

BAB III METODE PENELITIAN

3.1 Analisis Permasalahan

Indekos yang biasa disebut kos atau tinggal di rumah orang lain dengan atau tanpa makan dengan membayar setiap bulan (Departemen Pendidikan Nasional, 2008) banyak berdiri di perkotaan khususnya di daerah yang memiliki banyak perguruan tinggi. Sasaran pelanggan dari indekos ini adalah mahasiswa yang tempat tinggalnya jauh ataupun di luar kota dan memerlukan sebuah penginapan sementara atau indekos yang dekat dengan perguruan tinggi tempatnya menuntut ilmu.

Banyaknya indekos yang ada akan semakin menyulitkan mahasiswa untuk memilih indekos. Pencari indekos ini pastinya memiliki kriteria-kriteria tertentu dalam memilih suatu indekos yaitu dari segi jarak, wilayah daerah, fasilitas yang ada, harga dan universitas sebagai tempat belajarnya. Tiap pencari indekos pasti memiliki kriteria yang berbeda, begitu juga dengan para pemilik indekos tersebut pasti memiliki penawaran yang berbeda di setiap indekos yang ditawarkan. Selain itu juga belum ada suatu aplikasi yang menyediakan informasi indekos tersebut dalam sebuah peta digital yang mudah dimengerti.

Berdasarkan uraian di atas, dapat disimpulkan bahwa untuk mempermudah dan membantu para pencari indekos dalam proses pencarian lokasi dan rute menuju lokasi tersebut diperlukan suatu sistem aplikasi pencari dan pemberi informasi indekos yang dapat memberikan informasi lengkap tentang kamar yang disewakan dan memberikan navigasi menuju lokasi indekos yang

dipilih. Dengan adanya aplikasi ini diharapkan dapat membantu para pencari indekos agar dapat mengetahui gambaran informasi mengenai indekos yang akan dipilih dan tidak tersesat saat proses pencarian indekos tersebut. Dari sisi pemilik indekos sebagai sumber data, aplikasi ini juga berguna sebagai ajang promosi indekos-indekosa yang dimilikinya.

Untuk pencari indekos yang masih baru berpindah, sebelum datang ke kota besar tujuan belajarnya pasti membutuhkan suatu sistem pencari dan pemberi informasi indekos yang bisa diakses di mana saja dan kapan saja. Oleh karena itu pada aplikasi pencari ini akan disediakan dalam 2 basis pemrograman yaitu berbasis *web* dan *mobile Android*. Aplikasi berbasis *web* dapat memberikan informasi dasar serta peta *digital* lokasi indekos sehingga para pencari indekos bisa melakukan analisa awal mengenai indekos yang akan dipilihnya yang dapat diakses di mana pun dan kapan pun. Aplikasi *web* juga digunakan para pemilik indekos melakukan pendaftaran gratis sebagai anggota di *web* tersebut sehingga dapat menyimpan data indekos yang dimilikinya. Sedangkan aplikasi yang berbasis *mobile Android*, terdapat tambahan *fitur navigasi* yang memanfaatkan *GPS* pada *smartphone Android* sehingga dapat membantu para pencari indekos dalam menemukan indekos yang diinginkan berdasarkan rute yang tertera pada peta *digital*. Dipilih pemrograman *mobile Android* sebagai bahasa ke-dua karena pada *Smartphone Android* telah tertanam fitur *GPS* dan dapat menerapkan fungsi yang berjalan pada *background process*, sehingga dapat selalu terhubung dengan *web server* untuk melakukan *update* lokasi secara berkala dan otomatis. Diharapkan dengan adanya sistem pencari indekos dan navigasi ini dapat

membantu

cara

memiliki sehingga

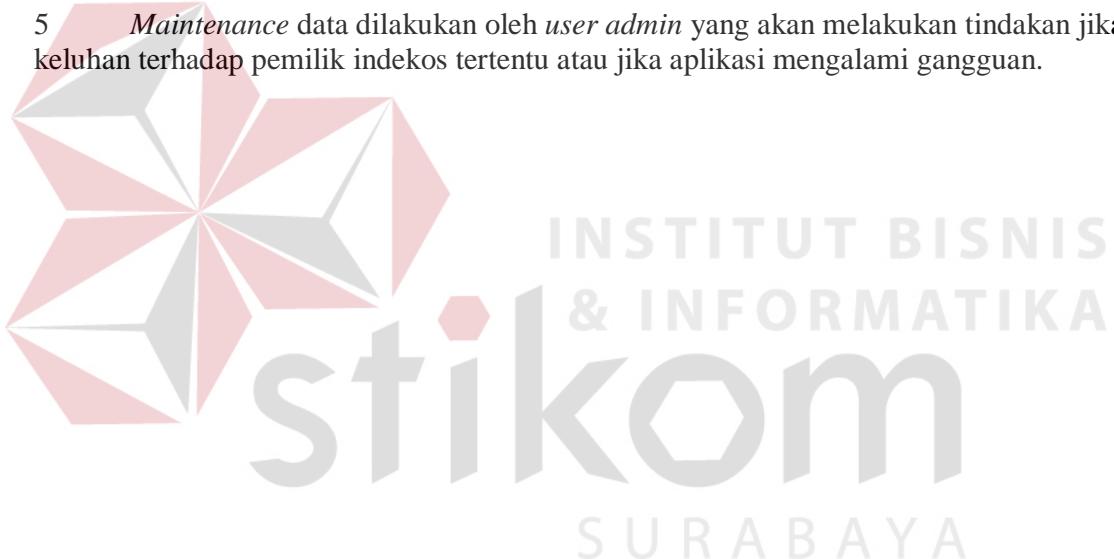
yaitu digunakan pengakses



Gambar 3.1 Gambaran Umum Aplikasi Sistem Pencari dan Navigasi Lokasi Indekos di Surabaya

Penjelasan Gambaran umum sistem tersebut adalah sebagai berikut :

- 1 *User* pemilik indekos dapat melakukan Registrasi hanya melalui *website* dengan format yang telah diatur, lalu untuk verifikasi akan dikirimkan kode verifikasi melalui pos ke alamat yang telah diisikan sebelumnya.
- 2 Setelah melakukan Registrasi, *user* pemilik indekos melakukan *Login* sesuai dengan *username* saat Registrasi lalu *user* pemilik indekos baru dapat melakukan penyimpanan data indekos yang akan didaftarkan dalam aplikasi ini.
- 3 Sedangkan proses pencarian dimulai saat *user* pencari indekos memasukkan data indekos yang diinginkan, data tersebut digunakan sistem sebagai *filter* dalam proses pencarian indekos sesuai keinginan *user* tersebut.
- 4 Setelah itu aplikasi mengirimkan posisi indekos berupa data lokasi (titik *latitude* dan *longitude*) ke GPS lalu GPS melakukan pencarian dan mengirimkan hasil pencarian posisi tersebut kembali ke aplikasi sehingga dapat ditampilkan dalam bentuk peta *digital* ditambah data-data informasi yang tersimpan sesuai indekos yang dipilih tersebut. Data ini dikirim menggunakan media internet sehingga *smartphone* diharuskan memiliki koneksi internet.
- 5 *Maintenance* data dilakukan oleh *user admin* yang akan melakukan tindakan jika ada keluhan terhadap pemilik indekos tertentu atau jika aplikasi mengalami gangguan.



3.3 Analisis Sistem

3.3.1 Block Diagram

Block diagram menunjukkan gambaran proses-proses yang dimiliki oleh aplikasi secara keseluruhan. Secara umum Aplikasi Pencari dan Navigasi Lokasi Indekos di Surabaya ini memiliki 3 proses yaitu proses pencarian indekos dengan data pemilik, data indekos dan data kamar sebagai data *input* sementara detail indekos, lokasi indekos, detail pemilik dan detail kamar sebagai data *output*. Proses selanjutnya. Selanjutnya proses pemesanan, data indekos dan data kamar sebagai data input diproses sehingga menghasilkan data pemesanan. Yang terakhir proses navigasi rute lokasi indekos, proses ini mengolah data lokasi dan data indekos sehingga menghasilkan rute lokasi indekos. Untuk lebih jelasnya dapat dilihat pada

Gambar 3.2.



Gambar 3.2 Block Diagram Aplikasi Sistem Pencari dan Navigasi Lokasi Indekos di Surabaya

3.3.2 Use Case Diagram

Use case diagram menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada di luar sistem (*actor*). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar, dan mendetailkan apa yang dapat dilakukan oleh sistem atau untuk mendetailkan kebutuhan fungsional utama dari sistem. Berikut akan dijelaskan *use case diagram* untuk sistem secara menyeluruh:

Gambar 3.3 *Use Case Diagram* Aplikasi Sistem Pencari dan Navigasi Lokasi Indekos di Surabaya



Penjelasan singkat dari *Use Case* diatas dapat dilihat pada Table 3.1 di bawah ini :

Tabel 3.1 Penjelasan Singkat *Use Case*

| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos |
| Pesan Indekos | Proses yang digunakan untuk melakukan pemesanan kamar indekos dalam jangka waktu tertentu. |

A Nama Tabel : *User*

3.3.3 *Class Diagram*

Class Diagram digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar kelas tersebut (menunjukkan interaksi antar kelas di dalam aplikasi).

Class Diagram dapat dilihat pada Gambar 3.3 dan Gambar 3.4.

Gambar 3.4 *Class Diagram* Pada *Mobile Application*

1. *Class* *MenuUtama* Kelas *MenuUtama* digunakan sebagai *form* utama pada *mobile application*. Kelas ini mengoordinasikan beberapa operasi seperti inisialisasi data awal saat aplikasi dijalankan, penentuan tampilan awal aplikasi, dan lainnya.



| MenuUtama |
|-------------------------|
| +kota |
| +svalue |
| +chk |
| +OnCreate() |
| +findViewById() |
| +view.OnClickListener() |
| +view.getContent() |

Dengan kata lain
kelas ini digunakan

sebagai
penghubung

Gambar 3.5 *Class Main* pada *Mobile Application*

dengan kelas-kelas yang lain. Untuk lebih jelasnya dapat dilihat pada Gambar 3.5.

Gambar 3.5 *Class Main* pada *Mobile Application*

2. *Class* Peta Kelas Peta digunakan oleh *mobile application* untuk menampilkan peta jika pencari indekos memilih menu Peta pada *form* utama. Kelas ini mengoordinasikan perintah-perintah sehingga dapat menampilkan sebuah peta

GoogleMaps beserta titik-titik lokasi indekos yang tersimpan pada *database*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.6.

3. *Class* DataPosisiSaya Kelas DataPosisiSaya digunakan oleh *mobile application* untuk memberikan titik koordinat lokasi *user* saat membuka aplikasi ini. Kelas ini mendefinisikan semua atribut dan operasi yang berguna untuk melakukan pelacakan koordinat lokasi pencari indekos dan menampilkannya pada peta. Untuk lebih jelasnya dapat dilihat pada Gambar 3.7.

4. *Class* CariFilter Kelas CariFilter digunakan pada menu pencarian dengan menggunakan filter. Kelas ini berfungsi untuk menampung data filter dan melakukan proses pencarian sesuai dengan filter yang dimasukkan oleh pencari indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.8.

5. *Class* ListCariRumah Kelas ListCariRumah digunakan untuk menampilkan daftar rumah hasil pencarian lokasi indekos. Kelas ini melakukan koneksi dengan *database* pusat untuk memberikan rumah indekos yang sesuai dengan kriteria yang dimasukkan pencari indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.9.

6. *Class* ListRumah Kelas ListRumah berfungsi sebagai penghubung dengan *database*. kelas ini membantu proses penampilan daftar pencarian pada kelas ListCariRumah.. Untuk lebih jelasnya dapat dilihat pada Gambar 3.10.

7. *Class* DetailRumah Kelas DetailRumah digunakan untuk menampilkan detail rumah indekos yang dipilih. Detail rumah indekos menampilkan profil pemilik dan data rumah indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.11.

8. *Class* ListKamar Kelas ListKamar digunakan mengoordinasi proses penampilan daftar kamar pada suatu rumah indekos yang dipilih. Kelas ini berfungsi untuk

menghubungkan aplikasi dengan data kamar pada *database*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.12.

9. *Class* DetailKamar Kelas DetailKamar merupakan kelanjutan dari kelas ListKamar. Kelas ini

menampilkan informasi kamar yang dipilih sesuai yang tersimpan pada *database*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.13.

10. *Class DirectionMap* Kelas *DirectionMap* digunakan untuk menampilkan rute menuju lokasi indekos yang dipilih oleh *user*. Kelas ini akan bekerja jika pencari indekos memilih menu rute pada *form* detail rumah. Untuk lebih jelasnya dapat dilihat pada Gambar 3.14.
11. *Class DataTanda* Kelas ini digunakan untuk menyimpan dan memproses data indekos yang telah diberi tanda bintang oleh pencari indekos. Kelas ini membantu pencari indekos yang ingin melihat kembali rumah indekos yang dipilih dengan diberi tanda bintang pada halaman detail rumah. Untuk lebih jelasnya dapat dilihat pada Gambar 3.15.
12. *Class ListTanda* Kelas *ListTanda* digunakan menampilkan daftar rumah indekos yang



Gambar 3.6 *Class Peta* pada *Mobile Application*



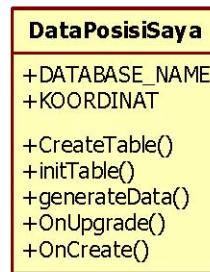
Gambar 3.7 *Class*

DataPosisiSaya
pada *Mobile*
Application

| Peta |
|------------------------|
| +GoogleMap |
| +JSONHelper |
| +ProgressDialog |
| +LatLng |
| +DataRumah |
| +SQLiteDatabase |
| +Cursor |
| +DataPosisiSaya |
| +KEY_ID_RUMAH_KOS |
| +KEY_LNG_ASAL |
| +KEY_LNG_TUJUAN |
| +KEY_LAT_ASAL |
| +KEY_LAT_TUJUAN |
| +OnCreate() |
| +SetContentView() |
| +getWritableDatabase() |
| +MoveToFirst() |
| +GenerateData() |
| +MoveToMyLocation() |
| +OnMyLocationChange() |

Gambar 3.6 *Class Peta* pada *Mobile Application*





Gambar 3.7 *Class DataPosisiSaya pada Mobile Application*

Gambar 3.8 *Class CariFilter pada Mobile Application*



Gambar 3.9

Class
ListCariRumah
pada *Mobile*
Application

| CariFilter |
|---------------------------|
| +id |
| +id_transaksi |
| +id_jadwal |
| +jenis_kelamin |
| +prev |
| +txtAlamat |
| +txtKelurahan |
| +RadioHarga |
| +radioHargaButton |
| +radioJK |
| +radioJKButton |
| +radioPSButton |
| +radioPeriode |
| +radioPeriodeButton |
| +radioKamar |
| +radioKamarButton |
| +OnCreate() |
| +getWritableDatabase() |
| +StartActivityForResult() |

Gambar 3.8 *Class CariFilter* pada *Mobile Application*

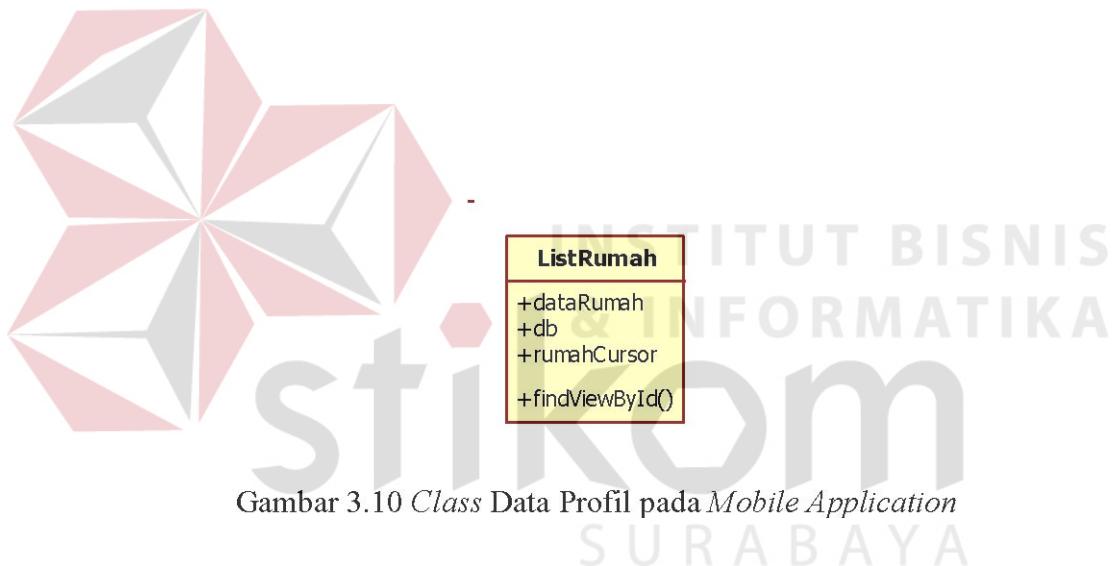




Gambar 3.9 *Class ListCariRumah pada Mobile Application*

Gambar 3.10 *Class Data Profil pada Mobile Application*

Gambar 3.11 *Class DetailRumah pada Mobile Application*



Gambar 3.10 *Class Data Profil pada Mobile Application*

| DetailRumah |
|-----------------|
| +mWebView |
| +id_rumah_kos |
| +tanda_adu |
| +menu |
| +rt |
| +btntanda |
| +datatanda |
| +dbtanda |
| +tandaCursor |
| +findViewById() |
| +rawQuery() |

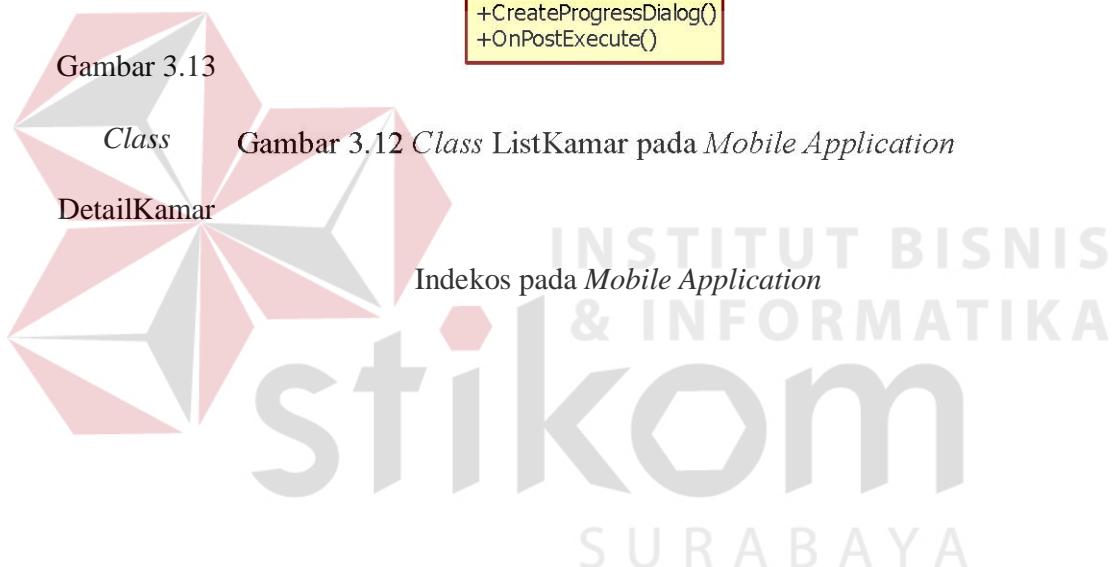
Gambar 3.11 *Class DetailRumah pada Mobile Application*

Gambar 3.12

| Class |
|--------------------|
| ListKamar pada |
| <i>Mobile</i> |
| <i>Application</i> |
| Gambar 3.13 |

| ListKamar |
|-------------------------|
| +dataKamar |
| +db |
| +kamarCursor |
| +srow |
| +id_rumah_kos |
| +menu |
| +CreateProgressDialog() |
| +OnPostExecute() |

Gambar 3.12 *Class ListKamar pada Mobile Application*



Gar

| DirectionMap |
|-----------------------|
| +map |
| +json |
| +pdialog |
| +listDirections |
| +id_rumah_kos |
| +menu |
| +koordinat |
| +alamat |
| +kelurahan |
| +datarumah |
| +dbrumah |
| +rumahCursor |
| +dataPosisiSaya |
| +posisiSayaCursor |
| +bundle.getString() |
| +rawQuery() |
| +setupMap() |
| +moveToMyLocation() |
| +gambarDirections() |
| +onMyLocationChange() |

Gambar 3.14 *Class DirectionMap* pada *Mobile Application*

Gambar 3.14 *Class DirectionMap* pada *Mobile Application*

Gambar 3.15 *Class DataTanda* pada *Mobile Application*



sebelumnya

telah ditandai

oleh *user*

pencari

Gambar 3.15 *Class DataTanda pada Mobile Application*

indekos. Kelas

ListTanda merupakan kelanjutan dari kelas DataTanda. Untuk lebih jelasnya dapat dilihat pada Gambar

3.16.

Gambar 3.16 *Class ListTanda pada Mobile Application*

| DataTanda |
|----------------|
| +DATABASE_NAME |
| +RUMAH_TANDA |
| +DataTable() |
| +CreateTable() |
| +InitTable() |
| +generate() |
| +delete_byID() |
| +OnUpgrade() |
| +OnCreate() |



13. *Class*
Petunjuk

Kelas
Petunjuk
digunakan

| ListTanda | |
|------------------------|--|
| +DataRumah | |
| +SQLiteDatabase | |
| +Cursor | |
| +DataTanda | |
| +OnCreate() | |
| +SetContentView() | |
| +FindViewById() | |
| +GetWritableDatabase() | |
| +InitTable() | |
| +Intent() | |
| +jobj.getString() | |
| +OnPostExecute() | |
| +OnProgressUpdate() | |

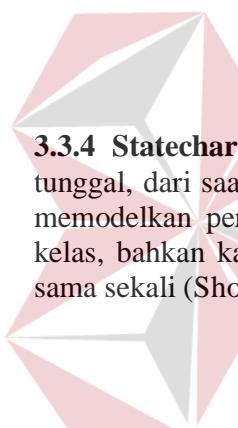
menampilkan

Gambar 3.16 *Class ListTanda* pada *Mobile Application file* petunjuk

penggunaan sistem informasi pencari indekos ini. Untuk lebih jelasnya dapat dilihat pada

Gambar

3.17.



Gambar 3.17 *Class Petunjuk* pada *Mobile Application*

3.3.4 Statechart Diagram *Statechart Diagram* menunjukkan siklus hidup sebuah obyek tunggal, dari saat dibuat sampai obyek tersebut dihapus. Diagram ini adalah cara tepat untuk memodelkan perilaku dinamis sebuah kelas. Diagram *statechart* tidak dibuat untuk setiap kelas, bahkan kadang-kadang untuk suatu proyek sistem informasi bisa tidak menggunakan sama sekali (Sholiq, 2010).

| filter pencarian | |
|------------------|--|
| • alamat | |
| • fasilitas | |
| • jenis_hunian | |
| • harga | |
| • universitas | |
| • radius | |
| | |
| ► load_data() | |
| ► get_code() | |
| ► get_lokasi() | |
| ► view_map() | |

A. *Statechart Diagram Booking Indekos*

Statechart Diagram Booking Indekos ini menjelaskan tentang perubahan status kamar indekos yang dipesan. Saat *user* indekos melakukan pemesanan, maka status kamar tersebut akan berubah jadi “pesan” dan sistem akan mengirimkan notifikasi ke *user* pemilik indekos. Lalu akan ada 2 kondisi yang terjadi, jika kamar tersebut cocok dan jadi disewa oleh *user* pencari indekos maka status kamar berubah menjadi “isi” tetapi jika pencari indekos tidak ingin menyewa kamar tersebut maka status kamar berubah menjadi “kosong”. Untuk lebih jelasnya dapat dilihat pada Gambar 3.18.

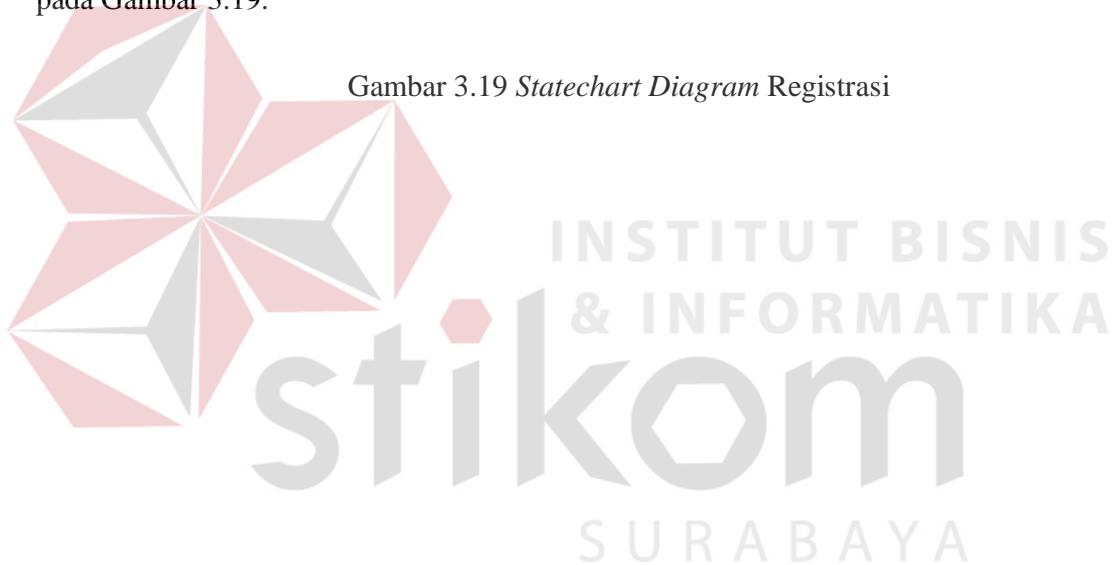
Gambar 3.18 *Statechart Diagram Booking Indekos*



B. *Statechart Diagram* Registrasi

Statechart Diagram Registrasi menjelaskan tentang pergantian sementara status *user* pemilik indekos saat registrasi awal. Saat proses registrasi, *user* pemilik diharuskan mengisi data registrasi yang akan disimpan pada *database* dan status *user* pemilik indekos sementara “belum aktif”. Setelah *user* pemilik melakukan validasi dengan memasukkan kode yang telah dikirim, maka akan terjadi 2 kondisi. Jika validasi berhasil maka *user* pemilik indekos akan berubah status menjadi “aktif” dan dapat mengisi data-data lebih lengkap, tetapi jika validasi gagal dan *user* pemilik indekos ingin melakukan pembatalan pendaftaran maka sistem akan melakukan pembatalan dan penghapusan data registrasi. Untuk lebih jelasnya dapat dilihat pada Gambar 3.19.

Gambar 3.19 *Statechart Diagram* Registrasi



3.3.5 Activity Diagram

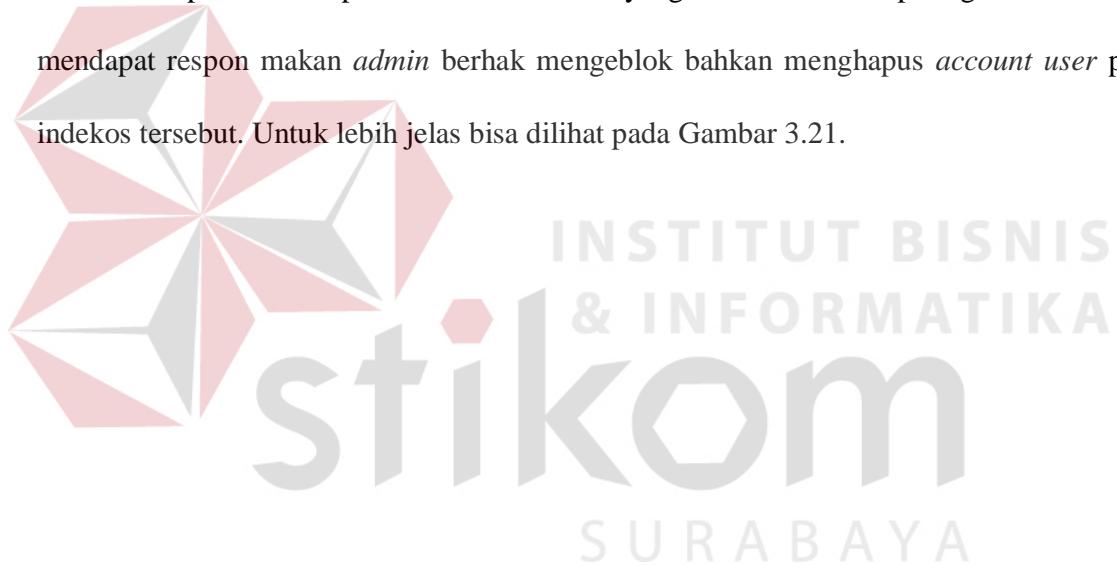
A. Activity Diagram Proses Maintenance Data Indekos

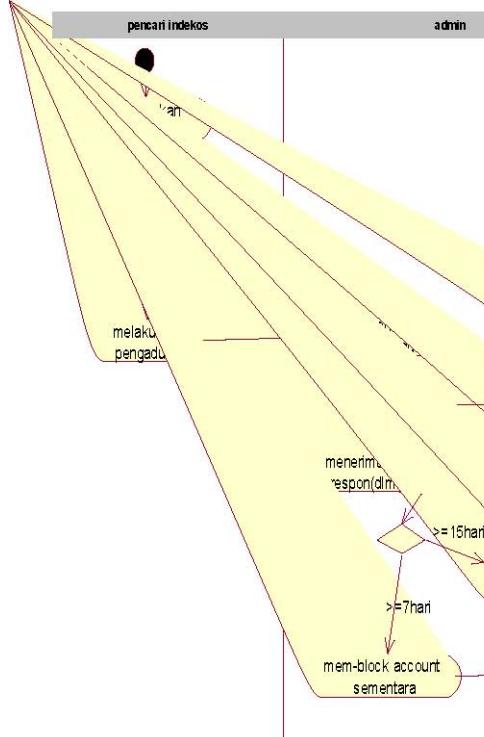
Proses *Maintenance* Data Indekos adalah proses yang melakukan manipulasi data (*insert, update, and delete*) baik untuk data indekos sendiri ataupun profil dari pemilik indekos. Proses ini dapat di lakukan hanya bagi *user* yang telah mendaftar sebagai pemilik indekos. Maka, *user* pemilik indekos baru harus melakukan pendaftaran dengan cara memilih menu Register dan mengisi *form* register yang telah disediakan pada *website*. Setelah mengisi *form* register dengan data yang benar lalu sistem akan menyimpan sementara data pemilik indekos baru tersebut dan mengirimkan pemberitahuan kepada *admin*. Saat *admin* menerima dan membaca pemberitahuan tersebut maka *admin* akan mencetak surat yang berisi kode validasi dan mengirimkannya melalui pos (manual) sebagai pembuktian bahwa alamat yang diberikan adalah *real* (nyata). Jika alamat tersebut benar dan nyata maka pemilik indekos pasti menerima surat validasi tersebut dan diimbau untuk segera mengaktifkan *account* pada aplikasi ini dengan cara memasukkan kode validasi tersebut setelah *login* menggunakan *username* dan *password* yang telah ditentukan saat pendaftaran sebelumnya. Untuk lebih jelas bisa dilihat pada Gambar 3.20.

Gambar 3.20 *Activity Diagram* untuk Proses *Maintenance* Data Indekos

B. *Activity Diagram* Proses *Maintenance User*

Proses *Maintenance User* digunakan untuk mengeblok dan menghapus *user* pemilik indekos yang dilaporkan oleh pencari indekos karena berbagai alasan yang merugikan. Proses ini dimulai ketika *user* pencari indekos merasa dirugikan dengan data yang dicantumkan oleh pemilik indekos. *User* pencari indekos harus masuk ke halaman profil pemilik indekos dengan cara klik lokasi indekos yang tertera pada peta. Lalu, *user* pencari indekos menuliskan keluhan yang dimiliki pada kotak yang telah disediakan. Setelah itu *admin* akan menerima pemberitahuan dan segera mengirim surat peringatan baik dalam bentuk pemberitahuan pada *website* ataupun melalui pos. Jika dalam waktu yang telah ditentukan peringatan tersebut tidak mendapat respon maka *admin* berhak mengeblok bahkan menghapus *account user* pemilik indekos tersebut. Untuk lebih jelas bisa dilihat pada Gambar 3.21.





Gambar 3.21 *Activity Diagram* untuk Proses Maintenance User

C. *Activity Diagram* Proses Pencarian Indekos

Proses Pencarian Indekos dimulai dengan membuka halaman *home* dan memilih menu pencarian. Ada 2 cara yang dapat dilakukan pencari indekos dalam melakukan pencarian yaitu pencarian biasa

Gambar 3.21 *Activity Diagram* untuk dengan langsung memasukkan obyek yang akan dicari dan pencarian dengan

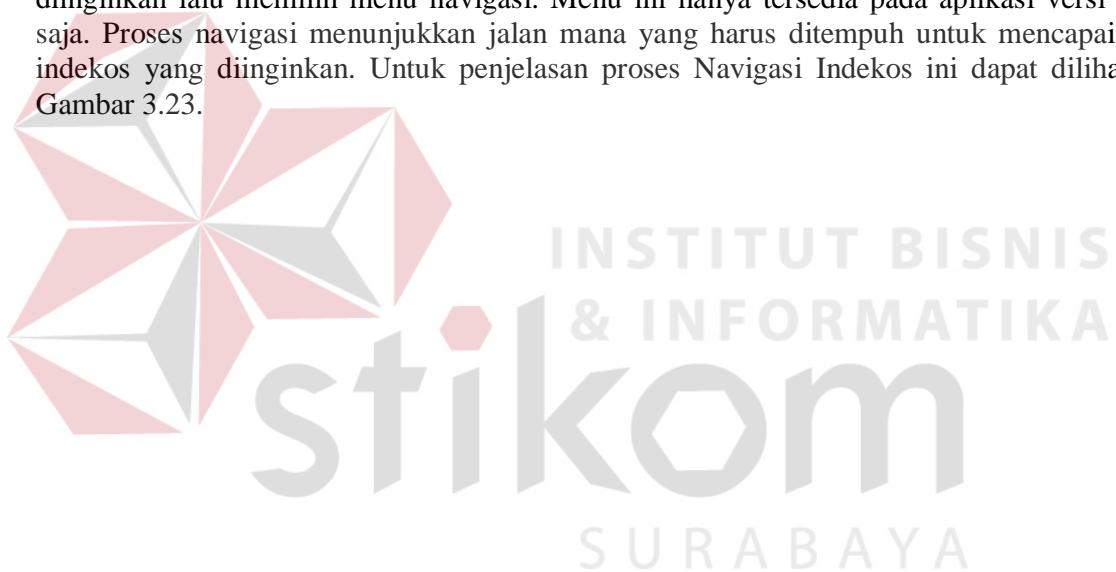
memasukkan filter-filter yang diinginkan.

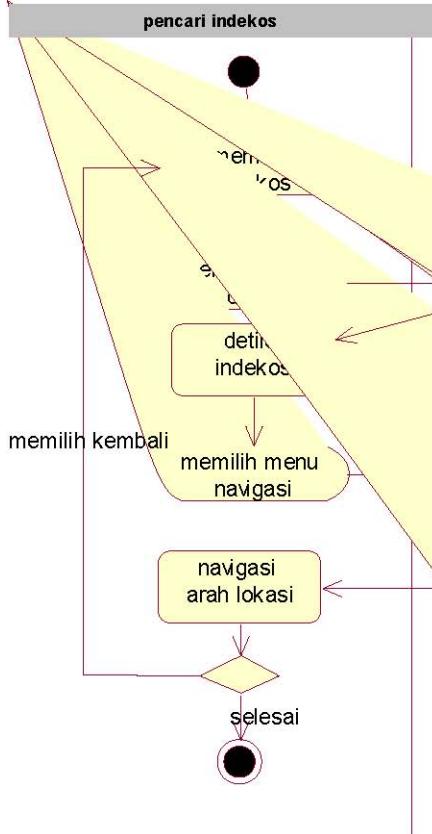


Setelah obyek dimasukkan, maka sistem akan melakukan pencarian pada *database*. Jika ada data yang sesuai dengan obyek yang dicari maka sistem akan mengirimkan koordinat untuk menampilkan lokasi pada *Google Maps*. Setelah menemukan indekos yang diinginkan maka *user* pencari indekos dapat melihat data *detile*-nya dengan memilih menu *detile* pada lokasi yang diinginkan tersebut. Untuk penjelasan proses Pencarian Indekos ini dapat dilihat pada Gambar 3.22.

Gambar 3.22 *Activity Diagram* untuk Proses Pencarian Indekos

D. *Activity Diagram* Proses Navigasi Indekos Proses Navigasi Indekos dapat dimulai setelah pencari indekos melakukan proses pencarian indekos dan menemukan indekos yang diinginkan lalu memilih menu navigasi. Menu ini hanya tersedia pada aplikasi versi *mobile* saja. Proses navigasi menunjukkan jalan mana yang harus ditempuh untuk mencapai lokasi indekos yang diinginkan. Untuk penjelasan proses Navigasi Indekos ini dapat dilihat pada Gambar 3.23.





Gambar 3.23 Activity Diagram untuk Proses Navigasi

Indekos

Gambar 3.23 Activity Diagram

E. Activity Diagram Proses Pesan Indekos

Proses Pesan Indekos merupakan lanjutan dari proses pencarian. Setelah pencari indekos melakukan pencarian dan menemukan kamar indekos yang sesuai dengan kriterinya, maka user pencari indekos dapat melakukan pemesanan kamar tersebut dalam jangka waktu tertentu dengan memasukkan data diri berupa nama, no.telepon, e-mail, jumlah kamar yang dipesan dan alamat asal. Data tersebut yang nantinya akan dikirim beserta data kamar yang dipesan baik ke *account* pemilik indekos ataupun *email* pencari indekos. Untuk penjelasan proses pesan indekos ini dapat dilihat pada Gambar 3.23./

INSTITUT BISNIS
& INFORMATIKA
ikom
SURABAYA

Gambar 3.24 Activity Diagram Proses Pesan Indekos

3.3.6 Sequence Diagram

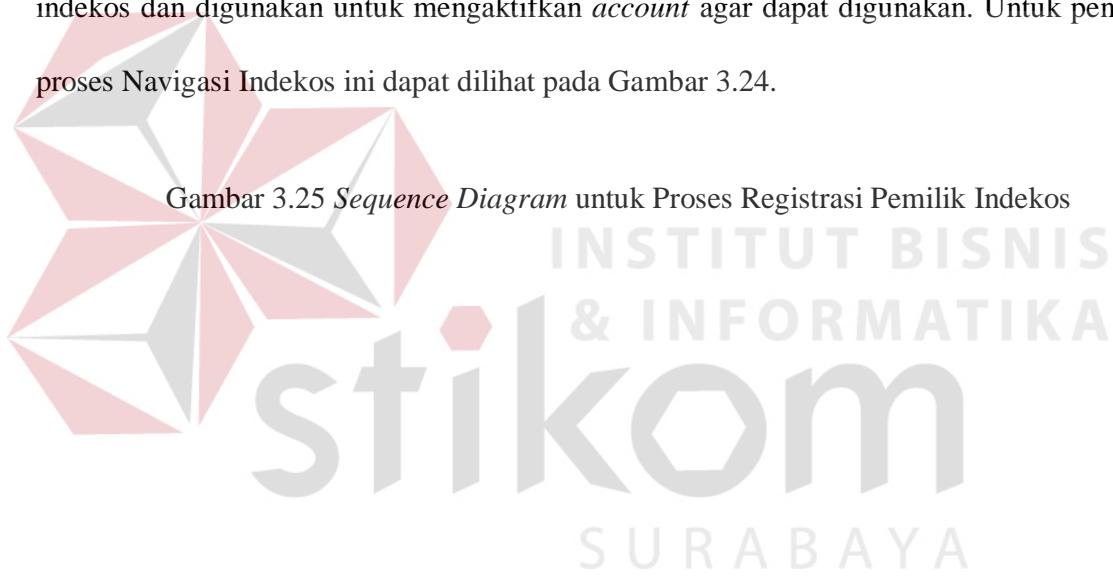
Sequence diagram digunakan untuk menggambarkan interaksi antar objek berdasarkan urutan waktu yang digambarkan dari atas ke bawah.



A. Sequence Diagram Proses Registrasi Pemilik Indekos

Proses dimulai ketika *user* membuka aplikasi dan pada tampilan awal akan keluar tampilan menu utama, kemudian *user* memilih menu registrasi. Menu ini diperuntukkan bagi pemilik indekos yang belum memiliki *account* pada aplikasi ini. Setelah memasuki *form* registrasi, pemilik indekos mengisikan data yang terdapat pada *form* tersebut dan memilih tombol simpan. Setelah itu data akan tersimpan sementara dalam *database* dan sistem akan mengirimkan pemberitahuan berupa notifikasi kepada *admin*. Saat *admin* membuka halaman *home*, *admin* akan menerima notifikasi tersebut dan selanjutnya *admin* akan mencetak surat yang berisi *username*, *password* dan kode aktivasi. Surat tersebut akan dikirimkan ke pemilik indekos dan digunakan untuk mengaktifkan *account* agar dapat digunakan. Untuk penjelasan proses Navigasi Indekos ini dapat dilihat pada Gambar 3.24.

Gambar 3.25 *Sequence Diagram* untuk Proses Registrasi Pemilik Indekos

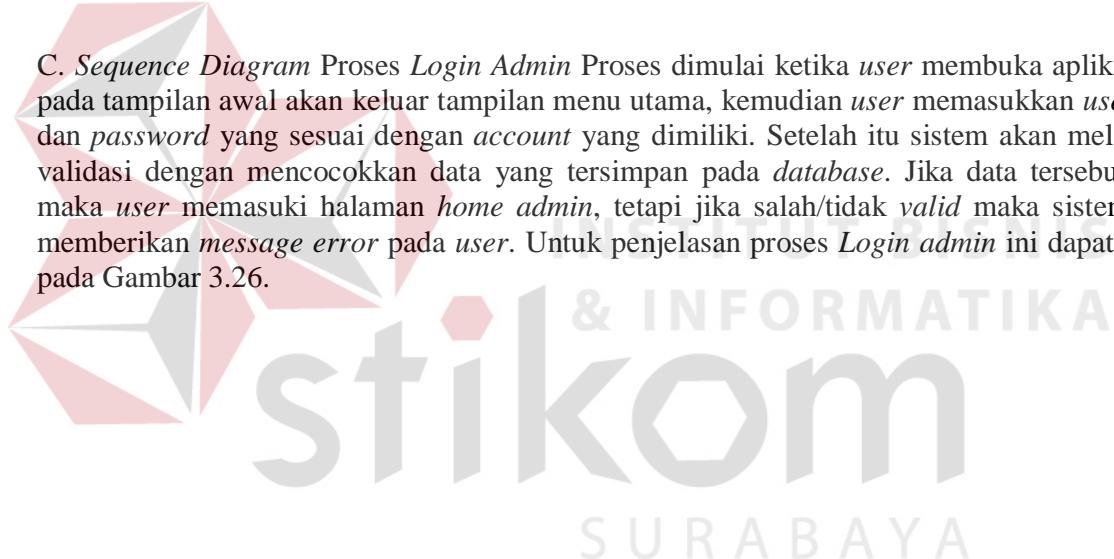


B. Sequence Diagram Proses Login Pemilik Indekos

Proses dimulai ketika *user* membuka aplikasi dan pada tampilan awal akan keluar tampilan menu utama, kemudian *user* memasukkan *username* dan *password* yang sesuai dengan *account* yang dimiliki. Setelah itu sistem akan melakukan validasi dengan mencocokkan data yang tersimpan pada *database*. Jika data tersebut valid maka *user* memasuki halaman *home* pemilik indekos, tetapi jika salah/tidak valid maka sistem akan memberikan *message error* pada *user*. Untuk penjelasan proses *Login* pemilik Indekos ini dapat dilihat pada Gambar 3.25.

Gambar 3.26 *Sequence Diagram* untuk Proses *Login* Pemilik Indekos

C. *Sequence Diagram* Proses *Login Admin* Proses dimulai ketika *user* membuka aplikasi dan pada tampilan awal akan keluar tampilan menu utama, kemudian *user* memasukkan *username* dan *password* yang sesuai dengan *account* yang dimiliki. Setelah itu sistem akan melakukan validasi dengan mencocokkan data yang tersimpan pada *database*. Jika data tersebut valid maka *user* memasuki halaman *home admin*, tetapi jika salah/tidak valid maka sistem akan memberikan *message error* pada *user*. Untuk penjelasan proses *Login admin* ini dapat dilihat pada Gambar 3.26.

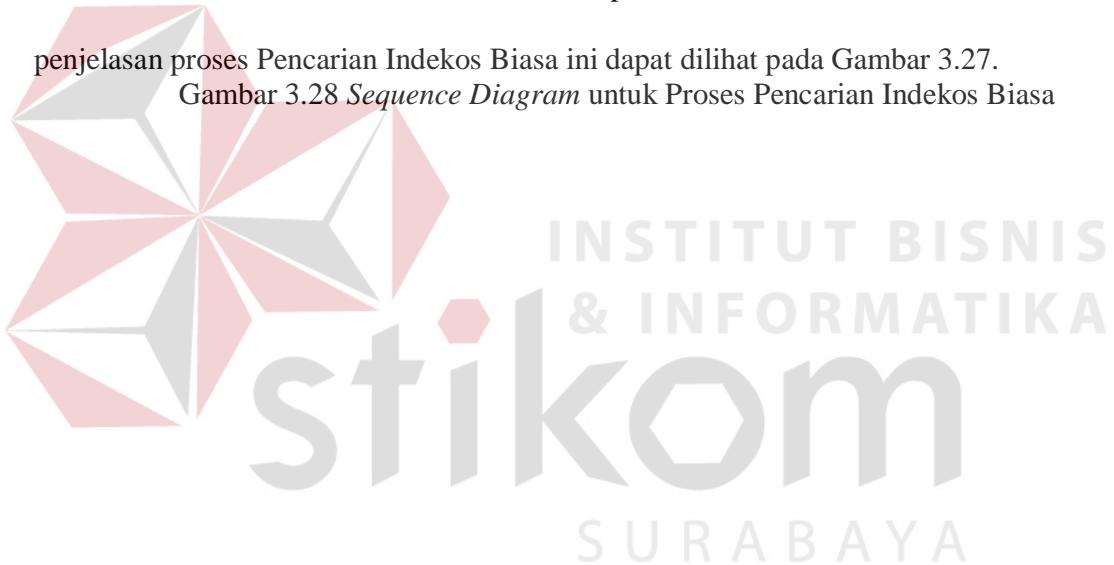


Gambar 3.27 *Sequence Diagram* untuk Proses *Login Admin*

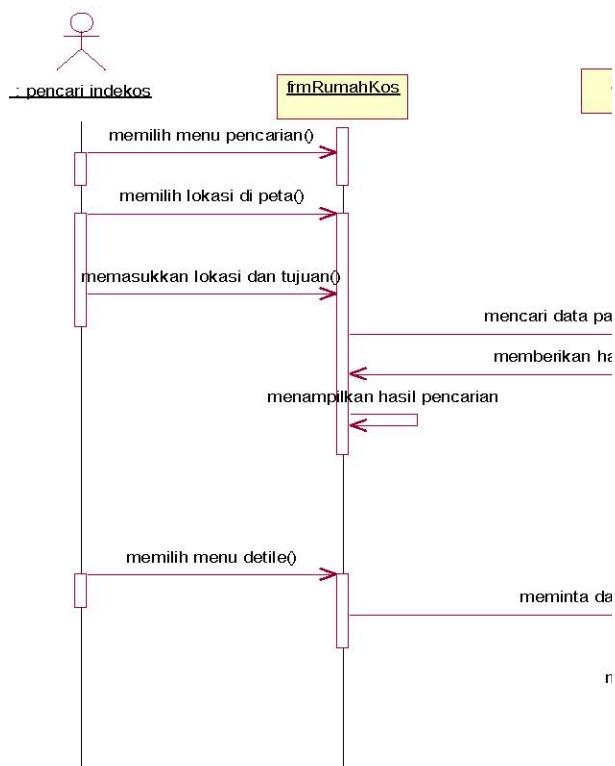
D. *Sequence Diagram* Proses Pencarian Indekos Biasa

Proses pencarian indekos biasa (tanpa filter) dimulai ketika *user* pencari indekos memilih menu pencarian pada halaman *home*. Setelah menu dipilih maka sistem akan membuka halaman rumah kos. Di halaman rumah kos ini terdapat kolom yang selanjutnya diisi dengan obyek lokasi tujuan yang diinginkan *user*. Obyek yang diisi selanjutnya akan dicocokkan dengan data pada *database*. Jika data ditemukan maka *database* akan memberikan hasil pencarian yang selanjutnya akan ditampilkan oleh sistem melalui peta pada *form* rumah kos. *User* pencari indekos yang ingin melihat *detile* data indekos tersebut dapat memilih menu *detile* dan sistem akan menampilkan *detile* data dari indekos tersebut. Untuk penjelasan proses Pencarian Indekos Biasa ini dapat dilihat pada Gambar 3.27.

Gambar 3.28 *Sequence Diagram* untuk Proses Pencarian Indekos Biasa



E. *Sequence Diagram* Proses Pencarian Indekos dengan Filter Proses pencarian indekos dengan filter dimulai ketika *user* pencari indekos memilih menu pencarian pada halaman *home*. Setelah menu dipilih maka sistem akan membuka halaman rumah kos. Di halaman rumah kos ini terdapat menu pencarian *advance*. Pencarian *advance* adalah pencarian dengan menggunakan beberapa filter yang diinginkan sehingga akan menampilkan indekos-indekos yang sesuai dengan kriteria pencari. Pada halaman pencarian *advance* ini akan terdapat beberapa kolom yang selanjutnya diisi sesuai keinginan *user*. Filter-filter yang diisi selanjutnya akan dicocokkan dengan data pada *database*. Jika data ditemukan maka *database* akan memberikan hasil pencarian yang selanjutnya akan ditampilkan oleh sistem melalui peta pada *form* rumah kos. *User* pencari indekos yang ingin melihat *detile* data indekos tersebut dapat memilih menu *detile* dan sistem akan menampilkan *detile* data dari indekos tersebut. Untuk penjelasan proses Pencarian Indekos dengan Filter ini dapat dilihat pada Gambar 3.28.



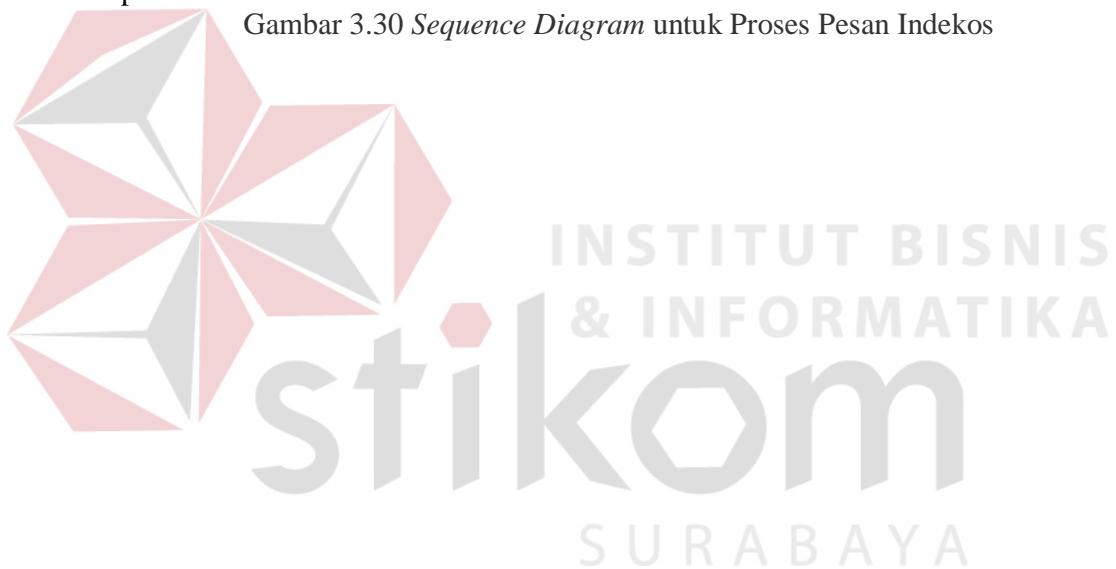
Gambar 3.28 *Sequence Diagram* untuk Proses

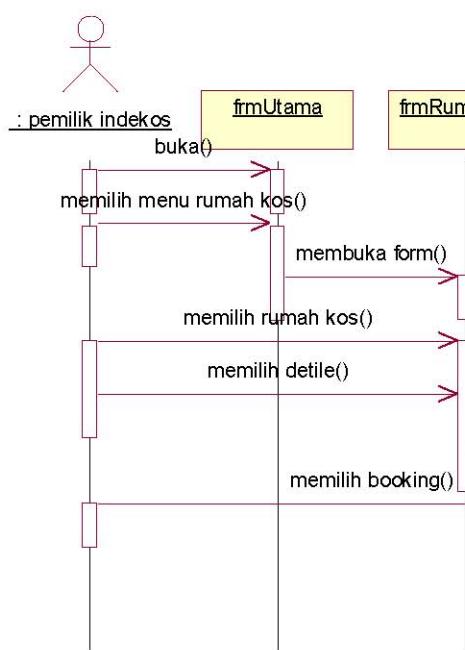
Gambar 3.29 *Sequence Diagram* untuk Proses Pencarian Indekos dengan Filter

F. *Sequence Diagram* Proses Pesan Indekos

Proses ini digunakan untuk memesan indekos yang dirasa cocok oleh *user* pencari indekos. Pencari indekos akan memilih indekos tersebut pada peta dan akan keluar *pop-up detile* indekos yang dipilih. Selanjutnya pencari indekos memilih menu *detile* untuk informasi lebih lanjut dan jika benar-benar dirasa cocok maka pencari indekos harus memilih menu pesan untuk selanjutnya mengisi form data diri dan disampaikan kepada pemilik indekos melalui sistem. Kamar pada indekos yang dipesan akan ditandai selama beberapa hari yang menandakan kamar tersebut telah dipesan. Untuk penjelasan proses Pesan Indekos ini dapat dilihat pada Gambar 3.29.

Gambar 3.30 *Sequence Diagram* untuk Proses Pesan Indekos





G. Sequence Diagram Proses Maintenance Data Indekos

Proses Maintenance Data Indekos ini hanya

bisa dilakukan oleh *user* pemilik indekos. Sebelum melakukan proses ini pemilik indekos diharuskan melakukan *login* dahulu. Setelah berhasil *login*, pemilik indekos akan masuk ke halaman *home* dan memilih menu *KosKu*. Pada halaman *KosKu* ini

Gambar 3.30 Sequence Diagram akan terdapat tabel yang berisi daftar

indekos-indekos yang dimiliki oleh pemilik tersebut. Pemilik indekos dapat melakukan hapus dan ubah pada daftar indekos tersebut. Tetapi jika pemilik indekos ingin menambah data indekosnya maka harus memilih menu tambah kos dan mengisi *form* indekos yang disediakan.

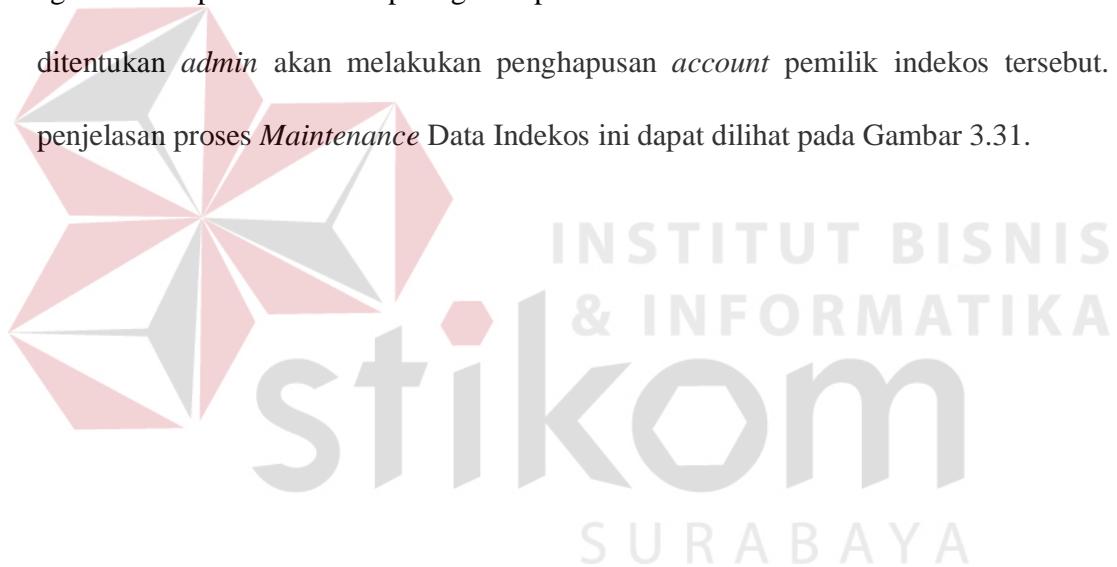
Untuk penjelasan proses *Maintenance Data Indekos* ini dapat dilihat pada Gambar 3.30.



Gambar 3.31 *Sequence Diagram* untuk Proses *Maintenance* Data Indekos

H. *Sequence Diagram* Proses *Maintenance User*

Proses *Maintenance User* ini diperlukan untuk mendisiplinkan pemilik indekos sebagai penyedia data agar data yang dimasukkan sesuai dengan keadaan di lapangan. Proses ini dilakukan saat *admin* menerima keluhan dari pencari indekos. *Admin* terlebih dahulu *login* dan masuk ke halaman *home admin*, kemudian memilih menu *user*. Pada halaman *user* ini terdapat *table* yang berisi daftar *user* pemilik indekos. *Admin* melakukan tindakan pendisiplinan ini sesuai dengan keluhan dari pencari indekos. Untuk peringatan pertama *admin* akan melakukan *block* sementara dan sistem akan mengirimkan notifikasi ke pemilik indekos agar data diperbaiki. Jika peringatan pertama tidak dihiraukan maka dalam waktu yang ditentukan *admin* akan melakukan penghapusan *account* pemilik indekos tersebut. Untuk penjelasan proses *Maintenance* Data Indekos ini dapat dilihat pada Gambar 3.31.



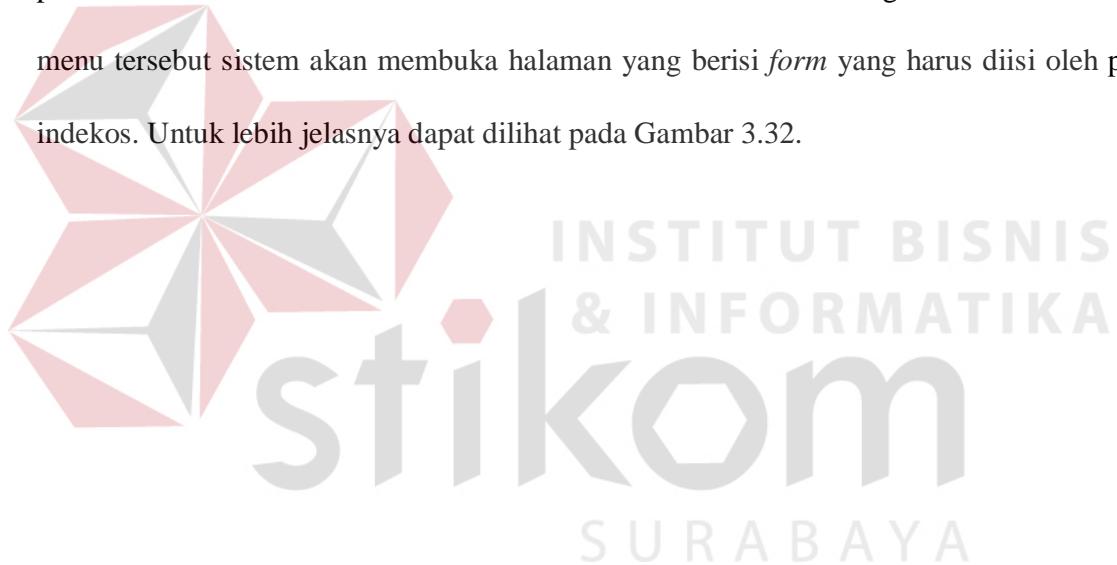
Gambar 3.32 *Sequence Diagram* untuk Proses *Maintenance User*

3.3.7 *Collaboration Diagram*

Diagram Kolaborasi (*Collaboration Diagram*) menunjukkan informasi yang sama persis dengan diagram sekuensial (*Sequence Diagram*), tetapi dalam bentuk dan tujuan yang berbeda. Pada diagram sekuensial, keseluruhan interaksi berdasarkan urutan waktu tetapi pada diagram kolaborasi interaksi antar obyek atau aktor ditunjukkan dengan arah panah tanpa keterangan waktu. (Sholiq, 2010).

A. *Collaboration Diagram* Proses Registrasi Pemilik Indekos

Collaboration Diagram pada Proses Registrasi Pemilik Indekos ini dimulai saat pemilik indekos membuka halaman utama lalu memilih menu registrasi. Setelah memilih menu tersebut sistem akan membuka halaman yang berisi *form* yang harus diisi oleh pemilik indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.32.



Gambar 3.33 *Collaboration Diagram* Proses Registrasi Pemilik Indekos

B. *Collaboration Diagram* Proses *Login* Pemilik Indekos

Proses pada diagram ini diawali dengan pemilik indekos membuka halaman *form* utama dan memasukkan *username* dan *password* yang dimiliki. *Username* dan *password* tersebut selanjutnya akan di validasi dan jika cocok maka *user* dapat membuka halaman *form* utama khusus pemilik indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.33.

Gambar 3.34 *Collaboration Diagram* Proses *Login* Pemilik Indekos



C. Collaboration Diagram Proses Login Admin

Collaboration Diagram pada proses ini hampir sama dengan proses *login* pemilik indekos. perbedaan ada pada halaman *form* yang dapat diakses sesuai dengan *user*-nya. Untuk lebih jelasnya dapat dilihat pada Gambar 3.34.

Gambar 3.35 *Collaboration Diagram* Proses Login Admin



D. *Collaboration Diagram* Proses Pencarian Indekos Biasa

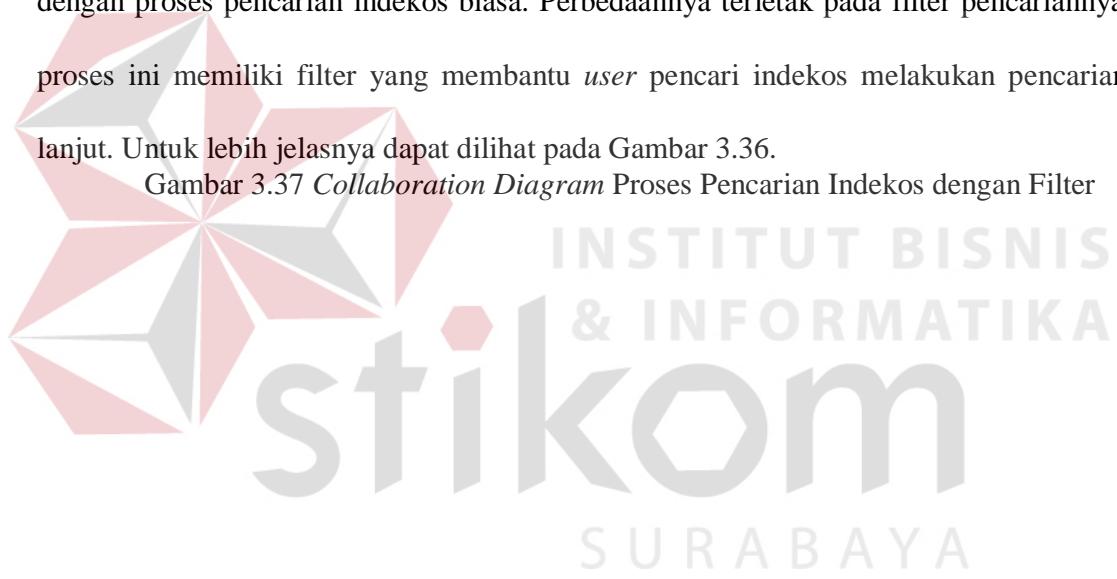
User pencari indekos memilih menu pencarian untuk memulai proses pencarian indekos biasa ini. Untuk proses pencarian biasa ini *user* hanya dapat melakukan pencarian standar dengan memasukkan langsung lokasi yang diinginkan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.35.

Gambar 3.36 *Collaboration Diagram* Proses Pencarian Indekos Biasa

E. *Collaboration Diagram* Proses Pencarian Indekos dengan Filter

Collaboration Diagram pada proses ini menjelaskan proses yang hampir sama dengan proses pencarian indekos biasa. Perbedaannya terletak pada filter pencarinya. Pada proses ini memiliki filter yang membantu *user* pencari indekos melakukan pencarian lebih lanjut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.36.

Gambar 3.37 *Collaboration Diagram* Proses Pencarian Indekos dengan Filter

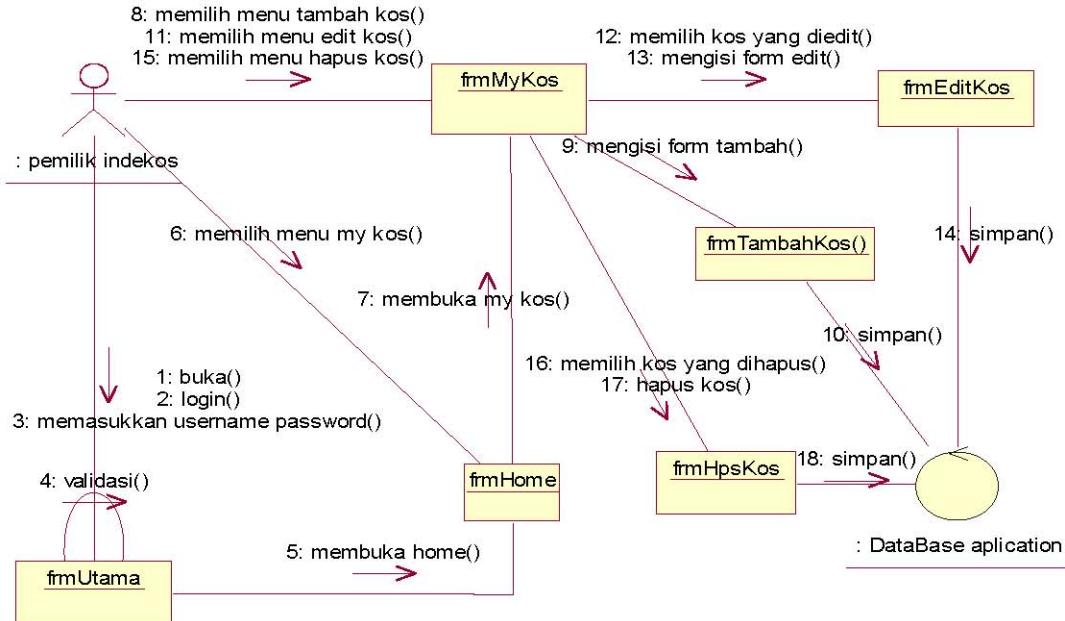


F. *Collaboration Diagram* Proses Pesan Indekos

Proses pesan indekos pada *collaboration diagram* ini dapat dimulai setelah *user* melakukan proses pencarian indekos atau telah mengetahui tujuan lokasi indekos yang diinginkan. Proses ini merupakan perantara antara pemilik indekos dan pencari indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.37.

Gambar 3.38 *Collaboration Diagram* Proses Pesan Indekos





Gambar 3.39 Collaboration Diagram Proses Maintenance Data Indekos

Collaboration diagram proses ini menjelaskan tentang bagaimana pemilik indekos melakukan proses *maintenance / update* data indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.38.

Gambar 3.39 Collaboration Diagram Proses Maintenance Data Indekos

H. Collaboration Diagram Proses Maintenance User

Collaboration diagram proses *maintenance user* ini menjelaskan tentang bagaimana *user admin* melakukan proses ini. proses ini memiliki 3 *form* yang diakses yaitu *form utama*, *form home admin* dan *form user*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.39.

Gambar 3.40 Collaboration Diagram Proses Maintenance User

3.3.8 Component Diagram

Diagram Komponen adalah diagram UML yang menampilkan komponen dalam sistem dan hubungan antara mereka. Dengan diagram ini, seseorang yang bertanggung jawab untuk mengompilasi dan men-deploy akan tahu kode pustaka mana saja yang di kompilasi lebih dulu. Jadi, diagram komponen adalah salah satunya berguna untuk mengetahui urutan kompilasi komponen-komponen yang akan dibuat.

Pada sistem pencari dan navigasi lokasi indekos, komponen-komponen di dalam *mobile application* dan *web application* dibagi menjadi dua komponen, yaitu *komponenGUI* yang menangani antar muka dengan pengguna dan *komponenControl* yang berisi semua kelas kontrol. Hubungan antar komponen dapat dilihat pada Gambar 3.40.

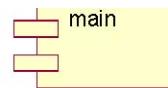


A. Package Specification Pada Mobile Application

1. Package Specification KomponenGUI

Package Specification yang terdapat dalam KomponenGUI mobile application ini dapat dilihat pada Gambar 3.41.

Gambar 3.42 Package Specification Application



KomponenGUI Mobile

2. Package Specification

Package Specification yang terdapat dalam KomponenKontrol mobile application ini dapat dilihat pada Gambar 3.42.

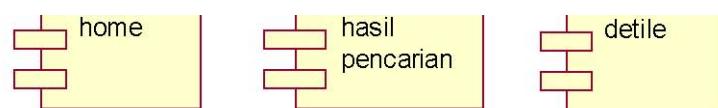
Gambar 3.43 Package Specification KomponenKontrol Mobile Application

B. Package Specification Pada Web Application

1. Package Specification KomponenGUI

Package Specification yang terdapat dalam KomponenGUI web application ini dapat dilihat pada Gambar 3.43.

Gambar 3.44 Package Specification KomponenGUI Web Application



2. *Package Specification KomponenKontrol*

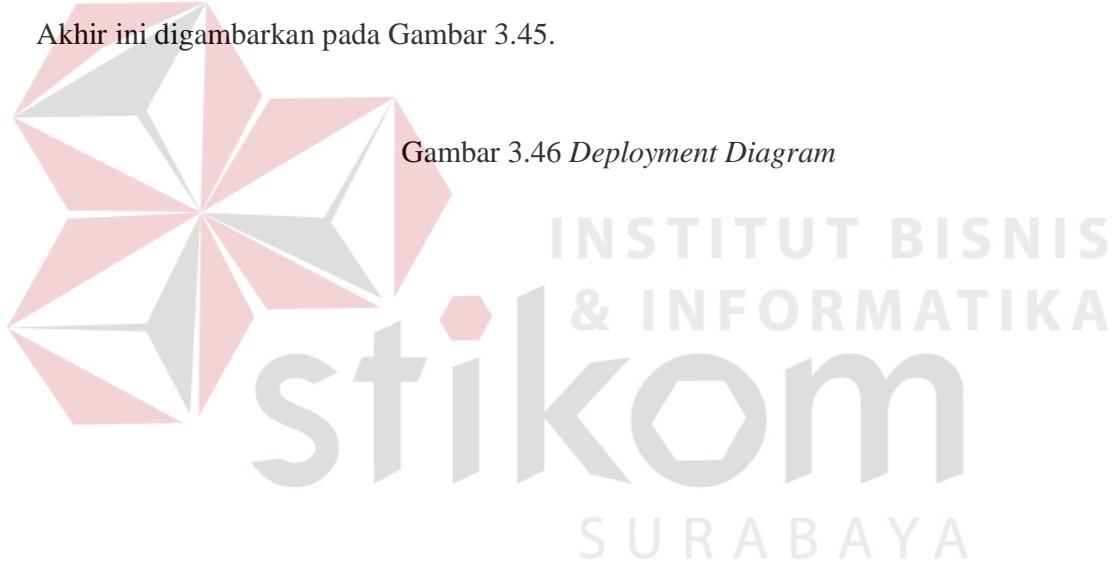
Package Specification yang terdapat dalam *KomponenKontrol* web application ini dapat dilihat pada Gambar 3.44.

Gambar 3.45 *Package Specification KomponenKontrol Web Application*

3.3.9 *Deployment Diagram*

Deployment Diagram menunjukkan pandangan secara fisik dari suatu sistem dan menunjukkan bagaimana sistem diimplementasikan di perangkat nyata. *Deployment Diagram* menampilkan semua *node* dalam suatu jaringan dan hubungan di antara mereka. *Node* adalah perangkat keras yang dapat menjadi *host* dari suatu aplikasi. *Deployment Diagram* pada Tugas Akhir ini digambarkan pada Gambar 3.45.

Gambar 3.46 *Deployment Diagram*



3.3.10 Struktur Tabel

Tabel-tabel yang digunakan dalam aplikasi sistem pencari dan navigasi lokasi indekos di Surabaya adalah sebagai berikut:

| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |

Tabel 3.2 Struktur Tabel *User*

| Nama Kolom | Type | Ukuran | Keterangan |
|------------------|---------|--------|------------|
| Id_user | Int | 11 | PK |
| Nama | Varchar | 50 | |
| Hak_akses | Varchar | 20 | |

| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |

| Nama kolom | Type | Ukuran | Keterangan |
|---------------------|---------|--------|------------|
| User_id | Int | 11 | PK |
| Id_user | Int | 11 | FK |
| User_name | Varchar | 50 | |
| Password | Varchar | 50 | |
| Nama_lengkap | Varchar | 100 | |
| Alamat | Varchar | 200 | |
| Jns_Kelamin | Varchar | 10 | |
| Kelurahan | Varchar | 50 | |
| Pekerjaan | Varchar | 50 | |
| No_telp | Varchar | 50 | |
| No_hp | Varchar | 50 | |
| Email | Varchar | 100 | |
| Pekerjaan | Varchar | 50 | |
| Foto | Varchar | 100 | |
| Secret | Varchar | 50 | |

C. Nama Tabel : Penyewa Kos

Fungsi : Menyimpan data penyewa indekos tiap kamar Tabel 3.4 Struktur Tabel Penyewa

Kos

| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos |
| Pesan Indekos | Proses yang digunakan untuk melakukan pemesanan kamar indekos dalam jangka waktu tertentu. |

A Nama Tabel : *User*

Fungsi : Menyimpan data *users*/pengguna yang dapat mengakses aplikasi.

| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos |
| Use Case | Keterangan |
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos |
| Pesan Indekos | Proses yang digunakan untuk melakukan pemesanan kamar indekos dalam jangka waktu tertentu. |

A Nama Tabel : *User*

Fungsi : Menyimpan data *users*/pengguna yang dapat mengakses aplikasi.

B Nama Tabel : Pemilik Kos

Fungsi : Menyimpan data pemilik indekos yang terdaftar.

Tabel 3.3 Struktur Tabel Pemilik Kos

| Nama kolom | Type | Ukuran | Keterangan |
|------------|------|--------|------------|
| No_KTP | Int | 20 | PK |

E. Nama Tabel : Kamar Kos

Fungsi : Menyimpan data kamar kos yang dimiliki tiap rumah kos. Tabel 3.6 Struktur

Tabel Kamar Kos

| Use Case | Keterangan | 3.4 |
|---------------------------------|--|-----|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos | |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. | |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos | |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos | |
| Pesan Indekos | Proses yang digunakan untuk melakukan pemesanan kamar indekos dalam jangka waktu tertentu. | |

Desain *Input / Output*

Pembuatan desain *input output* diperlukan untuk membantu pengguna berinteraksi dengan sistem. Desain *input output* yang dibuat meliputi desain aplikasi *web site* dan aplikasi *mobile*.

3.4.1 Desain aplikasi Web

A. Desain Untuk *User Admin*

1. Halaman *Home* Halaman *Home* adalah *form* yang pertama tampil setelah *user admin* melakukan *login*. Di dalam halaman *home admin* ini tertera pemberitahuan dan menu-menu yang dapat diakses oleh *user admin*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.46.

Gambar 3.47 Desain Halaman Home pada *User Admin*

2. Halaman *User* Halaman *user* adalah *form* yang hanya dapat diakses oleh *user admin*.

Halaman ini digunakan untuk memantau *user-user* tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.47.



Gambar 3.48 Desain Halaman *User* pada *User Admin*

3. Halaman Surat Validasi Halaman Surat Validasi berisi *preview* surat validasi yang akan dikirimkan ke *user* pemilik indekos yang baru melakukan registrasi. Surat validasi ini selanjutnya akan dicetak dan dikirimkan ke alamat *user* pemilik indekos yang sesuai dengan data registrasi yang tersimpan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.48.



Gambar 3.49 Desain Halaman Surat Validasi pada *User Admin*

B. Desain Untuk *User* Pemilik Indekos

1. Halaman *Home* Halaman *home* pada *user* pemilik indekos berisi pemberitahuan dan menu-menu yang dapat diakses oleh *user* pemilik indekos. menu-menu tersebut adalah menu *home*, menu profil, menu rumah kos, menu *KosKu*, menu aplikasi *mobile*, menu petunjuk dan menu logout. Untuk lebih jelasnya dapat dilihat pada Gambar 3.49.



Gambar 3.50 Desain Halaman *Home* pada *User* Pemilik Indekos

2. Halaman Profil Halaman profil akan tampil jika *user* pencari indekos ingin mengetahui lebih jauh tentang profil pemilik indekos. *User* pemilik indekos juga dapat mengakses halaman ini jika ingin mengetahui *preview* profil tentang dirinya. Untuk lebih jelasnya dapat dilihat pada Gambar 3.50.



Gambar 3.51 Desain Halaman Profil pada *User* Pemilik Indekos

3. Halaman Edit Profil *User* pemilik indekos yang ingin melakukan perubahan pada profilnya dapat menggunakan halaman edit profil ini. pada halaman ini akan terdapat kolom kolom yang dapat di ubah sesuai dengan keinginan *user* pemilik indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.51.



Gambar 3.52 Desain Halaman Edit Profil pada *User* Pemilik Indekos

4. Halaman Rumah Kos Halaman Rumah Kos digunakan jika *user* pemilik indekos ingin melihat

ataupun melakukan pencarian indekos yang ada di Surabaya. Untuk lebih jelasnya dapat dilihat pada Gambar 3.52.



Gambar 3.53 Desain Halaman Rumah Kos pada *User* Pemilik Indekos

5. Halaman *KosKu* Halaman *KosKu* berfungsi untuk menampilkan semua indekos yang dimiliki oleh *user* pemilik indekos dan telah tersimpan pada *database*. Indekosindekos yang dimiliki ditampilkan pada sebuah peta digital beserta lokasi dan jika lokasi tersebut diklik, maka akan tampil keterangan tentang indekos tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.53.



Gambar 3.54 Desain Halaman *KosKu* pada *User Admin*

6. Halaman Tambah Rumah Kos Halaman Tambah Rumah Kos berfungsi untuk *user* pemilik indekos yang ingin melakukan penambahan rumah indekos yang dimilikinya. Halaman ini terdiri dari kolom-kolom yang harus diisi sebagai data indekos tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.54.



Gambar 3.55 Desain Halaman Tambah Rumah Kos

7. Halaman Detail Indekos Halaman Detail Indekos digunakan saat *user* pemilik indekos ingin memberikan data secara detail mengenai setiap kamar indekos yang dimiliki. Untuk lebih jelasnya dapat dilihat pada Gambar 3.55.



Gambar 3.56 Desain Halaman Detail Indekos

8. Halaman Detail Kamar Halaman Detail Kamar berfungsi untuk *user* pemilik indekos ingin mengetahui keterangan setiap kamar secara merinci. Terdiri dari sebuah tabel yang berisi *id_kamar*, nama penyewa, status kamar, tempat kerja, umur, status pembayaran dan keterangan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.56.



9. Halaman Detail Pemberitahuan Halaman Detail Pemberitahuan berfungsi untuk menampilkan pemberitahuan-pemberitahuan yang diterima oleh *user* pemilik indekos secara rinci. Untuk lebih jelasnya dapat dilihat pada Gambar 3.57.



Gambar 3.58 Desain Halaman Detail Pemberitahuan

10. Halaman Peringatan Pemilik Halaman peringatan digunakan untuk memberikan peringatan kepada *user* pemilik indekos dari *admin*. Peringatan tersebut terjadi jika *user* pemilik indekos melanggar peraturan-peraturan yang telah disepakati pada awal registrasi. Peringatan tersebut dikirim secara 2 tahap yaitu tahap peringatan dan tahap penghapusan. Tahap peringatan diberikan untuk peringatan kesalahan dan memberikan tenggang waktu perbaikan, sedangkan penghapusan adalah peringatan penghapusan *account* saat peringatan awal tidak dihiraukan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.58.

Gambar 3.59 Desain Halaman Peringatan pada *User* Pemilik Indekos

11. Halaman Validasi Halaman Validasi digunakan untuk *user* pemilik indekos yang ingin mengaktifkan *account*-nya setelah melakukan registrasi. Validasi dapat dilakukan dengan memasukkan kode validasi yang telah dikirimkan oleh *admin* ke alamat *user* pemilik indekos yang telah diisikan saat registrasi. Untuk lebih jelasnya dapat dilihat pada Gambar 3.59.



Gambar 3.60 Desain Halaman Validasi pada *User* Pemilik Indekos

12. Halaman Registrasi Halaman Registrasi ini adalah halaman awal yang harus diisi oleh calon *user* pemilik indekos yang ingin memiliki *account* pada sistem ini. halaman ini terdiri dari kolom-kolom data diri *user* pemilik indekos. setelah mengisi, *user* menekan tombol simpan dan sistem akan menyimpan data serta mengirimkan notifikasi ke *user admin*. Selanjutnya *admin* melakukan proses validasi. Untuk lebih jelasnya dapat dilihat pada Gambar 3.60.



Gambar 3.61 Desain Halaman Registrasi pada *User* Pemilik Indekos

C. Desain Untuk *User* Pencari Indekos (*default*)

1. Halaman *Home* Halaman *Home user* pencari indekos merupakan halaman yang *default* tampil ketika semua *user* mengakses *website* sistem ini. halaman ini berisi menu umum yang dapat diakses semua *user*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.61.



Gambar 3.62 Desain Halaman Home pada *User* Pencari Indekos

2. Halaman Mencari Indekos Halaman Mencari Indekos ini digunakan saat *user* pencari indekos ingin melakukan pencarian. Halaman ini hanya melakukan pencarian secara umum dengan memasukkan kata kunci lokasi yang ingin dicari. Untuk lebih jelasnya dapat dilihat pada Gambar 3.62.



Gambar 3.63 Desain Halaman Mencari Indekos pada *User* Pencari Indekos

3. Halaman Mencari Indekos (dengan parameter) Halaman Mencari Indekos dengan Parameter ini diakses jika pencari indekos ingin melakukan pencarian secara terperinci (*advance search*). Pencarian ini dilakukan dengan memilih parameter-parameter yang diinginkan dalam pencarian dan mengisi parameter-parameter tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.63.



Gambar 3.64 Desain Halaman Mencari Indekos (parameter)

4. Halaman Detail Halaman Detail adalah halaman yang ditampilkan jika *user* pencari indekos ingin informasi lebih rinci mengenai indekos yang dipilihnya. Pada halaman ini juga terdapat menu detail pemilik dan detail kamar yang dapat dipilih *user* pencari indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.64.



Gambar 3.65 Desain Halaman Detail pada *User* Pencari Indekos

5. Halaman Detail Kamar Halaman Detail Kamar merupakan kelanjutan dari halaman detail sebelumnya. Halaman ini berisi mengenai spesifikasi kamar yang terdapat pada indekos tersebut. Jika *user* pemilik indekos tertarik dengan kamar pada indekos ini, *user* dapat melakukan pemesanan dengan memilih menu pesan pada halaman ini. Untuk lebih jelasnya dapat dilihat pada Gambar 3.65.



Gambar 3.66 Desain Halaman Detail Kamar pada *User* Pencari Indekos

6. Halaman Konfirmasi Pemesanan Halaman Konfirmasi Pemesanan merupakan halaman lanjutan setelah menu pesan dipilih. Halaman berisikan kolom nama dan nomor telepon yang akan disimpan sementara dan dikirimkan sebagai pemberitahuan ke *user* pemilik indekos. pada *user* pencari indekos, akan dikirimkan berupa pesan yang harus



ditunjukkan saat konfirmasi pemesanan ke pemilik indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.66.

Gambar 3.67

Desain Halaman

Konfirmasi

Pemesanan

7. Halaman

| Permintaan Pesan | |
|--------------------------------------|----------------------|
| Nama | <input type="text"/> |
| No.HP | <input type="text"/> |
| <input type="button" value="Kirim"/> | |

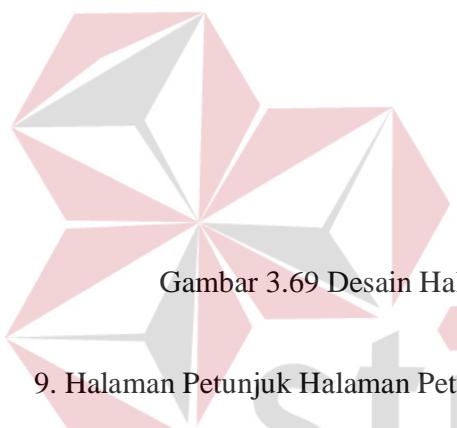
Gambar 3.67 Desain Halaman Konfirmasi Pemesanan

Profil

Pemilik Indekos Halaman Profil Pemilik Indekos merupakan halaman yang menampilkan data diri *user* pemilik indekos. halaman ini dapat dijadikan referensi jika *user* pencari indekos ingin mengenal lebih jauh tentang *user* pemilik indekos. Pada halaman ini juga terdapat kolom komentar yang digunakan sebagai kritik dan saran terhadap *user* pemilik indekos serta grafik reputasi sebagai bahan pertimbangan *user* pencari indekos dalam melakukan pemilihan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.67.

Gambar 3.68 Desain Halaman Profil Pemilik Indekos

8. Halaman Aplikasi *Mobile* Halaman Aplikasi *Mobile* diakses jika *user* pencari indekos ingin mengunduh aplikasi *mobile*. Aplikasi yang telah diunduh kemudian dipasang pada *smartphone* yang dimiliki. Untuk lebih jelasnya dapat dilihat pada Gambar 3.68.



Gambar 3.69 Desain Halaman Aplikasi *Mobile* pada *User* Pencari Indekos

9. Halaman Petunjuk Halaman Petunjuk menampilkan tata cara pemakaian aplikasi ini. halaman ini dapat diakses oleh semua *user*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.69.

Gambar 3.70 Desain Halaman Petunjuk pada *User* Pencari Indekos

10. Halaman Tentang Kami Halaman Tentang Kami menampilkan cara komunikasi dengan *admin*.

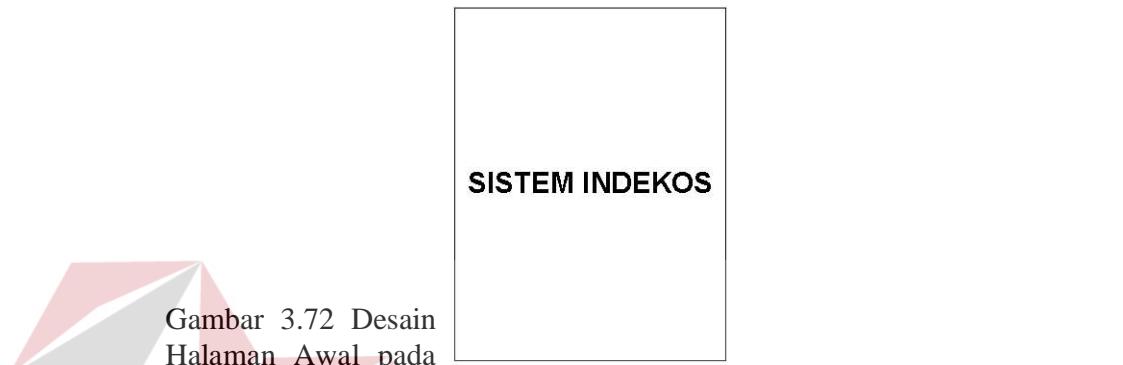
Pada halaman ini tertera nomor telepon yang dapat dihubungi dan alamat *e-mail admin*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.70.



Gambar 3.71 Desain Halaman Tentang Kami pada *User* Pencari Indekos

3.4.2 Desain Aplikasi Mobile

A. Halaman Awal

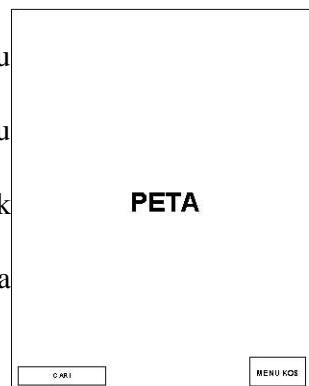


Gambar 3.72 Desain Halaman Awal pada *Mobile Application* in Halaman Awal pada *Mobile Application*

B. Halaman Utama

Halaman utama merupakan halaman yang tampil setelah halaman awal. merupakan halaman yang tampil *default* menampilkan peta Surabaya. Selain peta, ada setelah halaman awal. Halaman utama yaitu menu cari dan menu indekos. Untuk lebih utama akan secara *default* ambar 3.72. menampilkan peta Surabaya.

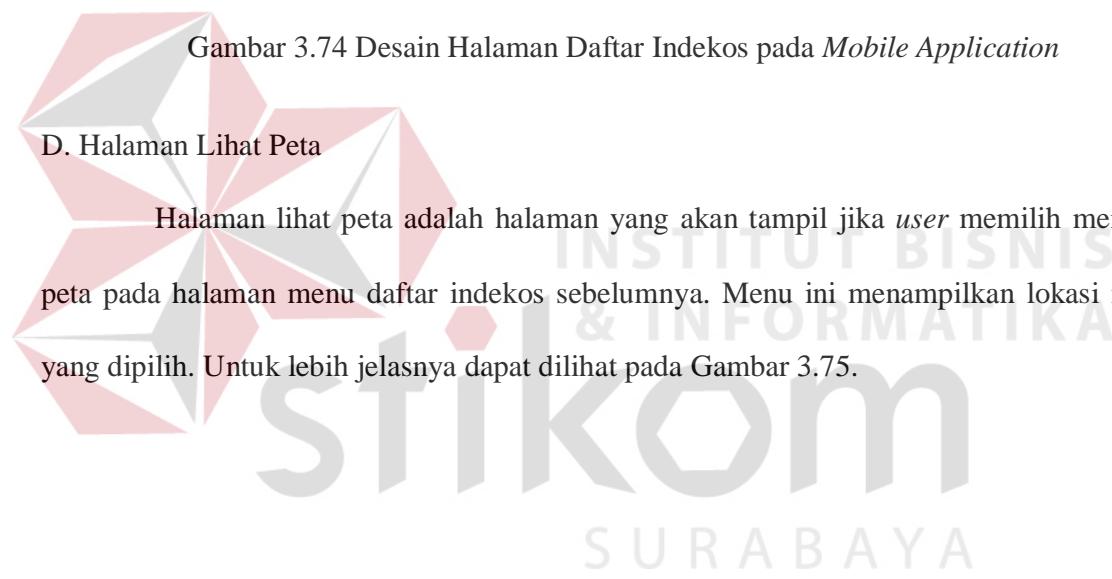
Selain peta, ada pula 2 menu yang ditampilkan yaitu menu cari dan menu indekos. Untuk lebih jelasnya dapat dilihat pada Gambar 3.72.



Gambar 3.73 Desain Halaman Utama pada *Mobile Application*

C.

Halaman daftar Halaman daftar



Gambar 3.74 Desain Halaman Daftar Indekos pada *Mobile Application*

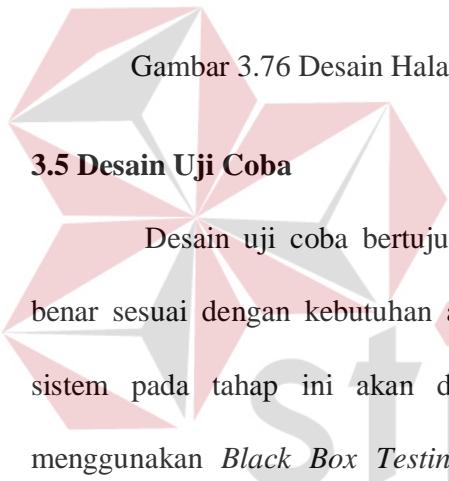
D. Halaman Lihat Peta

Halaman lihat peta adalah halaman yang akan tampil jika *user* memilih menu lihat peta pada halaman menu daftar indekos sebelumnya. Menu ini menampilkan lokasi indekos yang dipilih. Untuk lebih jelasnya dapat dilihat pada Gambar 3.75.

Gambar 3.75 Desain Halaman Lihat Peta pada *Mobile Application*

E. Halaman Detail Rumah Indekos

Halaman detail rumah indekos terdiri dari daftar spesifikasi indekos yang diinginkan oleh *user*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.76.



Gambar 3.76 Desain Halaman Detail Rumah Indekos pada *Mobile Application*

3.5 Desain Uji Coba

Desain uji coba bertujuan untuk memastikan bahwa aplikasi telah dibuat dengan benar sesuai dengan kebutuhan atau tujuan yang diharapkan. Kekurangan atau kelemahan sistem pada tahap ini akan dievaluasi sebelum diimplementasikan. Proses pengujian menggunakan *Black Box Testing* yaitu aplikasi akan diuji dengan melakukan berbagai percobaan untuk membuktikan bahwa aplikasi yang telah dibuat sudah sesuai dengan tujuan.

Uji coba yang akan dilakukan antara lain:

1. Uji coba fungsi aplikasi.
2. Uji coba kompatibilitas sistem.

3.5.1 Uji Coba Fungsi Aplikasi

A. Desain Uji Coba Aplikasi *Web*

Proses uji coba ini dilakukan untuk mengetahui apakah fungsi-fungsi dari Aplikasi Sistem Pencarian dan Navigasi Indekos di Surabaya ini telah berjalan dengan benar. Setiap fitur yang disediakan akan diuji hasilnya sesuai dengan tabel *test case*. Desain uji coba fungsi aplikasi adalah sebagai berikut:



1. Desain Uji Coba *Login* Fungsi *login* digunakan untuk melakukan pembatasan hak akses kepada pengguna. Pada fungsi ini, terdapat pemeriksaan kelengkapan masukan nama pengguna dan kata kunci. Sistem memberikan pesan agar pengguna mengisi data ketika ada yang masih kosong. Selain itu, sistem juga akan memberi peringatan apabila nama pengguna dan kata kuncinya salah. Desain untuk uji coba fungsi

login dapat dilihat pada Tabel 3.7. Tabel 3.7 Desain Uji Coba Fungsi *Login*

2. Desain Uji Coba *Block User* Fungsi *Block User* digunakan untuk melakukan pemblokiran hak akses

kepada *user* pemilik indekos jika *admin* mendapatkan pengaduan bahwa pemilik indekos melakukan kesalahan dalam memasukkan informasi indekos yang dimilikinya. Sistem akan melakukan pemblokiran untuk mengakses *account* yang dimiliki. Desain untuk uji coba fungsi *block user* dapat dilihat pada Tabel 3.8.

| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos |
| Pesan Indekos | Proses yang digunakan untuk melakukan pemesanan kamar indekos dalam jangka waktu tertentu. |

A Nama Tabel : *User*
 .
 Fungsi : Menyimpan data *users*/pengguna yang dapat mengakses aplikasi.

B Nama Tabel : Pemilik Kos
 .
 Fungsi : Menyimpan data pemilik indekos yang terdaftar.

Tabel 3.3 Struktur Tabel Pemilik Kos

| Nama kolom | Type | Ukuran | Keterangan |
|------------|------|--------|------------|
| No KTP | Int | 20 | PK |

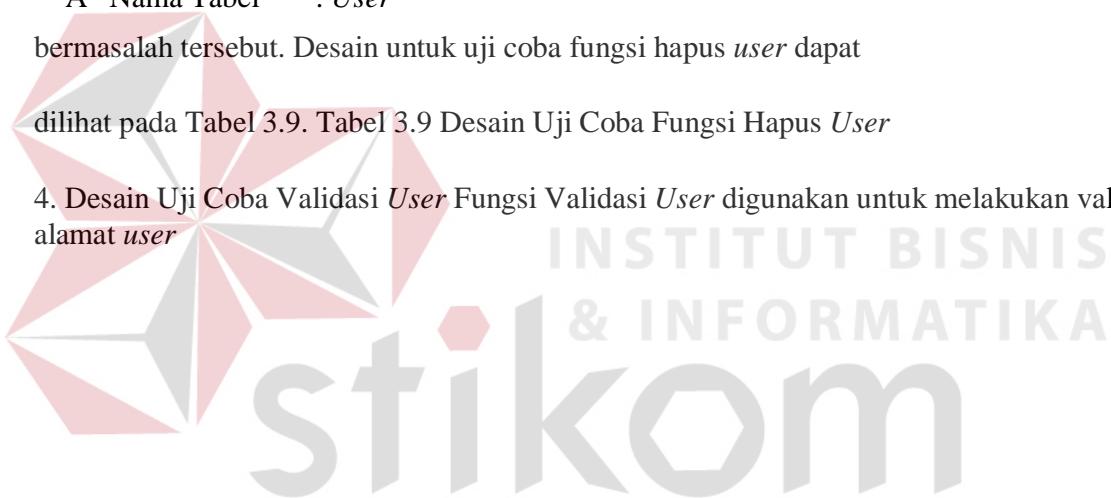
Tabel 3.8 Desain Uji Coba Fungsi *Block User*

3. Desain Uji Coba Hapus *User* Fungsi Hapus *User* digunakan untuk melakukan penghapusan *account user* pemilik indekos jika pemblokiran tidak dihiraukan oleh pemilik indekos dalam kurun waktu yang ditentukan. *Admin* akan melakukan penghapusan untuk *account* yang

| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos |
| Pesan Indekos | Proses yang digunakan untuk melakukan pemesanan kamar indekos dalam jangka waktu tertentu. |

A Nama Tabel : *User*
 bermasalah tersebut. Desain untuk uji coba fungsi hapus *user* dapat dilihat pada Tabel 3.9. Tabel 3.9 Desain Uji Coba Fungsi Hapus *User*

4. Desain Uji Coba Validasi *User* Fungsi Validasi *User* digunakan untuk melakukan validasi alamat *user*



| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos |
| Pesan Indekos | Proses yang digunakan untuk melakukan pemesanan kamar indekos dalam jangka waktu tertentu. |

A Nama Tabel : *User*
 Fungsi : Menyimpan data *users*/pengguna yang dapat mengakses aplikasi.

B Nama Tabel : Pemilik Kos

pemilik indekos. *Admin* akan mencetak dan mengirimkan surat validasi yang berisi kode validasi ke alamat pemilik indekos yang tersimpan. Kode validasi tersebut harus dimasukkan ke profil pemilik indekos dengan memilih menu “edit profil”. Desain untuk uji coba fungsi validasi *user* dapat dilihat pada Tabel 3.10.

Tabel 3.10 Desain Uji Coba Fungsi Validasi *User*

5. Desain Uji Coba Registrasi *User* Pemilik Indekos Fungsi Registrasi *User* Pemilik Indekos digunakan untuk melakukan pendaftaran menjadi *user* pemilik indekos dan menyimpannya ke

| Use Case | Keterangan |
|---------------------------------|--|
| Maintenance Data Indekos | Proses yang digunakan untuk menyimpan data-data indekos |
| Maintenance User | Proses menghapus dan mengeblok <i>user</i> yang dirasa mengganggu kenyamanan sistem. |
| Pencarian Indekos | Proses yang digunakan untuk melakukan pencarian lokasi indekos |
| Navigasi Lokasi Indekos | Proses yang digunakan sebagai petunjuk rute menuju lokasi indekos |
| Pesan Indekos | Proses yang digunakan untuk melakukan pemesanan kamar indekos dalam jangka waktu tertentu. |

- A Nama Tabel : *User*
Fungsi : Menyimpan data *users*/pengguna yang dapat mengakses aplikasi.
- B Nama Tabel : Pemilik Kos
Fungsi : Menyimpan data pemilik indekos yang terdaftar.

dalam *database*. Pemilik indekos diharuskan mengisi data diri pada kolom-kolom yang disediakan di halaman registrasi. Semua kolom harus diisi dengan lengkap dan benar, jika

kosong maka sistem akan memberikan peringatan “data harus diisi”. Desain untuk uji coba fungsi registrasi *user* pemilik indekos dapat dilihat pada Tabel 3.11.

6. Desain Uji Coba Tambah Data Indekos Fungsi Tambah Data Indekos digunakan oleh *user* pemilik indekos untuk melakukan penambahan data indekos yang dimiliki dan menyimpannya dalam *database*. Pemilik indekos memilih menu tambah dan mengisi kolom-kolom yang tersedia dengan data rumah indekos yang ingin ditambahkan. Untuk penambahan lokasi pada peta bisa dilakukan dengan menekan titik poin lokasi

Tabel 3.11 Desain Uji Coba Fungsi Registrasi User Pemilik Indekos

| 13 Test Case ID | Menghindari kolom data kosong. Tujuan | Mengisikan dengan data kosong. Input | Tampil pesan “data tidak boleh kosong” Output yang diharapkan |
|-----------------|--|--|---|
| 14 | Memastikan data diri tersimpan dalam database. | Mengisikan data dan menekan tombol simpan. | Tampilan pesan “Data telah tersimpan” |
| 15 | Menghindari data yang dimasukkan salah. | Memasukkan data dengan format yang salah. | Tampil pesan “kode yang anda inputkan salah” |

pada peta yang disediakan atau dengan memasukkan koordinat pada kolom yang ada. Desain untuk uji coba fungsi tambah rumah indekos dapat dilihat pada Tabel



Tabel 3.12 Desain Uji Coba Fungsi Tambah Data Indekos

| 16 Test Case ID | Menghindari kolom data kosong. Tujuan | Mengisikan dengan data kosong. Input | Tampil pesan “data tidak boleh kosong” Output yang diharapkan |
|-----------------|--|---|---|
| 17 | Memastikan data indekos tersimpan dalam database. | Mengisikan data dan menekan tombol simpan. | Tampilan pesan “Data telah tersimpan” |
| 18 19 | Menghindari data yang dimasukkan salah. Memastikan lokasi rumah indekos | Memasukkan data dengan format yang salah. Memasukkan koordinat lokasi rumah indekos | Tampil pesan “kode yang anda inputkan salah” Titik poin indekos secara dengar |

Te
st

i
m
a
s
u
k
k
a
n
.

| Test Case ID | Tujuan | Input |
|--------------|------------|-------|
| | pada peta. | |

Out
put
yang
Case
ID

k
o
o
r
d
i
n
a
d
i
h
a
r
a
g
p
k
d

| 16 Test Case ID | Menghindari kolom data kosong. Tujuan | Mengisikan dengan data kosong. Input | Tampil pesan “data tidak boleh kosong” Output yang diharapkan |
|-----------------|--|--|---|
| 17 | Memastikan data indekos tersimpan dalam database. | Mengisikan data dan menekan tombol simpan. | Tampilan pesan “Data telah tersimpan” |
| 7. Desain | Menghindari data yang dimasukkan salah. Memastikan | Memasukkan data dengan format yang salah. Memasukkan | Tampil pesan “kode yang anda inputkan |

Uji Coba Edit Data Indekos Fungsi Edit Data Indekos digunakan oleh *user* pemilik indekos untuk melakukan perubahan data indekos yang dimiliki dan menyimpannya dalam *database*. Pemilik indekos memilih menu ubah dan mengganti data pada kolom kolom yang tersedia dengan data rumah indekos yang baru. Desain untuk uji coba fungsi tambah rumah indekos dapat dilihat pada Tabel 3.13.

Tabel 3.13 Desain Uji Coba Fungsi Edit Data Indekos

8. Desain Uji Coba Hapus Data Indekos Fungsi Hapus Data Indekos digunakan oleh *user* pemilik indekos untuk melakukan penghapusan data indekos yang dimiliki. Pemilik indekos memilih menu hapus pada indekos yang ingin dihapus dari *database*. Desain untuk uji coba fungsi tambah rumah indekos dapat dilihat pada Tabel 3.14.



Tabel 3.14 Desain Uji Coba Fungsi Hapus Data Indekos

| Test | a n | pada pesan | d i h a p u s |
|--------------------|--------|---------------------------------|---------------------------------|
| Output yang | | M e n e k a n | |
| T | | t | i |
| u | | o | n |
| j | | m | d |
| u | | b | e |
| a | | o | l |
| n | | I | k |
| | | n | o |
| | | p | s |
| | | “ | |
| | | O | |
| | | 23 | |
| | | t | b |
| | | k | e |
| | | ” | r |
| | | p | Dh |
| | | e | a a |
| | | n | t s |
| | | g | |
| | | h | a i |
| | | a | l |
| | | p | . |
| | | u | i n |
| | | s | d e |
| | | a | k o |
| | | p | s |
| | | k | |
| | | a | y |
| | | a | a |
| | | n | n |
| | | d | g |
| | | a | . |
| | | t | |
| | | a | |
| | | penghapusan | |
| | | data | |
| | | | |
| Case ID | | | |
| M | | | |
| e | | | |
| m | | | |
| a | | | |
| s | | | |
| t | | | |
| i | | | |
| k | | | |
| | | fungsi | |

9. Desain Uji Coba Pencarian Indekos Fungsi Pencarian Indekos digunakan oleh *user* pencari indekos untuk melakukan pencarian indekos yang diinginkan. Pencari indekos memilih menu Rumah Kos yang melakukan pencarian indekos yang sesuai dengan kriteria. Desain untuk uji coba fungsi tambah rumah indekos dapat dilihat pada Tabel 3.15.

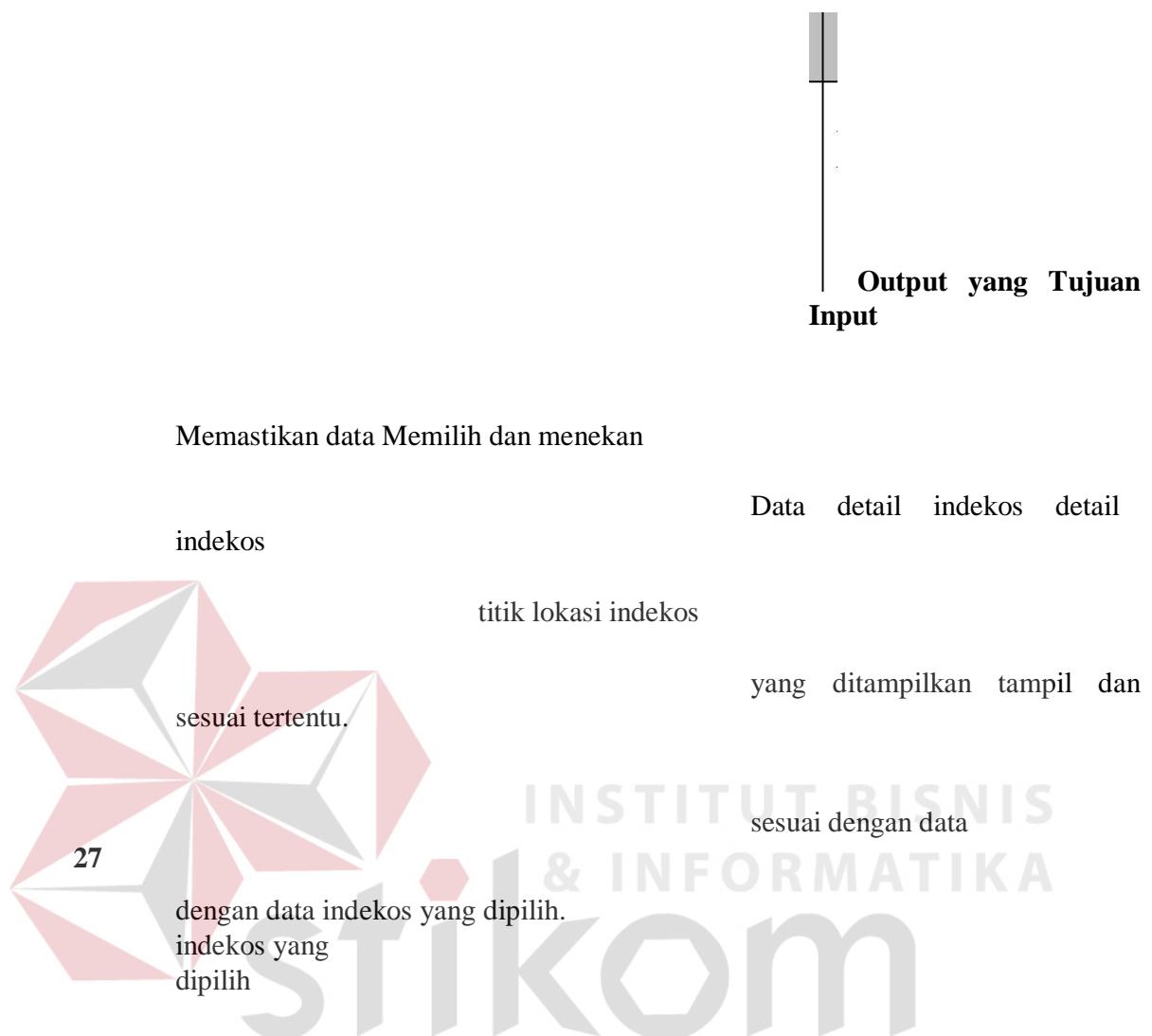
Tabel 3.15 Desain Uji Coba Fungsi Pencarian Data Indekos

10. Desain Uji Coba Detail Indekos Fungsi Detail Indekos digunakan oleh *user* pencari indekos untuk mengetahui data indekos lebih jelas. *User* pencari indekos memilih detail pada titik lokasi indekos di peta. Desain untuk uji coba fungsi tambah rumah indekos dapat dilihat pada Tabel 3.16.

| Test Case ID | Tujuan | Input | Output yang diharapkan |
|--------------|--|--|--|
| 16 | Menghindari kolom data kosong. Tujuan | Mengisikan dengan data kosong. Input | Tampil pesan “data tidak boleh kosong” |
| 17 | Memastikan data indekos tersimpan dalam database. | Mengisikan data dan menekan tombol simpan. | Output yang diharapkan Tampilan pesan “Data telah tersimpan” |
| 18 19 | Menghindari data yang dimasukkan salah. Memastikan lokasi rumah terdapat | Memasukkan data dengan format yang salah. Memasukkan koordinat lokasi rumah indekos. | Tampil pesan “kode yang anda inputkan salah” Titik poin lokasi sesuai dengan |
| 20 | Menghindari kolom data kosong. Tujuan | Mengisikan dengan data kosong. Input | Tampil pesan “data tidak boleh kosong” |

Tabel 3.16 Desain Uji Coba Fungsi Detail Indekos

Test



11. Desain Uji Coba Pesan Indekos Fungsi Pesan Indekos digunakan oleh *user* pencari indekos untuk melakukan pemesanan pada kamar indekos yang diinginkan. *User* pencari indekos memilih tombol pesan pada kamar yang dinginkan serta memasukkan nama dan nomor telepon sebagai validasi pemesanan. Desain untuk uji coba fungsi pesan indekos dapat dilihat pada Tabel 3.17.

Tabel 3.17 Desain Uji Coba Pesan Indekos

B. Desain Uji Coba Aplikasi *Mobile*

1. Desain Uji Coba Peta

Fungsi Peta digunakan oleh *user* pencari indekos untuk melihat semua lokasi indekos yang tersimpan di database dalam bentuk peta digital. *User* pencari indekos memilih menu peta pada halaman utama dan diharuskan menyalakan GPS pada ponsel. GPS digunakan agar sistem dapat mengetahui posisi user pencari indekos saat membuka aplikasi ini. Desain untuk uji coba fungsi peta dapat dilihat pada Tabel 3.18.



| 16 Test Case ID | Menghindari kolom data kosong. Tujuan | Mengisikan dengan data kosong. Input | Tampil pesan “data tidak boleh kosong” Output yang diharapkan |
|-----------------|--|---|---|
| 17 | Memastikan data indekos tersimpan dalam database. | Mengisikan data dan menekan tombol simpan. | Tampilan pesan “Data telah tersimpan” |
| | Menghindari data yang dimasukkan salah. Memastikan lokasi rumah | Memasukkan data dengan format yang salah. Memasukkan koordinat lokasi | Tampil pesan “kode yang anda inputkan salah” Titik poin |

Tabel 3.18 Desain Uji Coba Peta

2. Desain Uji Coba Detail Indekos Fungsi Detail Indekos digunakan oleh *user* pencari indekos untuk mengetahui informasi lebih lanjut mengenai rumah indekos yang dipilih. *User* pencari indekos menekan titik lokasi indekos yang dipilih dan sistem akan menampilkan halaman Detail Rumah. Desain untuk uji coba fungsi detail indekos dapat dilihat

| Test Case ID | Tujuan | Input | Output yang diharapkan |
|--------------|---|-------------------|--|
| 31 | Memastikan peta digital muncul dan menampilkan lokasi indekos | Memilih menu peta | Peta digital yang menampilkan lokasi indekos |

pada Tabel 3.19.

Tabel 3.19 Desain Uji Coba Detail Indekos

| Case ID | Tujuan | Input | Output yang diharapkan |
|---------|--|----------------------------------|-------------------------------|
| 31 | Memastikan halaman detail indekos pada peta menekan lokasi indekos | Menekan lokasi indekos pada peta | Halaman detail indekos muncul |
| 32 | rumah tampil | rumah tampil | |

3. Desain Uji Coba Detail Kamar

Fungsi Detail Kamar digunakan oleh *user* pencari indekos jika ingin mengetahui data kamar yang dimiliki oleh indekos yang dipilih. *User* pencari indekos memilih tombol kamar pada halaman detail rumah dan sistem akan menampilkan halaman detail kamar. Desain untuk uji coba fungsi detail kamar dapat dilihat pada Tabel 3.20.



Tabel 3.22 Desain Uji Coba Pesan Indekos

| | | | |
|------------------------|---|---|--|
| 35 Test Case ID | Memastikan status kamar berubah setelah pemesanan. Tujuan | Menekan tombol pesan Input | Status kamar berubah dari “kosong” menjadi “pesan” Output yang diharapkan |
| Test Case ID | Memastikan data pemesan tampil pada Tujuan pemilik indekos. | Memasukkan data pemesan Input menekan tombol “OK” Mengisi data pemesan dengan data kosong. | Email terkirim dan berisi data yang sesuai dengan pemesan. Output yang diharapkan |
| 36 37 | Memastikan Menghindari data halaman detail pemesan kosong. kamar muncul hasil | Menekan tombol kamar data pemesan dengan data kosong. | Muncul halaman detail Tampil pesan “Data kamar tidak boleh kosong” |
| 38 Test | Memastikan hasil pencarian tampil pada peta dan | Memilih filter yang diinginkan dan memasukkan inputan | Hasil Pencarian tampil pada peta sesuai dengan inputan user dan |

Tabel 3.20 Desain Uji Coba Detail Kamar

4. Desain Uji Coba Rute Fungsi Rute digunakan oleh *user* pencari indekos untuk mengetahui rute menuju lokasi indekos yang dipilih dari titik posisi user saat itu. *User* pencari indekos memilih tombol rute pada halaman detail rumah dan sistem akan menampilkan pada peta digital. Desain untuk uji coba fungsi rute dapat dilihat pada Tabel 3.21. Tabel 3.21 Desain Uji Coba rute

5. Desain Uji Coba Pesan Kamar Fungsi Pesan Indekos digunakan oleh *user* pencari indekos untuk melakukan pemesanan pada kamar indekos yang diinginkan. *User* pencari indekos

| Test Case ID | Tujuan | Input | Output yang diharapkan |
|---------------------|----------------------------------|---------------------|--------------------------------------|
| 34 | Memastikan rute tampil pada peta | Menekan tombol rute | Rute lokasi indekos tampil pada peta |

memilih tombol pesan pada kamar yang dinginkan serta memasukkan nama ,

nomor telepon, *email*, jumlah kamar, dan alamat asal sebagai validasi pemesanan. Desain untuk uji coba fungsi pesan indekos dapat dilihat pada Tabel 3.22.

6. Desain Uji Coba Pencarian Indekos Fungsi Pencarian Indekos digunakan oleh *user* pencari indekos untuk melakukan pencarian indekos yang diinginkan. *User* pencari indekos memilih

menu pencarian dan memasukkan kriteria indekos yang diinginkan. Desain untuk uji coba fungsi pencarian indekos dapat dilihat pada Tabel 3.23.

Tabel 3.23 Desain Uji Coba Pencarian Indekos

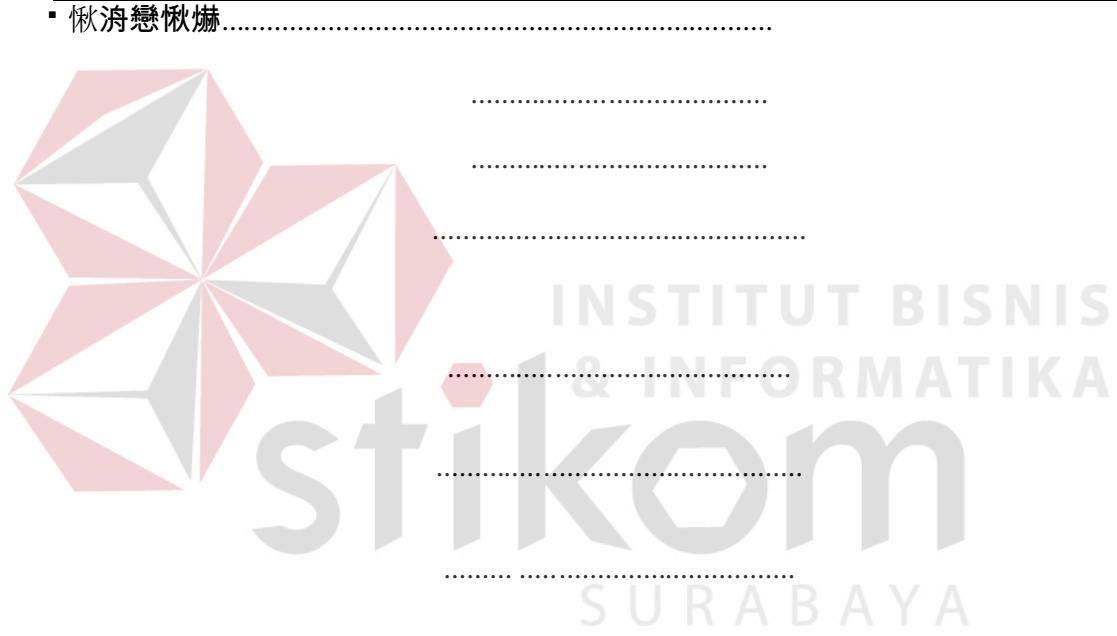
indekos menekan gambar bintang pada pojok kanan atas halaman detail rumah

| 35 Test Case ID | Memastikan status kamar berubah setelah pemesanan. Tujuan | Menekan tombol pesan Input | Status kamar berubah dari “kosong” menjadi “pesan” Output yang diharapkan |
|-----------------|---|--|--|
| 26 27 | Memastikan data pemesan tampil pada email pemilik indekos. Menghindari data pemesan kosong | Memasukkan data pemesan dan menekan tombol “OK” Mengisi data pemesan dengan data kosong | Email terkirim dan berisi data yang sesuai dengan pemesan. Tampil pesan “Data tidak boleh kosong” |
| 35 Test Case ID | Memastikan status kamar berubah setelah pemesanan. Tujuan | Menekan tombol pesan Input | Status kamar berubah dari “kosong” menjadi “pesan” Output yang diharapkan |

dan dapat melihatnya kembali dengan memilih menu *list* tandai. Desain untuk uji coba fungsi tandai dapat dilihat pada Tabel 3.24.

Tabel 3.24 Desain Uji Coba Tandai

| 40 Test Case ID | Memastikan gambar bintang berhasil menandai Tujuan | Menekan tombol bintang Input | Tampil pesan “Rumah Kos Berhasil di Tandai” Output yang diharapkan |
|-----------------|---|-------------------------------------|---|
| 41 | Memastikan data indekos yang ditandai tampil pada halaman <i>list</i> tandai. | Memilih menu <i>list</i> tandai | Ada data indekos yang ditandai pada halaman <i>list</i> tandai. |







INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA