

BAB III METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di sekitar kampus Universitas Muhammadiyah Yogyakarta. Adapun waktu penelitian Sistem Informasi Pencarian Kost di Sekitar Kampus UMY Berbasis *Android* ini dilaksanakan dari bulan Mei 2017 – Agustus 2017.

3.2 Alat dan Bahan

3.2.1 Software

Untuk membangun sistem berbasis *Android*, pada penelitian ini dibutuhkan beberapa *software*, yaitu:

1. *Android Studio* sebagai *software* pengembang aplikasi.
2. *MySQL* sebagai *database* yang digunakan aplikasi.
3. *FileZilla* digunakan untuk transfer data dari *local* ke *hosting database*.
4. *CorelDraw* digunakan untuk merancang antarmuka sistem.
5. *Microsoft Visio* digunakan untuk merancang *use case diagram*, *activity diagram*, dan *class diagram*.

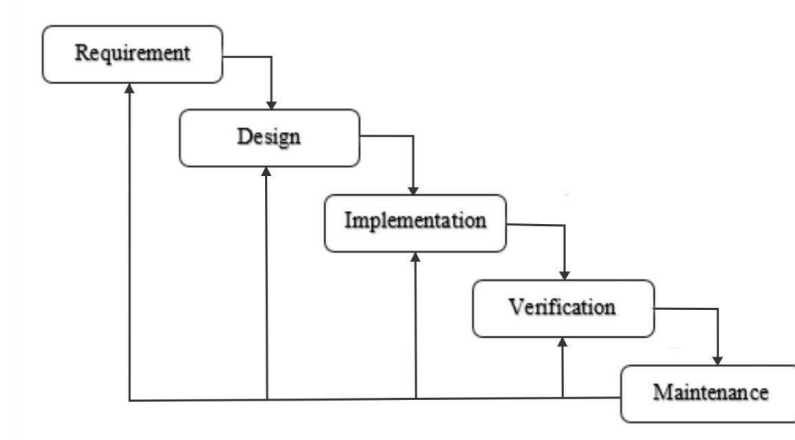
3.2.2 Hardware

Selain perangkat lunak (*software*), dibutuhkan pula perangkat keras (*hardware*) yang akan digunakan untuk mendukung pengembangan sistem informasi kost berbasis *android*, yaitu:

1. Laptop, spesifikasi 64bit *architecture*, Sistem Operasi Windows 10, 8GB RAM.
2. *Smartphone*, Sistem Operasi MIUI 8, 2GB RAM, 5 inci layar untuk menjalankan dan menguji aplikasi.

3.3 Langkah Penelitian

Dalam penelitian ini, pengembangan aplikasi menggunakan metode *System Development Life Cycle (SDLC) model waterfall*. Model ini merupakan model pengembangan *software* paling tua dan paling banyak digunakan. Model *waterfall* mengusung pengembangan *software* secara sistematis atau sekuensial. Inti dari model *waterfall* ini adalah pengerjaan secara berurutan atau secara linear. Model *waterfall* ini memiliki 5 tahapan, yaitu *requirement analysis, system design, implementation, testing dan maintenance*. Gambaran metode *waterfall* dapat dilihat pada gambar 3.1.



Gambar 3.1 Development Life Cycle (SDLC) model *waterfall*

3.3.1 Analisis Kebutuhan

Didalam tahapan ini dilakukan analisis terhadap seluruh kebutuhan *software* termasuk kegunaan *software* yang diharapkan oleh pengguna. Informasi ini dapat diperoleh dengan cara wawancara, survey atau diskusi. Informasi tersebut kemudian dianalisis sehingga mendapatkan dokumentasi kebutuhan pengguna yang akan digunakan dalam tahapan berikutnya.

Pada tahap ini dilakukan interview dengan mahasiswa dan diketahui bahwa dibutuhkan aplikasi yang dapat menampilkan informasi tentang penyewaan kost yang *up-to-date* serta lokasi kost dan informasi pemilik kost yang dapat membantu

mahasiswa lebih cepat dalam menemukan kost yang sesuai dengan apa yang mereka inginkan.

3.3.2 Perancangan Sistem

Berdasarkan analisis kebutuhan yang telah dilakukan, perancangan sistem dilakukan dengan membangun *use case* diagram, *class* diagram, *activity* diagram pada setiap proses menggunakan *software Microsoft Visio 2013*, kemudian dilakukan pengembangan antarmuka sistem dengan menggunakan *software CorelDraw*. Tahap ini bertujuan untuk memberikan gambaran sistem yang akan dikerjakan serta bagaimana rancangan antarmuka sistem.

3.3.3 Implementasi

Tahap ini merupakan tahap implementasi rancangan yang telah dibangun sebelumnya, implementasi perangkat lunak meliputi desain dan fungsi sistem berdasarkan hasil analisis, pengembangan dilakukan dengan menyusun kode-kode program. Dalam tahap ini pengembangan dilakukan dengan menggunakan bahasa *Java* dan menggunakan *software Android Studio*. Hasil dari tahap implementasi ini adalah nantinya merupakan sistem yang siap untuk dioperasikan.

3.3.4 Testing

Pada tahap ini sistem akan diuji dengan metode *black box testing*. Tujuan dari pengujian ini adalah untuk memastikan fungsi dapat berjalan sesuai dengan persyaratan perangkat lunak. Setelah sistem diuji, maka sistem siap disampaikan kepada *client* dan siap untuk dioperasikan.

3.3.5 Maintenance

Setelah sistem dibangun dan didistribusikan ke *client* untuk digunakan, masih ada kemungkinan sistem memiliki bug atau error. Seperti yang kita ketahui saat ini sangat banyak vendor dari *smartphone android* mengeluarkan handphone dengan spesifikasi yang berbeda-beda. Agar sistem dapat berjalan di semua *smarthphone*

android maka dibutuhkan *maintenance* terhadap *software* agar tetap kompatibel dengan versi sistem operasi *android* dan *hardware* yang berbeda

3.4 Analisis Kebutuhan

3.4.1 Analisis Kebutuhan Fungsional

Berikut ini merupakan kebutuhan-kebutuhan fungsional atau kebutuhan-kebutuhan yang berkaitan langsung dengan sistem. Kebutuhan fungsional dari aplikasi ini yaitu:

1. Mendapatkan informasi tentang kost yang berada di sekitar kampus UMY.
2. Dapat memberikan informasi tentang jumlah kamar yang masih tersedia.
3. Dapat memberikan informasi tentang harga penyewaan kost.
4. Dapat menunjukkan lokasi kost di dalam peta.
5. Melihat tampilan peta dalam bentuk digital.
6. Dapat terintegrasi dengan aplikasi *Google Maps*.

3.4.2 Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang tidak secara langsung terkait dengan fitur-fitur yang berada didalam sistem. Berikut merupakan kebutuhan non-fungsional meliputi:

1. Aksesibilitas: Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat diakses oleh mahasiswa ataupun pendatang yang ingin mencari kost di sekitar kampus UMY.
2. Keamanan: Pengelolaan sistem meliputi perubahan arsitektur sistem, menambah, mengganti dan menghapus data hanya dapat dilakukan oleh pengembang.
3. Bahasa: Tampilan antarmuka sistem ditampilkan dalam Bahasa Indonesia.

3.4.3 Analisis Kebutuhan Data

Data yang dikumpulkan berdasar jenis kost yaitu kost putra atau kost putri. Data yang diambil yaitu:

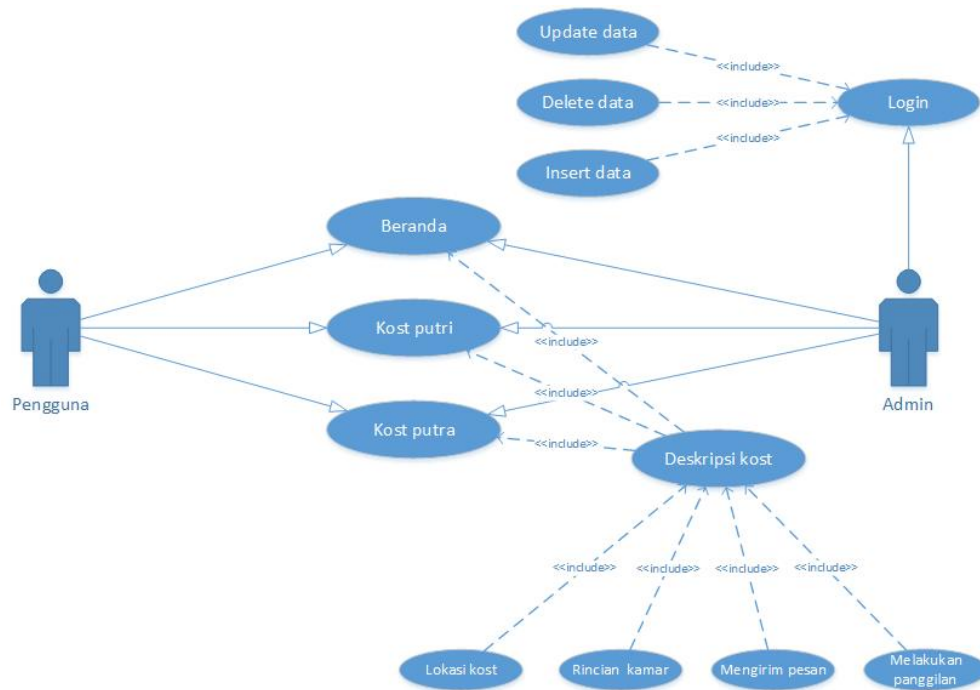
1. Data kost meliputi nama, alamat, jenis kost, harga sewa pertahun, harga sewa perbulan, minimal pembayaran.
2. Data kamar kost meliputi jumlah kamar, luas dan kamar yang masih tersedia.
3. Data pemilik kost meliputi nama dan nomor telepon.
4. Data fasilitas kost, fasilitas kamar dan fasilitas kamar mandi.
5. Gambar lokasi kost dan kamar kost.
6. Data koordinat lokasi kost.

3.5 Rancangan Aplikasi

Dalam pengembangan aplikasi metode yang digunakan adalah *Unified Markup Language* (UML). Model UML yang dipakai dalam pengembangan aplikasi ini yaitu *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*.

3.5.1 Use Case Diagram

Seperti yang kita ketahui *Use Case Diagram* merupakan bagian dari *Unified Markup Language* (UML). *Use case* diagram digunakan untuk menggambarkan suatu sistem dan bagaimana sistem tersebut bekerja. *Use case* diagram terdiri dari *use case* dan *actor*. Berikut adalah gambaran *Use case* diagram Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat dilihat pada Gambar 3.2:



Gambar 3.2 Use Case diagram

3.5.2 Class Diagram

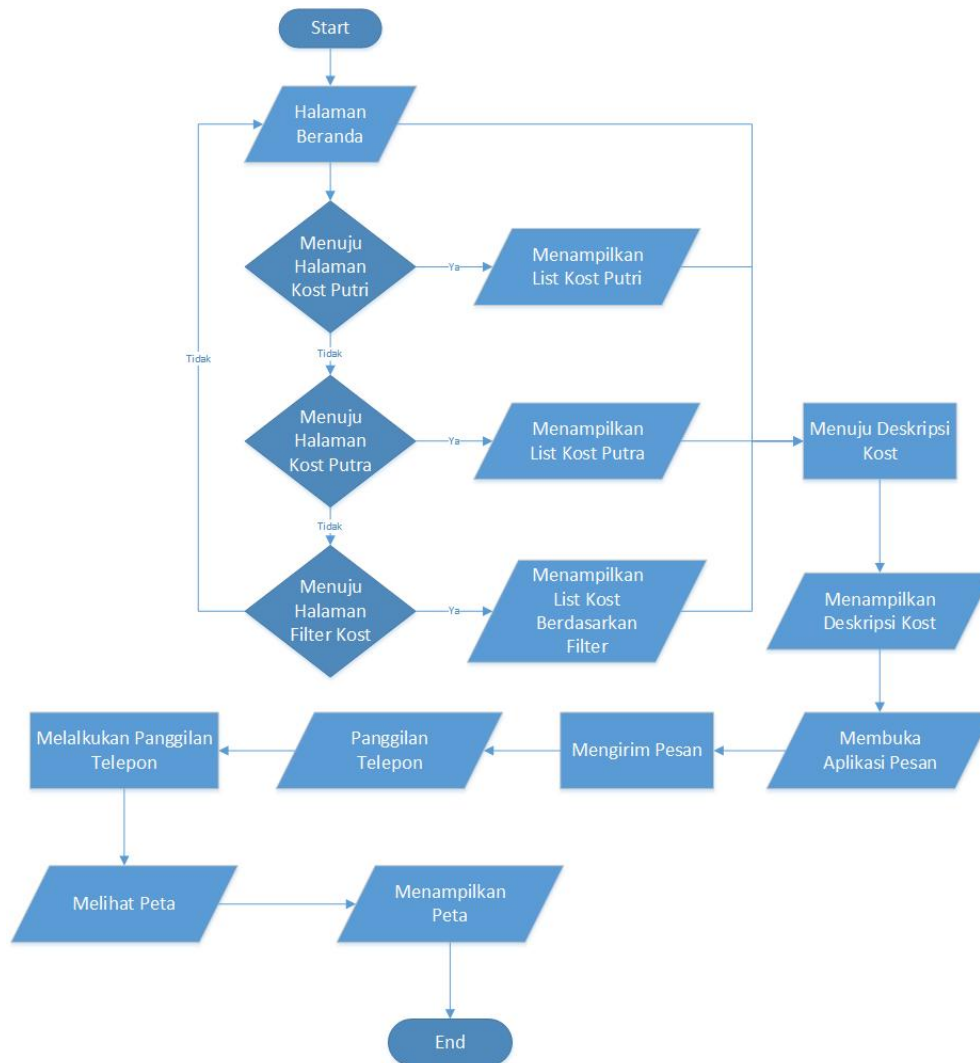
Gambaran *class* diagram dari Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat dilihat pada gambar 3.3.

Berikut merupakan penjelasan dari *class* diagram Sistem Informasi Pencarian Kost di Sekitar Kampus UMY:

1. *Class KostData* merupakan inialisasi sari data yang diambil dari database. *Class KostData* memiliki hubungan *Direct Association* dengan *class DataParser*.
2. *Class CustomAdapter* memiliki hubungan *Direct Association* dengan *class DataParser*.
3. *Class DataParser* berfungsi untuk mengubah data yang di *download* dari server manjadi data yang dapat dibaca oleh aplikasi.
4. *Class MainActivity* berfungsi sebagai *class* untuk menampilkan *class SemuaKost*, *class KostPutri*, *class KostPutra*, *class FilterHarga*, *class FilterHarga05jt*, *class FilterHarga610jt*, *class FilterHarga10jt*, *class FilterHargaTinggi* dan *class FilterHargaRendah* yang merupakan *class* yang memuat *list* kost.
5. *Class SemuaKost*, *class KostPutri*, *class KostPutra*, *class FilterHarga*, *class FilterHarga05jt*, *class FilterHarga610jt*, *class FilterHarga10jt*, *class FilterHargaTinggi* dan *class FilterHargaRendah* memiliki hubungan composition dengan *class Downloader* yang berfungsi untuk *download* data dari server.
6. *Class DetailActivity* berfungsi sebagai *class* yang menampilkan informasi kost secara keseluruhan dan memiliki hubungan *directed association* dengan *class SliderActivity* sebagai *class* yang berfungsi menampilkan *slide* gambar kost
7. *Class MapsActivity* berfungsi untuk menampilkan peta lokasi kost.
8. *Class PicassoClient* berfungsi untuk menampilkan gambar dari url.
9. *Class Connector* berfungsi sebagai jembatan antara aplikasi dengan server.
10. *Class SplashScreen* berfungsi untuk menampilkan halaman *SplashScreen* pada saat aplikasi dibuka.

3.5.3 *Flowchart* Diagram

Flowchart Diagram merupakan representasi visual dari langkah-langkah dan keputusan yang diperlukan untuk melakukan sebuah proses. Setiap langkah digambarkan dalam bentuk diagram dari mulai proses awal hingga akhir. Adapun *Flowchart* dari Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dijelaskan pada gambar 3.4:



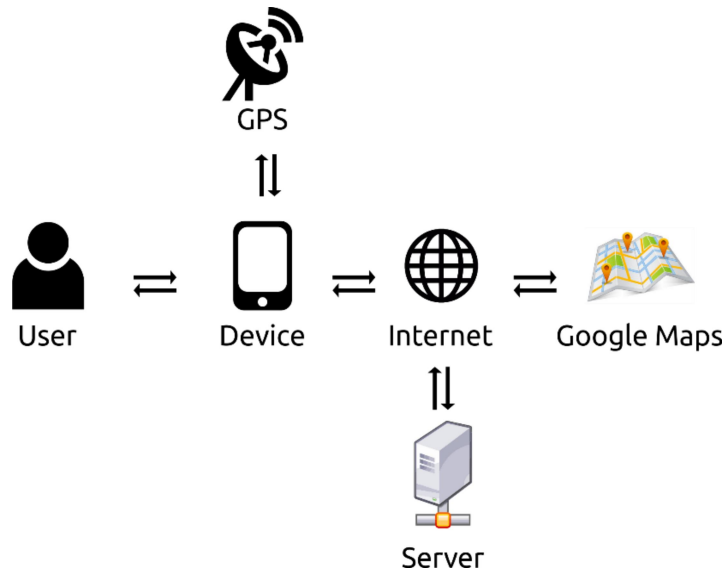
Gambar 3.4 *Flowchart* diagram

Berikut merupakan penjelasan mengenai *flowchart* diagram Sistem Informasi Pencarian Kost di Sekitar Kampus UMY:

1. Saat membuka aplikasi pengguna akan langsung dihadapkan dengan halaman beranda dari aplikasi yang menampilkan *list* dari semua kost baik itu kost putri atau kost putra.
2. Jika pengguna memilih halaman kost putri maka sistem akan menampilkan *list* kost putri.
3. Jika pengguna memilih halaman kost putra maka sistem akan menampilkan *list* kost putra.
4. Pengguna juga dapat memilih untuk menampilkan halaman filter harga dari kost maka sistem akan menampilkan halaman filter menurut pilihan pengguna.
5. Setelah pengguna memilih kost yang diinginkan maka sistem akan menampilkan deskripsi dari kost yang telah dipilih.
6. Setelah deskripsi muncul maka pengguna dapat memilih untuk mengirim pesan kepada pemilik kost.
7. Pengguna pun dapat melakukan panggilan telepon terhadap pemilik kost.
8. Jika pengguna ingin melihat peta maka sistem akan menampilkan peta lokasi dari kost yang telah dipilih.

3.5.4 Arsitektur Sistem

Gambaran arsitektur sistem yang digunakan dalam Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat dilihat pada gambar 3.5.



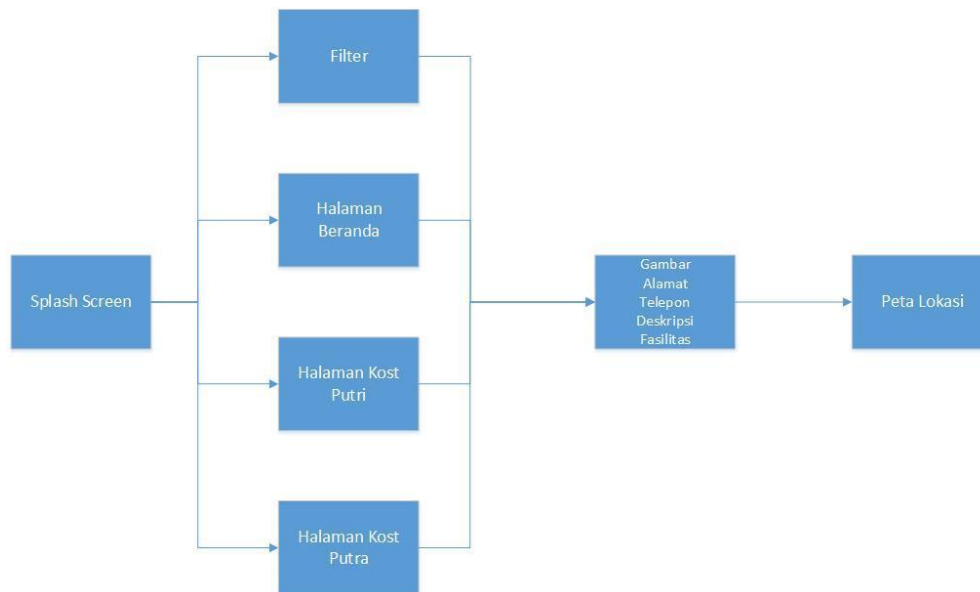
Gambar 3.5 Arsitektur sistem

Penjelasan tentang Gambar 3.5:

1. *User* memberikan perintah atau meminta aplikasi untuk menunjukkan kost yang diinginkan.
2. *User* memberikan perintah atau meminta aplikasi untuk menunjukkan lokasi kost dalam peta digital.
3. Aplikasi meminta data kost dari *database* server melalui koneksi internet.
4. Aplikasi meminta data peta lokasi (Google Maps) melalui koneksi internet.

3.6 Rancangan Struktur Menu

Berikut merupakan rancangan struktur menu dalam Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat dilihat pada gambar 3.6.



Gambar 3.6 Rancangan struktur menu

3.7 Rancangan Antar Muka

Salah satu elemen yang penting dalam pengembangan aplikasi adalah tampilan antarmuka. Rancangan antarmuka adalah tampilan dari suatu perangkat lunak yang berperan penting sebagai penghubung antara pengguna dan aplikasi. Dengan adanya antarmuka dapat membantu proses interaksi antara *user* dengan aplikasi. Berikut adalah perancangan antarmuka Sistem Informasi Pencarian Kost di Sekitar Kampus UMY:

3.7.1 Antarmuka *Splash Screen*

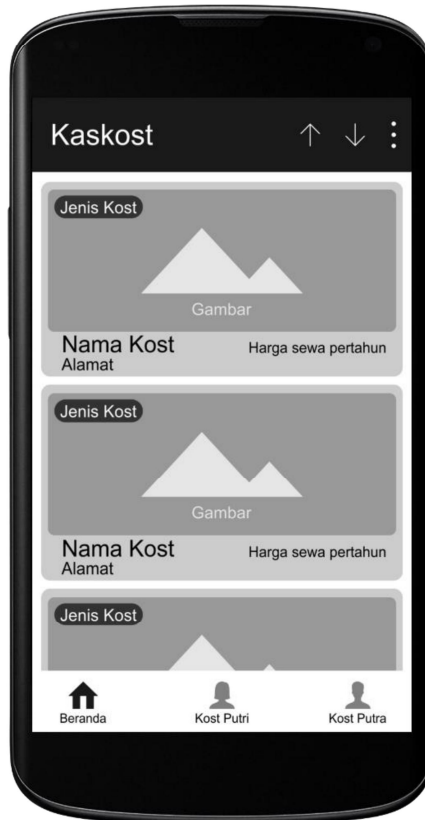
Berikut ini merupakan antarmuka *Splash screen* yang merupakan halaman pertama yang akan muncul ketika aplikasi dibuka. *Splash screen* dari aplikasi Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat dilihat pada gambar 3.7.



Gambar 3.7 Rancangan antarmuka *Splash screen*

3.7.2 Antarmuka Halaman Beranda

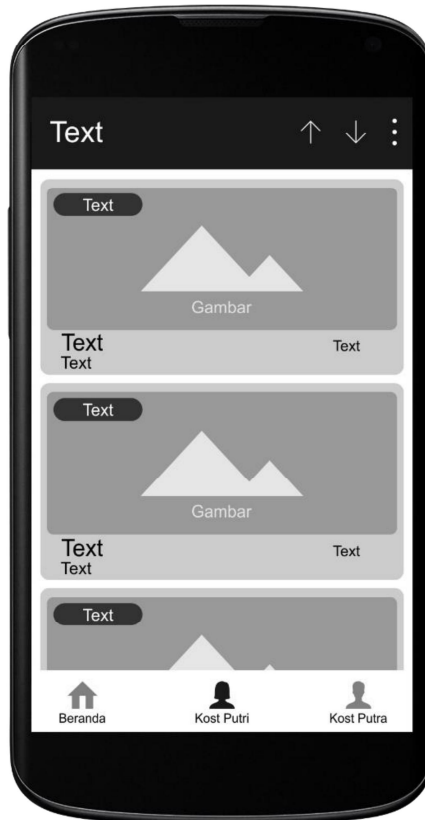
Halaman beranda merupakan halaman yang menampilkan list dari semua kost baik itu kost putra maupun putri. Halaman beranda aplikasi Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat dilihat pada gambar 3.8.



Gambar 3.8 Rancangan antarmuka Halaman Beranda

3.7.3 Antarmuka Halaman Kost Putri

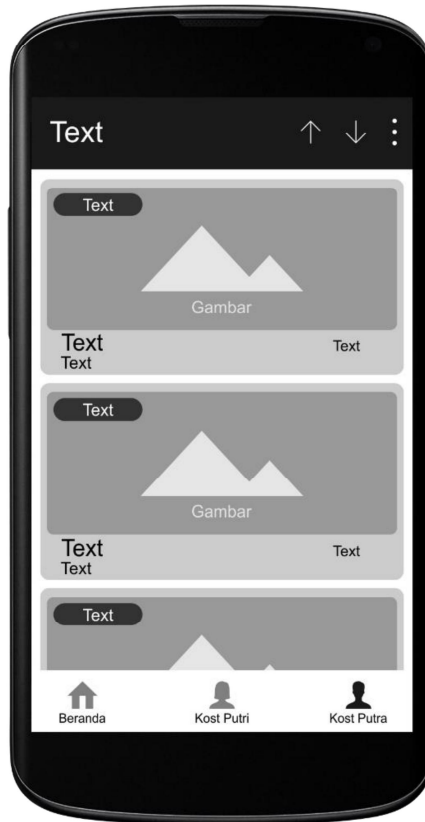
Halaman kost putri merupakan halaman yang menampilkan list dari semua kost putri. Halaman kost putri aplikasi Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat dilihat pada gambar 3.9.



Gambar 3.9 Rancangan antarmuka Kost Putri

3.7.4 Antarmuka Halaman Kost Putra

Halaman kost putra merupakan halaman yang menampilkan list dari semua kost putra. Halaman kost putra aplikasi Sistem Informasi Pencarian Kost di Sekitar Kampus UMY dapat dilihat pada gambar 3.10.



Gambar 3.10 Rancangan antarmuka Kost Putra

3.7.5 Antarmuka Deskripsi Kost

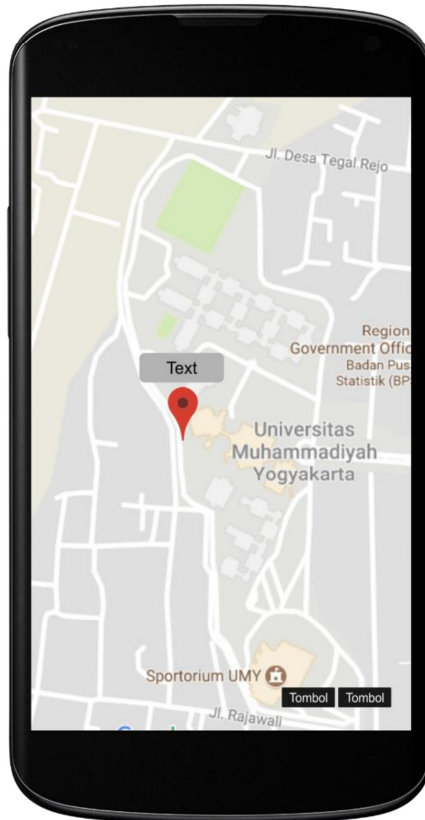
Antarmuka Deskripsi Kost menyajikan informasi lengkap mengenai kost. Antarmuka Deskripsi Kost dapat diakses dengan memilih salah satu kost dalam list yang ada di halaman beranda, halaman kost puri maupun kost putra. Antarmuka detail kost dapat dilihat pada gambar 3.11.



Gambar 3.11 Rancangan antarmuka Deskripsi Kost

3.7.6 Antarmuka Peta Lokasi

Berikut merupakan antarmuka lokasi kost yang ditunjukkan pada peta digital. Halaman antarmuka Peta Lokasi akan muncul ketika *user* memencet tombol “LIHAT PETA” pada halaman deskripsi kost. Antarmuka Peta Lokasi kost dapat dilihat pada gambar 3.12.



Gambar 3.12 Rancangan antarmuka Peta Lokasi Kost

3.8 Metode Pengumpulan Data

Ada beberapa cara yang dapat digunakan dalam mengumpulkan data diantaranya adalah kuesioner atau angket, observasi, wawancara atau bahkan gabungan ketiganya. Dalam penelitian ini peneliti menggunakan kuesioner sebagai alat untuk mengukur tingkat kelayakan aplikasi yang telah dibangun.

Kuesioner atau angket adalah sejumlah pertanyaan tertulis yang digunakan untuk memperoleh informasi dari responden untuk mendapatkan informasi yang diinginkan.

3.9 Analisis Data

Metode analisis data yang digunakan pada penelitian ini adalah metode deskriptif. Metode deskriptif adalah suatu metode yang digunakan dalam meneliti suatu objek, kondisi, atau peristiwa. Analisis deskriptif digunakan untuk menguji variabel yang bersifat kuantitatif. Tujuan dari penelitian deskriptif adalah untuk menggambarkan atau mendeskripsikan fakta-fakta, sifat serta hubungan antara fenomena yang diteliti.

Teknik pengolahan data untuk variabel bebas dapat menggunakan pengukuran dengan skala Likert. Menurut Sudaryono (2014). Skala Likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok orang sebuah kejadian atau gejala sosial yang telah ditetapkan secara spesifik oleh peneliti. Pertimbangan peneliti dalam memilih metode pengukuran ini karena dapat memudahkan responden dalam memilih jawaban. Dengan menggunakan skala Likert, variabel yang akan diukur dijabarkan menjadi dua dimensi. Dimensi lalu dijabarkan menjadi sub variabel. Kemudian sub variabel dijabarkan menjadi indikator-indikator yang dapat diukur.

Kriteria jawaban yang dibagikan kepada responden menggunakan kuesioner berupapegukuran skala Likert. Responden diminta untuk menggunakan sistem informasi secara keseluruhan dengan berhapan secara langsung. Responden diminta memberikan salah satu pilihan dari jawaban yang telah disediakan. Ada 5 pilihan jawaban yang diberikan, mulai dari sangat setuju sampai sangat tidak setuju. Data kuantitatif diubah berdasarkan bobot skor satu, dua, tiga, empat dan lima. Pembagian kategori dan skor pengukuran Skala Likert dapat dilihat pada tabel 3.1.

Tabel 3.1 Tabel Skor Liker

No	Kategori	Skor
1	Sangat Setuju	5
2	Setuju	4
3	Cukup Setuju	3
4	Tidak Setuju	2
5	Sangat Tidak Setuju	1

Hasil persentase digunakan untuk memberikan jawaban atas kelayakan dari aspek-aspek yang diteliti. Nilai maksimal yang diharapkan adalah 100% dan minimal 0%.

Pembagian kategori kelayakan dapat dilihat pada tabel 3.2.

Tabel 3.2 Tabel Kategori Kelayakan Aplikasi

No.	Kategori	Persentase
1	Sangat Layak	81% - 100%
2	Layak	61% - 80%
3	Cukup Layak	41% - 60%
4	Tidak Layak	21% - 40%
5	Sangat Tidak Layak	<20%