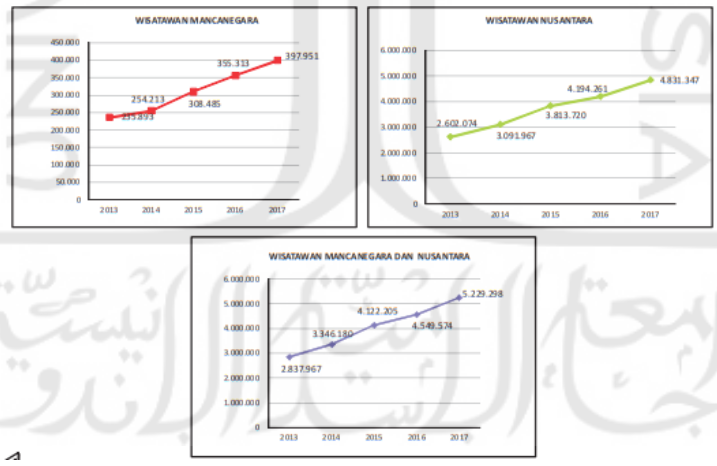
BAB III
METODE PENELITIAN

3.1 Pengumpulan Data

Dalam pengembangan aplikasi pembuatan *itinerary* wisata ini diperlukan data pendukung. Data dapat diperoleh lewat beberapa sumber seperti buku, jurnal, penelitian maupun informasi lain yang dapat diakses lewat internet. Selain itu, penulis melakukan wawancara untuk mengembangkan informasi yang didapatkan untuk membangun aplikasi ini. Proses pengumpulan data lewat wawancara dilakukan selama tiga minggu dengan memberikan pertanyaan kepada wisatawan, *tour guide*, *tour agent* dan pengelola tempat wisata. Lewat wawancara ini, diperoleh data dan informasi tambahan yang dapat digunakan untuk melengkapi data yang sudah didapatkan lewat metode yang lain.

3.2 Analisis Masalah

Daerah Istimewa Yogyakarta terkenal dengan banyaknya titik wisata, provinsi ini dapat menarik banyak wisatawan domestik maupun mancanegara untuk mengunjungi kawasan ini. Lewat informasi yang disajikan oleh Buku Statistik Kepariwisataan DI Yogyakarta tahun 2017, terdapat kenaikan jumlah wisatawan yang data ke daerah ini.



Gambar 3.1 Grafik pertumbuhan wisatawan di DIY

Pada Gambar 3.1 menunjukan grafik yang meningkat pada jumlah kunjungan wisatawan lokal maupun mancanegara ke Daerah Istimewa Yogyakarta kurun waktu 2013-2017. Kepedulian pemerintah dalam mengembangkan titik wisata baru turut mempegaruhi pertumbuhan jumlah kunjungan wisatawan ke provinsi DIY.

Banyaknya titik wisata yang dapt dikunjungi di DIY dapat membuat wisatawan membutuhkan alat bantu untuk menentukan urutan kunjungan ke tempat wisata yang akan dikunjungi. Terlebih informasi yang didapatkan dari sumber yang sama, diketahui bahwa lama tinggal wisatawan di hotel yang berada di provinsi DIY lebih dari satu hari pada tahun 2016-2017, sehingga masalah penentuan titik wisata terhadap hari berlibur dapat menjadi masalah efisiensi perjalanan. Beberapa *tour agent* yang sempat penulis wawancarai, memiliki kendala ketika konsumennya meminta mengunjungi beberapa tempat yang jarang disediakan oleh mereka. Mereka sedikit kesulitan dalam mengetahui rute ke tempat wisata tersebut, dan mengurutkan urutan kunjungan tempat wisata dengan adanya tempat baru yang belum mereka kunjungi sebelumnya.

Permasalahan yang dihadapi berupa penentuan titik wisata terhadap hari berlibur wisatawan dan urutan kunjungan titik wisata tentunya membutuhkan solusi untuk meningkatkan tingkat efisiensi dalam pembuatan *itinerary*.

3.3 Analisis Kebutuhan

Analisis Kebutuhan ini merangkum apa saja yang dibutuhkan dapam membangun aplikasi berdarkan kebutuhan masukan, kebutuhkan proses, kebutuhan keluaran, kebutuhan perangkat keras dan kebutuhan perangkat lunak.

3.3.1 Analisis Kebutuhan Masukan

Berdasarkan hasil analisis dari masalah yang didapatkan, maka aplikasi membutuhkan masukan data sebagai berikut:

1. Pengguna memasukan tanggal awal dan tanggal akhir hari berwisata
2. Pengguna memasukan lokasi awal perjalanan
3. Pengguna memasukan beberapa destinasi wisata yang akan dikunjungi

3.3.2 Analisis Kebutuhan Proses

Berdasarkan hasil analisis dari masalah yang didapatkan, maka aplikasi terdapat proses seperti berikut:

1. Proses mengakses lokasi pengguna saat ini
2. Proses mengakses lokasi awal perjalanan
3. Proses mengakses destinasi kunjungan
4. Proses pembuatan *itinerary* wisata

5. Proses menampilkan hasil *itinerary* wisata
6. Proses membuka aplikasi Google Maps untuk navigasi kearah lokasi yang akan dituju

3.3.3 Analisis Kebutuhan Keluaran

Berdasarkan hasil analisis dari masalah yang didapatkan, identifikasi keluaran apa saja yang harus ada pada aplikasi pembuat *itinerary* wisata ini sebagai berikut:

1. *Home Activity* berisi riwayat *itinerary* yang pernah dibuat dan menampilkan cuaca saat ini.
2. *Trip Plan Activity* berisi masukan yang dibutuhkan untuk membuat *itinerary*.
3. *Itinerary Activity* menampilkan hasil *itinerary* menggunakan *Balanced K-Means* dan *Traveling Salesman Problem*.

3.3.4 Analisis Kebutuhan Perangkat Keras

Pada poin analisis kebutuhan perangkat keras, merupakan daftar perangkat keras yang digunakan dalam membangun aplikasi ini sebagai berikut:

1. Laptop dengan spesifikasi *processor* Intel Core i5, RAM 8GB, dan HDD 1TB.
2. Ponsel Android dengan spesifikasi RAM 3GB dan *Internal Storage* 32GB.

3.3.5 Analisis Kebutuhan Perangkat Lunak

Pada poin analisis kebutuhan perangkat lunak, merupakan daftar perangkat lunak yang digunakan dalam membangun aplikasi ini sebagai berikut:

1. Ubuntu 18.04 Bionic Beaver

   Sistem Operasi yang digunakan untuk membangun aplikasi ini.
2. Android Studio

   IDE yang digunakan untuk membuat aplikasi Android.
3. Intellij IDEA 2019

   IDE yang digunakan untuk membuat API untuk komputasi dengan metode *Balanced K-Means* dan *Held-Karp*.
4. Java Virtual Machine

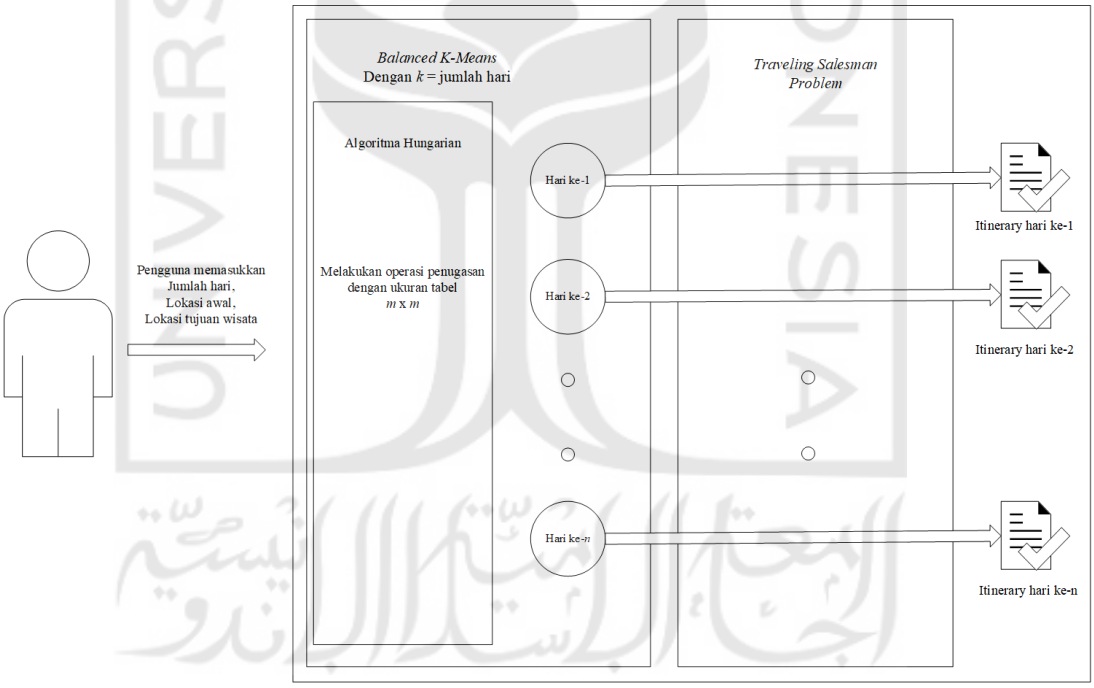
   Virtual machine yang digunakan untuk menguji metode yang digunakan dalam membangun aplikasi *itinerary* wisata.

3.4 Perancangan dan Pemodelan Aplikasi

3.4.1 Kerangka Kerja Aplikasi Pembuat Itinerary Wisata

Kerangka kerja dibutuhkan supaya mempermudah pembuatan aplikasi dan sesuai dengan analisis kebutuhan yang sudah dilakukan. Terdapat tiga analisis kebutuhan yang bersinggungan langsung dengan pembuatan aplikasi ini, yaitu analisis kebutuhan masukan, analisis kebutuhan proses, dan analisis kebutuhan keluaran. Dalam kerangka kerja ini, masukan akan dilakukan oleh pengguna (aksi), kemudian proses dan keluaran merupakan reaksi dari aksi yang telah dilakukan oleh pengguna. Perilaku ini merupakan sebuah aktifitas yang dapat diujikan dan untuk memastikan pemahaman atas analisis yang sudah dilakukan.

Pada analisis masukan, pengguna mengisi tanggal mulai dan tanggal selesai, kemudian mengisi lokasi awal dimana adalah tempat menginap pengguna. Destinasi tujuan wisata dimasukkan pengguna dengan menginputkan nama lokasi pada kolom pencarian. Data lokasi pada kasus ini, diambil dari data Google Maps API. Data yang dimasukkan pengguna pada aplikasi ini akan di masukkan kedalam dua proses utama.



Gambar 3.2 Diagram kerangka kerja sistem

Kerangka kerja sistem yang dapat dilihat pada Gambar 3.2 dimana terdapat dua modul utama yang digunakan untuk membuat *itinerary* wisata. Modul pertama adalah pengelompokan tempat wisata berdasarkan hari menggunakan *Balanced K-Means Clustering*, dan didalam modul ini terdapat sub modul *Hungarian Algorithm* untuk menyelesaikan

masalah penugasan yang membuat anggota setiap *cluster* dapat seimbang. Hasil dari modul pertama akan dieksekusi pada modul kedua, untuk menemukan solusi TSP dengan metode *Held-Karp*. Kedua modul akan menghasilkan rencana perjalanan wisata yang dapat dijadikan rekomendasi bagi wisatawan.

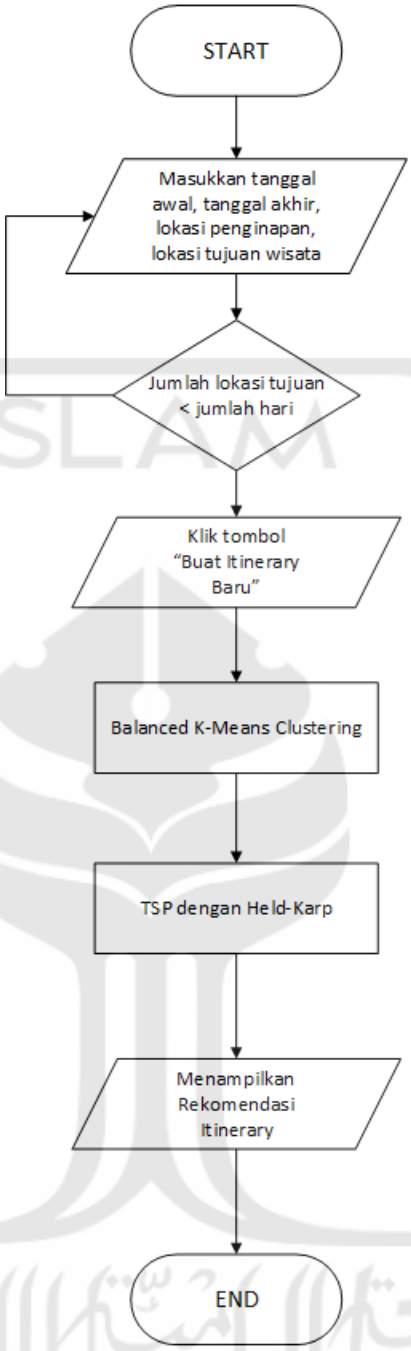
3.4.2 *Flowchart* Aplikasi Pembuat Itinerary Wisata

Flowchart merupakan diagram yang memiliki arus untuk menggambarkan langkah-langkah penyelesaian suatu masalah atau untuk menggambarkan sebuah algoritma. Dalam aplikasi ini, terdapat tiga *flowchart* yang digunakan untuk menggambarkan alur sistem. *Flowchart* yang digambarkan ini diharapkan bisa memperjelas pemahaman tentang aplikasi yang akan dibangun.

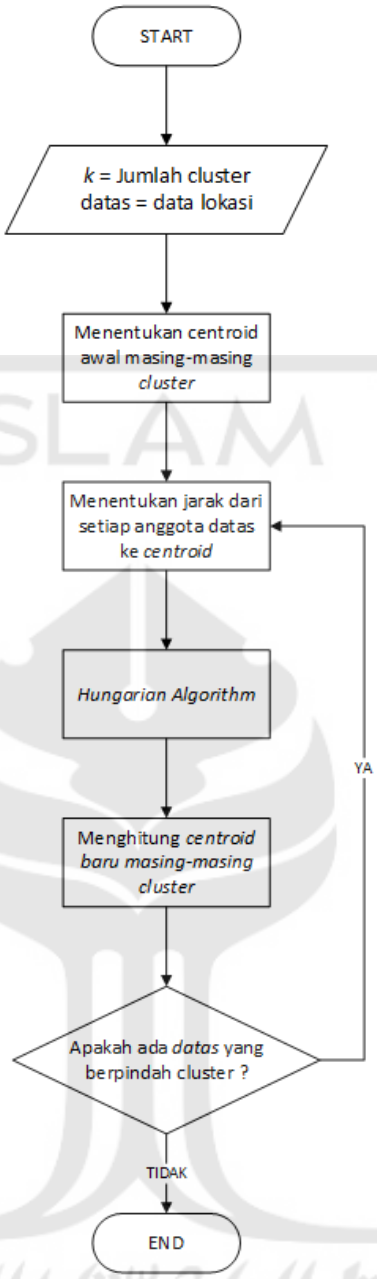
Diagram pertama digambarkan pada Gambar 3.3 *Flowchart* keseluruhan sistem memberikan penjelasan menyeluruh tentang alur aplikasi pembuatan *itinerary* wisata dimulai dengan pengguna memasukkan tanggal awal, tanggal akhir, lokasi penginapan, dan lokasi destinasi wisata yang akan dituju. Proses selanjutnya sistem akan melakukan pengecekan apakah jumlah lokasi lebih banyak dibandingkan total hari, apabila jumlah lokasi masih kurang, maka aplikasi akan memaksa pengguna melakukan penambahan destinasi wisata atau mengurangi hari berlibur. Aplikasi mendapatkan total hari, diambil dari selisih antara tanggal awal dan tanggal akhir berlibur. Jika kondisi sudah terpenuhi, pengguna dapat menekan tombol “Buat Itinerary Baru” untuk memproses ke proses *Balanced K-Means*.

Flowchart kedua dapat dilihat pada Gambar 3.4 berisi penjelasan yang lebih detail proses *clustering*, yang mengelompokan destinasi tujuan wisata berdasarkan jumlah hari. Proses dimulai dengan menentukan *k* dimana merupakan jumlah hari berlibur. Setelah *k* ditentukan, proses selanjutnya adalah menentukan *k centroid* awal. *Centroid* awal diambil dari data lokasi destinasi wisata yang dimasukkan oleh pengguna secara acak. Kemudian, proses *clustering* dilakukan dengan operasi *Hungarian algorithm* dan hasilnya akan digunakan untuk menghitung *centroid* baru.

Tahap selanjutnya adalah pengecekan apakan terdapat data lokasi yang berpindah *cluster*. Apabila terdapat data lokasi yang berpindah, maka program akan menghitung ulang jarak data terhadap *centroid* dan melanjutkan ke langkah selanjutnya. Iterasi ini akan terus dilakukan hingga tidak ada lokasi wisata yang berpindah *cluster*. Sehingga setiap *cluster* memiliki destinasi tujuan wisata yang memiliki kedekatan jarak koordinat.



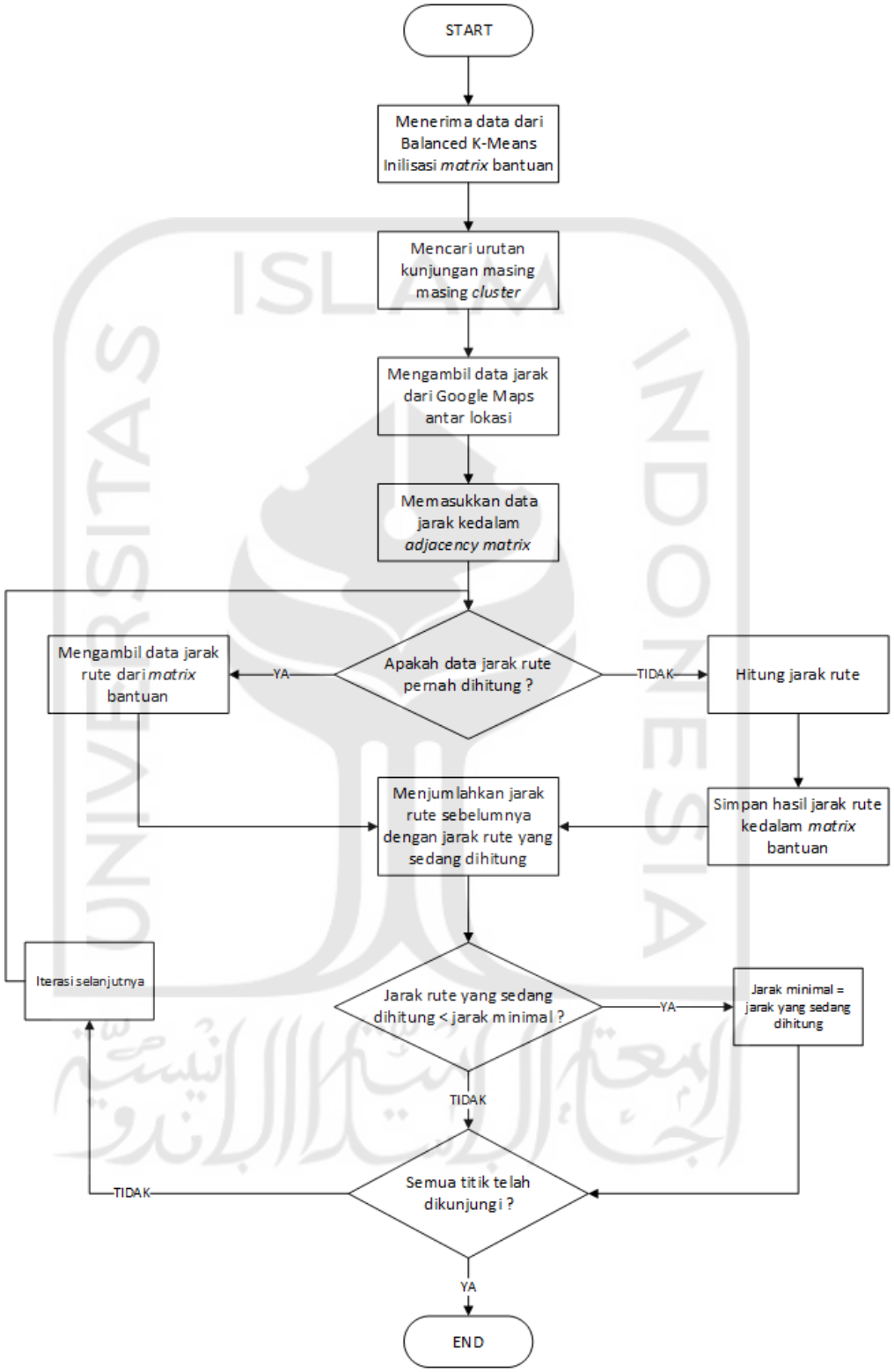
Gambar 3.3 *Flowchart* keseluruhan sistem



Gambar 3.4 *Flowchart* proses *Balanced K-Means*

Diagram ketiga pada Gambar 3.5 merupakan *flowchart* yang menjelaskan proses penyelesaian *Traveling Salesman Problem* dengan algoritma *held-karp* untuk mendapatkan rute perjalanan terpendek. Masukkan pada proses ini didapat dari hasil proses *clustering* yang dijelaskan pada Gambar 3.4. Kemudian proses selanjutnya mencari urutan destinasi tujuan wisata berdasarkan masing-masing *cluster*(hari), yang diikuti dengan mengambil data jarak antar lokasi tersebut dari Google Maps API. Langkah selanjutnya adalah, memasukkan data

jarak setiap lokasi tersebut kedalam *adjacency matrix* yang akan dicari permutasi rute kunjungan dengan algoritma *held-karp*.



Gambar 3.5 *Flowchart* proses penyelesaian *Traveling Salesman Problem*

Langkah selanjutnya melakukan iterasi dimulai dari titik lokasi awal hingga titik kembali ke lokasi awal. Kemudian aplikasi akan melakukan pengecekan, apakah pada iterasi tersebut, jarak antar lokasi sudah pernah dihitung, apabila sudah, maka akan diambil dari *matrix* bantuan yang digunakan untuk menyimpan data jarak yang pernah dihitung. Namun apabila jarak tersebut belum dihitung, maka aplikasi akan melakukan perhitungan jarak rute dan kemudian menyimpannya pada *matrix* bantuan kemudian menjumlahkan total rute kunjungan dengan jarak yang saat ini dihitung. Hasil penjumlahan tersebut akan dicek, apabila jarak rute kunjungan baru lebih kecil dibanding jarak rute kunjungan minimum saat ini, maka nilai jarak minimum akan digantikan dengan jarak rute kunjungan baru tersebut. Selanjutnya akan dilakukan pengecekan, apabila semua titik lokasi sudah dikunjungi seluruhnya, maka iterasi selesai, namun apabila terdapat titik yang belum dikunjungi, maka akan melanjutkan ke iterasi selanjutnya.

3.4.3 Rancangan Antarmuka Aplikasi Pembuat Itinerary Wisata

Rancangan antarmuka aplikasi digunakan untuk merepresentasikan tampilan aplikasi yang akan dibuat. Rancangan ini berupa gambar sederhana yang memberikan ilustrasi tampilan.

a. Rancangan Antarmuka *Home Activity*

Saat pengguna membuka aplikasi, pengguna akan melihat halaman beranda yang berisi informasi cuaca, lokasi saat ini, dan daftar *itinerary* yang sudah pernah dibuat.

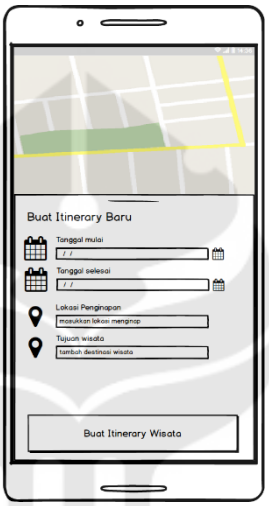
Gambar 3.6 Rancangan antarmuka *Home Activity*

Gambar 3.6 merupakan gambaran antar muka *Home Activity* yang terdiri dari beberapa komponen. Komponen pertama berupa gambar dan teks yang memberikan informasi cuaca dan lokasi pengguna saat ini. Komponen kedua adalah *list*

*itinerary*yang pernah dibuat, apabila salah satu dari data dalam *list* dipilih, maka akan menuju *Itinerary Activity*. Komponen terakhir yaitu tombol pada pojok kanan bawah layar untuk berpindah halaman ke *Trip Plan Itinerary* untuk membuat rencana perjalanan wisata baru.

b. Rancangan Antarmuka *Trip Plan Activity*

Halaman ini dibuka apabila pengguna menekan tombol lingkaran pada *Home Activity*. *Trip Plan Activity* berisi *maps* yang menginformasikan lokasi pengguna saat ini, dan tempat yang dipilih menginap dan destinasi wisata yang akan dituju.

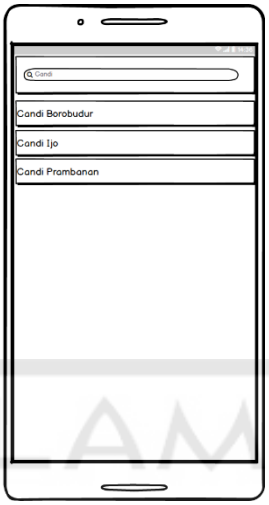


Gambar 3.7 Rancangan antarmuka *Trip Plan Activity*

Pengguna yang memasukkan tanggal mulai dan tanggal selesai, akan muncul *datepicker*. Tanggal yang dapat dipilih pada form tanggal mulai adalah mulai hari ini, sedangkan tanggal yang dapat dipilih pada form tanggal selesai adalah mulai tanggal selesai. Pada form Lokasi Menginap dan Tambah Destinasi, pengguna akan diarahkan pada *Place Search Activity*. Tombol Buat Itinerary Wisata akan memproses hasil masukkan pengguna dan menghasilkan rekomendasi *itinerary* wisata.

c. Rancangan Antarmuka *Place Search Activity*

Rancangan antarmuka ini merupakan rancangan asli yang disediakan oleh Google Maps Android SDK. Halaman ini digunakan untuk mencari lokasi yang dimasukkan oleh pengguna.



Gambar 3.8 Rancangan antarmuka *Place Search Activity*

Pengguna mengisi *search* box pada bagian atas halaman, kemudia mengetikan sebuah nama tempat, dan *search box* akan menghasilkan sugesti tempat yang dipilih oleh pengguna berdasarkan kata yang dimasukkan pengguna. Setelah pengguna memilih tempat, maka aplikasi mengembalikan ke *Trip Plan Itinerary*.

d. Rancangan Antarmuka *Itinerary Activity*

Setelah pengguna memasukkan data masukan yang lengkap dan menekan tombol Buat Itinerary Wisata pada *Trip Plan Activity*, aplikasi akan menuju halaman ini untuk menampilkn hasil proses pembuatan *itinerary*. Halaman ini dapat diakses lewat *list itinerary* yang pernah dibuat pada *Home Activity*.

Gambar 3.9 Rancangan antarmuka *Itinerary Activity*

Pada halaman ini terdapat komponen tombol kembali pada bagian atas halaman yang digunakan untuk kembali ke halaman sebelumnya. Informasi yang ditampilkan pada halaman ini adalah rekomendasi rencana perjalanan yang dikelompokkan berdasarkan hari. Setiap kelompok harinya, terdapat linimasa perjalanan wisatawan yang dimulai dari lokasi penginapan, perjalanan ke setiap lokasi yang akan dikunjungi, dan kembali ke lokasi penginapan. Setiap tempat dan keterangan perjalanan dapan di tekan untuk melihat detail lokasi dan navigasi perjalanan. Detail lokasi dan navigasi perjalanan akan dibuka oleh aplikasi Google Maps, sehingga pengguna wajib menginstall aplikasi Google Maps.

3.5 Rencana Pengujian Aplikasi

Rencana pengujian diperlukan untuk memastikan bahwa aplikasi yang dibuat dapat berjalan dengan baik dan sesuai kebutuhan. Pengujian dibagi menjadi tiga, yaitu pengujian fungsionalitas, pengujian pada pengguna atau *User Acceptance Test*, dan membandingkan algoritma TSP yang dipakai yaitu *Held-Karp* dengan *Brute Force*. Hal ini dilakukan untuk melihat berapa lama *runtime* kedua algoritma tersebut.

Pengujian yang pertama adalah pengujian fungsionalitas untuk mengetahui kebutuhan aplikasi sudah terpenuhi atau belum. Pengujian ini dilakukan oleh penulis atau orang terdekat yang dipercaya melakukan pengujian ini. Terdapat aktivitas dalam menguji fungsionalitas aplikasi ini. Pada Tabel 3.1 disebutkan dan dijelaskan aktivitas pengujian fungsionalitas.

Tabel 3.1 Pengujian Fungsionalitas

No	Fungsionalitas	Aktifitas	Hasil yang diharapkan
1.	Buat *itinerary* wisata	Memilih tombol “Buat Itinerary” berbentuk lingkaran pada pojok kanan bawah *Home Activity*. Swipe up pada halaman *Home Activity*. Menekan tombol “Pilih tanggal mulai” dan “Pilih tanggal mulai” kemudian memilih tanggal yang tersedia. Menekan tombol “Pilih tanggal mulai” atau “Pilih tanggal mulai” dan memilih tanggal yang tersedia. Menekan tombol “Pilih lokasi menginap”, aplikasi akan mengarah ke *Place Search Activity*¸ kemudian memasukkan nama lokasi. Menekan tombol “Pilih lokasi menginap”, aplikasi akan mengarah	Aplikasi dapat membuat rekomendasi *itinerary* wisata dengan solusi yang optimal.

		ke *Place Search Activity*¸ kemudian memasukkan nama lokasi. Menekan tombol "Buat Itinerary Wisata".	
2.	Deteksi lokasi pengguna	Mengaktifkan izin akses lokasi ponsel pengguna	Aplikasi dapat menampilkan penanda lokasi pengguna saat ini pada maps.
3.	Lihat ringkasan *itinerary* yang pernah dibuat	Memilih daftar *itinerary* yang pernah dibuat pada *Home Activity*	Aplikasi dapat menampilkan lokasi penginapan, ringkasan durasi jarak dan waktu yang harus ditempuh dan daftar lokasi wisata
4.	Lihat detail *itinerary* yang pernah dibuat	Menekan tombol Lihat Itinerary Wisata pada *activity Summary*	Aplikasi dapat menampilkan *timeline* perjalanan wisata yang pernah atau akan dilakukan.
5.	Lihat informasi tempat wisata	Memilih daftar *itinerary* yang pernah dibuat pada *Home Activity*, kemudian memilih lokasi yang akan dilihat informasinya pada halaman *Itinerary Activity*.	Aplikasi dapat membuka aplikasi Google Maps untuk mendapatkan informasi tentang lokasi yang dipilih.

Pengujian kedua adalah *User Acceptance Test* (UAT) dengan cara meminta responden dalam hal ini pengguna menjalankan aplikasi untuk mencoba semua fitur yang ada didalam aplikasi. Pada Tabel 3.2 berikut merupakan daftar pertanyaan untuk pengujian aplikasi kepada pengguna.

Tabel 3.2 Pengujian Aplikasi kepada Pengguna

No	Pertanyaan	Jawaban				
		STS	TS	N	S	SS
1.	Apakah aplikasi memiliki tata letak, menu, tombol yang mudah dipahami?					
2.	Apakah aplikasi dapat membuka Google Maps untuk menampilkan informasi lokasi yang dipilih?					
3.	Apakah aplikasi dapat membuka Google Maps untuk menampilkan navigasi perjalanan yang dipilih?					
4.	Apakah keseluruhan fitur aplikasi dapat berjalan dengan baik?					

Tabel 3.3 Pengujian Aplikasi kepada Pengguna (Terhadap Fitur dan Fungsionalitas Aplikasi)

No	Pertanyaan	Jawaban				
		STS	TS	N	S	SS
1.	Apakah aplikasi dapat mendeteksi lokasi ponsel dan cuaca dengan akurat?					
2.	Apakah aplikasi dapat memberikan lokasi penginapan yang dipilih dengan akurat?					
3.	Apakah aplikasi dapat memberikan lokasi wisata yang dipilih dengan akurat?					
4.	Apakah aplikasi dapat digunakan untuk membuat *itinerary* wisata?					
5.	Apakah aplikasi dapat menampilkan informasi jarak dan waktu tempuh antar lokasi dengan akurat?					
6.	Apakah aplikasi dapat memberikan rute perjalanan paling pendek dan cepat untuk dilalui?					

Nilai jawaban dari *User Acceptance Tests* di atas dan rumus untuk menghitung indeks keberhasilan pengujian sebagai berikut:

Sangat Tidak Setuju (STS) : 1
Tidak Setuju (TS) : 2
Netral (N) : 3
Setuju (S) : 4
Sangat Setuju (SS) : 5

Rumus indeks keberhasilan pengujian = $\frac{\boldsymbol{nilai\ total\ kuesioner}}{\boldsymbol{nilai\ maksimum\ kuesioner}} \times 100\%$ (3.1)

Nilai total kuesioner didapatkan dari hasil kuesioner yang diajukan kepada responden(pengguna). Nilai maksimum kuesioner didapatkan dari jumlah pengguna dikalikan dengan jumlah pertanyaan dikalikan indeks nilai SS (5). Hasil bagi antara nilai total kuesioner dan nilai maksimum kuesioner akan dikalikan dengan 100% untuk menghasilkan presentase. Presentase ini yang akan dijadikan acuan dalam pengujian UAT.

Pengujian yang terakhir adalah menguji waktu *runtime* permasalahan *Traveling Salesman Problem* menggunakan *brute force* dan *held-karp* untuk mengetahui berapa perbedaan waktu *runtime* antara kedua algoritma tersebut. Disini penulis menguji algoritma

ini dengan memasukkan beberapa jumlah koordinat lokasi dan program *testing* ini dibuat menggunakan Bahasa pemrograman Java.