

**PEMBANGUNAN SISTEM PENGELOLAAN MANAJEMEN
RUMAH INDEKOS PADA INDEKOS SEMANGGI DI KOTA
MALANG**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Susiawan Hastomo Ajie
NIM: 135150207111023



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

**PEMBANGUNAN SISTEM PENGELOLAAN MANAJEMEN RUMAH INDEKOS PADA
INDEKOS SEMANGGI DI KOTA MALANG**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer**

Disusun Oleh :
Susiawan Hastomo Ajie
NIM: 135150207111023

Skripsi ini telah diuji dan dinyatakan lulus pada

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2

Faizatul Amalia, S.Pd.,M.Pd
NIK: 201309 860821 2 001

Agus Wahyu Widodo, S.T, M.Cs.,
NIP: 19740805 200112 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D.,
NIP: 19710515 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 2 Juli 2019

Susiawan Hastomo Ajie

NIM: 135150207111023

PRAKATA

Puji syukur penulis panjatkan atas kehadiran Allah SWT. Karena berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan penyusunan skripsi dengan judul “Pembangunan Sistem Pengelolaan Manajemen Rumah Indekos Pada Indekos Semanggi di Kota Malang” dengan cukup baik.

Dalam Penulisan skripsi ini, penulis banyak mendapatkan bantuan, bimbingan serta dukungan baik materil maupun secara moril dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

1. Orang tua penulis, Bapak Subagio dan Ibu Suswati serta keluarga besar yang selalu memberikan dukungan dan doa untuk kelancaran kuliah penulis.
2. Ibu Faizatul Amalia, S.Pd.,M.Pd selaku dosen pembimbing I dan Bapak Agus Wahyu Widodo, S.T, M.Cs., selaku dosen pembimbing II yang selalu membimbing, membantu serta meluangkan waktu dan pikiran untuk penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D., selaku Dekan Fakultas Ilmu Komputer.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D., selaku Ketua Jurusan Teknik Informatika.
5. Bapak Agus Wahyu Widodo, S.T, M.Cs., selaku Ketua Program Studi Teknik Informatika.
6. Seluruh dosen dan civitas Program Studi Teknik Informatika, Universitas Brawijaya atas dukungan dan kerjasamanya.
7. Sahabat dan teman seperjuangan Fakultas Ilmu Komputer maupun diluar Fakultas Ilmu Komputer dalam menyelesaikan penulisan skripsi lainnya yang tidak dapat disebutkan satu persatu.

Malang, 2 Juli 2019

Susiawan Hastomo Ajie
Hastomo.ajie@gmail.com

ABSTRAK

Susiawan Hastomo Ajie, Pembangun Sistem Pengelolaan Manajemen Rumah Indekos Pada Rumah Indekos Semanggi Di Kota Malang

Pembimbing: Faizatul Amalia, S.Pd.,M.Pd dan Agus Wahyu Widodo, S.T, M.Cs.,

Rumah Indekos merupakan rumah yang digunakan untuk tinggal sementara di rumah orang lain dengan membayar uang sewa setiap bulannya. Seiring bertambahnya mahasiswa atau para pendatang yang berdatangan di kota Malang, tidak menuntut kemungkinan pemilik indekos ingin menambah rumah indekos serta menambah jumlah kamar lagi untuk menampung para pendatang. Pendataan manajemen rumah indekos menggunakan aplikasi *spreadsheet* yaitu Microsoft Excel. Namun pada saat melakukan pendataan, mendapatkan permasalahan diantaranya membutuhkan waktu yang lama mencari posisi data. Kesalahan dalam pembaruan data tersebut sering terjadi pada posisi kolom dan baris yang salah. *File spreadsheet* mengalami *corrupt*, sehingga data tidak dapat diakses dan harus melakukan *entry* ulang data. Permasalahan lainnya yang dihadapi adalah pemilik indekos sering mengalami keterlambatan atau bahkan lupa untuk memperbarui data terhadap penghuni indekos setiap kamarnya. Masalah keterlambatan pembayaran penghuni indekos setiap bulannya. Permasalahan dari sisi penghuni pada saat melakukan pembayaran, tidak adanya kwitansi yang digunakan sebagai bukti pembayaran. Penghuni indekos dibebankan untuk membayar lagi pada bulan yang sama. Solusi yang diberikan untuk mengatasi permasalahan-permasalahan tersebut, pembuatan sistem pengelolaan manajemen rumah indekos dengan fitur untuk 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*.

Kata Kunci: pengelolaan manajemen rumah indekos, rumah indekos, rekayasa perangkat lunak, operasional, *waterfall* model.

ABSTRACT

Susiawan Hastomo Ajie, Pembangun Sistem Pengelolaan Manajemen Rumah Indekos Pada Rumah Indekos Semanggi Di Kota Malang

Pembimbing: Faizatul Amalia, S.Pd.,M.Pd dan Agus Wahyu Widodo, S.T, M.Cs.,

Boarding House is a temporary place to live which belongs to the owner and has to be monthly paid as the rent. Along with the increase of students or outsiders that soon live in Malang, does not close up the possibility that boarding house owners want to build other boarding houses and rooms to hold more outsiders. The data management on boarding house management used spreadsheet application, Microsoft Excel. However, the owners of the boarding house found some problems when the data is going to be collected, such as requiring a long time to find the data position. The errors in updating the data often occur in the wrong column and row position. The spreadsheet file is corrupted, so the data cannot be accessed and the data has to be re-inputted. Another problem is found that they often postpone or even forget to update the data to boarders in each room. Then it comes to the issue of late payment for boarders every month. The other problems found from the boarders' side in paying the rent is no receipt used as proof of payment. Then it leads to the output that boarders are charged to pay the rent again for the same month. The solution is given to overcome these problems is the owners has to make boarding house management system followed by adding boarding house and boarders features, payment limit notifications, boarding payments, boarding house finance reports, room bookings, boarding house damage reports, printed payment as a proof, data check and also payment status. The development of this boarding house management system is used waterfall model. Unit testing comes to a value 100% valid as the result. The validation test gives 100% value too. The compatibility testing also gives 100% value using ten web browsers.

Keywords: boarding house management, boarding house, software engineering, operational, waterfall model.

DAFTAR ISI

PENGESAHANii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vi
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Rumah Indekos	5
2.3 Harga Sewa Indekos.....	5
2.4 Fasilitas yang Tersedia	6
2.5 Proses Manajemen	6
2.5.1 Data	6
2.5.2 Layanan	7
2.5.3 Laporan.....	7
2.6 Rekayasa Perangkat Lunak	7
2.7 <i>Waterfall</i>	7
2.8 MySQL.....	9
2.9 Pengujian Perangkat Lunak.....	9
2.9.1 <i>Black-Box Testing</i>	9

2.9.2 <i>White-Box Testing</i>	10
2.9.3 <i>Compatibility Testing</i>	12
2.10 <i>Activity Diagram</i>	12
BAB 3 METODOLOGI PENELITIAN	13
3.1 Metodologi Penelitian	13
3.2 Observasi.....	14
3.3 Studi Literatur	14
3.4 Analisis Kebutuhan	14
3.5 Perancangan	15
3.6 Implementasi	15
3.7 Pengujian	15
3.7.1 <i>Black-Box Testing</i>	15
3.7.2 <i>White-Box Testing</i>	16
3.7.3 <i>Compatibility Testing</i>	16
3.8 Pengambilan Kesimpulan dan Saran	16
BAB 4 REKAYASA KEBUTUHAN	17
4.1 Elitisasi dan Analisis Kebutuhan.....	17
4.2 Identifikasi Aktor.....	18
4.3 Spesifikasi Kebutuhan	19
4.3.1 Kebutuhan Fungsional.....	19
4.3.2 Kebutuhan Non Fungsional.....	21
4.3.3 <i>Use Case Diagram</i>	22
4.3.4 <i>Use Case Scenario</i>	24
4.3.5 <i>Activity Diagram</i>	37
BAB 5 PERANCANGAN DAN IMPLEMENTASI	40
5.1 Perancangan	40
5.1.1 <i>Class Diagram</i>	41
5.1.2 <i>Entity Relation Diagram</i>	41
5.1.3 Pemodelan <i>Sequence Diagram</i>	43
5.1.4 Perancangan <i>Pseudocode</i>	45
5.1.5 Perancangan Antarmuka	49
5.2 Implementasi	65

5.2.1 Spesifikasi Sistem	65
5.2.2 Implementasi Basis Data	66
5.2.3 Implementasi <i>Pseudocode</i>	68
5.2.4 Implementasi Antarmuka	72
BAB 6 PENGUJIAN	85
6.1 Pengujian Unit.....	85
6.1.1 Pengujian Unit Masuk ke Dalam Sistem	85
6.1.2 Pengujian Unit Menambahkan Rumah Indekos	88
6.1.3 Pengujian Unit Menambahkan Penghuni	91
6.2 Pengujian Validasi	94
6.2.1 Pengujian Validasi Kebutuhan Fungsional	94
6.2.2 Pengujian Validasi Kebutuhan Non-Fungsional	106
6.3 Analisis	112
BAB 7 PeNUTUP	114
7.1 Kesimpulan.....	114
7.2 Saran	114
LAMPIRAN A	116
LAMPIRAN B	118
DAFTAR REFERENSI	123

DAFTAR TABEL

Tabel 4.1 Kebutuhan Fungsional.....	18
Tabel 4.2 Identifikasi Aktor	19
Tabel 4.3 Kebutuhan Fungsional.....	19
Tabel 4.4 Kebutuhan Non-Fungsional.....	21
Tabel 4.5 <i>Use case</i> mendaftar ke sistem.....	24
Tabel 4.6 <i>Use case scenario</i> masuk ke dalam sistem.....	24
Tabel 4.7 <i>Use case scenario</i> menambahkan rumah indekos	25
Tabel 4.8 <i>Use case scenario</i> pemesanan kamar	26
Tabel 4.9 <i>Use case scenario</i> menambahkan penghuni	26
Tabel 4.10 <i>Use case scenario</i> mengubah penghuni.....	27
Tabel 4.11 <i>Use case scenario</i> menghapus penghuni	28
Tabel 4.12 <i>Use case scenario</i> menambahkan akun penghuni	29
Tabel 4.13 <i>Use case scenario</i> menambahkan pembayaran.....	29
Tabel 4.14 <i>Use case scenario</i> mengubah pembayaran.....	30
Tabel 4.15 <i>Use case scenario</i> mengirimkan notifikasi batas pembayaran	31
Tabel 4.16 <i>Use case scenario</i> menambahkan pengeluaran.....	32
Tabel 4.17 <i>Use case scenario</i> mengubah pemesanan	32
Tabel 4.18 <i>Use case scenario</i> verifikasi kendala	33
Tabel 4.19 <i>Use case scenario</i> mencetak laporan bukti pembayaran.....	34
Tabel 4.20 <i>Use case scenario</i> keluar dari sistem.....	34
Tabel 4.21 <i>Use case scenario</i> melihat status pembayaran	34
Tabel 4.22 <i>Use case scenario</i> melihat data pembayaran.....	35
Tabel 4.23 <i>Use case scenario</i> melaporkan kendala	35
Tabel 4.24 <i>Use case scenario</i> melaporkan kendala (lanjutan).....	36
Tabel 5.1 <i>Pseudocode</i> masuk ke dalam sistem	46
Tabel 5.2 <i>Pseudocode</i> menambahkan rumah indekos	47
Tabel 5.3 <i>Pseudocode</i> menambahkan penghuni	48
Tabel 5.4 Spesifikasi perangkat keras komputer	66
Tabel 5.5 Spesifikasi perangkat lunak komputer	66
Tabel 5.6 Implementasi <i>pseudocode</i> masuk ke dalam sistem.....	68

Tabel 5.7 Implementasi <i>pseudocode</i> menambahkan rumah indekos	70
Tabel 5.8 Implementasi <i>pseudocode</i> menambahkan penghuni	71
Tabel 6.1 Pengujian unit masuk ke dalam sistem	85
Tabel 6.2 Pengujian unit masuk ke dalam sistem	87
Tabel 6.3 Pengujian unit menambahkan rumah indekos	88
Tabel 6.4 Pengujian unit menambahkan rumah indekos	91
Tabel 6.5 Pengujian unit menambahkan penghuni	91
Tabel 6.6 Pengujian unit menambahkan penghuni	93
Tabel 6.7 Pengujian validasi mendaftar ke sistem.....	94
Tabel 6.8 Pengujian validasi masuk ke sistem	95
Tabel 6.9 Pengujian validasi menambahkan rumah indekos.....	96
Tabel 6.10 Pengujian validasi pemesanan kamar	96
Tabel 6.11 Pengujian validasi menambahkan penghuni.....	97
Tabel 6.12 Pengujian validasi mengubah penghuni	98
Tabel 6.13 Pengujian validasi menghapus penghuni	99
Tabel 6.14 Pengujian validasi menambahkan akun penghuni.....	99
Tabel 6.15 Pengujian validasi menambahkan pembayaran	100
Tabel 6.16 Pengujian validasi mengubah pembayaran	101
Tabel 6.17 Pengujian validasi mengirimkan notifikasi batas pembayaran.....	102
Tabel 6.18 Pengujian validasi menambahkan pengeluaran	102
Tabel 6.19 Pengujian validasi mengubah pemesanan.....	103
Tabel 6.20 Pengujian validasi verifikasi kendala	104
Tabel 6.21 Pengujian validasi mencetak laporan bukti pembayaran	104
Tabel 6.22 Pengujian validasi melihat status pembayaran.....	104
Tabel 6.23 Pengujian validasi melihat data pembayaran	105
Tabel 6.24 Pengujian validasi melaporkan kendala	105
Tabel 6.25 Pengujian validasi keluar dari sistem	106
Tabel 6.26 Pengujian validasi kompatibilitas <i>internet explorer</i>	106
Tabel 6.27 Pengujian validasi kompatibilitas <i>google chrome</i>	107
Tabel 6.28 Pengujian validasi kompatibilitas <i>opera</i>	107
Tabel 6.29 Pengujian validasi kompatibilitas <i>mozilla firefox</i>	108
Tabel 6.30 Pengujian validasi kompatibilitas <i>vivaldi</i>	109

Tabel 6.31 Pengujian validasi kompatibilitas <i>safari</i>	109
Tabel 6.32 Pengujian validasi kompatibilitas <i>yandex</i>	110
Tabel 6.33 Pengujian validasi kompatibilitas <i>comodo ice dragon</i>	110
Tabel 6.34 Pengujian validasi kompatibilitas <i>maxthon</i>	111
Tabel 6.35 Pengujian validasi kompatibilitas <i>torch</i>	112

DAFTAR GAMBAR

Gambar 2.1 WBS Manajemen Rumah Indekos.....	6
Gambar 2.2 Gambar <i>Waterfall Model</i>	8
Gambar 2.3 Format pengujian <i>black box</i>	10
Gambar 2.4 Struktur <i>flow graph</i>	11
Gambar 2.5 Contoh <i>flow graph</i>	11
Gambar 3.1 Metode Pelaksanaan Sistem Pengelolaan Manajemen.....	13
Gambar 4.1 <i>Use case diagram</i>	23
Gambar 4.2 <i>Activity diagram</i> masuk ke dalam	37
Gambar 4.3 <i>Activity diagram</i> menambahkan rumah indekos	38
Gambar 4.4 <i>Activity diagram</i> menambahkan penghuni	39
Gambar 5.1 Perancangan <i>class diagram</i> sistem manajemen rumah indekos.....	41
Gambar 5.2 Perancangan <i>Entity Relation Diagram</i>	42
Gambar 5.3 <i>Sequence diagram</i> masuk ke dalam sistem	43
Gambar 5.4 <i>Sequence diagram</i> menambahkan rumah indekos	44
Gambar 5.5 <i>Sequence diagram</i> menambahkan penghuni	45
Gambar 5.6 Perancangan antarmuka halaman utama sistem	49
Gambar 5.7 Perancangan antarmuka halaman masuk ke sistem	50
Gambar 5.8 Perancangan antarmuka halaman <i>dashboard</i> pemilik	50
Gambar 5.9 Perancangan antarmuka halaman rumah indekos	51
Gambar 5.10 Perancangan antarmuka halaman penghuni	52
Gambar 5.11 Perancangan antarmuka halaman <i>reminder whatsapp</i>	52
Gambar 5.12 Perancangan antarmuka data pembayaran.....	53
Gambar 5.13 Perancangan antarmuka pengeluaran.....	54
Gambar 5.14 Perancangan antarmuka <i>list</i> pemesanan.....	54
Gambar 5.15 Perancangan antarmuka <i>list</i> kendala.....	55
Gambar 5.16 Perancangan antarmuka tambah indekos	56
Gambar 5.17 Perancangan antarmuka tambah penghuni	57
Gambar 5.18 Perancangan antarmuka tambah akun penghuni.....	58
Gambar 5.19 Perancangan antarmuka mengubah penghuni.....	59
Gambar 5.20 Perancangan antarmuka mendaftar ke sistem	60

Gambar 5.21 Perancangan antarmuka pemesanan kamar	61
Gambar 5.22 Perancangan antarmuka tagihan	62
Gambar 5.23 Perancangan antarmuka tambah pengeluaran	62
Gambar 5.24 Perancangan antarmuka <i>detail</i> rumah indekos.....	63
Gambar 5.25 Perancangan antarmuka data pembayaran.....	64
Gambar 5.26 Perancangan antarmuka lapor kendala	64
Gambar 5.27 Perancangan antarmuka status pembayaran	65
Gambar 5.28 Implementasi Basis Data	67
Gambar 5.29 Implementasi antarmuka halaman utama sistem	73
Gambar 5.30 Implementasi antarmuka masuk ke dalam sistem	73
Gambar 5.31 Implementasi antarmuka halaman <i>dashboard</i> pemilik.....	74
Gambar 5.32 Implementasi antarmuka halaman rumah indekos	75
Gambar 5.33 Implementasi antarmuka halaman <i>reminder</i>	75
Gambar 5.34 Implementasi antarmuka halaman data pembayaran	75
Gambar 5.35 Implementasi antarmuka halaman pengeluaran.....	76
Gambar 5.36 Implementasi antarmuka halaman <i>list</i> pemesanan.....	76
Gambar 5.37 Implementasi antarmuka halaman <i>list</i> kendala.....	77
Gambar 5.38 Implementasi antarmuka halaman tambah rumah indekos	77
Gambar 5.39 Implementasi antarmuka halaman tambah penghuni	78
Gambar 5.40 Implementasi antarmuka halaman tambah akun penghuni	79
Gambar 5.41 Implementasi antarmuka halaman mengubah penghuni	80
Gambar 5.42 Implementasi antarmuka halaman mendaftar ke sistem.....	80
Gambar 5.43 Implementasi antarmuka halaman pemesanan kamar	81
Gambar 5.44 Implementasi antarmuka halaman pembayaran.....	81
Gambar 5.45 Implementasi antarmuka halaman tambah pembayaran	82
Gambar 5.46 Implementasi antarmuka halaman <i>detail</i> rumah indekos.....	82
Gambar 5.47 Implementasi antarmuka halaman data pembayaran	83
Gambar 5.48 Implementasi antarmuka halaman lapor kendala	83
Gambar 5.49 Implementasi antarmuka halaman status pembayaran indekos ...	83
Gambar 6.1 <i>Flow graph</i> masuk ke dalam sistem	87
Gambar 6.2 <i>Flow graph</i> menambahkan penghuni	90
Gambar 6.3 <i>Flow graph</i> menambahkan penghuni	93

Gambar 6.4 Hasil dari pengujian validasi kompatibilitas “ <i>Internet Explorer</i> ”	107
Gambar 6.5 Hasil dari pengujian validasi kompatibilitas “ <i>Google Chrome</i> ”.....	107
Gambar 6.6 Hasil dari pengujian validasi kompatibilitas “ <i>Opera</i> ”.....	108
Gambar 6.7 Hasil dari pengujian validasi kompatibilitas “ <i>Mozilla Firefox</i> ”.....	109
Gambar 6.8 Hasil dari pengujian validasi kompatibilitas “ <i>Vivaldi</i> ”	109
Gambar 6.9 Hasil dari pengujian validasi kompatibilitas “ <i>Safari</i> ”	110
Gambar 6.10 Hasil dari pengujian validasi kompatibilitas “ <i>Yandex</i> ”	110
Gambar 6.11 Hasil dari pengujian validasi kompatibilitas “ <i>Comodo Ice Dragon</i> ”	111
Gambar 6.12 Hasil dari pengujian validasi kompatibilitas “ <i>Maxthon</i> ”	112
Gambar 6.13 Hasil dari pengujian validasi kompatibilitas “ <i>Torch</i> ”.....	112

DAFTAR LAMPIRAN

A.1 Hasil Wawancara	116
B.1 <i>Sequence Diagram</i>	118

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Indekos yakni rumah yang akan digunakan untuk tinggal sementara di tempat orang dengan membayar uang sewa per bulannya (KBBI, 2005). Penghuni indekos berasal dari berbagai daerah di Indonesia yang diantaranya pelajar, mahasiswa, dan pegawai. Memutuskan untuk tinggal di rumah indekos agar berdekatan dengan keberadaan lokasi kampus dan tempat kerja menjadi alasan utama dari penghuni indekos. Kemajuan teknologi yang semakin terjangkau dapat mempermudah para pencari indekos menemukan tempat yang diinginkan. Selain itu bermunculannya pengusaha indekos menyebabkan banyaknya persaingan bisnis diantara mereka. Menurut dinas pendapatan kota Malang, yang diambil dari *website* BPPD kota Malang terdapat 400 rumah indekos yang sudah terdaftar harus membayar pajak. Jumlah tersebut belum termasuk rumah indekos di kecamatan lainnya (Perda, 2015). Dalam pesaingan tersebut pemilik indekos mulai memperhatikan masalah untuk meningkatkan pelayanan. Peningkatan tersebut diantaranya dapat ditinjau mulai dari segi fasilitas rumah indekos, fasilitas kamar, dan pelayanan agar banyak peminatnya.

Seiring bertambahnya mahasiswa atau para pendatang yang berdatangan di kota Malang, tidak menuntut kemungkinan pemilik indekos ingin menambah rumah indekos serta menambah jumlah kamar lagi untuk menampung para pendatang. Keinginan tersebut muncul ketika para pendatang yang menanyakan kamar kosong apakah masih tersedia. Dalam pengelolaan manajemen rumah indekos, sedikit dari pemilik rumah indekos yang menggunakan aplikasi atau sistem yang dapat membantu untuk manajemen rumah indekos yang dimiliki. Sehingga untuk manajemen rumah indekos pemilik menggunakan cara yang konvesional. Pada manajemen rumah indekos, pemilik rumah indekos untuk melakukan pengelolaan rumah indekos tanpa harus datang langsung ke tempat rumah indekos atau menelpon penjaga yang berada di rumah indekos. Manajemen rumah indekos sendiri mulai dari pengelolaan rumah indekos, pengelolaan data penghuni, serta laporan pembayaran.

Dalam pengelolahan rumah indekos yang saat ini berjumlah 20 kamar dibagi menjadi dua lokasi, pemilik indekos masih menggunakan pendataan dengan cara yang masih konvensional. Pendataan manajemen rumah indekos menggunakan aplikasi *spreadsheet* yaitu Microsoft Excel. Dalam pengoperasiannya, pemilik indekos melakukan pencarian data atau melakukan pembaruan, pemilik mencari letak data terlebih dahulu sehingga membutuhkan waktu. Ketika pemilik indekos melakukan pembaruan data sering terjadi kesalahan penulisan nama, tanggal masuk, dan penghuni kamar berada di nomor berapa. Kesalahan dalam pembaruan data tersebut sering terjadi pada posisi kolom dan baris yang salah. Masalah yang lain selama ini dialami pemilik indekos ingin melakukan *backup* data, *file* tersebut dipindahkan ke alat penyimpanan eksternal. *File spreadsheet* mengalami *corrupt*, sehingga data tidak dapat diakses dan harus melakukan *entry* ulang data.

Beberapa permasalahan lainnya yang dihadapi adalah pemilik indekos sering mengalami keterlambatan atau bahkan lupa untuk memperbarui data terhadap penghuni indekos setiap kamarnya. Pendataan data penghuni indekos yang dilakukan oleh pemilik indekos dengan menuliskan nama penghuni indekos dan waktu masuknya setiap kamarnya di buku catatan. Akibat sering mengalami keterlambatan untuk memperbarui data penghuni indekos setiap kamarnya. Ketika pada saat melakukan berpindah kamar dari satu kamar ke kamar yang lain yang dilakukan oleh penghuni indekos, maka penghuni indekos dibebankan untuk membayar lagi pada bulan yang sama.

Permasalahan lainnya adalah masalah keterlambatan pembayaran penghuni indekos setiap bulannya. Untuk pembayaran indekos tiap bulannya menggunakan cara bertatap muka antara pemilik dan penghuni indekos. Masalah ini dikarenakan pemilik indekos sering lupa tanggal pembayaran terakhir yang dilakukan oleh penghuni indekos. Kerugian yang dialami oleh pemilik indekos ketika keterlambatan pembayaran, melakukan tagihan secara manual dengan menegur setiap penghuni indekos yang belum membayar. Proses cara menegur tersebut menyulitkan pemilik indekos dikarenakan lokasi rumah pemilik indekos dengan lokasi rumah indekos tidak saling berdekatan. Permasalahan yang lain karena sering tidak bertemuanya dengan penghuni indekos, yang mengakibatkan pemilik indekos bolak-balik untuk menegur agar melakukan pembayaran.

Permasalahan dari segi penghuni indekos ialah pada saat melakukan pembayaran, tidak adanya kwitansi yang digunakan sebagai bukti pembayaran. Bukti pembayaran yang selama ini digunakan oleh pemilik indekos dan penghuni indekos masih menggunakan cara manual. Pendataan bukti pembayaran dengan menuliskan setiap bulannya pada buku catatan transaksi pembayaran yang dilakukan oleh pemilik dan penghuni indekos. Pendataan bukti transaksi pembayaran yang masih menggunakan buku catatan dirasa kurang efektif dan kurang transparansi oleh pemilik indekos. Faktor permasalahan yang lain pemilik indekos tidak memberikan bukti pembayaran sehingga penghuni indekos tidak memiliki bukti telah melakukan pembayaran. Akibat dari tidak adanya kwitansi yang digunakan sebagai bukti pembayaran, maka penghuni indekos dibebankan untuk membayar lagi pada bulan yang sama.

Berdasarkan permasalahan-permasalahan di atas maka dibuatlah aplikasi sistem manajemen indekos. Manajemen sendiri sebagai bidang ilmu pengetahuan yang berusaha secara sistematis, digunakan sebagai memahami mengapa dan bagaimana membuat sistem agar lebih bermanfaat bagi manusia (Husaini, 2006). Manajemen pengelolaan rumah indekos untuk mengelola rumah indekos agar lebih baik dan membantu agar lebih mudah pengelolaan. Sistem yang dibuat ini akan memenuhi semua kebutuhan yang ada pada rumah indekos semanggi kota Malang menggunakan pendekatan berorientasi objek. Mulai dari pengelolaan rumah indekos, pengelolaan penghuni, dan operasional rumah indekos. Pendekatan berorientasi objek digunakan sebagai pedoman untuk melakukan penelitian yang lebih lanjut. Dalam pembuatan sistem manajemen rumah indekos menggunakan metode *waterfall*, dengan menggunakan metode

ini memiliki kemudahan untuk analisis kebutuhan bersifat terstruktur dan bersifat stabil. Pengembangan yang bersifat terstruktur membuat *software* tetap terjaga kualitasnya (Fahrurrozi, 2012). Terdapat *framework* yang dapat mengimplementasi dan mengembangkan sistem ini, yaitu CodeIgniter. CodeIgniter memiliki beberapa kelebihan seperti kemudahan sintaks dan juga kecepatan. Maka dari itu sistem pengelolaan manajemen rumah indekos menggunakan CodeIgniter. Dengan dibuatnya sistem manajemen rumah indekos diharapkan dapat mengelola data dengan mudah dan menghilangkan kesalahan – kesalahan yang sering terjadi.

1.2 Rumusan Masalah

1. Bagaimana menganalisis, merancang dan implementasi sistem pengelolaan manajemen indekos pada rumah indekos semanggi ?
2. Bagaimana hasil pengujian dari sistem pengelolaan manajemen rumah indekos untuk mempermudah kinerja pemilik indekos ?

1.3 Tujuan

1. Mengetahui hasil analisis, perancangan serta implementasi sistem pengelolaan manajemen rumah indekos sesuai dengan kebutuhan.
2. Mengetahui hasil pengujian dari sistem pengelolaan manajemen rumah indekos.

1.4 Manfaat

Penulisan dokumentasi ini mempunyai beberapa kegunaan diantaranya berikut ini.

1. Mempersilahkan bagi pengembang sistem untuk berkontribusi dalam pengembangan dan pemanfaatan teknologi pada versi berikutnya seiring perkembangan teknologi.
2. Sistem ini dapat memudahkan bagi pemilik untuk mengelola rumah indekos yang dimiliki.

1.5 Batasan Masalah

Terdapat batasan-batasan masalah ketika pembuatan sistem diantaranya.

1. Kota Malang dijadikan sebagai lokasi untuk obyek penelitian khususnya indekos “Semanggi”.
2. Sistem ini terdiri dari dua *user*, yaitu pemilik dan penghuni indekos.
3. *Website* ini hanya dapat menampilkan informasi mengenai manajemen rumah indekos.
4. Penelitian tidak membahas tentang infrastruktur jaringan dan keamanan sistem.
5. Penelitian ini menggunakan metode penelitian *waterfall model* namun tidak sampai tahap *maintenance*.

1.6 Sistematika Pembahasan

Tahapan-tahapan sistematika pembahasan laporan digunakan penggambaran dari laporan dengan cara mengambil garis besar yang mempunyai 7 bab, antara lain dibawah ini.

BAB I Pendahuluan

Bab pendahuluan di dalamnya terdapat tentang penulisan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

BAB II Landasan Kepustakaan

Bab landasan kepustakaan berisi beberapa dasar teori-teori sistem aplikasi dalam perancangan, pembuatan dan pengujian.

BAB III Metodologi Penelitian

Bab metodologi penelitian membahas metode penelitian yang akan digunakan dalam penelitian.

BAB IV Rekayasa Kebutuhan

Bab rekayasa kebutuhan berisikan mengenai rekayasa kebutuhan yang akan digunakan dalam pembangunan Sistem Pengelolaan Manajemen Rumah Indekos.

BAB V Perancangan dan Implementasi

Bab perancangan dan implementasi menjelaskan mengenai tahapan-tahapan dalam perancangan dan implementasi Sistem Pengelolaan Manajemen Rumah Indekos.

BAB VI Pengujian

Bab pengujian menjelaskan mengenai tahap pengujian saat pembangunan Sistem Pengelolaan Manajemen Rumah Indekos untuk mengetahui sistem sudah berjalan dengan sesuai.

BAB VII Penutup

Bab penutup menjelaskan mengenai kesimpulan dan saran yang telah dilakukan dari hasil pengujian.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Berdasarkan penelitian yang sudah pernah dilakukan peniliti-peneliti antara lain.

Satria Novari (2017) yang berjudul Sistem Informasi Administrasi Pembayaran Kost Hj. Murni Baturaja Menggunakan *Embarcadero XE2*. Penelitian yang dilakukan salah satunya untuk merancang suatu sistem informasi untuk dapat membantu memudahkan dalam pengolahan data pembayaran pada kost Hj. Murni, dan dapat mengurangi kesalahan yang terjadi ketika menggunakan sistem yang lama dengan menggunakan pendataan di buku secara manual. Keunggulan dari sistem tersebut adalah untuk mengelola informasi yang berhubungan dengan data penghuni indekos dan informasi rumah indekos adalah pemilik yang menggunakan beberapa fitur yang berada pada aplikasi *Embarcadero XE2*. Macam – macam fitur tersebut yaitu pemilik dapat menambahkan data penghuni indekos, data kamar, data pembayaran sewa kamar dan nota.

Fery Setyawan (2014) yang berjudul Sistem Pembayaran Biaya Kost Terkomputerisasi Pada Rumah Kost Griya Ampel Muda Kabupaten Pacitan. Penelitian dilakukan untuk mempermudah pemilik indekos dalam melakukan pengecekan masalah data-data pembayaran serta data penghuni indekos. Kelebihan mengenai kemudahan yang dirasakan oleh pemilik indekos dalam melakukan pengolahan data sewanya, mengurangi kesalahan dalam melakukan pendataan maupun pengolahan data dan meningkatkan kinerja untuk memberikan hasil yang otimal. Kekurangannya terletak pada tampilan yang kaku dan kurang menarik.

2.2 Rumah Indekos

Disamping kebutuhan makanan dan sandang, rumah merupakan salah satu kebutuhan pokok manusia. Berdasarkan perundang-undangan Nomor 1 Tahun 2011 mengenai Perumahan dan Kawasan Permukiman mengatakan bahwa rumah merupakan kebutuhan dasar yang digunakan sebagai untuk peningkatan dan kesejahteraan masyarakat (KPUPR, 2013). Berdasarkan penjelasan mengenai perumahan dan kawasan permukiman, maka rumah merupakan kebutuhan pokok untuk mahasiswa yang sedang menuntut ilmu jauh dari tempat daerah asalnya membutuhkan tempat tinggal untuk sementara waktu.

2.3 Harga Sewa Indekos

Harga yakni nilai suatu barang yang dirupakan dengan uang (Kamus Besar Bahasa Indonesia, 2005), sedangkan untuk pengertian sewa yakni pemakaian suatu barang yang kemudian membayar dengan uang ketika sudah selesai memakainya(Kamus Besar Bahasa Indonesia, 2005). Pengertian tersebut dapat diartikan sebagai alat pembayaran yang menggunakan uang untuk membayarnya. Menurut pendapat (Soesabdo, 1981) bahwa yang dimaksud

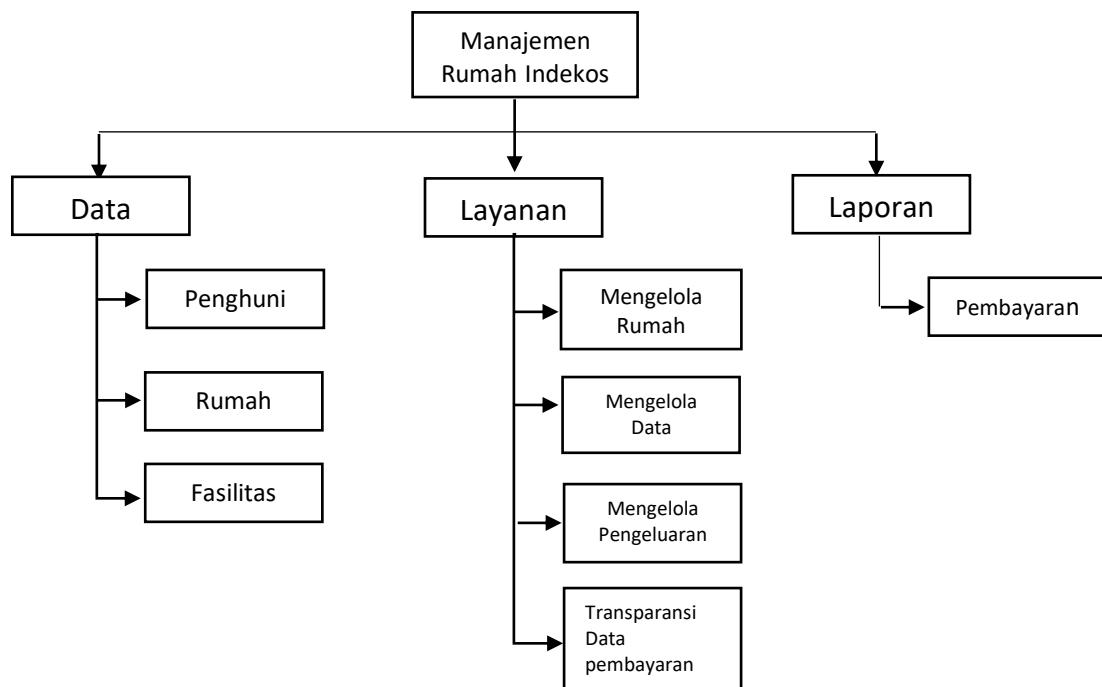
harga sewa yakni ketika sudah memakainya atas penggunaan perumahan mengganti dalam bentuk apapun.

2.4 Fasilitas yang Tersedia

Sarana, prasarana serta fasilitas merupakan dukungan dari lingkungan fisik untuk proses pendidikan (Sukmadinata, 2007). Berdasarkan pengertian tersebut fasilitas merupakan hal penting untuk proses pendidikan. Kenyamanan penghuni dinilai dari berdasarkan fasilitas yang cukup, untuk dikatakan sebagai rumah memiliki nilai standar kriteria (Panudju, 1999) kriteria yang dimaksud yakni dari segi kesehatan, kelembapan, serta prasarana listrik dan air yang baik.

2.5 Proses Manajemen

Pemilik rumah indekos melakukan manajemen yang didasari dengan proses manajemen rumah indekos. *Work Breakdown System* (WBS) sendiri merupakan ruang lingkup dari suatu target dari sebuah proyek yang terorganisir. *Work Breakdown System* (WBS) memiliki *work packages* yang digunakan untuk membuat komponen lebih kecil sehingga memudahkan untuk mengurnanya (Marchewka, 2015). Gambar 2.1 merupakan Diagram *Work Breakdown System* (WBS).



Gambar 2.1 WBS Manajemen Rumah Indekos

2.5.1 Data

Pada saat proses ini pemilik rumah indekos mempunyai data-data dari manajemen rumah indekos. Pada tahap ini data-data yang terdapat pada sistem diantaranya data penghuni, data rumah indekos dan data fasilitas. Data pada penghuni bisa didapatkan dari Kartu Tanda Kependudukan, indentitas pribadi

dan berserta nomor telepon dari penghuni. Kemudian data rumah indekos meliputi nama indekos, alamat indekos, harga dan luas kamar. Dalam data fasilitas meliputi fasilitas yang berada di dalam kamar dan fasilitas yang terdapat di rumah indekos (fasilitas umum).

2.5.2 Layanan

Pada saat manajemen rumah indekos, pemilik rumah indekos mempunyai layanan-layanan untuk mendukung manajemen rumah indekos tersebut. Layanan-layanan tersebut terdiri dari mengelola rumah indekos, mengelola data penghuni, mengelola pengeluaran dan transparansi data pembayaran. Ketika rumah indekos yang dimiliki belum terdaftar pada sistem, maka pemilik melakukan *register* rumah indekos. Setelah rumah indekos tersebut sudah terdaftar pada sistem, maka pemilik rumah indekos dapat melakukan pengelolaan terhadap penghuni terhadap kamar-kamar yang sudah tersedia didalam rumah indekos.

2.5.3 Laporan

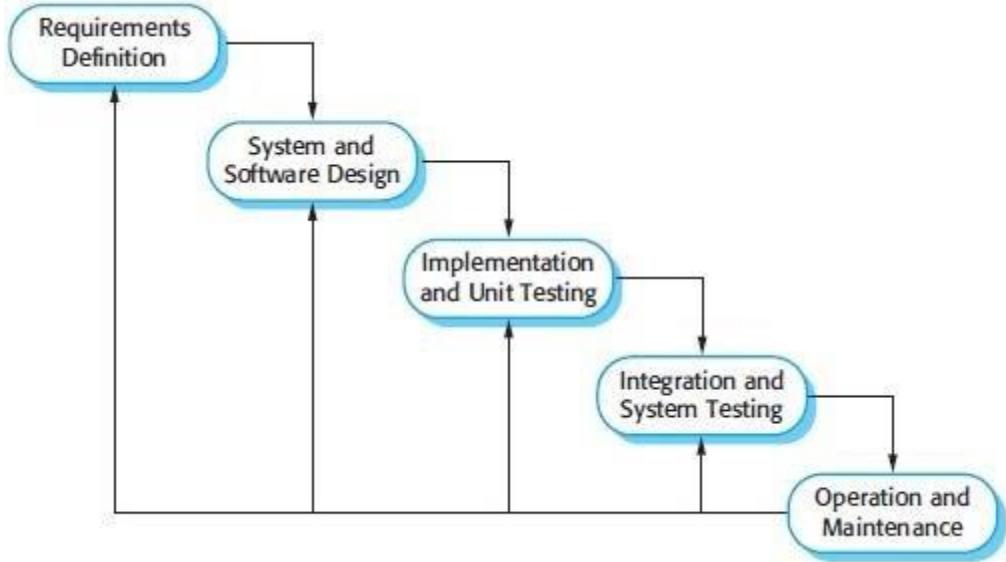
Pada tahap ini pemilik rumah indekos melakukan pengecekan terhadap manajemen rumah indekos yang dimiliki. Pengecekan tersebut bisa berupa laporan tiap bulannya. Laporan-laporan rumah indekos tersebut meliputi laporan pengeluaran.

2.6 Rekayasa Perangkat Lunak

Rekayasa perangkat yakni pengetahuan yang digunakan untuk merancang ataupun membuat sebuah program komputer atau alat elektronik yang lainnya (Laplante, 2007). Programmer membuat sebuah perangkat lunak dengan menggunakan metode yang baik agar menghasilkan kualitas baik. Perangkat lunak sendiri yang baik ketika memiliki kriteria yang mudah digunakan. Karena hal tersebut yakni cara yang paling efektif untuk menghasilkan perangkat lunak yang mempunyai kualitas yang baik. Dibutuhkan pengakuan dari universitas dan latihan yang baik untuk menulis kode agar menjadi seorang insinyur.

2.7 Waterfall

Model waterfall menggunakan pendekatan secara sistematis dan berurutan atau bisa disebut dengan model sekunsial linier (Pressman, 2010).



Gambar 2.2 Gambar *Waterfall Model*

Sumber : Sommerville (2003)

Model *Waterfall* mempunyai lima tahapan pada yakni analisis kebutuhan, perancangan, implementasi, pengujian, dan yang terakhir pemeliharaan. Setiap tahapan pengembangan dilakukan dokumentasi pada model *waterfall* (Sommerville, 2003). Tahapan-tahapan meliputi Analisis Kebutuhan, Perancangan, Implementasi dan Pengujian. Model pengembangan *waterfall* digunakan pada penelitian ini. Dipilih karena pada saat melakukan kebutuhan sistem ini sudah didefinisikan di tahap awal sehingga kebutuhan tidak berubah-ubah karena model *waterfall* mendefinisikan kebutuhan-kebutuhan dengan sangat jelas di tahap analisis kebutuhan (Pressman, 2010). *Waterfall model* memiliki tahapan diantaranya.

1. Analisis Kebutuhan merupakan proses untuk mengumpulkan kebutuhan yang dilakukan dengan lengkap. Sistem memiliki dua kebutuhan yaitu fungsional dan nonfungsional. Kebutuhan fungsional sendiri merupakan sebuah layanan yang harus dimiliki oleh sistem. Kebutuhan nonfungsional digunakan sebagai batasan yang telah diberikan oleh sistem
2. Perancangan, setelah dilakukan tahapan untuk analisis kebutuhan. Tahapan selanjutnya merupakan tahapan untuk perancangan. Perancangan sistem dibuat dari tahapan analisis kebutuhan. Tahapan perancangan mempunyai tujuan untuk menggambarkan dasar sistem dan hubungan-hubungannya.
3. Implementasi, setelah dilakukan tahapan untuk perancangan sistem maka tahapan selanjutnya yaitu implementasi. Memlilik tujuan untuk mengartikan ke bentuk kode program sehingga bisa dibaca oleh mesin.
4. Pengujian, setelah dilakukan tahapan untuk perancangan sistem maka tahapan selanjutnya adalah tahapan pengujian. Tahapan pengujian ini digunakan untuk menguji sistem sudah sesuai atau tidak dengan kebutuhan yang didefinisikan di tahapan analisis.

- Pemeliharaan, merupakan tahapan selanjutnya setelah dilakukan pengujian. Pada tahapan ini merupakan tahapan sistem mulai digunakan dan memperbaiki *error* yang pada tahapan sebelumnya tidak ditemukan. Pada tahapan ini juga dapat dilakukan untuk pengembangan lebih lanjut seperti penambahan fitur baru.

2.8 MySQL

MySQL yakni perangkat lunak sistem manajemen basis data *Database management system* yang memiliki sifat *multithread* dan *multi-user* (Adelheid dkk, 2012). Sampai sekarang MySQL merupakan *database server* yang sangat terkenal di dunia dikarenakan SQL merupakan bahasa dasar yang digunakan untuk mengakses *database*. Laboratorium riset San Jose pertama kali menerapkan SQL (*Structured Query Language*) pada sebuah proyek riset.

2.9 Pengujian Perangkat Lunak

Pengujian perangkat lunak tahapan yang digunakan untuk proses mengeksekusi program serta menemukan kesalahan (Myers, 2004). Kasus uji merupakan rancangan untuk melakukan pengujian yang digunakan untuk menemukan kesalahan. Pengujian perangkat lunak sendiri dibagi menjadi dua yaitu pengujian *black box* yang berguna untuk menguji fungsionalitas dari sistem dan pengujian *white box* yang digunakan untuk menguji logik dari sebuah sistem.

2.9.1 Black-Box Testing

Tahap pengujian ini merupakan pengujian yang berfokus pada spesifikasi kebutuhan dari suatu sistem yang sudah didefinisikan di tahap analisis kebutuhan (Pressman, 2010). *Black-box testing* dibagi menjadi beberapa jenis, yaitu *graph-based testing*, *scenario-based testing*, *equivalence partitioning*, *boundary value analysis*, dan *comparison testing*.

Scenario-based testing yakni jenis pengujian yang berfokus pada perlakuan pengguna. Jenis pengujian ini dilakukan dapat untuk mengetahui apakah ada kesalahan ketika pengguna menggunakan sistem. *Use-case scenario* akan diterapkan pada pengujian ini sebagai acuan menentukan langkah yang harus dilakukan pengguna terhadap sistem. Dalam pengujian *scenario-based testing* untuk melakukan pengujian menggunakan kasus uji dengan menerapkan *use scenario* yang sudah ditentukan berada di hasil eksperimen, agar mengetahui hasil eksperimen sama dengan hasil sebenarnya telah sesuai atau tidak. Berikut adalah sebuah format untuk pengujian *black box* yang ditunjukkan dengan Gambar 2.3 dibawah ini.

Test ID	Description	Expected Results	Actual Results

Gambar 2.3 Format pengujian *black box*

Sumber: Williams (2006)

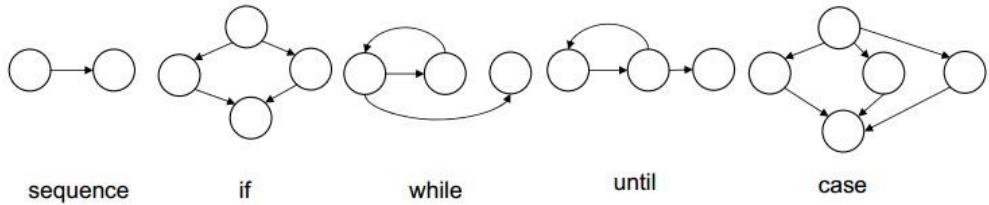
Equivalence Partitioning adalah jenis pengujian yang memisahkan ke dalam kelas-kelas data, kemudian mengecek apakah hasil yang ditampilkan oleh sistem sesuai dengan hasil yang diharapkan. Beberapa kelebihan menggunakan pengujian *equivalence partitioning* yaitu dapat menutupi satu klas kesalahan dengan satu kasus uji yang sudah dibuat, meminimalisir jumlah kasus uji sehingga bersifat efisiensi, kumpulan beberapa data yang ada dari setiap klas data mempunyai pengaruh yang sama terhadap program dan dapat merepresentasikan kondisi masukan yang mewakili keadaan valid atau tidak valid. Pedoman-pedoman umum yang digunakan untuk penentuan klas ekuivalen adalah sebagai berikut:

1. Kondisi masukan merupakan sebuah rentang: 1 klas ekuivalen valid dan 2 klas ekuivalen tidak valid.
2. Kondisi masukan merupakan sebuah nilai tertentu: 1 klas ekuivalen valid dan 2 klas ekuivalen tidak valid.
3. Kondisi masukan merupakan sebuah anggota dari suatu kumpulan: 1 klas ekuivalen valid per anggota dan 1 klas ekuivalen tidak valid.

2.9.2 White-Box Testing

Tahap pengujian *white box testing* merupakan jenis pengujian perangkat lunak yang lebih berkonsentrasi terhadap jalan program serta memastikan bahwa semua jalur sudah dilewati meskipun dilewati cuma sekali. Beberapa jenis *white box testing*, diantaranya basis *path testing*, *condition testing*, dan *loop testing* (Nidhra and Dondetti, 2012). Pada tahapan penelitian ini, menggunakan *basis path testing* dalam pengujian *white box*. Pengujian *Basis path testing* sendiri digunakan untuk menguji setiap operasi dari setiap klas-klas yang dihasilkan pada tahap perancangan. Pengujian yang dilakukan dengan menggambarkan alur algoritme sistem. Algoritme yang menggambarkan suatu kondisi menggunakan *node* dan *edge*. Pengujian ini memiliki tujuan untuk mengetahui jalur eksekusi yang harus dilalui berdasarkan perhitungan *cyclomatic complexity*. Kasus uji dibuat pada setiap jalur eksekusi. Pengujian ini sendiri memiliki *flow graph* untuk merepresentasikan jalan yang ada pada suatu program seperti pada Gambar 2.2.

Sedangkan untuk penghitungan jumlah *cyclomatic complexity* dapat dihitung dengan menggunakan persamaan 2.1.



Gambar 2.4 Struktur flow graph

Sumber : Laplante (2007)

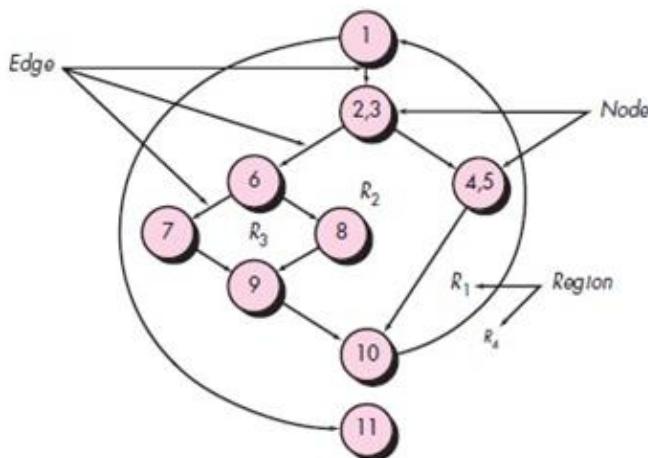
$$C = E - N + 2 \quad (2.1)$$

C = Cyclomatic complexity

E = Jumlah Edge

N = Jumlah Node

Gambar 2.3 merupakan contoh dari sebuah *flow graph*:



Gambar 2.5 Contoh flow graph

Sumber : Laplante (2007)

Flow graph dari contoh diatas, sebuah program didapatkan *cyclomatic complexity* dengan rumus dibawah ini:

$$C = E - N + 2 \quad (2.1)$$

Dengan rumus diatas dapat ditemukan hasil dari *cyclomatic complexity*nya sebagai berikut:

$$C = 11 - 9 + 2$$

$$= 4$$

Perhitungan hasil *cyclomatic complexity* menghasilkan jumlah *independent path* dari sebuah *basis path testing* berjumlah 4.

2.9.3 *Compatibility Testing*

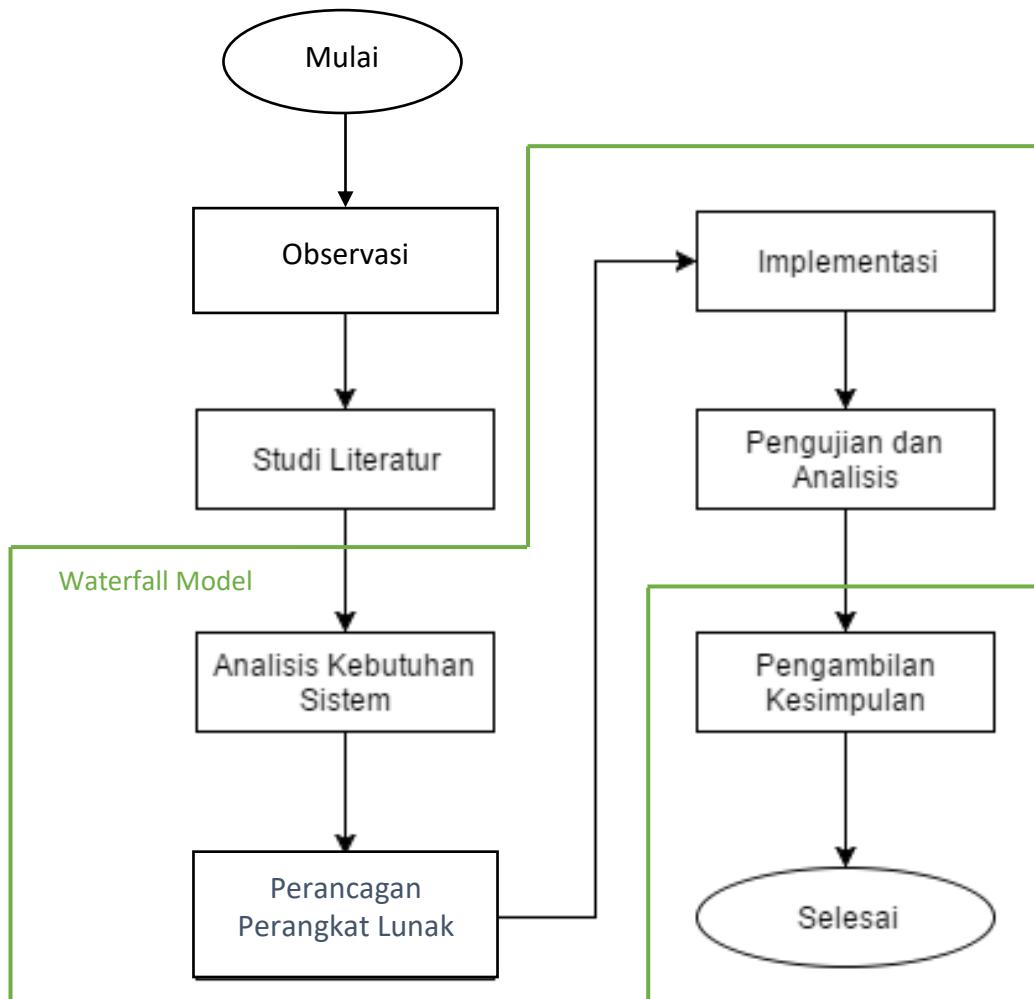
Pengujian *compatibility* merupakan pengujian dengan menjalankan sistem pada *web browser*. Pada pengujian *compatibility* menguji apakah sistem yang berada pada *web browser* sudah berjalan dengan baik, serta menjalankan fungsinya masing-masing tanpa ada mengalami gangguan. Ketika sistem sudah menjalankan fungsinya dengan baik maka dapat dibilang bahwa sistem berhasil dijalankan (Jamwal, 2010).

2.10 *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas satu ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *usecae* (Pressman, 2010).

BAB 3 METODOLOGI PENELITIAN

Penelitian ini digunakan secara khusus untuk membuat dokumentasi serta membahas dalam implementasi pembuatan sistem pengelolaan manajemen rumah indekos secara *online* berbasis *website*. Pada Gambar 3.1 menampilkan metode penelitian dari tahapan – tahapan pengerjaan penelitian ini.



Gambar 3.1 Metode Pelaksanaan Sistem Pengelolaan Manajemen Rumah Indekos

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan penelitian ini yakni menggunakan penelitian kualitatif. Penelitian kualitatif sendiri digunakan aspek pemahaman dengan secara mendalam agar mengetahui suatu masalah dalam penelitian lingkup generalisasi. Metode penelitian kualitatif lebih mengalisis suatu masalah kasus perkasus dengan cara mengumpulkan data secara primer dan sekunder. Data primer dan sekunder digunakan untuk memenuhi di tahap analis kebutuhan.

3.2 Observasi

Pada tahap observasi awal dilakukan studi pendahuluan yang mencakup tempat penelitian umum, permasalahan yang dihadapi oleh pemilik indekos dalam proses operasional dan identifikasi permasalahan yang dikembangkan. Setelah dilakukannya observasi awal maka akan diperoleh gambaran ruang lingkup dari objek penelitian dan identifikasi masalah yang ada, yang kemudian akan dikembangkan dan difokuskan ke dalam target rumusan masalah yang lebih spesifik.

3.3 Studi Literatur

Studi literatur diperlukan dalam tahap metode penelitian sebagai acuan untuk mempelajari dan literatur – literatur yang menunjang atau mendukung dalam proses perancangan, pembuatan dan pengujian sistem manajemen rumah indekos. Adapun literatur yang di dapat bersumber dari paper, jurnal, makalah dan berbagai macam tutorial – turorial yang berasal dari *internet*. Beberapa teori yang dipelajari untuk pembuatan laporan ini antara lain sebagai berikut.

1. Kajian pustaka
2. Rumah indekos
3. Harga sewa indekos
4. Fasilitas yang tersedia
5. Proses manajemen
 - Data
 - Manajemen
 - Laporan
6. Rekayasa perangkat lunak
7. Penggunaan *Unified Modeling Language* (UML)
 - Perancangan *use case diagram*
 - Perancangan *sequence diagram*
 - Perancangan *class diagram*
8. Waterfall
9. MySQL
10. Pemrograman HTML, PHP, CSS , JavaScript
11. Pengujian perangkat lunak
 - *Black-box testing*
 - *White-box tesing*
 - *Compatibility testing*

3.4 Analisis Kebutuhan

Analisis kebutuhan digunakan untuk mendapatkan semua kebutuhan dari perangkat lunak yang diinggakan baik itu kebutuhan fungsional maupun non

fungsional. Observasi dan wawancara secara langsung kepada pemilik indekos merupakan proses analisis kebutuhan ini dilakukan. Dari observasi dan wawancara tersebut peneliti bisa mengidentifikasi seluruh kebutuhan dari sistem.

3.5 Perancangan

Perancangan sistem berguna mempermudah dalam proses pembangunan sistem manajemen rumah indekos. Perancangan sistem dimulai dari analisis kebutuhan pengguna, tahapan analisis kebutuhan sistem mempunyai tujuan untuk memodelkan informasi yang akan digunakan dalam tahapan perancangan. Analisis kebutuhan sistem yang diperlukan meliputi identifikasi aktor dan kebutuhan fungsional. Dalam indentifikasi aktor bertujuan untuk mengidentifikasi aktor-aktor yang nantinya akan berhubungan dengan sistem. Kebutuhan fungsional itu sendiri bertujuan untuk menjelaskan fitur-fitur yang ada di dalam sistem.

Daftar kebutuhan sistem kemudian dilanjutkan dengan perancangan *use case* diagram, yang merupakan bentuk pemodelan dari perilaku (*behavior*) dari sistem. *Use case* menjelaskan interaksi antara aktor dengan sistem. Untuk mengetahui fungsi-fungsi apa saja yang terdapat di dalam sebuah sistem digambarkan dengan *use case* dan siapa saja yang mempunyai hak untuk menggunakan fungsi tersebut. Kemudian untuk mendapatkan deskripsi secara global mengenai *use case* maka menggunakan *use case scenario*. Setelah melakukan perancangan *use case scenario* dilanjutkan perancangan *sequence diagram*, perancangan *class diagram* dan perancangan *entity relationship diagram* (ERD).

3.6 Implementasi

Dalam tahap implementasi ini merupakan mengubah bentuk perancangan kedalam bentuk *code* atau bahasa pemrograman yang digunakan dalam pembuatan system manajemen rumah indekos. Implementasi ini menggunakan *framework CodeIgniter – Bootstrap* dan bahasa pemrograman PHP, HTML dan JavaScript.

3.7 Pengujian

Tahapan pengujian digunakan untuk memastikan bahwa sistem manajemen rumah indekos berjalan sesuai perancangan sistem dan memenuhi kebutuhan dari pengguna. Pengujian yang digunakan pada sistem ini adalah pengujian *black box* dan *white box*. Terdapat dua jenis pengujian yang akan dilakukan diantaranya.

3.7.1 Black-Box Testing

Pengujian *black-box* diterapkan selama tahap akhir pengujian. Pengujian *black-box* memperhatikan struktur kontrol. Teknik yang digunakan untuk

pengujian *black-box* yaitu menggunakan *Scenario-based testing* merupakan jenis pengujian yang berpusat pada perlakuan pengguna. Jenis pengujian ini dilakukan agar mengetahui apakah ada kesalahan saat pengguna berinteraksi dengan sistem. Pengujian *scenario-based testing* untuk melakukan pengujian menggunakan kasus uji dengan menerapkan *use scenario* yang sudah ditentukan berada di hasil ekspektasi, agar mengetahui hasil ekspektasi sama dengan hasil sebenarnya telah sesuai atau tidak.

3.7.2 White-Box Testing

Pengujian *white-box* yakni merupakan pengujian yang memastikan bahwa semua jalur harus terlewati atau dieksekusi meskipun cuma sekali. *Basis path testing* teknik yang digunakan dalam pengujian *white-box* digunakan untuk menguji operasi setiap klas yang ada pada tahap perancangan. Pengujian yang dilakukan dengan menggambarkan alur algoritme sistem. *Node* dan *edge* digunakan untuk menggambarkan suatu kondisi pada algoritme. Pengujian ini dilakukan untuk mengetahui jalur eksekusi yang harus dilalui berdasarkan perhitungan *cyclomatic complexity*. Kasus uji dibuat pada setiap jalur eksekusi. Pengujian ini sendiri memiliki flow graph untuk merepresentasikan jalan yang ada pada suatu program.

3.7.3 Compatibility Testing

Pengujian *compatibility* merupakan pengujian dengan menjalankan sistem pada *web browser*. Pada pengujian *compatibility* menguji apakah sistem yang berada pada *web browser* sudah berjalan dengan baik, serta menjalankan fungsinya masing-masing tanpa ada mengalami gangguan. Ketika sistem sudah menjalankan fungsinya dengan baik maka dapat dikatakan bahwa sistem berhasil dijalankan.

3.8 Pengambilan Kesimpulan dan Saran

Pada tahap akhir pembuatan laporan ini yakni pengambilan kesimpulan dan saran dari sistem yang telah didokumentasikan. Ketika semua tahapan analisis, perancangan, implementasi dan pengujian telah dilakukan. Maka kesimpulan diambil dari analisis dan pengujian. Tahap yang paling akhir yakni penulisan saran, yang digunakan sebagai memperbaiki kekurangan dan menyempurnakan sistem serta memberikan pertimbangan kepada pengembang sistem selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

Bab rekayasa kebutuhan ini menjelaskan tentang analisis kebutuhan sistem manajemen rumah indekos pada rumah indekos. Tahapan yang pertama kali dilakukan untuk membuat sistem yakni analisis kebutuhan. Tahap analisis kebutuhan pada penelitian ini diawali dengan elisitasi kebutuhan, analisis kebutuhan untuk mendapatkan kebutuhan secara umum, spesifikasi kebutuhan digunakan untuk mendapatkan kebutuhan fungsional dan non fungsional serta penggambaran *use case* dan *activity diagram*, dan yang terakhir validasi dan verifikasi untuk mengetahui kebutuhan yang valid dan terverifikasi. Namun tahap validasi dan verifikasi tidak dipublikasikan dalam dokumen ini.

4.1 Elitisasi dan Analisis Kebutuhan

Tahapan ini untuk mengetahui masalah-masalah yang perlu dipecahkan. Langkah-langkah yang dilakukan yakni, melakukan wawancara secara langsung kepada pemilik rumah indekos. Berlandaskan hasil wawancara yang terlampir pada lampiran A mengenai beberapa kendala yang dihadapi pemilik indekos. Kendala pertama, dalam pengelolahan rumah indekos yang berbeda lokasi masih menggunakan pendataan dengan cara yang masih konvensional. Pendataan transaksi pembayaran indekos menggunakan aplikasi *spreadsheet* yaitu Microsoft Excel. Sehingga memungkinkan terjadi kesalahan penulisan nama dan penghuni kamar berada di nomor berapa ketika mulai menempati kamar indekos pada saat pembaruan data. Kendala selanjutnya, sering terlambatnya atau lupa untuk memperbarui data terhadap penghuni indekos pada saat melakukan berpindah kamar dari satu kamar ke kamar yang lain yang dilakukan oleh penghuni indekos.

Berlandaskan masalah tersebut, sistem yang nantinya akan dibangun diharapkan dapat mengatasinya. Masalah-masalah tersebut nantinya sebagai lingkup untuk pembuatan sistem. Sistem ini meliputi tiga ruang lingkup, yaitu pengelolahan rumah indekos, data penghuni, dan laporan bukti pembayaran. Dalam lingkup pengelolahan rumah indekos, sistem harus dapat menambahkan rumah indekos. Hal ini berfungsi sebagai pembeda antara rumah indekos yang satu dengan yang lain ketika pemilik indekos memiliki lebih dari satu rumah indekos.

Dalam ruang lingkup data penghuni, sistem harus dapat menampilkan data-data dari penghuni indekos, menghapus penghuni indekos dan mengubah data penghuni indekos. Hal ini bertujuan untuk membantu pembaruan data penghuni yang dilakukan oleh pemilik indekos. Setelah proses pembaruan data yang dilakukan oleh pemilik indekos, sistem dapat menampilkan data yang telah diperbarui oleh pemilik indekos.

Dalam ruang lingkup laporan, sistem harus dapat menampilkan laporan setiap bulannya. Hal ini bertujuan sebagai pengganti dari bukti laporan secara konvensional yang selama ini digunakan oleh pemilik indekos ketika melakukan

pencatatan. Serta mengurangi risiko ketika buku catatan hilang yang dimiliki oleh pemilik indekos. Selain sebagai pengganti dari buku konvesional, fitur ini sebagai arsip ketika sewaktu-waktu pemilik membutuhkannya.

Dalam ruang lingkup penghuni indekos, sistem harus dapat menampilkan transparansi data yang sudah dibayarkan tiap bulannya. Serta penghuni dapat melaporkan kerusakan apa saja yang terjadi di rumah indekos ataupun di kamar penghuni.

Kebutuhan yang harus ada di dalam sistem berdasarkan penjabaran diatas. Kebutuhan yang harus ada dapat dilihat pada Tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional

Tipe Kebutuhan	Keterangan Kebutuhan	Solusi Untuk Masalah
Fungsional	Sistem perlu menyediakan fitur untuk hak akses pengguna	Fasilitas untuk daftar ke dalam sistem
Fungsional	Sistem perlu menyediakan fitur untuk hak akses pengguna	Fasilitas untuk masuk ke dalam sistem
Fungsional	Sistem perlu menyediakan fasilitas untuk mengelola rumah indekos	Fasilitas untuk kegiatan pengelolaan rumah indekos
Fungsional	Sistem perlu menyediakan fasilitas untuk mengelola data penghuni indekos	Fasilitas untuk kegiatan pengelolaan data penghuni indekos
Fungsional	Sistem perlu menyediakan fasilitas laporan	Fasilitas untuk kegiatan laporan bukti pembayaran
Fungsional	Sistem perlu menyediakan pengingat pembayaran	Fasilitas untuk <i>reminder</i> penghuni
Fungsional	Sistem perlu menyediakan fasilitas kerusakan	Fasilitas untuk melaporakan kerusakan
Fungsional	Sistem perlu menyediakan fasilitas untuk keluar dari hak akses pengguna	Fasilitas untuk <i>Logout</i> dari sistem

4.2 Identifikasi Aktor

Identifikasi aktor yakni aktor atau seseorang yang berhubungan langsung dengan sistem. Tahapan ini memiliki tujuan untuk mengidentifikasi aktor-aktor yang berhubungan langsung dengan sistem. Terdapat dua kolom yang diantaranya kolom aktor dan kolom deskripsi. Kolom aktor disebutkan aktor-aktor yang berperan di dalam sistem. Kolom deskripsi penjelasan dari masing-

masing aktor yang mengacu pada tahap elisitasi. Tabel 4.2 menunjukkan identifikasi aktor pada sistem.

Tabel 4.2 Identifikasi Aktor

Aktor	Deskripsi
User	Aktor yang belum masuk ke dalam sistem, sehingga aktor tersebut hanya bisa melakukan mendaftar ke sistem atau masuk ke sistem.
Pemilik	Aktor yang menggunakan sistem dan memiliki otorisasi untuk mengelola rumah indekos, mengelola data penghuni dan melihat laporan pengeluaran.
Penghuni	Aktor yang menggunakan sistem dan memiliki otorisasi untuk mengetahui transparansi data pembayaran dan melaporkan kerusakan yang berada di rumah indekos ataupun di kamar indekos.

4.3 Spesifikasi Kebutuhan

Di dalam spesifikasi kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional dari sistem yang akan dibuat.

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan-kebutuhan yang berikan sebuah proses-proses apa saja yang akan disediakan di dalam sistem. Sistem sendiri nantinya harus berfokus pada inputan dan situasi tertentu. Pada tahap ini bertujuan untuk menjelaskan fitur-fitur yang dimiliki sistem. Sebagaimana fitur tersebut akan digunakan oleh aktor untuk memenuhi kebutuhan informasi. Fungsionalitas yang dapat dilakukan oleh aktor pada Tabel 4.3 berikut ini.

Tabel 4.3 Kebutuhan Fungsional

No SRS	Kebutuhan	Use case	Aktor
SRS_1_01	Sistem perlu menyediakan untuk hak akses pengguna	Mendaftar ke Sistem	Pemilik
SRS_1_02	Sistem perlu menyediakan untuk masuk hak akses pengguna	Masuk ke dalam sistem	Pemilik dan penghuni
SRS_1_03	Sistem mampu untuk menambah rumah indekos	Menambahkan rumah indekos	Pemilik

Tabel 4.3 Kebutuhan Fungsional (lanjutan)

No SRS	Kebutuhan	Use case	Aktor
SRS_1_04	Sistem mampu untuk menambahkan penghuni	Menambahkan penghuni	Pemilik
SRS_1_05	Sistem mampu untuk mengubah data penghuni	Mengubah penghuni	Pemilik
SRS_1_06	Sistem perlu menyediakan tombol menghapus penghuni	Menghapus penghuni	Pemilik
SRS_1_07	Sistem perlu menyediakan untuk pemesanan kamar	Pemesanan kamar	Pemilik
SRS_1_08	Sistem mampu mengirimkan notifikasi batas pembayaran	Mengirimkan notifikasi batas pembayaran	Pemilik
SRS_1_09	Sistem perlu menyediakan untuk mencetak bukti pembayaran	Mencetak laporan bukti pembayaran	Penghuni
SRS_1_10	Sistem perlu menyediakan untuk melihat status pembayaran	Melihat status pembayaran	Penghuni
SRS_1_11	Sistem perlu menyediakan untuk melihat data pembayaran	Melihat data pembayaran	Penghuni
SRS_1_12	Sistem perlu menyediakan untuk melaporkan kerusakan	Melaporkan kerusakan	Penghuni
SRS_1_13	Sistem perlu menyediakan untuk menambahkan akun penghuni	Tambah Akun Penghuni	Pemilik

SRS_1_14	Sistem perlu menyediakan untuk menambahkan pembayaran	Menambahkan pembayaran	Pemilik
SRS_1_15	Sistem perlu menyediakan untuk mengubah pembayaran	Mengubah pembayaran	Pemilik
SRS_1_16	Sistem perlu menyediakan untuk menambahkan pengeluaran	Menambahkan pengeluaran	Pemilik
SRS_1_17	Sistem perlu menyediakan untuk mengubah pemesanan	Mengubah pemesanan	Pemilik
SRS_1_18	Sistem perlu menyediakan untuk verifikasi kendala	Verifikasi kendala	Pemilik
SRS_1_19	Sistem perlu menyediakan untuk keluar dari sistem	Keluar dari sistem	Pemilik dan penghuni

4.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah kebutuhan yang mengacu pada perilaku yang dimiliki sistem tersebut dan menentukan kualitas atau atribut secara keseluruhan pada sistem. Kebutuhan ini sering disebut sebagai sebuah batasan-batasan fungsi atau layanan yang ada di sistem. Pada tahap ini bertujuan untuk menjelaskan kebutuhan non fungsional yang dimiliki sistem. Pada Tabel 4.4 menunjukkan daftar kebutuhan non fungsional pada sistem.

Tabel 4.4 Kebutuhan Non-Fungsional

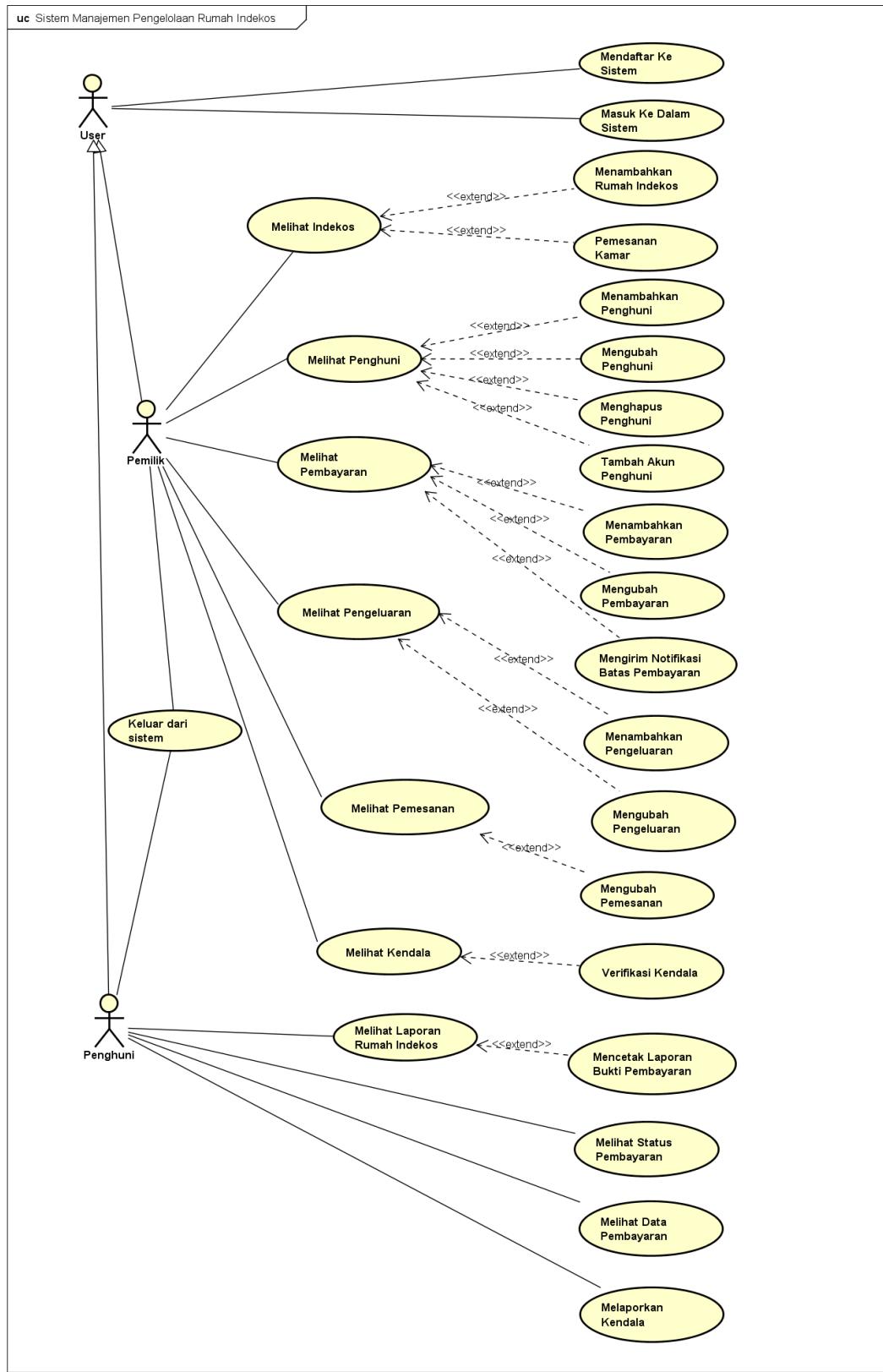
No.	Parameter	Deskripsi
1.	<i>Compatibility</i>	Sistem dapat dibuka pada sepuluh macam <i>web browser</i>

4.3.3 Use Case Diagram

Use case diagram merupakan bentuk perancangan atau pemodelan dari perilaku (*behavior*) dari sistem atau perangkat lunak. *Use case* menjelaskan interaksi perilaku yang diantaranya antara aktor Pemilik dengan sistem atau perangkat lunak. *Use case* tersebut digunakan untuk mengetahui fungsi-fungsi apa saja yang berada di dalam sebuah sistem atau perangkat lunak serta siapa saja yang berhak untuk menggunakan fungsi-fungsi tersebut. Pada diagram *use case* ini, memiliki dua *user* yaitu pemilik dan penghuni indekos. Aktor sebagai *member* pemilik memiliki *use case* melakukan Mendaftar Ke Sistem, Masuk Ke Dalam Sistem, Menambahkan Rumah Indekos, Pemesanan Kamar, Menambahkan Penghuni, Mengubah Penghuni, Menghapus Penghuni, Tambah Akun Penghuni, Menambahkan Pembayaran, Mengubah Pembayaran, Mengirimkan Notifikasi Batas Pembayaran, Menambahkan Pengeluaran, Mengubah Pemesanan, Verifikasi Kendala dan Keluar dari sistem. Aktor sebagai *member* penghuni memiliki *use case* melakukan Mencetak Laporan Bukti Pembayaran, Melihat Status Pembayaran, Melihat Data Pembayaran dan Melaporkan Kendala.

Use case Menambahkan Rumah Indekos, Pemesanan Kamar memiliki hubungan relasi *extend* terhadap *use case* Melihat Indekos. *Use case* Menambahkan Penghuni, Mengubah Penghuni, Menghapus Penghuni, Tambah Akun Penghuni memiliki hubungan relasi *extend* terhadap *use case* Melihat Penghuni. *Use case* Menambahkan Pembayaran, Mengubah Pembayaran, Mengirimkan Notifikasi Batas Pembayaran elasi *extend* terhadap *use case* Melihat Pembayaran. *Use case* Menambah Pengeluaran memiliki hubungan relasi *extend* terhadap *use case* Melihat Pengeluaran. *Use case* Mengubah Pemesanan memiliki hubungan relasi *extend* terhadap *use case* Melihat Pemesanan. *Use case* Verifikasi Kendala memiliki hubungan relasi *extend* terhadap *use case* Melihat Kendala. *Use case* Mencetak Laporan Bukti Pembayaran memiliki hubungan relasi *extend* terhadap *use case* Melihat Laporan Rumah Indekos.

Use case diagram yang digunakan sebagai acuan dalam pengembangan sistem terdapat pada Gambar 4.1.



powered by Astah

Gambar 4.1 Use case diagram

4.3.4 Use Case Scenario

Use case yang telah digambarkan dalam diagram *use case* akan lebih dijelaskan secara terperinci dalam skenario *use case*. *Use case scenario* ini memiliki alur rincian dari *use case diagram* kondisi awal dan akhir yang harus dipenuhi oleh *use case* setelah fungsionalitas selesai dijalankan. Tanggapan sistem terhadap aksi yang diberikan oleh pengguna akan dibahas di *scenario*. Pada tabel 4.5 sampai tabel 4.17 merupakan *use case scenario* untuk masing – masing kebutuhan fungsional sistem.

Tabel 4.5 Use case mendaftar ke sistem

<i>Use Case Name</i>	Mendaftar ke Sistem
<i>Objective</i>	Melakukan registrasi ke sistem
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik berada di halaman utama
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik memilih tombol Daftar 2. Sistem menampilkan <i>form</i> data diri 3. Pemilik mengisi data diri yang sudah disediakan ke dalam sistem 4. Pemilik menekan tombol <i>submit</i> 5. Sistem mengeksekusi yang diinputkan oleh pemilik 6. Jika yang diinputkan tidak sesuai atau kurang sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali 7. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Sistem akan menampilkan pesan “Please fill out this fill” bila data tidak di isi lengkap
<i>Post-Condition</i>	Pemilik terdaftar di dalam sistem

Tabel 4.6 Use case scenario masuk ke dalam sistem

<i>Use Case Name</i>	Masuk Ke Dalam Sistem
<i>Objective</i>	Proses untuk masuk ke dalam sistem
<i>Actors</i>	Pemilik dan penghuni
<i>Pre-Condition</i>	Pemilik berada di halaman utama <i>website</i> dan telah melakukan mendaftar ke sistem
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik dan penghuni memilih tombol <i>Login</i> 2. Sistem menampilkan <i>form</i> untuk masuk ke sistem 3. Pemilik dan penghuni memasukkan <i>email</i> dan <i>password</i> yang sudah didaftarkan ke dalam sistem

Tabel 4.6 Use case scenario masuk ke dalam sistem (lanjutan)

<i>Main flow</i>	<ol style="list-style-type: none">4. Sistem akan mengeksekusi validasi <i>email</i> dan <i>password</i> yang telah tersimpan di <i>database</i>5. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali6. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Sistem akan menampilkan pesan “ <i>Username</i> atau <i>password</i> salah !!!!”
<i>Post-Condition</i>	Pemilik dan penghuni dapat melihat data yang dimiliki di masing-masing halaman

Tabel 4.7 Use case scenario menambahkan rumah indekos

<i>Use Case Name</i>	Menambahkan Rumah Indekos
<i>Objective</i>	Menambahkan rumah indekos baru yang dimiliki oleh pemilik
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik
<i>Main flow</i>	<ol style="list-style-type: none">1. Pemilik memilih menu rumah indekos2. Sistem menampilkan data rumah indekos yang dimiliki oleh pemilik3. Pemilik memilih <i>icon</i> tambah rumah indekos4. Pemilik dapat menambahkan rumah indekos baru dengan mengisi <i>form</i> yang tersedia5. Pemilik menekan tombol simpan6. Sistem mengeksekusi yang diinputkan pemilik7. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali8. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pesan “Please fill out this fill” bila data tidak diisi lengkap

Tabel 4.7 Use case scenario menambahkan rumah indekos (lanjutan)

<i>Post-Condition</i>	Pemilik berhasil menambahkan rumah indekos baru
-----------------------	---

Tabel 4.8 Use case scenario pemesanan kamar

<i>Use Case Name</i>	Pemesanan kamar
<i>Objective</i>	Memesan kamar dengan jangka waktu tertentu agar tidak ditempati orang terlebih dahulu
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik memilih menu rumah indekos 2. Sistem menampilkan data rumah indekos yang dimiliki oleh pemilik 3. Pemilik memilih rumah indekos 4. Sistem menampilkan <i>form</i> pemesanan 5. Pemilik dapat menambahkan pemesanan kamar dengan mengisi <i>form</i> yang tersedia 6. Pemilik menekan tombol simpan 7. Sistem mengeksekusi yang diinputkan pemilik 8. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali 9. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pesan bila data tidak diisi lengkap
<i>Post-Condition</i>	Pemilik berhasil melakukan pemesanan kamar dengan jangka waktu tertentu di sistem

Tabel 4.9 Use case menambahkan penghuni

<i>Use Case Name</i>	Menambahkan Penghuni
<i>Objective</i>	Menambahkan penghuni indekos di rumah indekos tertentu

Tabel 4.9 Use case menambahkan penghuni (lanjutan)

<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik
<i>Main flow</i>	<ol style="list-style-type: none">1. Pemilik memilih menu penghuni2. Sistem menampilkan data-data penghuni3. Pemilik memilih <i>icon</i> tambah penghuni4. Sistem menampilkan <i>form</i> tambah penghuni5. Pemilik dapat mengisi <i>form</i> yang tersedia6. Pemilik dapat menekan tombol simpan7. Sistem mengeksekusi yang diinputkan pemilik8. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali9. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil10. Penghuni baru sudah ditambahkan pada rumah indekos tertentu
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Post-Condition</i>	Pemilik berhasil menambahkan penghuni baru yang berada di rumah indekos tertentu

Tabel 4.10 Use case scenario mengubah penghuni

<i>Use Case Name</i>	Mengubah Penghuni
<i>Objective</i>	Mengubah data penghuni yang diinginkan pemilik rumah indekos
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik
<i>Main flow</i>	<ol style="list-style-type: none">1. Pemilik memilih menu penghuni2. Sistem menampilkan data-data penghuni3. Pemilik memilih data penghuni yang akan dirubah

Tabel 4.10 Use case scenario mengubah penghuni (lanjutan)

<i>Main flow</i>	<ol style="list-style-type: none">4. Pemilik dapat merubah data lama dan mengganti data baru dengan mengisi <i>form</i> yang tersedia5. Pemilik memilih tombol simpan6. Sistem mengeksekusi yang diinputkan pemilik7. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali8. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil9. Data penghuni baru telah diperbarui
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Pre-Condition</i>	Pemilik berhasil merubah penghuni indekos yang baru

Tabel 4.11 Use case scenario menghapus penghuni

<i>Use Case Name</i>	Menghapus Penghuni
<i>Objective</i>	Menghapus penghuni yang sudah tidak berada di rumah indekos tersebut
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik dan menekan menu penghuni
<i>Main flow</i>	<ol style="list-style-type: none">1. Pemilik memilih menu penghuni2. Sistem menampilkan data-data penghuni3. Pemilik memilih data penghuni yang ingin dihapus4. Pemilik memilih <i>icon</i> hapus penghuni5. Sistem menampilkan <i>alert</i> apakah ingin menghapus6. Pemilik dapat menghapus penghuni
<i>Alternative flows</i>	Jika pemilik menekan tombol “batal” maka sistem tidak akan mengeksekusi hapus penghuni
<i>Post-Condition</i>	Pemilik berhasil mengubah data penghuni baru berada di sistem

Tabel 4.12 Use case scenario menambahkan akun penghuni

<i>Use Case Name</i>	Tambah Akun Penghuni
<i>Objective</i>	Menambahkan hak akses untuk penghuni
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik memilih menu penghuni 2. Sistem menampilkan data-data penghuni 3. Pemilik memilih <i>icon</i> akun penghuni 4. Sistem menampilkan <i>form</i> untuk menambahkan akun penghuni 5. Pemilik dapat menambahkan akun penghuni dengan mengisi <i>form</i> yang tersedia 6. Pemilik memilih tombol simpan 7. Sistem mengeksekusi yang diinputkan pemilik 8. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali 9. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Post-Condition</i>	Pemilik berhasil menambahkan akun penghuni

Tabel 4.13 Use case scenario menambahkan pembayaran

<i>Use Case Name</i>	Menambahkan Pembayaran
<i>Objective</i>	Menambahkan pembayaran yang telah dilakukan oleh penghuni
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik memilih menu pembayaran 2. Sistem menampilkan data-data pembayaran 3. Pemilik memilih <i>icon</i> pembayaran 4. Sistem menampilkan <i>form</i> tambah

	pembayaran
--	------------

Tabel 4.13 Use case scenario menambahkan pembayaran (lanjutan)

	<ul style="list-style-type: none"> 5. Pemilik dapat menambahkan pembayaran dengan mengisi <i>form</i> yang tersedia 6. Pemilik memilih tombol simpan 7. Sistem mengeksekusi yang diinputkan pemilik 8. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali 9. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Post-Condition</i>	Pemilik berhasil menambahkan pembayaran indekos

Tabel 4.14 Use case scenario mengubah pembayaran

<i>Use Case Name</i>	Mengubah Pembayaran
<i>Objective</i>	Menambahkan pembayaran yang telah dilakukan oleh penghuni
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik
<i>Main flow</i>	<ul style="list-style-type: none"> 1. Pemilik memilih menu pembayaran 2. Sistem menampilkan data-data pembayaran 3. Pemilik memilih penghuni yang telah membayar dan memilih <i>icon</i> ubah pembayaran 4. Sistem menampilkan <i>form</i> pembayaran yang akan dirubah 5. Pemilik dapat mengubah pembayaran dengan mengisi <i>form</i> yang tersedia 6. Pemilik memilih tombol simpan 7. Sistem mengeksekusi yang diinputkan pemilik 8. Jika yang diinputkan tidak sesuai sistem

	akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali
--	---

Tabel 4.14 Use case scenario mengubah pembayaran (lanjutan)

	9. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Post-Condition</i>	Pemilik berhasil mengubah pembayaran indekos

Tabel 4.15 Use case scenario mengirimkan notifikasi batas pembayaran

<i>Use Case Name</i>	Mengirimkan Notifikasi Batas Pembayaran
<i>Objective</i>	Dilakukan untuk mengirimkan notifikasi batas pembayaran kepada penghuni
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik memilih <i>menu pembayaran</i> 2. Sistem menampilkan data-data pembayaran 3. Pemilik menekan <i>icon reminder</i> 4. Sistem menampilkan <i>form reminder</i> 5. Pemilik mengisi <i>form</i> yang telah disediakan di dalam sistem 6. Pemilik menekan tombol <i>submit</i> 7. Sistem mengeksekusi yang diinputkan pemilik 8. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali 9. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Post-Condition</i>	Pemilik berhasil mengirimkan notifikasi batas pembayaran

Tabel 4.16 Use case scenario menambahkan pengeluaran

<i>Use Case Name</i>	Menambahkan Pengeluaran
<i>Objective</i>	Menambahkan biaya operasional rumah indekos
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik dan menekan menu penghuni
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik memilih menu pengeluaran 2. Sistem menampilkan data-data pengeluaran 3. Pemilik memilih <i>icon</i> menambahkan pengeluaran 4. Sistem menampilkan <i>form</i> pengeluaran 5. Pemilik dapat menambahkan pengeluaran dengan mengisi <i>form</i> yang tersedia 6. Pemilik menekan tombol simpan 7. Sistem mengeksekusi yang diinputkan pemilik 8. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali 9. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Post-Condition</i>	Pemilik berhasil menambahkan biaya pengeluaran operasional rumah indekos

Tabel 4.17 Use case scenario mengubah pemesanan

<i>Use Case Name</i>	Mengubah Pemesanan
<i>Objective</i>	Dilakukan untuk mengubah data pemesanan
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik memilih menu <i>list</i> pemesanan 2. Sistem menampilkan data pemesanan

	3. Pemilik memilih nama penghuni yang melakukan pemesanan
--	---

Tabel 4.17 Use case scenario mengubah pemesanan (lanjutan)

	4. Sistem menampilkan data penghuni yang dipilih 5. Pemilik mengisi <i>form</i> yang telah disediakan di dalam sistem 6. Pemilik menekan tombol simpan 7. Sistem mengeksekusi yang diinputkan pemilik 8. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali 9. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Post-Condition</i>	Pemilik berhasil mengubah data pemesanan

Tabel 4.18 Use case scenario verifikasi kendala

<i>Use Case Name</i>	Verifikasi Kendala
<i>Objective</i>	Melakukan verifikasi kendala yang telah dikirimkan oleh penghuni
<i>Actors</i>	Pemilik
<i>Pre-Condition</i>	Pemilik telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik dan menekan menu penghuni
<i>Main flow</i>	1. Pemilik memilih menu <i>list</i> kendala 2. Sistem menampilkan data kendala 3. Pemilik memilih <i>icon</i> verifikasi kendala 4. Sistem menampilkan status kendala 5. Pemilik memilih status kendala 6. Pemilik menekan tombol simpan 7. Sistem mengeksekusikan yang diinputkan oleh pemilik 8. Pemilik melihat status kendala yang telah diverifikasi
<i>Alternative flows</i>	-
<i>Post-Condition</i>	Pemilik berhasil memverifikasi laporan

	kendala dari penghuni
--	-----------------------

Tabel 4.19 Use case scenario mencetak laporan bukti pembayaran rumah indekos

<i>Use Case Name</i>	Mencetak Laporan Pembayaran Rumah Indekos
<i>Objective</i>	Mencetak laporan pembayaran berdasarkan kebutuhan yang diinginkan
<i>Actors</i>	Penghuni
<i>Pre-Condition</i>	Penghuni indekos telah terdaftar dalam sistem dan telah melakukan <i>login</i> sebagai pemilik
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Penghuni memilih menu laporan 2. Sistem menampilkan data keuangan 3. Penghuni memilih menu cetak laporan pembayaran 4. Sistem akan menampilkan laporan yang akan dicetak
<i>Alternative flows</i>	-
<i>Post-Condition</i>	Penghuni berhasil mencetak laporan data pembayaran indekos yang berada di sistem

Tabel 4.20 Use case scenario keluar dari sistem

<i>Use Case Name</i>	Keluar dari sistem
<i>Objective</i>	Dilakukan untuk keluar dari <i>login session</i>
<i>Actors</i>	Pemilik dan penghuni
<i>Pre-Condition</i>	Pemilik dan penghuni telah terdaftar dalam sistem dan telah melakukan <i>login</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pemilik dan penghuni memilih menu <i>logout</i> 2. Sistem menghapus <i>Session</i> 3. Pemilik dan penghuni keluar dari login session sistem
<i>Alternative flows</i>	-
<i>Post-Condition</i>	Pemilik dan penghuni berhasil keluar dari sistem

Tabel 4.21 Use case scenario melihat status pembayaran

<i>Use Case Name</i>	Melihat Status Pembayaran
<i>Objective</i>	Dilakukan untuk melihat apakah penghuni sudah membayar atau belum

<i>Actors</i>	Penghuni
<i>Pre-Condition</i>	Penghuni telah terdaftar dalam sistem dan telah melakukan <i>login</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Penghuni memilih menu lihat status pembayaran 2. Sistem menampilkan status pembayaran 3. Penghuni melihat status terakhir pembayaran
<i>Alternative flows</i>	-
<i>Post-Condition</i>	Penghuni berhasil melihat status pembayaran

Tabel 4.22 Use case scenario melihat data pembayaran

<i>Use Case Name</i>	Melihat Data Pembayaran
<i>Objective</i>	Dilakukan untuk transparansi antara pemilik dan penghuni agar tidak ada kesalahan
<i>Actors</i>	Penghuni
<i>Pre-Condition</i>	Penghuni telah terdaftar dalam sistem dan telah melakukan <i>login</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Penghuni memilih menu lihat data pembayaran 2. Sistem menampilkan data pembayaran 3. Penghuni melihat data pembayaran
<i>Alternative flows</i>	-
<i>Post-Condition</i>	Penghuni berhasil melihat data pembayaran

Tabel 4.23 Use case scenario melaporkan kendala

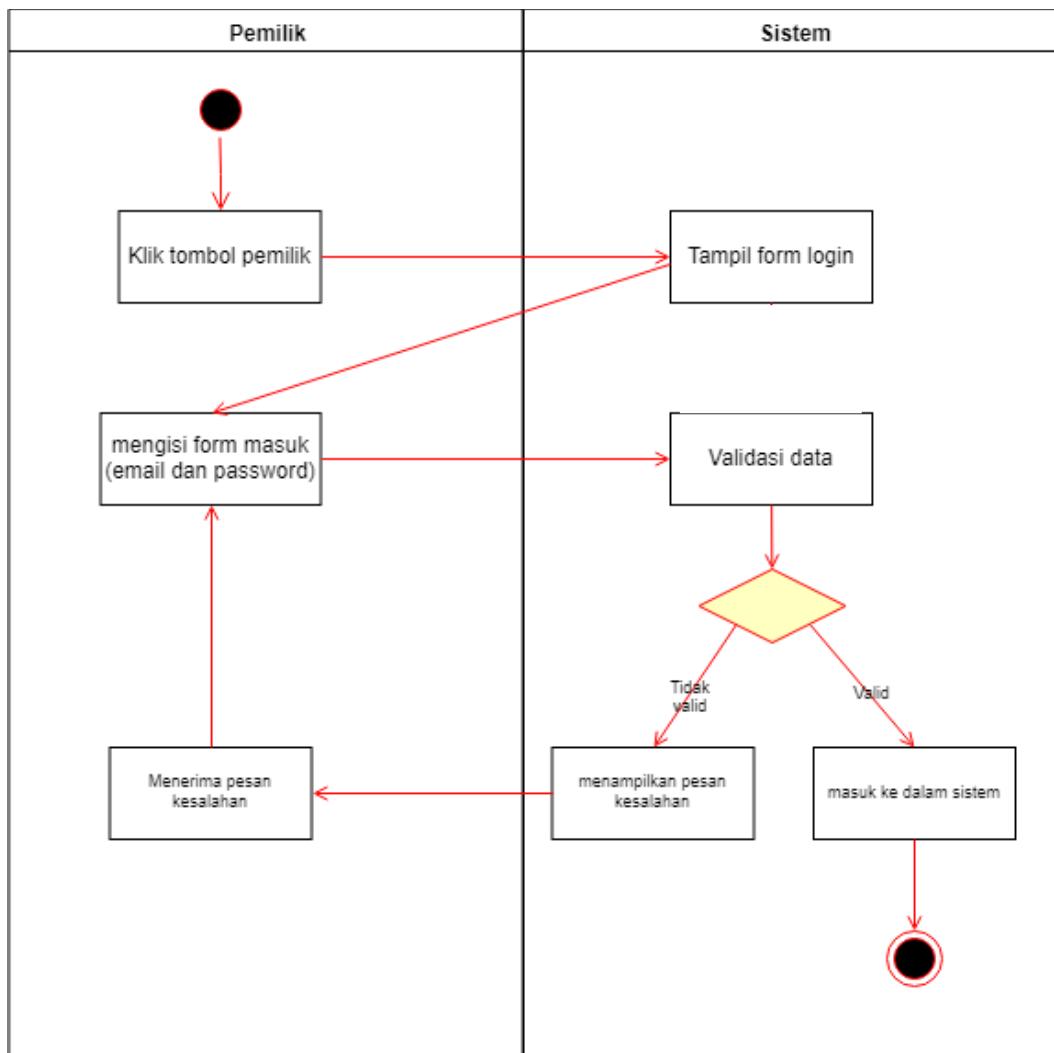
<i>Use Case Name</i>	Melaporkan Kendala
<i>Objective</i>	Dilakukan untuk melaporkan kerusakan yang ada di rumah indekos
<i>Actors</i>	Penghuni
<i>Pre-Condition</i>	Penghuni telah terdaftar dalam sistem dan telah melakukan <i>login</i>
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Penghuni memilih menu melaporkan kerusakan 2. Sistem menampilkan <i>form</i> kerusakan 3. Penghuni mengisi <i>form</i> yang telah disediakan di dalam sistem 4. Penghuni menekan tombol <i>submit</i> 5. Sistem mengeksekusi yang diinputkan pemilik

Tabel 4.24 Use case scenario melaporkan kendala (lanjutan)

<i>Main flow</i>	6. Jika yang diinputkan tidak sesuai sistem akan menampilkan notifikasi gagal, maka pemilik diminta untuk mengisi <i>form</i> kembali 7. Jika yang diinputkan sesuai maka akan menampilkan notifikasi berhasil
<i>Alternative flows</i>	Penghuni tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
<i>Post-Condition</i>	Penghuni berhasil melihat data pembayaran

4.3.5 Activity Diagram

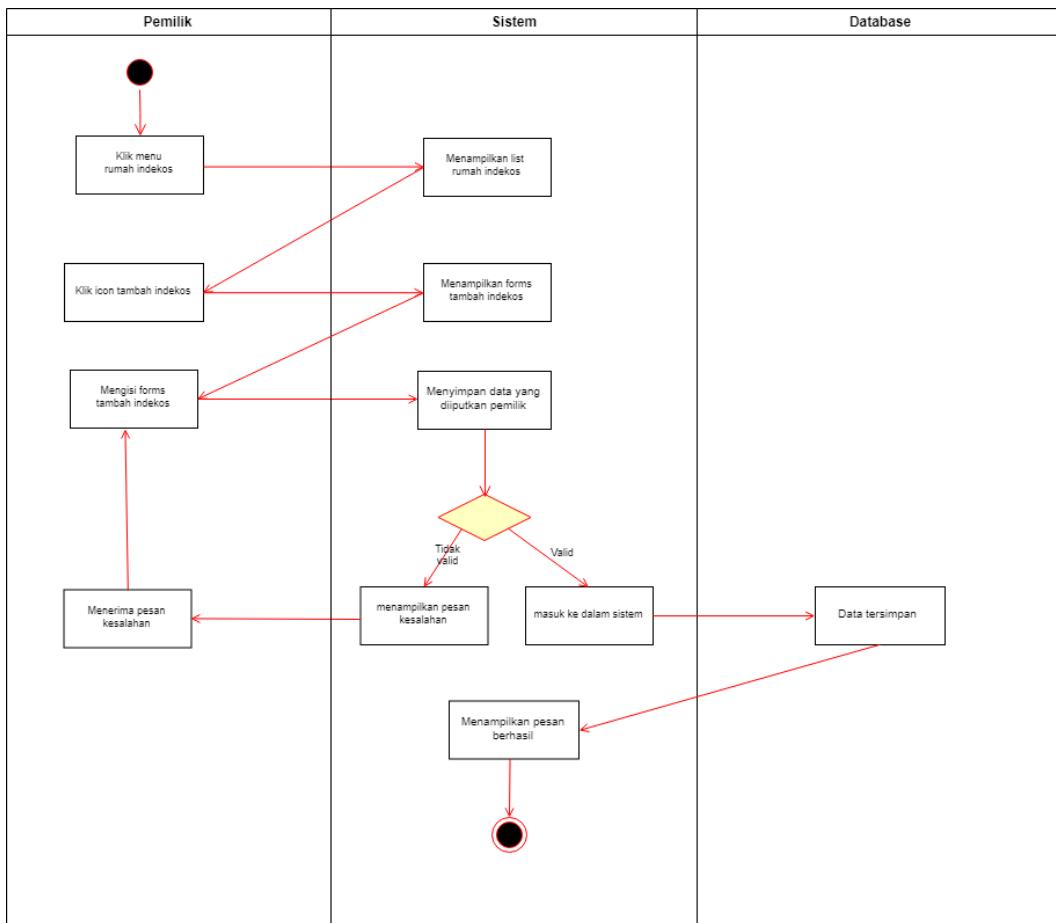
4.3.5.1 Activity Diagram untuk masuk ke sistem



Gambar 4.2 Activity diagram masuk ke dalam

Pada kasus yang tertera diatas untuk melakukan masuk ke dalam sistem, pertama pemilik menekan tombol *login* pemilik, kemudian sistem menampilkan *form*. Setelah pemilik melihat *form*, melakukan pengisian *field* dengan lengkap. Kemudian data dikirim ke sistem. Sistem akan melakukan validasi apakah data tersebut *valid* atau tidak *valid*. Kalau data tidak *valid*, maka sistem akan menampilkan pesan kesalahan kepada pemilik untuk melakukan pengisian ulang. Ketika data sudah *valid*. Maka pemilik dapat masuk ke dalam sistem.

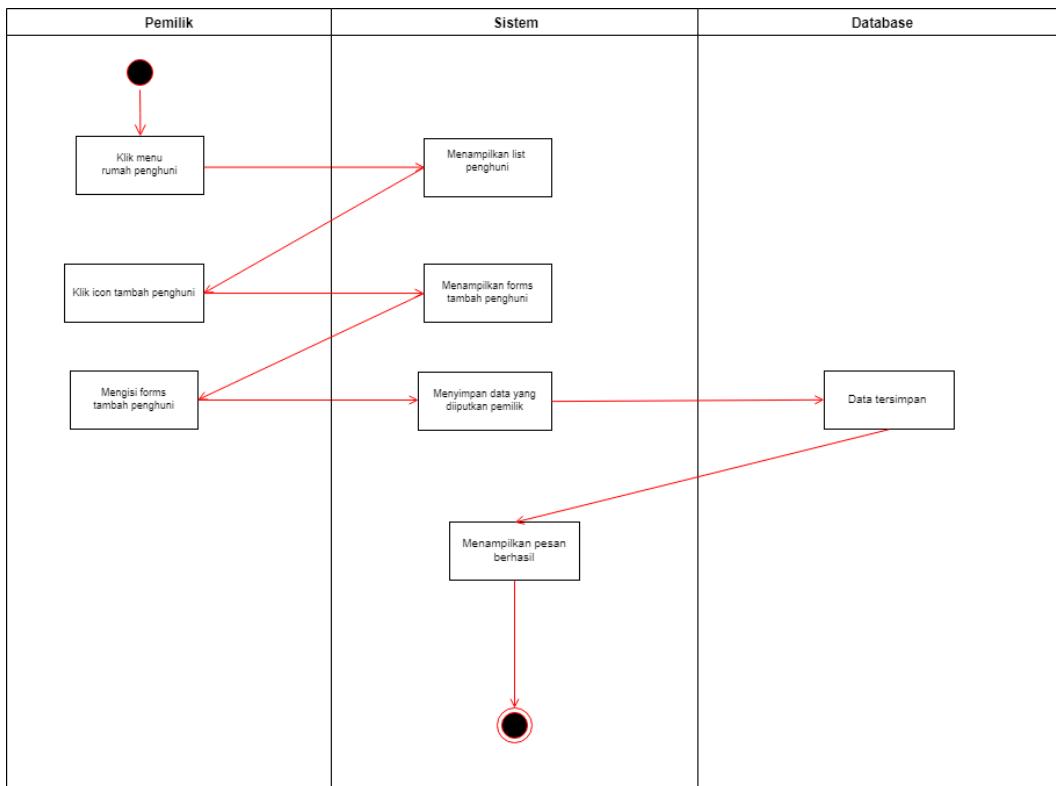
4.3.5.2 Activity Diagram untuk menambahkan rumah indekos



Gambar 4.3 Activity diagram menambahkan rumah indekos

Pada kasus yang tertera diatas untuk melakukan menambahkan rumah indekos. Pertama pemilik menekan menu rumah indekos. Sistem akan menampilkan *list* rumah indekos. Kemudian pemilik menekan *icon* tambah indekos, sistem akan menampilkan *form* yang untuk di isi oleh pemilik. Pemilik mengisi *form* dengan lengkap. Setelah data sudah di isi, sistem akan menyimpan data yang dari pemilik kemudian dicek apakah data yang dimasukkan *valid* atau tidak *valid*. Ketika data tidak valid, sistem akan menampilkan pesan kesalahan. Ketika pemilik sudah melihat pesan kesalahan, pemilik melakukan pengisian ulang tambah rumah indekos. Data yang sudah valid, akan disimpan ke *database*. Kemudian sistem akan menampilkan pesan bahwa data yang diinputkan sudah berhasil.

4.3.5.3 Activity Diagram untuk menambahkan penghuni



Gambar 4.4 Activity diagram menambahkan penghuni

Pada kasus yang tertera diatas untuk melakukan menambahkan penghuni. Pertama pemilik menekan menu penghuni. Kemudian sistem akan menampilkan *list* penghuni. Selanjutnya pemilik dapat menekan tombol tambah penghuni. Sistem akan menampilkan *forms* untuk tambah penghuni. Setelah sistem menampilkan *forms* untuk menambahkan penghuni, pemilik dapat mengisi data dengan lengkap dan benar pada *forms* tersebut. Setelah mengisi, data tambah penghuni tersebut disimpan oleh sistem. Kemudian di *parshing* ke *database* untuk disimpan. Ketika sudah berhasil, sistem akan menampilkan pesan bahwa data yang dimasukkan berhasil disimpan.

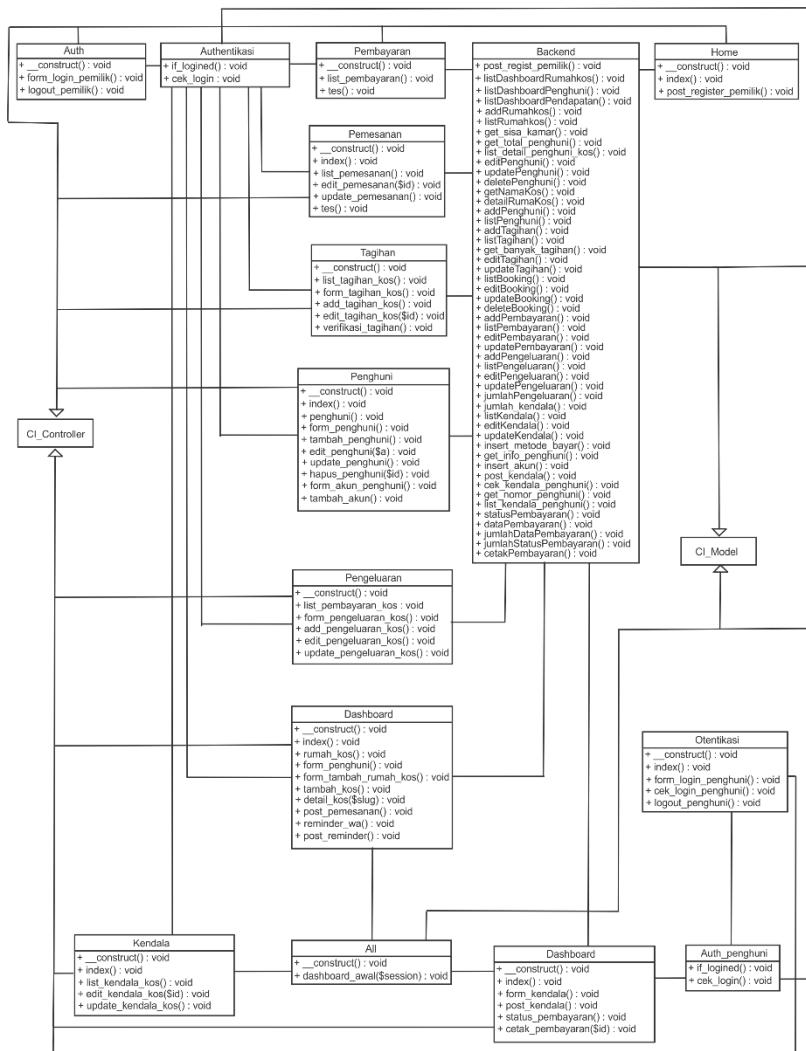
BAB 5 PERANCANGAN DAN IMPLEMENTASI

Setelah melakukan tahapan pada analisis kebutuhan, tahap selanjutnya yang dilakukan yaitu perancangan sistem. Perancangan yang akan dilakukan diantaranya perancangan arsitektur sistem, perancangan komponen, perancangan data, perancangan antarmuka dan perancangan *screen flow*. Kemudian pada tahapan implementasi akan dilakukan implementasi data, implementasi kode program dan implementasi antarmuka.

5.1 Perancangan

Tahap selanjutnya yang akan dilakukan setelah melakukan proses analisis kebutuhan, tahap berikutnya adalah perancangan. Perancangan dilakukan berdasarkan hasil dari analisis kebutuhan yang telah dibuat.

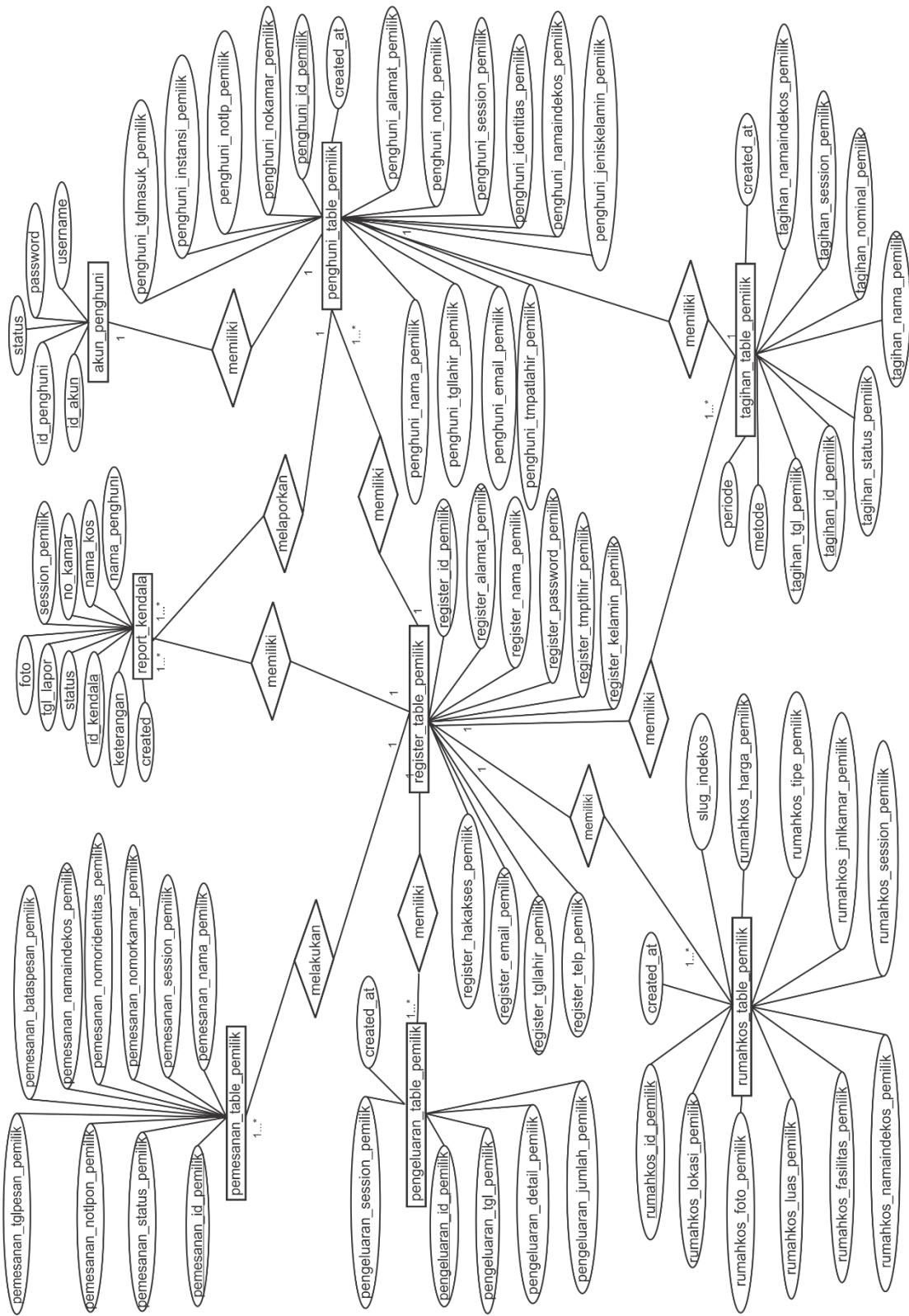
5.1.1 Class Diagram



Gambar 5.1 Perancangan *class diagram* sistem manajemen rumah indekos

5.1.2 Entity Relation Diagram

Gambar 5.2 Perancangan Entity Relation Diagram



Entity Relation Diagram (ERD) biasanya digunakan untuk menggambarkan hubungan antara entitas-entitas yang berada di dalam sistem. Nantinya akan digunakan sebagai dasar untuk melakukan implementasi basis data.

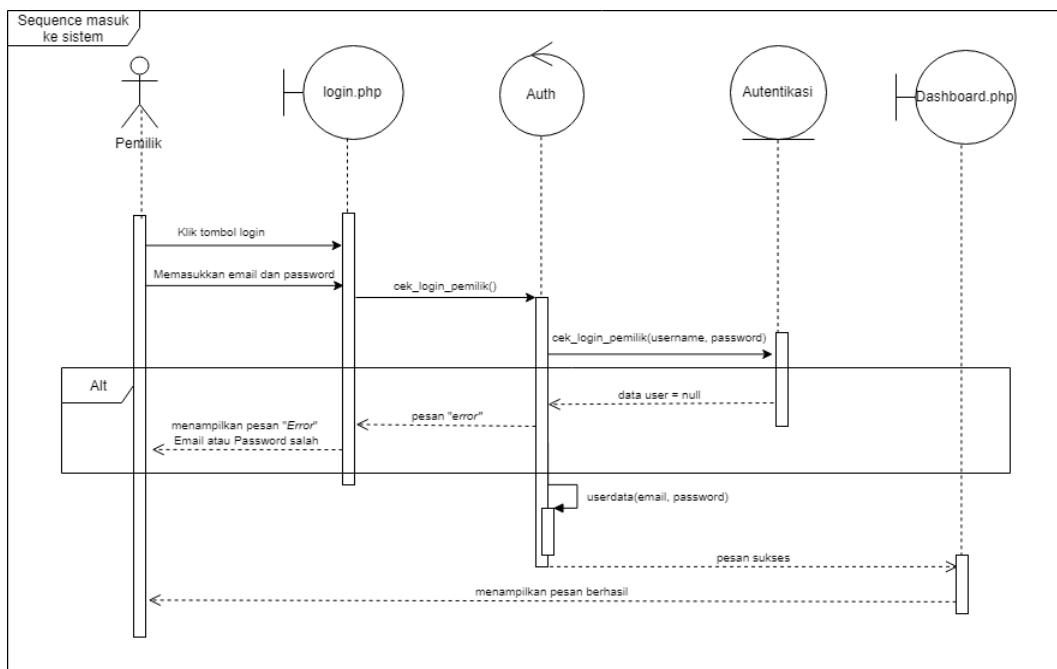
Pada gambar 5.2 di atas menggambarkan hubungan antara entitas pada sistem rumah indekos. *Database* rumah indekos memiliki beberapa entitas yaitu pemesanan, rumahkos, kendala, akun_penghuni, pengeluaran, register,

penghuni, pembayaran dan tagihan . Entitas register mempunyai hubungan *one-to-one* dengan entitas pengeluaran. Entitas register memiliki hubungan *one-to-many* dengan entitas rumahkos. Entitas register memiliki hubungan *one-to-many* dengan entitas kendala. Entitas register memiliki hubungan *one-to-many* dengan entitas penghuni. Entitas penghuni memiliki hubungan *one-to-one* dengan entitas akun_penghuni. Entitas penghuni memiliki hubungan *one-to-many* dengan entitas kendala. Entitas penghuni memiliki hubungan *one-to-one* dengan entitas tagihan. Entitas register memiliki hubungan *one-to-many* dengan entitas pembayaran.

5.1.3 Pemodelan Sequence Diagram

Sequence diagram pada sistem rumah indekos menjelaskan mengenai tentang urutan proses yang terjadi dalam mencapai kebutuhan sistem. *Sequence diagram* dapat dibuat berdasarkan *use case scenario* dari hasil indentifikasi kebutuhan sistem. Satu *use case* dapat digambarkan menggunakan satu *sequence diagram*.

1. Sequence diagram Masuk ke Dalam Sistem

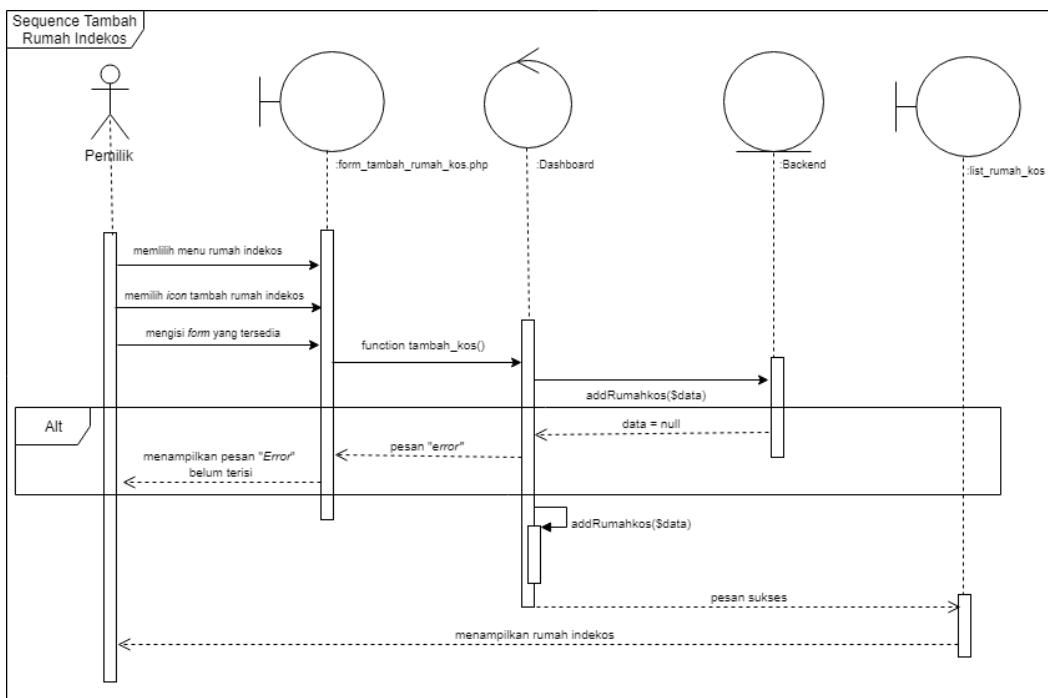


Gambar 5.3 *Sequence diagram* masuk ke dalam sistem

Pada gambar 5.3 menggambarkan interaksi antar objek yang saling berhubungan dalam fungsi *login*. Objek yang terlibat di dalamnya antara lain adalah aktor pemilik, *boundary* login.php dan Dashboard.php, *control* Auth dan *Entity* Autentikasi. Pemilik sebagai aktor tanpa otoritas akan melakukan *login* kedalam sistem sebelum dapat akses sistem rumah indekos dengan memasukkan data *email* dan *password*, kemudian *control* Auth menjalankan proses *login* dengan *method* *cek_login()*. Selanjutnya data yang telah dimasukkan akan

diperiksa di dalam *entity* Autentikasi dengan menggunakan *method* `cek_login_pemilik()`. Ketika data tidak ditemukan atau bernilai kosong maka akan menampilkan pesan *error*, namun ketika data ditemukan maka data akan dimasukkan ke dalam *session* dengan *method* `userdata (username,password)`. Ketika sudah masuk kedalam sistem akan menampilkan pesan berhasil.

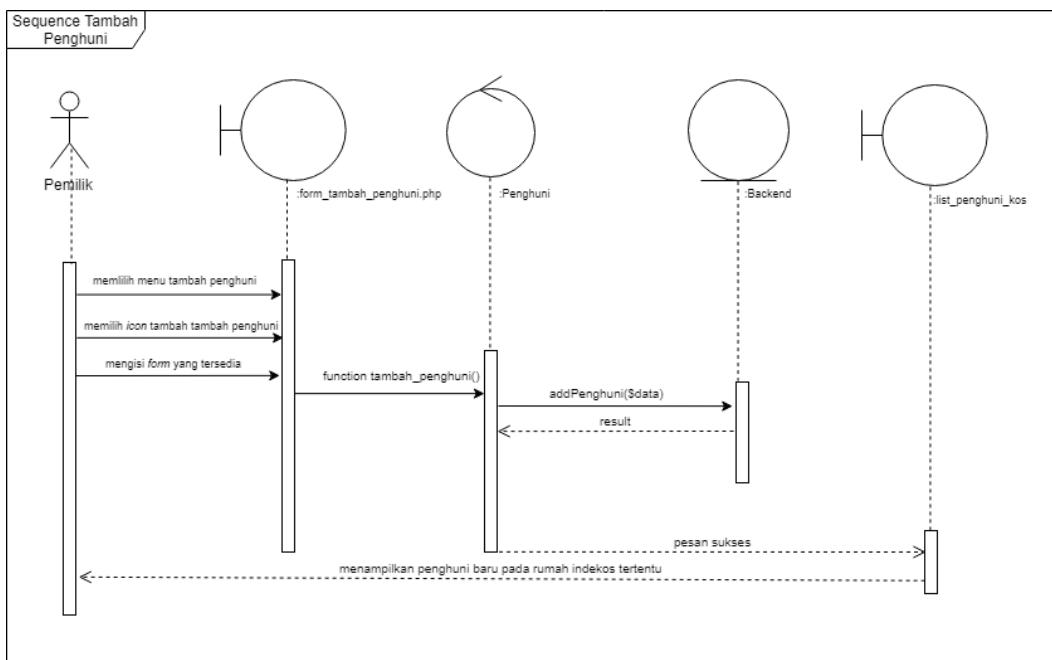
2. Sequence Diagram Menambah Rumah Indekos



Gambar 5.4 Sequence diagram menambahkan rumah indekos

Pada gambar 5.4 menggambarkan interaksi antar objek yang saling berhubungan dalam fungsi Tambah Rumah Indekos. Objek yang terlibat di dalamnya adalah aktor pemilik, *boundary* `list_tambah_rumah_kos` dan `list_rumah`, *control* `Dashboard` dan *Entity* `Backend`. Aktor sebagai pemilik melakukan menambahkan rumah indekos baru dengan memasukkan data yang tersedia di *form* dengan *field* yang tersedia. Kemudian *control* `Dashboard` dengan menjalankan *method* `tambah_kos()`. Selanjutnya data yang dimasukkan akan diperiksa di dalam *entity* `KosModel` dengan menggunakan *method* `addRumahkos()`. Namun ketika data yang dimasukkan tidak *valid* maka akan menampilkan pesan *error*. Ketika data yang dimasukkan sudah berhasil maka akan menampilkan pesan berhasil disimpan.

3. Sequence Diagram Menambahkan Penghuni



Gambar 5.5 *Sequence diagram menambahkan penghuni*

Pada gambar 5.5 berikut menggambarkan interaksi antar objek yang saling berhubungan dalam fungsi Menambahkan Penghuni. Objek yang terlibat di dalamnya adalah aktor *form_tambah_penghuni*, *boundary penghuni* dan *list_penghuni_kos*, *control backend*. Aktor sebagai pemilik melakukan menambahkan rumah indekos dengan menggunakan *method tambah_penghuni()*. Setelah data sudah diinputkan akan di *parhsing* ke *Entity Backend* dengan *method addPenghuni()*. Setelah data baru dimasukkan kemudian dan berhasil, akan menampilkan pesan data berhasil disimpan.

5.1.4 Perancangan *Pseudocode*

Perancangan *pseudocode* merupakan tahap selanjutnya untuk menjelaskan perancangan *pseudocode* sistem manajemen rumah indekos. *Pseudocode* tersebut dirancang dengan berdasarkan setiap operasi pada masing-masing *class diagram* yang telah dibuat. Perancangan *pseudocode* sendiri bertujuan untuk menjelaskan atau menjabarkan algoritma dari fungsi yang berada di dalam *class*. Pada perancangan *pseudocode* ini akan dicantumkan tiga *pseudocode* yang ada di dalam sistem.

5.1.4.1 Perancangan *Pseudocode* Masuk ke Dalam Sistem

Pada Tabel 5.1 perancangan *pseudocode* yang digunakan untuk masuk ke dalam sistem. *Pseudocode* berada di dalam fungsi *cek_login_pemilik()*, fungsi ini terdapat di dalam *class Auth*.

Tabel 5.1 Pseudocode masuk ke dalam sistem

No.	Pseudocode
1	start
2	
3	mengecek pemilik sudah login atau belum
4	if cek == pemilik
5	tampil dashboard_pemilik
6	else
7	Form validasi email, email
8	Form validasi password, password
9	If data tersimpan
10	email = masukkan email benar
11	password = mengenkripsi password ketika benar
12	c1 = array register_email_pemilik=>email
13	c2 = array register_password_pemilk'=>password
14	cek_user = this->autentikasi->cek_login(c1, c2)
15	if(cek_user != salah)
16	foreach cek_user as data_user
17	data = array(
18	register_id_pemilik => data_user->register_id_pemilik,
19	register_email_pemilik => data_user->register_email_pemilik,
20	register_nama_pemilik => data_user->register_nama_pemilik,
21	register_password_pemilk => data_user->register_password_pemilk,
22	register_hakakses_pemilik => data_user->register_hakakses_pemilik,
23);
24	Menyimpan data
25	End Foreach
26	If session= pemilik
27	tampil "alert berhasil masuk"
28	tampil "dashboard";
29	else
30	Tampil "Username atau Password Salah!!!"
31	End if
32	End if
33	
34	
35	
36	
37	End

5.1.4.2 Perancangan Algoritme Menambahkan Rumah Indekos

Pada Tabel 5.2 perancangan algoritme yang digunakan untuk menambahkan rumah indekos. *Pseudocode* berada di dalam fungsi tambah_kos(), fungsi ini terdapat di dalam class Dashboard.

Tabel 5.2 Pseudocode menambahkan rumah indekos

No.	Pseudocode
1	start
2	
3	session = input post(register_nama_pemilik)
4	nama_kos = input post (nama_kos)
5	lokasi_kos = input post(lokasi_kos)
6	jumlah_kamar = input post (jumlah_kamar)
7	tipe_kos = input post (tipe_kos)
8	luas_kos = input post (luas_kos)
9	fasilitas_kos = input post (fasilitas_kos)
10	harga_kos = input post (harga_kos)
11	foto_kos = input post (nama_kos)
12	slug = slug(nama_kos)
13	date = date(format tanggal)
14	
15	config[upload_path] = upload tempat foto
16	config[allowed_types] = jenis foto yang disimpan
17	config[max_size] = ukuran foto yang diupload
18	config[max_width] = ukuran lebar foto
19	config[max_height] = ukuran tinggi foto
20	config[file_name] = nama foto yang disimpan
21	upload inisialisasi config
22	if file != 0
23	if (upload foto)
24	foto berhasil diupload
25	data = array(
26	rumahkos_session_pemilik => session
27	rumahkos_namaindekos_pemilik => nama_kos
28	rumahkos_lokasi_pemilik => lokasi_kos
29	rumahkos_jmlkamar_pemilik => jumlah_kamar
30	rumahkos_tipe_pemilik => tipe_kos
31	rumahkos_foto_pemilik => foto
32	rumahkos_luas_pemilik=> luas_kos
33	rumahkos_fasilitas_pemilik => fasilitas_kos
34	rumahkos_harga_pemilik => harga_kos
35	slug_indekos=>slug
36	created_at =>date
37	simpan pemasukan (data);

```
38     tampil data
39     else
40         gagal diupload
41     end if
42 end
```

5.1.4.3 Perancangan *Pseudocode* Menambahkan Penghuni

Pada Tabel 5.3 perancangan *pseudocode* yang digunakan untuk menambahkan penghuni. *Pseudocode* berada di dalam fungsi tambah_penghuni(), fungsi ini terdapat di dalam *class* Penghuni.

Tabel 5.3 Pseudocode menambahkan penghuni

No.	Pseudocode
1	Start
2	created = date(format tanggal)
3	session = session(register_nama_pemilik)
4	nama_penghuni = input post(nama_penghuni)
5	identitas = input post(identitas)
6	alamat = input post(alamat)
7	tempat_lahir = input post(tempat_lahir)
8	tgl_lahir = input post(tgl_lahir);
9	telp_penghuni = input post(telp_penghuni)
10	email_penghuni = input post(email_penghuni)
11	pekerjaan_penghuni = input post(pekerjaan_penghuni)
12	kelamin_penghuni = input post(kelamin_penghuni)
13	tgl_masuk = input post(tgl_masuk)
14	indekos = input post(indekos)
15	no_kamar = input post(no_kamar)
16	
17	data = array (
18	penghuni_session_pemilik => session
19	penghuni_namaindekos_pemilik => indekos
20	penghuni_nokamar_pemilik => no_kamar
21	penghuni_nama_pemilik => nama_penghuni
22	penghuni_identitas_pemilik => identitas
23	penghuni_alamat_pemilik=>alamat
24	penghuni_tmpatlahir_pemilik=>tempat_lahir
25	penghuni_tgllahir_pemilik=>format tanggal
26	penghuni_notlp_pemilik=>telp_penghuni
27	penghuni_email_pemilik=>email_penghuni
28	penghuni_instansi_pemilik=>pekerjaan_penghuni
29	penghuni_jeniskelamin_pemilik=>kelamin_penghuni
30	penghuni_tglmasuk_pemilik=> format tanggal
31	data dikirim ke data base

32	data berhasil disimpan dan menampilkan pesan berhasil	
33	end	

5.1.5 Perancangan Antarmuka

Pada tahadapan perancangan antarmuka ini merupakan digunakan untuk menjelaskan tampilan-tampilan yang akan diterapkan pada Sistem Manajemen Rumah Indekos. Perancangan antarmuka ini akan menjelaskan dengan menggunakan *layout* yang sudah dibuat sebagai berikut.

5.1.5.1 Halaman Utama Sistem

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan saat sistem pertama kali dijalankan. Memiliki keterangan terdapat dua tombol untuk masuk ke dalam sistem. Rancangan antarmuka halaman utama sistem terdapat pada Gambar 5.9 sebagai berikut.



Gambar 5.6 Perancangan antarmuka halaman utama sistem

Keterangan :

1. Merupakan tampilan ucapan selamat datang ketika
2. Tombol untuk masuk ke sistem dengan hak akses pemilik
3. Tombol untuk masuk ke sistem dengan hak akses penghuni

5.1.5.2 Halaman Masuk ke Dalam Sistem

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman masuk ke dalam sistem pemilik. Terdapat dua kolom yang harus di isi, diantaranya memasukkan *email* dan *password*. Rancangan antarmuka halaman utama sistem terdapat pada Gambar 5.10 sebagai berikut.

TEXT 1	TEXT 2
<p>The diagram illustrates the layout of a login page. At the top, there is a header section labeled 'TEXT 1' on the left and 'TEXT 2' on the right. Below this is a large central area labeled 'Form masuk ke dalam sistem'. Inside this area, there are three horizontal input fields labeled 3, 4, and 5 from top to bottom. Below these fields is a small rectangular button labeled 6.</p>	

Gambar 5.7 Perancangan antarmuka halaman masuk ke sistem

Keterangan :

1. Nama sistem pengelolaan manajemen rumah indekos
2. Nama pemilik yang masuk ke dalam sistem
3. *Form* masuk ke dalam sistem
4. *Field* masukkan email
5. *Field* masukkan *password*
6. Tombol untuk masuk ke sistem

5.1.5.3 Halaman Dashboard Pemilik

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman *dashboard* pemilik. Terdapat tiga informasi data yang diantaranya adalah jumlah indekos yang dimiliki, jumlah penghuni, total pendapatan. Rancangan antarmuka halaman *dashboard pemilik* terdapat pada Gambar 5.11 sebagai berikut.

TEXT 1:	TEXT 2:								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">MENU 1:</td></tr> <tr><td style="padding: 2px;">MENU 2:</td></tr> <tr><td style="padding: 2px;">MENU 3:</td></tr> <tr><td style="padding: 2px;">MENU 4:</td></tr> <tr><td style="padding: 2px;">MENU 5:</td></tr> <tr><td style="padding: 2px;">MENU 6:</td></tr> <tr><td style="padding: 2px;">MENU 7:</td></tr> <tr><td style="padding: 2px;">MENU 8:</td></tr> </table>	MENU 1:	MENU 2:	MENU 3:	MENU 4:	MENU 5:	MENU 6:	MENU 7:	MENU 8:	<p>The diagram illustrates the layout of a dashboard page. On the left side, there is a sidebar labeled 'TEXT 1' containing a vertical list of menu items from 'MENU 1' to 'MENU 8'. To the right of the sidebar are three large, empty rectangular boxes labeled 3, 4, and 5 from left to right, representing areas for displaying data like total units, total tenants, and total income.</p>
MENU 1:									
MENU 2:									
MENU 3:									
MENU 4:									
MENU 5:									
MENU 6:									
MENU 7:									
MENU 8:									

Gambar 5.8 Perancangan antarmuka halaman *dashboard* pemilik

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1 – 8 merupakan menu yang berada di dalam sistem

Angka 3 : merupakan keterangan banyaknya rumah indekos yang dimiliki

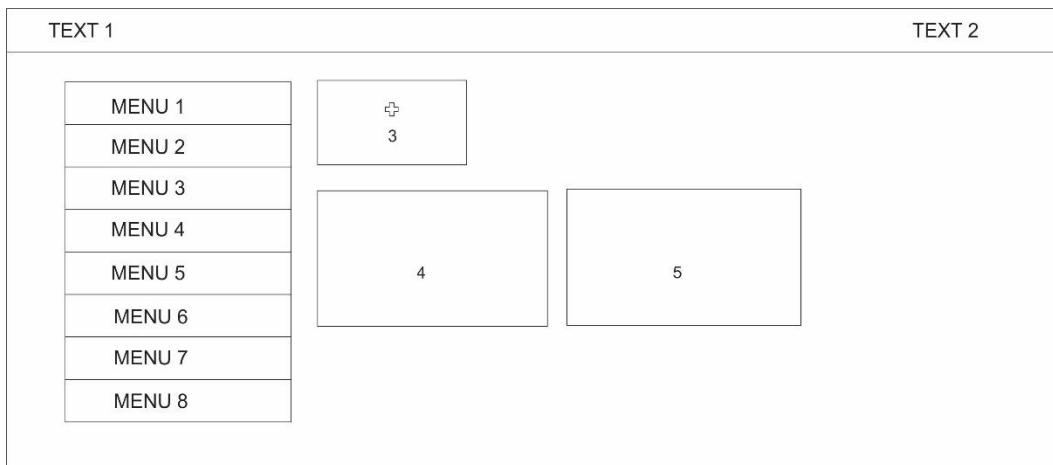
Keterangan lanjutan :

Angka 4 : merupakan keterangan banyaknya penghuni yang tinggal

Angka 5 : merupakan keterangan tota pendapatan

5.1.5.4 Halaman Rumah Kos

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman menu rumah kos pemilik. Terdapat informasi mengenai rumah indekos serta dapat menambahkan rumah indekos yang baru. Rancangan antarmuka halaman rumah indekos terdapat pada Gambar 5.12 sebagai berikut.



Gambar 5.9 Perancangan antarmuka halaman rumah indekos

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

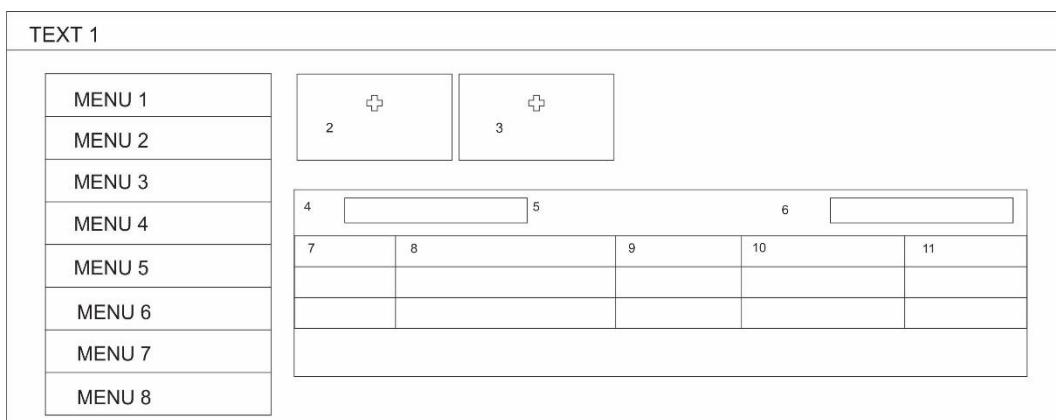
Menu 1 – 8 merupakan menu yang berada di dalam sistem

Angka 3 : merupakan tombol untuk menambahkan rumah indekos

Angka 4 – 5 : merupakan rumah indekos yang dimiliki

5.1.5.5 Halaman Penghuni

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman menu penghuni pemilik. Terdapat beberapa informasi mengenai data penghuni serta dapat menambahkan penghuni yang baru dan tambah akun yang baru. Rancangan antarmuka halaman penghuni terdapat pada Gambar 5.13 sebagai berikut.



Gambar 5.10 Perancangan antarmuka halaman penghuni

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1 – 8 merupakan menu yang berada di dalam sistem

Angka 2 : merupakan tombol untuk menambahkan penghuni

Angka 3 : merupakan tombol untuk menambahkan akun penghuni

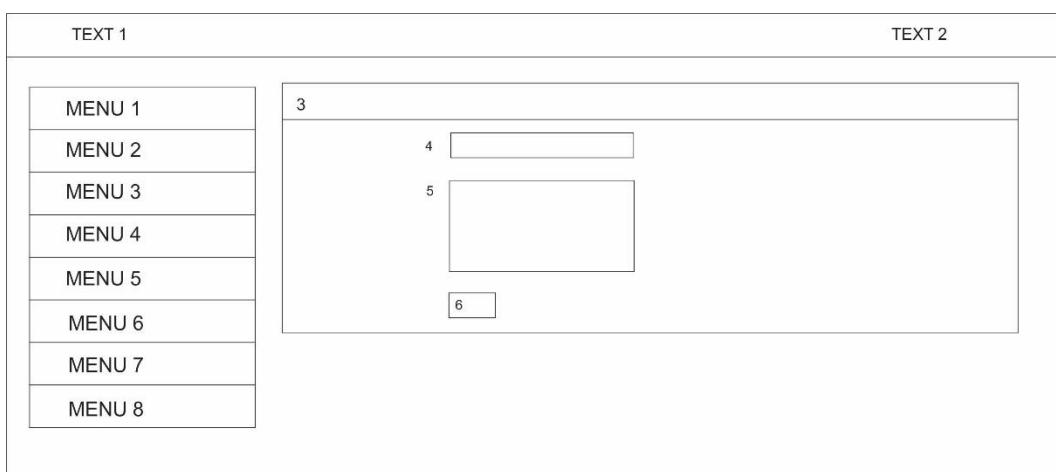
Angka 4 - 5: untuk menampilkan jumlah data yang diinginkan

Angka 6 : merupakan kolom untuk pencarian

Angka 7 – 11 : merupakan tabel informasi data penghuni

5.1.5.6 Halaman Reminder WhatsApp

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman menu *reminder*. Terdapat beberapa *field* yang harus diisi untuk dapat mengirimkan *reminder*. Rancangan antarmuka halaman *reminder whatsapp* terdapat pada Gambar 5.14 sebagai berikut.



Gambar 5.11 Perancangan antarmuka halaman *reminder whatsapp*

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Keterangan lanjutan :

Menu 1 – 8 merupakan menu yang berada di dalam sistem

Angka 3 : merupakan *form* untuk mengirimkan *reminder*

Angka 4 : merupakan kolom untuk memilih penghuni

Angka 5 : merupakan kolom untuk keterangan

Angka 6 : merupakan tombol untuk menyimpan

5.1.5.7 Halaman Data Pembayaran

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan saat pemilik melihat menu pembayaran indekos. Pada menu pembayaran indekos terdapat beberapa informasi mengenai pembayaran indekos diantaranya nama indekos, nama penghuni, nominal periode, status, tanggal tagihan dan aksi. Rancangan antarmuka halaman data pembayaran terdapat pada Gambar 5.15 sebagai berikut.

TEXT 1		TEXT 2									
<table border="1"><tr><td>MENU 1</td></tr><tr><td>MENU 2</td></tr><tr><td>MENU 3</td></tr><tr><td>MENU 4</td></tr><tr><td>MENU 5</td></tr><tr><td>MENU 6</td></tr><tr><td>MENU 7</td></tr><tr><td>MENU 8</td></tr></table>	MENU 1	MENU 2	MENU 3	MENU 4	MENU 5	MENU 6	MENU 7	MENU 8	<table border="1"><tr><td>3</td></tr></table>		3
	MENU 1										
	MENU 2										
	MENU 3										
	MENU 4										
	MENU 5										
	MENU 6										
	MENU 7										
	MENU 8										
3											
<table border="1"><tr><td>4</td><td>5</td></tr></table>		4	5								
4	5										
<table border="1"><tr><td>6</td><td>13</td></tr></table>		6	13								
6	13										
7	8	9	10								
11	12	13	14								

Gambar 5.12 Perancangan antarmuka data pembayaran

Keterangan lanjutan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1 – 8 merupakan menu yang berada di dalam sistem

Angka 3 : merupakan tombol untuk menambahkan data pembayaran

Angka 4 – 5 : untuk menampilkan jumlah data yang diinginkan

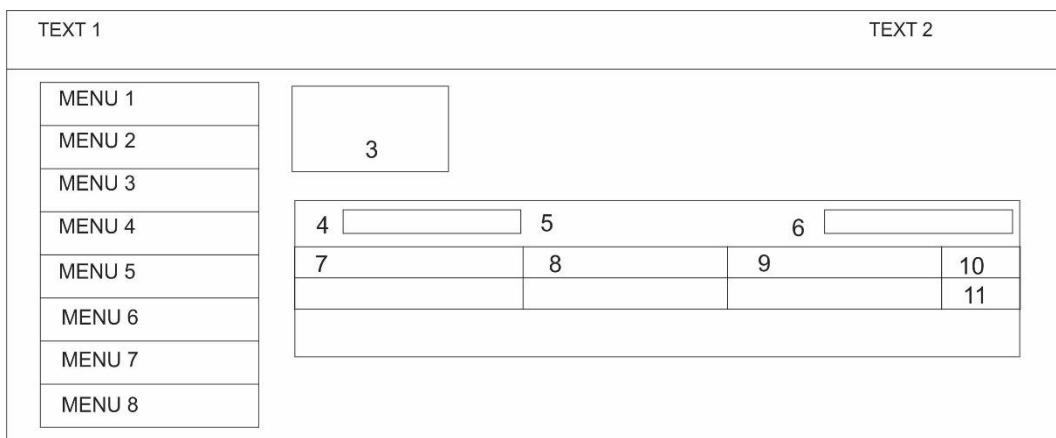
Angka 6 : merupakan kolom untuk pencarian

Angka 7 – 13 : merupakan tabel informasi data pembayaran

Angka 14 : merupakan tombol untuk mengubah data pembayaran

5.1.5.8 Halaman Pengeluaran

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan saat pemilik melihat menu laporan pengeluaran indekos. Pada menu laporan pengeluaran indekos terdapat beberapa informasi mengenai laporan pengeluaran indekos diantaranya detail pengeluaran, jumlah, tanggal dan aksi. Rancangan antarmuka halaman pengeluaran terdapat pada Gambar 5.16 sebagai berikut.



Gambar 5.13 Perancangan antarmuka pengeluaran

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1 – 8 merupakan menu yang berada di dalam sistem

Angka 3 : merupakan tombol untuk menambahkan data pengeluaran

Angka 4 – 5 : untuk menampilkan jumlah data yang diinginkan

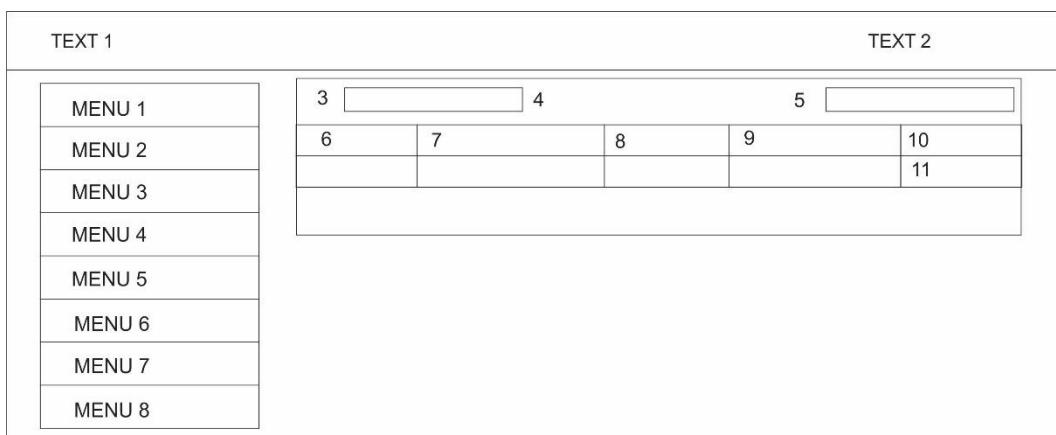
Angka 6 : merupakan kolom untuk pencarian

Angka 7 – 10 : merupakan tabel informasi data pengeluaran

Angka 11 : merupakan tombol untuk mengubah data pengeluaran

5.1.5.9 Halaman *List* Pemesanan

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan saat pemilik melihat menu *list* pemesanan indekos. Pada menu *list* pemesanan terdapat beberapa informasi mengenai *list* pemesanan indekos diantaranya detail nama, identitas, indekos, status dan aksi. Rancangan antarmuka halaman *list* pemesanan terdapat pada Gambar 5.17 sebagai berikut.



Gambar 5.14 Perancangan antarmuka *list* pemesanan

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1 – 8 merupakan menu yang berada di dalam sistem

Keterangan lanjutan :

Angkat 3-4 : untuk menampilkan jumlah data yang diinginkan

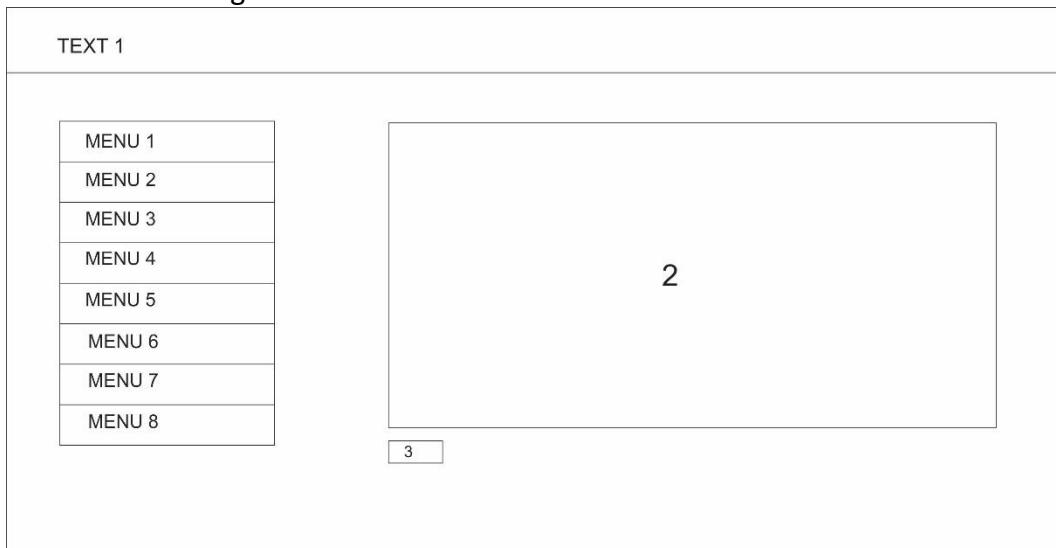
Angka 5 : merupakan kolom untuk pencarian

Angka 6-10 : merupakan tabel informasi data pemesan

Angka 11 : merupakan tombol untuk mengubah data pemesanan

5.1.5.10 Halaman *List Kendala*

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan saat pemilik melihat menu *list* kendala indekos. Pada menu *list* kendala indekos terdapat beberapa informasi mengenai *list* kendala indekos diantaranya foto dan tombol untuk *submit*. Rancangan antarmuka halaman *list* kendala terdapat pada Gambar 5.18 sebagai berikut.



Gambar 5.15 Perancangan antarmuka *list kendala*

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Menu 1 – 8 merupakan menu yang berada di dalam sistem

Angka 2 : merupakan *list* kendala yang dimasukkan dari penghuni

Angka 3 : merupakan tombol untuk *fix* ketika sudah dibenahi

5.1.5.11 Halaman Tambah Indekos

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk menambahkan rumah indekos. Terdapat beberapa kolom yang harus di isi, diantaranya memasukkan nama indekos, hargas indekos, jumlah kamar, tipe indekos, luas indekos, lokasi indekos, fasilitas indekos dan foto indekos. Rancangan antarmuka halaman tambah indekos terdapat pada Gambar 5.19 sebagai berikut.

TEXT 1	TEXT 2																		
<table border="1" style="width: 100px; margin-left: auto; margin-right: auto;"> <tr><td>MENU 1</td></tr> <tr><td>MENU 2</td></tr> <tr><td>MENU 3</td></tr> <tr><td>MENU 4</td></tr> <tr><td>MENU 5</td></tr> <tr><td>MENU 6</td></tr> <tr><td>MENU 7</td></tr> <tr><td>MENU 8</td></tr> </table>	MENU 1	MENU 2	MENU 3	MENU 4	MENU 5	MENU 6	MENU 7	MENU 8	<table border="1" style="width: 100px; margin-left: auto; margin-right: auto;"> <tr><td>3</td></tr> <tr><td>4 <input type="text"/></td></tr> <tr><td>5 <input type="text"/></td></tr> <tr><td>6 <input type="text"/></td></tr> <tr><td>7 <input type="text"/></td></tr> <tr><td>8 <input type="text"/></td></tr> <tr><td>9 <input type="text"/></td></tr> <tr><td>10 <input type="text"/></td></tr> <tr><td><input type="text"/> 11</td></tr> <tr><td><input type="text"/> 12</td></tr> </table>	3	4 <input type="text"/>	5 <input type="text"/>	6 <input type="text"/>	7 <input type="text"/>	8 <input type="text"/>	9 <input type="text"/>	10 <input type="text"/>	<input type="text"/> 11	<input type="text"/> 12
MENU 1																			
MENU 2																			
MENU 3																			
MENU 4																			
MENU 5																			
MENU 6																			
MENU 7																			
MENU 8																			
3																			
4 <input type="text"/>																			
5 <input type="text"/>																			
6 <input type="text"/>																			
7 <input type="text"/>																			
8 <input type="text"/>																			
9 <input type="text"/>																			
10 <input type="text"/>																			
<input type="text"/> 11																			
<input type="text"/> 12																			

Gambar 5.16 Perancangan antarmuka tambah indekos

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-8 : merupakan menu yang berada di dalam sistem

Angka 3 : *form* menambahkan rumah indekos

Angka 4 : kolom untuk memasukkan nama indekos

Angka 5 : kolom untuk memasukkan harga

Angka 6 : kolom untuk memasukkan jumlah kamar

Angka 7 : kolom untuk memasukkan tipe indekos

Angka 8 : kolom untuk memasukkan luas indekos

Angka 9 : kolom untuk memasukkan lokasi indekos

Angka 10 : kolom untuk memasukkan fasilitas indekos

Angka 11 : kolom untuk memasukkan foto

Angka 12 : tombol untuk menyimpan

5.1.5.12 Halaman Tambah Penghuni

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk menambahkan penghuni indekos. Terdapat beberapa kolom yang harus di isi, diantaranya memasukkan nama indekos, identitas, alamat, tempat lahir, tanggal lahir, nomor telepon, email, pekerjaan, jenis kelamin, tanggal masuk, indekos dan nomor kamar. Rancangan antarmuka halaman tambah penghuni terdapat pada Gambar 5.20 sebagai berikut.

TEXT 1	TEXT 2
MENU 1 MENU 2 MENU 3 MENU 4 MENU 5 MENU 6 MENU 7 MENU 8	3 4 <input type="text"/> 5 <input type="text"/> 6 <input type="text"/> 7 <input type="text"/> 8 <input type="text"/> 9 <input type="text"/> 10 <input type="text"/> 11 <input type="text"/> 12 <input type="text"/> 13 <input type="text"/> 14 <input type="text"/> 15 <input type="text"/> <input type="button" value="16"/>

Gambar 5.17 Perancangan antarmuka tambah penghuni

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-8 : merupakan menu yang berada di dalam sistem

Angka 3 : form menambahkan penghuni

Angka 4 : kolom untuk memasukkan nama penghuni

Angka 5 : kolom untuk memasukkan penghuni

Angka 6 : kolom untuk memasukkan alamat penghuni

Angka 7 : kolom untuk memasukkan tempat lahir penghuni

Angka 8 : kolom untuk memasukkan tanggal lahir penghuni

Angka 9 : kolom untuk memasukkan nomor telpon penghuni

Angka 10 : kolom untuk memasukkan *email* penghuni

Angka 11 : kolom untuk memasukkan pekerjaan penghuni

Angka 12 : kolom untuk memasukkan jenis kelamin penghuni

Angka 13 : kolom untuk memasukkan tanggal masuk

Angka 14 : kolom untuk memasukkan rumah indekos

Angka 15 : kolom untuk memasukkan nomor kamar

Angka 16 : tombol untuk menyimpan data yang sudah diisi

5.1.5.13 Halaman Tambah Akun Penghuni

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk menambahkan akun penghuni indekos. Terdapat beberapa

kolom yang harus diisi, diantaranya memasukkan nama, *username* dan *password*. Rancangan antarmuka halaman tambah akun penghuni terdapat pada Gambar 5.21 sebagai berikut.

TEXT 1	
MENU 1	3
MENU 2	4 <input type="text"/>
MENU 3	5 <input type="text"/>
MENU 4	6 <input type="text"/>
MENU 5	
MENU 6	
MENU 7	
MENU 8	7 <input type="text"/>

Gambar 5.18 Perancangan antarmuka tambah akun penghuni

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-8 : merupakan menu yang berada di dalam sistem

Angka 3 : *form* menambahkan rumah indekos

Angka 4 : kolom untuk memasukkan nama penghuni

Angka 5 : kolom untuk memasukkan *username*

Angka 6 : kolom untuk memasukkan *password*

Angka 7 : kolom untuk menyimpan data

5.1.5.14 Halaman Mengubah Penghuni

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk mengubah penghuni indekos. Terdapat beberapa kolom yang harus diisi, diantaranya memasukkan nama indekos, identitas, alamat, tempat lahir, tanggal lahir, nomor telepon, email, pekerjaan, jenis kelamin, tanggal masuk, indekos dan nomor kamar. Rancangan antarmuka halaman tambah penghuni terdapat pada Gambar 5.22 sebagai berikut.

TEXT 1	TEXT 2																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>MENU 1</td></tr> <tr><td>MENU 2</td></tr> <tr><td>MENU 3</td></tr> <tr><td>MENU 4</td></tr> <tr><td>MENU 5</td></tr> <tr><td>MENU 6</td></tr> <tr><td>MENU 7</td></tr> <tr><td>MENU 8</td></tr> </table>	MENU 1	MENU 2	MENU 3	MENU 4	MENU 5	MENU 6	MENU 7	MENU 8	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">3</td></tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;">4</td></tr> <tr><td style="height: 20px;">5</td></tr> <tr><td style="height: 20px;">6</td></tr> <tr><td style="height: 20px;">7</td></tr> <tr><td style="height: 20px;">8</td></tr> <tr><td style="height: 20px;">9</td></tr> <tr><td style="height: 20px;">10</td></tr> <tr><td style="height: 20px;">11</td></tr> <tr><td style="height: 20px;">12</td></tr> <tr><td style="height: 20px;">13</td></tr> <tr><td style="height: 20px;">14</td></tr> <tr><td style="height: 20px;">15</td></tr> <tr><td style="text-align: center; padding: 5px;">16</td></tr> </table>	3		4	5	6	7	8	9	10	11	12	13	14	15	16
MENU 1																								
MENU 2																								
MENU 3																								
MENU 4																								
MENU 5																								
MENU 6																								
MENU 7																								
MENU 8																								
3																								
4																								
5																								
6																								
7																								
8																								
9																								
10																								
11																								
12																								
13																								
14																								
15																								
16																								

Gambar 5.19 Perancangan antarmuka mengbah penghuni

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-8 : merupakan menu yang berada di dalam sistem

Angka 3 : form menambahkan penghuni

Angka 4 : kolom untuk memasukkan nama penghuni

Angka 5 : kolom untuk memasukkan penghuni

Angka 6 : kolom untuk memasukkan alamat penghuni

Angka 7 : kolom untuk memasukkan tempat lahir penghuni

Angka 8 : kolom untuk memasukkan tanggal lahir penghuni

Angka 9 : kolom untuk memasukkan nomor telpon penghuni

Angka 10 : kolom untuk memasukkan *email* penghuni

Angka 11 : kolom untuk memasukkan pekerjaan penghuni

Angka 12 : kolom untuk memasukkan jenis kelamin penghuni

Angka 13 : kolom untuk memasukkan tanggal masuk

Angka 14 : kolom untuk memasukkan rumah indekos

Angka 15 : kolom untuk memasukkan nomor kamar

Angka 16 : tombol untuk menyimpan data yang sudah diisi

5.1.5.15 Halaman mendaftar ke sistem

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk mendaftar ke dalam sistem. Terdapat beberapa kolom yang harus di isi, diantaranya memasukkan nama, *email*, *password*, alamat, nomor telepon,

tempat lahir, tanggal lahir dan jenis kelamin. Rancangan antarmuka halaman tambah penghuni terdapat pada Gambar 5.23 sebagai berikut.

TEXT 1	TEXT 2										
<table border="1"><tr><td>3</td></tr><tr><td>4</td></tr><tr><td>5</td></tr><tr><td>6</td></tr><tr><td>7</td></tr><tr><td>8</td></tr><tr><td>9</td></tr><tr><td>10</td></tr><tr><td>11</td></tr><tr><td>12</td></tr></table>		3	4	5	6	7	8	9	10	11	12
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											

Gambar 5.20 Perancangan antarmuka mendaftar ke sistem

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Angka 3 : *form* untuk mendaftar ke sistem sebagai pemilik

Angka 4 : kolom untuk memasukkan nama

Angka 5 : kolom untuk memasukkan *email*

Angka 6 : kolom untuk memasukkan *password*

Angka 7 : kolom untuk memasukkan alamat

Angka 8 : kolom untuk memasukkan nomor telefon

Angka 9 : kolom untuk memasukkan tempat tanggal lahir

Angka 10 : kolom untuk memasukkan tanggal lahir

Angka 11 : kolom untuk memasukkan jenis kelamin

5.1.5.16 Halaman Pemesanan Kamar

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk memesan kamar. Terdapat beberapa kolom yang harus di isi, diantaranya memasukkan nama, idenitas, nama kos, pilih kamar, nomor telefon, tanggal pesan dan batas tanggal pesan. Rancangan antarmuka halaman pemesanan kamar terdapat pada Gambar 5.24 sebagai berikut.

1	
2	<input type="text"/>
3	<input type="text"/>
4	<input type="text"/>
5	<input type="text"/>
6	<input type="text"/>
7	<input type="text"/>
8	<input type="text"/>
9	<input type="button" value=""/>

Gambar 5.21 Perancangan antarmuka pemesanan kamar

Keterangan :

- Angka 1 : *form* untuk melakukan pemesanan kamar
- Angka 2 : kolom untuk memasukkan nama
- Angka 3 : kolom untuk memasukkan identitas
- Angka 4 : kolom untuk memasukkan nama indekos
- Angka 5 : kolom untuk memasukkan nomor kamar
- Angka 6 : kolom untuk memasukkan nomor telepon
- Angka 7 : kolom untuk memasukkan tanggal pesan
- Angka 8 : kolom untuk memasukkan batas tanggal pesan
- Angka 9 : tombol untuk menyimpan data

5.1.5.17 Halaman Tagihan

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk menambahkan tagihan kepada penghuni indekos. Terdapat beberapa kolom yang harus di isi, diantaranya memasukkan nama indekos, nama penguni, metode pembayaran, nominal, periode pembayaran dan tanggal tagihan. Rancangan antarmuka halaman tagihan terdapat pada Gambar 5.25 sebagai berikut.

TEXT 1	TEXT 2																
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>MENU 1</td></tr> <tr><td>MENU 2</td></tr> <tr><td>MENU 3</td></tr> <tr><td>MENU 4</td></tr> <tr><td>MENU 5</td></tr> <tr><td>MENU 6</td></tr> <tr><td>MENU 7</td></tr> <tr><td>MENU 8</td></tr> </table>	MENU 1	MENU 2	MENU 3	MENU 4	MENU 5	MENU 6	MENU 7	MENU 8	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">3</td></tr> <tr><td style="text-align: center;">4 <input type="text"/></td></tr> <tr><td style="text-align: center;">5 <input type="text"/></td></tr> <tr><td style="text-align: center;">6 <input type="text"/></td></tr> <tr><td style="text-align: center;">7 <input type="text"/></td></tr> <tr><td style="text-align: center;">8 <input type="text"/></td></tr> <tr><td style="text-align: center;">9 <input type="text"/></td></tr> <tr><td style="text-align: center;"><input type="text"/> 10</td></tr> </table>	3	4 <input type="text"/>	5 <input type="text"/>	6 <input type="text"/>	7 <input type="text"/>	8 <input type="text"/>	9 <input type="text"/>	<input type="text"/> 10
MENU 1																	
MENU 2																	
MENU 3																	
MENU 4																	
MENU 5																	
MENU 6																	
MENU 7																	
MENU 8																	
3																	
4 <input type="text"/>																	
5 <input type="text"/>																	
6 <input type="text"/>																	
7 <input type="text"/>																	
8 <input type="text"/>																	
9 <input type="text"/>																	
<input type="text"/> 10																	

Gambar 5.22 Perancangan antarmuka tagihan

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-8 : merupakan menu yang berada di dalam sistem

Angka 3 : *form* untuk menambahkan tagihan

Angka 4 : kolom untuk memasukkan nama indekos

Angka 5 : kolom untuk memasukkan nama penghuni

Angka 6 : kolom untuk memasukkan metode pembayaran

Angka 7 : kolom untuk memasukkan nominal

Angka 8 : kolom untuk memasukkan periode

Angka 9 : kolom untuk memasukkan tanggal tagihan

5.1.5.18 Halaman Tambah Pengeluaran

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk menambahkan pengeluaran indekos. Terdapat beberapa kolom yang harus di isi, diantaranya memasukkan nominal, detail dan tanggal pengeluaran. Rancangan antarmuka halaman tambah pengeluaran terdapat pada Gambar 5.26 sebagai berikut.

TEXT 1	TEXT 2													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>MENU 1</td></tr> <tr><td>MENU 2</td></tr> <tr><td>MENU 3</td></tr> <tr><td>MENU 4</td></tr> <tr><td>MENU 5</td></tr> <tr><td>MENU 6</td></tr> <tr><td>MENU 7</td></tr> <tr><td>MENU 8</td></tr> </table>	MENU 1	MENU 2	MENU 3	MENU 4	MENU 5	MENU 6	MENU 7	MENU 8	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">3</td></tr> <tr><td style="text-align: center;">4 <input type="text"/></td></tr> <tr><td style="text-align: center;">5 <input type="text"/></td></tr> <tr><td style="text-align: center;">6 <input type="text"/></td></tr> <tr><td style="text-align: center;"><input type="text"/> 7</td></tr> </table>	3	4 <input type="text"/>	5 <input type="text"/>	6 <input type="text"/>	<input type="text"/> 7
MENU 1														
MENU 2														
MENU 3														
MENU 4														
MENU 5														
MENU 6														
MENU 7														
MENU 8														
3														
4 <input type="text"/>														
5 <input type="text"/>														
6 <input type="text"/>														
<input type="text"/> 7														

Gambar 5.23 Perancangan antarmuka tambah pengeluaran

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-8 : merupakan menu yang berada di dalam sistem

Angka 3 : *form* untuk pengeluaran

Angka 4 : kolom untuk memasukkan nominal

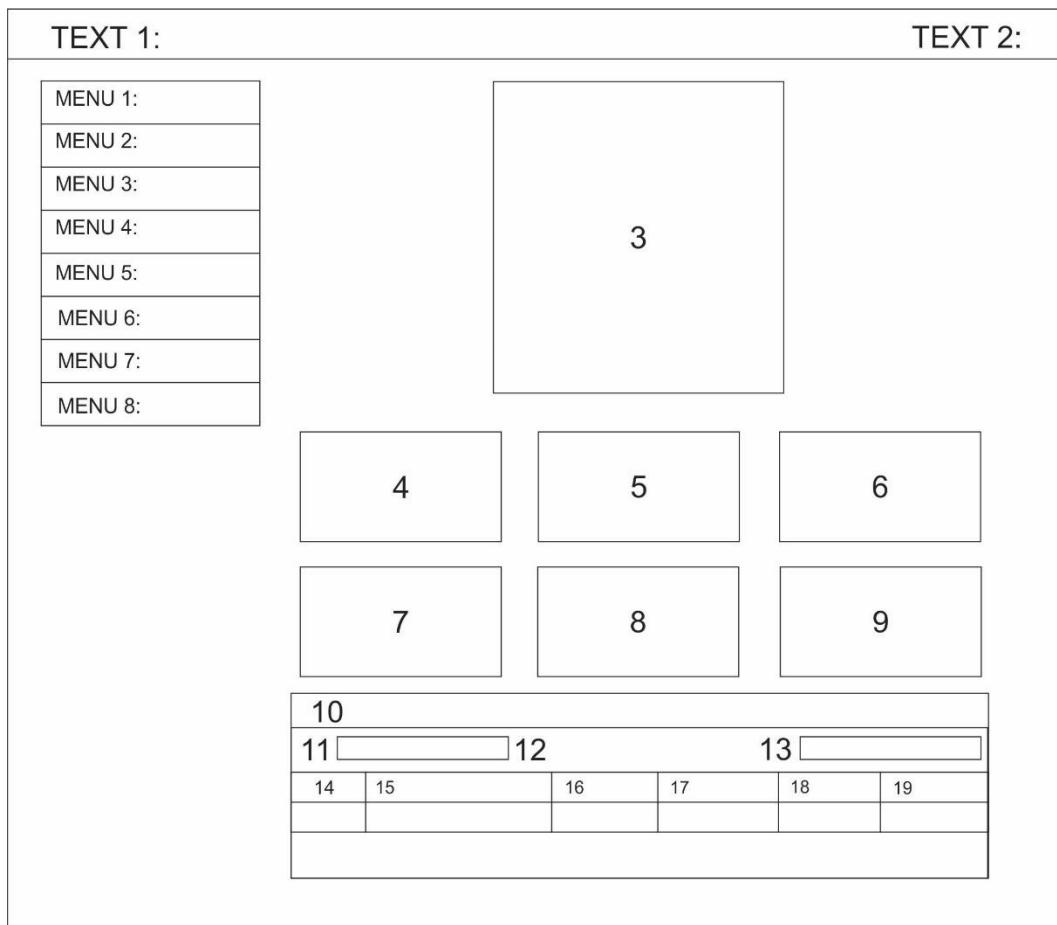
Angka 5 : kolom untuk memasukkan keterangan

Angka 6 : kolom untuk memasukkan tanggal pengeluaran

Angka 7 : tombol untuk menyimpan data yang telah diisi

5.1.5.19 Halaman *Detail* Indekos

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk menampilkan fasilitas-fasilitas beserta penghuni yang berada di dalam rumah indekos. Rancangan antarmuka halaman *detail* indekos terdapat pada Gambar 5.27 sebagai berikut.



Gambar 5.24 Perancangan antarmuka *detail* rumah indekos

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-8 : merupakan menu yang berada di dalam sistem

Keterangan lanjutan :

Angka 3 – 9 : merupakan fasilitas rumah indekos

Angka 10 : *list* penghuni rumah indekos

Angka 11-12 : menampilkan data yang diinginkan

Angka 13 : merupakan kolom untuk pencarian

Angka 14-15 : merupakan data informasi penghuni

5.1.5.20 Halaman Data Pembayaran

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk menampilkan data pembayaran penghuni yang berada di dalam sistem. Rancangan antarmuka halaman data pembayaran terdapat pada Gambar 5.28 sebagai berikut.

TEXT 1:	TEXT 2:																											
<table border="1"><tr><td>MENU 1:</td></tr><tr><td>MENU 2:</td></tr><tr><td>MENU 3:</td></tr></table>	MENU 1:	MENU 2:	MENU 3:	<table border="1"><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>13</td></tr><tr><td colspan="7"></td></tr></table>	3	4	5	6	7	8	9	10	11	12							13							
MENU 1:																												
MENU 2:																												
MENU 3:																												
3	4	5																										
6	7	8	9	10	11	12																						
						13																						

Gambar 5.25 Perancangan antarmuka data pembayaran

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-3 : merupakan menu yang berada di dalam sistem

Angka 3-4 : menampilkan data yang diinginkan

Angka 5 : merupakan kolom untuk pencarian

Angka 6-12 : merupakan data informasi pembayaran

Angka 13 : mencetak bukti pembayaran

5.1.5.21 Halaman Lapor Kendala

Halaman ini yang nantinya merupakan halaman yang akan ditampilkan halaman untuk melaporkan kendala yang ada di dalam rumah indekos berada di dalam sistem. Rancangan antarmuka halaman lapor kendala terdapat pada Gambar 5.29 sebagai berikut.

TEXT 1	TEXT 2									
<table border="1"><tr><td>MENU 1</td></tr><tr><td>MENU 2</td></tr><tr><td>MENU 3</td></tr></table>	MENU 1	MENU 2	MENU 3	<table border="1"><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td></td><td></td></tr></table>	3	4	5	6		
MENU 1										
MENU 2										
MENU 3										
3	4	5								
6										

Gambar 5.26 Perancangan antarmuka lapor kendala

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

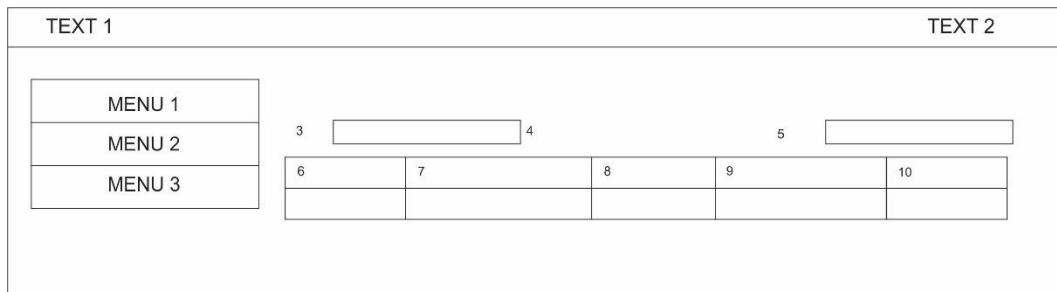
Menu 1-3 : merupakan menu yang berada di dalam sistem

Angka 4 : merupakan tombol untuk *upload* foto

Angka 6 : merupakan untuk menyimpan kendala

5.1.5.22 Halaman Melihat Status Pembayaran

Halaman ini yang nantinya merupakan halaman yang menampilkan status pembayaran yang telah dilakukan oleh penghuni indekos Rancangan antarmuka halaman melihat status pembayaran terdapat pada Gambar 5.30 sebagai berikut.



Gambar 5.27 Perancangan antarmuka status pembayaran

Keterangan :

Text 1 : nama sistem pengelolaan manajemen rumah indekos

Text 2 : pemilik yang masuk ke dalam sistem

Menu 1-3 : merupakan menu yang berada di dalam sistem

Angka 3-4 : merupakan untuk menampilkan data yang diinginkan

Angka 5 : merupakan kolom untuk pencarian

Angka 6-10: merupakan data informasi mengenai status pembayaran

5.2 Implementasi

Pada bagian sub bab implementasi ini akan membahas bagaimana Sistem Manajemen Rumah Indekos diimplementasikan berdasarkan dengan menggunakan metodologi dan perancangan yang sudah dibuat. Dalam implementasi ini akan menjelaskan tentang spesifikasi sistem yang digunakan, implementasi algoritma dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Pada saat mengimplementasikan sistem yang telah dirancang, dibutuhkan sebuah perangkat keras dan perangkat lunak. Untuk penspesifikasian sendiri memiliki tujuan untuk mengetahui spesifikasi standar dalam pembuatan sistem.

5.2.1.1 Spesifikasi Perangkat Keras

Berikut ini merupakan spesifikasi perangkat lunak yang digunakan dalam pembangunan sistem. Tabel 5.4 merupakan spesifikasi perangkat keras pada komputer yang digunakan.

Tabel 5.4 Spesifikasi perangkat keras komputer

Nama Komponen	Keterangan
<i>Harddisk</i>	1 TB
<i>Processor</i>	<i>Intel(R) Core(TM) i5 CPU @ 2.60GHz</i>
<i>RAM</i>	8 GB
<i>Graphic Card</i>	AMD Radeon HD 8600M Series

5.2.1.2 Spesifikasi Perangkat Lunak

Berikut ini merupakan spesifikasi perangkat lunak yang digunakan dalam pembangunan sistem. Tabel 5.5 merupakan spesifikasi perangkat lunak pada komputer yang digunakan.

Tabel 5.5 Spesifikasi perangkat lunak komputer

Hardware	Keterangan
Sistem Operasi	<i>Windows 8.1 Pro N</i>
<i>Tools</i>	<i>Bracket 1.11</i>
<i>DBMS</i>	<i>MySQL phpMyAdmin 4.2.11</i>
<i>Server</i>	<i>Localhost(XAMPP) 3.2.1</i>
<i>Framework</i>	<i>CodeIgniter 3.19</i>
<i>Browser</i>	<i>Google Chrome 70.0.3538.110 & Internet Explore 42.17134.1.0</i>

5.2.2 Implementasi Basis Data

Perancangan basis data merupakan tahapan-tahapan selanjutnya yang bertujuan untuk memodelkan relasi-relasi dari entitas yang berada di dalam sistem. Perancangan ini menggunakan *physical data model*. Dalam pemodelan ini dapat mewakili rancangan basis data serta dapat menjelaskan semua struktur-struktur tabel basis data yang ada di dalam sistem. Rancangan *physical data model* pada sistem manajemen rumah indekos akan ditunjukkan dengan Gambar 5.31.

Gambar 5.28 Implementasi Basis Data



5.2.3 Implementasi *Pseudocode*

Pada tahapan untuk mengimplementasikan *pseudocode*, perancangan *pseudocode* yang sudah dibuat akan diubah menjadi sebuah kode program yang mudah dimengerti oleh komputer. Berikut ini terdapat tiga sampel dari kode-kode program.

5.2.3.1 Implementasi *Pseudocode* Masuk ke Dalam Sistem

Pada tabel 5.6 merupakan implementasi dari kode program untuk melakukan masuk ke dalam sistem. *Pseudocode* tersebut terdapat berada di dalam *class* Auth. *Function* cek_login_pemilik() digunakan untuk melakukan validasi data yang dikirimkan pada *view login*. Sedangkan *function* cek_login digunakan menyimpan data ke dalam *database*.

Tabel 5.6 Implementasi *pseudocode* masuk ke dalam sistem

No.	Kode Program
1	public function cek_login_pemilik() { 3 \$cek = \$this->autentikasi->if_logged(); 4 if(\$cek == 'pemilik'){ 5 redirect('dashboard_pemilik'); 6 }else{ 7 \$this-> 8 form_validation->set_rules('email','email','required'); 9 \$this->form_validation- 10 >set_rules('password','password','required'); 11 //this->form_validation->set_message('required','<div 12 class="form-group">Harus Diisi</div>'); 13 if(\$this->form_validation->run() == TRUE){ 14 \$email = \$this->input->post('email',TRUE); 15 \$password = MD5(\$this->input->post('password', 16 TRUE)); 17 \$c1 = array('register_email_pemilik'=>\$email); 18 \$c2=array('register_password_pemilk'=>\$password); 19 \$checking_user = \$this->autentikasi->cek_login(\$c1,\$c2); 20 if(\$checking_user != FALSE){ 21 foreach (\$checking_user as \$data_user) { 22 \$data = array(23 'register_id_pemilik' => \$data_user->register_id_pemilik, 24 'register_email_pemilik' => \$data_user->register_email_pemilik, 25 'register_nama_pemilik' => \$data_user->register_nama_pemilik, 26 'register_password_pemilk' => \$data_user->register_password_pemilk, 27 'register_hakakses_pemilik' => \$data_user->register_hakakses_pemilik, 28); 29 \$this->session->set_userdata(\$data); 30 } 31 } 32 } 33 } 34 } 35 } 36 }

Tabel 5.6 Implementasi pseudocode masuk ke dalam sistem (lanjutan)

No.	Kode Program
37	}
38	if(\$this->session->userdata('register_hakakses_pemilik') == 'pemilik') {
39	echo "<script>alert('Selamat datang di Panel TRUSTEDKOS');</script>";
40	echo "<script>window.location.href = '" . base_url()
41	. "dashboard_pemilik';</script>";
42	} else {
43	\$data['error'] = '<div class="alert alert-success alert-dismissible">
44	<button type="button" href="#" class="close" data-dismiss="alert" aria-label="close">×</button>
45	Username atau Password Salah!!!.
46	</div> ';
47	\$this->load->view('pemilik/login_form/login', \$data);
48	}
49	}elseif(\$checking_user == FALSE) {
50	\$data['error'] = '<div class="alert alert-success alert-dismissible">
51	<button type="button" href="#" class="close" data-dismiss="alert" aria-label="close">×</button>
52	Username atau Password Salah!!!.</div> ';
53	\$this->load->view('pemilik/login_form/login', \$data);
54	}
55	} else {
56	\$this->load->view('pemilik/login_form/login');
57	}
58	}
59	}
60	
61	
62	
63	
64	
65	}

Penjelasan :

Pada baris 3 – 6 merupakan kondisi untuk mengecek pemilik sudah masuk ke sistem

Pada baris 7 – 12 merupakan form validasi

Pada baris 13 – 16 merupakan kondisi untuk form validasi yang sudah diisi pemilik

Pada baris 18 - 20 merupakan pendeklarasian variable array

Pada baris 21 merupakan mengecek data array

Pada baris 22 – 40 merupakan kondisi jika cek user yang dimasukkan pemilik tidak kosong maka akan menampilkan pesan sukses

Pada baris 40 – 58 merupakan kondisi jika cek user yang dimasukkan pemilik kosong atau salah maka akan menampilkan pesan gagal

5.2.3.2 Implementasi *Pseudocode* Menambahkan Rumah Indekos

Pada tabel 5.7 merupakan implementasi dari kode program untuk melakukan menambahkan indekos. *Pseudocode* tersebut terdapat berada di dalam *class Dashboard*. *Function* tambah_kos () digunakan untuk memarsing data dari *view form_tambah_rumah_kos* ke *function addRumahkos*. Sedangkan *function addPenghuni* digunakan menyimpan data ke dalam *database*.

Tabel 5.7 Implementasi *pseudocode* menambahkan rumah indekos

No.	Kode Program
1	<pre> 1 public function tambah_kos(){ 2 \$session = \$this->session- 3 >userdata('register_nama_pemilik'); 4 \$nama_kos = \$this->input->post('nama_kos'); 5 \$lokasi_kos = \$this->input->post('lokasi_kos'); 6 \$jumlah_kamar = \$this->input->post('jumlah_kamar'); 7 \$tipe_kos = \$this->input->post('tipe_kos'); 8 \$luas_kos = \$this->input->post('luas_kos'); 9 \$fasilitas_kos = \$this->input->post('fasilitas_kos'); 10 \$harga_kos = \$this->input->post('harga_kos'); 11 \$foto_kos = \$this->input->post('nama_kos'); 12 \$slug = slug(\$nama_kos); 13 \$date = date('Y-m-d'); 14 15 \$config['upload_path'] = './asset/foto_kos'; 16 \$config['allowed_types'] = 'jpg png jpeg gif'; 17 \$config['max_size'] = '4024'; //4MB max 18 \$config['max_width'] = '4480'; // pixel 19 \$config['max_height'] = '4480'; // pixel 20 \$config['file_name'] = \$_FILES['foto']['name']; 21 \$this->upload->initialize(\$config); 22 if(!empty(\$_FILES['foto']['name'])){ 23 if(\$this->upload->do_upload('foto')){ 24 \$foto = \$this->upload->data(); 25 \$data = array(26 'rumahkos_session_pemilik' => \$session, 27 'rumahkos_namaindekos_pemilik' => \$nama_kos, 28 'rumahkos_lokasi_pemilik' => \$lokasi_kos, 29 'rumahkos_jmlkamar_pemilik' => \$jumlah_kamar, 30 'rumahkos_tipe_pemilik' => \$tipe_kos, 31 'rumahkos_foto_pemilik' => \$foto['file_name'], 32 'rumahkos_luas_pemilik' => \$luas_kos, 33 'rumahkos_fasilitas_pemilik' => \$fasilitas_kos, 34 'rumahkos_harga_pemilik' => \$harga_kos, 35 'slug_indekos'=>\$slug, 36 'created_at' =>\$date, 37); 38 \$this->backend->addRumahkos (\$data); 39 } 40 } 41}</pre>

Tabel 5.7 Implementasi pseudocode menambahkan rumah indekos (lanjutan)

No.	Kode Program
40 41 42 43 44 45 46 47 48 49 50	<pre> echo "<script>alert('Data berhasil diupload'); window.location.href='form_tambah_kos'; </script>"; } else{ die('Fail Upload Data 1'); } } else{ die('Fail Upload Data 2'); } } </pre>

Penjelasan :

Pada baris 4 – 13 merupakan pendeklarasian variable data

Pada baris 15 merupakan *path* untuk meyimpan foto

Pada baris 16 merupakan jenis format foto

Pada baris 17 merupakan maksimal ukuran foto

Pada baris 18 – 19 merupakan maksimal ukuran pixel

Pada baris 22 – 39 merupakan kondisi untuk menyimpan data yang dimasukkan pemilik ke dalam database

Pada baris 40 merupakan pesan berhasil ketika data masuk ke dalam database

Pada baris 43, 48 merupakan pesan gagal ketika data masuk ke dalam database

5.2.3.3 Implementasi Pseudocode menambahkan penghuni

Pada tabel 5.8 merupakan implementasi dari kode program untuk melakukan menambahkan penghuni. *Pseudocode* tersebut terdapat berada di dalam *class Penghuni*. *Function* tambah_penghuni() digunakan untuk memarsing data dari *view form_tambah_penghuni* ke *function addPenghuni*. Sedangkan *function addPenghuni* digunakan menyimpan data ke dalam *database*.

Tabel 5.8 Implementasi pseudocode menambahkan penghuni

No.	Kode Program
1 2 3 4 5 6 7 8 9 10 11 12 13	<pre> public function tambah_penghuni(){ \$created = date('Y-m-d'); \$session = \$this->session- >userdata('register_nama_pemilik'); \$nama_penghuni = \$this->input->post('nama_penghuni'); \$identitas = \$this->input->post('identitas'); \$alamat = \$this->input->post('alamat'); \$tempat_lahir = \$this->input->post('tempat_lahir'); \$tgl_lahir = \$this->input->post('tgl_lahir'); \$telp_penghuni = \$this->input->post('telp_penghuni'); \$email_penghuni = \$this->input->post('email_penghuni'); \$pekerjaan_penghuni = \$this->input- >post('pekerjaan_penghuni'); } </pre>

14	\$kelamin_penghuni = \$this->input->post('kelamin_penghuni');
15	

Tabel 5.8 Implementasi pseudocode menambahkan penghuni (lanjutan)

No.	Kode Program
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42	<pre> \$tgl_masuk = \$this->input->post('tgl_masuk'); \$indekos = \$this->input->post('indekos'); \$no_kamar = \$this->input->post('no_kamar'); \$data = array ('penghuni_session_pemilik' => \$session, 'penghuni_namaindekos_pemilik' => \$indekos, 'penghuni_nokamar_pemilik' => \$no_kamar, 'penghuni_nama_pemilik' => \$nama_penghuni, 'penghuni_identitas_pemilik' =>\$identitas, 'penghuni_alamat_pemilik'=>\$alamat, 'penghuni_tmpatlahir_pemilik'=>\$tempat_lahir, 'penghuni_tgllahir_pemilik'=>date('Y-m- d',strtotime(\$tgl_lahir)), 'penghuni_notlp_pemilik'=>\$telp_penghuni, 'penghuni_email_pemilik'=>\$email_penghuni, 'penghuni_instansi_pemilik'=>\$pekerjaan_penghuni, 'penghuni_jeniskelamin_pemilik'=>\$kelamin_penghuni, 'penghuni_tglmasuk_pemilik'=>date('Y-m- d',strtotime(\$tgl_masuk)), 'created_at'=>\$created,); \$this->backend->addPenghuni(\$data); echo "<script> alert('Data berhasil diupload'); window.location.href='form_tambah_penghuni'; </script>"; </pre>

Penjelasan :

Pada baris 2 – 18 merupakan pendeklarasian variable data

Pada baris 20 – 38 merupakan kondisi untuk menyimpan data yang dimasukkan pemilik ke dalam database

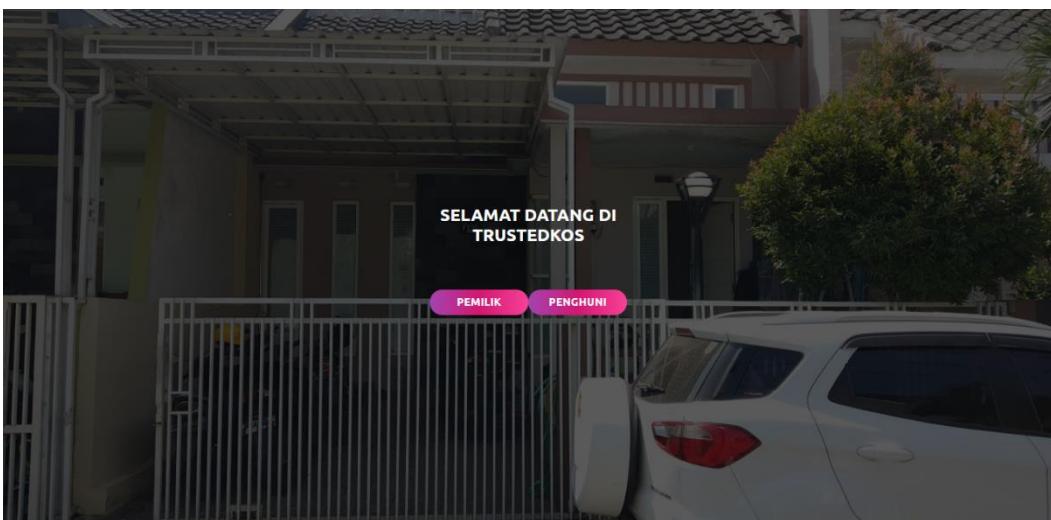
Pada baris 40 merupakan pesan berhasil ketika data masuk ke dalam database

5.2.4 Implementasi Antarmuka

Pada tahapan mengimplementasikan antarmuka merupakan penerapan dari perancangan antarmuka yang telah dibuat. Antarmuka tersebut digunakan untuk saran interaksi secara langsung antara pengguna dan sistem. Berikut di bawah ini merupakan implementasi antarmuka untuk Sistem Manajemen Rumah Indekos.

5.2.4.1 Implementasi Antarmuka Utama Sistem

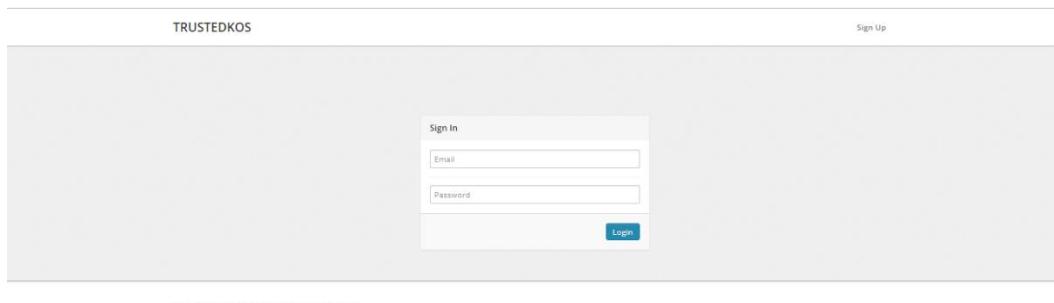
Antarmuka halaman utama sistem akan ditampilkan ketika pada saat sistem pertama kali dijalankan. Gambar 5.32 di bawah ini merupakan tampilan implementasi antarmuka dari halaman utama sistem.



Gambar 5.29 Implementasi antarmuka halaman utama sistem

5.2.4.2 Implementasi Antarmuka Masuk ke Dalam Sistem

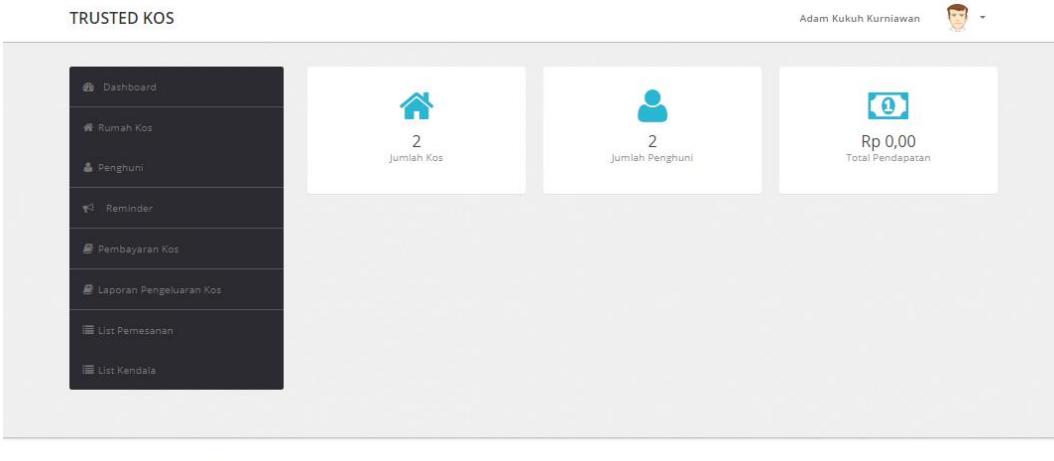
Antarmuka halaman masuk ke dalam sistem pemilik akan ditampilkan ketika pemilik sudah menekan tombol pemilik . Gambar 5.33 di bawah ini merupakan tampilan implementasi antarmuka dari halaman masuk ke dalam sistem.



Gambar 5.30 Implementasi antarmuka masuk ke dalam sistem

5.2.4.3 Implementasi Antarmuka Dashboard Pemilik

Antarmuka halaman *dashboard* pemilik akan ditampilkan ketika pemilik sudah masuk ke dalam sistem. Gambar 5.34 di bawah ini merupakan tampilan implementasi antarmuka dari halaman masuk ke dalam sistem.



Gambar 5.31 Implementasi antarmuka halaman *dashboard* pemilik

5.2.4.4 Implementasi Antarmuka Rumah Indekos

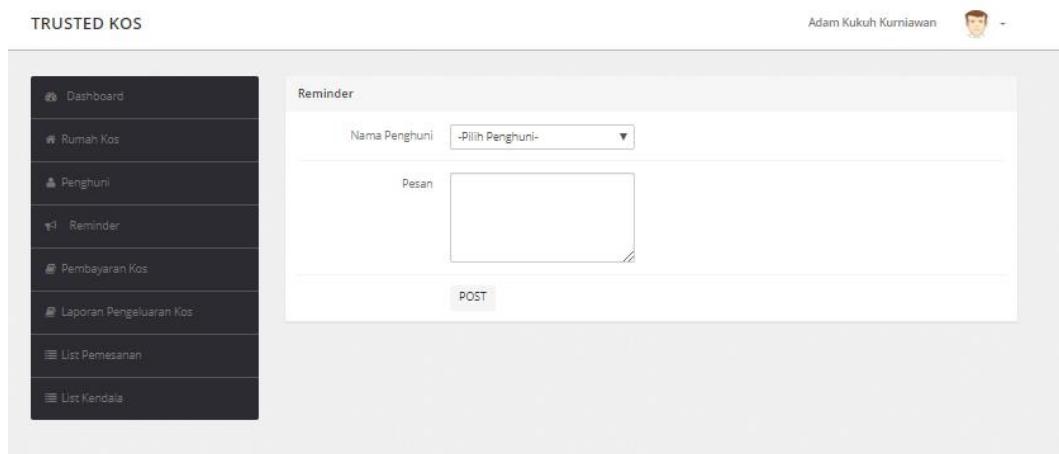
Antarmuka halaman rumah indekos akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu rumah indekos. Gambar 5.35 di bawah ini merupakan tampilan implementasi antarmuka dari rumah indekos.



Gambar 5.32 Implementasi antarmuka halaman rumah indekos

5.2.4.5 Implementasi *Reminder Whatsapp*

Antarmuka halaman *reminder whatsapp* akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu *reminder whatsapp*. Gambar 5.36 di bawah ini merupakan tampilan implementasi antarmuka dari *reminder whatsapp*.



Gambar 5.33 Implementasi antarmuka halaman *reminder*

5.2.4.6 Implementasi Antarmuka Data Pembayaran

Antarmuka halaman data pembayaran akan ditampilkan ketika penghuni sudah masuk ke dalam sistem dan menekan menu data pembayaran. Gambar 5.37 di bawah ini merupakan tampilan implementasi antarmuka dari data

Data Pembayaran Kos						
Nama	Nominal	Status	Tanggal	Periode	Metode Pembayaran	Aksi
Fadil	Rp 5.000.000	lunas	2019-06-27	1	transfer	Cetak Bukti Pembayaran

pembayaran.

Gambar 5.34 Implementasi antarmuka halaman data pembayaran

5.2.4.7 Implementasi Antarmuka Pengeluaran

Antarmuka halaman pengeluaran akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu pengeluaran. Gambar 5.38 di bawah ini merupakan tampilan implementasi antarmuka dari pengeluaran.

Detail Pengeluaran	Jumlah Pengeluaran	Tanggal Pengeluaran	Aksi
Beli rokok dan Minuman	Rp 5.000.000	2019-06-20	Edit

Gambar 5.35 Implementasi antarmuka halaman pengeluaran

5.2.4.8 Implementasi Antarmuka *List* Pemesanan

Antarmuka halaman *list pemesanan* akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu pemesanan. Gambar 5.39 di bawah ini merupakan tampilan implementasi antarmuka dari pemesanan.

Nama	Identitas	Indekos	Status	Aksi
Kosong	Kosong	Kosong	Kosong	Kosong

Gambar 5.36 Implementasi antarmuka halaman *list* pemesanan

5.2.4.9 Implementasi Antarmuka *List* Kendala

Antarmuka halaman *list kendala* akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu *list kendala*. Gambar 5.40 di bawah ini merupakan tampilan implementasi antarmuka dari pemesanan.

Gambar 5.37 Implementasi antarmuka halaman *list kendala*

5.2.4.10 Implementasi Antarmuka Tambah Indekos

Antarmuka halaman tambah indekos akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu tambah indekos. Gambar 5.41 di bawah ini merupakan tampilan implementasi antarmuka dari tambah indekos.

The screenshot shows a user interface for adding a room (Kos). On the left is a sidebar with navigation links: Dashboard, Rumah Kos, Penghuni, Reminder, Pembayaran Kos, Laporan Pengeluaran Kos, List Pemesanan, and List Kendala. The main area is titled 'Forms Tambah Kos' and contains fields for inputting room details. The fields include: Nama Kos (Name Kos), Harga Kos (Price Kos), Jumlah Kamar (Number of Rooms), Tipe Kos (Type Kos), Luas Kos (Area Kos), Lokasi Kos (Location Kos), Fasilitas Kos (Facilities Kos), and Foto Kos (Photo Kos). A 'Choose File' button is provided for uploading a photo, and a 'POST' button is at the bottom right.

Gambar 5.38 Implementasi antarmuka halaman tambah rumah indekos

5.2.4.11 Implementasi Antarmuka Tambah Penghuni

Antarmuka halaman tambah penghuni akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu tambah penghuni. Gambar 5.42 di bawah ini merupakan tampilan implementasi antarmuka dari tambah penghuni.



Forms Tambah Penghuni

Nama Penghuni	<input type="text" value="Nama Penghuni Kos"/>
Identitas Penghuni	<input type="text" value="(Kartu Identitas) (Nomor Identitas)"/>
Alamat Penghuni	<input type="text" value="Alamat Asal Penghuni Kos"/>
Tempat Lahir Penghuni	<input type="text" value="Tempat Lahir Penghuni Kos"/>
Tanggal Lahir Penghuni	<input type="text" value="Tanggal Lahir Penghuni"/>
No Telpon Penghuni	<input type="text" value="No Telepon Penghuni"/>
Email Penghuni	<input type="text" value="Email Penghuni"/>
Pekerjaan Penghuni	<input type="text" value="Pekerjaan Penghuni"/>
Jenis Kelamin Penghuni	<input type="text" value="Jenis Kelamin Penghuni"/>
Tanggal Masuk Penghuni	<input type="text" value="Tanggal Masuk Penghuni"/>
Indekos	<input type="text" value="•Pilih Indekos•"/>
No Kamar	<input type="text" value="No Kamar Penghuni"/>
POST	

Gambar 5.39 Implementasi antarmuka halaman tambah penghuni

5.2.4.12 Implementasi Antarmuka Tambah Akun Penghuni

Antarmuka halaman tambah akun penghuni akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu tambah akun penghuni. Gambar 5.43 di bawah ini merupakan tampilan implementasi antarmuka dari tambah akun penghuni.

Forms Akun

Nama Penghuni	<input type="text" value="•Pilih Penghuni•"/>
Username	<input type="text" value="Username"/>
Password	<input type="text" value="Password"/>
POST	

Gambar 5.40 Implementasi antarmuka halaman tambah akun penghuni

5.2.4.13 Implementasi Antarmuka Mengubah Penghuni

Antarmuka halaman mengubah penghuni akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu mengubah penghuni. Gambar 5.44 di bawah ini merupakan tampilan implementasi antarmuka dari mengubah penghuni.

The screenshot displays the 'Forms Edit Penghuni' page within the TRUSTED KOS application. On the left, a vertical sidebar lists several menu items: Dashboard, Rumah Kos, Penghuni (which is currently selected and highlighted in blue), Reminder, Pembayaran Kos, Laporan Pengeluaran Kos, List Pemesanan, and List Kendala. At the top right, there is a user profile icon and some text indicating the user's name. The main content area is titled 'Forms Edit Penghuni' and contains a form with the following fields and values:

Nama Penghuni	Bohong
Identitas Penghuni	(Kartu Identitas) (Nomor Identitas)
Alamat Penghuni	Alamat Asal Penghuni Kos
Tempat Lahir Penghuni	Tempat Lahir Penghuni Kos
Tanggal Lahir Penghuni	1970-01-01
No Telpon Penghuni	No Telepon Penghuni
Email Penghuni	Email Penghuni
Pekerjaan Penghuni	Pekerjaan Penghuni
Jenis Kelamin Penghuni	Jenis Kelamin Penghuni
Tanggal Masuk Penghuni	1970-01-01
Indekos	Semanggi 16 A
No Kamar	0

At the bottom right of the form, there is a 'POST' button.

Gambar 5.41 Implementasi antarmuka halaman mengubah penghuni

5.2.4.14 Implementasi Antarmuka Mendaftar ke Sistem

Antarmuka halaman mendaftar ke sistem akan ditampilkan ketika pemilik menekan tombol *sign up*. Gambar 5.43 di bawah ini merupakan tampilan implementasi antarmuka dari mendaftar ke sistem.

The screenshot shows a registration form titled 'Sign Up' on a website called 'TRUSTEDKOS'. The form includes fields for Name, Email, Password, Address, Phone Number, Birthplace, Birthdate, and Gender. A 'Register' button is at the bottom.

Name
Email
Password
Address
Phone
Birthplace
Birthdate
Gender

Register

Gambar 5.42 Implementasi antarmuka halaman mendaftar ke sistem

5.2.4.15 Implementasi Antarmuka Pemesanan Kamar

Antarmuka halaman pemesanan kamar akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu rumah indekos. Gambar 5.46 di bawah ini merupakan tampilan implementasi antarmuka dari pemesanan kamar.

The screenshot shows a booking form titled 'Forms Pemesanan'. It includes fields for Guest Name, Identity Card Number, Room Name, Phone Number, Booking Date, and Check-in Date. A dropdown menu for selecting a room is also present. A 'POST' button is at the bottom.

Nama Penghuni	Nama Penghuni Kos
Identitas Penghuni	(Kartu Identitas) (Nomor Identitas)
Nama Kos	Semanggi 16 A
Pilih Kamar	-Pilih No Kamar-
No Telepon	No Telepon
Tanggal Pesan	2019-06-29
Batas Tanggal Pemesanan	Batas Tanggal Pemesanan

POST

Gambar 5.43 Implementasi antarmuka halaman pemesanan kamar

5.2.4.16 Implementasi Antarmuka Tagihan

Antarmuka halaman tagihan akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu pembayaran. Gambar 5.47 di bawah ini merupakan tampilan implementasi antarmuka dari pembayaran.

The screenshot shows a user interface for managing bills. On the left is a dark sidebar menu with options: Dashboard, Rumah Kos, Penghuni, Reminder, Pembayaran Kos, Laporan Pengeluaran Kos, List Pemesanan, and List Kendala. The main area is titled 'Forms Tagihan'. It contains several input fields: 'Indekos' (dropdown menu with placeholder 'Pilih Indekos...'), 'Nama Penghuni' (dropdown menu with placeholder 'Pilih Nama Penghuni...'), 'Metode Pembayaran' (dropdown menu with placeholder 'Pilih Metode Pembayaran...'), 'Nominal' (text input field with placeholder 'Masukkan Jumlah Nominal'), 'Periode Pembayaran' (text input field with placeholder 'Masukkan Periode Pembayaran'), and 'Tanggal Tagihan' (text input field with placeholder 'Tanggal Tagihan'). At the bottom is a 'POST' button.

Gambar 5.44 Implementasi antarmuka halaman pembayaran

5.2.4.17 Implementasi Antarmuka Tambah Pengeluaran

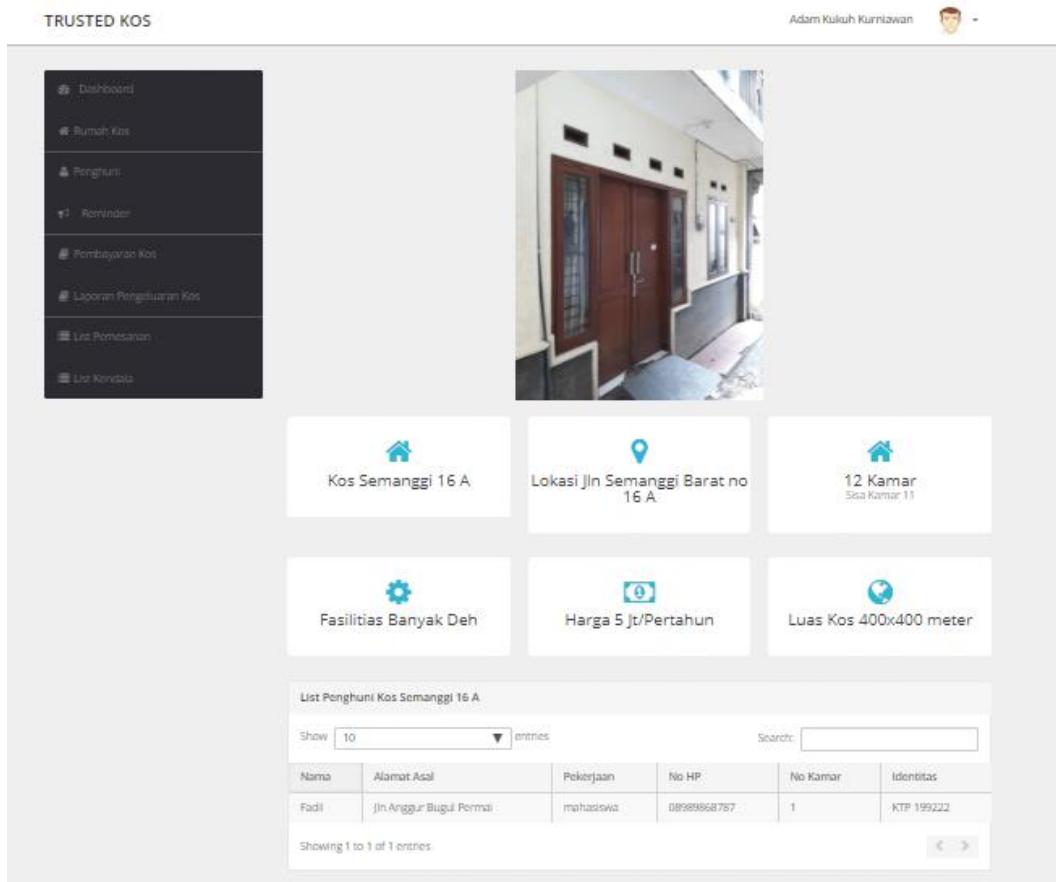
Antarmuka halaman tambah pengeluaran akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu pengeluaran. Gambar 5.48 di bawah ini merupakan tampilan implementasi antarmuka dari pengeluaran.

The screenshot shows a user interface for adding expenses. On the left is a dark sidebar menu with options: Dashboard, Rumah Kos, Penghuni, Reminder, Pembayaran Kos, Laporan Pengeluaran Kos, List Pemesanan, and List Kendala. The main area is titled 'Forms Tambah Pengeluaran'. It contains three input fields: 'Nominal' (text input field with placeholder 'Nominal'), 'Detail' (a large text area for expense details), and 'Tanggal Pengeluaran' (text input field with placeholder 'Tanggal Pengeluaran'). At the bottom is a 'POST' button.

Gambar 5.45 Implementasi antarmuka halaman tambah pembayaran

5.2.4.18 Implementasi Antarmuka *Detail* Rumah Indekos

Antarmuka halaman *detail* rumah indekos akan ditampilkan ketika pemilik sudah masuk ke dalam sistem dan menekan menu rumah indekos.



Gambar 5.49 di bawah ini merupakan tampilan implementasi antarmuka dari *detail* rumah indekos.

Gambar 5.46 Implementasi antarmuka halaman *detail* rumah indekos

5.2.4.19 Implementasi Antarmuka Data Pembayaran

Antarmuka status pembayaran akan ditampilkan ketika penghuni sudah masuk ke dalam sistem dan menekan data pembayaran. Gambar 5.50 di bawah ini merupakan tampilan implementasi antarmuka dari data pembayaran.

Nama	Nominal	Status	Tanggal	Periode	Metode Pembayaran	Aksi
Fadil	Rp 5.000.000	lunas	2019-06-27	1	transfer	Cetak Bukti Pembayaran

Gambar 5.47 Implementasi antarmuka halaman data pembayaran

5.2.4.20 Implementasi Antarmuka Lapor Kendala

Antarmuka lapor kendala akan ditampilkan ketika penghuni sudah masuk ke dalam sistem dan menekan lapor kendala. Gambar 5.51 di bawah ini merupakan tampilan implementasi antarmuka dari lapor kendala.

Belum ada postingan kendala

Gambar 5.48 Implementasi antarmuka halaman lapor kendala

5.2.4.21 Implementasi Antarmuka Status Pembayaran

Antarmuka status pembayaran akan ditampilkan ketika penghuni sudah masuk ke dalam sistem dan menekan status pembayaran indekos. Gambar 5.52 di bawah ini merupakan tampilan implementasi antarmuka dari status pembayaran indekos.

Nama	Nominal	Status	Tanggal	Periode
Fadil	Rp 5.000.000	lunas	2019-06-27	1

Gambar 5.49 Implementasi antarmuka halaman status pembayaran indekos

BAB 6 PENGUJIAN

Pada bab ini akan melakukan tahapan pengujian dari Sistem Manajemen Rumah Indekos. Tahapan pengujian yang dilakukan melalui 2 (dua) tahapan yaitu pengujian unit dan pengujian validasi. Analisis dari hasil pengujian tersebut digunakan untuk memastikan bahwa sistem telah memenuhi kebutuhan yang telah didefinisikan pada tahapan rekayasa kebutuhan serta dapat menemukan kesalahan yang ada di dalam sistem.

6.1 Pengujian Unit

Pengujian unit merupakan pengujian yang digunakan untuk menguji setiap detail pada tahapan perancangan sistem. Dalam pengujian unit ini menggunakan jenis pengujian *basis-path*. *Pseudocode* yang telah dibuat nantinya akan disusun menggunakan *flow-graph*. *Flow-graph* yang disusun bertujuan untuk mengetahui banyaknya dari *cyclomatic complexity* dan jalur independennya.

6.1.1 Pengujian Unit Masuk ke Dalam Sistem

Tabel 6.1 Pengujian unit masuk ke dalam sistem

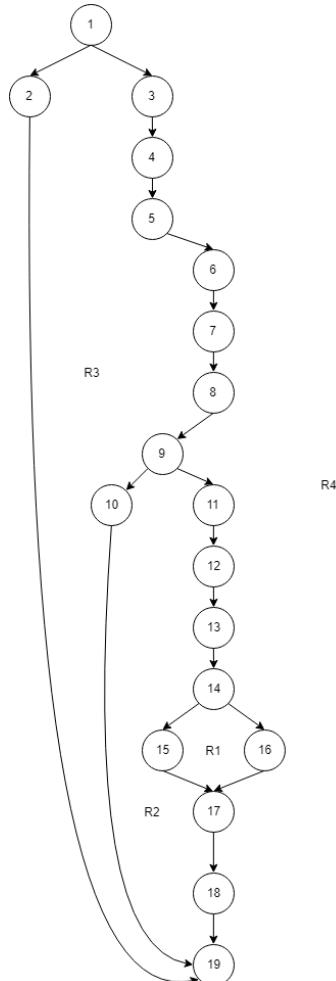
```
if(cek == pemilik) 1
    redirect(dashboard_pemilik); 2
else 3
    this->form_validation->set_rules(email, email, required); 4
    this->form_validation->set_rules(password, password, required); 4
    this->form_validation->set_message; 4
    if(this->form_validation->run() == TRUE) 5
        email = this->input->post(email, TRUE); 6
        password = MD5(this->input->post(password, TRUE)); 6
        c1 = array(register_email_pemilik=>email); 7
        c2 = array(register_password_pemilk=>$password); 7
        checking_user = this->autentikasi->cek_login(c1,c2); 8
        if(checking_user != FALSE) 9
            foreach (checking_user as data_user) 10
                data = array(11
                    register_id_pemilik => data_user->register_id_pemilik,
                    register_email_pemilik => data_user->register_email_pemilik,
                    register_nama_pemilik => data_user->register_nama_pemilik,
                    register_password_pemilk => data_user-
>register_password_pemilk,
                    register_hakakses_pemilik => data_user-
>register_hakakses_pemilik,
                );12
                this->session->set_userdata(data);13
```

```
}
```

Tabel 6.1 Pengujian unit masuk ke dalam sistem(lanjutan)

```
if(this->session->userdata(register_hakakses_pemilik) == pemilik)14  
    tampil "Selamat datang di Panel TRUSTEDKOS" 15  
    tampil dashboard_pemilik  
else  
    data[error] Username atau Password Salah!!!. 16  
  
this->load->view(pemilik/login_form/login,data);  
  
else  
    $this->load->view('pemilik/login_form/login');17  
End if 18  
End if 19
```

Berdasarkan dari *pseudocode* pada Tabel 6.1 diatas, dapat diperoleh bentuk *flow graph* yang akan ditunjukkan pada Gambar 6.1 di bawah.



Gambar 6.1 *Flow graph* masuk ke dalam sistem

Berdasarkan *flowgraph* yang diperoleh, maka dapat ditentukan jalur independen serta jumlah kompleksitas, seperti berikut.

Jalur 1 : 1-2-19

Jalur 2 : 1-3-4-5-6-7-8-9-10-19

Jalur 3 : 1-3-4-5-6-7-8-9-11-12-13-14-15-17-18-19

Jalur 4 : 1-3-4-5-6-7-8-9-11-12-13-14-16-17-18-19

Perhitungan *cyclomatic complexity*:

Jumlah region = 4

$$V(G) = E - N + 2$$

$$= 21 - 19 + 2$$

$$= 2 + 2$$

$$= 4$$

Berdasarkan dari jalur yang diperoleh dari perhitungan diatas, maka terbentuk kasus uji. Pemaparan kasus uji *pseudocode* fungsi masuk ke dalam sistem terdapat pada Tabel 6.2.

Tabel 6.2 Pengujian unit masuk ke dalam sistem

No.	Jalur	Prosedur Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
1.	1-2-19	Menjalankan operasi if(\$cek == 'pemilik')	Masuk ke dalam sistem dengan aktor sebagai pemilik	Masuk ke dalam sistem dengan aktor sebagai pemilik	Valid
2.	1-3-4-5-6-7-8-9-10-19	Menjalankan operasi if(\$checking_user != FALSE)	Sistem akan menampilkan pesan <i>error</i> di halaman masuk	Sistem akan menampilkan pesan <i>error</i> di halaman masuk	Valid
3.	1-3-4-5-6-7-8-9-11-12-13-14-15-17-18-19	Menjalankan operasi (this->session->userdata(register_hakkses	Sistem akan menampilkan pesan "Selamat datang di Panel TRUSTEDKOS"	Sistem akan menampilkan pesan "Selamat datang di Panel TRUSTEDKOS"	Valid

		<code>_pemilik) == pemilik)</code>	Panel TRUSTEDKOS” pada <i>dashboard</i> pemilik	pada <i>dashboard</i> pemilik	
--	--	------------------------------------	---	-------------------------------	--

Tabel 6.2 Pengujian unit masuk ke dalam sistem (lanjutan)

No.	Jalur	Prosedur Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
4.	1-3-4-5-6-7-8-9-11-12-13-14-16-17-18-19	Menjalankan operasi <code>elseif(\$checkin g_user == FALSE)</code>	Sistem akan menampilkan pesan “Username atau Password Salah!!!”	Sistem akan menampilkan pesan “Username atau Password Salah!!!”	Valid

6.1.2 Pengujian Unit Menambahkan Rumah Indekos

Tabel 6.3 Pengujian unit menambahkan rumah indekos

```

session = this->session->userdata(register_nama_pemilik);1
nama_kos = this->input->post(nama_kos);1
lokasi_kos = this->input->post(lokasi_kos);1
jumlah_kamar = this->input->post(jumlah_kamar);1
tipe_kos = this->input->post(tipe_kos);1
luas_kos = this->input->post(luas_kos);1
fasilitas_kos = this->input->post(fasilitas_kos);1
harga_kos = this->input->post(harga_kos);1
foto_kos = this->input->post(nama_kos);1
slug = slug(nama_kos);1
date = date(format tanggal);1

$config['upload_path'] = './asset/foto_kos';2
$config['allowed_types'] = 'jpg|png|jpeg|gif';2
$config['max_size'] = '1024'; //2MB max 2
$config['max_width'] = '4480'; // pixel 2

```

```

$config['max_height'] = '4480'; // pixel 2
$config['file_name'] = $_FILES['foto']['name']; 2
$this->upload->initialize($config); 2
if(!empty($_FILES['foto']['name'])) 3
    if($this->upload->do_upload('foto')) 4

```

Tabel 6.3 Pengujian unit menambahkan rumah indekos (lanjutan)

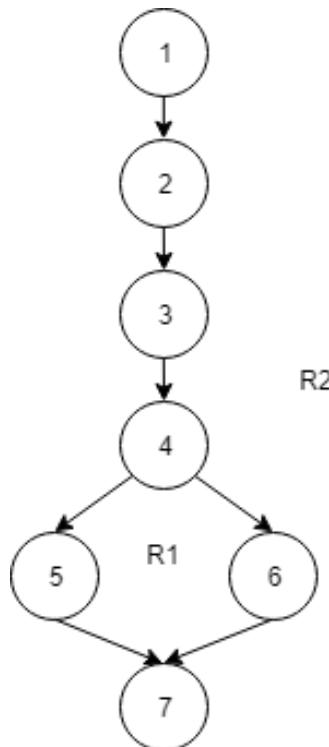
```

$ foto = $this->upload->data(); 4
    $data = array( 4
        'rumahkos_session_pemilik' => $session, 4
        'rumahkos_namaindekos_pemilik' => $nama_kos, 4
        'rumahkos_lokasi_pemilik' => $lokasi_kos, 4
        'rumahkos_jmlkamar_pemilik' => $jumlah_kamar, 4
        'rumahkos_tipe_pemilik' => $tipe_kos, 4
        'rumahkos_foto_pemilik' => $foto['file_name'], 4
        'rumahkos_luas_pemilik' => $luas_kos, 4
        'rumahkos_fasilitas_pemilik' => $fasilitas_kos, 4
        'rumahkos_harga_pemilik' => $harga_kos, 4
        'slug_indekos' => $slug, 4
        'created_at' => $date, 4
    this->backend->addRumahkos(data); 5
    menampilkan pesan berhasil 6
else
    menampilkan pesan gagal upload 7
else
    menampilkan pesan gagal upload 7
end if

```

```
end if
```

Berdasarkan dari *pseudocode* pada Tabel 6.3 diatas, dapat diperoleh bentuk *flow graph* yang akan ditunjukkan pada Gambar 6.2 di bawah.



Gambar 6.2 Flow graph menambahkan penghuni

Berdasarkan *flowgraph* yang diperoleh, maka dapat ditentukan jalur independen serta jumlah kompleksitas, seperti berikut.

Jalur 1 : 1-2-3-4-5-7

Jalur 2 : 1-2-3-4-6-7

Perhitungan *cyclomatic complexity*:

Jumlah region = 2

$$V(G) = E - N + 2$$

$$= 7 - 7 + 2$$

$$= 0 + 2$$

$$= 2$$

Berdasarkan dari jalur yang diperoleh dari perhitungan diatas, maka terbentuk kasus uji. Pemaparan kasus uji *pseudocode* fungsi masuk ke dalam sistem terdapat pada Tabel 6.3.

Tabel 6.4 Pengujian unit menambahkan rumah indekos

No.	Jalur	Prosedur Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
1.	1-2-3-4-5-7	Menjalankan operasi if((\$_FILES['foto']['name']) != empty)	Pemilik berhasil menambahkan rumah indekos	Pemilik berhasil menambahkan rumah indekos	Valid
2.	1-2-3-4-6-7	Menjalankan operasi if((\$_FILES['foto']['name']) = empty)	Pemilik tidak berhasil menambahkan rumah indekos	Pemilik tidak berhasil menambahkan rumah indekos	Valid

6.1.3 Pengujian Unit Menambahkan Penghuni

Tabel 6.5 Pengujian unit menambahkan penghuni

```

public function tambah_penghuni(){ 1
    $created = date('Y-m-d'); 1
    $session = $this->session->userdata('register_nama_pemilik'); 1
    $nama_penghuni = $this->input->post('nama_penghuni'); 1
    $identitas = $this->input->post('identitas'); 1
    $alamat = $this->input->post('alamat'); 1
    $tempat_lahir = $this->input->post('tempat_lahir'); 1
    $tgl_lahir = $this->input->post('tgl_lahir'); 1
    $telp_penghuni = $this->input->post('telp_penghuni'); 1
    $email_penghuni = $this->input->post('email_penghuni'); 1
    $pekerjaan_penghuni = $this->input->
post('pekerjaan_penghuni'); 1
    $kelamin_penghuni = $this->input->post('kelamin_penghuni'); 1
    $tgl_masuk = $this->input->post('tgl_masuk'); 1
    $indekos = $this->input->post('indekos'); 1
    $no_kamar = $this->input->post('no_kamar'); 1

```

Tabel 6.5 Pengujian unit menambahkan penghuni (lanjutan)

```

$data = array ( 2
    'penghuni_session_pemilik' => $session, 2
    'penghuni_namaindekos_pemilik' => $indekos, 2
    'penghuni_nokamar_pemilik' => $no_kamar, 2
    'penghuni_nama_pemilik' => $nama_penghuni, 2
    'penghuni_identitas_pemilik' => $identitas, 2
    'penghuni_alamat_pemilik' => $alamat, 2
    'penghuni_tmpatlahir_pemilik' => $tempat_lahir, 2
    'penghuni_tglahir_pemilik' => date('Y-m-
d', strtotime($tgl_lahir)), 2
    'penghuni_notlp_pemilik' => $telp_penghuni, 2
    'penghuni_email_pemilik' => $email_penghuni, 2
    'penghuni_instansi_pemilik' => $pekerjaan_penghuni, 2

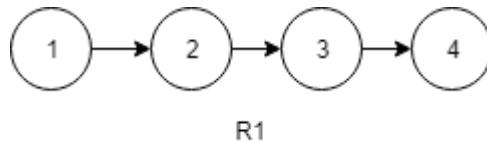
```

```

'penghuni_jeniskelamin_pemilik'=>$kelamin_penghuni, 2
'penghuni_tglmasuk_pemilik'=>date('Y-m-
d',strtotime($tgl_masuk)), 2
'created_at'=>$created, 2
);
$this->backend->addPenghuni($data); 3
echo 4
alert('Data berhasil diupload'); 4
window.location.href='form_tambah_penghuni';
</script>";
}

```

Berdasarkan dari *pseudocode* pada Tabel 6.5 diatas, dapat diperoleh bentuk *flow graph* yang akan ditunjukkan pada Gambar 6.3 di bawah.



Gambar 6.3 Flow graph menambahkan penghuni

Berdasarkan *flowgraph* yang diperoleh, maka dapat ditentukan jalur independen serta jumlah kompleksitas, seperti berikut.

Jalur 1 : 1-2-3-4

Perhitungan *cyclomatic complexity*:

Jumlah region = 1

$$V(G) = E - N + 2$$

$$= 3 - 4 + 2$$

$$= (-1) + 2$$

$$= 1$$

Berdasarkan dari jalur yang diperoleh dari perhitungan diatas, maka terbentuk kasus uji. Pemaparan kasus uji *pseudocode* fungsi masuk ke dalam sistem terdapat pada Tabel 6.6.

Tabel 6.6 Pengujian unit menambahkan penghuni

No.	Jalur	Prosedur Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
1.	1-2-3-4	Menjalankan operasi <i>function</i> tambah_penghuni = \$data	Pemilik berhasil menambahkan penghuni baru	Pemilik berhasil menambahkan penghuni baru	Valid

6.2 Pengujian Validasi

Semua kebutuhan pada sistem dilakukan pengujian validasi menggunakan pengujian *black box*. Dalam melakukan pengujian validasi mengacu pada daftar kebutuhan yang telah dibuat. Pengujian validasi sendiri digunakan untuk mengetahui apakah sistem yang sudah dibangun telah sesuai dengan seluruh kebutuhan yang telah ditetapkan atau tidak terdapat *error* pada sistem

6.2.1 Pengujian Validasi Kebutuhan Fungsional

Tabel 6.7 Pengujian validasi mendaftar ke sistem

Nama Kasus Uji	Kasus uji mendaftar ke sistem
Kode Kebutuhan	SRS_1_01
Prosedur	<ol style="list-style-type: none"> Pemilik memilih tombol <i>Daftar</i> Pemilik mengisi data diri yang sudah disediakan ke dalam sistem Pemilik menekan tombol <i>submit</i>

Tabel 6.7 Pengujian validasi mendaftar ke sistem (lanjutan)

Hasil yang diharapkan	Data akan tersimpan di dalam <i>database</i> dan pemilik indekos terdaftar di dalam sistem
Hasil	Data akan tersimpan di dalam <i>database</i> dan pemilik indekos terdaftar di dalam sistem
Status	Valid

Tabel 6.7 Pengujian validasi mendatar ke sistem

Nama Kasus Uji	Kasus uji mendaftar ke sistem
Kode Kebutuhan	SRS_1_01
Prosedur	<ol style="list-style-type: none"> Pemilik memilih tombol <i>Daftar</i> Pemilik mengisi data diri yang sudah disediakan ke dalam sistem

	3. Pemilik menekan tombol <i>submit</i>
Hasil yang diharapkan	Sistem akan menampilkan pesan “Please fill out this fill” bila data tidak diisi lengkap
Hasil	Data akan tersimpan di dalam <i>database</i> dan pemilik indekos terdaftar di dalam sistem
Status	Valid

Tabel 6.8 Pengujian validasi masuk ke sistem

Nama Kasus Uji	Kasus uji masuk ke sistem
Kode Kebutuhan	SRS_1_02
Prosedur	<ol style="list-style-type: none"> Pemilik dan penghuni memilih tombol <i>Login</i> Pemilik Pemilik dan penghuni memasukkan <i>email</i> dan <i>password</i> yang sudah didaftarkan ke dalam sistem Pemilik dan penghuni menekan tombol <i>submit</i>
Hasil yang diharapkan	Kombinasi <i>username</i> dan <i>password</i> salah. <i>User</i> akan melihat notifikasi gagal login dan kembali pada halaman login
Hasil	Pemilik dan penghuni dapat melihat data yang dimiliki di masing-masing halaman
Status	Valid

Tabel 6.8 Pengujian validasi masuk ke sistem

Nama Kasus Uji	Kasus uji masuk ke sistem
Kode Kebutuhan	SRS_1_02
Prosedur	<ol style="list-style-type: none"> Pemilik dan penghuni memilih tombol <i>Login</i> Pemilik Pemilik dan penghuni memilih memasukkan <i>email</i> dan <i>password</i> yang sudah didaftarkan ke dalam sistem Pemilik dan penghuni menekan tombol <i>submit</i>
Hasil yang diharapkan	Pemilik akan melihat notifikasi gagal “ <i>Username</i> atau <i>password</i> salah!!!!” dan kembali pada halaman login
Hasil	Pemilik dan penghuni dapat melihat data yang dimiliki di masing-masing halaman

Status	Valid
---------------	-------

Tabel 6.9 Pengujian validasi menambahkan rumah indekos

Nama Kasus Uji	Kasus uji menambahkan rumah indekos
Kode Kebutuhan	SRS_1_03
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu rumah indekos 2. Pemilik memilih <i>icon</i> tambah rumah indekos 3. Pemilik dapat menambahkan rumah indekos baru dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan
Hasil yang diharapkan	Pemilik berhasil menambah rumah indekos baru yang berada di menu rumah indekos
Hasil	Pemilik berhasil menambah rumah indekos baru yang berada di menu rumah indekos
Status	Valid

Tabel 6.9 Pengujian validasi menambahkan rumah indekos

Nama Kasus Uji	Kasus uji menambahkan rumah indekos
Kode Kebutuhan	SRS_1_03
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu rumah indekos 2. Pemilik memilih <i>icon</i> tambah rumah indekos 3. Pemilik dapat menambahkan rumah indekos baru dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan

Tabel 6.9 Pengujian validasi menambahkan rumah indekos (lanjutan)

Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi " <i>Please fill out this fill</i> "
Hasil	Pemilik berhasil menambah rumah indekos baru yang berada di menu rumah indekos
Status	Valid

Tabel 6.10 Pengujian validasi pemesanan kamar

Nama Kasus Uji	Kasus uji pemesanan kamar
Kode Kebutuhan	SRS_1_07
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu rumah indekos

	2. Pemilik memilih <i>icon</i> tambah rumah indekos 3. Pemilik dapat menambahkan pemesanan kamar dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan
Hasil yang diharapkan	Pemilik berhasil menambahkan pemesanan kamar pada rumah indekos tertentu
Hasil	Pemilik berhasil menambahkan pemesanan kamar pada rumah indekos tertentu
Status	Valid

Tabel 6.10 Pengujian validasi pemesanan kamar

Nama Kasus Uji	Kasus uji pemesanan kamar
Kode Kebutuhan	SRS_1_07
Prosedur	1. Pemilik memilih menu rumah indekos 2. Pemilik memilih <i>icon</i> tambah rumah indekos 3. Pemilik dapat menambahkan pemesanan kamar dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi " <i>Please fill out this fill</i> "
Hasil	Pemilik berhasil menambahkan pemesanan kamar pada rumah indekos tertentu
Status	Valid

Tabel 6.11 Pengujian validasi menambahkan penghuni

Nama Kasus Uji	Kasus uji menambahkan menambahkan penghuni
Kode Kebutuhan	SRS_1_04
Prosedur	1. Pemilik memilih menu penghuni 2. Pemilik memilih <i>icon</i> tambah penghuni 3. Pemilik dapat mengisi <i>form</i> yang tersedia 4. Pemilik dapat memilih tombol simpan 5. Penghuni baru sudah ditambahkan pada rumah indekos tertentu
Hasil yang diharapkan	Pemilik berhasil menambahkan penghuni baru yang berada di rumah indekos tertentu
Hasil	Pemilik berhasil menambahkan penghuni baru yang berada di rumah indekos tertentu
Status	Valid

Tabel 6.11 Pengujian validasi menambahkan penghuni

Nama Kasus Uji	Kasus uji menambahkan menambahkan penghuni
Kode Kebutuhan	SRS_1_04
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu penghuni 2. Pemilik memilih <i>icon</i> tambah penghuni 3. Pemilik dapat mengisi <i>form</i> yang tersedia 4. Pemilik dapat memilih tombol simpan 5. Penghuni baru sudah ditambahkan pada rumah indekos tertentu
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi "Please fill out this fill"
Hasil	Pemilik berhasil menambahkan penghuni baru yang berada di rumah indekos tertentu
Status	Valid

Tabel 6.12 Pengujian validasi mengubah penghuni

Nama Kasus Uji	Kasus uji menambahkan mengubah penghuni
Kode Kebutuhan	SRS_1_05
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu penghuni 2. Pemilik memilih data penghuni yang akan dirubah 3. Pemilik dapat merubah data lama dan mengganti data baru dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan

Tabel 6.12 Pengujian validasi mengubah penghuni (lanjutan)

	5. Data penghuni baru telah diperbarui
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
Hasil	Pemilik berhasil mengubah data penghuni yang berada di rumah indekos tertentu
Status	Valid

Tabel 6.12 Pengujian validasi mengubah penghuni

Nama Kasus Uji	Kasus uji menambahkan mengubah penghuni
Kode Kebutuhan	SRS_1_05
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu penghuni 2. Pemilik memilih data penghuni yang akan dirubah

	<ul style="list-style-type: none"> 3. Pemilik dapat merubah data lama dan mengganti data baru dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan 5. Data penghuni baru telah diperbarui
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi " <i>Please fill out this fill</i> "
Hasil	Pemilik berhasil mengubah data penghuni yang berada di rumah indekos tertentu
Status	Valid

Tabel 6.13 Pengujian validasi menghapus penghuni

Nama Kasus Uji	Kasus uji menghapus penghuni
Kode Kebutuhan	SRS_1_06
Prosedur	<ul style="list-style-type: none"> 1. Pemilik memilih menu penghuni 2. Pemilik memilih <i>icon</i> hapus penghuni 3. Pemilik dapat menghapus penghuni
Hasil yang diharapkan	Pemilik berhasil menghapus penghuni yang berada di sistem
Hasil	Pemilik berhasil menghapus penghuni yang berada di sistem
Status	Valid

Tabel 6.13 Pengujian validasi menghapus penghuni

Nama Kasus Uji	Kasus uji menghapus penghuni
Kode Kebutuhan	SRS_1_06
Prosedur	<ul style="list-style-type: none"> 1. Pemilik memilih menu penghuni 2. Pemilik memilih <i>icon</i> hapus penghuni 3. Pemilik dapat menghapus penghuni
Hasil yang diharapkan	Jika pemilik menekan tombol "tidak" maka sistem tidak akan mengeksekusi hapus penghuni
Hasil	Pemilik berhasil menghapus penghuni yang berada di sistem
Status	Valid

Tabel 6.14 Pengujian validasi menambahkan akun penghuni

Nama Kasus Uji	Kasus uji menambahkan akun penghuni
Kode Kebutuhan	SRS_1_13
Prosedur	<ul style="list-style-type: none"> 1. Pemilik memilih menu penghuni

	<ol style="list-style-type: none"> 2. Pemilik memilih <i>icon</i> akun penghuni 3. Pemilik dapat menambahkan akun penghuni dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan
Hasil yang diharapkan	Pemilik berhasil menambahkan akun penghuni
Hasil	Pemilik berhasil menambahkan akun penghuni
Status	Valid

Tabel 6.14 Pengujian validasi menambahkan akun penghuni

Nama Kasus Uji	Kasus uji menambahkan akun penghuni
Kode Kebutuhan	SRS_1_13
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu penghuni 2. Pemilik memilih <i>icon</i> akun penghuni 3. Pemilik dapat menambahkan akun penghuni dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi " <i>Please fill out this fill</i> "
Hasil	Pemilik berhasil menambahkan akun penghuni
Status	Valid

Tabel 6.15 Pengujian validasi menambahkan pembayaran

Nama Kasus Uji	Kasus uji menambahkan pembayaran
Kode Kebutuhan	SRS_1_15
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu pembayaran 2. Pemilik memilih <i>icon</i> pembayaran 3. Pemilik dapat menambahkan 4. Pemilik dapat menambahkan pembayaran dengan mengisi <i>form</i> yang tersedia 5. Pemilik memilih tombol simpan
Hasil yang diharapkan	Pemilik berhasil menambahkan pembayaran indekos
Hasil	Pemilik berhasil menambahkan pembayaran indekos
Status	Valid

Tabel 6.15 Pengujian validasi menambahkan pembayaran

Nama Kasus Uji	Kasus uji menambahkan pembayaran
Kode Kebutuhan	SRS_1_15
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu pembayaran 2. Pemilik memilih <i>icon</i> pembayaran 3. Pemilik dapat menambahkan 4. Pemilik dapat menambahkan pembayaran dengan mengisi <i>form</i> yang tersedia 5. Pemilik memilih tombol simpan
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi " <i>Please fill out this fill</i> "
Hasil	Pemilik berhasil menambahkan pembayaran indekos
Status	Valid

Tabel 6.16 Pengujian validasi mengubah pembayaran

Nama Kasus Uji	Kasus uji mengubah pembayaran
Kode Kebutuhan	SRS_1_15
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu pembayaran 2. Pemilik memilih penghuni yang telah membayar 3. Pemilik dapat mengubah pembayaran dengan mengisi <i>form</i> yang tersedia 4. Pemilik memilih tombol simpan

Tabel 6.16 Pengujian validasi mengubah pembayaran (lanjutan)

Hasil yang diharapkan	Pemilik berhasil mengubah pembayaran indekos
Hasil	Pemilik berhasil mengubah pembayaran indekos
Status	Valid

Tabel 6.16 Pengujian validasi mengubah pembayaran

Nama Kasus Uji	Kasus uji mengubah pembayaran
Kode Kebutuhan	SRS_1_15
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu pembayaran 2. Pemilik memilih penghuni yang telah membayar 3. Pemilik dapat mengubah pembayaran dengan mengisi <i>form</i> yang tersedia

	4. Pemilik memilih tombol simpan
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi " <i>Please fill out this fill</i> "
Hasil	Pemilik berhasil mengubah pembayaran indekos
Status	Valid

Tabel 6.17 Pengujian validasi mengirimkan notifikasi batas pembayaran

Nama Kasus Uji	Kasus uji mengirimkan notifikasi batas pembayaran
Kode Kebutuhan	SRS_1_08
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih <i>menu</i> melaporkan kerusakan 2. Mengisi <i>form</i> yang telah disediakan di dalam sistem 3. Pemilik menekan tombol <i>submit</i>
Hasil yang diharapkan	Pemilik berhasil mengirimkan notifikasi batas pembayaran
Hasil	Pemilik berhasil mengirimkan notifikasi batas pembayaran
Status	Valid

Tabel 6.17 Pengujian validasi mengirimkan notifikasi batas pembayaran

Nama Kasus Uji	Kasus uji mengirimkan notifikasi batas pembayaran
Kode Kebutuhan	SRS_1_08
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih <i>menu</i> melaporkan kerusakan 2. Mengisi <i>form</i> yang telah disediakan di dalam sistem

Tabel 6.17 Pengujian validasi mengirimkan notifikasi batas pembayaran (lanjutan)

	3. Pemilik menekan tombol <i>submit</i>
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi " <i>Please fill out this fill</i> "
Hasil	Pemilik berhasil mengirimkan batas pembayaran indekos
Status	Valid

Tabel 6.18 Pengujian validasi menambahkan pengeluaran

Nama Kasus Uji	Kasus uji menambahkan pengeluaran
Kode Kebutuhan	SRS_1_16
Prosedur	<ol style="list-style-type: none"> Pemilik memilih menu pengeluaran Pemilik memilih <i>icon</i> menambahkan pengeluaran Pemilik dapat menambahkan pengeluaran dengan mengisi <i>field</i> yang tersedia Pemilik menekan tombol simpan
Hasil yang diharapkan	Pemilik berhasil menambahkan biaya pengeluaran operasional rumah indekos
Hasil	Pemilik berhasil menambahkan biaya pengeluaran operasional rumah indekos
Status	Valid

Tabel 6.18 Pengujian validasi menambahkan pengeluaran

Nama Kasus Uji	Kasus uji menambahkan pengeluaran
Kode Kebutuhan	SRS_1_16
Prosedur	<ol style="list-style-type: none"> Pemilik memilih menu pengeluaran Pemilik memilih <i>icon</i> menambahkan pengeluaran Pemilik dapat menambahkan pengeluaran dengan mengisi <i>field</i> yang tersedia Pemilik menekan tombol simpan
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi "Please fill out this fill"
Hasil	Pemilik berhasil menambahkan biaya pengeluaran operasional rumah indekos

Tabel 6.18 Pengujian validasi menambahkan pengeluaran (lanjutan)

Status	Valid
---------------	-------

Tabel 6.19 Pengujian validasi mengubah pemesanan

Nama Kasus Uji	Kasus uji mengubah pemesanan
Kode Kebutuhan	SRS_1_17
Prosedur	<ol style="list-style-type: none"> Pemilik memilih menu <i>list</i> pemesanan Mengisi <i>form</i> yang telah disediakan di dalam sistem Pemilik menekan tombol <i>submit</i>
Hasil yang diharapkan	Pemilik berhasil mengubah data pemesanan
Hasil	Pemilik berhasil mengubah data pemesanan
Status	Valid

Tabel 6.19 Pengujian validasi mengubah pemesanan

Nama Kasus Uji	Kasus uji mengubah pemesanan
Kode Kebutuhan	SRS_1_17
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu <i>list</i> pemesanan 2. Mengisi <i>form</i> yang telah disediakan di dalam sistem 3. Pemilik menekan tombol <i>submit</i>
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi " <i>Please fill out this fill</i> "
Hasil	Pemilik berhasil mengubah data pemesanan
Status	Valid

Tabel 6.20 Pengujian validasi verifikasi kendala

Nama Kasus Uji	Kasus uji verifikasi kendala
Kode Kebutuhan	SRS_1_18
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu <i>list</i> kendala 2. Pemilik memilih <i>icon</i> verifikasi kendala 3. Mengganti status kendala 4. Menekan tombol simpan
Hasil yang diharapkan	Pemilik berhasil memverifikasi laporan kendala dari penghuni
Hasil	Pemilik berhasil memverifikasi laporan kendala dari penghuni
Status	Valid

Tabel 6.21 Pengujian validasi mencetak laporan bukti pembayaran

Nama Kasus Uji	Kasus uji mencetak laporan bukti pembayaran
Kode Kebutuhan	SRS_1_09
Prosedur	<ol style="list-style-type: none"> 1. Pemilik memilih menu laporan 2. Memilih menu cetak laporan pembayaran 3. Sistem akan menampilkan laporan yang akan dicetak
Hasil yang diharapkan	Pemilik berhasil mencetak laporan pembayaran yang berada di sistem
Hasil	Pemilik berhasil mencetak laporan pembayaran yang berada di sistem
Status	Valid

Tabel 6.22 Pengujian validasi melihat status pembayaran

Nama Kasus Uji	Kasus uji melihat status pembayaran
Kode Kebutuhan	SRS_1_10
Prosedur	Penghuni memilih menu lihat status pembayaran
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
Hasil	Penghuni berhasil melihat data status pembayaran
Status	Valid

Tabel 6.23 Pengujian validasi melihat data pembayaran

Nama Kasus Uji	Kasus uji melihat data pembayaran
Kode Kebutuhan	SRS_1_11
Prosedur	1. Penghuni memilih <i>menu</i> lihat data pembayaran
Hasil yang diharapkan	Penghuni berhasil melihat data pembayaran
Hasil	Penghuni berhasil melihat data pembayaran
Status	Valid

Tabel 6.24 Pengujian validasi melaporkan kendala

Nama Kasus Uji	Kasus uji melaporkan kendala
Kode Kebutuhan	SRS_1_12
Prosedur	1. Penghuni memilih <i>menu</i> melaporkan kerusakan 2. Mengisi <i>form</i> yang telah disediakan di dalam system 3. Penghuni menekan tombol <i>submit</i>

Tabel 6.24 Pengujian validasi melaporkan kendala (lanjutan)

Hasil yang diharapkan	Penghuni berhasil melihat data pembayaran pada bulan yang diinginkan
Hasil	Penghuni berhasil melaporkan kerusakan kepada pemilik indekos
Status	Valid

Tabel 6.24 Pengujian validasi melaporkan kerusakan

Nama Kasus Uji	Kasus uji menambahkan mengubah penghuni
Kode Kebutuhan	SRS_1_12
Prosedur	1. Penghuni memilih <i>menu</i> melaporkan kerusakan 2. Mengisi <i>form</i> yang telah disediakan di dalam

	<p>system</p> <ol style="list-style-type: none"> 3. Penghuni menekan tombol <i>submit</i> Data penghuni baru telah diperbarui 4. Penghuni memilih menu lihat status pembayaran
Hasil yang diharapkan	Pemilik tidak mengisi <i>form</i> sesuai atau yang ada maka sistem menampilkan pemberitahuan agar mengisi <i>form</i> yang belum terisi
Hasil	Penghuni berhasil melihat data status pembayaran
Status	Valid

Tabel 6.25 Pengujian validasi keluar dari sistem

Nama Kasus Uji	Kasus uji keluar dari sistem
Kode Kebutuhan	SRS_1_13
Prosedur	<ol style="list-style-type: none"> 1. Pemilik dan penghuni memilih menu <i>logout</i> 2. Sistem menghapus <i>Session</i> 3. Pemilik dan penghuni keluar dari login <i>session</i> sistem
Hasil yang diharapkan	Pemilik dan penghuni berhasil keluar dari sistem
Hasil	Pemilik dan penghuni berhasil keluar dari sistem
Status	Valid

6.2.2 Pengujian Validasi Kebutuhan Non-Fungsional

Tabel 6.26 Pengujian validasi kompatibilitas *internet explorer*

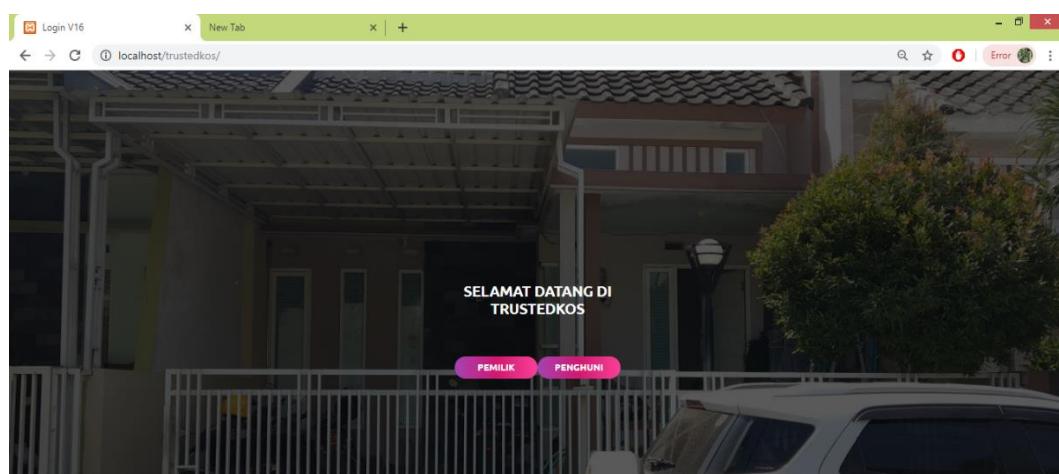
Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	<ol style="list-style-type: none"> 1. Pemilik dan penghuni membuka pada <i>web browser</i> "<i>internet explorer</i>"
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos
Status	Valid



Gambar 6.4 Hasil dari pengujian validasi kompatibilitas “Internet Explorer”

Tabel 6.27 Pengujian validasi kompatibilitas google chrome

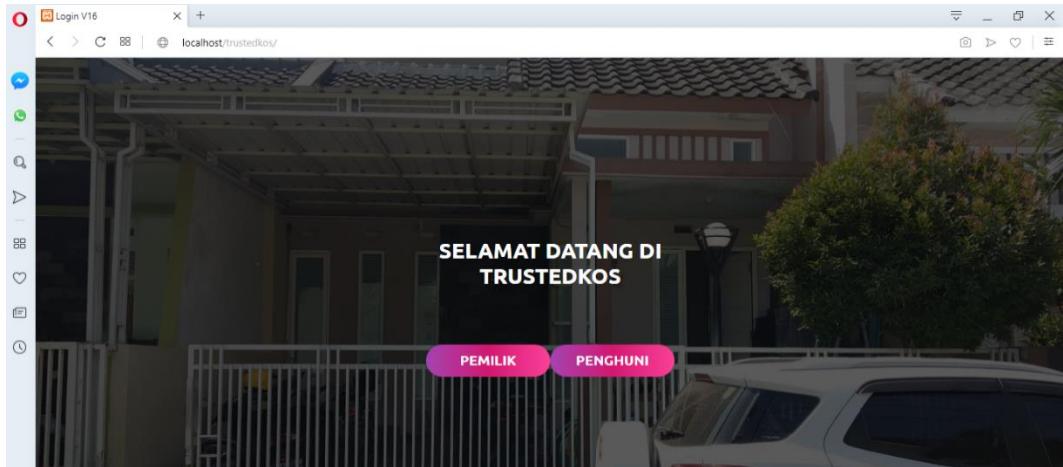
Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	1. Pemilik dan penghuni membuka pada web browser “Google Chrome”
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos
Status	Valid



Gambar 6.5 Hasil dari pengujian validasi kompatibilitas “Google Chrome”

Tabel 6.28 Pengujian validasi kompatibilitas opera

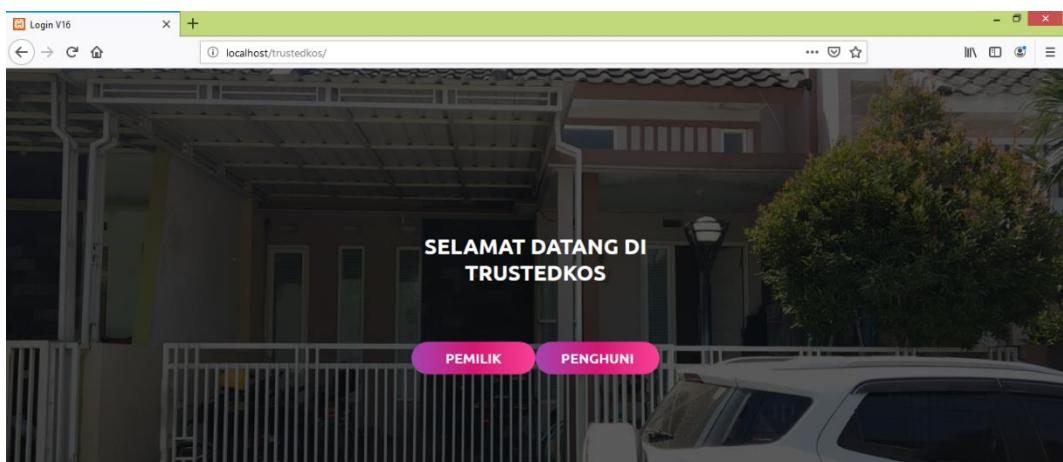
Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	1. Pemilik dan penghuni membuka pada web browser “Opera”
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos
Status	Valid



Gambar 6.6 Hasil dari pengujian validasi kompatibilitas “Opera”

Tabel 6.29 Pengujian validasi kompatibilitas *mozilla firefox*

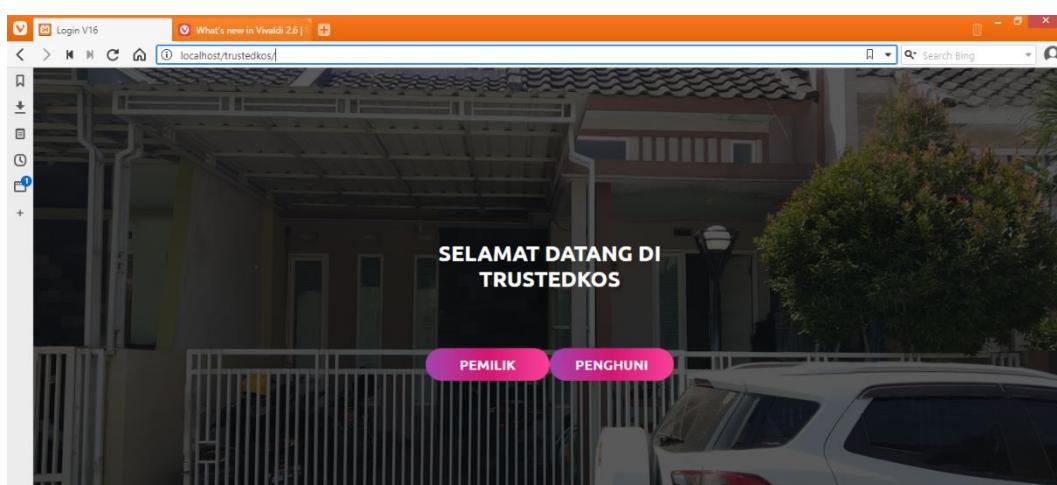
Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	1. Pemilik dan penghuni membuka pada <i>web browser “Mozilla Firefox”</i>
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos
Status	Valid



Gambar 6.7 Hasil dari pengujian validasi kompatibilitas “Mozilla Firefox”

Tabel 6.30 Pengujian validasi kompatibilitas vivaldi

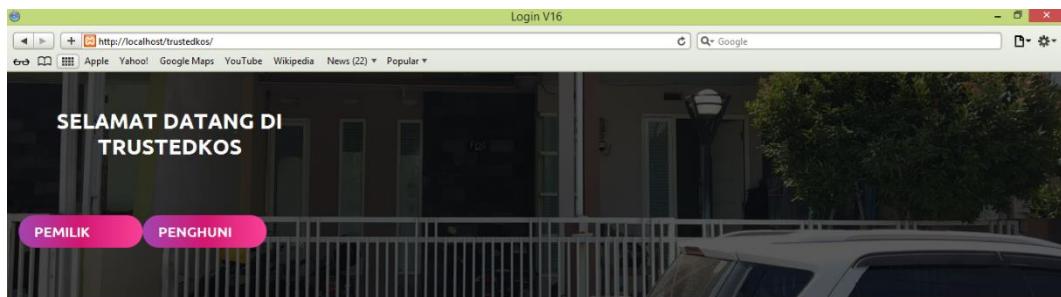
Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	1. Pemilik dan penghuni membuka pada <i>web browser</i> “Vivaldi”
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos
Status	Valid



Gambar 6.8 Hasil dari pengujian validasi kompatibilitas “Vivaldi”

Tabel 6.31 Pengujian validasi kompatibilitas safari

Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	1. Pemilik dan penghuni membuka pada <i>web browser</i> “Safari”
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos
Status	Valid



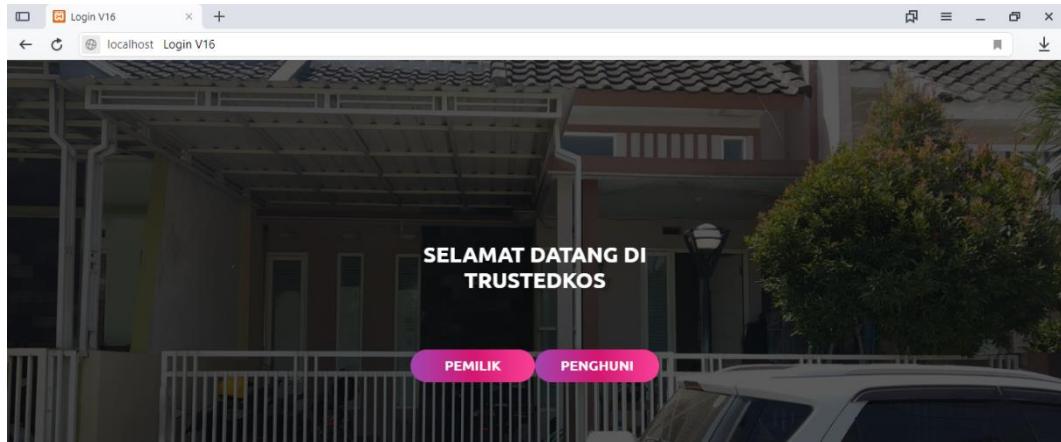
Gambar 6.9 Hasil dari pengujian validasi kompatibilitas “Safari”

Tabel 6.32 Pengujian validasi kompatibilitas yandex

Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	1. Pemilik dan penghuni membuka pada <i>web browser “Yandex”</i>
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos

Tabel 6.26 Pengujian validasi (lanjutan)

Status	Valid
---------------	-------

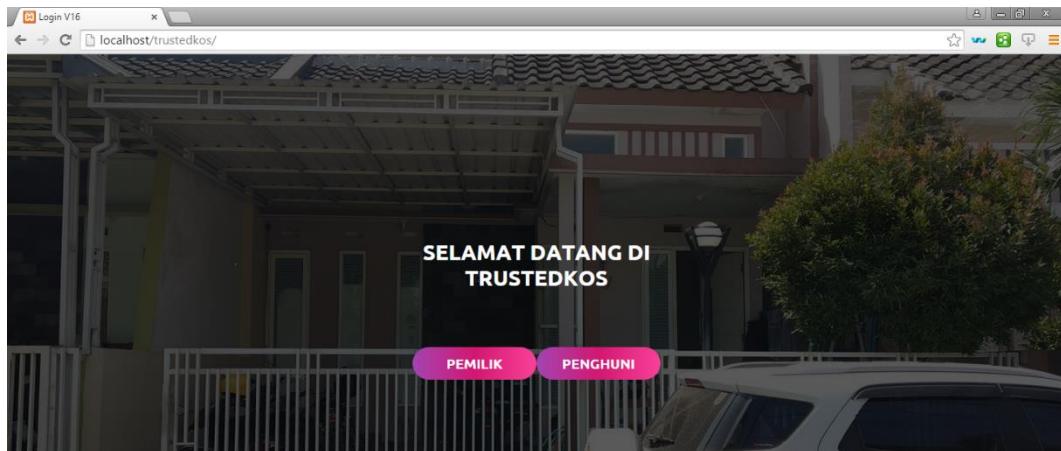


Gambar 6.10 Hasil dari pengujian validasi kompatibilitas “Yandex”

Tabel 6.33 Pengujian validasi kompatibilitas comodo ice dragon

Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	1. Pemilik dan penghuni membuka pada <i>web browser “Comodo Ice Dragon”</i>
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos

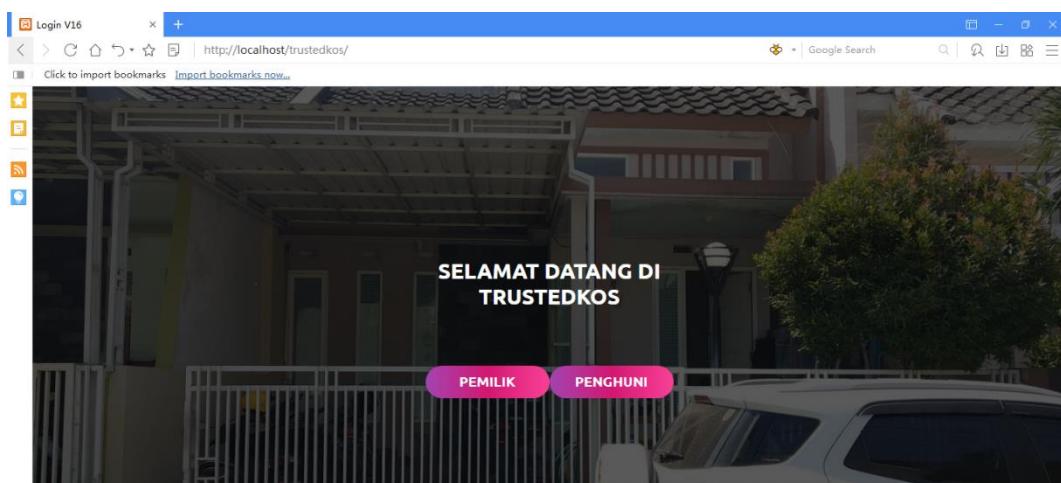
Status	Valid
--------	-------



Gambar 6.11 Hasil dari pengujian validasi kompatibilitas “Comodo Ice Dragon”

Tabel 6.34 Pengujian validasi kompatibilitas *maxthon*

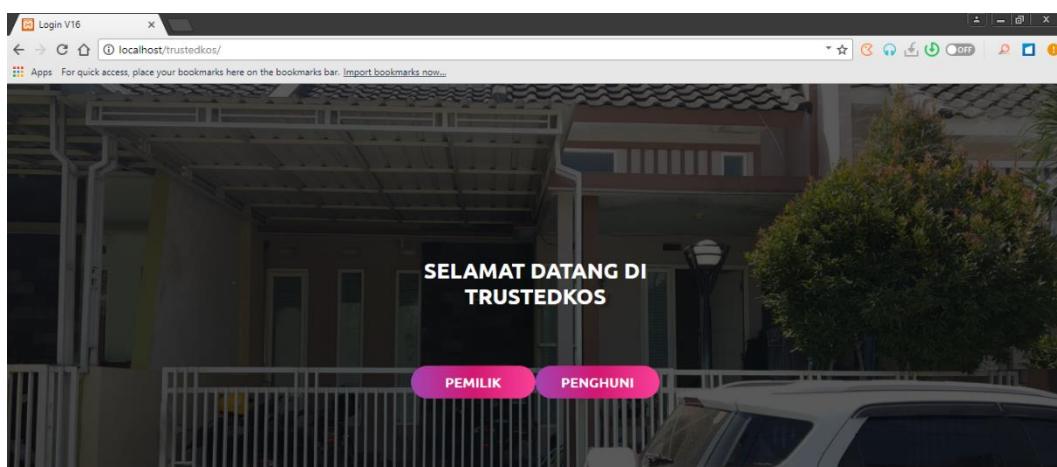
Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	2. Pemilik dan penghuni membuka pada web browser “Maxthon”
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos
Status	Valid



Gambar 6.12 Hasil dari pengujian validasi kompatibilitas “Maxthon”

Tabel 6.35 Pengujian validasi kompatibilitas *torch*

Nama Kasus Uji	Kasus uji kompatibilitas
Prosedur	1. Pemilik dan penghuni membuka pada <i>web browser</i> “Torch”
Hasil yang diharapkan	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos dan tidak ada kesalahan pada tampilan serta data informasinya
Hasil	Sistem akan tetap menampilkan data informasi sistem manajemen rumah indekos
Status	Valid



Gambar 6.13 Hasil dari pengujian validasi kompatibilitas “Torch”

6.3 Analisis

Berdasarkan dari hasil pengujian validasi yang sudah dilakukan, dapat disimpulkan bahwa seluruh kasus uji yang telah didefinisikan telah valid. Hal tersebut menggambarkan bahwa seluruh spesifikasi kebutuhan perangkat lunak telah terpenuhi. Dengan demikian, sistem manajemen rumah indekos telah dinyatakan lolos uji validasi.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan dari hasil penelitian skripsi “Sistem Pengelolaan Manajemen Rumah Indekos Pada Indekos Semanggi di Kota Malang” yang telah dilakukan dapat diambil kesimpulan sebagai berikut.

1. Berdasarkan dari hasil analisis kebutuhan yang sudah didefinisikan diawal, sistem pengelolaan manajemen rumah indekos memiliki 20 kebutuhan fungsional dan memiliki 1 kebutuhan non fungsional. Pada tahapan analisis kebutuhan sistem dimodelkan dengan menggunakan *use case* diagram, pemodelan *use case* diagram menghasilkan 20 *use case*. Masing-masing dari *use case* tersebut dijelaskan dengan secara detail menggunakan *use case scenario*. Setelah mendapatkan hasil analisis kebutuhan, maka dilanjutkan dengan perancangan *pseudocode* diambil dari tiga sampel yang digunakan untuk perancangan pada *sequence diagram*. Perancangan *sequence diagram* ini nantinya akan dijadikan pedoman untuk implementasi ke kode program. Pada perancangan antarmuka menghasilkan 22 *wireframe* yang dapat menggambarkan keseluruhan perancangan antarmuka pada sistem. Berdasarkan dari hasil implementasi yang telah dilakukan, mendapatkan spesifikasi perangkat lunak yang akan digunakan dalam melakukan implementasi sistem. Dalam proses untuk implementasi, melakukan implementasi basis data dari perancangan basis data. Selain itu juga dihasilkan implementasi kode program yang didapatkan melalui perancangan algoritma. Implementasi antar muka juga didapatkan melalui perancangan antarmuka.
2. Berdasarkan dari hasil pengujian *white-box testing* yang telah dilakukan, dalam pengujian *white-box* dilakukan pada pengujian unit menggunakan tiga sampel *pseudocode* dan menghasilkan nilai valid pada semua kasus uji. Pada saat pengujian *black-box testing* menghasilkan nilai yang valid untuk setiap masing-masing fungsionalitas. Pada saat pengujian non fungsionalitas *compatibility* menggunakan sepuluh *web browser* yang berbeda menghasilkan nilai yang valid. Hal tersebut dibuktikan bahwa sistem ini dapat digunakan untuk pengelolaan manajemen rumah indekos.

7.2 Saran

Adapun saran yang digunakan untuk penelitian selanjutnya yaitu sebagai berikut.

1. Memberikan akses kebebasan kepada pengguna untuk mendaftarkan dirinya ke dalam sistem dengan fitur *sign up* penghuni. Kebebasan ini memiliki tujuan agar program tidak bergantung kepada pemilik indekos
2. Sistem dapat dikembangkan dengan penambahan fitur pada hak akses penghuni. Fitur tersebut untuk *upload* bukti pembayaran ketika penghuni

melakukan pembayaran dengan menggunakan metode pembayaran transfer.

LAMPIRAN A

A.1 Hasil Wawancara

Berikut ini merupakan hasil wawancara yang sudah dilakukan dengan pemilik rumah indekos sebagai berikut.

Narasumber : Bapak Sunaryono

Jabatan : Pemilik rumah indekos semanggi

No.	Wawancara
1.	<p>Bagaimana perkembangan rumah indekos di kota Malang?</p> <p>Jawaban :</p> <p>Perkembangan untuk sampai sekarang sangat pesat mas. Dulunya saat di sini tidak mempunyai tetangga, bisa diitung jumlah orangnya. Namun sekarang sangat padat.</p>
2.	<p>Faktor apa yang membuat orang mulai usaha rumah indekos?</p> <p>Jawaban :</p> <p>Menurut saya, paling utama karena digunakan untuk investasi yang memiliki keuntungan lumayan besar dan memiliki kerugian yang sangat minim.</p>
3.	<p>Berapakah jumlah rumah indekos yang bapak memiliki?</p> <p>Ada dua mas, di daerah semanggi sama daerah griya santha. Keduanya memiliki jumlah kamar yang sama yaitu tiap rumah indekos berjumlah sepuluh kamar.</p>
4.	<p>Apakah di daerah ini semua menggunakan aplikasi untuk menajemen rumah indekos?</p> <p>Jawaban :</p> <p>Selama yang saya ketahui belum ada mas, masih menggunakan cara manual dengan menggunakan Microsoft excel.</p>

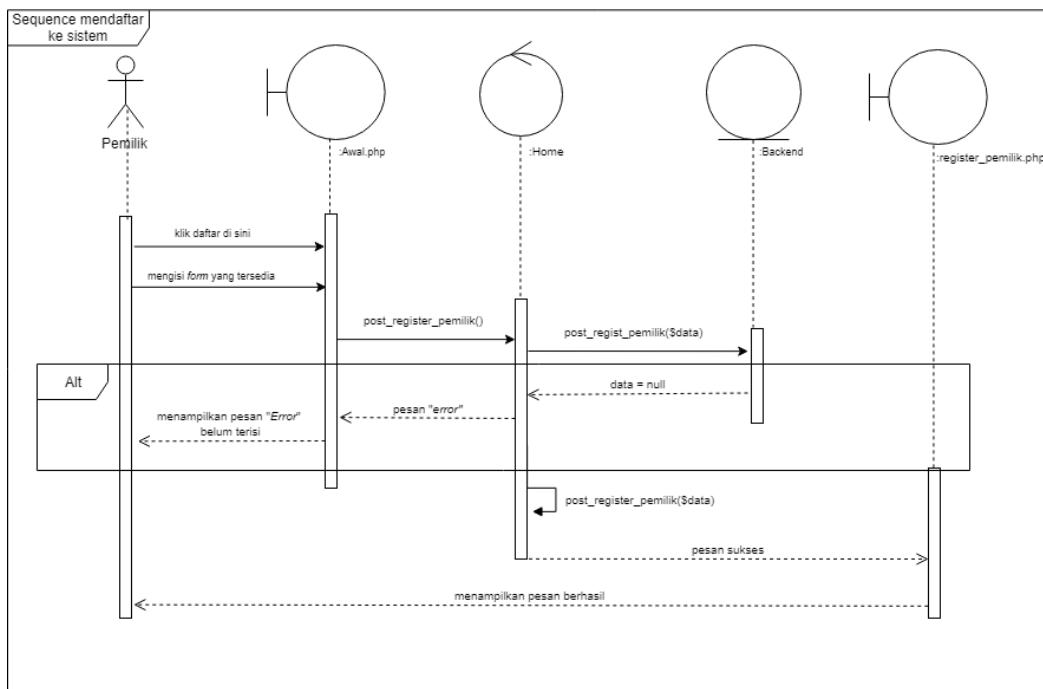
5.	<p>Apakah pada saat melakukan pencatatan secara manual mengalami kendala?</p> <p>Jawaban :</p> <p>Kendala yang selama ini saya alami adalah melakukan pencarian data atau pembaruan data membutuhkan waktu yang lama. Ketika melakukan pembaruan data sering mengalami kesalahan penulisan nama, tanggal masuk dan nomor kamar yang di tempati terjadi pada posisi kolom dan baris yang salah. Masalah yang lain ketika mau dipindahkan ke <i>flashdisk</i> untuk di backup, data yang telah dipindahkan mengalami <i>corrupt</i> dan tidak bisa diakses. Permasalahan yang lain sering terlambat bayar tiap bulannya.</p>
6.	<p>Apakah setiap habis melakukan pembayaran terdapat tanda bukti sudah membayar?</p> <p>Jawaban :</p> <p>Tidak ada mas, saya cuma mencatatnya saja di buku catatan.</p>
7.	<p>Bagaimana menurut bapak apabila dilakukan pembangunan aplikasi manajemen rumah indekos?</p> <p>Jawaban :</p> <p>Saya kira aplikasi tersebut bisa membantu saya.</p>
8.	<p>Apa yang bapak harapkan dari aplikasi manajemen rumah indekos ini?</p> <p>Jawaban :</p> <p>Semoga aplikasi ini dapat membantu saya untuk mempermudah manajemen indekos serta transparansi pembayaran.</p>

Malang,

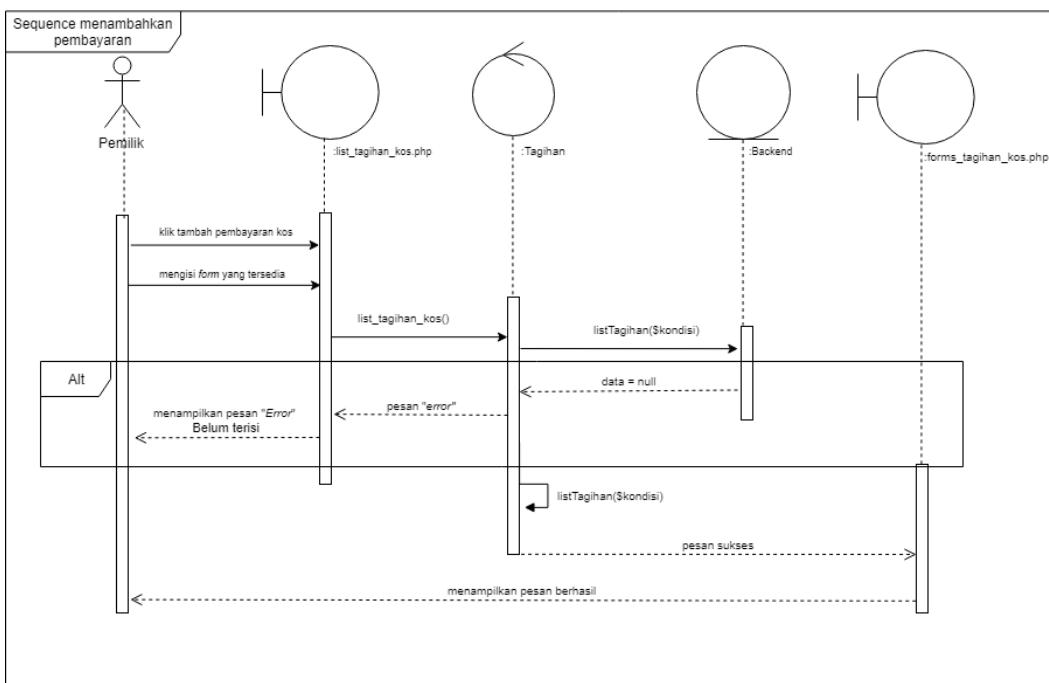
LAMPIRAN B

B.1 Sequence Diagram

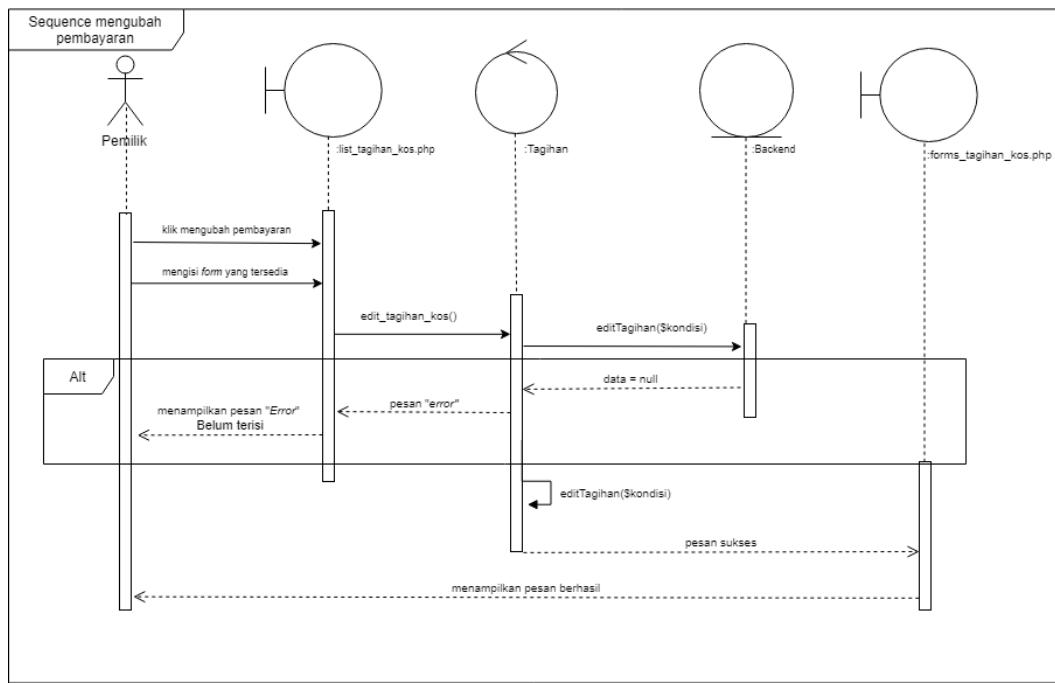
1. Sequence Diagram Mendaftar ke Sistem



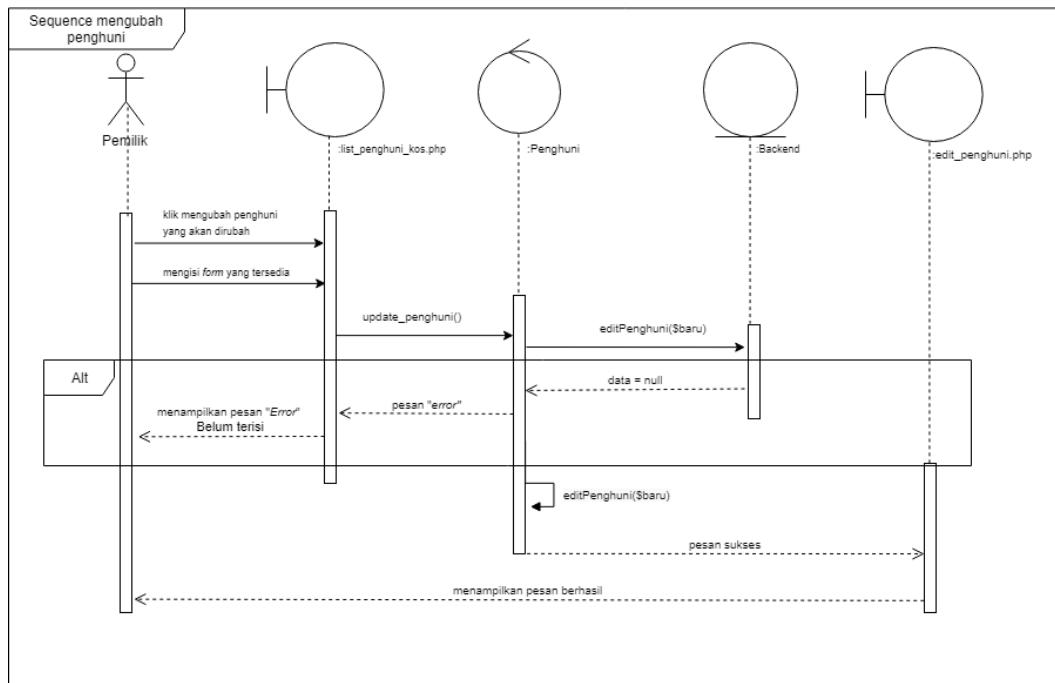
2. Sequence Diagram Tambah Pembayaran



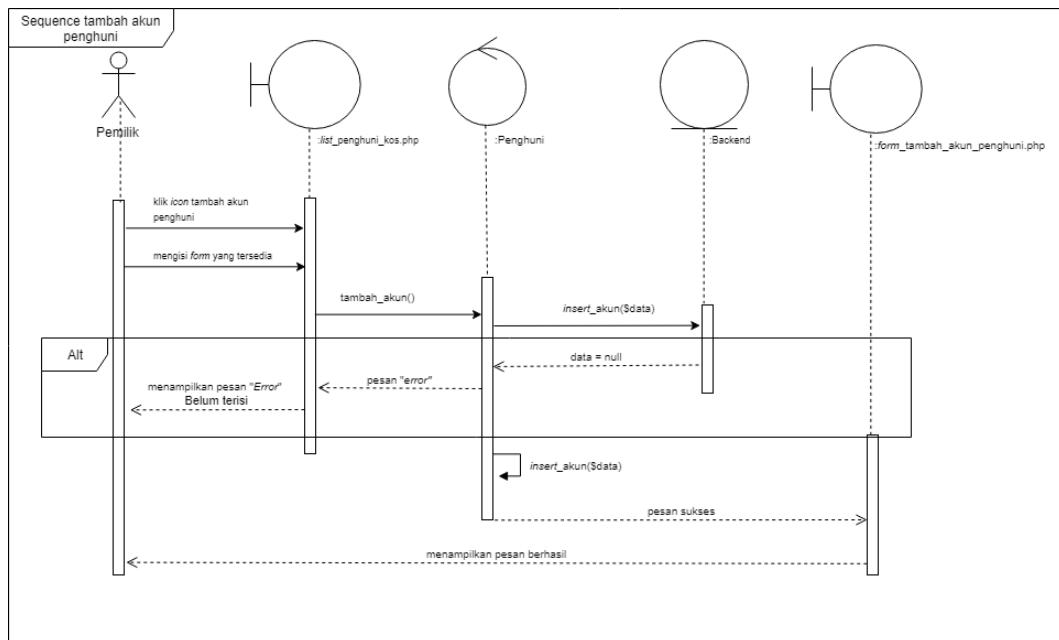
3. Sequence Diagram Mengubah Pembayaran



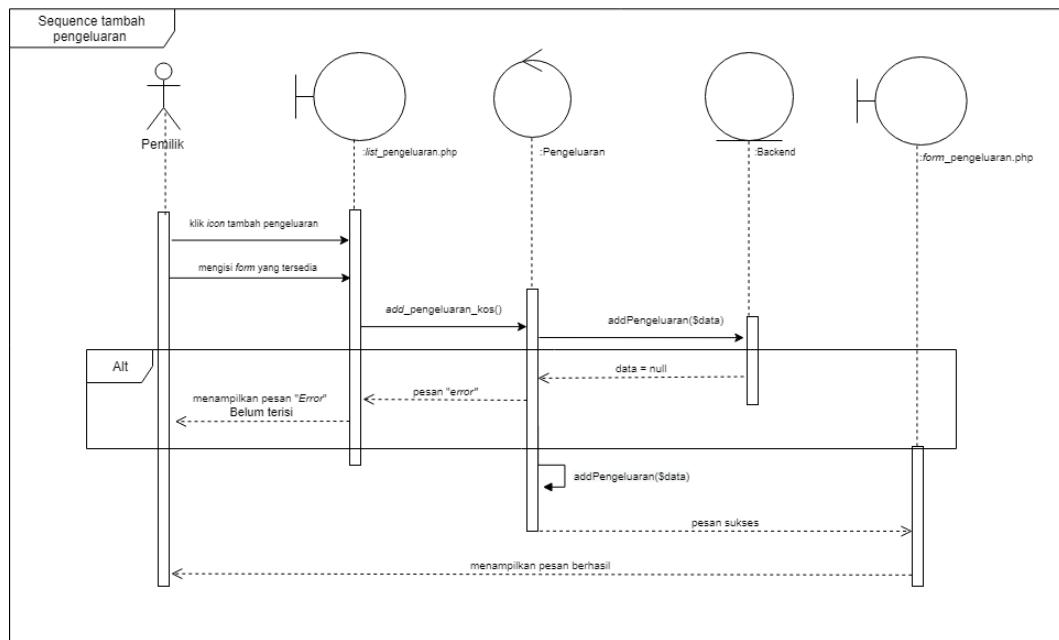
4. Sequence Diagram Mengubah Penghuni



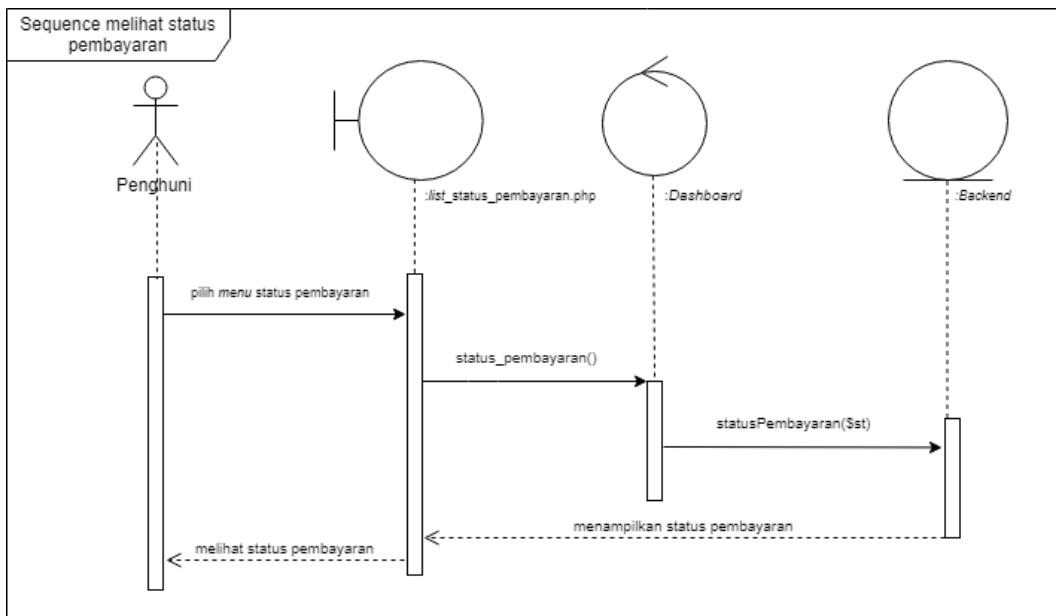
5. Sequence Diagram Tambah Akun Penghuni



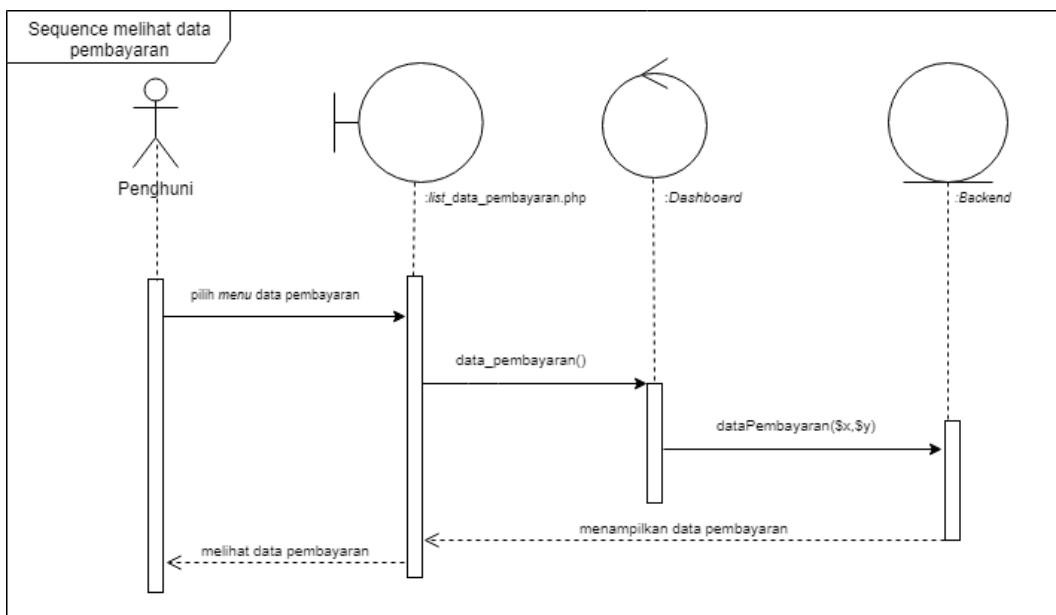
6. Sequence Diagram Tambah Pengeluaran



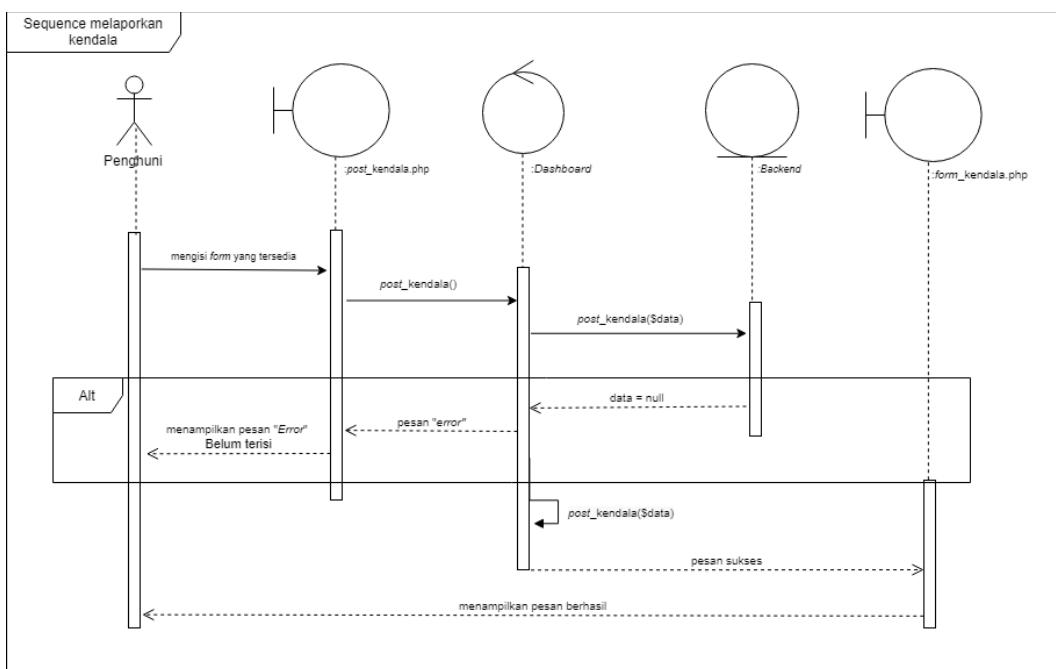
7. Sequence Diagram Melihat Status Pembayaran



8. Sequence Diagram Melihat Data Pembayaran



9. Sequence Diagram Melaporkan Kendala



DAFTAR REFERENSI

- Adelheid, Andre, dan Khairil. 2012. *Buku Pintar Mengusai PHP dan MySQL*. Jakarta: Mediakita.
- Fahrurrozi, imam dan Azhari. *Proses Pemodelan Software dengan Metode Waterfall dan Extreme Programming*. Yogyakarta: Vol. IV No. 1
- Jamwal, D., 2010. Analysis of Software Quality Models for Organizations. International Journal of Latest Trends in Computing, 1(2), pp. 19-23.
- Kementerian Pekerjaan Umum dan Perumahan Rakyat Republik Indonesia. 2012. *Buku Panduan Penanganan lingkungan perumahan dan permukiman kumuh berbasis kawasan TA 2013*. Jakarta: Deputi Pengembangan Kawasan.
- Laplante, P. A. 2007. What Every Engineer Should Know About. New York: CRC Press.
- Myers, Glen. 2004. *The Art of Software Testing*, Second Edition. New York: Wiley.
- Marchewka, J. T., 2015 *INFORMATION TECHNOLOGY PROJECT MANAGEMENT*. 5th ed. Hoboken: John Wiley.
- Nidhra, S. dan Jagruthi D. 2012. Black box and White box TestingmTechniques – A Literature Review. Internasional Journal of Embedded System and Applications (IJESA) Vol.2, No.2.
- Novari, Satria dan Haris Saputro. 2017. Sistem Informasi Administrasi Pembayaran Kost Hj. Murni Baturaja Menggunakan EmbaracaderoXE2. Baturaja: JIK Vol. 8, No. 1.
- Panudju, Bambang. 2006. *Pengadaan Perumahan Kota Dengan Perang Serta Masyarakat Berpenghasilan Rendah*. Bandung: Penerbit Alumni.
- Perda Malang. 2015. *Peraturan Daerah Kota Malang Nomor 16 Tahun 2010 Tentang Pajak Daerah*. [pdf] Tersedia di: < https://hukum.malangkota.go.id/download/perda/peraturan_daerah_tahun_2015/SALINAN%20PERDA%20NOMOR%202%20TAHUN%202015%20TENTANG%20PERUBAHAN%20PERDA%20NOMOR%2016%20TAHUN%202010%20ENTANG%20PAJAK%20DAERAH.pdf > [Diakses 02 Desember 2018].
- Pressman, R. S. 2010. Software Engineering : A practitioner's Approach. Seventh Edition. McGraw Hill.
- Pusat Bahasa Kementrian Pendidikan Nasional. 2005. *Kamus Besar Bahasa Indonesia*. Jakarta: Pusat Bahasa Kementrian Pendidikan Nasional.
- Setyawan, Ferry dan Sukadi. 2014. *Sistem Pembayaran Biaya Kost Terkomputerisasi pada Rumah Kost Griya Ampel Muda Kabupaten Pacitan*. Pacitan: Indonesian Journal on Networking and Security.

- Soesabdo, M. S. 1981. *Peraturan Perundangan Perumahan*. Jakarta: Bina Aksara.
- Sommerville, I., 2003. Software Engineering. America: Pearson.
- Sukmadinata, N. S. 2007. *Metode Penelitian Pendidikan*. Bandung: Remaja Rosda Karya.
- Usman, H. 2006. *Manajemen Teori, Praktik dan Riset Pendidikan*. Jakarta : PT Bumi Aksara.
- Williams, L. 2006. *Testing Overview and Black-Box Testing Techniques*. [pdf] Tersedia di : <<https://students.cs.byu.edu/~cs340ta/winter2018/readings/BlackBox.pdf>> [Diakses 02 Desember 2018].