

# Rancang Bangun Aplikasi Perangkat Bergerak Pendataan Kondisi Bencana (Studi Kasus: MDMC Jawa Timur)

Achmad Affandi dan Radityo Prasetianto Wibowo

Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: radityo\_pw@is.its.ac.id.

**Abstrak**—Bencana merupakan keresahan tersendiri bagi masyarakat yang menjadi korban. Segala bentuk bencana baik bencana alam, non-alam, ataupun sosial merupakan permasalahan bersama yang dihadapi masyarakat. Proses pencegahan dan mitigasi, kesiapsiagaan, tanggap darurat, hingga rehabilitasi dan rekonstruksi dilakukan untuk meminimalkan dampak dari bencana yang akan atau telah terjadi. Badan Nasional Penanggulangan Bencana (BNPB) merupakan lembaga utama yang mengoordinasikan berbagai lembaga-lembaga penanggulangan bencana di Indonesia. Muhammadiyah Disaster Management Center (MDMC) merupakan salah satu lembaga sosial kebencanaan yang berada di bawah Pimpinan Pusat Muhammadiyah. Dalam hal pelaksanaan penanggulangan bencana, MDMC dibantu oleh berbagai relawan dari regional Muhammadiyah. MDMC Jawa Timur sebagai salah satu regional juga turut membantu segala bentuk penanggulangan bencana yang terjadi di Indonesia. Dalam proses pengiriman bantuan pada umumnya diperlukan proses pendataan yang tepat dan cepat agar kebijakan lebih lanjut dapat segera dilaksanakan. Pengembangan aplikasi mobile disaster assessment berbasis Android dapat ditujukan untuk memudahkan relawan dalam proses pendataan kondisi dan kebutuhan saat terjadinya bencana. Dalam hal mengurangi risiko kurangnya akses terhadap internet, teknologi document oriented database dapat dimanfaatkan untuk tetap menyimpan data pada ponsel. Pada penelitian ini dikembangkan aplikasi Android yang dapat dijalankan dengan baik pada versi Android Marshmallow+. Aplikasi dapat menyimpan data saat tidak ada koneksi internet dan secara langsung mengirimkan data saat perangkat kembali terkoneksi ke internet.

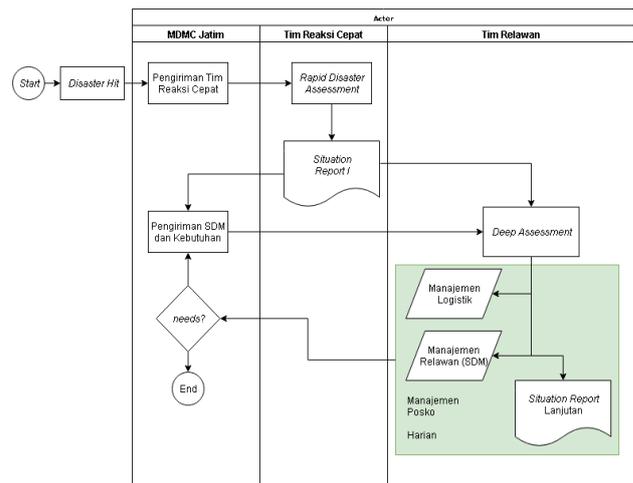
**Kata Kunci**—Android, Assessment, Bencana, Firebase, MDMC.

## I. PENDAHULUAN

**B**ENCANA adalah peristiwa atau rangkaian peristiwa yang mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan, baik oleh faktor alam dan/atau faktor non-alam maupun faktor manusia sehingga mengakibatkan timbulnya korban jiwa manusia, kerusakan lingkungan, kerugian harta benda, dan dampak psikologis [1].

Dalam kurun waktu tiga bulan awal 2018, Indonesia setidaknya telah mengalami 513 bencana alam. Dua bencana besar terjadi pada Agustus dan September 2018 dengan adanya gempa di Lombok, serta Tsunami di kawasan Sulawesi Tengah [2].

Lembaga-lembaga kemanusiaan menjadi lembaga terdepan dari proses penanggulangan bencana, diantaranya Tim SAR, Palang Merah Indonesia, serta berbagai lembaga kemanusiaan lain dari berbagai yayasan di Indonesia. Seluruh



Gambar 1. Garis besar proses tanggap darurat yang dilakukan oleh MDMC

aktivitas penanggulangan bencana dari masing-masing kelompok tersebut dikoordinasi oleh Badan Nasional Penanggulangan Bencana (BNPB).

*Muhammadiyah Disaster Management Center (MDMC)* merupakan salah satu lembaga sosial kebencanaan yang berada di bawah Pimpinan Pusat Muhammadiyah. MDMC bergerak dalam kegiatan penanggulangan bencana mencakup mitigasi dan kesiapsiagaan, tanggap darurat dan rehabilitasi. Dalam hal pelaksanaan penanggulangan bencana, MDMC dibantu oleh berbagai relawan dari regional Muhammadiyah.

MDMC Jawa Timur sebagai salah satu regional juga turut membantu segala bentuk penanggulangan bencana yang terjadi di Indonesia. Dalam proses pengiriman bantuan pada umumnya diperlukan proses pendataan yang tepat dan cepat agar kebijakan lebih lanjut dapat segera dilaksanakan.

Pada proses penanganan bencana, relawan melakukan pendataan kondisi dan kebutuhan saat terjadinya bencana. Saat ini proses pendataan masih dilakukan secara manual dan tanpa ada indikator tetap, sehingga koordinasi antarrelawan mengalami kendala.

Pihak MDMC Jatim juga membutuhkan laporan kondisi dan kebutuhan di lokasi bencana setiap harinya, untuk dijadikan dokumentasi dan referensi hingga proses rehabilitasi dan rekonstruksi.

Berdasarkan penjelasan latar belakang di atas, rumusan masalah yang menjadi fokus utama dalam studi ini adalah: 1) Bagaimana pihak MDMC Jawa Timur mendapatkan informasi mengenai kondisi dan kebutuhan saat terjadi bencana; 2) Bagaimana memastikan data yang dibutuhkan untuk rapid assessment diperoleh dan digunakan sebagai data dasar pada penyusunan langkah selanjutnya; 3) Bagaimana

```

{
  "users": {
    "alovelace": {
      "name": "Ada Lovelace",
      "contacts": { "ghopper": true },
    },
    "ghopper": { ... },
    "eclarke": { ... }
  }
}
    
```

Gambar 2. Contoh JSON tree Firebase.

```

Java Kotlin
Android Android
private lateinit var database: DatabaseReference // ...
database = FirebaseDatabase.getInstance().reference
NewPostActivity.kt
    
```

Gambar 3. DatabaseReference pada Java

menerapkan *document oriented database* sebagai penyimpanan data kebencanaan?

Batasan dalam pengerjaan tugas akhir ini adalah: 1) Hasil akhir penelitian ini adalah pengembangan aplikasi Android sebagai pengambilan data; 2) Data yang digunakan adalah kebutuhan organisasi MDMC Jawa Timur

Tujuan dari penelitian ini adalah mengembangkan sistem pengambilan data kondisi dan kebutuhan saat terjadi bencana sehingga dapat membantu MDMC Jawa Timur mendapatkan laporan mengenai kondisi dan kebutuhan saat terjadi bencana dengan menerapkan *document oriented database* yaitu Firebase sebagai penyimpanan data kebencanaan.

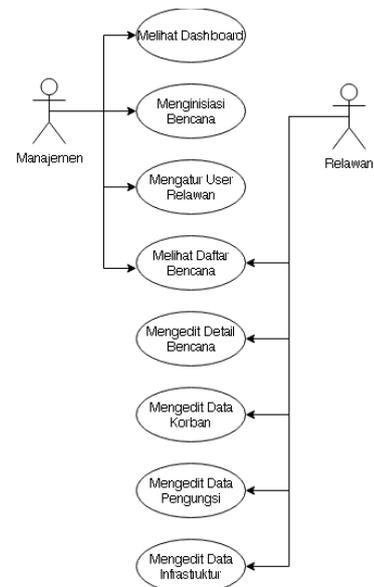
## II. TINJAUAN PUSTAKA

### A. Proses Tanggap Darurat

Proses tanggap darurat yang dilakukan MDMC Jatim saat terjadinya suatu bencana adalah sebagai berikut

- Pengiriman Tim Reaksi Cepat (TRC) yang akan mengambil data dan tindakan awal di lokasi bencana, pengiriman tim ini dilakukan sesegera mungkin dari proses terjadinya bencana.
- TRC selanjutnya memberikan laporan rekomendasi kepada pengurus MDMC Jatim terkait kebutuhan yang mendesak, serta kondisi di lokasi bencana.
- MDMC Jatim merespon laporan TRC dengan mengirim tim.
- Tim selanjutnya menganalisa kondisi dan menentukan pembagian SDM dan proses pengiriman logistik kepada korban yang membutuhkan.
- Tim relawan akan dibagi sesuai tugas dan penempatan yang disepakati, dan tetap menjaga koordinasi dengan posko koordinator.
- Tim relawan, logistik, atau kebutuhan lain secara berkala dipantau dan ditindak oleh MDMC Jatim.
- Bantuan tim relawan berlanjut hingga fase rehabilitasi dan rekonstruksi sesuai dengan kebutuhan terdampak bencana.

Setelah kondisi terdampak bencana dikatakan pulih, dilakukan proses manajemen risiko dan kesiapsiagaan terhadap kemungkinan bencana yang akan terjadi.



Gambar 4. Use case diagram aplikasi

Terdapat beberapa aktor yang berperan dalam sistem penanganan bencana berbasis Android ini. Berikut adalah daftar aktor dan fungsinya dalam proses penanganan laporan:

#### 1) Manajemen Pusat MDMC Jatim

Manajemen Pusat MDMC Jatim adalah yang memiliki akses utama aplikasi dan dapat melakukan segala manajemen sistem yang diperlukan. Pada proses tanggap bencana, Manajemen Pusat MDMC Jatim memiliki fungsi kontrol terhadap keseluruhan proses dan melakukan koordinasi untuk pengadaan bantuan yang mencakup sumber daya manusia (relawan), kendaraan, logistik, dan bantuan lainnya.

#### 2) Tim Reaksi Cepat

Tim Reaksi Cepat merupakan aktor yang memiliki fungsi melakukan rapid disaster assessment. Pada proses ini Tim Reaksi Cepat akan melakukan pendataan umum bencana yang terjadi dan mengirimkan laporan kepada Manajemen Pusat untuk memberikan keputusan bentuk bantuan yang diperlukan.

#### 3) Koordinator Posko

Koordinator Posko merupakan aktor yang memiliki fungsi utama dalam proses penanganan kebencanaan untuk melakukan koordinasi seluruh bantuan yang ada di lokasi bencana.

#### 4) Relawan

Relawan adalah aktor yang memiliki fungsi memasukkan semua data sesuai dengan kondisi yang ada di lokasi bencana.

### B. Android

Android adalah *software* untuk perangkat mobile yang meliputi sistem operasi, middleware dan aplikasi inti. Android dilengkapi dengan Android SDK (*Software Development Kit*) yang menyediakan tools dan mendukung kebutuhan API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java. Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode Java yang terkompilasi bersama-sama dengan data dan file *resources* yang dibutuhkan oleh aplikasi yang digabungkan oleh apt tools menjadi paket Android, sebuah file yang ditandai dengan suffix *.apk*. File ini didistribusikan sebagai aplikasi dan diinstal pada perangkat *mobile*.

Tabel 1.  
Kebutuhan Informasi Umum Bencana.

Informasi Umum Bencana	
Jenis Bencana	Jenis kejadian bencana memiliki pilihan diantaranya: 1. Banjir 2. Angin Topan 3. Gempa Bumi 4. Erupsi 5. Kebakaran 6. Lain-lain
Lokasi Kejadian	Lokasi kejadian memuat tampilan peta lokasi. Fitur ini memanfaatkan koordinat GPS dari Android. Sehingga dengan memanfaatkan <i>Google Location</i> , dapat terjabarkan pula lokasi kejadian seperti kabupaten, kecamatan, dsb
Tanggal Kejadian	
Akses Transportasi	Akses transportasi memiliki pilihan 1. Darat 2. Air 3. Udara Dan untuk pilihan darat, maka akan dilanjutkan pilihan berikutnya yaitu a. Mobil b. Truk c. Motor

Tabel 2.  
Kebutuhan Informasi Detail Bencana.

Informasi Detail Bencana		
Kondisi lokasi kejadian	Kondisi listrik	Memuat data ada atau tidaknya listrik di lokasi kejadian, dan sumber listrik terkait (genset, PLN)
	Kondisi Air	Memuat data ketersediaan air, dan sumber.
	Sanitasi	Memuat a. Kondisi saluran drainase (hancur/ tersumbat/ lancar) b. Ketersediaan jamban (jumlah) c. Ketersediaan tempat pembuangan sampah
Data korban	Meninggal	
	Hilang	
	Terluka	
	Mengungsi	Seluruh data dikategorikan berdasarkan usia a. Balita b. Anak-anak (5-12 tahun) c. Remaja (12-18 tahun) d. Dewasa (18-60 tahun) e. Lansia (>60 tahun) serta jenis kelamin dari setiap kategori tersebut laki-laki atau perempuan
Data infrastruktur	Rumah hancur	
	Rumah rusak berat	
	Rumah rusak ringan	

Fitur – fitur yang terdapat pada Android yaitu:

- *Application Framework*,
  - *Dalvik Virtual Machine*,
  - *Integrated Browser*,
  - *Optimized graphics*,
  - *SQLite*,
  - Media pendukung untuk audio, video, dan format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF),
  - *GSM Telephony* (tergantung perangkat *mobile*),
  - Bluetooth, EDGE, 3G, dan WiFi (tergantung perangkat *mobile*),
  - Kamera, GPS, kompas, dan *accelerometer* (tergantung perangkat *mobile*),
  - *Rich Development Environment*.
- Android merupakan kernel Linux yang menyediakan dan mengatur alur proses aplikasi. Arsitektur Android terdiri dari:
- 1) Application (System Apps).
  - 2) Application Framework,

Tabel 3.  
Kebutuhan Informasi Pada *Dashboard*

Kebutuhan pasokan pangan	Didapatkan dari data total pengungsi yang ada di lokasi bencana
Kebutuhan shelter/rumah	Didapatkan dari total rumah yang hancur, baik rusak berat maupun rusak ringan
Kebutuhan balita	Didapatkan dari total balita yang menjadi pengungsi. Kebutuhan yang dimaksud mencakup bubur bayi, popok
Kebutuhan lansia	Didapatkan dari total lansia yang menjadi pengungsi.
Kebutuhan khusus perempuan	Didapatkan dari total remaja perempuan ditambah jumlah perempuan dewasa yang menjadi pengungsi. Kebutuhan yang dimaksud seperti pembalut

Tabel 4.  
Data Bencana (*Disaster*).

Informasi	Data field	Keterangan
Bencana	jenisBencana	Merupakan data jenis bencana yang dapat digunakan sebagai identitas (nama) bencana.
Tanggal Kejadian	tglKejadian	Data tanggal kejadian bencana
Lokasi Kejadian	latLokasi	Data <i>latitude</i> lokasi bencana
	longLokasi	Data <i>longitude</i> lokasi bencana. latLokasi dan longLokasi digunakan untuk proses penampilan pada Google Maps
	alamat	Data alamat lengkap yang diambil dari <i>library</i> Googlelocation
	kabupaten	Data kabupaten yang diambil dari data alamat digunakan untuk mempersingkat lokasi
Akses Transportasi	aksesTransportasi	Data akses yang dapat dilalui untuk menuju lokasi, apakah darat, air, ataupun udara.
	alatTransportasi	Data alat transportasi yang memungkinkan untuk digunakan sebagai media utama ketika di lokasi bencana
Keterangan Lain	ketLain	Informasi atau keterangan lain yang mungkin diperlukan
Kondisi Listrik	kondisiListrik	Data ketersediaan listrik di lokasi bencana
	sumberListrik	Data sumber listrik yang ada
Kondisi Air	kondisiAir	Data ketersediaan akses terhadap air di lokasi bencana
	sumberAir	Data sumber air di lokasi bencana
Kondisi Sanitasi	kondisiDrainase	Informasi mengenai kondisi drainase, tersumbat, lancar, atau hancur.
	jumlahJamban	Data jumlah jamban yang mungkin untuk digunakan selama proses penanganan bencana
Data Pengungsi	lakiBalita	Jumlah pengungsi laki-laki berusia di bawah 5 tahun.
	perempuanBalita	Jumlah pengungsi perempuan berusia di bawah 5 tahun
	lakiAnak	Jumlah pengungsi laki-laki yang berusia antara 5 hingga 12 tahun
	perempuanAnak	Jumlah pengungsi perempuan yang berusia antara 5 hingga 12 tahun
	lakiRemaja	Jumlah pengungsi laki-laki yang berusia antara 13 hingga 17 tahun
	perempuanRemaja	Jumlah pengungsi perempuan yang berusia antara 13 hingga 17 tahun
	lakiDewasa	Jumlah pengungsi laki-laki yang berusia antara 18 hingga 60 tahun
	perempuanDewasa	Jumlah pengungsi perempuan yang berusia antara 18 hingga 60 tahun
	lakiLansia	Jumlah pengungsi laki-laki yang berusia lebih dari 60 tahun
perempuanLansia	Jumlah pengungsi perempuan yang berusia lebih dari 60 tahun	
Data Korban	korbanMeninggal	Jumlah korban meninggal
	korbanHilang	Jumlah korban yang hilang
	korbanLukaBerat	Jumlah korban yang mengalami luka berat
	korbanLukaRingan	Jumlah korban yang mengalami luka ringan
Data Infrastruktur	rumahHancur	Jumlah rumah yang hancur
	rumahRusakBerat	Jumlah rumah yang mengalami kerusakan berat
	rumahRusakRingan	Jumlah rumah yang mengalami kerusakan ringan

- 3) Libraries,
- 4) Android Runtime,
- 5) Hardware Abstraction Layer
- 6) Linux Kernel [3]

Dari keenam komponen diatas, komponen yang paling dikenal adalah *System Apps* karena itulah aplikasi-aplikasi yang terlihat dan digunakan oleh para user. Aplikasi juga dapat dikembangkan secara bebas dan gratis oleh para pengembang (*developer*) menggunakan *Android Software Development Kit*, baik Android Studio yang disediakan

langsung dari Google maupun menggunakan IDE lainnya. Dengan tersedianya *development platform* gratis, pengembang Android dapat membangun aplikasi yang memanfaatkan penuh fitur *smartphone* seperti akses informasi lokasi, menjalankan *background services*, kamera, membuat notifikasi di status bar, dll.

Android Studio adalah sebuah IDE untuk pengembangan aplikasi di platform Android. Pada Android Studio, SDK menggunakan IntelliJ IDEA Community Edition. Untuk manajemen proyeknya, Android Studio menggunakan Gradle

Tabel 5.  
Data Pengguna (User).

Informasi	Data field	Keterangan
Nama Pengguna	nama	Nama jelas dari pengguna
Kontak	email	Email dari pengguna
	phone	Nomor ponsel pengguna
Tipe Pengguna	type	Tipe pengguna, apakah manajemen, atau relawan

Tabel 6.  
Data penugasan relawan (userassign)

Informasi	Data field	Keterangan
Kunci data bencana	disasterID	Merupakan <i>node</i> yang akan berisikan pengguna siapa saja yang memiliki hak akses untuk bencana terkait
Kunci data pengguna	userID	Akan berisikan keterangan apakah pengguna ditugaskan di bencana terkait, sehingga memuat nilai "true" atau "false", namun tetap disimpan dalam bentuk String untuk kemudahan pengelolaan database

Tabel 7.  
Spesifikasi Komputer Pengembang

Perangkat	Deskripsi
Laptop	ASUS X453SA
Processor	Intel Dual-Core N3050 @1.60GHz
RAM	4GB
Hard Disk Drive	466GB

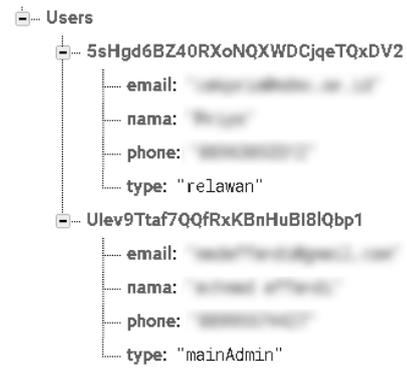
Tabel 8.  
Teknologi pengembangan yang digunakan

Perangkat	Deskripsi
Sistem Operasi	Microsoft Windows 10
Bahasa Pemrograman	Java
IDE	Android Studio 4
Database	Firestore
Library	- Googlelocation - GoogleMaps - CardView - Firebase RecyclerView - Firebase Authentication
minSdkVersion	API 16
compiledSdkVersion	API 28

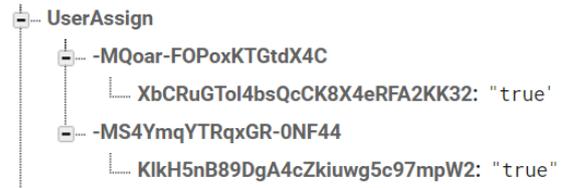
yakni *build automation tool* yang dapat dikonfigurasi melalui *Domain Specific Languages (DSL)* berbasis Groovy. Inilah yang membedakan Gradle dari Ant atau Maven yang memakai XML. Penggunaan DSL berbasis Groovy membuat Gradle menjadi lebih fleksibel dan dapat diprogram dengan mudah.

C. Firebase

Firebase adalah produk utama dari perusahaan Firebase, Inc. yang sudah diakuisisi oleh Google. Merupakan *platform* untuk pengembangan aplikasi *mobile* dan web. Layanan ini menyediakan pengembangan aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di *cloud* Firebase ini. Firebase menyediakan *library* untuk berbagai *client* yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Objective-C. Berbagai bentuk produk Firebase



Gambar 5. Struktur Basis Data Pengguna (User).



Gambar 6. Struktur Basis Data Penugasan (userassign)

```

69 //fungsi memasukkan data bencana awal ke database
70 public void inputDataBencana() {
71     mDatabase = FirebaseDatabase.getInstance().getReference().child("Disaster").push();
72     DisasterData dData = new DisasterData(getJenisBencana(), getTglKejadian(),
73     getLatLokasi(), getLongLokasi(), getAlamat(), getKabupaten(), getAksesTrans(),
74     getLatTrans(), getKetLain(), isCompleted: "false");
75     mDatabase.setValue(dData);
76     Toast.makeText(context: CreateDisasterActivity.this,
77     text: "Data bencana baru telah ditambahkan", Toast.LENGTH_SHORT).show();
78     Intent i = new Intent(context: CreateDisasterActivity.this, DisListActivity.class);
79     i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
80     startActivity(i);
81 }
    
```

Gambar 7. Potongan kode inisiasi data bencana

```

16 //method untuk menambahkan data ke database dengan menggunakan DIS_ID
17 private void updateDataBencana(String DIS_ID, String kondisiListrik,
18     String sumberListrik, String kondisiAir, String sumberAir,
19     String kondisiDrainase, String jumlahJamban) {
20     String key = DIS_ID;
21     mDataDis.child(key).child("kondisiListrik").setValue(kondisiListrik);
22     mDataDis.child(key).child("sumberListrik").setValue(sumberListrik);
23     mDataDis.child(key).child("kondisiAir").setValue(kondisiAir);
24     mDataDis.child(key).child("sumberAir").setValue(sumberAir);
25     mDataDis.child(key).child("kondisiDrainase").setValue(kondisiDrainase);
26     mDataDis.child(key).child("jumlahJamban").setValue(jumlahJamban);
27 }
    
```

Gambar 8. Potongan kode penambahan data

diantaranya Cloud Firestore, Authentication, Cloud Storage, dan Realtime Database.

Pada penelitian ini, fitur Firebase yang digunakan adalah Realtime Database dan Authentication. Firebase merupakan layanan *database cloud* tanpa SQL, data yang disimpan dalam bentuk JSON disinkronisasikan melalui perangkat yang terkoneksi dan dapat digunakan meski aplikasi tidak terhubung dengan internet.

Data dalam Firebase Realtime Database disimpan dalam bentuk JSON *object*. Semua data disimpan dalam *cloud* dalam bentuk JSON tree, tidak seperti SQL yang menyimpan data dalam bentuk tabel. Saat melakukan penambahan data dalam JSON tree maka data menjadi sebuah *node* dalam struktur JSON dan memiliki *key* tersendiri yang bisa kita buat dengan *userID* atau *semantic names*. Bentuk JSON tree yang menyimpan data dapat dilihat seperti pada Gambar 2

Pada skema di atas menunjukkan tree "users" memiliki tiga data, yaitu "alovelace", "ghopper", dan "eclarke". Serta pada data "alovelace" terdapat dua jenis data yang memiliki id "name" dan "contacts".

Untuk membaca atau menulis *database* hanya dapat dilakukan oleh *user* yang terautentifikasi. Pada Firebase

```

68 mDataDis.addValueEventListener(new ValueEventListener() {
69     @Override
70     public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
71         for (DataSnapshot ds : dataSnapshot.getChildren()) {
72             if (ds.getKey().equals(DIS_ID)) {
73                 tv_judulDis.setText(ds.child("jenisBencana").getValue().toString());
74                 tv_subJudulDis.setText(ds.child("kabupaten").getValue().toString());
75                 tv_tglKejadian.setText(ds.child("tanggalKejadian").getValue().toString());
76                 tv_latLongLokasi.setText(ds.child("latLokasi").getValue().toString()
77                     + ", " + ds.child("longLokasi").getValue().toString());
78                 tv_alamatLokasi.setText(ds.child("alamat").getValue().toString());
79                 tv_aksesTransportasi.setText(ds.child("aksesTransportasi").getValue().toString()
80                     + ", " + ds.child("alatTransportasi").getValue().toString());
81                 tv_keteranganLain.setText(ds.child("keteranganLain").getValue().toString());
82                 tv_kondisiListrik.setText(ds.child("kondisiListrik").getValue().toString()
83                     + ", " + ds.child("sumberListrik").getValue().toString());
84                 tv_kondisiAir.setText(ds.child("kondisiAir").getValue().toString()
85                     + ", " + ds.child("sumberAir").getValue().toString());
86                 tv_kondisiDrainase.setText(ds.child("kondisiDrainase").getValue().toString()
87                     + ", " + ds.child("jumlahJamban").getValue().toString() + " jamban");
88                 tv_pengungsiBalita.setText(ds.child("lakiBalita").getValue().toString()
89                     + " laki-laki, \n" + ds.child("perempuanBalita").getValue().toString()
90                     + " perempuan");
91                 tv_pengungsiAnak.setText(ds.child("lakiAnak").getValue().toString()
92                     + " laki-laki, \n" + ds.child("perempuanAnak").getValue().toString()
93                     + " perempuan");
94                 tv_pengungsiRemaja.setText(ds.child("lakiRemaja").getValue().toString()
95                     + " laki-laki, \n" + ds.child("perempuanRemaja").getValue().toString()
96                     + " perempuan");
97                 tv_pengungsiDewasa.setText(ds.child("lakiDewasa").getValue().toString()
98                     + " laki-laki, \n" + ds.child("perempuanDewasa").getValue().toString()
99                     + " perempuan");
100             }
101         }
102     }
103 }

```

Gambar 9. Potongan Kode Menampilkan Detail Bencana.

```

114 //menuju edit bencana
115 btn_editBencana.setOnClickListener(new View.OnClickListener() {
116     @Override
117     public void onClick(View v) {
118         Intent i = new Intent(DisReviewActivity.this,
119             CreateReviewDisasterActivity.class);
120         i.putExtra("DIS_ID", DIS_ID);
121         startActivity(i);
122     }
123 });

```

Gambar 10. Potongan Kode Edit Bencana.

```

150 //konfirmasi penghapusan data
151 private AlertDialog AskOption() {
152     AlertDialog myQuittingDialogBox = new AlertDialog.Builder(this)
153         .setTitle("Hapus")
154         .setMessage("apakah Anda yakin untuk menghapus data ini?")
155         //setIcon(R.drawable.delete)
156         .setPositiveButton("Hapus", new DialogInterface.OnClickListener() {
157             public void onClick(DialogInterface dialog, int whichButton) {
158                 //menghapus data dari database
159                 mDataDis.child(DIS_ID).removeValue();
160                 Intent i = new Intent(DisReviewActivity.this,
161                     DisListActivity.class);
162                 startActivity(i);
163                 Toast.makeText(DisReviewActivity.this, "Data berhasil dihapus",
164                     Toast.LENGTH_SHORT).show();
165                 dialog.dismiss();
166             }
167         })
168         .setNegativeButton("Batal", new DialogInterface.OnClickListener() {
169             public void onClick(DialogInterface dialog, int which) {
170                 dialog.dismiss();
171             }
172         })
173         .create();
174     return myQuittingDialogBox;
175 }

```

Gambar 11. Potongan kode konfirmasi penghapusan data

untuk membaca atau menulis data dari database, diperlukan *instance DatabaseReference* (lihat Gambar 3).

### III. METODOLOGI

#### A. Studi Literatur

Pada tahap ini dilakukan studi literatur terhadap permasalahan yang telah diidentifikasi, yaitu mengenai bagaimana MDMC Jawa Timur dapat menerima laporan yang efektif dalam proses penanganan tanggap bencana yang dilakukan oleh relawan Tim Relawan MDMC Jatim.

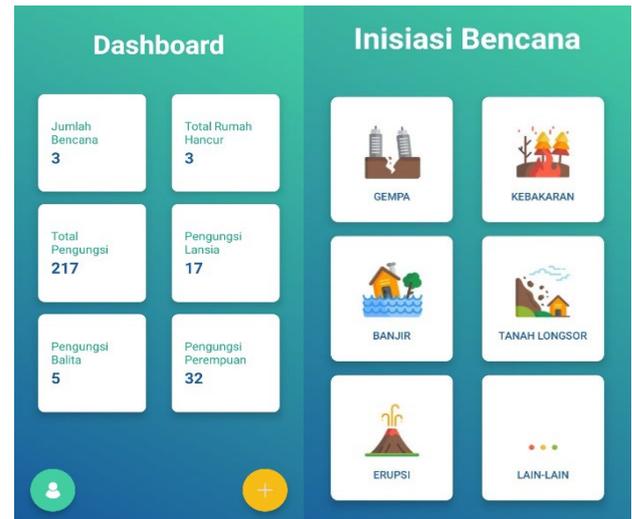
Untuk membantu MDMC Jatim dalam mendapatkan laporan kondisi kebencanaan perlu dipelajari metode pengembangan aplikasi Android, metode penggunaan *document oriented database* yaitu Firebase sebagai basis data, serta dasar proses tanggap bencana yang dilakukan oleh MDMC Jatim.

```

51 //mendapatkan peran pengguna, untuk dilanjutkan ke intent berikutnya
52 mDataRef.child("Users").child(FirebaseAuth.getInstance()
53     .getCurrentUser().getUid()).child("type")
54     .addListenerForSingleValueEvent(new ValueEventListener() {
55         @Override
56         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
57             setUserType(dataSnapshot.getValue().toString());
58         }
59     })
60     @Override
61     public void onCancelled(@NonNull DatabaseError databaseError) {
62     }
63 });

```

Gambar 12. Potongan kode mendapatkan peran pengguna



Gambar 13. Antarmuka aplikasi

#### B. Analisis Kebutuhan Sistem

Pada tahap ini dilakukan identifikasi kebutuhan pada sistem informasi yang akan dibangun. Proses identifikasi kebutuhan dilakukan dengan cara wawancara dengan Tim MDMC Jawa Timur. Selanjutnya hasil wawancara dianalisis untuk mengetahui kebutuhan fungsional dan kebutuhan non fungsional sistem. Hasil dari tahap ini adalah klasifikasi kebutuhan informasi, dan *use case diagram*.

#### C. Desain Sistem

Pada tahap ini peneliti melakukan pembuatan desain sistem. Desain sistem ini akan digunakan sebagai *“blueprint”* dalam pembangunan sistem ini. Desain sistem dirancang dengan menggunakan *Unified Modeling Language (UML)*. UML adalah bahasa standar yang digunakan untuk proses dokumentasi, pembuatan spesifikasi dan membangun sebuah sistem perangkat lunak. Pada tahap ini peneliti juga melakukan pembuatan desain/ *prototype* antarmuka aplikasi yang akan digunakan sebagai acuan pembuatan *user interface* aplikasi. Peneliti juga membuat rancangan struktur data berdasarkan data yang diperlukan dalam pengembangan aplikasi.

Hasil akhir dari tahap ini berupa *activity diagram*, desain interface aplikasi, dan desain struktur data aplikasi. Sebelum hasil akhir tahap ini digunakan pada tahap selanjutnya akan dilakukan validasi terlebih dahulu.

#### D. Pembangunan Sistem

Pada tahap ini peneliti melakukan pembangunan sistem yaitu pengkodean dengan menggunakan bahasa

Tabel 7.  
Hasil Pengujian Pada Berbagai Versi Android.

Versi Android	Keterangan
11.0 (API 30)	Berjalan dengan baik
10 (API 29)	Berjalan dengan baik
Pie (API 28)	Berjalan dengan baik
Oreo (API 26)	Berjalan dengan baik
Nougat (API 24)	Berjalan dengan baik
Marshmallow (API 23)	Berjalan dengan baik
Lollipop (API 21)	Dapat melakukan <i>login</i> , tetapi Cardview tidak ditampilkan
KitKat (API 19)	Dapat melakukan <i>login</i> , tetapi CardView tidak menampilkan isi
JellyBean (API 16)	Tidak dapat melakukan <i>login</i> .

pemrograman Java pada pengembangan aplikasi Android dan Firebase sebagai basis data dengan mengacu pada hasil dari desain sistem sebelumnya dan akan dilakukan validasi untuk memastikan fungsi yang dibangun dapat berfungsi.

E. Pengujian Sistem

Pada tahap ini peneliti melakukan pengujian sistem terhadap aplikasi yang dibangun. Tahap ini bertujuan untuk melihat kesesuaian dengan desain yang telah dibuat dengan fungsionalitasnya serta mengetahui apakah terdapat kesalahan-kesalahan, baik secara tampilan informasi maupun kode program. Proses ini dilakukan sampai aplikasi sesuai dengan desain yang dibuat pada proses sebelumnya.

F. Pembuatan Laporan Penelitian

Pada tahap ini peneliti melakukan dokumentasi terhadap hasil penelitian dari setiap langkah dan ditulis dalam bentuk studi hingga terbentuk buku.

IV. PERANCANGAN

A. Analisis Kebutuhan Sistem

Pada tahap ini dilakukan analisis kebutuhan dari aplikasi yang akan dibuat. Dari hasil wawancara dengan pihak Muhammadiyah Disaster Management Center (MDMC) Jawa Timur, dan dari Pedoman Assessment Tanggap Darurat MDMC dapat dirumuskan data-data apa saja yang dibutuhkan dalam proses penanggulangan kebencanaan yang akan diakumulasi oleh aplikasi ini.

Tabel 1 menunjukkan kebutuhan informasi umum. Kebutuhan ini disusun berdasarkan data paling penting dalam proses pendataan awal yang dilakukan oleh pihak manajemen. Selanjutnya Tim Reaksi Cepat (TRC) yang akan memberikan data lebih detail sebagaimana disebutkan pada Tabel 2.

B. Desain Sistem

Pembuatan Use Case Diagram digunakan untuk menganalisa bagai sistem berinteraksi dengan pengguna. Use Case Diagram menunjukkan autentikasi informasi yang dapat diakses oleh masing-masing pengguna. Aplikasi yang dibuat ini memiliki dua peran (role) pengguna yaitu Manajemen dan Relawan. Masing-masing peran memiliki hak akses yang berbeda terhadap data dan informasi yang ada di aplikasi.

```

64  if (android.os.Build.VERSION.SDK_INT <= android.os.Build.VERSION_CODES.LOLLIPOP){
65      //memberikan peringatan untuk versi Android
66      AlertDialog checkAndroidVersion = new AlertDialog.Builder( context: this)
67          .setTitle("Peringatan")
68          .setMessage("versi Android Anda tidak direkomendasikan, " +
69                    "gunakan API 23 (Marshmallow) ke atas")
70          .setPositiveButton( text: "OK", new DialogInterface.OnClickListener() {
71              public void onClick(DialogInterface dialog, int whichButton) {
72                  finish();
73                  System.exit( status: 0);
74                  dialog.dismiss();
75              }
76          })
77          .create();
78      checkAndroidVersion.show();
79  }
    
```

Gambar 14. Potongan kode peringatan versi Android

Perancangan terhadap data dilakukan untuk mengetahui data apa saja yang perlu digunakan dan penyusunan struktur data seperti apa yang akan digunakan oleh aplikasi. Pada pengembangan aplikasi ini data yang digunakan adalah dua data yaitu data pengguna dan data bencana. Tabel 4 menjelaskan isi dari masing-masing data.

Data pengguna pada Tabel 4 akan diinisiasi dalam bentuk node pada basis data Firebase. Masing-masing node akan merepresentasikan tabel terkait.

Selain data pada Tabel 5 dan Tabel 6, basis data juga menyimpan password pengguna dengan menggunakan library Firebase authentication. Penelitian akan menggunakan Firebase database untuk menyimpan data pengguna, dan menggunakan Firebase authentication untuk proses autentikasi aplikasi.

V. IMPLEMENTASI

A. Lingkungan Implementasi

Pada bagian ini akan dibahas terkait lingkungan implementasi untuk pengerjaan tugas akhir ini. Lingkungan yang dimaksud adalah perangkat keras, perangkat lunak, serta teknologi lain yang digunakan untuk proses pengerjaan tugas akhir ini. Tabel berikut menunjukkan tentang spesifikasi perangkat komputer pengembang untuk implementasi pada studi ini.

B. Implementasi Basis Data

Gambar 5 menunjukkan implementasi struktur data pengguna pada node Users yang memuat data mendasar pengguna. Sedangkan Gambar 6 menunjukkan struktur data penugasan pengguna (relawan) pada tiap bencana yang terjadi. Pada data ini, struktur data ditandai dengan ID (key) dari bencana yang ada pada data bencana, dan menautkan ID (key) pengguna sesuai dengan penugasan terkait.

C. Implementasi Kode

Pelaporan kebencanaan dilakukan melalui pembuatan data kebencanaan baru. Data yang dibuat pada tahap ini mencakup jenis bencana, tanggal kejadian, lokasi kejadian, dan kondisi akses lokasi.

Gambar 7 merupakan proses inisiasi pembuatan data bencana baru pada database. Gambar 8 menunjukkan proses penambahan data bencana dengan detail awal yaitu kondisi listrik, kondisi air, dan kondisi sanitasi. Proses ini memanfaatkan key yaitu key node pada data untuk bencana terkait. Proses penambahan data lain juga menggunakan metode yang sama dengan Gambar 8.

Tampilan detail bencana diakses dengan menggunakan kode pada Gambar 9. Data yang didapatkan dari basis data kemudian ditampilkan pada TextView terkait.

Proses perubahan dapat dilakukan jika data sudah dibuat. Setelah data dibuat, pengguna dapat melakukan perubahan terhadap data terkait dari laman ReviewDisaster. Perubahan data pada dasarnya sama dengan membuat data baru pada bencana terkait yang telah ada.

Gambar 10 menunjukkan potongan kode untuk akses tombol menuju laman CreateReviewDisaster yang selanjutnya dapat dilakukan pendataan lebih lanjut baik data korban, pengungsi, maupun infrastruktur.

Gambar 11 di bawah menunjukkan proses AlertDialog melakukan konfirmasi terhadap pengguna. Jika pengguna memilih untuk menghapus, maka data pada database akan terhapus.

Peran pengguna aplikasi ini adalah admin dan relawan. Proses identifikasi dilakukan dengan membaca data pengguna (Users) yang sebelumnya sudah dibuat. Gambar 12 menunjukkan pemanggilan peran pengguna dengan child "type".

UserType ini akan digunakan sebagai identifikasi untuk setiap fitur ataupun fungsi yang memerlukan pembeda antarperan.

#### D. Implementasi Antarmuka

Implementasi antarmuka aplikasi disesuaikan dengan rancangan sebelumnya. Berikut adalah implementasi dari masing-masing laman yang dikembangkan dalam penelitian.

## VI. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan menggunakan emulator Android Virtual Devices yang disediakan oleh Android Studio. Pengujian dilakukan pada smartphone yang sama yaitu Google Pixel dengan versi Android yang berbeda-beda seperti pada Tabel 7.

Untuk menghindari kesalahan performa dalam penggunaan aplikasi, maka dilakukan pengecekan pada versi Android yang terdapat pada perangkat. Jika versi Android

perangkat masih versi Lollipop atau sebelumnya, maka akan dimunculkan peringatan untuk menggunakan perangkat yang lebih mutakhir. Gambar 14 adalah kode untuk melakukan pengecekan tersebut.

## VII. KESIMPULAN DAN SARAN

Pada studi ini telah dikembangkan aplikasi perangkat bergerak (mobile application) sebagai media untuk membantu pendataan kondisi kebencanaan. Hasil pengujian fungsional menunjukkan bahwa aplikasi mampu menjalankan semua fungsinya sesuai dengan harapan. Pengujian non fungsional menunjukkan bahwa aplikasi dapat berjalan dengan baik pada Android versi Marshmallow+. Aplikasi dapat menyimpan data pada saat terputus dari koneksi internet dan secara langsung mengirimkan data pada basis data ketika kembali terkoneksi.

Untuk mengembangkan penelitian ini sangat direkomendasikan penelitian lebih lanjut. Rekomendasi dari penulis untuk pengembangan pada penelitian selanjutnya mencakup:

- Penambahan riwayat pengisian data bencana sehingga dapat dilakukan tracking dari hari ke hari;
- Ditambahkannya deep assessment terkait kesehatan, pendidikan, dan kontroling pasca bencana;
- Pengembangan dashboard yang lebih mutakhir untuk penentuan keputusan, menggunakan referensi Sphere Project;
- Pengembangan sistem yang dibangun secara terintegrasi antarlembaga sosial yang menangani kebencanaan.

## DAFTAR PUSTAKA

- [1] Presiden Republik Indonesia, "Peraturan Presiden Republik Indonesia Nomor 8 Tahun 2008 tentang Badan Nasional Penanggulangan Bencana." Jakarta, 2008.
- [2] Badan Nasional Penanggulangan Bencana, "BNPB-Badan Nasional Penanggulangan Bencana," 2018. <https://bnpb.go.id/publikasi/info-bencana> (accessed Sep. 14, 2018).
- [3] M. Mulyadi, *Membuat Aplikasi Untuk Android*. Yogyakarta, Indonesia: Multimedia Center Publishing, 2010.