

**APLIKASI *BOOKING SERVICE* ASTRA HONDA
MOTOR DI KOTA BATAM BERBASIS *CLIENT*
SERVER PADA *PLATFORM* ANDROID
(STUDI KASUS PT CAPELLA DINAMIK
NUSANTARA)**

TUGAS AKHIR

Oleh:

Agus Setia Budi 3311401046

Disusun untuk memenuhi syarat kelulusan Program Diploma 3 (DIII)



**PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2016**

HALAMAN PENGESAHAN

**APLIKASI *BOOKING SERVICE* ASTRA HONDA MOTOR DI KOTA BATAM BERBASIS
CLIENT SERVER PADA *PLATFORM ANDROID*
(STUDI KASUS PT CAPELLA DINAMIK NUSANTARA)**

Disusun oleh :

**Agus Setia Budi
3311401046**

Telah diuji dan dipertahankan di depan Tim Penguji
dalam Sidang Tugas Akhir
pada tanggal 24 Januari 2017
dan dinyatakan **LULUS**.

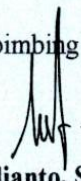
Batam, 04 Mei 2017

Diperiksa dan disetujui oleh;

Pembimbing I,


Nur Zahraati Janah, S.Kom., M.Sc.
NIP. 198610282015042004

Pembimbing II,


Supardianto, S.ST.
NIK. 113105

Penguji I,


Mira Chandra Kirana, S.T., M.T.
NIP. 197905302014042002

Penguji II,


Oktavianto Gustin, S.T., M.T.
NIK. 115138

HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : 3311401046

Nama : Agus Setia Budi

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

*APLIKASI BOOKING SERVICE ASTRA HONDA MOTOR DI KOTA BATAM
BERBASIS CLIENT SERVER PADA PLATFORM ANDROID
(STUDI KASUS PT CAPELLA DINAMIK NUSANTARA)*

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 24 Januari 2017

Agus Setia Budi
3311401046

KATA PENGANTAR

Puji syukur kehadiran Allah S.W.T., atas segala rahmat dan karunia-Nya sehingga laporan tugas akhir yang berjudul “Aplikasi *Booking Service* Astra Honda Motor di Kota Batam Berbasis *Client Server* Pada *Platform* Android (Studi Kasus PT Capella Dinamik Nusantara)” ini dapat terselesaikan.

Penulis mengucapkan terima kasih kepada semua pihak atas bantuan, nasehat, dukungan, serta bimbingan dalam pembuatan tugas akhir ini, sehingga penulis dapat menyelesaikan laporan tugas akhir ini tepat waktu. Dengan kerendahan hati, pada kesempatan ini penulis mengucapkan rasa terima kasih yang sebesar-besarnya kepada :

Secara umum, isi minimal adalah:

1. Orang tua penulis bapak Rusdi dan ibu Setia Ningsih yang selalu mendoakan, serta memberikan dukungan dan kasih sayang sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Ibu Nur Zahrati Janah, S.Kom., M.Sc selaku dosen pembimbing pertama yang telah memberikan kritik, nasehat, dan saran bimbingan yang sangat berguna dalam penyusunan tugas akhir ini.
3. Bapak Supardianto, S.St selaku dosen pembimbing kedua yang telah memberikan kritik, nasehat, dan saran bimbingan yang sangat berguna dalam penyusunan tugas akhir ini.
4. Saudara penulis Joko Purnomo, S.Kep, Dedy Wahyudi, S.Kep serta seluruh keluarga penulis atas doa dan dukungan baik dari segi moral maupun materil.
5. Selly Tri Amanda yang telah membantu dan memberikan dukungan dalam penyusunan tugas akhir ini.
6. Buat sahabat-sahabat penulis atas dukungan doa, kerja sama, kesetiaan, dan kebersamaan dalam penyelesaian tugas akhir ini.

Penulis menyadari bahwa selama melaksanakan tugas besar banyak hal-hal yang dapat dipelajari dan semuanya tidak lepas dari kesalahan maupun kekurangan.

Akhir kata mohon maaf atas kesalahan dan kekurangan serta keterbatasan pada data dan dokumentasi. Kritik saran yang membangun sangatlah diharapkan. Semoga laporan tugas akhir ini bermanfaat dan bisa disempurnakan pada masa yang akan datang

Batam, 24 Januari 2017

Penulis

ABSTRAK

APLIKASI *BOOKING SERVICE* ASTRA HONDA MOTOR DI KOTA BATAM BERBASIS *CLIENT SERVER* PADA *PLATFORM* ANDROID (STUDI KASUS PT CAPELLA DINAMIK NUSANTARA)

Servis sebagai salah satu cara bagi pengguna sepeda motor untuk merawat kendaraannya agar saat mengemudi merasa aman dan nyaman. Permasalahan yang sering dialami adalah ketika melakukan servis ke *dealer*, pelanggan harus menunggu antrian servis yang menghabiskan waktu yang lama dan kemudian digunakan sebuah sistem *booking service* dengan SMS. Namun sistem tersebut memiliki permasalahan karena tidak adanya kompleksitas data dan prosedur yang harus dikelola. Aplikasi berbasis *client server* pada *platform* Android dimungkinkan dapat menjadi salah satu alternatif solusi untuk permasalahan tersebut.

Aplikasi dirancang menggunakan pendekatan *UML* yaitu *use case* diagram, skenario *use case*, *activity* diagram, *sequence* diagram, *class* diagram, dan *ERD* dengan menerapkan MySQL untuk mengelola data serta *tools* yang digunakan untuk membangun yaitu PHP, AngularJS dan CSS. Penelitian ini menggunakan metode studi kasus, dimana data-data dikumpulkan berdasarkan hasil survey lapangan dan pengujian dilakukan dengan metode *black box*.

Hasil penelitian adalah *booking service* Astra Honda Motor berhasil dibangun dengan mengimplementasikan 10 fungsional sehingga dapat membantu kinerja pegawai *dealer* dalam mengelola data *booking*, data servis, serta data pelanggan dan membantu pelanggan dalam melakukan *booking service* serta mendapatkan informasi dari servis yang telah dilakukannya.

Kata Kunci : Aplikasi, *Booking service*, *Client Server*, Android

ABSTRACT

APPLICATION OF BOOKING SERVICE ASTRA HONDA MOTOR IN BATAM CITY CLIENT SERVER BASED ON ANDROID PLATFORM (PT CAPELLA DINAMIK NUSANTARA)

Service as a way for motorcyclists to take care while driving a vehicle in order to feel safe and comfortable. Problems are often experienced when doing a service to the dealer, the customer must wait for the queue servicing spent a long time and then used a system with SMS booking service. However, the system has a problem because of the lack of data and the complexity of the procedures that must be managed. Client server applications based on the Android platform can be one possible alternative solutions to these problems.

Applications designed using UML approach is use case diagrams, use case scenarios, activity diagrams, sequence diagrams, class diagrams, and ERD with implementing MySQL to manage data and tools used to build ie PHP, and CSS AngularJS. This study uses the case study method, where data is collected based on the results of field surveys and tests done by black box method.

The results of the study are booking service Astra Honda Motor successfully constructed by implementing the 10 functional so as to assist dealers in managing employee performance data is booking data services, as well as customer data and assist customers in making the booking service and get information from the service he has done

Keywords : Application, Booking service, Client Server, Android

DAFTAR ISI

ABSTRAK	ii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Tinjauan Pustaka	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Pemesanan	6
2.2 PT Capella Dinamik Nusantara	6
2.3 <i>Unified Modelling Language (UML)</i>	7
2.3.1 <i>Use Case Diagram</i>	7
2.3.2 <i>Activity Diagram</i>	9
2.3.3 <i>Sequence Diagram</i>	9
2.3.4 <i>Class Diagram</i>	10
2.4 HTML	11
2.5 PHP	12
2.6 <i>Java Script Object Notation (JSON)</i>	12
2.7 AngularJS	13
2.8 MySQL	13
2.9 Android	14
BAB III ANALISIS DAN PERANCANGAN	16
3.1 Deskripsi Umum Sistem	16
3.2 Kebutuhan Fungsional	16
3.3 Kebutuhan Non Fungsional	17
3.4 <i>Use Case Diagram</i>	17
3.5 Skenario Use Case	18
3.6 Activity Diagram	23
3.6.1 Activity Diagram Login	23
3.6.2 <i>Activity Diagram</i> Mengubah Kata Sandi	23
3.6.3 Activity Diagram Booking Service	24
3.6.4 <i>Activity Diagram</i> Mengakses Riwayat Servis	25
3.6.5 <i>Activity Diagram</i> Menambah Data Pelanggan	25
3.6.6 <i>Activity Diagram</i> Menghapus Data Pelanggan	26
3.6.7 <i>Activity Diagram</i> Menyunting Data Pelanggan	26
3.6.8 Activity Diagram Memverifikasi Booking	27
3.6.9 <i>Activity Diagram</i> Menambah Servis	27
3.6.10 <i>Activity Diagram</i> Membatalkan Servis	28

3.6.11	Activity Diagram Menambah Data Sparepart	28
3.6.12	Activity Diagram Menghapus Sparepart	29
3.6.13	Activity Diagram Menyunting Sparepart	29
3.6.14	<i>Activity Diagram</i> Menambah Layanan Servis.....	30
3.6.15	<i>Activity Diagram</i> Menghapus Layanan Servis	30
3.6.16	<i>Activity Diagram</i> Menyunting Layanan Servis	31
3.7	Sequence Diagram	31
3.7.1	Sequence Diagram Login	31
3.7.2	<i>Sequence Diagram</i> Mengubah Kata Sandi.....	32
3.7.3	Sequence Diagram Booking Service.....	32
3.7.4	<i>Sequence Diagram</i> Mengakses Riwayat Servis	33
3.7.5	<i>Sequence Diagram</i> Menambah Data Pelanggan	33
3.7.6	<i>Sequence Diagram</i> Menghapus Data Pelanggan.....	34
3.7.7	<i>Sequence Diagram</i> Menyunting Data Pelanggan.....	34
3.7.8	Sequence Diagram Memverifikasi Booking	35
3.7.9	<i>Sequence Diagram</i> Menambah Servis	35
3.7.10	<i>Sequence Diagram</i> Membatalkan Servis	36
3.7.11	Sequence Diagram Menambah Data Sparepart.....	36
3.7.12	Sequence Diagram Menyunting Data Sparepart	37
3.7.13	Sequence Diagram Menghapus Data Sparepart	37
3.7.14	<i>Sequence Diagram</i> Menambah Layanan Servis.....	38
3.7.15	<i>Sequence Diagram</i> Menghapus Layanan Servis	38
3.7.16	<i>Sequence Diagram</i> Menyunting Layanan Servis	39
3.8	Class Diagram.....	39
3.9	Perancangan Basis Data.....	40
3.9.1	Entity Relationship Diagram	40
3.9.2	Physical Data Modelling	40
3.10	Perancangan Antarmuka	41
3.10.1	Perancangan Antarmuka <i>Login</i> pada <i>Server</i>	41
3.10.2	Perancangan Antarmuka Ganti Kata Sandi pada <i>Server</i>	42
3.10.3	Perancangan Antarmuka <i>Booking Service</i> pada <i>Server</i>	43
3.10.4	Perancangan Antarmuka Data Servis pada <i>Server</i>	44
3.10.5	Perancangan Antarmuka Servis pada <i>Server</i>	45
3.10.6	Perancangan Antarmuka Pelanggan pada <i>Server</i>	46
3.10.7	Perancangan Antarmuka Pendaftaran Pelanggan pada <i>Server</i>	47
3.10.8	Perancangan Antarmuka <i>Sparepart</i> pada <i>Server</i>	48
3.10.9	Perancangan Antarmuka Tambah <i>Sparepart</i> pada <i>Server</i>	49
3.10.10	Perancangan Antarmuka Layanan Servis pada <i>Server</i>	50
3.10.11	Perancangan Antarmuka Tambah Layanan Servis pada <i>Server</i> ..	51
3.10.12	Perancangan Antarmuka <i>Login</i> pada <i>Client</i>	52
3.10.13	Perancangan Antarmuka <i>Booking</i> pada <i>Client</i>	53
3.10.14	Perancangan Antarmuka Riwayat Servis pada <i>Client</i>	54
3.10.15	Perancangan Antarmuka About pada <i>Client</i>	55
BAB IV IMPLEMENTASI DAN PENGUJIAN		56
4.1	Implementasi Struktur Tabel Basis Data	56

4.2	Implementasi Proses	57
4.2.1	Implementasi Proses <i>Login</i> pada <i>Server</i>	57
4.2.2	Implementasi Proses Ganti Kata Sandi pada <i>Server</i>	58
4.2.3	Implementasi Proses <i>Booking Service</i> pada <i>Server</i>	59
4.2.4	Implementasi Proses Data <i>Service</i> pada <i>Server</i>	61
4.2.5	Implementasi Proses <i>Service</i> pada <i>Server</i>	63
4.2.6	Implementasi Proses Pelanggan pada <i>Server</i>	65
4.2.7	Implementasi Proses Pendaftaran Pelanggan pada <i>Server</i>	67
4.2.8	Implementasi Proses <i>Sparepart</i> pada <i>Server</i>	69
4.2.9	Implementasi Proses Tambah Sparepart pada <i>Server</i>	71
4.2.10	Implementasi Proses Layanan Servis pada <i>Server</i>	72
4.2.11	Implementasi Proses Tambah Layanan Servis pada <i>Server</i>	74
4.2.12	Implementasi Proses <i>Login</i> pada <i>Client</i>	76
4.2.13	Implementasi Proses <i>Booking</i> pada <i>Client</i>	77
4.2.14	Implementasi Proses Riwayat Servis pada <i>Client</i>	80
4.2.15	Implementasi Proses About pada <i>Client</i>	82
4.3	Pengujian	83
4.3.1	Skenario Pengujian	83
BAB V KESIMPULAN DAN SARAN		89
5.1	Kesimpulan	89
5.2	Saran	89
DAFTAR PUSTAKA		90

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi HTML.....	11
Gambar 2. 2 Ilustrasi PHP.....	12
Gambar 3. 1 Deskripsi Umum Sistem.....	16
Gambar 3. 2 Use Case Diagram.....	17
Gambar 3. 3 <i>Activity Diagram Login</i>	23
Gambar 3. 4 <i>Activity Diagram Mengubah Password</i>	23
Gambar 3. 5 <i>Activity Diagram Booking Service</i>	24
Gambar 3. 6 <i>Activity Diagram Melihat Riwayat Servis</i>	25
Gambar 3. 7 <i>Activity Diagram Menambah Data Pelanggan</i>	25
Gambar 3. 8 <i>Activity Diagram Menghapus Data Pelanggan</i>	26
Gambar 3. 9 <i>Activity Diagram Menyunting Data Pelanggan</i>	26
Gambar 3. 10 <i>Activity Diagram Memverifikasi Booking</i>	27
Gambar 3. 11 <i>Activity Diagram Menambah Servis</i>	27
Gambar 3. 12 <i>Activity Diagram Menghapus Servis</i>	28
Gambar 3. 13 <i>Activity Diagram Menambah Data Sparepart</i>	28
Gambar 3. 14 <i>Activity Diagram Menghapus Sparepart</i>	29
Gambar 3. 15 <i>Activity Diagram Menyunting Sparepart</i>	30
Gambar 3. 16 <i>Activity Diagram Menambah Layanan Servis</i>	30
Gambar 3. 17 <i>Activity Diagram Menghapus Layanan Servis</i>	31
Gambar 3. 18 <i>Activity Diagram Menghapus Layanan Servis</i>	31
Gambar 3. 19 <i>Sequence Diagram Login</i>	31
Gambar 3. 20 <i>Sequence Diagram Mengubah Kata Sandi</i>	32
Gambar 3. 21 <i>Sequence Diagram Booking Service</i>	32
Gambar 3. 22 <i>Sequence Diagram Melihat Riwayat</i>	33
Gambar 3. 23 <i>Sequence Diagram Menambah Data Pelanggan</i>	33
Gambar 3. 24 <i>Sequence Diagram Menghapus Data Pelanggan</i>	34
Gambar 3. 25 <i>Sequence Diagram Menyunting Data Pelanggan</i>	34
Gambar 3. 26 <i>Sequence Diagram Memverifikasi Booking</i>	35
Gambar 3. 27 <i>Sequence Diagram Menambah Servis</i>	35
Gambar 3. 28 <i>Sequence Diagram Menghapus Servis</i>	36
Gambar 3. 29 <i>Sequence Diagram Menambah Data Sparepart</i>	36
Gambar 3. 30 <i>Sequence Diagram Menyunting Data Sparepart</i>	37
Gambar 3. 31 <i>Sequence Diagram Menghapus Data Sparepart</i>	37
Gambar 3. 32 <i>Sequence Diagram Menambah Layanan Servis</i>	38
Gambar 3. 33 <i>Sequence Diagram Menghapus Layanan Servis</i>	38
Gambar 3. 34 <i>Sequence Diagram Menyunting Layanan Servis</i>	39
Gambar 3. 35 <i>Class Diagram</i>	39
Gambar 3. 36 <i>Entity Relationship Diagram</i>	40
Gambar 3. 37 <i>Physical Data Modelling</i>	40
Gambar 3. 38 Perancangan Antarmuka <i>Login pada Server</i>	41
Gambar 3. 39 Perancangan Antarmuka Ganti Kata Sandi pada <i>Server</i>	42
Gambar 3. 40 Perancangan Antarmuka <i>Booking Service</i> pada <i>Server</i>	43
Gambar 3. 41 Perancangan Antarmuka Data Servis pada <i>Server</i>	44

Gambar 3. 42 Perancangan Antarmuka Servis pada <i>Server</i>	45
Gambar 3. 43 Perancangan Antarmuka Pelanggan pada <i>Server</i>	46
Gambar 3. 44 Perancangan Antarmuka Pendaftaran Pelanggan pada <i>Server</i>	47
Gambar 3. 45 Perancangan Antarmuka <i>Sparepart</i> pada <i>Server</i>	48
Gambar 3. 46 Perancangan Antarmuka Tambah <i>Sparepart</i> pada <i>Server</i>	49
Gambar 3. 47 Perancangan Antarmuka Layanan Servis pada <i>Server</i>	50
Gambar 3. 48 Perancangan Antarmuka Tambah Layanan Servis pada <i>Server</i>	51
Gambar 3. 49 Perancangan Antarmuka <i>Login</i> pada <i>Client</i>	52
Gambar 3. 50 Perancangan Antarmuka <i>Booking</i> pada <i>Client</i>	53
Gambar 3. 51 Perancangan Antarmuka Riwayat Servis pada <i>Client</i>	54
Gambar 3. 52 Perancangan Antarmuka <i>About</i> pada <i>Client</i>	55
Gambar 4.1 Implementasi Antarmuka <i>Login</i> pada <i>Server</i>	57
Gambar 4.2 Implementasi Antarmuka Ganti Kata Sandi pada <i>Server</i>	58
Gambar 4.3 Implementasi Antarmuka <i>Booking Service</i> pada <i>Server</i>	60
Gambar 4.4 Implementasi Antarmuka Data <i>Service</i> pada <i>Server</i>	62
Gambar 4.5 Implementasi Antarmuka <i>Service</i> pada <i>Server</i>	63
Gambar 4.6 Implementasi Antarmuka Pelanggan pada <i>Server</i>	65
Gambar 4.7 Implementasi Antarmuka Edit Data Pelanggan	66
Gambar 4.8 Implementasi Antarmuka Pendaftaran Pelanggan pada <i>Server</i>	68
Gambar 4.9 Implementasi Antarmuka <i>Sparepart</i> pada <i>Server</i>	69
Gambar 4.10 Implementasi Antarmuka Edit <i>Sparepart</i>	69
Gambar 4.11 Implementasi Antarmuka Tambah <i>Sparepart</i> pada <i>Server</i>	71
Gambar 4.12 Implementasi Antarmuka Layanan Servis pada <i>Server</i>	72
Gambar 4.13 Implementasi Edit Layanan	73
Gambar 4.14 Implementasi Antarmuka Tambah Layanan Servis pada <i>Server</i>	75
Gambar 4.15 Implementasi Antarmuka Proses <i>Login</i> pada <i>Client</i>	76
Gambar 4.16 Implementasi Antarmuka Proses <i>Booking</i> pada <i>Client</i>	78
Gambar 4.17 Implementasi Antarmuka Riwayat Servis pada <i>Client</i>	80
Gambar 4.18 Implementasi Antarmuka <i>Detail</i> Riwayat	81
Gambar 4.19 Implementasi Antarmuka <i>About</i> pada <i>client</i>	82

DAFTAR TABEL

Tabel 1. 1 Tabel perbandingan aplikasi <i>client server</i>	4
Tabel 2. 1 Notasi-notasi <i>use case diagram</i>	8
Tabel 2. 2 Notasi-notasi <i>activity diagram</i>	9
Tabel 2. 3 Notasi-notasi <i>sequence diagram</i>	10
Tabel 2. 4 Notasi-notasi <i>class diagram</i>	10
Tabel 3. 1 <i>Skenario Use Case Login</i>	18
Tabel 3. 2 <i>Skenario Use Case Mengubah Kata Sandi</i>	18
Tabel 3. 3 <i>Skenario Use Case Booking Service</i>	18
Tabel 3. 4 <i>Skenario Use Case Mengakses Riwayat Servis</i>	19
Tabel 3. 5 <i>Skenario Use Case Menambah Data Pelanggan</i>	19
Tabel 3. 6 <i>Skenario Use Case Menghapus Data Pelanggan</i>	19
Tabel 3. 7 <i>Skenario Use Case Menyunting Data Pelanggan</i>	19
Tabel 3. 8 <i>Skenario Use Case Memverifikasi Booking</i>	20
Tabel 3. 9 <i>Skenario Use Case Menambah Servis</i>	20
Tabel 3. 10 <i>Skenario Use Case Membatalkan Servis</i>	20
Tabel 3. 11 <i>Skenario Use Case Menambah Data Sparepart</i>	21
Tabel 3. 12 <i>Skenario Use Case Menghapus Data Sparepart</i>	21
Tabel 3. 13 <i>Skenario Use Case Menyunting Data Sparepart</i>	21
Tabel 3. 14 <i>Skenario Use Case Menambah Layanan Servis</i>	21
Tabel 3. 15 <i>Skenario Use Case Menyunting Layanan Servis</i>	22
Tabel 3. 16 <i>Skenario Use Case Menghapus Layanan Servis</i>	22
Tabel 3. 17 Deskripsi Perancangan Antarmuka <i>Login pada Server</i>	41
Tabel 3. 18 Deskripsi Perancangan Antarmuka Ganti Kata Sandi pada <i>Server</i>	42
Tabel 3. 19 Deskripsi Perancangan Antarmuka <i>Booking Service</i> pada <i>Server</i>	43
Tabel 3. 20 Perancangan Antarmuka Data Servis pada <i>Server</i>	44
Tabel 3. 21 Deskripsi Perancangan Antarmuka Servis pada <i>Server</i>	45
Tabel 3. 22 Deskripsi Perancangan Antarmuka Pelanggan pada <i>Server</i>	46
Tabel 3. 23 Deskripsi Perancangan Antarmuka Pendaftaran Pelanggan pada <i>Server</i>	47
Tabel 3. 24 Deskripsi Perancangan Antarmuka <i>Sparepart</i> pada <i>Server</i>	48
Tabel 3. 25 Deskripsi Perancangan Antarmuka Tambah <i>Sparepart</i> pada <i>Server</i> .	49
Tabel 3. 26 Deskripsi Perancangan Antarmuka Layanan Servis pada <i>Server</i>	50
Tabel 3. 27 Deskripsi Perancangan Antarmuka Tambah Layanan Servis pada <i>Server</i>	51
Tabel 3. 28 Deskripsi Perancangan Antarmuka <i>Login pada Client</i>	52
Tabel 3. 29 Deskripsi Perancangan Antarmuka <i>Booking</i> pada <i>Client</i>	53
Tabel 3. 30 Deskripsi Perancangan Antarmuka Riwayat Servis pada <i>Client</i>	54
Tabel 3. 31 Perancangan Antarmuka <i>About</i> pada <i>Client</i>	55
Tabel 4. 1 Implementasi Struktur Tabel Pelanggan	56
Tabel 4. 2 Implementasi Struktur Tabel Motor.....	56

Tabel 4. 3 Implementasi Struktur Tabel <i>Booking</i>	56
Tabel 4. 4 Implementasi Struktur Tabel Servis.....	56
Tabel 4. 5 Implementasi Struktur Tabel <i>Sparepart</i>	56
Tabel 4. 6 Implementasi Struktur Tabel Layananservis.....	57
Tabel 4. 7 Penjelasan <i>Source Code Login</i> pada <i>Server</i>	57
Tabel 4. 8 Penjelasan <i>Source Code</i> Ganti Kata Sandi.....	59
Tabel 4. 9 Penjelasan <i>Source Code Booking Service</i>	60
Tabel 4. 10 Penjelasan <i>Source Code Data Service</i>	62
Tabel 4. 11 Penjelasan <i>Source Code service</i>	63
Tabel 4. 12 Penjelasan <i>Source Code</i> Pelanggan	66
Tabel 4. 13 Penjelasan <i>Source Code</i> Pendaftaran Pelanggan	68
Tabel 4. 14 Penjelasan <i>Source Code Sparepart</i>	70
Tabel 4. 15 Penjelasan <i>Source Code</i> Tambah <i>Sparepart</i>	71
Tabel 4. 16 Penjelasan <i>Source Code</i> Proses Layanan Servis	73
Tabel 4. 17 Penjelasan <i>Source Code</i> Tambah Layanan Servis	75
Tabel 4. 18 Penjelasan <i>Source Code</i> Proses <i>Login</i> pada <i>Client</i>	76
Tabel 4. 19 Penjelasan <i>Source Code Booking</i>	78
Tabel 4. 20 Penjelasan <i>Source Code</i> Riwayat Servis.....	81
Tabel 4. 21 Penjelasan <i>Source Code About</i>	82

BAB I

PENDAHULUAN

1.1 Latar Belakang

Menurut data dari BPS (Badan Pusat Statistik) pada tahun 2013 lebih dari 84 juta unit sepeda motor digunakan masyarakat umum di Indonesia, karena pada umumnya masyarakat berasumsi dengan menggunakan sepeda motor, tidak membutuhkan waktu yang lama untuk melewati kemacetan pada kota-kota besar seperti kota Batam. Kota Batam itu sendiri dikenal sebagai kota industri, sehingga investor maupun para pekerja yang berdatangan terus meningkat, menimbulkan kemacetan di jalan raya setiap hari pada jam kerja. Salah satu daerah kota Batam yang terbilang macet yakni di daerah Bengkong. Bengkong merupakan daerah dengan kepadatan penduduk kelima berdasarkan data dari BPS (Badan Pusat Statistik) kota Batam. Mayoritas penduduk di Bengkong ialah pekerja yang menggunakan sepeda motor. sehingga untuk melewati kemacetan pada jam kerja, tentunya dibutuhkan kondisi sepeda motor yang prima agar pada saat melakukan perjalanan, pengemudi merasa aman dan nyaman. Hal yang perlu dilakukan agar sepeda motor selalu dalam kondisi prima yaitu, dengan melakukan servis motor secara berkala di *dealer* atau bengkel resmi.

Salah satu perusahaan di daerah Bengkong yang menangani proses perawatan dan penjualan sepeda motor merek Honda yakni PT Capella Dinamik Nusantara. Perusahaan ini berlokasi di Jl. Bengkong Ratu, komplek Green Town blok N No. 1 – 4, Batam. Sebagai salah satu cabang *dealer* resmi Honda yang berlokasi di Bengkong, PT Capella Dinamik Nusantara terus berusaha memberikan bentuk kepuasan dan layanan yang baik kepada pelanggannya dalam hal perawatan sepeda motor. Perusahaan ini sebelumnya sudah memberi kemudahan untuk pelanggan agar mendapatkan layanan servis sepeda motor tanpa harus menunggu lama dengan menggunakan sistem *SMS* . Namun penggunaan sistem tersebut masih kurang efektif karena kurangnya pemberian informasi kepada pelanggan tentang pemesanan waktu servis yang telah dilakukannya

Berdasarkan permasalahan yang dijelaskan maka akan dibangun aplikasi *booking service* Astra Honda Motor di kota Batam berbasis *client server* pada *platform* Android. Pemilihan Android menjadi pertimbangan karena merupakan media internet yang lebih mudah pengoperasiannya dan sifat dari *mobile* tersebut fleksibel (Ariefianto, 2012). Client server menjadi pertimbangan karena dapat melakukan pertukaran data pada peralatan *mobile* (Ariefianto, 2012). Aplikasi ini nantinya diharapkan mampu memudahkan dan menghemat waktu bagi pengguna motor untuk mendapatkan layanan servis di *dealer* dan memudahkan pegawai *dealer* untuk mengelola data pelanggan serta data servis.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, maka rumusan masalah pada penelitian ini adalah sebagai berikut :

1. Bagaimana aplikasi dapat membantu pelanggan untuk melakukan *booking service* dan menyediakan informasi dari *booking* yang telah dilakukan ?
2. Bagaimana aplikasi dapat membantu dalam pengelolaan data pelanggan dan data servis?

1.3 Batasan Masalah

Adapun batasan aplikasi ini adalah sebagai berikut:

1. Aplikasi ini tidak menangani administrasi pembayaran servis sepeda motor
2. Aplikasi ini hanya menangani administrasi servis dengan pemesanan online

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini ialah menghasilkan produk aplikasi untuk:

1. Menyediakan informasi *booking* yang telah dilakukan oleh pelanggan.
2. Melakukan pengelolaan data servis serta data pelanggan.

1.5 Manfaat Penelitian

Manfaat yang bisa didapatkan dari penelitian ialah :

1. Dapat menghemat waktu bagi pengguna saat menunggu antrian servis
2. Meningkatkan kepuasan pelanggan dalam pelayanan Astra Honda Motor
3. Mempermudah pengumpulan, pengelolaan data *booking service* dan data servis.

1.6 Tinjauan Pustaka

Beberapa acuan penelitian yang berkaitan dengan tugas akhir ini. Acuan pertama adalah Sirait (2016) mengembangkan aplikasi pengolahan data pasang surut air laut. Pada penelitian tersebut dilakukan perancangan, pembuatan, dan pengembangan sebuah program pengolahan data pasang surut air laut berbasis *client server* yang dibangun menggunakan bahasa pemrograman Java, XML, HTML, PHP. Aplikasi ini dapat menampilkan grafik pasang surut air laut serta melakukan perhitungan datum elevasi vertikal air laut. Sistem ini terdiri dari dua aplikasi yang bekerja sama dalam membangun pengolahan data pasang surut. Pertama sisi *server* berfungsi menyimpan data pasang surut, dan disisi *client* berhubungan langsung dengan pengguna aplikasi atau *surveyor* dan aplikasi ini dibangun pada platform Android. Untuk penyimpanan data, aplikasi ini menggunakan MySQL pada sisi *server* dan JSON untuk pertukaran data antara *client* dan *Server*.

Acuan kedua Ariefianto dkk. (2012) pengembangan aplikasi pemesanan makanan dan minuman. Pada penelitian tersebut dilakukan perancangan dan pembuatan sebuah program pemesanan makanan dan minuman berbasis *client server* yang dibangun menggunakan bahasa pemrograman Java, XML, HTML, PHP. Sistem ini terdiri dari dua aplikasi dalam membangun program pemesanan di restoran. Pertama sisi *server* berfungsi untuk menangani koneksi *client* yang berbasis Android, sedangkan disisi *client* berhubungan langsung dengan pelayan yang berfungsi untuk melakukan pemesanan, dan aplikasi ini dibangun pada platform Android. Untuk penyimpanan data, aplikasi ini menggunakan MySQL pada sisi *server* dan JSON untuk pertukaran data antara *client* dan *Server*.

Acuan ketiga Mahendra dkk. (2015) melakukan pengembangan aplikasi pelayanan administrasi perawatan sepeda motor berbasis web. Aplikasi tersebut dibangun dengan menggunakan Framework Codeigniter dan dilengkapi dengan data informasi yang berkenaan dengan administrasi perawatan sepeda motor seperti *history service*, laporan data pelanggan serta laporan data servis. Untuk penyimpanan data, aplikasi ini menggunakan MySQL.

Perbandingan antara aplikasi *client server* dan *booking service* yang menjadi acuan dalam penyusunan tugas akhir ini disajikan pada Tabel 1.1.

Tabel 1.1 Tabel perbandingan aplikasi *client server*

Perbandingan sistem	TIDE CALC (Sirait, 2016)	Aplikasi Pemesanan Makanan dan Minuman (Ariefianto, 2012)	Aplikasi pelayanan administrasi perawatan sepeda motor (Mahendra dkk., 2015)	Aplikasi Booking Service AHM
Implementasi	Web, dan Android	Web, dan Android	Web	Web, dan Android
Bahasa pemrograman	Java, XML, HTML, PHP	Java, XML, HTML, PHP	HTML, PHP	AngularJS, XML, HTML, PHP
Pertukaran data	JSON	JSON	-	JSON
Fungsi	Pengolahan data pasang surut air laut	Pengolahan pesanan makanan dan minuman	Pengolahan Administrasi servis dan penjualan spare part	Pengolahan pemesanan pemesanan servis serta menampilkan informasi tentang servis
Database	MySQL	MySQL	MySQL	MySQL

Tabel 1.1 mendeskripsikan perbandingan aplikasi *client server* dari aplikasi yang sudah ada dengan aplikasi *booking service* yaitu Tide Calc (Sirait, 2016) dan aplikasi pemesanan makanan dan minuman (Ariefianto, 2012). Perbandingan yang memiliki sistem yang sudah ada dengan aplikasi *booking service* yaitu aplikasi pelayanan administrasi perawatan sepeda motor (Mahendra dkk., 2015). Semua aplikasi tersebut sebagai acuan dalam perancangan dan pembangunan aplikasi.

1.7 Sistematika Penulisan

Bab I Pendahuluan

Bab ini berisi tentang alasan penulis memilih judul atau latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, tinjauan pustaka, dan sistematika penulisan.

Bab II Landasan Teori

Bab ini berisi tentang teori-teori dari berbagai referensi yang berkaitan dengan aplikasi *booking service* AHM yang mendasari dan menunjang penelitian.

Bab III Analisis dan Perancangan

Bab ini berisi tentang analisis perancangan aplikasi *booking service AHM*. Mulai dari deskripsi umum sistem, kebutuhan fungsional sistem, kebutuhan non-fungsional sistem, diagram use case, skenario use case, activity diagram, sequence diagram, class diagram, perancangan basis data, dan perancangan antarmuka..

Bab IV Implementasi dan Pengujian

Bab ini berisi tentang implementasi dan pengujian aplikasi yang sebelumnya telah dirancang. Apakah sudah sesuai dengan kebutuhan dan layanan yang diharapkan

Bab V Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan dari rangkaian penelitian yang telah dilaksanakan dan saran penulis untuk penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Pemesanan

Menurut Kotler (1998), pemesanan adalah kegiatan transaksi yang menyatakan keinginan atau rencana untuk memiliki atau membeli barang atau jasa tersebut sebelum melakukan transaksi finansial yakni berupa pembayaran finansial terhadap produk atau jasa yang diinginkan. Tujuan dari pemesanan adalah sebagai berikut :

1. Memonitoring keadaan pasar yang menjadi sasaran tujuan dalam hal penjualan.
2. Sebagai ukuran kelemahan dan kekuatan perusahaan dalam pemasaran.
3. Besar kecilnya pangsa pasar dapat mempengaruhi keberhasilan bersaing dengan perusahaan lainnya.

Dalam penelitian ini, teori pemesanan digunakan untuk mengetahui proses-proses dari pemesanan jasa servis motor yang dilakukan di *dealer* AHM.

2.2 PT Capella Dinamik Nusantara

PT Capella Dinamik Nusantara merupakan sebuah perusahaan yang menangani proses perawatan dan penjualan sepeda motor merek Honda (Anonim, 2016). Perusahaan ini berlokasi di Jl. Bengkong Ratu, komplek Green Town blok N No. 1 – 4, Batam. PT Capella Dinamik Nusantara memberikan pelayanan servis yang terbaik kepada konsumen dengan mekanik-mekanik yang telah terlatih dan menempuh pendidikan formal sebagai mekanik ASTRA Honda, suku cadang asli Honda juga mudah didapatkan di Outlet/Part Shop. Perusahaan ini menitikberatkan kepada Kepuasan Pelanggan dengan design Showroom Standar Modern, kebersihan yang terjaga dan pelayanan yang ramah pada setiap counter penjualan. Dalam penelitian ini, PT Capella Dinamik Nusantara berfungsi sebagai sumber informasi yang data-datanya akan digunakan untuk mengembangkan sebuah sistem *booking service* pada platform Android.

2.3 Unified Modelling Language (UML)

UML merupakan standarisasi bahasa permodelan untuk pembangunan perangkat lunak yang dibangun (Rosa dan Salahuddin, 2011). UML muncul karena adanya kebutuhan permodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa bisual untuk permodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Terdapat 3 kategori yang ada pada UML, yaitu :


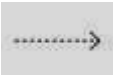








- a. *Structure diagram* adalah kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Contoh, class diagram.
- b. *Behaviour diagram* adalah kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada suatu sistem. Contoh, use case diagram, activity diagram.
- c. *Interaction diagram* adalah kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun antar sistem pada suatu sistem. Contoh, sequence diagram.

Dalam penelitian ini, UML digunakan sebagai bahasa untuk menggambarkan suatu perancangan sistem perangkat lunak dengan memvisualisasikan, menentukan, membangun dan mendokumentasikan dalam bentuk pemodelan.

2.3.1 Use Case Diagram

Menurut Henderi (2008) *use case diagram* merupakan permodelan untuk menggambarkan perilaku sistem yang akan dibuat. *Use case* mendeskripsikan interaksi antara aktor dengan sistem yang dibangun dan mengetahui fungsi apa saja yang berhak digunakan oleh aktor tertentu. Dalam penelitian ini, *Use Case Diagram* digunakan untuk menggambarkan hubungan saling interaksi antara aktor pelanggan dan aktor admin dengan sistem *booking service* berdasarkan kebutuhan fungsional. Berikut adalah notasi-notasi ada di *use case diagram* terdapat pada tabel 2.1.







Tabel 2.1 Notasi-notasi *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri(<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use cases</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.3.2 Activity Diagram

Activity Diagram biasa digunakan untuk memodelkan diagram alir sebuah sistem kerja atau proses bisnis dan operasi-operasi secara *internal* (Randy, 2008). Menurut Henderi(2007), *Activity Diagram* adalah diagram yang menggambarkan sifat dinamis secara alamiah dalam bentuk aliran dan kontrol dari aktivitas ke aktivitas lainnya. Dalam penelitian ini, activity diagram digunakan untuk menggambarkan proses aliran kerja dari sistem *booking service* yang telah dirancang untuk menghasilkan proses bisnis yang sesuai dengan fungsionalitas sistem. Adapun notasi-notasi dalam *Activity Diagram* terdapat pada tabel 2.2.

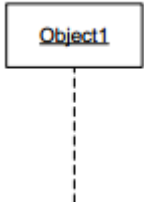


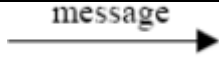
Tabel 2.2 Notasi-notasi Activity Diagram

No	Simbol	Nama	Keterangan
1		<i>Initial state</i>	Titik awal untuk memulai aktivitas
2		<i>Final State</i>	Titik akhir untuk mengakhiri suatu aktivitas
3		<i>Activity</i>	Menandakan suatu aktivitas
4		<i>Decision</i>	Pilihan untuk mengambil keputusan
5		<i>Control flow</i>	Arus aktivitas
6		<i>Note</i>	Catatan khusus untuk sebuah aktivitas

2.3.3 Sequence Diagram

Sequence Diagram merupakan diagram interaksi yang menggambarkan suatu operasi itu dilakukan (Neriswandi, 2014). Diagram ini berjalan berdasarkan waktu. Obyek-obyek yang berkaitan yang melakukan operasi, diurutkan dari kiri ke kanan sesuai waktu terjadinya. Dalam penelitian ini, *sequence diagram* digunakan untuk menggambarkan bagaimana objek yang ada pada sistem booking service dapat berinteraksi dengan objek lainnya. Adapun notasi-notasi dalam *Sequence Diagram* terdapat pada tabel 2.3.

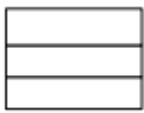
Tabel 2.3 Notasi-notasi *Sequence Diagram*


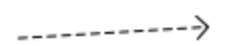

No	Simbol	Nama	Keterangan
1		<i>Object</i>	Instance dari sebuah <i>class</i> yang dituliskan tersusun secara horizontal diikuti lifeline.
2		<i>Activation</i>	Mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.
3		<i>Lifeline</i>	Mengindikasikan keberadaan sebuah obyek dalam basis waktu.
4		<i>Message</i>	Mengindikasikan komunikasi antara obyek-obyek. <i>Message</i> digambarkan anak panah horizontal antara <i>Activation</i> .

2.3.4 *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek dalam bentuk grafik tabel (Sirait, 2016). *Class Diagram* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Pada penelitian ini, *class diagram* digunakan untuk menggambarkan *class* maupun hubungan antar *class* pada perancangan sistem *booking service* tersebut. Adapun notasi-notasi dalam *Class Diagram* terdapat pada tabel 2.4.

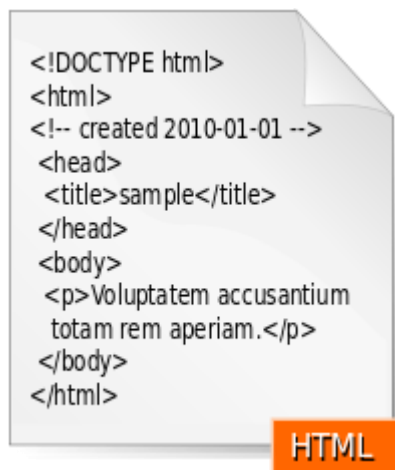
Tabel 2.4 Notasi-notasi *Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Blok-blok pembangun program. Bagian atas class menunjukkan nama dari class, bagian tengah mengindikasikan atribut dari class, dan bagian bawah mendefinisikan method dari sebuah class.

No	Simbol	Nama	Keterangan
2		<i>Association</i>	Menunjukkan relationship atau hubungan antar class.
3		<i>Dependency</i>	Menunjukkan ketergantungan antara satu class dengan class yang lain.
4		<i>Generalization</i>	Menunjukkan inheritance dari satu class ke beberapa class.

2.4 HTML

HTML singkatan dari *HyperText Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur website (Abdulloh, 2016). Untuk lebih memahami dapat dilihat pada Gambar 2.1



Gambar 2. 1 Ilustrasi HTML

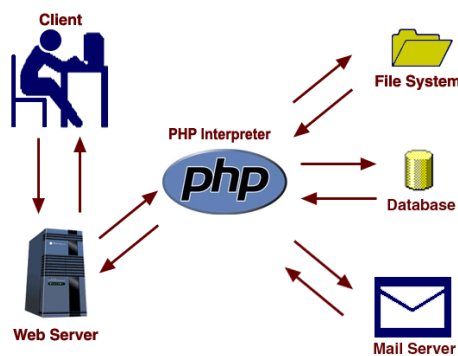
HTML dikenal sebagai web page, karena HTML merupakan dokumen yang disajikan dalam web surfer. Beberapa tugas utama HTML dalam membangun website, diantaranya sebagai berikut:

- a. Menentukan layout website.
- b. Memformat teks dasar.
- c. Membuat list.
- d. Membuat tabel.
- e. Menyisipkan gambar, video, dan audio
- f. Membuat link.
- g. Membuat formulir.

Dalam penelitian ini, HTML digunakan untuk menentukan layout, membuat list, membuat tabel, membuat link, dan membuat formulir yang ada pada sistem *booking service*.

2.5 PHP

PHP singkatan dari *Hypertext Preprocessor* merupakan bahasa pemrograman yang diproses pada bagian server yang paling populer dan banyak digunakan (Abdulloh, 2016). Untuk lebih memahami dapat dilihat pada Gambar 2. 2.



Gambar 2. 2 Ilustrasi PHP

PHP pertama kali dibuat oleh Rasmus Lerdford. Dengan adanya PHP, dunia situs Web menjadi lebih menarik dan interaktif. Di bawah ini beberapa keunggulan yang dimiliki PHP :

- Mendukung penggunaan *cookie*.
- Memiliki fitur dalam menangani *session*.
- Mendukung integrasi dengan database.
- Mendukung regular expression.
- Penanganan kesalahan yang dibuat secara bertingkat.

Dalam penelitian ini, PHP digunakan untuk mendukung penggunaan fitur *session* dalam fungsi *login* dan mendukung integrasi dengan *database* dalam sistem *booking service*.

2.6 *Java Script Object Notation (JSON)*

JSON merupakan suatu format teks untuk serialisasi data, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa

pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data (Safaat, 2011).

Fungsi JSON dalam PHP yaitu:

- a. JSON encode berfungsi untuk merubah array ke bentuk JSON.
- b. JSON decode berfungsi untuk merubah dari bentuk JSON ke Array.

Dalam penelitian ini, JSON berfungsi sebagai pemaketan data yang bertujuan untuk menjembatani antara antar muka dengan service background seperti database server, sama halnya seperti XML.

2.7 AngularJS

AngularJS merupakan open-source JavaScript framework yang dikembangkan oleh Google (Dietz, 2013). AngularJS menjadi populer dan banyak developer yang menggunakan karena performanya yang tinggi. AngularJS digunakan agar meningkatkan fungsi HTML untuk membangun Web Application.

Kelebihan dari AngularJs yaitu;

- a. Data binding yang berfungsi untuk sinkronisasi data antara Model dan View yang dimiliki Angular JS.
- b. Scope adalah suatu objek yang mengacu pada sebuah model, sebagai perekat antara Controller dan View.

Dalam penelitian ini, AngularJS digunakan sebagai *framework* yang menerapkan konsep MVC yang berfungsi untuk membangun sistem *booking service* pada *client*.

2.8 MySQL

Server database pada dasarnya berfungsi untuk mengelola, memproses, dan menampung berbagai macam data . Dalam bahasa SQL data akan diatur dalam baris dan kolom pada tabel, karena pada database data tersebut berupa tabel (Kadir, 2008). MySQL Salah satu database yang paling populer dan banyak digunakan orang yaitu. MySQL Adapun kelebihan yang dimiliki MySQL yaitu :

- a. Program database MySQL memang merupakan yang paling sering diintegrasikan dengan APACHE.
- b. MySQL mempunyai kemampuan server *multithread* sehingga dapat menangani banyak permintaan layanan secara bersamaan.
- c. Dapat berjalan pada banyak sistem operasi (Linux, Windows, , MacOS).
- d. Memiliki enkripsi yang baik.
- e. Mampu menangani data yang besar dan banyak.
- f. Menggunakan autentikasi penggunaan (*user*) dan kata sandi (*password*) sehingga menjadikan lebih aman.

Dalam penelitian ini, MySQL digunakan sebagai *database server* yang mampu menerima dan mengirimkan data yang dibutuhkan dalam sistem *booking service*.

2.9 Android

Android merupakan sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android itu sendiri adalah platform terbuka (Open Source) bagi pengembang untuk membuat aplikasi. Terdapat 2 jenis aplikasi yang terdapat pada Android yaitu;

- a. Aplikasi native adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk platform tertentu.
- b. Aplikasi hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau Android.

Android bukan bahasa pemrograman melainkan hanya menyediakan lingkungan hidup atau run time environment yang disebut DVM (Dalvik Virtual Machine) yang telah dioptimasi untuk *device* dengan sistem memori kecil. Secara garis besar arsitektur android terdiri atas *application* dan *widgets*, *Application Framework*, *Libraries*, *Android Run Time*, dan *Linux Kernel*. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. (Supardi, 2012) Adapun perkembangan 6 OS Android selama 5 tahun belakangan ini yaitu:

1. Android 3.0 (Honey Comp).

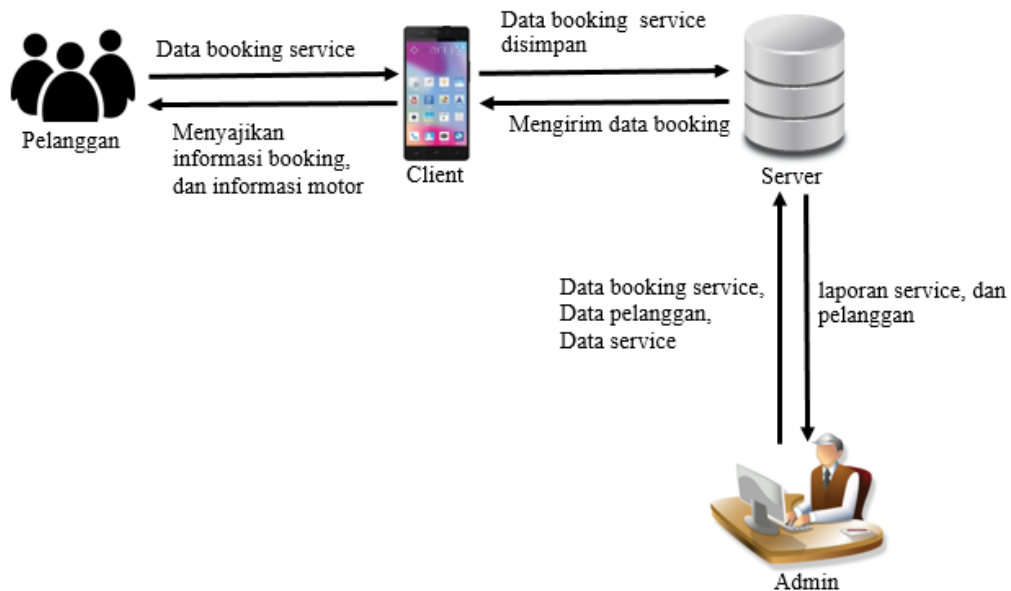
2. Android 4.0 (Ice Cream Sanddwich).
3. Android 4.1-4.3 (Jelly Bean).
4. Android 4.4 (KitKat).
5. Android 5.0 (Lollipop).
6. Anddroid 6.0 (Marshmallow).

Dalam penelitian ini, Android digunakan sebagai objek yang digunakan untuk mengembangkan sistem *booking service* pada sisi *client*.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Deskripsi Umum Sistem



Gambar 3.1 Deskripsi Umum Sistem

Berdasarkan Gambar 3.1, sistem ini terdiri dari dua aplikasi yaitu *client* dan *server*. *Client* adalah aplikasi yang memasukkan atau mengirimkan data *booking service* oleh pelanggan dengan menggunakan *smartphone* (Android). Data yang dimasukkan berupa waktu pemesanan servis, nomor polisi kendaraan motor, jenis servis yang akan dilakukan, jenis sepeda motor, tipe sepeda motor dan keluhan yang dirasakan pengguna sepeda motor. *Server* adalah aplikasi yang mengolah data *booking service*, data *service*, dan data pelanggan dalam web pada komputer, mulai dari menyunting, menghapus data *booking service*, menambah, menyunting, menghapus data *service*, menyunting, menambah dan menghapus data pelanggan dan kemudian memberikan informasi estimasi harga servis kepada *client*.

3.2 Kebutuhan Fungsional

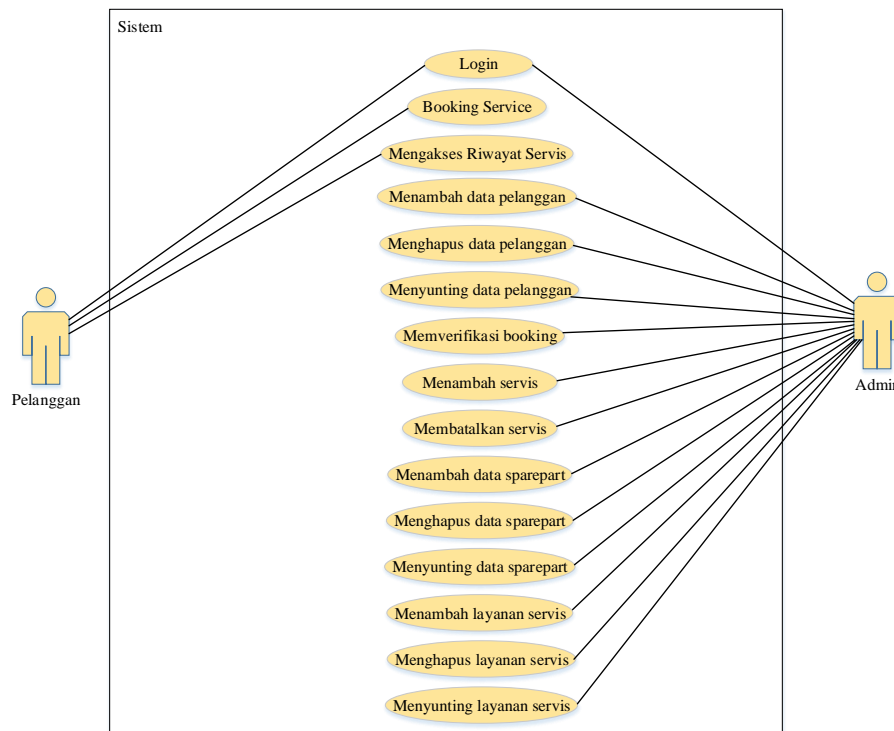
- F001 Sistem memiliki fitur login
- F002 Sistem menampilkan data informasi *dealer*
- F003 Sistem menampilkan riwayat servis

- F004 Admin dapat menambah, menyunting, menghapus data pelanggan
- F005 Admin dapat mengelola layanan servis
- F006 Sistem dapat melakukan *booking service* sepeda motor
- F007 Admin dapat menambah, menyunting, menghapus data *spare part*
- F008 Sistem dapat menampilkan data *booking*
- F009 Sistem dapat menambah *service*
- F010 Sistem menyediakan fitur untuk mengubah status *booking service*

3.3 Kebutuhan Non Fungsional

- NF001 Sistem harus mudah dioperasikan

3.4 Use Case Diagram



Gambar 3.2 Use Case Diagram

Berdasarkan Gambar 3.2, dapat dilihat *use case diagram* yang menggambarkan fungsi atau layanan yang diharapkan ada pada sistem dan hubungannya dengan aktor. Pada aplikasi *booking service* AHM memiliki 2 aktor yaitu pengguna dan *admin*. Pengguna dapat melakukan *booking service* sepeda motor dan melihat riwayat servis. Sedangkan *admin* dapat mengelola data *booking*, transaksi servis serta data pengguna.

3.5 Skenario Use Case

Tabel 3.1 Skenario Use Case Login

Nama use case	<i>Login</i>
Aktor	Pelanggan dan <i>admin</i>
Deskripsi	Pelanggan dan <i>admin</i> melakukan login untuk mendapatkan hak akses
Kondisi Awal	Pelanggan dan <i>admin</i> belum mendapatkan hak akses
Kondisi Akhir	Pelanggan dan <i>admin</i> sudah mendapatkan hak akses
Skenario	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman <i>login</i> 2. Aktor memasukkan nama pengguna dan kata sandi 3. Sistem melakukan verifikasi nama pengguna dan kata sandi 4. Sistem menampilkan halaman utama
Skenario alternatif	3.a [nama pengguna dan kata sandi salah] kembali ke langkah 2

Tabel 3.2 Skenario Use Case Mengubah Kata Sandi

Nama use case	Mengubah kata sandi
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> dapat mengubah kata sandi
Kondisi Awal	<i>Admin</i> telah <i>login</i>
Kondisi Akhir	Kata sandi baru telah tersimpan
Skenario	<ol style="list-style-type: none"> 1. Aktor memasukkan kata sandi baru 2. Sistem melakukan validasi 3. Sistem menyimpan kata sandi baru
Skenario alternatif	2.a [kata sandi tidak valid] kembali ke langkah 1

Tabel 3.3 Skenario Use Case Booking Service

Nama use case	<i>Booking service</i>
Aktor	Pelanggan
Deskripsi	Pelanggan melakukan <i>booking service</i> untuk mendapatkan waktu servis
Kondisi Awal	Pelanggan sudah mendapatkan hak akses
Kondisi Akhir	Pelanggan telah melakukan <i>booking service</i>
Skenario	<ol style="list-style-type: none"> 1. Pelanggan memilih tanggal <i>booking</i> 2. Pelanggan memilih waktu <i>booking</i> 3. Pelanggan memasukkan nomor polisi 4. Pelanggan memilih jenis motor 5. Pelanggan memilih tipe motor 6. Pelanggan memilih layanan 7. Sistem menampilkan estimasi biaya servis 8. Pelanggan memasukkan keluhan 9. Sistem memvalidasi data yang dimasukkan pelanggan 10. Sistem mengirim permintaan <i>booking service</i> ke server 11. Sistem menyimpan data <i>booking</i>
Skenario alternatif	9.a[data tidak valid] kembali ke langkah 1

Tabel 3.4 Skenario Use Case Mengakses Riwayat Servis

Nama use case	Mengakses riwayat servis
Aktor	Pengguna
Deskripsi	Mengakses riwayat servis
Kondisi Awal	Pengguna sudah melakukan servis
Kondisi Akhir	Sistem menampilkan semua data servis yang telah dilakukan
Skenario	1. Sistem mengambil data transaksi servis
	2. Sistem menampilkan riwayat servis
	3. Pelanggan memilih riwayat servis
	4. Sistem menampilkan detail riwayat servis
Skenario alternatif	-

Tabel 3.5 Skenario Use Case Menambah Data Pelanggan

Nama use case	Menambah data pelanggan
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> menambah data pelanggan
Kondisi Awal	<i>Admin</i> telah masuk ke menu data pelanggan
Kondisi Akhir	Data pelanggan telah tersimpan
Skenario	1. <i>Admin</i> memasukkan data pelanggan
	2. Sistem memvalidasi data yang dimasukkan
	3. Sistem menyimpan data pelanggan
Skenario alternatif	2.a[data tidak valid] kembali ke langkah 1

Tabel 3.6 Skenario Use Case Menghapus Data Pelanggan

Nama use case	Menghapus data pelanggan
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> menghapus data pelanggan
Kondisi Awal	Data pelanggan sudah ada
Kondisi Akhir	Sistem menghapus data pelanggan
hSkenario	1. Sistem menampilkan data pelanggan
	2. <i>Admin</i> memilih data pelanggan yang akan dihapus
	3. <i>Admin</i> mengkonfirmasi data pelanggan yang akan dihapus
	4. Sistem menghapus data pelanggan
Skenario alternatif	3.a[tidak dikonfirmasi] kembali ke langkah 2

Tabel 3.7 Skenario Use Case Menyunting Data Pelanggan

Nama use case	Menyunting data pelanggan
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> dapat menyunting data pelanggan
Kondisi Awal	Data pelanggan sudah ada
Kondisi Akhir	Sistem menyimpan dan menampilkan data pelanggan
	1. Sistem menampilkan data pelanggan
	2. <i>Admin</i> memilih data pelanggan yang akan disunting
	3. <i>Admin</i> memasukkan data yang baru

	4. Sistem memvalidasi data yang dimasukkan
	5. Sistem menyimpan data
Skenario alternatif	4.a[data tidak valid] kembali ke langkah 3

Tabel 3.8 Skenario Use Case Memverifikasi Booking

Nama use case	Memverifikasi <i>booking</i>
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> melakukan verifikasi data <i>booking</i>
Kondisi Awal	Data <i>booking</i> sudah ada
Kondisi Akhir	<i>Admin</i> telah memverifikasi data <i>booking</i> dan <i>booking</i> telah disetujui
Skenario	1. Sistem menampilkan semua data <i>booking</i> yang belum diverifikasi
	2. <i>Admin</i> mengecek data <i>booking</i>
	3. <i>Admin</i> mengubah status pengajuan <i>booking</i>
	4. Sistem menyimpan data <i>booking</i>
Skenario alternatif	3.a [Ditolak] Sistem menyimpan data <i>booking</i> dan pengajuan <i>booking</i> ditolak

Tabel 3.9 Skenario Use Case Menambah Servis

Nama use case	Menambah servis
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> dapat menambah servis
Kondisi Awal	Data <i>booking</i> yang statusnya diterima
Kondisi Akhir	Servis telah dilakukan dan tersimpan
Skenario	1. Sistem menampilkan data servis dari <i>booking</i> yang diterima
	2. <i>Admin</i> memilih data servis
	3. <i>Admin</i> memasukkan data <i>sparepart</i> dan layanan yang digunakan
	4. Sistem menyimpan data transaksi servis
Skenario alternatif	-

Tabel 3.10 Skenario Use Case Membatalkan Servis

Nama use case	Membatalkan servis
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> dapat Membatalkan data servis
Kondisi Awal	Data servis sudah ada
Kondisi Akhir	Sistem Membatalkan servis
Skenario	1. Sistem menampilkan data servis
	2. <i>Admin</i> memilih data servis yang akan dibatalkan
	3. <i>Admin</i> mengkonfirmasi data servis yang akan dibatalkan
	4. sistem membatalkan servis
Skenario alternatif	3.a [Tidak dikonfirmasi] kembali ke langkah 2

Tabel 3.11 Skenario Use Case Menambah Data Sparepart

Nama use case	Menambah data <i>sparepart</i>
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> menambah data <i>sparepart</i>
Kondisi Awal	<i>Admin</i> telah masuk ke menu data <i>sparepart</i>
Kondisi Akhir	Data <i>sparepart</i> telah tersimpan
Skenario	1. <i>Admin</i> memasukkan data <i>sparepart</i> 2. Sistem memvalidasi data yang dimasukkan 3. Sistem menyimpan data <i>sparepart</i>
Skenario alternatif	2.a [Data tidak valid] kembali ke langkah 1

Tabel 3.12 Skenario Use Case Menghapus Data Sparepart

Nama use case	Menghapus data <i>sparepart</i>
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> menghapus data <i>sparepart</i>
Kondisi Awal	Data <i>sparepart</i> sudah ada
Kondisi Akhir	Sistem menghapus data <i>sparepart</i> yang ada di <i>database</i>
Skenario	1. Sistem menampilkan data <i>sparepart</i> 2. <i>Admin</i> memilih data <i>sparepart</i> yang akan dihapus 3. <i>Admin</i> mengkonfirmasi data <i>sparepart</i> yang akan dihapus 4. Sistem menghapus data <i>sparepart</i>
Skenario alternatif	3.a [tidak dikonfirmasi] kembali ke langkah 2

Tabel 3.13 Skenario Use Case Menyunting Data Sparepart

Nama use case	Menyunting data <i>sparepart</i>
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> dapat menyunting data <i>sparepart</i>
Kondisi Awal	Data <i>sparepart</i> sudah ada
Kondisi Akhir	Sistem menyimpan data <i>sparepart</i> yang sudah telah disunting
Skenario	1. Sistem menampilkan data <i>sparepart</i> 2. <i>Admin</i> memilih data <i>sparepart</i> yang akan disunting 3. <i>Admin</i> memasukkan data yang baru 4. Sistem memvalidasi data yang dimasukkan 5. Sistem menyimpan data
Skenario alternatif	4.a[data tidak valid] kembali ke langkah 3

Tabel 3.14 Skenario Use Case Menambah Layanan Servis

Nama use case	Menambah layanan servis
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> menambah data layanan servis
Kondisi Awal	<i>Admin</i> telah masuk ke menu data layanan servis
Kondisi Akhir	Data layanan servis tersimpan
Skenario	1. <i>Admin</i> memasukkan data layanan servis 2. Sistem memvalidasi data yang dimasukkan

	3. Sistem menyimpan data layanan servis
Skenario alternatif	2.a [Data tidak valid] kembali ke langkah 1

Tabel 3.15 Skenario Use Case Menyunting Layanan Servis

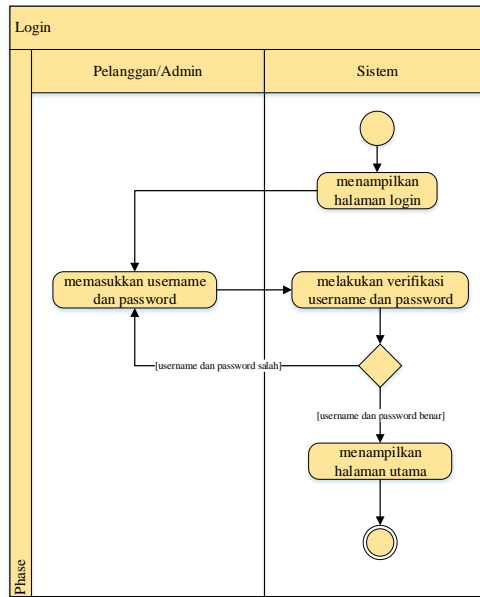
Nama use case	Menyunting layanan servis
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> dapat menyunting layanan servis
Kondisi Awal	Data layanan servis sudah ada
Kondisi Akhir	Sistem menyimpan data layanan servis yang sudah disunting
Skenario	1. Sistem menampilkan data layanan servis
	2. <i>Admin</i> memilih data layanan servis yang akan disunting
	3. <i>Admin</i> memasukkan data yang baru
	4. Sistem memvalidasi data yang dimasukkan
	5. Sistem menyimpan data
Skenario alternatif	4.a [data tidak valid] kembali ke langkah 3

Tabel 3.16 Skenario Use Case Menghapus Layanan Servis

Nama use case	Menghapus layanan servis
Aktor	<i>Admin</i>
Deskripsi	<i>Admin</i> menghapus data layanan servis
Kondisi Awal	Data layanan servis sudah ada
Kondisi Akhir	Sistem menghapus data layanan servis
Skenario	1. Sistem menampilkan data layanan servis
	2. <i>Admin</i> memilih data layanan servis yang akan dihapus
	3. <i>Admin</i> mengkonfirmasi data layanan servis yang akan dihapus
	4. Sistem menghapus data layanan servis
Skenario alternatif	3.a [tidak dikonfirmasi] kembali ke langkah 2

3.6 Activity Diagram

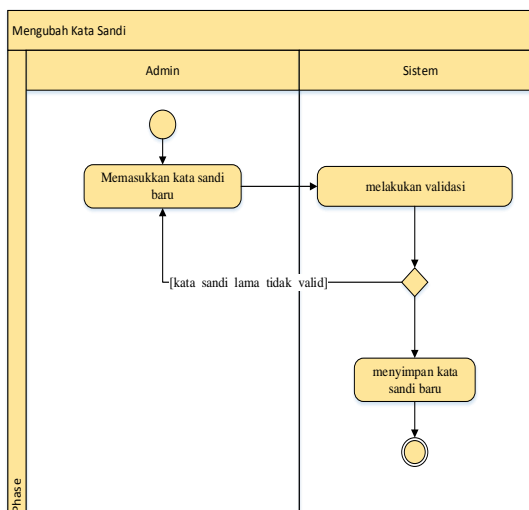
3.6.1 Activity Diagram Login



Gambar 3.3 Activity Diagram Login

Gambar 3.3 menggambarkan aliran kerja pada saat aktor melakukan login pada aplikasi dan terdapat beberapa kemungkinan proses tersebut akan gagal karena disebabkan kondisi tertentu.

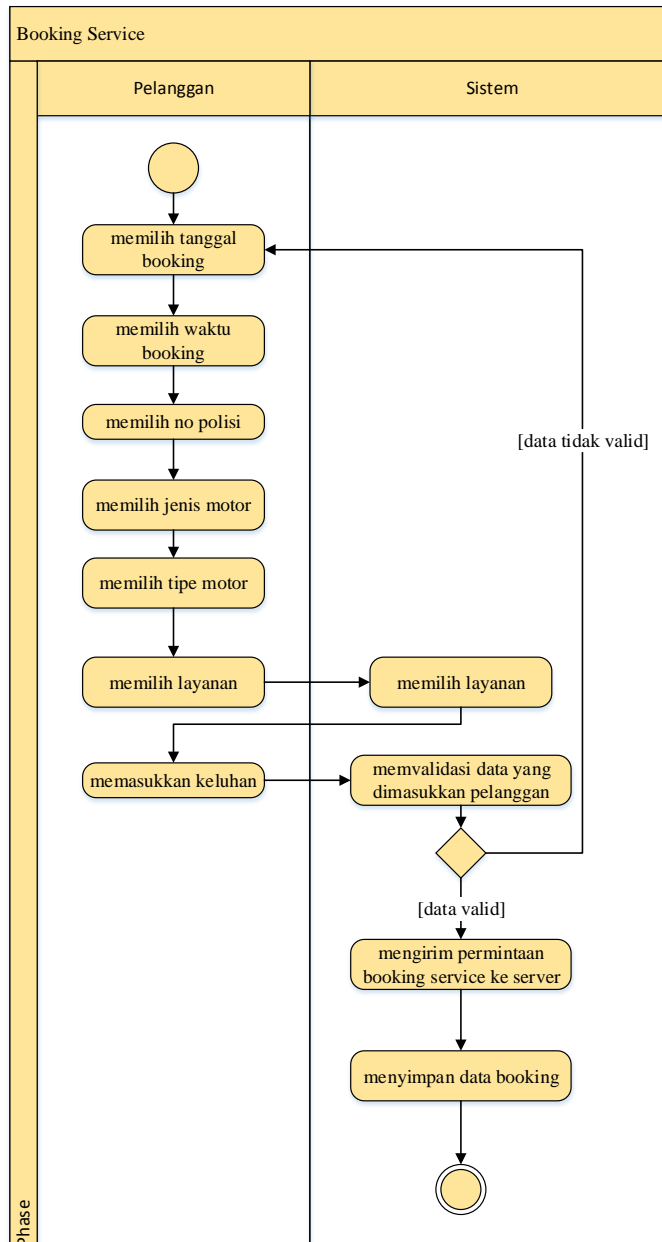
3.6.2 Activity Diagram Mengubah Kata Sandi



Gambar 3.4 Activity Diagram Mengubah Password

Gambar 3.4 menggambarkan aliran kerja pada saat aktor mengubah *password*.

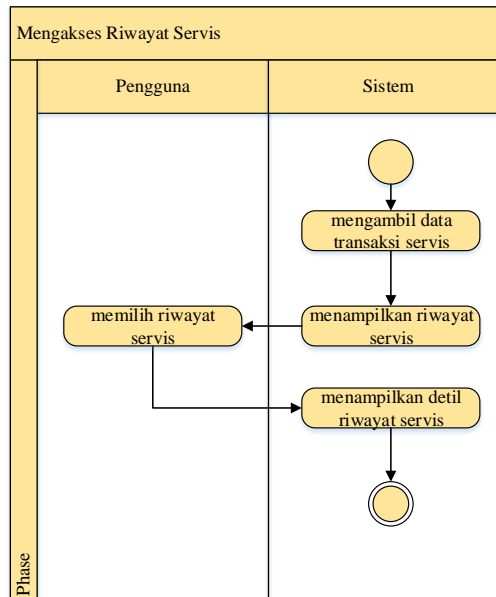
3.6.3 Activity Diagram Booking Service



Gambar 3.5 Activity Diagram Booking Service

Gambar 3.5 menggambarkan aliran kerja *booking service*, pelanggan memasukkan data *booking service* yang diperlukan lalu sistem menyimpan data *booking service*.

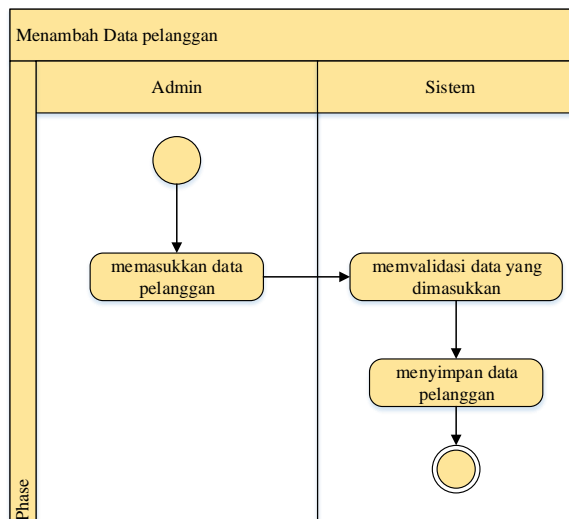
3.6.4 Activity Diagram Mengakses Riwayat Servis



Gambar 3.6 Activity Diagram Melihat Riwayat Servis

Gambar 3.6 menggambarkan aliran kerja pada saat pengguna melihat riwayat servis.

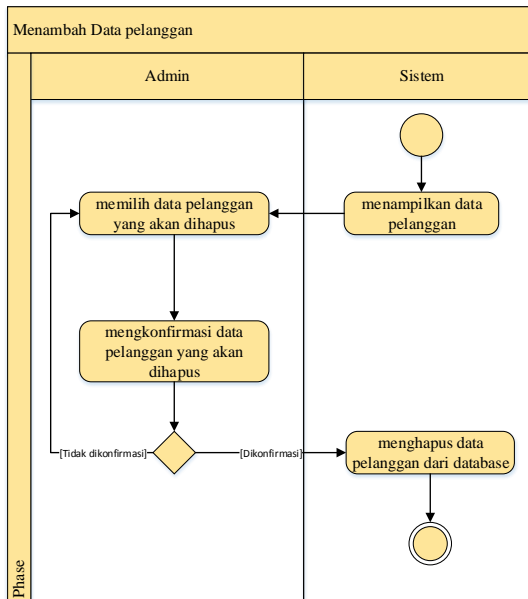
3.6.5 Activity Diagram Menambah Data Pelanggan



Gambar 3.7 Activity Diagram Menambah Data Pelanggan

Gambar 3.7 menggambarkan aliran kerja saat menambah data pelanggan

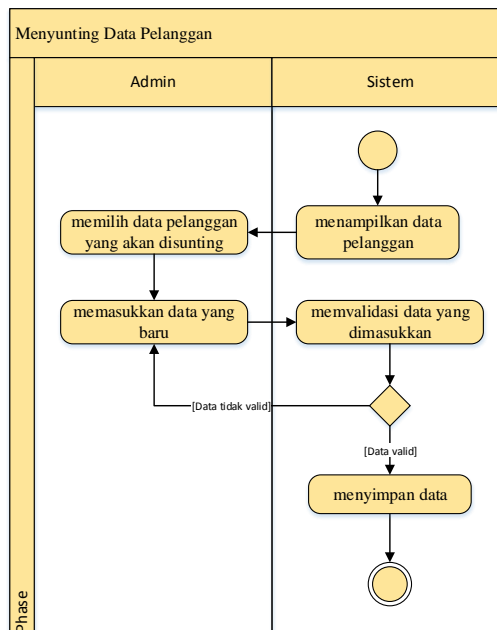
3.6.6 Activity Diagram Menghapus Data Pelanggan



Gambar 3.8 Activity Diagram Menghapus Data Pelanggan

Gambar 3.8 menggambarkan aliran kerja pada saat *admin* menghapus data pelanggan.

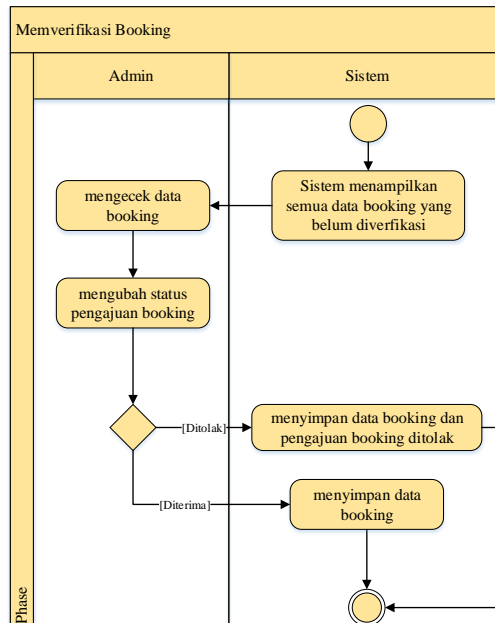
3.6.7 Activity Diagram Menyunting Data Pelanggan



Gambar 3.9 Activity Diagram Menyunting Data Pelanggan

Gambar 3.9 menggambarkan aliran kerja saat menyunting data pelanggan.

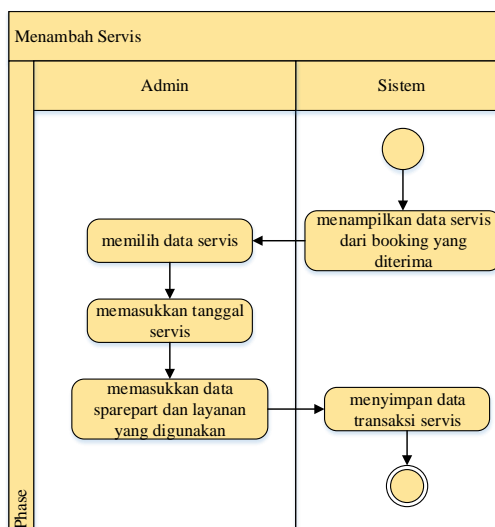
3.6.8 Activity Diagram Memverifikasi Booking



Gambar 3.10 Activity Diagram Memverifikasi Booking

Gambar 3.10 menggambarkan aliran kerja pada saat memverifikasi booking yang dilakukan pelanggan.

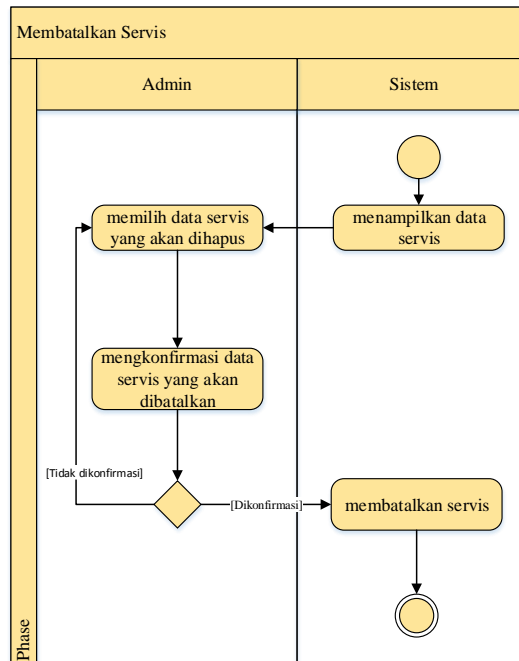
3.6.9 Activity Diagram Menambah Servis



Gambar 3.11 Activity Diagram Menambah Servis

Gambar 3.11 menggambarkan aliran kerja pada saat admin memasukkan data servis pada saat sepeda motor pelanggan tela diservis.

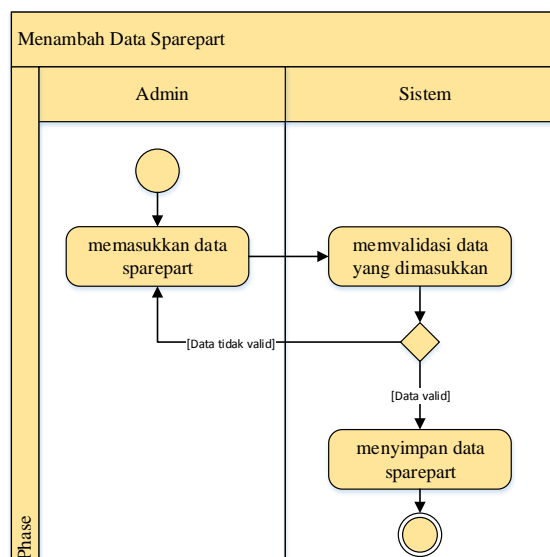
3.6.10 Activity Diagram Membatalkan Servis



Gambar 3.12 Activity Diagram Membatalkan Servis

Gambar 3.12 menggambarkan aliran kerja pada saat *admin* membatalkan data servis.

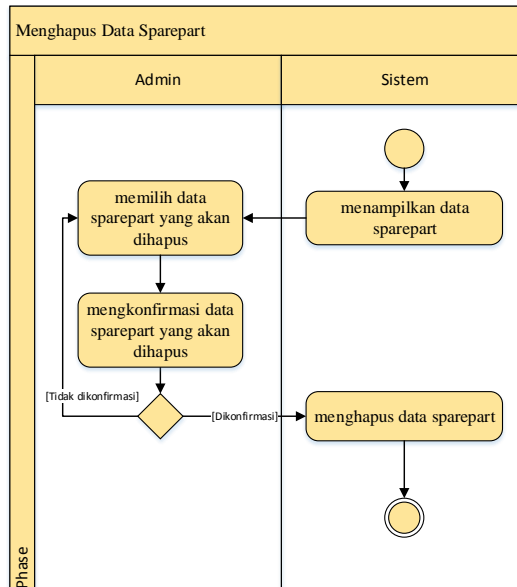
3.6.11 Activity Diagram Menambah Data Sparepart



Gambar 3.13 Activity Diagram Menambah Data Sparepart

Gambar 3.13 menggambarkan aliran kerja pada saat *admin* menambahkan data *sparepart*.

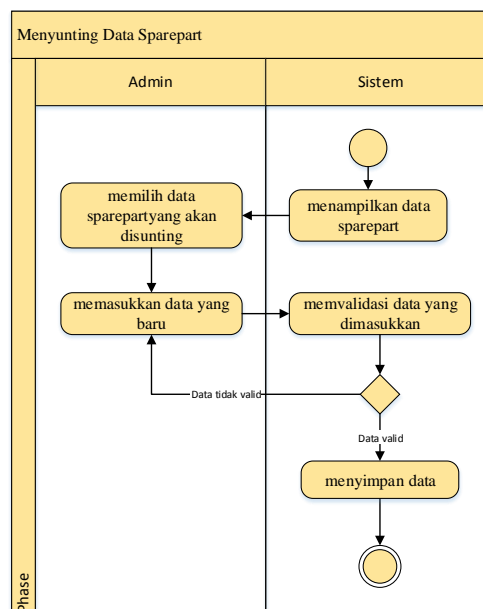
3.6.12 Activity Diagram Menghapus Sparepart



Gambar 3.14 Activity Diagram Menghapus Sparepart

Gambar 3.14 menggambarkan aliran kerja pada saat *admin* menghapus data *sparepart*.

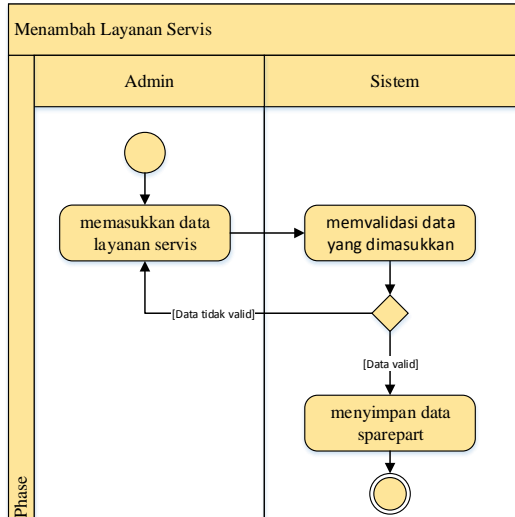
3.6.13 Activity Diagram Menyunting Sparepart



Gambar 3.15 Activity Diagram Menyunting Sparepart

Gambar 3.15 menggambarkan aliran kerja pada saat *admin* menyunting data *sparepart*.

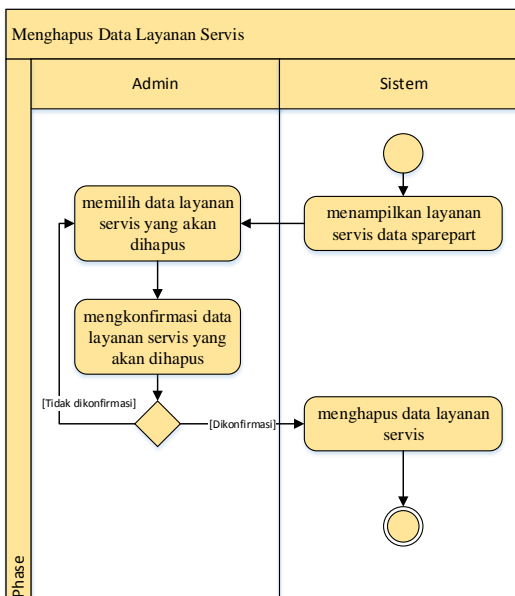
3.6.14 Activity Diagram Menambah Layanan Servis



Gambar 3.16 Activity Diagram Menambah Layanan Servis

Gambar 3.16 menggambarkan aliran kerja pada saat *admin* menambah layanan servis.

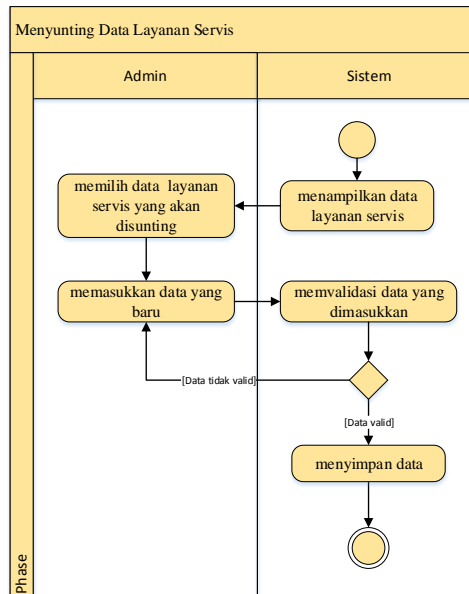
3.6.15 Activity Diagram Menghapus Layanan Servis



Gambar 3.17 Activity Diagram Menghapus Layanan Servis

Gambar 3.17 menggambarkan aliran kerja pada saat *admin* menghapus layanan servis.

3.6.16 Activity Diagram Menyunting Layanan Servis

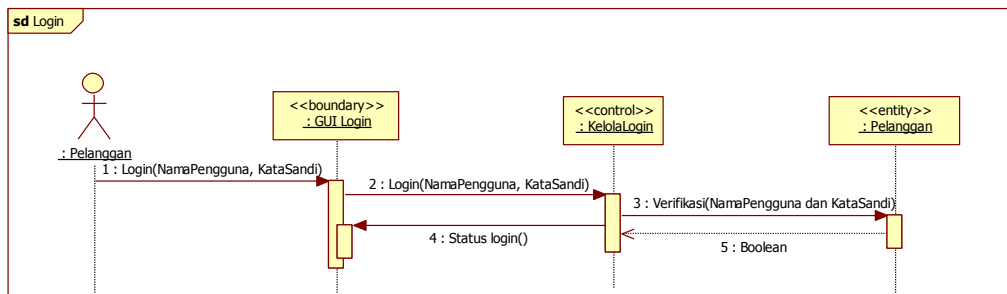


Gambar 3.18 Activity Diagram Menghapus Layanan Servis

Gambar 3.18 menggambarkan aliran kerja pada saat *admin* menyunting layanan servis.

3.7 Sequence Diagram

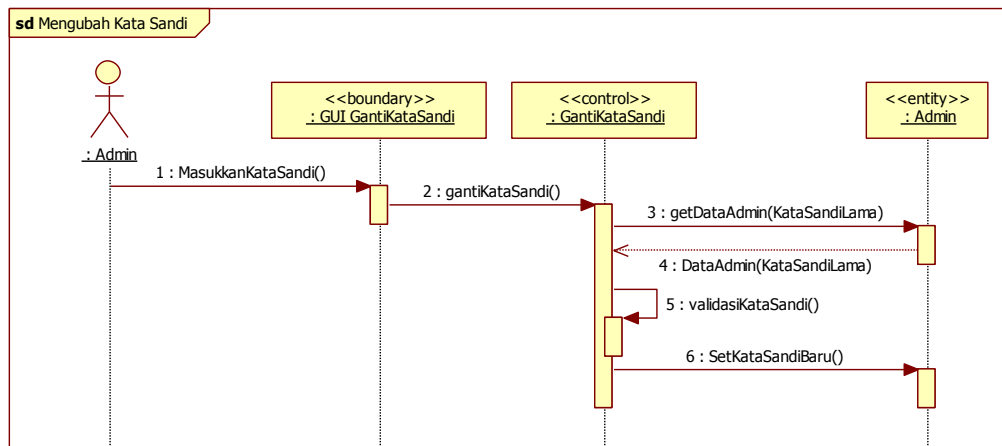
3.7.1 Sequence Diagram Login



Gambar 3.19 Sequence Diagram Login

Gambar 3.19 menunjukkan proses login pelanggan memasukkan nama pengguna dan kata sandi, lalu sistem akan memverifikasi. Setelah nama pengguna dan kata sandi yang dimasukkan benar maka login berhasil.

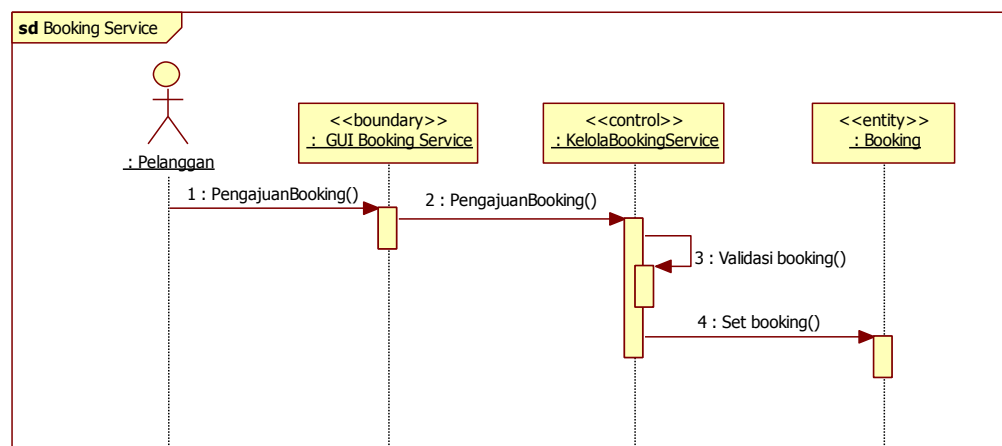
3.7.2 Sequence Diagram Mengubah Kata Sandi



Gambar 3.20 Sequence Diagram Mengubah Kata Sandi

Gambar 3.20 menunjukkan proses mengubah kata sandi, pada proses ini memasukkan kata sandi lama kemudian memasukkan kata sandi yang baru serta konfirmasi kata sandi baru, kemudian kata sandi baru selanjutnya disimpan.

3.7.3 Sequence Diagram Booking Service

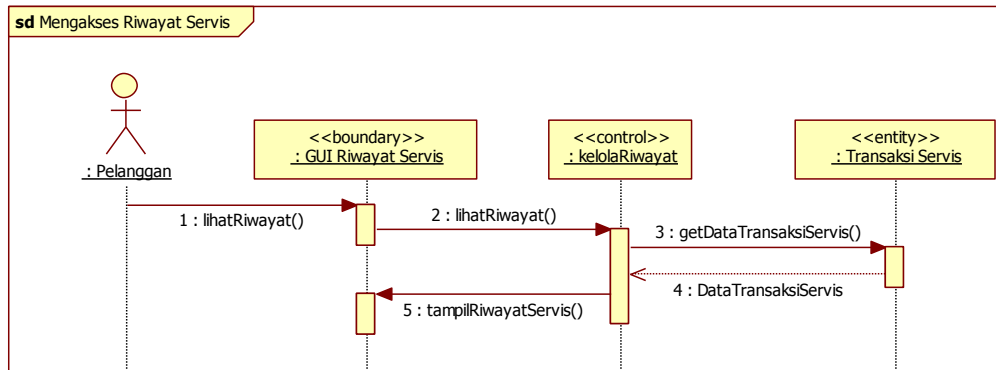


Gambar 3.21 Sequence Diagram Booking Service

Gambar 3.21 menunjukkan proses *booking service*, pada proses ini memasukkan data-data *booking* berupa tanggal *boeing*, waktu, no polisi, jenis

motor, tipe motor, keluhan dan kemudian divalidasi dan disimpan ke dalam *database*.

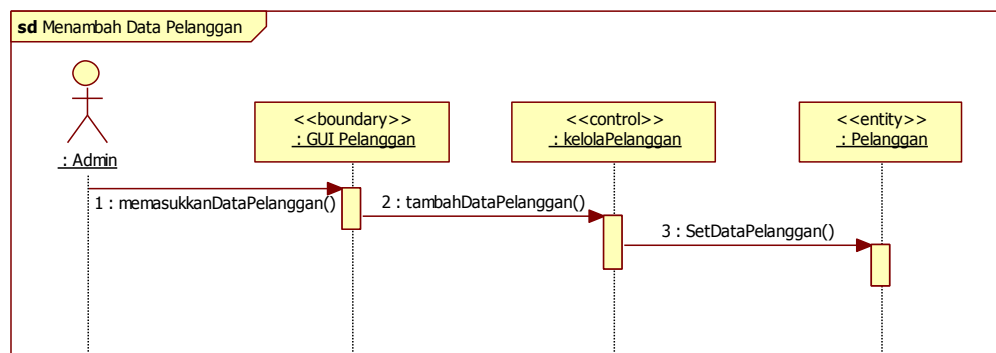
3.7.4 Sequence Diagram Mengakses Riwayat Servis



Gambar 3.22 Sequence Diagram Melihat Riwayat

Gambar 3.22 menunjukkan proses melihat riwayat, pertama kali sistem mengambil data transaksi servis kemudian sistem menampilkannya.

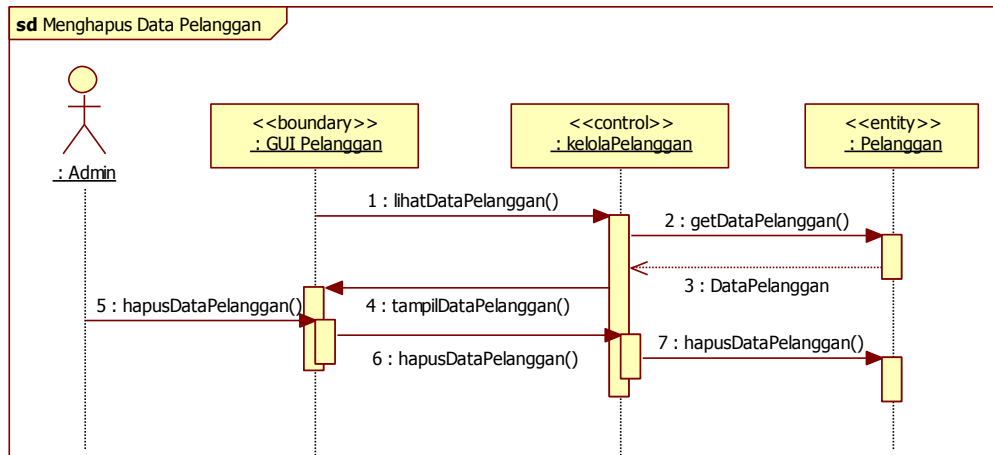
3.7.5 Sequence Diagram Menambah Data Pelanggan



Gambar 3.23 Sequence Diagram Menambah Data Pelanggan

Gambar 3.23 menunjukkan proses menambah data pelanggan, *admin* menambah data pelanggan dan memasukkan data yang diperlukan berupa NIK, nama pengguna dan kata sandi kemudian sistem menyimpan data tersebut.

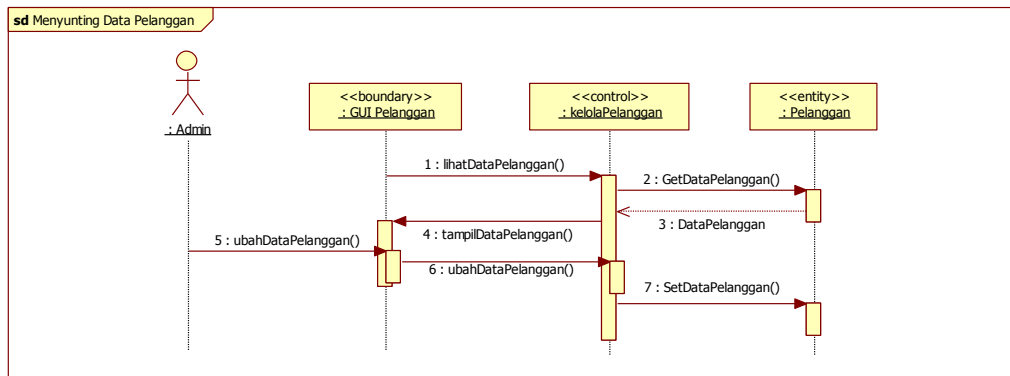
3.7.6 Sequence Diagram Menghapus Data Pelanggan



Gambar 3.24 Sequence Diagram Menghapus Data Pelanggan

Gambar 3.24 menunjukkan proses menghapus data pelanggan, dimana sistem melakukan penghapusan data pelanggan sesuai yang dipilih *admin*.

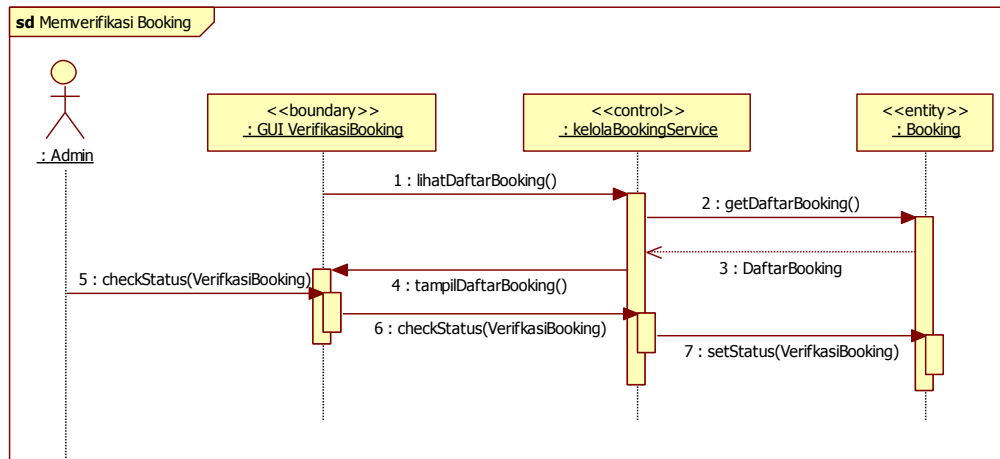
3.7.7 Sequence Diagram Menyunting Data Pelanggan



Gambar 3.25 Sequence Diagram Menyunting Data Pelanggan

Gambar 3.25 menunjukkan proses menyunting data pelanggan, pertama kontrol mengambil data pelanggan kemudian ditampilkan, setelah dipilih oleh *admin* maka memasukkan data baru pelanggan lalu sistem menyimpan ke *database*.

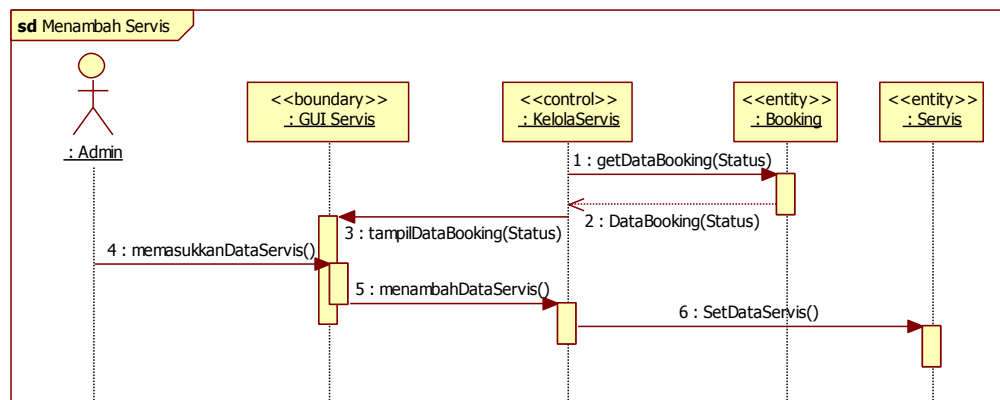
3.7.8 Sequence Diagram Memverifikasi Booking



Gambar 3.26 Sequence Diagram Memverifikasi Booking

Gambar 3.26 menunjukkan proses memverifikasi *booking*, pertama sistem mengambil data yang ada di entitas *booking* kemudian menampilkan dalam bentuk daftar tabel, lalu *admin* memilih data yang untuk diverifikasi agar pengajuan *booking* diterima.

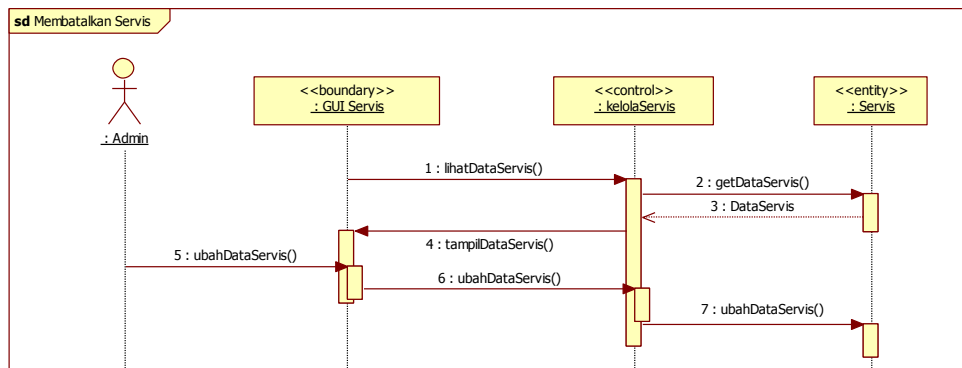
3.7.9 Sequence Diagram Menambah Servis



Gambar 3.27 Sequence Diagram Menambah Servis

Gambar 3.27 menunjukkan proses menambah servis, pertama sistem mengambil data booking dengan status diterima, kemudian *admin* memasukkan data-data servis yang diperlukan yaitu tanggal servis dan data *sparepart*.

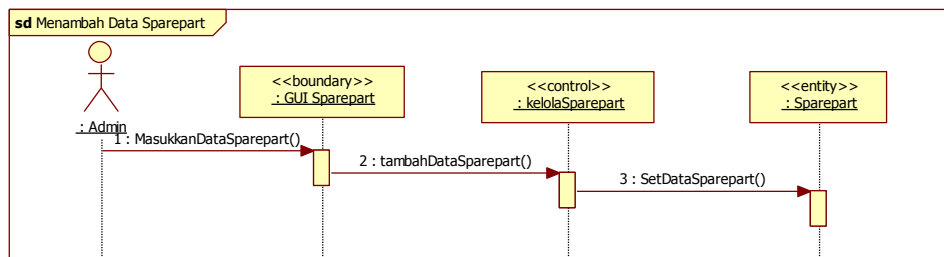
3.7.10 Sequence Diagram Membatalkan Servis



Gambar 3.28 Sequence Diagram Menghapus Servis

Gambar 3.28 menunjukkan proses menghapus servis, pertama sistem mengambil data dan menampilkan data servis kemudian memilih data yang akan dihapus dan sistem menghapus data tersebut.

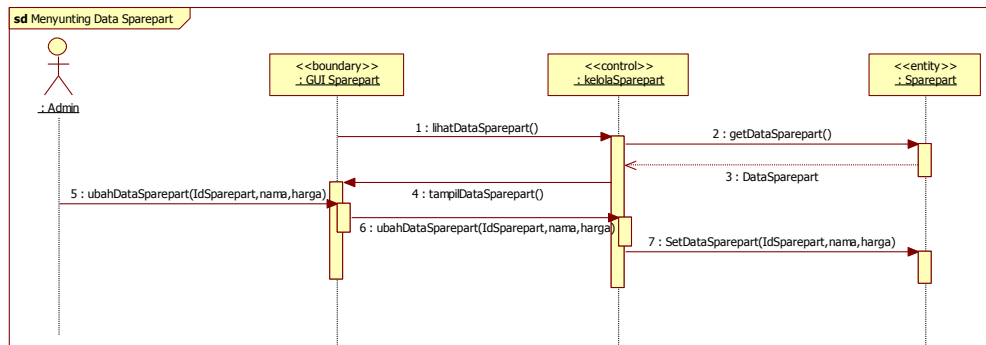
3.7.11 Sequence Diagram Menambah Data Sparepart



Gambar 3.29 Sequence Diagram Menambah Data Sparepart

Gambar 3.29 menunjukkan proses menambah data *sparepart*, pertama *admin* memasukkan data *sparepart* berupa id, nama, dan harga kemudian sistem menyimpan data tersebut.

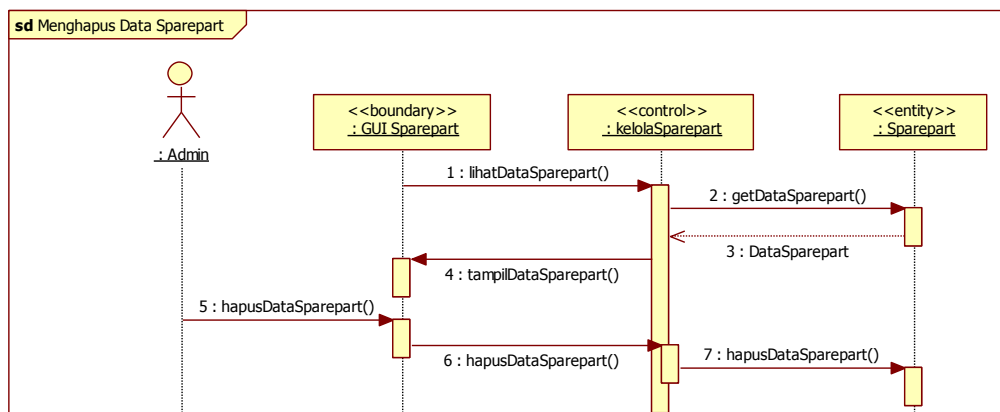
3.7.12 Sequence Diagram Menyunting Data Sparepart



Gambar 3.30 *Sequence Diagram Menyunting Data Sparepart*

Gambar 3.30 menunjukkan menunjukkan proses menyunting data *sparepart*, pertama kontrol mengambil data *sparepart* kemudian ditampilkan, setelah dipilih oleh *admin* maka memasukkan data baru *sparepart* lalu sistem menyimpan ke *database*.

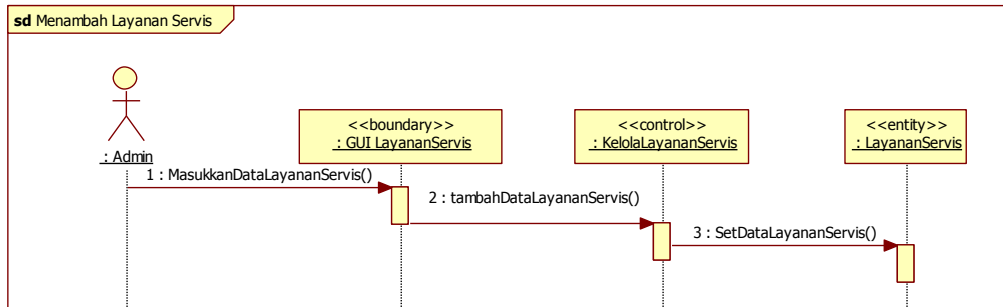
3.7.13 Sequence Diagram Menghapus Data Sparepart



Gambar 3.31 *Sequence Diagram Menghapus Data Sparepart*

Gambar 3.31 menunjukkan proses menghapus data *sparepart*, dimana sistem melakukan penghapusan data *sparepart* sesuai yang dipilih *admin*.

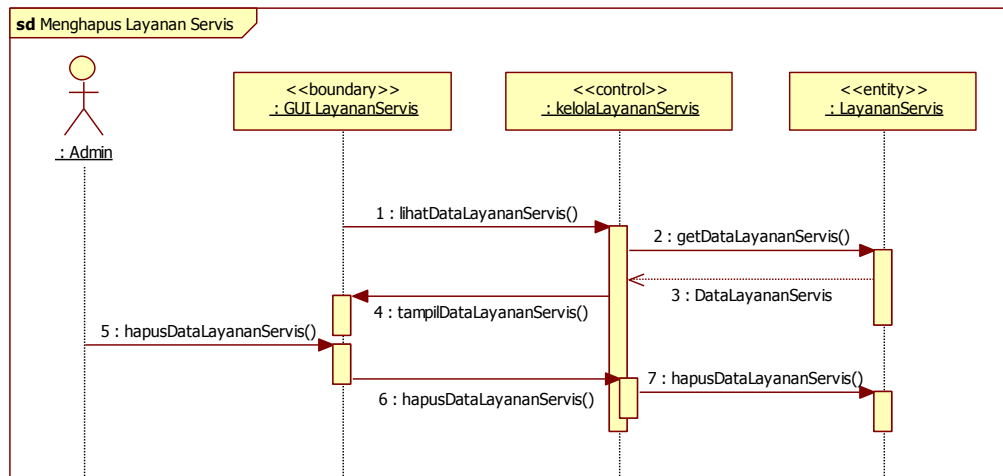
3.7.14 Sequence Diagram Menambah Layanan Servis



Gambar 3.32 Sequence Diagram Menambah Layanan Servis

Gambar 3.32 menunjukkan proses menambah layanan servis, pertama *admin* memasukkan data layanan servis berupa id, jenis, nama layanan, dan harga kemudian sistem menyimpan data tersebut.

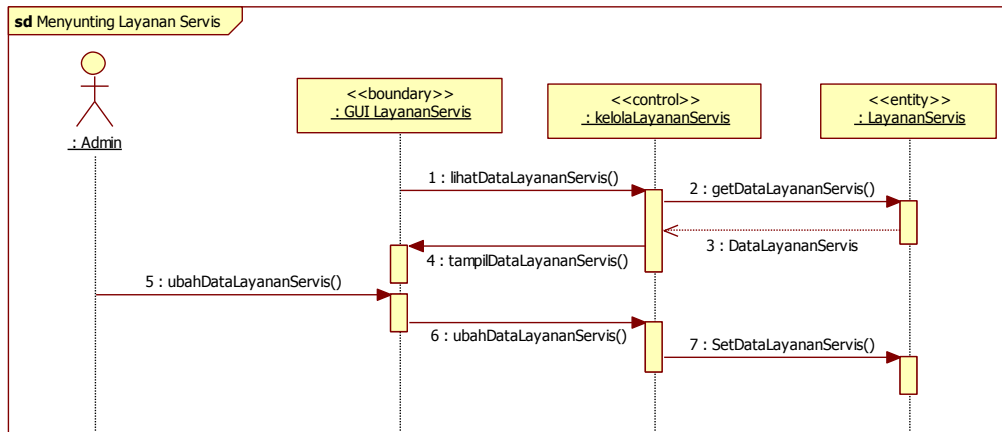
3.7.15 Sequence Diagram Menghapus Layanan Servis



Gambar 3.33 Sequence Diagram Menghapus Layanan Servis

Gambar 3.33 menunjukkan proses menghapus data layanan servis, dimana sistem melakukan penghapusan data layanan servis sesuai yang dipilih *admin*.

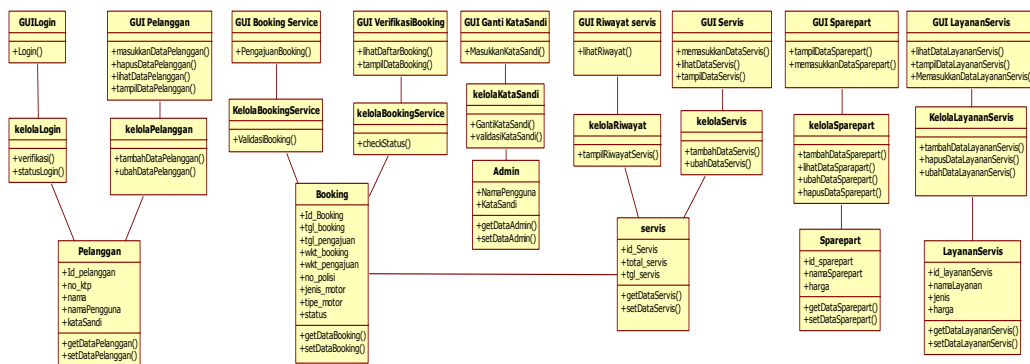
3.7.16 Sequence Diagram Menyunting Layanan Servis



Gambar 3.34 Sequence Diagram Menyunting Layanan Servis

Gambar 3.34 menunjukkan menunjukkan proses menyunting data layanan servis, pertama kontrol mengambil data layanan servis kemudian ditampilkan, setelah dipilih oleh *admin* maka memasukkan data baru layanan servis lalu sistem menyimpan ke *database*.

3.8 Class Diagram

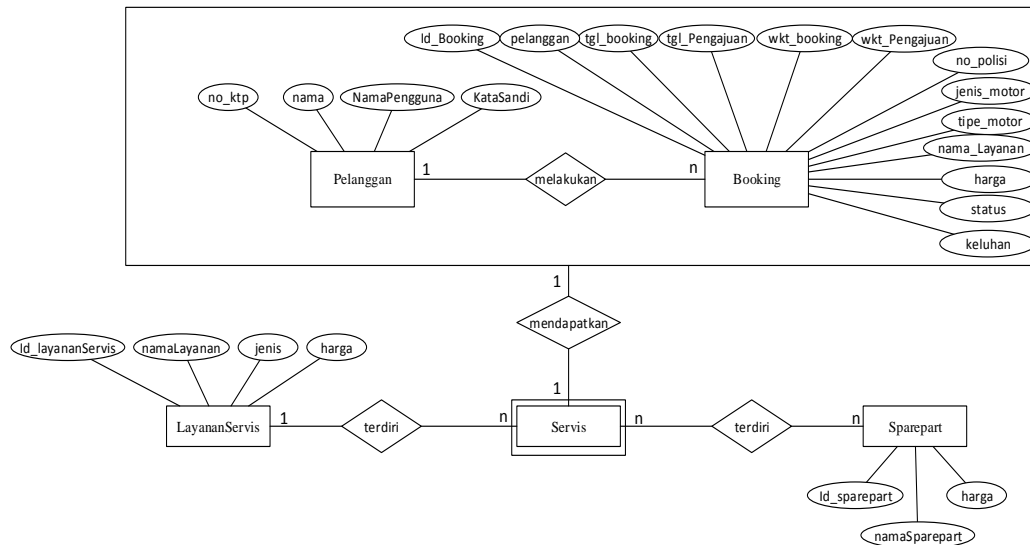


Gambar 3.35 Class Diagram

Gambar 3.35 merupakan class diagram yang memodalkan kelas-kelas dalam aplikasi serta atribut-atribut dan method yang dibutuhkan dalam aplikasi *booking servic* AHM.

3.9 Perancangan Basis Data

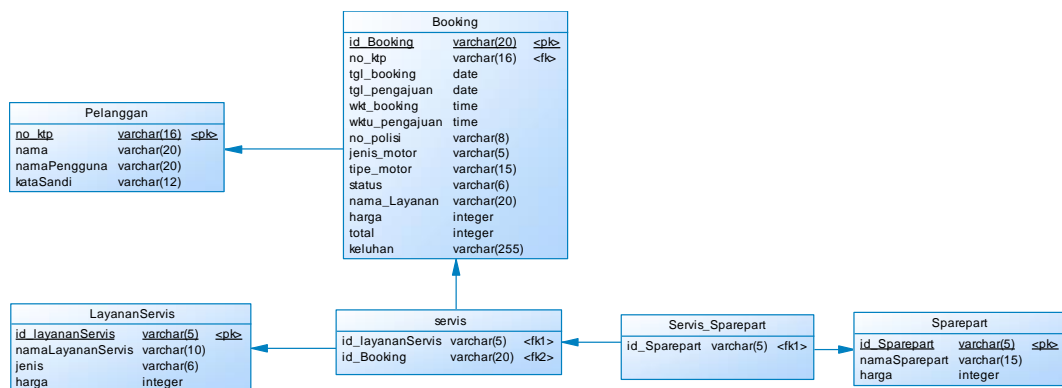
3.9.1 Entity Relationship Diagram



Gambar 3.36 Entity Relationship Diagram

Gambar 3.36 menunjukkan entitas-entitas beserta atribut serta relasi antar entitas yang ada pada aplikasi *booking service* AHM.

3.9.2 Physical Data Modelling

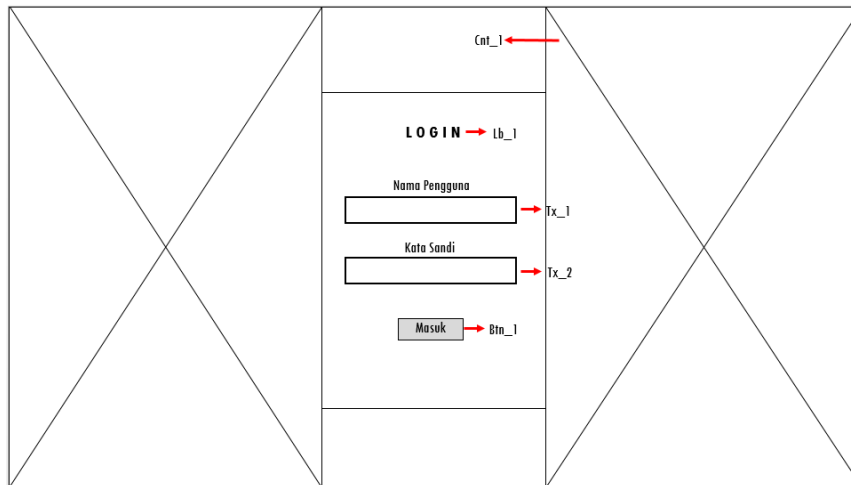


Gambar 3.37 Physical Data Modelling

Gambar 3.37 menunjukkan sejumlah tabel untuk menggambarkan data serta relasia antar data-data pada aplikasi *booking servic* AHM. Terdapat informasi nama atribut dan tipe datanya pada setiap tabel.

6.10 Perancangan Antarmuka

6.10.1 Perancangan Antarmuka *Login* pada *Server*



Gambar 3.38 Perancangan Antarmuka *Login* pada *Server*

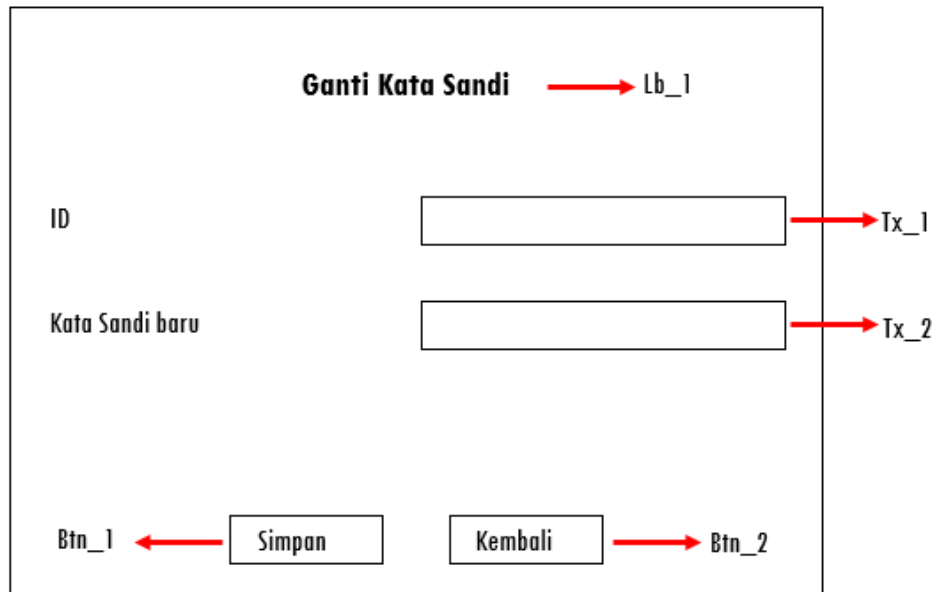
Gambar 3.38 merupakan perancangan antarmuka *login* pada *server* dimana pada tampilan ini pengguna memasukkan nama pengguna dan kata sandi untuk memulai menggunakan aplikasi.

Tabel 3. 17 Deskripsi Perancangan Antarmuka *Login* pada *Server*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	Login	Tulisan yang menandakan halaman <i>login</i>
Tx_1	Text field	Nama pengguna	Diisi dengan nama pengguna
Tx_2	Text field	Kata sandi	Diisi dengan kata sandi pengguna
Cnt_1	Content	Content	Berisi gambar <i>background</i>
Btn_1	Button	Masuk	Jika diklik akan memvalidasi nama pengguna dan kata sandi, jika benar akan masuk ke menu <i>booking</i>

Tabel 3.17 mendeskripsikan perancangan antarmuka *login server* dengan detail mulai dari nama objek (*id_objek*), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.2 Perancangan Antarmuka Ganti Kata Sandi pada *Server*



Gambar 3.39 Perancangan Antarmuka Ganti Kata Sandi pada *Server*

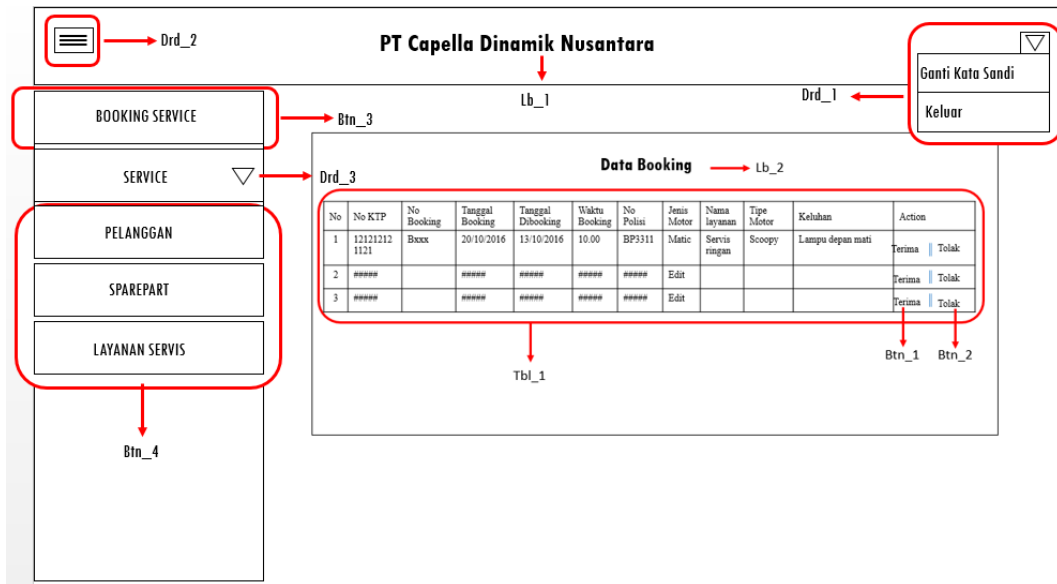
Gambar 3.39 merupakan perancangan antarmuka ganti kata sandi pada *server* dimana pada tampilan ini pengguna memasukkan kata sandi lama dan kata sandi baru untuk mengubah kata sandi.

Tabel 3.18 Deskripsi Perancangan Antarmuka Ganti Kata Sandi pada *Server*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	Ganti kata sandi	Tulisan yang menandakan halaman ganti kata sandi
Tx_1	Text field	Id	Berisi id admin
Tx_2	Text field	Kata sandi baru	Diisi dengan kata sandi baru
Btn_1	Button	Simpan	Jika diklik maka akan menyimpan kata sandi
Btn_2	Button	Kembali	Jika diklik akan keluar dari <i>form</i> ganti kata sandi

Tabel 3.18 mendeskripsikan perancangan antarmuka ganti kata sandi pada *server* dengan detail mulai dari nama objek (id_objek), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.3 Perancangan Antarmuka *Booking Service* pada *Server*



Gambar 3.40 Perancangan Antarmuka *Booking Service* pada *Server*

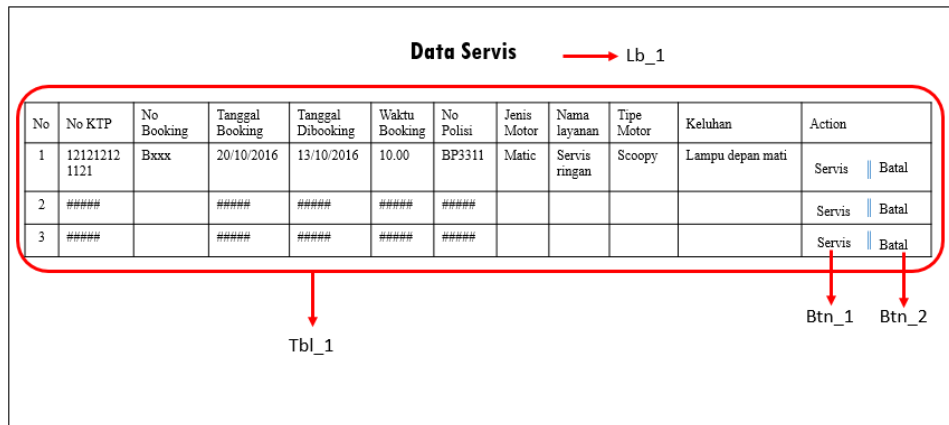
Gambar 3.40 merupakan perancangan antarmuka *booking service* pada server menampilkan semua data *booking* yang dibuat pada hari tersebut. *Admin* dapat menerima ataupun menolak pengajuan *booking*.

Tabel 3.19 Deskripsi Perancangan Antarmuka *Booking Service* pada *Server*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	PT Capella dinamik Nusantara	Logo aplikasi
Lb_2	Label	Data booking	Tulisan yang menandakan halaman <i>booking</i>
Drd_1	Dropdown	Menu pengguna	<i>Dropdown</i> yang berisi <i>button</i> untuk mengganti kata sandi dan keluar dari hak akses
Drd_2	Dropdown	Menu	Berisi menu-menu yang ada pada aplikasi
Drd_3	Dropdown	Menu	Berisi sub menu
Btn_1	Button	Terima	Jika diklik maka akan menerima pengajuan <i>booking</i>
Btn_2	Button	Tidak	Jika diklik maka akan menolak <i>booking</i>
Btn_3	Button	Menu	Jika diklik maka akan ke menu yang kita inginkan, contoh <i>booking service</i> dan servis
Btn_4	Button	Menu	Jika diklik maka akan ke menu pelanggan, <i>sparepart</i> , layanan servis
Tbl_1	Tabel	Tabel	Berisi data <i>booking</i> yang ada pada <i>database</i>

Tabel 3.19 mendeskripsikan perancangan antarmuka *booking service* pada server dengan detail mulai dari nama objek (id_objek), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.4 Perancangan Antarmuka Data Servis pada Server



Gambar 3.41 Perancangan Antarmuka Data Servis pada Server

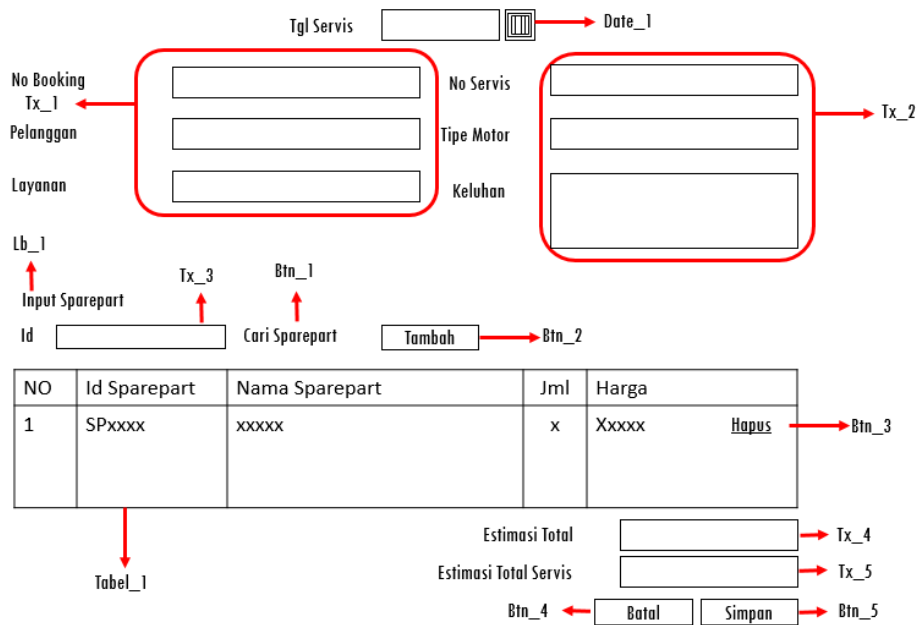
Gambar 3.41 merupakan perancangan antarmuka data servis pada server menampilkan semua data servis yang dibuat pada hari tersebut.

Tabel 3.20 Perancangan Antarmuka Data Servis pada Server

Id_Objek	Jenis	Nama	Keterangan
Btn_1	Button	Servis	Jika diklik maka akan ke <i>form</i> servis
Btn_2	Button	Batal	Jika diklik maka akan membatalkan atau menghapus
Tabel_1	Tabel	Tabel	Berisi data servis

Tabel 3.20 mendeskripsikan perancangan antarmuka data servis pada server dengan detail mulai dari nama objek (id_objek), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.5 Perancangan Antarmuka Servis pada Server



Gambar 3.42 Perancangan Antarmuka Servis pada Server

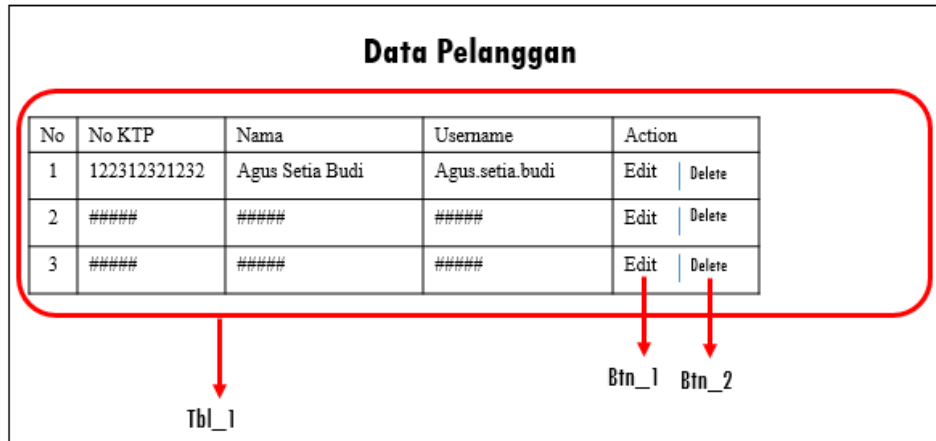
Gambar 3.42 merupakan perancangan antarmuka servis pada server

Tabel 3. 21 Deskripsi Perancangan Antarmuka Servis pada Server

Id_Objek	Jenis	Nama	Keterangan
Date_1	Date	Tgl servis	Diisi sesuai tanggal servis yang dilakukan
Tx_1	Text field	No booking, pelanggan, layanan	Terisi dari data <i>booking</i>
Tx_2	Text field	No servis, tipe motor, keluhan	Terisi dari data <i>booking</i>
Tx_3	Text field	Id sparepart	Diisi dengan id <i>sparepart</i>
Tx_4	Text field	Total	Diisi dengan total harga <i>sparepart</i> yang digunakan
Tx_5	Text field	Total servis	Diisi dengan total dari layanan dan total harga <i>sparepart</i>
Lb_1	Label	Input Sparepart	Tulisan yang menerangkan tempat menambahkan <i>sparepart</i> pada servis
Btn_1	Button	Cari sparepart	Jika diklik maka akan mencari <i>sparepart</i>
Btn_2	Button	Tambah	Jika diklik maka akan menambahkan <i>sparepart</i> pada tabel servis
Btn_3	Button	Hapus	Jika diklik maka akan menghapus <i>sparepart</i> dari tabel servis
Btn_4	Button	Batal	Jika diklik maka akan membatalkan proses servis
Btn_5	Button	Simpan	Jika diklik maka akan menyimpan data servis
Tabel_1	Tabel	Tabel	Berisi data-data <i>sparepart</i> yang digunakan dalam servis

Tabel 3.21 mendeskripsikan perancangan antarmuka servis pada *server* dengan detail mulai dari nama objek (*id_objek*), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.6 Perancangan Antarmuka Pelanggan pada *Server*



Gambar 3.43 Perancangan Antarmuka Pelanggan pada *Server*

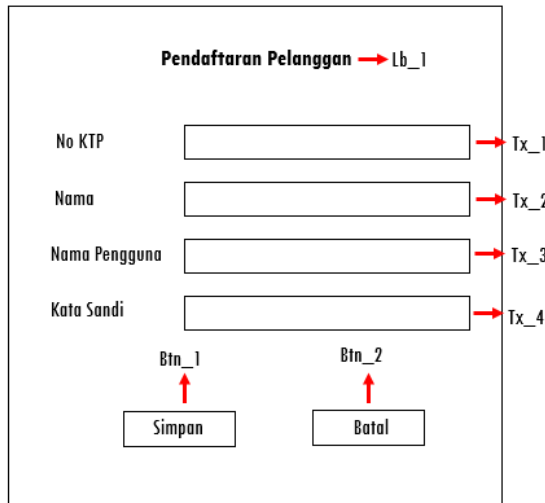
Gambar 3.43 merupakan perancangan antarmuka pelanggan pada *server*, dimana pada tampilan ini menampilkan semua data pelanggan. *Admin* dapat menyunting dan menghapus data pelanggan.

Tabel 3.22 Deskripsi Perancangan Antarmuka Pelanggan pada *Server*

Id_Objek	Jenis	Nama	Keterangan
Btn_1	Button	Edit	Jika diklik maka akan menyunting data pelanggan
Btn_2	Button	Delete	Jika diklik maka akan menghapus data pelanggan
Tbl_1	Tabel	Tabel	Berisi data pelanggan yang ada pada <i>database</i>

Tabel 3.22 mendeskripsikan perancangan antarmuka pelanggan pada *server* dengan detail mulai dari nama objek (*id_objek*), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.7 Perancangan Antarmuka Pendaftaran Pelanggan pada *Server*



Gambar 3.44 Perancangan Antarmuka Pendaftaran Pelanggan pada *Server*

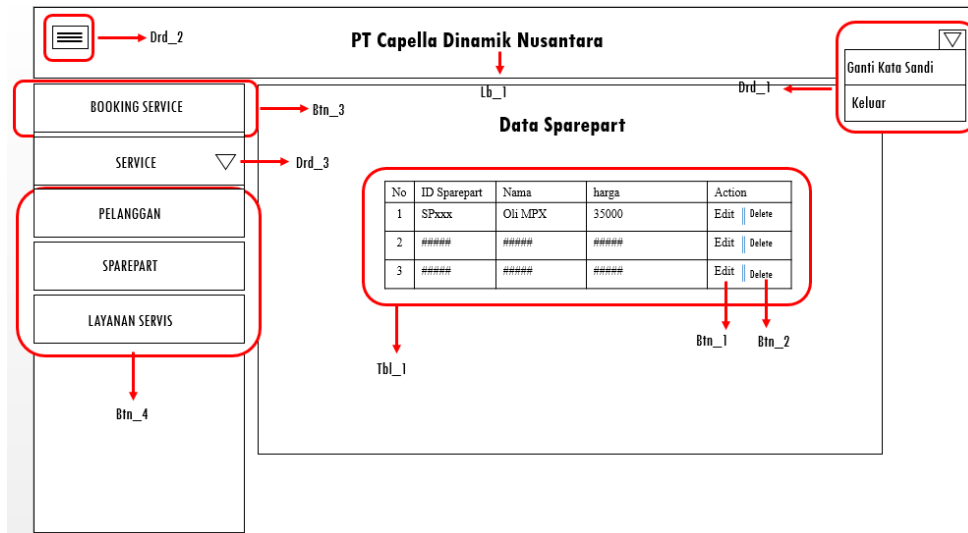
Gambar 3.44 merupakan perancangan antarmuka pendaftaran pelanggan, dimana *admin* memasukkan data-data pelanggan yang diperlukan.

Tabel 3.23 Deskripsi Perancangan Antarmuka Pendaftaran Pelanggan pada *Server*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	Pendaftaran pelanggan	Berada pada halaman pendaftaran pelanggan
Tx_1	Text field	No KTP	Diisi dengan no KTP
Tx_2	Text field	Nama	Diisi dengan nama sesuai KTP
Tx_3	Text field	Nama Pengguna	Diisi dengan nama pengguna untuk <i>login</i>
Tx_4	Text field	Kata sandi	Diisi dengan kata sandi pengguna baru
Btn_1	Button	Simpan	Jika diklik maka akan menyimpan data yang dimasukkan
Btn_2	Button	Batal	Jika diklik akan membatalkan pendaftaran

Tabel 3.23 mendeskripsikan perancangan antarmuka pendaftaran pelanggan pada *server* dengan detil mulai dari nama objek (id_objek), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.8 Perancangan Antarmuka Sparepart pada Server



Gambar 3.45 Perancangan Antarmuka Sparepart pada Server

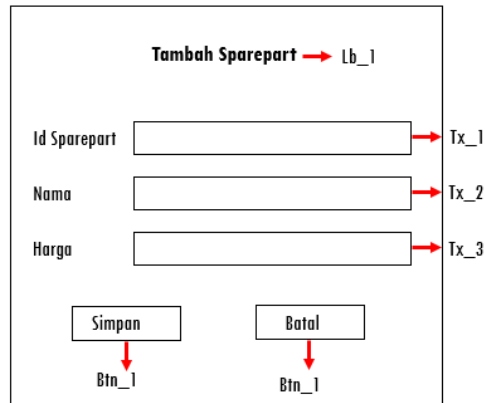
Gambar 3.45 merupakan perancangan antarmuka *sparepart* pada *server*, dimana pada tampilan ini menampilkan semua data *sparepart*. Admin dapat menyunting dan menghapus data *sparepart*.

Tabel 3.24 Deskripsi Perancangan Antarmuka Sparepart pada Server

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	PT Capella dinamik Nusantara	Logo aplikasi
Drd_1	Dropdown	Menu pengguna	Dropdown yang berisi <i>button</i> untuk mengganti kata sandi dan keluar dari hak akses
Drd_2	Dropdown	Menu	Berisi menu-menu yang ada pada aplikasi
Drd_3	Dropdown	Menu	Berisi sub menu
Btn_1	Button	Edit	Jika diklik maka akan menyunting data <i>sparepart</i>
Btn_2	Button	Delete	Jika diklik maka akan menghapus data <i>sparepart</i>
Btn_3	Button	Menu	Jika diklik maka akan ke menu yang kita inginkan, contoh <i>booking service</i> dan servis
Btn_4	Button	Menu	Jika diklik maka akan ke menu pelanggan, <i>sparepart</i> , layanan servis
Tbl_1	Tabel	Tabel	Berisi data <i>sparepart</i> yang ada pada <i>database</i>

Tabel 3.24 mendeskripsikan perancangan antarmuka *sparepart* pada *server* dengan detil mulai dari nama objek (*id_objek*), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.9 Perancangan Antarmuka Tambah *Sparepart* pada *Server*



Gambar 3.46 Perancangan Antarmuka Tambah *Sparepart* pada *Server*

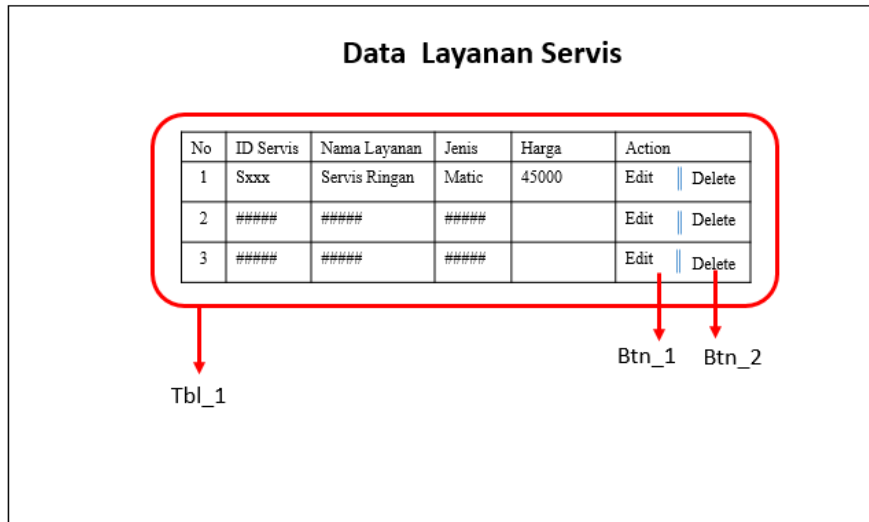
Gambar 3.46 merupakan perancangan antarmuka tambah *sparepart Admin* dapat menambahkan data *sparepart*.

Tabel 3.25 Deskripsi Perancangan Antarmuka Tambah *Sparepart* pada *Server*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	Pendaftaran pelanggan	Berada pada halaman pendaftaran pelanggan
Tx_1	Text field	Id <i>sparepart</i>	Diisi dengan id <i>sparepart</i>
Tx_2	Text field	Nama	Diisi dengan nama sesuai KTP
Tx_3	Text field	Harga	Diisi dengan harga
Btn_1	Button	Simpan	Jika diklik maka akan menyimpan data yang dimasukkan
Btn_2	Button	Batal	Jika diklik akan membatalkan tambah <i>sparepart</i>

Tabel 3.25 mendeskripsikan perancangan antarmuka tambah *sparepart* pada *server* dengan detail mulai dari nama objek (id_objek), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.10 Perancangan Antarmuka Layanan Servis pada Server



Gambar 3.47 Perancangan Antarmuka Layanan Servis pada Server

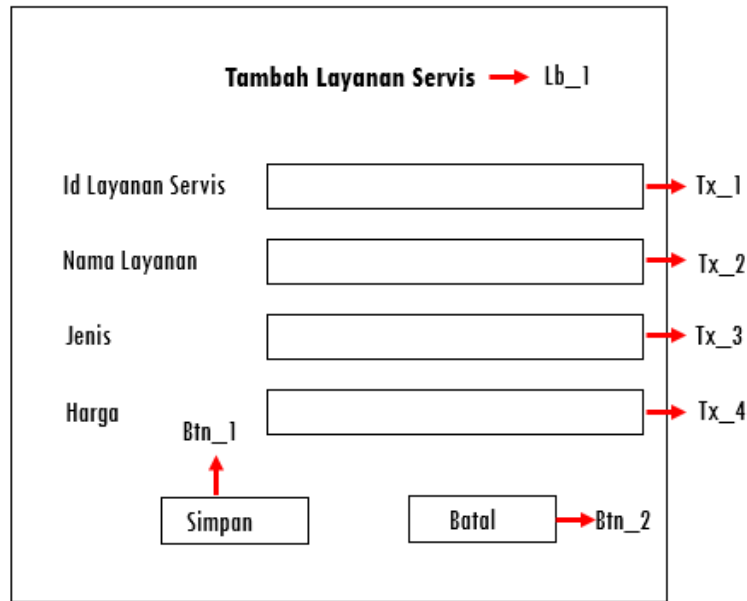
Gambar 3.47 merupakan perancangan antarmuka layanan servis pada server, dimana pada tampilan ini menampilkan semua data layanan servis. Admin dapat menyunting dan menghapus data layanan servis.

Tabel 3.26 Deskripsi Perancangan Antarmuka Layanan Servis pada Server

Id_Objek	Jenis	Nama	Keterangan
Btn_1	Button	Edit	Jika diklik maka akan menyunting data layanan servis
Btn_2	Button	Delete	Jika diklik maka akan menghapus data layanan servis
Tbl_1	Tabel	Tabel	Berisi data layanan servis yang ada pada database

Tabel 3.26 mendeskripsikan perancangan antarmuka layanan servis pada server dengan detail mulai dari nama objek (id_objek), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.11 Perancangan Antarmuka Tambah Layanan Servis pada *Server*



Gambar 3.48 Perancangan Antarmuka Tambah Layanan Servis pada *Server*

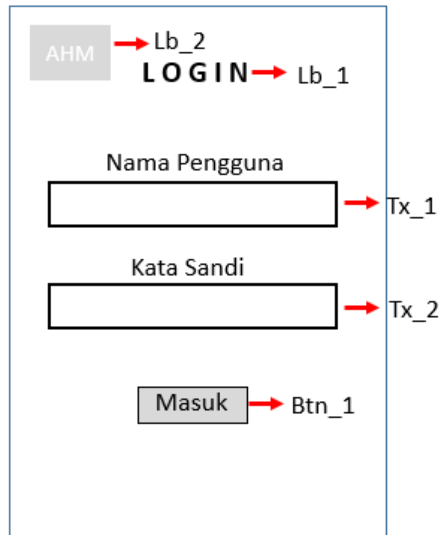
Gambar 3.48 merupakan perancangan antarmuka tambah layanan servis. *Admin* dapat menambahkan data layanan servis.

Tabel 3.27 Deskripsi Perancangan Antarmuka Tambah Layanan Servis pada *Server*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	Pendaftaran pelanggan	Berada pada halaman pendaftaran pelanggan
Tx_1	Text field	Id layanan servis	Diisi dengan id layanan servis
Tx_2	Text field	Nama Layanan	Diisi dengan nama layanan
Tx_3	Text field	Harga	Diisi dengan harga
Drd_1	Dropdown	Jenis	Berisi pilihan jenis yang ada pada
Btn_1	Button	Simpan	Jika diklik maka akan menyimpan data yang dimasukkan
Btn_2	Button	Batal	Jika diklik akan membatalkan tambah <i>sparepart</i>

Tabel 3.27 mendeskripsikan perancangan antarmuka tambah layanan servis pada *server* dengan detail mulai dari nama objek (id_objek), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.12 Perancangan Antarmuka *Login* pada *Client*



Gambar 3.49 Perancangan Antarmuka *Login* pada *Client*

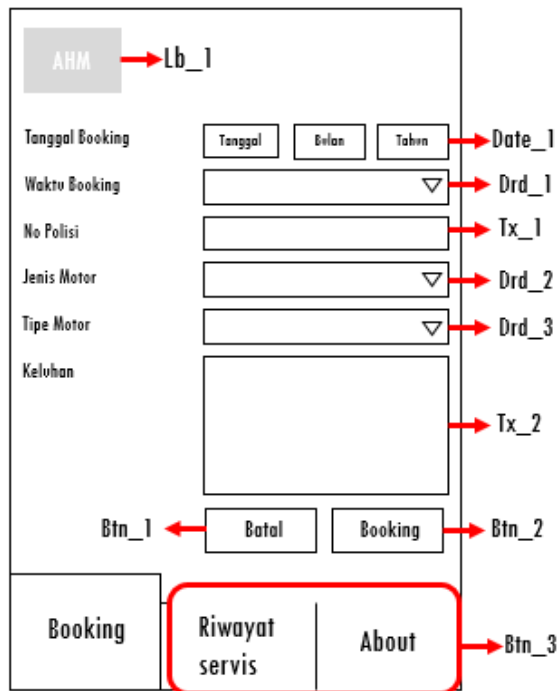
Gambar 3.49 merupakan perancangan antarmuka *login* pada *client* dimana pada tampilan ini pengguna memasukkan nama pengguna dan kata sandi untuk memulai menggunakan aplikasi.

Tabel 3.28 Deskripsi Perancangan Antarmuka *Login* pada *Client*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	Login	Tulisan yang menandakan halaman <i>login</i>
Lb_2	Label	AHM	Logo AHM
Tx_1	Text field	Nama pengguna	Diisi dengan nama pengguna
Tx_2	Text field	Kata sandi	Diisi dengan kata sandi pengguna
Btn_1	Button	Masuk	Jika diklik akan memvalidai nama pengguna dan kata sandi, jika benar akan masuk ke menu <i>booking</i>

Tabel 3.28 mendeskripsikan perancangan antarmuka *login client* dengan detail mulai dari nama objek (*id_objek*), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.13 Perancangan Antarmuka *Booking* pada *Client*



Gambar 3.50 Perancangan Antarmuka *Booking* pada *Client*

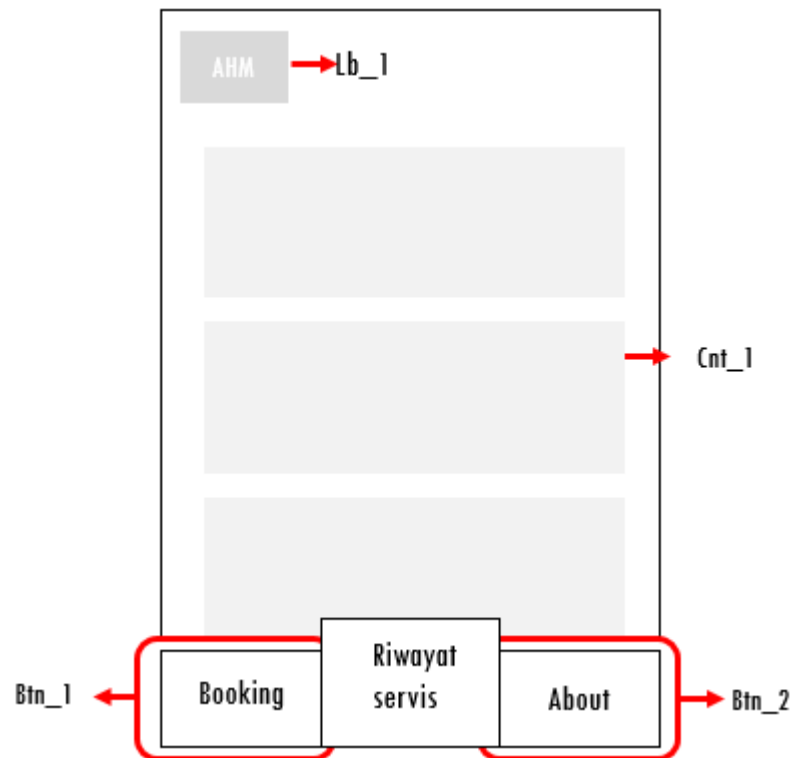
Gambar 3.50 merupakan perancangan antar muka *booking* pada *client*. Pelanggan memasukkan data-data *booking* yang dibutuhkan.

Tabel 3.29 Deskripsi Perancangan Antarmuka *Booking* pada *Client*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	AHM	Logo AHM
Drd_1	Dropdown	Waktu <i>booking</i>	Berisi waktu-waktu yang telah disediakan untuk <i>booking</i>
Drd_2	Dropdown	Jenis motor	Berisi jenis-jenis motor yang terdapat dalam <i>database</i>
Drd_3	Dropdown	Tipe motor	Berisi tipe-tipe motor yang terdapat dalam <i>database</i>
Date_1	Date	Tanggal <i>booking</i>	Diisi sesuai dengan tanggal <i>booking</i> yang diinginkan
Tx_1	Text field	No polisi	Diisi dengan no polisi kendaraan
Tx_2	Text field	keluhan	Diisi dengan keluhan
Btn_1	Button	Batal	Jika diklik maka akan membatalkan <i>booking</i>
Btn_2	Button	<i>Booking</i>	Jika diklik maka akan melakukan pengajuan <i>booking</i>
Btn_3	Button	Menu	Jika diklik maka akan ke menu lainnya seperti riwayat servis dan about

Tabel 3.29 mendeskripsikan perancangan antarmuka *booking* pada *client* dengan detail mulai dari nama objek (id_objek), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.14 Perancangan Antarmuka Riwayat Servis pada *Client*



Gambar 3.51 Perancangan Antarmuka Riwayat Servis pada *Client*

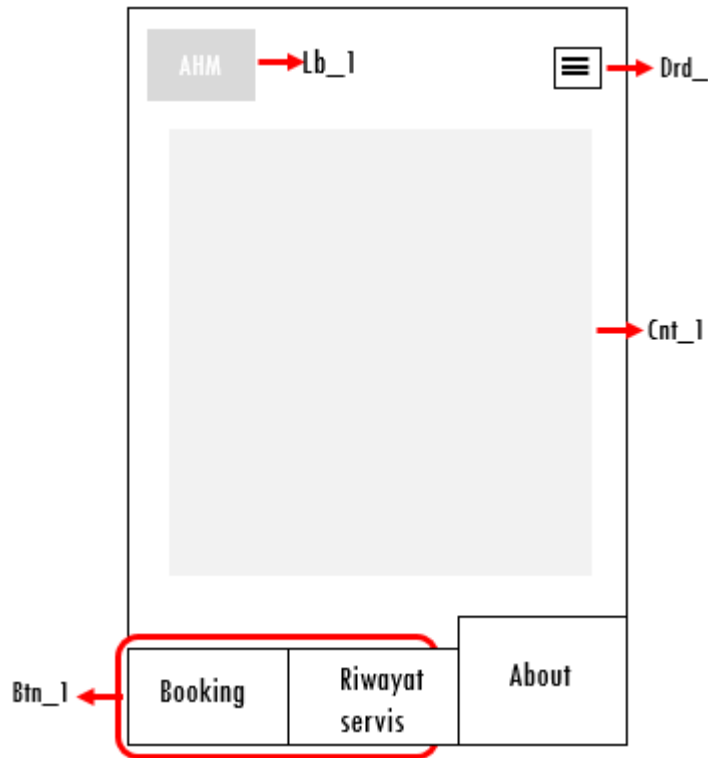
Gambar 3.51 merupakan perancangan antarmuka riwayat servis yang menampilkan *booking* yang telah dilakukan.

Tabel 3.30 Deskripsi Perancangan Antarmuka Riwayat Servis pada *Client*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	AHM	Logo AHM
Cnt_1	Content	Riwayat	Berisi data-data pengajuan <i>booking</i>
Btn_1	Button	Booking	Jika diklik maka akan ke menu <i>booking</i>
Btn_2	Button	About	Jika diklik maka akan ke menu <i>about</i>

Tabel 3.30 mendeskripsikan perancangan antarmuka riwayat servis pada *client* dengan detail mulai dari nama objek (*id_objek*), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

6.10.15 Perancangan Antarmuka About pada *Client*



Gambar 3.52 Perancangan Antarmuka *About* pada *Client*

Gambar 3.52 merupakan perancangan antarmuka *about* yang menampilkan konten tentang perusahaan.

Tabel 3.31 Perancangan Antarmuka *About* pada *Client*

Id_Objek	Jenis	Nama	Keterangan
Lb_1	Label	AHM	Logo AHM
Drd_1	Dropdown	Menu profile	Menampilkan menu seperti ubah kata sandi dan keluar
Cnt_1	Content	Content about	Berisi informasi tentang <i>dealer</i>
Btn_1	Button	Menu	Jika diklik maka akan ke menu booking atau riwayat servis

Tabel 3.31 mendeskripsikan perancangan antarmuka *about* pada *client* dengan detail mulai dari nama objek (*id_objek*), jenis objek, nama objek yang tertera serta keterangan dari objek tersebut.

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Struktur Tabel Basis Data

Implementasi struktur tabel basis data pada Sistem Aplikasi *Booking Service Astra Honda Motor* di Kota Batam berbasis *Client Server* pada *Platform* Android adalah sebagai berikut;

Tabel 4.1 Implementasi Struktur Tabel Pelanggan

Nama Field	Tipe	Panjang	Kunci	Atribut
no_ktp	Varchar	16	PK	NotNull
nama	Varchar	50		NotNull
namaPengguna	Varchar	20		NotNull
kataSandi	Varchar	25		NotNull

Tabel 4.2 Implementasi Struktur Tabel Motor

Nama Field	Tipe	Panjang	Kunci	Atribut
Nomor	int	3	PK	NotNull
tipe_mtr	varchar	30		NotNull
jns_mtr	varchar	6		NotNull

Tabel 4.3 Implementasi Struktur Tabel *Booking*

Nama Field	Tipe	Panjang	Kunci	Atribut
id_booking	int	255	PK	NotNull
Pelanggan	varchar	50		NotNull
tgl_booking	date			NotNull
tgl_pengajuan	date			NotNull
wkt_booking	time			NotNull
wkt_pengajuan	time			NotNull
no_polisi	varchar	6		NotNull
jns_mtr	varchar	15		NotNull
tipe_mtr	varchar	15		NotNull
nama_Layanan	varchar	20		NotNull
Harga	int	30		NotNull
Status	varchar	50		NotNull
Keluham	varchar	255		Null

Tabel 4.4 Implementasi Struktur Tabel Servis

Nama Field	Tipe	Panjang	Kunci	Atribut
id_booking	int	10		Null
id_sparepart	varchar	50		Null

Tabel 4.5 Implementasi Struktur Tabel *Sparepart*

Nama Field	Tipe	Panjang	Kunci	Atribut
id_sparepart	varchar	5	PK	NotNull

Nama Field	Tipe	Panjang	Kunci	Atribut
nama_sparepart	varchar	50		NotNull
hrg_sparepart	int	7		NotNull

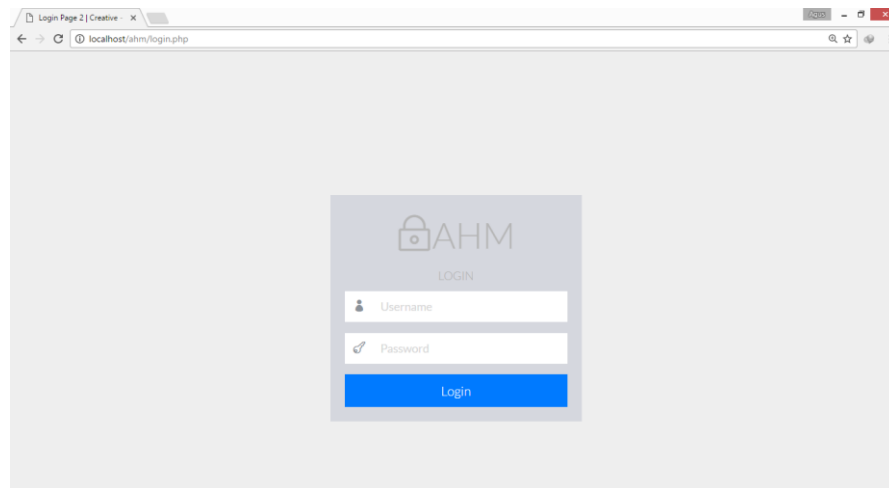
Tabel 4.6 Implementasi Struktur Tabel Layananservis

Nama Field	Tipe	Panjang	Kunci	Atribut
id_Layanan	int	4	PK	NotNull
nama_Layanan	varchar	20		NotNull
jns_Layanan	varchar	10		
hrg_Layanan	int	7		NotNull

4.2 Implementasi Proses

4.2.1 Implementasi Proses *Login* pada *Server*

Implementasi antarmuka prose *login*, terdiri dari dua buah textfield dan, satu button. Untuk lebih memahami dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi Antarmuka *Login* pada *Server*

Pada proses *login* sistem akan memvalidasi inputan *username* dan *password* aplikasi. Jika benar maka akan masuk ke halaman menu utama. Jika *username* dan *password* salah atau kosong kemudian admin mengklik tombol “login”, maka akan muncul pesan kesalahan. Potongan *source code* yang membangun antarmuka *login* terdapat pada *login.php* sedangkan yang membangun proses *login* terdapat pada *proses_login.php* yang dapat dilihat pada Tabel 4.7.

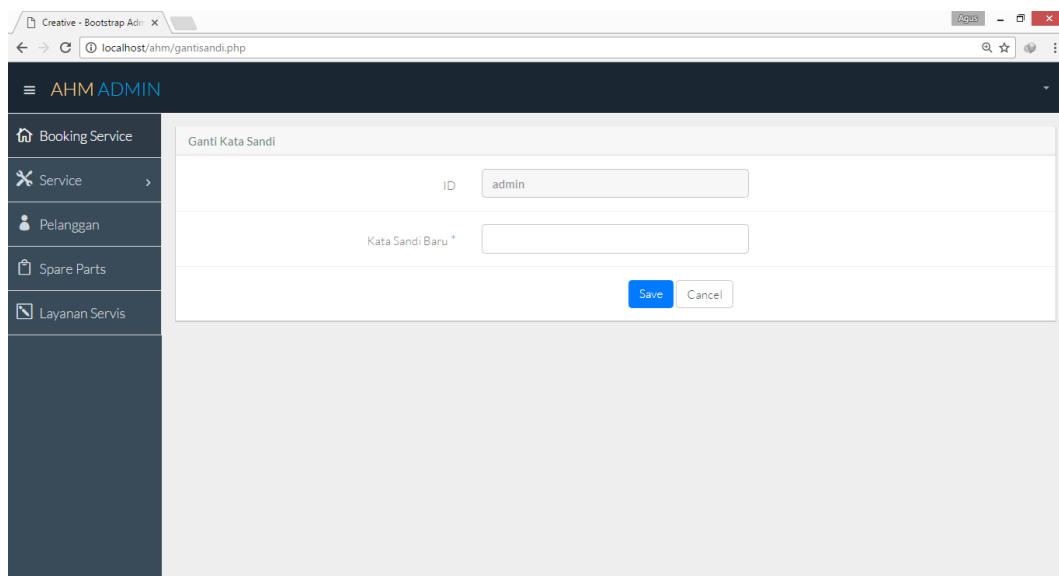
Tabel 4.7 Penjelasan *Source Code Login* pada *Server*

Kode	Penjelasan	File
------	------------	------

Kode	Penjelasan	File
<pre>\$id_pegawai = \$_POST['id_pegawai']; \$password = \$_POST['password'];</pre>	Mengambil variabel dari halaman <code>login.php</code>	<code>proses_login.php</code>
<pre>\$result = \$connect -> query ("SELECT * FROM pegawai WHERE id_pegawai = '\$id_pegawai' "); \$data = \$result -> fetch_assoc();</pre>	Query untuk memilih semua field pegawai dari tabel pegawai dimana <code>id_pegawai = variabel id_pegawai</code>	<code>proses_login.php</code>
<pre>if (\$id_pegawai == \$data['id_pegawai'] and \$password == \$data['password']) { \$_SESSION['id_pegawai'] = \$data['id_pegawai']; header('location:index.php'); }</pre>	Validasi apakah <code>id_pegawai</code> dan <code>password</code> sama dengan database, jika iya maka dibuat session untuk menyimpan data <code>id_pegawai</code> .	<code>proses_login.php</code>
<pre><script language="JavaScript"> window.alert("Maaf ID atau password yang anda masukkan salah"); window.location.href = 'login.php'; </script></pre>	Jika <code>id</code> dan <code>password</code> salah maka akan tampil pesan Maaf ID atau <code>password</code> yang anda masukkan salah	<code>proses_login.php</code>

4.2.2 Implementasi Proses Ganti Kata Sandi pada Server

Implementasi antarmuka ganti kata sandi, terdiri dari 1 inputan yaitu kata sandi baru kemudian 2 tombol yang berfungsi sebagai *save* dan *cancel*. Untuk lebih memahami bisa dilihat pada Gambar 4. 2.



Gambar 4.2 Implementasi Antarmuka Ganti Kata Sandi pada Server

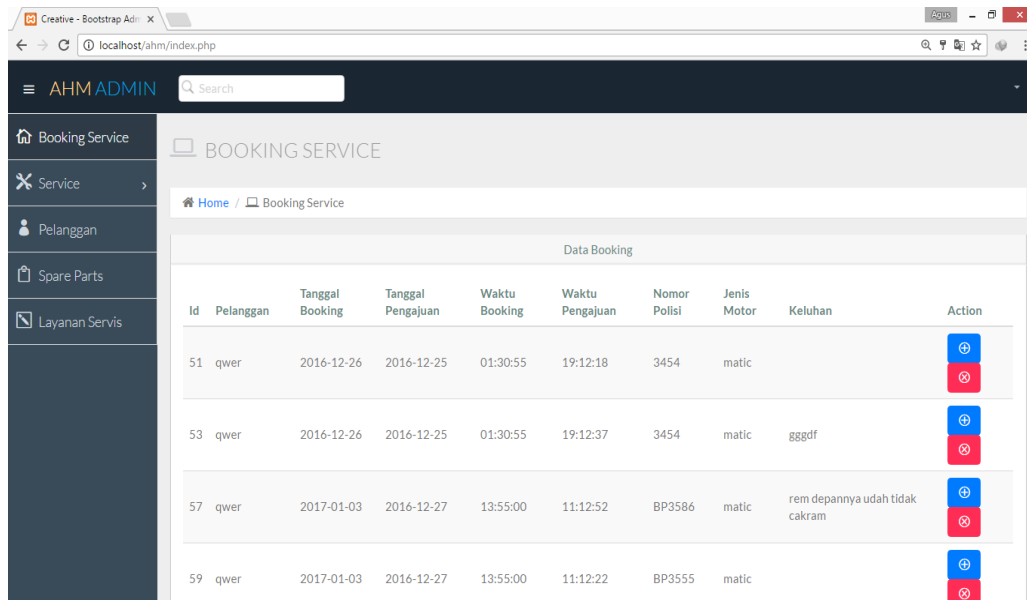
Pada proses ganti sandi, sistem menampilkan form ganti kata sandi yang terdapat pada `gantisandi.php` kemudian menyimpan kata sandi baru yang terdapat proses `ganti.php`. Untuk lebih memahami dapat dilihat pada Tabel 4.8.

Tabel 4.8 Penjelasan Source Code Ganti Kata Sandi

Kode	Penjelasan	File
<pre><input class="form-control" name="id_pegawai" value="<?=\$_SESSION["id_pegawai"];?>" type="text" readonly /> <input class="form-control" type="text" name="password" minlength="3" required /></pre>	Kode yang digunakan untuk inputan kata sandu baru	Gantisandi.php
<pre>\$id_pegawai = \$_POST['id_pegawai']; \$password = \$_POST['password'];</pre>	Kode yang digunakan untuk menampung variabel yang diinputkan	proses_ganti.php
<pre>\$tambah = \$connect -> query("UPDATE pegawai SET password='\$password' WHERE id_pegawai='\$id_pegawai'"); if (\$tambah){ echo "<script>alert('Berhasil Menyimpan')</script>"; echo "<script>window.open('index.php', '_se lf')</script>"; }else{ echo "<script>alert('Gagal Menyimpan')</script>"; echo "<script>window.open('index.php', '_se lf')</script>"; }</pre>	Kode yang digunakan untuk menyimpan kata sandi baru ke dalam database	proses_ganti.php

4.2.3 Implementasi Proses *Booking Service* pada *Server*

Implementasi antarmuka *booking service*, terdiri dari tabel daftar pelanggan yang mengajukan *booking* mulai dari nama pelanggan, tanggal *booking*, tanggal pengajuan, waktu *booking*, waktu pengajuan, nomor polisi, keluhan dan aksi yang dapat meyetujui maupun menolak pengajuan tersebut. Untuk lebih memahami bisa dilihat pada Gambar 4.3



Gambar 4.3 Implementasi Antarmuka *Booking Service* pada *Server*

Pada proses *booking service* yaitu menerima atau menolak pengajuan *booking* dengan cara mengubah status dari pengajuan *booking* yang dilakukan pelanggan. Data *booking service* yaitu dengan status memesan kemudian jika di terima maka akan menjadi dikonfirmasi sedangkan jika di tolak maka akan merubah statusnya menjadi ditolak. Potongan *source code* yang membangun antarmuka *booking service* terdapat pada *index.php*, sedangkan untuk mengubah status terdapat pada *status_booking.php* dan *hapusBooking.php* yang dapat dilihat pada Tabel 4.9.

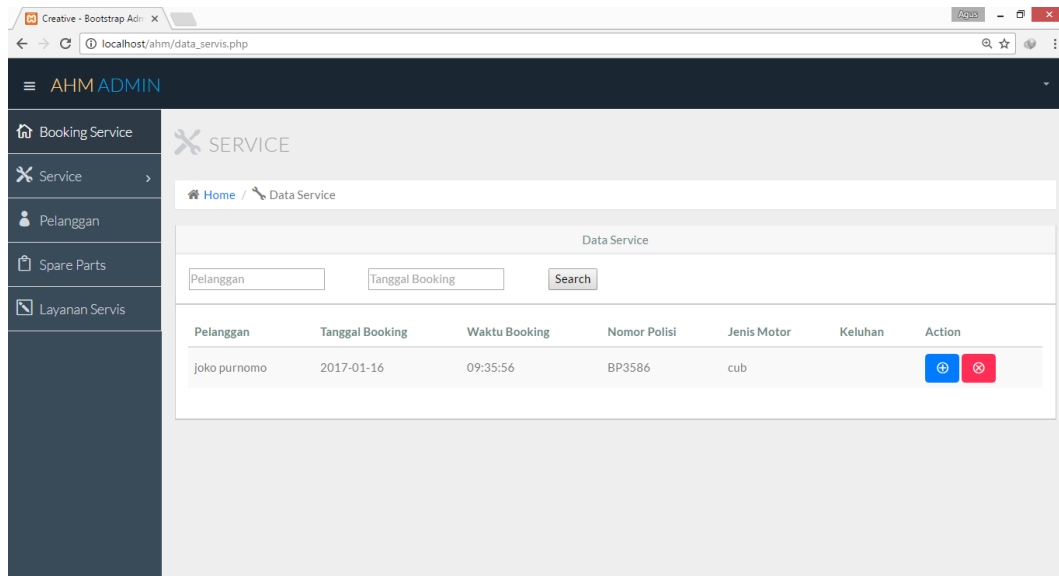
Tabel 4.9 Penjelasan *Source Code Booking Service*

Kode	Penjelasan	File
<pre>\$sql = "SELECT * FROM booking where status = 'memesan'"; \$kueri = \$connect -> query (\$sql); \$no = 1; while(\$data = \$kueri -> fetch_array())</pre>	Kode yang digunakan untuk memilih semua field <i>booking</i> pada tabel <i>booking</i> dengan status = memesan	Index.php
<pre><td class="center"><?php echo \$data['pelanggan']?></td> <td class="center"><?php echo \$data['tgl_booking']?></td> <td class="center"><?php echo \$data['tgl_pengajuan']?></td> <td class="center"><?php echo \$data['wkt_booking']?></td> <td class="center"><?php echo \$data['wkt_pengajuan']?></td> <td class="center"><?php echo \$data['no_polisi']?></td></pre>	Kode yang digunakan untuk menampilkan data pelanggan, tanggal <i>booking</i> , tanggal pengajuan, waktu <i>booking</i> , waktu pengajuan, nomor polisi. Jenis motor dan keluhan	Index.php

Kode	Penjelasan	File
<pre><td class="center"><?php echo \$data['jns_mtr']?></td> <td class="center"><?php echo \$data['Keluhan']?></td> <td class="center"></pre>		
<pre>if(isset(\$_GET['kode'])){ \$id_booking = \$_GET['kode']; }else{ echo"<script>alert('Gagal 1');document.location='index.php'</scr ipt>";} \$sql = "UPDATE booking SET status='Dikonfirmasi' WHERE id_booking='\$id_booking'"; \$kueri = mysql_query(\$sql); if(\$kueri){ echo"<script>alert('Booking telah dikonfirmasi');document.location='inde x.php'</script>"; }else{ echo"<script>alert('DATA GAGAL DIUPDATE')</script>"; echo mysql_error(); }</pre>	Kode yang digunakan untuk menyunting data <i>booking</i> dengan mengubah statusnya menjadi dikonfirmasi dengan id_booking = get["kode"] yang berasal dari pemilihan <i>booking</i>	Status_b ooking.p hp
<pre>if(isset(\$_GET['kode'])){ \$id_booking = \$_GET['kode']; }else{ echo"<script>alert('id_booking belum dipilih'); document.location='index.php' </script>";} \$sql = "UPDATE booking SET status='Ditolak' WHERE id_booking='\$id_booking'"; \$kueri = \$connect -> query(\$sql); if(\$kueri){ echo"<script>alert('BOOKING TELAH DITOLAK');document.location='index.php '</script>";} else{ echo"<script>alert('BOOKING GAGAL DITOLAK')</script>"; echo mysql_error();}</pre>	Kode yang digunakan untuk menyunting data <i>booking</i> dengan mengubah statusnya menjadi ditolak dengan id_booking = get["kode"] yang berasal dari pemilihan <i>booking</i>	hapusBo oking.ph p

4.2.4 Implementasi Proses Data Service pada Server

Implementasi antarmuka data *service*, terdiri dari tabel daftar data *service* mulai pelanggan, tanggal *booking*, waktu *booking*, nomor polisi, jenis motor, dan keluhan. Untuk lebih memahami bisa dilihat pada Gambar 4.4.



Gambar 4.4 Implementasi Antarmuka Data Service pada Server

Pada data *service* menampilkan data-data *service* yang terdapat pada *database* pada file *data_servis.php* serta melakukan servis atau membatalkan servis yang terdapat pada *servis.php* yang kemudian akan di proses servis dan *batalservis.php* untuk membatalkan *booking* yang telah dikonfirmasi, untuk lebih memahami dapat dilihat pada Tabel 4.10.

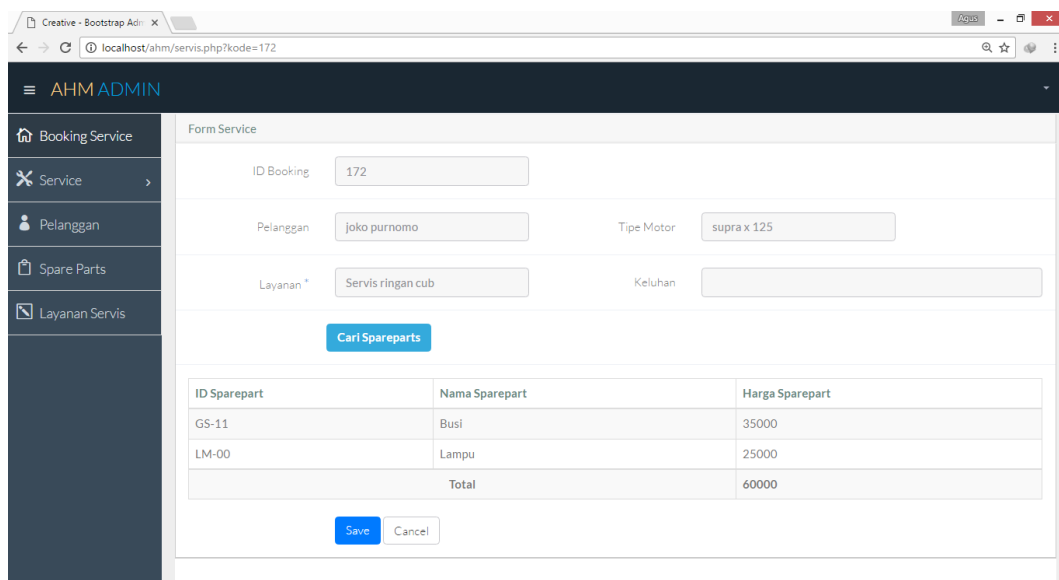
Tabel 4.10 Penjelasan Source Code Data Service

Kode	Penjelasan	File
<pre><?php include "koneksi.php"; \$name = isset(\$_GET['pelanggan']) ? \$_GET['pelanggan'] : null; \$tanggal = isset(\$_GET['tanggal']) ? \$_GET['tanggal'] : null; if (\$name != null OR \$tanggal != null) { \$sql = "SELECT * FROM booking where status = 'dikonfirmasi' AND (pelanggan = '\$name' OR tgl_booking = '\$tanggal')"; \$kueri = \$connect -> query (\$sql); \$no = 1; while(\$data = \$kueri -> fetch_array()){ ?></pre>	Kode yang digunakan untuk mengambil data <i>booking</i> dengan status = dikonfirmasi dari <i>database</i>	Data_servis.php
<pre><td class="center"><?php echo \$data['pelanggan']?></td> <td class="center"><?php echo \$data['tgl_booking']?></td> <td class="center"><?php echo \$data['wkt_booking']?></td> <td class="center"><?php echo \$data['no_polisi']?></td> <td class="center"><?php echo \$data['jns_mtr']?></td></pre>	Kode yang digunakan untuk menampilkan data pada tabel	Data_servis.php

Kode	Penjelasan	File
<pre><td class="center"><?php echo \$data['Keluhan']?></td> if(isset(\$_GET['kode'])){ \$id_booking = \$_GET['kode']; }else{ echo"<script>alert('id_booking belum dipilih');document.location='index.php'< /script>";} \$sql = "UPDATE booking SET status='Batal' WHERE id_booking='\$_id_booking'"; \$kueri = \$connect -> query(\$sql);</pre>	Kode yang digunakan untuk membatalkan <i>booking service</i> yang telah dikonfirmasi	Batalser vis.php

4.2.5 Implementasi Proses *Service* pada *Server*

Implementasi antarmuka *service* terdiri dari data-data *booking* yang dipilih, kemudian 1 *button* untuk mencari *sparepart* dan 1 *button* untuk menyimpan *service*. Untuk lebih memahami bisa dilihat pada Gambar 4. 5.



Gambar 4.5 Implementasi Antarmuka *Service* pada *Server*

Pada proses *service* yaitu menampilkan data pelanggan yang melakukan *service* kemudian memasukkan *sparepart* yang digunakan setelah *service* motor telah dilakukan dan data disimpan ke *database* dengan mengubah status menjadi selesai. Untuk dapat memahami dapat dilihat pada Tabel 4. 11

Tabel 4.11 Penjelasan *Source Code service*

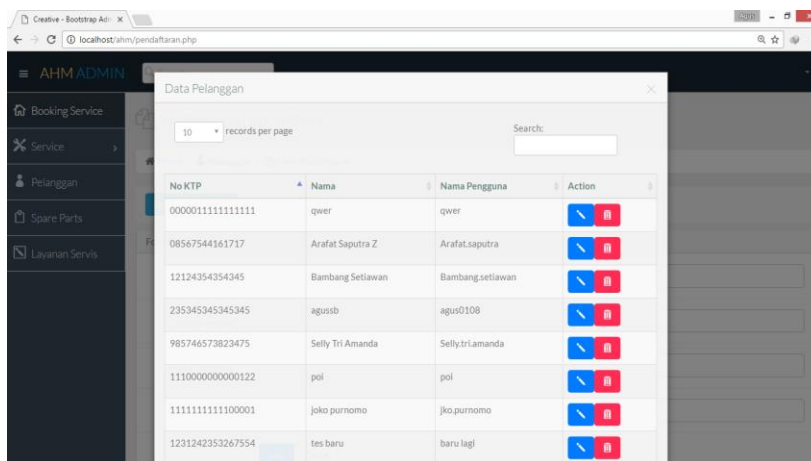
Kode	Penjelasan	File
<pre><?php include "koneksi.php"; if(isset(\$_GET['kode'])){ \$id_booking = \$_GET['kode'];</pre>	Kode yang digunakan untuk mengambil data <i>booking</i> berdasarkan	servis.php

Kode	Penjelasan	File
<pre> \$sql = "SELECT * FROM booking WHERE id_booking = '\$id_booking'"; \$kueri = \$connect -> query(\$sql); \$data = \$kueri -> fetch_array(); } ?> </pre>	id <i>booking</i> telah dipilih	
<pre> <input >="" <input="" <="" class="form-control" name="keluhan" pre="" readonly="" type="text" value="<?php echo \$data['Keluhan'] ;?>"/> </pre>	Kode yang digunakan untuk menampilkan isi dari data <i>booking</i> yang telah dipilih	servis.php
<pre> <?php //Data mentah yang ditampilkan ke tabel include "koneksi.php"; \$sql = 'SELECT * FROM sparepart'; \$kueri = \$connect -> query (\$sql); while (\$data = \$kueri -> fetch_array()) { ?> <tr class="pilih" data-id="<?php echo \$data['id_Sparepart']; ?>" data- nama="<?php echo \$data['nama_Sparepart']; ?>" data- hrg="<?php echo \$data['hrg_Sparepart']; ?>"> <td><?php echo \$data['id_Sparepart']; ?></td> <td><?php echo \$data['nama_Sparepart']; ?></td> <td><?php echo \$data['hrg_Sparepart']; ?></td> </tr> <?php } ?> </pre>	Kode yang digunakan untuk menampilkan pilihan <i>sparepart</i> yang digunakan dan menjumlahkan harga pada setiap <i>sparepartnya</i>	servis.php
<pre> var id = \$(this).attr('data-id'); var nama = \$(this).attr('data-nama'); var hrg = \$(this).attr('data-hrg'); \$('#addRow').append('<tr><td><input type="hidden" name="id[]" value="'+id+'"'>'+id+'</td><td>'+nama+'</ td><td class="harga">'+hrg+'</td></tr>'); var sum = 0; \$(".harga").each(function() { var value = \$(this).text(); if(!isNaN(value) && value.length != </pre>	Kode yang digunakan untuk mengambil data yang dipilih di modal dan ditampilkan di tabel baru	servis.php

Kode	Penjelasan	File
<pre> 0) { sum += parseFloat(value); } </pre>		
<pre> \$id_booking = \$_POST['id_booking']; if (isset(\$_POST['total'])) { \$kueri = 0; foreach (\$_POST['id'] as \$key => \$value) { \$sql = "INSERT INTO `servis`(`id_booking`, `id_sparepart`) VALUES ('\$id_booking', '\$value')"; \$kueri = \$connect -> query (\$sql); }if (\$kueri) { \$sqlupd = "UPDATE booking SET total= '".\$_POST['total']."' , status='Selesai' WHERE id_booking = '\$id_booking'"; \$kueriupd = \$connect- >query(\$sqlupd); } } else { \$sqlupd = "UPDATE booking SET status='Selesai' WHERE id_booking = '\$id_booking'"; \$kueriupd = \$connect->query(\$sqlupd); if (\$kueriupd) { echo "<script>alert('Telah Berhasil Melakukan Servis')</script>"; echo "<script>window.open('data_servis.php', ' self')</script>"; } } </pre>	Kode yang digunakan untuk menambahkan id booking serta id sparepart kedalam tabel servis dan menyimpan data servis yang telah dilakukan	Saveservis.php
<pre> else { \$sqlupd = "UPDATE booking SET status='Selesai' WHERE id_booking = '\$id_booking'"; \$kueriupd = \$connect->query(\$sqlupd); if (\$kueriupd) { echo "<script>alert('Telah Berhasil Melakukan Servis')</script>"; echo "<script>window.open('data_servis.php', ' self')</script>"; } } </pre>	Kode yang digunakan untuk menyimpan data servis yang telah dilakukan	saveservis.php

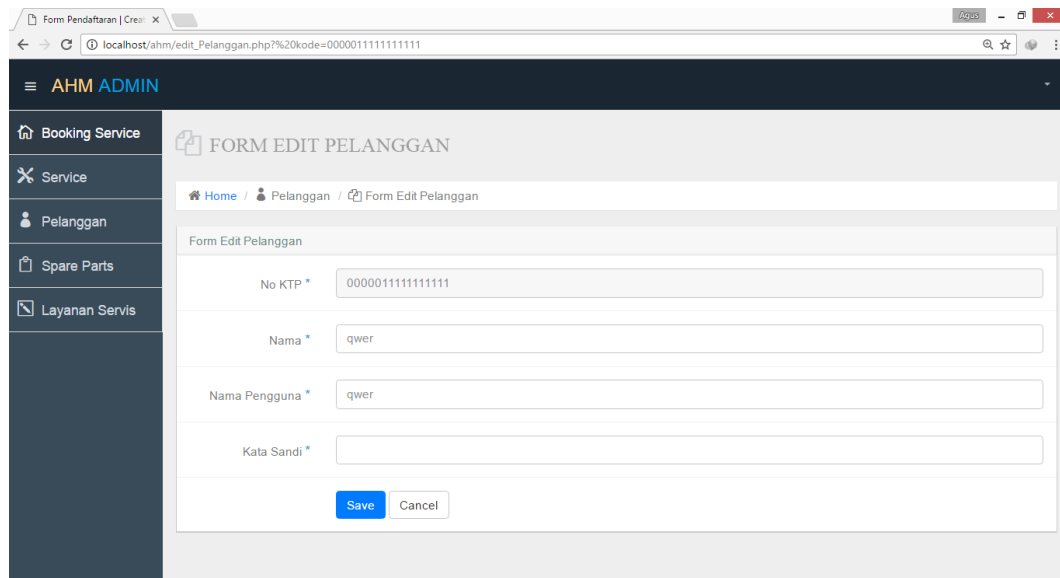
4.2.6 Implementasi Proses Pelanggan pada Server

Implementasi antarmuka pelanggan, terdiri dari tabel daftar data pelanggan mulai dari nomor ktp, nama, serta nama pengguna dan memiliki aksi untuk menyunting serta menghapus data pelanggan. Untuk lebih memahami bisa lihat Gambar 4.6.



Gambar 4.6 Implementasi Antarmuka Pelanggan pada Server

Implementasi antarmuka edit pelanggan, terdiri dari tabel masukkan data berupa nama, nama pengguna dan kata sandi yang baru. Untuk lebih memahami bisa lihat Gambar 4.7.



Gambar 4.7 Implementasi Antarmuka Edit Data Pelanggan

Implementasi pelanggan pada *server* yaitu menampilkan data-data pelanggan yang terdapat pada *database* pada file pendaftaran.php serta menyunting atau menghapus yang terdapat pada editPelanggan.php, edit_pelanggan.php dan hapusPelanggan.php, untuk lebih memahami dapat dilihat pada Tabel 4.12.

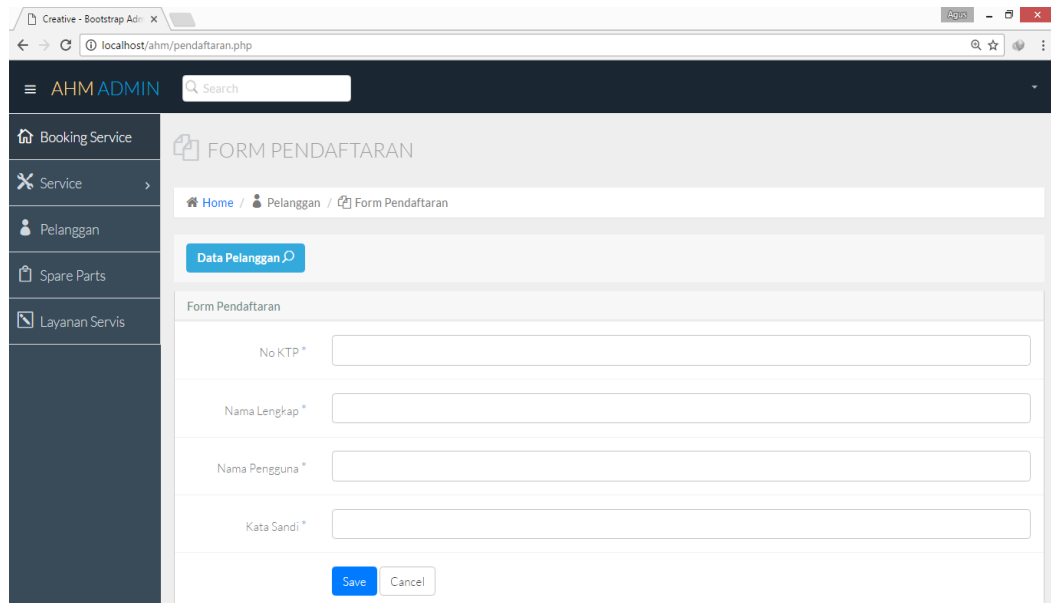
Tabel 4.12 Penjelasan Source Code Pelanggan

Kode	Penjelasan	File
<pre>mysql_connect('localhost', 'root', ''); mysql_select_db('latihan_ci'); \$query = mysql_query('SELECT * FROM pelanggan'); while (\$data = mysql_fetch_array(\$query)) {?> <tr><form method="post" action=""> <td><?php echo \$data['no_ktp']?></td> <td class="center"><?php echo \$data['nama']?></td> <td class="center"><?php echo \$data['namaPengguna']?></td> <td class="center"> <a class="btn btn-primary" href="edit_Pelanggan.php? kode=<?php echo \$data['no_ktp']?>"> <i class="icon_pencil"></i></pre>	<p>Kode yang digunakan untuk menampilkan semua field pelanggan dari tabel pelanggan kemudian menampilkan berupa no ktp, nama, nama pengguna. kode ini membuat tombol sunting untuk sunting data pelanggan, dan tombol hapus untuk menghapus data pelanggan</p>	<p>Pendaftaran.php</p>

Kode	Penjelasan	File
<pre><a onclick="return confirm('Apakah anda yakin menghapus data ini?');"class="btn btn-danger" href="hapusPelanggan.php?kode=<?php echo \$data['no_ktp']?>"> <i class="icon trash alt"></i></pre>		
<pre><input class="form-control " type="text" name="no_ktp" minlength="16" value="<?php echo \$data['no_ktp'] ;?>" required="required" readonly /> <input class="form-control" name="nama" minlength="3" value="<?php echo \$data['nama'] ;?>" type="text" required="required" /> <input class="form-control" name="namaPengguna" minlength="3" value="<?php echo \$data['namaPengguna'] ;?>" type="text" required /> <input class="form-control" name="kataSandi" minlength="8" type="password" required /></pre>	Inputan yang menampung nilai nomor ktp yang hanya bisa dilihat saja, kemudian nama, nama pengguna serta kata sandi	Edit_pelanggan.php
<pre>\$no_ktp = \$_POST['no_ktp']; \$nama = \$_POST['nama']; \$namaPengguna = \$_POST['namaPengguna']; \$kataSandi = \$_POST['kataSandi']; \$tambah = \$connect -> query("UPDATE pelanggan SET kataSandi='\$kataSandi', nama='\$nama', namaPengguna='\$namaPengguna' WHERE no_ktp='\$no_ktp'");</pre>	Kode yang digunakan untuk menyimpan data no ktp, nama, nama pengguna, serta kata sandi yang berasal dari inputan lalu disimpan ke dalam variabel yang sudah ditentukan	editPelanggan.php
<pre>if(isset(\$_GET['kode'])){ \$no_ktp = \$_GET['kode']; }else{ echo"<script>alert('no_ktp belum dipilih');document.location='pendaftaran.php'</script>";} \$sql = "delete from pelanggan WHERE no_ktp='\$no_ktp'"; \$kueri = \$connect -> query(\$sql);</pre>	Kode yang digunakan untuk menghapus data pelanggan sesuai data yang dipilih	hapusPelanggan.php

4.2.7 Implementasi Proses Pendaftaran Pelanggan pada Server

Implementasi antarmuka pendataran, terdiri dari inputan no ktp, nama lengkap, nama pengguna dan kata sandi. Untuk lebih memahami bisa lihat Gambar 4.8



Gambar 4.8 Implementasi Antarmuka Pendaftaran Pelanggan pada Server

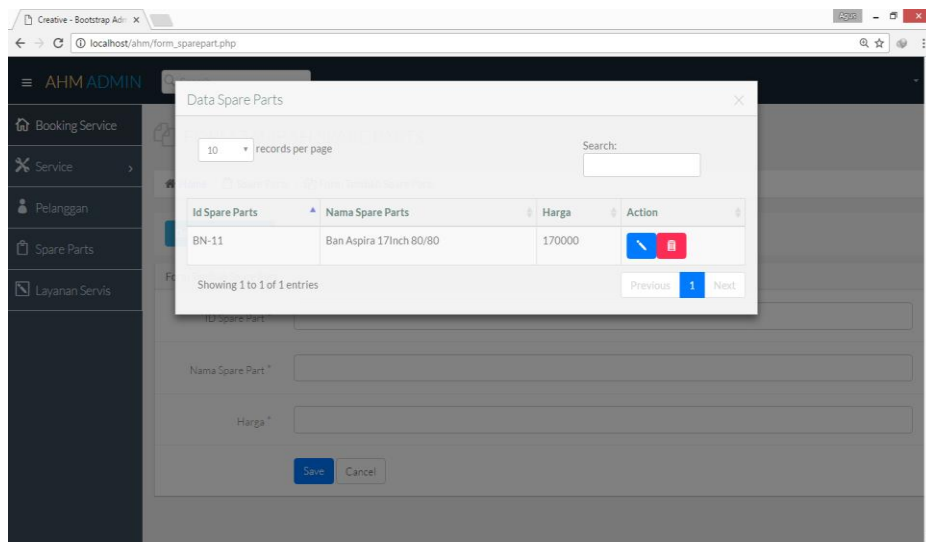
Implementasi pendaftaran pada server yaitu memasukkan data-data pelanggan baru yang belum memiliki akun terdapat pada pendaftaran.php dan tambahPelanggan.php. Untuk lebih memahami dapat dilihat pada Tabel 4. 13.

Tabel 4.13 Penjelasan Source Code Pendaftaran Pelanggan

Kode	Penjelasan	File
<pre><input class="form-control" name="no_ktp" minlength="16" maxlength="16" type="text" required /> <input class="form-control" type="text" name="nama" minlength="3" required /> <input class="form-control " type="text" name="namaPengguna" minlength="3" maxlength="20" required /><input class="form-control" id="password" name="kataSandi" minlength="8" type="password" required /></pre>	<p>Inputan yang menampung nilai no ktp, nama, nama pengguna, dan kata sandi</p>	<p>Pendaftaran.php</p>
<pre>\$no_ktp = \$_POST['no_ktp']; \$nama = \$_POST['nama']; \$namaPengguna = \$_POST['namaPengguna']; \$kataSandi = \$_POST['kataSandi']; \$tambah = \$connect -> query("INSERT INTO pelanggan values ('\$no_ktp', '\$nama', '\$namaPengguna', '\$k ataSandi')");</pre>	<p>Kode yang digunakan untuk menyimpan data no ktp, nama, nama pengguna, serta kata sandi yang berasal dari inputan lalu disimpan ke dalam variabel yang sudah ditentukan</p>	<p>tambahPelanggan.php</p>

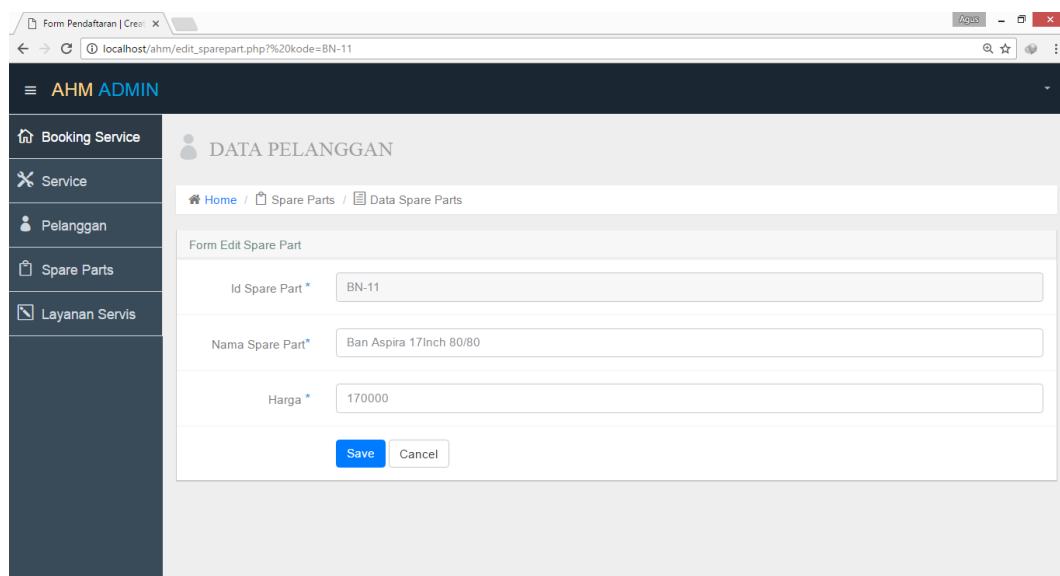
4.2.8 Implementasi Proses Sparepart pada Server

Implementasi antarmuka *sparepart*, terdiri dari tabel daftar data *sparepart* mulai dari id *sparepart*, nama *sparepart*, serta harga dan memiliki aksi untuk menyunting serta menghapus data *sparepart*. Untuk lebih memahami bisa lihat Gambar 4.9.



Gambar 4.9 Implementasi Antarmuka Sparepart pada Server

Implementasi antarmuka edit *sparepart* terdiri dari tabel masukkan data berupa id *sparepart*, nama *sparepart* dan harga. Untuk lebih memahami bisa lihat Gambar 4.10.



Gambar 4.10 Implementasi Antarmuka Edit Sparepart

Implementasi *sparepart* pada *server* yaitu menampilkan data-data *sparepart* yang terdapat pada *database* pada file *form_sparepart.php* serta menyunting atau menghapus yang terdapat pada *editSparepart.php*, *edit_sparepart.php* dan *hapusSparepart.php*, untuk lebih memahami dapat dilihat pada Tabel 4.14.

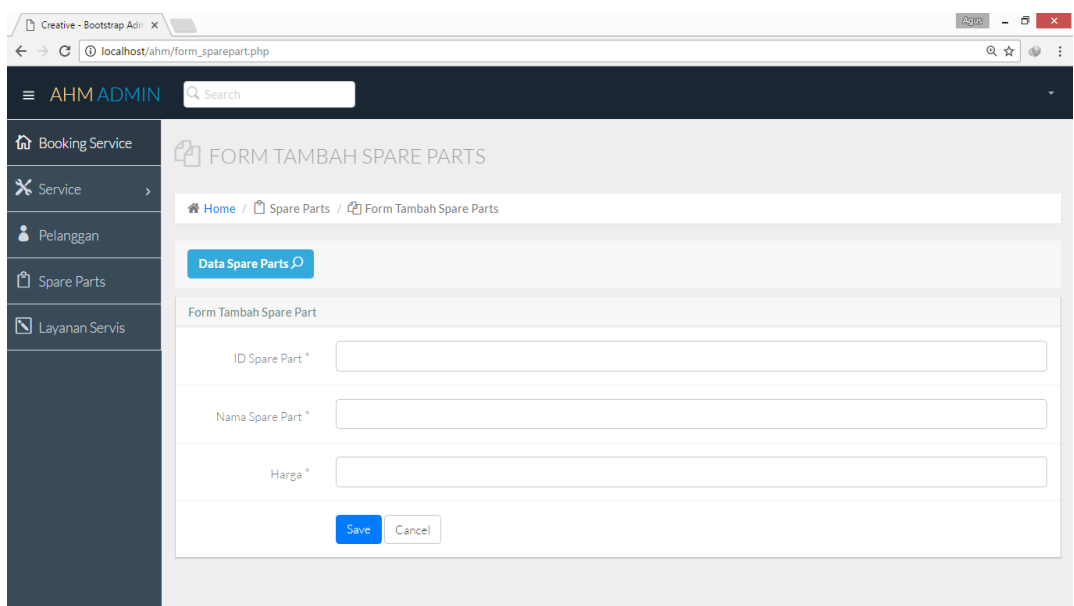
Tabel 4.14 Penjelasan Source Code Sparepart

Kode	Penjelasan	File
<pre>mysql_connect('localhost', 'root', ''); mysql_select_db('latihan_ci'); \$query = mysql_query('SELECT * FROM sparepart'); while (\$data = mysql_fetch_array(\$query)) ?><tr><form method="post" action=""><td><?php echo \$data['id_Sparepart']?></td> <td class="center"><?php echo \$data['nama_Sparepart']?></td> <td class="center"><?php echo \$data['hrg_Sparepart']?></td> <td class="center"> <a class="btn btn-primary" href="edit_sparepart.php? kode=<?php echo \$data['id_Sparepart']?>"> <i class="icon_pencil"></i> <a onclick="return confirm('Apakah anda yakin menghapus data ini?');" class="btn btn-danger" href="hapusSparepart.php?kode=<?php echo \$data['id_Sparepart']?>"> <i class="icon_trash_alt"></i> </pre>	<p>Kode yang digunakan untuk menampilkan semua field <i>sparepart</i> dari tabel <i>sparepart</i> kemudian menampilkan berupa id <i>sparepart</i>, nama <i>sparepart</i>, harga. kode ini membuat tombol sunting untuk nyunting data <i>sparepart</i>, dan tombol hapus untuk menghapus data <i>sparepart</i></p>	<p>Form_sparepart.php</p>
<pre><input class="form-control " type="text" name="id_Sparepart" minlength="16" value="<?php echo \$data['id_Sparepart'] ;?>" required="required" readonly /> <input class="form-control" name="nama_Sparepart" minlength="3" value="<?php echo \$data['nama_Sparepart'] ;?>" type="text" required="required" /> <input class="form-control" name="hrg_Sparepart" minlength="3" value="<?php echo \$data['hrg_Sparepart'] ;?>" type="text" required /></pre>	<p>Inputan yang menampung nilai id <i>sparepart</i>, nama <i>sparepart</i>, dan harga</p>	<p>Edit_sparepart.php</p>
<pre>\$id_Sparepart = \$_POST['id_Sparepart']; \$nama_Sparepart = \$_POST['nama_Sparepart']; \$hrg_Sparepart = \$_POST['hrg_Sparepart']; \$tambah = \$connect -> query("UPDATE</pre>	<p>Kode yang digunakan untuk menyimpan id <i>sparepart</i>, nama <i>sparepart</i>, dan harga yang berasal dari inputan lalu disimpan</p>	<p>editSparepart.php</p>

Kode	Penjelasan	File
<pre> sparepart SET nama_Sparepart='\$nama_Sparepart',hrg_Sp arepart='\$hrg_Sparepart'WHERE id_Sparepart='\$id_Sparepart'); </pre>	ke dalam variabel yang sudah ditentukan	
<pre> if(isset(\$_GET['kode'])){ \$id_Sparepart = \$_GET['kode']; }else{ echo"<script>alert('id_Sparepart belumdipilih');document.location='form_ sparepart.php'</script>";} \$sql = "delete from sparepart WHERE id_Sparepart='\$id_Sparepart'"; \$kueri = \$connect -> query(\$sql); </pre>	Query untuk menghapus data <i>sparepart</i> sesuai data yang dipilih	hapusSparepart.php

4.2.9 Implementasi Proses Tambah *Sparepart* pada Server

Implementasi antarmuka tambah *sparepart*, terdiri dari inputan id *sparepart*, nama *sparepart*, dan harga. Untuk lebih memahami bisa lihat Gambar 4.11.



Gambar 4.11 Implementasi Antarmuka Tambah *Sparepart* pada Server

Implementasi tambah *sparepart* pada server yaitu memasukkan data-data *sparepart* baru terdapat pada form_sparepart.php dan tambahSparepart.php. Untuk lebih memahami dapat dilihat pada Tabel 4.15.

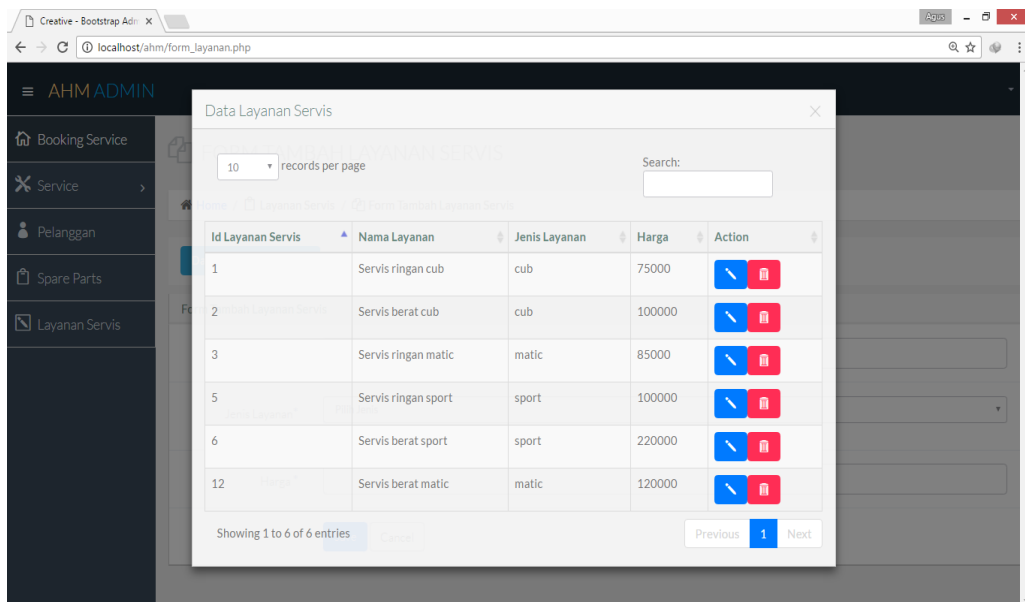
Tabel 4.15 Penjelasan Source Code Tambah *Sparepart*

Kode	Penjelasan	File
<pre> <input class="form-control" name="id_Sparepart" minlength="" type="text" required /> <input class="form-control" type="text" name="nama_Sparepart" minlength="" required /> </pre>	Inputan yang menampung nilai id <i>sparepart</i> , nama <i>sparepart</i> , dan harga	Form_sparepart.php

Kode	Penjelasan	File
<pre><input class="form-control " type="text" name="hrg_Sparepart" minlength="" /></pre>		
<pre>\$id_Sparepart = \$_POST['id_Sparepart']; \$nama_Sparepart = \$_POST['nama_Sparepart']; \$hrg_Sparepart = \$_POST['hrg_Sparepart']; \$tambah = \$connect -> query("INSERT INTO sparepart values ('\$id_Sparepart', '\$nama_Sparepart', '\$h rg_Sparepart')");</pre>	Kode yang digunakan untuk menyimpan id <i>sparepart</i> , nama <i>sparepart</i> , dan harga yang berasal dari inputan lalu disimpan ke dalam variabel yang sudah ditentukan	tambahSparepart.php

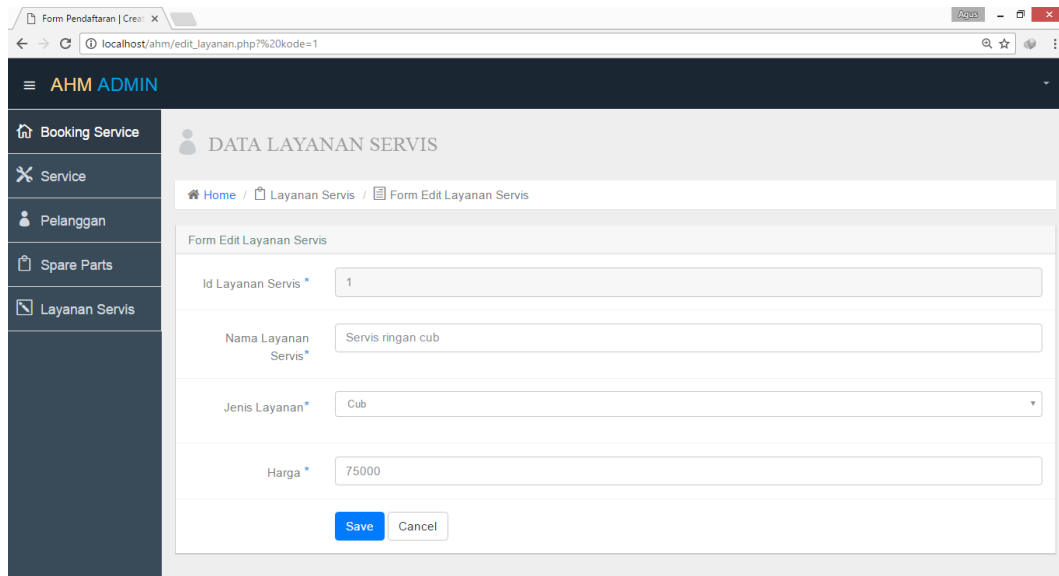
4.2.10 Implementasi Proses Layanan Servis pada Server

Implementasi antarmuka layanan servis, terdiri dari tabel data nama layanan, jenis layanan serta harga dan memiliki aksi untuk menyunting serta menghapus data *sparepart*. Untuk lebih memahami bisa lihat Gambar 4.12.



Gambar 4.12 Implementasi Antarmuka Layanan Servis pada Server

Implementasi antarmuka edit layanan servis terdiri dari tabel masukkan data berupa nama layanan, jenis layanan serta harga. Untuk lebih memahami bisa lihat Gambar 4.13.



Gambar 4.13 Implementasi Edit Layanan

Implementasi layanan servis pada *server* yaitu menampilkan data-data layanan servis yang terdapat pada *database* pada file `form_layanan.php` serta menyunting atau menghapus yang terdapat pada `editLayanan.php`, `edit_layanan.php` dan `hapusLayanan.php`, untuk lebih memahami dapat dilihat pada Tabel 4.16.

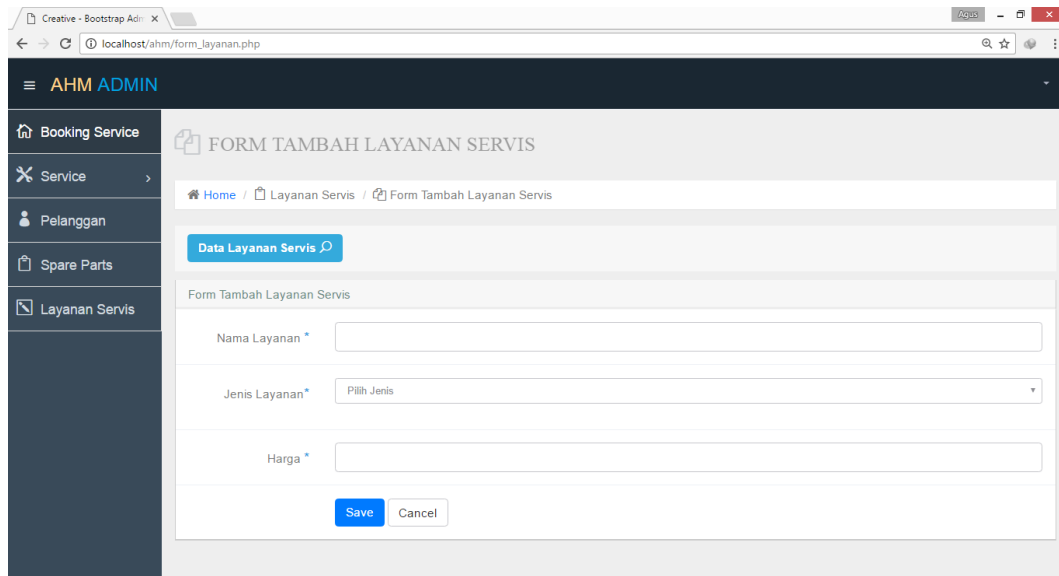
Tabel 4.16 Penjelasan *Source Code* Proses Layanan Servis

Kode	Penjelasan	File
<pre>mysql_connect('localhost', 'root', ''); mysql_select_db('latihan_ci'); \$query = mysql_query('SELECT * FROM layananservis'); while (\$data = mysql_fetch_array(\$query)) {?> <tr> <td><?php echo \$data['id_Layanan']?></td> <td class="center"><?php echo \$data['nama_Layanan']?></td> <td class="center"><?php echo \$data['jns_Layanan']?></td> <td class="center"><?php echo \$data['hrg_Layanan']?></td> <td class="center"> <a class="btn btn-primary" href="edit_layanan.php? kode=<?php echo \$data['id_Layanan']?>"> <i class="icon_pencil"></i> <a onclick="return confirm('Apakah anda yakin menghapus data ini?');" class="btn</pre>	<p>Kode yang digunakan untuk menampilkan semua field layanan dari tabel layanan servis kemudian menampilkan berupa nama layanan, jenis layanan serta harga. kode ini membuat tombol sunting untuk nyunting data layanan, dan tombol hapus untuk menghapus data layanan</p>	<p>form_layanan.php</p>

Kode	Penjelasan	File
<pre>btn-danger" href="hapusLayanan.php?kode=<?php echo \$data['id_Layanan']?>"> <i class="icon_trash_alt"></i> </pre>		
<pre><input class="form-control " type="text" name="id_Layanan" minlength="16" value="<?php echo \$data['id_Layanan'] ;?>" required="required" readonly /> <input class="form-control" name="nama_Layanan" minlength="3" value="<?php echo \$data['nama_Layanan'] ;?>" type="text" required="required" /> <select name="jns_Layanan" class="form- control input-sm m-bot15"> <option></option> <option value="cub">Cub</option> <option value="matic">Matic</option> <option value="sport">Sport</option> </select></pre>	Inputan yang menampung nilai nama layanan, jenis layanan serta harga	Edit_layanan.php
<pre>\$id_Layanan = \$_POST['id_Layanan']; \$nama_Layanan = \$_POST['nama_Layanan']; \$jns_Layanan = \$_POST['jns_Layanan']; \$hrng_Layanan = \$_POST['hrng_Layanan']; \$tambah = \$connect -> query("UPDATE layananservis SET nama_Layanan='\$nama_Layanan' ,jns_Layanan='\$jns_Layanan' ,hrng_Layanan='\$hrng_Layanan' WHERE id_Layanan='\$id_Layanan'");</pre>	Kode yang digunakan untuk menyimpan nama layanan, jenis layanan serta harga yang berasal dari inputan lalu disimpan ke dalam variabel yang sudah ditentukan	editLayanan.php
<pre>if(isset(\$_GET['kode'])){ \$id_Layanan = \$_GET['kode']; }else{ echo"<script>alert('id_Sparepart belum dipilih');document.location='form_layan an.php'</script>"; } \$sql = "delete from layananservis WHERE id_Layanan='\$id_Layanan'"; \$kueri = \$connect -> query(\$sql);</pre>	Kode yang digunakan untuk menghapus data layanan sesuai data yang dipilih	hapusLayanan.php

4.2.11 Implementasi Proses Tambah Layanan Servis pada Server

Implementasi antarmuka tambah layanan servis, terdiri dari inputan nama layanan, jenis layanan dan harga. Untuk lebih memahami bisa lihat Gambar 4.14



Gambar 4.14 Implementasi Antarmuka Tambah Layanan Servis pada Server

Implementasi tambah layanan servis pada server yaitu memasukkan data-data layanan servis baru terdapat pada form_layanan.php dan tambahLayanan.php untuk lebih memahami dapat dilihat pada Tabel 4.17.

Tabel 4.17 Penjelasan Source Code Tambah Layanan Servis

Kode	Penjelasan	File
<pre><input class="form-control" type="text" name="nama_Layanan" minlength="" required /> <select name="jns_Layanan" class="form-control input-sm m-bot15"> <option>Pilih Jenis</option> <option value="cub">Cub</option> <option value="matic">Matic</option> <option value="sport">Sport</option> </select> <input class="form-control " type="text" name="hrg_Layanan" minlength="" /></pre>	Inputan yang menampung nilai nama layanan servis, jenis layanan servis dan harga	Form_layanan.php
<pre>\$nama_Layanan = \$_POST['nama_Layanan']; \$jns_Layanan = \$_POST['jns_Layanan']; \$hrg_Layanan = \$_POST['hrg_Layanan']; \$tambah = \$connect -> query("INSERT INTO layananServis values ('','\$nama_Layanan','\$jns_Layanan','\$hrg_Layanan')");</pre>	Kode yang digunakan untuk menyimpan nama layanan servis, jenis layanan servis dan harga yang berasal dari inputan lalu disimpan ke dalam variabel yang sudah ditentukan	tambahLayanan.php

4.2.12 Implementasi Proses *Login* pada *Client*

Implementasi antarmuka login pada *client*, terdapat dua inputan *username* dan *password* dan tombol *login*. Untuk lebih memahami bisa dilihat pada Gambar 4.15



Gambar 4.15 Implementasi Antarmuka Proses *Login* pada *Client*

Pada proses *login* sistem akan memvalidasi masukan *username* dan *password* aplikasi. Jika benar maka akan masuk ke halaman menu utama. Jika *username* dan *password* salah kemudian pelanggan mengklik tombol “login”, maka akan muncul pesan kesalahan. Untuk lebih memahami dapat dilihat pada Tabel 4.18

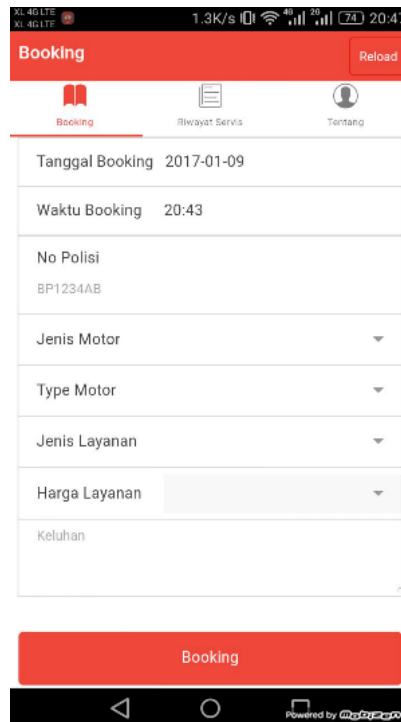
Tabel 4.18 Penjelasan *Source Code* Proses *Login* pada *Client*

Kode	Penjelasan	File
<pre><input type="text" placeholder="Username" ng- model="data.username"> <input type="password" placeholder="Password" ng- model="data.password"></pre>	Inputan yang menampung nilai <i>username</i> dan <i>password</i>	Login.html
<pre>login: function(params) { return \$http.post(base_url+'login.php', params); }</pre>	Kode yang digunakan untuk post data <i>username</i> dan <i>password</i> ke login.php	Service.js
<pre>if (res) { if (res.status == 1) {</pre>	Data yang digunakan dari	Controller.js

Kode	Penjelasan	File
<pre> window.localStorage.setItem("nama", res.data[0].nama); \$state.go('tab.dash'); } else { alert(res.message); } } else { alert("Koneksi bermasalah"); } }) </pre>	<p>hasil query login.php, jika status = 1 maka akan menyimpan di local storage nama dari <i>username</i> dan lanjut ke halaman tab dash yaitu halaman <i>booking</i></p>	
<pre> \$namapengguna = \$request->username; \$password = \$request->password; </pre>	<p>Kode yang digunakan untuk mendeklasrasikan variabel username dan password</p>	Login.php
<pre> \$sql = "SELECT * FROM pelanggan where namaPengguna = '\$namapengguna' and kataSandi = '\$password'"; \$query = mysql_query(\$sql); \$row = mysql_num_rows(\$query); \$hasil = array(); if (\$row > 0) { while (\$user = mysql_fetch_assoc(\$query)){ \$all[] = \$user; } \$hasil['status'] = 1; \$hasil['message'] = "Login berhasil"; \$hasil['data'] = \$all; echo json_encode(\$hasil); } else { \$hasil['status'] = 0; \$hasil['message'] = "Username atau password salah!"; echo json_encode(\$hasil); } </pre>	<p>Kode yang digunakan untuk memilih data pelanggan berdasarkan inputan kemudian jika datanya ada maka status = 1 dan jika tidak ada maka status = 0, lalu hasil tersebut dikirim kembali ke service.js</p>	Login.php

4.2.13 Implementasi Proses *Booking* pada *Client*

Implementasi antarmuka *booking* pada *client* terdapat *date picker* dan *time picker*, kemudian 2 inputan untuk no polisi dan keluhan, lalu terdapat 3 *dropdown* untuk memilih jenis motor, tipe motor dan jenis layanan. Untuk lebih memahami bisa dilihat pada Gambar 4.16.



Gambar 4.16 Implementasi Antarmuka Proses *Booking* pada *Client*

Pada proses *booking* setelah memunculkan form, sistem akan menampilkan pilihan tanggal dan waktu. Pada proses *booking* pengguna harus mengisi sesuai urutan mulai dari tanggal, waktu, no polisi, jenis motor, tipe motor, jenis layanan dan keluhan kemudian mengklik tombol “*booking*” untuk mengirmkan data *booking* tersebut. Untuk lebih memahami dapat dilihat pada Tabel 4.19.

Tabel 4.19 Penjelasan *Source Code Booking*

Kode	Penjelasan	File
<pre> \$scope.nama = localStorage.getItem("nama"); \$scope.dateValue = new Date(); \$scope.timeValue = new Date(); </pre>	Kode yang digunakan untuk mengambil nama dari login dan mengambil tanggal serta wakt dari sistem	controlle r.js
<pre> \$cat = isset(\$request->cat) ? \$request->cat : ""; switch (\$cat) { case 'type': \$sql = "SELECT tipe_mtr, nama_Layanan FROM motor a LEFT JOIN layananservis b ON a.jns_mtr=b.jns_Layanan WHERE a.jns_mtr = '\$request->jns'"; \$query = mysql_query(\$sql); \$row = mysql_num_rows(\$query); </pre>	Kode yang digunakan untuk menampilkan kategori pilihan tipe motor dan nama layanan yang di ambil dari <i>database</i> untuk ditampilkan pada <i>client</i>	Listmoto r.php

Kode	Penjelasan	File
<pre>\$hasil = array(); if (\$row > 0) { while (\$data = mysql_fetch_assoc(\$query)) { \$tipemtr[] = \$data['tipe_mtr']; \$namaLayanan[]=\$data['nama_Layanan'];}</pre>		
<pre>\$sql = "SELECT hrg_Layanan FROM layananservis WHERE nama_Layanan = '\$request->jns'"; \$query = mysql_query(\$sql); \$row = mysql_num_rows(\$query); \$hasil = array();</pre>	Kode yang digunakan untuk mengambil harga dari nama layanan yang dipilih	Listmotor.php
<pre>listmotor: function(data) { if (data) { return \$http.post(base_url+'listmotor.php', data); } else { var http = \$http.get(base_url+'listmotor.php'); return http; }},</pre>	Kode yang digunakan untuk menuju listmotor.php yang terdapat	Service.js
<pre><button ng-disabled=!harga class="button button-block button-assertive" ng- click="saveBook(dateValue, timeValue, nopolis, jns, type, nmlayanan, harga, keluhandesc)"> Booking </button></pre>	Kode yang digunakan untuk menampung value yang diinputkan	Tabdash.html
<pre>\$scope.saveBook = function(dateValue, timeValue, nopolis, jns, type, nmlayanan, harga, keluhandesc) { if (jns.jns_mtr) { var dates = \$filter('date')(dateValue, "yyyy-MM- dd"); var times = \$filter('date')(timeValue, "H:mm:ss"); \$scope.databook = { 'pelanggan': \$scope.nama, 'dateValue': dates, 'timeValue': times, 'nopolis': nopolis, 'jns': jns.jns_mtr, 'type': type, 'nmLayanan': nmlayanan, 'harga': harga, 'keluhandesc': keluhandesc };</pre>	Data yang diinputkan ditampung dalam objek	Service.js
<pre>\$pelanggan = \$request->pelanggan; \$tgl_booking = \$request->dateValue; \$tgl_pengajuan = date("Y-m-d"); \$wkt_booking = \$request->timeValue; \$wkt_pengajuan = date("H:i:s"); \$no_polisi = \$request->nopolis; \$jns = \$request->jns; \$type = \$request->type; \$nama_Layanan = \$request->nmLayanan; \$hrg_Layanan = \$request->harga; \$keluhan = isset(\$request->keluhandesc) ? \$request->keluhandesc : "";</pre>	Kode yang digunakan untuk mendeklarasikan variabel yang dibutuhkan dalam insertbooking.php	Insertbooking.php
<pre>\$sql = "insert into booking values</pre>	Kode yang digunakan	Insertbo

Kode	Penjelasan	File
<pre>(', '\$pelanggan', '\$tgl_booking', '\$tgl_pengajuan', '\$wkt_booking', '\$wkt_pengajuan', '\$no_polisi', '\$jns', '\$type', '\$nama_Layanan', '\$hrng_Layanan', '', ' Memesan', '\$keluhan'); \$query = mysql_query(\$sql); if (\$query) { \$hasil['status'] = 1; \$hasil['message'] = "Booking telah berhasil dikirim"; echo json_encode(\$hasil); } else { \$hasil['status'] = 0; \$hasil['message'] = "gagal mengirim, Kesalahan koneksi !!!"; echo json_encode(\$hasil); } ?></pre>	<p>untuk memasukkan value booking ke dalam <i>database</i></p>	<p>oking.php</p>

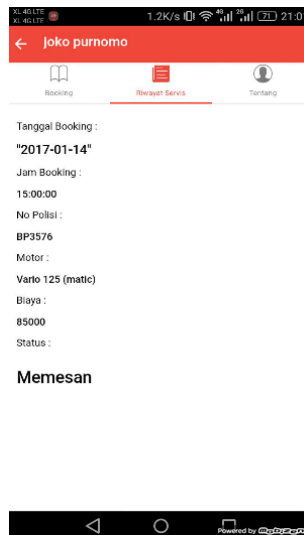
4.2.14 Implementasi Proses Riwayat Servis pada *Client*

Implementasi antarmuka riwayat servis terdiri dari informasi-informasi yang dilakukan pengguna saat *booking*. Data riwayat yang ditampilkan diambil dari *database*. Untuk lebih memahami dapat dilihat pada Gambar 4.17.



Gambar 4.17 Implementasi Antarmuka Riwayat Servis pada *Client*

Implementasi antarmuka detail riwayat servis terdiri dari data-data booking berdasarkan id booking yang diambil dari *database*. Untuk lebih memahami dapat dilihat pada Gambar 4.18.



Gambar 4.18 Implementasi Antarmuka *Detail Riwayat*

Pada proses riwayat servis, sistem menampilkan data *booking*, mulai dari pengambilan data dari *database* hingga menampilkannya. Untuk lebih memahami dapat dilihat pada Tabel 4.20.

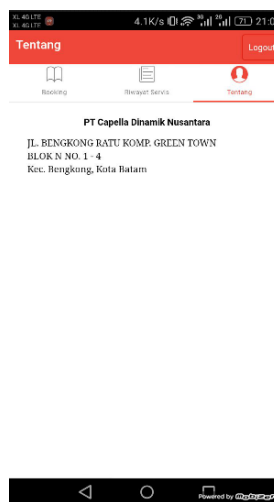
Tabel 4.20 Penjelasan *Source Code* Riwayat Servis

Kode	Penjelasan	File
<pre><h2>{{riwayat.tgl_booking}} {{riwayat.wkt_booking}}</h2> <p>{{riwayat.tipe_mtr}}</p> <p>{{riwayat.harga currency: 'Rp. '}}</p> <h2>{{riwayat.status}}</h2></pre>	Kode digunakan untuk menampilkan data <i>booking</i> yang diambil dari <i>database</i>	tab-chats.html
<pre>riwayat: function(name) { var data = {'nama': name}; return \$http.post(base_url+'riwayat.php', data); }, riwayatbyId: function(id) { var data = {'id': id}; return \$http.post(base_url+'riwayatbyId.php', data); }</pre>	Kode digunakan untuk mengambil data riwayat.php dan riwayatbyId.php	service.js
<pre>\$scope.riwayat = function() { booking.riwayat(\$scope.nama).success(function(res) { \$scope.riwayats = res.data; })} \$scope.riwayat();</pre>	Kode ini untuk mengembalikan <i>value</i> yang telah dipanggil	Controller.js
<pre>\$sql = "SELECT * FROM booking WHERE pelanggan = '\$request->nama' ORDER BY tgl_booking DESC"; \$query = mysql_query(\$sql); \$row = mysql_num_rows(\$query); \$hasil = array();</pre>	Kode yang digunakan untuk menampilkan riwayat dari <i>database</i>	riwayat.php

Kode	Penjelasan	File
<pre> \$mysql = "SELECT * FROM booking WHERE id_booking = '\$request->id'"; \$query = mysql_query(\$mysql); \$row = mysql_num_rows(\$query); \$hasil = array(); </pre>	Kode yang digunakan untuk menampilkan riwayat dari <i>database</i> berdasarkan id booking	riwayatbyId.php

4.2.15 Implementasi Proses *About* pada *Client*

Implementasi antarmuka *about* hanya terdiri dari text. Untuk lebih memahami bisa dilihat pada Gambar 4.19.



Gambar 4.19 Implementasi Antarmuka *About* pada *client*

Pada proses *about*, sistem hanya menampilkan berupa informasi dari bengkel tersebut. Untuk dapat memahami dapat dilihat pada Tabel 4.21.

Tabel 4.21 Penjelasan *Source Code About*

Kode	Penjelasan	File
<pre> <p align="center">PT Capella Dinamik Nusantara</p> <p><pre> JL. BENGKONG RATU KOMP. GREEN TOWN BLOK N NO. 1 - 4 Kec. Bengkong, Kota Batam </pre> </p> </pre>	Kode yang digunakan untuk menampilkan text	Tab-about.html

4.3 Pengujian

4.3.1 Skenario Pengujian

Nama Penguji : Nur Zahрати Janah. M.sc

Tanggal Pengujian : 13 Januari 2017

Tempat Pengujian : Ruang 702 Politeknik Negeri Batam

Tabel 4.1 Skenario Pengujian *Booking Service*

No	Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
1	Login client	<ol style="list-style-type: none"> Pengguna memasukan username dan password Pengguna menekan tombol login 	Username : "jko.purnomo" Password : "12345678"	Login berhasil dan masuk kedalam sistem	
			Username : "jko.purnomo" Password : "test1"	Login gagal dan muncul pesan kesalahan	
2	Melakukan booking service sepeda motor	<ol style="list-style-type: none"> Pengguna mengisi form booking (tanggal booking, waktu booking, no polisi, jenis motor, type motor, jenis layanan, harga layanan, keluhan) Pengguna menekan tombol booking 	Tanggal booking : 31-01-2017 Waktu booking : 14.00 No polisi : BP2345 Jenis motor : matic Type motor : scoopy Jenis layanan : service ringan matic Harga layanan : 85.000 Keluhan : oli habis	Data booking berhasil disimpan dan muncul pesan alert berhasil	
			Tanggal booking : 31-01-2017 Waktu booking : 14.00	Tombol booking tidak bisa ditekan dan booking gagal dilakukan.	

No	Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
			No polisi : Jenis motor : Type motor : Jenis layanan : Harga layanan : Keluhan : oli habis		
3	Menampilkan riwayat servis	1. Admin memilih menu riwayat servis 2. Sistem menampilkan riwayat servis	Data riwayat servis tersedia	Data riwayat servis berhasil ditampilkan	
			Data riwayat servis tidak tersedia	Data riwayat servis tidak berhasil ditampilkan	
4	Menampilkan data informasi dealer	1. Pengguna memilih menu about	Data informasi dealer	Berhasil menampilkan data informasi dealer	
5	Login server	1. Pengguna memasukkan username dan password 2. Pengguna menekan tombol login	Username : "admin" Password : "12345"	Login berhasil dan masuk kedalam sistem	
			Username : "admin" Password : "test1"	Login gagal dan muncul pesan kesalahan	
6	Menambah data pelanggan	1. Admin mengisi form menambah data pelanggan(no ktp, nama lengkap, nama pengguna, kata sandi) 2. Admin menekan tombol save	No KTP : 2343334242332344 Nama Lengkap : feбри Nama pengguna : ipeb Kata Sandi : feбри11	Data pelanggan berhasil disimpan kedalam sistem	

No	Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
			No KTP : 2343334242332344 Nama Lengkap : feбри Nema pengguna : ipeb Kata Sandi :	Data pelanggan tidak berhasil disimpan dan muncul pesan <i>alert</i>	
7	Menyunting data pelanggan	1. Admin mengubah atau menyunting data pelanggan (nama lengkap, nama pengguna, kata sandi) 2. Admin menekan tombol save	No KTP : 2343334242332344 Nama Lengkap : ratno Nema pengguna : eno Kata Sandi : ratno123	Data pelanggan berhasil diubah dan disimpan kedalam sistem	
			No KTP : 2343334242332344 Nama Lengkap : ratno Nema pengguna : Kata Sandi : ratno123	Data pelanggan tidak berhasil diubah dan muncul pesan alert	
8	Menghapus data pelanggan	1. Admin memilih data yang akan dihapus 2. Admin menekan tombol hapus 3. Muncul pesan konfirmasi	Admin memilih pesan konfirmasi : ya Admin memilih pesan konfirmasi : tidak	Data pelanggan berhasil dihapus Data pelanggan tidak dihapus	

No	Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
9	Menambah layanan service	<ol style="list-style-type: none"> Admin menambah data layanan service berupa nama layanan, jenis layanan, harga. Admin menekan tombol save 	Nama layanan : service ringan Jenis layanan : matic Harga : 85.000	Data layanan service berhasil ditambahkan dan disimpan	
			Nama layanan : service ringan Jenis layanan : matic Harga :	Data layanan service tidak berhasil ditambahkan dan muncul pesan alert	
10	Mengubah layanan service	<ol style="list-style-type: none"> Admin mengubah data layanan service berupa nama layanan, jenis layanan, harga. Admin menekan tombol save 	Nama layanan : service ringan Jenis layanan : matic Harga : 100.000	Data layanan service berhasil diubah dan disimpan	
			Nama layanan : service ringan Jenis layanan : matic Harga :	Data layanan service tidak berhasil ditambahkan dan muncul pesan alert	
11	Menghapus layanan service	<ol style="list-style-type: none"> Admin memilih data yang akan dihapus Admin menekan tombol hapus Muncul pesan konfirmasi 	Admin memilih pesan konfirmasi : ya	Data layanan service berhasil dihapus	
			Admin memilih pesan konfirmasi : tidak	Data layanan service gagal dihapus	
12	Menambah data spare part	<ol style="list-style-type: none"> Admin menambahkan data spare part pada form data spare part (id spare part, nama spare part, harga) 	ID Spare Part : LM-11 Nama Spare Part : Lampu Harga : 25.000	Data spare part berhasil ditambahkan dan disimpan	

No	Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
		2. Admin menekan tombol save	ID Spare Part : LM-11 Nama Spare Part : Lampu Harga :	Data spare part gagal ditambahkan dan muncul pesan alert	
13	Mengubah data spare part	1. Admin mengubah data spare part berupa nama spare part, harga 2. Admin menekan tombol save	ID Spare Part : LM-11 Nama Spare Part : Lampu Harga :50.000	Data spare part berhasil diubah dan disimpan	
			ID Spare Part : LM-11 Nama Spare Part : Lampu Harga :	Data spare part gagal diubah dan muncul pesan alert	
14	Menghapus data spare part	1. Admin memilih data spare part yang ingin dihapus 2. Admin menekan tombol hapus 3. Muncul pesan konfirmasi	Admin memilih pesan konfirmasi : ya	Data spare part berhasil dihapus dan muncul pesan alert	
			Admin memilih pesan konfirmasi : tidak	Data spare part tidak berhasil dihapus	
15	Menampilkan data booking	1. Admin memilih menu booking service 2. Sistem menampilkan data booking yang tersedia	Data booking tersedia	Data booking berhasil ditampilkan	
			Data booking tidak tersedia	Data booking gagal ditampilkan	
16	Fitur mengubah status booking service	1. Admin memilih menu booking service 2. Sistem menampilkan data booking yang tersedia 3. Admin memilih untuk mengkonfirmasi atau menolak booking	Data booking service dikonfirmasi	Data booking berhasil dikonfirmasi dan muncul pesan alert berhasil	
			Data booking service ditolak	Data booking ditolak dan muncul pesan alert	

No	Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
17	Menambah service	<ol style="list-style-type: none"> Admin memilih data booking dengan status yang sudah dikonfirmasi Sistem menampilkan data booking yang sudah dipilih Admin menambah spare part 	Spare part : - Busi - Lampu	Data service berhasil disimpan dan muncul pesan alert berhasil	
			Sparepart :	Data service berhasil disimpan dan muncul pesan alert berhasil	
18	Menampilkan riwayat servis	<ol style="list-style-type: none"> Admin memilih menu riwayat servis Sistem menampilkan riwayat servis 	Data riwayat servis tersedia	Data riwayat servis berhasil ditampilkan	
			Data riwayat servis tidak tersedia	Data riwayat servis tidak berhasil ditampilkan	

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian pada “Aplikasi *Booking Service* Astra Honda Motor berbasis *Client Server* pada *platform* Android” maka didapat kesimpulan bahwa aplikasi mampu :

1. Melakukan *booking service* pada sisi *client* serta melihat riwayat servis yang telah dilakukan.
2. Mengelola data servis berupa data booking, data layanan dan data sparepart serta dapat mengelola data pelanggan.

5.2 Saran

Sebagai langkah pengembangan aplikasi ini pada masa yang akan datang, adapun saran dari penulis antara lain:

1. Menambahkan fitur pembayaran pada aplikasi *client* dengan sistem saldo.
2. Menambahkan fitur kasir pada server.

DAFTAR PUSTAKA

- Abdulloh, R., 2016, Easy & Simple Web Programming, PT Elex Media Komputindo, Jakarta.
- Ariefianto, H., DKK, 2012, Perancangan Dan Implementasi Aplikasi Pemesanan Makanan Dan Minuman Berbasis Client Server Dengan Platform Android , *jurnal*, Politeknik Telkom, Bandung.
- Dietz, F, 2013, AngularJS Succintly, Syncfusion, USA.
- Henderi, 2008, Unified Modelling Language, Tangerang, Raharja Enrichment Centre (REC).
- Henderi, 2007, Diktat Kuliah UML STMIK AMIKOM, STMIK AMIKOM, Bandung.
- Kadir, A, 2008, Belajar Database menggunakan MYSQL, Yogyakarta, Graha Ilmu
- Kotler, P., 2007, Manajemen Pemasaran Jilid II (edisi bahasa indonesia.), PT Indeks, Jakarta.
- Mahendra, A., DKK, 2015, Rancang Bangun Aplikasi Pelayanan Administrasi Perawatan Sepeda Motor Di PT Surapita Unitrans Surabaya, *jurnal*, STMIK STIKOM Surabaya, Surabaya
- Maryadi, L., 2015 Rancang Bangun Sistem Dumping Pada Motor Roda Tiga Secara Mekanis(pembuatan), *Other Thesis*, Politeknik Negeri Sriwijaya.
- Neriswandi, T.D, 2014, Aplikasi Informasi Lokasi Wisata Belanja Batam Berbasis Android, *Tugas Akhir*, Jurusan Teknik Informatika Politeknik Negeri Batam, Batam.7
- Randy, 2008, Diagram UML OOAD, Bandung, Informatika.
- Rosa, A.S, Shalahuddin, M., 2011, Module Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Bandung, Modula.
- Safaat, N. H, 2011, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Penerbit Informatika Bandung, Bandung.
- Sirait, Y., 2016, TIDE CALC – Aplikasi P engolahan Data Pasang Surut Air Laut Berbasis Client Server pada Platform Android, *Tugas Akhir*, Jurusan Teknik Informatika Konsentrasi Geomatika, Politeknik Negeri Batam, Batam

Supardi, 2012, Semua Bisa Menjadi Programmer Android. Jakarta, Gramedia

Anonim, 2016, <http://www.cdn.co.id/kabupaten/keprih1/Barelang.php>, diakses pada tanggal 26 September 2016. Pukul 13.47 WIB