



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODE DAN PERANCANGAN SISTEM

1.1 Metode Penelitian

Sistem informasi yang berbasis android ini dibuat dengan menggunakan metode *forward chaining*. Langkah-langkah kerja penelitian adalah sebagai berikut.

1. Studi Literatur

Penulis melakukan studi mengenai sistem informasi, metode *forward chaining*, *tools* yang akan digunakan dan properti melalui literatur-literatur seperti buku, jurnal dan sumber ilmiah lain seperti laman web, artikel dan dokumen teks yang berhubungan.

2. Wawancara

Penulis melakukan wawancara kepada pakar untuk memperoleh data yang diperlukan untuk penelitian dan pembangunan sistem informasi.

3. Analisis Kebutuhan Sistem

Kebutuhan sistem didasarkan pada kebutuhan pengguna berupa: tipe pengguna sistem, antarmuka atau *interface*, dan kebutuhan proses.

4. Desain Sistem

Hasil analisis diubah ke dalam bentuk teknis berupa rancangan sistem. Antarmuka dirancang menggunakan konsep GUI.

5. Implementasi

Hasil desain diimplementasikan dengan bahasa pemrograman Java, baik dalam bentuk form untuk *user interface*, maupun implementasi untuk metodenya. Sedangkan tabel disajikan dengan database relasional.

6. Pengujian

Sistem informasi yang telah dibuat akan diuji coba. Hal ini bertujuan untuk mencari *bug* yang masih ada pada sistem tersebut.

1.2 Jadwal Penelitian

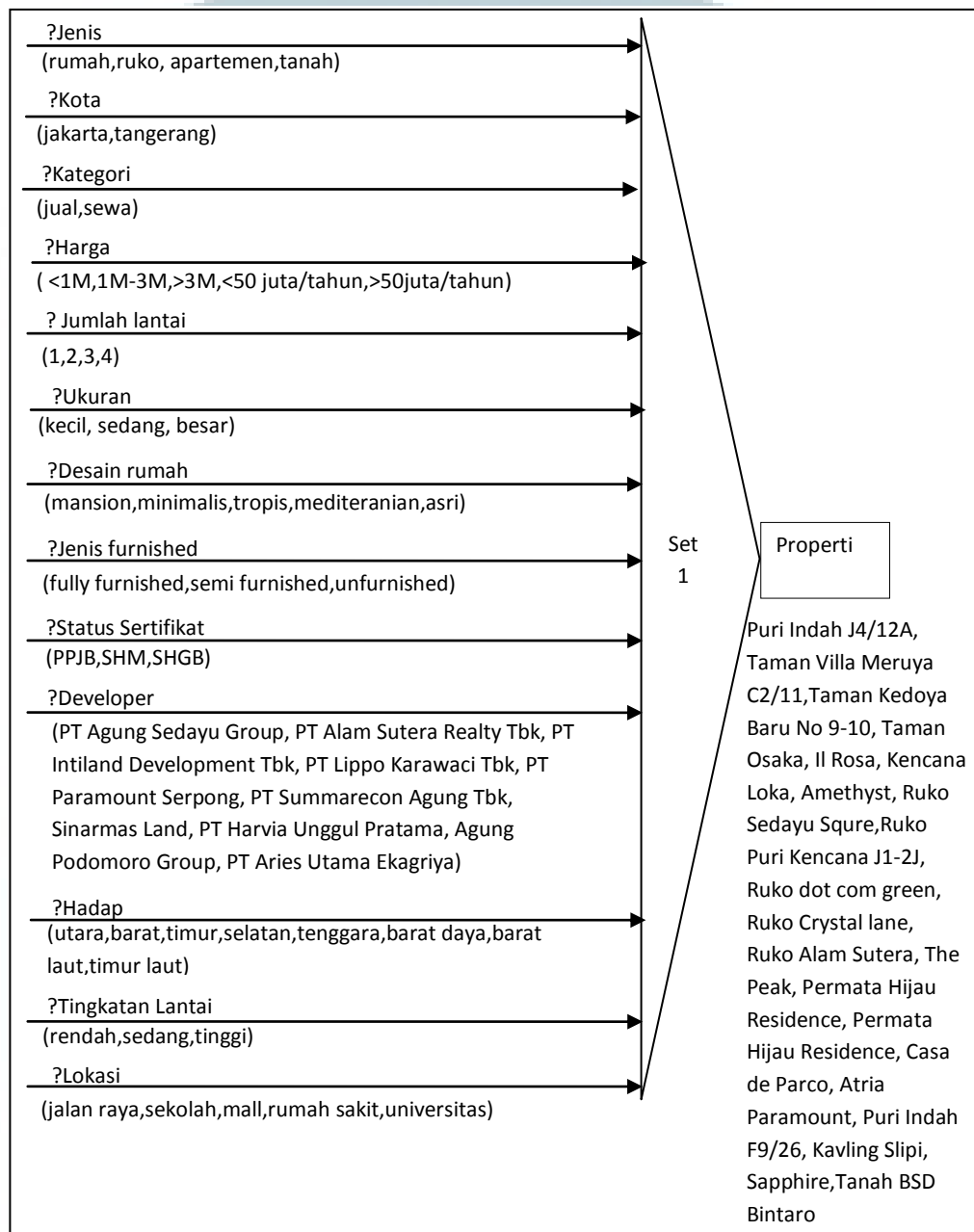
Kegiatan penelitian dijadwalkan sebagai berikut.

No.	Jenis Kegiatan	Minggu Ke -										
		1	2	3	4	5	6	7	8	9	10	11
1.	Studi Literatur											
2.	Wawancara											
3.	Analisis Kebutuhan Sistem											
4.	Desain Sistem											
5.	Implementasi											
6.	Pengujian											

3.3 Rule Based

Sistem informasi untuk properti membutuhkan basis pengetahuan dan mesin inferensi untuk mengetahui properti yang sesuai kriteria. Basis pengetahuan ini

merupakan fakta-fakta yang dibutuhkan oleh sistem, sedangkan mesin inferensi digunakan untuk menganalisa fakta-fakta yang dimasukkan *user* hingga dapat ditemukan suatu simpulan. Gambar 3.1 merupakan *dependency diagram* sistem informasi pemasaran properti.

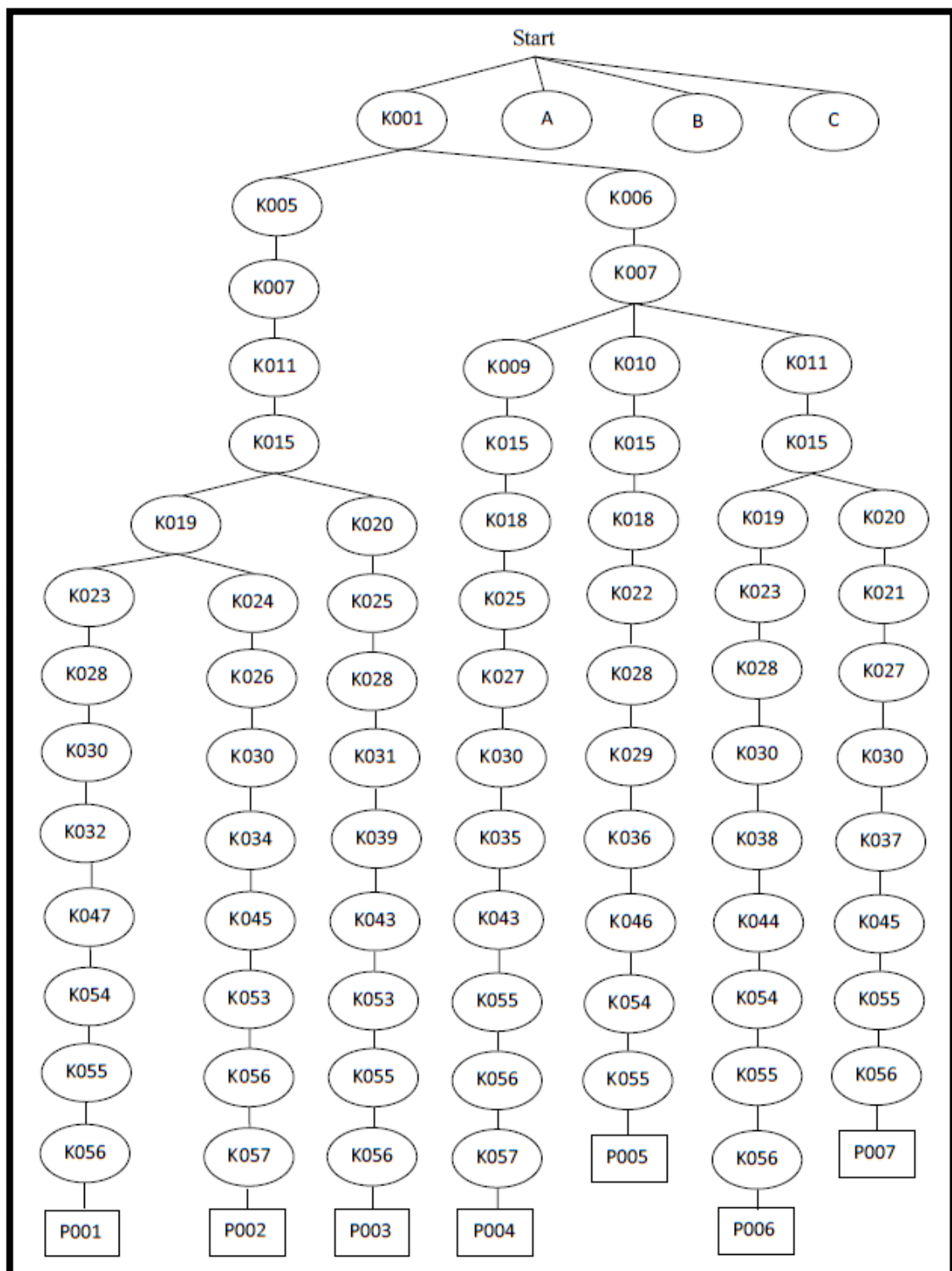


Gambar 3.1 *Dependency Diagram* Sistem Informasi Pemasaran Properti

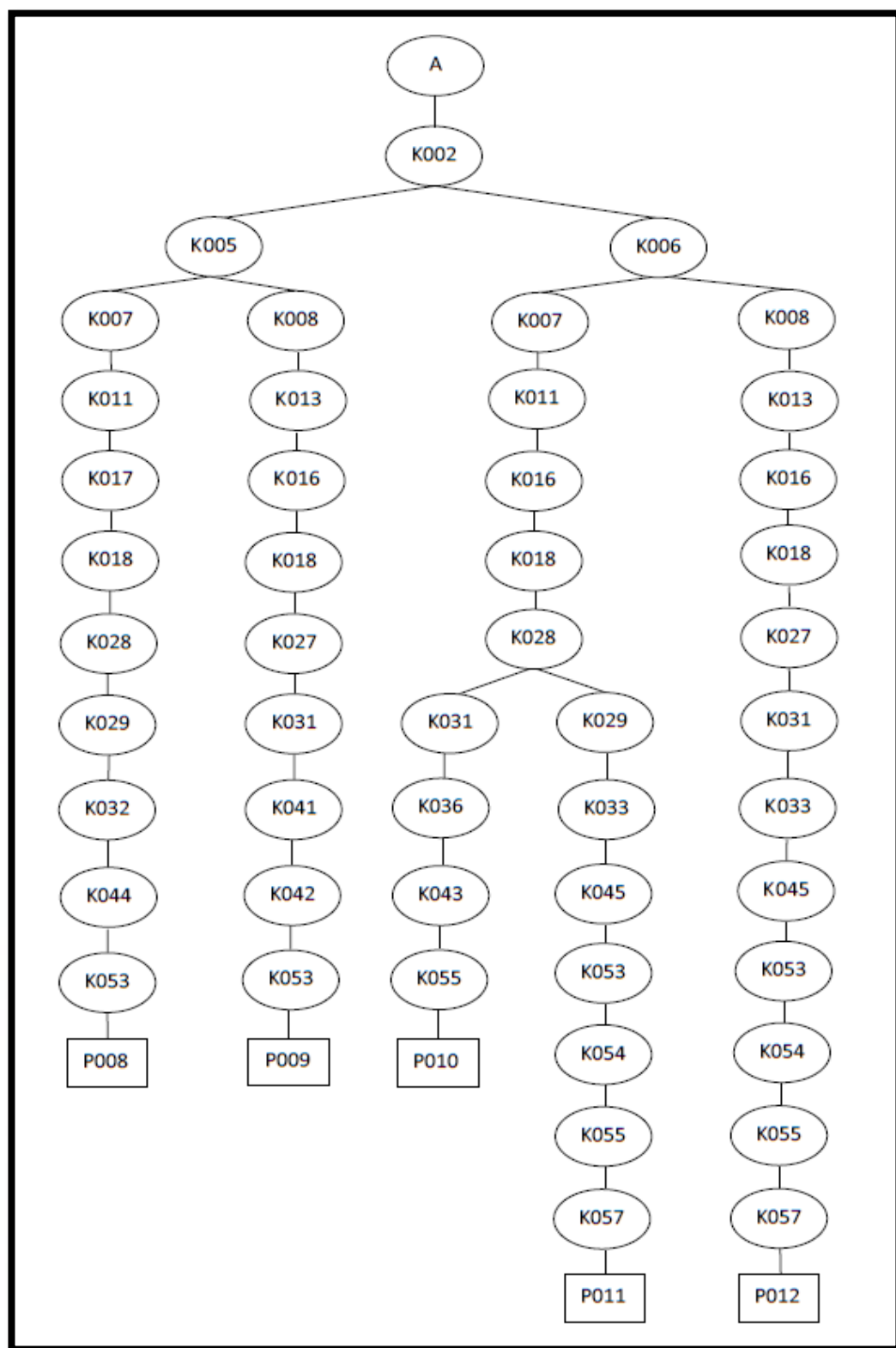
3.4 Pohon Keputusan

Suatu proses terhadap basis pengetahuan atau informasi yang didapat dari pakar, terlebih dahulu diubah kedalam bentuk pohon keputusan, sehingga didalam penyelesaian masalah lebih mudah dilakukan penelusuran untuk mendapatkan hasil simpulan akhir yang terbaik. Gambar 3.2 menunjukkan diagram pohon keputusan pada sistem pemasaran properti.

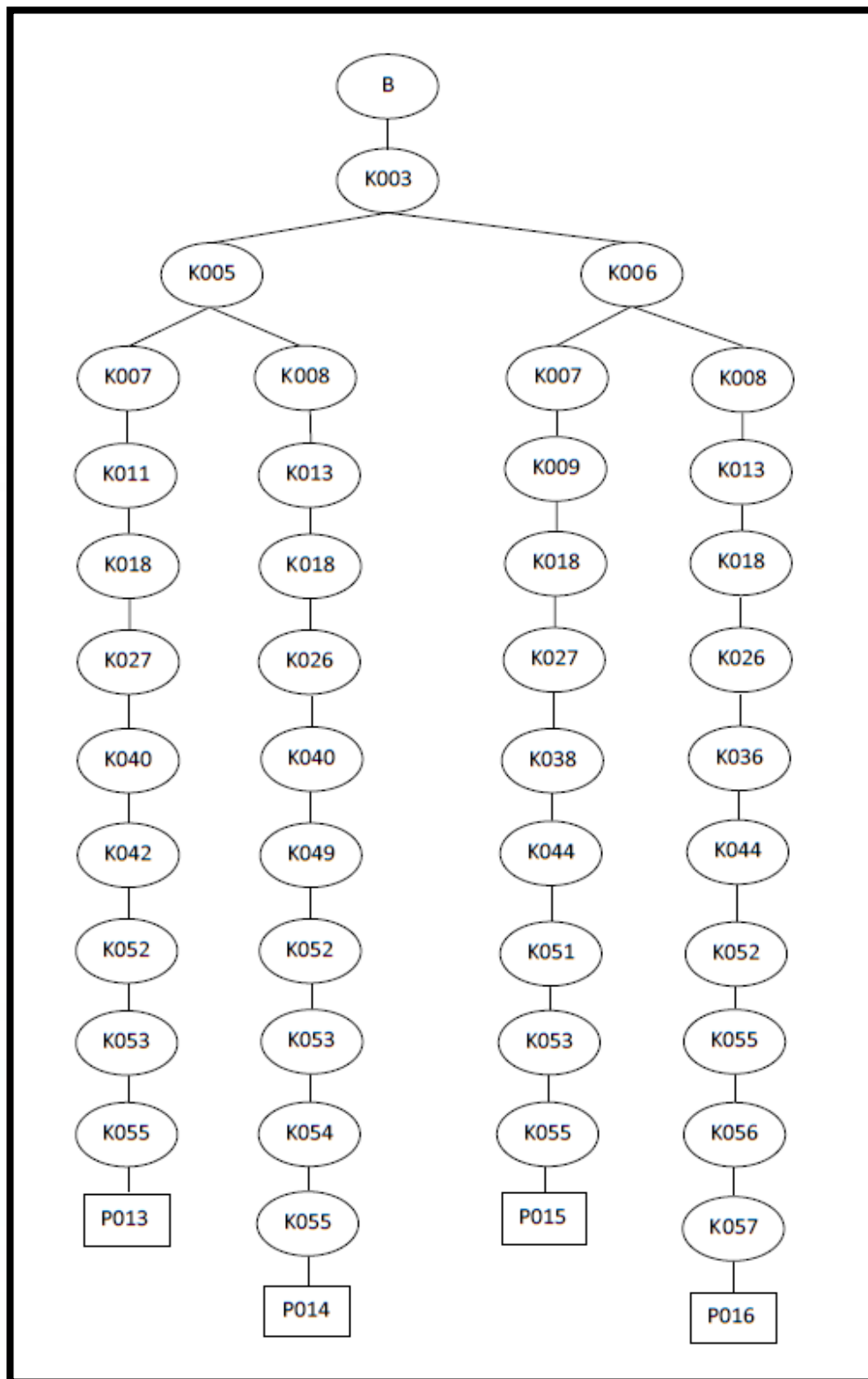




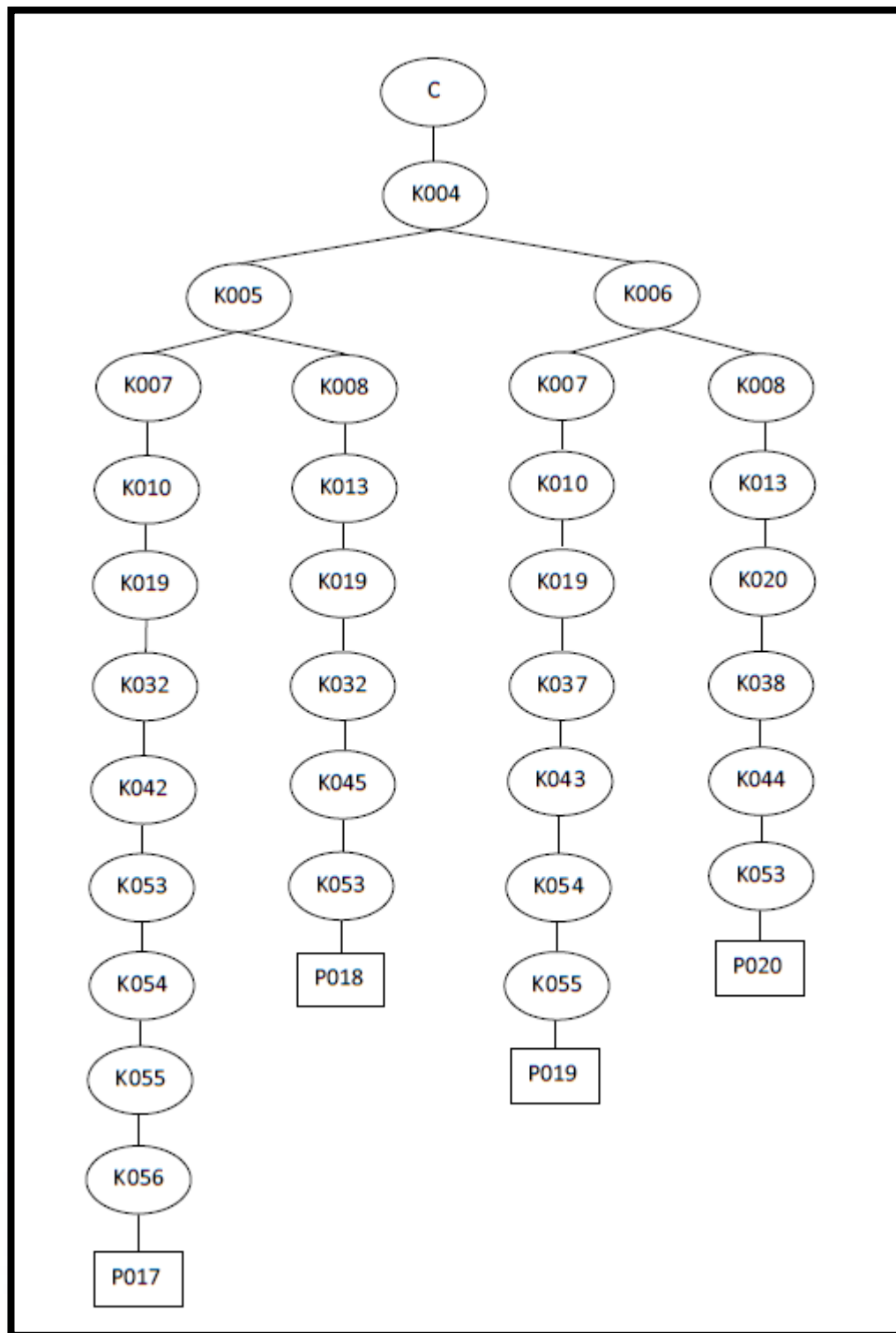
Gambar 3.2 Pohon Keputusan Pemasaran Properti



Gambar 3.3 Pohon Keputusan Pemasaran Properti (lanjutan)



Gambar 3.4 Pohon Keputusan Pemasaran Properti (lanjutan)



Gambar 3.5 Pohon Keputusan Pemasaran Properti (lanjutan)

Keterangan Kriteria:

K001 : Rumah

K002 : Ruko

K003 : Apartemen

K004 : Tanah

K005 : Jakarta

K006 : Tangerang

K007 : Jual

K008 : Sewa

K009 : < 1 Miliar

K010 : 1 Miliar – 3 Miliar

K011 : > 3Miliar

K012 : < 50 juta/tahun

K013 : >=50 juta/tahun

K014 : 1 lantai

K015 : 2 lantai

K016 : 3 lantai

K017 : 4 lantai

K018 : Kecil (< 200 meter persegi)

K019 : Sedang (200 – 400 meter persegi)

K020 : Besar (> 400 meter persegi)

K021 : Mansion

K022 : Minimalis

K023 : Tropis

K024 : Mediteranian

K025 : Asri

K026 : Fully furnished

K027 : Semi furnished

K028 : Unfurnished

K029 : PPJB (Perjanjian Pengikatan Jual Beli)

K030 : SHM (Surat Hak Milik)

K031 : SHGB (Sertifikat Hak Guna Bangun)

K032 : PT Agung Sedayu Group

K033 : PT Alam Sutera Realty Tbk

K034 : PT Intiland Development Tbk

K035 : PT Lippo Karawaci Tbk

K036 : PT Paramount Serpong

K037 : PT Summarecon Agung Tbk

K038 : Sinarmas Land

K039 : PT Harvia Unggul Pratama

K040 : Agung Podomoro Group

K041 : PT Aries Utama Ekagriya

K042 : Utara

K043 : Barat

K044 : Timur

K045 : Selatan

K046 : Tenggara

K047 : Barat Daya

K048 : Barat Laut

K049 : Timur Laut

K050 : Rendah (1 - 5)

K051 : Sedang (5 - 15)

K052 : Tinggi (>15)

K053 : Jalan Raya

K054 : Sekolah

K055 : Mall

K056 : Rumah Sakit

K057 : Universitas

Keterangan Properti:

P001 : Puri Indah J4/12A

P002 : Taman Villa Meruya C2/11

P003 : Taman Kedoya Baru No 9-10

P004 : Taman Osaka-Lippo Karawaci

P005 : Il Rosa

P006 : Kencana Loka – BSD

P007 : Amethyst

P008 : Ruko Sedayu Square

P009 : Ruko Puri Kencana J1-2J

P010 : Ruko dot com green

P011 : Ruko Crystal lane

P012 : Ruko Alam Sutera

P013 : The Peak – Sudirman

P014 : Permata Hijau Residence

P015 : Apart. Casa de Parco

P016 : Atria Paramount Residence Serpong

P017 : Kavling Puri Indah F9/26

P018 : Kavling Slipi, Jakarta Barat

P019 : PHG Sapphire Gading Serpong

P020 : Tanah BSD Bintaro

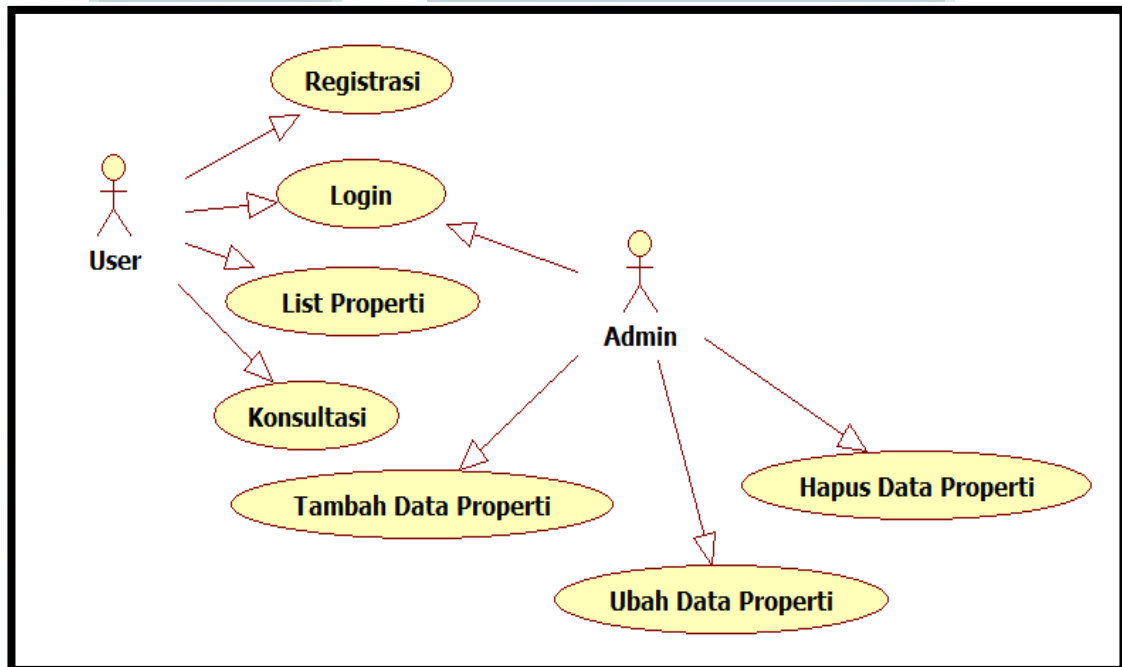
UMN

3.5 Perancangan Sistem

Perancangan sistem informasi pemasaran properti berupa *use case diagram*, *activity diagram*, *sequence diagram*, *state chart diagram*, *class diagram* dan *deployment diagram*. Software pembuatan diagram UML menggunakan StarUML.

3.5.1 Use Case Diagram

Use case diagram merupakan diagram yang menunjukkan hubungan antara aktor dan kasus penggunaan (*use case*) dalam sistem. Gambar 3.6 merupakan *use case diagram* sistem informasi pemasaran properti.



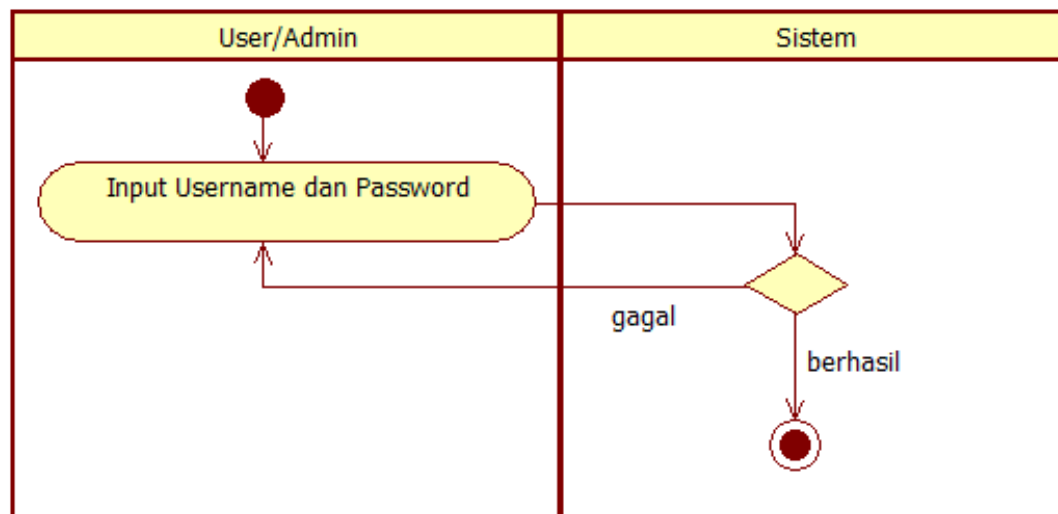
Gambar 3.6 *Use Case Diagram* Sistem Informasi Pemasaran Properti

Use case diagram sistem informasi pemasaran properti pada gambar 3.6 terdapat 2 aktor yaitu *user* dan *admin*. *User* berinteraksi dengan 4 proses yaitu

registrasi, login, list properti dan konsultasi. Admin berinteraksi dengan 4 proses yaitu login, tambah data properti, ubah data properti dan hapus data properti.

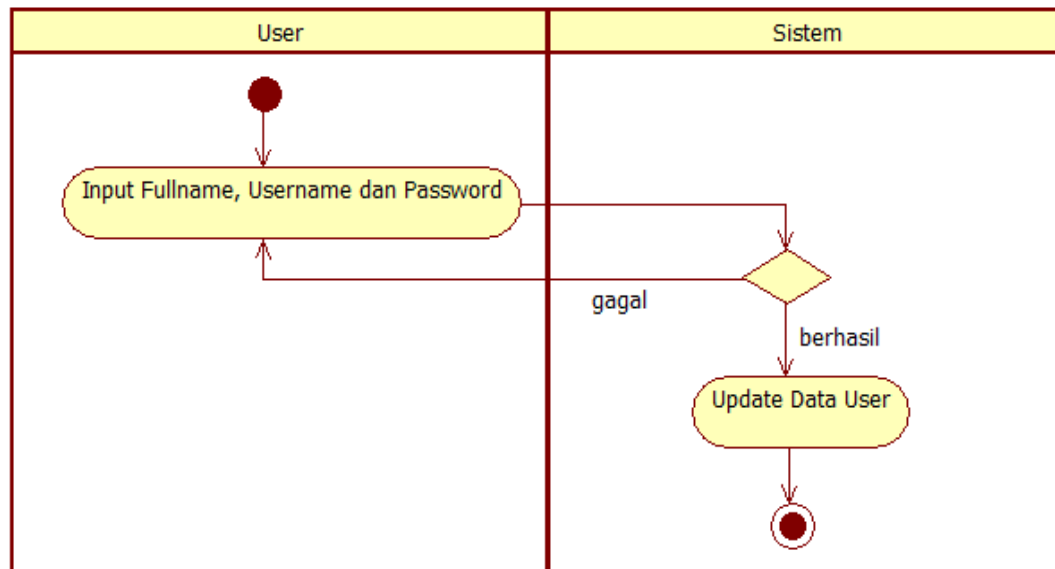
3.5.2 Activity Diagram

Activity diagram merupakan representasi grafis dari seluruh tahapan alur kerja. *Activity diagram* digunakan untuk menggambarkan *workflow* atau aliran kerja proses bisnis. Gambar 3.7 merupakan *activity diagram* untuk login.



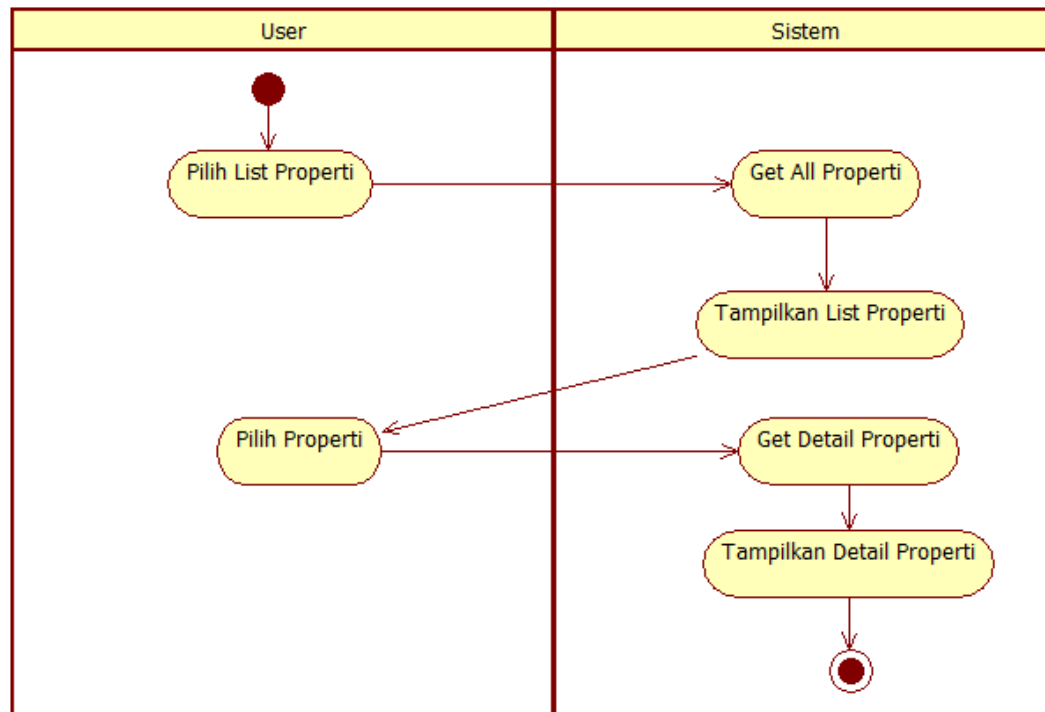
Gambar 3.7 Activity Diagram Untuk Login

User/admin dapat melakukan login. Jika login gagal, *user/admin* dapat melakukan login ulang. Jika login berhasil, *user/admin* dapat masuk ke menu.



Gambar 3.8 *Activity Diagram* Untuk Registrasi

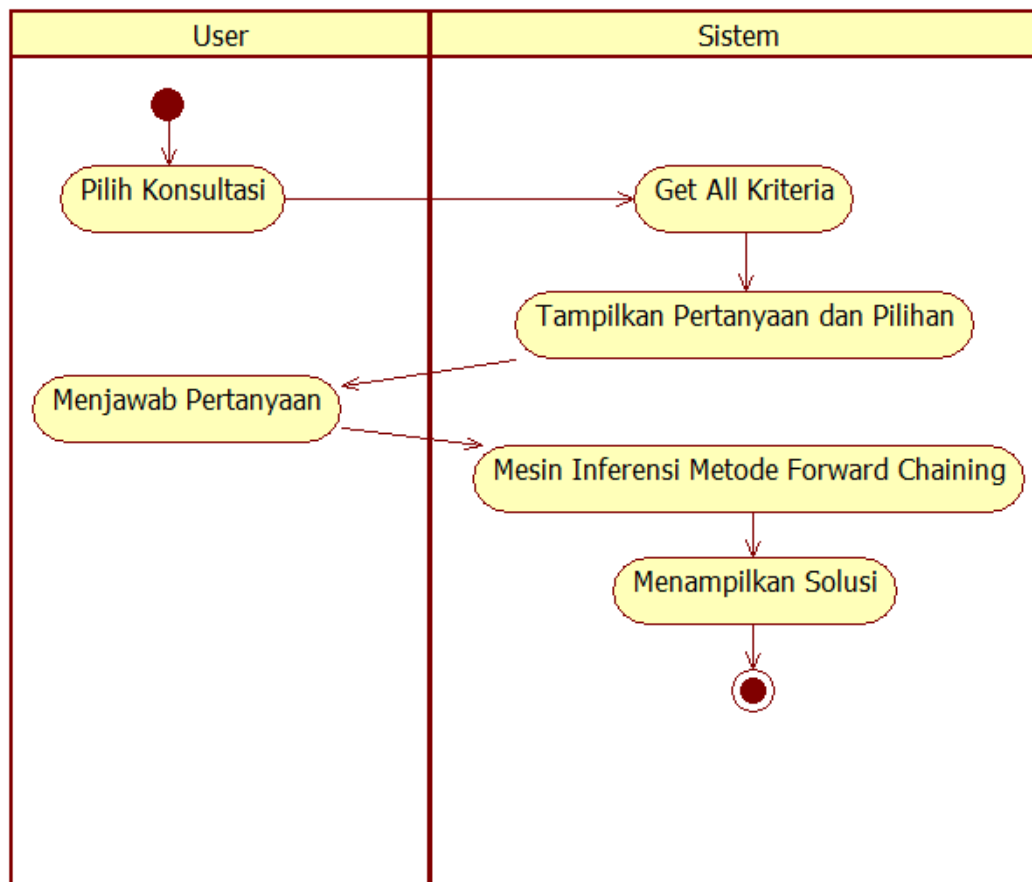
Gambar 3.8 merupakan *activity diagram* untuk registrasi. Setelah *user* melakukan registrasi, sistem melakukan verifikasi. Jika registrasi berhasil, sistem melakukan update data *user* sehingga *user* dapat login dengan melakukan input username dan password. Jika registrasi gagal, *user* dapat melakukan registrasi ulang.



Gambar 3.9 Activity Diagram Untuk List Properti

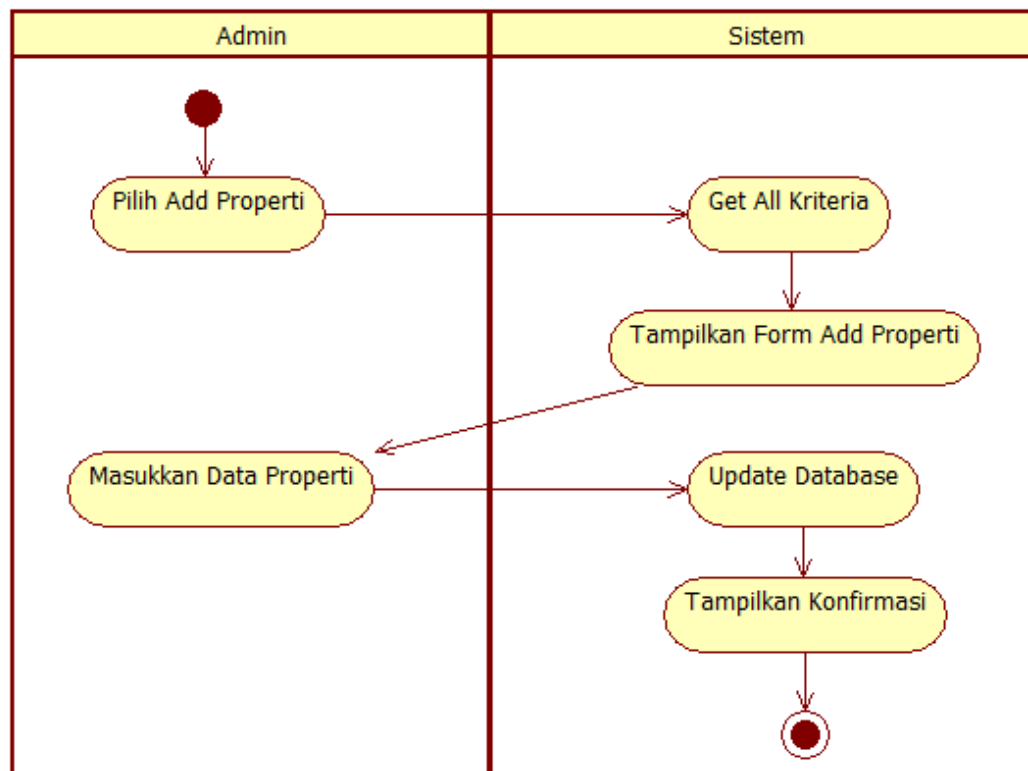
Gambar 3.9 merupakan *activity diagram* untuk list properti. Jika *user* memilih List Properti maka sistem melakukan get all properti dan menampilkan list data properti. *User* dapat memilih properti. Lalu sistem melakukan get detail properti dan menampilkan detail properti.

UMN



Gambar 3.10 *Activity Diagram* Untuk Konsultasi

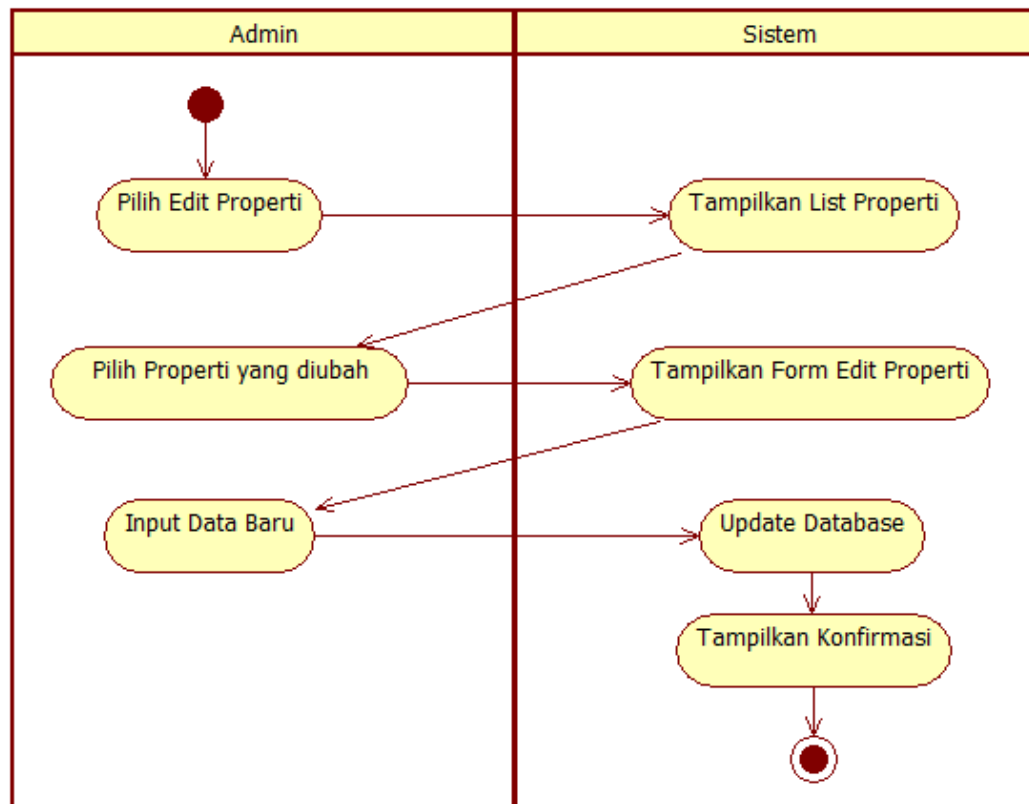
Gambar 3.10 merupakan *activity diagram* untuk konsultasi. Jika *user* memilih Konsultasi maka sistem melakukan get all kriteria dan menampilkan pertanyaan dan pilihan. Lalu *user* menjawab pertanyaan yang ditampilkan oleh sistem. Setelah *user* menjawab pertanyaan yang ditampilkan, sistem mencari solusi melalui mesin inferensi metode *forward chaining* dan menampilkan solusi tersebut.



Gambar 3.11 *Activity Diagram* Untuk Tambah Data Properti

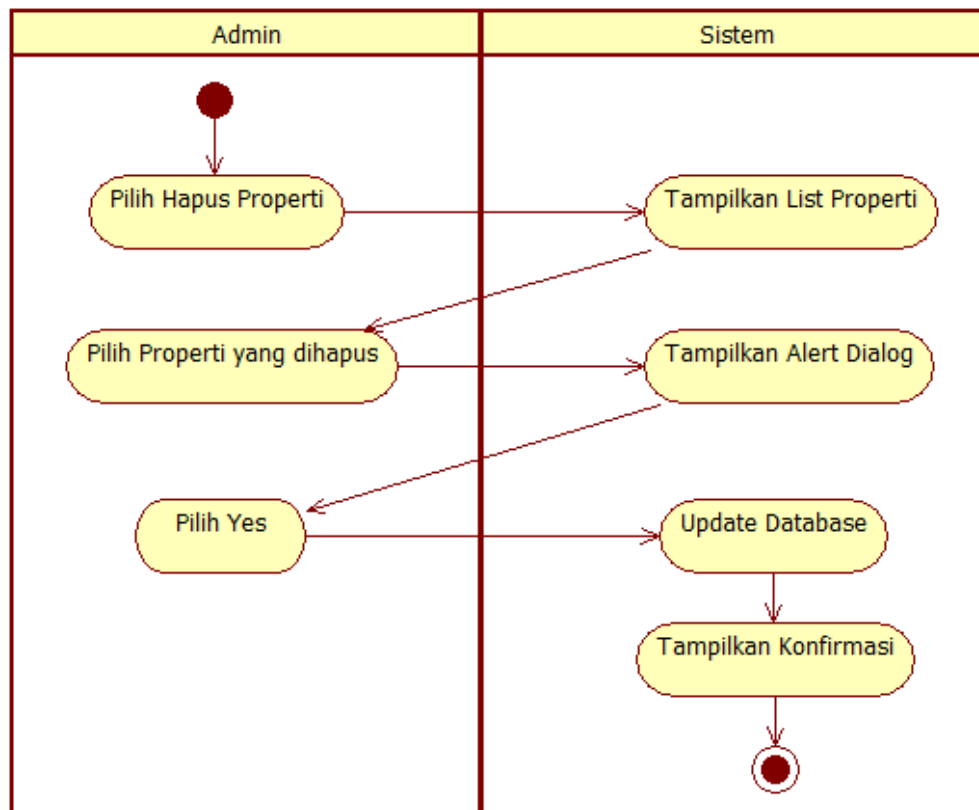
Gambar 3.11 merupakan *activity diagram* untuk tambah data properti. Jika admin memilih Tambah Properti maka sistem melakukan get all kriteria dan menampilkan form add properti. Lalu admin dapat memasukkan data properti sehingga sistem melakukan update database dan menampilkan konfirmasi ke admin.

UMN



Gambar 3.12 *Activity Diagram* Untuk Ubah Data Properti

Gambar 3.12 merupakan *activity diagram* untuk ubah data properti. Jika admin memilih Edit Properti maka sistem menampilkan list data properti. Lalu admin memilih properti yang diubah. Kemudian sistem menampilkan form edit properti. Admin melakukan input data baru ke dalam form tersebut. Setelah itu, sistem melakukan update database dan menampilkan konfirmasi ke admin.

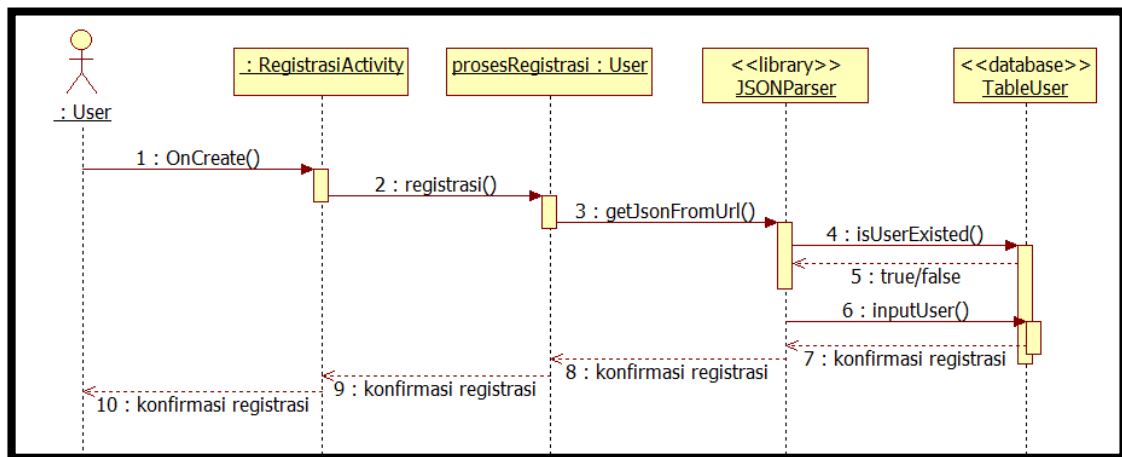


Gambar 3.13 *Activity Diagram* Untuk Hapus Data Properti

Gambar 3.13 merupakan *activity diagram* untuk hapus data properti. Jika admin memilih Hapus Properti maka sistem menampilkan list data properti. Lalu admin memilih properti yang dihapus. Kemudian sistem menampilkan alert dialog. Admin memilih Yes. Sistem melakukan update database dan menampilkan konfirmasi ke admin.

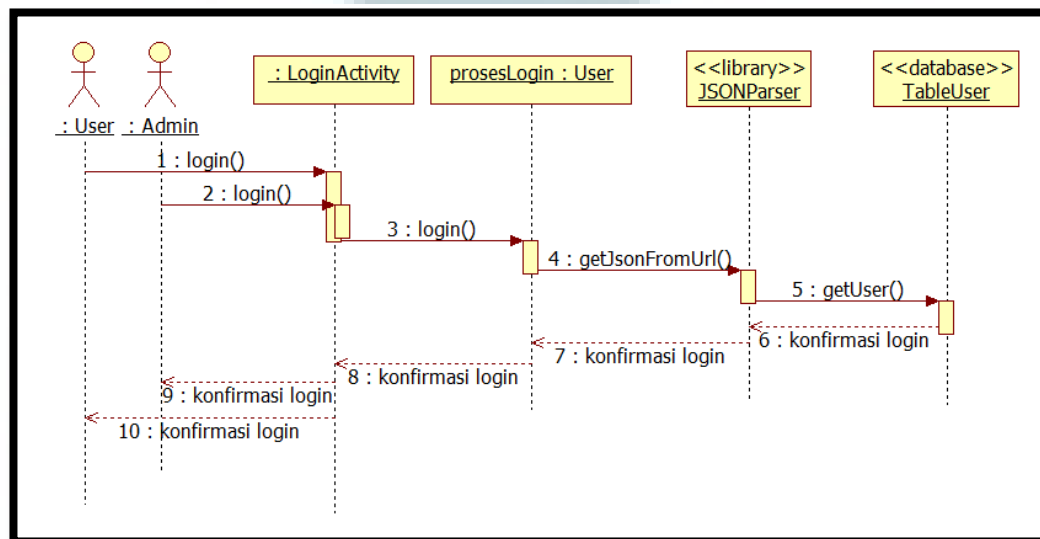
3.5.3 Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan interaksi antara obyek dan mengindikasikan komunikasi di antara obyek-obyek tersebut. Gambar 3.14 merupakan *sequence diagram* registrasi.



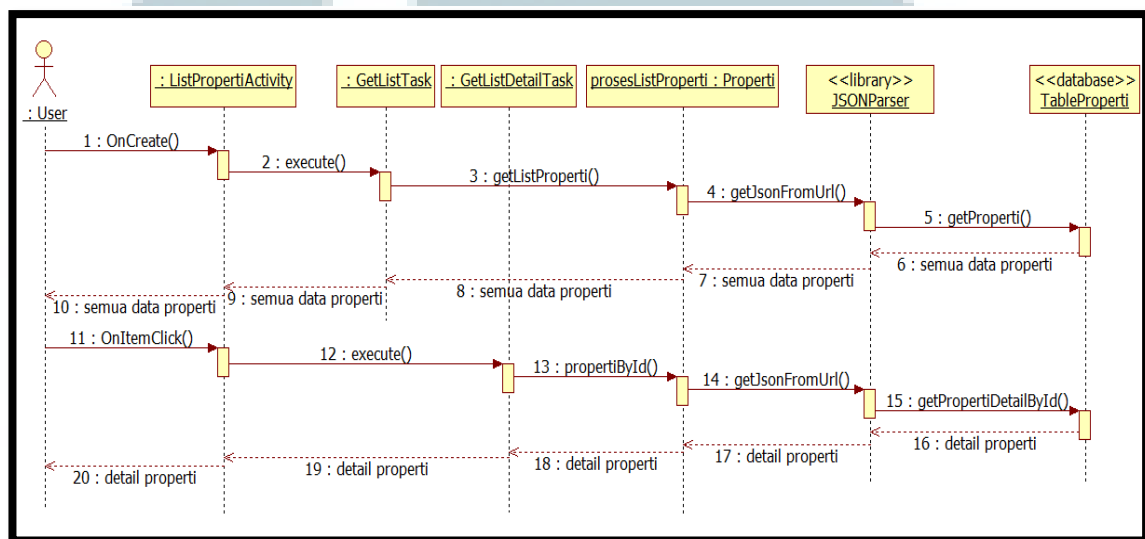
Gambar 3.14 *Sequence Diagram Registrasi*

Gambar 3.14 menunjukkan proses registrasi *user*. *User* men-trigger method *OnCreate* dengan mengklik *button* registrasi. Setelah itu *prosesRegistrasi* men-trigger method *getJsonFromUrl*. *JSONParser* melakukan query ke table *User*. Jika data berhasil atau gagal dimasukkan, *prosesRegistrasi* akan memberikan konfirmasi registrasi kepada *user* melalui *RegistrasiActivity*.



Gambar 3.15 *Sequence Diagram Login*

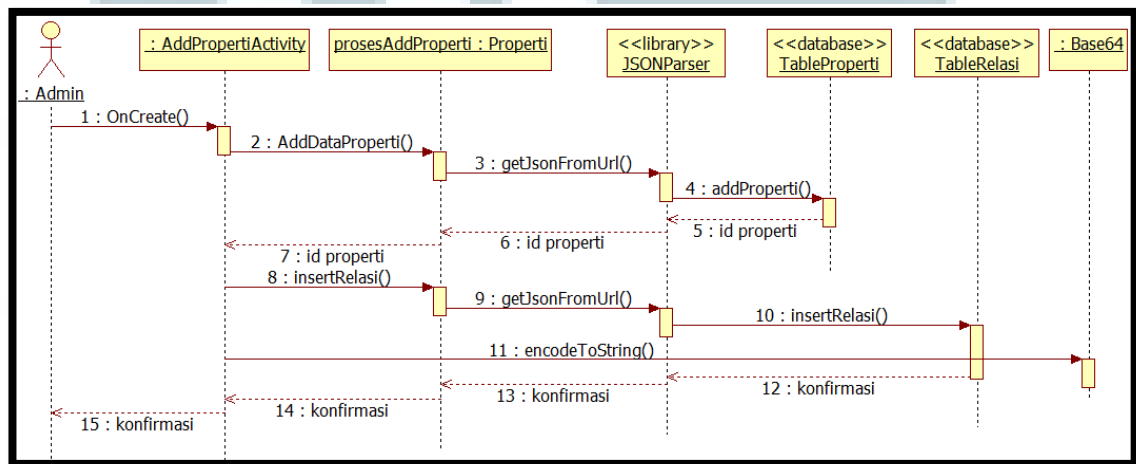
Gambar 3.15 menunjukkan proses login untuk *user* dan admin. *User/admin* memasukkan *username* dan password. Setelah itu prosesLogin men-*trigger* method *getJsonFromUrl*. *JSONParser* melakukan query ke table *User*. Kemudian, prosesLogin akan memberikan konfirmasi login kepada *user/admin* melalui *LoginActivity*.



Gambar 3.16 *Sequence Diagram* List Properti

Gambar 3.16 menunjukkan proses untuk menampilkan list (daftar) properti. *User* men-*trigger* method *OnCreate* jika masuk ke halaman *ListPropertiActivity*. *ListPropertiActivity* men-*trigger* method *execute* pada class *GetListTask*. *GetListTask* men-*trigger* method *getListProperti*. Setelah itu *prosesListProperti* men-*trigger* method *getJsonFromUrl*. *JSONParser* melakukan query ke table *Properti*. Jika berhasil, *prosesListProperti* akan menampilkan list properti kepada *user* melalui *ListPropertiActivity*.

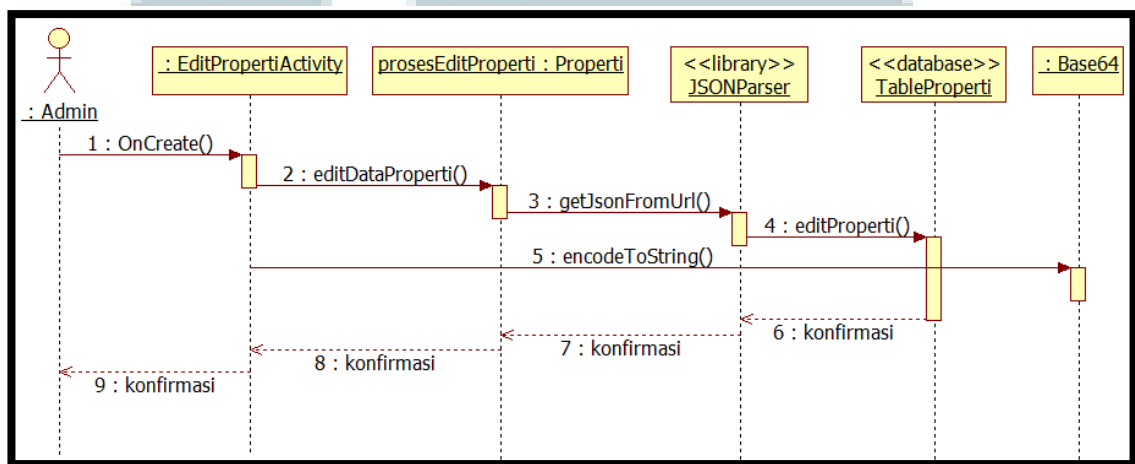
User dapat *men-trigger* method `OnItemClick` untuk mendapatkan detail properti yang dipilih dengan mengklik properti tersebut. `ListPropertiActivity` *men-trigger* method `execute` pada class `GetListDetailTask`. `GetListDetailTask` *men-trigger* method `propertiById`. Setelah itu `prosesListProperti` *men-trigger* method `getJsonFromUrl`. `JSONParser` melakukan query ke table `Properti`. Jika berhasil, `prosesListProperti` akan menampilkan detail properti berdasarkan id yang dipilih melalui `ListPropertiActivity`.



Gambar 3.17 *Sequence Diagram* Add Properti

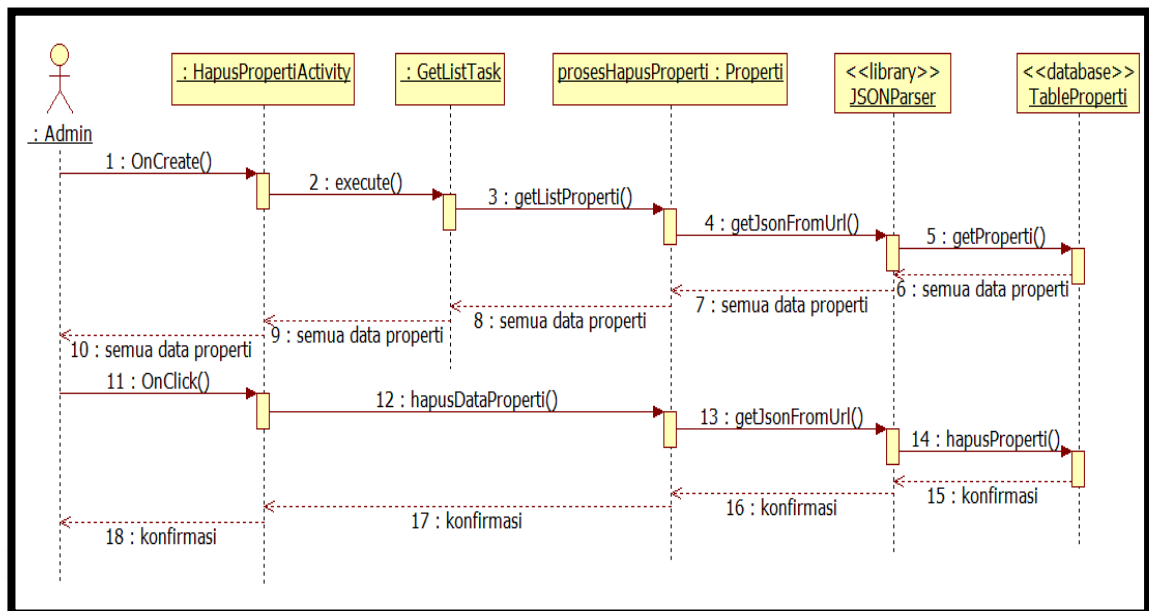
Gambar 3.17 menunjukkan proses untuk menambahkan data properti. Admin *men-trigger* method `OnCreate` dengan mengklik menu confirm. `AddPropertiActivity` *men-trigger* method `AddDataProperti`. Setelah itu `prosesAddProperti` *men-trigger* method `getJsonFromUrl`. `JSONParser` melakukan query ke table `Properti`. Jika data properti berhasil diinput, `prosesAddProperti` mengembalikan pesan berupa id properti yang diinput.

AddPropertiActivity men-trigger method insertRelasi. Setelah itu prosesAddProperti men-trigger method getJsonFromUrl. JSONParser melakukan query ke table Relasi. AddPropertiActivity men-trigger method encodeToString. Jika data berhasil diinput, prosesAddProperti akan menampilkan konfirmasi input sukses atau gagal kepada admin melalui AddPropertiActivity.



Gambar 3.18 Sequence Diagram Edit Properti

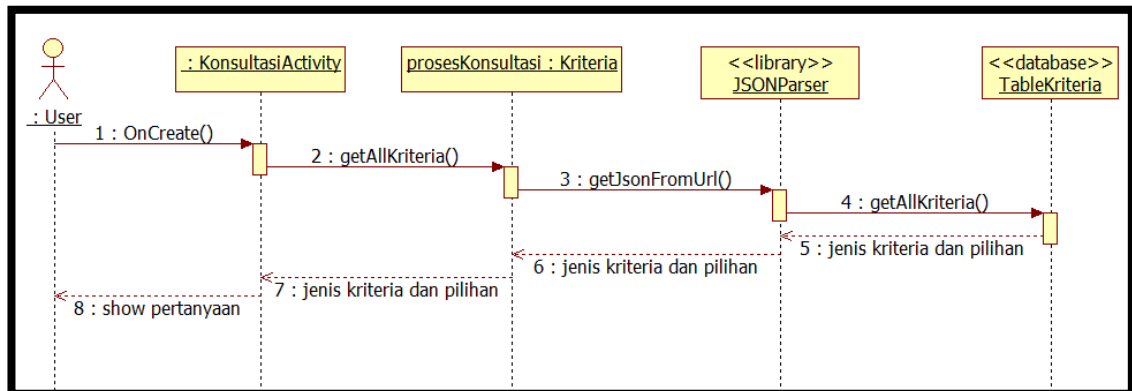
Gambar 3.18 menunjukkan proses untuk mengubah data properti. Admin men-trigger method OnCreate dengan mengklik *button* confirm. EditPropertiActivity men-trigger method editDataProperti. Setelah itu prosesEditProperti men-trigger method getJsonFromUrl. JSONParser melakukan query ke table Properti. EditPropertiActivity men-trigger method encodeToString. Jika data properti berhasil diubah, prosesEditProperti akan menampilkan konfirmasi edit sukses atau gagal kepada admin melalui EditPropertiActivity.



Gambar 3.19 *Sequence Diagram* Hapus Properti

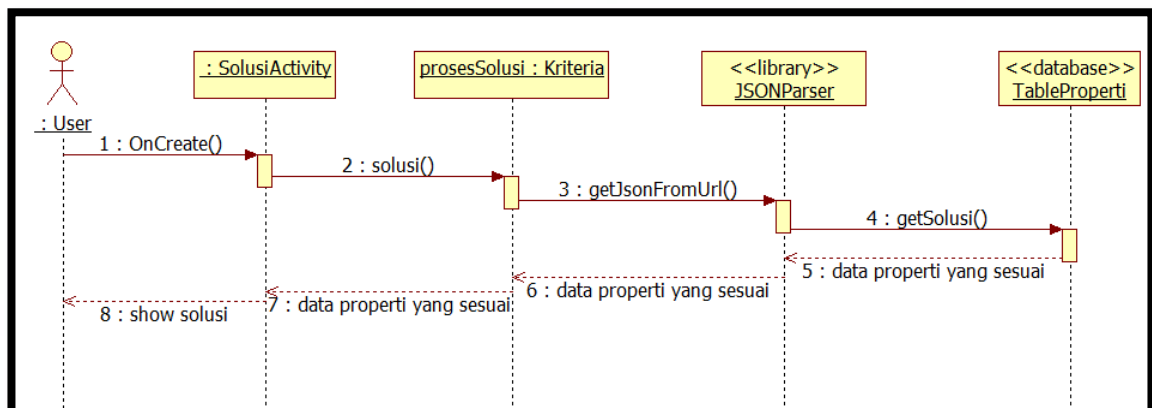
Gambar 3.19 menunjukkan proses untuk menghapus data properti. Admin men-trigger method `OnCreate` jika masuk ke halaman `HapusPropertiActivity`. `HapusPropertiActivity` men-trigger method `execute` pada class `GetListTask`. `GetListTask` men-trigger method `getListProperti`. Setelah itu `prosesHapusProperti` men-trigger method `getJsonFromUrl`. `JSONParser` melakukan query ke table `Properti`. Jika berhasil, `prosesHapusProperti` akan menampilkan list properti kepada admin melalui `HapusPropertiActivity`.

Admin men-trigger method `OnClick` dengan memilih properti yang dihapus. `HapusPropertiActivity` men-trigger method `hapusDataProperti`. Setelah itu `prosesHapusProperti` men-trigger method `getJsonFromUrl`. `JSONParser` melakukan query ke table `Properti`. Jika data properti berhasil dihapus, `prosesHapusProperti` akan menampilkan konfirmasi hapus sukses atau gagal kepada admin melalui `HapusPropertiActivity`.



Gambar 3.20 *Sequence Diagram* Konsultasi

Gambar 3.20 menunjukkan proses konsultasi untuk menampilkan pertanyaan. *User* men-trigger method *OnCreate* jika masuk ke halaman *KonsultasiActivity*. *KonsultasiActivity* men-trigger method *getAllKriteria*. Setelah itu *prosesKonsultasi* men-trigger method *getJsonFromUrl*. *JSONParser* melakukan query ke table *Kriteria*. Jika berhasil, *prosesKonsultasi* akan menampilkan pertanyaan kepada *user* melalui *KonsultasiActivity*.



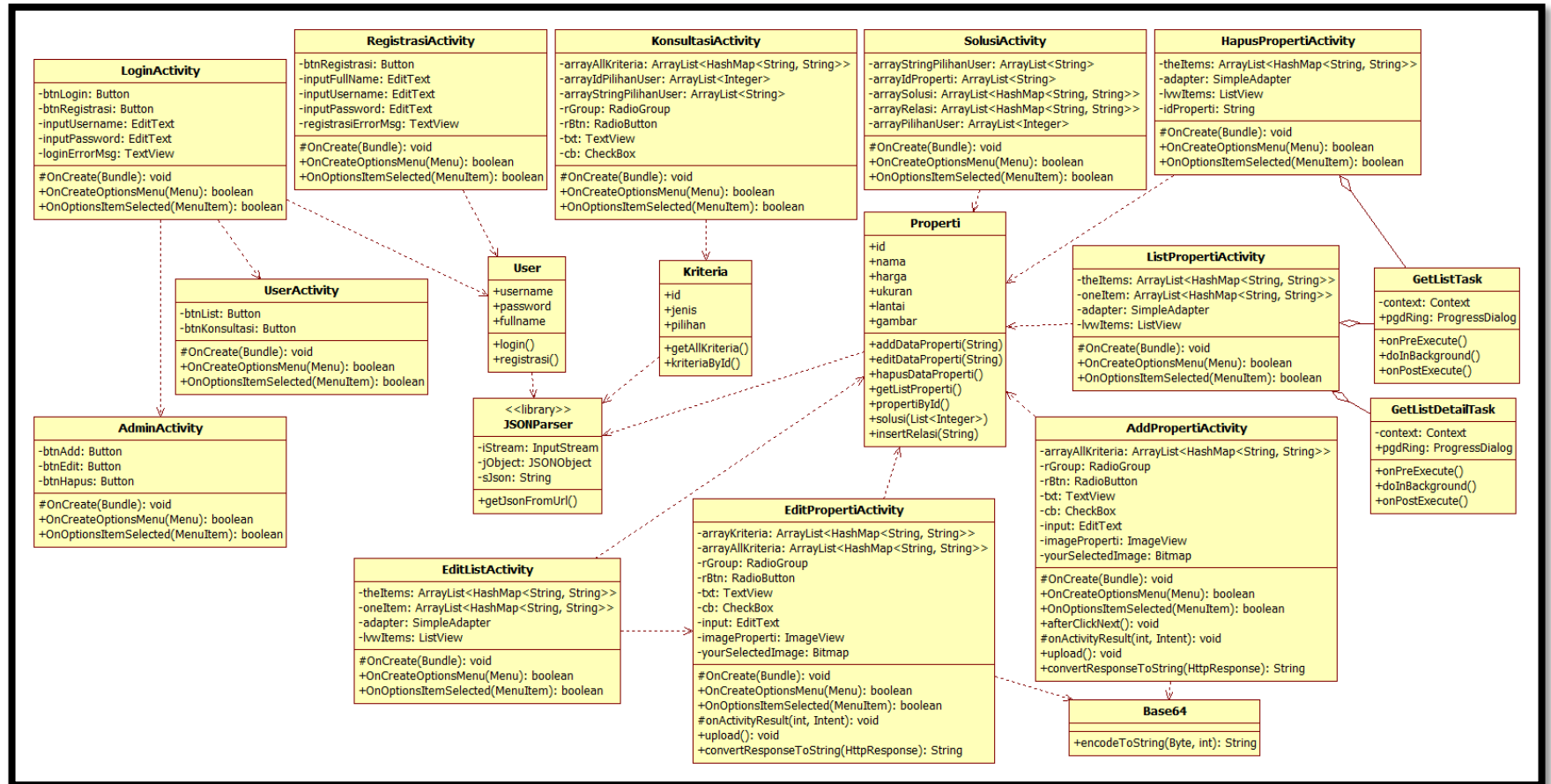
Gambar 3.21 *Sequence Diagram* Solusi

Gambar 3.21 menunjukkan proses untuk menampilkan solusi. *User* men-*trigger* method *OnCreate* setelah *user* menjawab pertanyaan-pertanyaan yang ditampilkan oleh sistem jika masuk ke halaman solusi. *SolusiActivity* men-*trigger* method *solusi*. Setelah itu *prosesSolusi* men-*trigger* method *getJsonFromUrl*. *JSONParser* melakukan query ke table *Properti*. Jika berhasil, *prosesSolusi* akan menampilkan solusi kepada *user* melalui *SolusiActivity*.

3.5.4 Class Diagram

Class diagram merupakan diagram yang menggambarkan kelas, atribut dan operasi dalam sebuah sistem dan hubungannya antara satu dengan yang lain. Gambar 3.22 merupakan *class diagram* dari sistem informasi pemasaran properti.

UMN

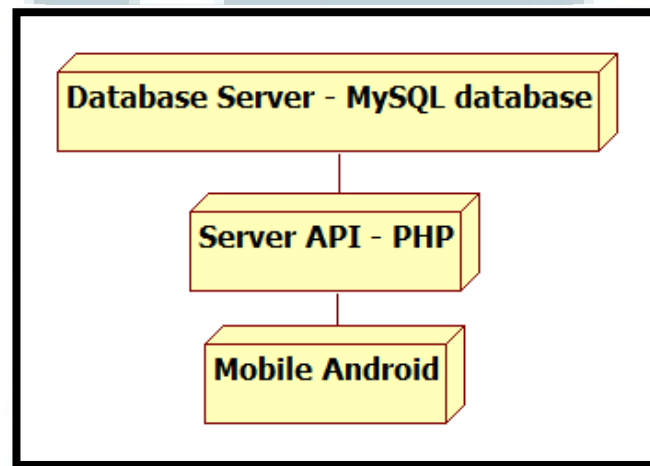


Gambar 3.22 Class Diagram Sistem Informasi Pemasaran Properti

Berdasarkan gambar 3.22 sistem informasi pemasaran properti terdiri dari 18 *class* yaitu LoginActivity, UserActivity, AdminActivity, RegistrasiActivity, ListPropertiActivity, KonsultasiActivity, SolusiActivity, AddPropertiActivity, EditListActivity, EditPropertiActivity, HapusPropertiActivity, GetListTask, GetListDetailTask, Base64, JSONParser, User, Kriteria dan Properti. Terdapat hubungan *dependency* dan *aggregation* pada kelas-kelas sistem informasi pemasaran properti.

3.5.5 Deployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem. Gambar 3.23 merupakan *deployment diagram* sistem informasi pemasaran properti.



Gambar 3.23 *Deployment Diagram* Sistem Informasi Pemasaran Properti

3.6 Perancangan Database

Perancangan database merupakan perancangan tabel dan menentukan hubungan (relasi) antar tabel tersebut. Perancangan database berupa struktur tabel.

3.6.1 Struktur Tabel

Terdapat tabel user untuk menyimpan data *user* sehingga *user* dapat login.

Tabel 3.1 adalah struktur tabel user.

Tabel 3.1 Struktur Table User

Nama Field	Tipe Data	Ukuran	Keterangan
Uid	Int	11	Menyimpan nomor id <i>user</i> (<i>auto increment</i>) - Primary Key
Unique_id	Char	23	Menyimpan unique id
Username	Char	10	Menyimpan username
Password	Char	80	Menyimpan password yang di enkripsi
Full_name	Char	20	Menyimpan nama lengkap <i>user</i>
Salt	Char	10	Menyimpan salt untuk enkripsi dan dekripsi password
Created_at	datetime	-	Menyimpan tanggal dan waktu <i>user</i> registrasi
Tipe	Char	10	Menyimpan tipe <i>user</i> atau admin

Terdapat tabel properti untuk menyimpan data properti. Tabel 3.2 adalah struktur tabel properti.

Tabel 3.2 Struktur Table Properti

Nama Field	Tipe Data	Ukuran	Keterangan
Id_properti	Int	11	Menyimpan nomor id properti (<i>auto increment</i>) - Primary Key
Nama	Char	50	Menyimpan nama properti
Harga	Char	20	Menyimpan harga properti
Ukuran	Char	25	Menyimpan ukuran properti
Lantai	Int	11	Menyimpan nomor lantai apartemen
Gambar	Char	50	Menyimpan <i>path</i> file gambar properti

Terdapat tabel kriteria untuk menyimpan data kriteria. Tabel 3.3 adalah struktur tabel kriteria.

Tabel 3.3 Struktur Tabel Kriteria

Nama Field	Tipe Data	Ukuran	Keterangan
Id_kriteria	Int	11	Menyimpan nomor id kriteria (<i>auto increment</i>) - Primary Key
Jenis_kriteria	Char	30	Menyimpan jenis kriteria
Pilihan	Char	70	Menyimpan pilihan kriteria

Terdapat tabel relasi untuk menyimpan relasi antara properti dan kriteria.

Tabel 3.4 adalah struktur tabel relasi.

Tabel 3.4 Struktur Tabel Relasi

Nama Field	Tipe Data	Ukuran	Keterangan
Id_kriteria	Int	11	Menyimpan nomor id kriteria - Foreign Key
Id_properti	Int	11	Menyimpan nomor id properti - Foreign Key

3.7 Desain Antarmuka

Desain antarmuka digunakan untuk membuat tampilan halaman pada sistem.

Halaman utama digunakan untuk login. Pada halaman utama terdapat menu exit untuk keluar dari sistem. Gambar 3.24 merupakan desain antarmuka halaman utama.

Icon Sistem Pakar Properti Exit

Login

Username

Password

(Teks konfirmasi jika gagal login)

Login

Registrasi

Gambar 3.24 Desain Antarmuka Halaman Utama

Jika button registrasi di klik, maka muncul halaman registrasi. Gambar 3.25 merupakan desain antarmuka halaman registrasi.

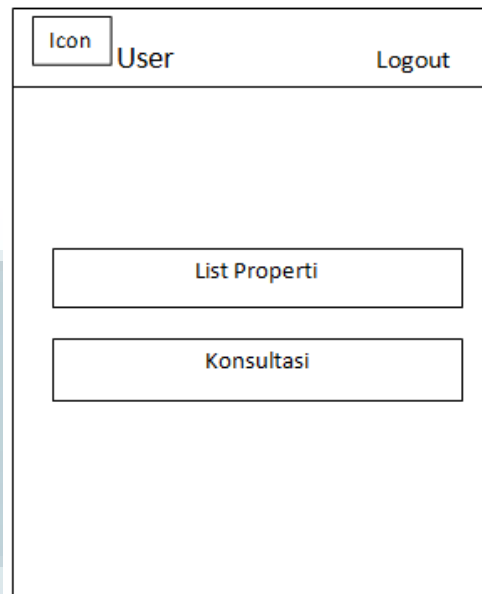


The image shows a mobile application registration screen. At the top, there is a header bar with a small icon on the left, the title "Registrasi" in the center, and a "Back" button on the right. Below the header, the title "Registrasi" is repeated in a larger font. The form contains three input fields labeled "Fullname", "Username", and "Password". Below these fields is a line of text: "(Teks konfirmasi jika gagal registrasi)". At the bottom of the form is a button labeled "Registrasi". The entire form is overlaid on a large, faint, light blue watermark of the UMN logo.

Gambar 3.25 Desain Antarmuka Halaman Registrasi

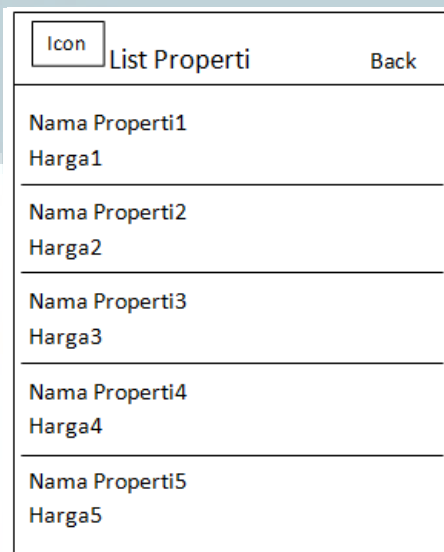
Jika *user* berhasil login, maka muncul halaman *user*. Gambar 3.26 merupakan desain antarmuka halaman *user*.

UMN



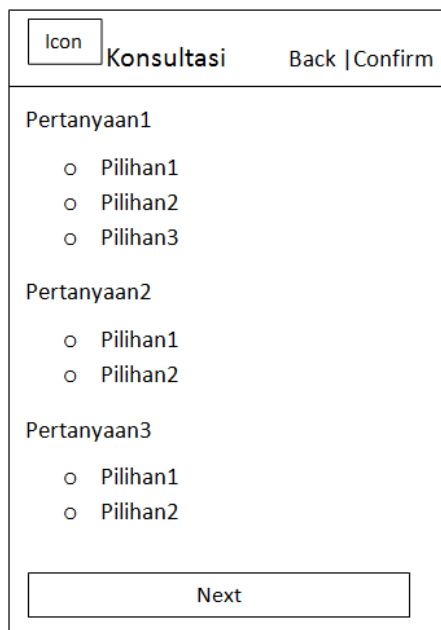
Gambar 3.26 Desain Antarmuka Halaman *User*

Jika *user* mengklik button List Properti, maka muncul halaman list properti berupa *list view*. Gambar 3.27 merupakan desain antarmuka halaman list properti.



Gambar 3.27 Desain Antarmuka Halaman List Properti

Jika *user* mengklik button Konsultasi, maka muncul halaman konsultasi. Gambar 3.28 merupakan desain antarmuka halaman konsultasi.



Icon

Konsultasi

Back | Confirm

Pertanyaan1

☐ Pilihan1
 ☐ Pilihan2
 ☐ Pilihan3

Pertanyaan2

☐ Pilihan1
 ☐ Pilihan2

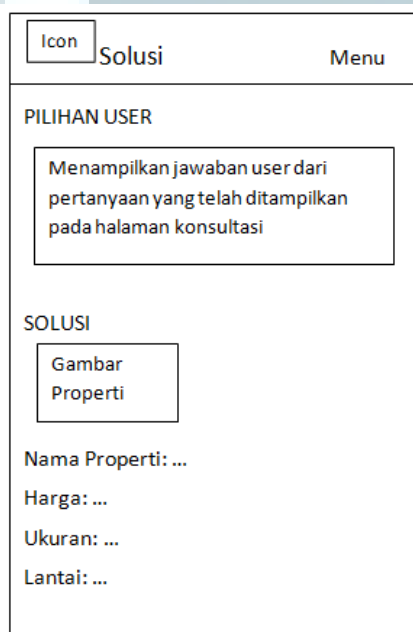
Pertanyaan3

☐ Pilihan1
 ☐ Pilihan2

Next

Gambar 3.28 Desain Antarmuka Halaman Konsultasi

Setelah *user* melakukan konsultasi, maka muncul halaman solusi. Gambar 3.29 merupakan desain antarmuka halaman solusi.



Icon

Solusi

Menu

PILIHAN USER

Menampilkan jawaban user dari pertanyaan yang telah ditampilkan pada halaman konsultasi

SOLUSI

Gambar Properti

Nama Properti: ...

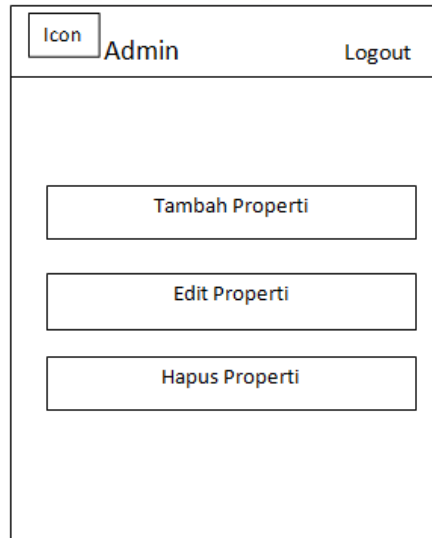
Harga: ...

Ukuran: ...

Lantai: ...

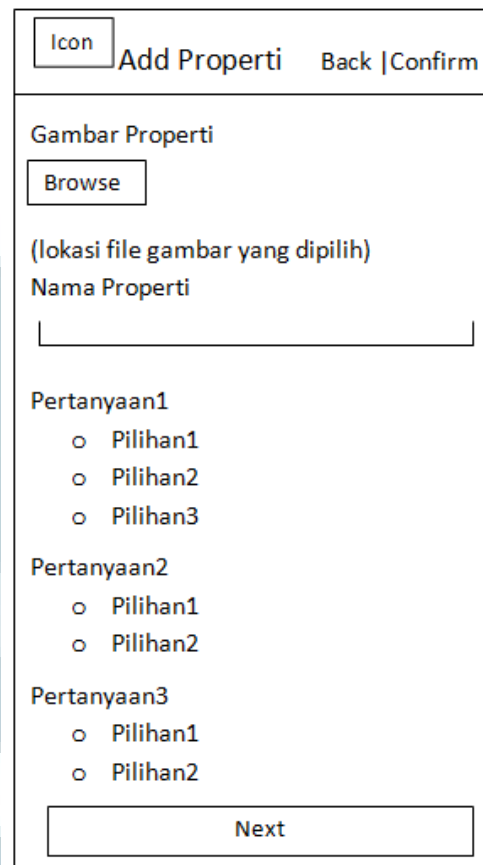
Gambar 3.29 Desain Antarmuka Halaman Solusi

Jika admin berhasil login, maka muncul halaman admin. Gambar 3.30 merupakan desain antarmuka halaman admin.



Gambar 3.30 Desain Antarmuka Halaman Admin

Jika admin mengklik button Tambah Properti, maka muncul halaman add properti. Gambar 3.31 merupakan desain antarmuka halaman add properti.



Icon Add Properti Back | Confirm

Gambar Properti

Browse

(lokasi file gambar yang dipilih)

Nama Properti

Pertanyaan1

- ☐ Pilihan1
- ☐ Pilihan2
- ☐ Pilihan3

Pertanyaan2

- ☐ Pilihan1
- ☐ Pilihan2

Pertanyaan3

- ☐ Pilihan1
- ☐ Pilihan2

Next

Gambar 3.31 Desain Antarmuka Halaman Add Properti

Jika admin mengklik button Edit Properti, maka muncul halaman list properti. Admin memilih properti yang akan diubah. Setelah memilih properti yang akan diubah, muncul halaman edit properti. Gambar 3.32 merupakan desain antarmuka halaman edit properti.

Gambar 3.32 Desain Antarmuka Halaman Edit Properti

Jika admin mengklik button Hapus Properti, maka muncul halaman hapus properti yang menampilkan daftar properti dengan menggunakan *list view*. Admin memilih properti yang akan dihapus. Setelah memilih properti yang akan dihapus, muncul *alert dialog*. Gambar 3.33 merupakan desain antarmuka halaman hapus properti dan *alert dialog* nya.

Gambar 3.33 Desain Antarmuka Halaman Hapus Properti dan *Alert Dialog*