            foreach ($package_grooming as $key) { ?>
24              <tr>
25                <td><?>= $no++ ?></td>
26                <td> </td>
27                <td><?>= $key->title ?></td>
28                <td>
29                  <a href="<?>= base_url('package_grooming/edit/'. $key->id_package_grooming) ?>" class="btn btn-info">Ubah</a>
30                  <a href="<?>= base_url('package_grooming/delete/'. $key->id_package_grooming) ?>" class="btn btn-danger">Hapus</a>
31                </td>
32              </tr>
33            <?php }
34            ?>
35          </tbody>
36        </table>
37      </div>
38    </div>
39  </div>

```

Gambar 4.23 Code program paket grooming admin

Pada code program paket *grooming* admin, yang terdapat pada bagian views. Fungsi `base_url` adalah menghasilkan URL berdasarkan konfigurasi `base_url`. Tetapi pada `base_url()` tidak disertakan `index.php` di belakangnya. Disini terdapat `<a href="<?>= base_url('package_grooming/add') ?>" class="ml-auto btn btn-primary">Tambah` yang artinya ketika admin *klik button* Tambah maka akan

dibawa ke `package_grooming`. Hal itu juga sama seperti yang terdapat pada `base_url` ubah dan hapus.



Gambar 4.24 Antarmuka halaman paket grooming admin

Gambar diatas merupakan implementasi dari halaman paket *grooming*. Halaman ini berisi informasi mengenai gambar dan judul yang akan dipasang pada halaman *grooming user*. Pada halaman ini admin dapat melakukan CRUD paket *grooming*. Admin dapat menambahkan paket *grooming* dengan cara *klik button* tambah. Admin juga dapat melakukan edit paket *grooming* dengan cara *klik button* ubah. Dan admin juga dapat menghapus paket *grooming* dengan cara *klik button* hapus.

4.1.11 Halaman Tambah Paket Grooming Admin

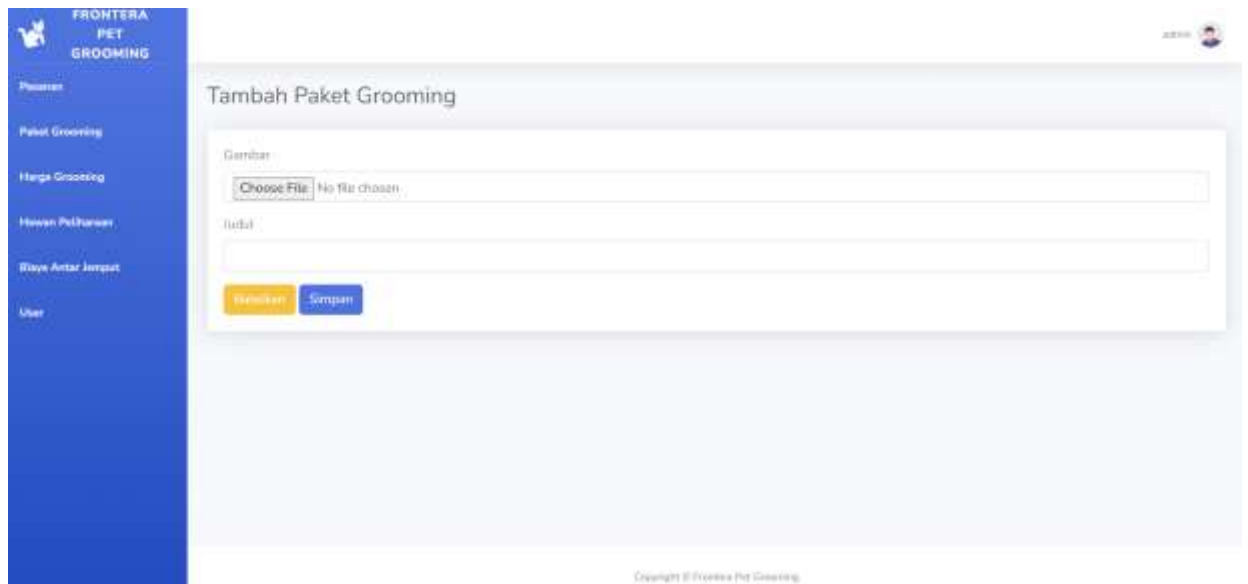
```

19 public function add()
20 {
21     $data['main_view'] = 'package_grooming/add_view';
22     $this->load->view('template_view', $data);
23 }
24 public function save_add()
25 {
26     // Upload File
27     $destination_path = 'uploads/package_grooming/';
28     $config['upload_path'] = './' . $destination_path;
29     $config['allowed_types'] = '*';
30     $this->load->library('upload', $config);
31     if ($this->upload->do_upload('image')) {
32         $fileData = $this->upload->data();
33         $insert['image'] = $destination_path . $fileData['file_name'];
34     }

```

Gambar 4.25 Code program tambah paket grooming admin

Pada code program tambah paket *grooming* admin, yang terdapat pada bagian controller. Terdapat function add dan function save_add. Disini function add terdapat `$this->load->view('template_view', $data);` yang bertujuan untuk menampilkan views dari template_view. Serta function save_add yang berisi fungsi untuk mengupload image dan file_name.



Gambar 4.26 Antarmuka halaman tambah paket grooming admin

Gambar diatas merupakan implementasi dari tambah paket *grooming*. Ketika admin mengklik *button* tambah pada halaman paket *grooming* maka akan diarahkan ke halaman tambah paket *grooming*. Disini admin dapat menginputkan gambar serta judul (nama paket *grooming*) ketika dirasa sudah benar admin dapat mengklik *button* simpan. Otomatis data yang ditambahkan akan masuk ke list paket *grooming*.

4.1.12 Halaman Ubah Paket *Grooming* Admin

```

42 public function edit($id_package_grooming)
43 {
44     $data['package_grooming'] = $this->db->where('id_package_grooming', $id_package_grooming)->get('tb_package_grooming')->row();
45     $data['main_view'] = 'package_grooming/edit_view';
46     $this->load->view('template_view', $data);
47 }
48 public function save_edit($id_package_grooming)
49 {
50     // Upload File
51     $destination_path = 'uploads/package_grooming/';
52     $config['upload_path'] = './' . $destination_path;
53     $config['allowed_types'] = '*';
54     $this->load->library('upload', $config);
55     if ($this->upload->do_upload('image')) {
56         $fileData = $this->upload->data();
57         $update['image'] = $destination_path . $fileData['file_name'];
58     }
59     $update['title'] = $this->input->post('title');
60     $this->db->where('id_package_grooming', $id_package_grooming)->update('tb_package_grooming', $update);
61
62     $this->session->set_flashdata('message', '<div class="alert alert-success">Paket grooming berhasil diperbarui !</div>');
63     redirect('package_grooming');
64 }

```

Gambar 4.27 Code program ubah paket grooming admin

Pada code program ubah paket *grooming* admin, yang terdapat pada bagian controller. Terdapat function edit yang berfungsi menampilkan form ubah paket *grooming* dan function save_edit yang berfungsi menyimpan perubahan yang telah dilakukan.

Gambar 4.28 Antarmuka halaman ubah paket grooming admin

Gambar diatas merupakan implementasi dari ubah paket *grooming*. Ketika admin mengklik *button* ubah pada halaman paket *grooming* maka akan diarahkan ke halaman ubah paket *grooming*. Disini admin dapat menginputkan gambar serta judul

(nama paket *grooming*) yang ingin diubah, ketika sudah diinputkan *klik button* simpan maka otomatis data akan berubah.

4.1.13 Halaman Harga *Grooming* Admin

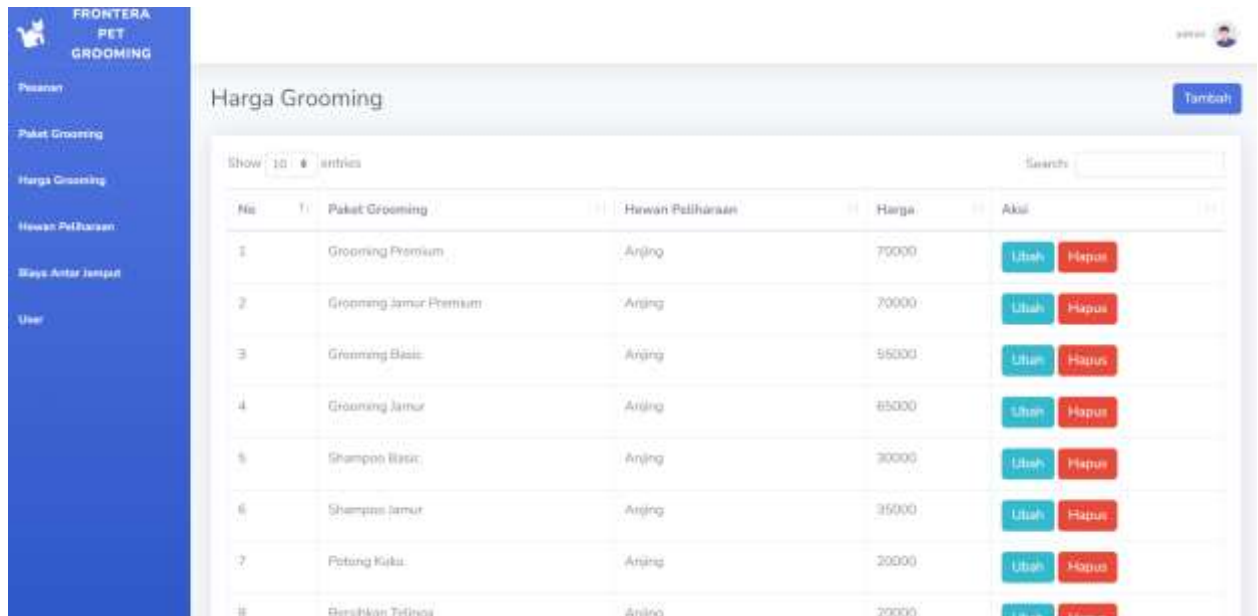
```

application > views > package_grooming_pet > index_view.php > div-d-sm-flex-align-items-center justify-content-between mb-4
1  <div class="d-sm-flex align-items-center justify-content-between mb-4">
2    <h1 class="h3 mb-0 text-gray-800">Harga Grooming </h1>
3    <a href="{?= base_url('package_grooming_pet/add') ?}" class="ml-auto btn btn-primary">Tambah</a>
4  </div>
5
6  <?php echo $this->session->flashdata('message'); ?>
7
8  <div class="card shadow">
9    <div class="card-body">
10   <div class="table-responsive">
11     <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
12       <thead>
13         <tr>
14           <th>No</th>
15           <th>Paket Grooming</th>
16           <th>Hewan Peliharaan</th>
17           <th>Harga</th>
18           <th>Aksi</th>
19         </tr>
20       </thead>
21       <tbody>
22         <?php
23         $no = 1;
24         foreach ($package_grooming_pet as $key) { ?>
25           <tr>
26             <td>{?= $no++ ?}</td>
27             <td>{?= $key->package_grooming_title ?}</td>
28             <td>{?= $key->pet_name ?}</td>
29             <td>{?= $key->price ?}</td>
30             <td>
31               <a href="{?= base_url('package_grooming_pet/edit/'. $key->id_package_grooming_pet) ?}" class="btn btn-info">Ubah</a>
32               <a href="{?= base_url('package_grooming_pet/delete/'. $key->id_package_grooming_pet) ?}" class="btn btn-danger">Hapus</a>
33             </td>
34           </tr>
35         <?php }
36       <?>
37     </tbody>
38   </table>
39 </div>
40 </div>
41 </div>

```

Gambar 4.29 Code program harga grooming admin

Pada code program harga *grooming* admin, yang terdapat pada bagian views. Fungsi `base_url` adalah menghasilkan URL berdasarkan konfigurasi `base_url`. Tetapi pada `base_url()` tidak disertakan `index.php` di belakangnya. Disini terdapat `Tambah` yang artinya ketika admin *klik button* Tambah maka akan dibawa ke `package_grooming_pet`. Hal itu juga sama seperti yang terdapat pada `base_url` ubah dan hapus.



No	Paket Grooming	Hewan Peliharaan	Harga	Aksi
1	Grooming Premium	Anjing	70000	Ubah Hapus
2	Grooming Jamur Premium	Anjing	70000	Ubah Hapus
3	Grooming Basic	Anjing	55000	Ubah Hapus
4	Grooming Jamur	Anjing	65000	Ubah Hapus
5	Shampoo Basic	Anjing	30000	Ubah Hapus
6	Shampoo Jamur	Anjing	35000	Ubah Hapus
7	Potong Kuku	Anjing	20000	Ubah Hapus
8	Bersihkan Telinga	Anjing	20000	Ubah Hapus

Gambar 4.30 Antarmuka halaman harga grooming admin

Gambar diatas merupakan implementasi dari halaman harga *grooming*. Halaman ini berisi informasi mengenai harga dari paket *grooming* serta jenis hewan peliharaanya. Pada halaman ini admin dapat melakukan CRUD harga *grooming*. Admin dapat menambahkan harga *grooming* dengan cara *klik button* tambah. Admin juga dapat melakukan edit harga *grooming* dengan cara *klik button* ubah. Dan admin juga dapat menghapus harga *grooming* dengan cara *klik button* hapus.

4.1.14 Halaman Tambah Harga Grooming Admin

```

22     public function add()
23     {
24         $data['package_grooming'] = $this->db->get('tb_package_grooming')->result();
25         $data['pet'] = $this->db->get('tb_pet')->result();
26         $data['main_view'] = 'package_grooming_pet/add_view';
27         $this->load->view('template_view', $data);
28     }
29     public function save_add()
30     {
31         $id_package_grooming = $this->input->post('id_package_grooming');
32         $id_pet = $this->input->post('id_pet');
33         $price = $this->input->post('price');

```

Gambar 4.31 Code program tambah harga grooming admin

Pada code program tambah harga *grooming* admin, yang terdapat pada bagian controller. Terdapat function add dan function save_add. Disini function add terdapat `$this->load->view('template_view', $data);` yang bertujuan untuk menampilkan views

dari `template_view`. Serta function `save_add` yang berisi fungsi untuk mengupload `id_package_grooming`, `id_pet`, dan `price`.

Gambar 4. 32 Antarmuka halaman tambah harga grooming admin

Gambar diatas merupakan implementasi dari tambah harga *grooming*. Ketika admin mengklik *button* tambah pada halaman harga *grooming* maka akan diarahkan ke halaman tambah harga *grooming*. Disini admin dapat menginputkan paket *grooming*, hewan peliharaan dan harga, ketika dirasa sudah benar admin dapat mengklik *button* simpan. Otomatis data yang ditambahkan akan masuk ke list harga *grooming*.

4.1.15 Halaman Ubah Harga Grooming Admin

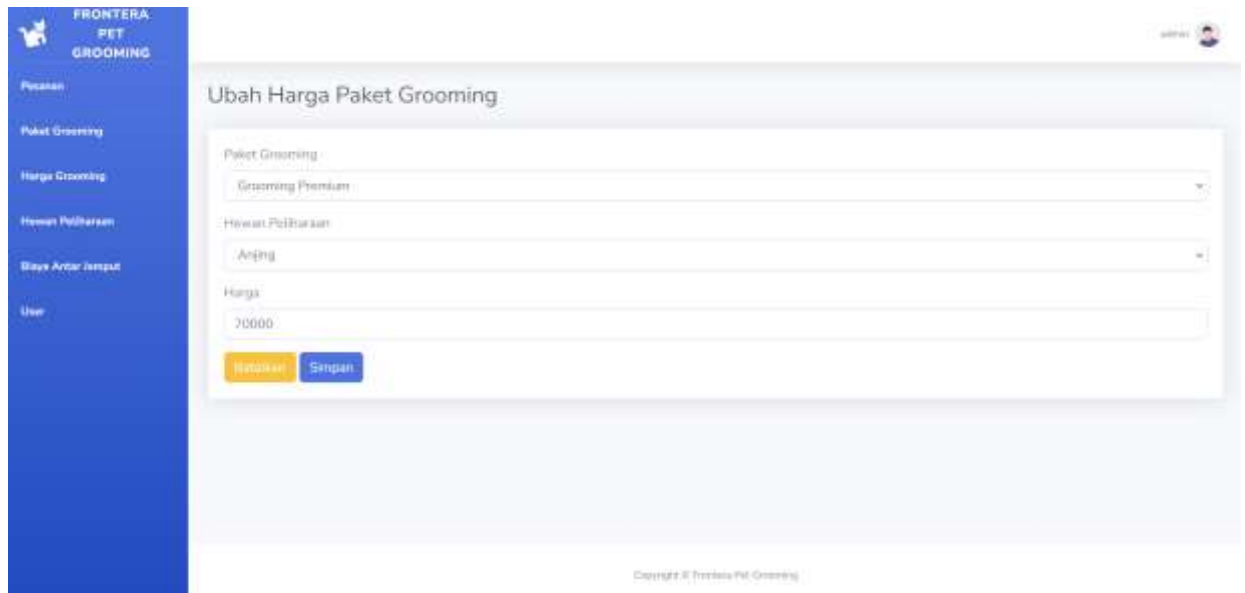
```

50 public function edit($id_package_grooming_pet)
51 {
52     $data['package_grooming_pet'] = $this->db->where('id_package_grooming_pet', $id_package_grooming_pet)->get('tb_package_grooming_pet')->row();
53     $data['package_grooming'] = $this->db->get('tb_package_grooming')->result();
54     $data['pet'] = $this->db->get('tb_pet')->result();
55     $data['main_view'] = 'package_grooming_pet/edit_view';
56     $this->load->view('template_view', $data);
57 }
58 public function save_edit($id_package_grooming_pet)
59 {
60     $id_package_grooming = $this->input->post('id_package_grooming');
61     $id_pet = $this->input->post('id_pet');
62     $price = $this->input->post('price');

```

Gambar 4.33 Code program ubah harga grooming admin

Pada code program ubah harga *grooming* admin, yang terdapat pada bagian controller. Terdapat function `edit` yang berfungsi menampilkan form ubah harga *grooming* dan function `save_edit` yang berfungsi menyimpan perubahan yang telah dilakukan.



Gambar 4.34 Antarmuka halaman ubah harga grooming admin

Gambar diatas merupakan implementasi dari ubah harga *grooming*. Ketika admin mengklik *button* ubah pada halaman harga *grooming* maka akan diarahkan ke halaman ubah harga *grooming*. Disini admin dapat menginputkan paket *grooming*, hewan peliharaan, dan harga yang ingin diubah, ketika sudah diinputkan klik *button* simpan maka otomatis data akan berubah.

4.1.16 Halaman Hewan Peliharaan Admin

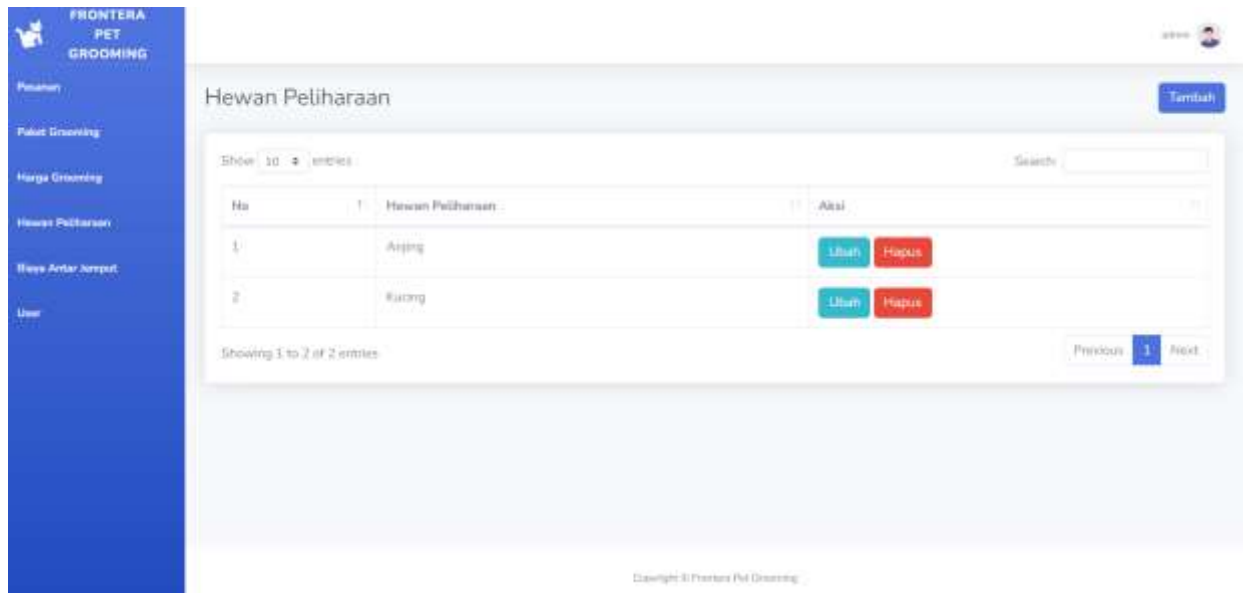
```

application > views > pet > index_view.php > div.card.shadow
1  <div class="d-sm-flex align-items-center justify-content-between mb-4">
2    <h1 class="h3 mb-0 text-gray-800">Hewan Peliharaan</h1>
3    <a href="<?= base_url('pet/add') ?>" class="ml-auto btn btn-primary">Tambah</a>
4  </div>
5
6  <?php echo $this->session->flashdata('message'); ?>
7
8  <div class="card shadow">
9    <div class="card-body">
10     <div class="table-responsive">
11       <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
12         <thead>
13           <tr>
14             <th>No</th>
15             <th>Hewan Peliharaan</th>
16             <th>Aksi</th>
17           </tr>
18         </thead>
19         <tbody>
20           <?php
21             $no = 1;
22             foreach ($pet as $key) { ?>
23               <tr>
24                 <td><?= $no++ ?></td>
25                 <td><?= $key->name ?></td>
26                 <td>
27                   <a href="<?= base_url('pet/edit/'. $key->id_pet) ?>" class="btn btn-info">Ubah</a>
28                   <a href="<?= base_url('pet/delete/'. $key->id_pet) ?>" class="btn btn-danger">Hapus</a>
29                 </td>
30               </tr>
31             <?php }
32             ?>
33           </tbody>
34         </table>
35       </div>
36     </div>
37   </div>

```

Gambar 4.35 Code program hewan peliharaan admin

Pada code program hewan peliharaan admin, yang terdapat pada bagian views. Fungsi `base_url` adalah menghasilkan URL berdasarkan konfigurasi `base_url`. Tetapi pada `base_url()` tidak disertakan `index.php` di belakangnya. Disini terdapat `<a href="<?= base_url('pet/add') ?>" class="ml-auto btn btn-primary">Tambah` yang artinya ketika admin *klik button* Tambah maka akan dibawa ke pet. Hal itu juga sama seperti yang terdapat pada `base_url` ubah dan hapus.



Gambar 4.36 Antarmuka halaman hewan peliharaan admin

Gambar diatas merupakan implementasi dari halaman hewan peliharaan. Halaman ini berisi informasi mengenai jenis dari hewan peliharaan. Pada halaman ini admin dapat melakukan CRUD hewan peliharaan. Admin dapat menambahkan jenis hewan peliharaan baru dengan cara *klik button* tambah. Admin juga dapat melakukan edit hewan peliharaan dengan cara *klik button* ubah. Dan admin juga dapat menghapus harga grooming dengan cara *klik button* hapus.

4.1.17 Halaman Tambah Hewan Peliharaan Admin

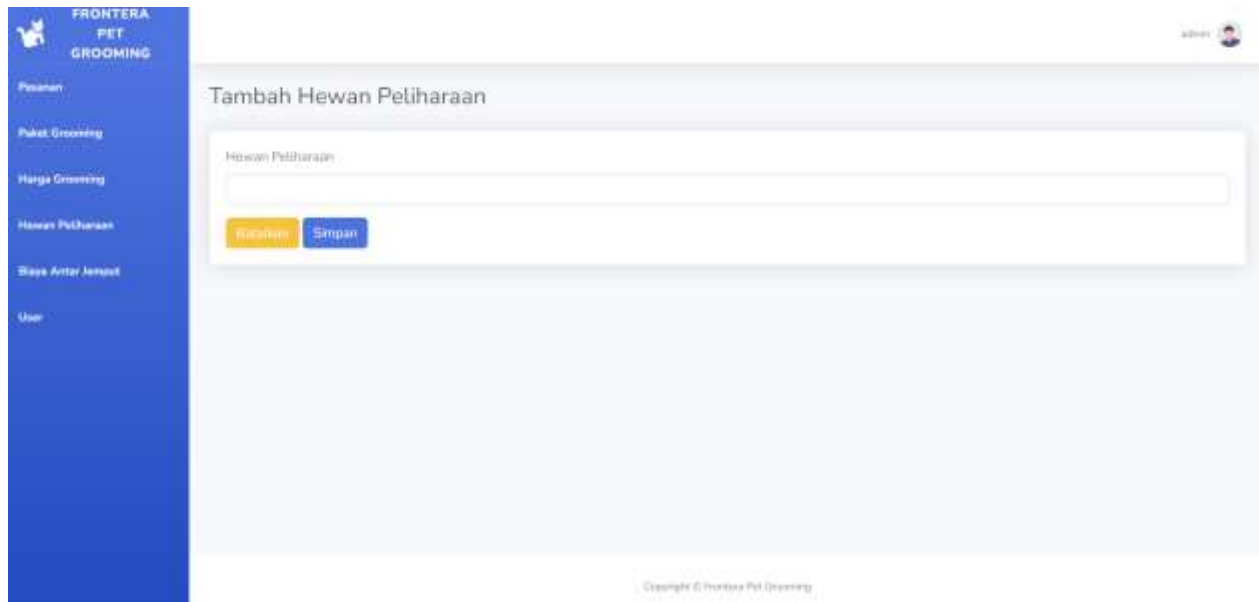
```

19 public function add()
20 {
21     $data['main_view'] = 'pet/add_view';
22     $this->load->view('template_view', $data);
23 }
24 public function save_add()
25 {
26     $insert['name'] = $this->input->post('name');
27     $this->db->insert('tb_pet', $insert);
28
29     $this->session->set_flashdata('message', '<div class="alert alert-success">Hewan Peliharaan berhasil ditambahkan !</div>');
30     redirect('pet');
31 }

```

Gambar 4.37 Code program tambah hewan peliharaan admin

Pada code program tambah hewan peliharaan admin, yang terdapat pada bagian controller. Terdapat function add dan function save_add. Disini function add terdapat `$this->load->view('template_view', $data);` yang bertujuan untuk menampilkan views dari template_view. Serta function save_add yang berisi fungsi untuk mengupload name dan tb_pet.



Gambar 4.38 Antarmuka halaman tambah hewan peliharaan admin

Gambar diatas merupakan implementasi dari tambah hewan peliharaan. Ketika admin mengklik *button* tambah pada halaman hewan peliharaan maka akan diarahkan ke halaman tambah hewan peliharaan. Disini admin dapat menginputkan jenis hewan baru, ketika dirasa sudah benar admin dapat mengklik *button* simpan. Otomatis data yang ditambahkan akan masuk ke list hewan peliharaan.

4.1.18 Halaman Ubah Hewan Peliharaan Admin

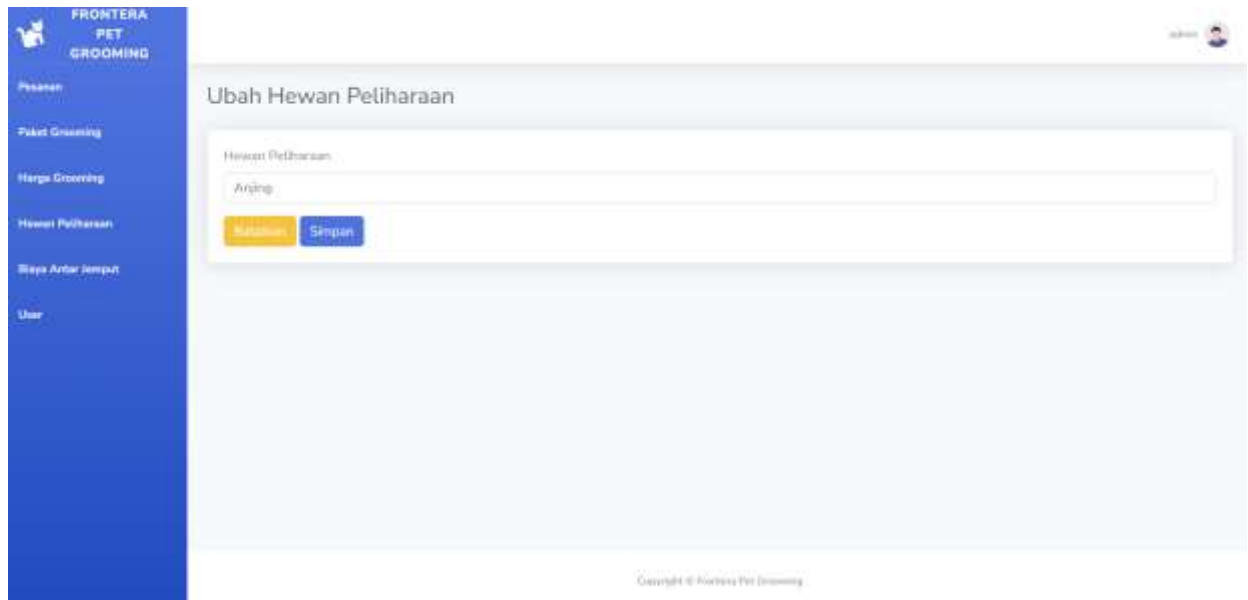
```

32 public function edit($id_pet)
33 {
34     $data['pet'] = $this->db->where('id_pet', $id_pet)->get('tb_pet')->row();
35     $data['main_view'] = 'pet/edit_view';
36     $this->load->view('template_view', $data);
37 }
38 public function save_edit($id_pet)
39 {
40     $update['name'] = $this->input->post('name');
41     $this->db->where('id_pet', $id_pet)->update('tb_pet', $update);
42
43     $this->session->set_flashdata('message', '<div class="alert alert-success">Hewan Peliharaan berhasil diperbarui !</div>');
44     redirect('pet');
45 }

```

Gambar 4.39 Code program ubah hewan peliharaan admin

Pada code program ubah hewan peliharaan admin, yang terdapat pada bagian controller. Terdapat function edit yang berfungsi menampilkan form ubah hewan peliharaan dan function save_edit yang berfungsi menyimpan perubahan yang telah dilakukan.



Gambar 4.40 Code program ubah hewan peliharaan admin

Gambar diatas merupakan implementasi dari ubah hewan peliharaan. Ketika admin mengklik *button* ubah pada halaman hewan peliharaan maka akan diarahkan ke halaman ubah hewan peliharaan. Disini admin dapat menginputkan jenis hewan peliharaan yang ingin diubah, ketika sudah diinputkan *klik button* simpan maka otomatis data akan berubah.

4.1.19 Halaman Biaya Antar Jemput Admin

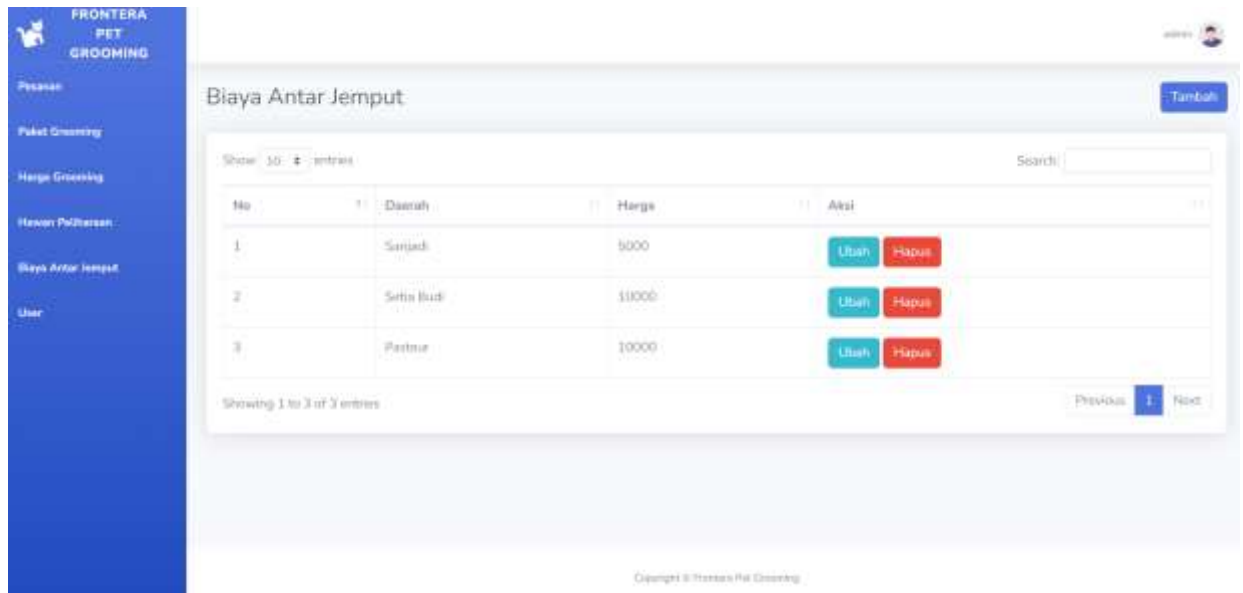
```

application > views > delivery_cost > index_view.php > div.card.shadow > div.card-body > div.table-responsive
1  <div class="d-sm-flex align-items-center justify-content-between mb-4">
2    <h1 class="h3 mb-0 text-gray-800">Biaya Antar Jemput</h1>
3    <a href="?= base_url('delivery_cost/add') ?>" class="ml-auto btn btn-primary">Tambah</a>
4  </div>
5
6  <?php echo $this->session->flashdata('message'); ?>
7
8  <div class="card shadow">
9    <div class="card-body">
10     <div class="table-responsive">
11       <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
12         <thead>
13           <tr>
14             <th>No</th>
15             <th>Daerah</th>
16             <th>Harga</th>
17             <th>Aksi</th>
18           </tr>
19         </thead>
20         <tbody>
21           <?php
22             $no = 1;
23             foreach ($delivery_cost as $key) { ?>
24               <tr>
25                 <td><?= $no++ ?></td>
26                 <td><?= $key->place ?></td>
27                 <td><?= $key->price ?></td>
28                 <td>
29                   <a href="?= base_url('delivery_cost/edit/' . $key->id_delivery_cost) ?>" class="btn btn-info">Ubah</a>
30                   <a href="?= base_url('delivery_cost/delete/' . $key->id_delivery_cost) ?>" class="btn btn-danger">Hapus</a>
31                 </td>
32               </tr>
33             <?php }
34           <?>
35         </tbody>
36       </table>
37     </div>
38   </div>
39 </div>

```

Gambar 4.41 Code program biaya antar jemput admin

Pada code program hewan biaya antar jemput admin, yang terdapat pada bagian views. Fungsi `base_url` adalah menghasilkan URL berdasarkan konfigurasi `base_url`. Tetapi pada `base_url()` tidak disertakan `index.php` di belakangnya. Disini terdapat `" class="ml-auto btn btn-primary">Tambah` yang artinya ketika admin *klik button* Tambah maka akan dibawa ke `delivery_cost`. Hal itu juga sama seperti yang terdapat pada `base_url` ubah dan hapus.



Gambar 4.42 Antarmuka halaman biaya antar jemput admin

Gambar diatas merupakan implementasi dari biaya antar jemput. Halaman ini berisi informasi mengenai biaya antar jemput layanan *grooming*. Pada halaman ini admin dapat melakukan CRUD daerah dan biaya antar jemput. Admin dapat menambahkan data layanan antar jemput dengan cara *klik button* tambah. Admin juga dapat melakukan edit data layanan antar jemput dengan cara *klik button* ubah. Dan admin juga dapat menghapus data layanan antar jemput dengan cara *klik button* hapus.

4.1.20 Halaman Tambah Biaya Antar Jemput Admin

```

17 public function add()
18 {
19     $data['main_view'] = 'delivery_cost/add_view';
20     $this->load->view('template_view', $data);
21 }
22 public function save_add()
23 {
24     $insert['place'] = $this->input->post('place');
25     $insert['price'] = $this->input->post('price');
26     $this->db->insert('tb_delivery_cost', $insert);
27
28     $this->session->set_flashdata('message', '<div class="alert alert-success">Biaya antar jemput berhasil ditambahkan !</div>');
29     redirect('delivery_cost');

```

Gambar 4.43 Code program tambah biaya antar jemput admin

Pada code program tambah biaya antar jemput admin, yang terdapat pada bagian controller. Terdapat function add dan function save_add. Disini function add terdapat `$this->load->view('template_view', $data);` yang bertujuan untuk menampilkan views dari template_view. Serta function save_add yang berisi fungsi untuk mengupload place dan price ke tabel database tb_delivery_cost.

Gambar 4.44 Antarmuka halaman tambah biaya antar jemput admin

Gambar diatas merupakan implementasi dari tambah layanan antar jemput. Ketika admin mengklik *button* tambah pada halaman layanan antar jemput maka akan diarahkan ke halaman tambah layanan antar jemput. Disini admin dapat menginputkan daerah dan harga layanan antar jemput baru, ketika dirasa sudah benar admin dapat mengklik *button* simpan. Otomatis data yang ditambahkan akan masuk ke list layanan antar jemput.

4.1.21 Halaman Ubah Biaya Antar Jemput Admin

```

31 public function edit($id_delivery_cost)
32 {
33     $data['delivery_cost'] = $this->db->where('id_delivery_cost', $id_delivery_cost)->get('tb_delivery_cost')->row();
34     $data['main_view'] = 'delivery_cost/edit_view';
35     $this->load->view('template_view', $data);
36 }
37 public function save_edit($id_delivery_cost)
38 {
39     $update['place'] = $this->input->post('place');
40     $update['price'] = $this->input->post('price');
41     $this->db->where('id_delivery_cost', $id_delivery_cost)->update('tb_delivery_cost', $update);
42
43     $this->session->set_flashdata('message', '<div class="alert alert-success">Biaya antar jemput berhasil diperbarui !</div>');
44     redirect('delivery_cost');
45 }

```

Gambar 4.45 Code program ubah biaya antar jemput admin

Pada code program ubah biaya antar jemput admin, yang terdapat pada bagian controller. Terdapat function edit yang berfungsi menampilkan form ubah biaya antar jemput dan function save_edit yang berfungsi menyimpan perubahan yang telah dilakukan.

Gambar 4.46 Antarmuka halaman ubah biaya antar jemput admin

Gambar diatas merupakan implementasi dari ubah layanan antar jemput. Ketika admin mengklik *button* ubah pada halaman layanan antar jemput maka akan diarahkan ke halaman ubah layanan antar jemput. Disini admin dapat menginputkan daerah dan harga yang ingin diubah, ketika sudah diinputkan *klik button* simpan maka otomatis data akan berubah.

4.1.22 Halaman *User Admin*

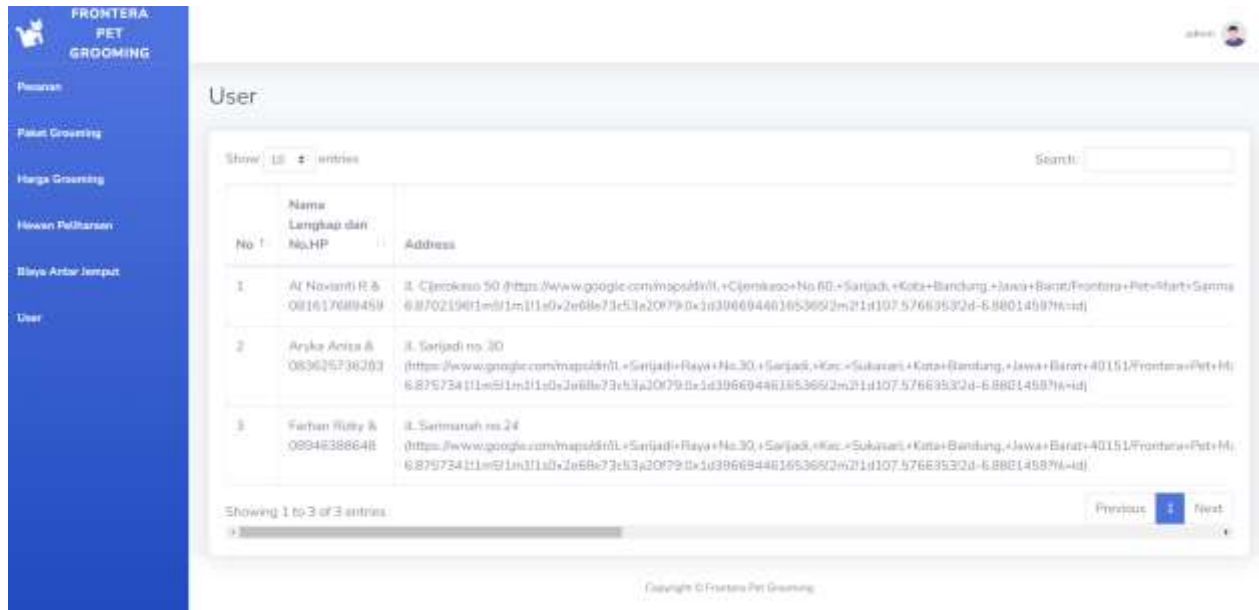
```

application > controllers > User.php > User
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3
4  class User extends CI_Controller
5  {
6      public function __construct()
7      {
8          parent::__construct();
9          if ($this->session->userdata('logged_in') != TRUE) {
10             redirect('login');
11         }
12     }
13     public function index()
14     {
15         $data['user'] = $this->db->where('role', 'user')->get('tb_user')->result();
16         $data['main_view'] = 'user/index_view';
17         $this->load->view('template_view', $data);
18     }
19 }

```

Gambar 4.47 Code program user admin

Pada code program *user*, yang terdapat pada controller. Terdapat beberapa function diantaranya adalah function construct dan function index. Function construct yang berfungsi untuk memastikan bahwa *user* sudah melakukan *login*. Function index yang berfungsi untuk menampilkan *template_view*.



Gambar 4.48 Antarmuka halaman user admin

Gambar diatas merupakan implementasi dari *user*. Pada halaman ini berisi informasi dari *user* seperti nama lengkap dan no. hp, alamat, dan email. Disini admin tidak dapat melakukan perubahan karena akun *user* merupakan kehendak dari *user* sehingga admin tidak dapat mengubahnya tanpa sepengetahuan dari *user*. Hal ini bertujuan untuk meminimalisir terjadinya perubahan secara tidak sengaja ke akun *user* yang dapat mengecewakan *user*.

4.1.23 Halaman Informasi Akun

```

19 public function save()
20 {
21     // Upload File
22     $destination_path = 'uploads/avatar/';
23     $config['upload_path'] = './' . $destination_path;
24     $config['allowed_types'] = '*';
25     $this->load->library('upload', $config);
26     if ($this->upload->do_upload('avatar')) {
27         $fileData = $this->upload->data();
28         $update['avatar'] = $destination_path . $fileData['file_name'];
29     }
30     $update['name'] = $this->input->post('name');
31     $update['address'] = $this->input->post('address');
32     $this->db->where('id_user', $this->session->id_user)->update('tb_user', $update);
33
34     $this->session->set_flashdata('message', '<div class="alert alert-success">Profil berhasil diperbarui !</div>');
35     redirect('profile');
36 }
37 }
38

```

Gambar 4.49 Code program informasi akun profil

Pada code program informasi akun profil, yang terdapat pada controller. Terdapat function save yang berfungsi ketika *user* atau admin mengubah profil maka akan tersimpan perubahannya menggunakan function save.

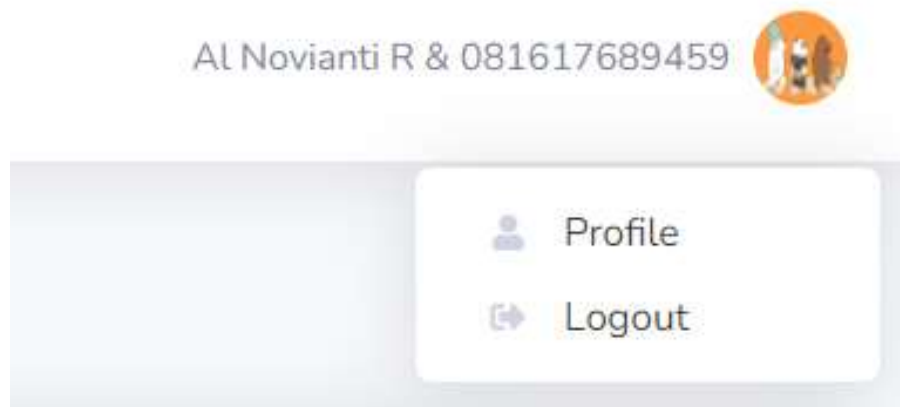
```

44 public function logout()
45 {
46     if ($this->session->userdata('logged_in') == TRUE) {
47         $this->session->sess_destroy();
48         redirect('login');
49     } else {
50         redirect('login');
51     }
52 }
53 }

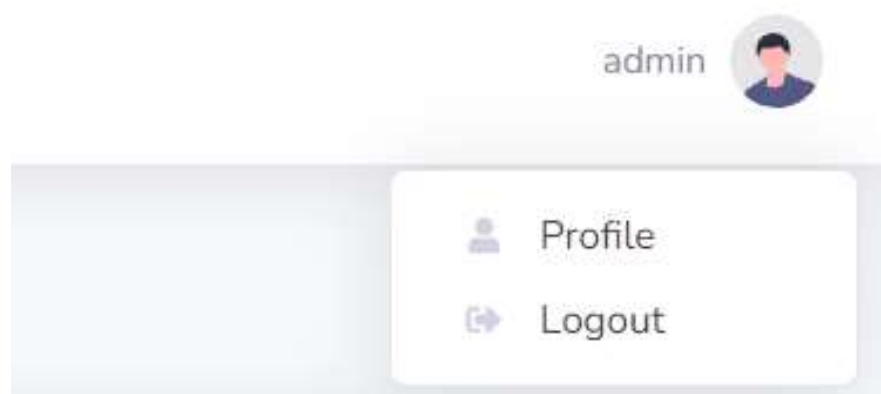
```

Gambar 4.50 Code program informasi akun logout

Pada code program informasi akun logout, yang terdapat pada controller. Terdapat function logout yang berisi `$this->session->sess_destroy(); redirect('login');` yang artinya ketika session `sess_destroy` maka akan di arahkan ke halaman *login*.



Gambar 4.51 Antarmuka halaman informasi akun user



Gambar 4.52 Antarmuka halaman informasi akun admin

Gambar diatas merupakan implementasi dari informasi akun dari *user* dan *admin*. Pada saat avatar di *klik* maka akan muncul menu *profile* dan *logout*. Ketika *diklik* *profile* maka akan diarahkan menuju halaman *profile*. Sedangkan apabila di *klik* *logout* maka akun otomatis akan keluar dan diarahkan ke menu *login*.

4.2 Pengujian dan Hasil Pengujian

Pengujian aplikasi yang dilakukan adalah pengujian instalasi, kompatibilitas dan fungsionalitas. Pengujian instalasi adalah memeriksa apakah aplikasi perangkat lunak berhasil diinstal & berfungsi seperti yang diharapkan setelah instalasi. [13]

4.2.1 Pengujian dan Hasil Pengujian Black Box

Pada pengujian kami menggunakan pengujian *Black Box*, *Black Box* adalah teknik pengujian perangkat lunak berorientasi tes kondisi memasuki sistem dan menguji spesifikasi fungsi sistem. Pengujian *Black Box* sering digunakan temukan

fungsi yang tidak kompatibel dan satu set kondisi input pengguna yang tidak sesuai dengan kebutuhan fungsional aplikasi.

4.2.1.1 Pengujian dan Hasil Pengujian Black Box Pada User

Berikut ini adalah tabel pengujian pada menu *user* dan hasil pengujian pada menu *user*.

Tabel 4.1 Pengujian user

No.	Menu	Submenu	Jenis Pengujian
1.	Registrasi	Membuat akun	<i>Black Box</i>
2.	Login	Masuk akun	<i>Black Box</i>
3.	Home	Menampilkan informasi Frontera <i>Pet grooming Grooming</i>	<i>Black Box</i>
4.	Grooming	Menampilkan katalog harga layanan <i>pet grooming grooming</i> serta harga layanan antar jemput	<i>Black Box</i>
5.	Order	Menampilkan form buat pesanan, list pesanan, dan informasi <i>user</i>	<i>Black Box</i>
6.	Buat pesanan	Menampilkan form pemesanan	<i>Black Box</i>
7.	Pesanan	Menampilkan list pesanan yang sudah dipesan	<i>Black Box</i>
8.	Search pesanan	Mencari data pesanan yang telah dipesan	<i>Black Box</i>
9.	Profil	Menampilkan informasi <i>user</i> seperti avatar, nama lengkap dan no. hp, alamat, email.	<i>Black Box</i>
10.	Logout	Keluar Akun	<i>Black Box</i>

Tabel 4.2 Hasil pengujian user

No.	Menu	Jenis Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil Pengujian
-----	------	-----------------	-----------------	-----------------------	-----------------

1.	Registrasi	<i>Black Box</i>	Memasukkan data <i>user</i> seperti nama lengkap dan no.hp, alamat, email, dan password	<i>User</i> berhasil membuat akun	Sesuai
2.	<i>Login</i>	<i>Black Box</i>	Memasukkan email dan password yang telah terdaftar	<i>User</i> berhasil <i>login</i> dan masuk ke halaman home	Sesuai
3.	Home	<i>Black Box</i>	<i>User</i> klik menu home	Menampilkan informasi Frontera <i>Pet grooming</i> <i>Grooming</i>	Sesuai
4.	<i>Grooming</i>	<i>Black Box</i>	<i>User</i> klik menu <i>grooming</i>	Menampilkan katalog harga layanan <i>pet grooming</i> serta harga layanan antar jemput	Sesuai
5.	<i>Order</i>	<i>Black Box</i>	<i>User</i> klik menu <i>order</i>	Menampilkan form buat pesanan, list pesanan, dan informasi <i>user</i>	Sesuai
6.	Buat pesanan	<i>Black Box</i>	<i>User</i> klik menu buat pesanan dan mengisi form pemesanan	Menampilkan form pemesanan	Sesuai

7.	Pesanan	<i>Black Box</i>	<i>User klik menu pesanan</i>	Menampilkan list pesanan yang sudah dipesan	Sesuai
8.	<i>Search pesanan</i>	<i>Black Box</i>	<i>User mengisi menu search dengan data yang ingin ditampilkan</i>	Menampilkan list data yang sudah dituliskan di menu <i>search</i>	Sesuai
9.	Profil	<i>Black Box</i>	<i>User klik menu profil</i>	Menampilkan profil <i>user</i> seperti avatar, nama lengkap dan no.hp, alamat, dan email pada menu ini <i>user</i> dapat mengedit profilnya	Sesuai
10.	<i>Logout</i>	<i>Black Box</i>	<i>User klik menu logout</i>	<i>User berhasil keluar akun dan masuk ke menu login</i>	Sesuai

4.2.1.2 Pengujian dan Hasil Pengujian Black Box Pada Admin

Setelah dilakukan pengujian terhadap menu pada *user*, selanjutnya melakukan pengujian terhadap menu pada admin. Berikut ini adalah tabel pengujian menu admin dan hasil pengujian menu admin.

Tabel 4. 3 Pengujian admin

No.	Menu	Submenu	Jenis Pengujian
1.	<i>Login</i>	Masuk akun	<i>Black Box</i>
2.	Dashboard	Menampilkan foto Frontera <i>Pet grooming Grooming</i> pada sidebar terdapat menu pesanan,	<i>Black Box</i>

		paket <i>grooming</i> , harga <i>grooming</i> , hewan peliharaan, biaya antar jemput, dan <i>user</i>	
3.	Pesanan	Menampilkan list pesanan pelanggan	<i>Black Box</i>
4.	Buat pesanan	Menampilkan form pemesanan	<i>Black Box</i>
5.	<i>Search</i> pesanan	Mencari data pesanan pelanggan	<i>Black Box</i>
6.	Aksi pesanan	Mengubah status pada pelanggan (proses dan selesai)	<i>Black Box</i>
7.	Paket <i>grooming</i>	Menampilkan list paket <i>grooming</i>	<i>Black Box</i>
8.	Tambah paket <i>grooming</i>	Menambah paket <i>grooming</i> yang baru	<i>Black Box</i>
9.	Ubah paket <i>grooming</i>	Mengubah atau mengedit paket <i>grooming</i>	<i>Black Box</i>
10.	Hapus paket <i>grooming</i>	Menghapus paket <i>grooming</i>	<i>Black Box</i>
11.	<i>Search</i> paket <i>grooming</i>	Mencari data paket <i>grooming</i>	<i>Black Box</i>
12.	Harga <i>grooming</i>	Menampilkan list harga <i>grooming</i>	<i>Black Box</i>
13.	Tambah harga <i>grooming</i>	Menambah harga <i>grooming</i> yang baru	<i>Black Box</i>
14.	Ubah harga <i>grooming</i>	Mengubah atau mengedit harga <i>grooming</i>	<i>Black Box</i>
15.	Hapus harga <i>grooming</i>	Menghapus harga <i>grooming</i>	<i>Black Box</i>
16.	<i>Search</i> harga <i>grooming</i>	Mencari data harga <i>grooming</i>	<i>Black Box</i>
17.	Hewan peliharaan	Menampilkan list hewan peliharaan	<i>Black Box</i>

18.	Tambah hewan peliharaan	Menambah hewan peliharaan yang baru	<i>Black Box</i>
19.	Ubah hewan peliharaan	Mengubah atau mengedit hewan peliharaan	<i>Black Box</i>
20.	Hapus hewan peliharaan	Menghapus hewan peliharaan	<i>Black Box</i>
21.	<i>Search</i> hewan peliharaan	Mencari data hewan peliharaan	<i>Black Box</i>
22.	Biaya antar jemput	Menampilkan list biaya antar jemput	<i>Black Box</i>
23.	Tambah biaya antar jemput	Menambah biaya antar jemput yang baru	<i>Black Box</i>
24.	Ubah biaya antar jemput	Mengubah atau mengedit biaya antar jemput	<i>Black Box</i>
25.	Hapus biaya antar jemput	Menghapus biaya antar jemput	<i>Black Box</i>
26.	<i>Search</i> biaya antar jemput	Mencari data biaya antar jemput	<i>Black Box</i>
27.	<i>User</i>	Menampilkan data <i>user</i>	<i>Black Box</i>
28.	<i>Search user</i>	Mencari data <i>user</i>	<i>Black Box</i>
29.	Profile	Menampilkan informasi admin seperti avatar, nama lengkap dan no. hp, alamat, email.	<i>Black Box</i>
30.	<i>Logout</i>	Keluar Akun	<i>Black Box</i>

Tabel 4.4 Hasil pengujian admin

No.	Menu	Jenis Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil Pengujian
1.	<i>Login</i>	<i>Black Box</i>	Memasukkan email dan password admin	Admin berhasil <i>login</i> dan masuk	Sesuai

				ke halaman dashboard	
2.	Dashboard	<i>Black Box</i>	Ketika <i>user</i> berhasil <i>login</i> akan otomatis masuk ke halaman dashboard	Menampilkan foto <i>Frontera Pet grooming</i> pada sidebar terdapat menu pesanan, paket <i>grooming</i> , harga <i>grooming</i> , hewan peliharaan, biaya antar jemput, dan <i>user</i>	Sesuai
3.	Pesanan	<i>Black Box</i>	Admin <i>klik</i> menu pesanan	Menampilkan list pesanan pelanggan	Sesuai
4.	Buat pesanan	<i>Black Box</i>	Admin <i>klik button</i> buat pesanan	Menampilkan form pemesanan	Sesuai
5.	<i>Search</i> pesanan	<i>Black Box</i>	Admin mengisi menu <i>search</i> dengan data pesanan yang ingin ditampilkan	Menampilkan list data pesanan yang sudah dituliskan di menu <i>search</i>	Sesuai
6.	Aksi pesanan	<i>Black Box</i>	Admin memilih <i>button</i> aksi (proses dan selesai) untuk status yang ditampilkan pada <i>user</i>	Status pada <i>user</i> akan otomatis berganti sesuai yang admin inputkan	Sesuai

7.	Paket <i>grooming</i>	<i>Black Box</i>	Admin <i>klik</i> menu paket <i>grooming</i>	Menampilkan list paket <i>grooming</i> serta menu tambah, ubah, hapus, dan <i>search</i> paket <i>grooming</i>	Sesuai
8.	Tambah paket <i>grooming</i>	<i>Black Box</i>	Admin <i>klik</i> <i>button</i> tambah paket <i>grooming</i>	Menampilkan form tambah paket <i>grooming</i>	Sesuai
9.	Ubah paket <i>grooming</i>	<i>Black Box</i>	Admin <i>klik</i> <i>button</i> ubah	Menampilkan form ubah paket <i>grooming</i>	Sesuai
10.	Hapus paket <i>grooming</i>	<i>Black Box</i>	Admin <i>klik</i> <i>button</i> hapus	Otomatis data paket <i>grooming</i> akan terhapus	Sesuai
11.	<i>Search</i> paket <i>grooming</i>	<i>Black Box</i>	Admin mengisi menu <i>search</i> dengan data paket <i>grooming</i> yang ingin ditampilkan	Menampilkan list data paket <i>grooming</i> yang sudah dituliskan di menu <i>search</i>	Sesuai
12.	Harga <i>grooming</i>	<i>Black Box</i>	Admin <i>klik</i> menu harga <i>grooming</i>	Menampilkan list harga <i>grooming</i> serta menu tambah, ubah, hapus, dan <i>search</i> harga <i>grooming</i>	Sesuai
13.	Tambah harga <i>grooming</i>	<i>Black Box</i>	Admin <i>klik</i> <i>button</i> tambah harga <i>grooming</i>	Menampilkan form tambah harga <i>grooming</i>	Sesuai

14.	Ubah harga grooming	<i>Black Box</i>	Admin klik <i>button</i> ubah	Menampilkan form ubah harga grooming	Sesuai
15.	Hapus harga grooming	<i>Black Box</i>	Admin klik <i>button</i> hapus	Otomatis harga grooming akan terhapus	Sesuai
16.	<i>Search</i> harga grooming	<i>Black Box</i>	Admin mengisi menu <i>search</i> dengan data harga grooming yang ingin ditampilkan	Menampilkan list data harga grooming yang sudah dituliskan di menu <i>search</i>	Sesuai
17.	Hewan peliharaan	<i>Black Box</i>	Admin klik menu hewan peliharaan	Menampilkan list hewan peliharaan serta menu tambah, ubah, hapus, dan <i>search</i> hewan peliharaan	Sesuai
18.	Tambah hewan peliharaan	<i>Black Box</i>	Admin klik <i>button</i> tambah hewan peliharaan	Menampilkan form tambah hewan peliharaan	Sesuai
19.	Ubah hewan peliharaan	<i>Black Box</i>	Admin klik <i>button</i> ubah	Menampilkan form ubah hewan peliharaan	Sesuai
20.	Hapus hewan peliharaan	<i>Black Box</i>	Admin klik <i>button</i> hapus	Otomatis hewan peliharaan akan terhapus	Sesuai
21.	<i>Search</i> hewan peliharaan	<i>Black Box</i>	Admin mengisi menu <i>search</i> dengan data hewan peliharaan	Menampilkan list data hewan peliharaan yang	Sesuai

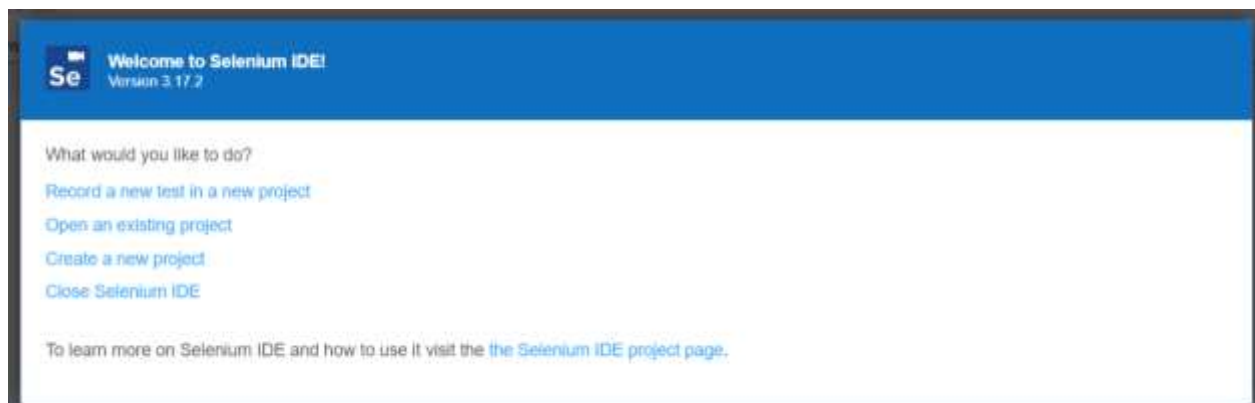
			yang ingin ditampilkan	sudah dituliskan di menu <i>search</i>	
22.	Biaya antar jemput	<i>Black Box</i>	Admin <i>klik</i> menu biaya antar jemput	Menampilkan list biaya antar jemput serta menu tambah, ubah, hapus, dan <i>search</i> biaya antar jemput	Sesuai
23.	Tambah biaya antar jemput	<i>Black Box</i>	Admin <i>klik button</i> tambah biaya antar jemput	Menampilkan form tambah biaya antar jemput	Sesuai
24.	Ubah biaya antar jemput	<i>Black Box</i>	Admin <i>klik button</i> ubah	Menampilkan form ubah biaya antar jemput	Sesuai
25.	Hapus biaya antar jemput	<i>Black Box</i>	Admin <i>klik button</i> hapus	Otomatis biaya antar jemput akan terhapus	Sesuai
26.	<i>Search</i> biaya antar jemput	<i>Black Box</i>	Admin mengisi menu <i>search</i> dengan data biaya antar jemput yang ingin ditampilkan	Menampilkan list data biaya antar jemput yang sudah dituliskan di menu <i>search</i>	Sesuai
27.	<i>User</i>	<i>Black Box</i>	Admin <i>klik</i> menu <i>user</i>	Menampilkan profil <i>user</i> seperti nama lengkap dan no.hp, alamat, dan email	Sesuai

28.	<i>Search user</i>	<i>Black Box</i>	Admin mengisi menu <i>search</i> dengan data <i>user</i> yang ingin ditampilkan	Menampilkan list data <i>user</i> yang sudah dituliskan di menu <i>search</i>	Sesuai
29.	Profile	<i>Black Box</i>	Admin klik menu profile	Menampilkan profil admin seperti avatar, nama lengkap dan no.hp, alamat, dan email pada menu ini admin dapat mengedit profilnya	Sesuai
30.	<i>Logout</i>	<i>Black Box</i>	Admin klik menu <i>logout</i>	Admin berhasil keluar akun dan masuk ke menu <i>login</i>	Sesuai

4.2.2 Pengujian dan Hasil Pengujian Selenium IDE

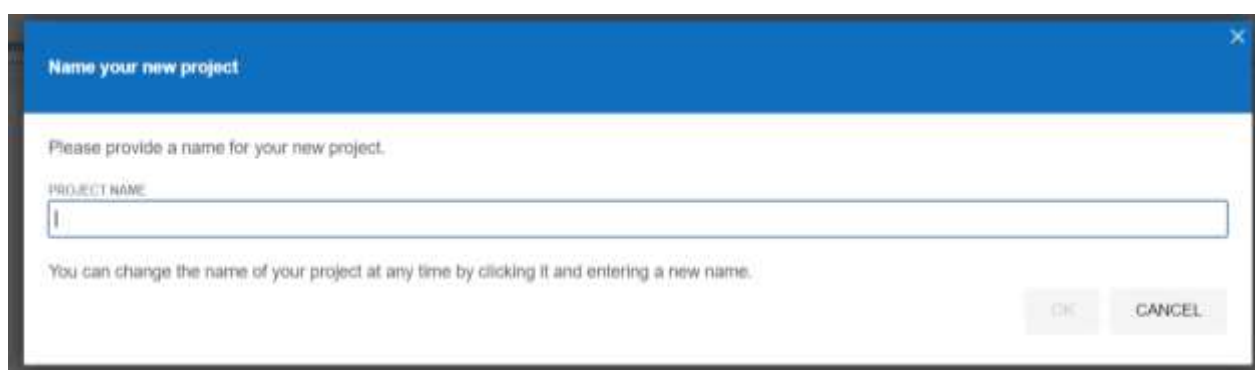
Pada bagian pengujian dan hasil pengujian aplikasi, di paparkan hasil pengujian pada setiap proses berjalannya sistem yang telah di bangun. Dalam melakukan pengujian digunakan selenium IDE atau extension khusus untuk pengujian. Selenium merupakan sebuah *tools auto testing* yang digunakan untuk mengotomatisasi tes aplikasi web yang dilakukan pada browser. Berikut adalah pengujian dan hasil pengujian sistem:

Cara kerja Selenium IDE adalah pertama kita akan memilih *Record a new test in a new project*.



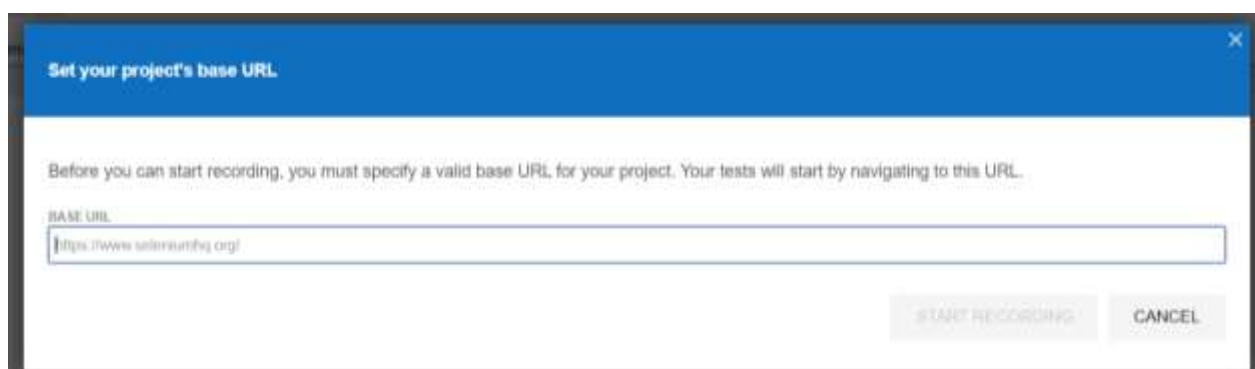
Gambar 4.53 Selenium IDE

Kemudian kita memasukan nama *test* yang akan kita jalankan, disini bisa diisi apa saja bebas.



Gambar 4.54 Cara kerja selenium IDE

Kemudian kita akan memasukan link URL *website* yang akan kita lakukan *code coverage* atau pengetestan.



Gambar 4.55 Cara kerja selenium IDE

Kemudian setelah kita memasukkan link yang ingin kita test, kita bisa menekan tombol start recording, dan kita akan langsung mengarah ke tab window baru dengan

link yang sudah dimasukkan tadi. Dan pada bagian ini kita hanya tinggal lakukan proses yang sesuai dengan *website*. Misalnya disini kita lakukan saja input data *user*. Dan semua proses yang kita lakukan akan terkekam.



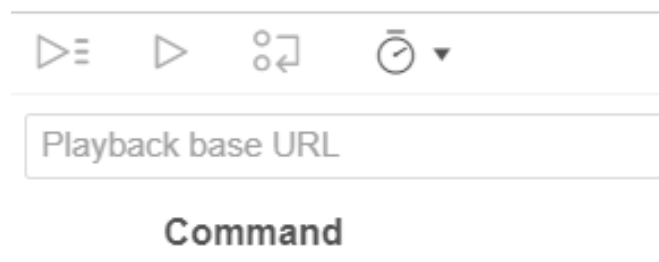
Gambar 4.56 Prsoses testing selenium IDE

Setelah kita melakukan record test, kita dapat selesaikan dengan menekan tombol pada bagian atas kanan berwarna merah kotak.



Gambar 4.57 Record testing

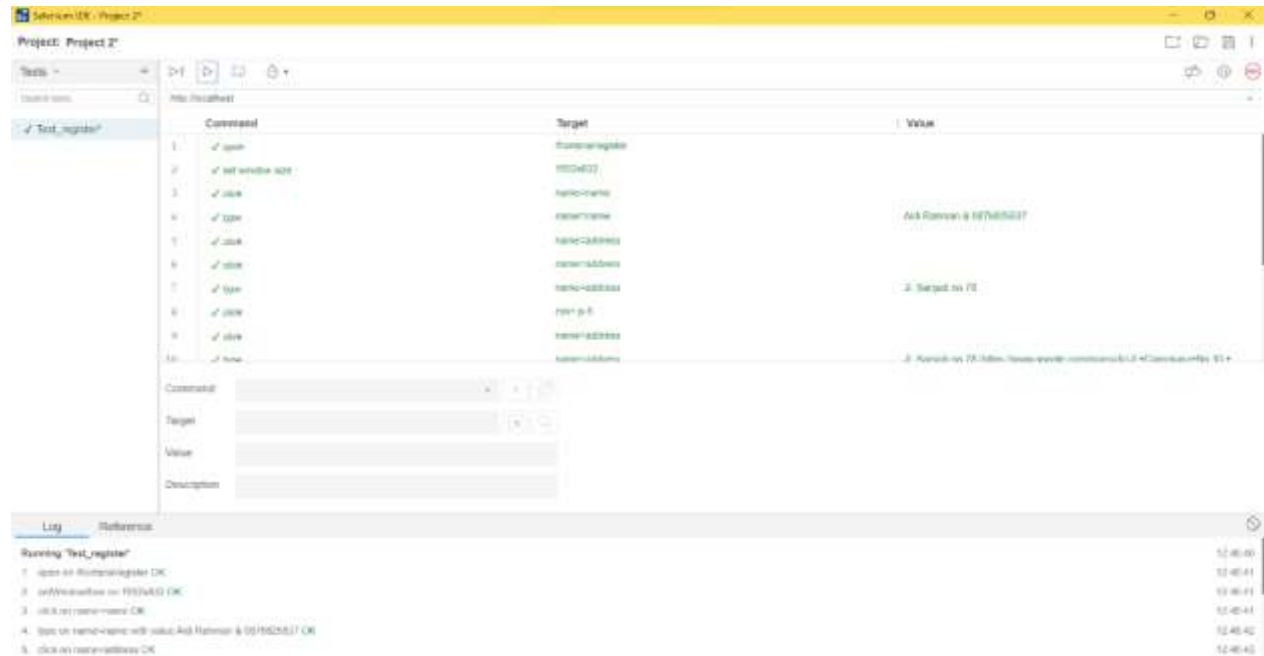
Kemudian setelah kita simpan hasil record tadi. Kita dapat menekan tombol Run all test untuk mengulang recordan yang tadi untuk melihat hasil ujinya.



Gambar 4.58 Run testing

Dan terakhir kita dapat melihat hasil uji nya dengan, jika OK berarti proses tersebut berhasil tanpa error. Dan bagian waktu pada kanan bawah merupakan waktu ketika testingnya di lakukan, dan kita dapat melihat berapa lama satu proses dapat bekerja dengan melihat berapa selisih detik dari setiap proses.

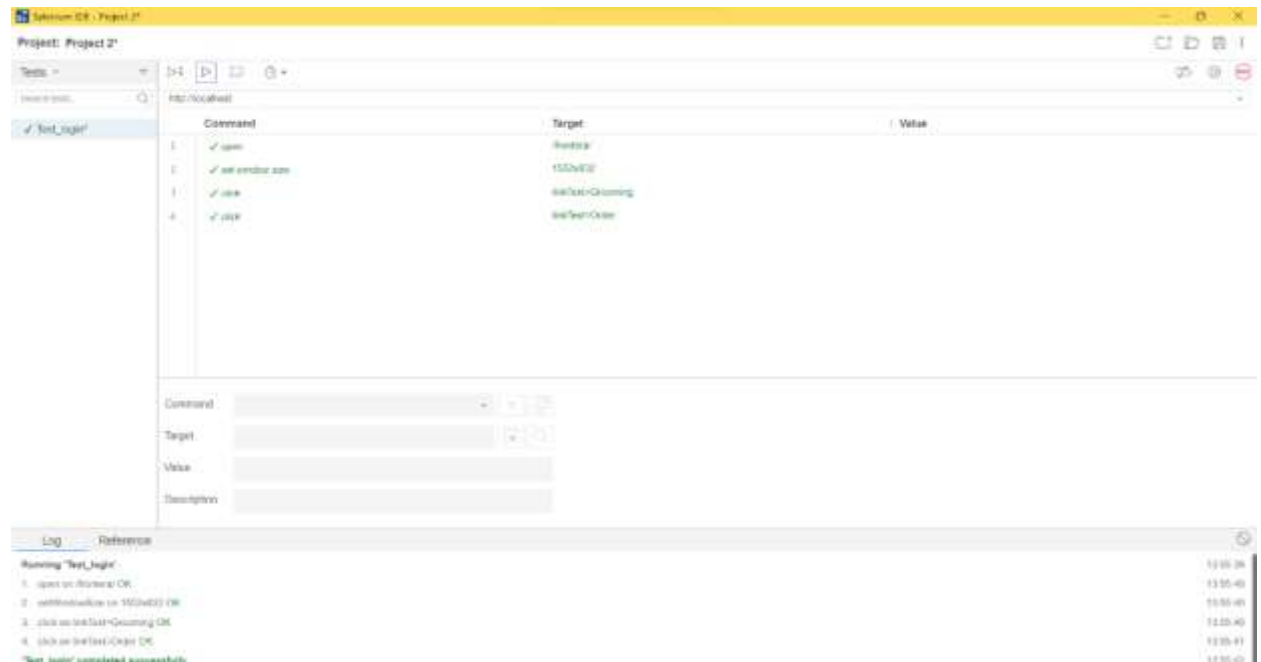
4.2.2.1 Pengujian dan Hasil Pengujian Menu Register



Gambar 4.59 Pengujian dan hasil pengujian menu register

Dalam pengujian ini dilakukan oleh *user* dengan memasukkan informasi *user* seperti, nama lengkap dan no.telp, alamat, email dan password. Setelah pengisian informasi *user* dilakukan *klik* register maka data telah di daftarkan maka sistem akan mengarahkan ke halaman *login*. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

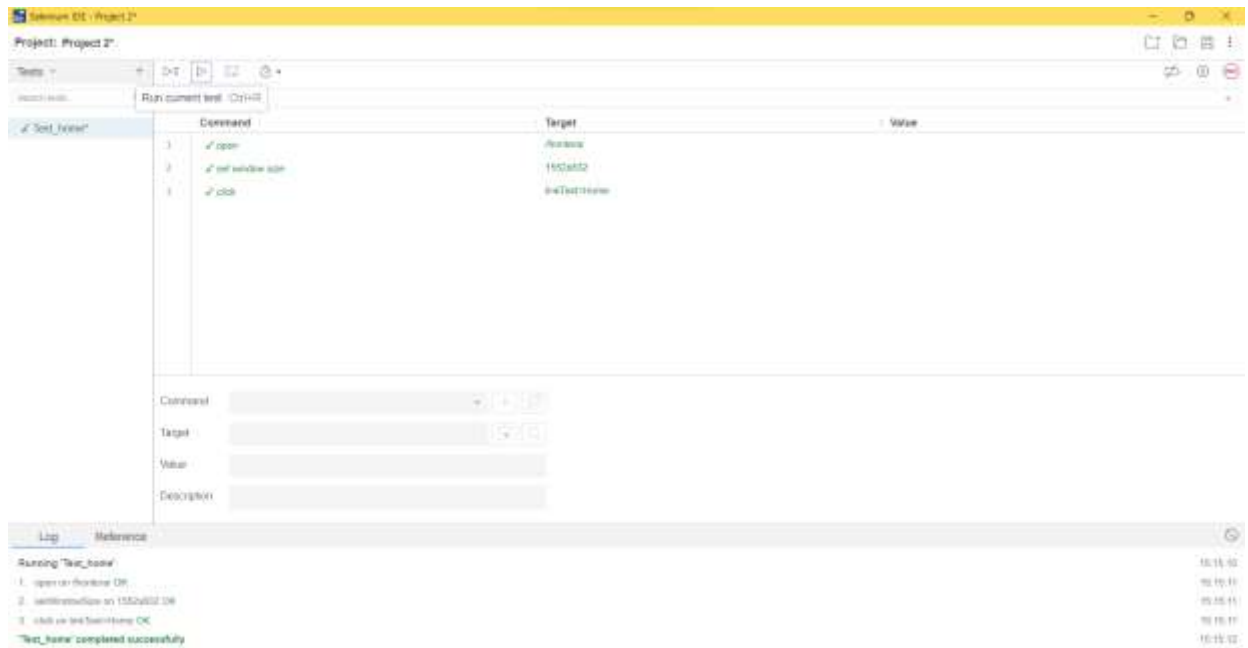
4.2.2.2 Pengujian dan Hasil Pengujian Menu Login



Gambar 4.60 Pengujian dan hasil pengujian menu login

Dalam pengujian ini dilakukan oleh *user* atau admin dengan memasukkan email dan password yang telah dibuat. Setelah pengisian email dan password dilakukan *klik login* dan jika sesuai dengan email dan password yang telah di daftarkan maka sistem akan mengarahkan ke halaman dashboard resepsionis. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

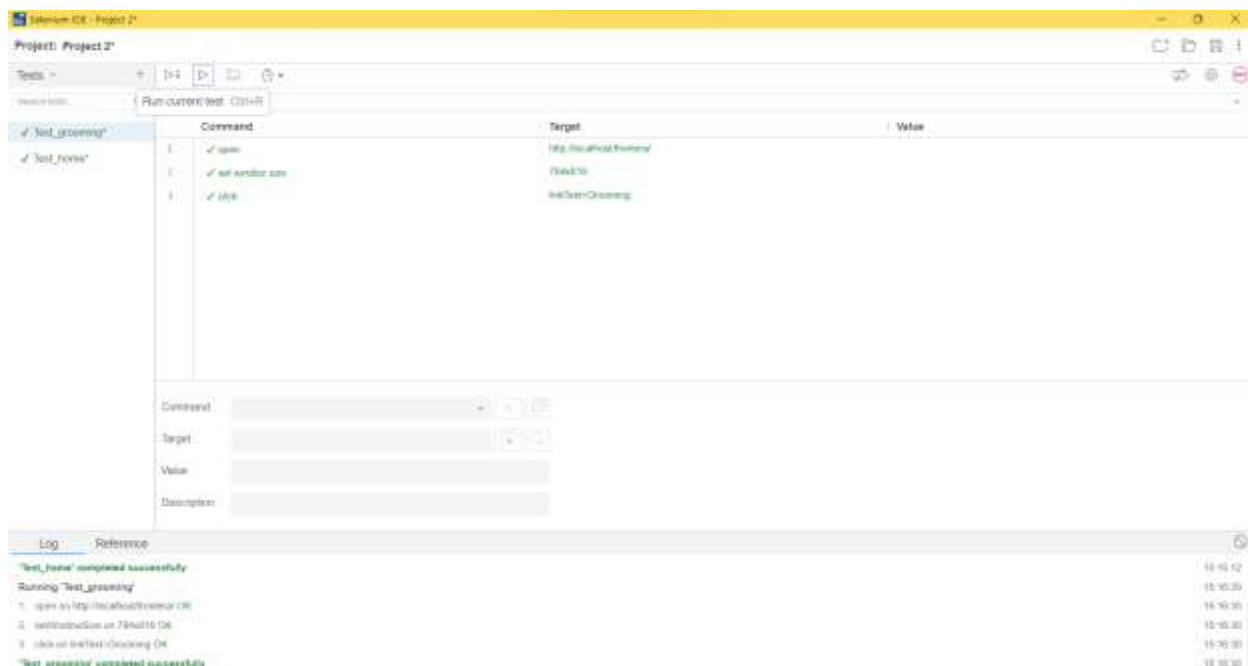
4.2.2.3 Pengujian dan Hasil Pengujian Menu Home



Gambar 4.61 Pengujian dan hasil pengujian menu home

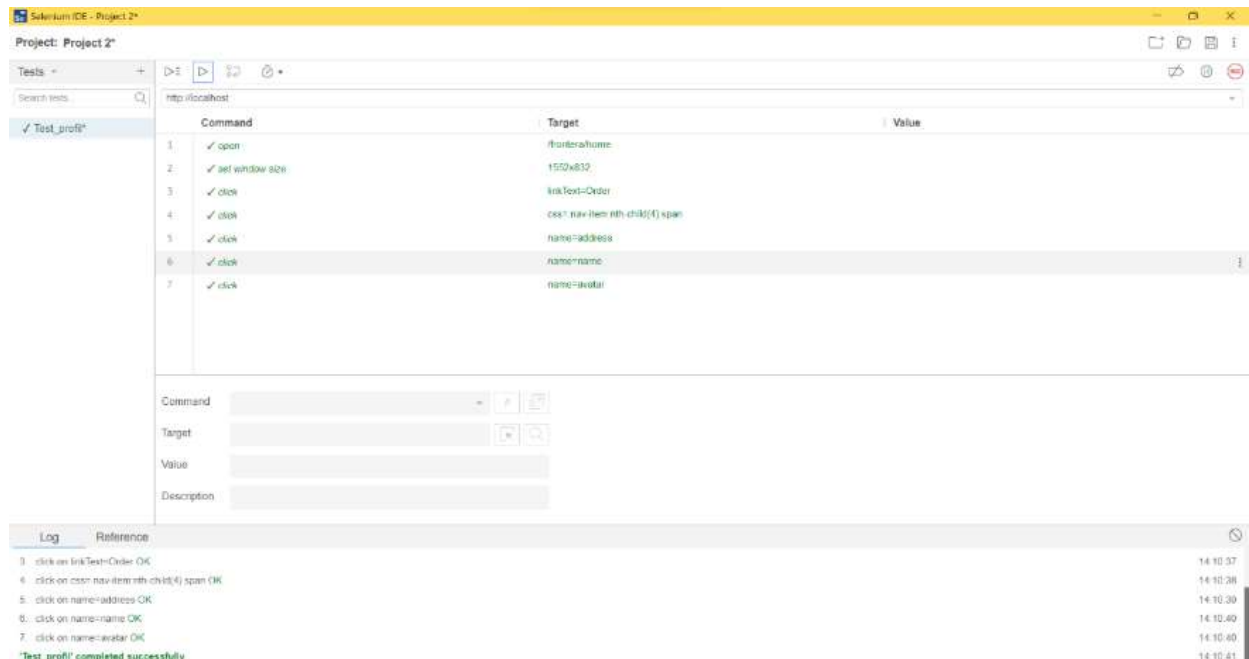
Dalam pengujian ini dilakukan oleh *user* pada saat masuk ke halaman home, maka halaman home akan menampilkan informasi tentang Frontera Pet *Grooming*. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

4.2.2.4 Pengujian dan Hasil Pengujian Menu *Grooming*



Gambar 4.62 Pengujian dan hasil pengujian menu grooming

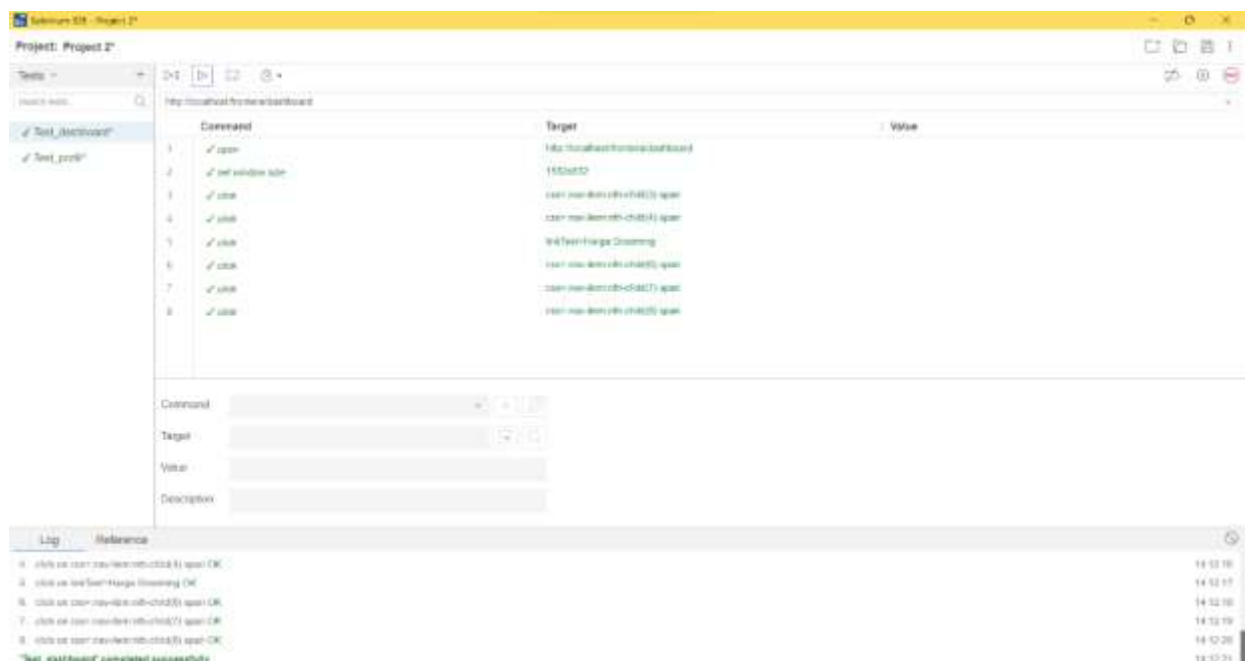
4.2.2.6 Pengujian dan Hasil Pengujian Menu Profil



Gambar 4.64 Pengujian dan hasil pengujian menu profil

Dalam pengujian ini dilakukan oleh *user* pada saat masuk ke halaman profil, maka halaman profil akan menampilkan informasi *user* seperti, avatar, nama lengkap dan no.hp, alamat, dan email. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

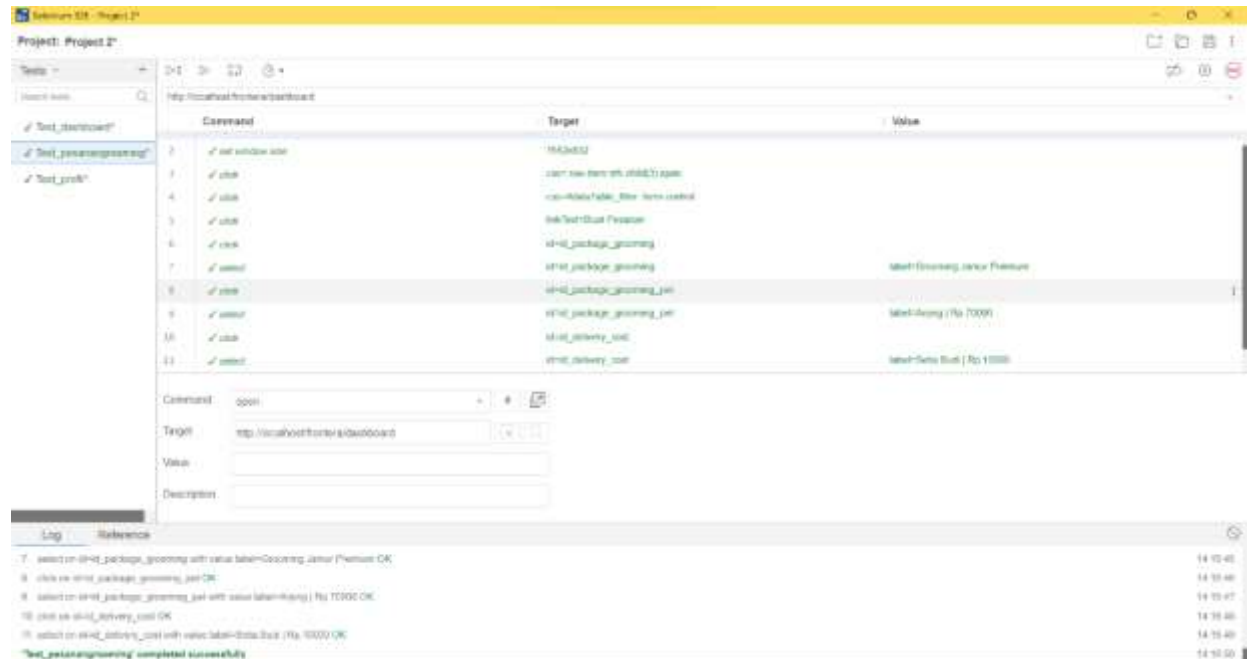
4.2.2.7 Pengujian dan Hasil Pengujian Menu DASHBOARD Admin



Gambar 4.65 Pengujian dan hasil pengujian menu dashboard admin

Dalam pengujian ini dilakukan oleh admin pada saat masuk ke halaman dashboard, maka halaman dashboard akan menampilkan foto dari Frontera Pet Grooming. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

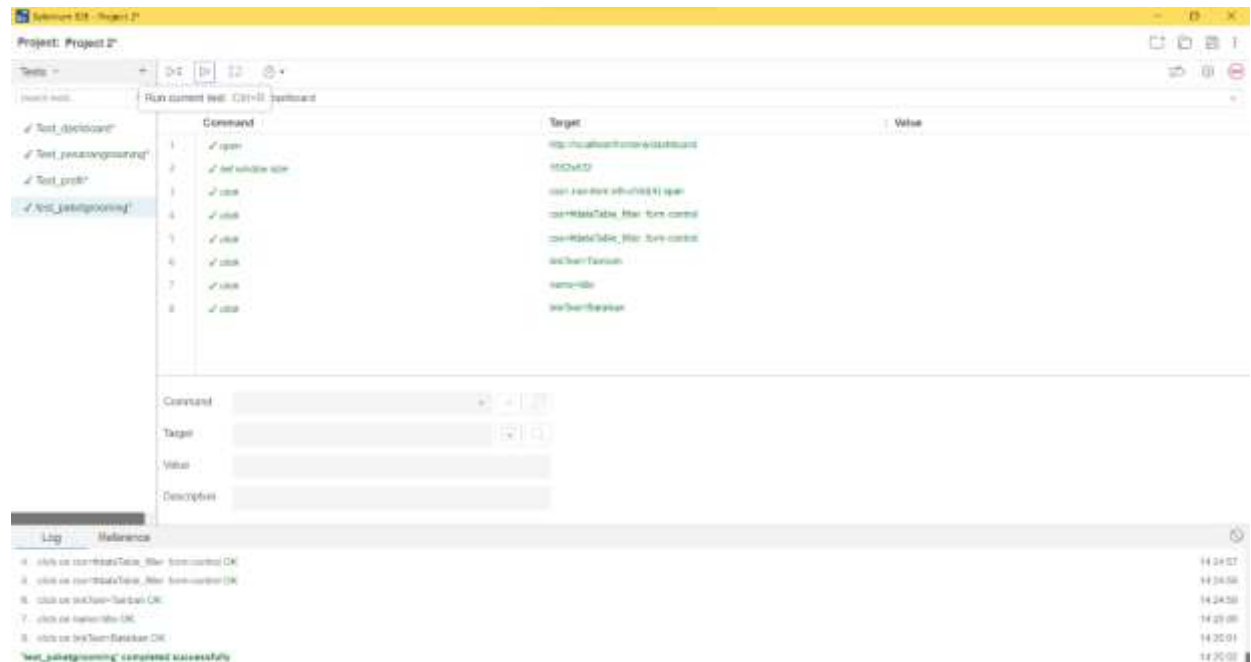
4.2.2.8 Pengujian dan Hasil Pengujian Menu Pesanan Admin



Gambar 4.66 Pengujian dan hasil pengujian menu pesanan admin

Dalam pengujian ini dilakukan oleh admin pada saat masuk ke halaman pesanan, maka halaman pesanan akan menampilkan list pesanan dan form pemesanan. Pada form pemesanan dapat digunakan untuk melakukan pemesanan. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

4.2.2.9 Pengujian dan Hasil Pengujian Menu Paket *Grooming* Admin



Gambar 4.67 Pengujian dan hasil pengujian menu paket grooming admin

Dalam pengujian ini dilakukan oleh admin pada saat masuk ke halaman paket grooming, maka halaman paket grooming akan menampilkan informasi paket grooming. Pada halaman ini dapat dilakukan tambah paket grooming, ubah paket grooming dan hapus paket grooming. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

4.2.2.12 Pengujian dan Hasil Pengujian Menu Biaya Antar Jemput Admin

Test Case	Step	Command	Target	Value
Test_Mayantarjemput	11	click	name=glbbtn	Tambah
	12	type	name=glbbtn	
	13	click	css=btn primary	
	14	click	css=btn primary	
	15	click	name=glbbtn	Ubah
	16	type	name=glbbtn	
	17	click	name=glbbtn	Hapus
	18	type	name=glbbtn	
	19	click	css=btn primary	
	20	click	css=btn primary	

Log	Reference
11. type on name=glbbtn with value: Ubah OK	14:20:45
12. click on name=glbbtn OK	14:20:45
13. type on name=glbbtn with value: 7000 OK	14:20:46
14. click on css=btn primary OK	14:20:46
15. click on css=btn primary OK	14:20:46
16. type on name=glbbtn with value: 7000 OK	14:20:46
17. click on name=glbbtn OK	14:20:46
18. type on name=glbbtn with value: Hapus OK	14:20:46
19. click on css=btn primary OK	14:20:46
20. click on css=btn primary OK	14:20:46
Test_Mayantarjemput completed successfully	

Gambar 4.70 Pengujian dan hasil pengujian menu biaya antar jemput admin

Dalam pengujian ini dilakukan oleh admin pada saat masuk ke halaman biaya antar jemput, maka halaman biaya antar jemput akan menampilkan informasi biaya antar jemput. Pada halaman ini dapat dilakukan tambah biaya antar jemput, ubah biaya antar jemput dan hapus biaya antar jemput. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

Dalam pengujian ini dilakukan oleh admin pada saat masuk ke halaman profil, maka halaman profil akan menampilkan informasi admin seperti, avatar, nama lengkap dan no.hp, alamat, dan email. Dalam pengujian ini sistem berjalan dengan baik sesuai dengan permintaan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pembahasan yang sebelumnya terdapat beberapa masalah pada Frontera *Petshop*. Dengan adanya Aplikasi *Pet Grooming* berbasis *website* ini maka dapat ditarik kesimpulan yaitu :

1. Aplikasi *pet grooming* ini dapat diakses dari mana saja karena berbasis online. Informasi pesanan *pet grooming* juga sudah dilakukan secara terkomputerisasi sehingga resiko kehilangan data lebih kecil dibanding masih menggunakan cara konvensional.
2. Aplikasi *pet grooming* mampu menampilkan informasi mengenai informasi Frontera *pet grooming* seperti katalog harga *grooming* dan biaya antar jemput agar pelanggan tidak harus datang langsung ke *petshop* untuk mencari informasi.

5.2 Saran

Agar sistem yang dirancang dapat digunakan berjalan sesuai dengan yang diharapkan, maka ada beberapa saran yang dapat dijadikan untuk bahan pertimbangan selanjutnya yaitu :

1. Kedepannya disarankan untuk dibuat sistem berbasis android agar lebih simple dan fleksibel dalam penggunaannya. Agar pelanggan lebih mudah dalam mengakses Frontera *Pet Grooming*.
2. Disarankan dapat menambahkan fitur pembayaran secara transfer atau pembayaran via online lainnya. Karena dimasa sekarang orang jarang membawa uang *cash* sehingga lebih baik jika ditambahkan fitur tersebut.
3. Disarankan kedepannya Aplikasi *Pet Grooming* ini dapat ditambahkan dengan penjualan barang-barang kebutuhan seperti pada *petshop*. Dengan ditambahkan penjualan barang – barang diharapkan mampu menambah daya tarik pelanggan serta meningkatkan penjualan Frontera *Petshop*.

DAFTAR PUSTAKA

- [1] A. Mubarak, "RANCANG BANGUN APLIKASI WEB SEKOLAH MENGGUNAKAN UML (UNIFIED MODELING LANGUAGE) DAN BAHASA PEMROGRAMAN PHP (PHP HYPERTEXT PREPROCESSOR) BERORIENTASI OBJEK," *JIKO (Jurnal Informatika dan Komputer)*, vol. 2, no. 112, p. 7, 2019.
- [2] o. A. O. Sari, A. Abdilah and Sunarti, *Web Programming*, Jakarta: Graha Ilmu, 2019.
- [3] o. A. O. Sari, A. Abdilah and Sunarti, *Web Programming*, Jakarta: Graha Ilmu, 2019.
- [4] Y. N. R. ElisaUsada 1, "RANCANG BANGUN SISTEM INFORMASIJADWAL PERKULIAHAN BERBASIS JQUERY MOBILEDENGAN MENGGUNAKAN PHP DAN MySQL," *Jurnal Infotel*, vol. 4, no. 64, p. 12, 2014.
- [5] S. Eddy Purwanto, "PERBANDINGAN STRATEGI REPLIKASI PADA SISTEM BASIS DATA TERDISTRIBUSI," *Jurnal Ilmiah*, vol. 10, no. 33, p. 8, 2012.
- [6] S. R. Adi Widarma1, "PERANCANGAN APLIKASI GAJI KARYAWAN PADA PT. PP LONDON SUMATRA INDONESIA Tbk. GUNUNG MALAYU ESTATE - KABUPATEN ASAHAN," *Jurnal Teknologi Informasi*, vol. 1, no. 50, p. 8, 2017.
- [7] S. A. S. K. W. M. Ali Mustopa1, "Analisa Kepuasan Pengguna *Website* Layanan Akademik Kemahasiswaan (LYKAN) Menggunakan Metode Webqual 4.0," *Jurnal Ekonomi & Manajemen Universitas Bina Sarana Informatika*, vol. 18, no. 21, p. 7, 2020.

- [8] W. komputer, Mudah Membuat Aplikasi SMS Gateway dengan Codeigniter, Jakarta: PT Elex Media Komputindo, 2014.
- [9] M. T. Ramlah, "Tingkah laku *grooming* kelelawar *Pteropus alecto* di penangkaran ex-situ," *Zootec*, p. 41, 2021.
- [10] P. R. A. Yudi Permana¹⁾, "PERANCANGAN SISTEM INFORMASI PENJUALAN PERUMAHAN MENGGUNAKAN METODE SDLC PADA PT. MANDIRI LAND PROSPEROUS BERBASIS MOBILE," *Jurnal Teknologi Pelita Bangsa*, vol. 10, no. 70, p. 15, 2019.
- [11] M. A. Yunita Trimarsiah¹⁾, "ANALISIS DAN PERANCANGAN *WEBSITE* SEBAGAI SARANA INFORMASI PADA LEMBAGA BAHASA KEWIRAUSAHAAN DAN KOMPUTER AKMI BATURAJA," *Jurnal Ilmiah MATRIK*, vol. 19, no. 200, p. 10, 2017.
- [12] A. S. 2. C. E. F. 3. Muhammad Saed Novendri¹⁾, "APLIKASI INVENTARIS BARANG PADA MTS NURUL ISLAM DUMAI MENGGUNAKAN PHP DAN MYSQL," *JURNAL MANAJENEN DAN TEKNOLOGI INFORMASI*, vol. 10, no. 200, p. 12, 2019.
- [13] D. A. Wasesha, "PERANCANGAN APLIKASI LAYANAN SALON HEWAN SECARA DARING," p. 48, 1 AGUSTUS 2021.
- [14] Peranginangin, Perancangan sistem informasi e-commerce silky pet shop bekasi, Bandung: sandrajeanawuy@umika.ac.id, 2006.
- [15] Nugroho, sistem informasi penjualan pada lucky paw's petshop berbasis web, Palembang: Jurnal Teknik Informatika Politeknik Sekayu, 2017.
- [16] M. A. Yunita Trimarsiah¹⁾, "ANALISIS DAN PERANCANGAN *WEBSITE* SEBAGAI SARANA INFORMASI PADA LEMBAGA BAHASA KEWIRAUSAHAAN DAN KOMPUTER AKMI BATURAJA," *Jurnal Ilmiah MATRIK*, vol. 19, no. 200, p. 10, 2017.

LAMPIRAN

CODE COVERAGE

1. Menu Register

Running 'Register'	13:32:26
1. open on /frontera/ OK	13:32:27
2. setWindowSize on 1552x832 OK	13:32:27
3. click on linkText=Home OK	13:32:27
4. click on linkText=Grooming OK	13:32:28
5. click on linkText=Order OK	13:32:29
6. click on linkText=Create an Account! OK	13:32:29
7. click on name=name OK	13:32:30
8. type on name=name with value Alya Avara & 08192263628 OK	13:32:30
9. click on name=address OK	13:32:30
10. type on name=address with value J. Sarijadi no 26 OK	13:32:31
11. click on name=email OK	13:32:31
12. type on name=email with value avara@gmail.com OK	13:32:31
13. click on name=password OK	13:32:31
14. type on name=password with value 12345 OK	13:32:31
15. click on css=.btn OK	13:32:31
'Register' completed successfully	13:32:31

Lampiran 1 Menu Register

2. Menu Login

Running 'Login'	13:37:31
1. open on /frontera/ OK	13:37:32
2. setWindowSize on 1552x832 OK	13:37:32
3. click on id=userDropdown OK	13:37:32
4. click on linkText=Logout OK	13:37:33
5. click on name=email OK	13:37:34
6. type on name=email with value avara@gmail.com OK	13:37:35
7. click on name=password OK	13:37:36
8. type on name=password with value 12345 OK	13:37:37
9. click on css=.btn OK	13:37:37
10. click on linkText=Order OK	13:37:37
'Login' completed successfully	13:37:37

Lampiran 2 Menu Login

3. Menu Home

Running 'Home'	13:28:47
1. open on /frontera/ OK	13:28:48
2. setWindowSize on 1552x832 OK	13:28:48
3. click on linkText=Grooming OK	13:28:48
4. click on linkText=Order OK	13:28:49
'Home' completed successfully	13:28:49

Lampiran 3 Menu Home

4. Menu Grooming

Running 'Grooming'	13:40:00
1. open on /frontera/ OK	13:40:01
2. setWindowSize on 1552x832 OK	13:40:01
3. click on linkText=Grooming OK	13:40:01
4. click on linkText=Home OK	13:40:02
5. click on linkText=Grooming OK	13:40:03
'Grooming' completed successfully	13:40:03

Lampiran 4 Menu Grooming

5. Menu Order

Running 'Order'	13:42:00
1. open on /frontera/home OK	13:42:01
2. setWindowSize on 1552x832 OK	13:42:01
3. click on linkText=Grooming OK	13:42:01
4. click on linkText=Order OK	13:42:02
5. click on linkText=Buat Pesanan OK	13:42:03
6. click on id=id_package_grooming OK	13:42:04
7. select on id=id_package_grooming with value label=Grooming Premium OK	13:42:04
8. mouseDownAt on css=.form-group:nth-child(4) with value 296.1999816894531,21 OK	13:42:04
9. mouseMoveAt on css=.form-group:nth-child(4) with value 296.1999816894531,21 OK	13:42:04
10. mouseUpAt on css=.form-group:nth-child(4) with value 296.1999816894531,21 OK	13:42:04
11. click on css=form OK	13:42:04
12. select on id=id_package_grooming_pet with value label=Anjing Rp.70000 OK	13:42:05
13. click on id=id_delivery_cost OK	13:42:05
14. select on id=id_delivery_cost with value label=Sarijadi Rp.5000 OK	13:42:05
15. click on css=.form-group:nth-child(5) OK	13:42:05
16. click on name=notes OK	13:42:05
17. type on name=notes with value kucing oren OK	13:42:05
18. click on css=.btn-primary OK	13:42:05
'Order' completed successfully	13:42:05

Lampiran 5 Menu Order

6. Menu Profil

Running 'Profile'	13:48:34
1. open on /frontera/home OK	13:48:35
2. setWindowSize on 1552x832 OK	13:48:35
3. click on linkText=Order OK	13:48:35
4. click on css=.nav-item:nth-child(4) span OK	13:48:36
5. click on name=name OK	13:48:36
6. click on name=address OK	13:48:36
'Profile' completed successfully	13:48:36

Lampiran 6 Menu Profil

7. Menu Dashboard Admin

Running 'Dashboard'	13:54:25
1. open on /frontera/dashboard OK	13:54:26
2. setWindowSize on 1552x832 OK	13:54:26
3. click on css=.nav-item:nth-child(4) span OK	13:54:26
4. click on css=.nav-item:nth-child(5) span OK	13:54:27
5. click on linkText=Hewan Peliharaan OK	13:54:28
6. click on css=.nav-item:nth-child(6) span OK	13:54:29
7. click on css=.nav-item:nth-child(7) span OK	13:54:30
8. click on css=.nav-item:nth-child(8) span OK	13:54:31
9. click on css=.nav-item:nth-child(3) span OK	13:54:32
'Dashboard' completed successfully	13:54:33

Lampiran 7 Menu Dashboard Admin

8. Menu Pesanan Admin

Running 'Pesanan Admin'	14:03:08
1. open on /frontera/dashboard OK	14:03:09
2. setWindowSize on 1552x832 OK	14:03:09
3. click on css=.nav-item:nth-child(4) span OK	14:03:09
4. click on css=.nav-item:nth-child(3) span OK	14:03:10
5. click on linkText=Buat Pesanan OK	14:03:10
6. click on id=id_package_grooming OK	14:03:10
7. select on id=id_package_grooming with value label=Grooming Premium OK	14:03:11
8. click on id=id_package_grooming_pet OK	14:03:11
9. select on id=id_package_grooming_pet with value label=Anjing Rp.70000 OK	14:03:11
10. click on id=id_delivery_cost OK	14:03:11
11. select on id=id_delivery_cost with value label=Pasteur Rp.10000 OK	14:03:12
12. click on name=notes OK	14:03:12
13. click on linkText=Batalikan OK	14:03:12
'Pesanan Admin' completed successfully	14:03:12

Lampiran 8 Menu Pesanan Admin

9. Menu Paket *Grooming* Admin

Running 'Paket Grooming'	14:06:45
1. open on /frontera/dashboard OK	14:06:46
2. setWindowSize on 1552x832 OK	14:06:46
3. click on css=.nav-item:nth-child(4) span OK	14:06:46
4. click on linkText=Tambah OK	14:06:47
5. click on linkText=Batalikan OK	14:06:48
6. click on linkText=Ubah OK	14:06:48
7. click on name=title OK	14:06:48
8. type on name=title with value Grooming Premiums OK	14:06:48
9. click on css=.btn-primary OK	14:06:49
10. click on css=.odd:nth-child(1) > td:nth-child(4) OK	14:06:49
11. click on linkText=Ubah OK	14:06:49
12. click on name=title OK	14:06:49
13. type on name=title with value Grooming Premium OK	14:06:49
14. click on css=.btn-primary OK	14:06:49
'Paket Grooming' completed successfully	14:06:49

Lampiran 9 Menu Paket Grooming Admin

10. Menu Harga *Grooming* Admin

Running 'Harga Grooming'	14:09:47
1. open on /frontera/dashboard OK	14:09:47
2. setWindowSize on 1552x832 OK	14:09:47
3. click on css=.nav-item:nth-child(5) span OK	14:09:47
4. click on linkText=Tambah OK	14:09:47
5. click on name=id_package_grooming OK	14:09:47
6. select on name=id_package_grooming with value label=Grooming Premium OK	14:09:48
7. click on name=id_pet OK	14:09:48
8. select on name=id_pet with value label=Anjing OK	14:09:48
9. click on linkText=Batalikan OK	14:09:48
10. click on linkText=Ubah OK	14:09:48

11. click on name=id_package_grooming OK	14:09:48
12. click on name=id_package_grooming OK	14:09:48
13. click on name=id_pet OK	14:09:49
14. click on name=price OK	14:09:49
15. click on name=price OK	14:09:49
16. type on name=price with value 71000 OK	14:09:49
17. click on css=.btn-primary OK	14:09:49
18. click on linkText=Ubah OK	14:09:49
19. click on name=price OK	14:09:49
20. type on name=price with value 70000 OK	14:09:50
21. click on css=.btn-primary OK	14:09:50
'Harga Grooming' completed successfully	14:09:50

Lampiran 10 Menu Harga Grooming Admin

11. Menu Hewan Peliharaan Admin

Running 'Hewan Peliharaan'	14:12:38
1. open on /frontera/dashboard OK	14:12:39
2. setWindowSize on 1552x832 OK	14:12:39
3. click on css=.nav-item:nth-child(6) span OK	14:12:39
4. click on linkText=Ubah OK	14:12:40
5. click on name=name OK	14:12:41
6. type on name=name with value Anjings OK	14:12:41
7. click on css=.btn-primary OK	14:12:42
8. click on linkText=Ubah OK	14:12:42
9. click on name=name OK	14:12:42
10. type on name=name with value Anjing OK	14:12:42
11. click on css=.card-body OK	14:12:42
12. click on css=.btn-primary OK	14:12:42
13. click on linkText=Tambah OK	14:12:42
14. click on linkText=Batalikan OK	14:12:42
'Hewan Peliharaan' completed successfully	14:12:43

Lampiran 11 Menu Hewan Peliharaan Admin

12. Menu Biaya Antar Jemput Admin

Running 'Biaya Antar Jemput'	14:15:55
1. open on /frontera/dashboard OK	14:15:56
2. setWindowSize on 1552x832 OK	14:15:56
3. click on css=.nav-item:nth-child(7) span OK	14:15:56
4. click on linkText=Tambah OK	14:15:57
5. click on linkText=Batalan OK	14:15:58
6. click on linkText=Ubah OK	14:15:58
7. click on name=price OK	14:15:58
8. type on name=price with value 5100 OK	14:15:58
9. click on css=.btn-primary OK	14:15:58
10. click on linkText=Ubah OK	14:15:59
11. click on name=price OK	14:15:59
12. type on name=price with value 5000 OK	14:15:59
13. click on css=.btn-primary OK	14:15:59
'Biaya Antar Jemput' completed successfully	14:15:59

Lampiran 12 Menu Biaya Antar Jemput Admin

13. Menu User Admin

Running 'User Admin'	14:17:47
1. open on /frontera/dashboard OK	14:17:48
2. setWindowSize on 1552x832 OK	14:17:48
3. click on linkText=User OK	14:17:48
4. mouseDownAt on css=.table-responsive with value 240.19998168945312,387.59999084472656 OK	14:17:48
5. mouseMoveAt on css=.table-responsive with value 240.19998168945312,387.59999084472656 OK	14:17:48
6. mouseUpAt on css=.table-responsive with value 240.19998168945312,387.59999084472656 OK	14:17:48
7. click on css=#dataTable_filter .form-control OK	14:17:48
8. click on css=#dataTable_filter .form-control OK	14:17:49
9. click on css=#dataTable_filter .form-control OK	14:17:50
'User Admin' completed successfully	14:17:51

Lampiran 13 Menu User Admin

14. Menu Profile admin

Running 'Profil Admin'	14:18:53
1. open on /frontera/dashboard OK	14:18:54
2. setWindowSize on 1552x832 OK	14:18:54
3. click on css=.img-profile OK	14:18:54
4. click on linkText=Profile OK	14:18:54
5. click on name=address OK	14:18:55
'Profil Admin' completed successfully	14:18:55

Lampiran 14 Menu Profile Admin

GLOSARIUM

Glosarium Non Teknis (Daftar istilah-istilah umum)

No	Istilah	Keterangan
1.	Flowchart	Penggambaran secara grafik dari langkah-langkah dan urutan-prosedur dari suatu program.
2.	Use case	Menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas.
3.	Actor	Orang/proses/sistem lain yang berinteraksi dengan sistem informasi.
4.	Sequence Diagram	Menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan di terima antar objek.
5.	Data flow diagram (DFD)	Suatu gambaran logis yang menggambarkan aliran data dari proses satu ke proses lainnya sehingga didapatkan informasi yang diinginkan, sekaligus menggambarkan bagaimana suatu informasi menjadi data dan bagaimana proses selanjutnya.
6.	Class diagram	Struktur diagram statik yang menggambarkan struktur sebuah sistem dengan dengan menunjukkan kelas, atribut dan method / proses.
7.	Collaboration Diagram	Diagram yang menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan.
8.	Component Diagram	Menggambarkan bagian software, controller yang menyatu dsb. Yang membentuk sebuah sistem
9.	Deployment diagram	Menggambarkan <i>hardware</i> untuk sistem dan <i>software</i> yang di instal dalam hardware tersebut.
10.	UML (unified modelling language)	Merancang perangkat lunak, Sarana komunikasi perangkat lunak dengan proses bisnis, Menjabarkan sistem secara rinci untuk analisa dan yg diperlukan system, Mendokumentasi sistem yang ada.

Glosarium Teknik

No	Istilah	Keterangan
1.	<i>BASEPATH</i>	Barisan <i>code</i> yang berfungsi untuk mencegah akses langsung ke <i>script</i> atau controller secara langsung dan memastikan bahwa request dari client telah melalui <i>index.php</i> pada <i>directory project</i> . Cara penulisannya tepat pada dibawah tag pembuka php yaitu <i><?php</i> . Code lengkapnya seperti pada dibawah ini.
2.	<i>Construct</i>	Untuk membuat proses awal dalam mempersiapkan objek, seperti memberi nilai awal kepada property, memanggil method internal dan beberapa proses lainnya yang digunakan untuk mempersiapkan objek.
3.	<i>Foreach</i>	Digunakan untuk perulangan yang datanya dalam bentuk array dan memungkinkan untuk menampilkan data dari array tanpa variabel <i>counter</i> .
4.	<i>Echo</i>	Fungsi untuk menampilkan teks ke layar. Fungsi ini dapat digunakan dengan tanda kurung maupun tanpa tanda kurung.
5.	<i>Required</i>	Untuk meminimalisir kesalahan pengguna saat pengisian form, misalnya harus mengisi semua kolom form yang tertera, namun jika ada yang terlewat maka atribut <i>required</i> ini akan memberi tahu kepada pengguna ada form yang terlewat.
6.	<i>Enctype</i>	Atribut yang berperan menentukan bagaimana data form dikirim ke server.