

Rancang Bangun Aplikasi Mobile Geotagging Lokasi Musibah dan Bencana Alam Pada Platform Android

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Adi Wiratama
115060807111156



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

Rancang Bangun Aplikasi Mobile Geotagging Lokasi Musibah dan Bencana Alam
Pada Platform Android

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Adi Wiratama
115060807111156

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Juli 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Herman Tolle, ST., MT
NIP: 197408232000121001

Aswin Suharsono, S.T, M.T
NIK: 201102 840919 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

LEMBAR PERNYATAAN

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Juni 2016



Adi Wiratama

NIM: 115060807111156

KATA PENGANTAR

Puji syukur kehadirat Allah SWT karena atas rahmat dan hidayah-Nya, penulis dapat menyelesaikan skripsi dengan judul **“Rancang Bangun Aplikasi Mobile Geotagging Lokasi Musibah dan Bencana Alam Pada Platform Android”**.

Pada penyusunan Skripsi ini tidak semata-mata hasil kerja penulis sendiri, melainkan juga berkat dan bimbingan dan dorongan dari pihak-pihak yang telah membantu, baik secara materi maupun non materi. Maka dari itu penulis ingin mengucapkan banyak terima kasih yang tak terhingga serta penghargaan yang setinggi-tingginya kepada orang-orang yang telah membantu penulis secara langsung maupun tidak langsung kepada yang terhormat:

1. Dr. Herman Tolle, ST., MT dan Aswin Suharsono, S.T, M.T, selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Issa Arwani, S.Kom., M.Sc, selaku Ketua Prodi Informatika / Ilmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan kesempatan kepada penulis untuk menyelesaikan skripsi ini.
3. Bapak, Ibu, dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian, dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesainya skripsi ini.
4. Segenap bapak dan ibu dosen program studi Informatika / Ilmu Komputer beserta seluruh staff administrasi yang telah membantu selama perkuliahan.
5. Seluruh Civitas Akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
6. Teman – teman FILKOM Informatika angkatan 2011 yang telah memberikan masukan dan inspirasi kepada penulis selama menempuh studi dan menyelesaikan skripsi ini.
7. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat kami sebutkan satu persatu.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca terutama mahasiswa FILKOM Universitas Brawijaya.

Malang, 23 Juni 2016

Penulis

adiadiwiratama@yahoo.com

ABSTRAK

Negara Indonesia adalah salah satu Negara yang banyak mengalami bencana alam dan musibah. Tidak jarang musibah dan bencana alam yang terjadi di Indonesia memakan puluhan korban jiwa. Besarnya korban jiwa dan kerugian yang terjadi tidak jarang dikarenakan oleh lambatnya bantuan penanganan yang dilakukan oleh pemerintah. Ini juga disebabkan oleh sulitnya cara masyarakat untuk melaporkan dimana bencana alam dan musibah ini terjadi. Dengan adanya fitur *geotagging* pada smartphone kita bisa memanfaatkannya sebagai sarana pelaporan bencana alam.

Geotagging adalah proses menambahkan data identifikasi geografis ke berbagai media seperti *website* atau gambar. *Geotagging* sendiri sudah menjadi tren yang berkembang pesat pada dunia fotografi digital dan komunitas berbagi foto. Dengan menggunakan geotagging ini kita bisa membuat aplikasi yang dapat mengambil gambar bencana alam dan musibah yang sudah memiliki identitas lokasi didalamnya. Nantinya pelapor cukup melakukan pengambilan gambar di lokasi bencana kemudian melakukan *upload* laporan itu ke *server* yang akan melakukan penanganan pada bencana alam itu.

Dari hasil penelitian yang dilakukan, Aplikasi *geotagging* lokasi bencana alam dan musibah yang dibangun ini berhasil diimplementasikan pada smartphone android dan telah sesuai dengan perancangan dan kebutuhan pengguna. Hasil pengujian fungsionalitas menunjukkan sistem telah sesuai dengan spesifikasi kebutuhan fungsional. Hasil pengujian usability berdasarkan 5 aspek usability testing mencapai rata-rata sebanyak 82.5% yang menunjukkan aplikasi layak diterima dan memudahkan pengguna dengan hasil sangat memuaskan.

Kata kunci: Bencana alam , *Geotagging* dan *Android*

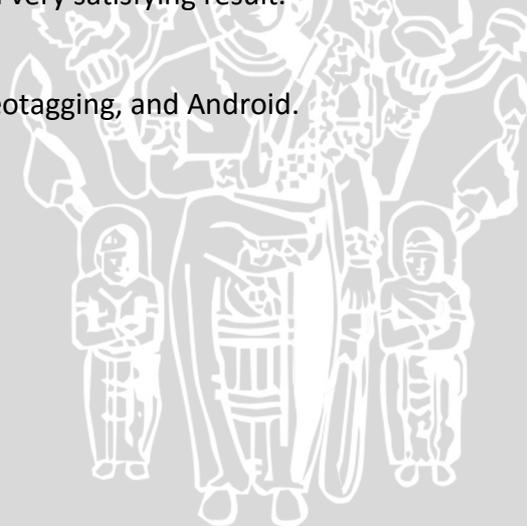
ABSTRACT

Indonesia is one of country that have many kind of disasters. Frequently Indonesian disasters causes many victims. The number of victims and loss that happened is often caused by slow handling by government. This is also caused difficulty of people to make a report where the disaster happened. With geotagging feature on smarphone we can use it for disaster report tool.

Geotagging is a process to add geographic identification to many kind of object, for example website or picture. Geotagging has become evolve trend on digital photographic and various phot community. With geotagging we can make an app that able to take a disaster picture that have location identity on it. Eventually reporter only have to take a picture where disaster happened an then upload the report to server that will take responsibility to resolve the disaster.

Form the result of research that have done, disaster location geotagging app succeed to implemented on android smartphone and already suitable with design and user needs. The result of functionality test show that system already suitable with functionality needs specification. Usability test result according to 5 usability testing aspect gain average 82.5% that shiw this app decent to approve and ease the user with very satisfying result.

Keyword : Disaster, Geotagging, and Android.



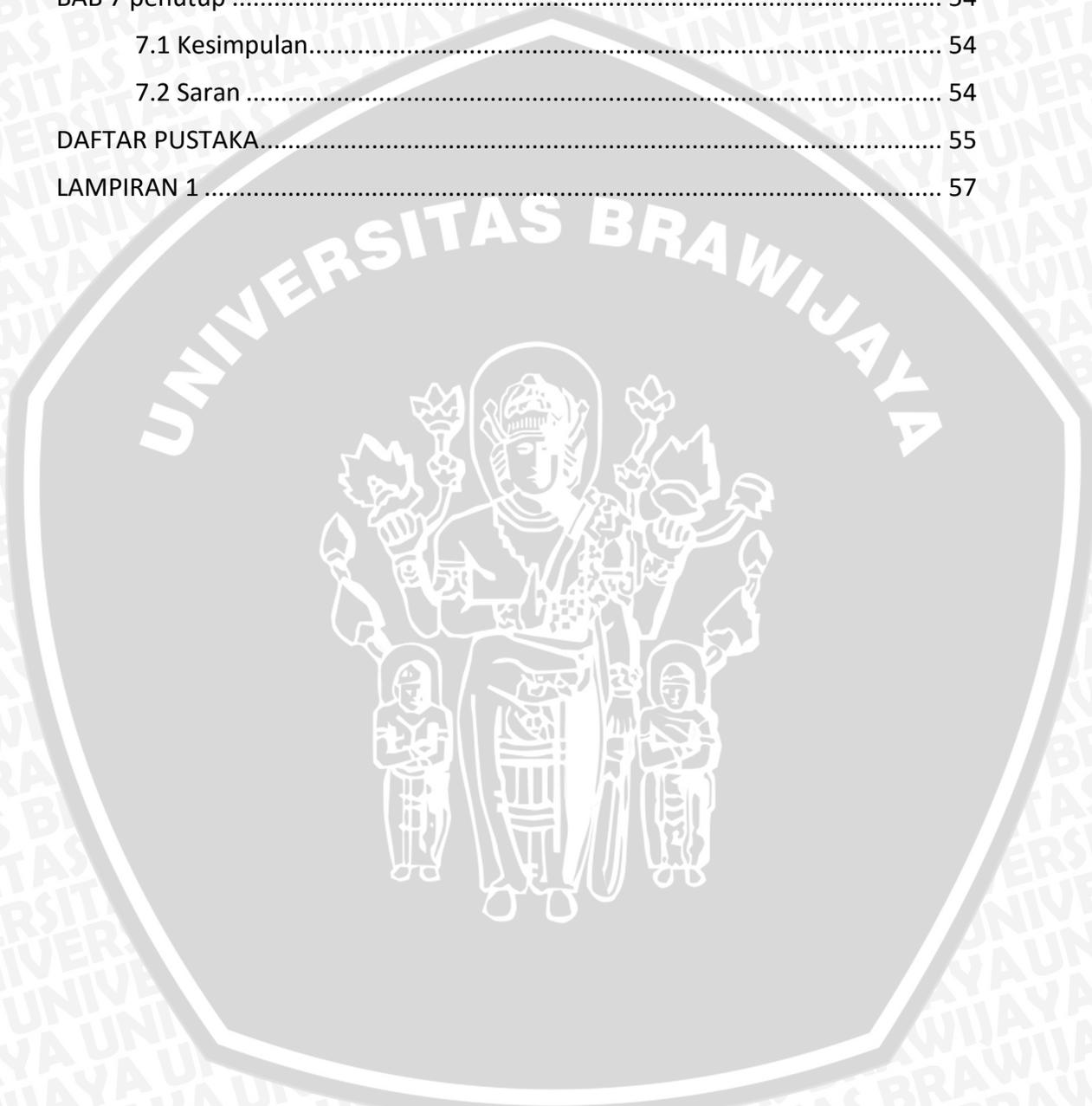
DAFTAR ISI

PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	1
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah.....	2
1.6 Sistematika pembahasan.....	2
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 <i>Web Service</i>	5
2.3 <i>Javascript Object Notation</i>	6
2.4 <i>Google Maps API</i>	8
2.5 <i>Geotagging</i>	8
2.6 Badan Nasional Penanggulangan Bencana.....	9
2.7 Pengujian <i>Usability</i>	10
BAB 3 METODOLOGI	12
3.1 Studi Literature	12
3.2 Analisis Kebutuhan	13
3.3 Perancangan Sistem.....	13
3.4 Implementasi	13
3.5 Pengujian	14
3.6 Kesimpulan dan Saran	14
BAB 4 PERANCANGAN.....	15

4.1 Analisis Kebutuhan	15
4.1.1 Gambaran Umum Sistem	15
4.1.2 Identifikasi Aktor	15
4.1.3 Analisis Kebutuhan Fungsional	16
4.1.4 Analisi Kebutuhan Non Fungsional	17
4.1.5 Diagram Use Case	17
4.1.6 Skenario Use Case	18
4.2 Perancangan	22
4.2.1 Perancangan Aritektural	22
4.2.2 Perancangan Activity Diagram	23
4.2.3 Perancangan Sequence diagram	27
4.2.4 Perancangan Class Diagram	30
4.2.5 Perancangan Basis Data	30
4.2.6 Perancangan Page Flow	31
4.2.7 Perancangan Antarmuka	32
4.2.8 Perancangan Drag and Drop Lokasi Bencana Alam	36
BAB 5 IMPLEMENTASI	38
5.1 Spesifikasi Sistem	38
5.1.1 Spesifikasi Perangkat Keras	38
5.1.2 Spesifikasi perangkat lunak	38
5.2 Implementasi Kode Program	39
5.2.1 Algoritma Pengambilan Gambar	39
5.2.2 Algoritma Upload Laporan	41
5.2.3 Algoritma Edit Lokasi Marker	43
5.3 Implementasi Antarmuka Pengguna	43
5.3.1 Halaman Utama Mobile	43
5.3.2 Halaman Map Activity	44
5.3.3 Halaman Utama Web Server	45
5.3.4 Halaman Daftar Insiden	45
BAB 6 PENGUJIAN DAN ANALISIS	46
6.1 Pengujian	46
6.1.1 Pengujian Validasi	46

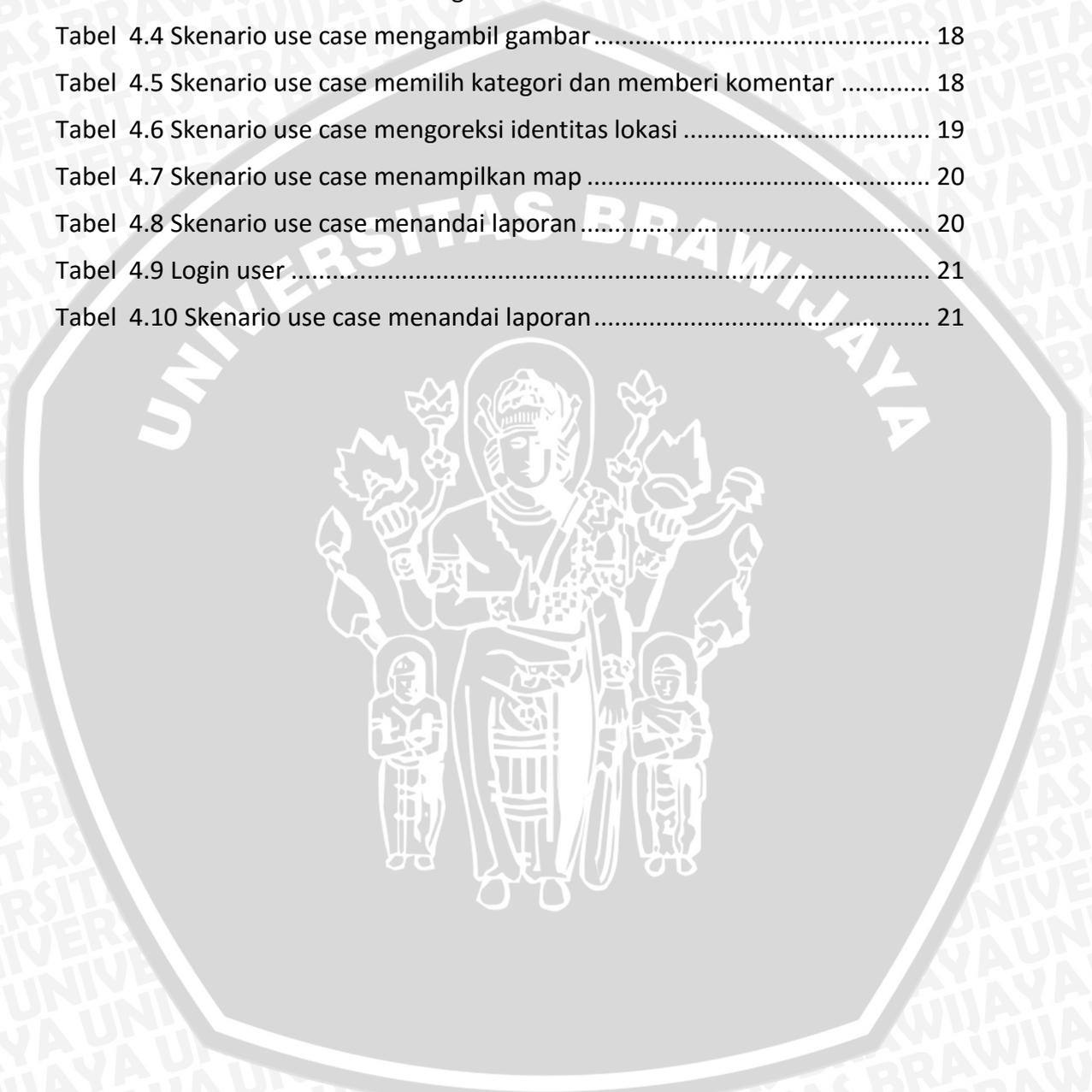


6.1.2 Pengujian <i>Usability</i>	48
6.2 Analisis	50
6.2.1 Analisis Hasil Pengujian Validasi	50
6.2.2 Analisis Hasil Pengujian <i>Usability</i>	51
BAB 7 penutup	54
7.1 Kesimpulan.....	54
7.2 Saran	54
DAFTAR PUSTAKA.....	55
LAMPIRAN 1	57



DAFTAR TABEL

Tabel 4.1 Tabel Identifikasi Aktor.....	16
Tabel 4.2 Tabel Kebutuhan Fungsional.....	16
Tabel 4.3 Tabel kebutuhan non fungsional.....	17
Tabel 4.4 Skenario use case mengambil gambar.....	18
Tabel 4.5 Skenario use case memilih kategori dan memberi komentar.....	18
Tabel 4.6 Skenario use case mengoreksi identitas lokasi.....	19
Tabel 4.7 Skenario use case menampilkan map.....	20
Tabel 4.8 Skenario use case menandai laporan.....	20
Tabel 4.9 Login user.....	21
Tabel 4.10 Skenario use case menandai laporan.....	21



DAFTAR GAMBAR

Gambar 2.1 Gambar Struktur Objek Dalam Format <i>JSON</i>	6
Gambar 2.2 Gambar Struktur <i>Array</i> Dalam Format <i>JSON</i>	7
Gambar 2.3 Gambar Struktur Nilai Dalam Format <i>JSON</i>	7
Gambar 2.4 Gambar Struktur String dalam Format <i>JSON</i>	7
Gambar 2.5 Gambar Struktur numerik dalam Format <i>JSON</i>	8
Gambar 4.1 Diagram use case aplikasi pelaporan musibah dan bencana.....	17
Gambar 4.2 Rancangan arsitektural sistem	22
Gambar 4.3 Activity diagram login.....	23
Gambar 4.4 Activity diagram mengambil gambar	24
Gambar 4.5 Activity diagram memilih kategori menambah komentar.....	24
Gambar 4.6 Activity diagram koreksi identitas lokasi.....	25
Gambar 4.7 Activity diagram login admin.....	26
Gambar 4.8 Activity diagram menandai laporan	26
Gambar 4.9 Activity diagram menampilkan map	27
Gambar 4.10 Diagram sekuensial ambil gambar	27
Gambar 4.11 Diagram sekuensial memilih kategori.....	28
Gambar 4.12 Menampilkan map	28
Gambar 4.13 menampilkan map admin	29
Gambar 4.14 Menandai laporan	29
Gambar 4.15 Class diagram.....	30
Gambar 4.16 Class diagram IncidentMapActivity.....	30
Gambar 4.17 Entity diagram sistem.....	31
Gambar 4.18 Page flow diagram mobile.....	31
Gambar 4.19 Page flow diagram server.....	32
Gambar 4.20 Halaman Utama.....	33
Gambar 4.21 Halaman ambil gambar	34
Gambar 4.22 Halaman menampilkan map	34
Gambar 4.23 Halaman utama server	35

BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia dengan luas area $\pm 1,9$ juta km^2 dan 13.677 buah pulau, serta sebaran populasi yang belum merata dengan komplikasi keadaan social ekonomi masyarakat merupakan suatu modal dasar pembangunan bangsa dan negara. Dengan kondisi wilayah yang sangat luas ini besar juga kemungkinan terjadinya musibah dan bencana alam (Simatupang, 2005).

Sebagian besar daerah yang rawan terhadap bencana alam adalah daerah yang masih terpencil, contohnya daerah yang rawan terhadap bencana tanah longsor. Pada provinsi Jawa Tengah terdapat 327 lokasi rawan tanah longsor, beberapa diantaranya adalah Wonosobo, Batang, Pati, Jepara, Purwodadi dan Kudus (Supriyono, 2014). Contoh lain adalah daerah yang rawan terhadap bencana gunung meletus, diantaranya adalah Kota Salatiga, Kabupaten Cianjur, Kabupaten Purwokerto, dan dataran tinggi Dieng di Kabupaten Wonosobo. Dengan fakta bahwa banyak kejadian musibah dan bencana alam terjadi pada daerah terpencil maka ini menjadi masalah terhadap sulitnya dan efisiensi waktu pelaporan bencana itu sendiri (Supriyono, 2014).

Solusi alternatifnya adalah dengan mengimplementasikan teknologi geotagging pada proses pelaporan keadian bencana alam dan musibah. *Geotagging* adalah proses menambahkan data identifikasi geografis ke berbagai media seperti *website* atau gambar. *Geotagging* sendiri sudah menjadi tren yang berkembang pesat pada dunia fotografi digital dan komunitas berbagi foto. Fenomena dari *geotagging* sudah menghasilkan kesadaran letak pada perangkat multimedia. *Geotagging* dapat menolong pengguna untuk menemukan berbagai macam informasi spesifik suatu lokasi. Misalnya, seseorang dapat menemukan gambar yang diambil di dekat lokasi tertentu yang telah dimasukkan koordinatnya (Joshi, et al., 2010). Tetapi teknologi *geotagging* sendiri memiliki kelemahan, yaitu terkadang letak identitas lokasi yang tertera tidak sesuai dengan lokasi yang sesungguhnya. Maka dari itu pengimplementasian skripsi ini diharap dapat menemukan jalan keluar dari masalah akurasi dari *GPS* ini.

Selain itu, sistem pelaporan dengan *geotagging* juga membutuhkan tingkat fleksibilitas yang tinggi agar dalam hampir semua kondisi user dapat melakukan pelaporan. Fleksibilitas tersebut dijawab dengan memanfaatkan teknologi *mobile* yang memiliki tingkat fleksibilitas yang tinggi (Luckin, 2005).

Skripsi ini bertujuan untuk merancang dan mengimplementasikan aplikasi dengan memanfaatkan kamera dan *geotagging* pada *smartphone* sebagai media untuk pelaporan kejadian musibah dan bencana alam yang ada di Indonesia.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat diuraikan rumusan masalah sebagai berikut:

1. Bagaimana rancangan dan implementasi aplikasi mobile geotagging untuk pelaporan musibah dan bencana alam pada platform android yang mempermudah masyarakat untuk melakukan pelaporan?
2. Bagaimana melakukan *edit* lokasi sebelum melakukan *submit* untuk memastikan lokasi yang ditunjukkan sesuai
3. Bagaimana mengimplementasikan ampilkasi dengan tingkat kemudahan penggunaan yang baik?

1.3 Tujuan

Tujuan dari penelitian ini adalah merancang dan mengimplementasikan sebuah aplikasi mobile geotagging untuk mempermudah penanganan musibah dan bencana alam.

1.4 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

1. Masyarakat umum dapat melaporkan bencana alam dan musibah secara cepat dan mudah.
2. Dengan proses pelaporan yang cepat diharapkan penanganan bencana alam dan musibah bisa lebih cepat dan baik.

1.5 Batasan masalah

Beberapa batasan masalah untuk memfokuskan masalah ini antara lain adalah:

1. Aplikasi ini membutuhkan koneksi ke internet untuk dapat digunakan karena data yang akan digunakan diambil langsung dari bagian *server*.
2. Aplikasi hanya dapat digunakan pada perangkat *smartphone* yang memiliki fitur *geo-tagging*.

1.6 Sistematika pembahasan

Gambaran secara garis besar pembahasan dari keseluruhan isi penelitian untuk setiap bab adalah sebagai berikut:

BAB I PENDAHULUAN

Dalam bab Pendahuluan akan diuraikan tentang latar belakang penulisan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika pelaporan, dan jadwal penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang teori-teori dasar yang diambil dari sumber pustaka dan referensi yang terkait dengan teori-teori pembuatan aplikasi mobile geotagging untuk musibah dan bencana alam pada platform android.

BAB III METODOLOGI PENELITIAN

Menguraikan tentang langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri Studi Literatur, Observasi dan Wawancara, Pengumpulan Data, Analisis Kebutuhan, Perancangan Sistem, Implementasi Sistem, Pengujian Sistem, dan Pengambilan Kesimpulan.

BAB IV HASIL

Pada bab Perancangan akan membahas tentang analisis kebutuhan, perancangan antarmuka pengguna, dan perancangan mesin inferensi dalam membangun aplikasi mobile geotagging untuk musibah dan bencana alam pada platform android.

BAB V IMPLEMENTASI

Pada bab ini akan diulas tentang implementasi berdasarkan metodologi dan perancangan.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab Pengujian dan Analisis akan membahas tentang proses dan hasil pengujian terhadap sistem yang telah dibangun. Selain itu pada bab ini akan memastikan bahwa implementasi program telah sesuai dengan perancangan yang dibuat.

BAB VII PENUTUP

Bab Penutup berisi tentang kesimpulan dan saran dari penulis. Kesimpulan diambil berdasarkan pengujian serta analisis yang telah dilakukan, serta saran berisi tentang rekomendasi dari penulis agar sistem pakar yang telah dibuat dapat dikembangkan kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Perkembangan teknologi perangkat bergerak (*smartphone*) saat ini sudah berkembang sangat pesat dan sudah menjadi kebutuhan pokok bagi sebagian masyarakat. Kegunaannya pun sudah sangat berkembang untuk menyelesaikan masalah dalam kehidupan sehari-hari. Kini fungsi dari *smartphone* bisa digunakan untuk melakukan pengelolaan penanggulangan pada bencana alam dan musibah. Pengelolaan bencana alam dengan menggunakan teknologi *smartphone* sangat mungkin seperti pada penelitian sebelumnya yang dilakukan Amit Gosavi dan S. Vishnu dengan judul penelitian “Disaster Alert and Notification System Via Android Mobile Phone by Using Google Map” . Pada penelitian ini dijelaskan bahwa dikarenakan terus berubahnya cuaca dan struktur geografis bumi maka mengakibatkan seringnya terjadi bencana alam. Dikarenakan oleh ini peringatan bencana alam yang baik dan pedoman evakuasi dapat menyelamatkan banyak nyawa. Sehingga diusulkan aplikasi alarm peringatan bencana alam dan jalur evakuasi yang memanfaatkan teknologi *Google Map*. Karena penggunaan ponsel pintar di negara-negara berkembang meningkat, sistem ini diterapkan pada ponsel android. Sistem ini terbagi menjadi tiga bagian yaitu *Disaster Management System (DMS)*, perangkat android yang sudah terdapat aplikasi ini dan pengguna. Update dari bencana (tsunami, angin topan atau banjir) diletakkan di *DMS* oleh kantor cuaca lokal. Untuk mendapatkan pemberitahuan otomatis dari bencana alam yang akan datang *device* pengguna harus didaftarkan terlebih dahulu pada *Disaster Management System*, sedangkan yang tidak terdaftar dapat mendapat pemberitahuan secara manual. Pengguna tetap terhubung dengan *DMS* untuk mendapatkan pemberitahuan terbaru yang diperoleh dari *GSM*. Aplikasi ini juga akan mendeteksi dimana lokasi *user* sekarang untuk memberikan rute evakuasi yang aman dari bencana alam yang sedang terjadi (Gosavi & Vishnu, 2014).

Pada penelitian sebelumnya berjudul “Rancang Bangun Aplikasi Mobile Social Geotagging Keragaman Budaya Indonesia Pada Platform Android” yang dilakukan oleh Eka Aryansyah seorang mahasiswa Universitas Brawijaya. Peneliti mengembangkan sebuah aplikasi yang bertujuan untuk mempromosikan budaya Indonesia melalui internet dengan mengunggah foto-foto kebudayaan Indonesia. Aplikasi ini memiliki fitur geotagging yang berguna untuk memberikan identitas lokasi saat dilakukan pengambilan foto ke dalam data EXIF, jadi pengguna tidak perlu memberikan identitas lokasi foto dengan cara manual. Setelah itu foto yang telah diambil tadi akan di *upload* ke bagian *server* yang nantinya akan ditampilkan pada sebuah website sebagai sebuah *icon marker* yang terdapat pada sebuah peta Indonesia. *Website* ini menampilkan semua foto beserta informasinya yang telah diunggah oleh pengguna dari pihak *mobile*.

Pada bagian *website* foto-foto kebudayaan memiliki sebuah fitur untuk *login* sebagai admin yang berfungsi untuk mengakses fitur-fitur khusus diantara

lainnya adalah melakukan eksekusi terhadap laporan yang kurang layak dari pengguna aplikasi pada perangkat mobile. Bentuk tanggapan terhadap laporan dapat berupa pengiriman pemberitahuan pada pengguna *mobile* melalui *email* kemudian melakukan penghapusan terhadap laporan yang sudah dikirimkan. Bagi pengguna yang tidak memiliki akses untuk admin maka hanya dapat melihat sebaran foto-foto kebudayaan Indonesia yang sudah dilaporkan bersama dengan informasi yang terdapat pada laporan itu (Aryansyah, 2016).

Pada penelitian yang dilakukan Jovilyn T.B, Fajardo dan Carlos M. Oppus yang berjudul “*A Mobile Disaster Management System Using the Android Technology*”. Mereka mengembangkan aplikasi bernama MyDisasterDroid yang berguna sebagai sarana pelaporan bencana alam. Aplikasi ini bisa menentukan rute terbaik yang harus dilewati penolong untuk sampai ditempat kejadian. Penentuan rute pada aplikasi ini sangat fleksibel karena kita diperbolehkan untuk mengganti prioritas atau diperpanjang rutenya dengan upaya yang lebih ringan.

Tetapi kekurangannya dikarenakan proses pelaporan aplikasi ini menggunakan *SMS* atau menginputkan langsung dari aplikasi itu sendiri sehingga besar kemungkinan orang bisa melakukan pelaporan palsu. Ini dikarenakan tidak adanya bukti bahwa di lokasi itu benar-benar terjadi bencana alam atau tidak (Jovilyn Therese B, n.d.).

2.2 Web Service

Web service merupakan *software* yang digunakan untuk mendukung interaksi dari mesin melalui jaringan. *Web service* memiliki layanan terbuka untuk kepentingan integrasi data dan kolaborasi informasi yang bisa diakses melalui internet oleh berbagai pihak menggunakan teknologi yang dimiliki oleh masing-masing pengguna (Sutanta, 2012)

Teknologi *web service* telah banyak digunakan oleh organisasi bisnis untuk mengkolaborasikan sistem-sistem di dalam organisasi. *Web service* menjanjikan banyak keuntungan seperti peningkatan produktivitas, efisiensi dan akurasi. Fokus utama dari *web service* ini adaka untuk membantu agar aplikasi bisa berkomunikasi antar satu dan yang lainnya. Semua *protocol* transport yang berbasis internet seperti *HTTP*, *SMTP* (*Simple Mail Transfer Protocol*) dan *FTP* (*File Transfer Protocol*) bisa diimplementasikan untuk saling bertukar pesan melalui *web service*.

Alasan menggunakan *web service* adalah kemudahan dalam penggunaan kembali dan berbagi logika yang sama dengan klien yang beragam seperti *mobile*, desktop dan aplikasi *web*. Jangkauan *web service* yang luas karena *web service* bergantung pada standar yang terbuka, dapat beroperasi pada *platform* yang berbeda, serta tidak bergantung pada teknologi eksekusi yang mendasarinya. Semua *web service* setidaknya menggunakan *HTTP* dan format pertukaran data standar berupa *XML*, *JSON* atau media lain. Seain itu, *web service* menggunakan *HTTP* dengan dua cara yang berbeda yaitu sebagai *protocol* standar untuk

menentukan perilaku standar serta sebagai media transportasi untuk menyampaikan data (Daigneau, 2012).

Dalam penelitian ini, web service inilah yang akan melakukan pemrosesan data. *Web service* ini nantinya akan menerima permintaan data dari perangkat bergerak yang nanti selanjutnya melakukan proses ke server hingga data yang telah didapat dikembalikan lagi ke perangkat bergerak untuk selanjutnya ditampilkan dengan tampilan visual pada perangkat bergerak tersebut.

2.3 Javascript Object Notation

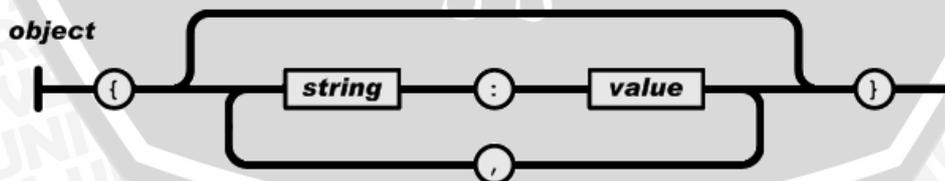
Javascript Object Notation adalah format teks yang memfasilitasi pertukaran struktur data antara semua bahasa pemrograman. *JSON* juga merupakan sebuah format standar terbuka yang menggunakan teks yang dapat dibaca manusia untuk mengirimkan objek data yang terdiri dari pasangan atribut – nilai.

JSON dibangun di atas dua struktur :

- Sebuah kumpulan dari pasangan nama / nilai. Di beberapa bahasa, hal ini direalisasikan sebagai sebuah *objek*, *record*, *struct*, *dictionary*, *hash table*, *keyed list* atau *array asosiatif*.
- Sebuah daftar nilai terurut. Di hamper semua bahasa, hal ini direalisasikan sebagai sebuah *array*, *vector*, *list* atau *sequence*. Kedua hal di atas merupakan data struktur yang universal. Sebenarnya semua bahasa pemrograman modern mendukung mereka dalam satu bentuk atau lainnya. Hal ini menjadi masuk akal bahwa format data yang bisa dipertukarkan dengan bahasa pemrograman juga didasarkan pada struktur ini. (Team, 2015)

Di *JSON* sendiri, keduanya mengambil bentuk-bentuk:

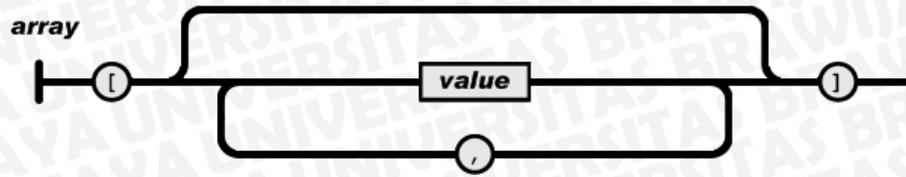
- Sebuah objek merupakan sebuah set dari pasangan nama / nilai. Sebuah objek dimulai dengan { (kurawal kiri) dan diakhiri dengan } (kurawal kanan). Setiap nama diikuti dengan : (titik dua) dan nama / nilai dipisahkan oleh , (koma). Seperti terlihat gambar di bawah.



Gambar 2.1 Gambar Struktur Objek Dalam Format JSON

Sumber : (Team, 2015)

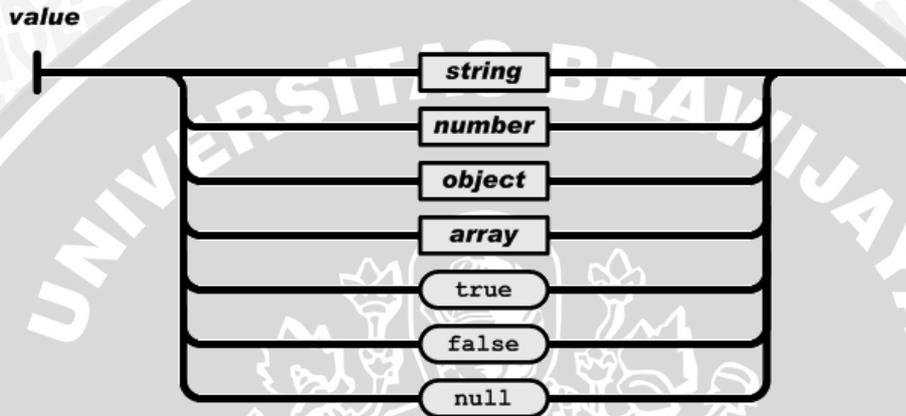
- Sebuah *array* merupakan sebuah daftar nilai yang terurut. Sebuah *array* dimulai dengan [(kurung siku kiri) dan diakhiri dengan] (kurung siku kanan). Nilainya dipisahkan oleh , (koma). Seperti terlihat gambar di bawah



Gambar 2.2 Gambar Struktur Array Dalam Format JSON

Sumber : (Team, 2015)

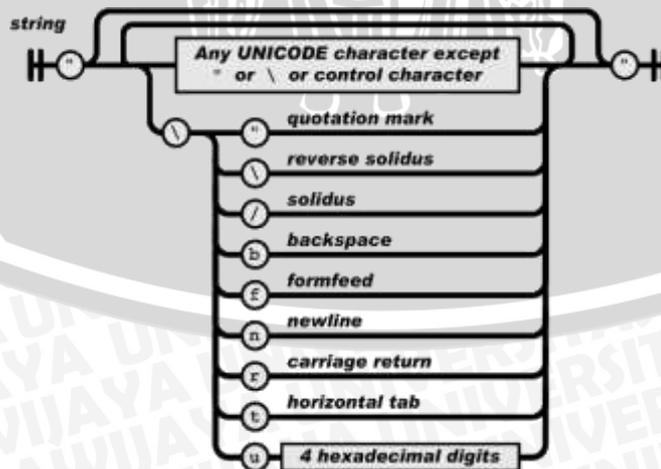
- c. Sebuah nilai dapat berupa string dalam petik ganda, atau sebuah angka, atau *true* or *false* or *null*, atau sebuah objek atau sebuah *array*. Struktur ini dapat bersarang.



Gambar 2.3 Gambar Struktur Nilai Dalam Format JSON

Sumber : (Team, 2015)

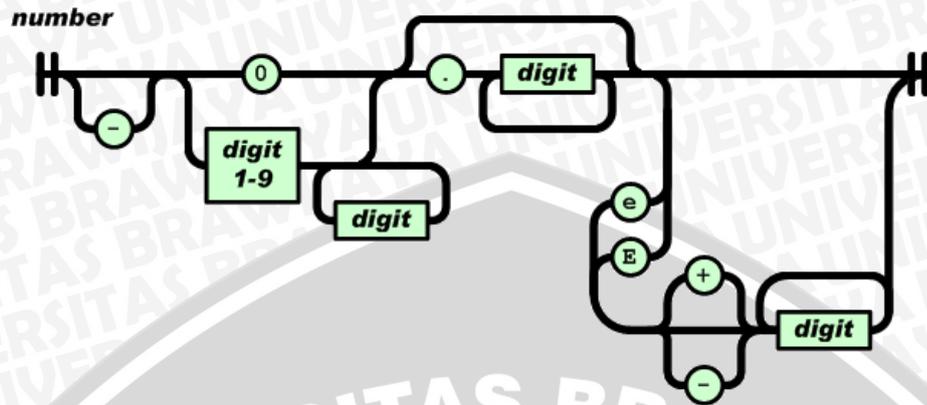
- d. Sebuah string merupakan rangkaian dari nol atau lebih karakter *Unicode*, yang dibungkus dalam petik ganda, menggunakan *backslash escapes*. Sebuah karakter di representasikan sebagai sebuah karakter di string. Sebuah string di *JSON* sangat mirip dengan string di *C* atau *Java*.



Gambar 2.4 Gambar Struktur String dalam Format JSON

Sumber : (Team, 2015)

- e. Sebuah angka di *JSON* sangat mirip dengan sebuah angka di *C* atau *Java*, kecuali format *octal* dan *hexadecimal* tidak digunakan.



Gambar 2.5 Gambar Struktur numerik dalam Format JSON

Sumber : (Team, 2015)

- f. *Whitespace* dapat dimasukkan di antara token dari pasangan. Pengecualian beberapa rincian *encoding*, yang benar-benar menggambarkan bahasanya.

2.4 Google Maps API

Google maps merupakan sebuah layanan peta dunia virtual berbasis web yang disediakan oleh *Google Inc.* *Google maps* dapat dilekatkan sebagai elemen dalam tata letak antarmuka pengguna yang di rancang oleh pemrogram. Untuk menghubungkan aplikasi perangkat lunak dengan *google maps* diperlukan sebuah kunci yang diistilahkan sebagai *API Key*.

Google maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *google maps* merupakan suatu peta yang dapat dilihat dari sebuah browser. Kita dapat menambahkan fitur *google maps* dalam *web* ataupun blog yang telah kita buat. *Google maps API* adalah suatu *library* yang berbentuk *javascript*. Pustaka eksternal yang ditawarkan oleh *Google* untuk menghubungkan aplikasi android dengan *google maps* adalah *com.google.android.maps*. Untuk memanfaatkan pustaka ini, diperlukan *google API's add-on* sehingga dapat dibuat aplikasi berbasis *google maps* pada *android SDK* dengan akses terhadap data *google maps* (Firdaus, 2013).

2.5 Geotagging

Geotagging merupakan proses penambahan informasi geospasial pada berbagai media seperti foto dan video dalam bentuk metadata. Media yang telah mengalami proses geotagging akan memiliki informasi koordinat berupa *longitude* (bujur), *latitude* (lintang) dan *altitude* (ketinggan). Hal tersebut memungkinkan media dapat diposisikan secara tepat pada peta (Entwistle, J. 2010). Terdapat tiga cara melakukan *geotagging* pada suatu media sebagai berikut (Nandipati, 2011):

- a. Kamera digital yang dilengkapi dengan GPS.
Beberapa kamera digital dan kamera pada telepon seluler telah dilengkapi dengan built-in GPS. Semua hasil foto dari perangkat tersebut secara otomatis akan memiliki informasi lokasi seperti *longitude*, *latitude* dan *latitude* pada EXIF header-nya.
- b. Kamera digital disinkronkan dengan GPS terpisah.
Kamera digital yang tidak memiliki built-in GPS tetap dapat melakukan geotagging dengan cara menyingkronkan dengan GPS. Melalui bantuan perangkat lunak, waktu yang direkam oleh kamera akan disinkronkan dengan waktu yang direkam oleh GPS dan berdasarkan waktu tersebut koordinat yang direkam oleh GPS akan ditambahkan sebagai informasi EXIF pada foto. Faktor terpenting pada cara ini adalah mengatur agar waktu pada kamera dan GPS sama.
- c. *Geocoding* manual.
Pada *Geocoding* manual, informasi lokasi ditambahkan ke EXIF foto dengan cara menambahkan koordinat atau dengan memilih lokasi pada peta saat melakukan upload gambar ke internet. Tingkat akurasi posisi tergantung pada tools atau GPS *receiver* yang digunakan untuk mendapatkan koordinat.

Pada penelitian ini metode Geotagging yang digunakan untuk melakukan penambahan informasi lokasi pada metadata foto adalah Geocoding manual. Algoritme penambahan ditulis di dalam *script* aplikasi dan otomatis akan berjalan saat pengguna mengunggah foto. Lokasi yang ditambahkan merupakan lokasi pengguna yang didapatkan oleh aplikasi.

2.6 Badan Nasional Penanggulangan Bencana

Badan Nasional Penanggulangan Bencana (disingkat BNPB) adalah sebuah Lembaga Pemerintah Non Departemen yang mempunyai tugas membantu Presiden Republik Indonesia dalam: mengkoordinasikan perencanaan dan pelaksanaan kegiatan penanganan bencana dan kedaruratan secara terpadu; serta melaksanakan penanganan bencana dan kedaruratan mulai dari sebelum, pada saat, dan setelah terjadi bencana yang meliputi pencegahan, kesiapsiagaan, penanganan darurat, dan pemulihan.

BNPB dibentuk berdasarkan Peraturan Presiden Nomor 8 Tahun 2008. Sebelumnya badan ini bernama Badan Koordinasi Nasional Penanggulangan Bencana yang dibentuk berdasarkan Peraturan Presiden Nomor 83 Tahun 2005, menggantikan Badan Koordinasi Nasional Penanggulangan Bencana dan Penanganan Pengungsi yang dibentuk dengan Keputusan Presiden Nomor 3 Tahun 2001. (Badan Nasional Penanggulangan Bencana, 2016)

Petugas dari BNPB inilah yang nantinya memegang aplikasi dari sisi *server* sebagai penerima laporan kejadian bencana alam. Setelah menerima laporan admin inilah yang nantinya menginformasikan kepada petugas untuk langsung terjun ke lapangan.

2.7 Pengujian Usability

Terdapat banyak pengertian mengenai usability. Namun pengertian usability (kebergunaan) secara umum adalah kemampuan perangkat lunak dalam mempermudah pengguna untuk melakukan sebuah tugas. Nielsen memberikan pengertian usability sebagai atribut kualitas yang dapat menilai seberapa mudah antarmuka pengguna digunakan (Santoso & Ferdina, 2015). Berdasarkan ISO 9241-11 untuk mengukur usability harus mencakup tiga aspek berikut (Santoso & Ferdina, 2015) (Brooke, 1996).

- a. Effectiveness (tingkat akurasi dan kesempurnaan yang dicapai pengguna saat menjalankan tugas tertentu).
- b. Efficiency (tingkat konsumsi sumber daya yang digunakan saat menjalankan tugas).
- c. Satisfaction (pendapat subjektif pengguna dalam menggunakan sistem).

Sedangkan menurut Jacob Nielsen (Munaiseche, 2012) komponen usability terdiri dari 5 hal sebagai berikut.

- a. Learnability (mudah dipelajari)
- b. Efficiency (efisiensi)
- c. Memorability (mudah diingat)
- d. Errors (kesalahan dan keamanan)
- e. Satisfaction (kepuasan)

Pengujian Usability merupakan pengujian yang berhubungan dengan pengguna akhir baik secara langsung maupun tidak langsung untuk menilai bagaimana pengguna merasakan perangkat lunak dan bagaimana mereka berinteraksi dengan perangkat lunak (Simarmata, 2010). Menurut Munaiseche, C. P. C. (2012), pengujian usability merupakan evaluasi perangkat lunak yang bertujuan untuk mengetahui seberapa besar kemudahan suatu antarmuka dapat digunakan oleh pengguna saat berinteraksi dengan sistem. Pengujian usability juga dapat diartikan sebagai pengujian yang bertujuan untuk mengetahui seberapa efektif, efisien dan memuaskan sebuah website menurut penggunanya (Santoso & Ferdina, 2015).

Pengujian usability dilakukan dengan menggunakan kuesioner. Terdapat beberapa kuesioner yang dapat digunakan untuk pengujian usability seperti System Usability Scale (SUS), Questionnaire for User Interface Satisfaction (QUIS), Software Usability Measurement Inventory (SUMI) dan Post-Study Usability Questionnaires. Menurut penelitian yang dilakukan oleh Sauro (2011) serta Tullis dan Stetson (2004) menunjukkan bahwa SUS merupakan kuesioner yang dapat dijadikan alat pengujian usability yang valid dan reliabel (Santoso & Ferdina, 2015). Sehingga pada penelitian ini dalam melakukan pengujian usability digunakan kuesioner SUS.

Menurut Nielsen (2014) terdapat tiga hal yang perlu diperhatikan dalam melakukan pengujian usability yaitu (Ersa, 2015):

1. Buat Task serealistis mungkin.
2. Buat Task yang mudah dikerjakan.
3. Hindari memberikan petunjuk dan menjelaskan langkah-langkah yang harus dikerjakan.



BAB 3 METODOLOGI

Struktur yang ada pada Gambar 3.1 adalah tahap pengerjaan pengerjaan pembuatan aplikasi pelaporan bencana alam dengan menggunakan fitur *geotagging*. Penelitian diawali dari studi literatur, Analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan kesimpulan & saran.



Gambar 3. 1 Diagram Alir Metodologi Penelitian

3.1 Studi Literature

Studi literature adalah penjelasan tentang dasar teori yang dipakai sebagai pendukung pada penelitian ini. Studi literature dibutuhkan karena kualitas dari penelitian ini berdasarkan pada studi literature yang dipelajari oleh penulis. Teori-teori pendukung yaitu :

1. *Geotaging*
2. Sistem Operasi Android
3. *Android SDK*
4. *UML(Unified Modelling Language)*

3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk mendapatkan semua kebutuhan perangkat lunak. Metode analisis yang digunakan adalah *object-oriented analysis* dengan menggunakan bahasa pemodelan UML. Analisis kebutuhan dalam penelitian ini dilakukan dengan mengidentifikasi kebutuhan aplikasi *Geotagging* Lokasi Musibah dan Bencana Alam.

Proses analisis kebutuhan terdiri dari tiga tahap yaitu identifikasi aktor, analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional. Identifikasi aktor merupakan tahap pengidentifikasian elemen yang dapat berinteraksi dan melakukan perubahan pada sistem. Identifikasi aktor menjelaskan deskripsi dari setiap aktor.

Analisis kebutuhan fungsional dilakukan untuk mengetahui fungsi apa saja yang harus terdapat di dalam aplikasi dan sistem. Terdapat dua analisis kebutuhan yang dikerjakan yaitu analisis kebutuhan fungsional aplikasi perangkat bergerak yang digunakan masyarakat untuk melakukan pelaporan bencana dan analisis kebutuhan fungsional sistem penampil (*website*) laporan bencana alam dan musibah. Setiap kebutuhan fungsional diberikan idenfier dan akan menjadi sebuah *usecase* dalam *usecase* diagram. Kemudian setiap *usecase* dalam *usecase diagram* dijabarkan dalam tabel skenario *usecase*. Skenario *usecase* menjelaskan bagaimana suatu *usecase* dikerjakan, siapa yang mengerjakan dan apa tujuan dari *usecase* tersebut.

Sedangkan analisis kebutuhan non-fungsional merupakan proses analisis untuk menentukan atribut atau kualitas secara keseluruhan dari sistem. Analisis kebutuhan non-fungsional dilakukan terhadap aplikasi perangkat bergerak pelaporan kejadian musibah dan bencana alam.

3.3 Perancangan Sistem

Perancangan dapat dilakukan setelah melakukan analisis kebutuhan. Pada tahap perancangan, dilakukan identifikasi terhadap *class – class* yang dibutuhkan, dan kemudian dimodelkan pada *class diagram*. Kemudian hubungan dan interaksi antar objek diidentifikasi dan dimodelkan dengan *sequence diagram*. Selanjutnya adalah perancangan antar muka pengguna.

3.4 Implementasi

Bahasa pemrograman yang digunakan adalah *java* dengan memanfaatkan *platform android*. Dalam implementasi proses pengambilan gambar diambil dengan menggunakan kamera kemudian diidentifikasi lokasinya dengan teknologi *geotagging*. Seluruh data disimpan pada *server* sehingga tidak harus menyimpan data yang sama dua kali. Data di *server* akan dilakukan *update* secara berkala agar selalu tersimpan data yang terbaru. Selanjutnya akan dilakukan simulasi implementasi secara langsung pada *hardware*.

3.5 Pengujian

Pada bagian ini dilakukan pengujian aplikasi *mobile geotagging* lokasi musibah dan bencana alam dengan cara metode pengujian validasi yaitu pengujian aplikasi melakukan pembuatan laporan bencana alam yang diawali dengan pengambilan gambar, pemilihan kategori, penambahan info, koreksi lokasi dan *upload* laporan. Ketika pengujian pertama ini berhasil maka bisa dikatakan pengujian pada tahap ini tervalidasi.

Selanjutnya dilakukan pengujian dari sisi *server*, yang pertama dilakukan adalah melihat map sebaran bencana alam apakah semua sudah bisa ditampilkan dengan baik tanpa ada masalah. Selanjutnya pengujian fitur penandaan kejadian bencana alam yang sudah ditangani, nantinya bencana alam yang sudah ditangani akan memiliki tanda yang berbeda dari bencana alam yang belum ditangani. Jika fitur ini sudah bisa dijalankan dengan benar maka tahap ini sudah tervalidasi

3.6 Kesimpulan dan Saran

Tahap kesimpulan dan saran hanya bisa dilakukan setelah tahap perancangan perangkat lunak, implimentasi perangkat lunak, dan pengujian perangkat lunak telah selesai dikerjakan. Kesimpulan berisi analisa terhadap sistem perangkat lunak dan hasil analisis dari pengujian yang telah dilakukan.

Setelah kesimpulan didapat selanjutnya adalah pengambilan saran untuk memperbaiki celah dan kesalahan yang ada pada penelitian untuk menyempurnakan penelitian yang telah dibuat. Kemudian penelitian ini bisa dijadikan sebagai refrensi untuk penelitian dikemudian hari.



BAB 4 PERANCANGAN

Pada bab ini dibahas tentang analisis dan perancangan perangkat lunak aplikasi pelaporan musibah dan bencana alam memanfaatkan teknologi geotagging berbasis Android. Proses perancangan aplikasi ini meliputi dua tahap. Tahap pertama adalah analisis kebutuhan. Di dalam analisis kebutuhan terdapat berapa tahapan yaitu gambaran umum aplikasi, identifikasi aktor, analisis kebutuhan fungsional, kemudian dimodelkan kedalam use case. Sedangkan pada tahap perancangan terdiri dari perancangan arsitektural, perancangan activity diagram, perancangan sequence diagram, perancangan class diagram, perancangan basis data dan perancangan page flow

4.1 Analisis Kebutuhan

4.1.1 Gambaran Umum Sistem

Konsep dari “Rancang Bangun Aplikasi *Mobile Geotagging* Lokasi Musibah dan Bencana Alam Pada *Platform Android*” adalah melaporkan kejadian musibah atau bencana alam menggunakan teknologi *geotagging* yang disematkan pada foto yang diambil saat itu juga ditempat kejadian kemudian dikirimkan ke *web server* sebagai laporan bencana alam.

Pada *prototype* aplikasi ini akan digunakan sebuah *smartphone android* dengan menggunakan kamera sebagai perangkat pengambilan gambar. Output dari aplikasi ini berupa sebuah gambar yang sudah memiliki identitas lokasi yang siap dikirimkan ke server.

Aplikasi pelaporan bencana alam ini nantinya akan dibagi menjadi dua sisi yaitu pengguna dan admin. Dari sisi pengguna cara kerja aplikasi ini adalah dengan mengambil gambar menggunakan kamera, setelah itu informasi identitas lokasi yang dimiliki oleh gambar nantinya akan diambil untuk menentukan lokasi pengambilan gambar pada *google map*. Gambar yang diambil ini nantinya akan ditambahkan informasi komentar dan kategori bencana alam untuk memisahkan antara laporan bencana alam yang satu dengan yang lainnya.

Sedangkan dari sisi admin aplikasi ini bertugas untuk menerima laporan yang dikirimkan oleh pengguna ke server. Pada bagian web server nantinya akan dapat melihat map sebaran dari lokasi pelaporan bencana alam dan juga memiliki fitur untuk menandai bencana alam. Fitur ini berguna untuk membedakan bencana alam yang sudah ditangani dan yang belum ditangani. Nantinya petugas dari BNPB yang akan memegang posisi admin dari server ini.

4.1.2 Identifikasi Aktor

Pada tahap ini dilakukan proses identifikasi terhadap aktor-aktor yang berperan atau berinteraksi dengan aplikasi ini. Pada Tabel 4.1 memperlihatkan aktor-aktor yang terlibat beserta penjelasannya masing-masing.

Tabel 4.1 Tabel Identifikasi Aktor

Aktor	Deskripsi
<i>User</i>	Pengguna aplikasi yang mengambil gambar kemudian melaporkannya ke admin.
Admin	Pengguna yang memegang aplikasi pada bagian server untuk menerima laporan dari <i>user</i>

4.1.3 Analisis Kebutuhan Fungsional

Pada Tabel 4.2 ditunjukkan daftar kebutuhan fungsional yang memuat fitur utama yang dimiliki oleh aplikasi tersebut.

Tabel 4.2 Tabel Kebutuhan Fungsional

Kode	Kebutuhan Fungsional	Use Case
F01	Aplikasi mampu mengambil gambar menggunakan kamera	Mengambil gambar
F02	Aplikasi mampu menampilkan fitur agar <i>user</i> dapat memilih kategori bencana dan menambahkan komentar pada laporan yang akan dikirim	Memilih kategori dan menambah komentar
F03	Aplikasi mampu menampilkan fitur agar <i>user</i> dapat mengoreksi jika identitas lokasi yang didapat kurang tepat	Mengoreksi identitas lokasi
F04	Server mampu menampilkan map sebaran lokasi bencana alam	Menampilkan Map
F05	Server mampu menandai untuk menandakan laporan sudah ditangani	Menandai laporan
F06	Aplikasi mampu memberikan fitur login untuk <i>user</i> pada <i>mobile</i>	<i>Login user</i>
F07	Aplikasi mampu memberikan fitur login untuk <i>admin</i> pada <i>server</i>	<i>Login admin</i>

4.1.4 Analisa Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah kebutuhan aplikasi yang tidak diminta oleh pengguna namun dibutuhkan agar aplikasi yang dibuat dapat berfungsi sesuai dengan kebutuhan pengguna.

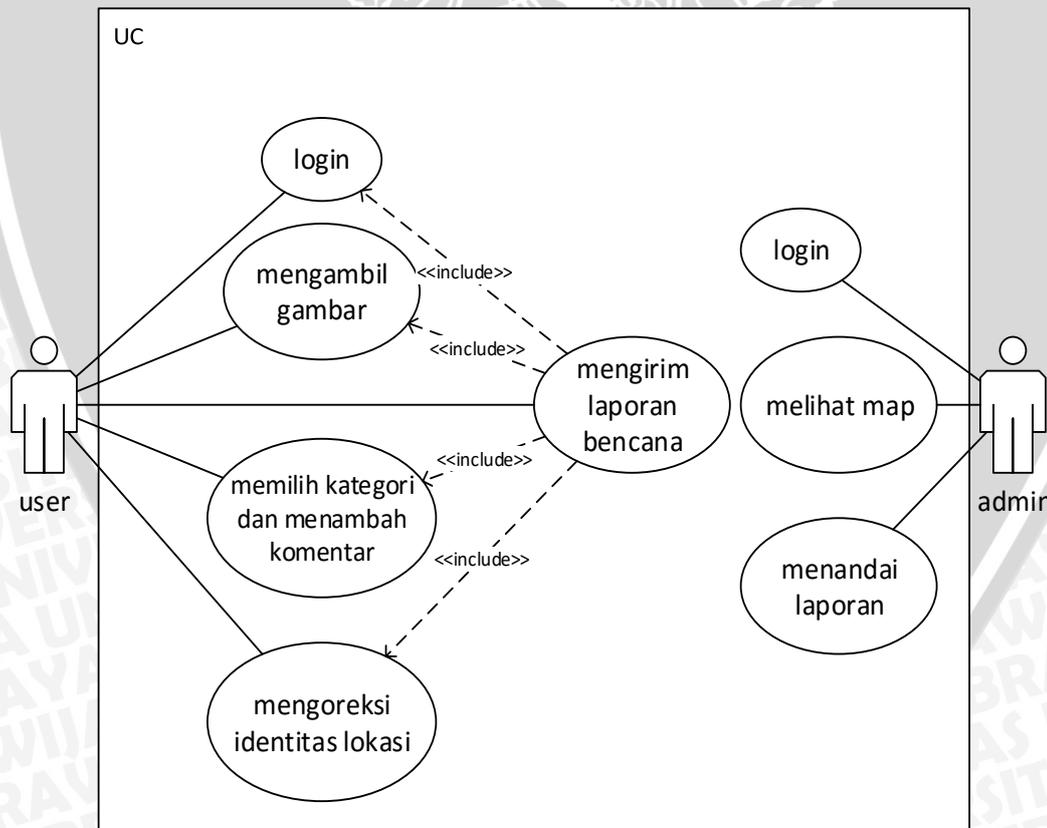
Analisis kebutuhan non fungsional dari aplikasi ini dijelaskan pada Tabel 4.3 berikut ini.

Tabel 4.3 Tabel kebutuhan non fungsional

Parameter	Dsekripsi Kebutuhan
<i>Usability</i>	Desain <i>interface</i> dirancang agar mudah dimengerti dan digunakan oleh pengguna

4.1.5 Diagram Use Case

Diagram use case merupakan diagram *UML(Unified Modeling Language)* yang menggambarkan fungsionalitas dari sebuah sistem. Gambar 4.1 menunjukkan use case dari aplikasi pelaporan bencana alam. Pada use case tersebut digambarkan sistem memiliki aktor yaitu user (pengguna) yang mampu mengambil gambar dan mengirimkan ke server dan server yang mampu menerima laporan bencana alam.



Gambar 4.1 Diagram use case aplikasi pelaporan musibah dan bencana

4.1.6 Skenario Use Case

Tabel 4.4 merupakan penjabaran dari use case upload gambar.

Tabel 4.4 Skenario use case mengambil gambar

Kode use case	F01
Nama use case	Mengambil gambar
Aktor	<i>User</i>
Tujuan	Untuk mendapatkan gambar bencana alam
Deskripsi	Fitur yang ada dalam aplikasi yang berguna mengambil gambar bencana yang diinginkan
Pra-Kondisi	User mentrigger sistem untuk melakukan pengambilan gambar melalui kamera <i>smartphone</i>
Pos-Kondisi	User akan mendapatkan gambar bencana alam
Alur-utama	Aktifitas 1. User membuka aplikasi 2. Aplikasi akan menampilkan menu pengambilan gambar 3. Aplikasi mengambil gambar. 4. Aplikasi menampilkan gambar yang sudah di ambil

Pada Tabel 4.5 menjelaskan proses user menambahkan komentar pada laporan.

Tabel 4.5 Skenario use case memilih kategori dan memberi komentar

Kode use case	F02
Nama use case	Memilih kategori dan memberi komentar
Aktor	<i>User</i>
Tujuan	Untuk memilih kategori bencana dan menambah komentar atau info pada laporan bencana

Deskripsi	Fitur yang ada dalam aplikasi berguna untuk menyediakan sarana <i>user</i> untuk memilih kategori bencana dan memberikan komentar atau info tentang kejadian bencana yang dilaporkan
Pra-Kondisi	User telah mendapatkan gambar dengan identitas lokasi yang sesuai. Kemudian <i>user</i> diberi pilihan untuk mengisi kolom komentar.
Pos-Kondisi	User telah memilih kategori dan memberikan tambahan komentar.
Alur-utama	Aktifitas <ol style="list-style-type: none"> 1. <i>User</i> telah menjalankan fitur ambil gambar. 2. <i>User</i> diberi pilihan untuk memilih kategori bencana dan mengisi kolom komentar

Pada Tabel 4.6 menjelaskan proses menampilkan map.

Tabel 4.6 Skenario use case mengoreksi identitas lokasi

Kode <i>use case</i>	F03
Nama <i>use case</i>	Mengoreksi identitas lokasi
Aktor	<i>User</i>
Tujuan	Mengoreksi identitas lokasi yang kurang tepat
Deskripsi	Fitur yang ada dalam aplikasi berguna untuk mengoreksi identitas lokasi yang didapatkan secara otomatis dengan teknologi <i>geotagging</i> .
Pra-Kondisi	User telah melakukan pengambilan gambar. Hasilnya maka akan muncul gambar yang sudah memiliki identitas lokasi secara otomatis.
Pos-Kondisi	User mendapatkan identitas lokasi gambar

	yang sesuai.
Alur-utama	<p>Aktifitas</p> <ol style="list-style-type: none"> 1. User telah menjalankan fitur ambil gambar dan mendapatkan gambar bencana. 2. Teknologi <i>geotagging</i> memberikan identitas lokasi secara otomatis pada gambar. 3. <i>User</i> mengoreksi identitas lokasi jika dirasa kurang tepat. 4. Aplikasi menampilkan gambar dengan identitas lokasi yang tepat

Pada Tabel 4.7 menjelaskan proses mengirim laporn ke server.

Tabel 4.7 Skenario use case menampilkan map

Kode <i>use case</i>	F04
Nama <i>use case</i>	Menampilkan map
Aktor	<i>Admin</i>
Tujuan	Untuk menampilkan map lokasi bencana alam
Deskripsi	Fitur yang ada dalam aplikasi berguna Untuk menampilkan lokasi laporan bencana alam dala bentuk map.
Pra-Kondisi	<i>User</i> berhasil melakukan pengiriman laporan bencana alam.
Pos-Kondisi	<i>Admin</i> dapat melihat lokasi tempat pelaporan bencana alam.
Alur-utama	<p>Aktifitas</p> <ol style="list-style-type: none"> 1. <i>Admin</i> membuka halaman utama <i>server</i> 2. <i>Admin</i> melihat laporan bencana alam

Pada Tabel 4.8 menjelaskan proses admin menandai laporan.

Tabel 4.8 Skenario use case menandai laporan

Kode <i>use case</i>	F05
Nama <i>use case</i>	Menandai laporan

Aktor	<i>Admin</i>
Tujuan	Untuk menandai laporan dari <i>user</i> .
Deskripsi	Fitur yang ada dalam aplikasi berguna untuk menandai laporan dari <i>user</i> yang sudah ditangani.
Pra-Kondisi	Admin telah mendapatkan laporan bencana dari <i>user</i> .
Pos-Kondisi	Laporan yang sudah ditangani diberi tanda.
Alur-utama	Aktifitas <ol style="list-style-type: none"> 1. Admin menerima laporan bencana alam 2. Admin menandai laporan bencana alam yang sudah ditangani

Pada Tabel 4.9 menjelaskan proses *login user* dari sisi *mobile*.

Tabel 4.9 Login user

Kode <i>use case</i>	F06
Nama <i>use case</i>	<i>Login user</i>
Aktor	<i>User</i>
Tujuan	Untuk melakukan <i>login</i> dari sisi <i>user</i> .
Deskripsi	Fitur yang ada dalam aplikasi berguna untuk <i>user</i> yang akan melakukan <i>login</i> ke dalam aplikasi
Pra-Kondisi	<i>User</i> membuka aplikasi
Pos-Kondisi	<i>User</i> masuk kedalam aplikasi
Alur-utama	Aktifitas <ol style="list-style-type: none"> 1. <i>User</i> memasukkan <i>username</i> dan <i>password</i> 2. <i>User</i> menekan tombol <i>login</i>

Pada Tabel 4.10 menjelaskan proses *login admin* dari sisi *server*.

Tabel 4.10 Skenario use case menandai laporan

Kode <i>use case</i>	F07
Nama <i>use case</i>	<i>Login admin</i>

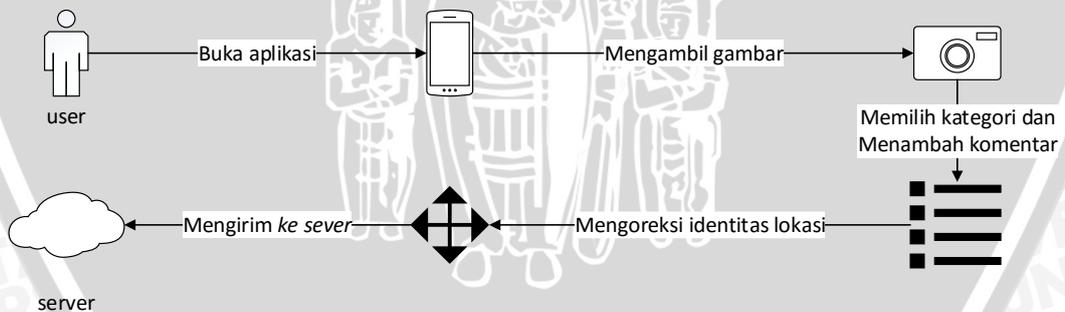
Aktor	<i>Admin</i>
Tujuan	Untuk <i>login</i> dari sisi admin.
Deskripsi	Fitur yang ada dalam aplikasi berguna untuk <i>admin</i> yang akan melakukan <i>login</i> ke dalam server
Pra-Kondisi	<i>Admin</i> membuka aplikasi
Pos-Kondisi	<i>Admin</i> masuk kedalam aplikasi
Alur-utama	Aktifitas <ol style="list-style-type: none"> 3. <i>Admin</i> memasukkan <i>username</i> dan <i>password</i> 4. <i>User</i> menekan tombol <i>login</i>

4.2 Perancangan

Pada tahap ini dilakukan proses perancangan sistem berdasarkan analisis kebutuhan yang dilakukan pada tahap sebelumnya. Pada tahap perancangan akan dimulai dengan merancang arsitektural sistem, *activity diagram*, *sequence diagram*, *class diagram*, basis data, dan *page flow*.

4.2.1 Perancangan Aritektural

Perancangan arsitektural menjelaskan gambaran umum kinerja dari sistem yang akan dibuat secara keseluruhan.



Gambar 4.2 Rancangan arsitektural sistem

Pada Gambar 4.2 digambarkan rancangan arsitektural system yang dimulai dengan user membuka aplikasi kemudian menangkap objek bencana alam yang berupa gambar melalui kamera. Setelah itu dengan teknologi geotagging gambar secara otomatis akan memiliki identitas lokasi. User kemudian memeriksa apakah identitas lokasi yang secara otomatis terdaat pada gambar sudah sesuai atau belum, jika belum maka user bisa melakukan koreksi identitas lokasi dengan menggeser letak pin yang menandakan dimana lokasi gambar tersebut.

Kemudian ada kolom komentar, kolom ini bertujuan agar user bisa

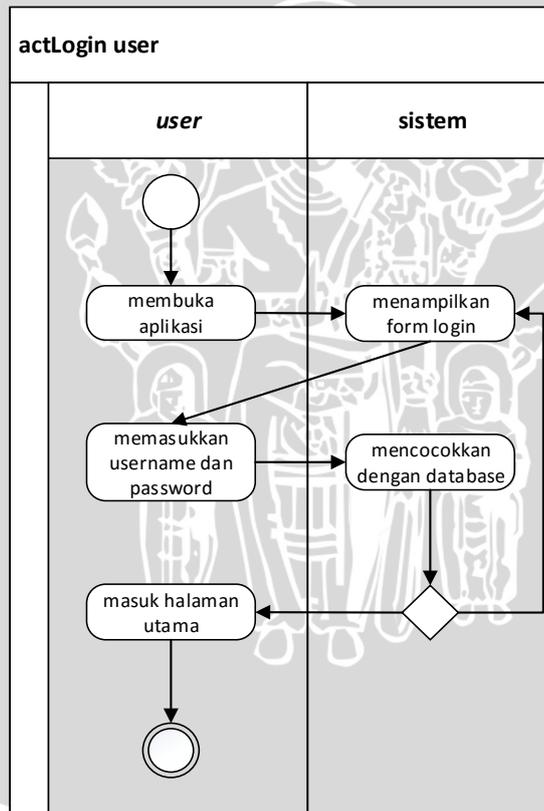
menambahkan info tentang laporan bencana alam jika memang dibutuhkan. Terakhir adalah proses upload laporan, jadi laporan bencana alam yang tadi telah dibuat oleh user di upload ke server untuk diterima oleh admin.

4.2.2 Perancangan Activity Diagram

Activity diagram merupakan representasi grafis (diagram) dari seluruh tahapan alur kerja. diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem.

4.2.2.1 Activity diagram login user

Pada Gambar 4.3, *activity login* dimulai dari *user* membuka aplikasi kemudian sistem akan menampilkan form login. Setelah *user* mengisi form login dengan username dan password maka sistem akan melakukan pengecekan apakah sudah sesuai atau belum. Jika sesuai dengan *database* maka user akan ditampilkan halaman utama.

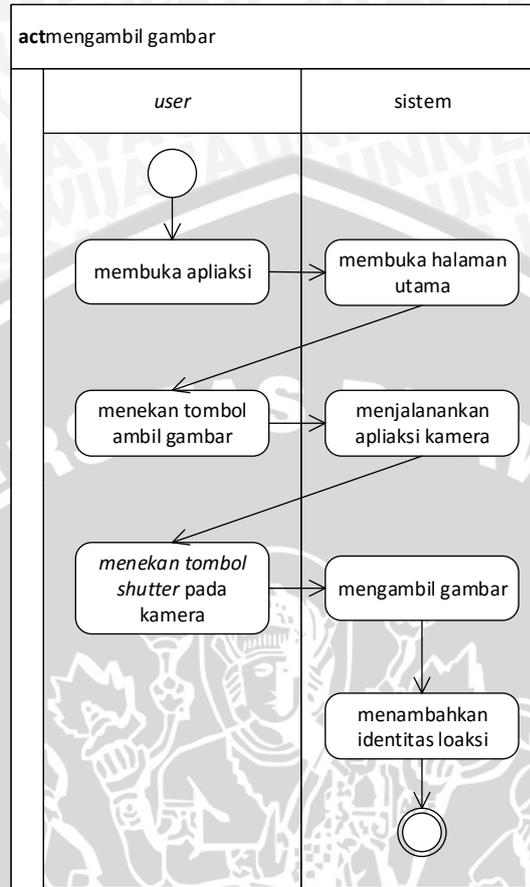


Gambar 4.3 Activity diagram login

4.2.2.2 Activity diagram mengambil gambar

Pada Gambar 4.4, *activity diagram* mengambil gambar dimulai dengan user membuka aplikasi. Sistem menampilkan halaman utama dari aplikasi yang berisi menu ambil gambar. User menekan tombol ambil gambar kemudian melakukan

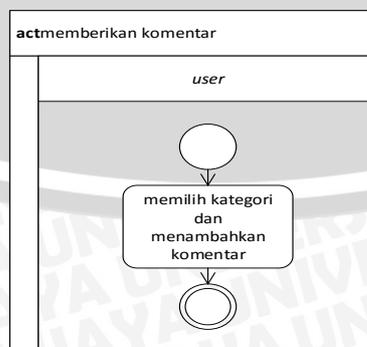
proses pengambilan gambar bencana. Setelah itu identitas lokasi akan secara otomatis terdapat pada gambar yang sudah diambil



Gambar 4.4 Activity diagram mengambil gambar

4.2.2.3 Activity diagram memilih kategori dan menambahkan komentar

Activity diagram yang sesuai dengan Gambar 4.5 merupakan proses untuk memilih kategori dan menambahkan komentar pada laporan bencana alam. Komentar bisa berupa info untuk memperjelas apa yang sedang terjadi di lokasi kejadian. Jika dirasa penambahan komentar tidak diperlukan maka user tidak perlu menambah komentar dan melanjutkan ke tahap yang selanjutnya.



Gambar 4.5 Activity diagram memilih kategori menambah komentar

4.2.2.4 Activity diagram koreksi identitas lokasi

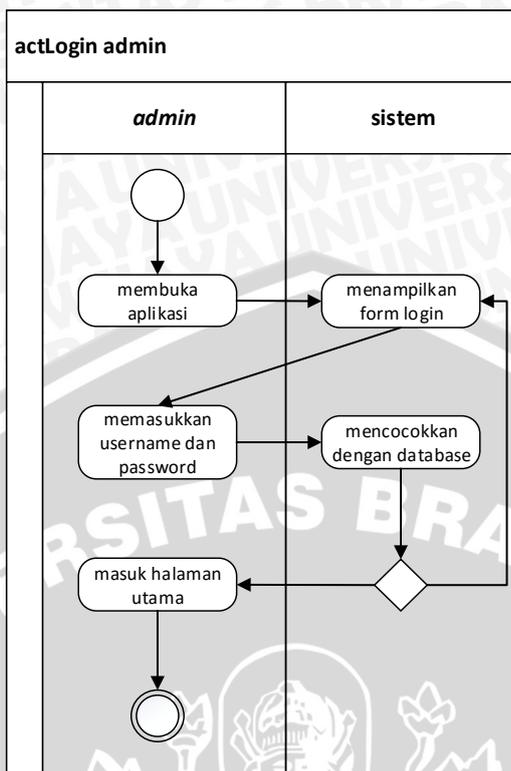
Activity diagram yang sesuai dengan Gambar 4.6 merupakan proses untuk mengoreksi jika identitas lokasi yang secara otomatis ditentukan dengan teknologi geotagging dirasa kurang tepat. Jika dirasa penempatan identitas lokasi sudah tepat maka proses ini tidak perlu dilakukan



Gambar 4.6 Activity diagram koreksi identitas lokasi

4.2.2.5 Activity diagram login admin

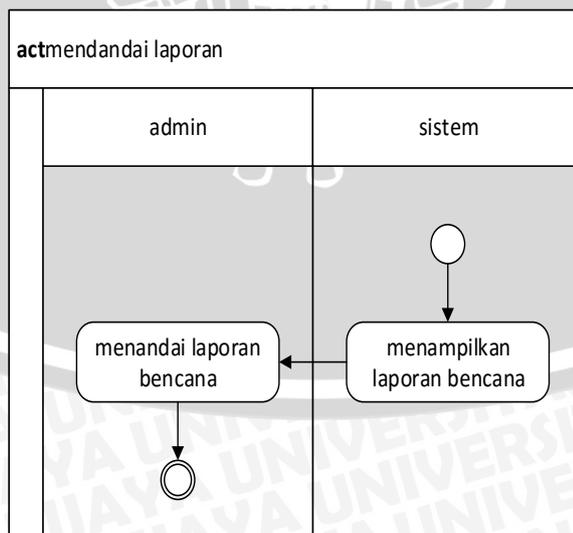
Pada Gambar 4.7, activity *login* dimulai dari *admin* membuka aplikasi kemudian sistem akan menampilkan form login. Setelah *admin* mengisi form login dengan username dan password maka sistem akan melakukan pengecekan apakah sudah sesuai atau belum. Jika sesuai dengan *database* maka admin akan ditampilkan halaman utama.



Gambar 4.7 Activity diagram login admin

4.2.2.6 Activity diagram menandai laporan

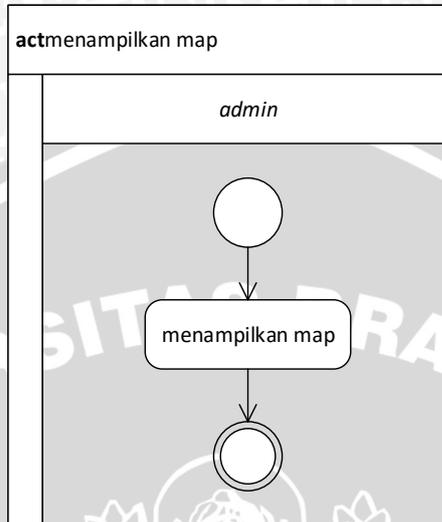
Activity diagram yang sesuai dengan Gambar 4.8 merupakan proses yang dilakukan *admin* untuk menandai laporan bencana alam yang sudah tidak digunakan lagi. Jadi setelah *user* melakukan mengirimkan laporan bencana alam ke *server*, maka akan muncul laporan berupa pin lokasi pada sebuah map. Setelah itu *admin* akan menandai laporan yang sudah ditangani.



Gambar 4.8 Activity diagram menandai laporan

4.2.2.7 Activity diagram menampilkan map

Activity diagram yang sesuai dengan Gambar 4.9 merupakan proses untuk melihat *map*. Jadi setelah laporan berhasil dikirimkan ke *server* maka *admin* dapat melihat sebaran lokasi bencana alam yang ada di *map*.

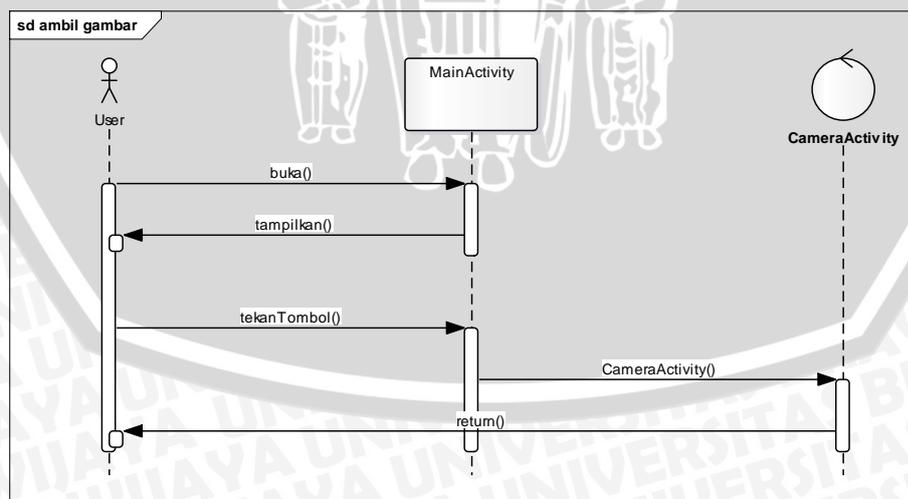


Gambar 4.9 Activity diagram menampilkan map

4.2.3 Perancangan Sequence diagram

Diagram skuensial menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek. Oleh karena itu untuk menggambar diagram skuensial maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

4.2.3.1 Diagram Sekuensial ambil gambar

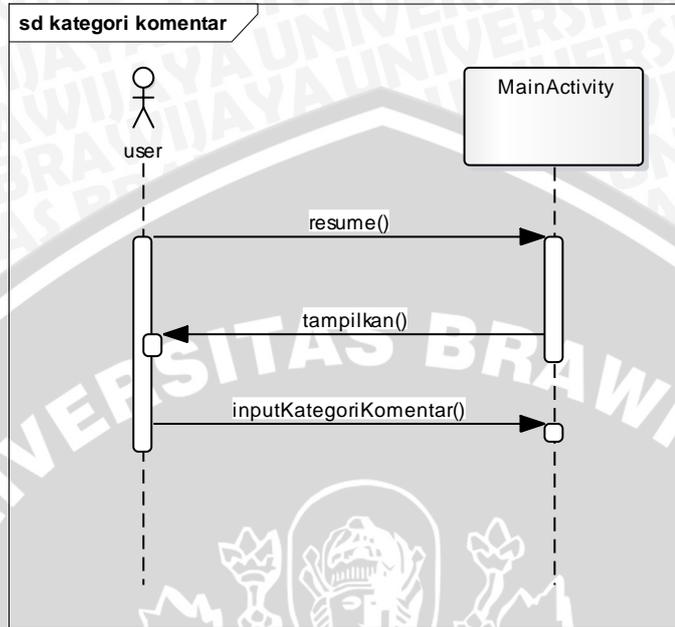


Gambar 4.10 Diagram sekuensial ambil gambar

Gambar 4.10 adalah sequence diagram yang digunakan saat *user* akan melakukan proses pengambilan gambar. Pertama *user* membuka halaman

MainActivity kemudian menekan tombol *capture* dan diteruskan pada halaman CameraActivity yang akan diteruskan pada proses pengambilan gambar.

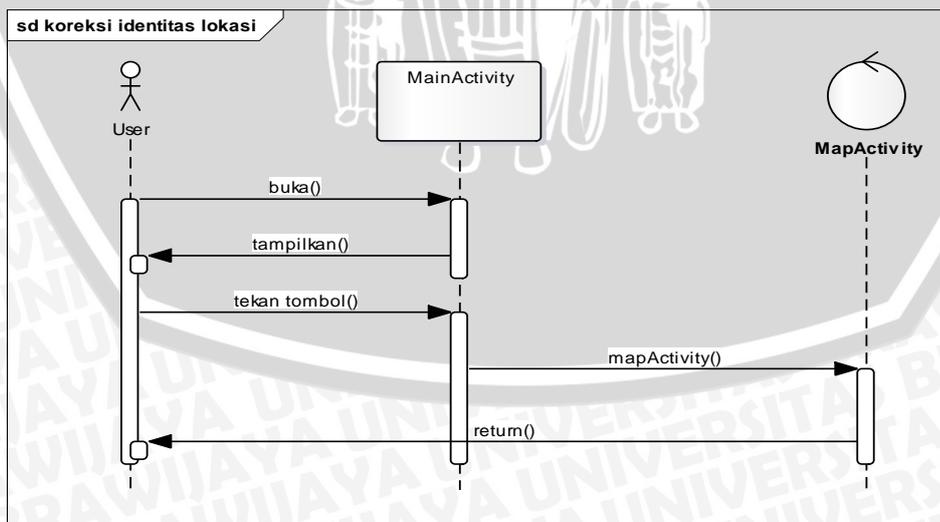
4.2.3.2 Diagram sekuensial memilih kategori dan menambah komentar



Gambar 4.11 Diagram sekuensial memilih kategori

Diagram 4.11 adalah sequence diagram yang digunakan *user* saat memilih kategori bencana dan menambahkan komentar. Setelah membuka menu utama pada halaman MainActivity maka user akan diberikan pilihan untuk mengisikan komentar dan memilih kategori bencana alam apakah yang akan dilaporkan.

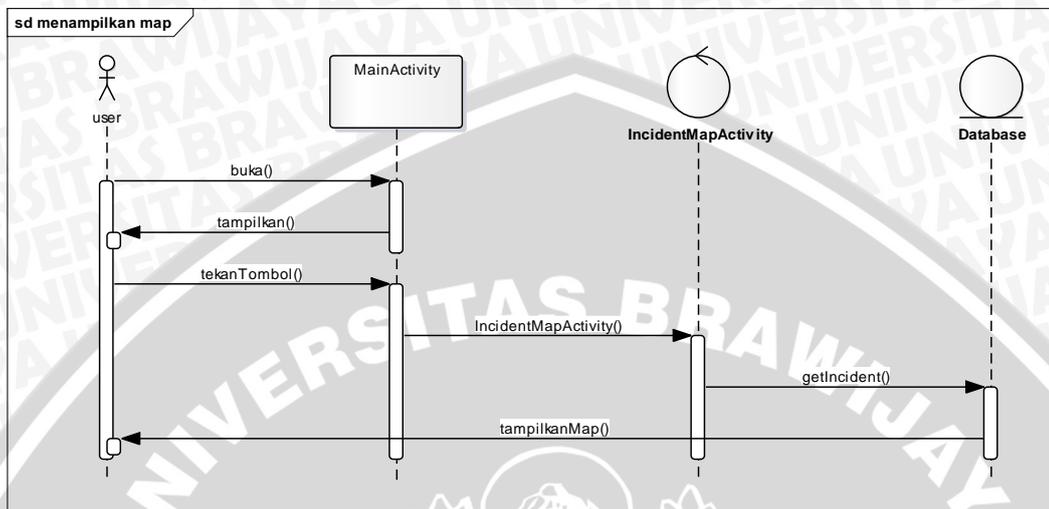
4.2.3.3 Diagram sekuensial koreksi identitas lokasi



Gambar 4.12 Menampilkan map

Diagram 4.12 adalah *sequence diagram* yang digunakan user untuk melakukan koreksi identitas lokasi. Setelah masuk ke menu utama user akan diberikan tombol *submit* dibagian bawah layar.

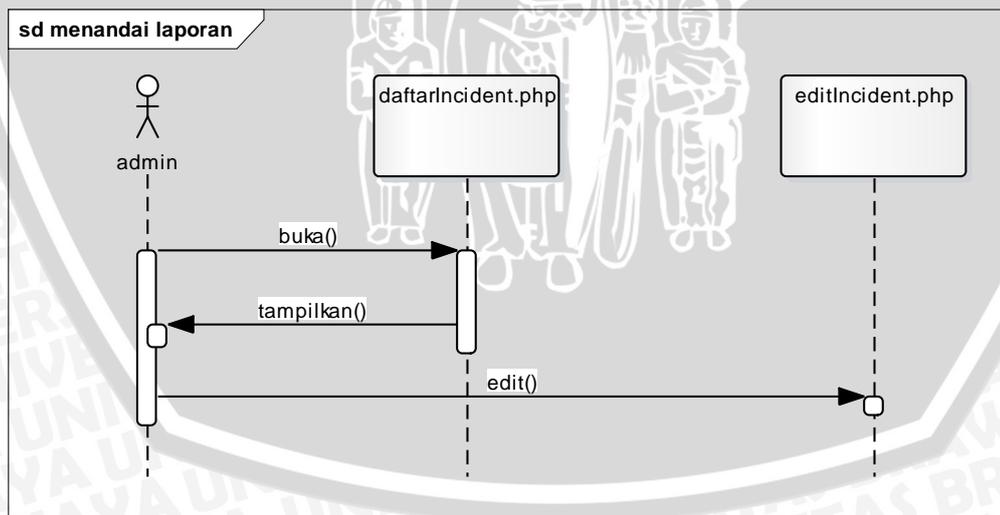
4.2.3.4 Diagram sekuensial menampilkan map admin



Gambar 4.13 menampilkan map admin

Diagram 4.13 adalah *sequence diagram* yang digunakan ketika admin akan melihat map. Pertama admin harus masuk ke halaman utama, kemudian *index.php* akan menghubungi database untuk memintak data letak sebaran laporn bencana alam yang akan ditampilkan di map.

4.2.3.5 Diagram sekuensial menandai laporan

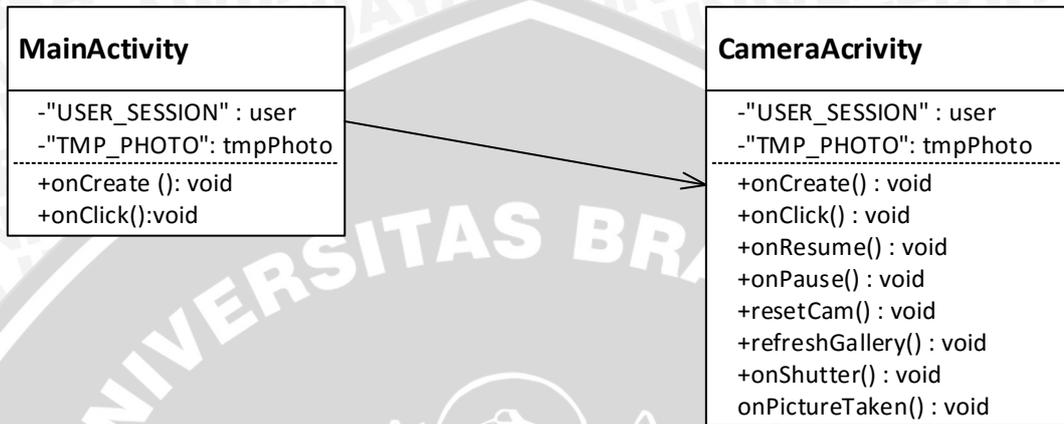


Gambar 4.14 Menandai laporan

Diagram 4.14 adalah *sequence diagram* yang digunakan admin untuk menandai laporan. Pertama admin membuka halaman *daftarincident.php*, kemudian masuk ke menu edit disana admin akan diberikan pilihan untuk menandai yang mana laporan yang sudah ditangani atau belum.

4.2.4 Perancangan Class Diagram

Class diagram merupakan diagram yang memodelkan elemen-elemen class yang membentuk sebuah sistem. *Class diagram* juga menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berhubungan. Gambar berikut ini merupakan *class diagram* dari aplikasi.



Gambar 4.15 Class diagram

Pada gambar 4.15 *class CameraActivity* digunakan untuk melakukan proses pengambilan foto kemudian menyimpan data foto, *longitude* dan *latitudenya* yang nanti akan dikirimkan kembali ke *class MainActivity*

IncidentMapActivity

```
-selectedMarkerId : string
--"USER_SESSION" : user
+onCreate() : void
+onOptionsItemSelected () : void
+onMapReady () : void
+onMarkerClick () : void
```

Gambar 4.16 Class diagram IncidentMapActivity

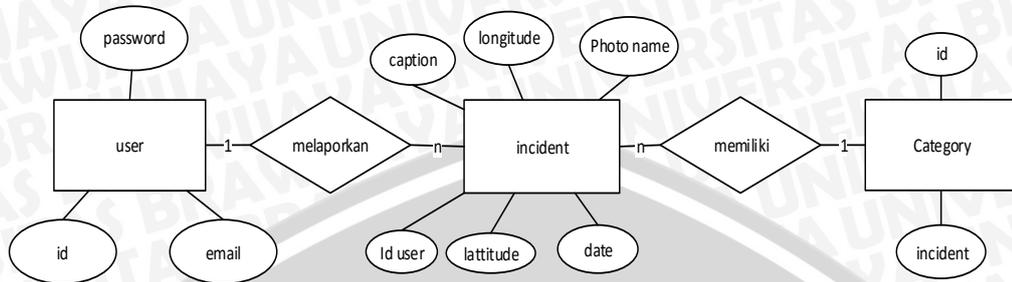
Pada gambar 4.16 adalah *class IncidentMapActivity* yang berguna untuk menampilkan sebaran bencana alam yang sudah dilaporkan oleh user.

4.2.5 Perancangan Basis Data

Perancangan basis data digunakan untuk merancang basis data yang akan dibuat agar masukan dan keluaran program sesuai dengan hasil analisis kebutuhan. Perancangan basis data menggunakan hasil dari proses analisis kebutuhan sebagai acuan dalam merancang basis data

Pada Gambar 4.17, terdapat tiga tabel yaitu, yang pertama adalah tabel tabel insiden yang berguna untuk menyimpan data laporan bencana alam. Tabel ini pun digunakan untuk mengelola laporan bencana alam yang ada di *server*.

Yang kedua adalah tabel tabel user yang berguna untuk *login* user. Dan yang ketiga adalah tabel *category*, tabel ini berfungsi untuk menyimpan jenis kategori yang akan dilaporkan nantinya.

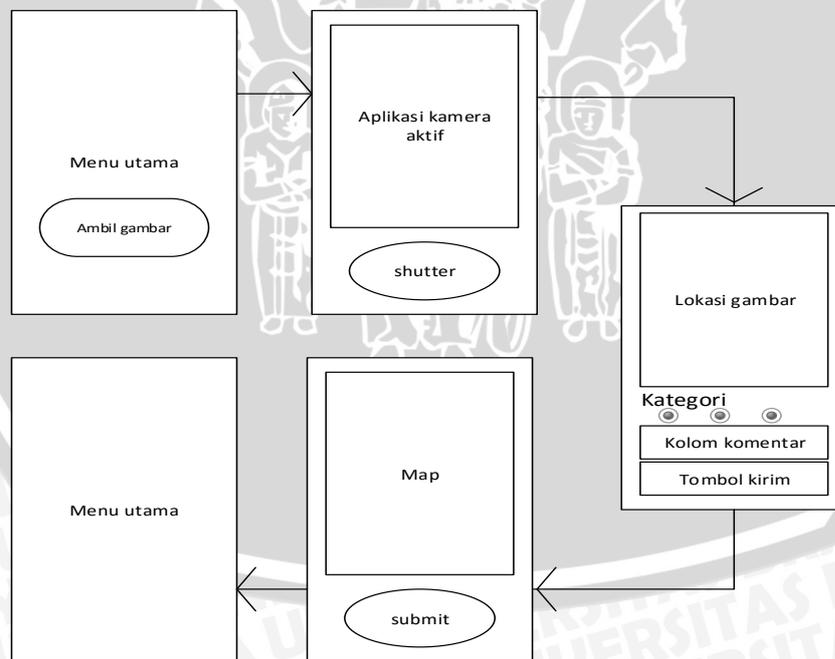


Gambar 4.17 Entity diagram sistem

4.2.6 Perancangan Page Flow

Pada perancangan *page flow* dijelaskan alur yang terdapat pada antarmuka aplikasi. Dalam perancangan *page flow* harus mengacu hasil proses analisis kebutuhan.

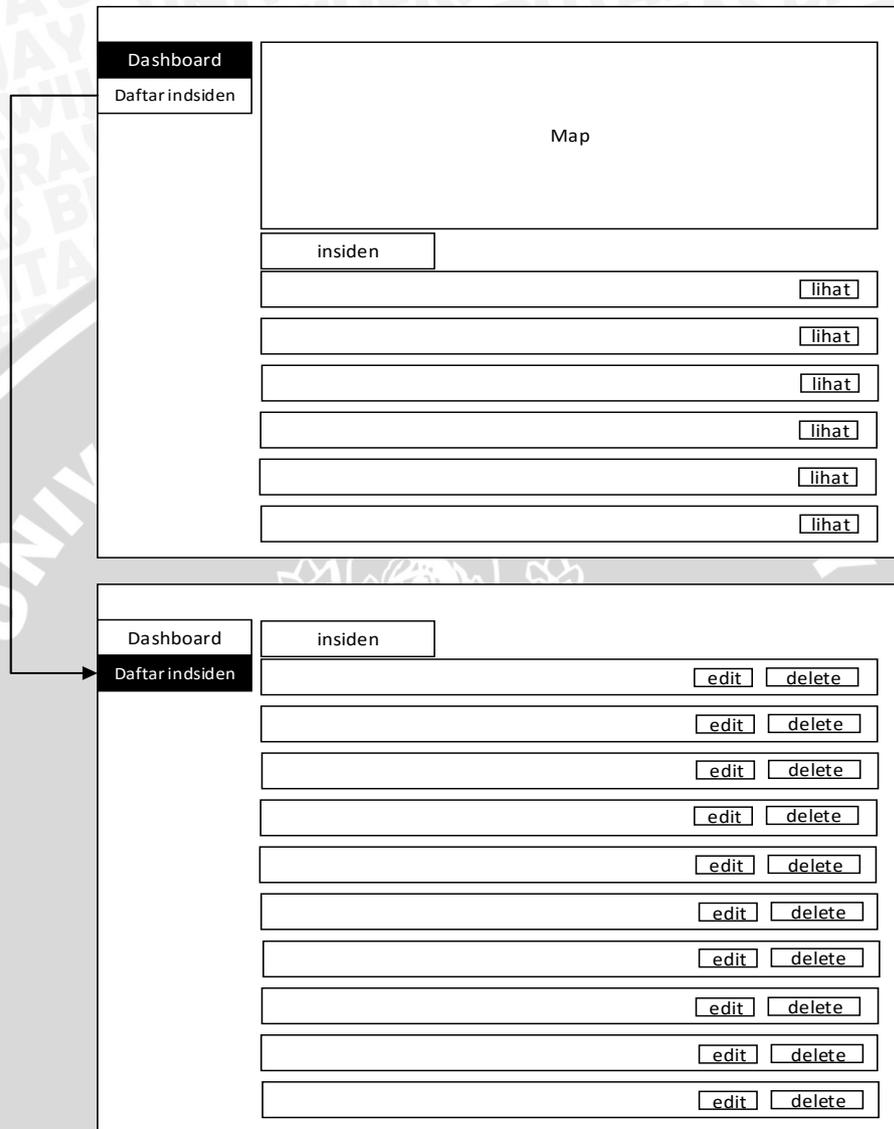
Pada Gambar 4.18, merupakan *page flow* diagram yang diawali dengan menu utama, yaitu menu dengan tombol ambil gambar. Setelah anda menekan tombol ambil gambar kamera akan melakukan proses menangkap gambar. Setelah menangkap gambar anda akan ditampilkan tampilan map untuk mengoreksi identitas lokasi dan kolom komentar yang berguna untuk menambahkan komentar atau info.



Gambar 4.18 Page flow diagram mobile

Pada Gambar 4.19, merupakan *page flow* diagram dari sisi server yang diawali dengan menu utama atau halaman *dashboard*, yaitu menu dengan tampilan map di bagian atas dan juga ada daftar tombol insiden yang telah

dilaporkan dibawahnya. Disini kita disediakan tombol daftar insiden untuk masuk ke halaman daftar insiden, di halaman ini nantinya kita dapat melakukan *edit* pada insiden yang ingin ditandai apakah sudah di tangani atau belum.



Gambar 4.19 Page flow diagram server

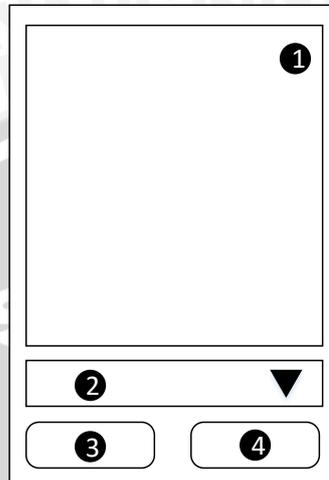
4.2.7 Perancangan Antarmuka

Kemudian terdapat beberapa gambar rancangan antarmuka untuk aplikasi pelaporan bencana alam ini yang dibagi menjadi dua yaitu bagian *mobile* dan *server*. bisa dilihat mulai dari Gambar 4.20 sampai dengan Gambar 4.24.

4.2.7.1 Halaman Utama Mobile

Halaman utama adalah halaman yang pertama kali muncul ketika *user* masuk ke dalam aplikasi. Pada halaman utama ini terdapat tombol shutter yang berguna untuk mengambil gambar. Disini nantinya akan teredia tombol *drop*

down untuk memilih kategori bencana alam yang akan di laporkan. Dibagian bawah akan terdapat dua tombol. Yaitu tombol *capture* dan tombol *submit*. Tombol *capture* berguna untuk melakukan proses pengambilan gambar sedangkan tombol *submit* berguna untuk memproses laporan ke tahap yang selanjutnya.



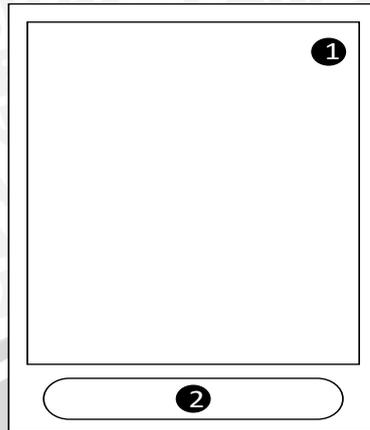
Gambar 4.20 Halaman Utama

Pada Gambar 4.20 rancangan antarmuka halaman utama, tiap-tiap bagian pada antarmuka memiliki keterangan sebagai berikut:

1. Layar untuk menampilkan hasil pengambilan gambar
2. Tombol *drop down* pemilihan kategori bencana alam
3. Tombol *capture*
4. Tombol *submit*

4.2.7.2 Halaman Ambil Gambar Mobile

Halaman ambil gambar ini adalah halaman yang akan muncul etelah *user* menekan tombol *capture* yang ada di menu utama. Pada halaman ini hanya terdapat satu tombol yaitu tombol *shutter* yang berguna untuk mengambil gambar dan juga sebuah *view finder* yang berguna untuk menampilkan gambar yang akan kita ambil.



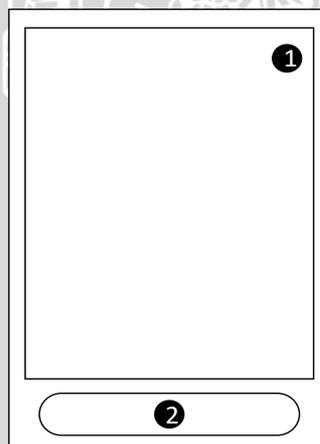
Gambar 4.21 Halaman ambil gambar

Pada Gambar 4.21 rancangan antarmuka halaman ambil gambar, tiap-tiap bagian pada antarmuka memiliki keterangan sebagai berikut:

1. Layar *view finder* untuk menampilkan gambar yang akan diambil
2. Tombol *shutter*

4.2.7.3 Halaman Menampilkan Map Mobile

Halaman menampilkan map ini adalah halaman yang akan muncul setelah user menekan tombol *submit* yang ada pada halaman utama. Pada halaman ini terdapat layar yang akan menampilkan *map* dan lokasi tempat pengambilan gambar yang baru saja kita lakukan. Pada halaman ini juga terdapat satu tombol *submit*, yaitu tombol yang akan digunakan untuk meng *upload* laporan ke *server*.



Gambar 4.22 Halaman menampilkan map

Pada Gambar 4.22 rancangan antarmuka halaman menampilkan map, tiap-tiap bagian pada antarmuka memiliki keterangan sebagai berikut:

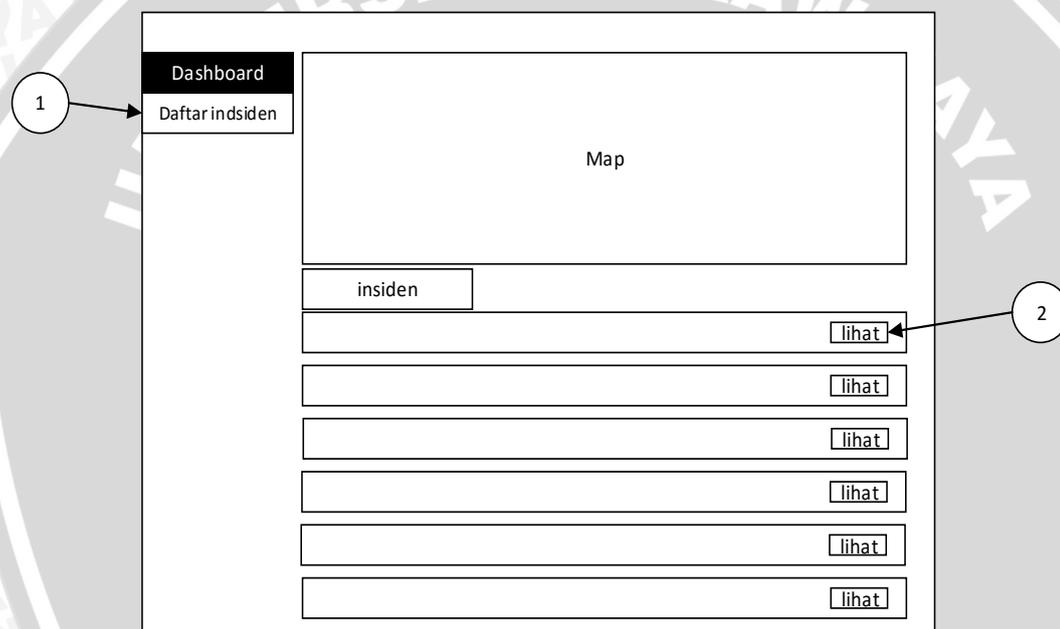
1. Layar untuk menampilkan gambar *map* dan lokasi letak pengambilan gambar
2. Tombol *submit*

4.2.7.4 Halaman Utama Server

Halaman utaman ini adalah halaman yang pertama kali muncul setelah admin masuk ke *server*. Pada halaman ini terdapat sebuah layar *map* yang menampilkan sebaran bencana alam yang sudah dilaporkan oleh *user*. Dibagian bawah dari layar *map* terdapat daftar insiden dengan urutan yang paling baru dilaporkan. Dalam *list* insiden ini terdapat tombol lihat yang berguna untuk menuju lokasi bencana alam secara spesifik. Dan dibagian kiri layar terdapat tombol daftar insiden untuk masuk kedalam halaman daftar insiden.

Pada Gambar 4.23 rancangan antarmuka halaman utama, tiap-tiap bagian pada antarmuka memiliki keterangan sebagai berikut:

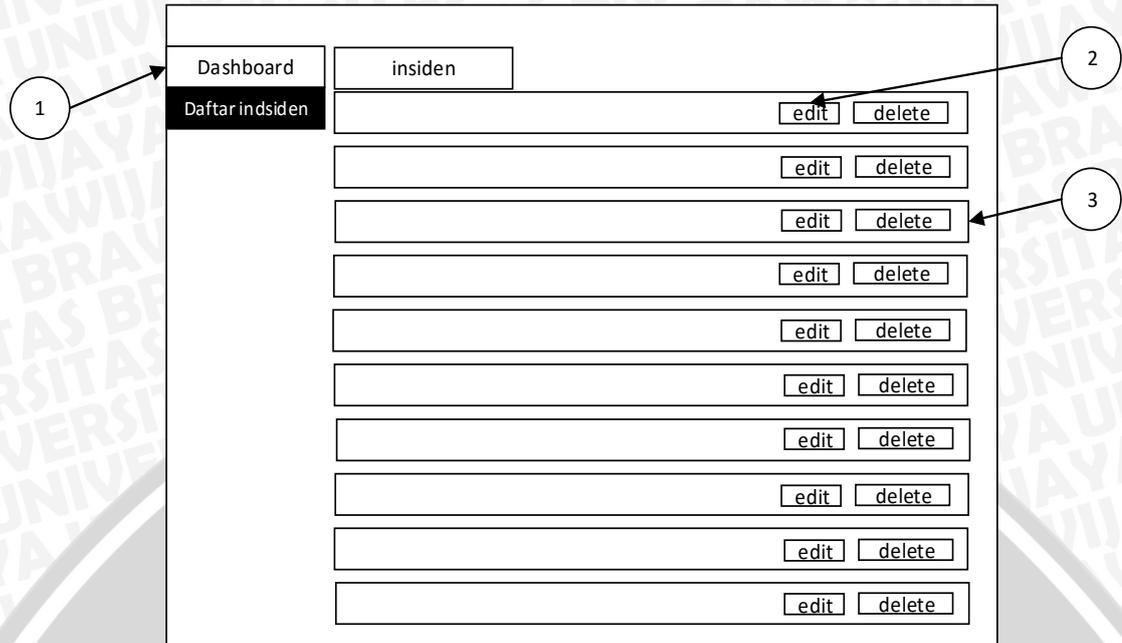
1. Tombol daftar insiden
2. Tombol lihat laporan



Gambar 4.23 Halaman utama server

4.2.7.5 Halaman Daftar Insiden

Halaman daftar insiden ini adalah halaman yang akan muncul setelah admin menekan tombol daftar insiden yang ada pada halaman utama. Pada halaman ini terdapat tomol *dashboard* untuk kembali ke halaman utama. Pada daftar insiden tiap insiden akan mempunyai tombol *edit* dan *delete*. Tombol *edit* ini nantinya akan berfungsi untuk mengedit laporan, seperti menandai apakah laporan sudah ditangani atau belum. Sedangkan tombol *delete* ini berfungsi untuk menghapus laporan.



Gambar 4.24 Halaman daftar insiden

Pada Gambar 4.24 rancangan antarmuka halaman daftar insiden, tiap-tiap bagian pada antarmuka memiliki keterangan sebagai berikut:

1. Tombol *dashboard*
2. Tombol *edit*
3. Tombol *delete*

4.2.8 Perancangan *Drag and Drop* Lokasi Bencana Alam

Perancangan *drag and drop* lokasi bencana alam adalah perancangan bagaimana mengimplementasikan fitur *draggable position* yang telah tersedia pada *googlemaps. GPS* pada *smartphone* tidak jarang memberikan informasi lokasi yang salah ataupun tidak akurat. Hal ini tentu saja akan memiliki dampak yang sangat besar terutama pada aplikasi pelaporan bencana alam yang mengandalkan *GPS* sebagai sumber informasi lokasi pelaporan bencana alam.

Oleh karena itu fitur *drag and drop* ini dibutuhkan untuk aplikasi pelaporan bencana alam. Dengan fitur ini marker yang tampak pada *google maps* setelah kita melakukan pengambilan gambar nantinya dapat digeser ke tempat yang kita inginkan. Berikut ini adalah beberapa fungsi yang digunakan untuk melakukan *drag and drop* pada *google maps*.

1. Public abstract void onMarkerDragStart (Marker marker)

Dipanggil ketika jari pengguna mulai menyentuh *marker* yang ada pada map, posisi ini mungkin berbeda dengan posisi sebelum dimulainya penggeseran marker karena marker nantinya muncul di atas titik sentuh.

2. Public abstract void onMarkerDrag (Marker marker)

Dipanggil berulang kali selama marker digeser.

3. Public abstract void onMarkerDragEnd (Marker marker)

Dipanggil ketika marker selesai digeser, lebih tepatnya ketika pengguna melepaskan jari yang menempel pada *marker* yang terletak pada di *map*.



BAB 5 IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari aplikasi berdasarkan analisis kebutuhan dan proses perancangan yang telah dilakukan. Pembahasan pada bab ini terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, dan implementasi kode program.

5.1 Spesifikasi Sistem

Dalam pembuatan aplikasi geotagging lokasi bencana alam dan musibah ini membutuhkan perangkat keras dan perangkat lunak. Berikut ini akan dijelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pengembangan perangkat ini terdapat pada tabel 5.1

Tabel 5. 1 Spesifikasi perangkat keras komputer

Nama Komponen	Spesifikasi
<i>System Model</i>	HP Pavilion 14
<i>Processor</i>	2,2 GHz Core i5-5200U
<i>Memory</i>	4GB
<i>Display</i>	NVIDIA GT840M

Dalam proses implementasi dan pengujian, perangkat keras yang digunakan adalah smartphone android dengan spesifikasi perangkat keras pada Tabel 5.2 berikut.

Tabel 5. 2 Spesifikasi perangkat keras *android*

Nama Komponen	Spesifikasi
<i>System Model</i>	Onepus One
<i>Processor</i>	Quad-core 2.5 GHz Krait 400
<i>Memory</i>	3GB
<i>Camera Feature</i>	<i>Geo-tagging</i>

5.1.2 Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan aplikasi ini ditampilkan pada tabel 5.3 .

Tabel 5. 3 Spesifikasi perangkat lunak komputer

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 8.1
Bahasa Pemrograman	Java dan PHP
IDE	Android Studio dan Netbeans

Perangkat lunak android yang digunakan memiliki spesifikasi pada Tabel 5.4 berikut.

Tabel 5. 4 Spesifikasi perangkat lunak *mobile*

Nama Komponen	Spesifikasi
OS mobile	Android OS 5.1.1 (Lollipop)

5.2 Implementasi Kode Program

Pada aplikasi geotagging lokasi bencana dan musibah ini memiliki beberapa fungsi utama yang ada di beberapa kelas. Pada penulisan laporan skripsi ini hanya dicantumkan algoritma dari beberapa proses saja.

5.2.1 Algoritma Pengambilan Gambar

```

1  public class CameraActivity extends Activity {
2      private static final String TAG = "CameraActivity";
3      Preview preview;
4      Button buttonClick;
5      Camera camera;
6      private User user;
7      Activity act;
8      Context ctx;
9      String dateTakenTmpPhoto;
10     String tmpPhotoName;
11     private Intent intent;
12     private GPSTracker gps;
13     private LatLng taggedCoordinate;
14
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17
18         super.onCreate(savedInstanceState);
19         ctx = this;
20         act = this;
21         requestWindowFeature(Window.FEATURE_NO_TITLE);
22         getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
23
24         setContentView(R.layout.activity_camera);
25
26         intent = getIntent();
27         if (intent != null) {
28             if (intent.getSerializableExtra("USER_SESSION") != null)
29     
```

```

30         user = (User)
31         intent.getSerializableExtra("USER_SESSION");
32         Log.e("_USER_ID_", String.valueOf(user.getId()));
33         Log.e("_USER_EMAIL_", user.getEmail());
34         Log.e("_USER_PASSWORD_", user.getPassword());
35     }
36 }
37
38     gps = new GPSTracker(CameraActivity.this);
39     if (gps.canGetLocation()) {
40
41         double latitude = gps.getLatitude();
42         double longitude = gps.getLongitude();
43
44
45         taggedCoordinate = new LatLng(latitude, longitude);
46         Log.d("_onLocationChanged_", "Lat: " +
47 String.valueOf(String.format("%.8f", latitude)) + " Long: " +
48 String.valueOf(String.format("%.8f", longitude)));
49     } else {
50         gps.showSettingsAlert();
51     }
52
53
54     preview = new Preview(this, (SurfaceView)
55 findViewById(R.id.surfaceView));
56     preview.setLayoutParams(new
57 LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
58     ((FrameLayout) findViewById(R.id.layout)).addView(preview);
59     preview.setKeepScreenOn(true);
60
61     preview.setOnClickListener(new OnClickListener() {
62
63         @Override
64         public void onClick(View arg0) {
65             camera.takePicture(shutterCallback, rawCallback,
66 jpegCallback);
67         }
68     });

```

Gambar 5. 1 Algoritma pengambilan gambar

Pada Gambar 5.1, ditampilkan implementasi kode program algoritma pengambilan gambar. Kode program pengambilan gambar ini di *trigger* melalui tombol yang ada pada kode program *main activity* yang terdapat pada Lampiran 1. Berikut beberapa penjelasan mengenai kode program pada Gambar 5.1:

1. Baris 26-36 merupakan instansiasi object intent yang berfungsi untuk menangkap variable dari Activity pemanggil CameraActivity.
2. Baris 38-51 merupakan kode dari pembuatan class object untuk GPS.
3. Baris 53-68 merupakan instansiasi object Preview yang digunakan untuk menampilkan Camera preview pada view.

5.2.2 Algoritma *Upoad* Laporan

```

1  public void uploadImage() {
2
3      // When Image is selected from Gallery
4
5      if (imgPath != null && !imgPath.isEmpty()) {
6          prgDialog.setMessage("Converting Image to Binary Data");
7          prgDialog.show();
8          // Convert image to String using Base64
9          encodeImagetoString();
10         // When Image is not selected from Gallery
11     } else {
12         Helper.toast(
13             getApplicationContext(),
14             "You must select image from gallery before you
15 try to upload");
16     }
17 }
18
19
20
21 // AsyncTask - Mengkonversi image/foto ke string base64_encode
22 untuk dikirim ke server dalam bentuk string
23 public void encodeImagetoString() {
24     new AsyncTask<Void, Void, String>() {
25
26         protected void onPreExecute() {
27
28         };
29
30
31
32         @Override
33         protected String doInBackground(Void... params) {
34             BitmapFactory.Options options = null;
35             options = new BitmapFactory.Options();
36             options.inSampleSize = 2;
37             bitmap = BitmapFactory.decodeFile(imgPath, options);
38
39             ByteArrayOutputStream stream = new
40 ByteArrayOutputStream();
41             // Must compress the Image to reduce image size to
42 make upload easy
43             bitmap.compress(Bitmap.CompressFormat.JPEG, 60,
44 stream);
45             byte[] byte_arr = stream.toByteArray();
46             // Encode Image to String
47             encodedString = Base64.encodeToString(byte_arr, 0);
48
49             Log.d("BASE64_", encodedString);
50             return "";
51         }
52     }
53
54
55     @Override
56     protected void onPostExecute(String msg) {
57         prgDialog.setMessage("Calling Upload");
58         // Put converted Image string into Async Http Post
59 param
60
61         Log.d(" CAPTION ", tmpPhoto.getPhotoCaption());

```

```

62         Log.d("CAPTIONN OK BROOOS2",
63         String.valueOf(tmpPhoto.getUserId()));
64         params.put("image", encodedString);
65         params.put("user_id",
66         String.valueOf(tmpPhoto.getUserId()));
67         params.put("caption", tmpPhoto.getPhotoCaption());
68         params.put("latitude", tmpPhoto.getPhotoLatitude());
69         params.put("longitude",
70         tmpPhoto.getPhotoLongitude());
71         params.put("date_taken",
72         tmpPhoto.getDateTakenPhoto());
73         params.put("category", tmpPhoto.getCategory());
74         // Trigger Image upload
75         triggerImageUpload();
76     }
77     }.execute(null, null, null);
78 }
79
80
81
82 public void triggerImageUpload() {
83     makeHTTPCall();
84 }
85
86
87
88 // Memanggil php untuk mengupload foto ke server beserta data
89 incident
90 public void makeHTTPCall() {
91     prgDialog.setMessage("Uploading photo...");
92     AsyncHttpClient client = new AsyncHttpClient();
93
94     client.post(FILE_UPLOAD_URL_PROD, params, new
95     AsyncHttpResponseHandler() {
96         // When the response returned by REST has Http
97         // response code '200'
98

```

Gambar 5.2 Algoritma *upload* laporan

Pada Gambar 5.2, ditampilkan implementasi kode program algoritma menampilkan map. Berikut beberapa penjelasan mengenai kode program pada Gambar 5.2:

1. Baris 1-16 merupakan kode untuk memastikan ketersediaan image
2. Baris 17-46 merupakan kode untuk melakukan konversi *image* ke dalam bentuk base64 string
3. Baris 48-86 melakukan pengiriman data ke *server*

5.2.3 Algoritma Edit Lokasi Marker

```
1 public void onMarkerDragStart(Marker marker) {
2     Log.d("onMarkerDragStart",
3     marker.getPosition().toString());
4     }
5
6     @Override
7     public void onMarkerDrag(Marker marker) {
8         Log.d("onMarkerDrag",
9         marker.getPosition().toString());
10        }
11
12        @Override
13        public void onMarkerDragEnd(Marker marker) {
14            Log.d("onMarkerDragEnd",
15            marker.getPosition().toString());
16        }
17
18        tmpPhoto.setPhotoLatitude(String.valueOf(String.format("%.8
19        f", marker.getPosition().latitude)));
20
21        tmpPhoto.setPhotoLongitude(String.valueOf(String.format("%.8
22        f", marker.getPosition().longitude)));
23    }
```

Gambar 5.3 Algoritma Edit Lokasi

Pada Gambar 5.3, ditampilkan implementasi kode program algoritma edit lokasi. Berikut beberapa penjelasan mengenai kode program pada Gambar 5.3:

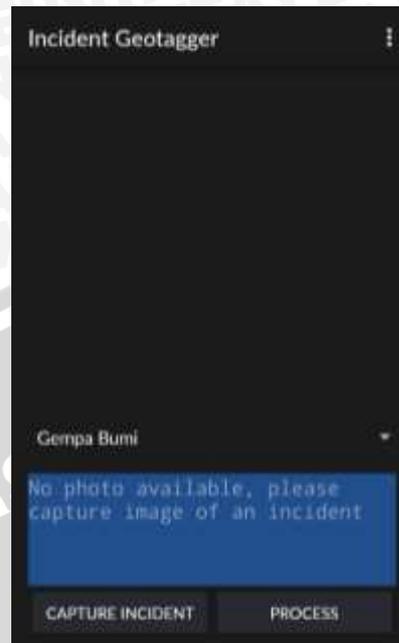
1. Baris 1-5 merupakan kode untuk memulai proses *drag* pada *marker* yang terdapat pada *map*
2. Baris 8-10 merupakan kode untuk melakukan pergeseran *marker*
3. Baris 14-17 merupakan kode untuk mengakhiri proses *drag* pada *marker*.
4. Baris 19-23 merupakan kode untuk mengambil informasi lokasi berupa *latitude* dan *longitude*.

5.3 Implementasi Antarmuka Pengguna

Pada tahap ini akan ditampilkan hasil implementasi antarmuka pengguna aplikasi geotagging lokasi bencana alam dan musibah. Tampilan dari implementasi antarmuka pengguna aplikasi ini ditunjukkan pada Gambar 5.4 sampai dengan 5.6 berikut.

5.3.1 Halaman Utama *Mobile*

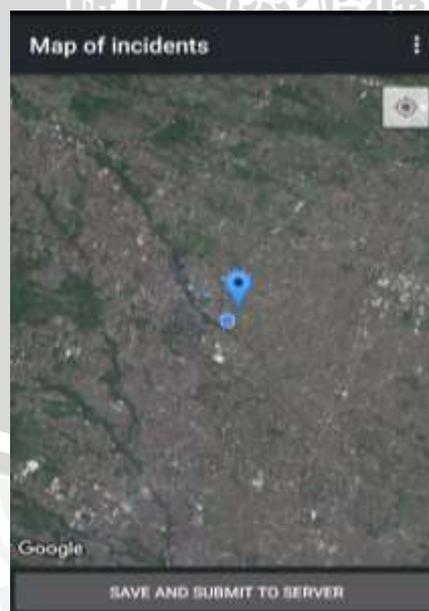
Gambar 5.4 ini adalah tampilah utama aplikasi ini. Halaman ini adalah halaman yang pertama kali keluar setelah *user* masuk ke dalam aplikasi. Pada halaman utama ini tersedia tombol *capture incident*. Tombol ini berguna untuk menampilkan halaman pengambilan gambar. Tombol kategori yang berfungsi untuk memilih kategori bencana alam yang akan dilaporkan, dan tombol *process* untuk masuk ke proses selanjutnya.



Gambar 5. 4 Halaman utama aplikasi

5.3.2 Halaman *Map Activity*

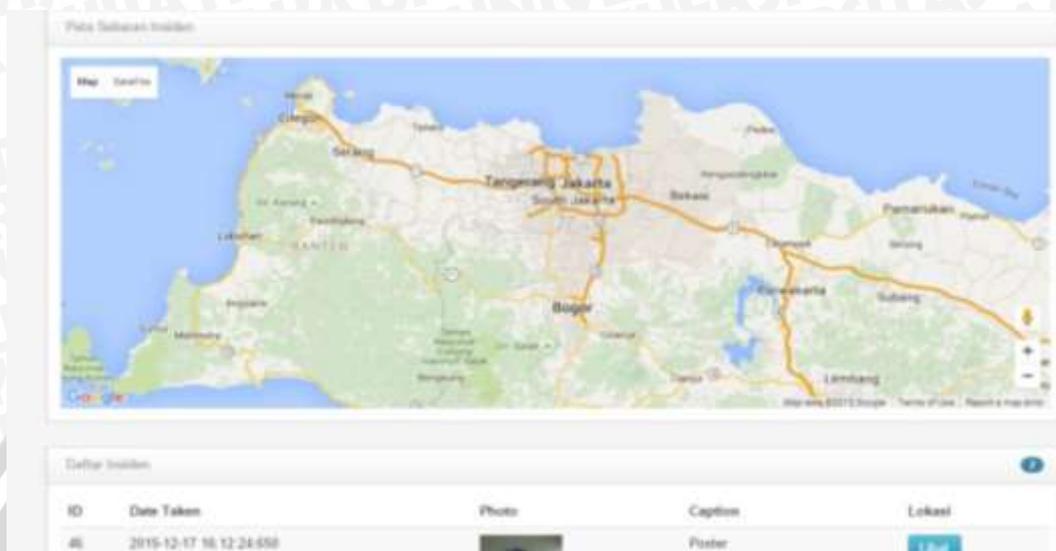
Gambar 5.5 ini adalah tampilan dari *map activity*, yaitu halaman yang akan muncul setelah kita menekan tombol *process* pada *main menu*. Halaman ini berguna untuk mengoreksi letak identitas lokasi yang kita laporkan dengan cara melakukan *drag and drop* pada *pin* yang terletak pada *map*.



Gambar 5. 5 Halaman *map activity*

5.3.3 Halaman Utama *Web Server*

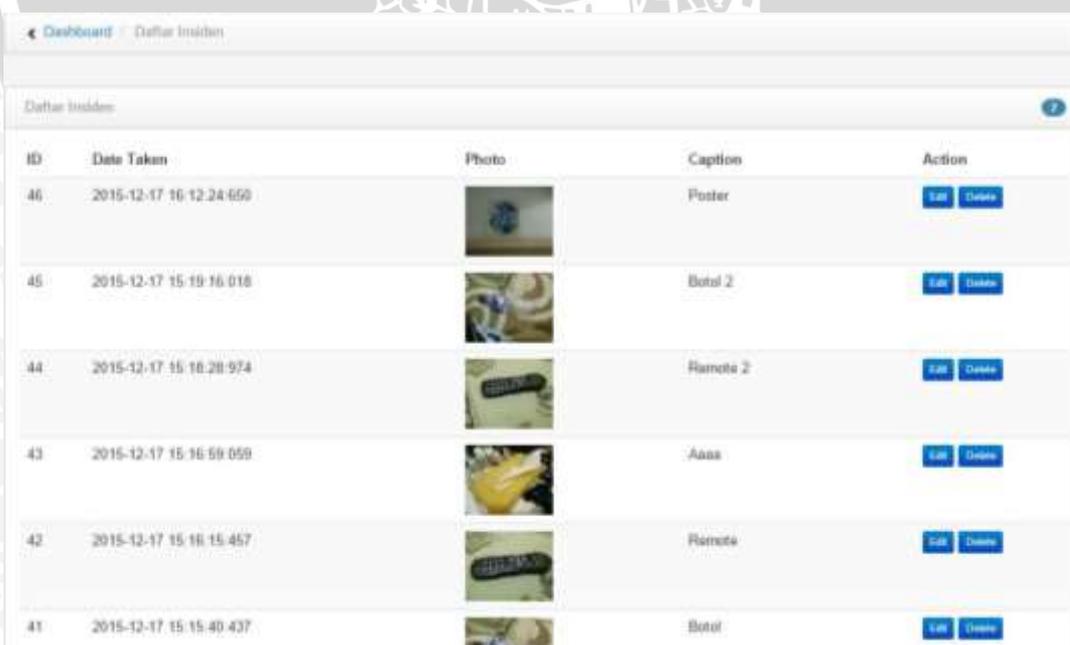
Pada Gambar 5.6 ini adalah tampilan halaman utama dari bagian *web server*. Pada halaman utama ini akan ditampilkan map dan juga sebaran laporan bencana alam yang sudah dikirimkan oleh *user*.



Gambar 5. 6 Halaman Utama *Web Server*

5.3.4 Halaman Daftar Insiden

Gambar 5.7 ini adalah tampilan dari halaman daftar insiden dari bagian *server*. Pada halaman ini akan ditampilkan daftar insiden yang telah masuk ke *server*, disini juga tersedia fitur edit untuk keperluan *admin* mengolah data.



Gambar 5. 7 Halaman Daftar Insiden

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan pengujian dan analisis pada aplikasi *geotagging* lokasi bencana alam dan musibah. Proses pengujian dilakukan melalui 2 tahap, yaitu pengujian validasi dan *usability*. Pengujian validasi dilakukan dengan menggunakan teknik pengujian *black-box* (*black-box testing*). Sedangkan pengujian *usability* dilakukan dengan metode kuisiner. Selanjutnya dilakukan analisis terhadap sistem untuk mengetahui hasil dari pengujian.

6.1 Pengujian

Proses pengujian pada aplikasi menggunakan pengujian validasi dan pengujian *usability*. Pengujian validasi dilakukan untuk menguji sisi fungsional sistem sedangkan pengujian *usability* digunakan untuk menguji sisi non-fungsional sistem.

6.1.1 Pengujian Validasi

Pengujian validasi pada penelitian ini digunakan untuk membuktikan kesesuaian antara daftar-daftar kebutuhan yang merupakan hasil dari analisis kebutuhan dan kinerja sistem, setiap kebutuhan dilakukan proses pengujian masing-masing akan ditunjukkan pada tabel 6.1 sampai 6.8.

Tabel 6.1 Kasus uji validasi mengambil gambar

Nomor Kasus Uji	VAL_01
Nama Kasus Uji	Mengambil gambar
Objek Uji	Kebutuhan Fungsional(F01)
Tujuan Pengujian	Memastikan aplikasi mampu mengmil gambar.
Prosedur Pengujian	Menekan tombol <i>capture</i> untuk mengambil gambar.
Hasil yang Diharapkan	Gambar yang diinginkan berhasil diambil.

Tabel 6.2 Kasus uji validasi memilih kategori dan memberikan komentar

Nomor Kasus Uji	VAL_02
Nama Kasus Uji	Memilih kategori dan Memberikan komentar
Objek Uji	Kebutuhan Fungsional(F03)
Tujuan Pengujian	Memastikan aplikasi mampu menyediakan fitur untuk <i>user memilih kategori dan</i> memberikan komentar.
Prosedur Pengujian	Memilih kategori dan mengisikan kata pada kolom komentar.
Hasil yang Diharapkan	Kategori dan komentar berhasil ditambahkan.

Tabel 6.3 Kasus uji validasi menampilkan map *mobile*

Nomor Kasus Uji	VAL_03
Nama Kasus Uji	Menampilkan map <i>mobile</i>
Objek Uji	Kebutuhan Fungsional(F05)
Tujuan Pengujian	Memastikan aplikasi mampu menampilkan map pada perangkat <i>mobile</i> .
Prosedur Pengujian	<i>User</i> memilih tampilkan map pada menu.
Hasil yang Diharapkan	<i>Map</i> berhasil ditampilkan oleh <i>user</i> .

Tabel 6.4 Kasus uji validasi menampilkan map *server*

Nomor Kasus Uji	VAL_04
Nama Kasus Uji	Memilih kategori <i>server</i>
Objek Uji	Kebutuhan Fungsional(F06)
Tujuan Pengujian	Memastikan aplikasi <i>server</i> mampu menampilkan map.
Prosedur Pengujian	<i>Admin</i> membuka halaman utama.
Hasil yang Diharapkan	<i>Map</i> berhasil ditampilkan oleh <i>admin</i> .

Tabel 6.5 Kasus uji validasi menandai laporan

Nomor Kasus Uji	VAL_05
Nama Kasus Uji	Menandai laporan
Objek Uji	Kebutuhan Fungsional(F07)
Tujuan Pengujian	Memastikan aplikasi mampu menandai laporan yang sudah ditangani.
Prosedur Pengujian	Memilih salah satu lokasi yang ingin di <i>esit</i> kemudian ditandai.
Hasil yang Diharapkan	<i>Pin</i> lokasi berhasil ditandai.

Berdasarkan pada kasus uji yang dilakukan akan diperoleh hasil pengujian. Hasil pengujian validasi akan ditampilkan pada Tabel 6.8 berikut.

Tabel 6.8 Hasil pengujian validasi

Nomor Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validasi
VAL_01	Gambar yang diinginkan berhasil diambil.	Gambar yang diinginkan berhasil diambil.	Valid
VAL_02	Terjadi perubahan lokasi dari gambar.	Terjadi perubahan lokasi dari gambar.	Valid
VAL_03	Komentar berhasil	Komentar berhasil	Valid

	ditambahkan.	ditambahkan.	
VAL_04	Kategori berhasil ditambahkan	Kategori berhasil ditambahkan	Valid
VAL_05	Map berhasil ditampilkan oleh user.	Map berhasil ditampilkan oleh user.	Valid
VAL_06	Map berhasil ditampilkan oleh admin.	Map berhasil ditampilkan oleh admin.	Valid
VAL_07	Pin lokasi berhasil ditandai	Pin lokasi berhasil ditandai	Valid

6.1.2 Pengujian Usability

Pengujian *usability* pada penelitian ini menggunakan metode kuisiioner dengan cara mengajukan beberapa pertanyaan kepada 13 orang responden yang berasal dari kalangan mahasiswa. Kemudian dari data kuisiioner tersebut diambil kesimpulannya terhadap sistem yang telah dibuat. Kategori jawaban menggunakan lima rentang, diantaranya:

3. Sangat tidak setuju
4. Tidak setuju
5. Biasa saja
6. Setuju
7. Sangat setuju

Pertanyaan yang diberikan kepada responden dibuat berdasarkan metode *usability testing* yaitu *learnability*, *efficiency*, mudah *memorability*, perlindungan terhadap *errors*, dan *satisfaction* (Saputra, 2014).

Pada komponen *learnability*, diberikan pertanyaan dengan tujuan menjelaskan tingkat kemudahan pengguna atau *user* dalam berinteraksi saat pertama kali melihat atau berhadapan dengan sistem yang ada. Kasus uji pada komponen *learnability* dapat dilihat pada Tabel 6.9 berikut.

Tabel 6.9 Kasus uji *learnability*

No.	Pertanyaan
1.	Tulisan teks yang digunakan pada sistem mudah dan jelas
2.	Simbol-simbol gambar mudah pada aplikasi dipahami

Pada komponen *efficiency*, diberikan pertanyaan dengan tujuan menjelaskan seberapa cepat pengguna berinteraksi saat pertama kali mempelajari sistem. Kasus uji pada komponen *efficiency* dapat dilihat pada Tabel 6.10 berikut.

Tabel 6.10 Kasus uji *efficiency*

No.	Pertanyaan
-----	------------

1.	Saat berpindah dari halaman yang satu ke yang lainnya dapat dilakukan dengan cepat(tidak terdapat delay)
----	--

Pada komponen *memorability*, diberikan pertanyaan dengan tujuan menjelaskan tingkat kemudahan pengguna dalam menggunakan sistem dengan baik, setelah beberapa lama tidak menggunakannya. Kasus uji pada komponen *memorability* dapat dilihat pada Tabel 6.11 berikut.

Tabel 6.11 Kasus uji *memorability*

No.	Pertanyaan
1.	Sistem mudah dioperasikan
2.	Halaman pada sistem mudah diingat
3.	Icon yang ada telah lazim dan sudah anda kenal

Pada komponen *errors*, diberikan pertanyaan dengan tujuan menjelaskan kemungkinan terjadinya *error* atau kesalahan yang dilakukan oleh pengguna dan seberapa mudah mereka dapat mengatasinya. Kasus uji pada komponen *errors* dapat dilihat pada Tabel 6.12 berikut.

Tabel 6.12 Kasus uji *errors*

No.	Pertanyaan
1.	Semua tombol dapat berfungsi dengan baik(tidak terdapat <i>error</i>)
2.	Lokasi tempat pengambilan gambar sudah sesuai dengan yang di tampilkan pada <i>map</i>

Pada komponen *satisfaction*, diberikan pertanyaan dengan tujuan menjelaskan tingkat kepuasan pengguna dalam menggunakan sistem yang telah dibuat. Kasus uji pada komponen *satisfaction* dapat dilihat pada Tabel 6.13 berikut.

Tabel 6.13 Kasus uji *satisfaction*

No.	Pertanyaan
1.	Semua fitur yang ada telah mencakup kebutuhan yang diharapkan
2.	Anda merasa terbantu dengan adanya aplikasi ini dalam proses pelaporan bencana alam

Berdasarkan hasil pengujian *usability* melalui kuisioner yang disebarkan kepada 13 orang responder, didapatkan akumulasi hasil pada Tabel 6.14 berikut.

Tabel 6.14 Hasil pengujian *usability*

No.	Pertanyaan	1	2	3	4	5
1.	Tulisan teks yang digunakan pada sistem mudah dan jelas	0	0	0	9	4
2.	Simbol-simbol gambar pada aplikasi mudah dipahami	0	0	2	7	4
3.	Saat berpindah dari halaman yang satu ke yang lainnya dapat dilakukan dengan cepat (tidak terdapat delay)	0	0	1	7	5
4.	Sistem mudah dioperasikan	0	0	3	7	3
5.	Halaman pada sistem mudah diingat	0	0	3	9	1
6.	Icon yang ada telah lazim dan sudah anda kenal	0	0	3	8	2
7.	Semua tombol dapat berfungsi dengan baik (tidak terdapat <i>error</i>)	0	0	0	2	11
8.	Lokasi tempat pengambilan gambar sudah sesuai dengan yang di tampilkan pada <i>map</i>	0	0	4	8	1
9.	Semua fitur yang ada telah mencakup kebutuhan yang diharapkan	0	0	0	11	2
10.	Anda merasa terbantu dengan adanya aplikasi ini dalam proses pelaporan bencana alam	0	0	0	7	6

Keterangan:

- 1 = Sangat Tidak Setuju 3 = Netral 5 = Sangat Setuju
 2 = Tidak Setuju 4 = Setuju

6.2 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian aplikasi *geotagging* lokasi bencana alam dan musibah. Proses analisis yang dilakukan adalah analisis pada pengujian validasi dan pengujian *usability*.

6.2.1 Analisis Hasil Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan cara melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan fungsional. Berdasarkan dengan hasil pengujian validasi pada tabel 6.8 dapat disimpulkan bahwa implementasi dan fungsionalitas aplikasi *geotagging* lokasi bencana alam dan musibah telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan. Analisis kebutuhan yang dimaksud adalah dapat menjalankan semua fitur aplikasi, yaitu mengambil gambar, memilih kategori bencana, menambahkan komentar, dan mengoreksi identitas lokasi.

6.2.2 Analisis Hasil Pengujian *Usability*

Proses analisis terhadap hasil pengujian *usability* dilakukan dengan menggunakan skala likert. Hasil perhitungan index presentase dari setiap pertanyaan ditunjukkan pada Tabel 6.15 dengan menggunakan rumus untuk menghitung total skor yang ditunjukkan pada persamaan 6.1 dan rumus untuk melakukan perhitungan index persentase ditunjukkan pada persamaan 6.2 [12].

Tabel 6.15 Index persentase

No.	Pertanyaan	STS	TS	N	S	SS	Total Skor	Index (%)
<i>Learnability</i>								
1.	Tulisan teks yang digunakan pada sistem mudah dan jelas	0	0	0	9	4	56	86.2%
2.	Simbol-simbol gambar mudah dipahami	0	0	2	7	4	42	64.6%
<i>Efficiency</i>								
3.	Saat berpindah dari halaman yang satu ke yang lainnya dapat dilakukan dengan cepat (tidak terdapat delay)	0	0	1	7	5	56	86.2%
<i>Memorability</i>								
4.	Sistem mudah dioperasikan	0	0	3	7	3	52	80%
5.	Halaman pada sistem mudah diingat	0	0	3	9	1	50	76.9%
6.	Icon yang ada telah lazim dan sudah anda kenal	0	0	3	8	2	51	78.5%
<i>Errors</i>								
7.	Semua tombol dapat berfungsi dengan baik (tidak terdapat error)	0	0	0	2	11	63	96.9%
8.	Lokasi tempat pengambilan gambar sudah sesuai dengan yang di tampilkan pada <i>map</i>	0	0	4	8	1	49	75.4%
<i>Satisfaction</i>								
9.	Semua fitur yang ada telah mencakup kebutuhan yang diharapkan	0	0	0	11	2	54	83%
10.	Anda merasa terbantu dengan adanya aplikasi ini dalam proses pelaporan bencana alam	0	0	0	7	6	58	89.2%

Keterangan:

STS = Sangat Tidak Setuju N = Netral SS = Sangat Setuju
TS = Tidak Setuju S = Setuju

$$\text{Total Skor} = S_{STS} \times 1 + S_{TS} \times 2 + S_N \times 3 + S_S \times 4 + S_{SS} \times 5 \quad (6.1)$$

$$\text{Index}(\%) = (\text{Total Skor} / Y) 100 \quad (6.2)$$

Y = Skor Likert Tertinggi × Jumlah Responder

Untuk dapat menentukan status dari Index persentase dalam berbagai aspek penilaian digunakan interpretasi skor likert pada tabel 6.16 berikut.

Tabel 6.16 Interpretasi skor Likert

Skor likert	Interpretasi Skor dengan interval = 20	Pilihan
1	0%-19.99%	Sangat Tidak Memuaskan
2	20%-39.99%	Tidak Memuaskan
3	40%-59.99%	Biasa
4	60%-79.99%	Memuaskan
5	80%-100%	Sangat Memuaskan

Berdasarkan dengan interpretasi skor likert maka dapat ditentukan status dari hasil pengujian *usability* yang ditunjukkan pada tabel 6.17 berikut.

Tabel 6.17 Status pengujian *usability*

Aspek Penilaian	Rata-rata persentase(%)	Status
<i>Learnability</i>	75.4%	Memuaskan
<i>Efficiency</i>	86.2%	Sangat Memuaskan
<i>Memorability</i>	78.5%	Memuaskan
<i>Errors</i>	86.2%	Sangat Memuaskan
<i>Satisfaction</i>	86.1%	Sangat Memuaskan
Rata-rata	82.5%	Sangat Memuaskan

Dari hasil pengujian *usability* yang telah dilakukan, menyatakan bahwa index persentase (%) yang didapatkan pada setiap aspek penelitian, mendapatkan hasil

rata-rata sebanyak 82.5% yaitu dengan status sangat memuaskan. Dengan adanya hasil tersebut menunjukkan aplikasi layak diterima dan memudahkan pengguna dengan hasil sangat memuaskan.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil analisis perancangan, implementasi dan pengujian yang telah dilakukan, maka diambil kesimpulan sebagai berikut:

1. Perancangan aplikasi *geotagging* lokasi musibah dan bencana alam pada *platform android* pertama-tama dilakukan dengan membuat daftar kebutuhan. Dari daftar kebutuhan tersebut, didapatkan skenario yang digambarkan dari *use case* diagram. Kemudian *use case* diagram dikembangkan menjadi *activity* diagram, *class* diagram, dan *sequential* diagram.
2. Sistem telah diimplementasikan sesuai dengan hasil perancangan pada *smartphone android* menggunakan bahasa pemrograman *java* dengan memanfaatkan fitur *geotagging* untuk mendapatkan identitas lokasi dari gambar yang di *capture*.
3. Berdasarkan hasil pengujian validasi dengan metode *black box testing*, menunjukkan bahwa semua fitur atau kebutuhan fungsional telah sesuai dengan yang dibutuhkan.
4. Tingkat *usability* sistem berdasarkan aspek pengujian *usability* yang meliputi aspek penilaian *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction* menunjukkan rata-rata sebesar 82.5 % yaitu dengan status sangat memuaskan.

7.2 Saran

Saran yang bisa diberikan untuk pengembangan lebih lanjut terhadap aplikasi ini adalah:

1. Dapat dilakukan pengembangan pada perangkat *mobile* berbasis *IOS* untuk lebih mempermudah masyarakat dalam pelaporan bencana alam yang terjadi.

DAFTAR PUSTAKA

- Andi, 2013. *"Step by step Menjadi Programmer Android"*. s.l.:Andi.
- Aryansyah, E., 2016. "Rancang Bangun Aplikasi Mobile Social Geotagging Keragaman Budaya Indonesia Pada Platform Android".
- Badan Nasional Penanggulangan Bencana, 2016. *Badan Nasional Penanggulangan Bencana*. [Online] Available at: bnpb.go.id [Accessed Desember 2015].
- Brooke, J., 1996. "A quick and dirty usability scale".
- Daigneau, R., 2012. "Service Design Pattern: Fundamental Design Solutions for SOAP/WSDL dan RESTful Web Services".
- Firdaus, A., 2013. *"Dokumentasi Google Maps API"*. [Online] Available at: <http://www.developers.google.com/maps/documentation/android/> [Accessed Mei 2013].
- Gosavi, A. & Vishnu, 2014. "Disaster Alert and Notification System Via Android Mobile Phone by Using Google Map". Volume 4.
- Joshi, D., Gallagher, A., Yu, J. & Luo, J., 2010. "Exploring User Image Tags for Geo-Location Inference".
- Jovilyn Therese B, F. C. M. O., 2016. "A Mobile Disaster Management System Using the Android Technology".
- Luckin, R., 2005. "Using Mobile Teknologi to Create Flexible Learning Contexts". *Journal of Interactive Media in Education*.
- Martin, F., 2005. *"UML Ddistilled"*. Yogyakarta: ANDI.
- Munaiseche, C. P. C., 2012. "Pengujian Web Aplikasi DSS Berdasarkan pada Aspek Usability". Volume 8, pp. 63-68.
- Rahadi, D. R., 2014. *Pengukuran Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi Android*. s.l.:Program Pascasarjana Universitas Bina Darma Palembang.
- Santoso, P. & Ferdina, R., 2015. "Pengujian Usability Website Menggunakan System Usability Scale". Volume 17, pp. 31-38.
- Saputra, E., 2014. *Usability Testing Untuk Mengukur Penggunaan Website Inspektorat Kota Palembang*. s.l.:Universitas Bina Darma.
- Simarmata, J., 2010. "Rekayasa Perangkat Lunak".
- Simatupang, R., 2005. *"Bencana Alam dan Masalahnya"*. s.l.:Yayasan Lembaga Alam Indonesia.
- Supriyono, P., 2014. *"Seri Pendidikan Pengurangan Risiko Bencana Gunung Meletus"*. Yogyakarta: Andi Offset.

Team, J., 2015. "Introducing JSON". [Online] Available at: [http:// json.org](http://json.org)
[Accessed 8 Oktober 2015].



LAMPIRAN 1

KODE PROGRAM

MainActivity.java

```

package co.fppc.ig;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;

import java.io.File;

import co.fppc.ig.database.model.TmpPhoto;
import co.fppc.ig.database.model.User;

public class MainActivity extends Activity {
    private TmpPhoto tmpPhoto;
    private User user;
    private EditText editTextPhotoCaption;
    private Intent intent;
    private Spinner spinner;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        intent = getIntent();
        if (intent != null) {
            if (intent.getSerializableExtra("USER_SESSION") != null) {

                user = (User)
intent.getSerializableExtra("USER_SESSION");
                Log.e("_USER_ID_", String.valueOf(user.getId()));
                Log.e("_USER_EMAIL_", user.getEmail());
                Log.e("_USER_PASSWORD_", user.getPassword());

                Button buttonReCapture = (Button)
findViewById(R.id.buttonCaptureIncident);
                buttonReCapture.setOnClickListener(new
View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Intent intent = new
Intent(getApplicationContext(), CameraActivity.class);
                        intent.putExtra("USER_SESSION", user);
                        startActivity(intent);
                    }
                });
                editTextPhotoCaption = (EditText)

```

```

findViewById(R.id.editTextCaption);

        }

        if (intent.getSerializableExtra("TMP_PHOTO") != null) {

            tmpPhoto = (TmpPhoto)
intent.getSerializableExtra("TMP_PHOTO");
            editTextPhotoCaption.setEnabled(true);
            editTextPhotoCaption.setHint("Insert caption");

            final String tmpPhotoName = tmpPhoto.getPhotoName();

            File sdCard =
Environment.getExternalStorageDirectory();
            File dir = new File(sdCard.getAbsolutePath() + "/" +
BuildConfig.APPLICATION_ID + "/" + tmpPhotoName);
            Uri imgUri = Uri.fromFile(dir);
            ImageView imageView = (ImageView)
findViewById(R.id.imageViewTmpPhotoPreviewer);
            imageView.setImageURI(imgUri);

            spinner = (Spinner) findViewById(R.id.spinnerKategori);

            Button buttonProcessToMap = (Button)
findViewById(R.id.buttonProcessToMap);
            buttonProcessToMap = (Button)
findViewById(R.id.buttonProcessToMap);
            buttonProcessToMap.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {

                    String text =
spinner.getSelectedItem().toString();
                    Log.e("__SPINNER__", text);
                    tmpPhoto.setCategory(text);
                    Intent intent = new
Intent(getApplicationContext(), MapActivity.class);

                    tmpPhoto.setPhotoCaption(editTextPhotoCaption.getText().toString());
                    intent.putExtra("USER_SESSION", user);
                    intent.putExtra("TMP_PHOTO", tmpPhoto);
                    // HEREEEEEE...
                    //Log.d("CAPTIONN_OK__BROOOS",
                    editTextPhotoCaption.getText().toString());
                    startActivity(intent);

                }
            });

        } else {
            editTextPhotoCaption.setHint("No photo available,
please capture image of an incident");
            editTextPhotoCaption.setEnabled(false);
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

```

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_show_camera) {
        Intent intent = new Intent(getApplicationContext(),
CameraActivity.class);
        intent.putExtra("USER_SESSION", user);
        startActivity(intent);
    } else if (id == R.id.action_process) {
        if (intent != null) {
            if (intent.getSerializableExtra("TMP_PHOTO") != null) {

                Intent intent = new Intent(getApplicationContext(),
MapActivity.class);

                tmpPhoto.setPhotoCaption(editTextPhotoCaption.getText().toString());
                intent.putExtra("TMP_PHOTO", tmpPhoto);
                // HEREEEEEE...
                //Log.d("CAPTIONN_OK_BROOS",
                editTextPhotoCaption.getText().toString());
                startActivity(intent);
            }
        }
    } else if (id == R.id.action_show_incidents_map) {
        Intent intent = new Intent(getApplicationContext(),
IncidentsMapActivity.class);
        intent.putExtra("USER_SESSION", user);
        startActivity(intent);
    } else if (id == R.id.action_logout) {
        Intent intent = new Intent(getApplicationContext(),
LoginActivity.class);
        startActivity(intent);
        finish();
    }

    return super.onOptionsItemSelected(item);
}
```