

**APLIKASI PENGELOLAAN KAMAR INDEKOS
BERBASIS WEB DI KAMAR INDEKOS AL-NAJAH CIPARAY**

SKRIPSI

Karya Tulis sebagai Syarat untuk Memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung

Disusun Oleh :

ASRI HARDIYANTI
NPM C1A160023 / 311160023



**PROGRAM STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
BANDUNG
2021**

LEMBAR PERSETUJUAN PEMBIMBING

**APLIKASI PENGELOLAAN KAMAR INDEKOS
BERBASIS WEB DI KAMAR INDEKOS AL-NAJAH
CIPARAY**

Disusun Oleh:

**ASRI HARDIYANTI
NPM C1A160023 / 311160023**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2021

Disetujui oleh:

Pembimbing 1

Pembimbing 2

Yudi Herdiana, S.T., M.T
NIK. 0428027501

Rosmalina, S.T., M.Kom
NIK. 0425038203

LEMBAR PERSETUJUAN PENGUJI

**APLIKASI PENGELOLAAN KAMAR INDEKOS
BERBASIS WEB DI KAMAR INDEKOS AL-NAJAH
CIPARAY**

Disusun Oleh:

**ASRI HARDIYANTI
NPM C1A160023 / 311160023**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2021

Disetujui oleh:

Penguji 1

Penguji 2

Yusuf Muharam, M.Kom
NIK. 04104820003

Sutiyono, S.T, M.Kom
NIK.01043180002

LEMBAR PENGESAHAN LEMBAGA

**APLIKASI PENGELOLAAN KAMAR INDEKOS
BERBASIS WEB DI KAMAR INDEKOS AL-NAJAH
CIPARAY**

Disusun Oleh:

**ASRI HARDIYANTI
NPM C1A160023 / 311160023**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2021

Mengetahui,
Dekan,

Mengesahkan,
Ketua Program Studi,

Yudi Herdiana, S.T., M.T
NIK. 0428027501

Yusuf Muharam, M.Kom
NIK. 04104820003

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : ASRI HARDIYANTI

NIM : C1A160023

Judul Skripsi : APLIKASI PENGELOLAAN KAMAR INDEKOS BERBASIS
WEB DI KAMAR INDEKOS AL-NAJAH CIPARAY

Menyatakan bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik yang berlaku di Fakultas Teknologi Informasi Universitas Bale Bandung. Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Maret 2021

Yang membuat pernyataan,

ASRI HARDIYANTI

NIM. C1A160023 / 311160023

ABSTRACT

Boarding Room Is a Home-based business that provides a place or room for rent to people who need it. Al-Najah Ciparay Boarding Room is one of the boarding houses located in the city of Bandung, to be precise at Kp. Babakan Caringin RT 01 / RW 10 Ciparay Village, Ciparay District. Having been around for more than 20 years, Al-Najah's boarding room still uses manual methods in managing its data. In managing boarding rooms, of course, a good management process is needed to get convenience for both the admin or boarding room manager, as well as service satisfaction for customers or boarding room tenants. With management that is still done manually, of course the boarding house management becomes inefficient and encounters many obstacles, both in terms of promotion, management of admin records, financial management and others. With the discovery of problems and constraints in the boarding room Al-Najah, the author will design a web-based boarding room application using the Waterfall system development method. Broadly speaking, the Waterfall method has steps, namely Requerment Analysis & Specification, Design, Coding and Testing, Integration, Maintenance. The result is a web-based boarding room management application, which can make it easy for prospective tenants to get information about boarding rooms, then the application can also help the admin or boarding room manager in carrying out promotions, data collection, and providing financial reports and others. easily.

Keywords: Management application, Boarding Room, Web-based, Waterfall, Information.

ABSTRAK

Kamar Indekos merupakan sebuah usaha rumahan yang menyediakan tempat atau kamar untuk disewakan kepada orang yang membutuhkannya. Kamar Indekos Al-Najah Ciparay merupakan salah satu usaha Indekos yang berada di kabupaten Bandung, tepatnya di Kp. Babakan Caringin RT 01/RW 10 Desa Ciparay kecamatan Ciparay. Telah berdiri lebih dari 20 tahun, kamar indekos Al-Najah masih menggunakan cara manual dalam pengelolaan data-datanya. Dalam pengelolaan kamar indekos, tentunya diperlukan sebuah proses pengelolaan yang baik untuk mendapatkan kemudahan baik bagi admin atau pengelola kamar indekos, dan memberikan kepuasan pelayanan bagi pelanggan kamar Indekos. Dengan pengelolaan yang masih dilakukan dengan cara manual, tentunya pengelolaan kamar Indekos menjadi tidak efisien dan banyak menemukan kendala, baik dalam hal promosi, pengelolaan pencatatan admin, pengelolaan keuangan dan lain-lain. Dengan ditemukannya masalah-masalah dan kendala di kamar Indekos Al-najah tersebut, penulis akan membuat sebuah aplikasi kamar indekos berbasis web dengan menggunakan metode pengembangan sistem *Waterfall*. Secara garis besarnya, metode *Waterfall* mempunyai Langkah-langkah yaitu *requerment analisys & Specification, Design, Coding and Testing, Integration, Maintenance*. Hasilnya adalah sebuah aplikasi pengelolaan kamar indekos berbasis web, yang dapat memberikan kemudahan bagi para pelanggan dalam mendapatkan informasi mengenai kamar indekos, kemudian aplikasi tersebut juga dapat membantu admin atau pengelola kamar indekos dalam melakukan promosi, pendataan, dan memberikan laporan keuangan dan lain-lain secara mudah.

Kata Kunci : Aplikasi Pengelolaan, Kamar Indekos, berbasis *Web*, *Waterfall*, Informasi.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT yang maha kuasa karena berkat rahmat dan karunianya penulis dapat menyelesaikan penulisan laporan skripsi yang berjudul “APLIKASI PENGELOLAAN KAMAR INDEKOS BERBASIS WEB DI KAMAR INDEKOS AL-NAJAH CIPARAY” ini tepat pada waktunya. Laporan Skripsi ini disusun sebagai salah satu syarat untuk mengerjakan skripsi pada strata 1 Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Universitas Bale Bandung.

Laporan tugas akhir skripsi ini dibuat dengan beberapa bantuan dari berbagai pihak untuk menyelesaikan tantangan dan hambatan selama pengeraannya. Oleh karena itu, pada kesempatan kali ini penyusun ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua dan keluarga besar yang tidak pernah berhenti mencerahkan do'a, dukungan moril maupun materil untuk penulis hingga dapat menyelesaikan penulisan laporan ini.
2. Bapak Yudi Herdiana, S.T., M.T, selaku Dekan Fakultas Teknologi Informasi, Sekaligus pembimbing 1.
3. Bapak Yusuf Muhamram, M.Kom, selaku Ketua Program Studi Teknik Informatika.
4. Ibu Rosmalina, S.T., M.Kom, selaku Dosen Pembimbing Skripsi yang telah membantu dalam pembuatan laporan Skripsi.
5. Seluruh Dosen dan Staff Fakultas Teknologi Informasi Universitas Bale Bandung.
6. Suami, berserta putri tercinta Arsyila Maika Maharani, yang selalu menjadi semangat dan motivasi bagi penulis dalam menyelesaikan penulisan laporan ini.
7. Sahabat-sahabat dan rekan seperjuangan di Fakultas Teknologi Informasi yang selalu membantu dan memberi dukungan selama pengeraaan laporan ini.

Penulis menyadari bahwa masih banyak kekurangan dalam laporan ini, segala kritik dan saran yang membangun akan penulis terima dengan baik. Akhir kata semoga laporan skripsi ini bisa diterima dan bermanfaat bagi kita semua. Aamiin.

Bandung, Juli 2021

Asri Hardiyanti

DAFTAR ISI

| | |
|--|------|
| <i>ABSTRACT</i> | v |
| ABSTRAK | vi |
| KATA PENGANTAR | vii |
| DAFTAR ISI..... | ix |
| DAFTAR TABEL..... | xii |
| DAFTAR GAMBAR | xiii |
| DAFTAR LAMPIRAN..... | xv |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah..... | 3 |
| 1.4 Tujuan Penelitian | 3 |
| 1.5 Metodologi Penelitian | 4 |
| 1.5.1 Metode Pengumpulan Data..... | 4 |
| 1.5.2 Metode Pengembangan Sistem | 5 |
| 1.6 Sistematika Penulisan | 6 |
| BAB II TINJAUAN PUSTAKA..... | 7 |
| 2.1 Landasan Teori..... | 7 |
| 2.2 Dasar Teori..... | 9 |
| 2.2.1 Aplikasi..... | 9 |
| 2.2.2 Pengelolaan | 12 |
| 2.2.3 Kamar Indekos | 13 |
| 2.2.4 Web Browser | 16 |
| 2.2.5 HTML (<i>HyperText Markup Language</i>) | 17 |
| 2.2.6 CSS (<i>Cascading Style Sheet</i>) | 17 |
| 2.2.7 PHP (<i>HyperText Pre Processor/Personal Home Page</i>) | 18 |
| 2.2.8 MySQL | 18 |
| 2.2.9 Bootstrap..... | 18 |
| 2.2.10 JavaScript..... | 19 |
| 2.2.11 Sublime Text..... | 20 |
| 2.2.12 XAMPP..... | 20 |

| | | |
|---------|--|-----|
| 2.2.13 | UML(<i>Unified Modelling Language</i>) | 21 |
| 2.2.14 | Metode <i>Waterfall</i> | 24 |
| 2.2.15 | <i>Black Box</i> | 27 |
| 2.2.16 | <i>Web Hosting</i> | 28 |
| BAB III | METODOLOGI PENELITIAN | 29 |
| 3.1 | Kerangka Pikir | 29 |
| 3.2 | Deskripsi Teori..... | 30 |
| 3.2.1 | Inisialisasi | 30 |
| 3.2.2 | Identifikasi Masalah..... | 30 |
| 3.2.3 | Metodologi Pengumpulan Data | 31 |
| 3.2.4 | Metodologi Pengembangan Sistem | 32 |
| 3.2.5 | Pembuatan Laporan | 34 |
| BAB IV | ANALISIS DAN PERANCANGAN | 35 |
| 4.1 | Analisis..... | 35 |
| 4.1.1 | Analisis Masalah..... | 35 |
| 4.1.2 | Analisis <i>Software</i> | 35 |
| 4.1.3 | Analisis Pengguna..... | 36 |
| 4.1.4 | <i>User Interface</i> | 38 |
| 4.1.5 | Fitur-fitur | 38 |
| 4.1.6 | Analisis Data..... | 39 |
| 4.2 | Perancangan | 40 |
| 4.2.1 | <i>Usecase Diagram</i> | 41 |
| 4.2.2 | <i>Activity Diagram</i> | 57 |
| 4.2.3 | <i>Class Diagram</i> | 68 |
| 4.2.4 | Perancangan Tabel..... | 69 |
| 4.2.5 | <i>User Interface</i> | 72 |
| BAB V | IMPLEMENTASI DAN PENGUJIAN | 84 |
| 5.1 | Implementasi | 84 |
| 5.1.1 | Implementasi Sistem..... | 84 |
| 5.1.2 | Spesifikasi Sistem | 84 |
| 5.1.3 | Implementasi Antar Muka | 85 |
| 5.2 | Pengujian Sistem..... | 102 |
| 5.2.1 | Pengujian form Admin | 103 |
| 5.2.2 | Pengujian Form Pelanggan | 110 |
| BAB VI | KESIMPULAN..... | 115 |

| | |
|----------------------|-----|
| 6.1 Kesimpulan..... | 115 |
| 6.2 Saran..... | 115 |
| DAFTAR PUSTAKA | 116 |
| LAMPIRAN..... | 118 |

DAFTAR TABEL

| | |
|---|-----|
| Tabel 2.1 Simbol <i>Usecase Diagram</i> | 22 |
| Tabel 2.2 Simbol <i>Activity Diagram</i> | 22 |
| Tabel 2.3 Simbol <i>Class Diagram</i> | 23 |
| Tabel 4.1 Analisis Software | 35 |
| Tabel 4.2 Spesifikasi Server Hosting | 36 |
| Tabel 4.3 Tabel Analisis Data..... | 39 |
| Tabel 4.4 Deskripsi Aktor pada <i>Usecase Diagram</i> | 41 |
| Tabel 4.5 Deskripsi <i>Usecase Diagram</i> | 42 |
| Tabel 4.6 <i>Usecase Diagram Login</i> admin | 44 |
| Tabel 4.7 Skenario <i>Usecase Diagram Dashboard</i> admin..... | 45 |
| Tabel 4.8 Skenario <i>Usecase Diagram</i> tambah data admin | 47 |
| Tabel 4.9 Skenario <i>Usecase Diagram</i> edit data admin | 48 |
| Tabel 4.10 Skenario <i>Usecase Diagram</i> hapus data admin..... | 50 |
| Tabel 4.11 Skenario <i>Usecase Diagram</i> cari data | 52 |
| Tabel 4.12 Skenario <i>Usecase Diagram Login</i> pelanggan..... | 53 |
| Tabel 4.13 Skenario <i>Usecase Diagram</i> daftar | 54 |
| Tabel 4.14 Skenario <i>Usecase Diagram</i> pemesanan kamar | 55 |
| Tabel 4.15 Skenario <i>Usecase Diagram Profile</i> | 56 |
| Tabel 4.16 Tabel Admin | 69 |
| Tabel 4.17 Tabel kamar..... | 69 |
| Tabel 4.18 Tabel pemesanan..... | 70 |
| Tabel 4.19 Tabel pelanggan | 70 |
| Tabel 4.20 Tabel pembayaran | 71 |
| Tabel 4.21 Tabel data pelanggan | 71 |
| Tabel 5.1 Spesifikasi perangkat keras minimum | 84 |
| Tabel 5.2 Pengujian <i>Login</i> admin | 103 |
| Tabel 5.3 Pengujian menu admin..... | 103 |
| Tabel 5.4 Pengujian menu kamar..... | 105 |
| Tabel 5.5 Pengujian menu pelanggan | 106 |
| Tabel 5.6 Pengujian menu pemesanan | 107 |
| Tabel 5.7 pengujian menu pembayaran | 108 |
| Tabel 5.8 pengujian menu laporan | 109 |
| Tabel 5.9 Pengujian <i>Logout</i> Admin | 110 |
| Tabel 5.10 Pengujian <i>Login</i> pelanggan | 110 |
| Tabel 5.11 Pengujian pendaftaran pelanggan | 111 |
| Tabel 5.12 Pengujian menu kamar..... | 112 |
| Tabel 5.13 Pengujian menu <i>Profile</i> | 113 |
| Tabel 5.14 Pengujian <i>Logout</i> pelanggan..... | 114 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1.1 Metode pengembangan Sistem..... | 5 |
| Gambar 2.1 Metode <i>Waterfall</i> | 24 |
| Gambar 3.1 Kerangka pikir..... | 29 |
| Gambar 4.1 <i>Usecase Diagram</i> Aplikasi..... | 44 |
| Gambar 4.2 <i>Activity Diagram</i> admin | 57 |
| Gambar 4.3 <i>Activity Diagram Login</i> admin..... | 57 |
| Gambar 4.4 <i>Activity Diagram</i> tambah data admin..... | 58 |
| Gambar 4.5 <i>Activity Diagram</i> edit data admin | 58 |
| Gambar 4.6 <i>Activity Diagram</i> hapus data admin | 59 |
| Gambar 4.7 <i>Activity Diagram</i> tambah data kamar..... | 59 |
| Gambar 4.8 <i>Activity Diagram</i> edit data kamar | 60 |
| Gambar 4.9 <i>Activity Diagram</i> hapus data kamar | 60 |
| Gambar 4.10 <i>Activity Diagram</i> tambah data pemesanan | 61 |
| Gambar 4.11 <i>Activity Diagram</i> ubah status pemesanan | 61 |
| Gambar 4.12 <i>Activity Diagram</i> tambah data pelanggan | 62 |
| Gambar 4.13 <i>Activity Diagram</i> edit data pelanggan | 62 |
| Gambar 4.14 <i>Activity Diagram</i> hapus data pelanggan..... | 63 |
| Gambar 4.15 <i>Activity Diagram</i> tambah data pembayaran | 63 |
| Gambar 4.16 <i>Activity Diagram</i> edit data pembayaran | 64 |
| Gambar 4.17 <i>Activity Diagram</i> hapus data pembayaran..... | 64 |
| Gambar 4.18 <i>Activity Diagram</i> cetak laporan..... | 65 |
| Gambar 4.19 <i>Activity Diagram</i> Cari data..... | 65 |
| Gambar 4.20 <i>Activity Diagram</i> pelanggan..... | 66 |
| Gambar 4.21 <i>Activity Diagram</i> Daftar dan <i>Login</i> pelanggan | 66 |
| Gambar 4.22 <i>Activity Diagram</i> pesan kamar | 67 |
| Gambar 4.23 <i>Activity Diagram</i> cetak detail pemesanan | 67 |
| Gambar 4.24 <i>Activity Diagram</i> cetak riwayat pembayaran | 68 |
| Gambar 4.25 <i>Class Diagram</i> Aplikasi pengelolaan kamar indekos | 68 |
| Gambar 4.26 <i>User Interface</i> <i>Login</i> admin | 72 |
| Gambar 4.27 <i>User Interface</i> <i>Dashboard</i> admin..... | 73 |
| Gambar 4.28 <i>User Interface</i> menu Admin..... | 73 |
| Gambar 4.29 <i>User Interface</i> menu kamar..... | 74 |
| Gambar 4.30 <i>User Interface</i> tambah kamar..... | 74 |
| Gambar 4.31 <i>User Interface</i> pemesanan kamar..... | 75 |
| Gambar 4.32 <i>User Interface</i> tambah Pemesanan | 75 |
| Gambar 4.33 <i>User Interface</i> pelanggan | 76 |
| Gambar 4.34 <i>User Interface</i> tambah pelanggan | 76 |
| Gambar 4.35 <i>User Interface</i> pembayaran | 77 |
| Gambar 4.36 <i>User Interface</i> tambah pembayaran | 77 |
| Gambar 4.37 <i>User Interface</i> Laporan | 78 |
| Gambar 4.38 <i>User Interface</i> halaman awal atau <i>Home</i> | 79 |
| Gambar 4.39 <i>User Interface</i> <i>Login</i> pelanggan..... | 79 |

| | |
|--|-----|
| Gambar 4.40 <i>User Interface</i> pendaftaran pelanggan baru | 80 |
| Gambar 4.41 <i>User Interface</i> kamar..... | 80 |
| Gambar 4.42 <i>User Interface Booking</i> | 81 |
| Gambar 4.43 <i>User Interface</i> Konfirmasi pembayaran..... | 82 |
| Gambar 4.44 <i>User Interface</i> cetak detail pemesanan..... | 82 |
| Gambar 4.45 <i>User Interface Profile</i> | 83 |
| Gambar 5.1 Tampilan Halaman <i>Login Admin</i> | 85 |
| Gambar 5.2 Tampilan halaman <i>Dashboard</i> | 86 |
| Gambar 5.3 Tampilan halaman menu admin | 86 |
| Gambar 5.4 Tampilan halaman tambah data admin..... | 87 |
| Gambar 5.5 Tampilan halaman edit data admin | 87 |
| Gambar 5.6 Tampilan halaman kamar | 88 |
| Gambar 5.7 Tampilan halaman <i>Create</i> kamar | 88 |
| Gambar 5.8 Halaman edit data kamar | 89 |
| Gambar 5.9 Tampilan halaman menu pelanggan..... | 89 |
| Gambar 5.10 Tampilan halaman tambah pelanggan..... | 90 |
| Gambar 5.11 Halaman edit data pelanggan | 90 |
| Gambar 5.12 Tampilan Halaman detail pelanggan..... | 91 |
| Gambar 5.13 Tampilan halaman pemesanan | 91 |
| Gambar 5.14 Tampilan halaman tambah pemesanan | 92 |
| Gambar 5.15 Tampilan halaman ubah status pemesanan | 92 |
| Gambar 5.16 Tampilan halaman pembayaran | 93 |
| Gambar 5.17 Tampilan halaman tambah pembayaran..... | 93 |
| Gambar 5.18 Tampilan halaman edit pembayaran | 94 |
| Gambar 5.19 Tampilan halaman Laporan..... | 94 |
| Gambar 5.20 Tampilan halaman cetak laporan..... | 95 |
| Gambar 5.21 Tampilan pesan saat admin salah <i>Username</i> | 95 |
| Gambar 5.22 Tampilan pesan saat admin salah <i>Password</i> | 96 |
| Gambar 5.23 Tampilan halaman <i>Home</i> | 96 |
| Gambar 5.24 Tampilan halaman <i>Login</i> pelanggan | 97 |
| Gambar 5.25 Tampilan halaman pendaftaran | 97 |
| Gambar 5.26 Tampilan halaman menu kamar | 98 |
| Gambar 5.27 Tampilan halaman detail kamar | 98 |
| Gambar 5.28 Tampilan halaman <i>Booking</i> | 99 |
| Gambar 5.29 Tampilan halaman Detail pemesanan kamar | 99 |
| Gambar 5.30 Tampilan konfirmasi pembayaran di whatsapp | 100 |
| Gambar 5.31 Tampilan halaman detail pemesanan | 101 |
| Gambar 5.32 Tampilan halaman cetak detail pemesanan | 101 |
| Gambar 5.33 Tampilan halaman <i>Profile</i> | 102 |
| Gambar 5.34 Tampilan halaman cetak riwayat pembayaran | 102 |

DAFTAR LAMPIRAN

| | |
|---|-----|
| Lampiran 1 Kuesioner..... | 118 |
| Lampiran 2 Dokumentasi..... | 120 |
| Lampiran 3 Source Code Aplikasi | 123 |
| Lampiran 4 Daftar Riwayat Hidup Penulis | 149 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era modern seperti sekarang ini, masih banyak usaha yang belum menggunakan kemajuan teknologi untuk membantu meningkatkan mutu atau kualitas pelayanan. Oleh karena itu untuk menunjang peningkatan mutu atau kualitas pelayanan tersebut perlu adanya penggantian sistem yang belum menggunakan teknologi tersebut dengan teknologi saat ini. Agar terciptanya pengelolaan perusahaan yang optimal dibutuhkan pengelolaan yang baik. Karena Pengelolaan berdampak pada tercapainya efektifitas dan efisiensi program kerja.

Kamar Indekos Al-Najah Ciparay merupakan salah satu usaha yang bergerak dibidang jasa penyewaan hunian atau tempat tinggal. Berdiri pada tahun 2000 oleh Bapak H.Ama Basri, yang beralamat di Kp. Babakan Caringin RT 01, RW 10 Desa Ciparay, Kecamatan Ciparay, Kabupaten Bandung. Saat ini kamar indekos Al-Najah sudah berjumlah kurang lebih 25 pintu. Kamar Indekos Al-Najah memang terbilang sederhana, rata-rata ukuran satu kamarnya 3x3 meter persegi untuk kamar saja, dengan kamar mandi umum. Harga sewanya terbilang relatif murah, pemilik beralasan agar Kamar Indekos Al-Najah dapat dijangkau oleh semua kalangan, seperti pegawai pabrik, pedagang, bahkan masyarakat menengah kebawah. Meskipun kamar Indekos Al-Najah ini terbilang sederhana, tetapi fasilitas yang diberikan cukup baik, terdapat lahan parkir kendaraan bermotor yang luas, air yang bersih dari PDAM, kantin, dan lokasinya yang berdekatan dengan sarana kesehatan, pasar tradisional, sekolah, kampus, pabrik-pabrik tekstil dan lain-lain.

Lokasi Kamar Indekos Al-Najah yang berdekatan dengan fasilitas-fasilitas umum tersebut, tentunya menjadi peluang usaha yang besar bagi pemilik kamar Indekos, pemilik kamar Indekos tidak ingin menyia-nyiakan kesempatan dalam mempromosikan tempat usahanya untuk di sewa oleh para pelanggan kamar Indekos. Para pelanggan kamar Indekos juga tidak ingin

ambil pusing dalam mencari informasi mengenai kamar indekos. Informasi tentang tempat, lokasi, harga, dan fasilitas tentunya sangat di butuhkan dalam pencarian informasi. Ini merupakan salah satu masalah yang penulis temui saat wawancara dengan pengelola Kamar Indekos Al- Najah, karena kamar Indekos Al-Najah masih melakukan promosi dengan cara manual yaitu melalui pembicaraan dari mulut ke mulut dan menempelkan brosur di tembok-tembok. Pelanggan juga harus mendatangi tempat indekos secara langsung untuk menanyakan fasilitas, harga, dan ketersediaan kamar Indekos. Masalah lain yang penulis temui adalah seluruh pengelolaan data pada kamar Indekos Al-Najah yang masih menggunakan cara manual, yaitu mencatat daftar nama penghuni kamar indekos pada sebuah buku. Karena itu proses pendataan terkadang terhambat karena faktor manual tersebut. Kemudian untuk pembayaran yang masih banyak menemui kendala, terutama uang pembayaran yang terkadang lupa dicatat, sehingga data orang yang sudah membayar dan yang belum menjadi tidak jelas. Laporan keuangan nya pun masih diberikan secara manual oleh pengelola kepada pemilik kamar Indekos dengan memperlihatkan buku catatan pembayaran secara langsung.

Dengan adanya permasalahan-permasalahan yang terjadi di kamar Indekos Al-Najah yang telah diuraikan diatas, maka diperlukan adanya sebuah aplikasi yang dapat memudahkan pengelola dalam melakukan promosi atau memberikan informasi bagi pelanggan kamar Indekos, dan juga dapat mempermudah pengelola dalam proses pendataan, pembayaran, pembuatan laporan keuangan, dan lain-lain. Selain itu, aplikasi ini diharapkan dapat membantu pengelola dalam memberikan pelayanan yang optimal.

Berdasarkan permasalahan diatas maka judul yang akan diambil adalah "Aplikasi pengelolaan kamar indekos berbasis web di kamar indekos Al-Najah Ciparay"

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka rumusan masalah terletak pada :

1. Bagaimana agar pengelola kamar Indekos Al-Najah dapat melakukan promosi dan memberikan Informasi kepada pelanggan Kamar Indekos dengan mudah?
2. Bagaimana membuat aplikasi yang dapat mempermudah pengelola kamar Indekos dalam pendataan dan pencatatan pembayaran?
3. Bagaimana agar pengelola dapat memberikan laporan keuangan kepada pemilik kamar Indekos dengan mudah?

1.3 Batasan Masalah

Adapun batasan masalah yang digunakan untuk menghindari adanya pelebaran pokok masalah agar penelitian ini menjadi lebih terarah dan memudahkan dalam pembahasannya sehingga tujuan penelitian akan tercapai. Ada beberapa batasan masalah dalam penelitian ini yaitu sebagai berikut:

1. Lokasi yang dijadikan obyek penelitian adalah Di kamar Indekos Al-Najah Desa Ciparay.
2. Website ini terdiri dari tiga *User Interface*, yaitu untuk pengelola, pelanggan dan pemilik Kamar Indekos.
3. Aplikasi yang dibuat hanya mengelola tentang promosi, pendataan dan keuangan Kamar Indekos Al-Najah Ciparay saja.
4. Pemrograman di buat menggunakan bahasa pemrograman PHP dan MySQL sebagai *DataBase*.
5. Kamar Indekos hanya bisa disewa perbulan dan harus dibayar penuh (tidak di cicil)

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Memberikan kemudahan untuk pengelola kamar Indekos dalam mempromosikan dan memberikan informasi mengenai kamar Indekos.

2. Membuat sebuah aplikasi yang memudahkan admin untuk mengelola berbagai hal yang berkaitan dengan seluruh pengelolaan data kamar Indekos, termasuk data pelanggan, data kamar, pembayaran, dan lain-lain.
3. Membuat sebuah aplikasi yang menyediakan laporan keuangan untuk diserahkan kepada pemilik kamar Indekos.

1.5 Metodologi Penelitian

1.5.1 Metode Pengumpulan Data

1. Observasi

Observasi dilakukan di Kamar Indekos Al-Najah Ciparay, observasi bertujuan untuk mengamati secara langsung bagaimana sistem pengelolaan kamar Indekos yang sedang berjalan dan mengetahui masalah-masalah apa saja yang ada pada sistem yang sedang digunakan. Kemudian dari masalah-masalah yang ditemukan dapat dianalisa sistem seperti apa yang akan dikembangkan untuk membantu proses pengelolaan kamar Indekos agar lebih mudah dan efisien.

2. Wawancara

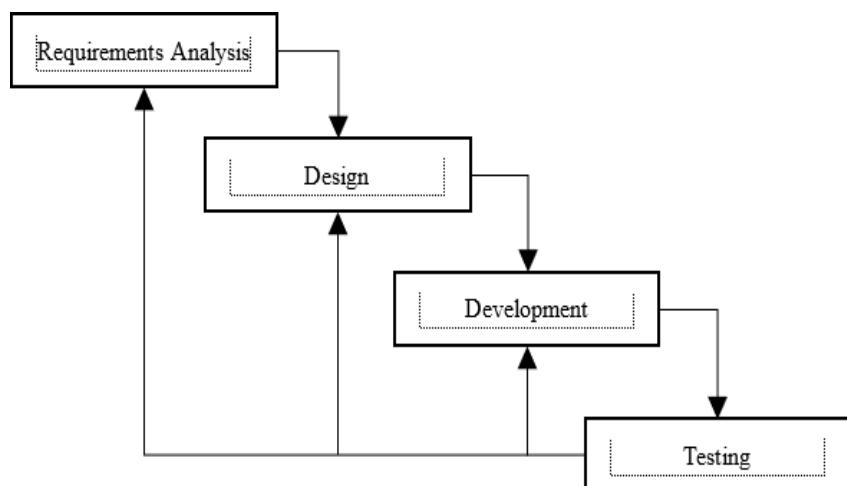
Wawancara dilakukan secara langsung dengan Pengelola kamar Indekos Al-Najah Ciparay yang bertugas dalam proses pengelolaan Kamar Indekos A-Najah. Tujuan dilakukannya wawancara untuk mengetahui secara detail bagaimana alur sistem pengelolaan yang sedang berjalan sebagai data untuk dilakukan pemecahan masalah dengan pengembangan dari sistem yang ada.

3. Studi Pustaka

Pada tahap ini penyusun melakukan pencarian data, membandingkan dan mempelajari penelitian sebelumnya yang berkaitan dengan topik penelitian ini seperti beberapa jurnal maupun buku yang digunakan sebagai referensi seperti yang tercantum di BAB II.

1.5.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada pembuatan aplikasi Pengelolaan Kamar Indekos Al-Najah Ciparay ini adalah metode *Waterfall*. Alasan menggunakan metode ini adalah dikarenakan metode *Waterfall* melakukan pendekatan secara sistematis dan berurutan dalam membangun sebuah sistem. Berikut tahapan metode *Waterfall* (Pressman,2012):



Gambar 1.1 Metode pengembangan Sistem

Berikut ini merupakan tahapan tahapan dalam metode pengembangan sistem *Waterfall* yaitu:

1. *Requirements analysis*, pada proses ini dilakukan untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.
2. *Design*, pada proses ini dilakukan *desain* pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.
3. *Development*, pada tahap ini dilakukan pembuatan kode program, desain ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. *Testing*, pada tahap ini dilakukan pengujian fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

1.6 Sistematika Penulisan

1. BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan tentang latar belakang dilakukannya penelitian serta diuraikan permasalahan dan tujuan yang ingin dicapai dalam proses penelitian.

2. BAB II : TINJAUAN PUSTAKA

Pada bab ini akan di jelaskan landasan teori dari jurnal – jurnal yang berhubungan dengan penelitian, serta di jelaskan dasar teori yang di pakai selama proses penelitian.

3. BAB III : METODOLOGI PENELITIAN

Pada bab ini akan dibahas tentang langkah langkah penelitian yang dilakukan di kamar Indekos Al-Najah Ciparay dalam menindaklanjuti permasalahan yang ada serta tahap - tahap pemecahannya pada penelitian

4. BAB IV : ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini, akan dijelaskan tentang analisis yang telah dilakukan lalu dilanjutkan dengan perancangan sistem yang akan dibuat.

5. BAB V : IMPLEMENTASI DAN PENGUJIAN

Pada bab ini, hasil perancangan yang telah dibuat di bab sebelumnya akan dilanjutkan dengan membangun suatu sistem. Setelah sistem telah dibangun, maka akan dilakukan pengujian untuk mengetahui berhasil atau tidaknya sistem yang telah dibangun.

6. BAB VI : KESIMPULAN DAN SARAN

Pada bagian ini akan disimpulkan tentang seluruh pembahasan dan pemecahan masalah yang telah dilakukan di Program Studi Teknik Informatika serta diberikan saran saran untuk pengembangan sistem berikutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Dalam melakukan penelitian, penulis juga mempelajari referensi dari penelitian yang terdahulu sebagai landasan teori yang melandasi penelitian yang tengah dilakukan. Beberapa landasan teori tersebut yaitu:

2.1.1 "APLIKASI MANAJEMEN KOST PADA KOST KEMBAR" (I Wayan Rendra Adiputra, 2020)

Kost kembar berdiri pada tahun 2009 dan sampai saat ini sudah ada sekitar 40 kamar. Jasa ini tidaklah gratis, yaitu dengan melibatkan sejumlah pembayaran tertentu untuk setiap periode, yang biasanya dihitung per bulan. Proses pembayaran kost akan dapat berjalan dengan lancar apabila faktor-faktor pendukungnya juga memadai. Salah satu faktor yang penting dalam hal ini adalah pembayaran uang kost setiap bulan nya. Proses pembayaran kost masih di lakukan dengan cara menagih uang kost dari kamar satu ke kamar yang lain. Tentu saja hal ini tidak efisien karena akan membutuhkan waktu yang tidak sedikit karena terdapat banyak anak kost yang akan membayar uang kost.

Masalah lain yang timbul adalah keterlambatan pembayaran juga sering terjadi. Jika anak kost terlambat membayar karena tidak ingat tanggal bayar kost. Masalah lain yang terjadi adalah masih ditemukan anak kost yang tidak jujur dalam proses pembayaran uang kost. Uang pembayaran telah diberikan oleh orang tua kepada putra/putrinya namun oleh anak tersebut tidak segera dibayarkan kepada pemilik kost, bahkan digunakan untuk keperluan pribadinya sendiri. Tentu saja hal ini sangat memprihatinkan dan harus segera diatasi. Diharapkan dengan membangun sebuah sistem informasi pembayaran kost dapat memudahkan pemilik dalam mengelola administrasi pembayaran. Selain itu, juga dapat membantu pemilik dalam memberikan pelayanan yang optimal dalam proses pembayaran uang kost.

2.1.2 “PEMBANGUNAN SISTEM PENGELOLAAN MANAJEMEN RUMAH INDEKOS PADA INDEKOS SEMANGGI KOTA MALANG” (Ajie et al., 2020)

Mengatakan, Pengelolaan manajemen rumah indekos yang baik akan memenuhi seluruh kebutuhan rumah indekos. Ketikan melakukan manajemen rumah indekos sering terjadi kesalahan pendataan. Permasalahan tersebut terjadi di rumah indekos semanggi kota Malang, permasalahannya yakni membutuhkan waktu yang lama mencari posisi kolom dan baris yang salah, file mengalami corrupt, keterlambatan untuk memperbarui data, tidak adanya kwitansi ketika melakukan pembayaran sehingga penghuni kamar akan dibebankan untuk membayar lagi pada saat pemilik tidak melakukan pendataan. Dalam penyelesaian untuk mengatasi permasalahan yang ada, pembuatan sistem pengelolaan rumah indekos dengan fitur menambahkan rumah indekos, menambahkan penghuni, notifikasi batas pembayaran, pembayaran indekos, laporan pengeluaran indekos, pemesanan kamar, laporan kerusakan yang terjadi di rumah indekos, cetak bukti pembayaran, lihat data dan status pembayaran, Pada pembangunan sistem pengelolaan manajemen rumah indekos ini menggunakan *Waterfall* model. Pengujian unit mendapatkan hasil dengan nilai 100% valid. Pengujian validasinya dengan demikian juga mendapatkan hasil nilai 100% valid. Pengujian compatibility juga mendapatkan hasil nilai 100% dengan menggunakan sepuluh web browser.

2.1.3 PERANCANGAN SISTEM INFORMASI PENCARIAN DAN PEMESANAN RUMAH KOS BERBASIS WEB (Budiman et al., 2019)

Mengatakan, Rumah kost merupakan sebuah usaha rumahan yang menyediakan tempat atau kamar untuk disewakan kepada yang membutuhkan. Daerah Bandar Lampung ditemukan sekitar 71 (tujuh puluh satu) lebih rumah kost baik yang berupa kamar bedengan dan kamar yang berada di dalam rumah pemilik kost serta variasi harga yang berbeda-beda berdasarkan bentuk dan fasilitas yang ada didalam kamar

kost. Pada saat ini di daerah Bandar Lampung, masih ada masyarakat yang belum memanfaatkan teknologi informasi untuk pencarian dan pemesanan rumah kost. Sistem yang sering dipakai, mereka masih harus bertanya kepada teman atau bertanya langsung kepada pemilik rumah kost untuk membandingkan letak, fasilitas maupun harga. Bagi pemilik rumah kost masih mempromosikan rumah kostnya dengan memasang pamflet atau tulisan “Terima kost” di sebuah papan. Dari permasalahan di atas maka penulis merancang sebuah sistem informasi pencarian dan pemesanan rumah kost berbasis web dengan menggunakan bahasa pemrograman PHP, database MySQL dan dirancang menggunakan Unified Modeling Language (UML). Teknik pengumpulan data dalam penelitian ini menggunakan wawancara (interview), pengamatan (observasi), studi literatur, dan dokumentasi. Hasil pengujian kualitas menggunakan ISO 9126 aspek fungsionalitas yang menunjukkan bahwa sistem melakukan 100% fungsinya dengan benar dan hasil pengujian kebutuhan user menggunakan ISO 9126 aspek usability diperoleh nilai persentase sebesar 88.36%.

2.2 Dasar Teori

2.2.1 Aplikasi

Aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Pengertian Aplikasi (Pratama, 2012) :

1. Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas.
2. Aplikasi adalah sistem lengkap yang mengerjakan tugas spesifik.
3. Aplikasi basis data terdiri atas sekumpulan menu, formulir,

laporan dan program yang memenuhi kebutuhan suatu fungsional unit bisnis / organisasi / instansi.

Aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya. Beberapa aplikasi yang digabung bersama menjadi suatu paket disebut sebagai suatu paket atau *application suite*. Aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. *Software application* adalah *Software* program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu. *Software application* terdiri dari bahasa pemrograman (*programming language*), program aplikasi (*application program*), program paket atau paket aplikasi (*package program*), program utilitas (*utility program*), *games*, *entertainment*, dan lain-lain. Untuk mendukung operasi *Software application* di atas, tugas pengguna komputer dibagi menjadi beberapa bagian yaitu sebagai Analis Sistem, *Programmer*, *Operator*, *Administrator DataBase*, *Administrator Jaringan*, Klasifikasi aplikasi menurut Barry Pratama dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi *Software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu. Macam-macam data yang digunakan untuk membuat aplikasi adalah:
 - a. Data Sumber (*source data*), adalah fakta yang disimpan di dalam basis data, misalnya: nama, tempat lahir, tanggal lahir, dan lain-lain.
 - b. *Meta Data*, digunakan untuk menjelaskan struktur dari basis data, *type* dan format penyimpanan data item dan berbagai pembatas (*constraint*) pada data.
 - c. *Data Dictionary* atau *Data Repository*, digunakan untuk menyimpan informasi katalog skema dan pembatas serta

data lain seperti: pembakuan, deskripsi program aplikasi dan informasi pemakai.

- d. *Overhead Data*, berisi *linked list*, *indeks* dan struktur data lain yang digunakan untuk menyajikan *relationship record*.

Klasifikasi aplikasi dapat digolongkan menjadi beberapa kelas, yaitu:

1. Perangkat Lunak Perusahaan (*Enterprise*)

Perangkat Lunak Perusahaan (*Enterprise Software*) adalah aplikasi yang digunakan perusahaan untuk melakukan pengorganisasian kegiatan perusahaan

2. Perangkat Lunak Infrastruktur Perusahaan

Perangkat Lunak Infrastruktur Perusahaan (*Enterprise Infrastructure Software*) adalah aplikasi yang dibuat untuk menyediakan kemampuan-kemampuan umum yang dibutuhkan untuk membantu perangkat lunak perusahaan (*enterprise Software*)

3. Perangkat Lunak Informasi Kerja

Perangkat Lunak Informasi Kerja (*Information Worker Software*) adalah aplikasi yang biasa dipakai untuk menunjukkan kebutuhan individual untuk membuat dan mengolah informasi. Umumnya untuk tugas-tugas individu dalam sebuah departemen.

4. Perangkat Lunak Media dan Hiburan

Perangkat Lunak Media Dan Hiburan (*Content Access Software*) adalah aplikasi yang biasa digunakan untuk mengakses konten tanpa *editing*, tapi bisa saja termasuk *Software* yang memungkinkan mengedit konten. Seperti *Software* yang menunjukkan kebutuhan individu dan grup untuk mengkonsumsi hiburan digital dan mempublikasikan konten digital.

5. Perangkat Lunak Pendidikan (*Educational Software*)

Perangkat Lunak Pendidikan (*Educational Software*) adalah aplikasi yang hampir sama dengan Perangkat Lunak Media Dan Hiburan (*Content access Software*) tapi biasanya menampilkan konten yang berbeda.

6. Perangkat Lunak Pengembangan media (*Media Development Software*)

Perangkat Lunak Pengembangan media (*Media Development Software*) adalah aplikasi yang digunakan untuk menunjukkan kebutuhan individu untuk menghasilkan media cetak dan elektronik, umumnya pada bidang komersial atau pendidikan.

7. Perangkat Lunak Pengembangan Produk (*Product Engineering Software*)

Perangkat Lunak Pengembangan Produk (*Product Engineering Software*) adalah aplikasi yang biasa digunakan untuk pengembangan produk *Hardware* dan *Software*.

(Pratama, 2012)

2.2.2 Pengelolaan

Pengelolaan sama artinya dengan manajemen, yang mana menurut (Priansa, 2017, p. 2) manajemen berasal dari Bahasa Inggris “*Management*” dengan kata kerja ”*To Manage*” yang secara umum berarti mengurus, mengemudikan, mengelola, membina atau memimpin. Kata benda ”*Management*” dan ”*Manage*” berarti orang yang melakukan kegiatan manajemen atau pengelolaan.

Sedangkan Menurut Choirunnida (Djamarah, 2006 : 174) menjelaskan pengelolaan berasal dari kata kelola yang berarti : ‘Mengendalikan, menyelenggarakan, mengurus, menjalankan yang mendapat imbuhan pe-an menjadi pengelolaan yang artinya mengurus suatu perusahaan dan organisasi dan sebagainya’.

Adapun menurut Trisnawati dan Saefullah (Cairunida, 2009 : 25) mendefinisikan pengelolaan sebagai Seni atau proses dalam menyelesaikan sesuatu yang terkait dengan pencapaian tujuan. Dalam penyelesaian akan sesuatu tersebut, terdapat tiga faktor yang terlibat :

- a. Adanya penggunaan sumber daya organisasi, baik sumber daya manusia maupun faktor-faktor produksi lainnya;

- b. Proses yang bertahap mulai dari perencanaan, pengorganisasian, pengarahan dan pengimplementasian hingga pengendalian dan pengawasan;
- c. Adanya seni dalam penyelesaian kinerja.

Dari uraian di atas dapat dikatakan bahwa pengelolaan merupakan suatu cara atau proses yang dimulai dari perencanaan, pengorganisasian, pengarahan, dan pengawasan sumber daya organisasi untuk mencapai tujuan yang telah ditetapkan.

Adapun fungsi dari pengelolaan menurut Terry (Torang, 2016 : 166) membagi empat fungsi dasar manajemen (pengelolaan), yaitu :

- 1) Perencanaan (*planning*)
- 2) pengorganisasian (*organizing*)
- 3) pengarahan (*actuating*)
- 4) pengawasan (*controlling*)’.

Fungsi-fungsi dasar manajemen tersebut bersifat universal, dimana saja dan dalam organisasi apa saja. Oleh sebab itu, agar manajemen pada organisasi dengan mudah mencapai tujuannya secara efektif, efisien dan rasional maka seorang pimpinan organisasi harus mampu menjalankan fungsi-fungsi dasar manajemen tersebut

2.2.3 Kamar Indekos

Kamar Indekos, menurut KBBI (Lee, 2017) adalah tinggal di rumah orang lain dengan atau tanpa makan (dengan membayar setiap bulan); memondok. Menilik sejarahnya, sebutan ini berasal dari istilah Belanda "in de kost" yang secara harafiah berarti "makan di dalam" atau secara luas "tinggal dan ikut makan" di dalam rumah tempat menumpang tinggal.

Pada zaman kolonial Belanda di Indonesia, "*in de kost*" adalah sebuah gaya hidup yang cukup populer di kalangan menengah ke atas untuk kaum pribumi, terutama sebagian kalangan yang mengagung-agungkan budaya barat / Eropa khususnya adat Belanda, dengan trend ini mereka berharap banyak agar anaknya dapat bersikap dan berprilaku layaknya bangsa Belanda atau Eropa yang dirasa lebih terhormat saat itu.

Dalam masa penjajahan, bangsa Belanda ataupun bangsa Eropa pada umumnya mendapat status sangat terpandang dan memiliki kedudukan tinggi dalam strata sosial di masyarakat, terutama di kalangan masyarakat pribumi Indonesia. Orang-orang yang bukan orang Belanda dan berpandangan non-tradisional menganggap perlunya anak mereka bersikap "seperti layaknya" orang Belanda. Dengan membayar sejumlah uang tertentu sebagai jaminan, anaknya diperbolehkan untuk tinggal di rumah orang Belanda yang mereka inginkan, dengan beberapa syarat yang sudah diperhitungkan, dan resmilah si anak diangkat sebagai anak angkat oleh keluarga Belanda tersebut.

Setelah tinggal serumah dengan keluarga Belanda tersebut, selain diperbolehkan makan dan tidur di rumah tersebut, si anak tetap dapat bersekolah dan belajar menyesuaikan diri dengan gaya hidup keluarga tempat ia menumpang. Dari situasi inilah mungkin sisi paling penting dari konsep "*in de kost*" zaman dulu, yaitu mengadaptasi dan meniru budaya hidup, bukan sekadar hanya makan dan tidur saja, tetapi diharapkan setelah berhenti menumpang, sang anak dapat cukup terdidik untuk mampu hidup mandiri sesuai dengan tradisi keluarga tempat di mana ia pernah tinggal. Hal ini dianggap mirip atau sama dengan konsep "*Home stay*" (bahasa Inggris) pada zaman sekarang.

Seiring berjalannya waktu dan berubahnya zaman, sekarang khalayak umum di Indonesia menyebut istilah "*in de kost*" dengan menyingkatnya menjadi "kost" saja. Di mana-mana, terutama di berbagai daerah di Indonesia, sentra pendidikan tumbuh berjamuran, terutama akademi dan universitas swasta. Hal ini diikuti dengan bertambahnya jumlah rumah-rumah atau bangunan khusus yang menawarkan jasa "kost" bagi para pelajar/mahasiswa yang membutuhkannya. Jasa ini tidaklah gratis, yaitu dengan melibatkan sejumlah pembayaran tertentu untuk setiap periode, yang biasanya dihitung per bulan atau per minggu. Hal ini berbeda dengan kontrak rumah, karena umumnya "kost" hanya menawarkan sebuah kamar untuk

ditinggali. Setelah melakukan transaksi pembayaran barulah seseorang dapat menumpang hidup di tempat yang dia inginkan.

Usaha Kamar Indekos kebanyakan dihuni oleh karyawan atau mahasiswa yang tidak memiliki keluarga atau orang tua di mana mereka melakukan aktivitas pendidikan atau pekerjaan. Tidak sedikit pula para mahasiswa atau karyawan ini yang lebih memilih tinggal di kamar Indekos, meski mereka punya orangtua atau keluarga di kota itu. Alasan paling umum, ingin mandiri atau kamar Indekos lebih dekat dengan kampus atau kantor.

Dari sinilah orang lalu oleh sebagian orang memanfaatkan sebagai peluang usaha tambahan (Dianawati, 2007). Usaha sewa kamar atau yang lebih dikenal dengan istilah rumah kost ini, sekarang merebak di hampir setiap rumah, baik yang berbentuk asrama (putra atau putri) maupun berbentuk kost yang menyatu dengan rumah induknya.

Dilihat dari fasilitasnya, terdapat beberapa ‘kelas’ usaha kamar Indekos. Dari yang paling sederhana, yaitu kamar dengan fasilitas seadanya hingga yang paling mewah, yaitu dengan fasilitas kamar mandi, AC, TV, kulkas dan telepon per kamar. Dilihat dari segi manajemen, hampir semua usaha kost ini bersifat informal, artinya tidak ada standar harga yang seragam.

Harga rumah kost dipengaruhi oleh lokasi terhadap pusat-pusat pertumbuhan, seperti dekat sekolah, kampus, areal pertokoan, pabrik, dan lain-lain. Hal-hal yang diperhatikan dalam usaha rumah kost :

a. Lokasi

Hal pertama untuk menentukan bagus tidaknya prospek usaha rumah kost di masa yang akan datang adalah lokasi. Membeli atau membangun rumah kost di daerah dekat kampus, misalnya. Tentunya dengan harapan agar kelak para mahasiswa akan memerlukan rumah kost untuk disewa. Biasanya harga sewanya adalah antara Rp 200.000,- s/d Rp 500.000,- tergantung dari seberapa lengkap fasilitas yang disediakan (TV, AC, kulkas dan komputer). Selain itu, suasana juga

akan ikut menentukan tarif sewa. Kebanyakan para pelanggan yang jauh dari keluarga cenderung memilih rumah kost yang terasa *homely*.

b. Pembayaran

Sewa Sistem pembayaran untuk rumah kost ini berbeda dengan rumah kontrakan biasa. Disini pada umumnya si pelanggan rumah kost tidak menggunakan kontrak sewa. Si pelanggan harus membayar di muka terlebih dahulu untuk tiga bulan pertama, selanjutnya pembayaran per bulan seperti biasa. Hal ini juga digunakan sebagai bentuk antisipasi dalam hal penagihan uang sewa. Karena ada pun resiko bagi si pemilik rumah adalah adanya kemungkinan penghuni yang ‘bandel’ dan mangkir membayar kewajibannya (Ajen Dianawati, 2006).

2.2.4 Web Browser

Menurut (Sibero, 2011, p. 12) Web browser adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi web. Sumber informasi web diidentifikasi dengan dengan uniform resource identifier (URI) yang dapat terdiri dari halaman web, video, gambar ataupun konten lainnya.

Web browser menampilkan halaman halaman web dengan cara membaca instruksi-sintruksi yang biasa disebut dengan tag-tag HTML (Kawistara, 2017, p. 4) Dari pernyataan diatas dapat disimpulkan bahwa web browser merupakan sebuah perangkat lunak yang digunakan untuk mengakses sumber informasi di dalam web. Beberapa web browser yang banyak digunakan antara lain:

1. Mozilla firefox adalah web browser yang cukup populer digunakan sebagian pengguna internet.
2. Google Chrome adalah web browser yang dirancang oleh Google INC, yang terkenal dengan mesin pencarinya.
3. Microsoft Internet Explorer adalah sebuah web dan *Software* yang tidak bebas namun gratis dari microsoft, dan diikutkan dalam setiap peluncuran sistem operasi Microsoft Windows.

4. Opera adalah browser popular yang mudah digunakan, serta perangkat lunak untuk membaca dan mengirim surat.

2.2.5 HTML (*HyperText Markup Language*)

Menurut (Kawistara, 2017, p. 15) HyperText Markup Language adalah bahasa standard yang digunakan untuk menampilkan halaman web. Yang bisa dilakukan dengan HTML yaitu:

- Mengatur tampilan dari halaman web dan isinya.
- Membuat table dalam halaman web.
- Mempublikasikan halaman web secara online.
- Membuat form yang bisa di gunakan untuk menangani registrasi dan transaksi via web.
- Menambahkan objek-objek seperti citra, audio, video, animasi, java applet dalam halaman web.
- Menampilkan area gambar (canvas) di browser.

2.2.6 CSS (*Cascading Style Sheet*)

Sebuah *website* bisa terdiri dari berpuluhan hingga beratus-ratus halaman. Jika setiap pengguna mengubah halaman *website* tersebut pengguna harus merubah formatnya satu per satu maka hal tersebut akan sangat merepotkan. Namun, jika kita menggunakan CSS maka hal tersebut bukan lagi sebuah masalah karena dengan menggunakan CSS kita dapat menyimpan format dan menggunakannya kapanpun dan dimanapun kita inginkan.

Cascading Style Sheet (CSS) merupakan salah satu bahasa desain *web* (*style sheet language*) yang mengontrol format tampilan sebuah halaman *web* yang ditulis dengan menggunakan penanda *markup language*. Biasanya CSS digunakan untuk membuat desain sebuah halaman HTML atau XHTML. (Kawistara, 2017, p. 55)

2.2.7 PHP (*HyperText Pre Processor/Personal Home Page*)

Menurut (Kawistara, 2017, p. 223) “PHP singkatan dari *Perl Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*.” PHP merupakan *script* yang berintergrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). ”

PHP adalah *script* yang digunakan untuk membuat halaman *web* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. *Mekanisme* ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

PHP adalah bahasa pemograman *script server-side* yang didesain untuk pengembangan *web*. Disebut bahasa pemograman *server-side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemograman *client-side* seperti *JavaScript* yang diproses pada web browser (*client*).

PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirillis dalam lisensi PHP *License*, sedikit berbeda dengan lisensi *GNU General Public License* (GPL) yang biasa digunakan untuk *Open Source*.

2.2.8 MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrograman aplikasi *web*. Kelebihan dari MySQL adalah gratis, handal, selalu *di-update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering dibanding dengan *web server* sehingga proses instalasi nya jadi lebih mudah. (Kawistara, 2017, p. 175)

2.2.9 Bootstrap

Menurut (Kaban & Fajrillah, 2017) Bootstrap merupakan sebuah CSS *framework* yang dikembangkan pengembang *Twitter* pada

pertengahan tahun 2010. Sebelum resmi menjadi *open-source*, Bootstrap dikenal sebagai *Twitter Blueprint*. Hingga saat ini bootstrap sudah dirilis versi 3.3.7 dan telah menjadi salah satu *front-end framework* yang paling populer serta menjadi proyek *open source* di dunia. Bootstrap digambarkan sebagai CSS sederhana namun dibangun dengan *pre-processor* yang menyediakan lebih banyak daya dan fleksibilitas dibanding CSS standar. Bootstrap sudah menyediakan *class-class* CSS dan terintegrasi dengan JQuery. *Responsive layout* pada css bootstrap dengan 12 kolom *grid system* menghasilkan *layout website* yang secara otomatis menyesuaikan dengan lebar browser pengguna. Hal ini yang menyebabkan bootstrap mendukung semua jenis perangkat seperti *smartphone*, tablet, laptop ataupun PC Desktop. Di samping itu, bootstrap juga sudah support untuk HTML 5 dan CSS 3. Banyak fungsi bootstrap yang bisa dipakai untuk sebuah website. Berikut fungsinya:

1. Bisa mempercepat waktu untuk memproses pembuatan front end sebuah website
2. Menampilkan sisi website yang lebih modern dan juga khas anak jaman sekarang
3. Tampilan dari bootstrap sendiri sudah sangat responsive sehingga sangat mendukung untuk segala jenis resolusi, entah itu tablet, smartphone ataupun juga PC dan laptop.
4. Website yang menggunakan bootstrap umumnya lebih ringan karena lebih terstruktur.

2.2.10 JavaScript

Menurut (Abdullah, 2016, p. 3), menyimpulkan bahwa peran *javascript* dalam membuat *website* adalah memberikan efek animasi yang menarik dan interaktivitas dalam penanganan *event* yang dilakukan oleh pengguna *website*.

Menurut Suryana dalam Saifudin & Maharani (2017:11), mengemukakan bahwa “*JavaScript* adalah bahasa *script* berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak

aspek interaksi pemakai pada suatu dokumen *HTML*".

Jadi dapat disimpulkan bahwa *Javascript* merupakan bahasa *script* berdasarkan pada objek yang memberikan efek animasi menarik dan interaktivitas pemakai pada suatu dokumen *HTML*.

2.2.11 Sublime Text

Sublime text merupakan salah satu kode editor yang biasa digunakan oleh para programmer untuk membuat suatu program. Menurut Supono & Putratama (2016:15), "Sublime text merupakan perangkat lunak text editor yang digunakan untuk membuat atau meng-edit suatu aplikasi. Sublime text mempunyai fitur plugin tambahan yang memudahkan programmer".

Selain itu, menurut (Faridl, 2015, p. 1) menjelaskan bahwa "Sublime Text 3 adalah editor berbasis python, sebuah teks editor yang elegan, kaya akan fitur, cross platform, mudah dan simple yang cukup terkenal di kalangan developer (pengembang), penulis dan desainer".

Jadi dapat disimpulkan bahwa sublime text ialah teks editor yang digunakan untuk membuat program aplikasi yang secara otomatis untuk mempermudah progremer dalam mengetikkan kode editor.

2.2.12 XAMPP

Menurut Wahana dalam Agus & Safitri (2015:1) mengemukakan bahwa "Xampp adalah salah satu paket instalasi *Apache*, *PHP*, dan *MySQL* secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut".

Menurut Wahana dalam Prayitno (2015:2), mengemukakan bahwa "XAMPP adalah salah satu paket instalasi *apache*, *PHP*, dan *MySQL* secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut". *XAMPP* juga dapat disebut sebuah *CPanel* server *virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

Jadi dapat disimpulkan bahwa *XAMPP* ialah salah satu paket instalasi *Apache*, PHP, dan MySQL berbasis *open source* yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP. (Wibawanto, 2018)

2.2.13 UML(*Unified Modelling Language*)

Unified Modelling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (M.Shalahuddin, 2018)

1. Tujuan UML

Tujuan dari UML adalah :

- a. Memberikan model yang siap pakai, bahasa permodelan visual yang ekspresif untuk mengembangkan model dan dimengerti secara umum.
- b. Memberikan bahasa permodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- c. Menyatukan praktek-praktek yang terdapat dalam permodelan.

2. Macam-macam diagram dalam UML

Ada beberapa diagram dalam UML (*Unified Modelling Language*) antara lain :

a. *Usecase Diagram*

Usecase Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Usecase* merepresentasikan sebuah interaksi antara aktor dengan sistem. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Adapun simbol dari *Usecase Diagram* antara lain:

Tabel 2.1 Simbol *Usecase Diagram*

| No | Simbol | Nama | DESKRIPSI |
|----|--------|-----------------------|---|
| 1 | | <i>Actor</i> | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>Usecase</i> . |
| 2 | | <i>Usecase</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> . |
| 3 | | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 4 | | <i>Include</i> | Menspesifikasikan bahwa <i>Usecase</i> sumber secara eksplisit. |
| 5 | | <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>). |
| 6 | | <i>Extend</i> | Menspesifikasikan bahwa <i>Usecase</i> target memperluas perilaku dari <i>Usecase</i> sumber pada suatu titik yang diberikan |

b. *Activity Diagram*

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, decicion yang mungkin terjadi, dan bagaimana mereka berakhir. Adapun simbol dari *Activity Diagram* antara lain :

Tabel 2.2 Simbol *Activity Diagram*

| NO | SIMBOL | NAMA | DESKRIPSI |
|----|--------|-------------|--|
| 1 | | Status Awal | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 2 | | Aktifitas | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang |

| | | | |
|---|---|--------------|--|
| | | | menghasilkan suatu hasil yang terukur bagi suatu Actor. |
| 3 |  | Percabangan | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4 |  | penggabungan | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5 |  | Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

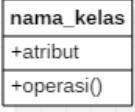
c. Class Diagram

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

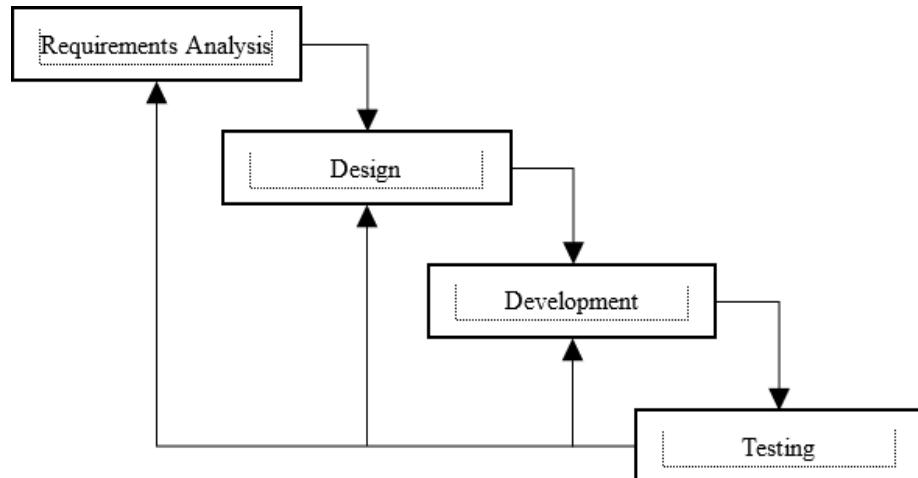
Diagram kelas dibuat agar pembuat program membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Berikut adalah simbol-simbol yang dipakai dalam membuat kelas diagram:

Tabel 2.3 Simbol *Class Diagram*

| NO | SIMBOL | NAMA | DESKRIPSI |
|----|---|-------|-------------------------|
| 1 |  | Kelas | Kelas pada suatu sistem |

| | | | |
|---|--|------------------|--|
| 2 | | Antar Muka | Menyerupai Konsep <i>Interface</i> dalam pemrograman berorientasi Objek. |
| 3 | | Asosiasi | Yang menggabungkan objek satu dengan Objek lainnya |
| 4 | | Asosiasi Berarah | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
| 5 | | Generalisasi | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
| 6 | | Kebergantungan | Kebergantungan antar kelas |
| 7 | | Agregasi | Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>) |

2.2.14 Metode Waterfall



Gambar 2.1 Metode Waterfall

Menurut Pressman (2015:42), Metode *Waterfall* adalah metode klasik yang memiliki sifat sistematis, berurutan dalam membangun *Software*. Nama model ini sebenarnya adalah “Linear Sequential Model”. Model ini sering disebut juga dengan “classic life cycle” atau metode *Waterfall*. Model ini termasuk ke dalam model generic pada rekayasa

perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Fase-fase dalam metode *Waterfall* Adalah sebagai berikut :

a. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *Software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

b. *Planning (Estimating, Scheduling, Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resikoresiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem.

c. *Modeling (Analysis & Design)*

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *Software*, tampilan interface, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

d. *Construction (Code & Test)*

Tahapan Construction ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca

oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

e. *Deployment (Delivery, Support, Feedback)*

Tahapan Deployment merupakan tahapan implementasi *Software* ke customer, pemeliharaan *Software* secara berkala, perbaikan *Software*, evaluasi *Software*, dan pengembangan *Software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015)

Keuntungan menggunakan metode *Waterfall* adalah prosesnya lebih terstruktur, hal ini membuat kualitas *Software* baik dan tetap terjaga. Dari sisi user juga lebih menguntungkan, karena dapat merencanakan dan menyiapkan kebutuhan data dan proses yang diperlukan sejak awal. Penjadwalan juga menjadi lebih menentu, karena jadwal setiap proses dapat ditentukan secara pasti. Sehingga dapat dilihat jelas target penyelesaian pengembangan program. Dengan adanya urutan yang pasti, dapat dilihat pula perkembangan untuk setiap tahap secara pasti. Dari sisi lain, model ini merupakan jenis model yang bersifat dokumen lengkap sehingga proses pemeliharaan dapat dilakukan dengan mudah.

Kelemahan menggunakan metode *Waterfall* adalah bersifat kaku, sehingga sulit melakukan perubahan di tengah proses. Jika terdapat kekurangan proses/prosedur dari tahap sebelumnya, maka tahapan pengembangan harus dilakukan mulai dari awal lagi. Hal ini akan memakan waktu yang lebih lama. Karena jika proses sebelumnya belum selesai sampai akhir, maka proses selanjutnya juga tidak dapat berjalan. Oleh karena itu, jika terdapat kekurangan dalam permintaan user maka proses pengembangan harus dimulai kembali dari awal. Karena itu, dapat dikatakan proses pengembangan *Software* dengan metode *Waterfall* bersifat lambat.

Kelemahan lainnya menggunakan metode *Waterfall* adalah membutuhkan daftar kebutuhan yang lengkap sejak awal. Tetapi, biasanya jarang sekali customer yang dapat memenuhi itu. Untuk menghindari pengulangan tahap dari awal, user harus memberikan seluruh prosedur, data, dan laporan yang diinginkan mulai dari tahap awal pengembangan. Tetapi pada banyak kondisi, user sering melakukan permintaan di tahap pertengahan pengembangan sistem. Dengan metode ini, maka development harus dilakukan mulai lagi dari tahap awal. Karena development disesuaikan dengan desain hasil user pada saat tahap pengembangan awal. Di sisi lain, user tidak dapat mencoba sistem sebelum sistem benar-benar selesai. Selain itu, kinerja personil menjadi kurang optimal karena terdapat proses menunggu suatu tahap selesai terlebih dahulu. Oleh karena itu, seringkali diperlukan personil yang “multi-skilled” sehingga minimal dapat membantu pengerjaan untuk tahapan berikutnya. (Pressman, 2015:42-43)

2.2.15 Black Box

(Perry, 1995, pp. 1–3) Mengemukakan bahwa Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *Black Box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan. Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses *DataBase* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

2.2.16 Web Hosting

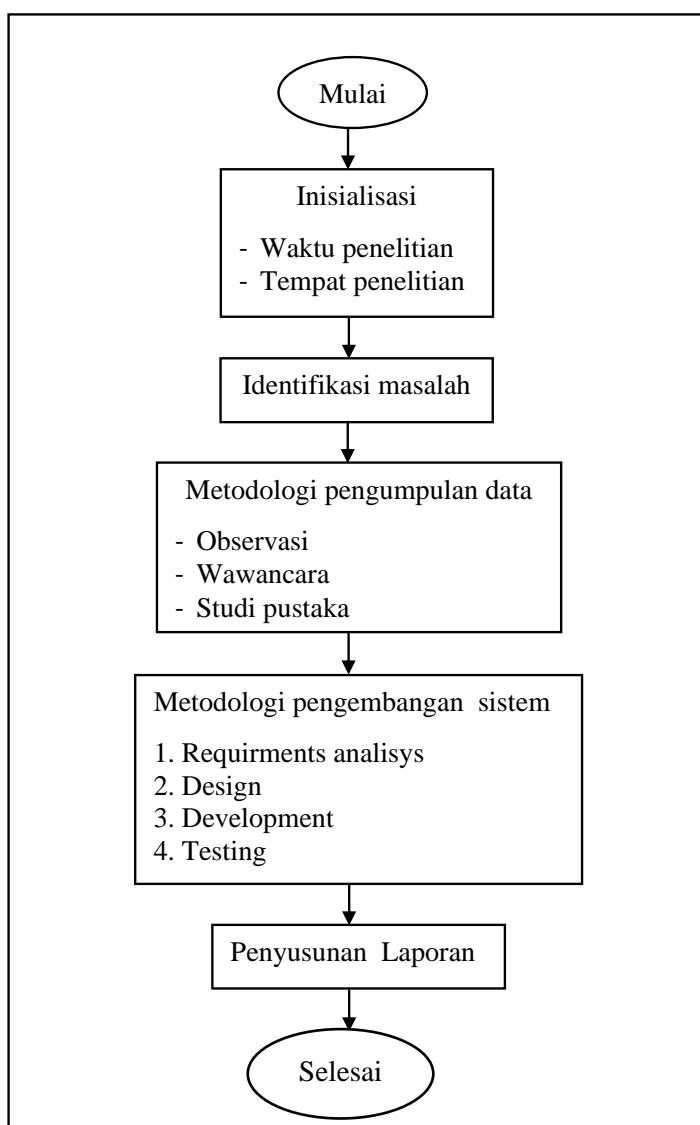
Layanan *web hosting* adalah jenis layanan *hosting* internet yang memungkinkan individu dan organisasi untuk membuat situs *web* mereka dapat diakses melalui *World Wide Web*. *Host web* adalah perusahaan yang menyediakan ruang pada *server* yang dimiliki atau disewa untuk digunakan oleh klien, serta menyediakan konektivitas internet, biasanya di pusat data. *Host web* juga dapat menyediakan ruang pusat data dan konektivitas ke internet untuk *server* lain yang terletak di pusat data mereka, yang disebut *colocation*.

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Pikir

Kerangka pikir merupakan langkah-langkah yang akan di lakukan dalam penyelesaian masalah penelitian ini. Dalam penelitian ini, penulis melakukan tahapan sesuai dengan rencana kegiatan yang telah dibuat untuk membangun Aplikasi Pengelolaan kamar Indekos Al-Najah Ciparay, dibutuhkan beberapa tahapan yang dilakukan peneliti, untuk penjelasan sebagai berikut :



Gambar 3.1 Kerangka pikir

3.2 Deskripsi Teori

3.2.1 Inisialisasi

Pada tahap ini dilakukan penentuan tempat dan waktu penelitian. Setelah waktu dan tempat ditentukan selanjutnya dilakukan proses pengumpulan data dan identifikasi masalah untuk dijadikan bahan serta judul penelitian

1. Tempat Penelitian

Tempat Penelitian dilaksanakan yaitu di Kamar Indekos Al-Najah yang beralamatkan di Kp. Babakan Caringin RT01/RW10 Desa Ciparay, Kecamatan Ciparay Kabupaten Bandung.

2. Waktu Penelitian

Penelitian Mengenai pembuatan Aplikasi Pengelolaan Kamar Indekos Al-Najah Ciparay ini dilakukan selama 5 bulan, dari mulai bulan maret sampai dengan bulan juli 2021.

3.2.2 Identifikasi Masalah

Pada tahap ini penyusun melakukan identifikasi masalah sebagai upaya mendefinisikan masalah dan membuat definisi tersebut dapat diukur sebagai langkah awal penelitian. Pada tahap ini ditemukan beberapa masalah yang terjadi di Kamar Indekos Al-Najah Ciparay Diantaranya:

1. Belum maksimal dan efektifnya promosi yang dilakukan pengelola kamar Indekos.
2. Belum tersedianya Informasi mengenai kamar Indekos Al-Najah bagi para pencari kamar Indekos, sehingga para pencari kamar Indekos harus mendatangi tempat Indekos secara langsung.
3. Belum tersedianya pemesanan kamar Indekos secara Online.
4. Pencatatan dan pembukuan mengenai seluruh pengelolaan kamar Indekos (data-data pelanggan, pembayaran, laporan, dll) yang masih dilakukan secara manual, sehingga

pengelolaan kamar Indekos banyak kendala karena tidak efektifnya pengelolaan kost.

3.2.3 Metodologi Pengumpulan Data

Dalam pelaksanaan penelitian ini diperlukan data dan informasi terkait yang nantinya akan digunakan sebagai bahan rujukan untuk proses penyelesaian masalah maupun dalam pengembangan sistem. Adapun metodologi pengumpulan data yang digunakan dalam penelitian ini adalah observasi, wawancara dan studi pustaka.

1. Observasi

Observasi dilakukan di Kamar Indekos Al-Najah Ciparay, observasi bertujuan untuk mengamati secara langsung bagaimana sistem pengelolaan kamar Indekos yang sedang berjalan dan mengetahui masalah-masalah apa saja yang ada pada sistem yang sedang digunakan. Kemudian dari masalah-masalah yang ditemukan dapat dianalisa sistem seperti apa yang akan dikembangkan untuk membantu proses pengelolaan kamar Indekos agar lebih mudah dan efisien.

2. Wawancara

Wawancara dilakukan secara langsung dengan Pengelola kamar Indekos Al-Najah Ciparay yang bertugas dalam proses pengelolaan Kamar Indekos A-Najah. Tujuan dilakukannya wawancara untuk mengetahui secara detail bagaimana alur sistem pengelolaan yang sedang berjalan sebagai data untuk dilakukan pemecahan masalah dengan pengembangan dari sistem yang ada.

3. Studi Pustaka

Pada tahap ini penyusun melakukan pencarian data, membandingkan dan mempelajari penelitian sebelumnya yang berkaitan dengan topik penelitian ini seperti beberapa jurnal maupun buku yang digunakan sebagai referensi seperti yang tercantum di BAB II.

3.2.4 Metodologi Pengembangan Sistem

Metodologi penelitian ini digunakan sebagai pedoman peneliti dalam pelaksanaan penelitian ini agar hasil yang dicapai tidak menyimpang dari tujuan yang telah ditentukan sebelumnya. Seperti yang telah diuraikan sebelumnya, penulis menggunakan metode *Waterfall* sebagai metodologi pengembangan sistem dalam penelitian ini, dengan Langkah-langkah sebagai berikut:

1. *Requirement Analysis*

Berikut adalah analisis kebutuhan yang diperlukan untuk menjawab permasalahan yang ada, yaitu :

a. Kebutuhan Fungsional

Kebutuhan fungsional meliputi sistem yang berfungsi mengelola promosi, pendataan, dan laporan keuangan. Maka dari itu dibuat sistem yang mendukung aktifitas tersebut diantaranya :

- 1) Mengelola data mengenai promosi dan informasi mengenai kamar indekos yang dapat diakses oleh pelanggan.
- 2) Mengelola pendataan yang meliputi, data kamar, data pembayaran, data pelanggan dan lain-lain.
- 3) Menghasilkan laporan keuangan bagi pemilik kamar indekos

b. Kebutuhan Non fungsional

1) Analisis Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras yang digunakan untuk membuat aplikasi pengelolaan kamar indekos berbasis web tersebut adalah seperangkat komputer dengan spesifikasi sebagai berikut:

1. Laptop Lenovo™ Yoga™ 510
2. Processor Intel® core™ i5 7200U CPU 2.50GHz 2.70 GHz 7th Generation.
3. Hardisk berkapasitas 700 GB
4. RAM berkapasitas 4.00 GB

5. Video Grapich AMD RADEON Graphich
- 2) Analisis Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk membangun Aplikasi pengelolaan kamar indekos berbasis *Web* tersebut adalah sebagai berikut:

1. Sistem operasi Windows 10 *Home Single Language*
2. Sublime Text 3
3. Xampp
4. Web browser Microsoft Edge, Google Chrome, dll.
5. Balsamiq Mockups 3
6. Umlet

2. *Design*

Setelah selesai dengan pengumpulan data dan analisis kebutuhan selanjutnya dilakukan tahap desain. Proses *Design* atau desain sistem meliputi proses gambaran alur kerja sistem, basis data (*DataBase*), dan tampilan (*User Interface*).

a. Perancangan sistem

Perancangan sistem dilakukan bertujuan untuk mendapatkan gambaran jelas tentang alur aplikasi seperti apa yang akan dibuat dan dikerjakan yang akan membantu dalam proses pembuatan aplikasi. Perancangan sistem dilakukan dengan menggunakan pemodelan *Unified Modelling Language* (UML).

b. Perancangan Basis Data

Perancangan Basis Data harus dilakukan untuk mendapatkan kumpulan data yang saling terhubung. Perancangan basis data harus dilakukan dikarenakan akan menjadi tempat penyimpanan data dari sebuah sistem. Perancangan Basis Data yang akan digunakan dalam penelitian ini adalah Tabel data Admin yang berfungsi untuk menyimpan seluruh data yang akan di Kelola admin, dan Tabel data pelanggan yang berfungsi untuk menyimpan data diri dan pemesanan.

c. Perancangan *User Interface*

Perancangan *User Interface* dilakukan penyusun bertujuan untuk menggambarkan tampilan yang akan dibuat agar mempermudah pada tahap implementasi program. *User Interface* dibuat sesederhana mungkin untuk memudahkan pengguna dalam mengakses aplikasi yang akan dibuat nantinya.

3. *Development*

Pada tahap *Coding* atau implementasi desain, perancangan yang telah dibuat di terjemahkan kedalam bentuk pengkodean secara nyata ke dalam bahasa pemrograman. Pada aplikasi ini implementasi kode dibuat ke dalam bahasa pemrograman PHP dan MySQL sebagai *DataBase*. Implementasi kode untuk membentuk fungsi-fungsi yang dibutuhkan oleh program dibuat menggunakan bantuan *framework* bootstrap agar sesuai dengan keinginan pengguna dan implementasi dapat berjalan cepat dan mudah. Tahap ini dibuat sesuai dengan desain yang telah dibuat sebelumnya.

4. *Testing*

Pengujian atau *testing* dilakukan dengan metode *Black Box* setelah aplikasi selesai dibuat dan memenuhi kebutuhan-kebutuhan sesuai desain. Pengujian dilakukan untuk memastikan aplikasi berjalan dengan baik dan sesuai dengan sistem yang ada.

3.2.5 Pembuatan Laporan

Tahapan pembuatan laporan sebagai salah satu persyaratan kelulusan di Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung. Pembuatan laporan ini terdiri dari hasil identifikasi masalah dan implementasi dalam bentuk laporan tertulis yang terdiri dari 6 bab dengan sistematika penulisan yang tertera pada BAB I.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

4.1.1 Analisis Masalah

Sistem pengelolaan data yang sedang berjalan di Kamar Indekos Al-Najah saat ini masih menggunakan cara konvensional, baik dalam hal Promosi, pencatatan keuangan dan laporannya masih dicatat secara manual di buku tulis. Pencatatan yang masih dilakukan dengan cara mencatat dibuku tulis tentu saja tidak efisien, banyak kendala dan masalah yang ditemui dalam pengelolaan kamar indekos Al-Najah, salah satunya adalah kurangnya kegiatan promosi, sehingga pelanggan kurang mendapatkan informasi mengenai kamar indekos, kemudian dalam hal pencatatan keuangan atau pembayaran juga masih dilakukan dengan cara manual sehingga sering terjadi kesalahan dalam pembayan, tagihan, dan lain-lain. Juga dalam penyimpanan berkas atau data-data pelanggan yang masih diberikan secara langsung kepada pengelola, sehingga data-data tersebut sering sekali hilang, begitu juga dalam hal penyetoran laporan keuangan perbulan untuk pemilik kamar Indekos yang belum efektif.

4.1.2 Analisis *Software*

Pada pembuatan Aplikasi pengelolaan kamar indekos berbasis web ini, penulis menggunakan beberapa perangkat lunak atau *Software* dengan spesifikasi sebagai berikut :

Tabel 4.1 Analisis *Software*

| NO | <i>Software</i> | Fungsi |
|----|--|------------------------|
| 1 | Windows 10 <i>Home Single Language</i> | Sistem Operasi |
| 2 | Sublime Text 3 | Teks Editor |
| 3 | Xampp (PHP 7.4) | Untuk membuat Aplikasi |

| | | |
|---|-------------------------------|---|
| 4 | Php myadmin | DBMS |
| 5 | Microsoft Edge, Google Chrome | Web Browser |
| 6 | Balsamiq Mockup 3 | Untuk membuat rancangan <i>User Interface</i> |
| 7 | Umlet | Untuk membuat UML diagram |
| 8 | Microsoft Word 2016 | Untuk membuat laporan |

Tabel 4.2 Spesifikasi Server Hosting

| No | Spesifikasi | Deskripsi |
|----|----------------|-----------|
| 1 | Core | 1 |
| 2 | RAM | 768 MB |
| 3 | SSD | 2 GB |
| 4 | OS | Linux |
| 5 | Apache Version | 2.4.43 |
| 6 | PHP Version | 7.3.17 |

4.1.3 Analisis Pengguna

Aplikasi yang dibuat akan membatasi layanan-layanan yang disediakan berdasarkan hak akses yang diberikan kepada pengguna dalam sesuai keperluannya. Secara umum, pengguna memiliki hak untuk, *Login*, melihat dan mengubah *profil* pribadinya, mengubah *Password*, dan *Logout*. Berdasarkan hak akses, pengguna yang akan menggunakan aplikasi pengelolaan kamar indekos berbasis web ini dibagi menjadi 3(tiga) pengguna yaitu :

a. Administrator

Administrator adalah pengelola kamar indekos yang diberikan kewenangan untuk melakukan pengelolaan aplikasi. Administrator dapat melihat dan mengubah semua data sistem, dan dapat membenahi kesalahan yang terjadi dalam sistem. Semua layanan yang ada pada sistem dapat digunakan oleh administrator dengan tujuan agar sistem bisa dimonitor secara penuh oleh administrator untuk perbaikan-

perbaikan jika terjadi kesalahan yang dilakukan oleh pengguna baik secara disengaja maupun tidak disengaja. Adapun hal-hal yang dapat dilakukan seorang Administrator dalam sistem adalah sebagai berikut:

1. *Login*
 2. Mengelola data admin
 3. Mengelola data kamar
 4. mengelola data pemesanan
 5. mengelola data pelanggan
 6. mengelola data pembayaran
 7. Mencetak Laporan
- b. Pelanggan
- Adapun hal-hal yang dapat dilakukan seorang Member dalam sistem adalah sebagai berikut:
1. Melakukan pendaftaran
 2. *Login*
 3. Melakukan pemesanan
 4. Mencetak detail pemesanan
 5. Melihat data diri atau profil
 6. Melihat dan mencetak Riwayat pembayaran
- c. Pemilik Kamar Indekos

Pemilik kamar indekos disini hanya mendapatkan *print out* laporan keuangan setiap bulan nya dari pengelola kamar indekos yang memuat beberapa informasi antaralain :

1. Nama pelanggan
2. Kamar yang di sewa
3. Tanggal sewa
4. Nominal atau uang yang sudah dibayarkan
5. Status lunas.
6. Jumlah Total pendapatan

4.1.4 User Interface

Pada dasarnya, fokus dari Interaksi Manusia dan Komputer adalah perancangan dan evaluasi antarmuka pemakai (*User Interface*) pada suatu sistem. *User Interface* dapat diartikan sebagai bagian sistem komputer yang memungkinkan manusia berinteraksi dengan komputer secara langsung. *User Interface* dari aplikasi pengelolaan kamar indekos berbasis web ini akan dibuat *user friendly*, yaitu dengan membuat tampilan yang sederhana, sehingga aplikasi dapat dengan mudah dipelajari, mudah dioperasikan dan tetap dapat memberikan informasi mengenai kamar indekos Al-Najah kepada penggunanya.

4.1.5 Fitur-fitur

Fitur-fitur yang akan di buat dalam aplikasi ini akan memudahkan admin atau pengelola kamar indekos dalam melakukan seluruh pengelolaan data kamar indekos dan memudahkan pelanggan dalam mendapatkan informasi mengenai kamar indekos. Berikut ini adalah fitur-fitur yang terdapat pada aplikasi tersebut :

1. Pendaftaran atau buat akun, dapat dilakukan oleh pelanggan untuk bisa *Login* mengakses informasi detail mengenai informasi kamar indekos
2. Fitur *Booking*, yang dapat digunakan pelanggan untuk melakukan pemesanan secara *online* yang nantinya akan diarahkan untuk terhubung langsung dengan pengelola melalui *chat Whatsapp*.
3. Cetak Riwayat pembayaran
4. Cetak detail pemesanan
5. Fitur *Dashboard*, untuk melihat data jumlah pelanggan, jumlah pemesanan, total pemasukan dan lain-lain.
6. Pengelolaan data admin, untuk mencari, melihat, menambah, mengedit dan menghapus data admin

7. Pengelolaan data kamar, untuk mencari, menambahkan kamar baru, upload foto kamar, menghapus, atau mengedit data-data kamar indekos.
8. Pengelolaan data pemesanan, untuk mencari, menambahkan, dan mengubah status pembayaran.
9. Pengelolaan data Pelanggan, untuk mencari, melihat detail, menambahkan, menghapus atau mengedit data pelanggan yang tidak bisa melakukan pendaftaran secara mandiri.
10. Pengelolaan data pembayaran, untuk mencari, menambahkan, menghapus dan mengedit data pembayaran
11. Cetak laporan, untuk mencetak laporan bulanan yang akan diberikan kepada pemilik kamar indekos.

4.1.6 Analisis Data

Berikut ini merupakan analisis data berupa data masukan (*input*), proses, dan keluaran (*Output*) untuk penunjang bagi aplikasi pengelolaan kamar indekos yang akan dibuat :

Tabel 4.3 Tabel Analisis Data

| No | Masukan | Proses | Keluaran |
|----|----------------------------|--|---------------------------------------|
| 1 | <i>Login</i> | <ul style="list-style-type: none"> • Memeriksa Status <i>Login</i> • Validasi | Informasi <i>Login</i> |
| 2 | Pendaftaran pelanggan baru | <ul style="list-style-type: none"> • Memeriksa data pelanggan • Validasi • Mengurutkan data pelanggan | Informasi profil/ data diri pelanggan |
| 3 | Data Admin | <ul style="list-style-type: none"> • Memeriksa data Admin • Validasi | Informasi data admin |

| | | | |
|---|--------------------|--|-------------------------------------|
| | | <ul style="list-style-type: none"> • Mengurutkan data admin | |
| 3 | Data kamar indekos | <ul style="list-style-type: none"> • Memeriksa data kamar • Validasi • Mengurutkan data kamar | Informasi data kamar indekos |
| 4 | Data pemesanan | <ul style="list-style-type: none"> • Memeriksa data pemesanan • Validasi • Mengurutkan data pemesanan | Informasi data transaksi pemesanan |
| 5 | Data pelanggan | <ul style="list-style-type: none"> • Memeriksa data pelanggan • Validasi • Mengurutkan data pelanggan | Informasi data pelanggan |
| 6 | Data pembayaran | <ul style="list-style-type: none"> • Memeriksa data pembayaran • Validasi • Mengurutkan data pembayaran | Informasi data transaksi pembayaran |
| 7 | Cetak Laporan | <ul style="list-style-type: none"> • Memeriksa data Laporan • Mencetak Laporan | Informasi laporan perbulan |

4.2 Perancangan

Perancangan Aplikasi pengelolaan kamar indekos Al-najah dibuat dengan menggunakan UML (*unified modelling language*) sebagai Bahasa pemodelan. Diagram UML yang digunakan adalah *UseCase Diagram*, *Activity Diagram*, dan *Class Diagram*.

4.2.1 Usecase Diagram

Usecase Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *Usecase* merepresentasikan sebuah interaksi antara aktor dengan sistem.

1. Aktor

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu Berikut ini merupakan deskripsi setiap actor pada *Usecase Diagram* tersebut :

Tabel 4.4 Deskripsi Aktor pada *Usecase Diagram*

| no | aktor | deskripsi |
|----|-----------|--|
| 1 | Admin | 1. <i>Login</i> 2. Lihat tampilan <i>Dashboard</i> 3. Mencari, menambah, menghapus , mengedit data admin 4. Mencari, menambah, menghapus , mengedit data dan upload foto kamar 5. Mencari, menambah dan mengubah status pemesanan 6. Mencari, menambah, melihat detail,menghapus, mengedit data pelanggan 7. Mencari, menambah, menghapus, mengedit data pembayaran 8. Mencari dan Mencetak Laporan |
| 2 | pelanggan | 1. Melakukan pendaftaran 2. Melakukan <i>Login</i> 3. Melihat tampilan <i>Home</i> 4. Melihat informasi kamar secara detail 5. Melakukan pemesanan secara online 6. Melakukan Konfirmasi pembayaran melalui Whatsapp |

| | | |
|---|---------|---|
| | | <p>7. Melihat status pemesanan/<i>Booking</i> 8. Melihat Profil atau data diri 9. Mencetak Riwayat pembayaran 10. Mencetak detail pemesanan</p> |
| 3 | Pemilik | <p>Menerima laporan dari pengelola yang berisi tentang :</p> <p>1. laporan keuangan atau total pemasukan 2. Nama pelanggan 3. Kamar yang disewa 4. Tanggal penyewaan 5. Status pembayaran</p> |

2. Deskripsi *Usecase Diagram*

Berikut merupakan deskripsi dari *Usecase Diagram* yang menjelaskan mengenai proses-proses yang dilakukan dalam aplikasi:

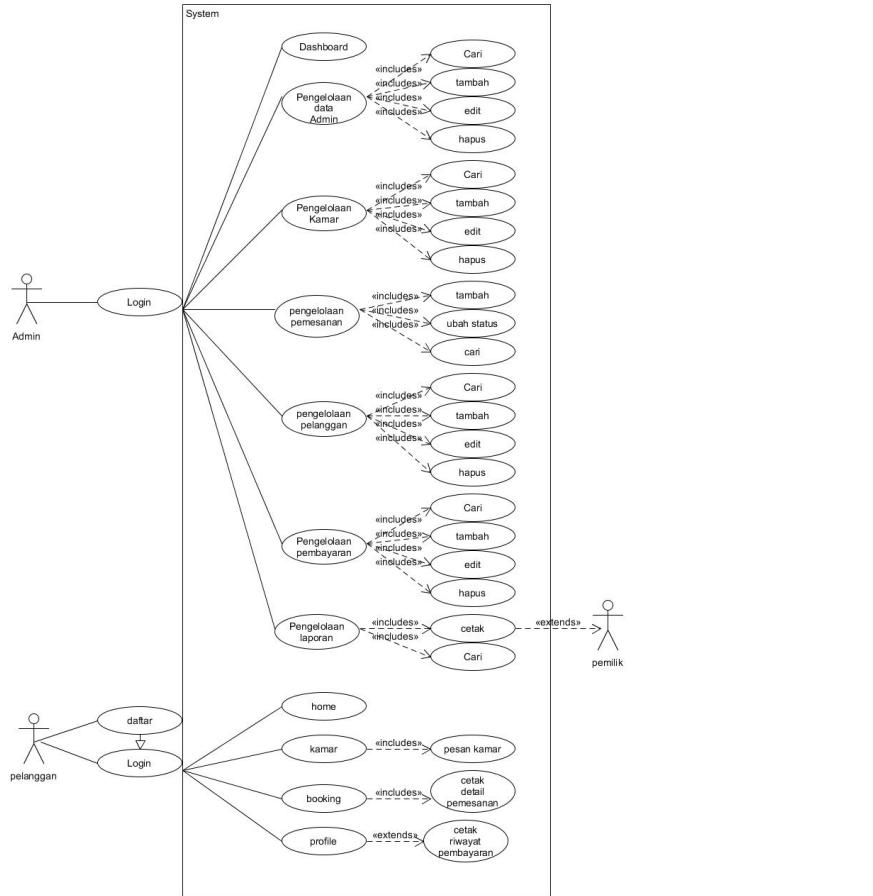
Tabel 4.5 Deskripsi *Usecase Diagram*

| No | Aktor | Deskripsi |
|----|--------------|---|
| 1 | <i>Login</i> | aktor diharuskan untuk melakukan <i>Login</i> terlebih dahulu. Yaitu dengan memasukan <i>Username</i> dan <i>Password</i> . |
| 2 | Pendaftaran | Pendaftaran dilakukan oleh aktor (pelanggan) untuk membuat akun pribadi yang dapat digunakan untuk <i>Login</i> |
| 3 | <i>Home</i> | Setelah <i>Login</i> , pelanggan akan masuk ke tampilan <i>Home</i> |
| 4 | Kamar | Pelanggan dapat memilih menu kamar untuk mendapatkan informasi mengenai ketersediaan kamar, dan dapat melakukan pemesanan kamar jika dirasa kamar yang dipilih sudah cocok. |

| | | |
|----|------------------|--|
| 5 | <i>Booking</i> | Pada menu <i>Booking</i> pelanggan dapat melihat status pesanan setelah mengirimkan bukti pembayaran melalui <i>whatsapp</i> kepada pengelola |
| 6 | <i>Profile</i> | Menu profil digunakan untuk melihat Kembali data diri dan Riwayat pembayaran. Pelanggan juga dapat mencetak Riwayat pembayaran tersebut. |
| 7 | <i>Dashboard</i> | Setelah admin melakukan <i>Login</i> tampilan <i>Dashboard</i> akan muncul dengan menampilkan informasi mengenai kamar yang sudah dipesan, Riwayat pembayaran dan jumlah total pemasukan dari pembayaran kamar indekos |
| 8 | Admin | Proses pengelolaan data admin, yaitu dapat menambahkan, menghapus dan mengedit data admin |
| 9 | Kamar | Proses pengelolaan data kamar, yaitu dapat menambahkan, menghapus, mengupload foto kamar dan mengedit data kamar |
| 10 | Pemesanan | Proses pengelolaan data pemesanan, yaitu dapat menambahkan, menghapus dan mengedit data pemesanan |
| 11 | Pelanggan | Proses pengelolaan data pelanggan, yaitu dapat melihat detail, menambahkan, menghapus dan mengedit data pelanggan |
| 12 | Pembayaran | Proses pengelolaan data pembayaran, yaitu dapat menambahkan, menghapus dan mengedit data pembayaran |
| 13 | Laporan | Menampilkan laporan bulanan yang dapat dicetak untuk dilaporkan kepada pemilik kamar indekos |

3. Usecase aplikasi

Berikut ini merupakan *Usecase* aplikasi pengelolaan kamar indekos Al-Najah :



Gambar 4.1 *Usecase Diagram* Aplikasi

a. Skenario *Usecase Diagram* admin

Berikut adalah Skenario *Usecase Diagram* admin, yang pertama adalah sekernario dari *Usecase Diagram Login* untuk admin :

Tabel 4.6 *Usecase Diagram Login* admin

| | |
|--------|--|
| Nama | <i>Login</i> admin |
| Level | 1 |
| Aktor | Admin |
| Tujuan | Masuk kedalam aplikasi untuk mengelola semua data dalam aplikasi |

| | | |
|----------------------|--|---|
| <i>Precondition</i> | Memasukan <i>Username</i> dan <i>Password</i> | |
| <i>Postcondition</i> | Berhasil memasuki <i>Dashboard</i> | |
| Langkah | User | Sistem |
| | 1. Memasukan <i>Username</i> dan <i>Password</i> | |
| | | 2. Melakukan Validasi <i>Username</i> dan <i>Password</i> |
| | | 3. Menampilkan Halaman <i>Dashboard</i> |

Kemudian setelah *Login* admin akan masuk ke tampilan *Dashboard* yang menampilkan menu lainnya, lalu admin dapat melihat data-data dihalaman admin, kamar, pemesanan, pelanggan, pembayaran, dan laporan. Berikut ini Skenario *Usecase Diagram* admin untuk melihat data:

Tabel 4.7 Skenario *Usecase Diagram Dashboard* admin

| | | |
|----------------------|--|--------|
| Nama | <i>Dashboard</i> | |
| Level | 1 | |
| Aktor | Admin | |
| Tujuan | Masuk ke <i>Dashboard</i> untuk mengelola data-data yang terdapat pada tampilan <i>Dashboard</i> | |
| <i>Precondition</i> | Masuk ke menu admin, kamar, pemesanan, pelanggan, pembayaran dan laporan | |
| <i>Postcondition</i> | Melihat data admin, kamar, pemesanan, pelanggan, pembayaran. Serta mencetak laporan. | |
| Langkah | User | Sistem |
| | 1. Masuk ke menu admin | |

| | |
|----------------------------|--|
| | 2.Menampilkan halaman admin |
| 3. Masuk ke menu kamar | |
| | 4.Menampilkan halaman kamar |
| 5,Masuk ke menu pemesanan | |
| | 6.Menampilkan halaman pemesanan |
| 7.Masuk ke menu pelanggan | |
| | 8.Menampilkan halaman pelanggan |
| 9.Masuk ke menu pembayaran | |
| | 10.Menampilkan halaman pembayaran |
| 11.Masuk ke menu laporan | |
| | 12.Menampilkan halaman Laporan |
| 13. pilih tombol cetak | |
| | 14.Menampilkan halaman cetak Laporan Mencetak Laporan |

Selain admin dapat melihat informasi mengenai admin, pemesanan, pelanggan, pembayaran, dan laporan, admin juga dapat

menambahkan data. Berikut *Usecase* aplikasi admin untuk menambah data :

Tabel 4. 8 Skenario *Usecase Diagram* tambah data admin

| | | |
|----------------------|--|---|
| Nama | <i>Dashboard</i> | |
| Level | 1 | |
| Aktor | Admin | |
| Tujuan | Masuk ke <i>Dashboard</i> untuk menambah data-data yang terdapat pada tampilan <i>Dashboard</i> | |
| <i>Precondition</i> | Masuk ke menu admin, kamar, pemesanan, pelanggan, pembayaran dan laporan. Lalu pilih tombol tambah | |
| <i>Postcondition</i> | Menambah data admin, kamar, pemesanan, pelanggan, dan pembayaran. | |
| Langkah | User | Sistem |
| | 1.Masuk ke menu admin, tambah data admin baru | |
| | | 2.Menampilkan halaman admin, data admin ditambahkan |
| | 3.Masuk ke menu kamar, tambah data kamar baru | |
| | | 4.Menampilkan halaman kamar, data kamar ditambahkan |
| | 5.Masuk ke menu pemesanan, | |

| | | |
|--|--|---|
| | tambahkan data pemesanan baru | |
| | | 6.Menampilkan halaman pemesanan, data pemesanan baru ditambahkan |
| | 7.Masuk ke menu pelanggan, tambahkan data pelanggan baru | |
| | | 8.Menampilkan halaman pelanggan, data pelanggan baru ditambahkan |
| | 9.Masuk ke menu pembayaran, tambahkan data pembayaran baru | |
| | | 10.Menampilkan halaman pembayaran, data pembayaran baru ditambahkan |

Selanjutnya admin dapat mengedit informasi mengenai admin, pelanggan, pembayaran, dan merubah status pemesanan. Berikut *Usecase* aplikasi admin untuk mengedit data :

Tabel 4. 9 Skenario *Usecase Diagram* edit data admin

| | |
|-------|------------------|
| Nama | <i>Dashboard</i> |
| Level | 1 |
| Aktor | Admin |

| | | |
|----------------------|--|--|
| Tujuan | Masuk ke <i>Dashboard</i> untuk mengedit data-data yang terdapat pada tampilan <i>Dashboard</i> | |
| <i>Precondition</i> | Masuk ke menu admin, kamar, pemesanan, pelanggan, pembayaran dan laporan. Lalu pilih tombol edit | |
| <i>Postcondition</i> | Mengedit data admin, kamar, pembayaran, pelanggan dan merubah status pemesanan. | |
| Langkah | User | Sistem |
| | 1.Masuk ke menu admin, edit data admin | |
| | | 2.Menampilkan halaman admin, data admin akan berubah |
| | 3.Masuk ke menu kamar, edit data kamar | |
| | | 4.Menampilkan halaman kamar, data kamar akan berubah |
| | 5.Masuk ke menu pemesanan, ubah status pemesanan | |
| | | 6.Menampilkan halaman pemesanan, status pemesanan akan berubah |
| | 7.Masuk ke menu pelanggan, edit data pelanggan | |

| | | |
|--|--|---|
| | | 8.Menampilkan halaman pelanggan, data pelanggan akan berubah |
| | 9.Masuk ke menu pembayaran, edit data pembayaran | |
| | | 10.Menampilkan halaman pembayaran, data pembayaran akan berubah |

Kemudian admin dapat menghapus informasi mengenai admin, pelanggan, dan pembayaran, Berikut *Usecase* aplikasi admin untuk menghapus data :

Tabel 4.10 Skenario *Usecase Diagram* hapus data admin

| | | |
|----------------------|---|--------|
| Nama | <i>Dashboard</i> | |
| Level | 1 | |
| Aktor | Admin | |
| Tujuan | Masuk ke <i>Dashboard</i> untuk menghapus data-data yang terdapat pada tampilan <i>Dashboard</i> | |
| <i>Precondition</i> | Masuk ke menu admin, kamar, pemesanan, pelanggan, pembayaran dan laporan. Lalu pilih tombol hapus | |
| <i>Postcondition</i> | Menghapus data admin, kamar, pemesanan, pelanggan, dan pembayaran. | |
| Langkah | User | Sistem |
| | 1.Masuk ke menu admin, pilih hapus data admin | |

| | |
|--|--|
| | 2.Menampilkan halaman admin, data admin akan dihapus |
| 3.Masuk ke menu kamar, pilih hapus data kamar | |
| | 4.Menampilkan halaman kamar, data kamar akan dihapus |
| 5.Masuk ke menu pelanggan, hapus data pelanggan | |
| | 6.Menampilkan halaman pelanggan, data pelanggan akan dihapus |
| 7. Masuk ke menu pembayaran, hapus data pembayaran | |
| | 8.Menampilkan halaman pembayaran, data pembayaran akan dihapus |

Admin juga dapat melakukan pencarian data di menu admin, kamar, pelanggan, pemesanan, pembayaran, dan laporan dengan memasukkan kata kunci yang ingin dicari. Berikut adalah Skenario *Use Case Diagram* cari data :

Tabel 4.11 Skenario *Usecase Diagram* cari data

| | | |
|----------------------|---|---------------------------------|
| Nama | <i>Dashboard</i> | |
| Level | 1 | |
| Aktor | Admin | |
| Tujuan | Masuk ke <i>Dashboard</i> untuk mencari data | |
| <i>Precondition</i> | Masuk ke menu admin, kamar, pemesanan, pelanggan, pembayaran dan laporan. Lalu masukkan kata kunci di kolom pencarian | |
| <i>Postcondition</i> | Menghapus data admin, kamar, pemesanan, pelanggan, dan pembayaran. | |
| Langkah | User | Sistem |
| | 1.Masuk ke menu admin, masukkan kata kunci di kolom pencarian | |
| | | 2.Menampilkan data yang dicari |
| | 3.Masuk ke menu kamar, masukkan kata kunci di kolom pencarian | |
| | | 4. Menampilkan data yang dicari |
| | 5.Masuk ke menu pelanggan, masukkan kata kunci di kolom pencarian | |
| | | 6. Menampilkan data yang dicari |

| | | |
|--|---|--|
| | 7. Masuk ke menu pembayaran, masukkan kata kunci di kolom pencarian | |
| | 8. Menampilkan data yang dicari | |
| | 9. masuk ke menu laporan, masukkan tanggal di kolom tanggal, klik tombol cari | |
| | 10. menampilkan data yang dicari | |

b. Skenario *Usecase Diagram* pelanggan

Berikut adalah skernario dari *Usecase Diagram* pelanggan, yang pertama adalah *Usecase Diagram Login* pelanggan :

Tabel 4.12 Skenario *Usecase Diagram Login* pelanggan

| | | |
|----------------------|--|--------|
| Nama | <i>Login</i> pelanggan | |
| Level | 2 | |
| Aktor | Pelanggan | |
| Tujuan | Masuk kedalam aplikasi untuk melihat informasi kamar indekos | |
| <i>Precondition</i> | Memasukan <i>Username</i> dan <i>Password</i> | |
| <i>Postcondition</i> | Berhasil memasuki <i>Home</i> | |
| Langkah | User | Sistem |
| | 1.Memasukan <i>Username</i> dan <i>Password</i> | |
| | 2.Melakukan Validasi <i>Username</i> dan <i>Password</i> | |

| | | |
|--|--|--------------------------------------|
| | | 3.Menampilkan Halaman <i>Home</i> |
|--|--|--------------------------------------|

Jika pelanggan atau pelanggan belum mempunyai akun untuk *Login*, maka pelanggan harus daftar terlebih dahulu, setelah daftar dan mengisi data diri, pelnaggan dapat *Login* ke aplikasi untuk memesan kamar, atau untuk terhubung melalui *whatsapp*. Berikut ini Skenario pendaftaran pelanggan baru :

Tabel 4.13 Skenario *Usecase Diagram* daftar

| | | |
|----------------------|---|------------------|
| Nama | Daftar | |
| Level | 2 | |
| Aktor | Pelanggan | |
| Tujuan | Melakukan pendaftaran untuk membuat akun agar dapat <i>Login</i> | |
| <i>Precondition</i> | Mengisi data diri yaitu nama, alamat, no hp, status, email, user, <i>Password</i> | |
| <i>Postcondition</i> | Berhasil daftar dan membuat akun untuk <i>Login</i> | |
| Langkah | User | Sistem |
| | 1.Memasukan atau mengisi data diri | |
| | | 2.Menyimpan data |
| | 3.Berhasil daftar | |

Jika pelanggan sudah berhasil daftar dan sudah bisa *Login*, pelanggan bisa melihat informasi ketersediaan kamar dan melakukan pemesanan di menu kamar. Berikut adalah Skenario *Usecase Diagram* kamar pelanggan untuk melihat status kamar dan melakukan pemesanan :

Tabel 4.14 Skenario *Usecase Diagram* pemesanan kamar

| | | |
|----------------------|--|-----------------------------------|
| Nama | Kamar | |
| Level | 2 | |
| Aktor | Pelanggan | |
| Tujuan | Masuk ke menu kamar untuk melihat status kamar dan melakukan pemesanan | |
| <i>Precondition</i> | Pilih menu kamar | |
| <i>Postcondition</i> | Melihat halaman kamar dan melakukan pemesanan | |
| Langkah | User | Sistem |
| | 1.masuk ke menu kamar | |
| | | 2.Menampilkan halaman kamar |
| | 3.pilih tombol pesan sekarang | |
| | | 4.Masuk ke halaman <i>Booking</i> |
| | 5.pilih detail pemesanan | |
| | | 6.Menampilkan detail pemesanan |
| | 7.konfirmasi pemesanan | |
| | | 8.Masuk ke <i>whatsapp</i> |
| | 9.memberikan bukti pembayaran | |
| | | 10.Berhasil memesan kamar |

Setelah berhasil memesan, pelanggan juga dapat melihat data diri dan dapat mencetak Riwayat pembayaran apabila diperlukan di menu *Profile*. Berikut adalah Skenario melihat data diri dan mencetak Riwayat pembayaran :

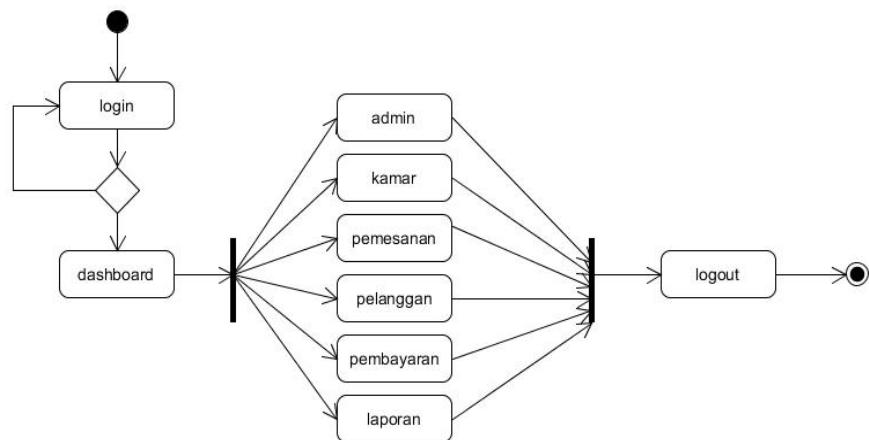
Tabel 4. 15 Skenario *Usecase Diagram Profile*

| | | |
|----------------------|--|--|
| Nama | <i>Profile</i> | |
| Level | 2 | |
| Aktor | Pelanggan | |
| Tujuan | Masuk kedalam aplikasi untuk melihat data diri dan mencetak riwayat pembayaran | |
| <i>Precondition</i> | Pilih menu <i>Profile</i> | |
| <i>Postcondition</i> | Berhasil memasuki halaman <i>Profile</i> | |
| Langkah | User | Sistem |
| | 1.pilih menu <i>Profile</i> | |
| | | 2.menampilkan halaman <i>Profile</i> 3.menampilkan data diri dan Riwayat pembayaran |
| | 4.pilih tombol cetak | |
| | | 5.menampilkan halaman cetak 6.mencetak Riwayat pembayaran |

4.2.2 Activity Diagram

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut ini merupakan *Activity Diagram* dari aplikasi pengelolaan kamar indekos berbasis web di kamar indekos Al-Najah Ciparay :

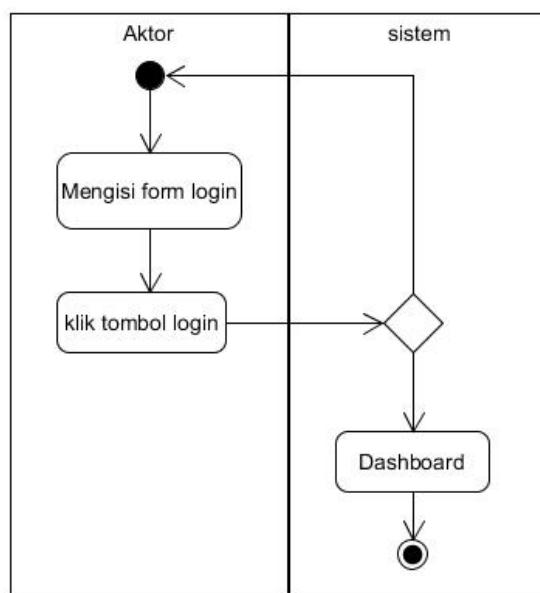
1. Activity Diagram Admin



Gambar 4.2 *Activity Diagram* admin

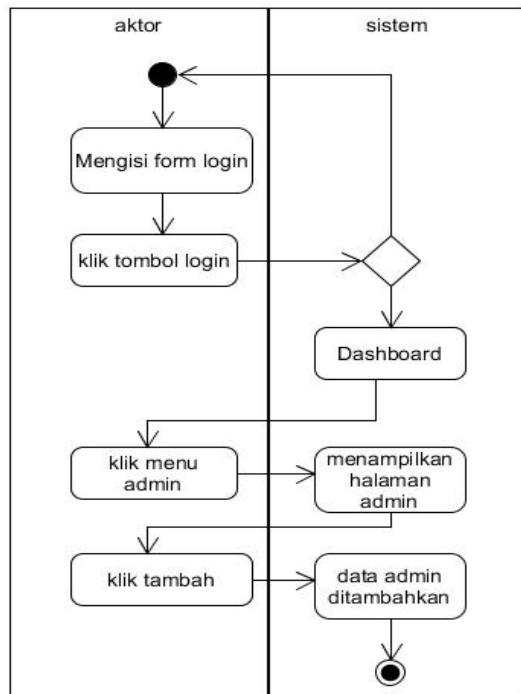
➤ Activity Diagram Login

Berikut adalah *Activity Diagram* untuk *Login* admin :



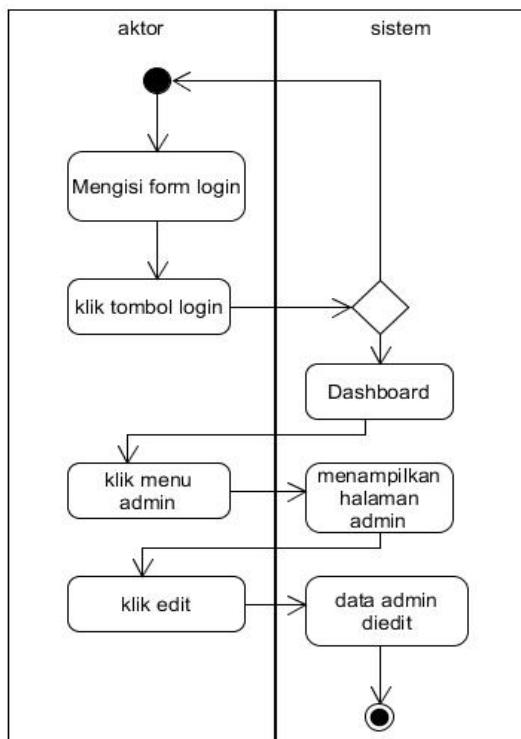
Gambar 4.3 *Activity Diagram* Login admin

➤ *Activity Diagram Tambah data admin*



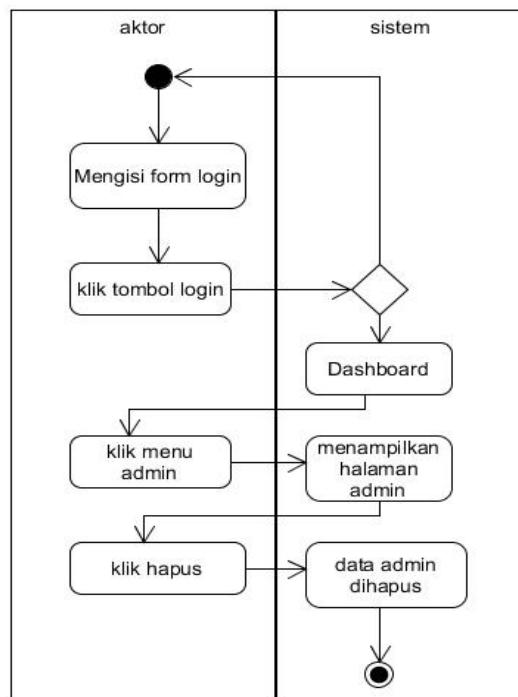
Gambar 4.4 *Activity Diagram* tambah data admin

➤ *Activity Diagram edit data admin*



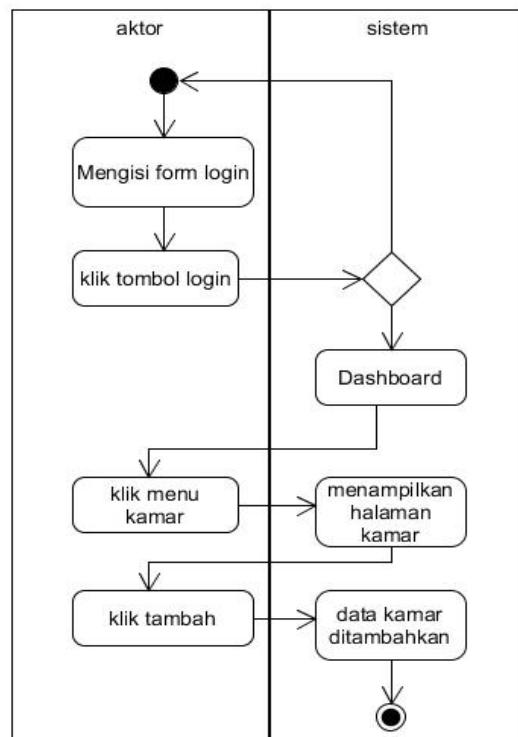
Gambar 4.5 *Activity Diagram* edit data admin

➤ *Activity Diagram* hapus data admin



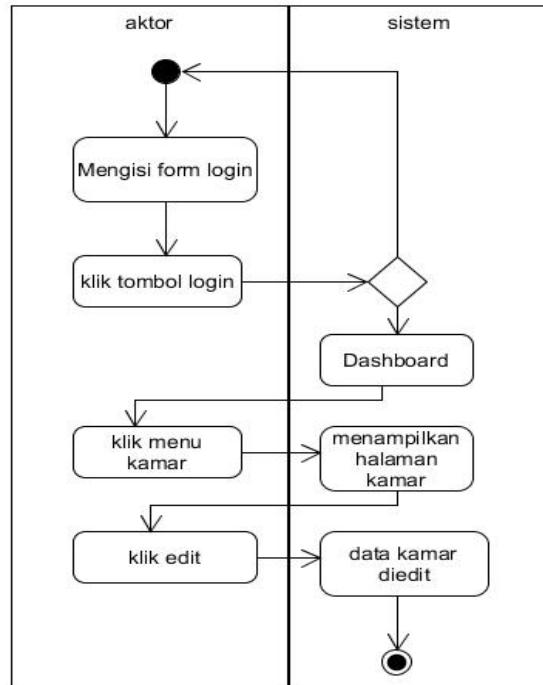
Gambar 4.6 *Activity Diagram* hapus data admin

➤ *Activity Diagram* tambah data kamar



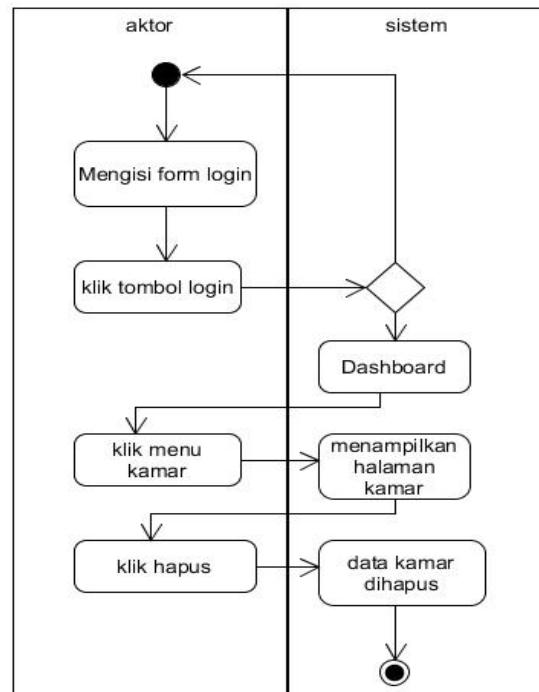
Gambar 4.7 *Activity Diagram* tambah data kamar

➤ *Activity Diagram* edit data kamar



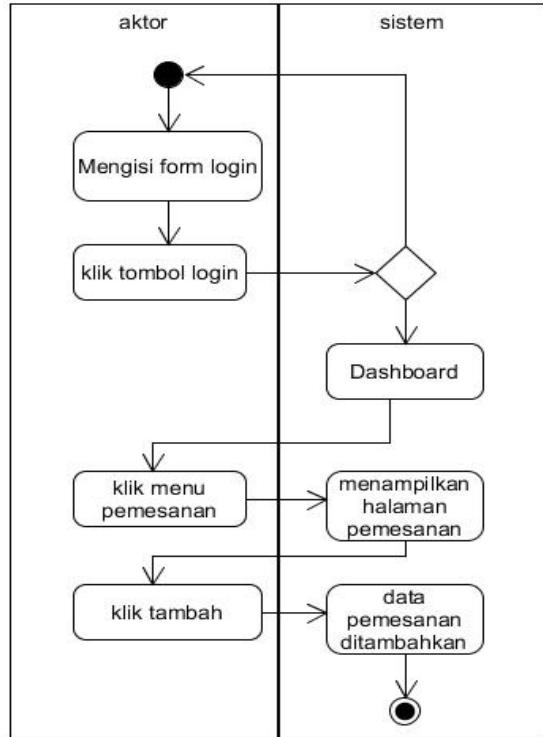
Gambar 4.8 *Activity Diagram* edit data kamar

➤ *Activity Diagram* hapus data kamar



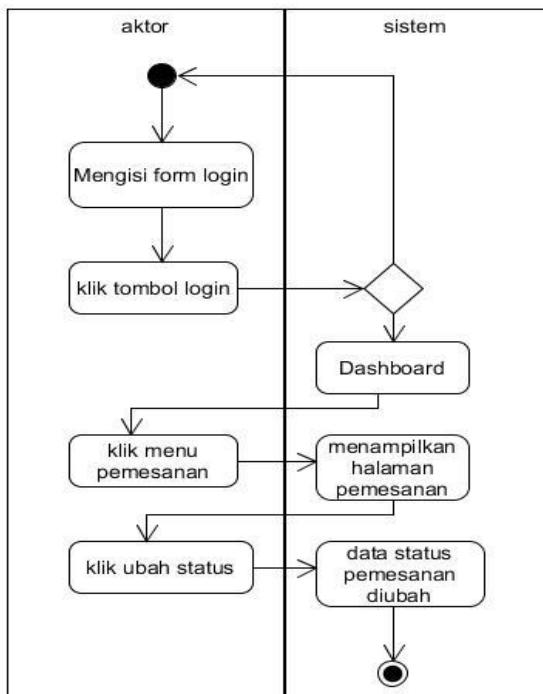
Gambar 4.9 *Activity Diagram* hapus data kamar

➤ *Activity Diagram* tambah data pemesanan



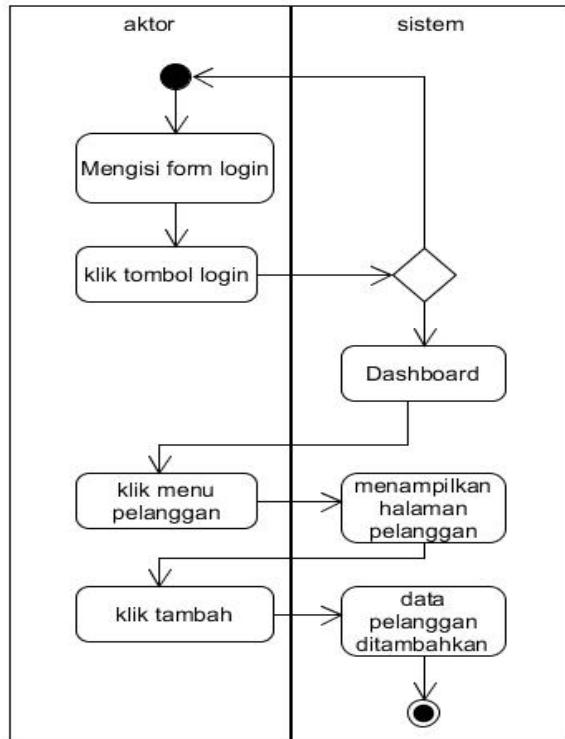
Gambar 4. 10 *Activity Diagram* tambah data pemesanan

➤ *Activity Diagram* ubah status pemesanan



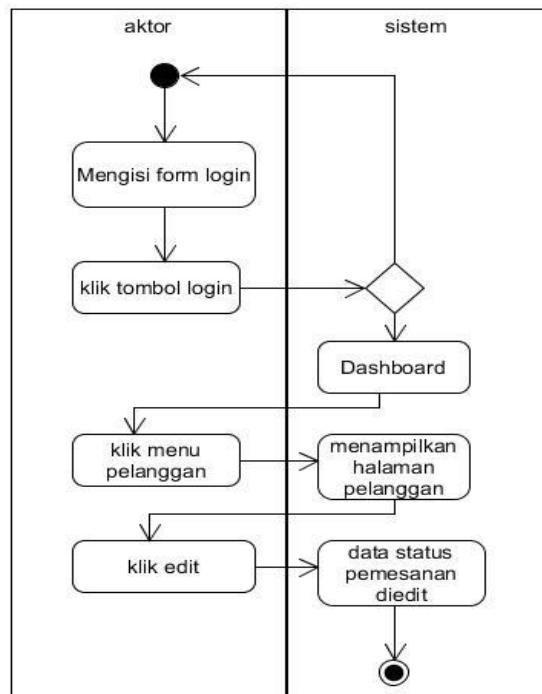
Gambar 4. 11 *Activity Diagram* ubah status pemesanan

➤ *Activity Diagram* tambah data pelanggan



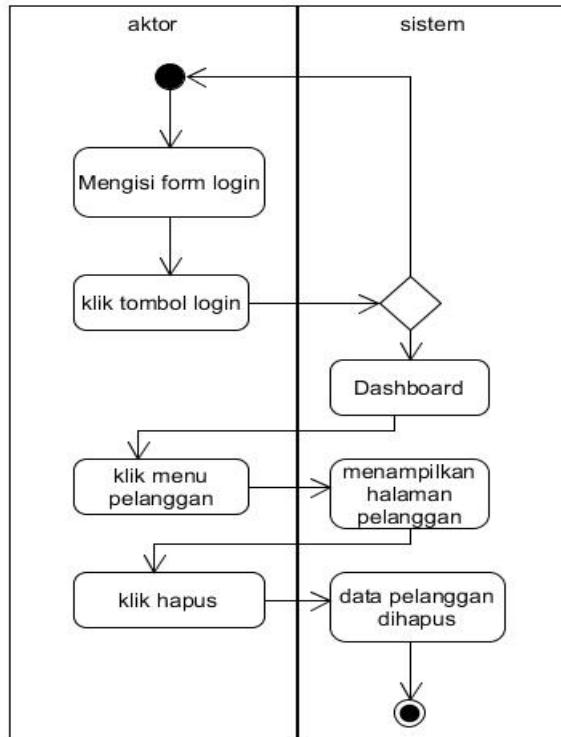
Gambar 4. 12 *Activity Diagram* tambah data pelanggan

➤ *Activity Diagram* edit data pelanggan



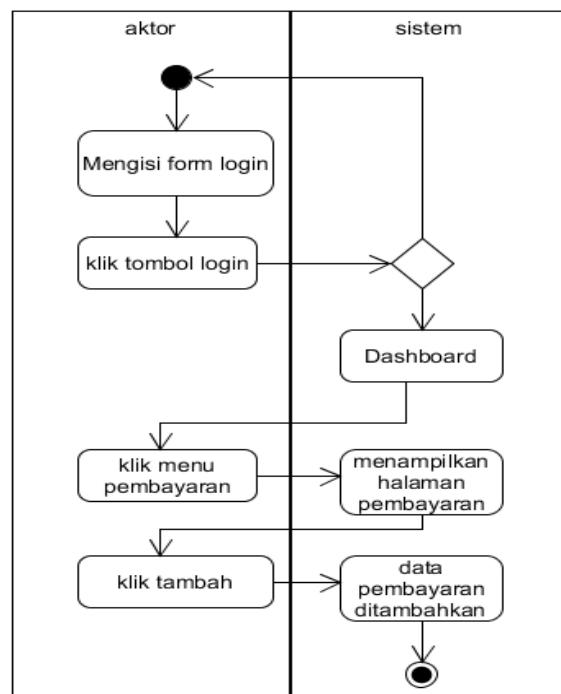
Gambar 4. 13 *Activity Diagram* edit data pelanggan

➤ *Activity Diagram* hapus data pelanggan



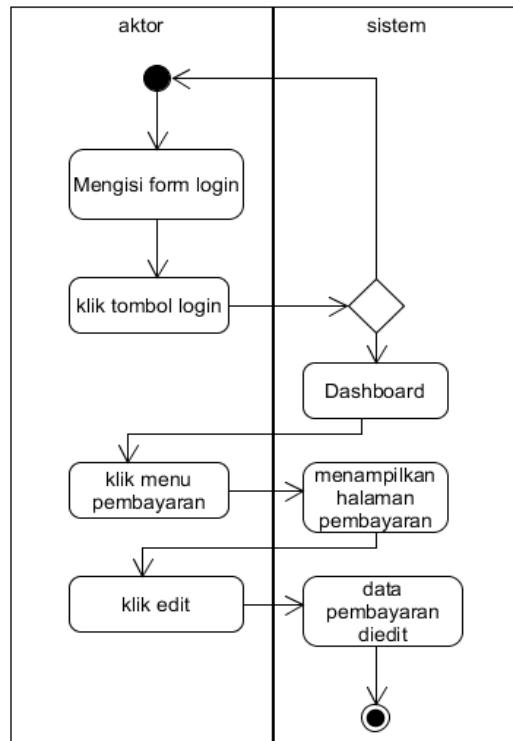
Gambar 4. 14 *Activity Diagram* hapus data pelanggan

➤ *Activity Diagram* tambah data pembayaran



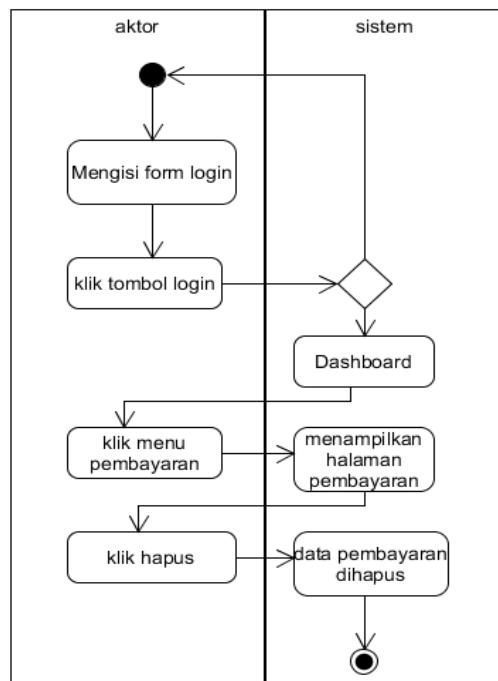
Gambar 4. 15 *Activity Diagram* tambah data pembayaran

➤ *Activity Diagram* edit data pembayaran



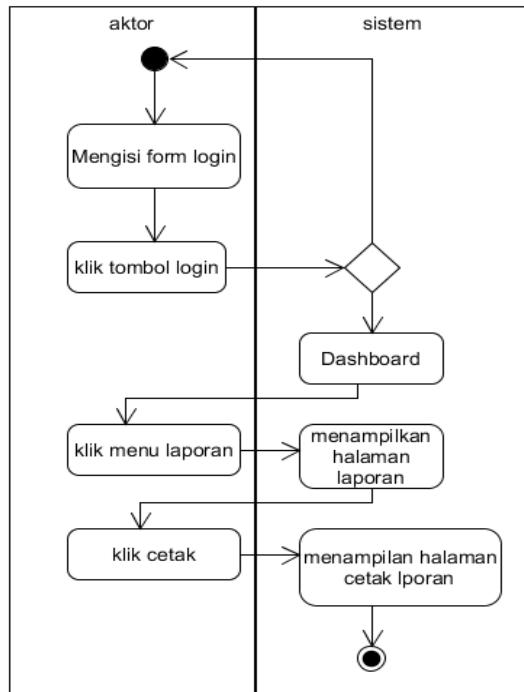
Gambar 4.16 *Activity Diagram* edit data pembayaran

➤ *Activity Diagram* hapus data pembayaran



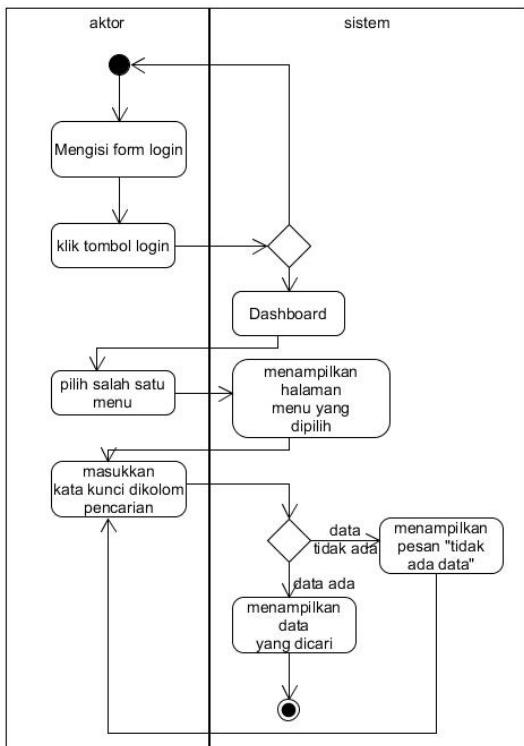
Gambar 4.17 *Activity Diagram* hapus data pembayaran

➤ *Activity Diagram* cetak laporan



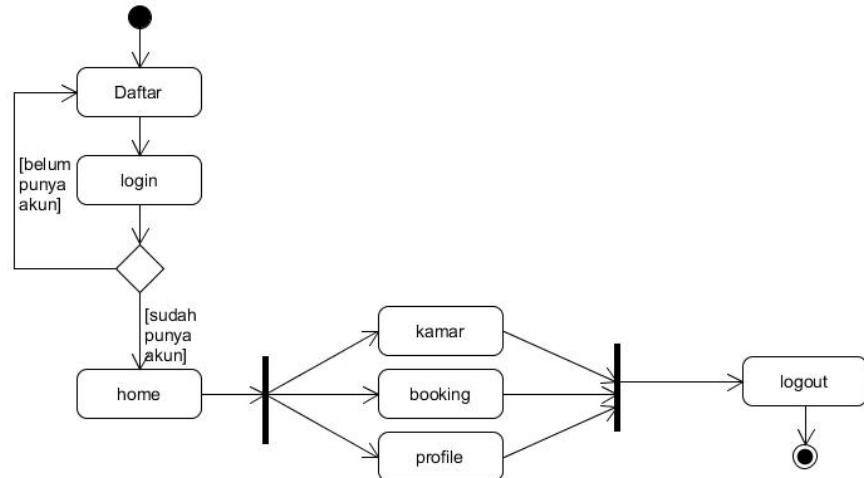
Gambar 4.18 *Activity Diagram* cetak laporan

➤ *Activity Diagram* cari data



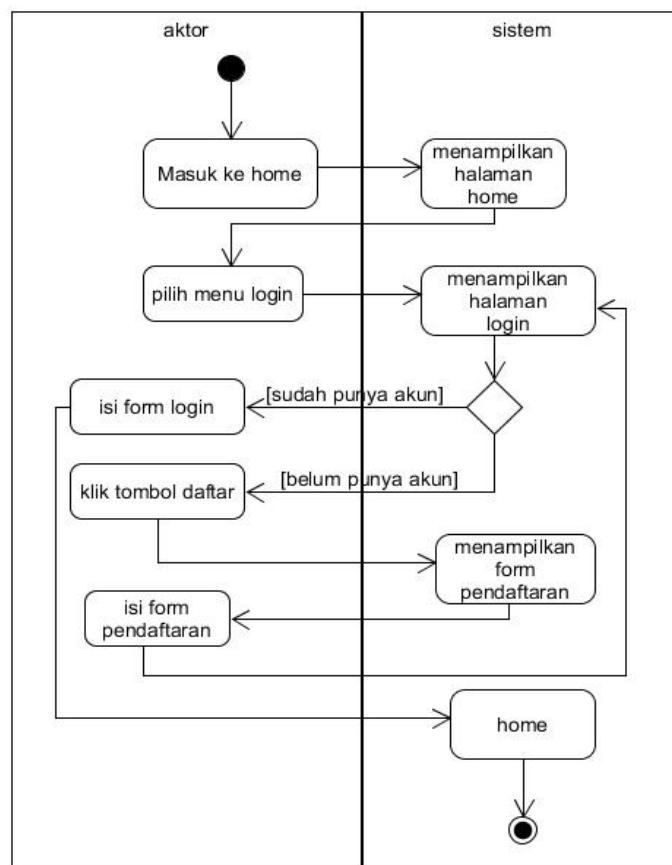
Gambar 4.19 *Activity Diagram* Cari data

2. Activity Diagram Pelanggan



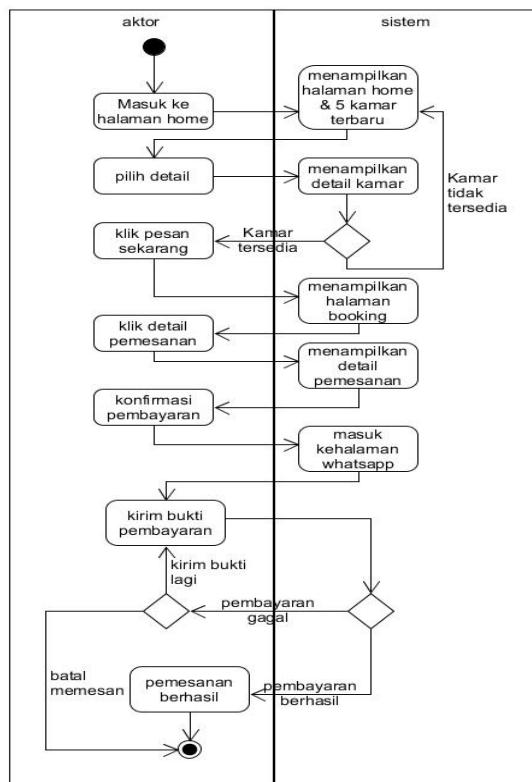
Gambar 4.20 *Activity Diagram* pelanggan

➤ *Activity Diagram* Daftar dan *Login* pelanggan



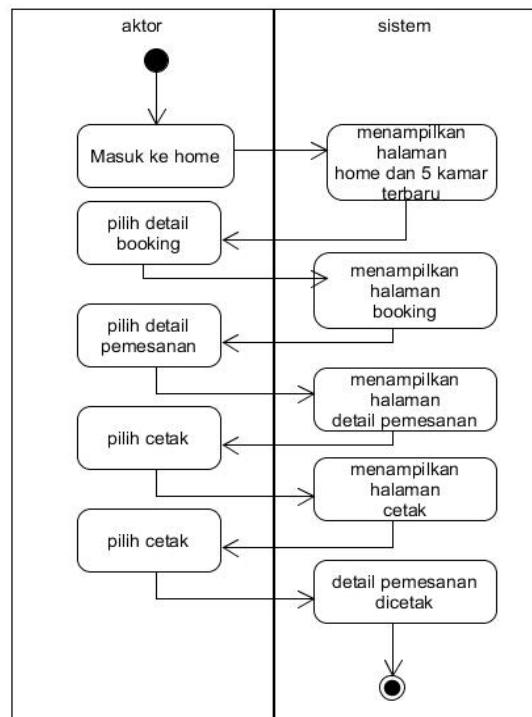
Gambar 4.21 *Activity Diagram* Daftar dan *Login* pelanggan

➤ *Activity Diagram* pesan kamar



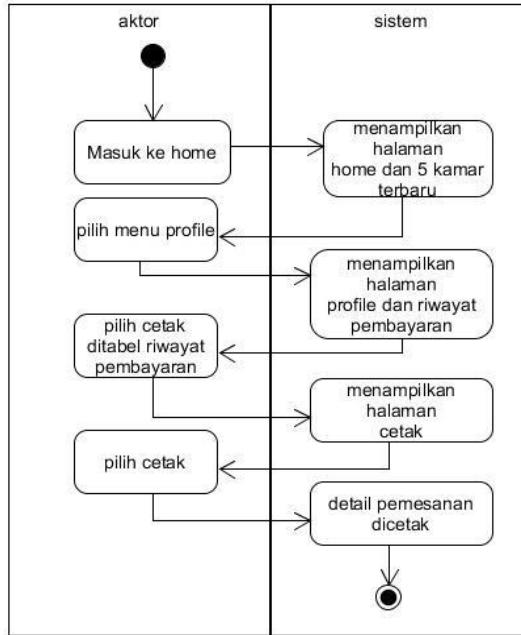
Gambar 4.22 *Activity Diagram* pesan kamar

➤ *Activity Diagram* cetak bukti pemesanan kamar



Gambar 4.23 *Activity Diagram* cetak detail pemesanan

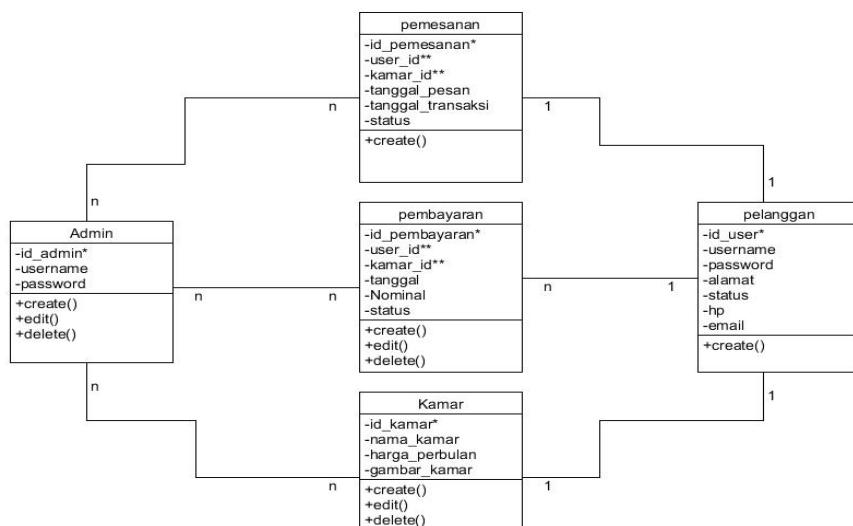
➤ *Activity Diagram* cetak Riwayat pembayaran



Gambar 4. 24 *Activity Diagram* cetak riwayat pembayaran

4.2.3 Class Diagram

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisiannya. Berikut merupakan *Class Diagram* Aplikasi pengelolaan kamar indekos berbasis web di kamar indekos al-najah Ciparay :



Gambar 4. 25 *Class Diagram* Aplikasi pengelolaan kamar indekos

4.2.4 Perancangan Tabel

Berikut ini dijelaskan tabel-tabel yang digunakan dalam perancangan tabel dengan deskripsi sebagai berikut :

1. Tabel admin

Nama Tabel : Admin

Keterangan : Menampung data Admin

Tabel 4.16 Tabel Admin

| NO | Nama field | Tipe Data | Keterangan |
|----|-----------------|---------------|---|
| 1 | <i>id_admin</i> | Int (12) | Nomor identitas untuk pengurutan otomatis |
| 2 | <i>Username</i> | Varchar (255) | Nama pengguna |
| 3 | <i>Password</i> | Varchar (255) | <i>Password</i> pengguna |

2. Tabel kamar

Nama Tabel : kamar

Keterangan : Menampung data kamar

Tabel 4. 17 Tabel kamar

| NO | Nama field | Tipe Data | Keterangan |
|----|-----------------------|---------------|---|
| 1 | <i>id_kamar</i> | Int (12) | Nomor identitas untuk pengurutan otomatis |
| 2 | <i>Nama_kamar</i> | Varchar (255) | Nama kamar |
| 3 | Fasilitas | text | Fasilitas kamar |
| 4 | <i>Harga_perbulan</i> | Int(12) | Harga kamar perbulannya |
| 5 | Gambar kamar | Blob | Menambahkan foto kamar |

3. Tabel pemesanan

Nama Tabel : pemesanan

Keterangan : Menampung data pemesanan kamar

Tabel 4. 18 Tabel pemesanan

| NO | Nama field | Tipe Data | Keterangan |
|----|-------------------|------------|---|
| 1 | id_pemesanan | Int (12) | Nomor identitas untuk pengurutan otomatis |
| 2 | User_id | Int(12) | Nomor identitas user |
| 3 | Kamar_id | Int(12) | Nomor identitas kamar |
| 4 | Tanggal_pemesanan | Date time | Tanggal pemesanan |
| 5 | Tanggal_transaksi | Date time | Tanggal transaksi pemesanan |
| 6 | Status | Tinyint(1) | Status pemesanan |

4. Tabel pelanggan

Nama Tabel : user_kamar

Keterangan : Menampung data pelanggan

Tabel 4.19 Tabel pelanggan

| NO | Nama field | Tipe Data | Keterangan |
|----|---------------|-----------|---|
| 1 | Id | Int (11) | Nomor identitas untuk pengurutan otomatis |
| 2 | User_id | Int(12) | Nomor identitas user |
| 3 | Kamar_id | Int(12) | Nomor identitas kamar |
| 4 | Tanggal_masuk | Date time | Tanggal pelanggan masuk atau memesan |

5. Tabel pembayaran

Nama Tabel : pembayaran

Keterangan : Menampung data pembayaran

Tabel 4.20 Tabel pembayaran

| NO | Nama field | Tipe Data | Keterangan |
|----|---------------|------------|---|
| 1 | id_pembayaran | Int (12) | Nomor identitas untuk pengurutan otomatis |
| 2 | User_id | Int(12) | Nomor identitas user |
| 3 | Kamar_id | Int(12) | Nomor identitas kamar |
| 4 | Tanggal | Date time | Tanggal pembayaran |
| 5 | Nominal | Int(12) | Jumlah bayar |
| 6 | Status | Tinyint(1) | Status pembayaran |

6. Tabel data pelanggan

Nama Tabel : user

Keterangan : Menampung data pelanggan

Tabel 4. 21 Tabel data pelanggan

| NO | Nama field | Tipe Data | Keterangan |
|----|------------|--------------|---|
| 1 | id_user | Int (12) | Nomor identitas untuk pengurutan otomatis |
| 2 | Nama | Varchar(255) | Nama pelanggan |
| 3 | Username | Varchar(255) | Username yang akan digunakan untuk Login |
| 4 | Password | Varchar(255) | Password pelanggan |
| 5 | Alamat | Text | Alamat pelanggan |
| 6 | Status | Tinyint(1) | Status pelanggan |
| 7 | Hp | Varchar(255) | Nomor handphone pelanggan |
| 8 | Email | Varchar(255) | Email pelanggan |

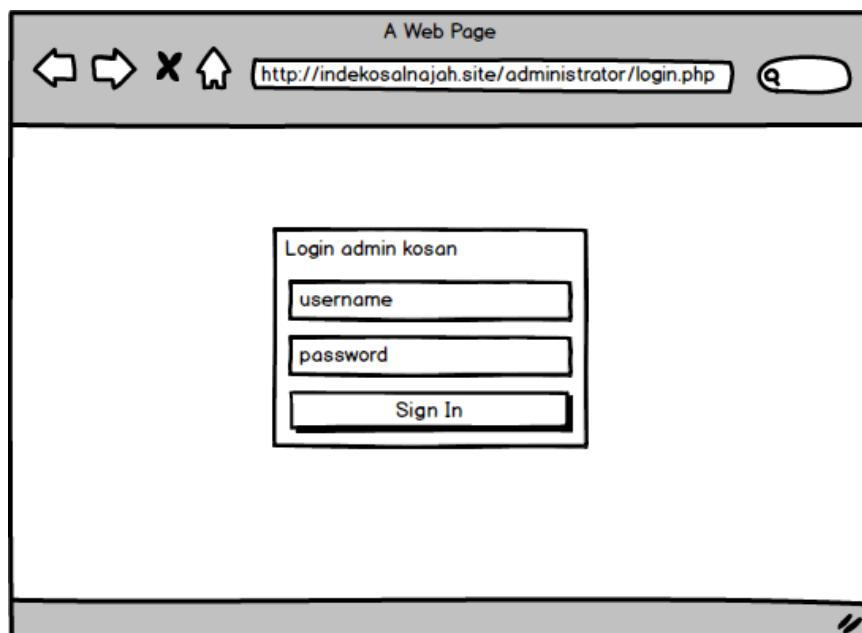
4.2.5 User Interface

User Interface merupakan tampilan yang dibuat agar memudahkan komunikasi antara pengguna dengan sistem aplikasi. Perancangan *User Interface* sendiri dibuat sesederhana mungkin untuk memudahkan pengguna dalam mengakses aplikasi sehingga dapat mengurangi tingkat kebingungan pengguna dalam menggunakan aplikasi.

1. *User Interface admin*

- *User Interface Login* admin

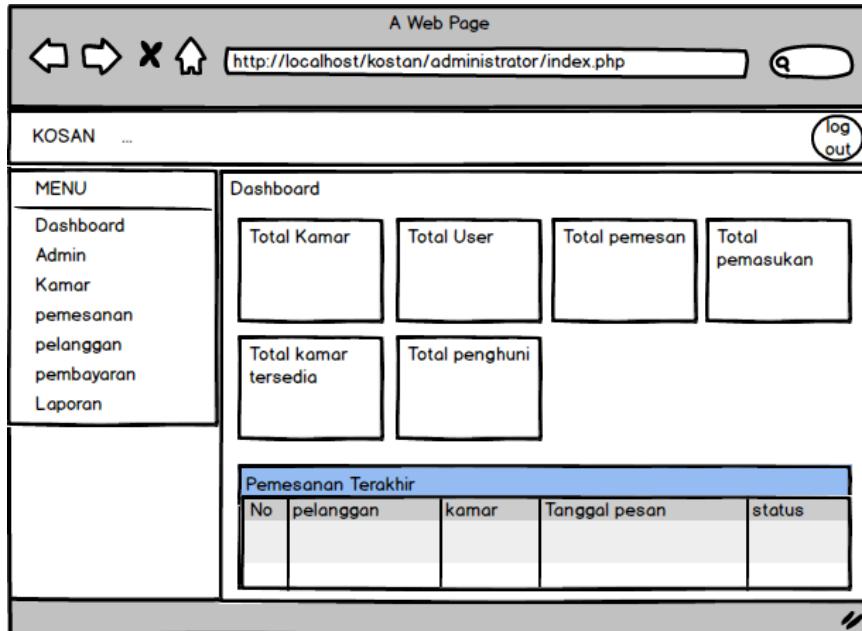
Halaman *Login* admin memuat form untuk mengisi user name dan *Password* agar admin dapat masuk ke *Dashboard* untuk mengelola data kamar indekos.



Gambar 4.26 *User Interface Login* admin

- *User Interface Dashboard* admin

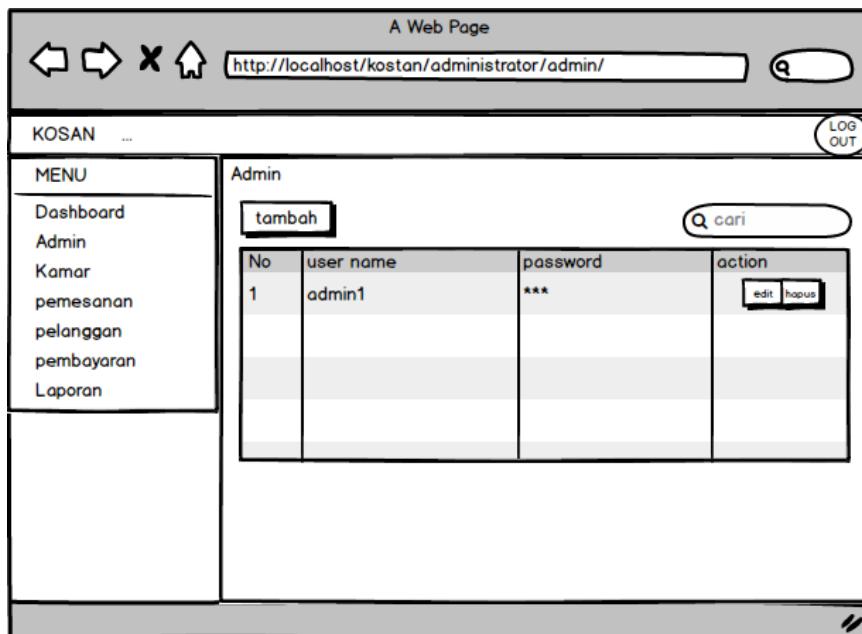
Setelah admin berhasil melakukan *Login*, admin akan masuk ke halaman *Dashboard* yang memuat informasi mengenai total kamar, total pelanggan ,total pemesanan, total pemasukan, dan pemesanan terakhir. Pada halaman *Dashboard* juga memuat beberapa menu lain, yaitu menu kamar, admin, pemesanan,pelanggan,pembayaran, dan laporan.



Gambar 4.27 User Interface Dashboard admin

➤ User Interface Menu Admin

Pada menu Admin memuat informasi mengenai data admin, yaitu user name, dan *Password* admin, kemudian admin juga dapat menambahkan admin yang baru, mencari, mengedit , dan menghapus data datanya.



Gambar 4.28 User Interface menu Admin

➤ *User Interface* menu kamar admin

Setelah di menu *Dashboard* admin dapat memilih menu kamar, dihalaman menu kamar memuat informasi nama kamar, fasilitas, harga kamar, status ketersediaan kamar. Admin juga dapat mencari, menambah, mengedit dan menghapus data kamar.

Gambar 4.29 *User Interface* menu kamar

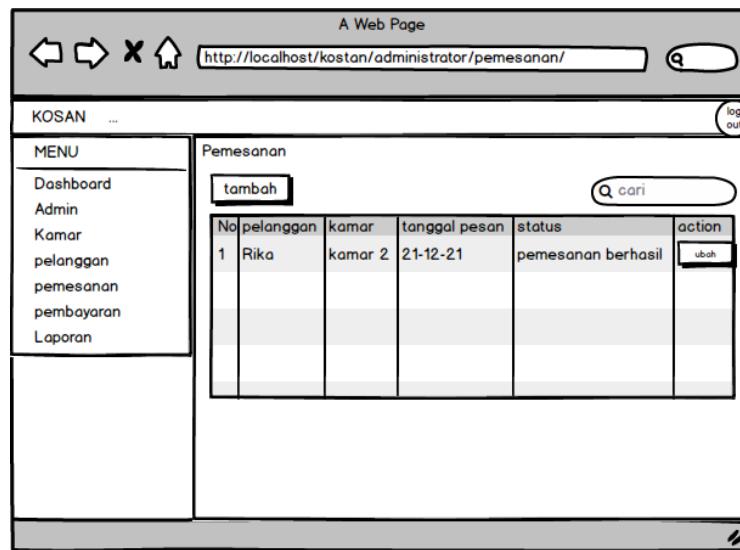
➤ *User Interface* tambah data kamar

Dalam menu kamar admin dapat menambahkan kamar baru, ada beberapa form yang harus diisi, diantaranya nama kamar, fasilitas, harga kamar perbulan, dan mengunggah gambar kamar.

Gambar 4.30 *User Interface* tambah kamar

➤ *User Interface* pemesanan

Data pemesanan memuat informasi pelanggan yang sudah melakukan pemesanan baik dari aplikasi maupun pemesanan yang dilakukan secara langsung, terdapat beberapa data yaitu data pelanggan, kamar, tanggal pesan dan status yang dapat diubah oleh admin, apakah pemesanannya berhasil, atau gagal.



Gambar 4.31 *User Interface* pemesanan kamar

➤ *User Interface* Tambah Pemesanan

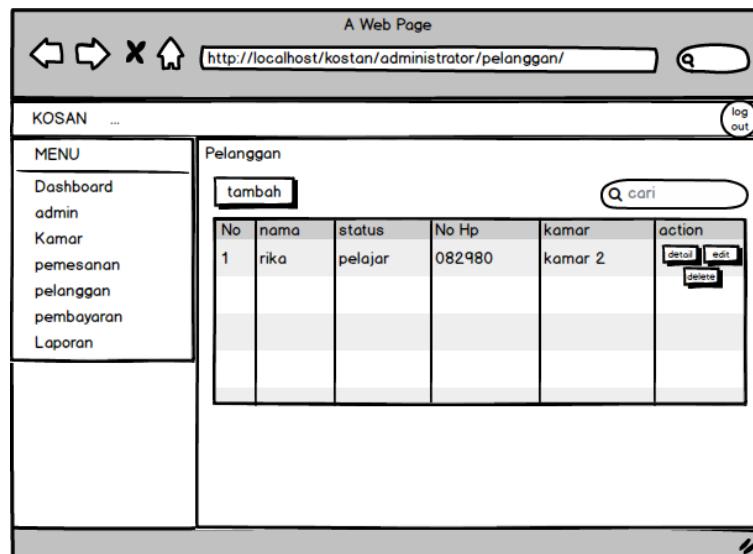
Dalam menu pemesanan terdapat pilihan untuk menambah data pemesanan yang memuat form yang harus diisi antara lain nama pelanggan, kamar yang dipesan, tanggal pemesanan, tanggal transaksi, dan status pemesanan.

The screenshot shows a web page titled 'A Web Page' with the URL 'http://indekosalnajah.site/administrator/pemesanan/'. The page has a header with navigation icons and a 'log out' link. On the left is a sidebar with 'KOSAN ...' and a 'MENU' section containing 'Dashboard', 'Admin', 'Kamar', 'pelanggan', 'pemesanan', 'pembayaran', and 'Laporan'. The main content area is titled 'Create pemesanan' and contains a form with fields for 'pelanggan', 'kamar', 'tanggal pesan', 'tanggal transaksi', 'status', and a 'submit' button.

Gambar 4.32 *User Interface* tambah Pemesanan

➤ *User Interface* pelanggan

User Interface pelanggan memuat informasi mengenai pelanggan, yaitu nama pelanggan, status, Nomber handphone,nama kamar yang disewa. Dan admin juga dapat melakukan penambahan, pengeditan dan penghapusan data pelanggan.



Gambar 4.33 *User Interface* pelanggan

➤ *User Interface* tambah pelanggan baru

Dalam menu pelanggan, admin dapat menambahkan pelanggan baru, ada beberapa form yang harus diisi, diantaranya nama, *Username*, *Password*, alamat, status pekerjaan, number handphone dan email.

| |
|----------|
| nama |
| username |
| password |
| alamat |
| status |
| HP |
| Email |
| submit |

Gambar 4. 34 *User Interface* tambah pelanggan

➤ *User Interface* pembayaran

User Interface pembayaran memuat infomasi data pelanggan, nama kamar yang di sewa, tanggal pembayaran, nominal uang yang dibayarkan, dan status pembayaran lunas. Admin juga dapat menambahkan data pembayaran baru jika ada yang membayar secara langsung.

The screenshot shows a web browser window titled 'A Web Page' with the URL <http://localhost/kostan/administrator/pembayaran/>. The page is titled 'KOSAN ...' and has a 'log out' button in the top right. A left sidebar labeled 'MENU' contains links: Dashboard, Admin, Kamar, pelanggan, pemesanan, pembayaran, Laporan. The main content area is titled 'Pembayaran' and contains a table with one row of data:

| No | pelanggan | kamar | tanggal | nominal | status | action |
|----|-----------|---------|----------|---------|--------|---|
| 1 | Rika | Kamar 2 | 08-02-21 | 500.000 | Lunas | delete edit |

Below the table is a search bar with a 'cari' button. At the top left of the main area is a 'tambah' button.

Gambar 4.35 *User Interface* pembayaran

➤ *User Interface* tambah pembayaran

Dalam menu pembayaran, admin dapat menambahkan pembayaran baru, ada beberapa form yang harus diisi, diantaranya user id, kamar, tanggal bayar, dan nominal uang yang dibayarkan.

The screenshot shows a web browser window titled 'A Web Page' with the URL <http://indekosalnajah.site/administrator/pembayaran/>. The page is titled 'KOSAN ...' and has a 'log out' button in the top right. A left sidebar labeled 'MENU' contains links: Dashboard, Admin, Kamar, pelanggan, pemesanan, pembayaran, Laporan. The main content area is titled 'Create pembayaran' and contains a form with four input fields and a 'submit' button:

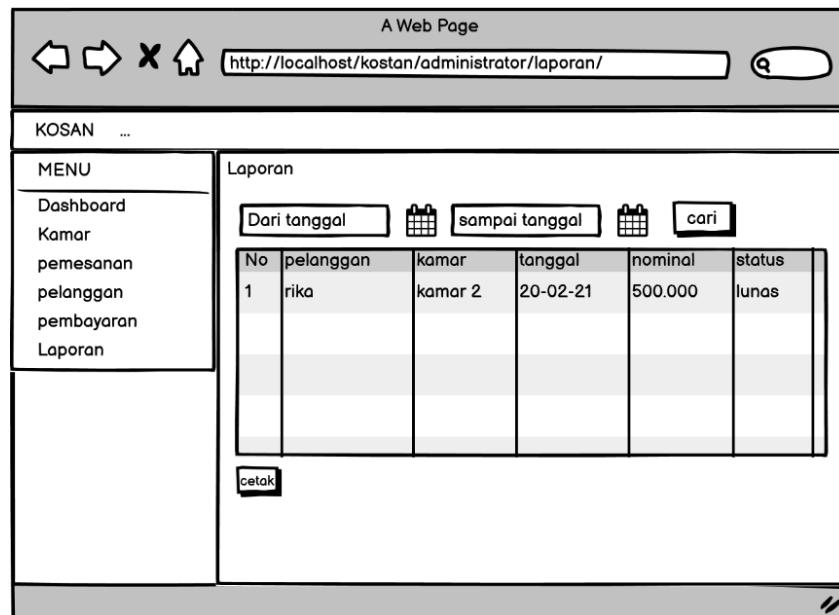
-
-
-
-

Below the inputs is a 'submit' button.

Gambar 4. 36 *User Interface* tambah pembayaran

➤ *User Interface* laporan admin

User Interface laporan admin memuat informasi mengenai laporan keuangan dan data pelanggan, kamar yang disewa, tanggal, nominal, dan status pembayaran para pelanggan selama perbulannya. Admin dapat mencari laporan pengelolaan dari dan sampai tanggal berapa, kemudian admin dapat mencetak laporannya untuk diberikan kepada pemilik kamar indekos setiap bulannya

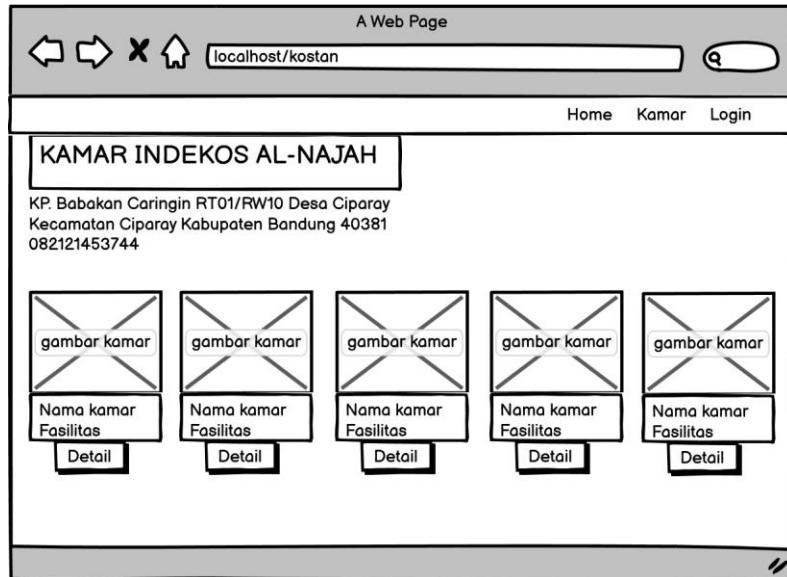


Gambar 4.37 *User Interface* Laporan

2. *User Interface* pelanggan

➤ *User Interface* halaman awal atau *Home*

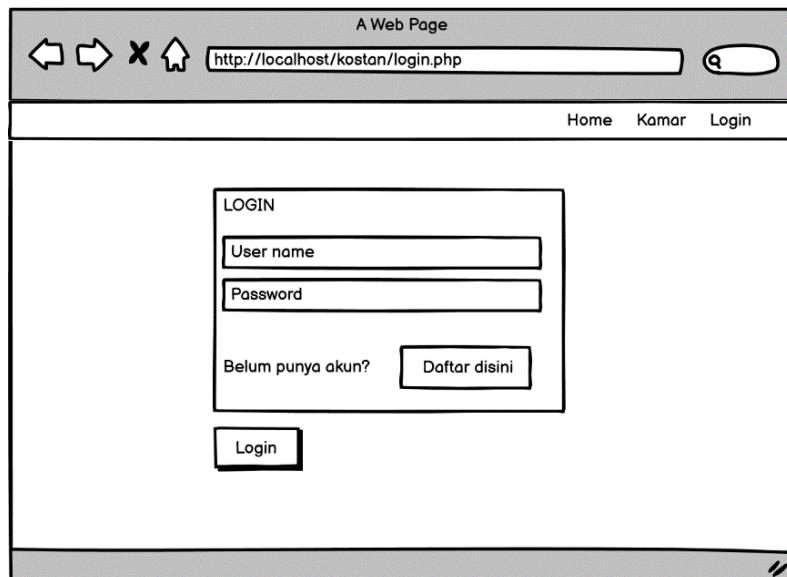
User Interface Home muncul setelah pengguna atau pelanggan mengakses link website aplikasi kamar indekos al-najah. Yang memuat informasi mengenai alamat kamar indekos, gambar gambar 5 kamar tebaru, menu *Home*, kamar dan *Login*.



Gambar 4.38 *User Interface* halaman awal atau *Home*

➤ *User Interface* halaman *Login*

User Interface halaman *Login* memuat form user name dan *Password* yang harus diisi oleh pelanggan agar dapat melakukan pemesanan.



Gambar 4. 39 *User Interface* *Login* pelanggan

➤ *User Interface* daftar pelanggan baru

User Interface dafatr memuat form pendaftaran yang berisi nama, alamat, no hp, status, email, *Username*, dan *Password* yang harus diisi oleh pelanggan baru .

A Web Page
 http://localhost/kostan/daftar.php

Home Kamar Login

DAFTAR

Nama
alamat
No Hp
status
email
username
password

sudah punya akun? Login disini

daftar

Gambar 4.40 *User Interface* pendaftaran pelanggan baru

➤ *User Interface* kamar

User Interface halaman kamar memuat informasi mengenai gambar kamar, nama kamar, fasilitas kamar, dan tombol pesan sekarang untuk memesan kamar

A Web Page
 http://localhost/kostan/detail.php?kamar=7

Home Kamar Booking profile logout

gambar kamar

Nama Kamar

Detail

Fasilitas

Kamar mandi

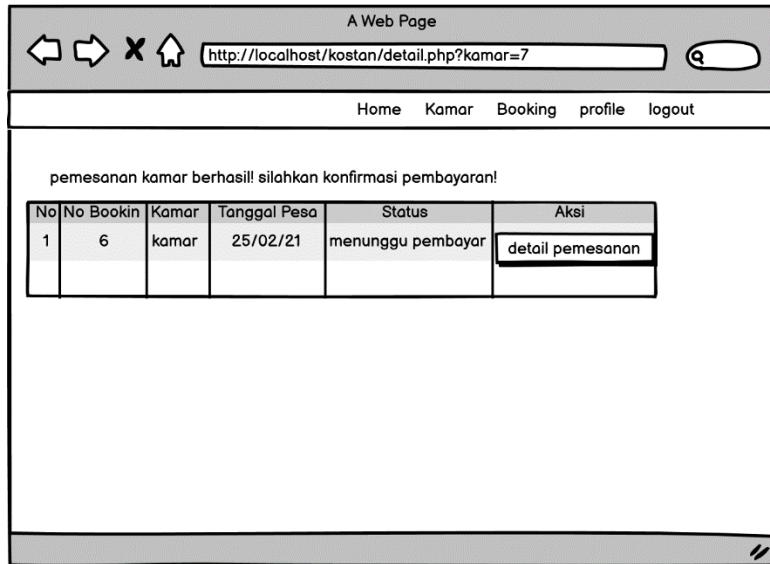
Dapur

Pesan sekarang

Gambar 4.41 *User Interface* kamar

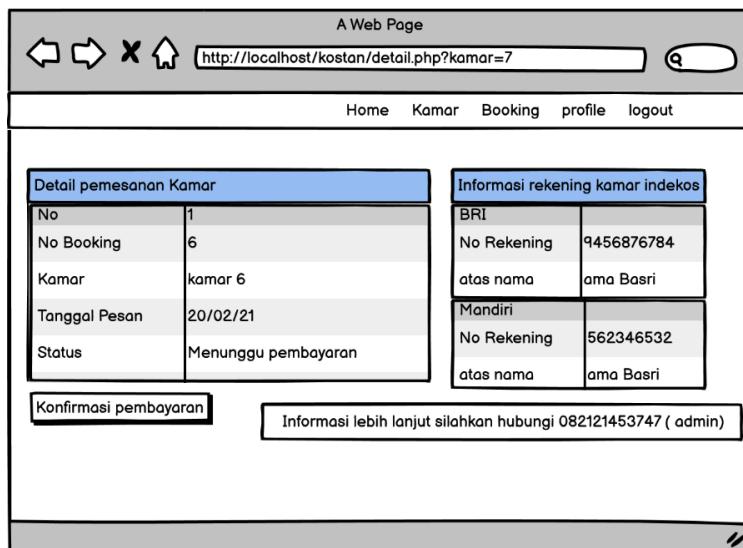
➤ *User Interface Booking*

User Interface halaman *Booking* memuat informasi mengenai status pemesanan kamar, pelanggan dapat menekan tombol detail pemesanan kamar.

Gambar 4.42 *User Interface Booking*

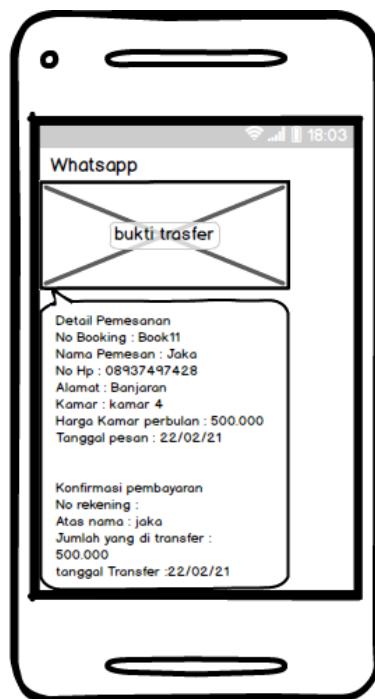
➤ *User Interface* detail pemesanan kamar

User Interface halaman detail pemesanan kamar memuat informasi status pemesanan, nomor rekening dan nomor handphone admin, dan pelanggan dapat menekan tombol konfirmasi pembayaran untuk terhubung langsung dengan pengelola kamar indekos atau admin untuk melakukan konfirmasi pembayaran.

Gambar 4.49 *User Interface* detail pemesanan

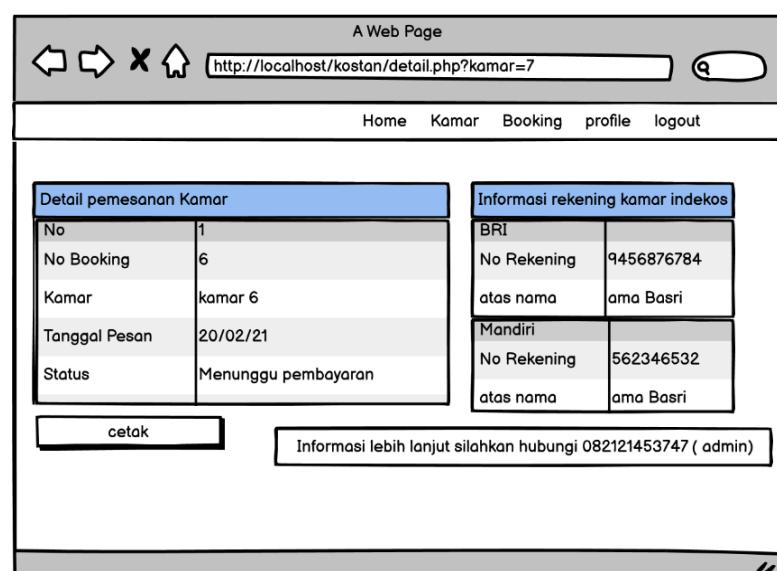
➤ *User Interface* Konfirmasi pembayaran

Berikut ini merupakan tampilan *whatsapp* pelanggan yang terhubung langsung dengan admin untuk melakukan konfirmasi pembayaran.



Gambar 4.43 *User Interface* Konfirmasi pembayaran

➤ *User Interface* Cetak detail pemesanan



Gambar 4.44 *User Interface* cetak detail pemesanan

➤ *User Interface Profile*

User Interface halaman *Profile* memuat informasi data diri, status kamar yang dipesan dan Riwayat seluruh pembayaran yang sudah dilakukan, Riwayat pembayaran tersebut dapat dicetak apabila diperlukan.

The screenshot shows a web browser window titled "A Web Page". The address bar contains the URL "http://localhost/kostan/detail.php?kamar=7". The navigation bar includes links for "Home", "Kamar", "Booking", "profile", and "logout". The main content area is divided into several sections:

- Profile:** A table with columns for "nama", "user name", "email", "alamat", and "no hp".
- status:** A table with columns for "kamar", "biaya perbulan", and "tanggal masuk".
- Riwayat pembayaran:** A table with columns for "Tanggal", "nominal", and "status". A "cetak" button is located to the right of this section.

Gambar 4.45 *User Interface Profile*

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Implementasi merupakan tahap saat pembuatan aplikasi dimulai setelah dilakukan analisis dan desain. Rancangan sistem yang telah disiapkan kemudian diimplementasikan dalam bahasa pemrograman, sehingga semua fungsi dapat dijalankan dengan baik sesuai dengan yang diharapkan.

5.1.1 Implementasi Sistem

Implementasi Aplikasi pengelolaan kamar indekos Al-Najah berbasis web ini dilaksanakan pada :

Hari/Tanggal : Senin, 5 Juli 2021

Waktu : 10:00 s/d Selesai

Tempat : Kamar Indekos Al-Najah, Kp. Babakan Caringin RT01/RW10 Desa Ciparay, Kecamatan Ciparay, Kabupaten Bandung.

5.1.2 Spesifikasi Sistem

Perangkat keras komputer atau *hardware* sangat mendukung dalam kinerja sistem. Semakin baik spesifikasi perangkat keras yang ada maka kinerja sistem pun akan semakin baik. Berikut adalah spesifikasi sistem minimum untuk menjalankan Aplikasi Pengelolaan kamar Indekos A-Najah Ciparay :

- Perangkat Keras

Spesifikasi perangkat keras minimum yang dapat digunakan untuk menjalankan Aplikasi pengelolaan kamar indekos Al-Najah Ciparay ini adalah sebagai berikut :

Tabel 5.1 Spesifikasi perangkat keras minimum

| No | Spesifikasi | Deskripsi |
|----|-------------|-----------|
| 1 | Processor | CPU 1 GHz |
| 2 | RAM | 2 GB |
| 3 | Harddisk | 80 GB |

| | | |
|---|---------------|---|
| 4 | Video Graphic | DirectX 9 graphic device dengan WDDM driver |
|---|---------------|---|

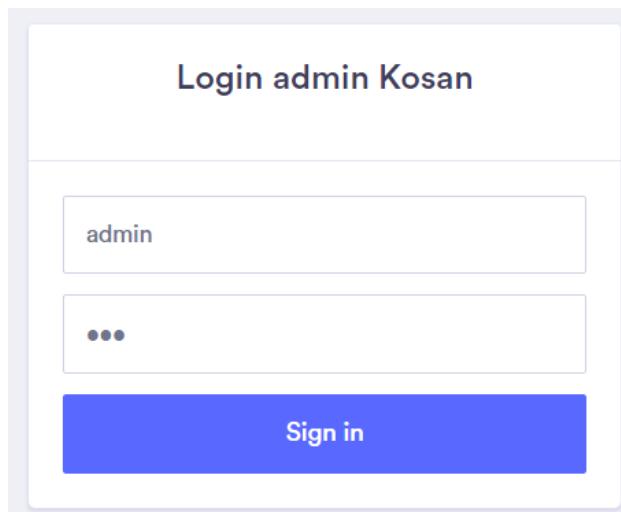
- Perangkat Lunak

Spesifikasi perangkat lunak minimum yang dapat digunakan untuk menjalankan Aplikasi pengelolaan kamar indekos Al-Najah Ciparay ini adalah Komputer dengan Sistem Operasi Windows 7 32-bit yang sudah terinstall Xampp dan web browser.

5.1.3 Implementasi Antar Muka

Implementasi *User Interface* atau antarmuka dilakukan menggunakan bahasa pemrograman PHP dan bootstrap. *User Interface* dibuat sesuai dengan rancangan yang sebelumnya telah dibuat.

- 1) Halaman *Login Admin*



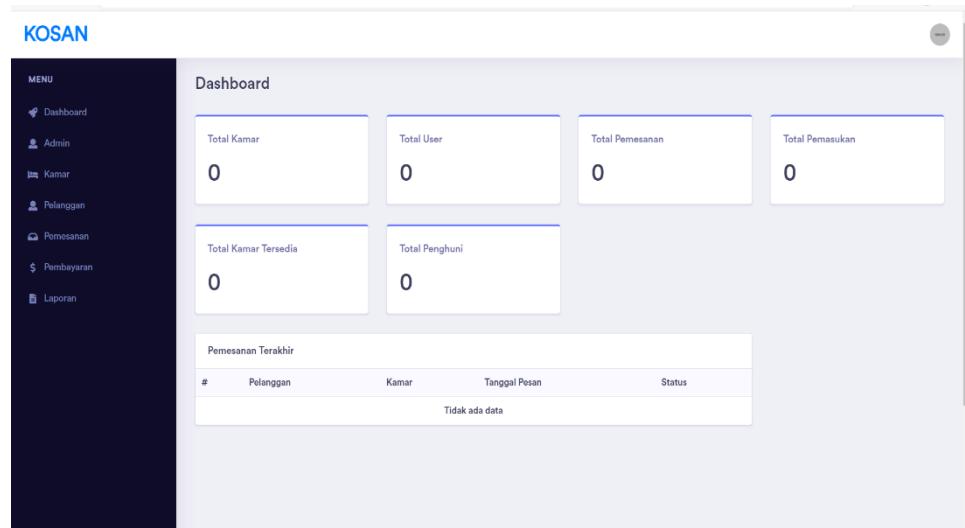
Gambar 5.1 Tampilan Halaman *Login Admin*

Pada saat admin masuk ke link *indekosalnajah.net* admin akan masuk ke halaman *Login*, dimana admin diharuskan untuk memasukkan *Username* dan *Password* untuk dapat masuk ke halaman selanjutnya.

- 2) Halaman *Dashboard*

Setelah berhasil *Login*, sistem akan menampilkan halaman *Dashboard* yang memuat beberapa menu yaitu menu admin, kamar, pelanggan, pemesanan, pembayaran, laporan. Pada halaman *Dashboard* juga memuat

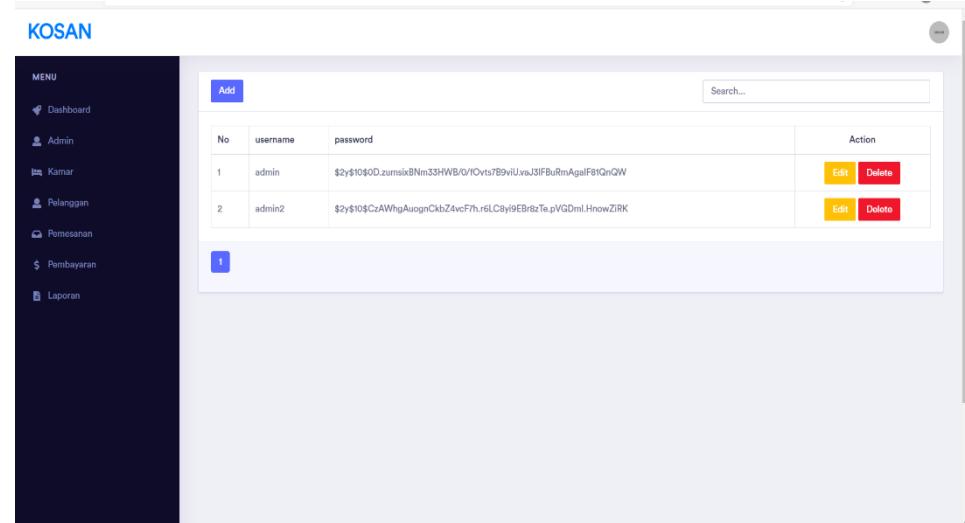
informasi jumlah kamar, jumlah user, total pemasukan, total pemesanan, total kamar tersedia, dan 5 transaksi pemesanan terakhir.



Gambar 5.2 Tampilan halaman *Dashboard*

3) Halaman menu Admin

Setelah masuk ke menu *Dashboard*, admin dapat melakukan pengelolaan data-data di menu admin, yaitu menambah, mengedit dan menghapus data admin.



Gambar 5.3 Tampilan halaman menu admin

4) Halaman menu tambah admin

Pada menu admin terdapat pilihan untuk menambah data admin yang baru, admin harus mengisi form penambahan data yang memuat *Username*, dan *Password* yang nantinya akan digunakan untuk *Login*.

KOSAN

MENU

- Dashboard
- Admin
- Kamar
- Pelanggan
- Pemesanan
- Pembayaran
- Laporan

Tambah admin

username
admin2

password
123

Submit

Gambar 5.4 Tampilan halaman tambah data admin

5) Halaman menu edit data admin

Pada menu admin juga terdapat tombol edit data admin untuk mengubah data admin yang lama dengan data yang baru.

KOSAN

MENU

- Dashboard
- Admin
- Kamar
- Pelanggan
- Pemesanan
- Pembayaran
- Laporan

Edit admin

Username
admin

Password
123

Update

Gambar 5.5 Tampilan halaman edit data admin

6) Halaman menu Kamar

Pada menu kamar terdapat informasi mengenai nama kamar, fasilitas kamar,harga kamar perbulan, dan status ketersediaan kamar. Admin juga dapat menambah, mengedit, menghapus dan mencari data kamar dengan menekan salah satu tombol yang ada di kolom aksi.

| No | Nama Kamar | Fasilitas | Harga perbulan | Status | Aksi |
|----|------------|-----------------------------------|----------------|-----------|---|
| 1 | kamar 1 | kamar mandi, Tempat Tidur, Lemari | Rp 1,000,000 | Sudah ada | <button>Edit</button> <button>Delete</button> |
| 2 | kamar 2 | kamar mandi, Tempat Tidur, Lemari | Rp 1,000,000 | Belum ada | <button>Edit</button> <button>Delete</button> |
| 3 | kamar 3 | kamar mandi, Tempat Tidur, Lemari | Rp 1,000,000 | Sudah ada | <button>Edit</button> <button>Delete</button> |

Gambar 5. 6 Tampilan halaman kamar

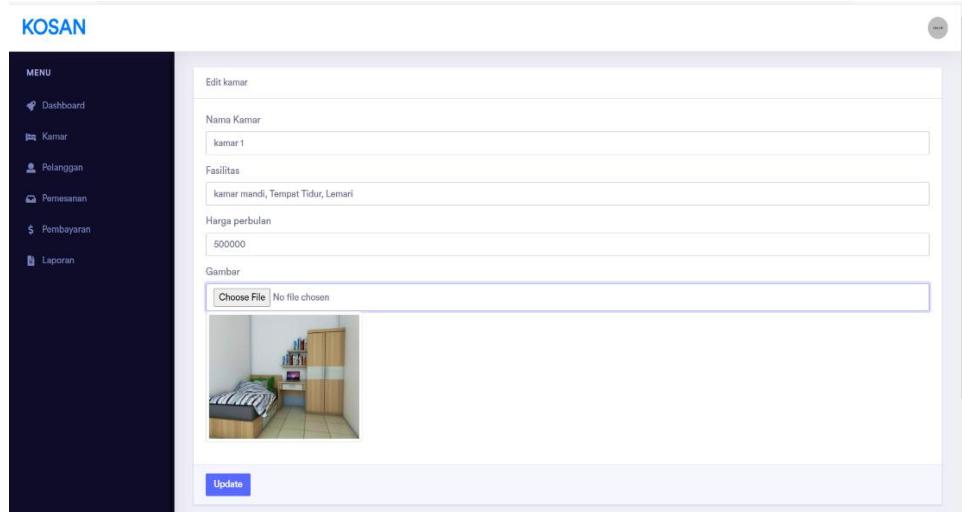
7) Halaman tambah kamar

Pada halaman tambah kamar baru, admin diharuskan untuk mengisi form yang memuat nama kamar, harga kamar perbulan, fasilitas dan unggah gambar kamar.

Gambar 5.7 Tampilan halaman *Create* kamar

8) Halaman Edit data Kamar

Pada menu kamar terdapat tombol edit untuk merubah atau mengupdate data kamar yang lama dengan data yang baru.



Gambar 5.8 Halaman edit data kamar

9) Halaman menu pelanggan

Pada menu pelanggan terdapat informasi mengenai nama pelanggan, status pekerjaan pelanggan, No HP pelanggan, dan nama kamar yang pelanggan pesan. Admin juga dapat menambah data pelanggan jika ada pelanggan yang memesan secara langsung, melihat detail pelanggan, mengedit, dan menghapus data pelanggan.

| No | Nama | Status | Hp | Kamar | Aksi |
|----|-----------------|-----------|--------------|---------|---|
| 1 | Asri Hardiyanti | mahasiswa | 082121453746 | kamar 1 | <button>Detail</button> <button>Edit</button> <button>Delete</button> |
| 2 | Arsyila maika | polajar | 982397982374 | kamar 3 | <button>Detail</button> <button>Edit</button> <button>Delete</button> |

Gambar 5.9 Tampilan halaman menu pelanggan

10) Halaman tambah pelanggan

Pada menu pelanggan admin dapat menambahkan pelanggan baru, ini dilakukan jika ada pelanggan yang tidak bisa daftar secara mandiri di aplikasi. Admin diharuskan mengisi form data pelanggan, diantaranya nama pelanggan, alamat, status pekerjaan, No Hp, Email, *Username*, dan *Password*.

KOSAN

MENU

- Dashboard
- Kamar
- Pelanggan
- Pemesanan
- Pembayaran
- Laporan

Tambah Pelanggan

Nama:

Username:

Password:

Alamat:

Status:

Hp:

Email:

Submit

Gambar 5.10 Tampilan halaman tambah pelanggan

11) Halaman menu edit pelanggan

Pada menu pelanggan terdapat tombol edit untuk merubah atau mengupdate data pelanggan yang lama dengan data yang baru.

KOSAN

MENU

- Dashboard
- Kamar
- Pelanggan
- Pemesanan
- Pembayaran
- Laporan

Edit Pelanggan

Nama: Asri Hardiyanti

Username: asri

Alamat: kp babakan carengin rt01 rw 10 desa ciparay

Status: mahasiswa

Hp: 08212453746

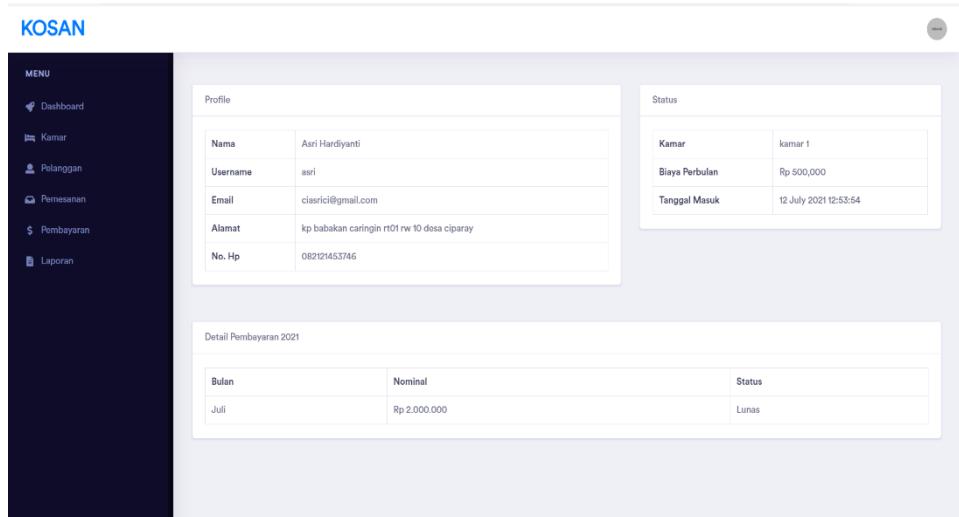
Email: cisrici@gmail.com

Update

Gambar 5.11 Halaman edit data pelanggan

12) Halaman detail pelanggan

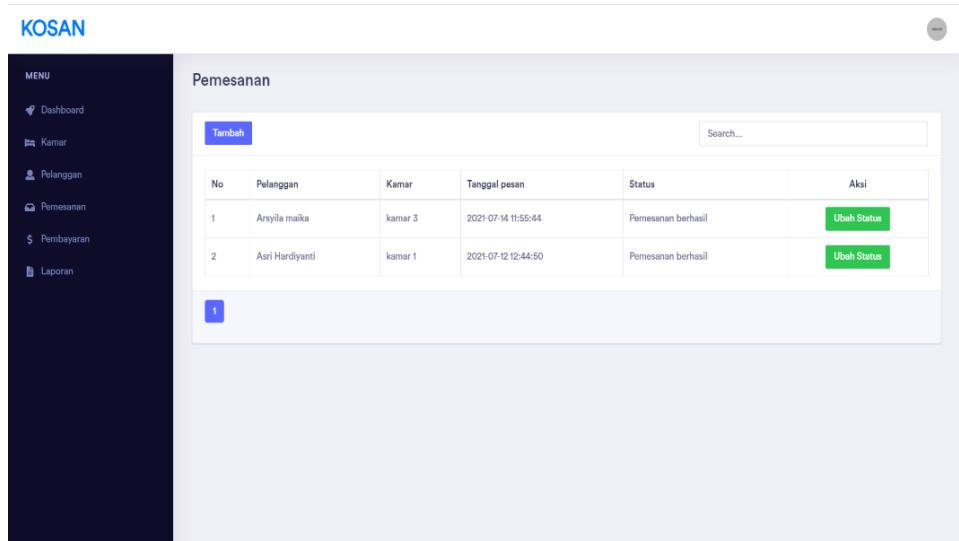
Pada menu pelanggan juga terdapat tombol detail untuk melihat secara detail data pelanggan, data kamar yang dipesan, dan riwayat pembayaran pelanggan tersebut.



Gambar 5.12 Tampilan Halaman detail pelanggan

13) Halaman menu pemesanan

Pada menu pemesanan terdapat informasi mengenai nama pelanggan yang memesan, kamar yang dipesan, tanggal pemesanan dan status pemesanan apakah berhasil memesan, sedang divalidasi ataupun gagal memesan. Admin juga dapat menambah, menghapus, dan mengubah status pemesanan.



Gambar 5.13 Tampilan halaman pemesanan

14) Halaman tambah pemesanan

Pada menu tambah pemesanan ini dilakukan apabila ada pelanggan yang tidak bisa memesan kamar secara mandiri melalui aplikasi, jadi admin lah yang diharuskan mengisi form nama pelanggan yang sebelumnya sudah diisi terlebih dahulu di form pelanggan, pilih kamar yang dipesan, tanggal

pemesanan, tanggal transaksi, dan status pemesanan apakah berhasil memesan atau tidak.

KOSAN

Tambah pemesanan

Pelanggan
- Pilih Pelanggan -

Kamar
- Pilih Kamar -

Tanggal Pesan
dd/mm/yyyy

Tanggal Transaksi
dd/mm/yyyy

Status
- Pilih status pemesanan -

Submit

Gambar 5.14 Tampilan halaman tambah pemesanan

15) Halaman ubah status pemesanan

Pada halaman pemesanan terdapat tombol ubah status, status disini merupakan status pemesanan apakah sudah berhasil, sedang divalidasi, menunggu pembayaran atau pemesanan gagal/berhenti berlangganan.

KOSAN

Detail pemesanan kamar

Nama:
Arsyilia maika

Kamar:
kamar 3

tanggal_pesan:
2021-07-14 11:55:44

status:
- Pilih status pemesanan -
- Pilih status pemesanan -
Menunggu Pembayaran
Sedang divalidasi
Pemesanan Berhasil
Pemesanan Gagal / Berhenti langganan

Gambar 5.15 Tampilan halaman ubah status pemesanan

16) Halaman menu pembayaran

Pada halaman pembayaran berisi informasi mengenai nama pelanggan, kamar yang dipesan, tanggal bayar, nominal uang yang

dibayarkan, status lunas. Admin juga dapat menambahkan, mencari, mengedit dan menghapus data pembayaran.

Gambar 5.16 Tampilan halaman pembayaran

17) Halaman tambah pembayaran

Pada halaman tambah pembayaran ini, admin harus mengisi form nama pelanggan, kamar yang dipesan, tanggal bayar, dan nominal uang yang dibayarkan.

Gambar 5.17 Tampilan halaman tambah pembayaran

18) Halaman edit pembayaran

Pada halaman pembayaran terdapat tombol edit yang berfungsi untuk mengubah data pembayaran yang lama dengan data yang baru.

Gambar 5.18 Tampilan halaman edit pembayaran

19) Halaman menu laporan

Pada halaman menu laporan memuat informasi mengenai nama-nama pelanggan yang sudah memesan, nama kamar yang dipesan, tanggal pemesanan, nominal uang yang sudah dibayarkan dan total pemasukan uang. Admin juga dapat melihat laporan keuangan dari dan sampai tanggal berapa dan admin dapat mencetaknya untuk diberikan kepada pemilik kamar indekos.

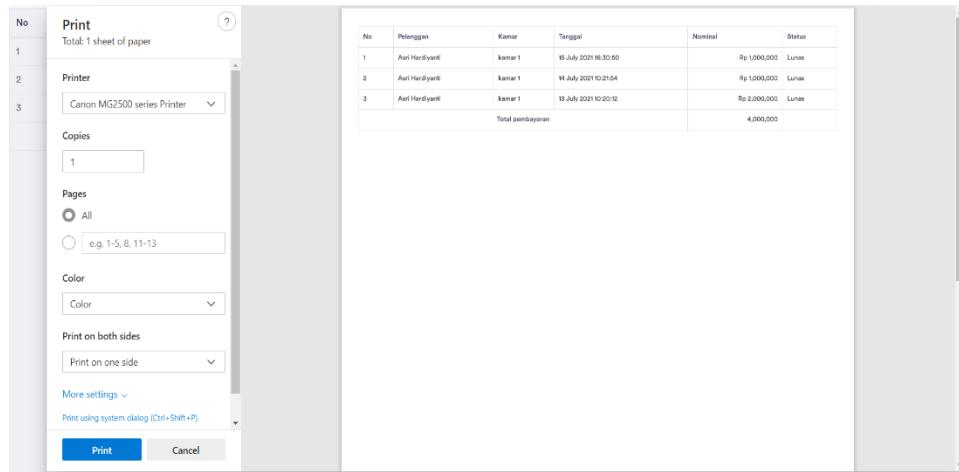
| No | Pelanggan | Kamar | Tanggal | Nominal | Status |
|------------------|----------------|---------|-----------------------|--------------|-----------|
| 1 | Ari Hardiyanti | kamar 1 | 15 July 2021 16:30:50 | Rp 1,000,000 | Lunas |
| 2 | Ari Hardiyanti | kamar 1 | 14 July 2021 10:21:54 | Rp 1,000,000 | Lunas |
| 3 | Ari Hardiyanti | kamar 1 | 13 July 2021 10:20:12 | Rp 2,000,000 | Lunas |
| Total pembayaran | | | | | 4,000,000 |

Gambar 5.19 Tampilan halaman Laporan

20) Halaman Cetak Laporan

Pada halaman laporan terdapat tombol cetak, saat admin menekan tombol cetak maka akan tampil halaman *Print Preview*. Kemudian admin

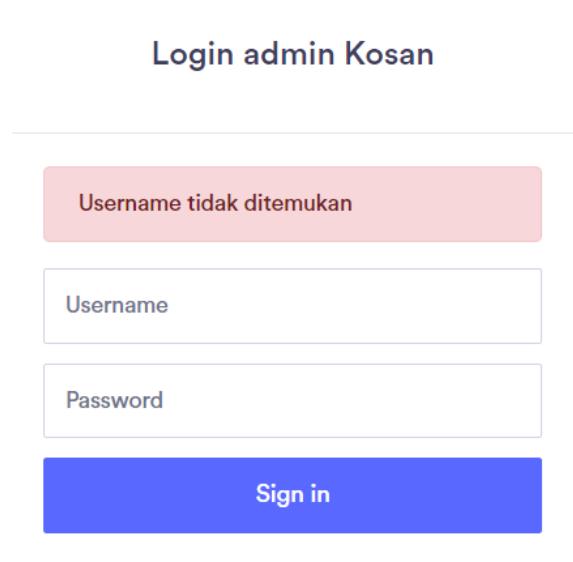
dapat mengatur halaman yang akan dicetak, seperti menentukan printer yang akan digunakan, banyak laporan yang akan dicetak, dan lain-lain.



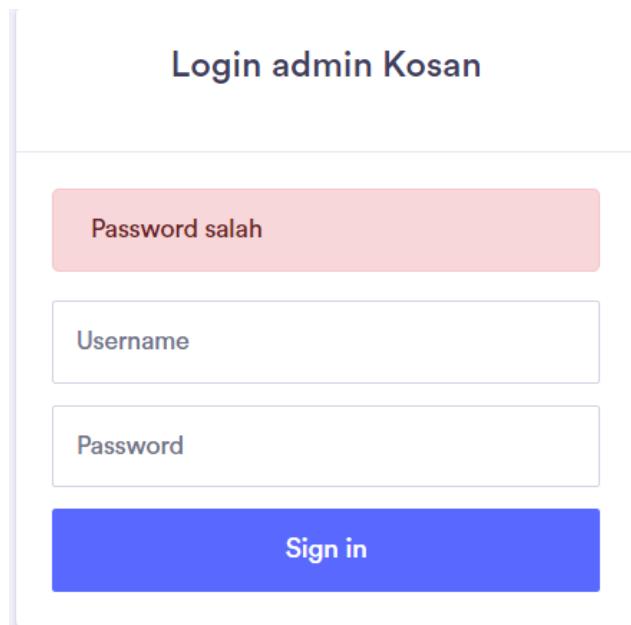
Gambar 5.20 Tampilan halaman cetak laporan

21) Tampilan *Login* admin yang gagal

Jika pada saat melakukan *Login* admin salah dalam memasukkan username atau *Password*, maka sistem akan menampilkan pesan sebagai berikut :



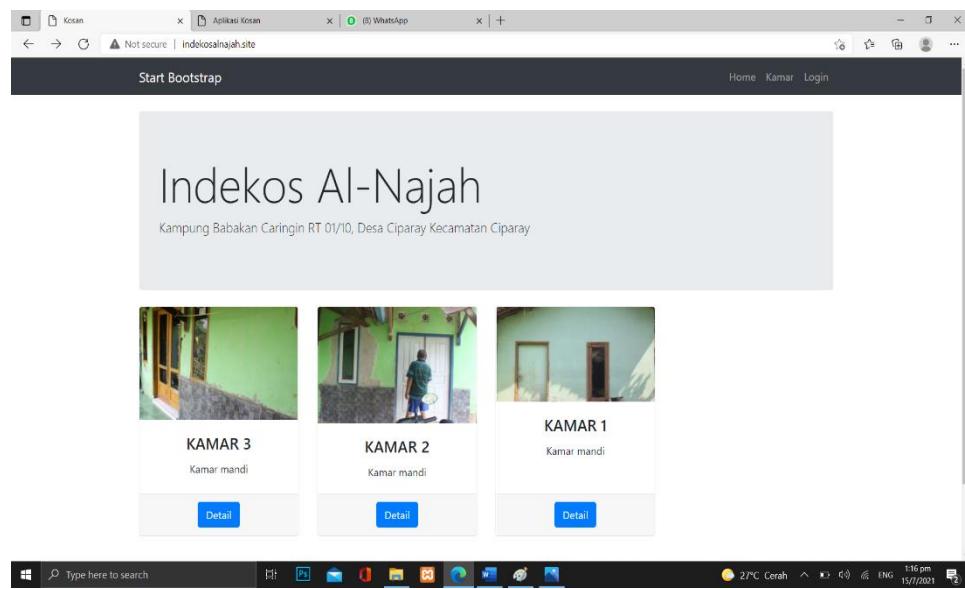
Gambar 5.21 Tampilan pesan saat admin salah *Username*



Gambar 5.22 Tampilan pesan saat admin salah *Password*

22) Halaman *Home*

Pada saat pelanggan masuk ke link indekosalnajah.site, pelanggan akan masuk ke halaman *Home* dari aplikasi kamar indekos, yang mana halaman *Home* tersebut menampilkan 5 kamar terbaru. Pelanggan juga dapat menekan tombol detail untuk dapat melihat detail kamar.

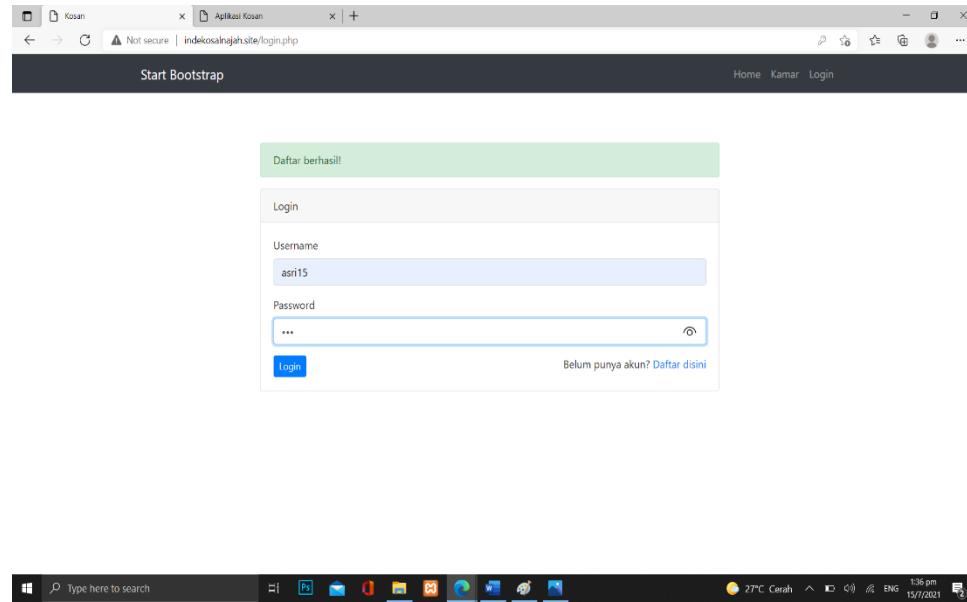


Gambar 5.23 Tampilan halaman *Home*

23) Halaman *Login* pelanggan

Pada halaman *Login* pelanggan, pelanggan harus mengisi *Username* dan *Password* untuk dapat memesan kamar, melihat profil ataupun melihat

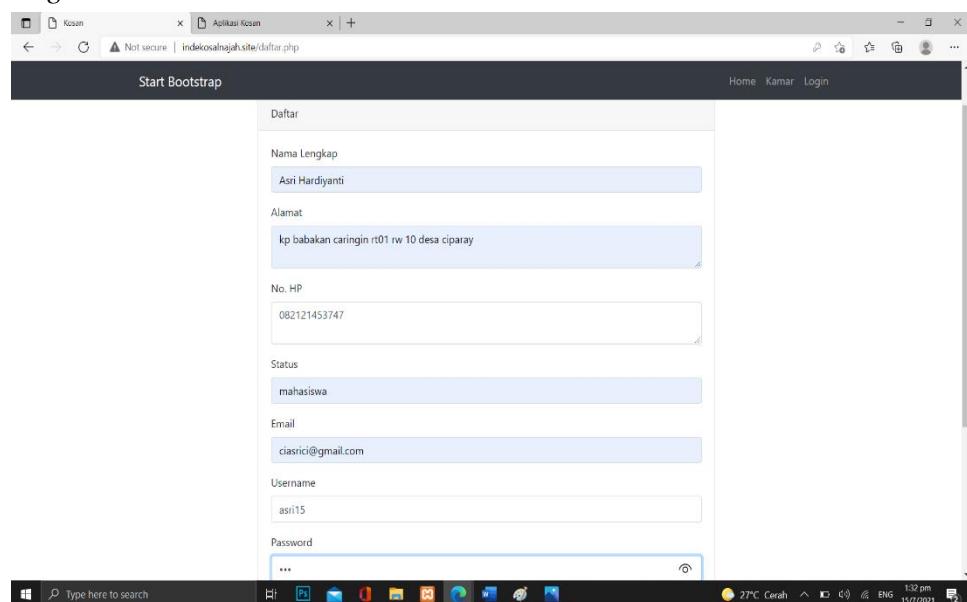
Riwayat pembayaran. Pelanggan juga dapat memilih daftar jika belum mempunyai *Username* dan *Password* untuk *Login*.



Gambar 5.24 Tampilan halaman *Login* pelanggan

24) Halaman pendaftaran pelanggan baru

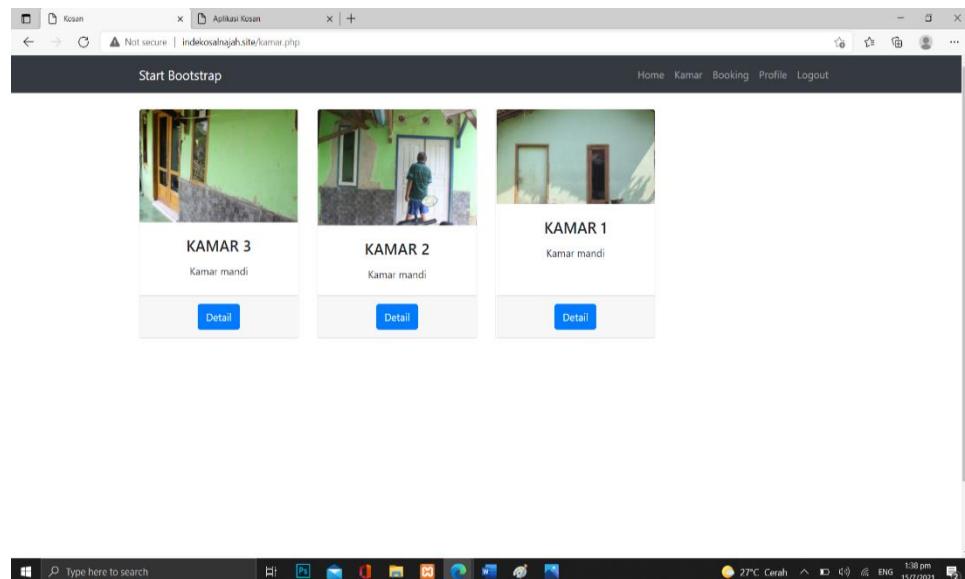
Halaman pendaftaran memuat form yang harus diisi data diri oleh pelanggan, diantaranya nama lengkap, alamat, no hp, status pekerjaan, email, *Username*, dan *Password* yang nantinya akan digunakan untuk *Login*.



Gambar 5.25 Tampilan halaman pendaftaran

25) Halaman kamar pelanggan

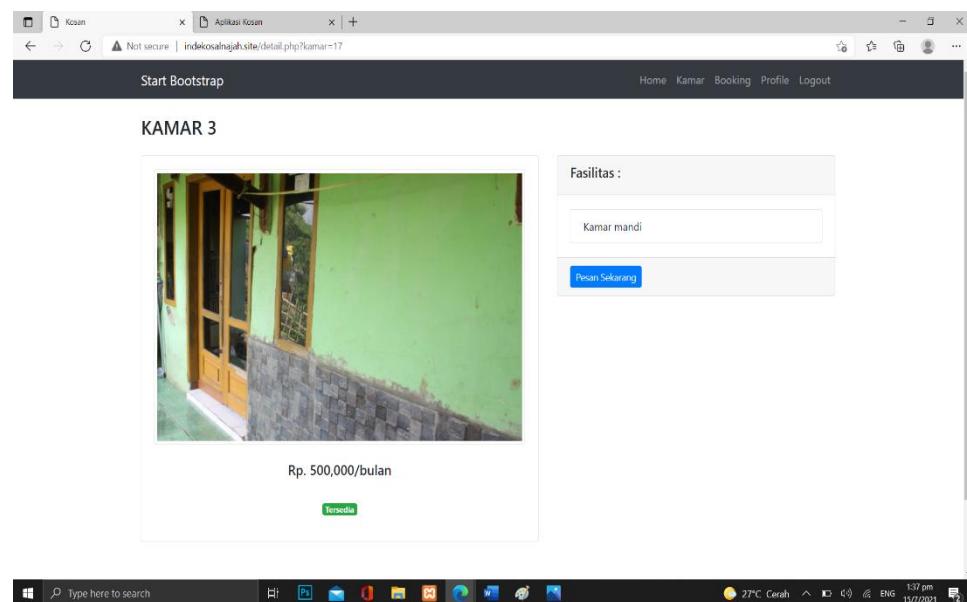
Pada halaman kamar pelanggan memuat gambar gambar kamar, dan pengguna harus menekan tombol detail untuk melihat detail kamar.



Gambar 5.26 Tampilan halaman menu kamar

26) Halaman detail kamar

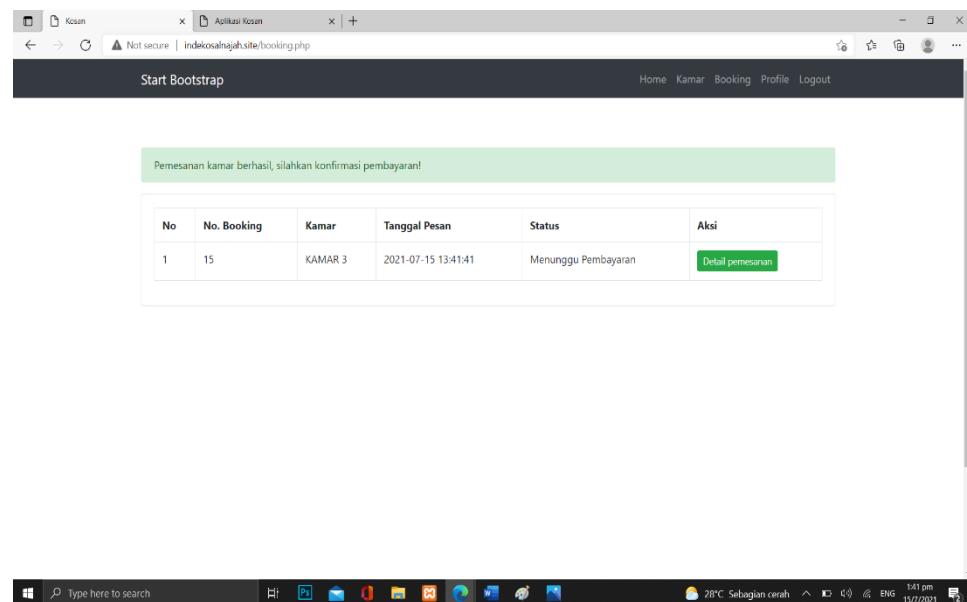
Halaman detail kamar memuat informasi mengenai gambar kamar, ketersediaan kamar, harga kamar perbulan, dan tombol pesan untuk mulai memesan kamar.



Gambar 5. 27 Tampilan halaman detail kamar

27) Halaman *Booking*

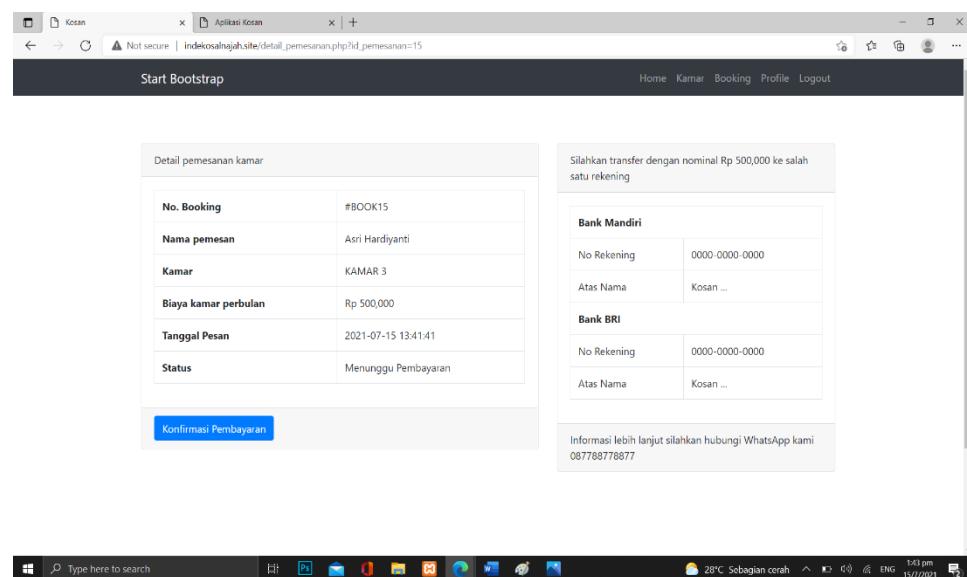
Pada halaman *Booking* terdapat informasi berisi no *Booking*, nama kamar, tanggal pesan, status pemesanan, dan tombol tombol detail pemesanan untuk melihat detail pemesanan yang telah dilakukan.



Gambar 5. 28 Tampilan halaman *Booking*

28) Halaman detail pemesanan

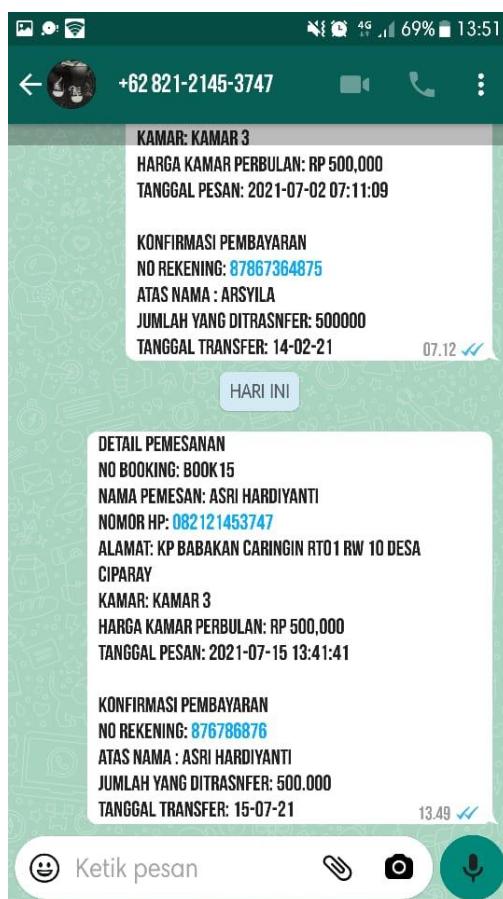
Pada halaman detail pemesanan memuat informasi mengenai data *Booking*, informasi rekening pengelola kamar indekos dan tombol konfirmasi pembayaran.



Gambar 5.29 Tampilan halaman Detail pemesanan kamar

29) Konfirmasi pembayaran di Whatsapp

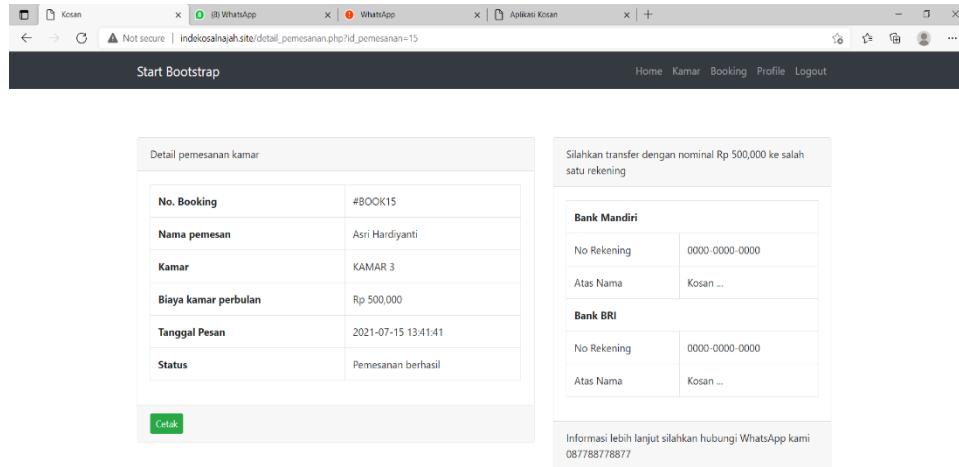
Setelah menekan tombol konfirmasi pembayaran di halaman detail pemesanan, pelanggan akan diarahkan ke aplikasi whatsapp untuk melakukan konfirmasi pembayaran langsung kepada admin, dengan mengisi form konfirmasi pembayaran dan mengirimkan bukti transfer. Jika bukti transfer sudah dikirimkan maka admin akan merubah status pemesanan apakah sudah berhasil atau gagal. Jika gagal pelanggan dapat mengulangi pemesanan atau mengkonfirmasi ulang pemesanannya melalui whatsapp.



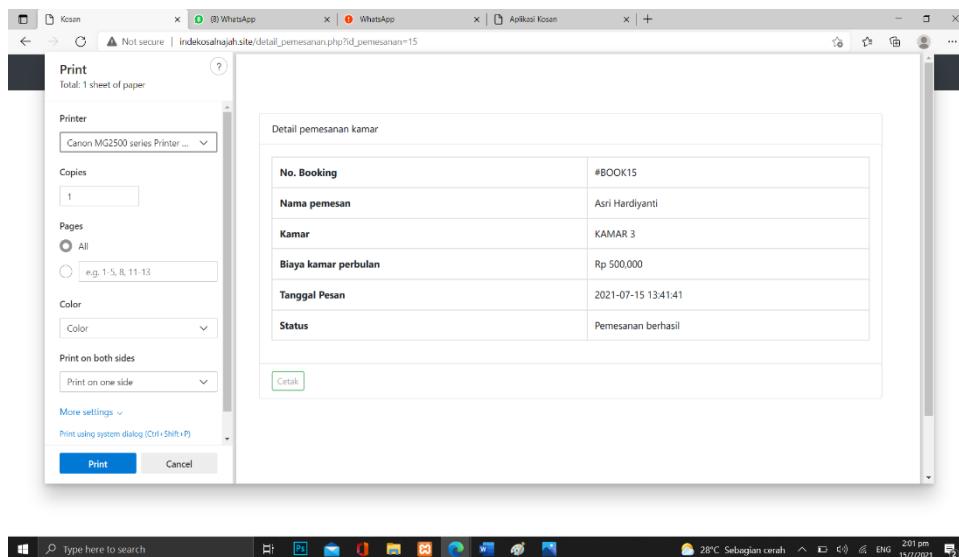
Gambar 5.30 Tampilan konfirmasi pembayaran di whatsapp

30) Halaman detail pemesanan

Pada halaman detail pemesanan terdapat informasi mengenai data pemesanan yang sudah berhasil dilakukan. Pelanggan juga dapat mencetaknya jika diperlukan.



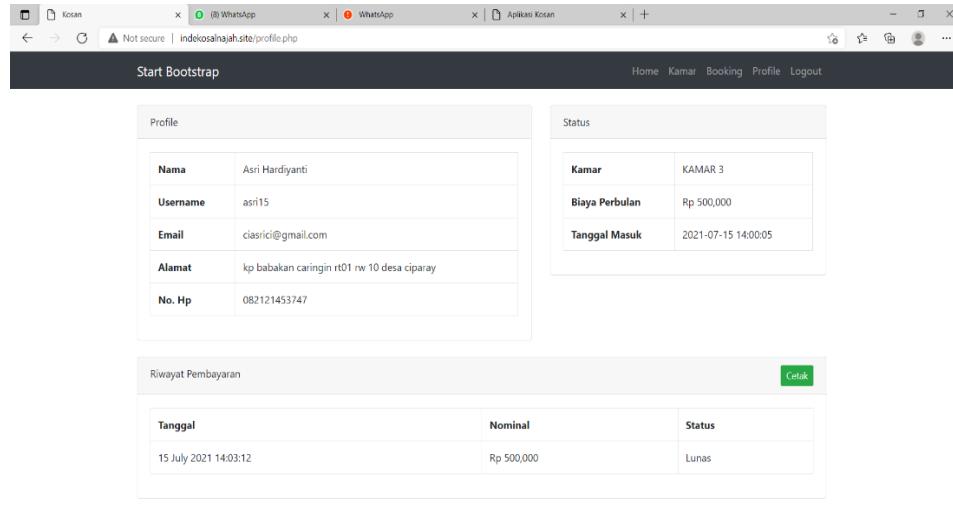
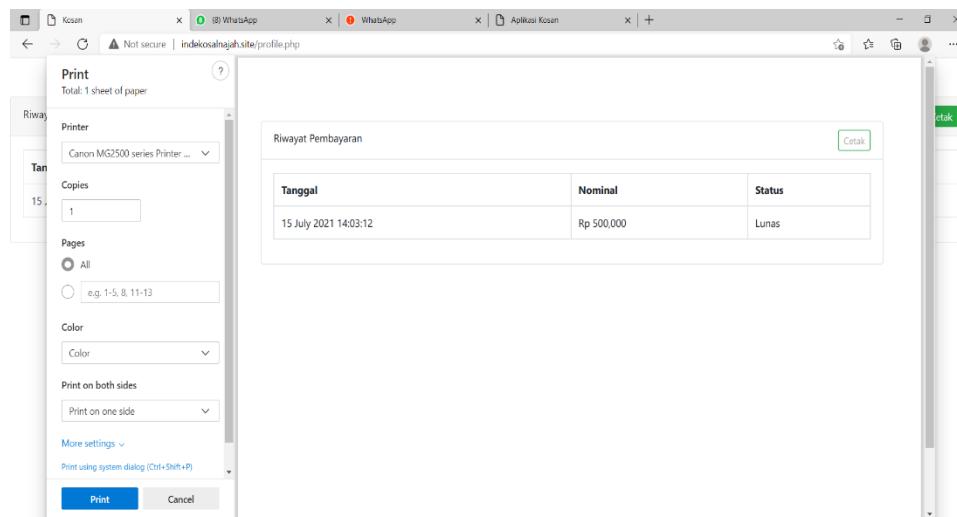
Gambar 5.31 Tampilan halaman detail pemesanan



Gambar 5.32 Tampilan halaman cetak detail pemesanan

31) Halaman *Profile*

Pada halaman *Profile* pelanggan berisi informasi mengenai profil atau data diri pelanggan, status kamar yang dipesan dan riwayat pembayaran yang dapat dicetak jika diperlukan.

Gambar 5.33 Tampilan halaman *Profile*

Gambar 5.34 Tampilan halaman cetak riwayat pembayaran

5.2 Pengujian Sistem

Sebelum diterapkan di Kamar indekos Al-Najah Ciparay, perlu adanya proses pengujian untuk menemukan kesalahan pada aplikasi. Pada tahap ini penulis menggunakan metode blackbox yaitu metode pengujian perangkat lunak dengan hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan diberikan untuk fungsi yang ada tanpa melihat bagaimana proses mendapatkan keluaran tersebut.

Berikut adalah hasil pengujian yang telah dilakukan oleh penulis:

5.2.1 Pengujian form Admin

1. Pengujian *Login*

Tabel 5.2 Pengujian *Login* admin

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|--|--|-----------------|
| 1 | Admin memasukkan link <i>indekosalnajah.net/administrator</i> di web browser | Masuk ke halaman <i>Login</i> | Berhasil |
| 2 | Admin memasukkan <i>Username</i> dan <i>Password</i> pada form <i>Login</i> | Masuk ke halaman <i>Dashboard</i> | Berhasil |
| 3 | Admin memasukkan <i>Username</i> dengan kondisi (salah) | Menampilkan pesan “ <i>Username</i> tidak ditemukan” | Berhasil |
| 4 | Admin memasukkan <i>Password</i> dengan kondisi (salah) | Menampilkan pesan “ <i>Password</i> salah” | Berhasil |

2. Pengujian menu admin

Tabel 5.3 Pengujian menu admin

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|--|---|-----------------|
| 1 | Admin menekan menu admin di halaman <i>Dashboard</i> | Masuk ke halaman admin dan menampilkan data admin | Berhasil |

| | | | |
|---|---|--|----------|
| 2 | Admin menekan tombol tambah | Masuk kehalaman tambah admin | Berhasil |
| 3 | Admin mengisi data admin | Menyimpan data admin, menampilkan pesan “data ditambahkan”, dan menampilkan tabel data admin | Berhasil |
| 4 | Admin menulis nama admin yang dicari di kolom pencarian | Menampilkan data admin yang dicari | Berhasil |
| 5 | Admin menulis nama admin dengan keadaan (salah) dikolom pencarian | Menampilkan pesan “tidak ada data” | Berhasil |
| 6 | Admin menekan tombol edit | Merubah data yang lama dengan data baru dan menampilkan pesan “data diupdate” | Berhasil |
| 7 | Admin menekan tombol hapus | Menampilkan pesan “are you sure?” (apakah anda yakin untuk menghapus data) | Berhasil |
| 8 | Admin menekan tombol “ok” untuk yakin menghapus data | Menghapus data dan menampilkan pesan “data terhapus” | Berhasil |

3. Pengujian menu kamar

Tabel 5.4 Pengujian menu kamar

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|---|--|-----------------|
| 1 | Admin menekan menu kamar di halaman <i>Dashboard</i> | Masuk ke halaman kamar dan menampilkan data kamar | Berhasil |
| 2 | Admin menekan tombol tambah | Masuk kehalaman tambah kamar | Berhasil |
| 3 | Admin mengisi data kamar | Menyimpan data kamar, menampilkan pesan “data ditambahkan”, dan menampilkan tabel data kamar | Berhasil |
| 4 | Admin menulis kata kunci yang dicari di kolom pencarian | Menampilkan data kamar yang dicari | Berhasil |
| 5 | Admin menulis kata kunci yang ingin dicari dengan keadaan (salah) dikolom pencarian | Menampilkan pesan “tidak ada data” | Berhasil |
| 6 | Admin menekan tombol edit | Merubah data yang lama dengan data baru dan menampilkan pesan “data diupdate” | Berhasil |
| 7 | Admin menekan tombol hapus | Menampilkan pesan “are you sure?” | Berhasil |

| | | | |
|---|--|--|----------|
| | | (apakah anda yakin untuk menghapus data) | |
| 8 | Admin menekan tombol “ok” untuk yakin menghapus data | Menghapus data dan menampilkan pesan “data terhapus” | Berhasil |

4. Pengujian menu pelanggan

Tabel 5.5 Pengujian menu pelanggan

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|---|--|-----------------|
| 1 | Admin menekan menu pelanggan di halaman <i>Dashboard</i> | Masuk ke halaman pelanggan dan menampilkan data pelanggan | Berhasil |
| 2 | Admin menekan tombol tambah | Masuk kehalaman tambah pelanggan | Berhasil |
| 3 | Admin mengisi data pelanggan | Menyimpan data pelanggan, menampilkan pesan “data ditambahkan”, dan menampilkan tabel data pelanggan | Berhasil |
| 4 | Admin menulis kata kunci yang dicari di kolom pencarian | Menampilkan data pelanggan yang dicari | Berhasil |
| 5 | Admin menulis kata kunci yang ingin dicari dengan keadaan (salah) dikolom pencarian | Menampilkan pesan “tidak ada data” | Berhasil |

| | | | |
|---|--|---|----------|
| 6 | Admin menekan tombol edit | Merubah data yang lama dengan data baru dan menampilkan pesan “data diupdate” | Berhasil |
| 7 | Admin menekan tombol hapus | Menampilkan pesan “are you sure?” (apakah anda yakin untuk menghapus data) | Berhasil |
| 8 | Admin menekan tombol “ok” untuk yakin menghapus data | Menghapus data dan menampilkan pesan “data terhapus” | Berhasil |
| 9 | Admin menekan tombol detail | Menampilkan data detail pelanggan | Berhasil |

5. Pengujian menu pemesanan

Tabel 5.6 Pengujian menu pemesanan

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|--|--|-----------------|
| 1 | Admin menekan menu pemesanan di halaman <i>Dashboard</i> | Masuk ke halaman pemesanan dan menampilkan data pemesanan | Berhasil |
| 2 | Admin menekan tombol tambah | Masuk kehalaman tambah pemesanan | Berhasil |
| 3 | Admin mengisi data pemesanan | Menyimpan data pemesanan, menampilkan pesan “data ditambahkan”, dan menampilkan tabel data pemesanan | Berhasil |

| | | | |
|---|---|--|----------|
| 4 | Admin menekan tombol ubah status | Merubah status pemesanan dan menampilkan pesan “data diupdate” | Berhasil |
| 5 | Admin menulis kata kunci yang dicari di kolom pencarian | Menampilkan data pemesanan yang dicari | Berhasil |
| 6 | Admin menulis kata kunci yang ingin dicari dengan keadaan (salah) dikolom pencarian | Menampilkan pesan “tidak ada data” | Berhasil |

6. Pengujian menu pembayaran

Tabel 5.7 pengujian menu pembayaran

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|---|--|-----------------|
| 1 | Admin menekan menu pembayaran di halaman <i>Dashboard</i> | Masuk ke halaman pembayaran dan menampilkan data pembayaran | Berhasil |
| 2 | Admin menekan tombol tambah | Masuk kehalaman tambah pembayaran | Berhasil |
| 3 | Admin mengisi data pembayaran | Menyimpan data pembayaran, menampilkan pesan “data ditambahkan”, dan menampilkan tabel data pembayaran | Berhasil |

| | | | |
|---|---|---|----------|
| 4 | Admin menulis kata kunci yang dicari di kolom pencarian | Menampilkan data pembayaran yang dicari | Berhasil |
| 5 | Admin menulis kata kunci yang ingin dicari dengan keadaan (salah) dikolom pencarian | Menampilkan pesan “tidak ada data” | Berhasil |
| 6 | Admin menekan tombol edit | Merubah data yang lama dengan data baru dan menampilkan pesan “data diupdate” | Berhasil |
| 7 | Admin menekan tombol hapus | Menampilkan pesan “are you sure?” (apakah anda yakin untuk menghapus data) | Berhasil |
| 8 | Admin menekan tombol “ok” untuk yakin menghapus data | Menghapus data dan menampilkan pesan “data terhapus” | Berhasil |

7. Pengujian menu Laporan

Tabel 5.8 pengujian menu laporan

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|--|---|-----------------|
| 1 | Admin menekan menu laporan di halaman <i>Dashboard</i> | Masuk ke halaman laporan dan menampilkan data laporan | Berhasil |

| | | | |
|---|---|--|----------|
| 2 | Admin menekan tombol cetak | Masuk ke halaman cetak | Berhasil |
| 3 | Admin mengisi tanggal dari dan sampai dan menekan tombol cari | Menampilkan data dari tanggal sekian sampai tanggal sekian | Berhasil |

8. Pengujian *Logout* Admin

Tabel 5.9 Pengujian *Logout* Admin

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|--|--|-----------------|
| 1 | admin memilih <i>Logout</i> di halaman <i>Home</i> | Menampilkan pesan “apa anda yakin ingin keluar” | Berhasil |
| 2 | Admin menekan tombol ok | Keluar dari <i>Dashboard</i> atau menu dan Kembali ke halaman <i>Login</i> | Berhasil |

5.2.2 Pengujian Form Pelanggan

1. Pengujian *Login* pelanggan

Tabel 5.10 Pengujian *Login* pelanggan

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|---|--|-----------------|
| 1 | User/pelanggan memasukkan link <u>indekosalnajah.net</u> di web browser | Masuk ke halaman <i>Home</i> dan menampilkan pilihan menu dan beberapa kamar | Berhasil |

| | | | |
|---|--|--|----------|
| 2 | Pelanggan memilih menu <i>Login</i> di halaman <i>Home</i> | Menampilkan form <i>Login</i> | Berhasil |
| 3 | Pelanggan memasukkan <i>Username</i> dan <i>Password</i> pada form <i>Login</i> | Masuk ke halaman <i>Home</i> | Berhasil |
| 4 | Pelanggan memasukkan <i>Username</i> dengan keadaan (salah) pada form <i>Login</i> | Menampilkan pesan “ <i>Username</i> tidak ditemukan” | Berhasil |
| 5 | Pelanggan memasukkan <i>Password</i> dengan kondisi (salah) pada form <i>Login</i> | Menampilkan pesan “ <i>Password</i> salah” | Berhasil |

2. Pengujian pendaftaran pelanggan

Tabel 5.11 Pengujian pendaftaran pelanggan

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|---|---|-----------------|
| 1 | User/pelanggan memilih tombol daftar disini di halaman <i>Login</i> | Masuk ke halaman pendaftaran | Berhasil |
| 2 | Pelanggan mengisi data diri di form pendaftaran dan menekan tombol daftar | Menampilkan form <i>Login</i> Kembali dan menampilkan pesan “berhasil daftar” | Berhasil |

| | | | |
|---|--|---|----------|
| 3 | Pelanggan mengisi data diri di form pendaftaran dengan keadaan data yang (tidak lengkap) dan menekan tombol daftar | Menampilkan pesan “ <i>please fill out this field</i> ” (Silahkan isi bidang ini) | Berhasil |
|---|--|---|----------|

3. Pengujian pemesanan kamar

Tabel 5.12 Pengujian menu kamar

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|---|--|-----------------|
| 1 | User/pelanggan memilih menu kamar di halaman <i>Home</i> | Masuk ke halaman kamar | Berhasil |
| 2 | Pelanggan menekan tombol detail kamar dihalaman kamar | Menampilkan halaman detail kamar | Berhasil |
| 3 | Pelanggan menekan tombol pesan sekarang di halaman detail kamar | Menampilkan pesan “ berhasil memesan, silahkan konfirmasi pembayaran” dan menampilkan halaman <i>Booking</i> | Berhasil |
| 4 | Pelanggan menekan tombol detail pemesanan di halaman <i>Booking</i> | Menampilkan halaman detail pemesanan dan informasi rekening kamar indekos | Berhasil |
| 5 | Pelanggan menekan tombol | Masuk ke aplikasi whatsapp | Berhasil |

| | | | |
|---|---|--|----------|
| | konfirmasi pembayaran | | |
| 6 | Pelanggan masuk kehalaman <i>Booking</i> untuk melihat status pemesanan “menunggu pembayaran” | Menampilkan status pemesanan “menunggu pembayaran” | Berhasil |
| 7 | Pelanggan masuk kehalaman <i>Booking</i> untuk melihat status pemesanan “pemesanan berhasil” | Menampilkan status pemesanan “pemesanan berhasil” | Berhasil |
| 8 | Pelanggan menekan tombol detail pemesanan | Menampilkan detail pemesanan yang berhasil | Berhasil |
| 9 | Pelanggan menekan tombol cetak detail pemesanan | Menampilkan halaman cetak | Berhasil |

4. Pengujian menu *Profile*

Tabel 5.13 Pengujian menu *Profile*

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|---|--|-----------------|
| 1 | User/pelanggan memilih menu <i>Profile</i> di halaman <i>Home</i> | Masuk ke halaman <i>Profile</i> dan menampilkan data <i>Profile</i> pelanggan, data kamar yang | Berhasil |

| | | | |
|---|--|--|----------|
| | | dipesan dan riwayat pembayaran | |
| 2 | User menekan tombol cetak di tabel riwayat pembayaran pada menu <i>Profile</i> | Menampilkan halaman cetak riwayat pembayaran | Berhasil |

5. Pengujian *Logout* pelanggan

Tabel 5.14 Pengujian *Logout* pelanggan

| No | Cara pengujian | Hasil yang diharapkan | Hasil pengujian |
|----|---|---|-----------------|
| 1 | User/pelanggan memilih <i>Logout</i> di halaman <i>Home</i> | Menampilkan pesan “apa anda yakin ingin keluar” | Berhasil |
| 2 | User/pelanggan menekan tombol ok | Keluar dari menu dan Kembali ke halaman <i>Home</i> | Berhasil |

BAB VI

KESIMPULAN

Berdasarkan hasil dari penelitian yang telah penyusun lakukan, penyusun mencoba membuat suatu kesimpulan dan mengajukan beberapa saran-saran yang berhubungan dengan pembahasan yang telah dikemukakan pada bab-bab sebelumnya.

6.1 Kesimpulan

Dari laporan yang telah diuraikan pada bab-bab sebelumnya, maka penulis menyimpulkan beberapa hal, diantaranya :

1. Aplikasi pengelolaan kamar indekos Al-Najah dapat mengelola seluruh data-data kamar indekos, sehingga tidak lagi memerlukan buku catatan karena seluruh data sudah tersimpan secara digital dalam aplikasi.
2. Kegiatan Promosi dan pemberian informasi untuk pelanggan menjadi lebih mudah karena informasi mengenai kamar indekos sudah dapat diakses melalui internet, sehingga pelanggan tidak perlu mendatangi langsung kamar indekos hanya untuk mendapatkan informasi
3. Pelanggan dapat dengan mudah memesan kamar indekos secara online.
4. Laporan keuangan menjadi lebih jelas, transparan, dan mudah untuk diberikan kepada pemilik kamar indekos.

6.2 Saran

Adapun saran yang penulis berikan untuk pengelolaan kamar indekos Al-najah Ciparay ini agar menjadi lebih baik, diantaranya : Kegiatan promosi harus lebih gencar dilakukan, contoh nya di promosikan di media sosial agar jangkauan promosinya lebih luas.

DAFTAR PUSTAKA

- Abdullah, R. (2016). *Easy & Simple Web Programming*. Elex Media Komputindo.
- Ajie, S. H., Amalia, F., & Widodo, A. W. (2020). *Pembangunan Sistem Pengelolaan Manajemen Rumah Indekos pada Indekos Semanggi di Kota Malang*. 4(2), 690–697.
- Budiman, A., Wahyuni, L. S., & Bantun, S. (2019). *Rumah Kos Berbasis Web (Studi Kasus : Kota Bandar Lampung)*. 13(2), 24–30.
- Dennis, Alan. et al. (2009). “*Systems Analysis Design with UML An Object-Oriented Approach Third Edition*”. p:159- 177.
- Dianawati, A. (2007). *Peluang Usaha rumahan paling menguntungkan*. MediaKita. https://books.google.co.id/books/about/Peluang_Usaha_Rumahan_yang_Menguntungkan.html?id=zgowZloEQGMC&redir_esc=y
- Faridl, M. (2015). Fitur Dasyat Sublime Text 3. *Lug Stikom*, 1–12. <http://lug.stikom.edu/wp-content/media/Fitur-Dahsyat-Sublime-Text-3.pdf>
- Hirin A.M dan Virgi. (2011). *Cepat Mahir Pemrograman Web Dengan PHP dan MySQL (Level Dasar Sampai Mahir)*. Jakarta: PT. Prestasi Pustakaraya.
- I Wayan Rendra Adiputra. (2020). Aplikasi manajemen kost pada kost kembar. *Aplikasi Manjemen Kost Pada Kost Kembar*.
- Kaban, R., & Fajrillah, F. (2017). Pengembangan Sistem Informasi Perpustakaan Dengan Framework Css Bootstrap Dan Web Development Life Cycle. *Jurnal Ilmiah Informatika*, 2(1), 83–89. <https://doi.org/10.35316/jimi.v2i1.454>
- Kawistara, P. H. J. K. (2017). *Pemrograman WEB* (revisi). Informatika Bandung.
- Lee, Z. (2017). *Arti Kata indekos, Makna, Pengertian dan Definisi - Kamus Besar Bahasa Indonesia (KBBI) Online*. <https://artikbbi.com/indekos/>
- M.Salahuddin, R. A. . &. (2018). *Rekayasa Perangkat Lunak Terstruktur dan berorientasi Objek*. Informatika.
- Nugroho, Bunafit. (2007). *Trik dan Rahasia Membuat Aplikasi Web dengan PHP*. Yogyakarta:Gava Media.
- Nur Elfi Husda, S. K. M. S. (2016). *Pengantar Teknologi Informasi*. (S. K. M. S) Eka Putria, Ed.) (Edisi Revi). Jakarta: Baduose Media.
- Perry, W. &. (1995). *Effective Methods For software Testing*.
- Pratama, B. (2012). *Sistem informasi dan Implementasi*. Informatika Bandung.
- Priansa, D. J. (2017). *Manajemen Pelayanan Prima Fokus pada organisasi publik dan peningkatan kualitas aparatur*. Alfabeta Bandung.
- Rachmawati, Ariefah, (2017), “*Membangun Informasi Layanan Umum Rumah Kos*

- Melalui Aplikasi Berbasis Web*”, Fakultas Ilmu Komputer, Universitas Mercu Buana, Jakarta Barat
- Ramadhan, Arief. (2006). *Pemrograman Web Database dengan PHP dan MySQL*. Jakarta:PT.Elex Media Komputindo
- Sibero, A. F. . (2011). *Kitab suci Web Programming*. Mediakom.
- Wibawanto. (2018). Bab Ii Landasan Teori. *Journal of Chemical Information and Modeling*, 53(9), 8–24.

LAMPIRAN

Lampiran 1 Kuesioner

Kuesioner

Aplikasi pengelolaan Kamar Indekos berbasis WEB di kamar indekos Al-Najah Ciparay

Nama : *Kustandi*
 Jabatan : *pengelola Kamar Indekos*

Petunjuk Pengisian : Berilah tanda checklist (✓) pada jawaban yang sesuai dengan pendapat anda.

Kriteria Penilaian : S (Setuju)
 N (Netral)
 TS (Tidak Setuju)

| No | Pernyataan | S | N | TS |
|----|---|---|---|----|
| 1 | Kamar indekos Al-Najah adalah usaha dalam bidang penyewaan tempat tinggal | ✓ | | |
| 2 | Kamar Indekos Al-Najah sudah memiliki 25 pintu kamar | ✓ | | |
| 3 | Pengelolaan kamar indekos Al-Najah masih dilakukan dengan cara konvensional | ✓ | | |
| 4 | Promosi yang dilakukan oleh pengelola kamar indekos Al-Najah belum maksimal | | ✓ | |
| 5 | Pencatatan data di kamar indekos Al-Najah masih dilakukan dengan cara menulis di buku catatan | ✓ | | |
| 6 | Transaksi pemesanan kamar Indekos Al-Najah masih dilakukan secara langsung | ✓ | | |
| 7 | Pelanggan mendatangi kamar indekos Al-Najah secara langsung untuk mendapatkan informasi mengenai ketersediaan kamar indekos | ✓ | | |
| 8 | Laporan keuangan dari pengelola kepada pemilik kamar indekos masih dilakukan dengan cara hanya memperlihatkan buku catatan | ✓ | | |
| 9 | Sering terjadi kesalahan pada pencatatan data, laporan keuangan dan lain-lain sehingga | | ✓ | |
| 10 | Buku catatan yang sudah habis menumpuk sehingga ruang penyimpanan menjadi penuh | ✓ | | |
| 11 | Buku catatan rentan rusak atau hilang | | ✓ | |

| | | | | |
|----|--|---|---|--|
| 12 | Direkomendasikan Aplikasi pengelolaan Kamar indekos berbasis WEB untuk membantu pengelolaan kamar indekos | ✓ | | |
| 13 | Pemberkasan catatan di basis data Komputer membantu proses pencatatan dengan mudah | ✓ | | |
| 14 | Penyimpanan pencatatan di basis data mengurangi tempat penyimpanan berkas | ✓ | | |
| 15 | Data kamar indekos lebih aman dan mudah dicari | | ✓ | |
| 16 | Promosi kamar indekos Al-Najah lebih gencar dan memperluas jangkauan promosinya | | ✓ | |
| 17 | Informasi mengenai kamar indekos lebih mudah didapatkan pelanggan karena dapat diakses melalui internet | | ✓ | |
| 18 | Pelanggan tidak perlu lagi mendatangi kamar indekos satu persatu untuk sekedar menanyakan ketersediaan dan fasilitas kamar | ✓ | | |
| 19 | Pemesanan dapat dilakukan secara online | ✓ | | |
| 20 | Pemesanan secara langsung masih dapat dilakukan oleh pengelola untuk antisipasi apabila ada pelanggan yang memesan secara langsung | ✓ | | |
| 21 | Laporan keuangan menjadi lebih transparan dan dapat dengan mudah disampaikan kepada pemilik kamar indekos | ✓ | | |
| 22 | Laporan keuangan di aplikasi berbasis web dapat meminimalisir kesalahan dan selisih | ✓ | | |
| 23 | Di aplikasi kamar indekos dapat memakai beberapa pengelola atau admin | ✓ | | |
| 24 | Terdapat fitur tambah,hapus,edit data di aplikasi untuk pengelola/admin | ✓ | | |
| 25 | Admin dapat merubah status pemesanan, apakah pemesanan yang dilakukan pelanggan berhasil, atau gagal | ✓ | | |
| 26 | Terdapat fitur pencarian untuk memudahkan admin mencari data yang diperlukan | ✓ | | |
| 27 | Terdapat fitur cetak data riwayat pembayaran di aplikasi pelanggan untuk mengetahui riwayat pembayaran yang pernah dilakukan | ✓ | | |
| 28 | Terdapat fitur pesan di aplikasi pelanggan sehingga pelanggan dapat melakukan pemesanan secara online | ✓ | | |
| 29 | Konfirmasi pembayaran diarahkan sistem ke aplikasi Whatsapp | ✓ | | |
| 30 | Hak akses pengelola dapat melakukan pengelolaan seluruh data kamar indekos | | ✓ | |
| 31 | Hak akses pelanggan dapat melakukan pemesanan, melihat data diri, melihat Riwayat pembayaran | ✓ | | |
| 32 | Hak akses pemilik hanya menerima laporan dari pengelola berupa print out | | ✓ | |

Lampiran 2 Dokumentasi

1. Gambar Indekos Al-Najah Ciparay (Tampak Luar)



2. Gambar Kamar Indekos Al-Najah Ciparay (Tampak Dalam)



3. Wawancara dengan Pemilik Kamar Indekos



4. Wawancara dengan Pengelola Kamar Indekos



Lampiran 3 *Source Code* Aplikasi

Halaman Utama

```
<?php
// session_start();
require_once 'function.php';
require "../templates/header.html";
require "../templates/navbar.html";
require "../templates/sidebar.html";
?>
<!-- wrapper -->
<!--
=====
===== -->
<div class="dashboard-wrapper">
    <div class="container-fluid dashboard-content">
        <?php if (isset($_SESSION['success'])) : ?>
            <div class='alert alert-success'>
                <?= $_SESSION['success']; ?>
            </div>
        <?php unset($_SESSION['success']);
        endif; ?>

        <?php if (isset($_SESSION['failed'])) : ?>
            <div class='alert alert-danger'>
                <?= $_SESSION['failed']; ?>
            </div>
        <?php unset($_SESSION['failed']);
        endif; ?>
        <div class='card'>
            <div class='card-header'>
                <div class='row'>
                    <div class='col-8'><a href='create.php'
class='btn btn-primary btn-sm'>Add</a></div>
                    <div class='col-4' align='right'>
                        <form action="" method='get'>
                            <input type='text'
name='search' class='form-control' placeholder='Search...'">
                        </form>
                    </div>
                </div>
            </div>
            <div class='card-body'>
                <div class='table-responsive'>
                    <table class='table table-bordered'>
                        <tr>
```

```

<th>No</th>
<th>username</th>
<th>password</th>
<th class='text-
center'>Action</th>
</tr>

<?php
if (isset($_GET['search'])) {
    $data =
GetBySearch($_GET['search']);
} else {
    $data = getAll();
}
$no = ($_GET['page'] > 1) ?
($_GET['page'] * 10) - 9 : 1;
?>

<?php if ($data) : ?>

<tbody class='tbody'>
<?php foreach ($data
as $td) : ?>
<tr>

<td><?= $no; ?></td>

<td><?= $td['username']; ?></td>

<td><?= $td['password']; ?></td>

<td
class='text-center'>

<form method='POST' action='edit.php' class='d-inline'>

<input type='hidden' name='id_admin' value='<?= $td['id_admin']; ?>'>

<input type='submit' name='edit' Value='Edit' class='btn btn-warning btn-
sm text-white'>

</form>

<form method='POST' action='function.php' class='d-inline'
onclick="return confirm('Are you sure?')">

<input type='hidden' name='id_admin' value='<?= $td['id_admin']; ?>'>

```

```

<input type='submit' name='delete' Value='Delete' class='btn btn-danger
btn-sm'>

</form>
</td>

</tr>
<?php $no++;
endforeach; ?>
</tbody>
<?php else : ?>
<td colspan='10' class='text-
center'>Tidak ada data</td>
<?php endif; ?>
</table>
</div>
</div>
<div class='card-footer'>
<nav aria-label='Page navigation example'>
<ul class='pagination'>
<?php for ($i = 1; $i <=
pagination()['total_page']; $i++) : ?>
<?php if ($i ==
pagination()['page']) : ?>
<li class='page-item
active'><a class='page-link' href='?page=<?= $i; ?>'><?= $i; ?></a></li>
<?php else : ?>
<li class='page-
item'><a class='page-link' href='?page=<?= $i; ?>'><?= $i; ?></a></li>
<?php endif; ?>
<?php endfor; ?>
</ul>
</nav>
</div>
</div>
<!--
=====
-->
<!-- end wrapper -->
<!--
=====
-->
<?php require "../templates/footer.html"; ?>

```

Tambah data kamar

```

<?php
// session_start();
require_once 'function.php';
require "../templates/header.html";
require "../templates/navbar.html";
require "../templates/sidebar.html";
?>

<div class="dashboard-wrapper">
    <div class="container-fluid dashboard-content">
        <div class='card'>
            <div class='card-header'>Create kamar</div>
            <div class='card-body'>
                <form action='function.php' method='post'
enctype="multipart/form-data">
                    <div class='form-group'>
                        <label for="nama_kamar">Nama
Kamar</label>
                        <input type='text'
name='nama_kamar' class='form-control'>
                    </div>
                    <div class='form-group'>
                        <label for="fasilitas">Fasilitas
<small>(pisahkan dengan koma per fasilitas)</small></label>
                        <input type='text' name='fasilitas'
class='form-control' placeholder="contoh: Tempat tidur, Kamar mandi, ...">
                    </div>
                    <div class='form-group'>
                        <label for="harga_perbulan">Harga
perbulan</label>
                        <input type='text'
name='harga_perbulan' class='form-control'>
                    </div>
                    <div class="form-group">
                        <label
for="gambar">Gambar</label>
                        <input type="file" name="gambar"
class="form-control">
                    </div>
                </div>
                <div class='card-footer'>
                    <button type='submit' name='insert' class='btn btn-primary btn-sm'>Submit</button>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </form>
    </div>
</div>

<?php require "../templates/footer.html"; ?>

```

Edit data Kamar

```

<?php
// session_start();
require_once 'function.php';
require "../templates/header.html";
require "../templates/navbar.html";
require "../templates/sidebar.html";
$id_kamar = $_POST['id_kamar'];
$datas = GetById($id_kamar);
?>

<div class="dashboard-wrapper">
    <div class="container-fluid dashboard-content">
        <div class='card'>
            <div class='card-header'>Edit kamar</div>
            <div class='card-body'>
                <form action='function.php' method='post'
enctype="multipart/form-data">
                    <input type='hidden'
name='id_kamar' value="<?php echo $_POST['id_kamar']; ?>">
                    <?php
                    foreach ($datas as $data) { ?>
                        <div class="form-group">
                            <label
for="nama_kamar"> Nama Kamar</label>
                            <input type="text"
class="form-control" id="nama_kamar" name='nama_kamar' value="<?php
echo $data['nama_kamar']; ?>">
                        </div>
                        <div class="form-group">
                            <label
for="fasilitas">Fasilitas</label>
                            <input type="text"
class="form-control" id="fasilitas" name='fasilitas' value="<?php echo
$data['fasilitas']; ?>">
                        </div>
                    <div class="form-group">

```

```

        <label
for="harga_perbulan"> Harga perbulan</label>
        <input type="text"
class="form-control" id="harga_perbulan" name="harga_perbulan"
value="<?php echo $data['harga_perbulan']; ?>">
        </div>

        <div class="form-group">
        <label
for="gambar">Gambar</label>
        <input type="file"
name="gambar" class="form-control">
        <input type="hidden"
name="gambar_lama" value="<?php echo $data['gambar']; ?>">
        <?php
if($data['gambar']) : ?>
        
        <?php endif; ?>
        </div>

        <?php } ?>
        </div>
        <div class='card-footer'>
        <button type='submit' name='update'
class='btn btn-primary btn-sm'>Update</button>
        </form>
        </div>
        </div>
    </div>
    <?php require "..//templates/footer.html"; ?>

```

Update data Kamar

```

function Update($id)
{
    $nama_kamar = $_POST['nama_kamar'];
    $fasilitas = $_POST['fasilitas'];
    $harga_perbulan = $_POST['harga_perbulan'];

    $nama_file = (( $_POST['gambar_lama'] ) &&
( $_POST['gambar_lama'] != " " || $_POST['gambar_lama'] != null)) ?
$_POST['gambar_lama'] : null;
    $gambar_lama = $_POST['gambar_lama'];

    if($_FILES['gambar']['name']){

```

```

$folder = '../assets/images/kamar/';
$nama_file = time()."_" . $_FILES['gambar']['name'];
$source = $_FILES['gambar']['tmp_name'];
unlink($folder.$gambar_lama);
move_uploaded_file($source, $folder . $nama_file);
}

$query = "UPDATE `kamar` SET `nama_kamar` =
'$nama_kamar', `fasilitas` = '$fasilitas', `harga_perbulan` = '$harga_perbulan',
`gambar` = '$nama_file' WHERE `id_kamar` = '$id'";
$exe = mysqli_query(connect(), $query);
if ($exe) {
    // kalau berhasil
    $_SESSION['success'] = " Data berhasil diubah! ";
    header("Location: index.php");
} else {
    $_SESSION['failed'] = " Data gagal diubah! ";
    header("Location: index.php");
}

```

Hapus data Kamar

```

function Delete($id)
{
    $query = "DELETE FROM `kamar` WHERE `id_kamar` =
$id";
    $exe = mysqli_query(connect(), $query);
    if ($exe) {
        // kalau berhasil
        $_SESSION['success'] = " Data berhasil dihapus! ";
        header("Location: index.php");
    } else {
        $_SESSION['failed'] = " Data gagal dihapus! ";
        header("Location: index.php");
    }
}

```

Detail data pelanggan

```

<?php
// session_start();
require_once 'function.php';
require "../templates/header.html";
require "../templates/navbar.html";
require "../templates/sidebar.html";

$namaBulan = array("01" => "Januari", "02" => "Februari", "03" => "Maret",
"04" => "April", "05" => "Mei", "06" => "Juni", "07" => "Juli", "08" =>
"Agustus", "09" => "September", "10" => "Oktober", "11" => "November",
"12" => "Desember");

$user_id = $_GET['user_id'];
$user = mysqli_query(connect(), "SELECT * FROM user WHERE id_user =
'$user_id'");
$row_user = mysqli_fetch_assoc($user);

$query_pembayaran = mysqli_query(connect(), "SELECT * FROM
pembayaran WHERE user_id = '$user_id' ORDER BY tanggal ASC");
$query_user_kamar = mysqli_query(connect(), "SELECT kamar.nama_kamar,
kamar.harga_perbulan, user_kamar.tanggal_masuk FROM user_kamar JOIN
kamar ON user_kamar.kamar_id = kamar.id_kamar WHERE user_id =
'$user_id'");
$row_user_kamar = mysqli_fetch_assoc($query_user_kamar);

$tanggal = [];
while ($row = mysqli_fetch_assoc($query_pembayaran)) {
    $tanggal[] = $row;
}

$tanggal_masuk = date('Y-m', strtotime($row_user_kamar['tanggal_masuk']));
$begin = new DateTime($tanggal_masuk);
$tanggal_sekarang = date('Y-m');
$end = new DateTime($tanggal_sekarang);

$j = 0;
for ($i = $begin; $i <= $end; $i->modify('+1 month')) {
    if ($i->format('Y-m') == date('Y-m', strtotime($tanggal[$j]['tanggal']))) {
        $data[] = [
            'tanggal' => date('m', strtotime($tanggal[$j]['tanggal'])),
            'nominal' => $tanggal[$j]['nominal'],
            'status' => 'Lunas'
        ];
    } else {
        $data[] = [
            'tanggal' => $i->format('m'),
            'nominal' => 0,
        ];
    }
}

```



```

$row_user['email']; ?></td>
</tr>
<tr>
    <th>Alamat</th>
    <td><?=

$row_user['alamat']; ?></td>
</tr>
<tr>
    <th>No. Hp</th>
    <td><?=

$row_user['hp']; ?></td>
</tr>
</table>
</div>
</div>
</div>

<div class="col-lg-5">
    <div class="card">
        <div class="card-header">Status</div>
        <div class="card-body">
            <table class="table table-bordered">
                <tr>
                    <th>Kamar</th>
                    <td><?=

$row_user_kamar['nama_kamar']; ?></td>
</tr>
<tr>
    <th>Biaya
Perbulan</th>
                    <td>Rp <?=

number_format($row_user_kamar['harga_perbulan']); ?></td>
</tr>
<tr>
    <th>Tanggal
Masuk</th>
                    <td><?= date('d F Y
H:i:s', strtotime($row_user_kamar['tanggal_masuk'])); ?></td>
</tr>
</table>
</div>
</div>

</div>
<div class="row">
    <div class="col-lg-12">

```

```

<div class="card">
    <div class="card-header">Detail
Pembayaran <?= date('Y'); ?></div>
    <div class="card-body">
        <table class="table table-bordered
table-sm">
            <tr>
                <th>Bulan</th>
                <th>Nominal</th>
                <th>Status</th>
            </tr>
            <?php foreach ($data as
$pembayaran) : ?>
                <tr>
                    <td><?=
$namaBulan[$pembayaran['tanggal']]; ?></td>
                    <td>Rp <?=
number_format($pembayaran['nominal'], 0, ',', ','); ?></td>
                    <td><?=
$pembayaran['status']; ?></td>
                </tr>
            <?php endforeach; ?>
        </table>
    </div>
</div>
</div>
</div>

<!--
=====
===== -->
<!-- end wrapper -->
<!--
=====
===== -->
<?php require "../templates/footer.html"; ?>

```

Ubah Status pemesanan

```

<?php
// session_start();
require_once 'function.php';
require "../templates/header.html";
require "../templates/navbar.html";
require "../templates/sidebar.html";
$id_pemesanan = $_POST['id_pemesanan'];
$datas = GetById($id_pemesanan);
?>

<div class="dashboard-wrapper">
    <div class="container-fluid dashboard-content">
        <div class='card'>
            <div class='card-header'>Detail pemesanan kamar</div>
            <div class='card-body'>
                <form action='function.php' method='post'>
                    <input type='hidden' name='id_pemesanan' value="<?php echo
$_POST['id_pemesanan']; ?>">
                    <?php
                    foreach ($datas as $data) { ?>
                        <div class="form-group">
                            <label for="user"> Nama:</label>
                            <input type="text" class="form-control" id="user" name='user'
value="<?php echo $data['nama']; ?>" readonly>
                        </div>
                        <input type="hidden" name="user_id" value="<?=
$data['user_id']; ?>">
                        <input type="hidden" name="kamar_id" value="<?=
$data['kamar_id']; ?>">
                        <input type="hidden" name="harga_perbulan" value="<?=
$data['harga_perbulan']; ?>">
                        <div class="form-group">
                            <label for="kamar"> Kamar:</label>
                            <input type="text" class="form-control" id="kamar"
name='kamar' value="<?php echo $data['nama_kamar']; ?>" readonly>
                        </div>
                        <div class="form-group">
                            <label for="tanggal_pesan"> tanggal_pesan:</label>
                            <input type="text" class="form-control" id="tanggal_pesan"
name='tanggal_pesan' value="<?php echo $data['tanggal_pesan']; ?>" readonly>
                        </div>
                    <div class="form-group">

```

```

<label for="status"> status:</label>
<select class="form-control" id="status" name='status'>
    <option value="" selected disabled>- Pilih status pemesanan
-</option>
    <option value="0">Menunggu Pembayaran</option>
    <option value="1">Sedang diverifikasi</option>
    <option value="2">Pemesanan Berhasil</option>
    <option value="3">Pemesanan Gagal / Berhenti
langganan</option>
</select>
</div>

<?php } ?>
</div>
<div class='card-footer'>
    <button type='submit' name='update' class='btn btn-primary btn-sm'>Update</button>
    </form>
</div>
</div>
</div>

<?php require "../templates/footer.html"; ?>

```

Booking pelanggan

```

<?php
require 'templates/header.php';
require 'config/get_connection.php';

if (!isset($_SESSION['user'])) {
    $_SESSION['failed'] = 'Anda harus login dulu!';
    header('Location: login.php');
}

$status = [
    0 => 'Menunggu Pembayaran',
    1 => 'Pembayaran sedang diverifikasi',
    2 => 'Pemesanan berhasil',
    3 => 'Pemesanan gagal',
];
$id_user = $_SESSION['user']['id_user'];
$query = mysqli_query(connect(), "SELECT pemesanan.*, kamar.nama_kamar
FROM pemesanan JOIN kamar ON pemesanan.kamar_id = kamar.id_kamar
WHERE user_id = '$id_user'");

?>

```

```

<div class="container mt-4 mb-5">
    <div class="row mt-5">
        <div class="col-lg-12">
            <?php if (isset($_SESSION['success'])) : ?>
                <div class='alert alert-success'>
                    <?= $_SESSION['success']; ?>
                </div>
            <?php unset($_SESSION['success']);
            endif; ?>
            <div class="card">
                <div class="card-body">
                    <div class="table-responsive">
                        <table class="table table-bordered">
                            <tr>
                                <th>No</th>
                                <th>No. Booking</th>
                                <th>Kamar</th>
                                <th>Tanggal Pesan</th>
                                <th>Status</th>
                                <th>Aksi</th>
                            </tr>
                            <?php if (mysqli_num_rows($query) == 0) : ?>
                                <td class="text-center" colspan="6">Tidak ada data</td>
                            <?php endif; ?>
                            <?php $no = 1;
                            while ($row = mysqli_fetch_assoc($query)) : ?>
                                <tr>
                                    <td><?= $no; ?></td>
                                    <td><?= $row['id_pemesanan']; ?></td>
                                    <td><?= $row['nama_kamar']; ?></td>
                                    <td><?= $row['tanggal_pesan']; ?></td>
                                    <td><?= $status[$row['status']]; ?></td>
                                    <td>
                                        <a href="detail_pemesanan.php?id_pemesanan=<?=
                                        $row['id_pemesanan']; ?>" class="btn btn-success btn-sm">Detail
                                        pemesanan</a>
                                    </td>
                                </tr>
                            <?php $no++;
                            endwhile; ?>
                        </table>
                    </div>
                </div>
            </div>
        </div>
    </div>
<?php require 'templates/footer.html'; ?>

```

Daftar pelanggan baru

```

<?php
require 'templates/header.php';
require 'config/get_connection.php';
?>

<!-- Page Content -->
<div class="container">
<br>
<div class="row justify-content-center mt-5 mb-5">
<div class="col-lg-8">

<?php if (isset($_SESSION['success'])) : ?>
<div class='alert alert-success'>
<?= $_SESSION['success']; ?>
</div>
<?php unset($_SESSION['success']);
endif; ?>

<?php if (isset($_SESSION['failed'])) : ?>
<div class='alert alert-danger'>
<?= $_SESSION['failed']; ?>
</div>
<?php unset($_SESSION['failed']);
endif; ?>

<div class="card">
<div class="card-header">Daftar</div>
<div class="card-body">
<form action="administrator/pelanggan/function.php"
method="post">
<input type="hidden" name="daftar" value="daftar">
<div class="form-group">
<label for="nama">Nama Lengkap</label>
<input type="text" class="form-control" name="nama"
required>
</div>

<div class="form-group">
<label for="alamat">Alamat</label>
<textarea class="form-control" name="alamat"
required></textarea>
</div>

<div class="form-group">
<label for="hp">No. HP</label>

```

```

        <textarea class="form-control" name="hp"
required></textarea>
</div>

        <div class="form-group">
            <label for="status">Status</label>
            <input type="text" class="form-control" name="status"
required>
        </div>

        <div class="form-group">
            <label for="email">Email</label>
            <input type="email" class="form-control" name="email"
required>
        </div>

        <div class="form-group">
            <label for="username">Username</label>
            <input type="text" class="form-control" name="username"
required>
        </div>

        <div class="form-group">
            <label for="password">Password</label>
            <input type="password" class="form-control"
name="password" required>
        </div>

        <button type="submit" name="insert" class="btn btn-primary
btn-sm">Daftar</button>
        <span class="float-right">Sudah punya akun? <a
href="login.php">Login disini</a></span>
    </form>
</div>
</div>
</div>
</div>

</div>
<!-- /.container -->
<br><br><br><br>
<?php require 'templates/footer.html'; ?>
```

Detail Pemesanan

```

<?php
require 'templates/header.php';
require 'config/get_connection.php';

if (!isset($_SESSION['user'])) {
    $_SESSION['failed'] = 'Anda harus login dulu!';
    header('Location: login.php');
}

$status = [
    0 => 'Menunggu Pembayaran',
    1 => 'Pembayaran sedang divalidasi',
    2 => 'Pemesanan berhasil',
    3 => 'Pemesanan gagal',
];

$id_pemesanan = $_GET['id_pemesanan'];
$query = mysqli_query(connect(), "SELECT pemesanan.*, kamar.nama_kamar,
kamar.harga_perbulan, user.nama, user.alamat, user.hp FROM pemesanan JOIN
kamar ON pemesanan.kamar_id = kamar.id_kamar JOIN user ON
pemesanan.user_id = user.id_user WHERE id_pemesanan = '$id_pemesanan'");
$row = mysqli_fetch_assoc($query);
?>
<div class="container mt-4 mb-5">
    <div class="row mt-5">
        <div class="col-lg-7">
            <?php if (isset($_SESSION['success'])) : ?>
                <div class='alert alert-success'>
                    <?= $_SESSION['success']; ?>
                </div>
            <?php unset($_SESSION['success']); ?>
            <endif; ?>
            <div id="printThis">
                <div class="card">
                    <div class="card-header">Detail pemesanan kamar</div>
                    <div class="card-body">
                        <div class="table-responsive">
                            <table class="table table-bordered">
                                <tr>
                                    <th>No. Booking</th>
                                    <td>#BOOK<?= $row['id_pemesanan']; ?></td>
                                </tr>
                                <tr>
                                    <th>Nama pemesan</th>
                                    <td><?= $row['nama']; ?></td>
                                </tr>
                                <tr>

```

```

<th>Kamar</th>
<td><?= $row['nama_kamar']; ?></td>
</tr>
<tr>
    <th>Biaya kamar perbulan</th>
    <td>Rp <?= number_format($row['harga_perbulan']); ?></td>
?></td>
</tr>
<tr>
    <th>Tanggal Pesan</th>
    <td><?= $row['tanggal_pesan']; ?></td>
</tr>
<tr>
    <th>Status</th>
    <td><?= $status[$row['status']]; ?></td>
</tr>

</table>
</div>
</div>
<div class="card-footer">
    <?php if ($row['status'] == 0 || $row['status'] == 3) : ?>
        <a
            href="https://api.whatsapp.com/send?phone=+6282121453747&text=Detail
            pemesanan %0ANo Booking: BOOK<?php echo $row['id_pemesanan'];
            ?>%0ANama pemesan: <?php echo $row['nama']; ?>%0ANomor HP: <?php
            echo $row['hp']; ?>%0AAlamat: <?php echo $row['alamat']; ?>%0AKamar:
            <?php echo $row['nama_kamar']; ?>%0AHarga Kamar perbulan: Rp <?php
            echo number_format($row['harga_perbulan']); ?>%0ATanggal Pesan: <?php
            echo $row['tanggal_pesan']; ?>%0A%0AKonfirmasi Pembayaran%0ANo
            Rekening: %0AAatas Nama : %0AJumlah yang ditrasnfer: %0ATanggal
            Transfer: %0A
                " class="btn
                btn-primary text-center" target="_blank">Konfirmasi Pembayaran</a>
        <?php endif; ?>

        <?php if ($row['status'] == 2) : ?>
            <button onclick="printDiv('printThis')" id="btnPrint"
            class="btn btn-success btn-sm">Cetak</button>
        <?php endif; ?>
    </div>

    </div>
    </div>
    <br>
</div>
<div class="col-md-5">
    <div class="card">

```

```

<div class="card-header">Silahkan transfer dengan nominal Rp <?= number_format($row['harga_perbulan']); ?> ke salah satu rekening</div>
<div class="card-body">
    <table class="table table-bordered">
        <tr>
        <tr>
            <th colspan="2">Bank Mandiri</th>
        </tr>
        <tr>
            <td>No Rekening</td>
            <td>0000-0000-0000</td>
        </tr>
        <tr>
            <td>Atas Nama</td>
            <td>Kosan ...</td>
        </tr>
        <tr>
        <tr>
            <th colspan="2">Bank BRI</th>
        </tr>
        <tr>
            <td>No Rekening</td>
            <td>0000-0000-0000</td>
        </tr>
        <tr>
            <td>Atas Nama</td>
            <td>Kosan ...</td>
        </tr>
        </table>
    </div>
    <div class="card-footer">
        Informasi lebih lanjut silahkan hubungi WhatsApp kami
        087788778877
    </div>
    </div>
    </div>
    </div>
</div>
<?php require 'templates/footer.html'; ?>
<script>
    function printDiv(divName) {
        var printContents = document.getElementById(divName).innerHTML;
        var originalContents = document.body.innerHTML;

        document.body.innerHTML = printContents;

```

```

var css = '@page { size: landscape; }',
    head = document.head || document.getElementsByTagName('head')[0],
    style = document.createElement('style');

style.type = 'text/css';
style.media = 'print';

if (style.styleSheet) {
    style.styleSheet.cssText = css;
} else {
    style.appendChild(document.createTextNode(css));
}

head.appendChild(style);
window.print();

document.body.innerHTML = originalContents;
}
</script>

```

Profile Pelanggan

```

<?php
require 'templates/header.php';
require 'config/get_connection.php';

$id_user = $_SESSION['user']['id_user'];
$user = mysqli_query(connect(), "SELECT * FROM user WHERE id_user = '$id_user'");
$row = mysqli_fetch_assoc($user);
$user_kamar = mysqli_query(connect(), "SELECT kamar.nama_kamar,
kamar.harga_perbulan, user_kamar.tanggal_masuk FROM user_kamar JOIN
kamar ON user_kamar.kamar_id = kamar.id_kamar WHERE user_id = '$id_user'");
$row_user_kamar = mysqli_fetch_assoc($user_kamar);
$pembayaran = mysqli_query(connect(), "SELECT * FROM pembayaran
WHERE user_id = '$id_user' ORDER BY tanggal DESC");

?>

<!-- Page Content -->
<div class="container">
<!-- Page Features -->
<div class="row my-4">

<div class="col-lg-7">
<div class="card">
<div class="card-header">

```

```

Profile
</div>
<div class="card-body">
<table class="table table-bordered">
<tr>
<th>>Nama</th>
<td><?= $row['nama']; ?></td>
</tr>
<tr>
<th>Username</th>
<td><?= $row['username']; ?></td>
</tr>
<tr>
<th>Email</th>
<td><?= $row['email']; ?></td>
</tr>
<tr>
<th>Alamat</th>
<td><?= $row['alamat']; ?></td>
</tr>
<tr>
<th>No. Hp</th>
<td><?= $row['hp']; ?></td>
</tr>
</table>
</div>
</div>
</div>

<div class="col-lg-5">
<div class="card">
<div class="card-header">Status</div>
<div class="card-body">
<table class="table table-bordered">
<tr>
<th>Kamar</th>
<td><?= $row_user_kamar['nama_kamar']; ?></td>
</tr>
<tr>
<th>Biaya Perbulan</th>
<td>Rp <?= number_format($row_user_kamar['harga_perbulan']); ?></td>
</tr>
<tr>
<th>Tanggal Masuk</th>
<td><?= $row_user_kamar['tanggal_masuk']; ?></td>
</tr>
</table>

```

```

        </div>
        </div>
        </div>

    </div>

    <div class="row">
        <div class="col-lg-12">
            <div id="printPembayaran">
                <div class="card">
                    <div class="card-header">
                        Riwayat Pembayaran
                        <div align="right" class="float-right">
                            <button onclick="printDiv('printPembayaran')" id="btnPrint"
                                class="btn btn-success btn-sm">Cetak</button>
                        </div>
                    </div>
                    <div class="card-body">
                        <table class="table table-bordered">
                            <tr>
                                <th>Tanggal</th>
                                <th>Nominal</th>
                                <th>Status</th>
                            </tr>
                            <?php while ($row_pembayaran = mysqli_fetch_assoc($pembayaran))
                            : ?>
                            <tr>
                                <td><?= date('d F Y H:i:s', strtotime($row_pembayaran['tanggal']));
                            ?></td>
                                <td>Rp <?= number_format($row_pembayaran['nominal']); ?></td>
                                <td><?= ($row_pembayaran['status'] == 1) ? 'Lunas' : ''; ?></td>
                            </tr>
                            <?php endwhile; ?>
                        </table>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <!-- /.row -->

</div>
<!-- /.container -->

<?php require 'templates/footer.html'; ?>
<script>
    function printDiv(divName) {
        var printContents = document.getElementById(divName).innerHTML;

```

```

var originalContents = document.body.innerHTML;

document.body.innerHTML = printContents;
var css = '@page { size: landscape; }',
head = document.head || document.getElementsByTagName('head')[0],
style = document.createElement('style');

style.type = 'text/css';
style.media = 'print';

if (style.styleSheet) {
  style.styleSheet.cssText = css;
} else {
  style.appendChild(document.createTextNode(css));
}

head.appendChild(style);
window.print();

document.body.innerHTML = originalContents;
}
</script>

```

Lihat dan cetak Laporan

```

<?php
// session_start();
require_once '../pembayaran/function.php';
require "../templates/header.html";
require "../templates/navbar.html";
require "../templates/sidebar.html";
?>
<!-- wrapper -->
<!--
=====
===== -->
<div class="dashboard-wrapper">
  <div class="container-fluid dashboard-content">
    <div class="row">
      <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-12">
        <div class="page-header">
          <h2 class="pageheader-title">Laporan</h2>
        </div>
      </div>
    </div>
    <?php if (isset($_SESSION['success'])) : ?>
      <div class='alert alert-success'>
        <?= $_SESSION['success']; ?>
      </div>
    </?php >
  </div>
</div>

```

```

</div>
<?php unset($_SESSION['success']);
endif; ?>

<?php if (isset($_SESSION['failed'])) : ?>
<div class='alert alert-danger'>
<?= $_SESSION['failed']; ?>
</div>
<?php unset($_SESSION['failed']);
endif; ?>
<div class='card'>
<div class='card-header'>
<form action="" method="GET">
<div class="row">
<div class="col-lg-4">
<label for="">Dari</label>
<input type="date" name="dari" class="form-control">
</div>
<div class="col-lg-4">
<label for="">Sampai</label>
<input type="date" name="sampai" class="form-control">
</div>
<div class="col-lg-4 mt-2">
<br>
<button type="submit" name="date" class="btn btn-primary
btn-sm">Cari</button>
<button onclick="printDiv('printThis')" id="btnPrint"
class="btn btn-success btn-sm">Cetak</button>
</div>
</div>
</form>
</div>
<div class='card-body'>
<h3 id="titleLaporan" style="display: none;">Laporan
Pembayaran</h3>
<div class='table-responsive' id="printThis">
<table class='table table-bordered'>
<tr>
<th>No</th>
<th>Pelanggan</th>
<th>Kamar</th>
<th>Tanggal</th>
<th>Nominal</th>
<th>Status</th>
</tr>
<?php
if (isset($_GET['date'])) {

```

```

        $data = GetByDate();
    } else {
        $data = getAll();
    }
$no = ($_GET['page'] > 1) ? ($_GET['page'] * 10) - 9 : 1;
?>

<?php if ($data) : ?>

<tbody class='tbody'>
    <?php $total = 0;
    foreach ($data as $td) :
        $total += $td['nominal'];
    ?>
    <tr>
        <td><?= $no; ?></td>
        <td><?= $td['nama']; ?></td>
        <td><?= $td['nama_kamar']; ?></td>
        <td><?= date('d F Y H:i:s', strtotime($td['tanggal']));
?></td>
        <td class="text-right">Rp <?= number_format($td['nominal']); ?></td>
        <td><?= ($td['status'] == 1) ? 'Lunas' : ''; ?></td>
    </tr>
    <?php $no++;
endforeach; ?>
    <tr>
        <th colspan="4" class="text-center">Total
pembayaran</th>
        <td class="text-right"><?= number_format($total);
?></td>
        <td></td>
    </tr>
</tbody>
<?php else : ?>
    <td colspan='10' class='text-center'>Tidak ada data</td>
    <?php endif; ?>
</table>
</div>
</div>
<div class='card-footer'>
    <nav aria-label='Page navigation example'>
        <ul class='pagination'>
            <?php for ($i = 1; $i <= pagination()['total_page']; $i++) : ?>
                <?php if ($i == pagination()['page']) : ?>
                    <li class='page-item active'><a class='page-link'
href='?page=<?= $i; ?>'><?= $i; ?></a></li>
                <?php else : ?>

```

```

<li class='page-item'><a class='page-link' href=?page=<?=$i; ?>><?= $i; ?></a></li>
    <?php endif; ?>
    <?php endfor; ?>
</ul>
</nav>
</div>
</div>
</div>

<!--
=====
===== -->
<!-- end wrapper -->
<!--
=====
===== -->
<?php require "../templates/footer.html"; ?>
<script>
    function printDiv(divName) {
        var printContents = document.getElementById(divName).innerHTML;
        var originalContents = document.body.innerHTML;

        document.body.innerHTML = printContents;
        var css = '@page { size: portrait; }',
            head = document.head ||
        document.getElementsByTagName('head')[0],
            style = document.createElement('style');

        style.type = 'text/css';
        style.media = 'print';

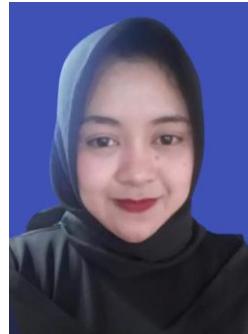
        if (style.styleSheet) {
            style.styleSheet.cssText = css;
        } else {
            style.appendChild(document.createTextNode(css));
        }

        head.appendChild(style);
        window.print();

        document.body.innerHTML = originalContents;
    }
</script>

```

Lampiran 4 Daftar Riwayat Hidup Penulis



Asri Hardiyanti, Lahir di Bandung, pada tanggal 15-Maret-1998, anak ke 5 dari 8 bersaudara. Memulai Pendidikan di Madrasah Ibtidaiyah (MI) AT-TAQWA Ciparay pada tahun (2003-2009), setelah itu melanjutkan Pendidikan di Madrasah Tsanawiyah (MTs) AT-TAQWA Ciparay pada tahun (2009-2012), setelah lulus diteruskan di SMK ITIKURIH HIBARNA pada tahun (2012-2015) dengan mengambil jurusan Teknik Komputer Jaringan (TKJ). Setelah lulus dari SMK ITIKURIH HIBARNA, penulis tidak langsung melanjutkan perguruan tinggi, penulis berkerja terlebih dahulu selama kurang lebih 1 tahun di PT Metro Garmin, setelah itu penulis *resign*, dan memutuskan untuk melanjutkan Pendidikan jenjang Strata 1 (S1) di UNIVERSITAS BALE BANDUNG (UNIBBA) tahun (2016-2021) dengan mengambil jurusan Teknologi Informasi dengan program studi Teknik Informatika, penulis sempat cuti selama 1 semester dikarenakan melahirkan. Penulis juga aktif dalam organisasi kampus EPOCH DIGITAL TECHNOLOGI dan menjabat sebagai sekertaris.