

BAB III

ANALISA DAN PERANCANGAN *SOFTWARE*

3.1. Analisa Kebutuhan *Software*

Dalam melakukan perancangan *software*, langkah pertama yang dilakukan adalah mengidentifikasi masalah yang ditemukan. Langkah berikutnya adalah melakukan analisa kebutuhan *software*, supaya aplikasi yang akan dibuat dapat memberikan solusi dan menyelesaikan permasalahan yang ada.

3.1.1. Identifikasi Masalah

Dengan adanya lokasi wisata di Nabire yang sangat banyak dan menjamin masyarakat Nabire serta turis namun kurangnya pengetahuan letak dan *direction* menuju tempat wisata tersebut menjadi salah satu penghambat masyarakat dan turis untuk menuju lokasi wisata di daerah Nabire. Untuk mengetahui lokasi wisata di daerah Nabire, penulis menggunakan *Google Maps* sebagai pencitraan lokasi wisata yang ada di daerah Nabire. Dengan mengetahui koordinat lokasi wisata di daerah Nabire, penulis membuat marker lokasi wisata di daerah Nabire. Aplikasi ini diharapkan mampu membantu masyarakat dan turi yang ingin mengunjungi lokasi wisata di daerah Nabire.

3.1.2. Analisa Kebutuhan

Berlandaskan idntifikasi masalah, analisa kebutuhan program sesuai dengan *user requirement* adalah sebagai berikut: Mengakses dan mengetahui posisi user menggunakan *Location Based Service (LBS)* pada perangkat *smartphone* user. Dan kemudian dapat memilih kemana tujuan wisata user di daerah Nabire.

3.2. Desain

Dalam sub bab ini akan dijelaskan secara lebih mendetail tentang perancangan desain yang penulis terapkan dalam program.

3.2.1. Rancangan Algoritma

Algoritma yang digunakan adalah algoritma string matching, dengan menggunakan string yang sudah di buat menjadi class maka pencarian akan menjadi lebih mudah karena hanya menggunakan sting yang ada. Berikut kode *listening* string matching:

```

if (location.equals("Pantai Gedo")) {
    mMap.clear();
    mMap.addMarker(new MarkerOptions().anchor(0.0f,
1.0f).position(pantai_gedo).icon(BitmapDescriptorFactory
.fromResource(R.drawable.pantai)).title("Pantai").snippet("Pantai
Gedo"));
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
    {
        if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_FINE_LOCATION)
            == PackageManager.PERMISSION_GRANTED) {
            buildGoogleApiClient();
            mMap.setMyLocationEnabled(true);
        }
    }
    else {
        buildGoogleApiClient();
        mMap.setMyLocationEnabled(true);
    }
    gps = new Gps (Maps_Aktiviti_Cari.this);
    LatLng lokasi=new
LatLng(gps.getLatitude(),gps.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions();
    markerOptions.position(lokasi);
    markerOptions.title("Posisi saat ini");

markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable
.user));
    MarkerOptions options = new MarkerOptions();
    options.position(lokasi);
    LatLng origin=lokasi;
    LatLng dest = pantai_gedo;
    String url = getUrl(origin, dest);
    Log.d("lokasi", url.toString());
    FetchUrl FetchUrl = new FetchUrl();
    FetchUrl.execute(url);
    //move map camera

```

```

        mMap.moveCamera(CameraUpdateFactory.newLatLng(pantai_gedo));
        mMap.animateCamera(CameraUpdateFactory.zoomTo(15));
    }

```

Untuk dapat mengetahui lokasi user, maka perlu aktifkan GPS pada perangkat user agar program bisa mengakses lokasi user menggunakan internet. Lokasi user tersebut menggunakan kodenya sebagai berikut:

```

private Location getLocation() {

    try {
        locationManager = (LocationManager)
        _context.getSystemService(LOCATION_SERVICE);

        // cek GPS status
        isGPSEnable =
        locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
        // cek status koneksi
        isNetworkEnable =
        locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (!isGPSEnable && !isNetworkEnable) {
            // tidak ada koneksi ke GPS dan Jaringan
        } else {
            // bisa dapatkan lokasi
            canGetLocation = true;

            // cek apakah koneksi internet bisa ?
            if (isNetworkEnable) {
                // ambil posisi berdasarkan Network

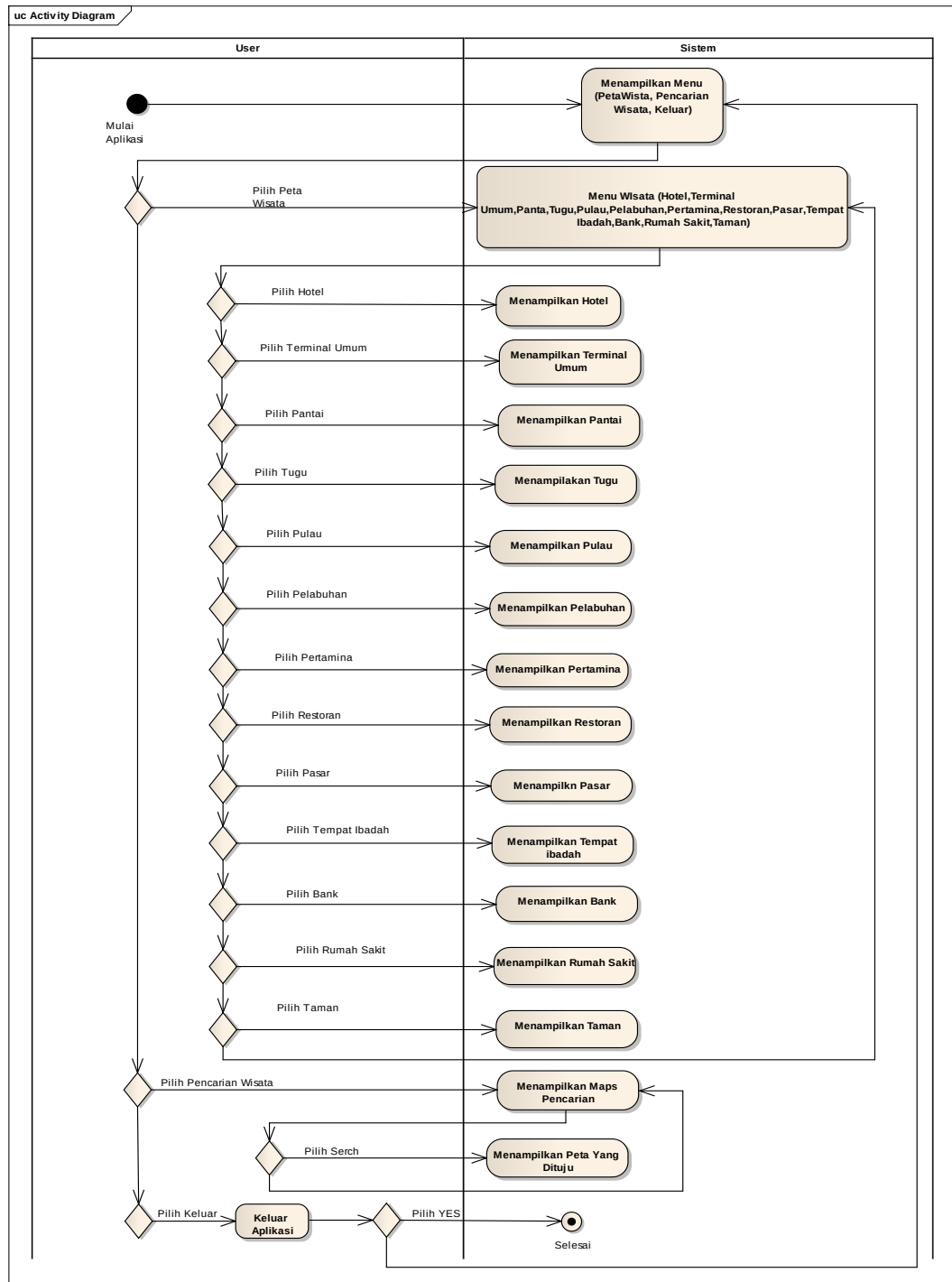
                locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER
                , MIN_WAKTU_GPS_UPDATE,
                    MIN_JARAK_GPS_UPDATE, this);
                if (locationManager != null) {
                    // ambil posisi terakhir user menggunakan
                    Network
                        location =
                locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                // jika lokasi berhasil didapat
                if (location != null) {
                    // ambil latitude
                    latitude = location.getLatitude();
                    // ambil longitude
                    longitude = location.getLongitude();
                }
            }
        }
    }
}

```

3.2.2. Software Arsitektur

1. UML

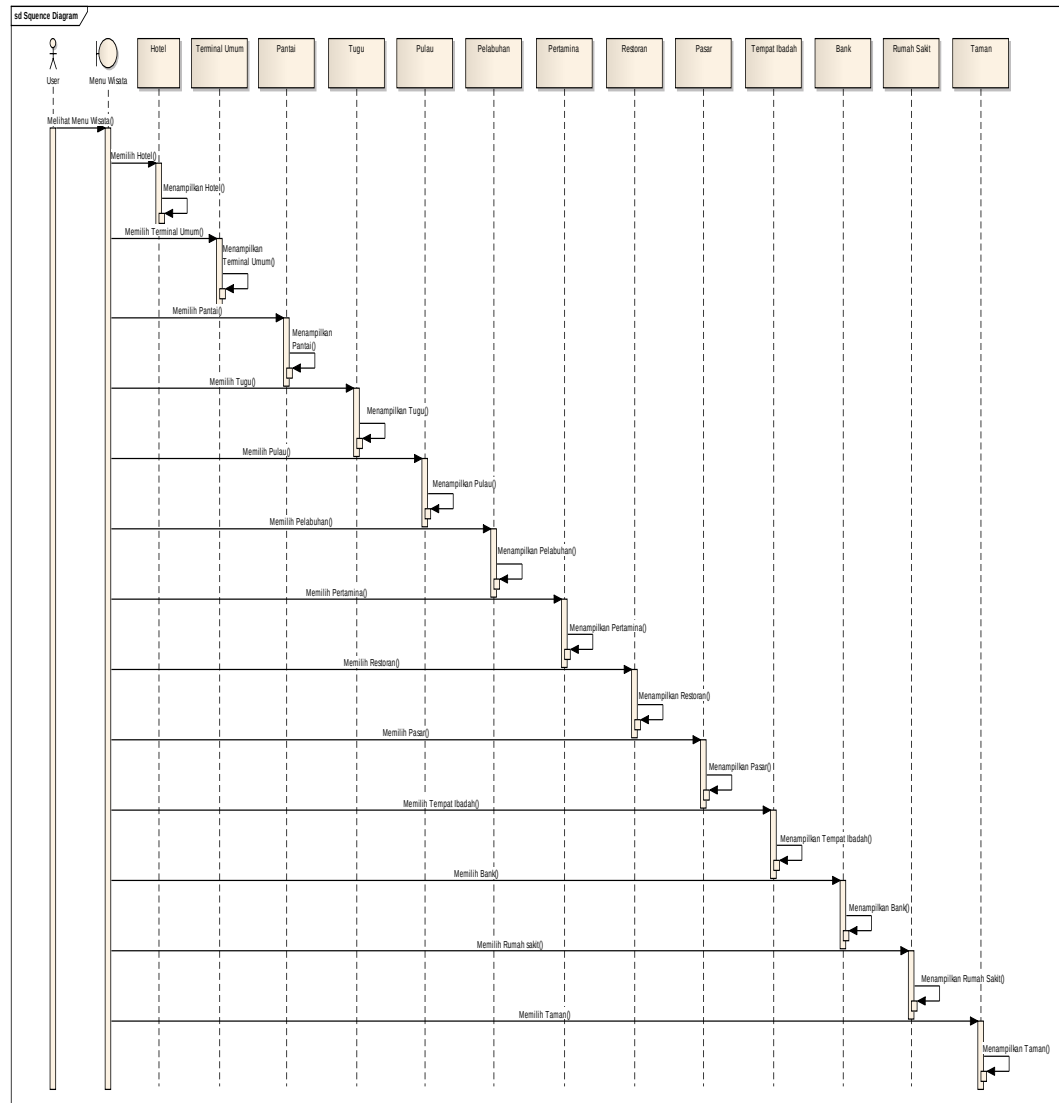
a. Activity diagram



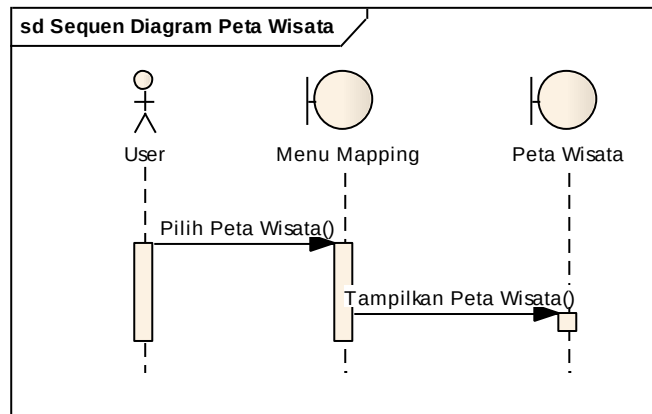
Gambar III.1

Activity Diagram Aplikasi Pencarian Lokasi Wisata

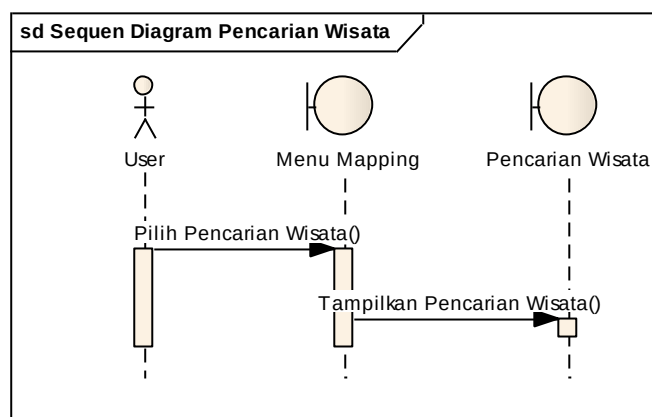
c. Sequence diagram



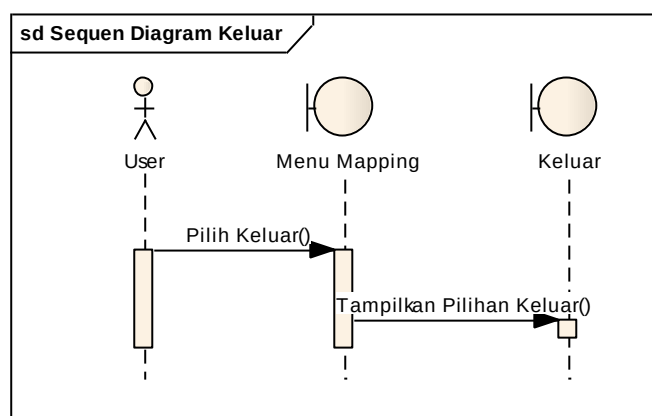
Gambar III.3
Sequence Diagram Aplikasi Pencarian Lokasi Wisata



Gambar III.4
Sequence diagram peta wisata

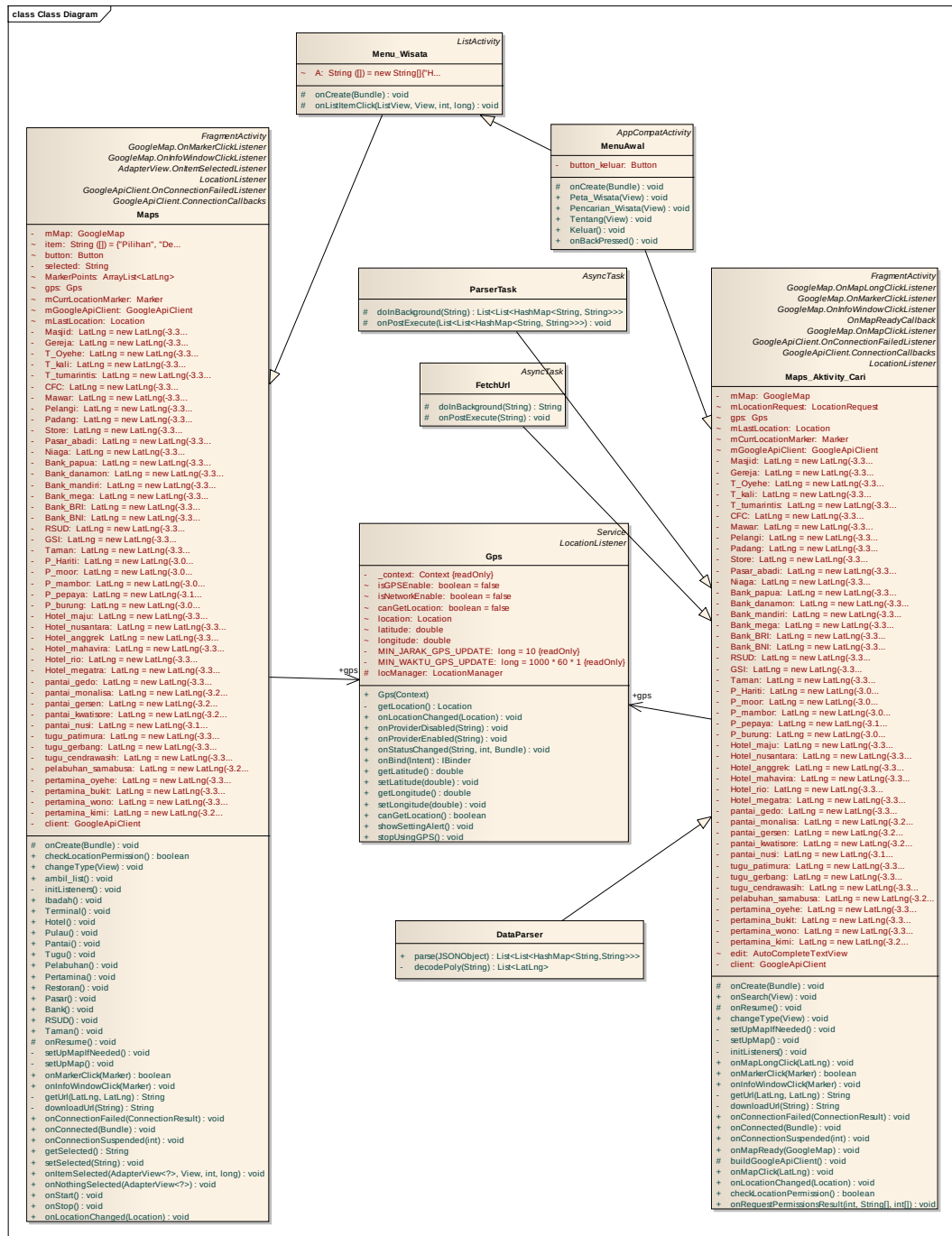


Gambar III.5
Sequence Diagram Pencarian Wisata



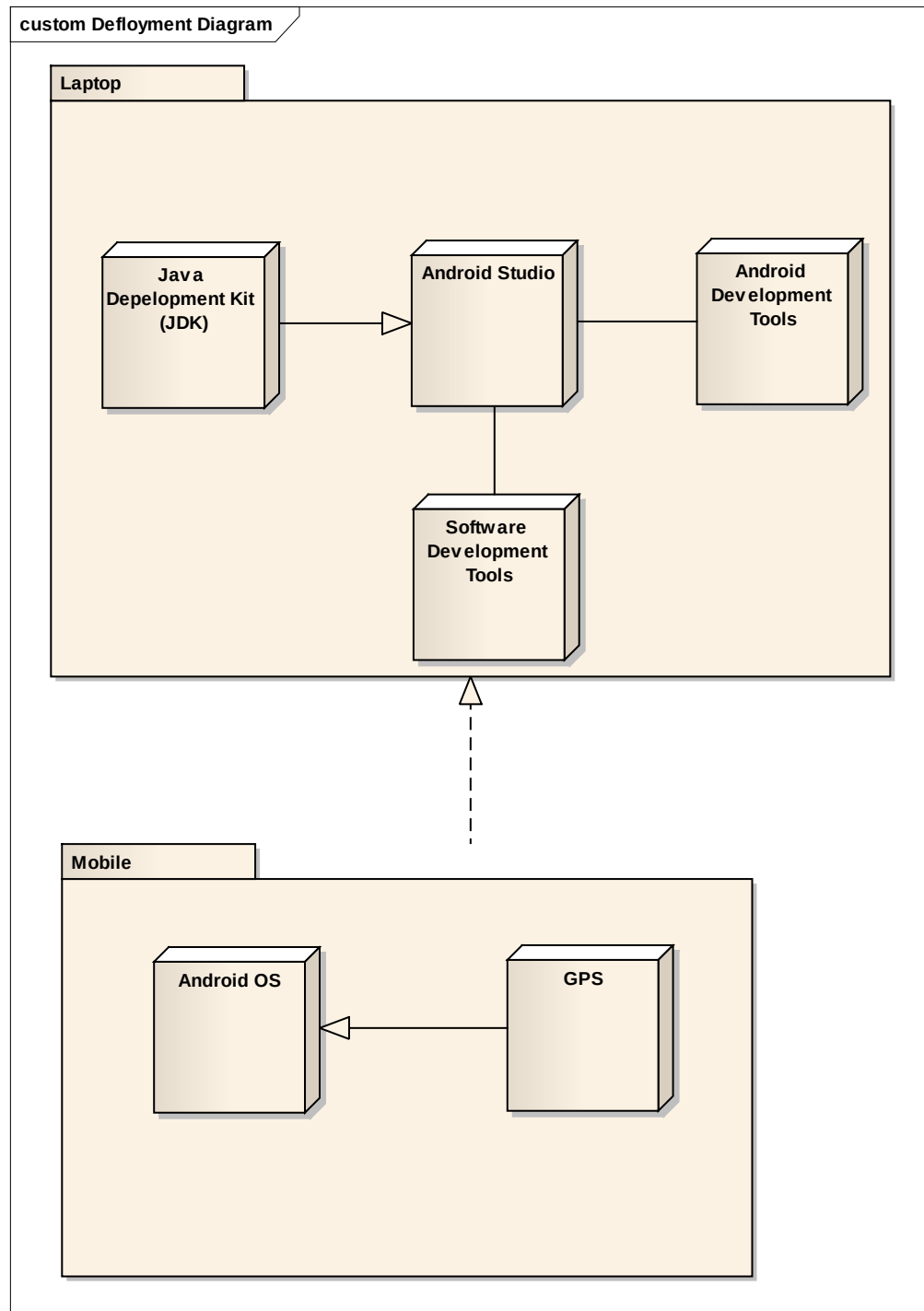
Gambar III.6
Sequence Diagram Keluar

d. Class diagram



Gambar III.7
Class diagram Aplikasi Pencarian Lokasi Wisata

e. Deployment diagram

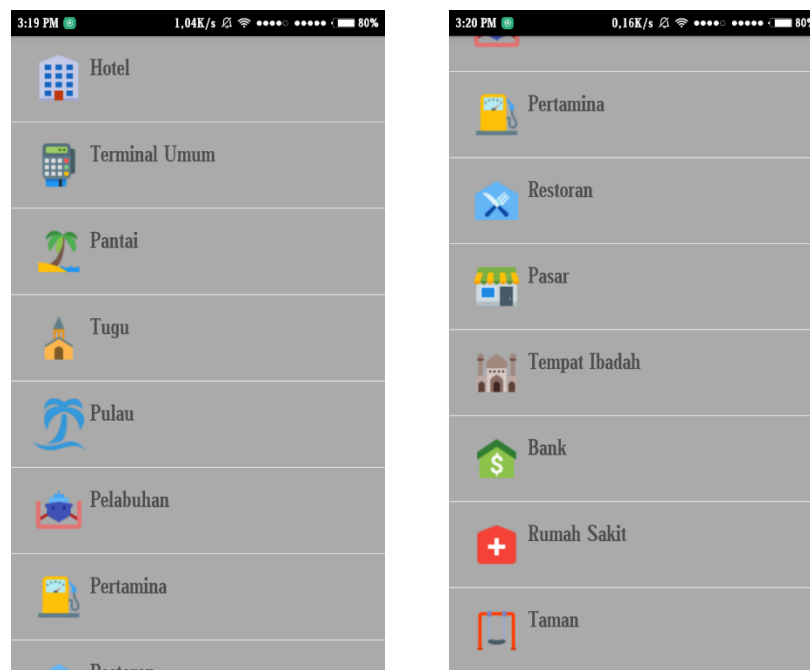


Gambar III.8
Deployment Diagram Aplikasi Pencarian Lokasi Wisata

3.2.3. User Interface

1. Peta Wisata

Pada halaman ini terdapat tiga belas (13) Kategori lokasi, diantaranya adalah Hotel, Terminal Umum, Pantai, Tugu, Pulau, Pelabuhan, Pertamina, Restoran, Pasar, Tempat Ibadah, Bank, Rumah Sakit, dan Taman. Setelah *user* memilih kemana dia akan tujuh, maka aplikasi langsung akan menampilkan pilihan yang dipilih oleh *user*. Dan kemudian untuk dapat *directionnya* *user* bisa langsung mengklik atau menyentuh pada bagian mana dia akan tujuh, maka otomatis aplikasi akan menampilkan Pilihan Detail dan Mulai Perjalanan. Berikut adalah tampilan menu Peta wisata:



Gambar III.9

Tampilan Peta Wisata

3. Keluar

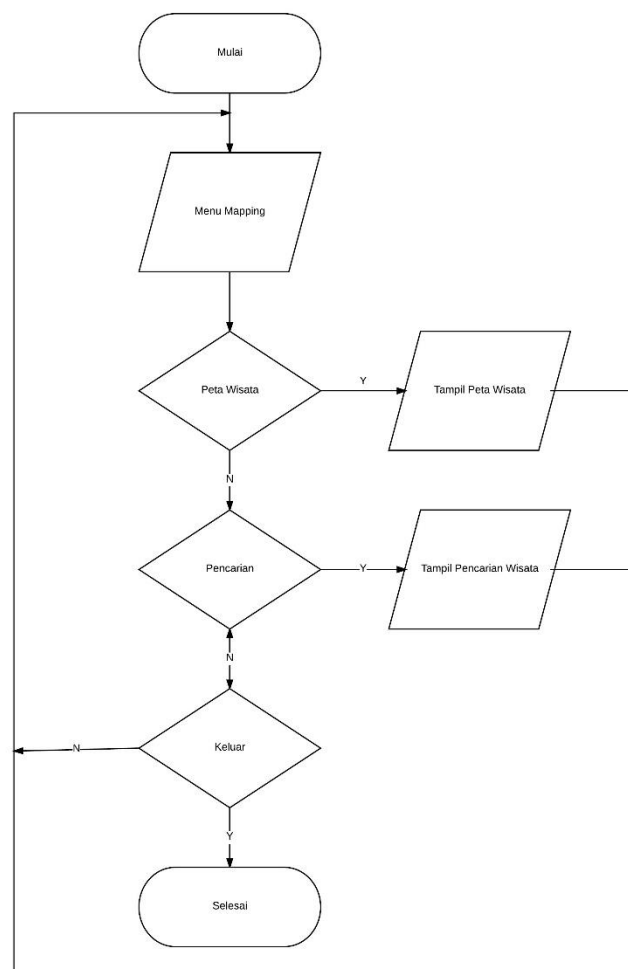
Di halaman keluar ini, user akan melihat tampilan *Pop-Up* Pilihan memilih keluar atau tidak. Jika *Yes* berarti aplikasi akan keluar, dan jika *No* berarti aplikasi tidak akan keluar.



Gambar III.11
Tampilan Keluar

3.3. Implementasi

Perancangan aplikasi pada skripsi ini menggunakan bahasa pemrograman java, *software* yang digunakan untuk membangun aplikasi tersebut adalah Android Studio ADT (*Android Development Tool*). Berikut adalah *flowchart* dari perancangan program:

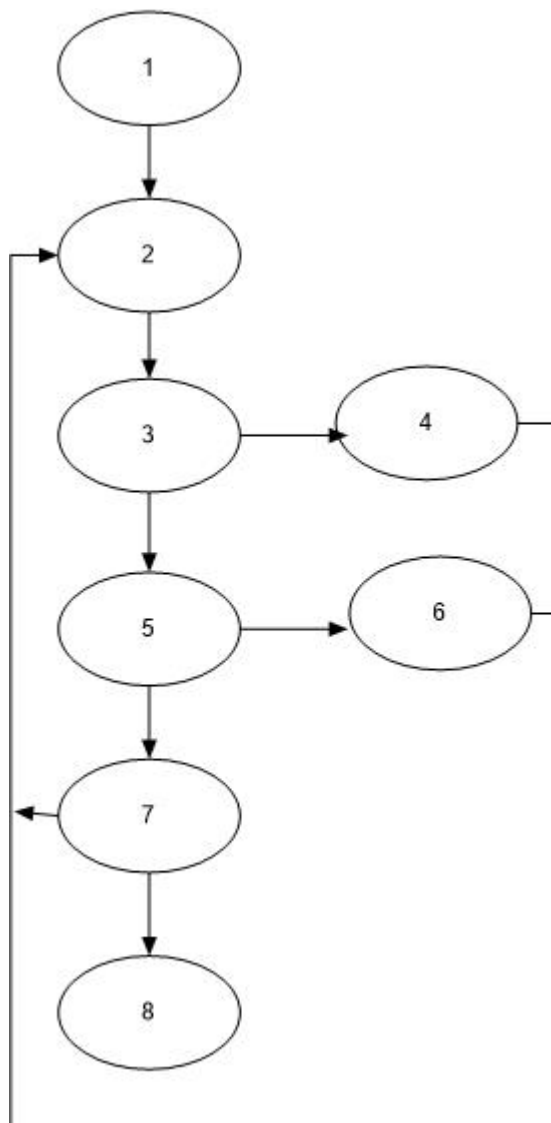


Gambar III.12
Flowchart Aplikasi Pencarian Lokasi Wisata

3.4. Testing

3.4.1. Pengujian White Box Testing

Flowchart



Rancangan Aplikasi Pencarian Lokasi Wisata

Gambar III.13

Pengujian White Box

Kompleksitas siklomatis dari grafik alir *white box* dapat diperoleh dengan

perhitungan:

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

Dimana:

E = Jumlah *Edge* yang ditentukan gambar panah

N = Jumlah simpul grafik alir ditentukan dengan gambar lingkaran

$$V(G) = 9 - 8 + 2 = 3$$

Baris set yang dihasilkan dari jalur *independent* adalah sebagai berikut:

- a. 1-2-3-5-7-8
- b. 1-2-3-4-2
- c. 1-2-3-5-6-2
- d. 1-2-3-5-7-8
- e. Ketika aplikasi dijalankan, maka terlihat bahwa satu set baris yang dihasilkan adalah 1-2-3-5-7-8-1-2-3-4-2-1-2-3-5-6-2-1-2-3-5-7-8

3.4.2. Pengujian Black Box Testing

Dalam pengujian software skripsi ini penulis menggunakan metode pengujian *Black Box Testing*. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *Black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

Tabel III.1
Pengujian *Black box* Aplikasi Pencarian Lokasi Wisata

Input Event	Proses	Output/next Stege	Hasil
Tombol Menu Maps	Menampilkan Layout Menu Maps	Tampilan layout Menu Maps	Sesuai
Tombol Pencarian Maps	Menampilkan layout Pencarian Maps	Menampilkan layout Pencarian Maps	Sesuai
Tombol Serch	Menampilkan layout Maps Direction	Menampilkan layout Maps Direction	Sesuai
Tombol Keluar	Menampilkan dialog keluar	Tempilan dialog keluar	Sesuai

3.5. Support

Dalam membangun aplikasi ini, perangkat keras dan perangkat lunak yang digunakan untuk perancangan dan pengujian aplikasi yang telah berhasil diujikan adalah sebagai berikut:

1. Laptop

Sistem Operasi	Windows 8.1 (64bit)
Video Type	Intel® HD Graphics 3000
Display	14" WXGA LED 1366 x 768
CPU	Core i3 2328 2.2Ghz (2.2Ghz)

Memori	8 GB DDR3
Storage	500 GB Serial ATA
Wireless Network	Data WLAN 802.11 b/g/n

2. Handphone

Sistem Operasi	Android 4.4 (KitKat)
<i>Prosesor</i>	Quad-core 1.6GHz Cortex-A7
Layar	720 x 1280 pixels
Kamera	8 megapiksel
Baterai	Li-Ion 2000 mAh
Storage	8 GB
<i>Memory / RAM</i>	1 GB